

- 1 Introduction
- 2 DPL Forms
- 3 Creating a Basic DPL Form
- 4 Overview
- 5 The General Property Page
- 6 Output Device Control
 - 6.1 Output Properties
- 7 Code Set Control
 - 7.1 Code Set Properties
- 8 Date Time Control
 - 8.1 Date Time Properties
 - 8.1.1 Prompt Options Area
 - 8.1.2 Command Line Format Area
 - 8.1.3 Date Options Area
 - 8.1.4 Time Options Area
 - 8.2 Calculate Default Properties
 - 8.3 Setting the Default Value for the Date Time Control
- 9 Text Edit Control
 - 9.1 Text Properties
 - 9.2 Numeric Input:
 - 9.3 Validation Properties
 - 9.4 Validation Scripting
- 10 List Box and Combo Box Controls
 - 10.1 Working with String Tables
 - 10.2 String Table Properties
 - 10.3 Working with Database Tables
 - 10.4 Database Table Properties
- 11 Master-Detail Linking
- 12 Audit Logs and Troubleshooting
- 13 Conclusion

Introduction

This tutorial introduces the fundamental concepts of using Discern Prompt Builder to create prompts for Discern Explorer Programs. The Discern Prompt Builder can be used within VisualExplorer.exe (VE) or DiscernVisualDeveloper.exe (DVDev). This tutorial is intended for programmers familiar with the Discern Explorer program language. Some familiarity with MS Windows controls is helpful, but not required.

Cerner believes the information in this document is accurate as of its publication date. While Cerner has made a conscientious effort to avoid errors, some may still exist. The information in this document is subject to change without notice, and should not be construed as a commitment by Cerner Corporation.

A Discern Explorer program can contain prompts to gather input from the user at execution time. This input is then used to flex the behavior of the program as it executes. You could create a program that asks the user to enter a last name and then have the program display information about people that had that last name. In the past Discern Explorer has placed some restrictions on prompts. They must appear at the beginning of the program. They can display a text string to the user and pass what the user enters to the program. Using just a prompt you cannot validate what the user enters and return to the prompt if they entered invalid information. You cannot display a list of values that the user can pick from or limit the user to only entering a certain set of values. The Discern Prompt Library (DPL) has been created to overcome the restrictions that exist on prompts. Using the DPL separates getting the user input from the execution of the program. The DPL allows you to create a form that is associated with a Discern Explorer program. When a Discern Explorer program is executed from a front-end application that uses the DPL, the application checks to see if a DPL form exists for the program. If the DPL form exists, it is displayed and the user completes the DPL form. The DPL form will contain a control that relates to each prompt in the program. The controls can be used to provide drop down lists of possible values, limit the input to only certain values, provide a drop down calendar that allows picking a date, validate that the data entered is in a specific format, look up names that begin with a certain string, and many other features. When the DPL form is completed, the program is executed and the values entered at each control are passed as parameters and assigned to the corresponding prompts in the program.

DPL Forms

A DPL form is collection of controls that form a single end-user conversation or dialog. DPL forms are used to get information from the user prior to executing a Discern Explorer program. Currently there are six types of controls that can be added to a DPL form:

- Output Device
- Text Edit
- Code Set
- Date and Time
- Combo Box
- List Box

Creating a Basic DPL Form

In order to see how these controls function, we will create a simple program and then create a DPL form that uses the six control types.

Using VisualExplorer.exe (VE) or DiscernVisualDeveloper.exe (DVDev) create a program named 1_INITIALS_PROMPT_TUTORIAL1. You should follow the naming convention that is established at your site when naming your program. Cerner recommends that you name all of your custom programs beginning with a number to prevent any conflicts with Cerner Millennium program names. The examples in this tutorial will use dpl_prompt_tutorial1 for the name of this program. Create a select statement in your program that selects the PERSON_ID, NAME_FULL_FORMATTED, NAME_LAST_KEY, SEX_CD, DISPLAY VALUE OF THE SEX_CD, and UPDT_DT_TM from the Person table. Format the updt_dt_tm to display both the date and time the record was last updated.

From the File menu, select **Include/Compile** in VE, or in DVDev from the Build menu, select **Include/Compile** to create the program object in the Discern Explorer object library. Correct any syntax errors and get a clean compile before proceeding. Use the **Prompt Builder** button on the toolbar or Tools > **Prompt Builder** to open the Discern Prompt Builder. If you are working in DVDev you will be asked to verify the name of the program that you want to create prompts for. In VE you will not be asked to verify the program name. Discern Prompt Builder opens and looks similar to the following example:

Discern Prompt Builder3 - [JG_DPL_PROMPT_TUTORIAL1:GROUP1*] New Form

The Prompt Display text appears in the prompt control's label.
Enter a string of text used with the prompt to instruct the user on what to enter. The text can be any length up to 100. You may define a mnemonic key by placing a & character in front of the mnemonic. To break a label enter \n where the new line should start.

Prompt Controls

OUTDEV

AddDeleteImport

☒ Auto arrange

GeneralOutput

*Prompt Display :&Output to File/Printer/MINE

*Prompt Name :OUTDEV

*Control Type :Output Device

Status Bar Text :Enter or select the printer or file name to send this report to.

*Prompt Type :String

☐ Advanced

Form ToolsProgram InfoSaveCancelHelp

Overview

When Discern Prompt Builder first opens it will check to see if the program has a DPL form already defined. If so then the DPL form is loaded. If the program does not have a DPL form defined Discern Prompt Builder checks the Discern Explorer object library for the program definition. If the program exists the prompts are read from the object library and a default form is created. The first prompt is assumed to be the output destination and an Output Device control is selected. The remaining prompts are set up to use a Text control. Note that the dialogs caption displays the programs name and group number, an asterisk (*) indicates that the form has been modified, and New Form is letting you know that the form is being created. You should get in the habit of checking the dialog caption for this text. For existing forms being modified the caption should not be displaying New Form or have an * until you change something. If a DPL form does not exist nor could a Discern Explorer program object be loaded then the builder will create a new form with one prompt (an Output Device).

Discern Prompt Builder is divided into four sections: Instructions, Prompt Control List, Control Property Pages, and Dialog Button Area. The Instruction section (located below the builders caption) displays instructions about the currently selected field. As you move from one field to the next the instructions will give you a short hint on what you are to do. For more detailed instructions press the Help button in the dialog button section. The Prompt Control List area (located on the left edge of the builder) shows all of the controls defined on the DPL form in the order, top to bottom, that the prompts will appear (tab order). This also determines what order is used when the prompts are passed to the Discern Explorer program. The Control Property Pages (the tabs located on the center-right) differ depending on the control selected in the Prompt Control List. But, the General property page will be present for every control.

The General Property Page

This property page defines the common properties for all six prompt control types.

*Prompt Display	The display text or label for the control. By default, the text is displayed to the left of the control at runtime. The advanced option "Label Left" can be used to display the text to the left of the control or above the control at runtime.
*Prompt Name	Assigns a name to the prompt that can then be used in the Discern Explorer program and for master-detail linking. It is recommended that you give each control a meaningful name. For example, OUTDEV for the Output device is better than prompt1. Prompt names must conform to the Discern Explorer identifier naming rules. Namely, start with a letter or underscore and can be followed by letters, digits, or underscores. If you name your first prompt OUTDEV then in your Discern Explorer program you can refer to value entered at the first prompt as \$1 or \$OUTDEV.

Page Version:
37

Page Identifier:
694454

Page Title:
Use Discern Prompt Builder

Page Effective Date:
Feb 1, 2013

© Cerner Corporation. All rights reserved. This document contains confidential and/or proprietary information belonging to Cerner Corporation and/or its related affiliates which may not be reproduced or transmitted in any form or by any means without the express written consent of Cerner.

2/34

*Control Type	Identifies the type of prompt control to use on this prompt. By default the first prompt is an Output Device and additional prompts are defaulted to Text.
Status Bar Text	With this field, you can provide a short helpful hint to your users at runtime. This text appears in Explorer Menus status line.
Prompt Type	Determines how DPL should format the parameter when it is passed to the report program. If the program is expecting a parameter wrapped between quotes, select String . If the program is expecting numeric values or calculations, select Expression .
Advanced	When checked shows the following properties.
Label Left	When not checked, the label is placed above the control. For auto-arranged forms the control is actually moved below the label or to the right of the label. For manually arranged forms the label is moved above the control or left of the control.
Control Position	Width and Height can be specified to change the controls default size. However, for non-auto arranged forms the left and top fields specify the top, left (x, y) coordinates of the control. The first release of DPL does not have a visual placement or preview. So if you want to arrange the controls yourself you have to do it by trial and error.
Prompt Only	When checked, the value entered in the control is not passed to the Discern Explorer program. This option can be used to pass a value from one control to a second control.

Selecting a control name in the Prompt Controls List brings the selected control into focus. The prompt control that has focus then has its property pages listed in the property page section. You can tab to this list and press the UP ARROW or DOWN ARROW to change the current focus. To change the order of the prompt controls on the DPL form, select the prompt control and click the **Up** or **Down** arrows on the left edge of the list.

Output Device Control

Because you are working on a new program Discern Prompt Builder created a new DPL form for your program that contains a single Output Device control name OUTDEV.

To see how the Output Device control functions by default, save the form using the **Save** button. Saving the form adds the following code to your source code file:

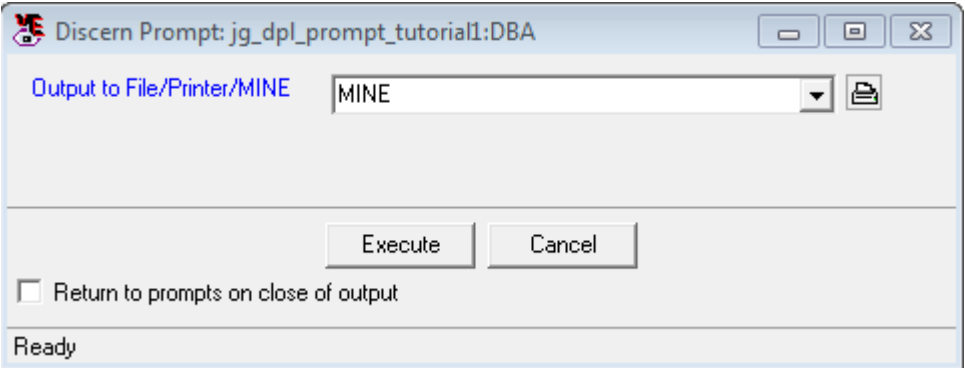
```
prompt
"Output to File/Printer/MINE" = "MINE"
with OUTDEV
```

If you are working in DVDev, this code is placed directly in the file you are editing. If you are working in VE, this code can be seen using the **View Query** tab in the Query Builder or from the Report menu, select **View Program**.

If you are working in DVDev, modify your select command to select into \$OUTDEV, include/compile your file, and execute your program using CTRL+R or from the Build menu, select **Run Prompt Program**.

If you are working in VE, execute your program using the **Run Query** button.

Executing a program that has a DPL form associated with it, from any application that uses DPL, causes the form to be displayed. The DPL form associated with your program should be similar to the following:



This DPL form contains a single output device control. This control consists of a combo box and a printer browser button. At this point, the only item listed in the drop-down list is the default value MINE. The user can enter any value they want in the text box or use the printer browser button. Clicking the printer browse button displays the following dialog box:

Output Device

☐ Printer
☐ File
☒ Display

OK Cancel

By selecting the Printer radio button the user can get a list of all of the print queues for the host system.

Output Device

☒ Printer
☐ File
☐ Display

Output To:

1hermontest2G
 1PrintTekle1
 1TestTekle1
 2tekle6G
 a_dev_name_be

So at this point the user can send the output of the program to any device or file.

Using the Output Device control you can restrict the user to only selecting a printer from a small subset of all the available printers, sending the output to a file that you specify, or having the output displayed on the screen. To do this, you need to make some changes to the properties on the **Output** tab.

Click **Cancel** to close the DPL Form without executing the program.

Use the Prompt Builder icon on the toolbar or from the Tools menu select **Prompt Builder** to open the Discern Prompt Builder.

Click the **Output** tab to display the Output properties page. Your output properties page should look similar to the following:

Discern Prompt Builder3 - [JG_DPL_PROMPT_TUTORIAL1:DBA*] New Form

Check the Show Only box to create a sub-set of printers that the user may choose from. When this box is checked the Available and Selected boxes below will be enabled. The first device in the Select list will be the default device. When unchecked all printers on the system will be listed at runtime.

☐ Show only printers listed below.
☐ Hide browser button.
☐ Disallow freetext devices.

Add Output File

Device List

Available Devices : { Empty } Load

Default Device: MINE

> >>

Form Tools Program Info Save Cancel Help

Output Properties

Show only printers listed below. If this option is checked, the output control only shows the devices that are explicitly listed in the **Selected Devices** list in the drop-down list.

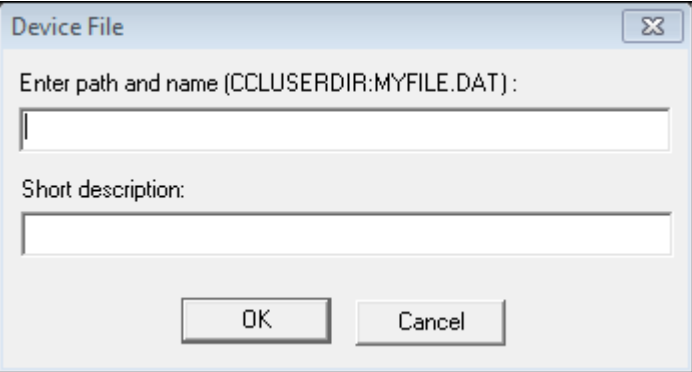
Hide browser button. If this is checked, the printer icon next to the output device combo box is not displayed. The browser button allows the user to search for printers not defined in the combo box list box or to create a data file for the report.

Disallow free text devices. If this is checked, the user is not allowed to type in the edit box and is forced to select a printer device from the combo list or if available, the browser.

The first output device listed in the Selected Devices list is the default output device. If **Show only printers listed below** is not checked, then the default is MINE. Use the up-down control to the right of the **Select Devices** list to reorder the devices.

The **Load** button is used to load the Available Devices list with all available printers defined on the host. Be careful not to select printers already selected. The output property page does not remove duplicates in the list.

The **Add Output File** button allows you to add a predefined file as an output device. This button displays the following dialog box.



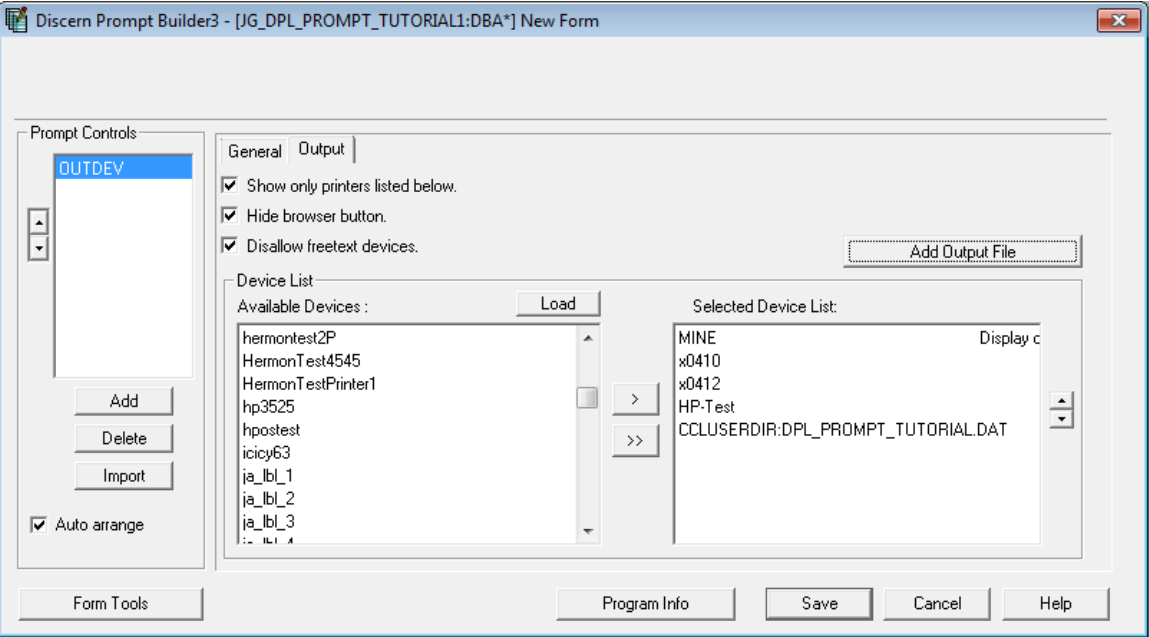
The 'Device File' dialog box has a title bar with a close button. It contains two text input fields: 'Enter path and name (CCLUSERDIR:MYFILE.DAT) :' and 'Short description:'. At the bottom are 'OK' and 'Cancel' buttons.

Enter the file name in the first field. The file name must be compatible with the Discern Explorer SELECT INTO command. The Short description is currently displayed on the Properties page but is not shown to the user at runtime. This may change in future releases.

Select the **Show only printers listed below**, **Hide browser button**, and **Disallow free text devices** check boxes.

- Click **Load** to populate the Available Devices: list.
- Highlight MINE and click the arrow button to move MINE to the Selected Devices: list.
- Highlight several printers in the Available Devices list and click the arrow button to move them to the Selected Devices: list.

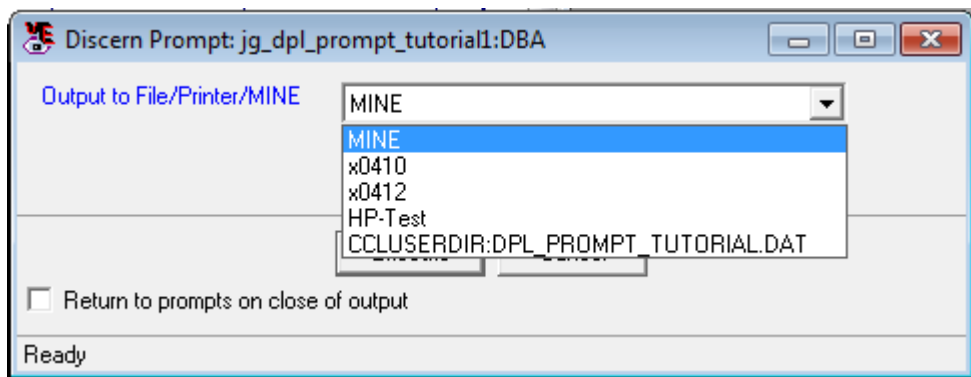
Click **Add Output File**. Use CCLUSERDIR:DPL_PROMPT_TUTORIAL1.DAT as the path and file name and Output file for tutorial program as the short description and click **OK**. Your Discern Prompt Builder dialog should look similar to the following example:



The 'Discern Prompt Builder3' window shows a 'Prompt Controls' sidebar with 'OUTDEV' selected. The main area has 'General' and 'Output' tabs. Under 'General', three checkboxes are checked: 'Show only printers listed below.', 'Hide browser button.', and 'Disallow freetext devices.'. An 'Add Output File' button is on the right. Below, the 'Device List' section shows an 'Available Devices' list with items like 'hermontest2P', 'HermontTest4545', and 'hp3525'. A 'Load' button is next to it. To the right is a 'Selected Device List' containing 'MINE', 'x0410', 'x0412', 'HP-Test', and 'CCLUSERDIR:DPL_PROMPT_TUTORIAL.DAT'. Navigation arrows (> and >>) are between the lists. At the bottom are 'Form Tools', 'Program Info', 'Save', 'Cancel', and 'Help' buttons.

Click **Save** to save the changes to your form.

Execute your program using CTRL+R or from the Build menu, select **Run Prompt Program** in DVDev or click **Run Query** in VE. When you select the drop-down list on your form it should look similar to the following:



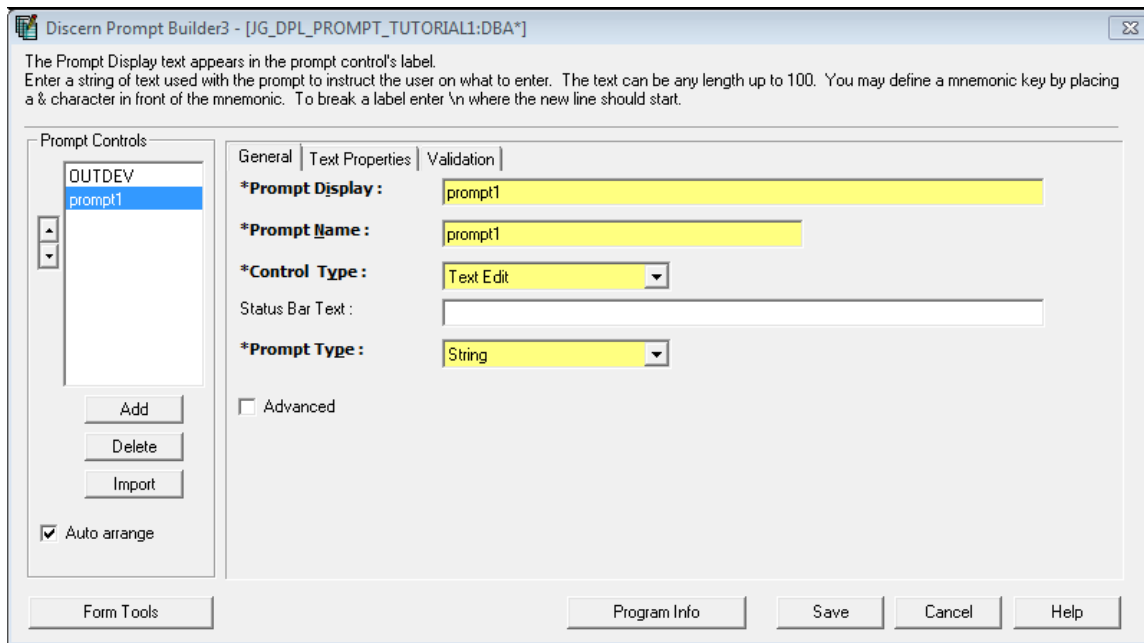
Now you can only choose one of the options on the list as the output device when your program is executed. The printer browser button is no longer shown and typing a free text value is not allowed.

Code Set Control

At run time, the Code Set control allows the user pick one or more code values from a list of display values and descriptions. When a display value or description is selected from the list, the control passes the actual code value to the Discern Explorer program. The Code Set control type can be used to create either a combo box or a list box control on the form. Either of these controls force the user to select a value from the list but the list box allows multiple values to be selected.

Lets assume that we want to give the user the ability to only select people of a specific gender when they execute our program. To accomplish this, we can add a code set control to the form that is associated with the program.

Open Discern Prompt Builder and click **Add** on the lower left side of the Prompt Control list to add a second prompt to the form. Your Discern Prompt Builder dialog should look similar to the following example:



Enter **Select the Gender** in the Prompt Display field

Change the Prompt Name to Gender

Select **Code Set** as the control type

Because the control will pass a code value number, verify that the Prompt Type changed to Expression. Your Discern Prompt Builder dialog should look similar to the following example:

Click the **Code Set** tab to begin setting the properties that are specific to the Code Set control.

Code Set Properties

Code Set. This is the numeric value of the code set that is displayed in the controls list. This number can be seen in several places:

- From the Query Builder within VE or DVDev click **Glossary** on the **Fields** tab and the code set is displayed for all fields that end in **_CD**.
- From the Query Builder within VE or DVDev on the **Fields** tab right-click on a field name in the Fields list. The code set is displayed as one of the properties of a field that ends in **_CD**
- From the **Tables/Fields** tab in DVDev, right-click on a field name. The code set is displayed as one of the properties of a field that ends in **_CD**

Find Button. Used to look up Code Sets

Combo Box. Creates a combo box on the form. A combo box displays one value with an attached drop-down list.

List Box. Creates a list box on the form. A list box displays a list of values.

Multiple Values. This option is only valid in conjunction with a list box. Allows the user to select multiple values from the list to be passed to the Discern Explorer program. If this option is selected, the codes that the user selects are passed to the Discern Explorer program using the Value() function. When used in a qualification, the Value() function causes an equality statement to be treated as an IN statement. For example in the qualification:

```
Where p.sex_cd = $gender
```

If \$gender is equal to value(123.0, 124.0, 125.0) then the qualification would be treated as

```
Where p.sex_cd in (123.0, 124.0, 125.0)
```

Set Default. Activates the Default Code Values list.

Default Code Values. Sets the default value that will be active in the control at run time.

Because we want to give the user the ability to only select people with a specific gender we need to provide them with a list of values that can be compared to the Sex_CD field on the Person table. The Sex_CD field uses code set 57.

Enter **57** as the Code Set.

Select **Combo box** in the Display Type area.

Select the **Set Default** check box.

Highlight the Unknown sex code in the Default Code Values list. Your Discern Prompt Builder dialog should look similar to the following example:

Select the default code value for the control.

Prompt Controls

OUTDEV
Gender

Add
Delete
Import

☒ Auto arrange

Form Tools

General Code Set

Code Set
57 Find

Display Type
☐ List box ☐ Multiple Values
☒ Combo box
☐ Include Any(*) Define Any(*)

☒ Set Default

Default Code Value

Female	: Sex is female
Indeterminate	: Indeterminate
Male	: Sex is male
Unknown123	: Sex is unknown
Unspecific	: Unspecified

Program Info Save Cancel Help

Click **Save**. Saving the form modifies the Prompt clause in your source code file. It should look similar to the following example:

```
prompt
"Output to File/Printer/MINE" = "MINE"
, "Select the Gender" = 197.00
with OUTDEV, Gender
```

If you are working in DVDev, this code is placed directly in the file you are editing. If you are working in VE, this code can be seen using the **View Query** tab in the Query Builder or using the **View Program** option on the Report menu.

The default value for your second prompt will most likely not be 197.0. Instead it is the code value number that represents the Unknown sex code in your environment.

Now that the control is set up, you need to add a qualification that uses the value that is selected at run time to only select people with the selected gender.

Add the qualification **WHERE P.SEX_CD = \$Gender** to your select statement.

If you are working in DVDev, include/compile your file, and execute your program using CTRL+R or from the Build menu, select **Run Prompt Program**.

If you are working in VE, click **Run Query** to execute your program.

When the program is executed, the DPL form should look similar to the following example:

Discern Prompt: jg_dpl_prompt_tutorial1:DBA

Output to File/Printer/MINE MINE

Select the Gender Unknown123

Execute Cancel

☐ Return to prompts on close of output

Ready

Use the drop-down list on the Select the Gender combo box to select people with a specific gender. Execute your program and verify that only people with the selected gender are displayed.

If instead of only having the ability to select one gender, the user needs the ability to select one or more genders, you can modify the Gender control on the form to provide this functionality.

Open Discern Prompt Builder and select the **Gender** control on the Prompt Controls list.

Click the **Code Set** tab to display the code set properties.

Select **List Box** and check the Multiple Values check box in the Display Type area. Your Discern Prompt Builder dialog should look similar to the following example:

Click **Save** to save the changes to the form.

If you are working in DVDev, execute your program using CTRL+R or from the Build menu, select **Run Prompt Program**.

If you are working in VE, click **Run Query** to execute your program. When the program is executed, the DPL form should look similar to the following example:

Check a couple of the genders in the Select the Gender list box. Execute your program and verify that only people with the selected genders are displayed. The check boxes in the above display are the result of using the Multiple Values check box on the Code Set properties page. If this option is not selected, the check boxes are not displayed. The code value for the item that is highlighted in the list is passed to the Discern Explorer program at run time.

Date Time Control

At run time, the Date Time control uses the Cerner standard Date Time control, which allows the user to enter a date or time, increment a date or time using spin buttons, or pick a date from calendar. The date and time values are then passed to the Discern Explorer program where they can be converted to a date time

value using the Cnvtdatetime() function or used by other date and time functions.
Lets assume that we want to give the user the ability to only select records from the Person table that have been updated on or after a specific date. To accomplish this we can add a date time control to the form that is associated with the program.

Open Discern Prompt Builder and click the **Add** button on the lower left side of the Prompt Control list to add a third prompt to the form.

For the Prompt Display, use Select the Beginning Date.

For the Prompt Name, enter **Updt**.

For the Control Type, select **Date Time**.

For the Prompt Type, select **String**. Your Discern Prompt Builder dialog should look similar to the following example:

Select the **Date / Time** tab to review the date time properties.

Date Time Properties

The Date Time Properties are divided into four areas: Prompt Options, Command Line Format, Date Options, Time Options.

Prompt Options Area

- Date Only.** Only prompts the user for a date.
- Time Only.** Only prompts the user for a time.
- Date and Time.** Prompts the user for both a date and time.

Command Line Format Area

Command Line Format. Controls how the date and time are formatted when passed to the Discern Explorer program. The available formats flex based on what is selected as the prompt type on the **General** tab and the prompt option. The Command Line Format does not affect the way the date or time is displayed in the form at run time. The Command Line Format only affects how the date or time is passed to the Discern Explorer program. The format for how the date or time is displayed at run time is controlled by the PC locale settings.

Date Options Area

- ComboCalendar.** When selected, the ComboCalendar check box creates a drop-down button on the form that allows the user to pick a date from a calendar.
- Spin Button.** When selected, the Spin Button check box creates up and down arrow buttons next to the date on the form that allow the date to be incremented or decremented.
- Fill Style.** Determines what character is used to fill in the date control when no date has been specified.

+MM/DD -MM/DD The Plus/Minus (Months/Days) Only applies in Discern Prompt Builder Version 1. It sets the default date that is displayed in the control by adding or subtracting the number of days and months from the current date. For example, +01/00 sets the default to one month in the future. -01/15 sets the default to one month and fifteen days in the past.

Time Options Area

- Allow Relative Input.** Select this option to allow relative time entries using the UP and DOWN arrow keys. Selecting this option activates the Spin Box option and Increment box.
- Spin Button.** Select this option to display up and down arrow buttons next to the time prompt control. The spin box allows the user to change the time based on the increments entered in the Increment box.
- Increment box.** Enter the increment or decrement value that is used by the spin box when selecting a time. The time advances or retracts to the next multiple (for example, if the increment is 60, the time advances to the top of the hour, and if the increment is 30, the time advances to the top or bottom of the hour, whichever is first).
- Allow N Key Input.** Select this option to allow the user to set the time to the current time by pressing N.

+HH/MM -HH/MM The Plus/Minus (Hours:Minutes) Is only applicable in Discern Prompt Builder Version 1. It allows you to enter the number of hours and minutes to add or subtract from the current time (for example, 05:30 is 5 hours and 30 minutes ahead of the current time and -05:30 is 5 hours and 30 minutes prior to the current time).

Calculate Default Properties

Note
The **Calculate Default** tab was added in version 2 of Prompt Builder.

The **Calculate Default** tab is used to set the default value of the date time control. By default, the date time control displays the current date and time.

The screenshot shows the 'Discern Prompt Builder3 - [BGR_DPL_PROMPT_TUTORIAL1:DBA]' window. The 'Calculate Default' tab is selected. The 'Anchor date & time' section has 'Current date & time' checked. The 'Day of:' section has 'Use anchor date & time' selected. The 'Offset date & time' section is empty. The 'Time' section has 'Begin time of day' selected. The 'Link input from:' dropdown is empty. The 'Test' button is visible. The 'Form Tools' section on the left includes 'Add', 'Delete', 'Import', and 'Auto arrange' (checked). The bottom of the window has 'Program Info', 'Save', 'Cancel', and 'Help' buttons.

Anchor date & time - Sets the starting point for calculating the date and time values displayed in the control.

- **Current date & time** - This option is selected by default. Deslecting allows you to specify a different date and time value.
- **Year/Month/Day Hour:Minute** - Can be used to specify an absolute or relative date time. For example, if you enter **2013** for the year, **01** for the month, **30** for the day, **15** for the hour and **45** for the minute, the control displays 30-Jan-2013 at 3:45 p.m. as the default. Entering **-1** for the year, **-1** for the month and **15** for the day would calculate the date time displayed in the control by subtracting one from the current year, one from the current month and then choosing the 15th day of that month as the default. Entering **+4** for the hour would cause the control to default to four hours after the current time.
- **Auto update default time** - This option causes the time displayed in the control to automatically update when the prompt form is displayed at runtime.
- **Link input from:-** Sets the anchor date and time equal to the date and time from another date time control on the prompt form. The drop-down contains the names of all the other date time controls on the form.

Day of: - Allows you to set the default by adjusting the anchor date and time to the beginning or end date and time of a period. Use the drop-down to select the period. For example, **Use anchor date & time** does no further adjustment. Selecting **Month** on the drop-down allows you to adjust the default date to be either the first or last day of the month of the anchor date and time. You can also adjust to the beginning, ending, or current time on that day.

Offset date & time - Allows you to adjust either the anchor date and time or the date and time that resulted from the Day of: adjustment.

- Year/Month/Day/ Hour:Minute - Can be used to specify a relative date and time. For example entering **-1** for the year and **-1** for the month subtracts one year and one month from the anchor date and time or the date time that resulted from the Day of: adjustment. You can enter absolute values in the Year/Month/Day Hour:Minute controls, but doing so sets the default to that absolute value. Generally, if you want to default to an absolute date time, you would set it in the Anchor date & time area.

Test button - Allows you to test the behavior of the settings to see the default that is calculated without needing to save the prompt form and execute the program.

Setting the Default Value for the Date Time Control

In our program we want the user to be able to qualify records on the person table that have been added or updated on or after a specified date. On the **Date / Time** tab, select **Date Only** as the Prompt Option. The Command Line Format should default to DD-MMM-YYYY. Verify that this is what you have. If you have a different command line format, verify that the Prompt Type on the **General** tab is set to String and then change the command line format as needed.



Release Consideration

If you are using version 1 of Discern Prompt Builder, complete the following steps.

Enter **-01/00** in the +MM/DD -MM/DD field. Your Discern Prompt Builder dialog should look similar to the following example:

Click **Save** to save your form.



Release Consideration

If you are using version 2 or higher of Discern Prompt Builder, complete the following steps.

On the **Calculate Default** tab, deselect the option for Current date & time. Enter **-1** in the Month. These settings cause the default date displayed in the Updt prompt to be one month in the past. Your Discern Prompt Builder dialog should look similar to the following example:

Click **Save** to save your form.



Release Consideration

Complete the following regardless of the version of Discern Prompt Builder you are using.

We now need to modify the program to qualify records from the Person table where the UPDT_DT_TM is on or after the date that the user enters in the Date Time control. Because the Prompt Type is String and the Command Line Format is DD-MMM-YYYY, at run time the date the user picks is passed to our program as a string in DD-MMM-YYYY format. This string is assigned to the \$Updt parameter. We need to convert the string into a datetime value that we can compare to the UPDT_DT_TM field on the person table. The CNVTDATETIME() function takes a string in DD-MMM-YYYY format, assumes zero as the time, and creates a datetime value that represents the earliest possible time on that date. So in our program all we need to do is pass \$Updt to the CNVTDATETIME() function and compare the result to the UPDT_DT_TM field on the Person table.

Currently your qualifications should be

```
WHERE P.SEX_CD = $Gender
```

Add

```
AND P.UPDT_DT_TM >= CNVTDATETIME($UPDT)
```

to your qualifications. Your qualifications should now be

```
WHERE P.SEX_CD = $Gender AND
P.UPDT_DT_TM >= CNVTDATETIME($UPDT)
```

If you are working in DVDev, include/compile your file, and execute your program using CTRL+R or from the Build menu, select **Run Prompt Program**.

If you are working in VE, click **Run Query** to execute your program.

When the program is executed, the DPL form should look similar to the following example:

Use the spin buttons and the drop-down calendar to select different dates and execute the program. Verify that records returned by the query were updated on or after the chosen dates.

Write down the update dates and times from a couple of records that are displayed in your output.

Suppose we want to change our program to only qualify records from the Person table that were updated on the date entered at the prompt. We can accomplish this by just changing our qualifications. There is no need to change the DPL form.

Change your qualifications from:

```
WHERE P.SEX_CD = $Gender AND
P.UPDT_DT_TM >= CNVTDATETIME($UPDT)
```

To:

```
WHERE P.SEX_CD = $Gender AND
P.UPDT_DT_TM BETWEEN CNVTDATETIME($UPDT) AND
CNVTDATETIME(CONCAT($UPDT, CHAR(32), "23:59:59"))
```

CNVTDATETIME() accepts a string in the format DD-MMM-YYYY HH:MM:SS and returns the datetime value that represents that date at that time. There must be one and only one space between the date and time. CHAR(32) returns a single space. In this example the CONCAT() function takes the string in DD-MMM-YYYY that is passed from the DPL form, the space returned from the CHAR(32), and the hard coded string 23:59:59, and creates a string that is in the DD-MM-YYYY HH:MM:SS format that the CNVTDATETIME() function recognizes.

If you are working in DVDev, include/compile your file, and execute your program using CTRL+R or from the Build menu, select **Run Prompt Program**.

If you are working in VE, click **Run Query** to execute your program.

Choose a date that you know records have been updated on. Verify that only records that were updated on the date you entered are displayed when the program is executed.

Suppose we wanted to qualify on records that were updated between two specific dates and times. We can accomplish this by adding a second date time prompt to the DPL form.

Open Discern Prompt Builder and Delete the Updt prompt control by highlighting it and clicking **Delete**.

Click **Add** to add a new prompt control to the form.

For the Prompt Display use **Select the Beginning Date and Time**.

For the Prompt Name use **Beg_dt_tm**.

For the Control Type select **Date Time**.

For the Prompt Type select **String**.

On the Date / Time properties page select **Date and Time** as the Prompt Option.

Verify that the Command Line Format is DD-MMM-YYYY HH:MM:SS.

Enter -00/01 in the +MM/DD -MM/DD field. Your Prompt Builder dialog should look similar to the following example:

Discern Prompt Builder :[DPL_PROMPT_TUTORIAL1:DBA*]

The number of months and days to add or subtract from the current date. (Example: 5/12 is 5 months 12 days from the current date).

Prompt Controls

OUTDEV

Gender

End_dt_tm

Add

Delete

☒ Auto arrange

GeneralDate / Time

Prompt Options

☐ Date Only

☐ Time Only

☒ Date and Time

Command Line Format:

DD-MMM-YYYY HH:MM:SS

Date Options

☒ ComboCalendar

Fill Style

Blanks

☒ Spin Button

+MM/DD

-00/01

Time Options

☒ Allow Relative Input

☒ Spin Button

Increment :

1

☒ Allow N key Input

+HH:MM

- HH:MM

00:00

Program Info

Save

Cancel

Help

Click **Add** to add a new prompt control to the form.

For the Prompt Display, enter **Select the Ending Date and Time**.

For the Prompt Name, enter **End_dt_tm**.

For the Control Type select **Date Time**.

For the Prompt Type select **String**.

On the **Date / Time** properties tab, select **Date and Time** as the Prompt Option.

Verify that the Command Line Format is DD-MMM-YYYY HH:MM:SS. Your Prompt Builder dialog should look similar to the following example:

Page Version: 37	Page Identifier: 694454	Page Title: Use Discern Prompt Builder	Page Effective Date: Feb 1, 2013
---------------------	----------------------------	---	-------------------------------------

© Cerner Corporation. All rights reserved. This document contains confidential and/or proprietary information belonging to Cerner Corporation and/or its related affiliates which may not be reproduced or transmitted in any form or by any means without the express written consent of Cerner.

15/34

Discern Prompt Builder :[DPL_PROMPT_TUTORIAL1:DBA*]

Create a prompt that will accept date and time elements.

Prompt Controls
OUTDEV
Gender
Beg_dt_tm
End_dt_tm
Add
Delete
☒ Auto arrange

General **Date / Time**
Prompt Options
☐ Date Only ☐ Time Only ☒ Date and Time
Command Line Format: DD-MMM-YYYY HH:MM:SS
Date Options
☒ ComboCalendar Fill Style: Blanks
☒ Spin Button +MM/DD: 00/00
 -MM/DD:
Time Options
☒ Allow Relative Input ☒ Spin Button Increment: 1
☒ Allow N key Input +HH:MM: 00:00
 -HH:MM:
Program Info Save Cancel Help

Click **Save** to save your form.

Change your qualifications from:

```
WHERE P.SEX_CD = $Gender AND
      P.UPDT_DT_TM BETWEEN CNVDTATETIME($UPDT) AND
      CNVDTATETIME( CONCAT( $UPDT, CHAR( 32 ), " 23:59:59" ) )
```

To:

```
WHERE P.SEX_CD = $Gender AND
      P.UPDT_DT_TM BETWEEN CNVDTATETIME($BEG_DT_TM) AND
      CNVDTATETIME( $END_DT_TM)
```

If you are working in DVDev, include/compile your file, and execute your program using CTRL+R or from the Build menu, select **Run Prompt Program**.

If you are working in VE, click **Run Query** to execute your program.

Enter a beginning date and time and an ending date and time. Verify that only records that were updated between the dates and times you entered are displayed when the program is executed.

Remember the Between operator is an inclusive operator, so records that were updated on your beginning and ending dates and times also qualify.

Write down the names from the Name_Last_key field from several records that qualify. We will use these names to test the next control we add to our DPL form.

All of our date time examples have used string as the prompt type and DD-MMM-YYYY or DD-MMM-YYYY HH:MM:SS as the command line format. Depending on what you are going to do in the Discern Explorer program, you might want to use expression as the prompt type and a different command line format. For example, many programs that were written before the introduction of Discern Prompt Builder prompted the user to enter a date in MMDDYY format. These programs then used something like:

```
where p.updt_dt_tm >= cnvdatetime(cnvdate($2),0)
```

in a qualification. If you were going to add a prompt form to a program that used this type of logic, you would want to use expression as the prompt type, select **Date Only** as the Prompt Option and use MMDDYY as the command line format.

Text Edit Control

Text edit controls can be used to create free text and numeric prompts. The control supports forcing text to upper case or lower case, accepting only alphabetic, numeric, alpha-numeric input or a definable character masked input. The text control can also validate what the user entered in the field and provide a lookup list for the user to select from.

Lets assume that we want to give the user the ability to enter a last name and only select records from the Person table that match this name. To accomplish this we can add a text edit control to the DPL form that is associated with the program.

Open Discern Prompt Builder and click **Add** on the lower left side of the Prompt Control list to add a prompt to the form.

For the Prompt Display use **Enter a last name**.

For the Prompt Name enter **Name**.

For the Control Type use the default Text Edit.

For the Prompt Type select **String**. Your Prompt Builder dialog should look similar to the following example:

The Text Edit control has two property tabs: **Text Properties** and **Validation**.

Text Properties

Text Properties define the runtime behavior of the edit box.

Discern Prompt Builder3 - [JG_DPL_PROMPT_TUTORIAL1:DBA*]

Enter the value the prompt should have if the user does not enter a response. This will be the initial value of the prompt when the program first runs.

Prompt Controls

OUTDEV
Gender
Beg_dt_tm
End_dt_tm
Name

Add
Delete
Import

☒ Auto arrange

General | Text Properties | Validation

Default Value :

Maximum Characters :

Character Case

☒ Mixed
☐ Upper
☐ Lower

Character Validation

☒ None ☐ Numeric only
☐ Alphanumeric
☐ Alphabetic only

☐ Display (*) for each character typed

Display Templates
None

Display Mask

Form Tools

Program Info Save Cancel Help

Default Value. Defines the default text for the control. Leave empty for no default. Be careful of extraneous spaces! A space is interpreted as a default value and is hard to see in the default box.

Maximum Characters. If 0 then the edit box allows any length of text (although Discern Explorer only accepts up to 130 characters). Otherwise, the number entered in the field limits the number of characters the user may type into the prompt control.

Character Case. Defines what the control should do when the user types a letter. If Mixed, then the letter is accepted as is. If Upper, the letter is converted to upper case. If Lower, the letter is converted to lower case.

Character Validation. Determines what type of character to accept. None accepts any character the user types, Alphanumeric accepts letters and numbers only. Special characters are ignored. Alphabetic only accepts letters. Numeric only accepts digits. For this control, the underscore is considered a letter.

Display(y) *.* Mask the users input with asterisks in the display. The value of the text is whatever the user typed.

Display Templates. Currently only None and User Defined templates are supported. Templates allow the edit box to use a Display Mask in the edit box window. The Display Mask allows you to define a picture of what the input must look like. Prompt Builder supports the following characters in the Display Mask picture.

'9"0"A"U"L"D"N"P"X"*

9	Numeric.
9(n)	Repeat the 9 mask for the number of times indicated by (n) .
A	Alpha (uppercase and lowercase letters).
A(n)	Repeat the A mask for the number of times indicated by (n) .
D	Display. Use the corresponding default character in the default clause and skip over for accept.
N	Numeric or punctuation (0-9, period, or minus sign).
N(n)	Repeat the N mask for the number of times indicated by (n) .
P	Any printable ASCII character.
P(n)	Repeat the P mask for the number of times indicated by (n) .
X	Alphanumeric (letters or numbers).
0	Zero filled numeric. Example : The mask 00009 would display 4 as 00004. As a user types into a zero fill area, the digits are filled in at the last 0 or 9 digit. Deleting removes from the end of a sequence of 0s or 9s. For example, if the format is 0009 and the user enters 12 the field shows as 0012 and if they then press DELETE, it will show as 0001.
*	Hide character typed with *. Same as the <i>Display (*)</i> for each character typed option but character by character. This allows you to hide portions of the characters being typed.
U	Converts the character typed to upper case. Both the displayed character and the corresponding character in the input are upper case.
L	Converts the character typed to lower case. Both the displayed character and the corresponding character in the input are lower case.
(x)	Repeat operator. Can be applied to any mask character. Example: X(5) is the same as XXXXX.

x	Backslash. The character following the backslash character is placed "as is" in the display and input for the control. For example if the display mask should have the " character in it the mask " ** would be translated as a literal " instead of the hide character mask. To include a backslash in the display mask you would use two backslashes "\\\\".
---	--

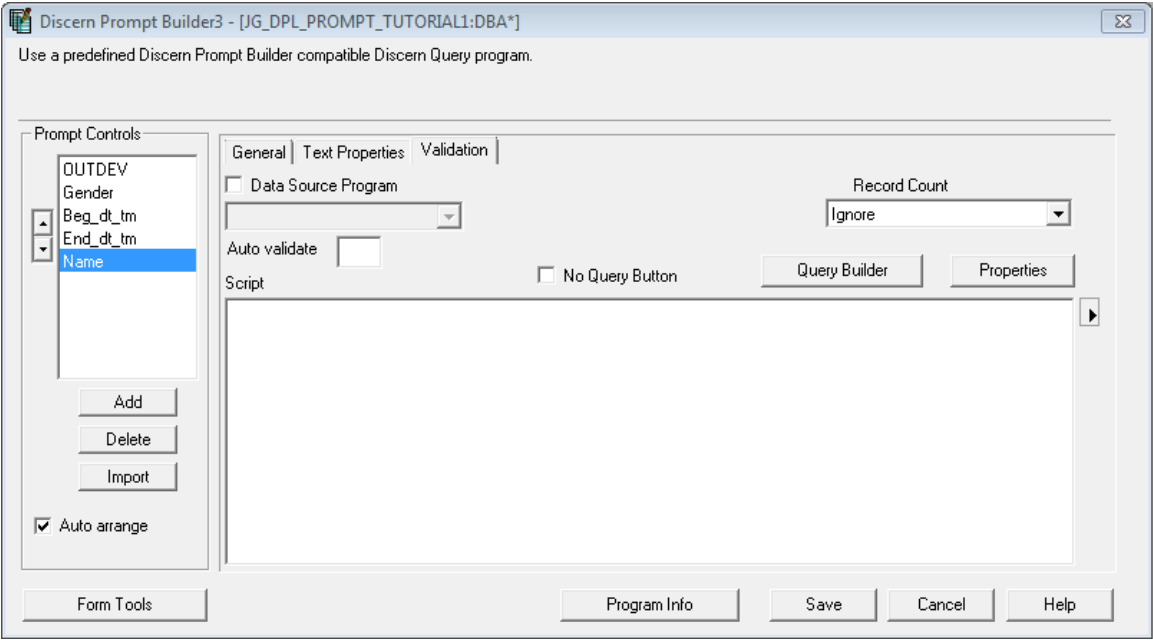
Any character that is not a mask character or the escape character is interpreted as a literal character mask. Literal characters in the input/display are skipped over and are not editable.

Numeric Input:

When the Character Validation is Numeric Only, additional fields appear in the property page. This allows the DPL programmer to input a lower and upper limit. If one of the ranges is not needed, leave the field blank. For example, to create a control with no upper limit but a lower limit of 5, leave the upper limit blank and enter the lower limit of 5.

Validation Properties

This property page allows the user to define a select statement for lookup and record validation. Validation occurs when the user enters text into the control and attempts to TAB or click to another control.



- Data Source Program:** When checked, the user may select from a list of Discern Explorer programs using the combo box. Creating Data Source Programs is outside the scope of this tutorial.
- Record Count:** Defines what type of validation, if any, to perform. Ignore performs no action. Match none sets an error if the SELECT returns any records. Match one and only one sets the error flag if the SELECT returns no records or more than one record. And Match one or more sets the error flag if the SELECT fails to return any records. When the error flag is set, the controls background color turns red with white text. An error message is also sent to the status text line.
- Script:** Contains a Discern Explorer SELECT statement that is executed when the user attempts to TAB or click to another control. The reserved special variable \$CURACCEPT can be used to pass the value that the user entered in the control to the SELECT statement. SELECT commands placed in the Script area can use most of the Discern Explorer SELECT clauses. However, they cannot have the SELECT INTO clause or use a report writer section (HEAD REPORT/DETAIL and so on).
- No Query Button** check box: Prevents the query lookup button from displaying at runtime.
- Query Builder** button: Opens the query builder, which can be used to create a select statement in the script area. If a select statement exists in the Script area, it is imported into the query builder. When the query builder is closed, the select statement is written to the script area.
- Properties** button: The **Properties** button opens the Column Properties dialog box. Columns properties describe how a dataset field is handled in the control at runtime

Validation Scripting

Validation is performed by the Text control when the control loses focus (either the user tabs or clicks out of the control). The current value in the edit box is passed to the validation script as \$CURACCEPT and the select is executed. When the select completes the control compares the number of rows that qualified to the Record Count criteria. If the record count criteria does not match the number of rows that qualified then the value that is in the control is invalid. When the value is invalid, the controls background is painted red and an error sound wave is played. The focus is immediately switched back to the Text control.

When the user enters a last name at the prompt, we want to validate that there are records on the Person table with that name.

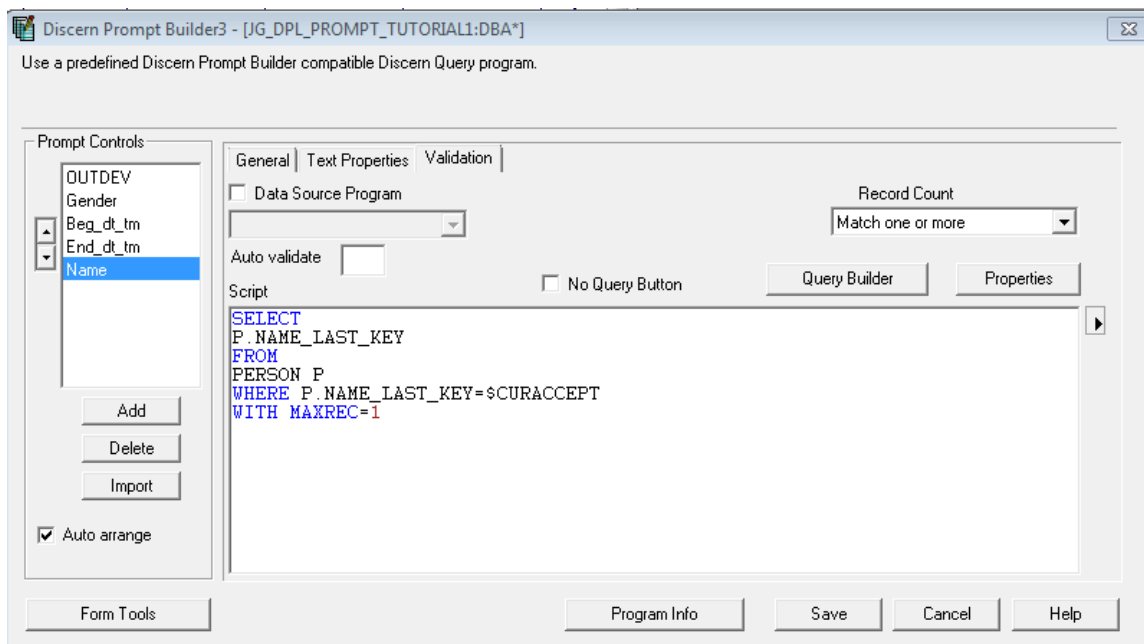
Click the **Text Properties** tab and select Upper in the Character Case area to have the control convert what the user enters to upper case.

On the **Validation** tab, set the Record Count to Match One or More.

Use the Query Builder to create the following query or type the following query in the Script area:

```
SELECT
P.NAME_LAST_KEY
FROM
PERSON P
WHERE P.NAME_LAST_KEY = $CURACCEPT
WITH MAXREC = 1
```

Your Prompt Builder dialog should look similar to the following example:



Click **Save** to save your form.

Now you need to modify the qualifications in your program to use the name that the user enters at the prompt.

Change your qualifications from:

```
WHERE P.SEX_CD = $Gender AND
P.UPDT_DT_TM BETWEEN CNVTDATETIME($BEG_DT_TM) AND
CNVTDATETIME($END_DT_TM)
```

To:

```
WHERE P.SEX_CD = $Gender AND
P.UPDT_DT_TM BETWEEN CNVTDATETIME($BEG_DT_TM) AND
CNVTDATETIME($END_DT_TM) AND
P.NAME_LAST_KEY = $Name
```

If you are working in DVDev, include/compile your file, and execute your program using CTRL+R or from the Build menu, select **Run Prompt Program**.

If you are working in VE, click **Run Query** to execute your program.

Enter one of the names you wrote down earlier in the Enter Last Name control and execute your program. Verify that only records where the NAME_LAST_KEY value matches the name you are entering at the prompt are qualifying.

To test the validation feature of the Text Edit control, enter a last name that you know does not exist in the NAME_LAST_KEY field on the Person table and change the Beginning date to a different date. The Text Edit control that you are using to get the last name loses focus when you change the date. At that point the background color of the control should change to red to indicate that the value you entered is invalid.

List Box and Combo Box Controls

The List Box and the Combo Box controls allow the user to pick a value from a pre-defined list. The list can be created by entering values into a string table or selecting values from the database. We will refer to the list of values as a dataset. The dataset contains rows and columns of data. One of the columns is designated as the Key column. At runtime, the user selects at least one of the rows from the dataset. The value in the key column of the selected row or rows is passed to the Discern Explorer program that is associated with the DPL form.

Working with String Tables

Combo Boxes show one column of a dataset. The first visible column in the dataset is displayed. To create a string table for a Combo box, select **From String Table** on the **Data Source** property tab.

List Boxes support up to ten columns and fifty rows. All the columns may be visible, but only one can be designated as the key. As before, to create a string table select **From String Table** on the **Data Source** tab.

We will create a new program to use for working with the list and combo box controls.

Using VisualExplorer.exe (VE) or DiscernVisualDeveloper.exe (DVDev) create a program named 1_initials_prompt_tutorial2. You should follow the naming convention that is established at your site when naming your program. Cerner recommends that you name all of your custom programs beginning with a number to prevent any conflicts with Cerner Millennium program names.

Create a select statement in your program that selects the Order_id, Active_ind, the Catalog_cd, and the display value of the Catalog_cd field from the Orders table. Use the MaxRec control option to only read 1000 records.

From the File menu, select **Include/Compile** in VE or from the Build menu, select **Include/Compile** in DVDev to create the program object in the Discern Explorer object library. Correct any syntax errors and get a clean compile before proceeding.

Use the Prompt Builder icon on the toolbar or from the Tools menu, select **Prompt Builder** to open the Discern Prompt Builder.

Discern Prompt Builder should have created a prompt for an output device. Verify that you are selecting into the output device.

Suppose we want to give the user the ability to select active or inactive orders when the program is executed. We can do this using a Combo Box.

Click **Add** on the lower left side of the Prompt Control list to add a prompt to the form.

For the Prompt Display, enter **Active or Inactive?**

For the Prompt Name, enter **Active**.

For the Control Type, select **Combo Box**.

For the Prompt Type select **Expression**. Your Prompt Builder dialog should look similar to the following example:

The Active_ind field on the orders table is set to 1 (one) or 0 (zero) to indicate if the row is active or inactive. We want to set our prompt up so that at run time the user can select **Active** or **Inactive** from a list. Then, based on their selection the DPL form will pass either a 1 or 0 to the program to be used to qualify active or inactive rows.

Click the **Data Source** tab and verify that the From String Table radio button is selected.

In row 1 of the String Literals table enter **1** under column A and **Active** under column B.

In row 2 of the String Literals table enter **0** under column A and **Inactive** under column B.

Discern Prompt Builder3 - [1_JG_PROMPT_TUTORIAL_2:DBA*] New Form

The Data Source page defines the data used in list controls. Two types of sources may be defined: String Tables and Select Tables. A String Table defines a list of literal string values. The Select Table defines a SELECT statement to query the data from the database.

Prompt Controls

OUTDEV
Active

General Data Source

Where will the list data come from ?

☐ From Database ☒ From String Table

☐ Multiple Selection ☐ Label Columns
☐ Use Dual List
☐ Keep
☐ Modifiable ☐ Include Any(*)

String Literals : ☐ Parameter editor

	[A]	B
1	1	Active
2	0	Inactive
3		
4		
5		
6		
7		
8		

Add
Delete
Import

☒ Auto arrange

Form Tools

Program Info Save Cancel Help

Right-click in column A and select **Column > Hide**.

Right-click in row 1 and select **Set Default Row**. Your Prompt Builder dialog should look similar to the following example:

Discern Prompt Builder3 - [1_JG_PROMPT_TUTORIAL_2:DBA*] New Form

The Data Source page defines the data used in list controls. Two types of sources may be defined: String Tables and Select Tables. A String Table defines a list of literal string values. The Select Table defines a SELECT statement to query the data from the database.

Prompt Controls

OUTDEV
Active

General Data Source

Where will the list data come from ?

☐ From Database ☒ From String Table

☐ Multiple Selection ☐ Label Columns
☐ Use Dual List
☐ Keep
☐ Modifiable ☐ Include Any(*)

String Literals : ☐ Parameter editor

	[A]	B
*1	1	Active
2	0	Inactive
3		
4		
5		
6		
7		
8		

Add
Delete
Import

☒ Auto arrange

Form Tools

Program Info Save Cancel Help

When a string table is used as the dataset for a combo box control, the first visible column is the list of values that the user sees. Hiding column A causes column B to be seen in the drop down list by the user at run time. The light blue background indicates that column A is hidden. The asterisk (*) at the beginning of row 1 indicates that row 1 is the default row. The square brackets [A] on column A indicate that column A is the key column. At run time, the user selects a value from the drop-down list. The value in the key column from the row the user selects are passed to the Discern Explorer program. So in our case, when the user selects Inactive from the drop-down list, 0 is passed to the program.

Click **Save** to save your form.

Now you need to modify the qualifications in your program to use the active indicator that was selected at the prompt.

Add the following qualification to your select command:

```
WHERE O.ACTIVE_IND = $Active
```

If you are working in DVDev, include/compile your file, and execute your program using CTRL+R or from the Build menu, select **Run Prompt Program**.

If you are working in VE, click **Run Query** to execute your program.

Validate that the query returns active or inactive orders based on what is selected at the prompt.

Suppose we want to give the user the ability to select active, inactive, or both active and inactive orders when the program is executed. We can do this by changing the Active control from a Combo Box to a List Box.

Click the Prompt Builder icon on the toolbar or from the Tools menu, select **Prompt Builder** to open the Discern Prompt Builder.

Select the **Active** control on the Prompt Control list.

On the **General** tab, change the control type from Combo Box to List Box.

Making this change deletes the properties that were formerly set on the Data Source page for this control.

Click the **Data Source** tab and verify that the From String Table radio button is selected.

Check the Multiple Selection check box.

In row 1 of the String Literals table enter **1** under column A and **Active** under column B.

In row 2 of the String Literals table enter **0** under column A and **Inactive** under column B.

Right-click somewhere in column A and select **Column > Hide**.

Right-click somewhere in row 1 and select **Set Default Row**. Your Prompt Builder dialog should look similar to the following example:

Discern Prompt Builder3 - [1_JG_PROMPT_TUTORIAL_2:DBA*]

Check the Multiple selection box to allow the user to select one or more items from the list. The results are passed in a VALUE() statement in the parameter list.

Prompt Controls:

- OUTDEV
- Active

Buttons: Add, Delete, Import, Auto arrange

General Data Source

Where will the list data come from ?

☐ From Database ☒ From String Table

☒ Multiple Selection ☐ Use Dual List ☐ Keep ☐ Modifiable ☐ Label Columns ☐ Sort header ☐ Include Any(*)

String Literals : ☐ Parameter editor

	[A]	B
*1	1	Active
2	0	
3		
4		
5		
6		
7		
8		

Buttons: Properties, Form Tools, Program Info, Save, Cancel, Help

Click **Save** to save your form.

If you are working in DVDev, execute your program using CTRL+R or from the Build menu, select **Run Prompt Program**.

If you are working in VE, click **Run Query**.

Validate that the query is returning active or inactive orders based on what is selected at the prompt.

Depending on the data in the environment you are working in you might need to modify you qualifications to display active and inactive orders. For example, you might need to do an ad hoc query to find the Order_ID of an inactive order then add and o.order_id >= the order_id you found using the ad hoc query

String Table Properties

Combo boxes and list boxes have many common properties but some of the options that are available on the Data Source properties page flex for combo and

list boxes.

Modifiable: Available on combo boxes. Allows the user to enter free text data into the combo box edit field. If the user picks from the dataset, the value in the key column is passed to the Discern Explorer program. If the user types a value in the edit field, that value is passed as is.

Multiple Selection: Available on list boxes. Allows the user to pick one or more rows in the data set. The values in the key field for each of the selected rows is passed in the parameter.

Label Columns: Available on list boxes. Determines if the datasets column titles are displayed or not. Column titles can be defined using the **Properties** button.

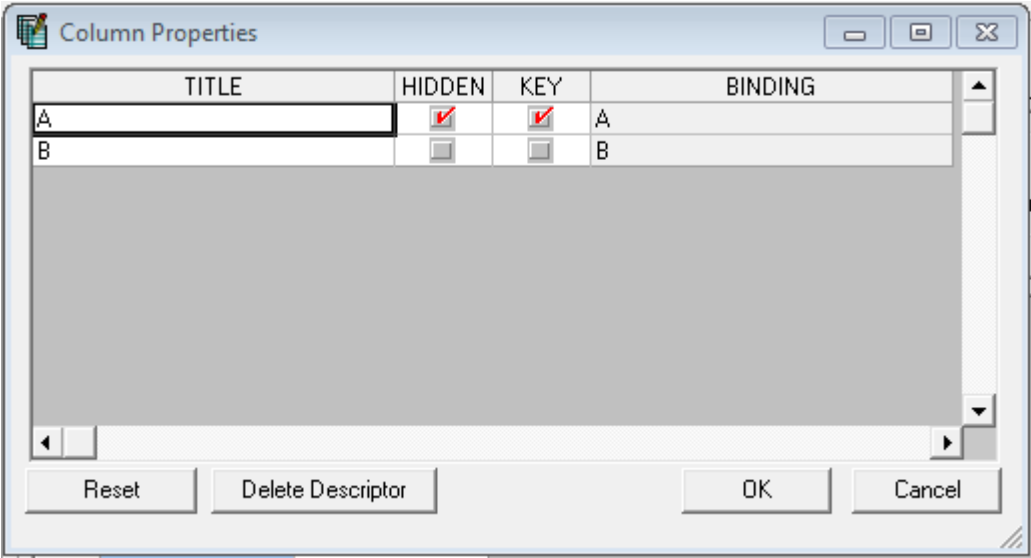
String Literals: The literal values that make up the dataset.

Setting the key column: The key column is indicated with [] around the column name. To set the key column, highlight the column, right-click and select **Set Key Column** or click **Properties** and select the key column. Each dataset must have one key column. The key column should be unique and is the value that is passed to the report program or other prompt control from the selected row.

Setting the default row: The default row is indicated with an asterisk (*) left of the row number. To set the default row, highlight the row, right-click and select **Set Default Row**.

Rows and columns can be inserted, deleted, and re-arranged using the context menu.

Properties button: The **Properties** button opens the Column Properties dialog box. Column properties describe how a dataset field is handled in the control at runtime.



The Column Properties editor allows the user to change the title of the column, determine if the column should appear in the control, or set the key column.

Title: For controls that support titles this is the heading of the column.

Hidden: If checked, the column is not shown in the control.

Key: Indicates which column will be the key column. Only one key column can be set.

Binding: The name of the column. This field cannot be modified.

Reset: Restores the columns properties to the default settings.

Hiding Columns:

Hidden columns appear as light blue columns in the spreadsheet and are not displayed at runtime.

To hide a column open the Column Properties editor and check the HIDDEN check box next to the column to hide, or in the Dataset property page, select the column in the editor right-click and select **Column > Hide**.

Working with Database Tables

A Database Table is the result of a Discern Explorer SELECT or EXECUTE statement. EXECUTE statements are limited to programs that meet interface specifications for Prompt Forms. We will not discuss how to create Discern Explorer programs that meet the interface specifications in this tutorial. Combo Boxes show one column of a dataset. The first visible column in the dataset is displayed. To create a database table for a Combo box, select **From Database** on the Data Source property page.

List Boxes can display multiple columns and rows. All the columns may be visible but only one can be designated as the key. As before to create a database table select **From Database** on the Data Source property page.

Suppose we want to give the user the ability to select a specific orderable from a subset of all orderables when the program is executed. The entire order catalog is contained in code set 200. We could use a Code Set control to display the complete list of orderables. However the list returned by a Code Set control would be very long, would take several seconds to load, and would be difficult for the user to look through to find the specific order they want. Instead of using a Code Set control to display the entire order catalog, we can use a Combo Box to display a subset of the entire order catalog.

Use the Prompt Builder icon on the toolbar or from the Tools menu, select **Prompt Builder** to open the Discern Prompt Builder.

Click **Add** on the lower left side of the Prompt Controls list to add a prompt to the form.

For the Prompt Display enter **Choose the Order**.

For the Prompt Name, enter **Order_Code**.

For the Control Type, select **Combo Box**.

For the Prompt Type, select **Expression**. Your Prompt Builder dialog should look similar to the following example:

Discern Prompt Builder3 - [1_JG_PROMPT_TUTORIAL_2:DBA*]

Select the Prompt Type for this prompt that is passed as the parameter of the program. Prompt Type "String" will pass the value of the control as a string (encapsulated between " characters) and expression passes the value as is.

Prompt Controls

OUTDEV

Active

Order_Code

Add

Delete

Import

☒ Auto arrange

General | Data Source

*Prompt Display :

Choose the Order

*Prompt Name :

Order_Code

*Control Type :

Combo Box

Status Bar Text :

*Prompt Type :

Expression

☐ Advanced

Form Tools

Program Info

Save

Cancel

Help

Click the **Data Source** tab to open the Data Source property page.

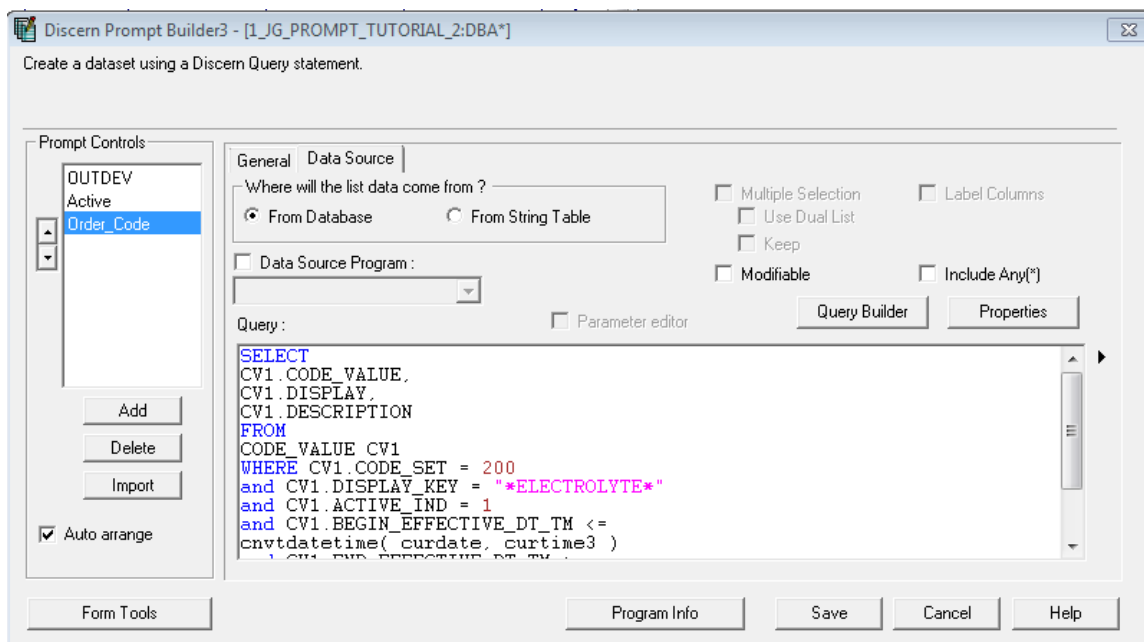
Select the **From Database** radio button.

Use the Query Builder or type the following Select statement in the Query edit box:

```
SELECT
CV1.CODE_VALUE,
CV1.DISPLAY,
CV1.DESCRPTION
FROM
CODE_VALUE CV1
WHERE CV1.CODE_SET = 200
and CV1.DISPLAY_KEY = "**ELECTROLYTE*"
and CV1.ACTIVE_IND = 1
and CV1.BEGIN_EFFECTIVE_DT_TM <=
cnvtdatetime( curdate, curtime3 )
and CV1.END_EFFECTIVE_DT_TM >=
cnvtdatetime( curdate, curtime3 )
```

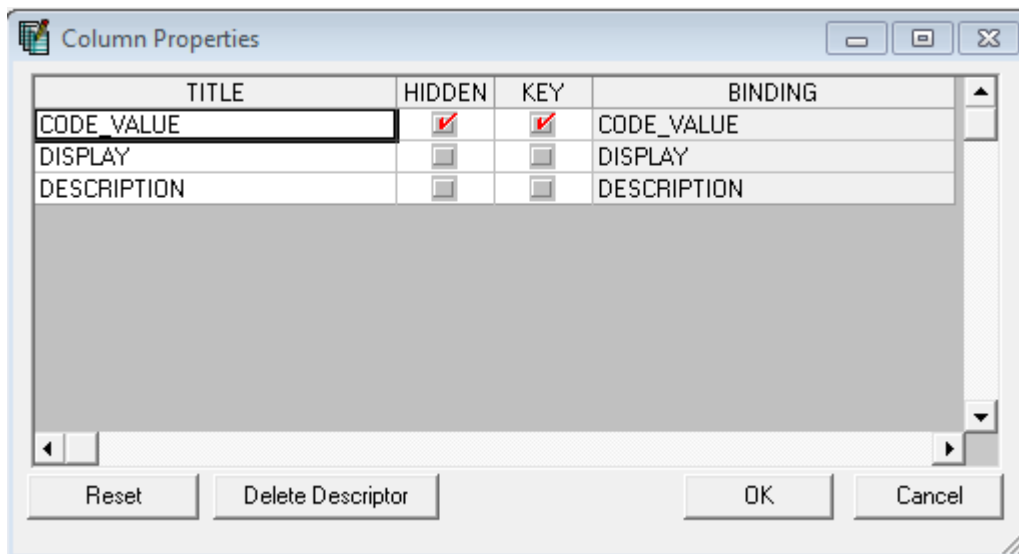
If you do not have any orderables in code set 200 with a Display_Key that contains the string ELECTROLYTE you will need to use a different string in your qualification.

Your Prompt Builder dialog should look similar to the following example:



Click **Properties** to open the Column Properties editor.

Use the check boxes to hide the Code_Value field and make it the key field. Your Column Properties dialog should look similar to the following example:



At run time this Order_Code control displays a list of orderables from Code Set 200 that have a Display Key that contains the string you used in your qualification above. When the user selects a row from the list, the Code_Value is passed to your program. You now need to add the code to your program to use the Code_Value in the qualifications.

Click **OK** to close the Column Properties editor.

Click **Save** button to save your DPL Form.

Change the qualification in your select command from:

```
WHERE O.ACTIVE_IND = $Active
```

To:

```
WHERE O.ACTIVE_IND = $Active
and O.CATALOG_CD = $Order_Code
```

If you are working in DVDev, include/compile your file, and execute your program using CTRL+R or from the Build menu, select **Run Prompt Program**.

If you are working in VE, click **Run Query**.

Validate that the query is returning the type of orders you selected at the prompt.

Suppose we want to give the user the ability to select one or more orderables when the program is executed. We can do this by changing the Order_Code control from a Combo Box to a List Box.

Open the Discern Prompt Builder and select the Order_Code control.

Open the Data Source properties page, highlight the existing query and use CTRL+C to copy it to the clipboard.

Click the **General** tab and change the Control Type from Combo Box to List Box.

Click the **Data Source** tab and click the **From Database** radio button.

Use CTRL+V to paste the query you copied above back into the Query edit box.

Note
When you changed the control type the properties for the control were reset back to the defaults, which erased the query. Check the **Multiple Selection** check box. Click **Properties** to open the Column Properties editor. Use the check boxes to hide the Code_Value field and make it the key field. Click **OK** to close the Column Properties editor. Your Prompt Builder dialog should look similar to the following example:

TITLE	HIDDEN	KEY	BINDING
CODE_VALUE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	CODE_VALUE
DISPLAY	<input type="checkbox"/>	<input type="checkbox"/>	DISPLAY
DESCRIPTION	<input type="checkbox"/>	<input type="checkbox"/>	DESCRIPTION

Click **Save** to save the changes to your DPL form.
If you are working in DVDev, execute your program using CTRL+R or from the Build menu, select **Run Prompt Program**.
If you are working in VE, click **Run Query** to execute your program.
Validate that the query is returning orders based on what is selected at the prompt.

Database Table Properties

Combo boxes and list boxes have many common properties but some of the options that are available on the Data Source properties page flex for combo and list boxes.

Data Source Program: When checked, the user can select from a predefined list of Discern Explorer programs using the combo box. These programs must be written to meet interface specifications for Prompt Forms. We will not discuss how to create this type of program in this tutorial.

Multiple Selection: For List boxes only. This field determines if the user is allowed to select more than one row from the list. If checked, the values are passed to the program using the Discern Explorer VALUES statement. Example (execute abc MINE, VALUES(param1, param2)).

Modifiable: This allows the end user to enter free text data into the combo box edit field. If the user picks from the list, the datasets key for that item is passed, however, if the user types text in the edit field the text entered is passed as is.

Label Columns: Available for List boxes only. Determines if the datasets column titles are displayed or not. Column displays can be defined using the **Properties** button.

Use Dual List. Available for List boxes only. Determines if the values of the list box can be selected using a dual list dialog window. Dual list should be limited to lists that have a query parameter that can be re-entered to get a different selection. For example, if the dual list is picking patient names, the user may query for last name of Johnson and pick a few people from that list. Then query all of the Smiths to pick a few more names from the second list.

Query Builder. Opens the Discern Query Builder. The Query Builder allows you to build a Discern Explorer select statement using a point and click interface. When the Query Builder is closed, the select statement is written in text form in the Query edit box. If a select statement exists in the Query edit box when the Query Builder is opened, that select is pulled into the builder.

Discern Prompt Builder3 - [1_JG_PROMPT_TUTORIAL_2:DBA*]

Define or modify column visibility, display order, and which column contains the key to the dataset.

Prompt Controls

OUTDEV
Active
Order_Code

Add
Delete
Import

☒ Auto arrange

Form Tools

General Data Source

Where will the list data come from ?
☒ From Database ☐ From String Table

☐ Multiple Selection ☐ Label Columns
☐ Use Dual List ☐ Sort header
☐ Keep
☐ Modifiable ☐ Include Any(*)

Data Source Program :

Query : ☐ Parameter editor

```
SELECT
CV1.CODE_VALUE,
CV1.DISPLAY,
CV1.DESCRPTION
FROM
CODE_VALUE CV1
WHERE CV1.CODE_SET = 200
and CV1.DISPLAY_KEY = "*ELECTROLYTE*"
and CV1.ACTIVE_IND = 1
and CV1.BEGIN_EFFECTIVE_DT_TM <=
cnvtdatetime( curdate, curtime3 )
1 CV1.END_EFFECTIVE_DT_TM
```

Program Info

Query. Edit box to type the select or executable statement. Most of the Discern Explorer SELECT clauses can be used. SELECT INTO a table, file, or printer queue cannot be used. Report writer sections (HEAD REPORT/DETAIL etc) cannot be used.

Queries can use \$CURACCEPT which will open the Query Viewer and prompt the user for a value. The control uses the value that is entered in the query to build a dataset to populate the list. Once the list is populated, the user can use the F2 key or right-click in the control to reset the list. Queries can also use Master-Detail links, which are discussed later in this tutorial.

Properties. Opens the Column Properties editor, which can be used to define column properties for the dataset. Column properties are the runtime visible aspects of the fields in the SELECT or Discern Explorer program. By default, column properties are created for each field listed in the SELECT statement. It is important therefore, not to use Select * in the query. Adding or removing fields from the select may not be automatically reflected in the columns properties. Having Discern Prompt Builder automatically reflect changes to the select statement in the column properties could inadvertently destroy properties that were set manually. When changes to the select are made, you should verify that the changes are correctly reflected in the column properties.

Discern Query Builder

TABLES | FIELDS | QUALIFICATIONS | GROUP BY | SORT | CONTROL OPTIONS | VIEW QUERY

CATEGORIES

- COMMON
- FAVORITES
- INLINE
- MILLENNIUM
- DICTIONARY
- RDBMS
- ALL
- APPLICATIONS
- DOMAINS
- RECORD STRUCTURES

Tables

ACCESSION
CLINICAL_EVENT
CODE_VALUE
DUMMYT
ENCOUNTER
ORDER_CATALOG
ORDERS
PERSON
PRSNL
RESULT

Selected Tables

Table	Alias
PERSON	P

Customize

Table Filter:

TITLE property defines the column title to display in the list box.

HIDDEN check box can be used to prevent the column from being displayed in the control. At least one column must be visible in the data set.

KEY identifies which field is passed to Detail controls or to the report program. Each data set must have one and only one KEY. By default, the first column in the data set is the KEY.

BINDING is the physical name of the field in the data set that the column is representing. The Binding name is set by the DPL.

Reset button restores the columns properties to match the default column properties. With a dataset from a SELECT statement the columns are re-parsed.

Master-Detail Linking

Master detail links can be used to have one prompt effect what is displayed in a second prompt. Using master-detail linking you can create parent child relationships where the parent (Master) control passes data to the child (Detail) control(s). The child control responds using data from the parent as an input source. Master controls can have one or more detail controls and can themselves be detail controls to another master control. Detail controls can have one or more master controls. When a detail's run condition is met, the control executes the query and populates the control with the selected data. Master-Detail forms are created by referencing the master control in the detail control's query as a condition in the where clause using the \$ operator. When a detail references a master we say the detail is "linked" to the master.

Currently our program displays a list of orderables that have ELECTROLYTE in the Display_Key field. We can use master detail linking to give the user the ability to enter a value and then display a list of orderables that contain that value. The user can then pick one or more of the orderables from the list.

Open the Discern Prompt Builder click **Add** to create a new prompt control.

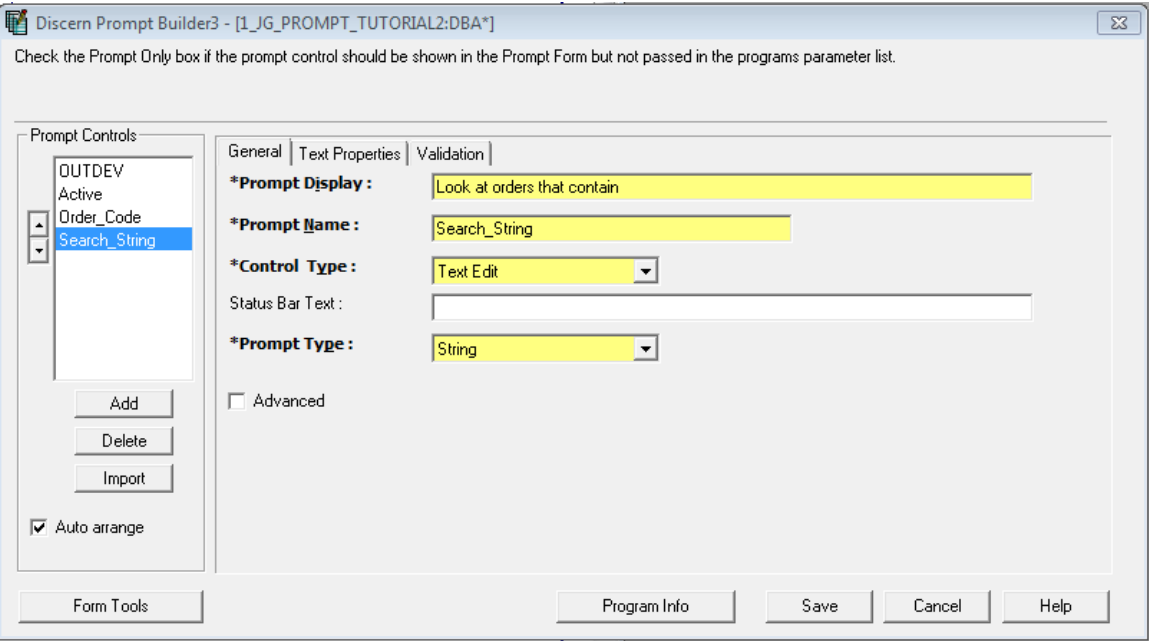
For the Prompt Display, enter **Look for orders that contain**.

For the Prompt Name, enter **Search_String**.

For the Control Type, select **Text_Edit**.

For the Prompt Type, select **String**.

Use the Arrow buttons on the left of the Prompt Controls list to move the Search_String control above the Order_Code control. Your Prompt Builder dialog should look similar to the following example:



With all of the prompt controls we have created up to this point, we have passed the value the user entered or selected to the Discern Explorer program that is associated with our DPL form. We want to use the value the user enters at the Search_String control to populate the list that is displayed in the Order_Code control. To prevent the Search_String value from being passed as a parameter to the program, we want to set this control up as a prompt only control. On the General properties page, select the Advanced check box to display additional properties.

Select the Prompt Only check box. Your Prompt Builder dialog should look similar to the following example:

Discern Prompt Builder3 - [1_JG_PROMPT_TUTORIAL2:DBA*]

Check the Prompt Only box if the prompt control should be shown in the Prompt Form but not passed in the programs parameter list.

Prompt Controls

- OUTDEV
- Active
- Search_String**
- Order_Code

Add
Delete
Import

☒ Auto arrange

Form Tools

General | Text Properties | Validation

*Prompt Display : Look at orders that contain

*Prompt Name : Search_String

*Control Type : Text Edit

Status Bar Text :

*Prompt Type : String

☒ Advanced

☒ Label left ☐ Hide label

☒ Prompt only ☐ Auto group

☐ Required

☐ Trim

Control Position

Left : 160 Top : 69

Width : 250 Height : 18

H.offset V.offset

Program Info Save Cancel Help

Using the Prompt Only check box causes the value that the user enters during runtime to be assigned to the \$Search_String variable but \$Search_String is not passed to the Discern Explorer program when it is executed. In our example, we want to use the \$Search_String in the query that is executed by the Order_Code control so that the list of orderables that is displayed by the Order_Code control changes based on what the user enters.

Using the Text Properties page on the Search_String control set the Character Case to Upper. This converts what the user enters to upper case.

Enter **ELECTROLYTE** as the default value.

Select the Data Source page on the Order_Code control.

Check the Label Columns check box.

Change the qualification in the query edit box from:

```
SELECT
    CV1.CODE_VALUE ,
    CV1.DISPLAY ,
    CV1.DESCRPTION
FROM
    CODE_VALUE CV1
WHERE CV1.CODE_SET = 200
    and CV1.DISPLAY_KEY = "**ELECTROLYTE*"
    and CV1.ACTIVE_IND = 1
    and CV1.BEGIN_EFFECTIVE_DT_TM <=
cnvtdatetime( curdate, curtime3 )
    and CV1.END_EFFECTIVE_DT_TM >=
cnvtdatetime( curdate, curtime3 )
```

To:

```

SELECT
    CV1.CODE_VALUE,
    CV1.DISPLAY,
    CV1.DESCRPTION
FROM
    CODE_VALUE CV1
WHERE CV1.CODE_SET = 200
and CV1.DISPLAY_KEY =
VALUE(CONCAT(" ", $SEARCH_STRING, " "))
and CV1.ACTIVE_IND = 1
and CV1.BEGIN_EFFECTIVE_DT_TM <=
cnvtdatetime( curdate, curtime3 )
and CV1.END_EFFECTIVE_DT_TM >=
cnvtdatetime( curdate, curtime3 )

```

Your Discern Prompt Builder dialog should look similar to the following example:

We are using the Concat() function to place the wildcard in front of and in back of the string the user enters in the Search_String control, and the Value() function to pass the result of the Concat() by value instead of by reference.

Click **Save** to save your DPL form.

If you are working in DVDev, execute your program using CTRL+R or from the Build menu, select **Run Prompt Program**.

If you are working in VE, click **Run Query** to execute your program. The DPL form that is displayed should look similar to the following example:

Discern Prompt: 1_jg_prompt_tutorial2:DBA

Output to File/Printer/MINE: MINE

Active or Inactive?:

Look at orders that contain: ELECTROLYTE

Choose the Order:

- ☐ DISPLAY
- ☐ Electrolyte Solution (Isolyte-E)
- ☐ Electrolyte Solution (Plasma-Lyte)
- ☐ Electrolyte Solution (Plasma-Lyte 56)
- ☐ intravenous electrolyte solution

Execute Cancel

☐ Return to prompts on close of output

Ready

When a dataset is created for a List Box or Combo Box using a query, the query is executed by the DPL as soon as it has everything it needs. Because we are referencing \$Search_String in the query that creates the dataset for the Order_Code control, the query is executed by the DPL as soon as \$Search_String is given a value. Because we are using ELECTROLYTE as the default, \$Search_String is given a value when the form is opened. Because \$Search_String has a value, the query to build the dataset is also executed when the form is opened. The result is that the list of orderables that the user can choose is automatically populated with orderables that contain the string ELECTROLYTE. To display a different list of orderables, the user can change ELECTROLYTE to some other value and tab or click out of the Search_String control. As soon as the Search_String control no longer has focus, the value of \$Search_String is changed and the query that populates the Order_Code dataset is re-executed. One thing to keep in mind is that it would not be a good idea to make the default value for \$Search_String an * (asterisk). Doing so would cause the query that builds the dataset for the Order_Code to select all of the rows from code set 200. Because code set 200 is generally very large, this query would take a while to execute and would cause the form to appear to lock up. It is a good idea to make sure that any query that populates a dataset only take a second or two to execute.

Change ELECTROLYTE to some other value (try using CBC or BUN) and tab out of the Look at orders that contain field.

Validate that the list of orders that are displayed changes. The DPL form that is displayed should look similar to the following example:

Discern Prompt: 1_jg_prompt_tutorial2:DBA

Output to File/Printer/MINE: MINE

Active or Inactive? ☒ Active ☐ Inactive

Look at orders that contain: ELECTROLYTE

Choose the Order: DISPLAY

- ☐ Acides Aminés-Electrolytes (Aminomix 800)
- ☐ Amino Acids 3.5% with 25% Dextrose and E
- ☐ Amino Acids 3% with Electrolytes (Procal
- ☐ Amino Acids 8.5% with Electrolytes (Amin

Execute Cancel

☐ Return to prompts on close of output

Ready

Select several of the orders that are displayed in the list and execute your program.

Verify that only orders that match the selection criteria are displayed in the output.

Audit Logs and Troubleshooting

When a program is executed from Explorer Menu (EM), Visual Explorer (VE), or Discern Visual Developer (DVDev), the DiscernReport.dll is used to display the DPL form. When you click **Execute**, the DiscernReport.dll uses a script to execute your program and passes the values selected on the form to the program as parameters. The script also logs information including what program was executed, who executed it, and what parameters were passed to the program. (Note: When executing an Ad Hoc query from DVDev or using Run Query in VE, no information is logged. Executing a prompt program from DVDev, executing any program from EM, or using Run Report from VE logs the information.) Three standard reports have been created that read the log information 1) CCL_RPT_AUDIT_LOG, 2) CCL_RPT_AUDIT_SUMMARY and 3) CCL_RPT_QUERY. These reports are documented in Discern Explorer Help (DiscernExplorerHelp.exe) under System Reference > **Useful CCL Utilities**. These reports are also helpful for troubleshooting problems that occur when executing a program that uses a DPL form. For example, they can be used to see exactly what parameters the form is passing to the program.

If you have been using the **Run Query** button in VE to execute your program, no information has been written to the logs. To write some information to the logs, open DVDev and use CTRL+R or from the Build menu, select **Run Prompt Program** to execute your program.

In DVDev use CTRL+R or from the Build menu, select **Run Prompt Program** to execute CCL_RPT_AUDIT_LOG. The following form should be displayed:

Discern Prompt: 1_jg_prompt_tutorial2:DBA

Output to File/Printer/MINE: MINE

Active or Inactive? ☒ Active ☐ Inactive

Look at orders that contain: CBC

[Choose the Order](#)

DISPLAY	DESCRIPTION
<input type="checkbox"/> CBC1	CBC1
<input type="checkbox"/> CBC-JPS	CBC-JPS
<input type="checkbox"/> CHIA 85 CBC	CHIA 85 CBC
<input type="checkbox"/> DG CBC with Diff	DG CBC with Diff

☐ Return to prompts on close of output

Ready

Enter the name of the program that you have been executing, (DPL_PROMPT_TUTORIAL2 for example) and click **Execute**.

The information displayed by the report includes the parameters that were passed from the DPL form to the program when it was executed.

Conclusion

This tutorial has attempted to show the basic operation and use of the six controls that are available in the initial release of Discern Prompt Builder. We have looked at the controls from a very high level and used a very small part of the functionality that is available. We are confident that with practice, investigation, and experimentation, you will be able to expand upon this basic functionality and gain a complete knowledge of the power and flexibility the Discern Prompt Builder can provide.