Discern Prompt Builder Tutorial

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

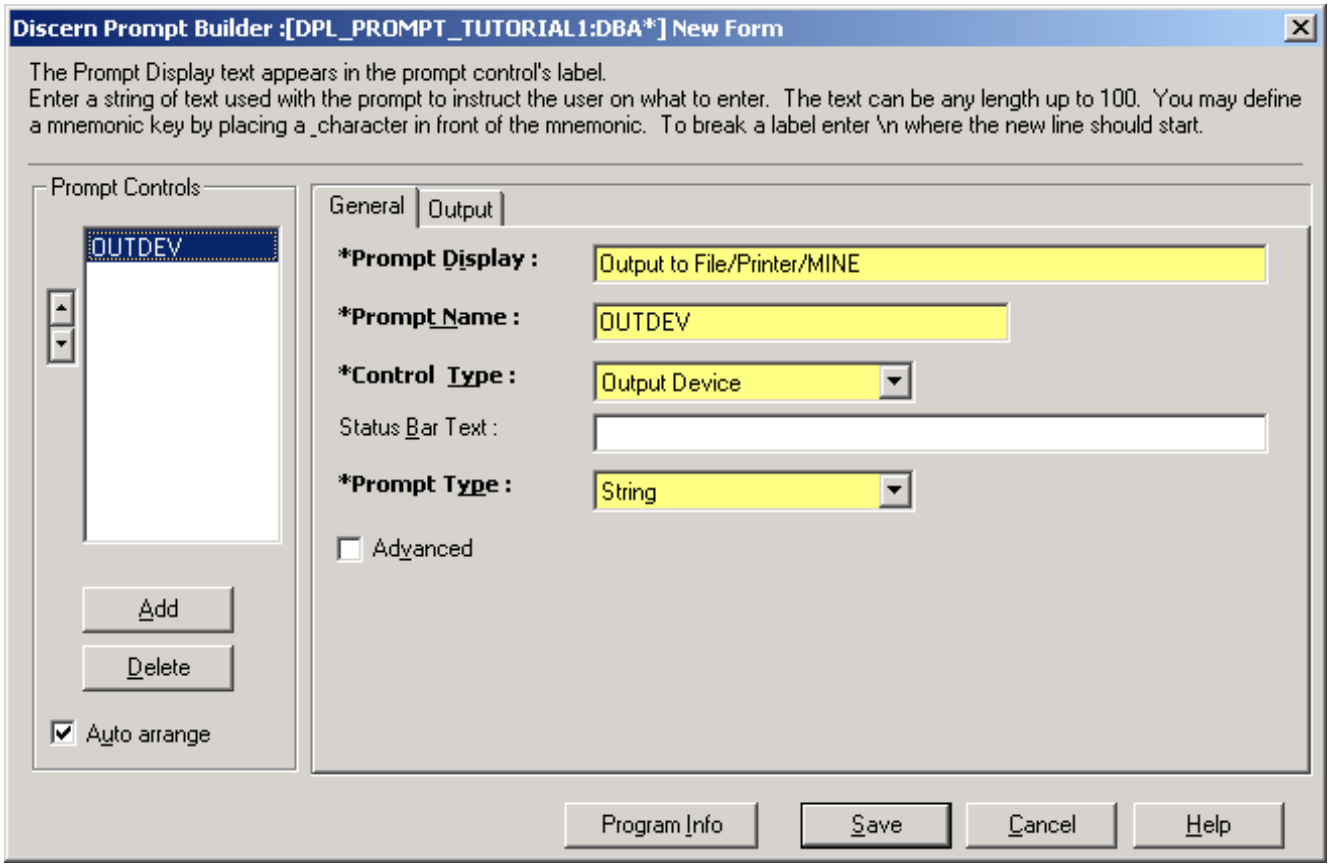
In order to see how these controls function, we will create a simple program and then create a DPL form that uses the six control types.

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.

Use File > Include/Compile in VE or Build > 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 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 look similar to the following example:



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 setup to use a Text control. Note that the dialogs caption displays the programs name and group number, an * indicates that the form has been modified, and New Form is letting you known 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 will be 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.
*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 will appear 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 then select String. If the program is expecting numeric values or calculations then 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 use the keyboard up or down arrows to change the current focus. To change the ordering of the prompt controls on the DPL form select the prompt control and press the up/down arrows on the left edge of the list with the mouse.

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.

In order to see how the Output Device control functions by default, save the form using the Save button. Saving the form will add the following code to your source code file:

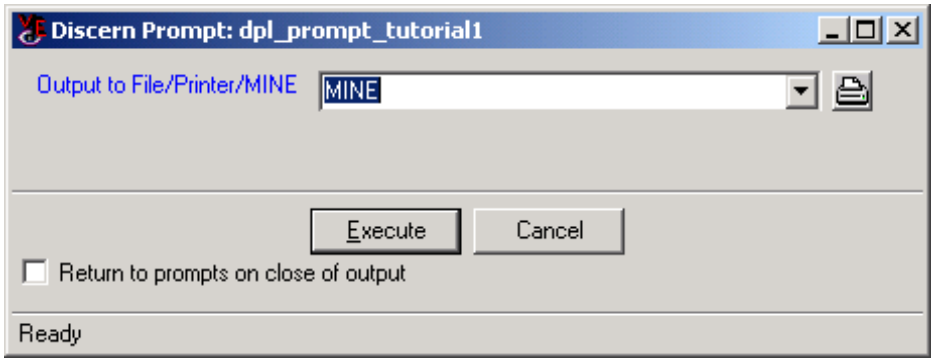
```
prompt
 "Output to File/Printer/MINE" = "MINE"
with OUTDEV
```

If you are working in DVDev this code will be 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.

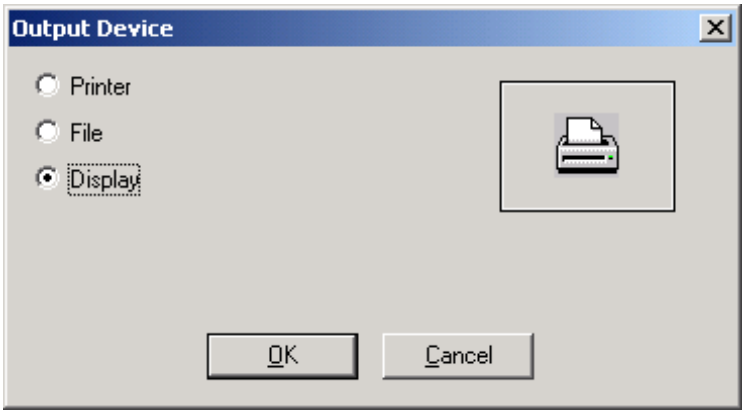
If you are working in DVDev, modify your select command to select into $OUTDEV, include/compile your file, and execute your program using Ctrl+R or the Run Prompt Program option on the Build menu.

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

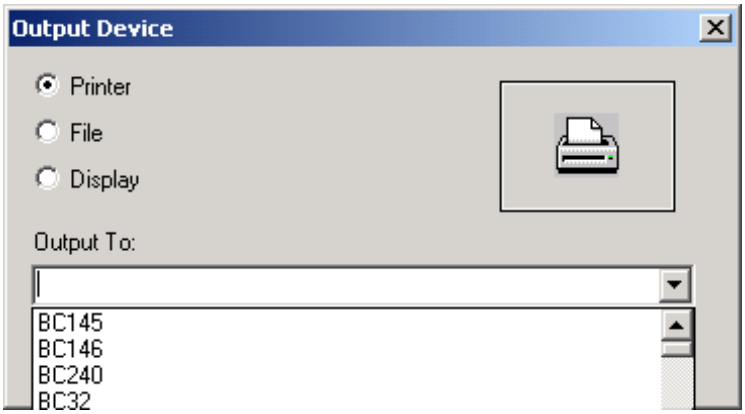
Executing a program that has a DPL form associated with it, from any application that uses DPL, will cause the form to be displayed. The DPL form associated with your program should be similar to the following:



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 will be the default value MINE. The user can enter any value they want in the text box or use the printer browser button. Pressing the printer browse button will display the following dialog box:



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



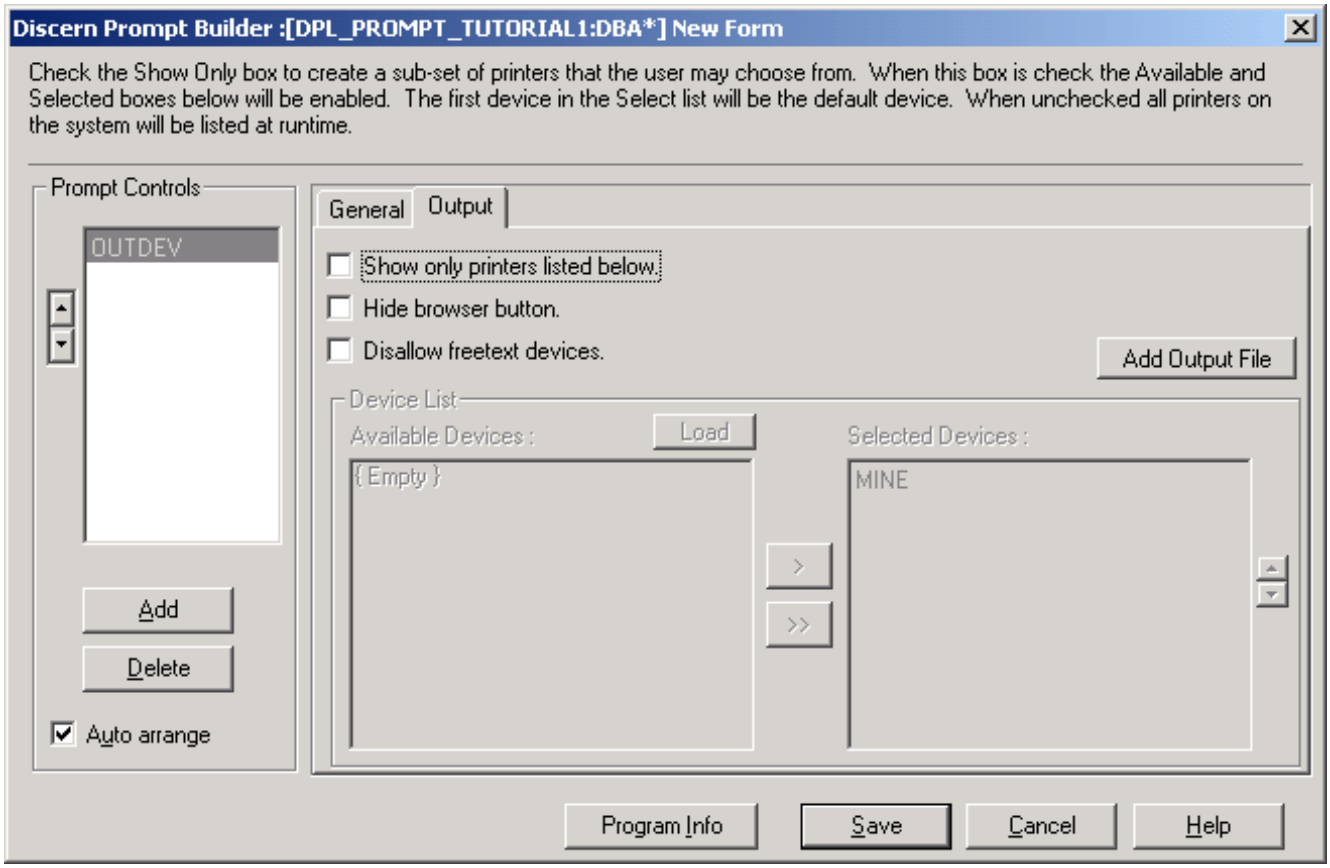
So at this point the user can send the output of the program to any device or file.

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 will need to make some changes to the properties on the Output tab.

Use the Cancel buttons to close the DPL Form with out executing the program.

Use the Prompt Builder icon on the toolbar or Tools>Prompt Builder to open the Discern Prompt Builder.

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



Output Properties

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

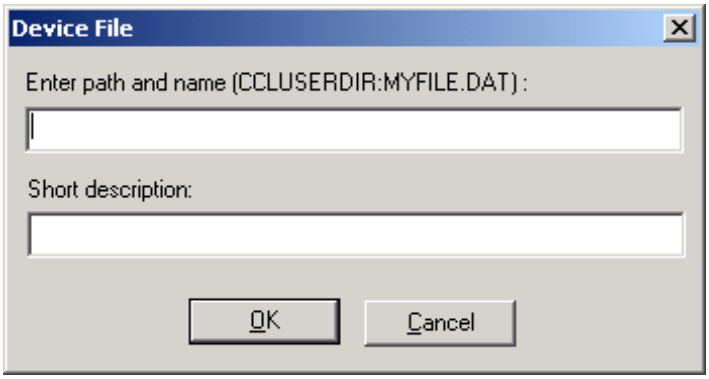
Hide browser button. If this is checked the printer icon next to the output device combo box will not be displayed. The browser button allows the user to search for printers not defined in the combo box list box or to create a data file for the report.

Disallow free text devices. If this is checked the end-user will not be allowed to type in the edit box and will be 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 will be the default output device. If **Show only printers listed below** is not checked then the default is MINE. Use the up-down control 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 will display the following dialog box.



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.

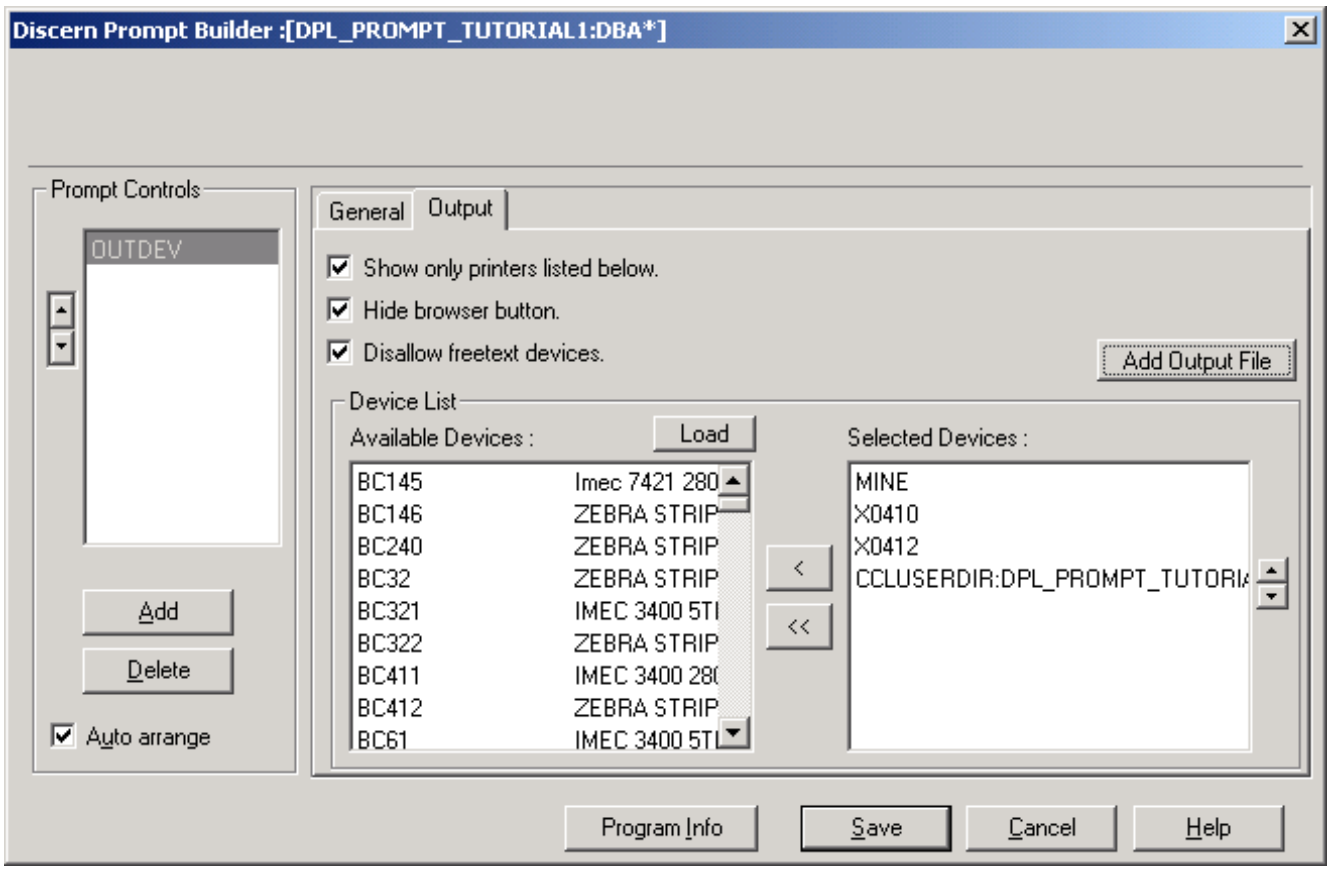
Check the Show only printers listed below, Hide browser button, and Disallow free text devices check boxes.

Click the Load button to populate the Available Devices: list.

Highlight MINE and click the arrow button to move MINE to the Selected Devices: list.

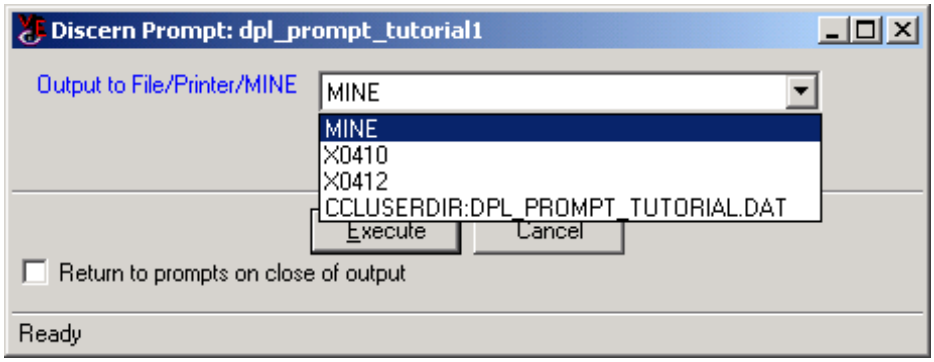
Highlight several printers in the Available Devices list and click the arrow button to move them to the Selected Devices: list.

Click the Add Output File button. 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:



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

Execute your program using Ctrl+R or the Run Prompt Program option on the Build menu in DVDev or the Run Query button in VE. When you select the drop down list on your form it should look similar to the following:



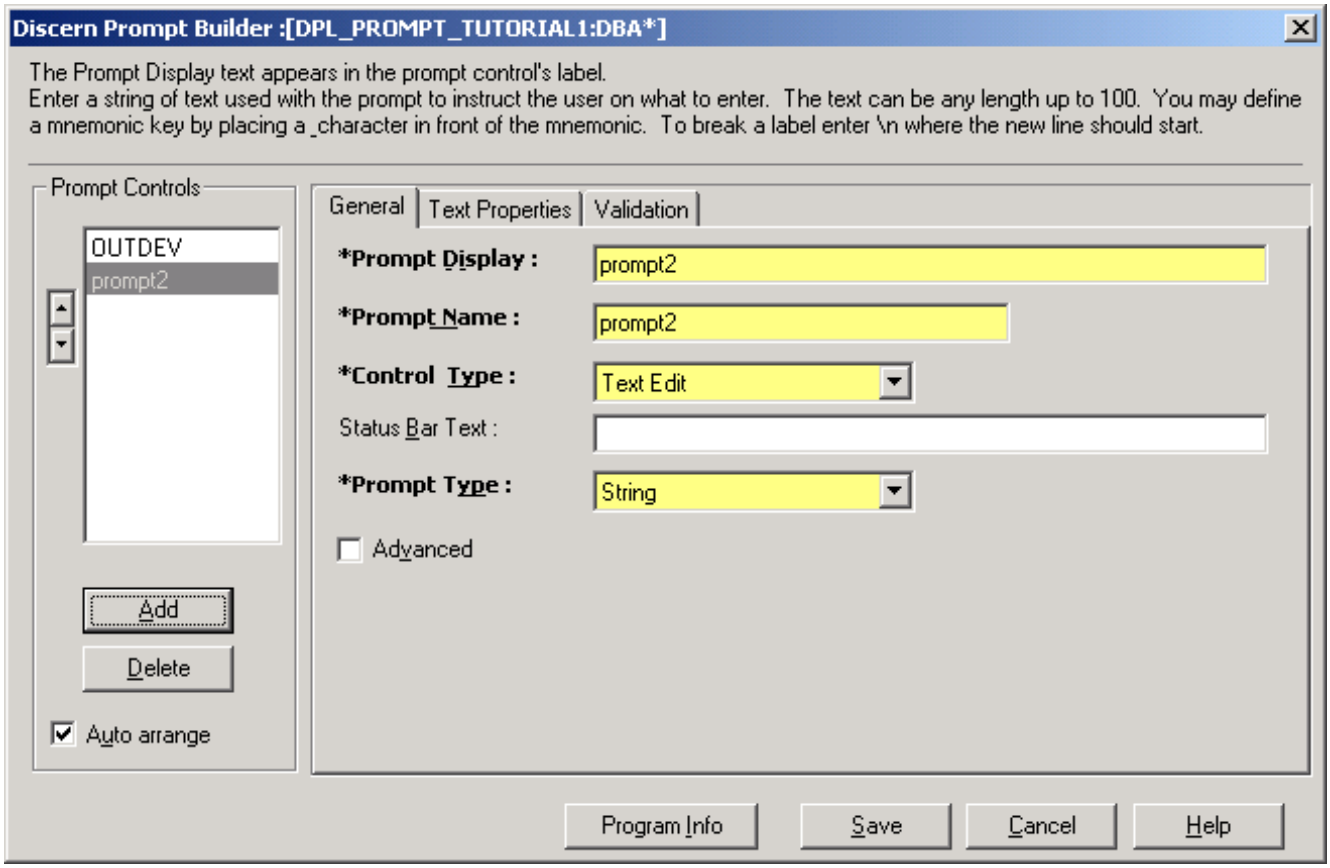
Now you can only choose one of the options on the list as the output device when your program is executed. The printer browser button is no longer shown and typing a free text value is not allowed.

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 will pass 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 will force the user to select a value from the list but the list box will allow 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 the Add button 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:

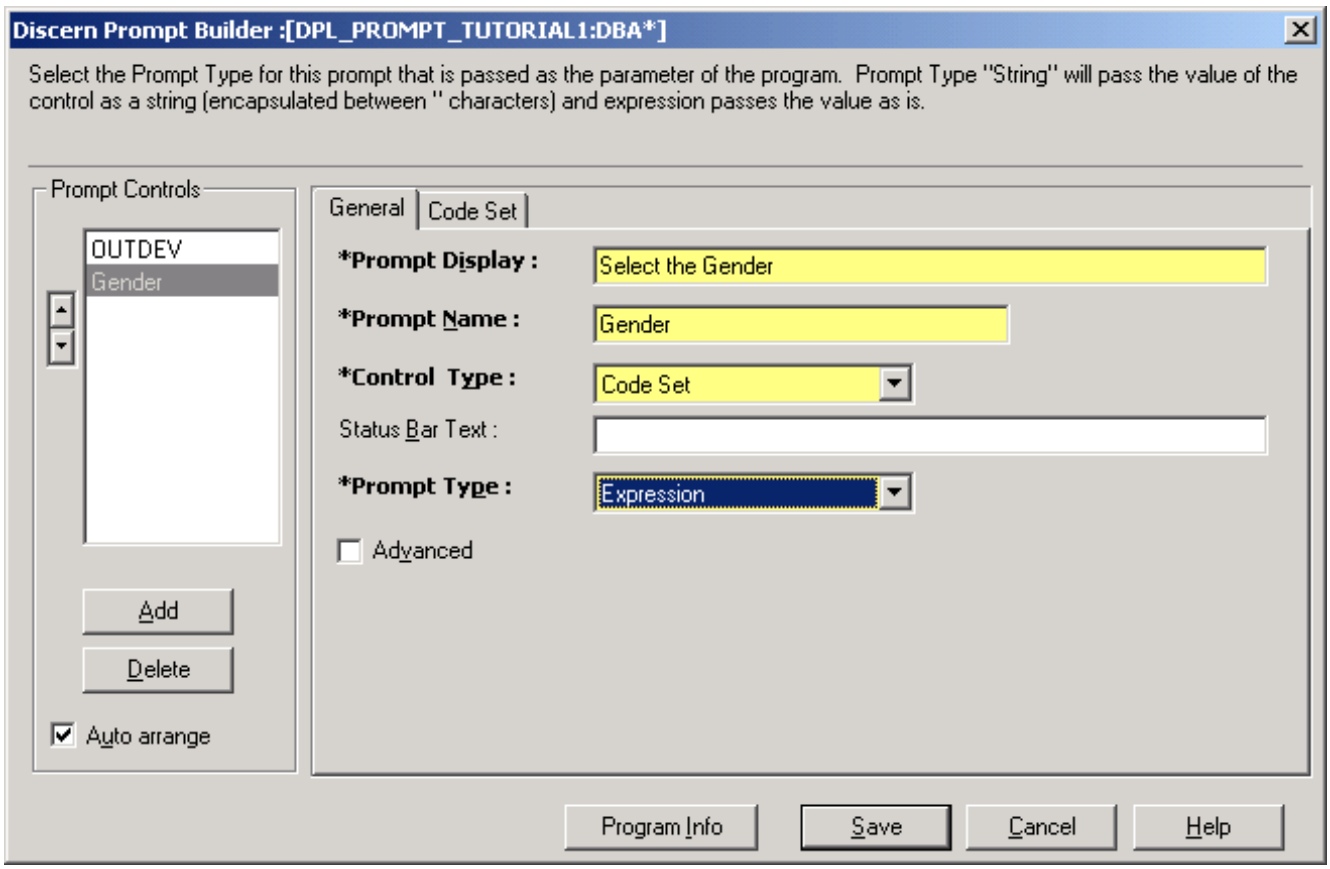


Enter Select the Gender in the Prompt Display field

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:



Select 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 will be displayed in the controls list. This number can be seen in several different places:

- From the Query Builder within VE or DVDev select the Glossary button on the Fields tab and the code set will be 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 will be 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 will be 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 will be passed to the Discern Explorer program using the Value() function. When used in a qualification the Value() function will cause 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.

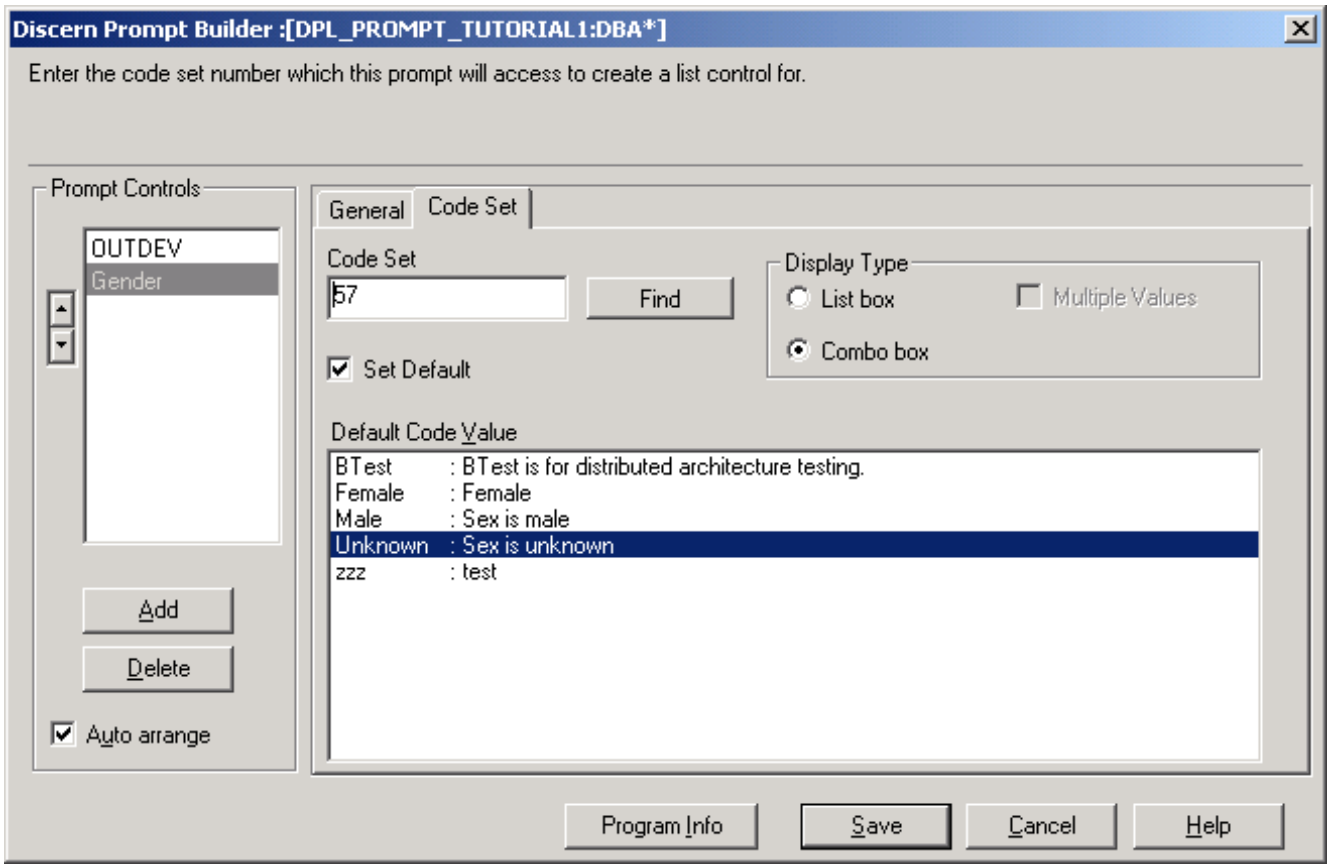
Because we want to give the user the ability to only select people with a specific gender we need to provide them with a list of values that can be compared to the Sex_CD field on the Person table. The Sex_CD field uses code set 57.

Enter 57 as the Code Set

Select Combo box in the Display Type area

Check 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:



Save the changes to the form by clicking the Save button. Saving the form will modify 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 will be 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 will be the code value number that represents the Unknown sex code in your environment.

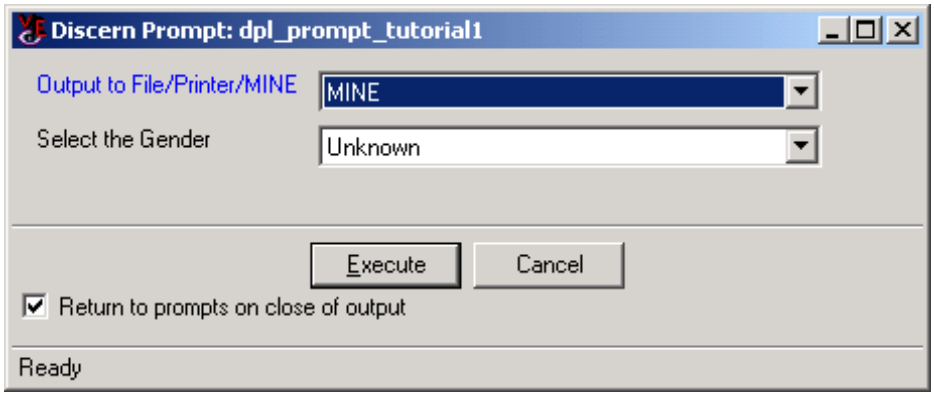
Now that the control is set up we need to add a qualification that uses the value that is selected at run time to only select people with the selected gender.

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 the Run Prompt Program option on the Build menu.

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

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



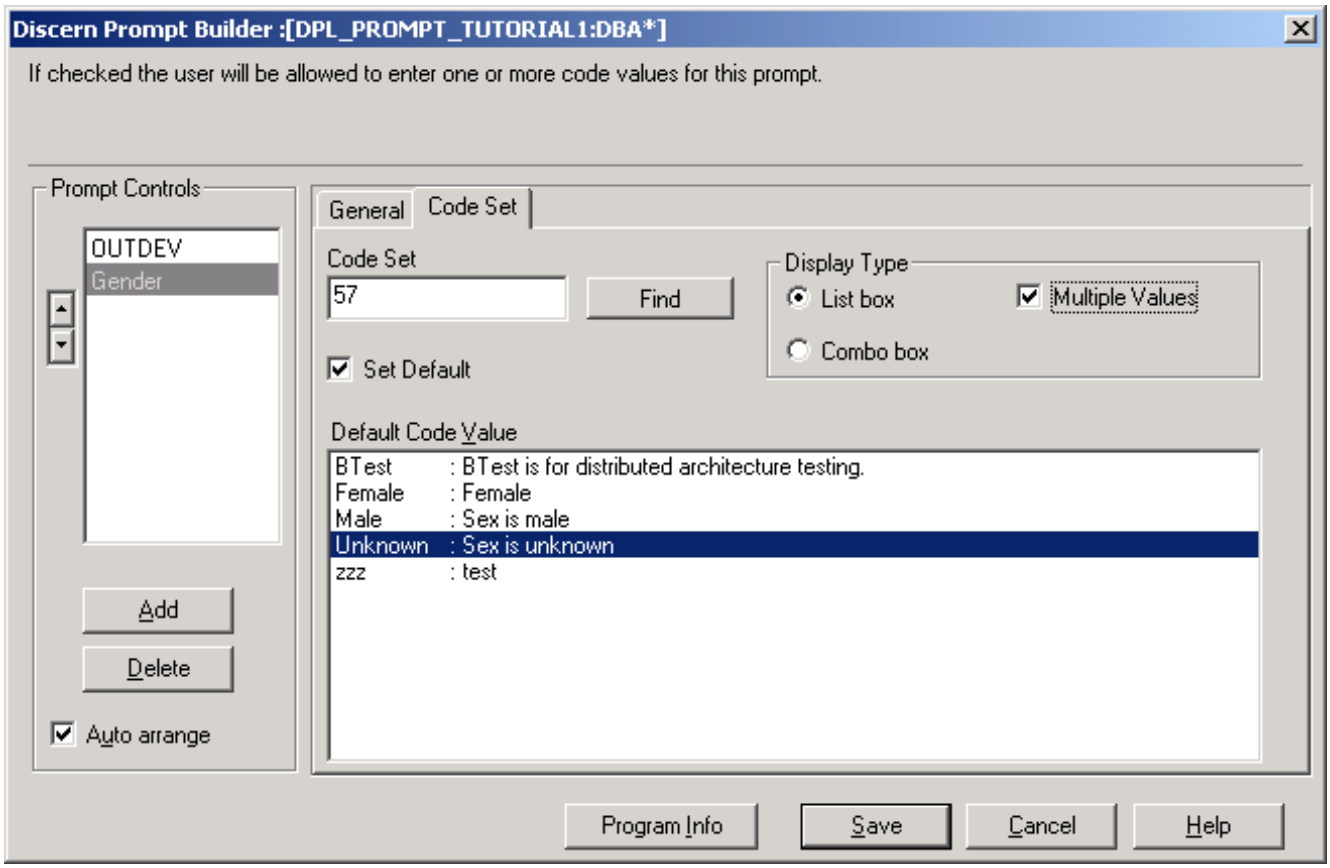
Use the drop down list on the Select the Gender combo box to select people with a specific gender. Execute your program and verify that only people with the selected gender are displayed.

If instead of only having the ability to select one gender, the user needs the ability to select one or more genders, we can modify the Gender control on our 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.

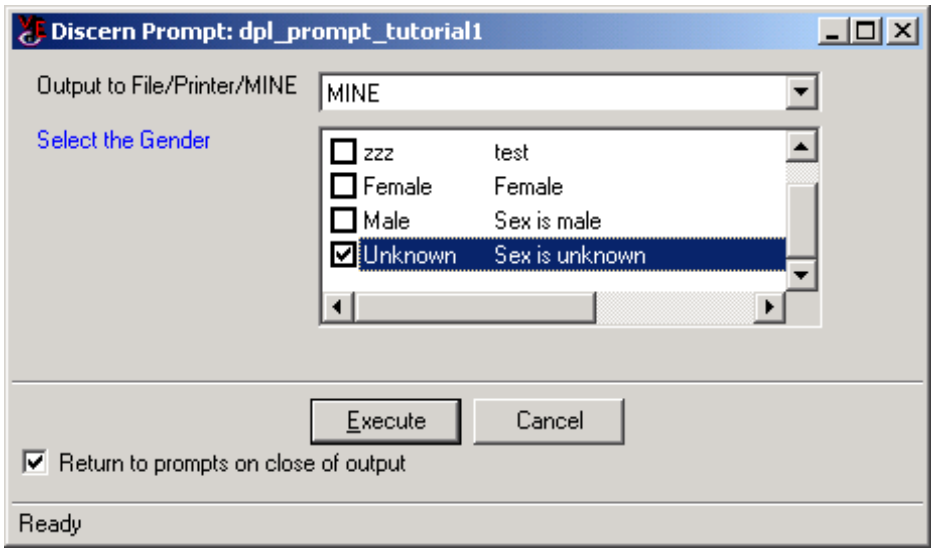
Select List Box and check the Multiple Values check box in the Display Type area. Your Discern Prompt Builder dialog should look similar to the following example:



Save the changes to the form by clicking the Save button.

If you are working in DVDev, execute your program using Ctrl+R or the Run Prompt Program option on the Build menu.

If you are working in VE, execute your program using the Run Query button. 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 will not be displayed. The code value for the item that is highlighted in the list will be 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.

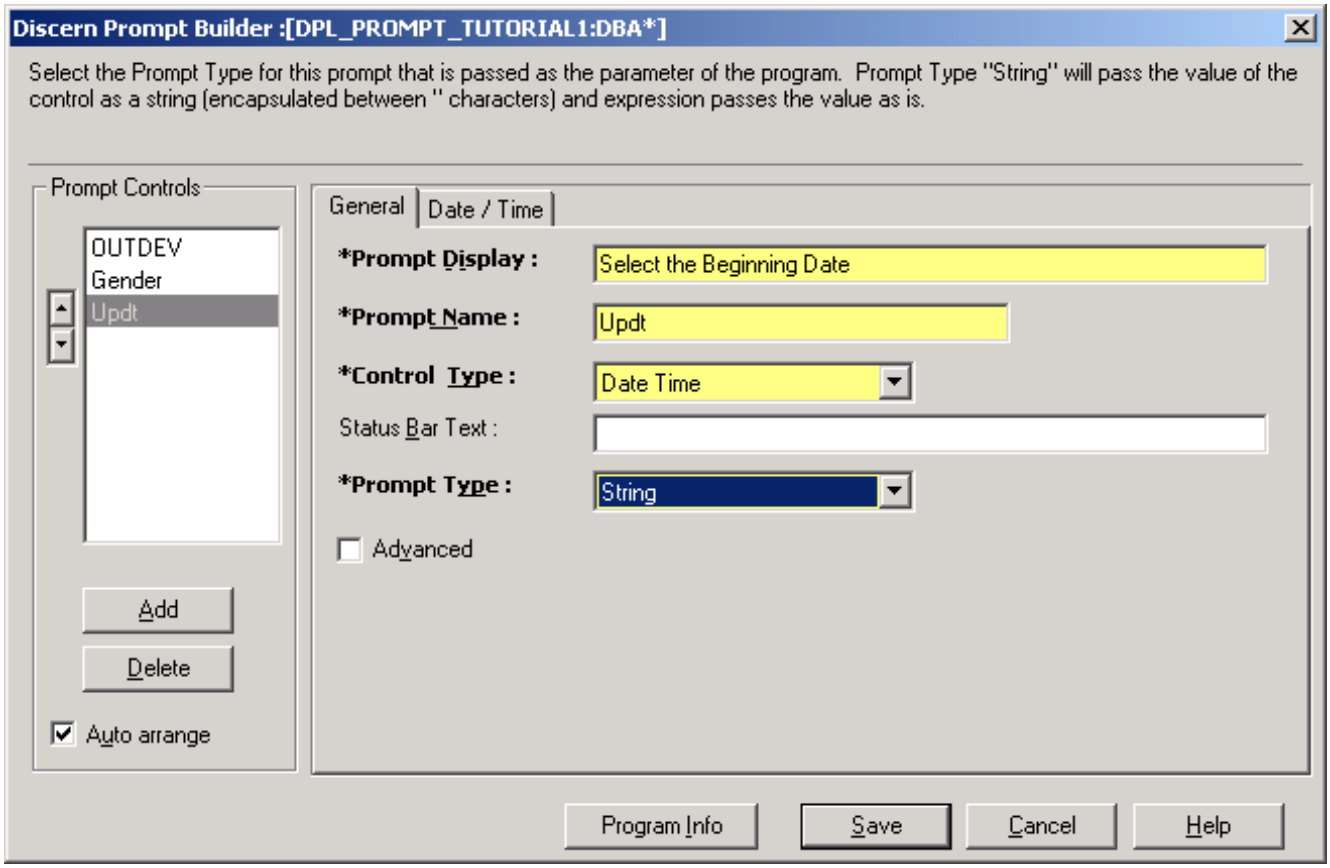
Open Discern Prompt Builder and click the Add button on the lower left side of the Prompt Control list to add a third prompt to the form.

For the Prompt Display use Select the Beginning Date.

For the Prompt Name use 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 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 or time will be formatted when it is 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 will allow 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 will be used to fill in the date control when no date has been specified.

+MM/DD -MM/DD Sets the default date that will be displayed in the control by adding or subtracting the number of days and or months from the current date. +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 on the keyboard. 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 will be used by the spin box when selecting a time. The time will advance or retract 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 the N key.

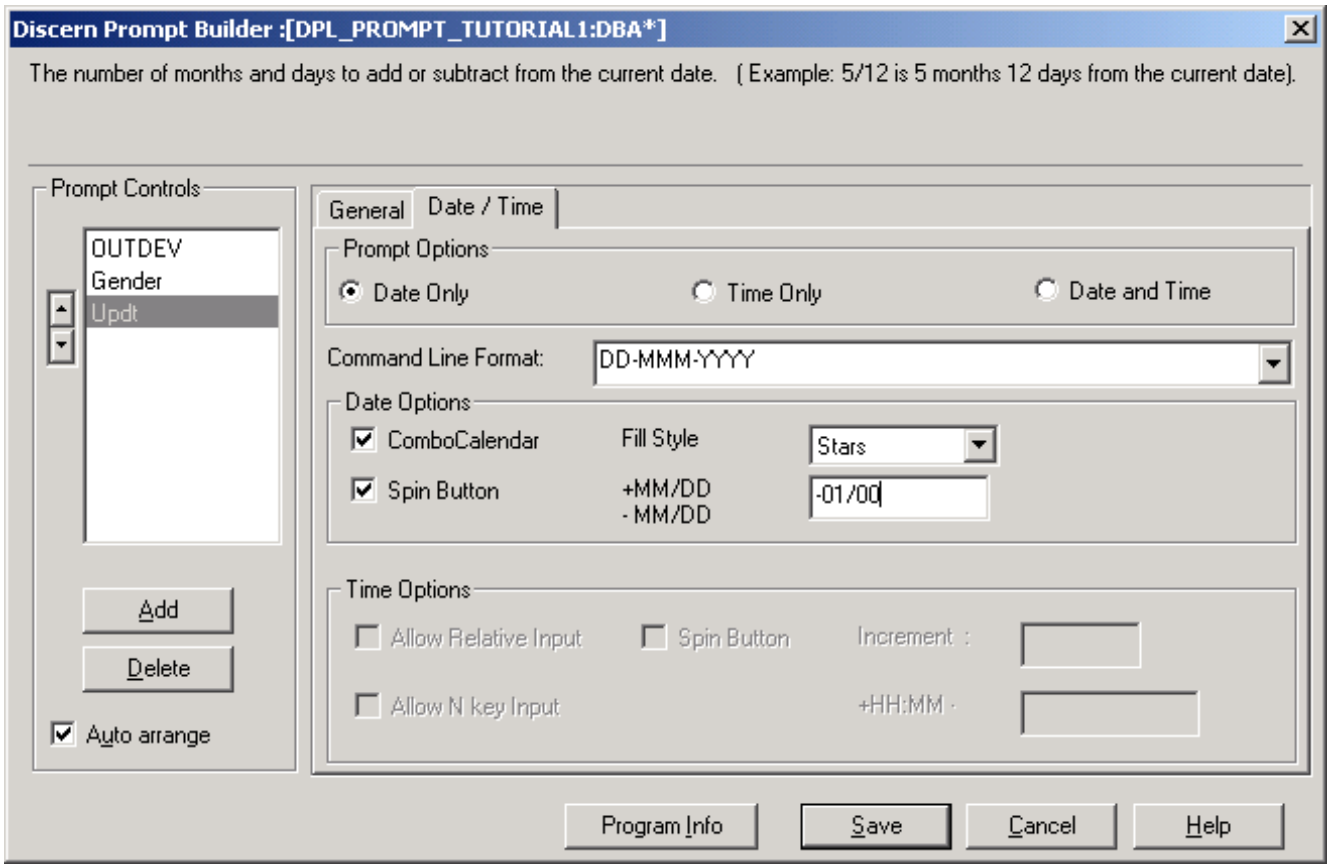
+HH/MM -HH/MM The Plus/Minus (Hours:Minutes) box allows you to enter the number of hours and minutes to add or subtract from the current time (for example, 05:30 is 5 hours and 30 minutes ahead of the current time and -05:30 is 5 hours and 30 minutes prior to the current time).

In our program we want to give the user the ability to qualify records on the person table that have been added or updated on or after a specified date. We want the default date to be one month ago.

On the Date / Time properties page 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.

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



Click the Save button to save your form.

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 will be passed to our program as a string in DD-MMM-YYYY format. This string will be 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 will take a string in DD-MMM-YYYY format, assume zero as the time, and create 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.
```

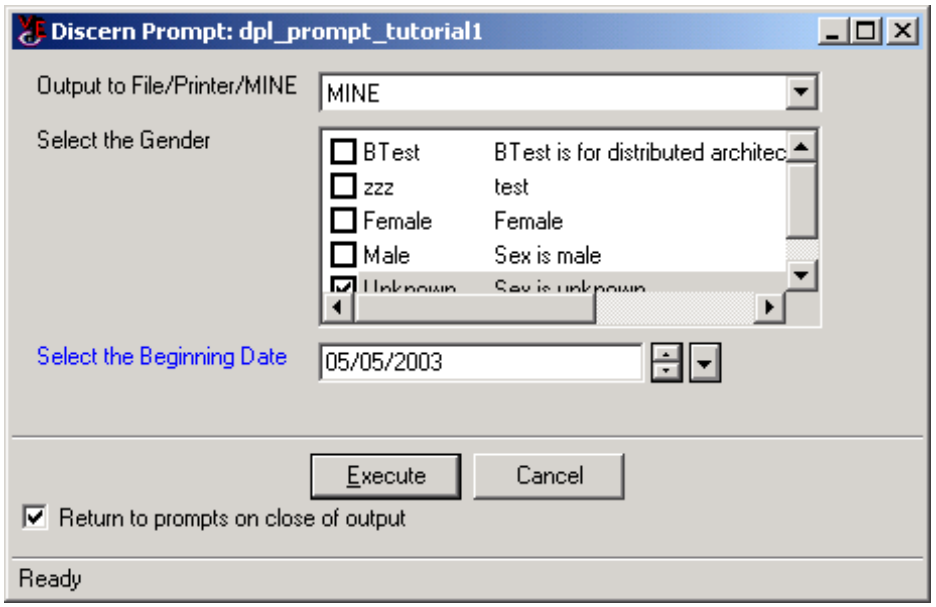
Your qualifications should now be

```
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 the Run Prompt Program option on the Build menu.

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

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 the Run Prompt Program option on the Build menu.

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

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 frm.

Open Discern Prompt Builder and Delete the Updt prompt control by highlighting it and selecting the Delete button.

Add a new prompt control to the form using the Add button.

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

For the Prompt Name use Beg_dt_tm.

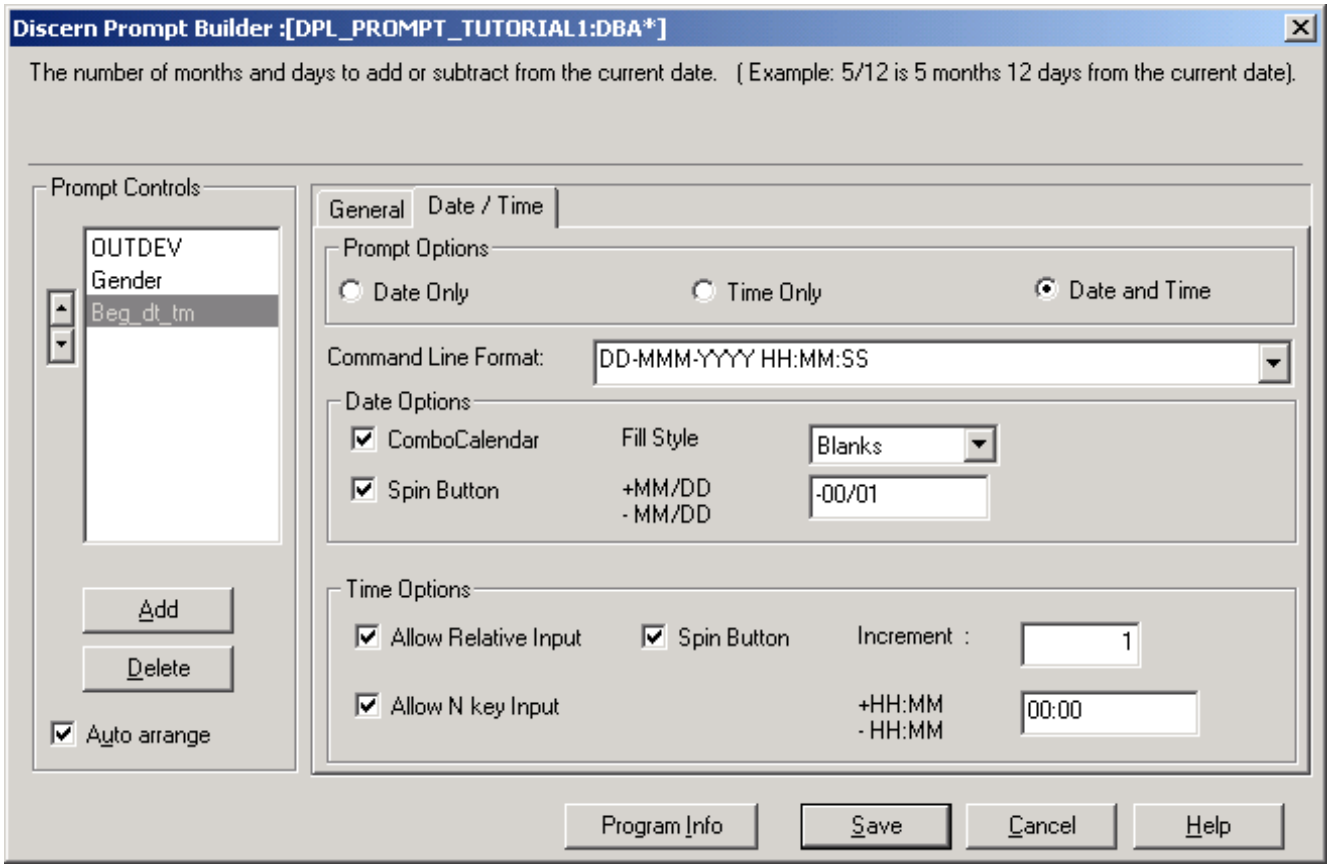
For the Control Type select Date Time.

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:



Add a new prompt control to the form using the Add button.

For the Prompt Display use Select the Ending Date and Time.

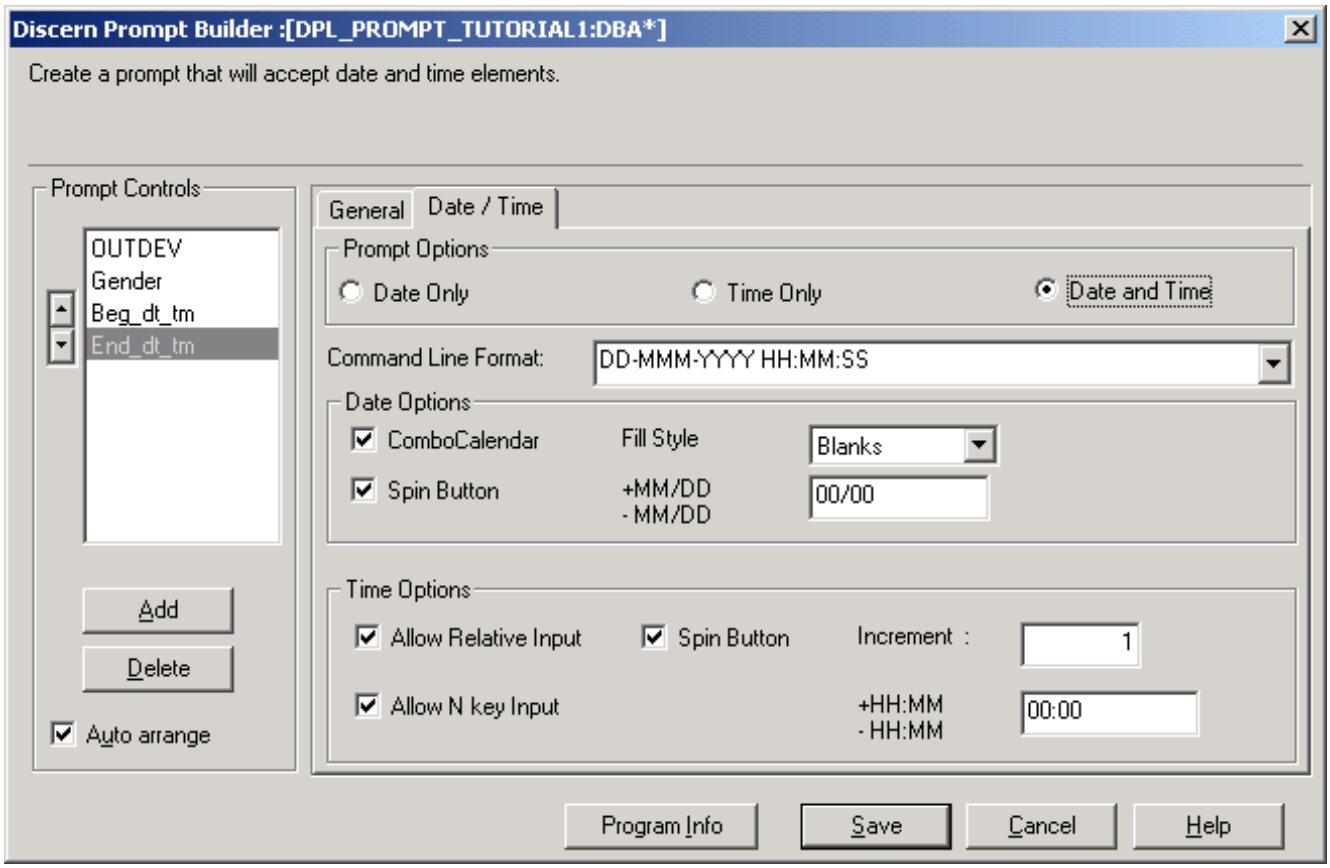
For the Prompt Name use End_dt_tm.

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



Click the Save button to save your form.

Change your qualifications from:

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

To:

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

If you are working in DVDev, include/compile your file, and execute your program using Ctrl+R or the Run Prompt Program option on the Build menu.

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

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 will 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 may 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 >= cnvtdatetime(cnvtdate($2),0)
```

in a qualification. If you were going to add a prompt form to a program that used this type of logic then 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/or 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.

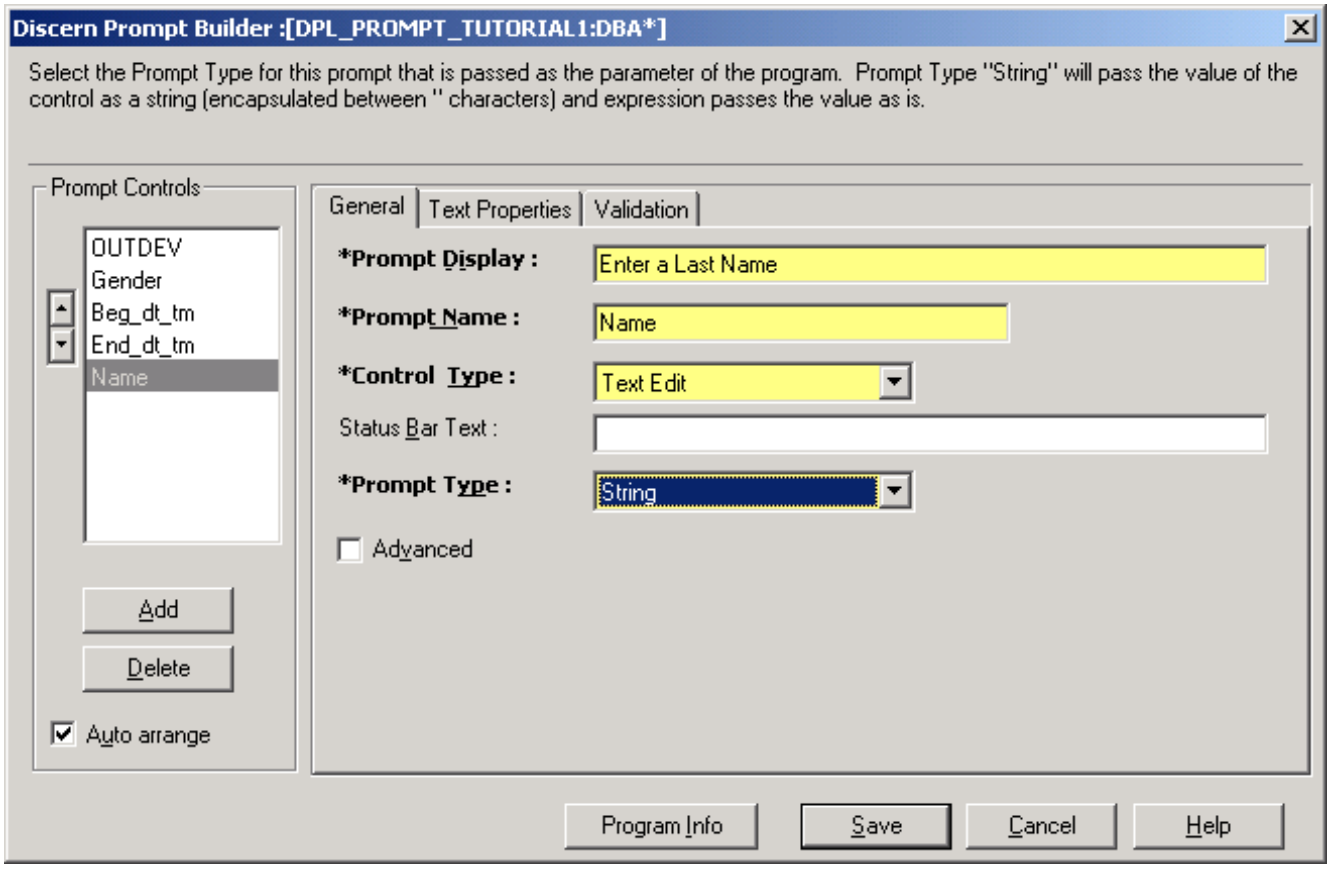
Open Discern Prompt Builder and click the Add button on the lower left side of the Prompt Control list to add a prompt to the form.

For the Prompt Display use Enter a last name.

For the Prompt Name use Name.

For the Control Type use the default Text Edit.

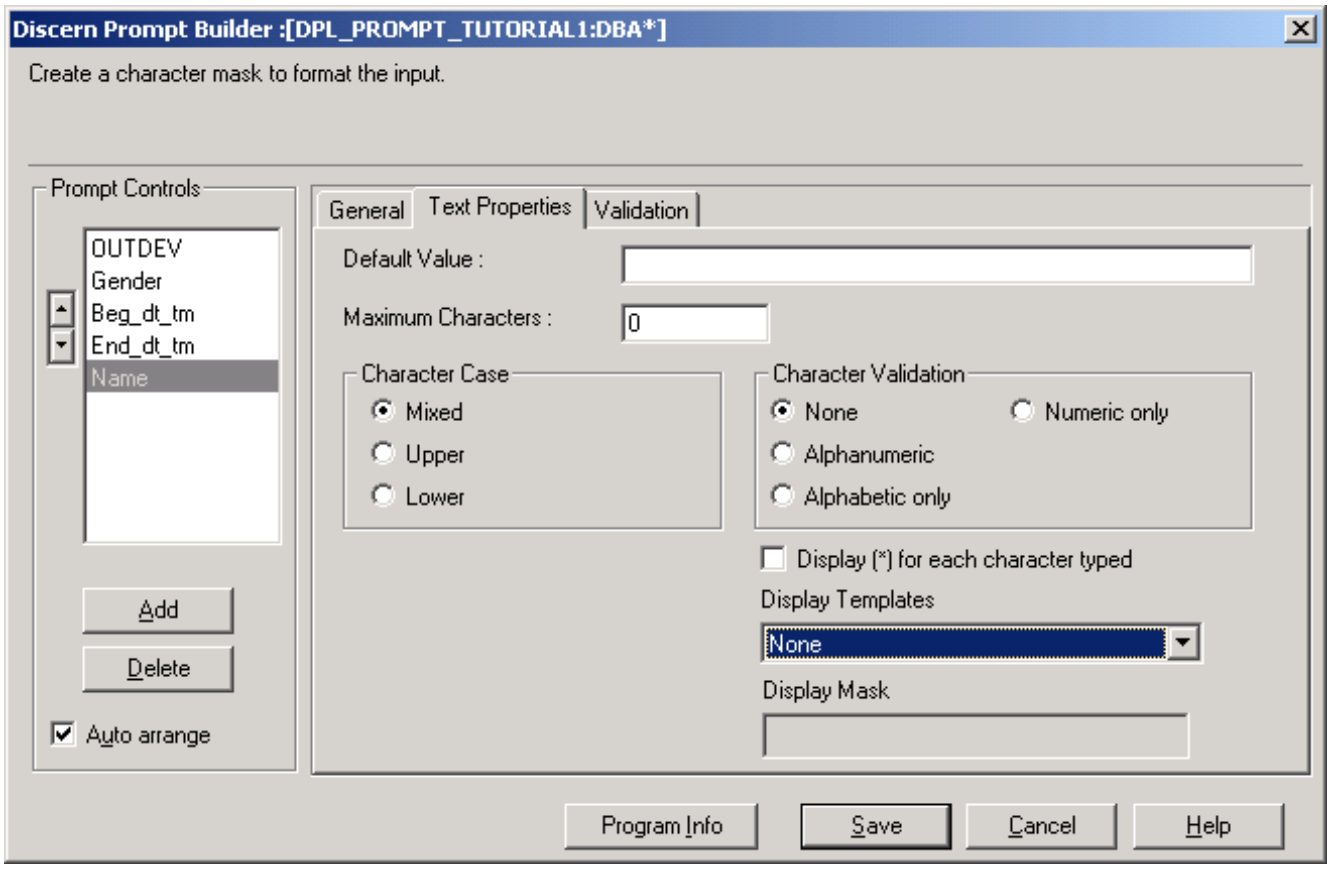
For the Prompt Type select String. Your Prompt Builder dialog should look similar to the following example:



The text edit control has two property pages: Text Properties and Validation.

Text Properties

Text properties define the runtime behavior of the edit box.



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

Maximum Characters. If 0 then the edit box will allow any length of text (although *Discern Explorer* will only accept 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, Upper the letter is converted to upper case, and 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 will accept letters and numbers only. Special characters are ignored. Alphabetic only will accept only letters. Numeric only will accept digits only. For this control the underscore is considered a letter.

Display(*). 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 will show as 0012 and if they then press the delete key it will show as 0001.
*	Hide character typed with *. Same as the *Display (*) 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 "\\".

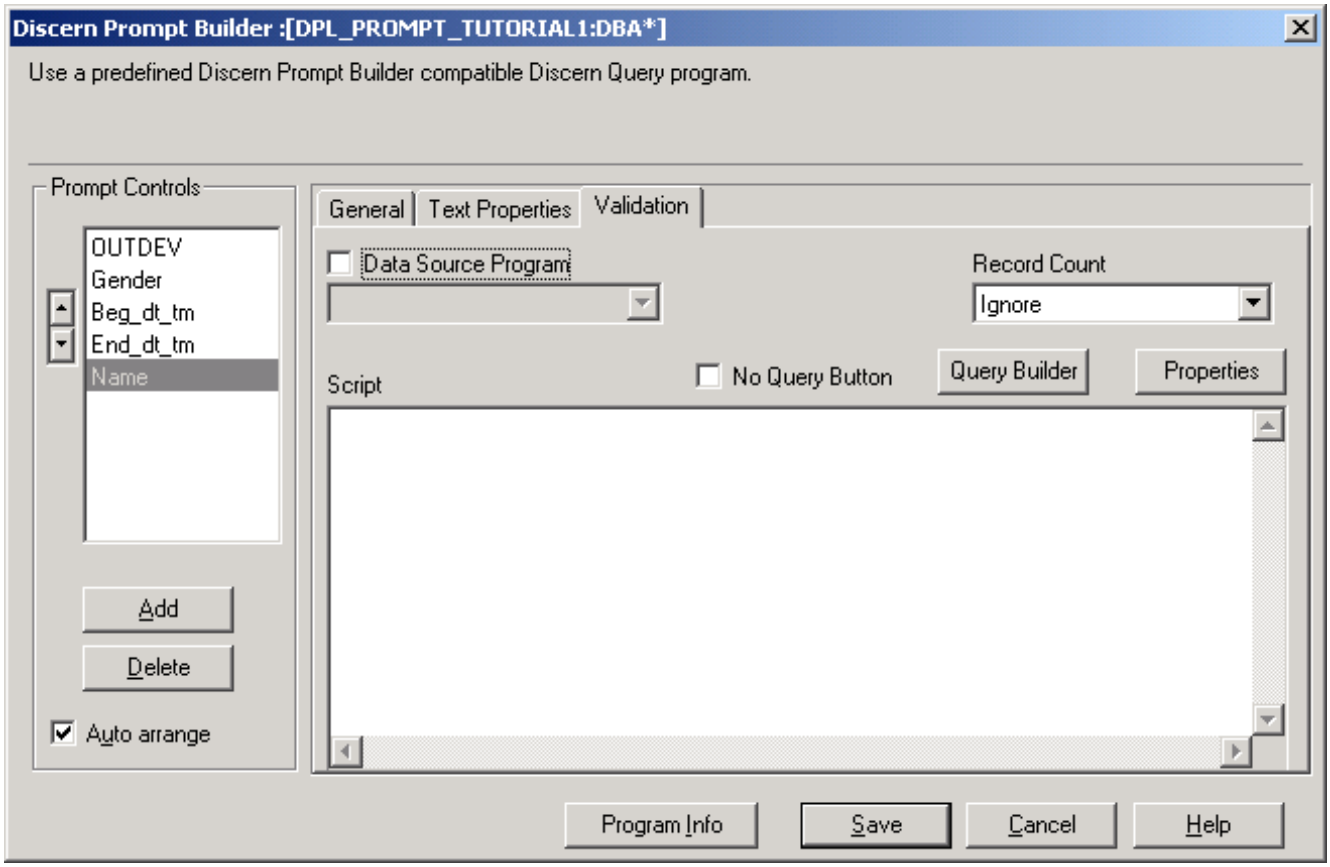
Any character that is not a mask character or the escape character is interpreted as a literal character mask. Literal characters in the input/display are skipped over and are not editable.

Numeric Input:

When the Character Validation is Numeric Only additional fields will appear in the property page. This allows the DPL programmer to input a lower and/or upper limit. If one of the ranges is not needed then 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. None sets an error if the SELECT returns any records. One and only one, sets the error flag if the SELECT returns no records or more that than one record. And 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 will turn red with white text. An error message is also sent to the status text line.

Script. Contains a *Discern Explorer* SELECT statement that will be 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 etc).

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 will be imported into the query builder. When the query builder is closed the select statement will be written to the script area.

Properties Button. The Properties button opens the Column Properties dialog box. Columns properties describe how a dataset field will be handled in the control at runtime

Validation Scripting

Validation is performed by the Text control when the control looses focus (either the user tabs out of the control or clicked out). 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 criterion 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.

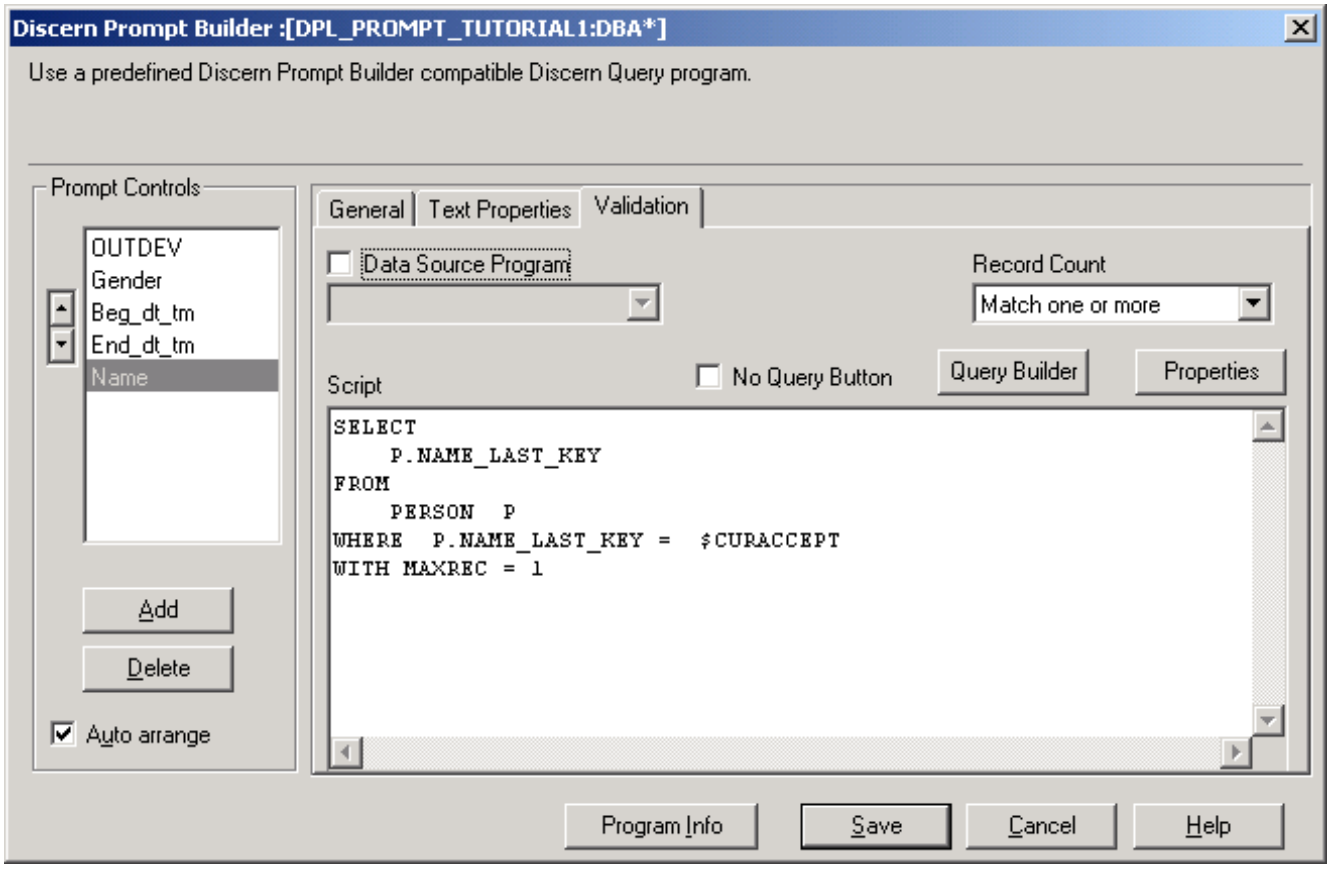
Click the Text Properties tab and select Upper in the Character Case area to have the control convert what the user enters to upper case.

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 the Save button 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:

If you are working in DVDev, include/compile your file, and execute your program using Ctrl+R or the Run Prompt Program option on the Build menu.

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

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 will contain rows and columns of data. One of the columns will be designated as the Key column. At runtime the user will select at least one of the rows from the dataset. The value in the key column of the selected row or rows will be 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 page.

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 property page.

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.

Use File>Include/Compile in VE or Build>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 Tools>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.

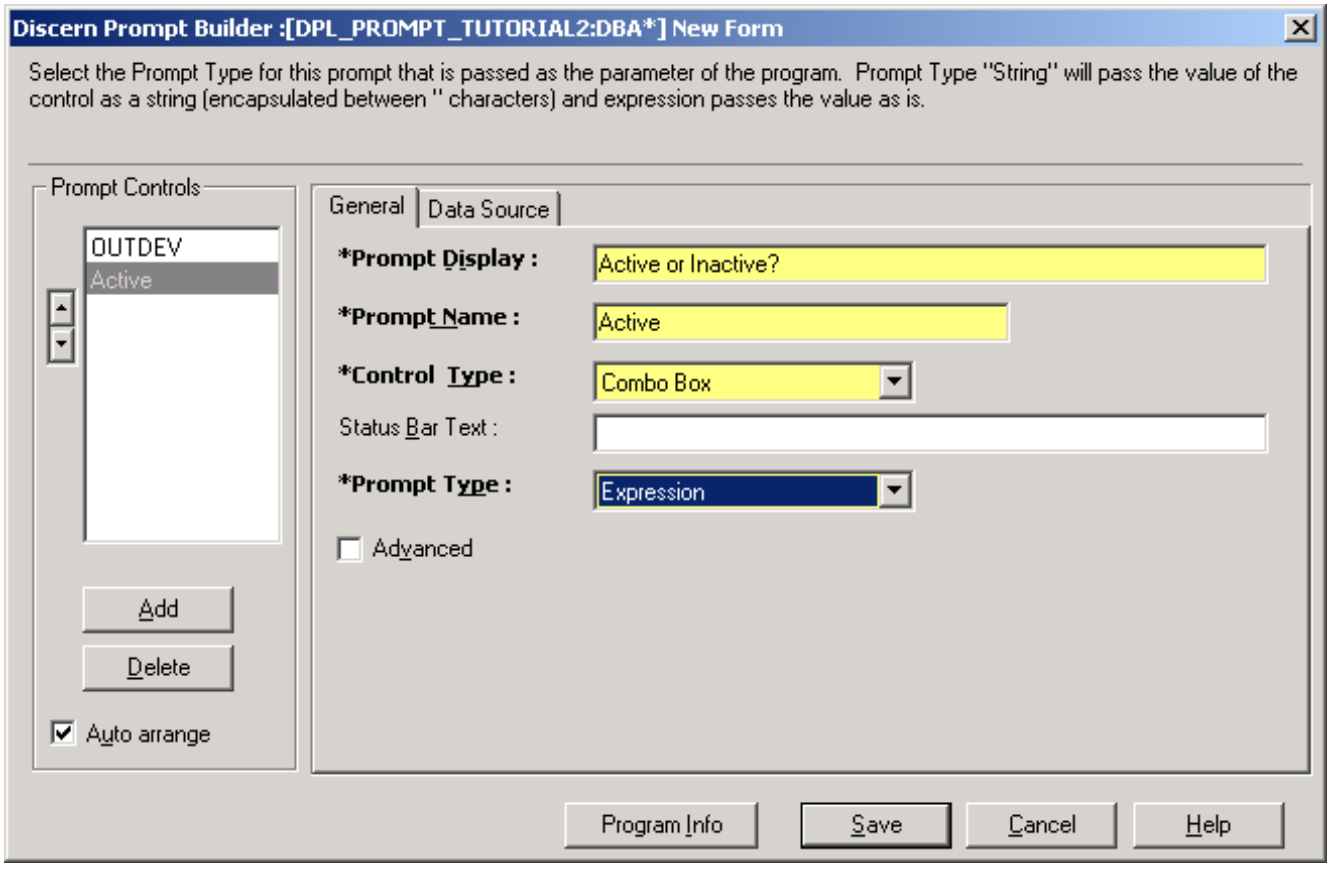
Click the Add button on the lower left side of the Prompt Control list to add a prompt to the form.

For the Prompt Display use Active or Inactive?

For the Prompt Name use Active.

For the Control Type choose Combo Box.

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

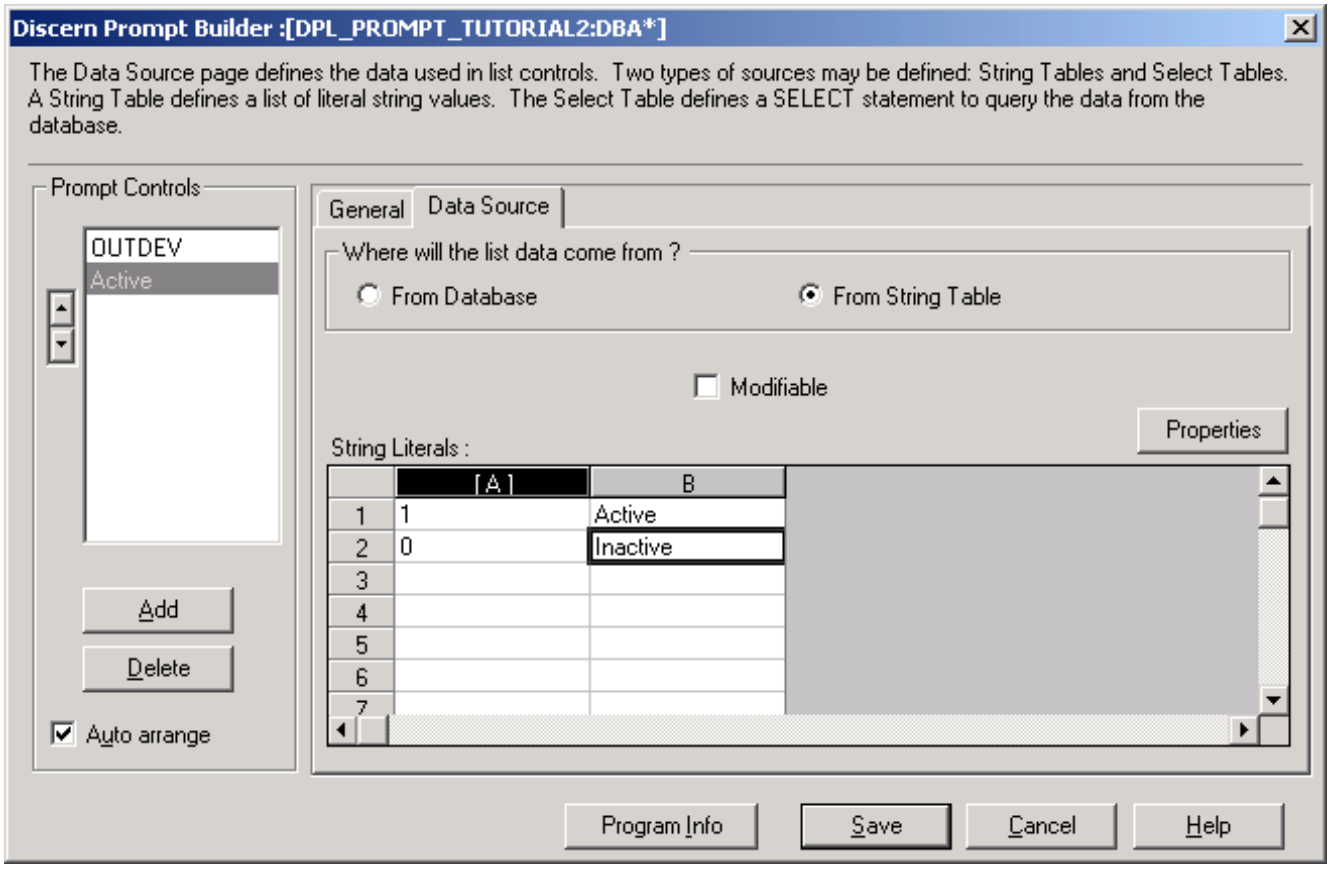


The Active_ind field on the orders table will be set to 1 (one) or 0 (zero) to indicate if the row is active or inactive. We want to set our prompt up so that at run time the user can select Active or Inactive from a list. Then based on their selection the DPL form will pass either a 1 or 0 to the program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

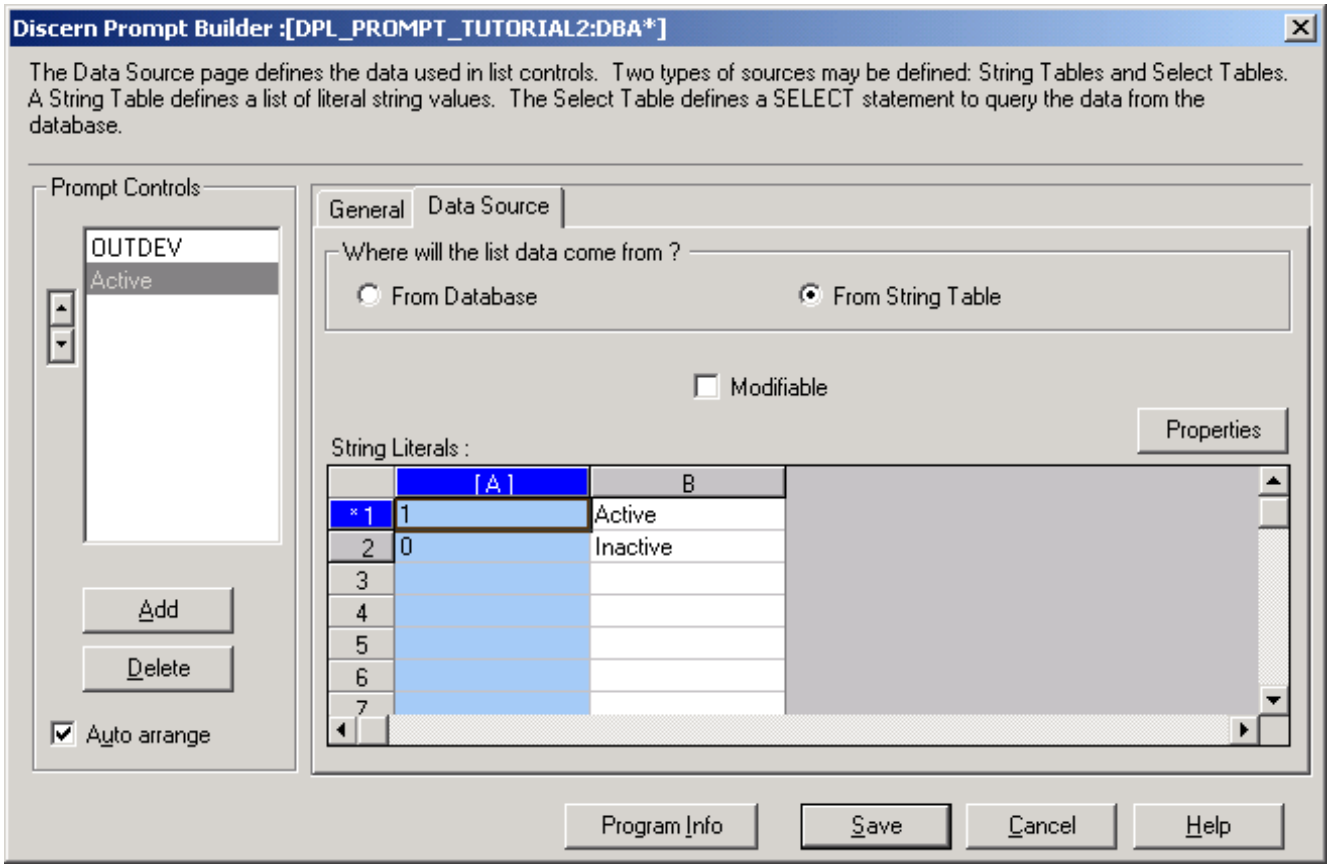
In row 1 of the String Literals table enter 1 under column A and Active under column B.

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



Open the context menu by right clicking somewhere in column A and select Hide from the Column menu.

Open the context menu by right clicking somewhere in row 1 and select Set Default Row. Your Prompt Builder dialog should look similar to the following example:



When a string table is used as the dataset for a combo box control, the first visible column will be 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 will select a value from the drop down list. The value in the key column from the row the user selects will be passed to the

Discern Explorer program. So in our case when the user selects Inactive from the drop down list, 0 will be passed to the program.

Click the Save button 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 the Run Prompt Program option on the Build menu.

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

Validate that the query is returning 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.

Use the Prompt Builder icon on the toolbar or Tools>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 will delete 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.

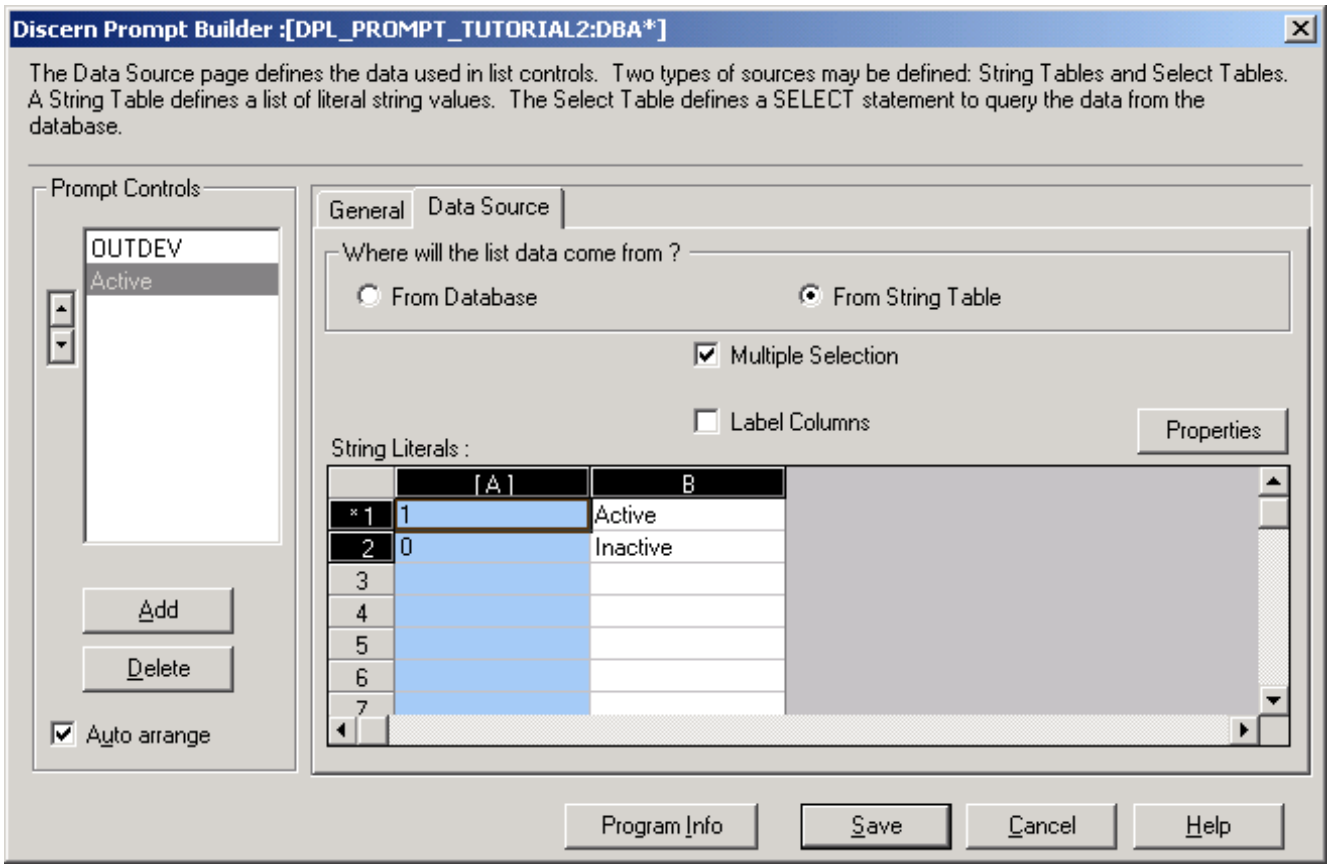
Check the Multiple Selection check box.

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.

Open the context menu by right clicking somewhere in column A and select Hide from the Column menu.

Open the context menu by right clicking somewhere in row 1 and select Set Default Row. Your Prompt Builder dialog should look similar to the following example:



Click the Save button to save your form.

If you are working in DVDev, execute your program using Ctrl+R or the Run Prompt Program option on the Build menu.

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

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 may need to modify you qualifications to display active and inactive orders. For example you may 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 will flex for combo and list boxes.

Modifiable: Available on combo boxes. Allows the end 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 will be passed as is.

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

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

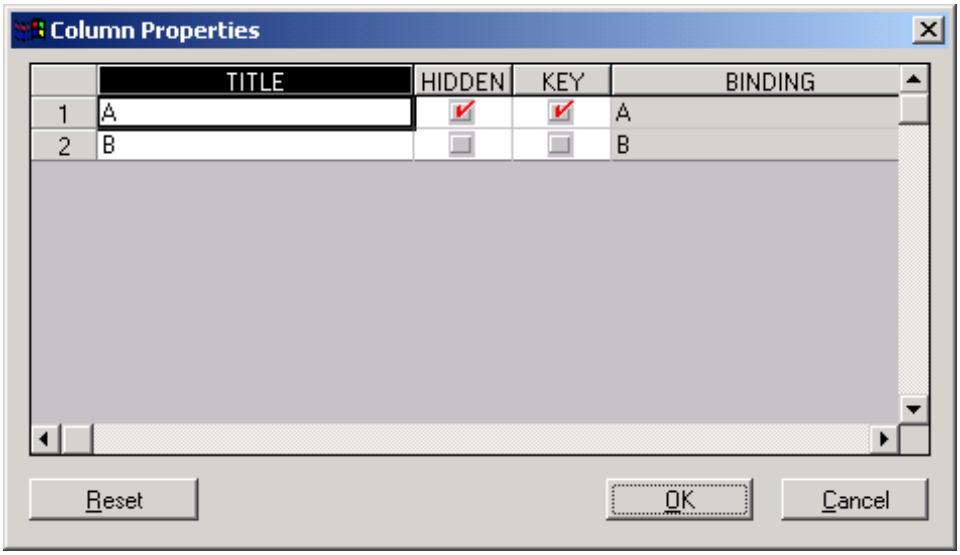
String Literals. The literal values that make up the dataset.

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

Setting the default row. The default row is indicated with * left of the row number. To set the default row, highlight the row, open the context menu and then 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. Columns properties describe how a dataset field will be 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 will be the heading of the column.

Hidden. If checked the column will not be shown in the control.

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

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

The Reset button 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 and open the context menu, select Column and then 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 discus 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 Tools>Prompt Builder to open the Discern Prompt Builder.

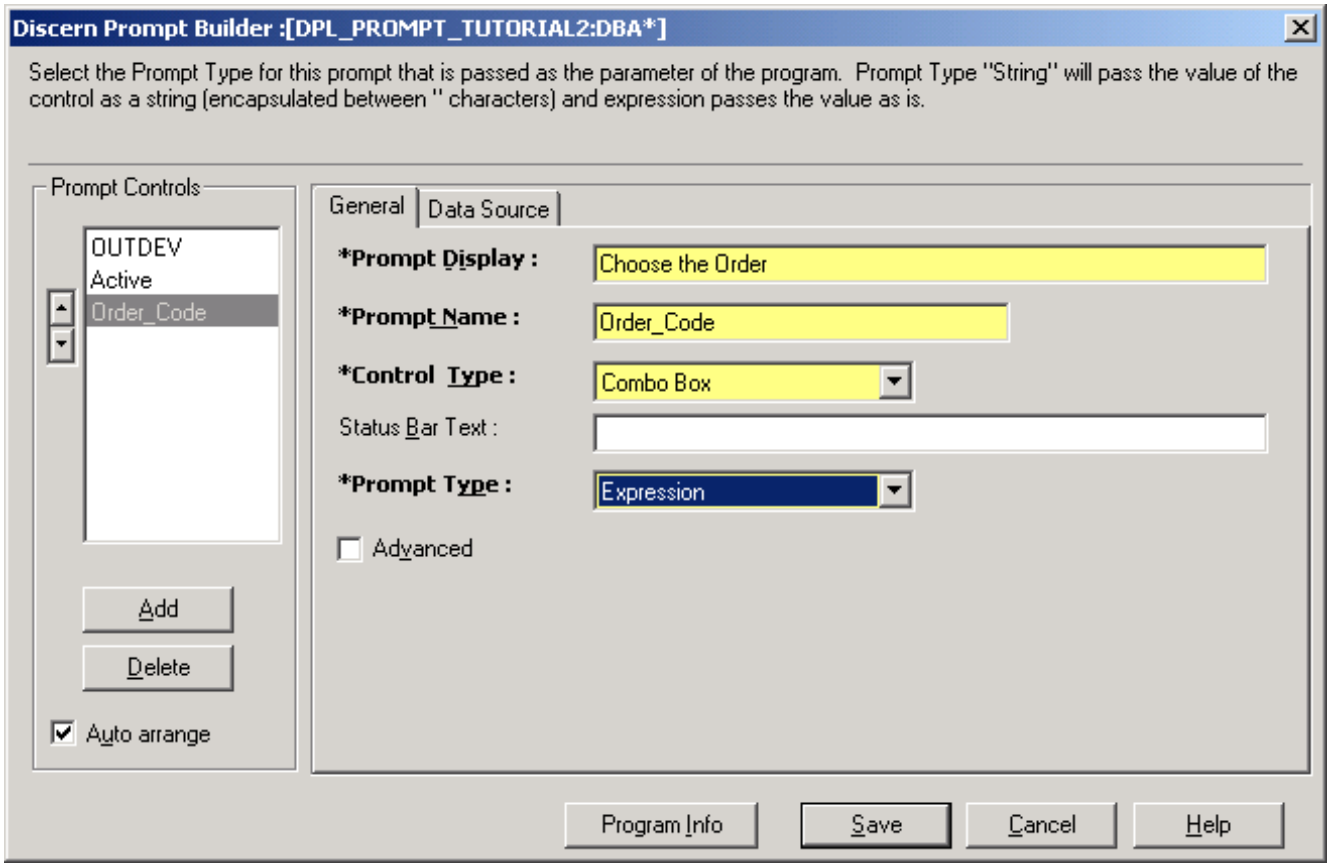
Click the Add button on the lower left side of the Prompt Control list to add a prompt to the form.

For the Prompt Display enter Choose the Order.

For the Prompt Name use Order_Code.

For the Control Type choose Combo Box.

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



Open the Data Source property page by clicking the Data Source tab.

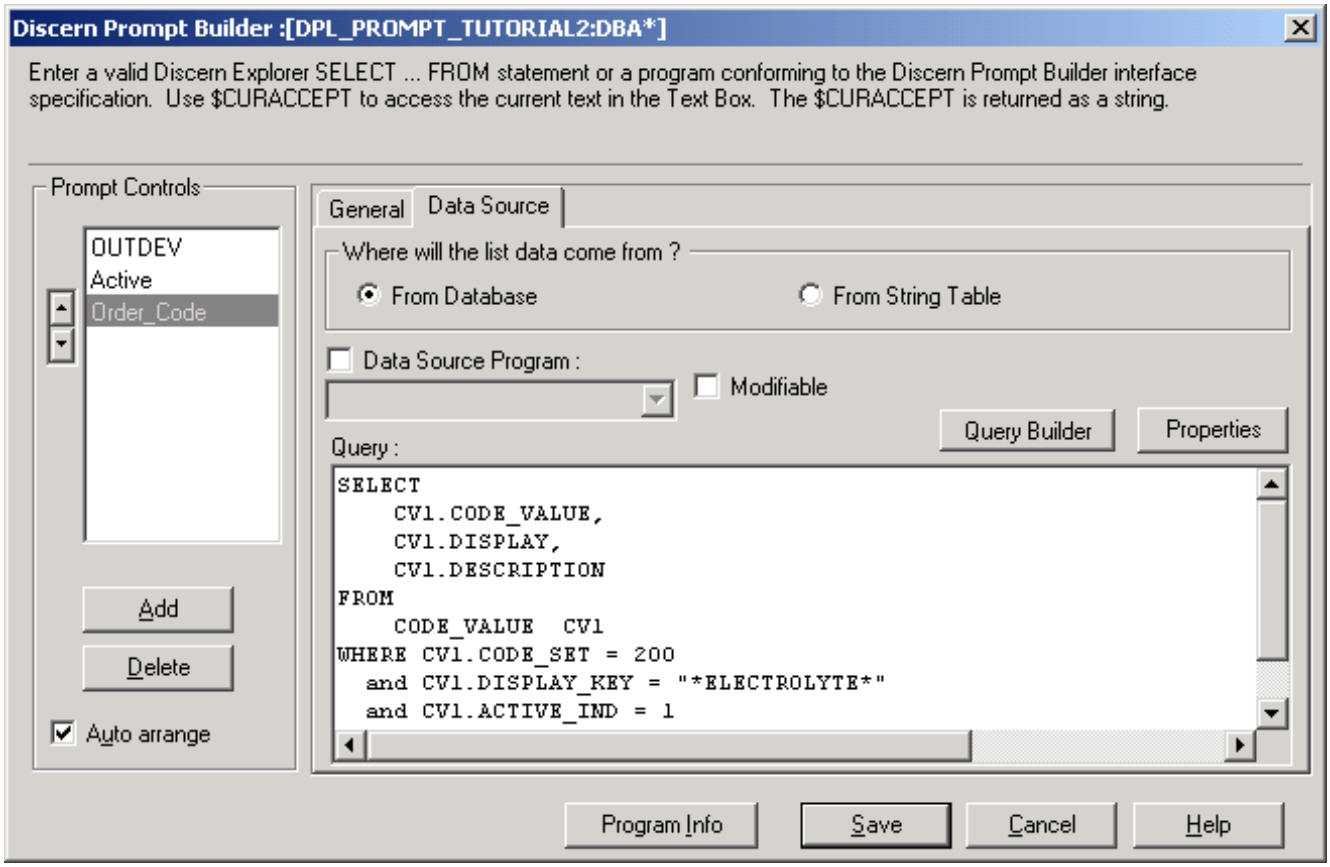
Check the From Database radio button.

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

```
SELECT
CV1.CODE_VALUE,
CV1.DISPLAY,
CV1.DESCRIPTION
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 )
```

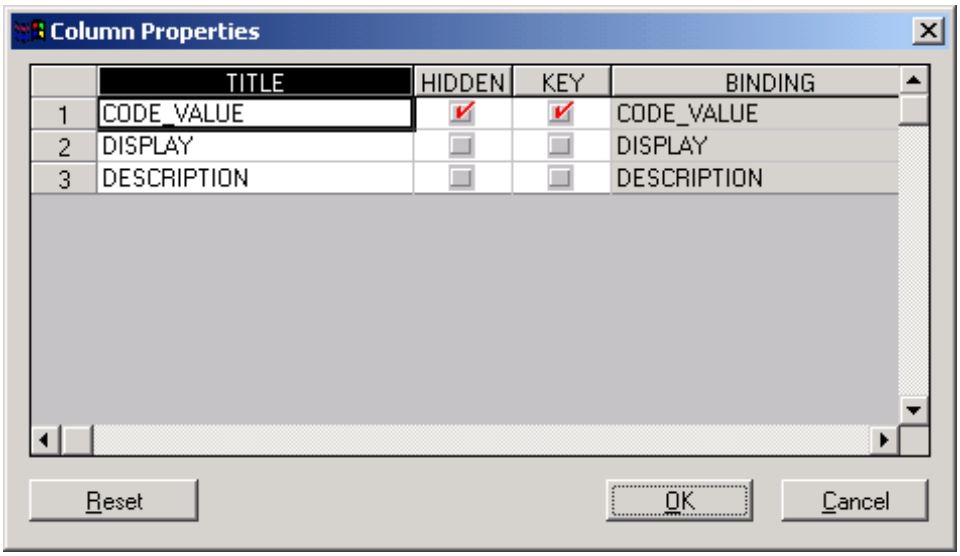
If you do not have any orderables in code set 200 with a Display_Key that contains the string ELECTROLYTE you will need to use a different string in your qualification.

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



Click the Properties button 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 will display 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 will be passed to your program. You now need to add the code to your program to use the Code_Value in the qualifications.

Use the OK button to close the Column Properties editor.

Use the 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 the Run Prompt Program option on the Build menu.

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

Validate that the query is returning the type of orders that 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.

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

Select the Data Source tab and click the From Database radio button.

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

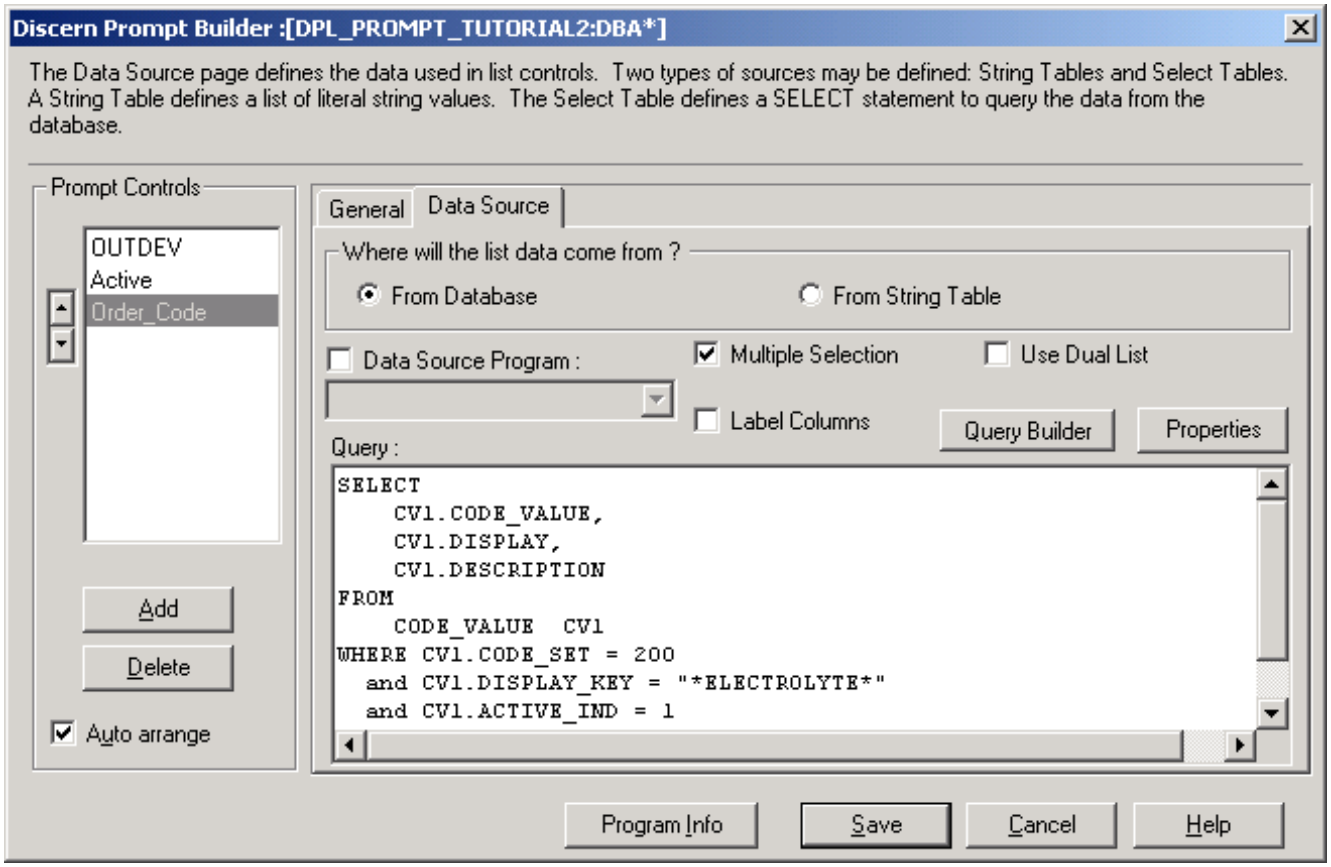
Note: When you changed the control type the properties for the control were reset back to the defaults, which erased the query.

Check the Multiple Selection check box.

Click the Properties button to open the Column Properties editor.

Use the check boxes to hide the Code_Value field and make it the key field.

Use the OK button to close the Column Properties editor. Your Prompt Builder dialog should look similar to the following example:



Use the Save button to save the changes to your DPL form.

If you are working in DVDev, execute your program using Ctrl+R or the Run Prompt Program option on the Build menu.

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

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 will flex for combo and list boxes.

Data Source Program: When checked the user may 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 discus how to create this type of program in this tutorial

Multiple Selection: For List boxes only. This field determines if the end-user will be 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).

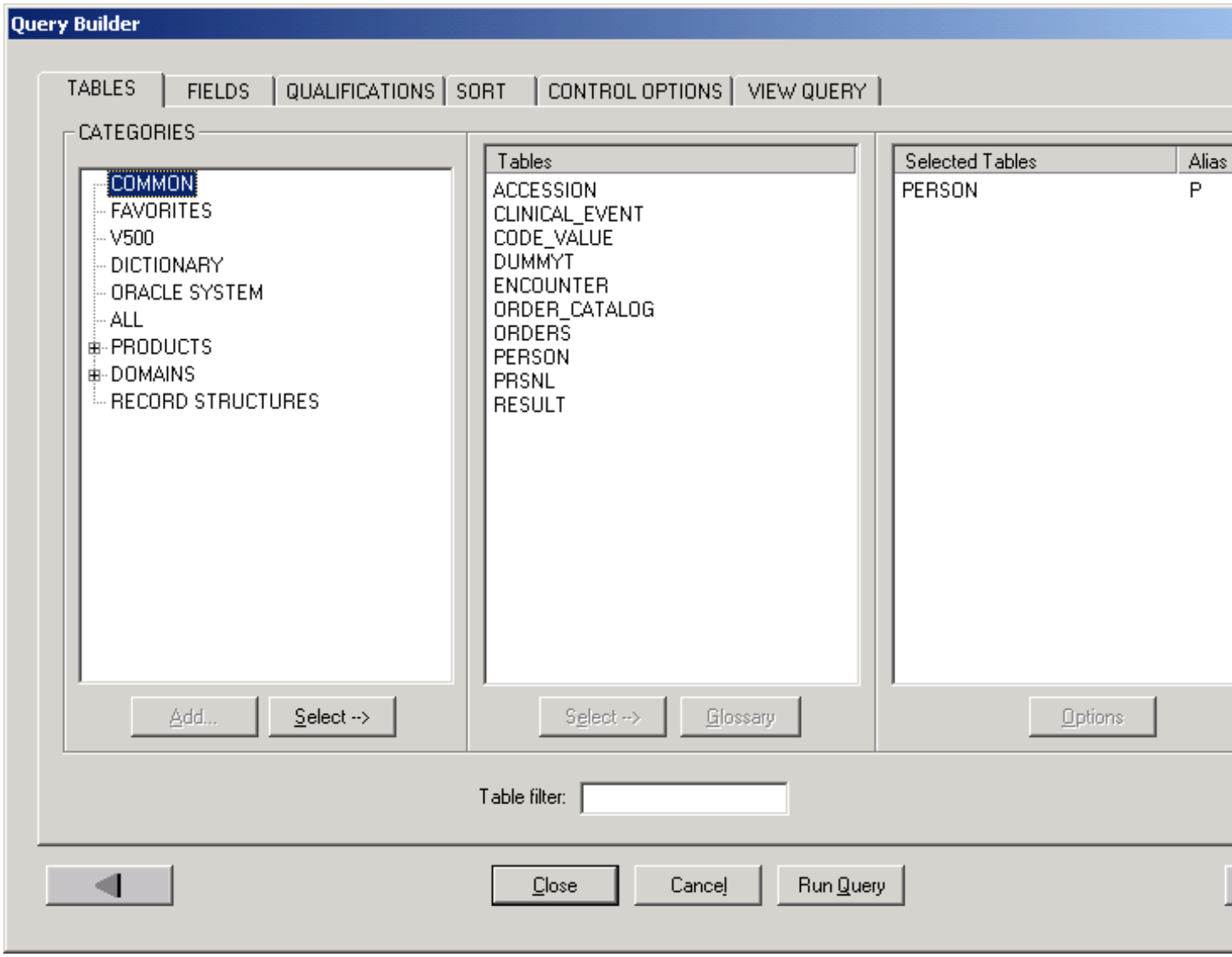
Modifiable: This allows the end user to enter free text data into the combo box edit field. If the user picks from the list the datasets key for that item is passed, however, if the user types text in the edit field the text entered is passed as is.

Label Columns. Available for List boxes only. Determines if the datasets column titles will be 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 will be 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 will be pulled into the builder.

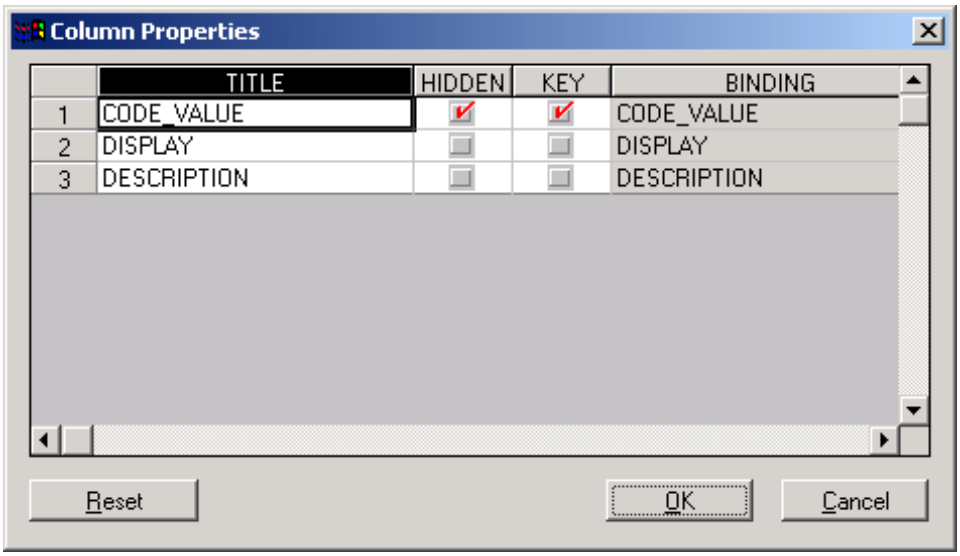


Query. Edit box to type the select or executable statement. Most of the *Discern Explorer* SELECT clauses can be used. SELECT INTO a table, file, or printer queue cannot be used. Report writer sections (HEAD REPORT/DETAIL etc) cannot be used.

Queries can use $CURACCEPT which will open the Query Viewer and prompt the user for a value. The control will use the value that is entered in the query to build a dataset to populate the list. Once the list has been 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 will be 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.



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 will be 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 may themselves be detail controls to another master control. Detail controls may 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 is displaying 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 and use the Add button to create a new prompt control.

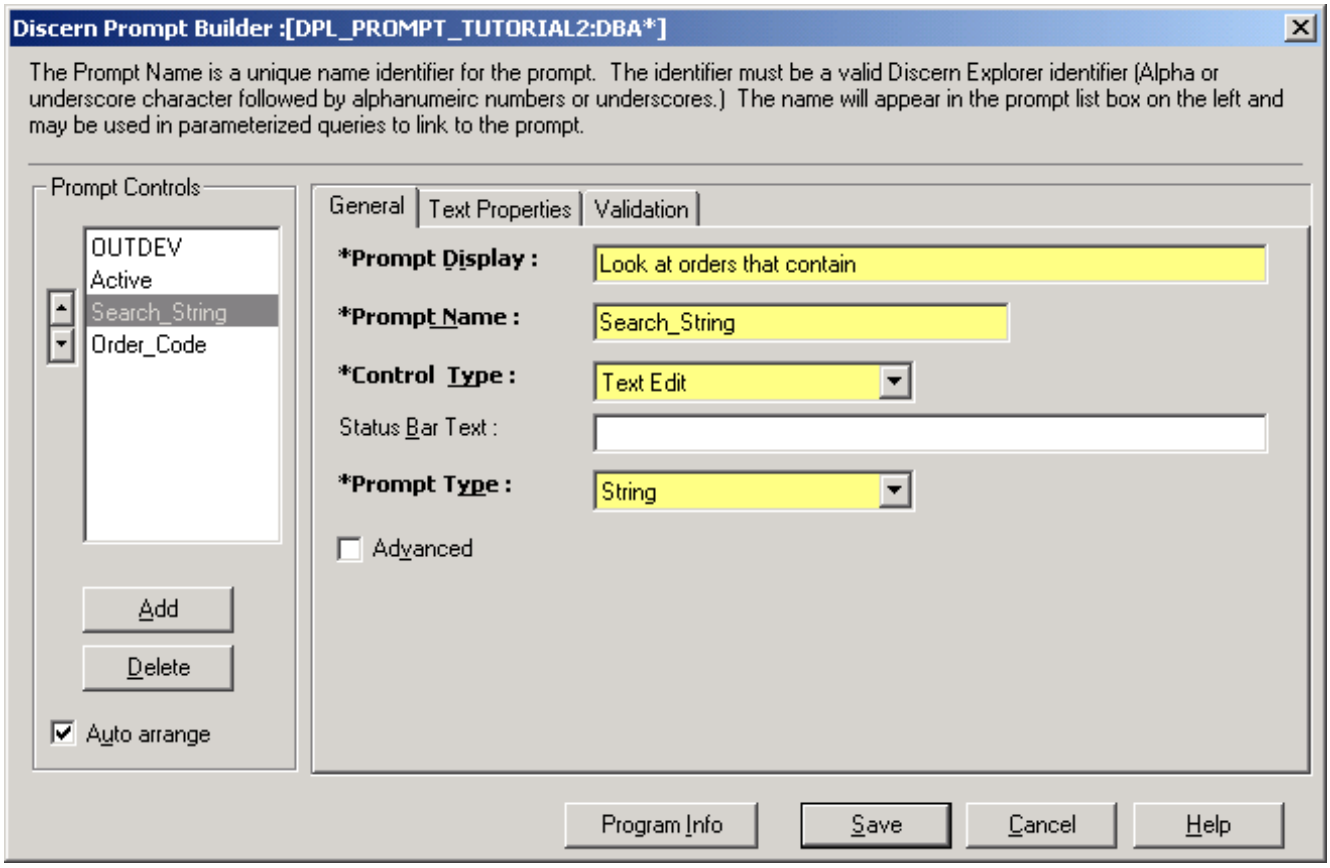
For the Prompt Display enter Look for orders that contain.

For the Prompt Name use Search_String.

For the Control Type choose Text_Edit.

For the Prompt Type select String.

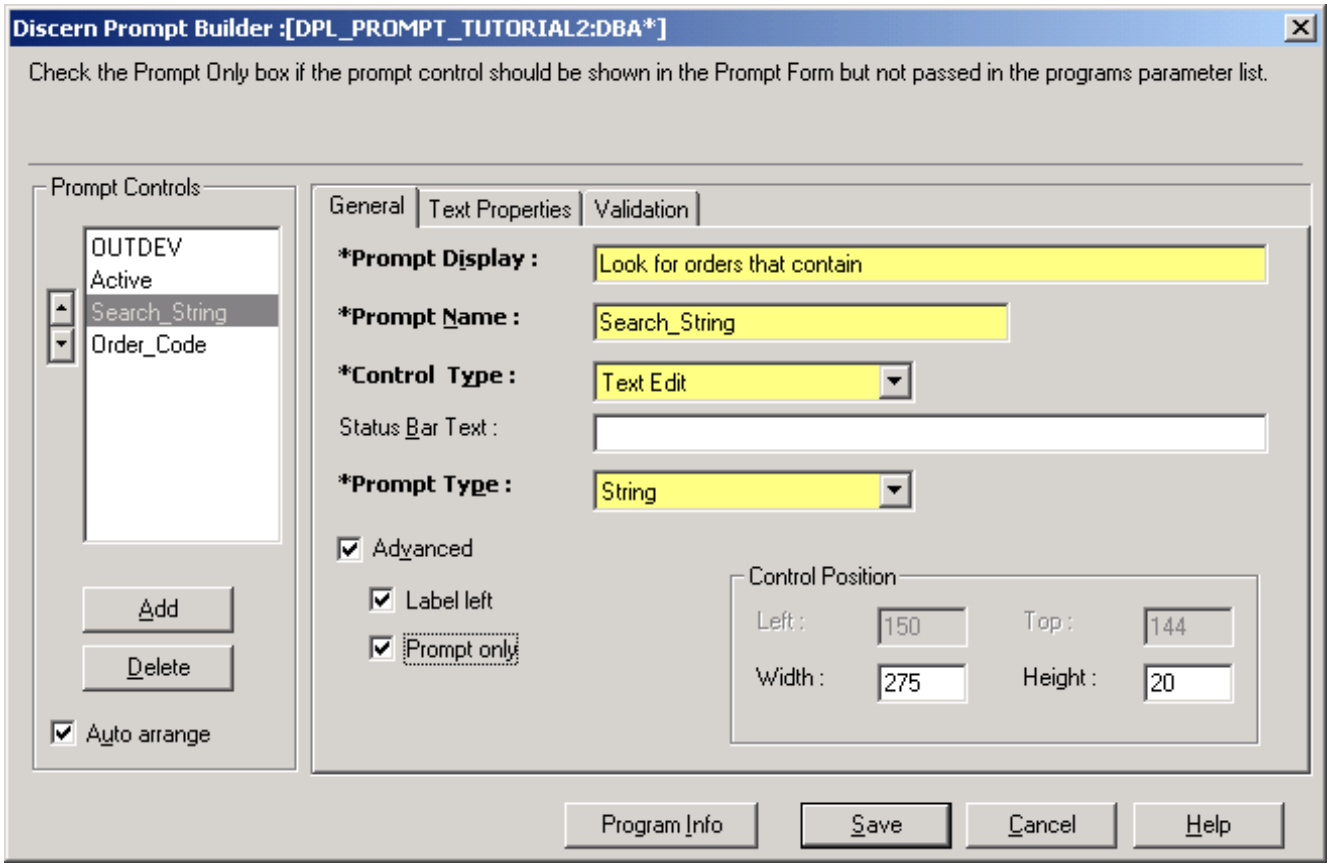
Use the Arrow buttons on the left of the Prompt Controls list to move the Search_String control above the Order_Code control. Your Prompt Builder dialog should look similar to the following example:



With all of the prompt controls that 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 check the Advanced check box to display additional properties.

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



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 will change based on what the user enters.

Using the Text Properties page on the Search_String control set the Character Case to Upper. This will convert 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:

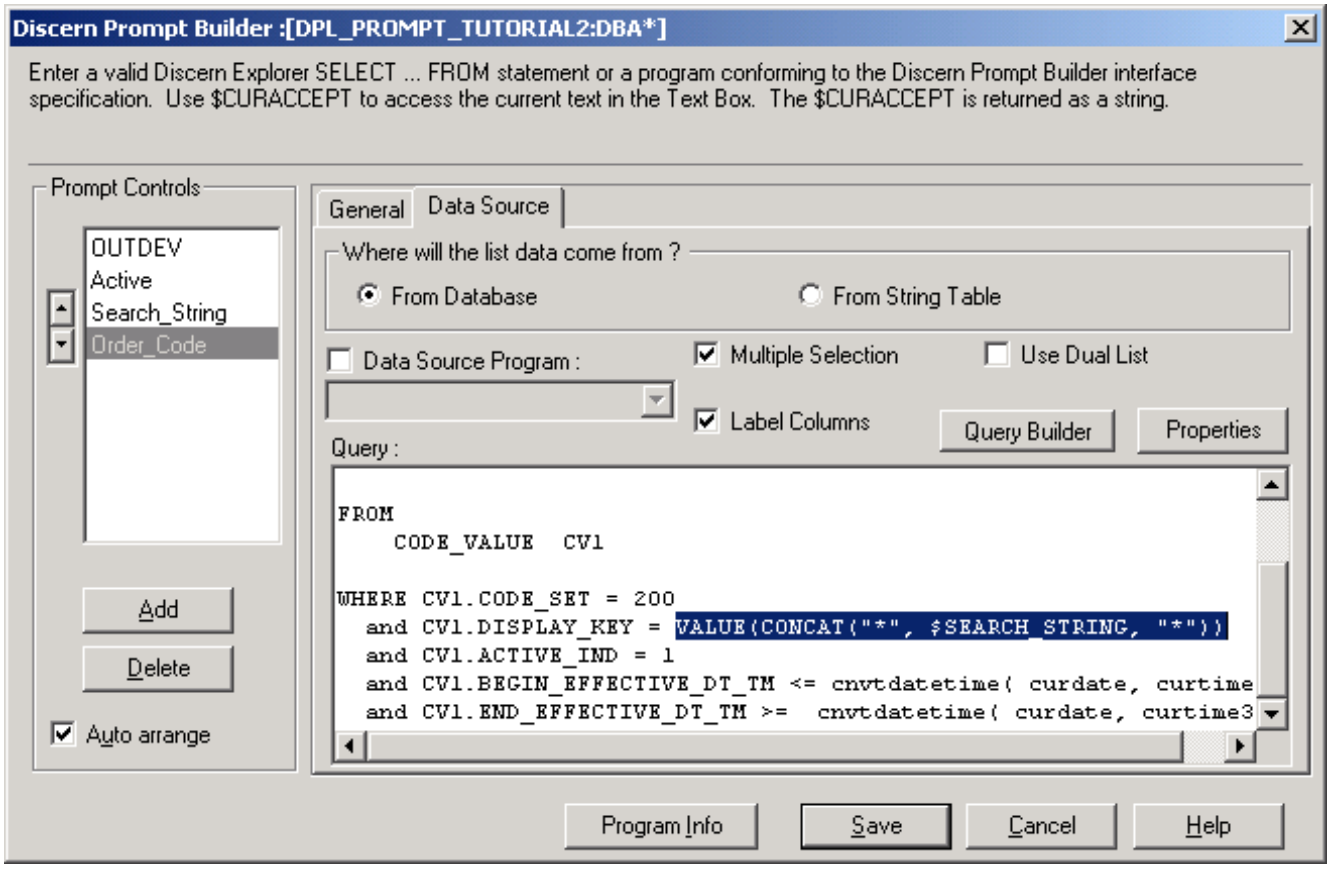
```
SELECT
    CV1.CODE_VALUE,
    CV1.DISPLAY,
```

```
    CV1.DESCRIPTION
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.DESCRIPTION
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:

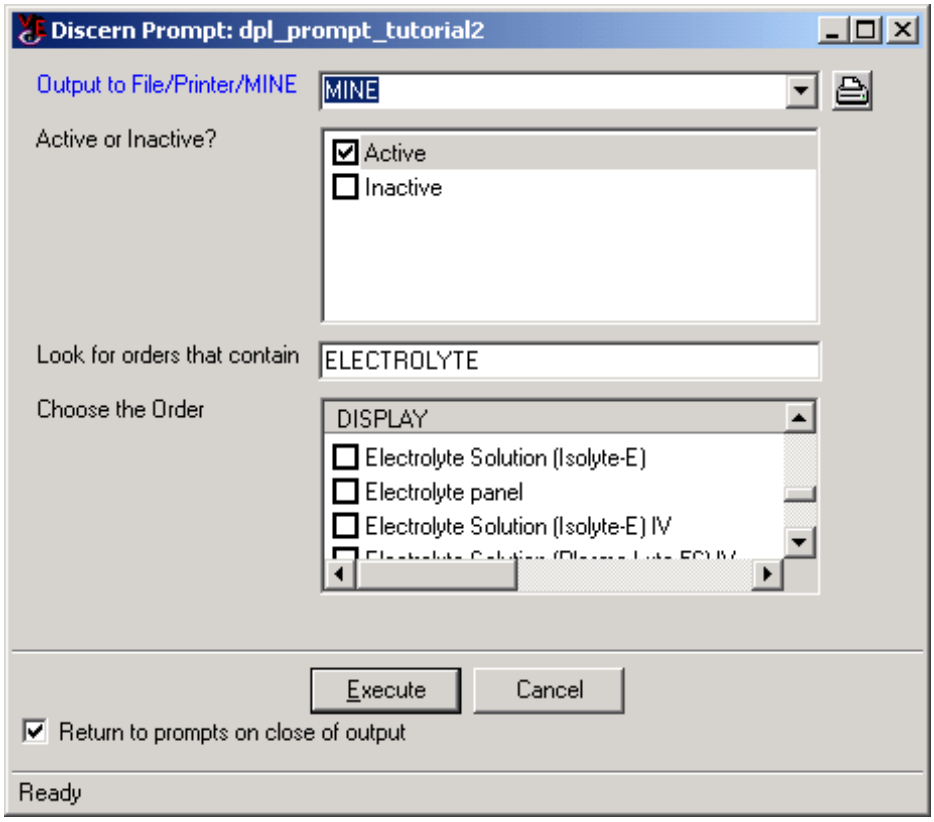


We are using the Concat() function to place the wildcard in front of and in back of the string the user enters in the Search_String control, and the Value() function to pass the result of the Concat() by value instead of by reference.

Use the Save button to save your DPL form.

If you are working in DVDev, execute your program using Ctrl+R or the Run Prompt Program option on the Build menu.

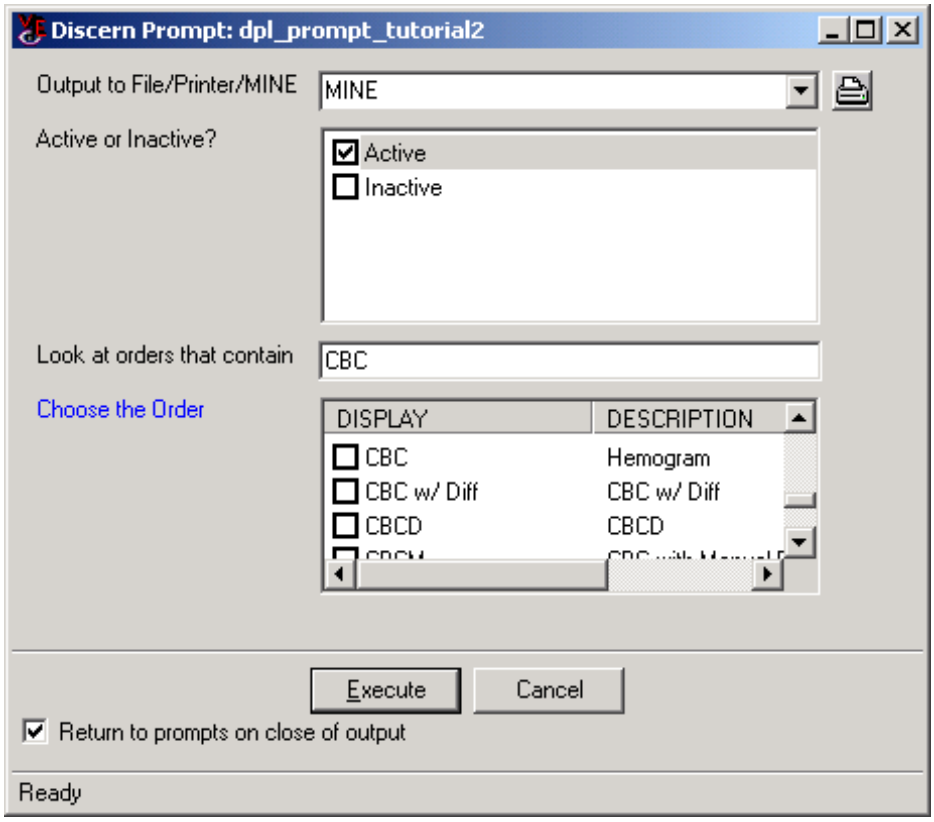
If you are working in VE, execute your program using the Run Query button. The DPL form that is display should look similar to the following example:



When a dataset is created for a List Box or Combo Box using a query, the query will be 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 will be 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 will 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 display should look similar to the following example:



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.

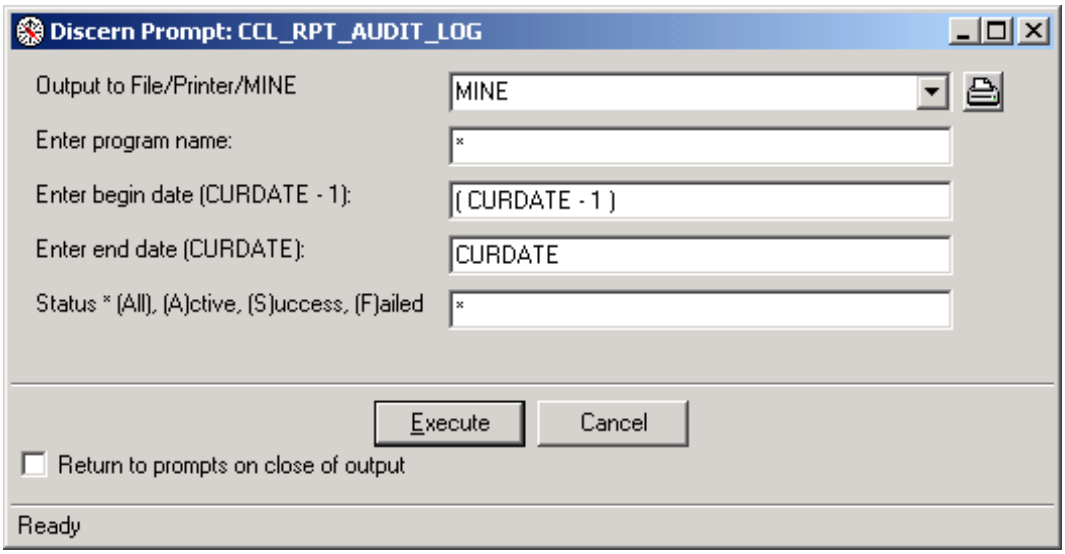
Audit Logs and Troubleshooting

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 the Execute button is selected, the DiscernReport.dll uses a script to execute your program and pass 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 will log 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 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. In order to write some information to the logs, open DVDev and use Ctrl+R or the Run Prompt Program option on the Build menu to execute your program.

In DVDev use Ctrl+R or the Run Prompt Program option on the Build menu to execute CCL_RPT_AUDIT_LOG. The following form should be displayed:



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

The information that is displayed by the report will include 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.

Document Revision History

Revision Number:	Revision Date:	Description:
001	October 2003	Initial release