- 1 Discern Explorer Layout Builder Tutorial
- 2 Creating HTML Output

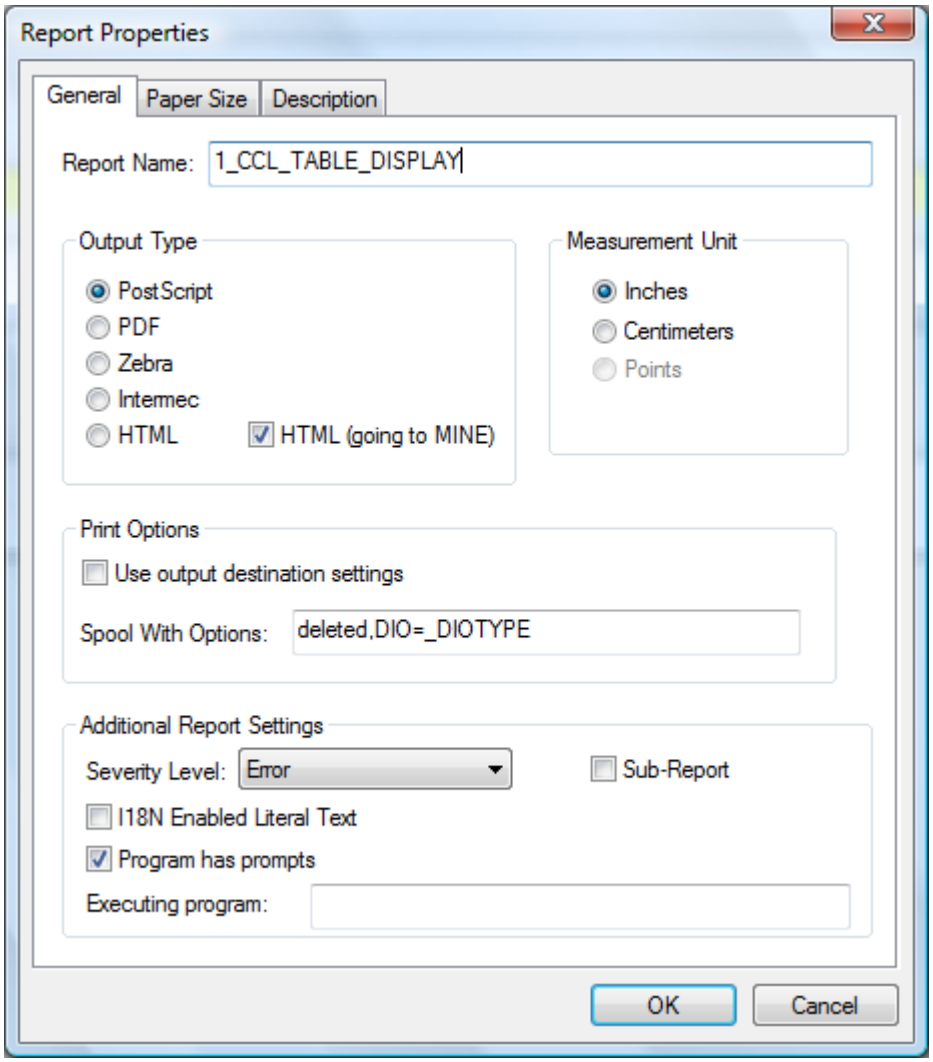
Discern Explorer Layout Builder Tutorial

Creating HTML Output

This section assumes you have already completed the Creating a Table View section of this tutorial.

There might be times when you want to create a report that supplies high level information, but then gives you the ability to navigate down to a finer level of detail. Layout Builder enables you to create Table Views and layout programs that produce output in HTML format. Using HTML format enables you to create links which can then be used to execute further programs allowing the user to navigate down to see additional details. Using Table Views is the recommended type of layout to use when creating HTML output. Table Views have intrinsic tabular type logic which lends itself to creating HTML output. However, using a normal layout program is acceptable when consideration is given to the tabular type logic flow.

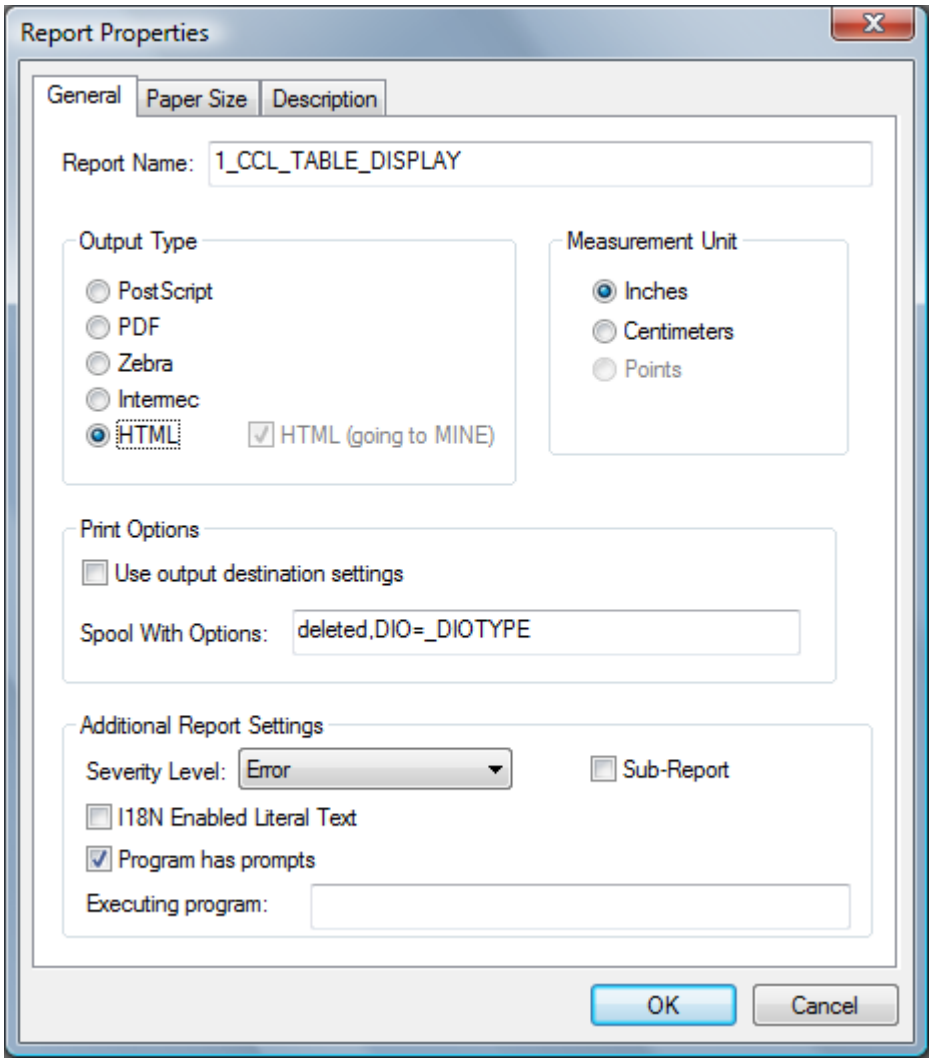
For existing layout programs, the Report Properties dialog box provides the option to create output in HTML table format.
One way to create HTML output is by selecting the HTML (going to MINE) option for the Report Property.



Using this method allows you to flex the output to different types of formats.
When a layout program is created, Discern Visual Developer (DVDev) creates a prompt form for the layout program. The prompt form contains a prompt for the output device. The output of the layout is sent to the output device. If a layout program has the Output Type option of HTML (going to MINE) selected and if MINE (in uppercase) is used as the output device, the output of the layout program is displayed on your screen in HTML format. If any value other than uppercase MINE is used as the output device, the output is sent to that device in the specified output format.

For example, if the PostScript Output option is selected and the HTML (going to MINE) option is checked and MINE is entered or selected at the prompt for output device, the output of the layout program is displayed on the screen in HTML format. If a PostScript printer queue name is used as the output device, the output of the layout program is sent to that PostScript printer queue in PostScript format. It is important to understand that MINE must be entered with uppercase characters to create HTML output. If Mine (not all uppercase) is entered as the output device, the results of the layout program would be displayed on the screen in PostScript format.

The alternative way to create HTML output is to select the HTML option for the Output Type.



Selecting the HTML option locks in the format type. When selecting MINE for the output device on the first prompt, the display of the output will be in HTML format. Typing a file name for the output device on the first prompt sends the output to a file in HTML format.
For any new programs, the selection to use the HTML options is available in the New Layout Program dialog box.

Using HTML output enables you to create linked reports you can use to navigate down through multiple levels to see additional detail. To demonstrate this functionality, create a report that displays information about people. For each person that is displayed, create a link to a report that displays information about that person's encounters. For each encounter, we will create a link to a report to display information about the orders for that encounter. To build these reports, identify the person IDs of some people, and the encounter IDs of some of their encounters for use in testing. The following query selects information about people, their encounters, and orders and can be used to find some testing data:

```
SELECT
P.PERSON_ID,
P.NAME_FULL_FORMATTED,
P.NAME_LAST_KEY,
P.NAME_FIRST_KEY,
E.ENCNTR_ID,
E_ENCNTR_TYPE_CLASS_DISP = UAR_GET_CODE_DISPLAY(E.ENCNTR_TYPE_CLASS_CD),
O.ORDER_ID,
O_CATALOG_DISP = UAR_GET_CODE_DISPLAY(O.CATALOG_CD)
FROM
PERSON P,
ENCOUNTER E,
ORDERS O
PLAN P WHERE P.PERSON_ID > 0.0
JOIN E WHERE P.PERSON_ID = E.PERSON_ID
JOIN O WHERE E.ENCNTR_ID = O.ENCNTR_ID
ORDER BY
P.PERSON_ID,
E.ENCNTR_ID,
O_CATALOG_DISP
WITH MAXREC = 100, NOCOUNTER, SEPARATOR=" ", FORMAT
```

1. Using DVDev, from the File menu, select **New**. The New dialog box opens.
2. From the File Type list, select **Blank** and click **OK**.
3. Copy the select statement above and paste it into the blank file in DVDev.
4. To execute the select statement, press CTRL+Q, or from the Build menu, select **Run Ad Hoc Query**.
5. Review the output from the query and record several person_ids, encntr_ ids, last_name_keys, and first_name_keys. We will use this information while building and testing the linked reports.

   Now that you have identified some values to use for testing, you can begin building the reports that will eventually be linked to provide the navigate down capability. First, create the report that displays information about orders for an encounter. To test this report, create it as a Table View layout program that prompts for an output device and encntr_id and displays the results in a table.
6. Using DVDev, from the File menu, select **New**. The New dialog box opens.
7. From the File Type list, select **Layout Program**.
8. In the Program Name box, enter **1_*Your*_*Initials*_Linked_Orders**, and click **OK**. The New Layout Program dialog box opens.
9. Select Table View from the Report Layout option. Enter **3** for the Number of Columns: option.
10. Select the HTML Output Type. After you click **Finish**, three rows are created: HeadReportRow, DetailRow and FootReportRow. Each row contains three equally spaced cells.
11. From the Tools menu, select **Prompt Builder**. The Prompt Builder dialog box opens with a single control for an output device.
12. Click **Add** to add a second prompt to ask for an encntr_id from the user.
13. In the **General** tab, set the following:

Prompt Display:	Enter an Encntr_ID
Prompt Name:	EID
Control Type:	Text Edit
Prompt Type:	Expression

14. Click **Save** to save the prompt form and close the Prompt Builder dialog box.

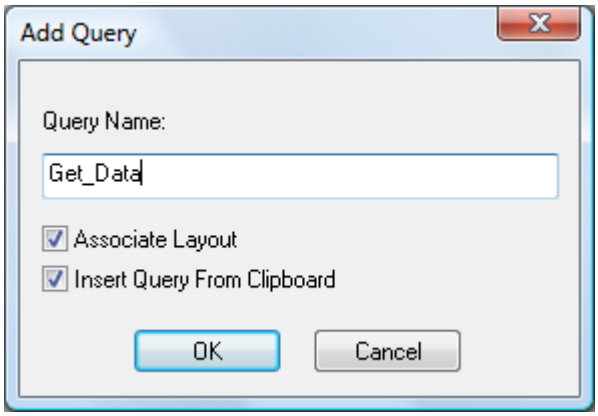
    The following query selects information about orders for a specific encntr_id that is entered at the EID prompt you created.

```
SELECT into "NL:"
O.ORDER_ID,
O_CATALOG_DISP = UAR_GET_CODE_DISPLAY(O.CATALOG_CD),
O.ORIG_ORDER_DT_TM ";;q"
FROM ORDERS  O
WHERE  O.ENCNTR_ID =  $EID
ORDER BY
O.ORIG_ORDER_DT_TM,
O_CATALOG_DISP
WITH MAXREC = 100, NOCOUNTER, SEPARATOR=" ", FORMAT
```

15. Copy the above query to the clipboard.
16. From the Tools menu, select **Query Builder**. The Add Queries dialog box opens.
17. Click **Add** to add a new query. The Add Query dialog box opens.

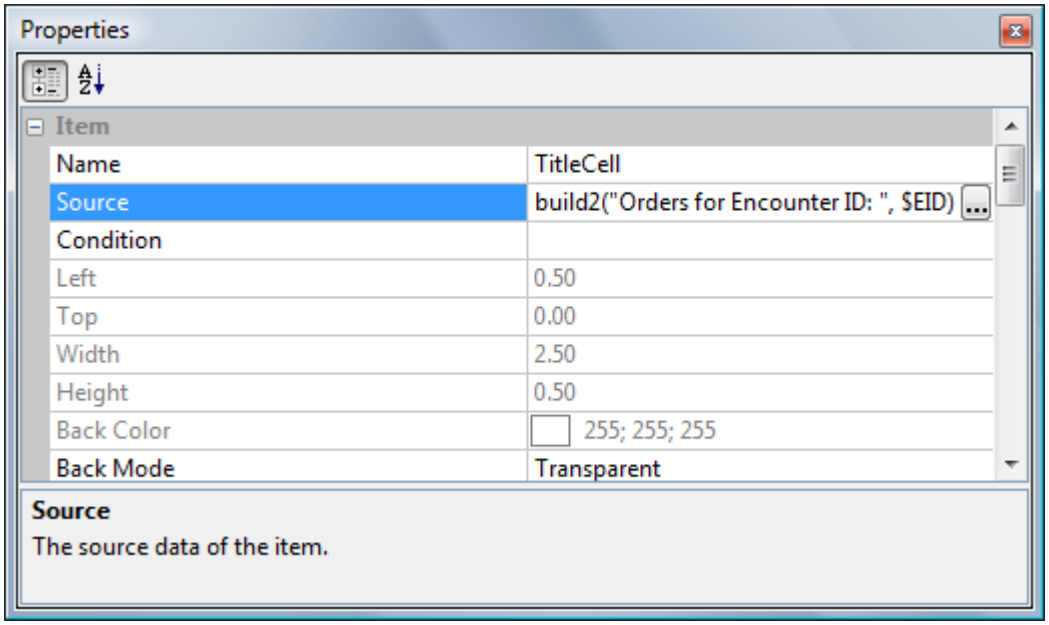
18. In the Query Name box, enter **Get_Data**.
19. Select the Associate Layout and Insert Query From Clipboard options.

    Your Add Query dialog should be similar the following:



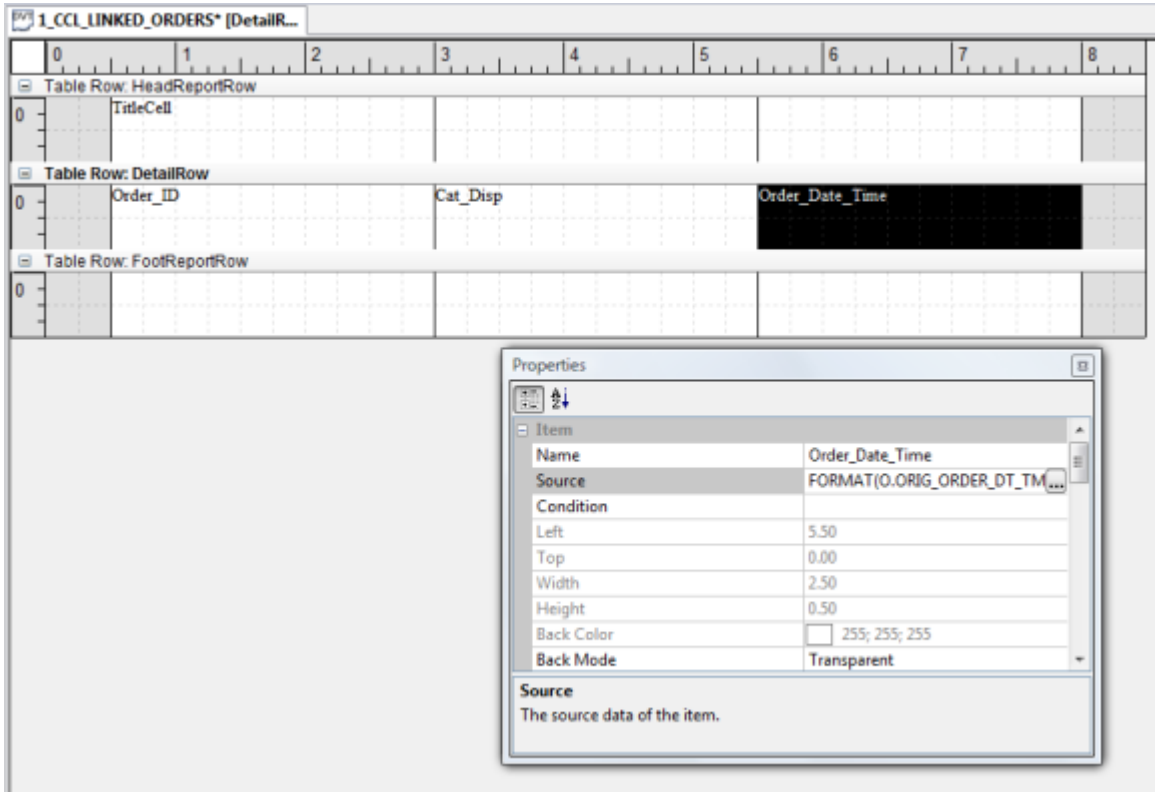
20. Click **OK** to add the query. The Discern Query Builder dialog box opens.
21. Click **Close** to close the Discern Query Builder dialog box.
22. Click **OK** to close the Add Queries dialog box.
23. Click in the first cell of the HeadReportRow on your table. The Properties window displays the information for the call you selected.
24. Modify the Name to **TitleCell**.
25. In the Source box, enter the following command:

    To create this command, click in the Source box to activate the ellipsis  button, and click the ellipsis  button to open the Layout Source [TitleCell] dialog. The Build2() function can be selected from the CCL Functions list in the Functions column.
    Your Properties dialog box should look like the following:



26. Click in the first cell of the DetailRow on your table.
27. In the Properties window, modify the Name property to **Order_ID**.
28. In the Source box, enter **O.ORDER_ID**, or click in the Source property box to activate the ellipsis  button to open the Layout Source [Order_ID] dialog box and select O.ORDER_ID from the Select Fields list in the Variables column.
29. Click in the middle cell of the DetailRow on your table.
30. In the Properties dialog box, modify the Name property to **Cat_Disp**.
31. In the Source box, enter **O_CATALOG_DISP,** or click in the Source property box to activate the ellipsis  button to open the Layout Source [Cat_Disp] dialog box and select O_CATALOG_DISP from the Select Fields list in the Variables column.
32. Click in the cell on the right side of the DetailRow on your table.
33. In the Properties window, modify the Name property to **Order_Date_Time**.
34. In the Source box, enter the following command:

To create this command, click in the Source box to activate the ellipsis button, and click the ellipsis button to open the Layout Source [Order_Date_Time] dialog. Select the Format() function from the CCL Functions list in the Functions column and select O.Orig_Order_DT_TM from the Select Fields list in the Variables column.
The following layout is displayed:



35. Save your 1_*Your_Initials*_Linked_Orders layout.
36. From the Build menu, select **Run "1_*your_initials*_Linked_Orders"**, or press CTRL+F5 to execute your layout program. The Discern Prompt dialog box opens.
37. Select MINE as the output device, enter one of the encntr_ids you recorded earlier at the *Enter an Encntr_ID* prompt, and click **Execute**.

    The Report Output window opens and the output of your 1_*Your_Initials*_Linked_Orders layout should look like the following:

Orders for Encounter ID: 590102		
592919.00	Hemogram	11/26/03 08:40:28
592920.00	Prostate Specific Antigen	11/26/03 08:40:31
595609.00	hepatitis A vaccine	01/06/04 03:34:49
595614.00	hepatitis A-hepatitis B vaccine	01/06/04 03:42:00
597619.00	amoxicillin	01/06/04 07:49:08

    The 1_*Your_Initials*_Linked_Orders layout program is the lowest level of detail in our linked reports. You will now create the intermediate level of our linked reports. For this level, create a report that displays information about all encounters for a specific person and provides a link that passes the encntr_id and executes the 1_*Your_Initials*_Linked_Orders report created above. To test the intermediate report, we will create it as a layout program that prompts for an output device and person _id and displays the results in a table.
38. From the File menu, select **New**. The New dialog box opens.
39. From the File Type list, select **Layout Program**.
40. In the Program Name box, enter **1_*Your_Initials*_Linked_Encntrs**, and click **OK**. The New Layout Program dialog box opens.
41. Select Table View from the Report Layout option. Enter **3** for the Number of Columns: option.
42. Select the HTML option for the Output Type. After you click **Finish**, three rows are created: HeadReportRow, DetailRow and FootReportRow. Each row contains three equally-spaced cells.
43. From the Tools menu, select **Prompt Builder**. The Prompt Builder dialog box opens with a single control for an output device.
44. Click **Add** to add a second prompt to ask for a person_id from the user.
45. In the **General** tab set the following:

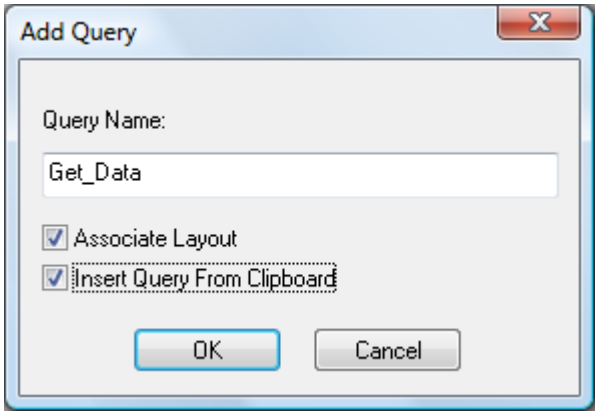
Prompt Display:	Enter a Person_ID
Prompt Name:	PID
Control Type:	Text Edit

Prompt Type:	Expression

46. Click **Save** to save the prompt form and close the Prompt Builder dialog box.

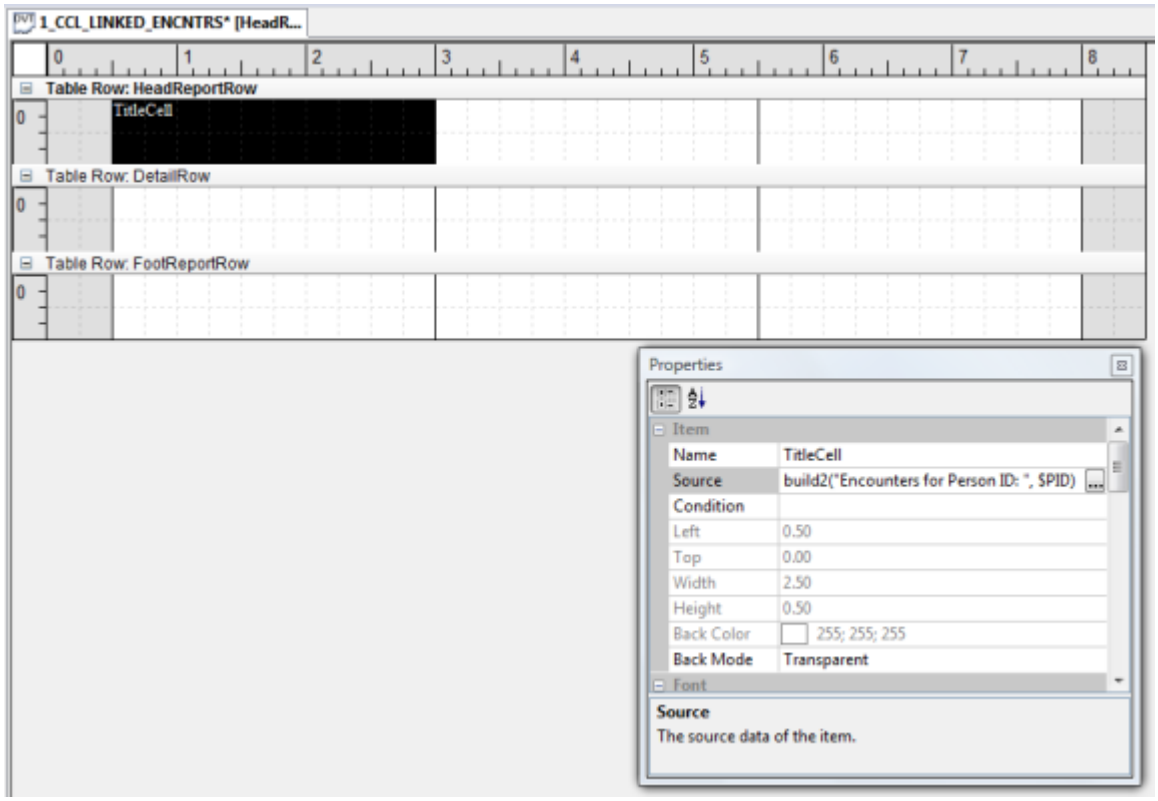
The following query selects information about encounters for a specific person_id that is entered at the PID prompt you created.

47. Copy the above query to the clipboard.
48. From the Tools menu, select **Query Builder**. The Add Queries dialog box opens.
49. Click **Add** to add a new query. The Add Query dialog box opens.
50. In the Query Name box, enter **Get_Data**.
51. Select the **Associate Layout** and the **Insert Query From Clipboard** options. The following Add Query dialog box is displayed:



52. Click **OK** to add the query. The Discern Query Builder dialog box opens.
53. Click **Close** to close the Discern Query Builder dialog box.
54. Click **OK** to close the Add Queries dialog.
55. Click in the first cell of the HeadReportRow on your table. The Properties dialog box displays the information for the cell you selected.
56. Modify the Name property to **TitleCell**.
57. In the Source property, enter the following command:

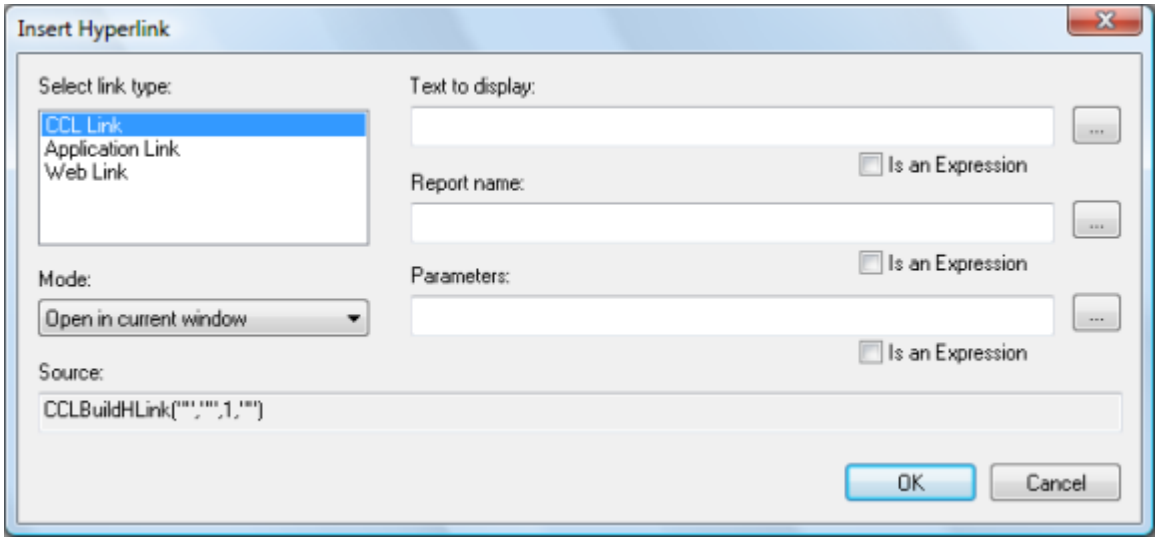
The following layout is displayed:



58. Click in the first cell of the DetailRow on your table.
59. In the Properties dialog box, modify the Name to **Encntr_ID**.
60. In the Source box, enter **E.ENCNTR_ID**, or click in the Source box to activate the ellipsis  button to open the Layout Source [Encntr_ID] dialog box and select E.ENCNTR_ID from the Select Fields list in the Variables column.
61. Click in the middle cell of the DetailRow on your table.
62. In the Properties dialog box, modify the Name to **EType_Disp**.
63. In the Source box, enter **E_ENCNTR_TYPE_CLASS_DISP**, or click in the Source property box, then click the ellipsis  button to open the Layout Source [EType_Disp] dialog box and select E_ENCNTR_TYPE_CLASS_DISP from the Select Fields list in the Variables column.

    Use the cell on the right side of the DetailRow on the table to create the link to execute the 1_*Your_Initials*_Linked_Orders program you created earlier. To create the link, use the special CCLBuildHLink() subroutine. The syntax for this subroutine is:
    CCLBuildHLink(*Program_Name, Prompt_Values, Display_Mode, Link_Text)Program_Name* – Name of program to execute. A literal value must be enclosed in quotation marks. A VC data type variable that is equal to the name of the program to execute can be used without the quotation marks.

    Prompt_Values – Prompt values to pass to the program. If the program to be executed is expecting multiple prompt values, the prompt values must be separated by commas. A literal list of comma-separated values must be enclosed in quotation marks. The Build(), Build2(), or Concat() function can be used to concatenate literal values and or values contained in variables into a single parameter value.
    Display_Mode – 0 = open new window, 1 = open in current window.
    Link_Text – Text to display in the link. The text can be a literal, a field with a character data type, a character data type variable, or a character data type value that is returned by a function.
64. Click in the cell on the right side of the DetailRow on your table. The Properties dialog box displays the information for the cell you selected.
65. In the Properties dialog box, modify the Name property to **OrderLink**.
66. Change the Is Hyperlink property to **Yes**. The Insert Hyperlink dialog box is displayed.



The Insert Hyperlink dialog box is used to populate the CCLBuildHLink() subroutine. When all of the parameters are filled out, the subroutine is placed as the value for the Source property.

> **Note**
> When you create a hyperlink, if there is text already associated to the source property, a warning of its deletion is displayed.

67. Verify the CCL Link is selected for the Select link type.
68. Click the menu on the Mode option and select **Launch DiscernReportViewer.exe**.

> **Note**
> As the parameters are populated, the CCLBuildHLink() subroutine is filled out and displayed in the Source box.

69. In the Text to display box, enter the following command:

    To create this command, click the ellipsis  button that is to the right of the Text to display box. The Layout Source dialog opens. The Concat() and Cnvtstring() functions can be selected from the CCL Function list in the Functions column. The E.Encntr_id can be selected from the Select Fields list in the Variables column.
70. Check the box for the Is an Expression option. Selecting the option enables the system to treat the command as an expression instead of a literal value.

    Filling out the Text to display box populates the Link_Text parameter of the CCLBuildHLink() subroutine.

71. In the Report Name box, enter the name of the program that you created to find the orders for an encounter:

    1_ *your_initials* Linked_Orders

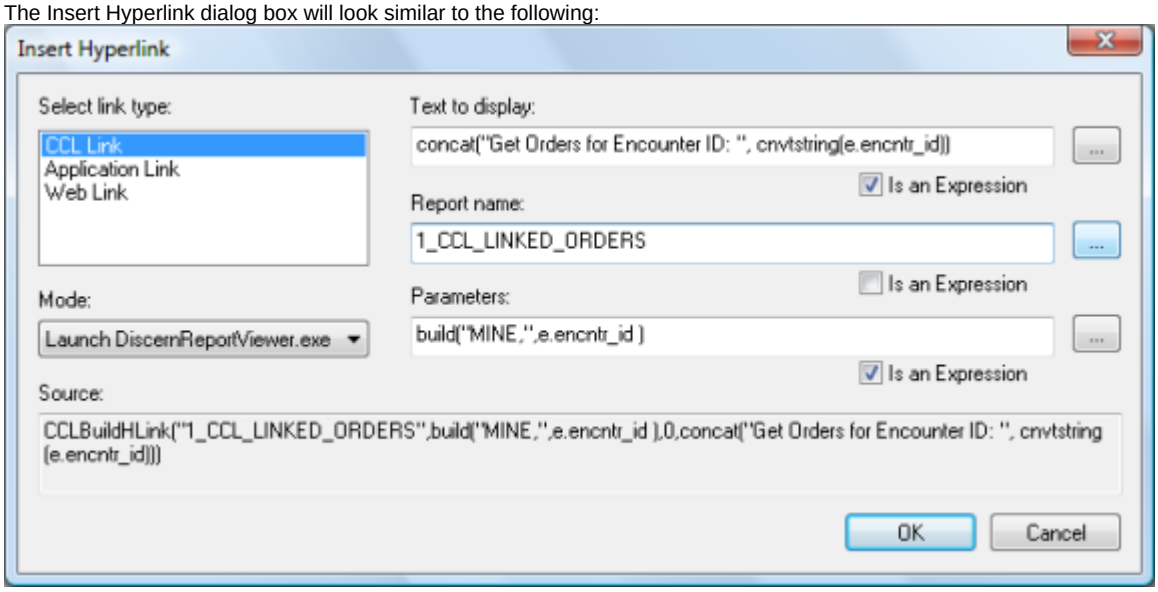
    The above command assumes that the 1_*Your_Initials*_Linked_Orders program was created as a cclgroup0 (DBA) program. If the username that was used to log in to DVDev when the 1your_initialslinked_orders program was created is not in cclgroup0, then by default the program would have been created as a cclgroup1 object. If the 1_*Your_Initials*_Linked_Orders program is a cclgroup1 object, you need to append :group1 to the end of the program name as shown below:
    1_ *your_initials* Linked_Orders:group1
    Use CCLPROT to determine if the 1_*your_initials*_linked_orders program is a cclgroup1 or cclgroup0 program.
72. Verify that the Is an Expression is not checked. This ensures the program name is treated as a literal value and quotes are placed around the value in the subroutine.

    Filling out the Report Name box populates the Program_Name parameter of the CCLBuildHLink() subroutine.
73. In the Parameters box, enter the command as shown below:

    To create this command, click the ellipsis  button to the right of the Parameters: box. The Layout Source dialog opens. The Build( ) function can be selected from the CCL Function list in the Functions column. The E.Encntr_id can be selected from the Select Fields list in the Variables column.

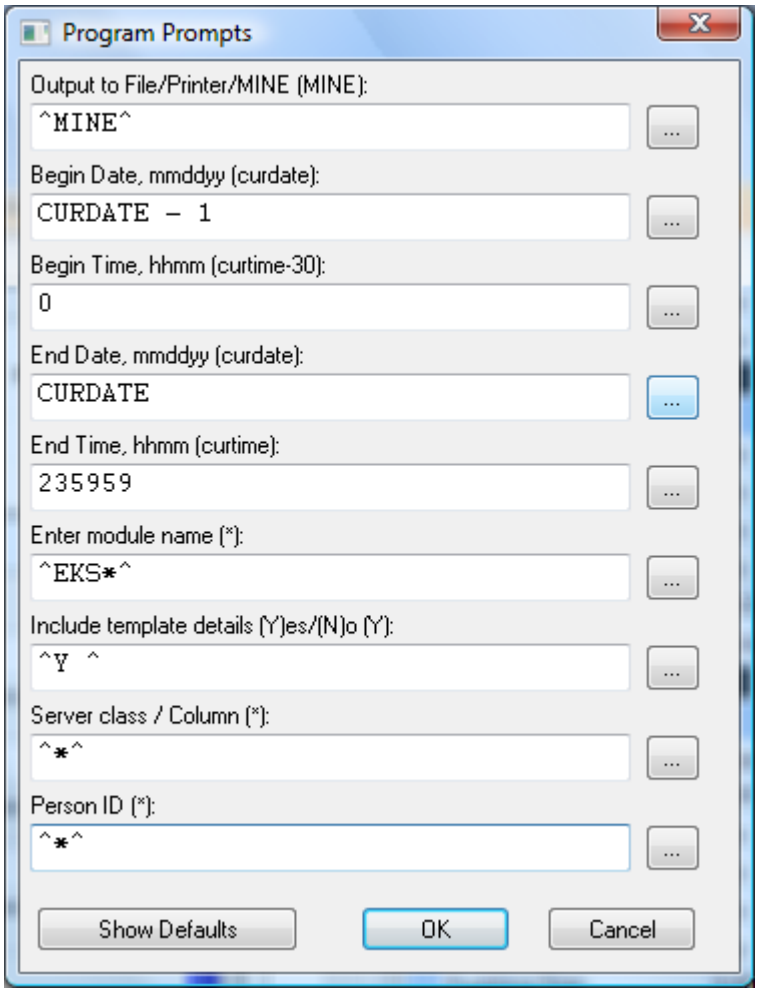
74. Check the box for the Is an Expression option. Selecting the option enables the system to treat the command as an expression instead of as a literal value.

    The Insert Hyperlink dialog box will look similar to the following:

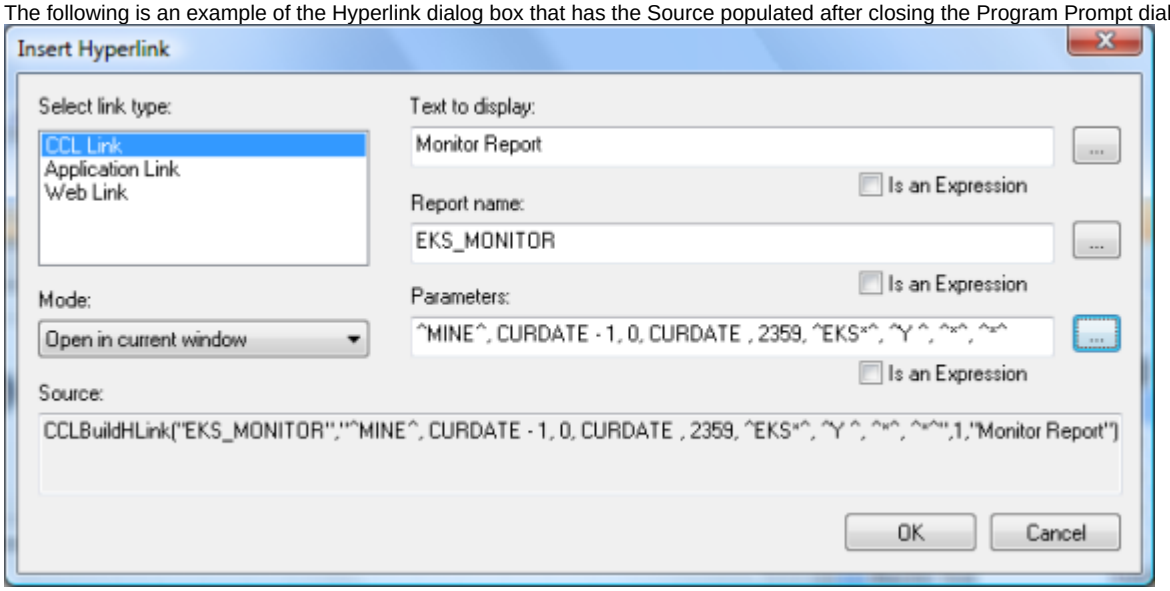


Filling out the Parameters box populates Prompt_Values parameter of the CCLBuildDHLink() subroutine.
In the above example, the e.encntr_id references a field from the query. To render the ENCNTR_ID and to have the actual ENCNTR_ID available to show in the link, the *Is an Expression* option is selected so that the value is treated as an expression and quotes are not wrapped around the value. In this example, to render the value, you used the Build() function to manually create the parameter string of comma-separated values consisting of the parameter of "MINE" and the actual ENCNTR_ID brought back from the query.
Any time you reference a field or expression from the query in the Parameters string, select the *Is an Expression* option and then use the Build(),Build2() or Concat() functions to manually build the parameter string. Using this method tells the system you have a command that needs to be executed and rendered for the display of the link. Otherwise, the parameter string is not evaluated until the link is selected. When the *Is an Expression* option is selected, you need to know the parameters required to pass to the program.
You do not need to select the *Is an Expression* option if the parameters being passed to the program are global variables, prompts, reserved system variables, or literal values. These expressions do not directly reference an item in the query. When parameters do not contain any reference to an item in the query, the *Is an Expression* option is not needed.

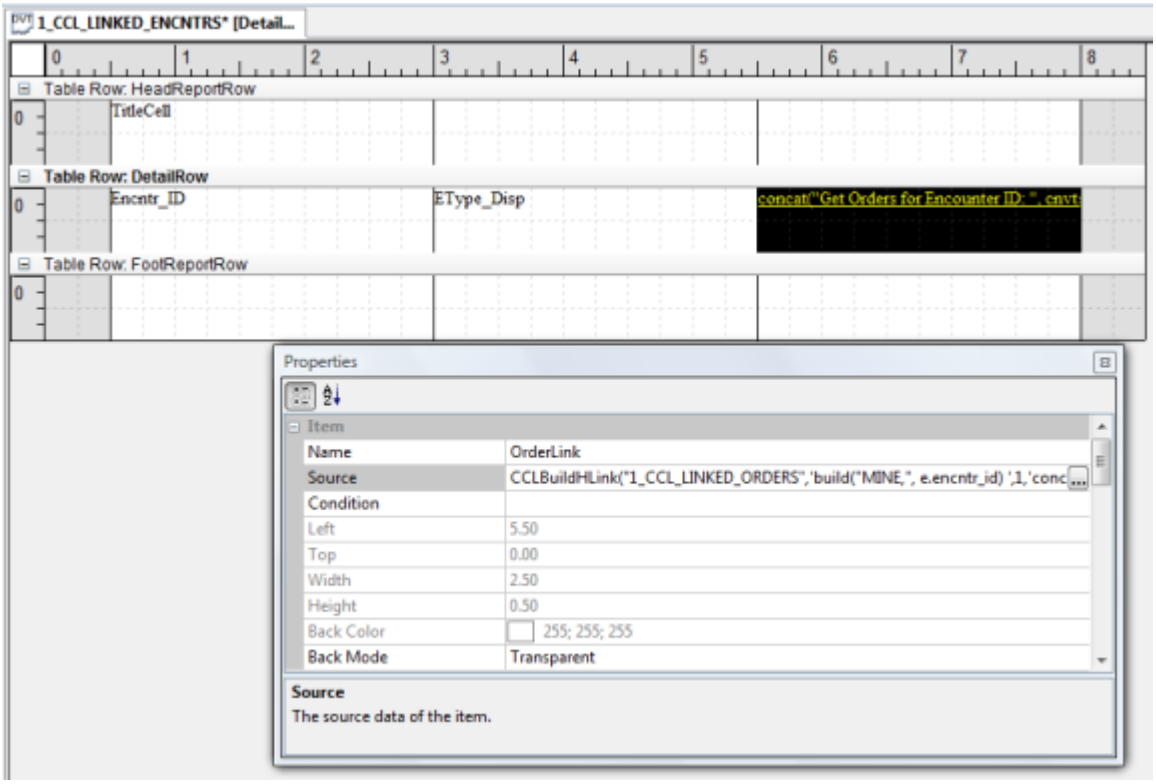
Click the ellipsis button  . A different dialog box, Program Prompts, is displayed and shows the prompts for the program. You do not need to remember all of the prompts required for the program as the Program Prompts dialog box displays them for you. However, the Program Prompts dialog box is only available when the *Is an Expression* is not selected and the ellipsis button is clicked.
The following is an example of a Program Prompts dialog box when the *Is an Expression* option is not selected and the ellipsis button is clicked.



The user can enter any expression in the text boxes or click any of the ellipsis buttons to display the Layout Source dialog box, where the values can be specified. Clicking **OK** assembles a parameter string of comma-separated values and returns you to the Insert Hyperlink dialog.
The following is an example of the Hyperlink dialog box that has the Source populated after closing the Program Prompt dialog box.



75. Click **OK** to close the Insert Hyperlink dialog box. The following layout is displayed:



The CCLBuildHLink() subroutine is populated and placed as the value for the Source property and should look similar to the following:

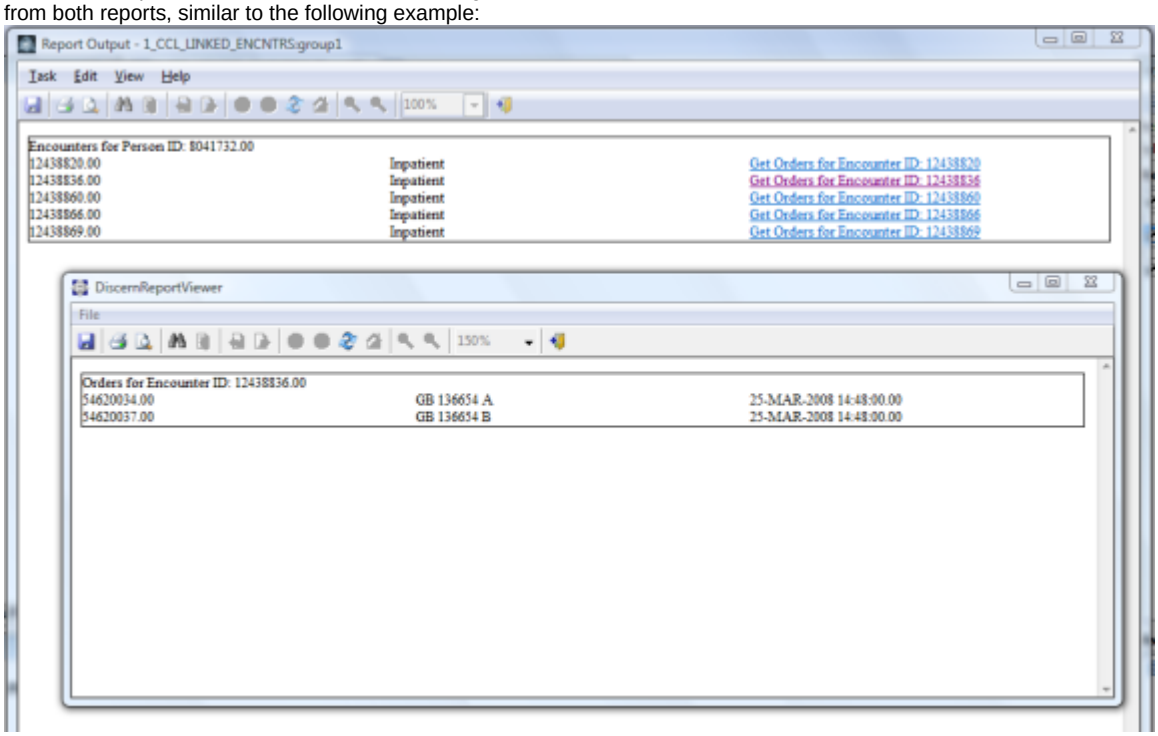
```
CCLBuildHLink("1_your_initials_linked_orders",
build("MINE,", e.encntr_id),
0,
concat("Get Orders for Encounter ID: ", cnvtstring(e.encntr_id)))
```

1. From the Build menu, select **Run "1_*Your_Initials*_Linked_Encntrs"**, or press CTRL+F5 to execute your layout program. The Discern Prompt dialog box opens.
2. Select MINE as the output device.
3. At the Person_ID prompt, enter one of the person_ids you recorded earlier and click **Execute**. The Report Output window opens and the output of your 1_*Your_Initials*_Linked_Encntrs layout should resemble the following example:

4. Click one of the Get Orders for Encounter ID links on the right side of the output.

   Because the Launch DiscernReportViewer.exe was selected as the Mode, a 0 (zero) was passed in to the Display_Mode parameter of the CCLBuildHLink() subroutine. This opens a new DiscernReportViewer window and the output of your 1_*your_initials*_Linked_Orders report is displayed. If

the DiscernReportViewer window is maximized, right-click the window title bar, and select **Restore**. Then resize the window so you can see the output from both reports, similar to the following example:



5. Close the DiscernReportViewer and Report Output windows.

   You can now create the top-level program of our linked reports. For this level, create a report that displays information about people with a specific name and provide a link that passes the person_id and executes the 1_*your_initials*_Linked_Encntrs report you created above. You will create this report as a Table View that prompts for an output device, a first and last name, and displays the information about people with that name in a table.
6. Using DVDev, from the File menu, select **New**. The New dialog box opens.
7. From the File Type list, select **Layout Program**.
8. In the Program Name box, enter **1_*your_initials*_Linked_People**, and click **OK**. The New Layout Program dialog box opens.
9. Select **Table View** from the Report Layout option. Enter **3** for the Number of Columns option.
10. Select the HTML option for the Output Type. When you click **Finish**, three rows are created: HeadReportRow, DetailRow and FootReportRow. Each row contains three equally-spaced cells.
11. From the Tools menu, select **Prompt Builder**. The Prompt Builder dialog box is displayed with a single control for an output device.
12. Click **Add** to add a second prompt to get a last name from the user.
13. In the **General** tab, set the following:

Prompt Display:	Enter a Last Name
Prompt Name:	LName
Control Type:	Text Edit
Prompt Type:	String

14. In the **Text Properties** tab, set the Character Case to Upper.
15. Leave the default values for all other properties.
16. Click **Add** to add a third prompt to get a first name from the user.
17. In the **General** tab, set the following:

Prompt Display:	Enter a First Name
Prompt Name:	FName
Control Type:	Text Edit
Prompt Type:	String

18. In the **Text Properties** tab set the Character Case to Upper.
19. Leave the defaults for all other properties.
20. Click **Save** to save the prompt form and close the Prompt Builder.

    The following query selects information about people that have a specific first and last name that is entered at prompts named FName and LName. You created these prompts above.

21. Copy this query to the clipboard.
22. From the Tools menu, select **Query Builder**.
23. To add a new query, click **Add** in the Add Queries dialog box.
24. In the Add Query dialog box, enter **Get_Data** in the Query Name box.

25. Select the **Associate Layout** and **Insert Query From Clipboard** options and click **OK** to add the query. The Discern Query Builder dialog box opens.
26. Click **Close** to close the Discern Query Builder.
27. Click **OK** to close the Add Queries dialog box.
28. Click in the center cell of the HeadReportRow on your table. The properties for that cell are displayed in the Properties window.
29. Modify the Name property to **TitleCell**.
30. In the Source property, enter **"Linked Person Encounters Orders Report"** for this cell.
31. In the Align Horizontal property, select **Center**.
32. Click in the first cell of the DetailRow on your table, and in the Properties window, modify the Name property to **Person_ID**.
33. In the Source box, enter P.PERSON_ID, or click the ellipsis  button to open the Layout Source [Person_ID] dialog box and select P.PERSON_ID from the Select Fields list in the Variables column.
34. Click in the middle cell in the second row of your table, and in the Properties dialog box, modify the Name property to **Formatted_Name**.
35. In the Source property, enter **P.NAME_FULL_FORMATTED**, or click the ellipsis  button to open the Layout Source [Formatted_Name] dialog box and select P.NAME_FULL_FORMATTED from the Select Fields list in the Variables column.

    Use the cell on the right side of the DetailRow on the table to create the link to execute the 1_*your_initials*_Linked_Encntrs program you previously created using the special CCLBuildHLink() subroutine.
36. Click in the cell on the right side of the DetailRow on your table. The properties for the cell are displayed.
37. In the Properties window, modify the Name property to **EncntrLink**.
38. Change the Is Hyperlink property to **Yes**. The Insert Hyperlink dialog box is displayed.
39. Verify the CCL link is selected for the Select link type.
40. Click the menu on the Mode option and select Launch DiscernReportViewer.exe.
41. In the Text to display box, enter the following command:

    To create this command, click the ellipsis  button that is to the right of the Text to display box. The Layout Source dialog opens. You can select the Concat( ) function from the CCL Function list in the Functions column. The P.Name_full_formatted can be selected from the Select Fields list in the Variables column.

42. Check the box for the Is Expression option. Selecting the option enables the system to treat the command as an expression instead of a literal value.
43. In the Report Name box, enter the name of the program that you created to find the encounters for a person as shown below:

    1_ *your_initials* Linked_Encntrs

    Note
    The above command assumes that the 1_*your_initials*_Linked_Encntrs program was created as a cclgroup0 (DBA) program. If the username that was used to log in to DVDev when the 1_*your_initials*_Linked_Encntrs program was created is not in cclgroup0, then by default the program would have been created as a cclgroup1 object. If the 1_*your_initials*_Linked_Encntrs program is a cclgroup1 object, you need to append ":group1" to the end of the program name as shown below:
    1_ *your_initials* Linked_Encntrs:group1

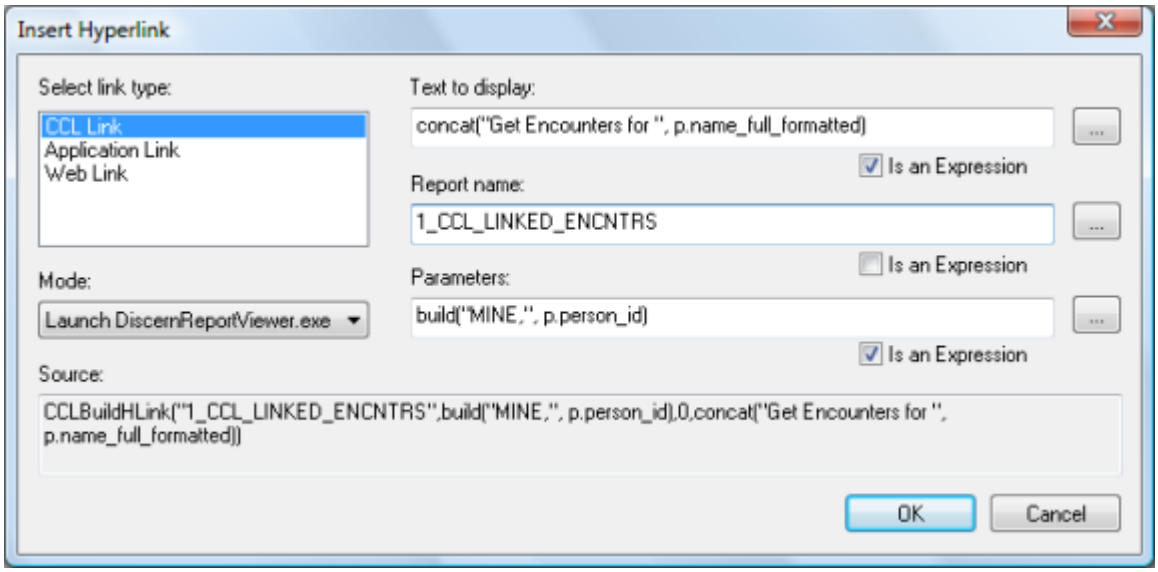
    Use CCLPROT to determine if the 1_*your_initials*_linked_encntrs program is a cclgroup1 or cclgroup0 program.

44. Verify that Is an Expression is not checked. This ensures that the program name is treated as a literal value and quotes are placed around the value in the subroutine.
45. In the Parameters box, enter the command as shown below:

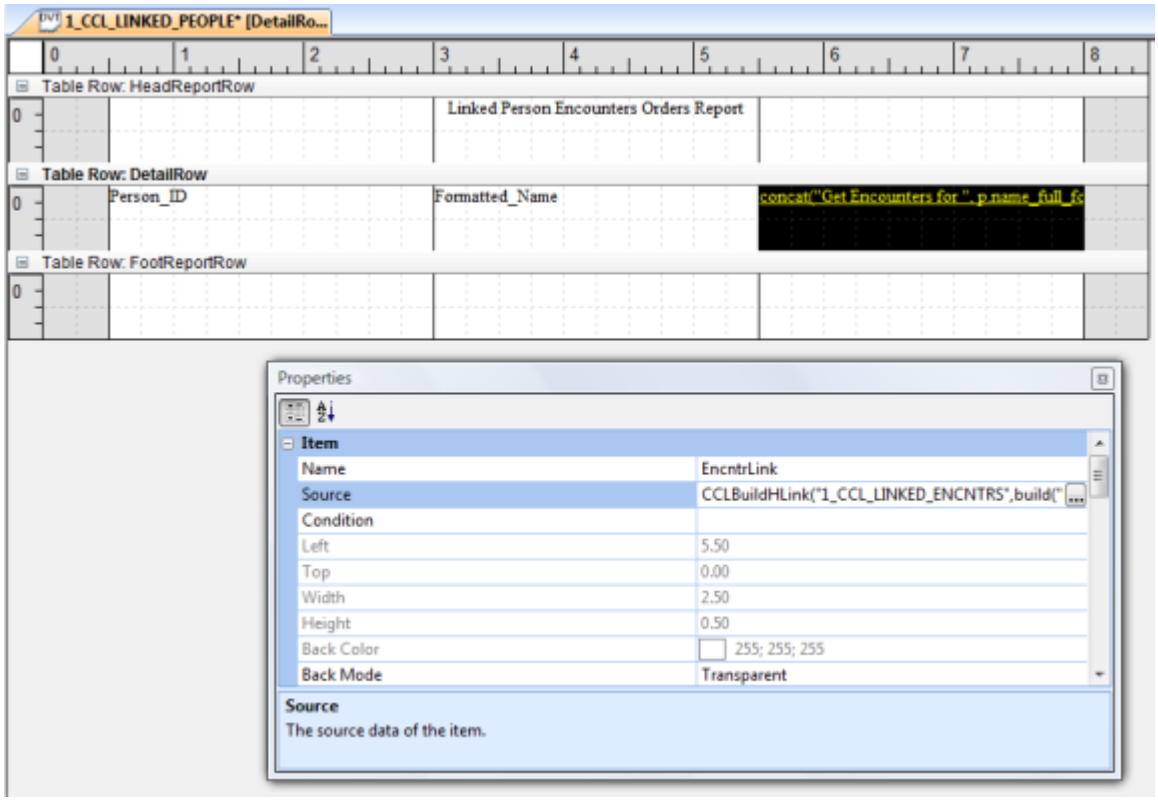
    To create this command, click the ellipsis  button that is to the right of the Parameters box. The Layout Source dialog opens. The Build( ) functions can be selected from the CCL Function list in the Functions column. You can select P.Person_id from the Select Fields list in the Variables column.

46. Check the box for the Is an Expression option. Selecting this option lets the system treat the command as an expression instead of a literal value.

    The Insert Hyperlink dialog box should look similar to the following:

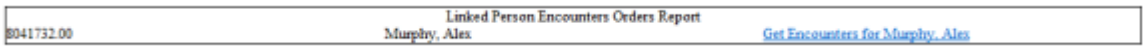


47. Click **OK** to close the Insert Hyperlink dialog box. The following layout is displayed:



The CCLBuildHLink() subroutine is populated and placed as the value for the Source property and should look similar to the following:
CCLBuildHLink("1_*your_initials*_linked_encntrs",
build("MINE,", p.person_id),
0,
concat("Get Encounters for ", p.name_full_formatted))

48. From the Build menu, select **Run "1_ Your_Initials_Linked_People"** or press CTRL+F5 to execute your layout program. The Discern Prompt dialog box opens.
49. Select MINE as the output device.
50. Enter one of the last names that you recorded earlier at the last name prompt. Enter one of the first names that you recorded earlier at the first name prompt. The Report Output window opens and the output of your 1_your_initials_Linked_People layout should look like the following example:

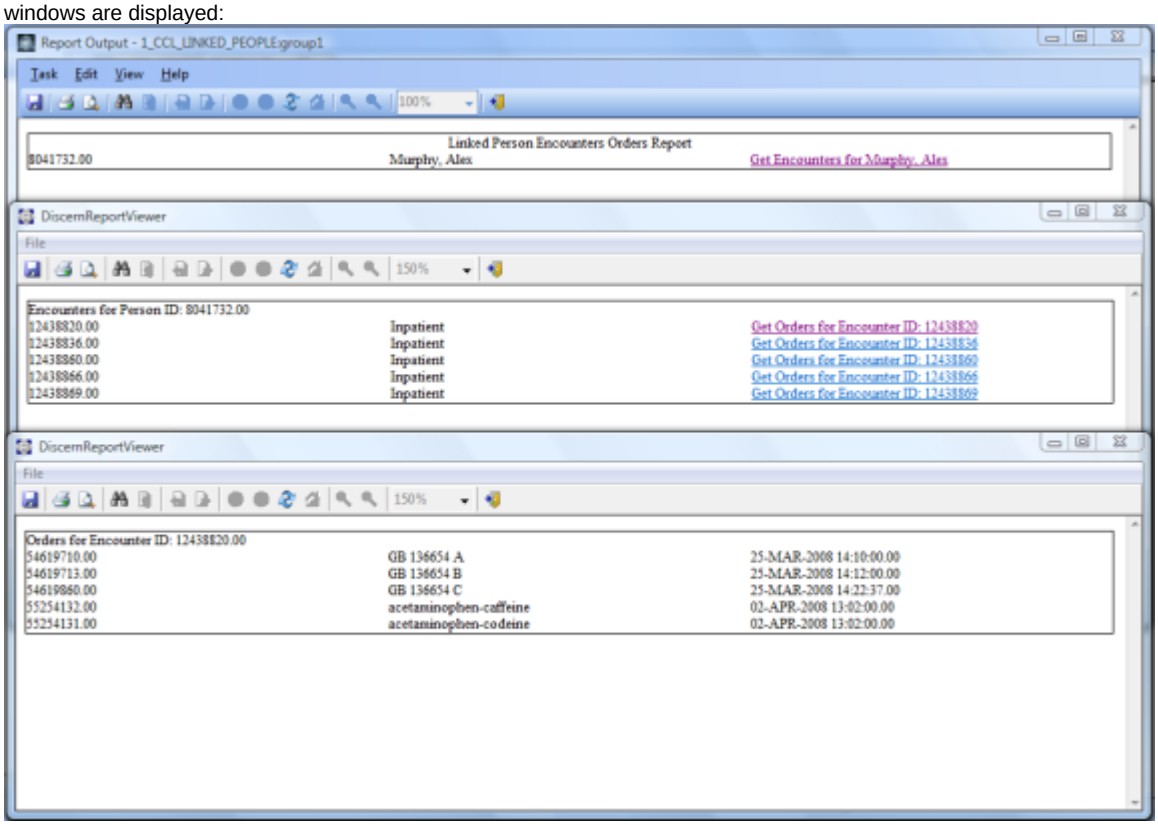


51. Click the **Get Encounters for *Name_Full_Formatted*** link on the right side of the output.

Because the Launch DiscernReportViewer.exe was selected as the Mode, a 0 (zero) was passed in to the Display_Mode parameter of the CCLBuildHLink() subroutine. This opens a new DiscernReportViewer window and the output of your 1_*your_initials*_Linked_Encntrs report is displayed. If the DiscernReportViewer window is maximized, resize it so you can see the output from both reports.

52. Click one of the Get Orders for Encounter ID: Encntr_ID links on the right side of the output.

    Because the Launch DiscernReportViewer.exe was selected as the Mode, a 0 (zero) was passed in to the Display_Mode parameter of the CCLBuildHLink() subroutine. This opens a new DiscernReportViewer window and displays the output of your 1_*your_initials*_Linked_Orders report is displayed. If the DiscernReportViewer window is maximized, resize it so you can see the output from all three reports. The following Report Output windows are displayed:



53. Close the DiscernReportViewer and the Report Output windows.

    Once you complete these steps, continue to the next part of Use Discern Layout Builder 2007.18, Converting Existing Programs to Layout Programs Using a Driver Program.