Use Discern Layout Builder Part H - Manually Calling Layout Subroutines From Report Programs

# Manually Calling Layout Subroutines From Report Programs

Most likely, the easiest way to add layout functionality to an existing query program is to convert the existing program to a driver program and call it from a layout program (see the Converting Existing Programs to Layout Programs Using a Driver Program section for more information on converting an existing program into a driver program).

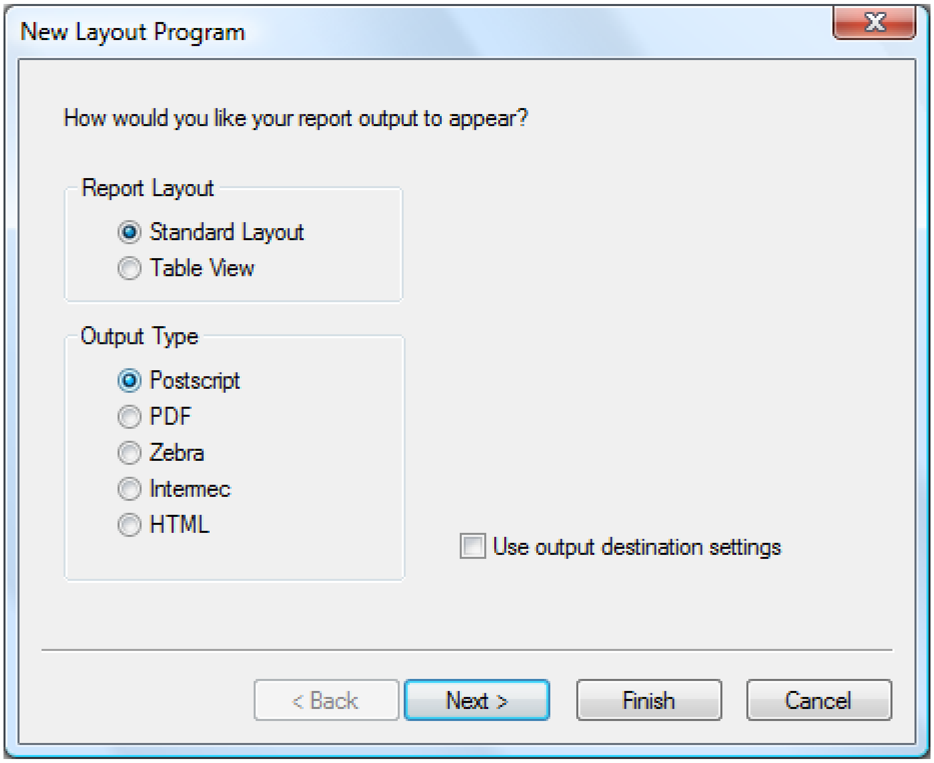
When you create a layout program, DVDev creates subroutines for each layout section and places commands to call those subroutines into the appropriate reportwriter sections of the query associated with the layout. Appendix A contains additional information about the subroutines that DVDev creates. You can use the Layout Builder to create layout sections and to add the code to call those sections to an existing program; however, doing so requires you to manually add the code to call the layout section subroutines to the source code that creates the query or program. You will also need to control page breaks and manage the overall flow of the program.

For example, suppose you had the following source code, which creates a program that prompts for a last name and then selects person and encounter information for people with a last name equal to the value entered at the prompt. You want to use the layout builder functionality to format the results of the query.

1. Copy the text from the above example and paste it into a blank file in DVDev.
2. Modify the name of the program to match the naming conventions of your site. For example, modify 1_ccl_add_layt_to_prg to **1_*your_initials*_ add_layt_to_prg**. If you are logged in to DVDev using a cclgroup1 account, ensure that the total length of the program name is 23 or fewer characters. Subsequent steps in this exercise will create a layout and generate a file with a .DVL extension that contains the source code for the layout subroutines. The program name will be used as the file name when creating the .DVL file. When using a cclgroup1 account, if the program name exceeds 23 characters, the name of the .dvl file will be truncated. The subsequent instructions assume the name of the .DVL file is the same as the program name used in this step. To avoid confusion, ensure your program name is 23 or fewer characters in length when using a cclgroup1 account. If you are using a cclgroup0 (DBA) account, the name of the .DVL file will not be truncated, so the program name can use up to the 30 characters as allowed by Discern Explorer for program names.
3. Save the source code file and verify that it will include/compile without any errors.
4. Execute the program as a prompt program to verify that it executes without error and returns information about people and their encounters.

## Creating the Layout

1. Add a layout to the program created in the source code in the section above by placing your cursor within the body of the Create Program command; from the Tools menu, select Layout Builder. The New Layout Program dialog box opens.



New Layout Program

How would you like your report output to appear?

Report Layout

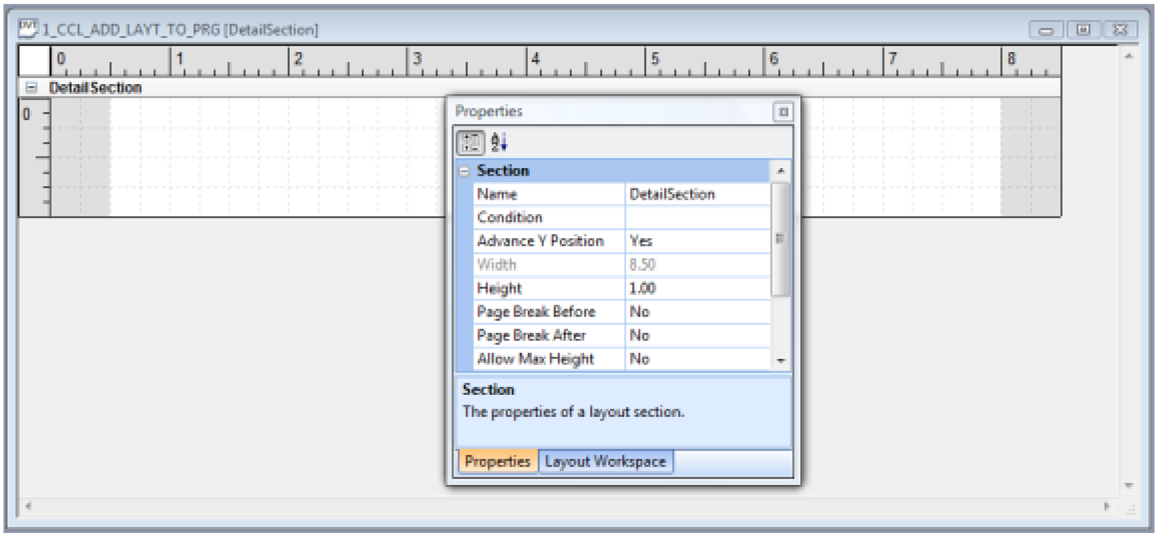
● Standard Layout
○ Table View

Output Type

● Postscript
○ PDF
○ Zebra
○ Intermec
○ HTML

□ Use output destination settings

2. Verify that the Standard Layout option for the Report Layout and PostScript option for the Output Type are selected, and click Finish. A layout with a single section named DetailSection is created.



1_CCL_ADD_LAYT_TO_PRG [DetailSection]

Detail Section

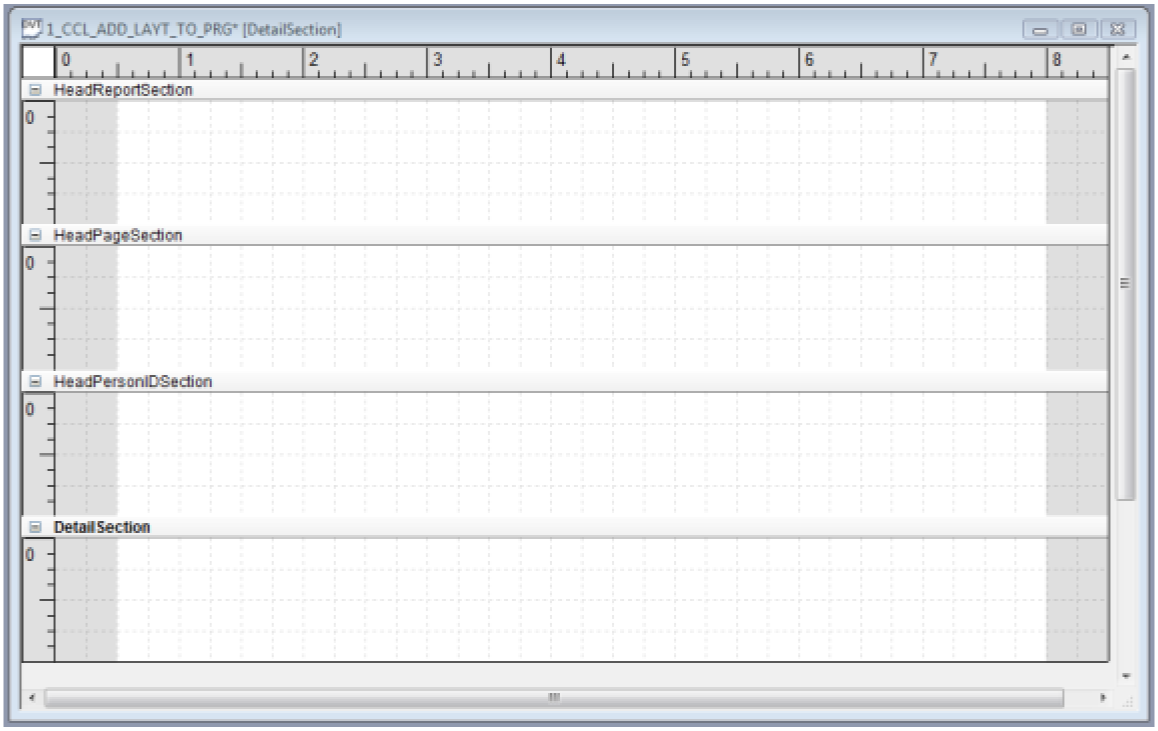
Properties

| Section | |
|---|---|
| Name | DetailSection |
| Condition | |
| Advance Y Position | Yes |
| Width | 8.50 |
| Height | 1.00 |
| Page Break Before | No |
| Page Break After | No |
| Allow Max Height | No |

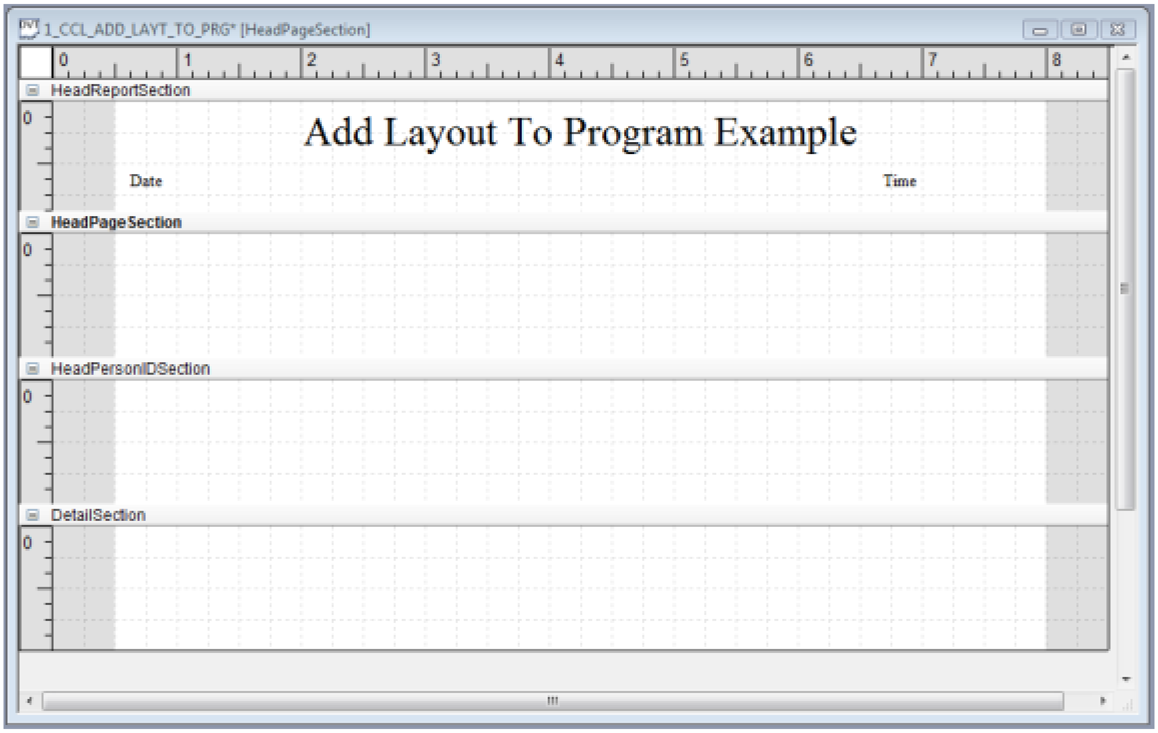
**Section**
The properties of a layout section.

For our example, we want to format the data returned from the PERSON and ENCOUNTER tables, so we need a report header at the top of the report, the page number and column headings at the top of each page, some specific information about each person, and some specific information about each encounter. To accomplish this we must add layout sections for the head report, head page, head PERSON_ID, and detail.

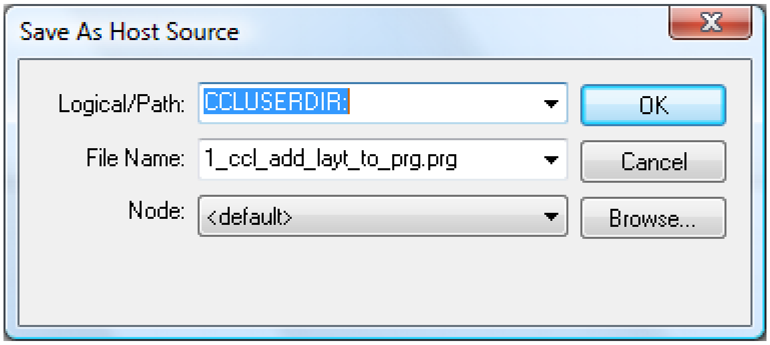
3. From the Edit menu, select Insert > Section to add three more sections to your layout.
4. Click in the first section; in the Properties dialog box, modify the Name property to **HeadReportSection**.
5. Click in the second section; in the Properties dialog box, modify the Name property to **HeadPageSection**.
6. Click in the third section; in the Properties dialog box, modify the Name property to **HeadPersonIDSection**.
7. The fourth section should already be labeled DetailSection. Modify the Name property if needed. You can expect your layout to be similar to the following screen:



8. Use the Label Tool Aa to add a report title Add Layout To Program Example to the HeadReportSection, and use the Formatting toolbar to set the following values:
    1. The font to Times
    2. The font size to 24
    3. Center the text
9. Use the Text Tool A to add a field to the HeadReportSection to display the current date when the layout program is executed. In the Properties dialog box, modify the Name property to **Date** and the Source to **CURDATE**.
10. Use the Text too A to add a field to the HeadReportSection to display the current time when the layout program is executed. In the Properties dialog box, modify the Name property to **Time** and the Source to **CURTIME2**. You can expect your layout to be similar to the following screen:



11. From the File menu, select Save As. The Save As Host Source dialog box opens.



Save As Host Source

Logical/Path: CCLUSERDIR:

File Name: 1_ccl_add_layt_to_prg.prg

Node: <default>

12. Verify the Logical/Path is CCLUSERDIR: and the File Name: is 1_*your_initials*_add_layt_to_prg.prg and click OK.

    Saving the layout creates a file that has the same name as the program name that is used in the Create Program command with a .DVL extension. By default, the *Program_Name*.dvl file is saved in the CCLUSERDIR: directory on the back-end host. The *Program_Name*.dvl file contains the commands to create the layout subroutines used to render the output. We now need to add the commands to the source code

# Calling Layout Sections in the Existing Program/Query

1. Add the following commands to your source code file. These commands must be placed after the Prompt clause and before the Select command.

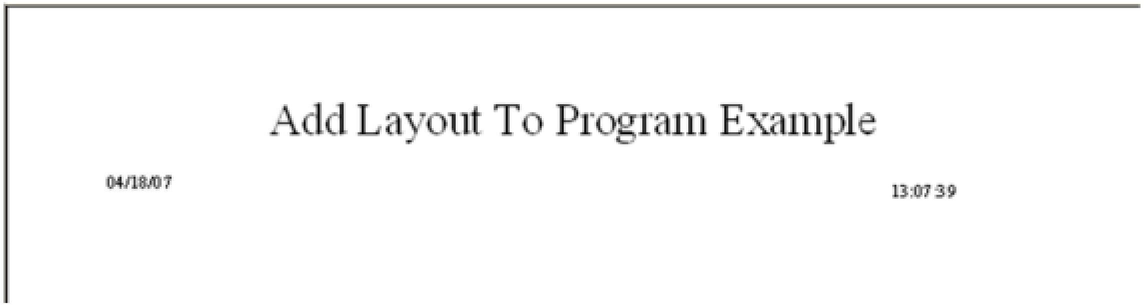
2. Modify Select Into $OutDev to **Select Into "NL:"**.

   Instead of selecting into $OutDev, you will send the information to the Null Device ("NL;"). Then, later in the program, you will pass $OutDev to a subroutine to direct the output to the device that the user enters at the prompt.
3. Add a Head Report section to the Select command that contains the following command:

   This is assuming you created a layout section named HeadReportSection.

4. Add the following command after the Select command's With clause, but before the End Go:

5. After adding the commands to your source code, save the file and include/compile it.
6. Execute the program as a prompt program by using CTRL+R, by clicking Run Prompt Program, or by selecting the Run Prompt Program option from the Build menu.
7. Use MINE as the output device and enter the last name of a person that you know has encounters when prompted. You can expect your output to be similar to the following screen:

   ```
   Add Layout To Program Example
   04/18/07                                  13:07:39
   ```

   At this point your program creates only the output from the HeadReportSection. To display additional information in the output, you need to add the information to the layout and modify the source code to call the layout subroutines.

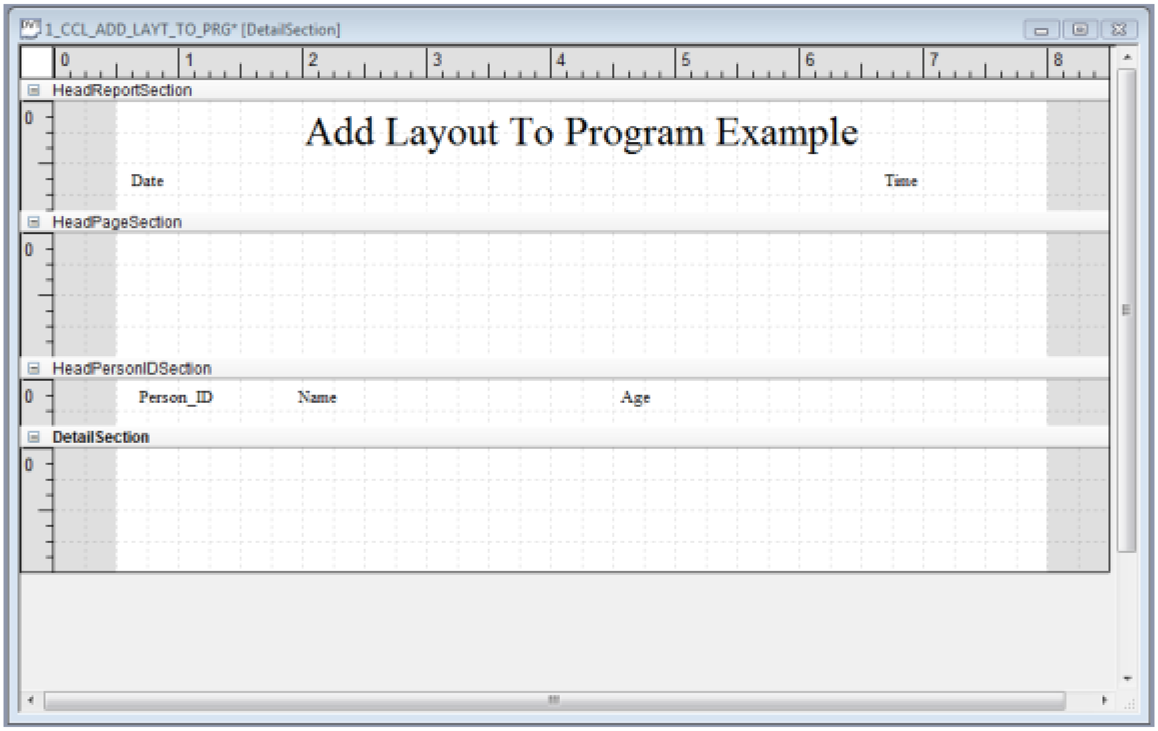
8. Use the Text Tool A to add a field to the HeadPersonIDSection to display the Person_ID. In the Properties dialog box, modify the Name property to **Person_ID** and the Source to **P.Person_ID**.

9. Use the Text Tool A to add a field to the HeadPersonIDSection to display the person's full name. In Properties dialog box, modify the Name property to **Name** and the Source to **P.Name_Full_Formatted**.

10. Use the Text Tool A to add a field to the HeadPersonIDSection to display the person's gender. In the Properties dialog box, modify the Name property to **Gender** and the Source to **P_Sex_Disp**.

11. Use the Text Tool A to add a field to the HeadPersonIDSection to display the person's age. In the Properties dialog box, modify the Name property to **Age** and the Source to **Age**.
12. Slowly move your pointer over the DetailSection title bar. At the top of the title bar the pointer will change to the vertical resize pointer. When the pointer changes, click and drag the DetailSection title bar up to the bottom of the fields you placed in the HeadPersonIDSection. You can expect your layout to

be similar to the following screen:



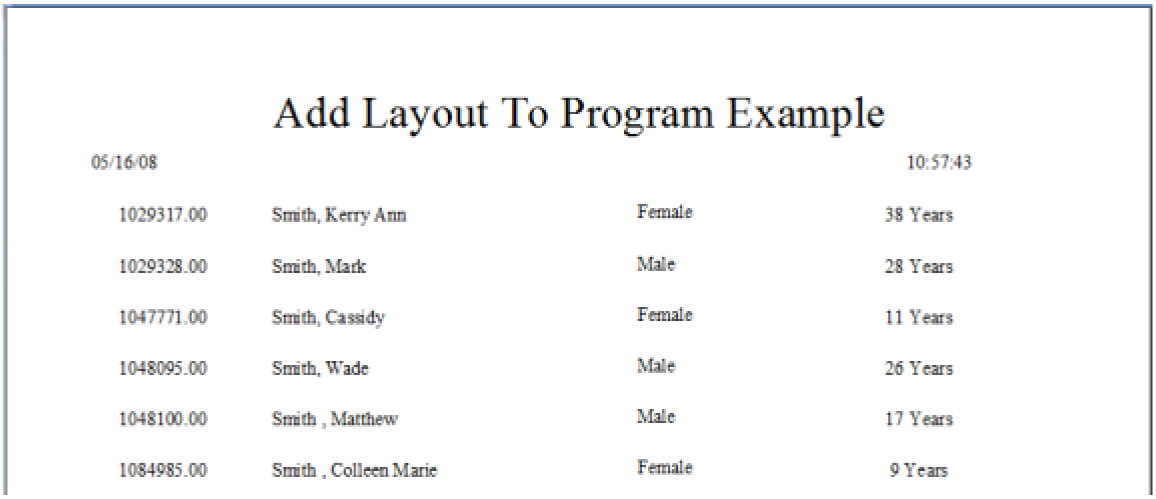
13. Save the layout.

    You are now ready to add the commands to your source code to execute the HeadPersonIDSection subroutine in the Head P.Person_ID reportwriter section of the Select command. Here, however, we run into an interesting issue that results from the way *Discern Explorer* performs the internal reportwriter section processing. If the query uses reportwriter sections, then only expressions created in the select list, the fields that are used as Head/Foot group expression sections, or fields that are used in a reportwriter section are recognized in the internal processing. Layout Builder creates subroutines that are outside the reportwriter sections. You then add the code in the reportwriter sections to call the subroutines.
    Any fields that are used on the layout must be referenced somewhere within a reportwriter section to be recognized by the internal processing. If the field is used only in the layout subroutine, it is not recognized by the internal processing and generates a %CCL-E-85 error. Currently, you have added the P.Person_ID field, the P.Name_Full_Formatted field, the expression P_Sex_Disp, and the expression Age to your HeadPersonIDSection layout section. In order for the fields P.Name_Full_Formatted and P.Person_ID to be recognized when the layout subroutine is called, they must be referenced somewhere in a reportwriter section before the subroutine is called.
14. Add the following command to the Head Report section of your source code:

    The command above is used only to make the P.Name_Full_Formatted field known to the internal processing. If we did not use this command, *Discern Explorer* would return an error similar to the following when the layout program was executed:
    %CCL-E-85-1_CCL_ADD_LAYT_TO_PRG(0,0)S58,L7,Rpt{}Report attribute (P.NAME_FULL_FORMATTED) was not selected for retrieval in select clause.
    We do not need to add a similar command for the P.Person_ID field because we use P.Person_ID to create a Head group expression section. Also, we also do not need to add similar commands for the P_Sex_Disp or Age expressions. However, we will need to add similar commands for other fields that we want to display that are not used as a Head/Foot group expression.

15. Use the following code to create a Head P.Person_ID reportwriter section and call the HeadPersonIDSection subroutine:

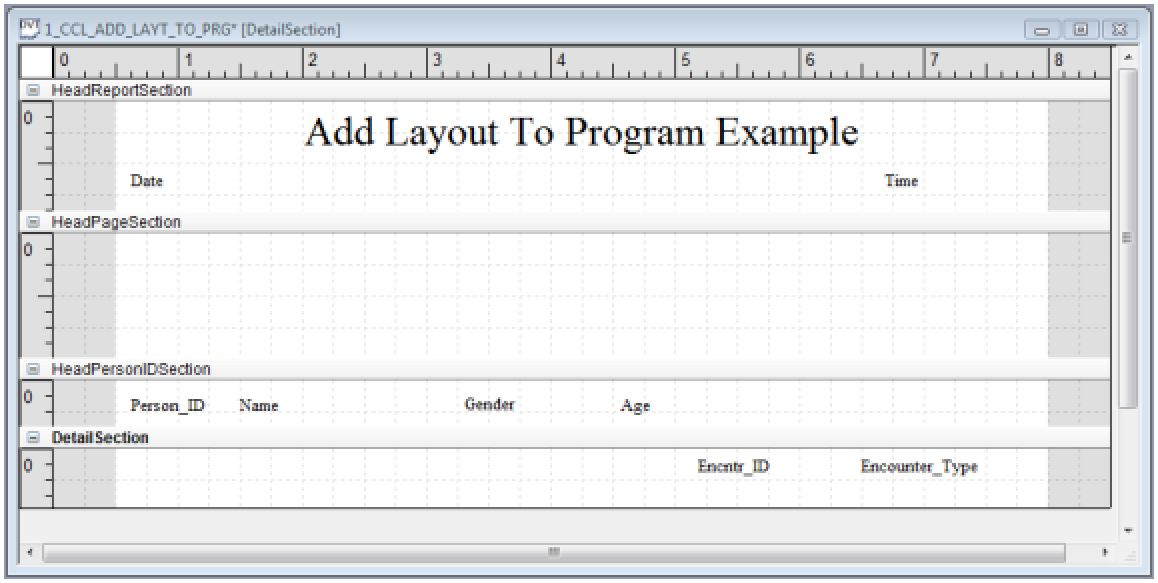
16. After adding the commands to your source code, save the file and include/compile it.
17. Execute the program as a prompt program by using CTRL+R, by clicking Run Prompt Program, or by selecting the Run Prompt Program option from the Build menu.
18. Use MINE as the output device and, when prompted, enter the last name of a person that you know has encounters. Your output should resemble the following screen:



```
                  Add Layout To Program Example
   05/16/08                                                    10:57:43

     1029317.00    Smith, Kerry Ann              Female        38 Years

     1029328.00    Smith, Mark                   Male          28 Years

     1047771.00    Smith, Cassidy                Female        11 Years

     1048095.00    Smith, Wade                   Male          26 Years

     1048100.00    Smith , Matthew               Male          17 Years

     1084985.00    Smith , Colleen Marie         Female         9 Years
```

To continue, we need to add information about each person's encounters to the DetailSection layout section.

19. Close the output window and return to the layout.
20. Use the Text Tool A to add a field towards the right hand side of the DetailSection to display the Encntr_ID. In the Properties dialog box, modify the property to **Encntr_ID** and the Source to **E.Encntr_ID**.
21. Use the Text Tool A to add a field to the right of the Encntr_ID of the DetailSection to show the display value of the encounter type code. In the Properties dialog box, modify the Name property to **Encounter_Type** and the Source to **E_Encntr_Type_Class_Disp**.
22. Use the vertical resize to move the bottom of the DetailSection up to just below the items you added. If necessary, use the scroll bar on the right of the layout to scroll down until you can see the bottom of the DetailSection so you can grab and resize it.



1_CCL_ADD_LAYT_TO_PRG* [DetailSection]

HeadReportSection

Add Layout To Program Example

Date Time

HeadPageSection

HeadPersonIDSection

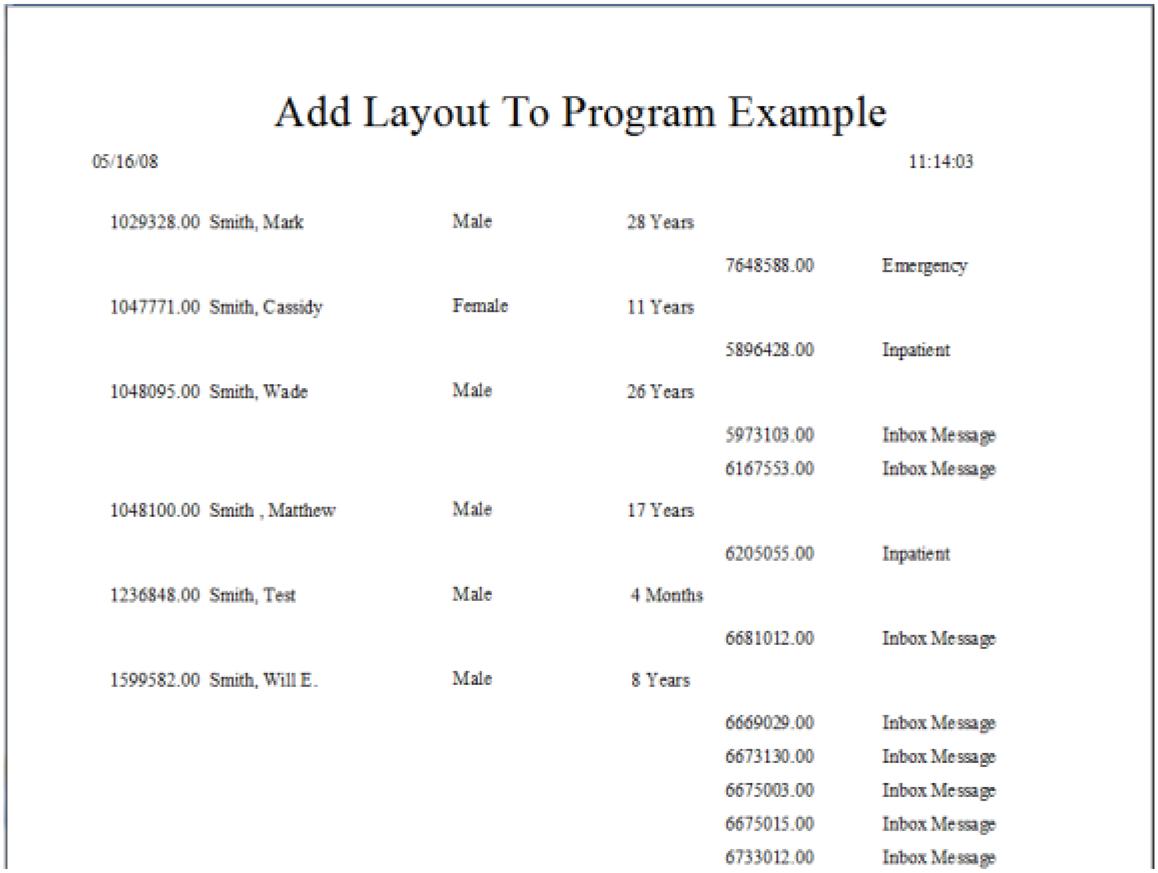
Person_ID Name Gender Age

DetailSection

Encntr_ID Encounter_Type

23. Save the layout.

    You are now ready to add the commands to your source code to execute the DetailSection subroutine in the Detail reportwriter section of the Select command.
24. Add the following command to the Head Report section of your source code:
25. Use the following code to create a Detail reportwriter section and call the DetailSection subroutine:
26. After adding the commands to your source code, save the file and include/compile it.
27. Execute the program as a prompt program by using CTRL+R, by clicking Run Prompt Program, or by selecting Run Prompt Program option from the Build menu.
28. Use MINE as the output device and, when prompted, enter the last name of a person that you know has encounters. You can expect your output to be similar to the following screen:



```
                 Add Layout To Program Example
05/16/08                                                      11:14:03

  1029328.00  Smith, Mark         Male        28 Years
                                                      7648588.00    Emergency
  1047771.00  Smith, Cassidy      Female      11 Years
                                                      5896428.00    Inpatient
  1048095.00  Smith, Wade         Male        26 Years
                                                      5973103.00    Inbox Message
                                                      6167553.00    Inbox Message
  1048100.00  Smith , Matthew     Male        17 Years
                                                      6205055.00    Inpatient
  1236848.00  Smith, Test         Male         4 Months
                                                      6681012.00    Inbox Message
  1599582.00  Smith, Will E.      Male         8 Years
                                                      6669029.00    Inbox Message
                                                      6673130.00    Inbox Message
                                                      6675003.00    Inbox Message
                                                      6675015.00    Inbox Message
                                                      6733012.00    Inbox Message
```

You may notice a couple of things about the output that you want to modify. For example, a person's first encounter is displayed one row below their person information. To align the first encounter with the person information you need to set the Advance Y Position section property to No.

29. Close the output window and return to the layout.
30. Click in the HeadPersonIDSection and in the Properties dialog box; modify the Advance Y Position from Yes to No.
31. Save the layout.
32. Include/compile your source code file.
33. Execute the program as a prompt program.
34. Use MINE as the output device and, when prompted, enter the last name of a person that you know has encounters. When the output is displayed, verify the person's first encounter is displayed on the same line as their person information.
35. Close the output window and return to the layout.

## Generating Page Breaks

You may have noticed that you are getting only one page of output. Depending on the last name that you have been using at the prompt and the amount of data you have in your environment, you may have already noticed that when the output appears to fill a page and you attempt to click the next page button , the next page and previous page buttons are unavailable and your output will still display the same page. If your query has not returned enough data to fill the first page, try entering a different name or part of a name, followed by an asterisk ( * ) to qualify enough data so that more than one page is required to display it. The output only has one page because, up to this point, we have only added code to call the layout section subroutines. We have not added the code needed to generate page breaks when the amount of space used by the subroutines exceeds the amount of space on the page. Since we are calling the subroutines, we also need to control when a page break is created. It is important to realize that since the code does not generate a page break, any information that does not fit on the page will not be displayed when we attempt to render.

When you first create a layout, the Report Properties default paper size is 8.5 by 11 inches with .5 inch margins and portrait orientation. The default properties can be modified using the Report > Report Properties command from the Edit menu. Layout Builder subroutines use a variable named _YOffset to track the position going down the page. Using the default settings, the bottom line on the page is rendered 10.5 inches down from the top of the page. When the height of a layout section, plus the _YOffset is greater than the height of the page minus the bottom margin, the program needs to generate a page break before the section is rendered. Using the defaults, if the height of a layout section plus the current value of _YOffset is greater than 10.5, a page break must be generated before rendering the layout section.

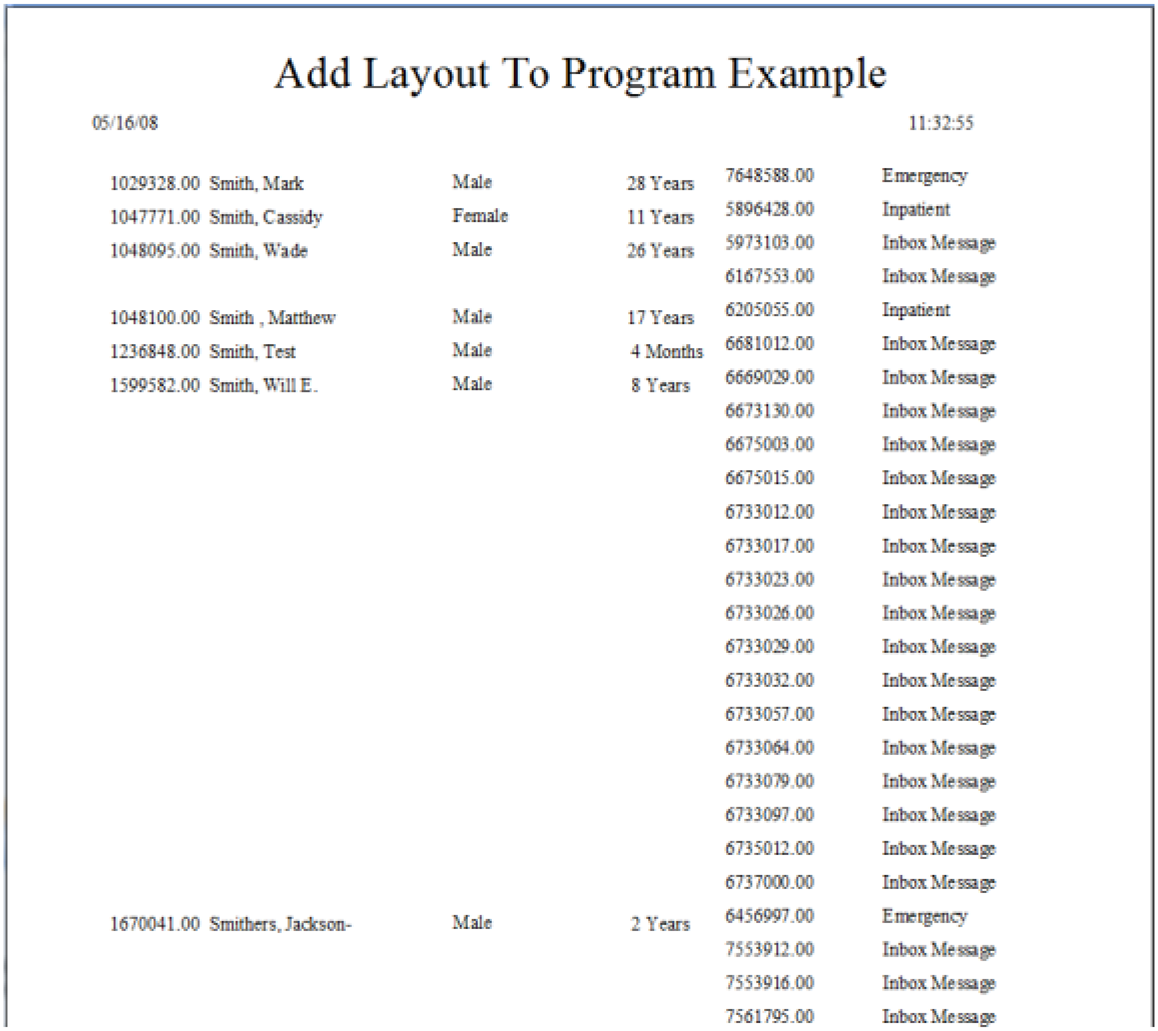
Two steps are required to generate the page break. First, the Break command needs to be issued to force *Discern Explorer* to execute the Head/Foot Page reportwriter sections. Next the PageBreak() layout subroutine, created by Layout Builder, must be executed to generate the page break command and reset *YOffset to the top margin of the page. A layout section subroutine is called using the format _Layout_Section_Subroutine(parameter)* where *parameter* is either Rpt_Render or Rpt_CalcHeight. When passing either Rpt_Render or Rpt_CalcHeight, the height of the layout section is returned. If Rpt_Render is used, the layout section is rendered and the _YOffset variable is incremented. If Rpt_CalcHeight is used, the layout section is not rendered and the _YOffset variable is not incremented.

1. Add the following IF statement to the top of your reportwriter detail section to generate a page break if the layout DetailSection will not fit on the current

page:

2. Add the following statement after your reportwriter Head Report section to create a reportwriter Head Page section and generate a page break when necessary:

3. After adding the commands to your source code, save the file and include/compile it.
4. Execute the program as a prompt program.
5. Use MINE as the output device and, when prompted, enter the last name of a person that you know has encounters. You can expect your output to be similar to the following screen:



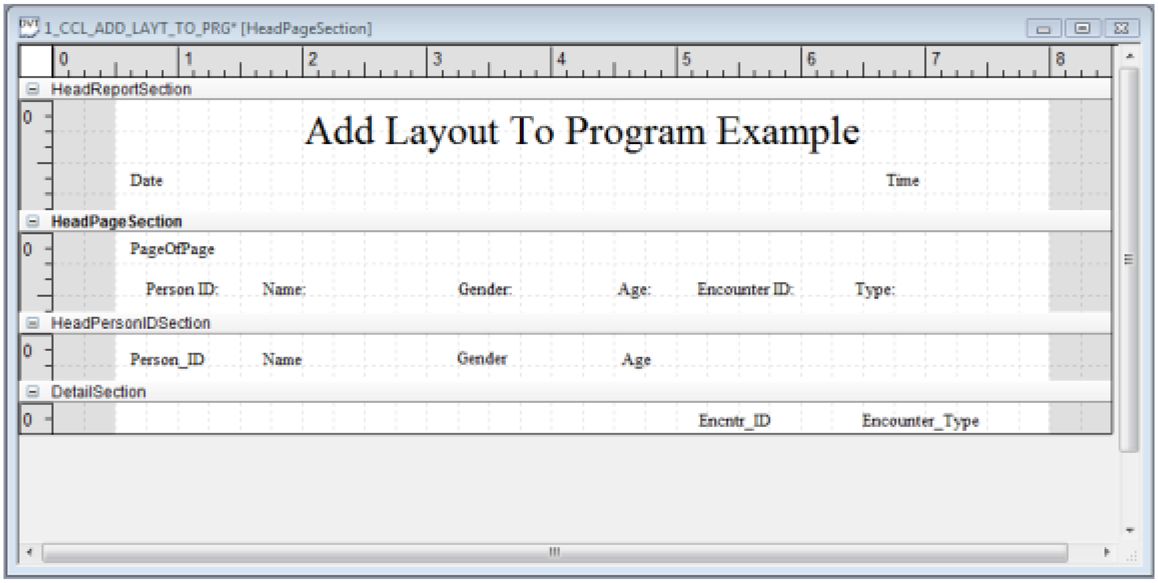
```
                    Add Layout To Program Example
  05/16/08                                                        11:32:55

   1029328.00  Smith, Mark          Male       28 Years   7648588.00   Emergency
   1047771.00  Smith, Cassidy       Female     11 Years   5896428.00   Inpatient
   1048095.00  Smith, Wade          Male       26 Years   5973103.00   Inbox Message
                                                          6167553.00   Inbox Message
   1048100.00  Smith , Matthew      Male       17 Years   6205055.00   Inpatient
   1236848.00  Smith, Test          Male       4 Months   6681012.00   Inbox Message
   1599582.00  Smith, Will E.       Male       8 Years    6669029.00   Inbox Message
                                                          6673130.00   Inbox Message
                                                          6675003.00   Inbox Message
                                                          6675015.00   Inbox Message
                                                          6733012.00   Inbox Message
                                                          6733017.00   Inbox Message
                                                          6733023.00   Inbox Message
                                                          6733026.00   Inbox Message
                                                          6733029.00   Inbox Message
                                                          6733032.00   Inbox Message
                                                          6733057.00   Inbox Message
                                                          6733064.00   Inbox Message
                                                          6733079.00   Inbox Message
                                                          6733097.00   Inbox Message
                                                          6735012.00   Inbox Message
                                                          6737000.00   Inbox Message
   1670041.00  Smithers, Jackson-   Male       2 Years    6456997.00   Emergency
                                                          7553912.00   Inbox Message
                                                          7553916.00   Inbox Message
                                                          7561795.00   Inbox Message
```

Provided your query returned enough data you are able to view multiple pages of output.

## Adding and Calling Additional Layout Sections

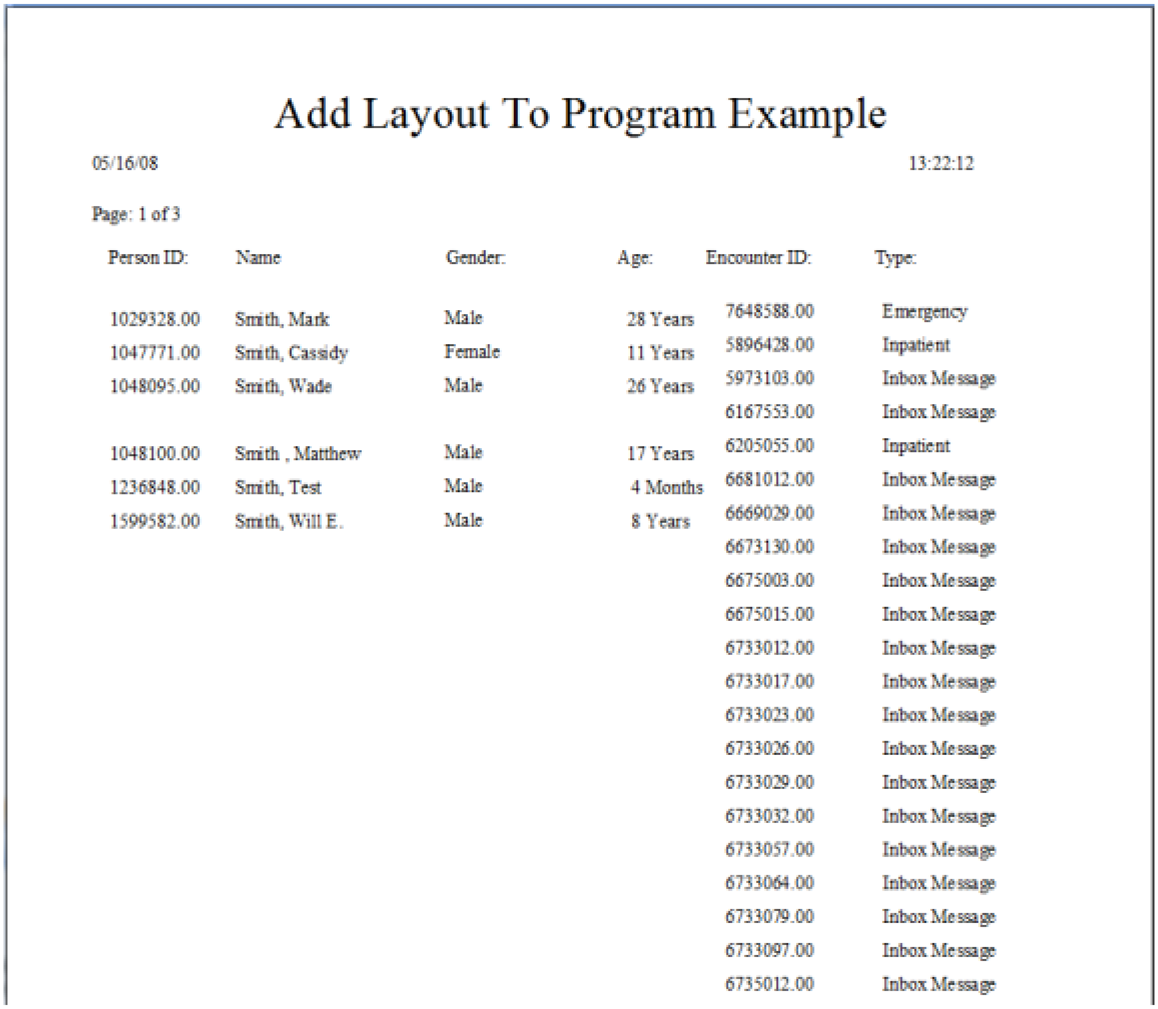
To make the output more readable, you can add column headings and page numbers to the HeadPageSection layout section.

1. Close the output and return to the layout.
2. Use the Text Tool to add a field to the HeadPageSection to display the special Rpt_PageOfPage variable created by Layout Builder. In the Properties dialog box, modify the Name property to **PageOfPage** and the Source to **Rpt_PageOfPage**.
3. Use the Label Tool to place the column headings Person ID:, Name:, Gender:, Age:, Encounter ID:, and Type: in the HeadPageSection of your layout.
4. Use the vertical resize to move the bottom of the HeadPageSection up to just underneath the items you just added.
5. Save the layout. Your layout should look similar to the following example:



6. Add the following code to your reportwriter Head Page section to render the HeadPageSection layout section:

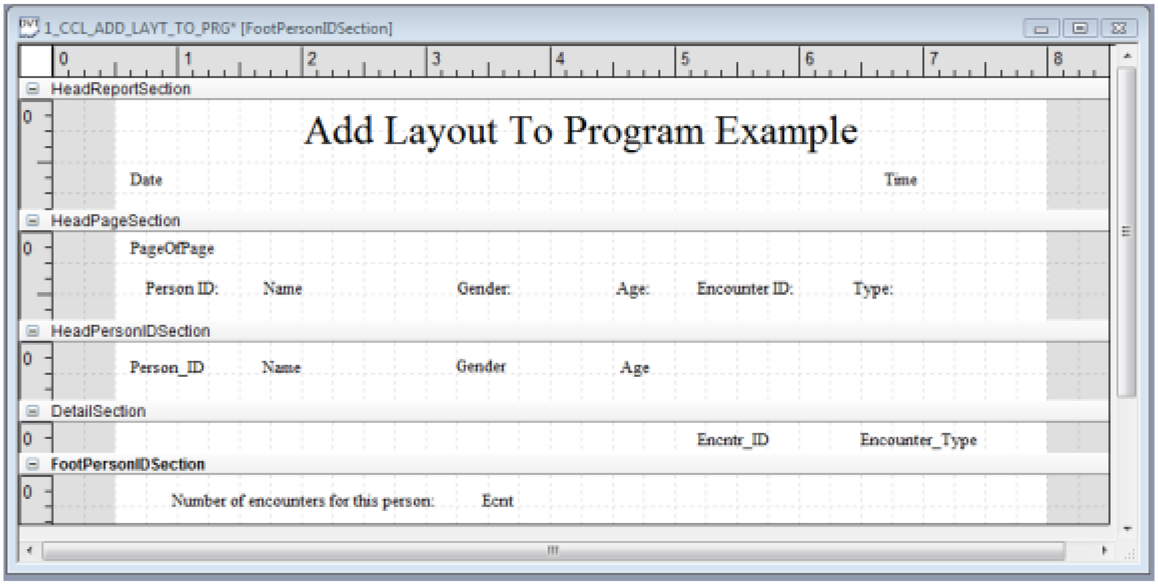
7. After adding the commands to your source code, save the file and include/compile it.
8. Execute the program as a prompt program.
9. Use MINE as the output device and, when prompted, enter the last name of a person that you know has encounters. You can expect your output to be similar to the following screen:



```
                 Add Layout To Program Example
05/16/08                                                        13:22:12

Page: 1 of 3

  Person ID:   Name               Gender:        Age:   Encounter ID:   Type:

  1029328.00   Smith, Mark        Male        28 Years   7648588.00   Emergency
  1047771.00   Smith, Cassidy     Female      11 Years   5896428.00   Inpatient
  1048095.00   Smith, Wade        Male        26 Years   5973103.00   Inbox Message
                                                         6167553.00   Inbox Message
  1048100.00   Smith , Matthew    Male        17 Years   6205055.00   Inpatient
  1236848.00   Smith, Test        Male        4 Months   6681012.00   Inbox Message
  1599582.00   Smith, Will E.     Male         8 Years   6669029.00   Inbox Message
                                                         6673130.00   Inbox Message
                                                         6675003.00   Inbox Message
                                                         6675015.00   Inbox Message
                                                         6733012.00   Inbox Message
                                                         6733017.00   Inbox Message
                                                         6733023.00   Inbox Message
                                                         6733026.00   Inbox Message
                                                         6733029.00   Inbox Message
                                                         6733032.00   Inbox Message
                                                         6733057.00   Inbox Message
                                                         6733064.00   Inbox Message
                                                         6733079.00   Inbox Message
                                                         6733097.00   Inbox Message
                                                         6735012.00   Inbox Message
```

Since you are creating the layout section subroutines using Layout Builder and then calling them in your reportwriter sections, you can use any of *Discern Explorer* commands allowed in a reportwriter section, such as to create variables, load record structure elements, and so on. You can then reference those items in the layout sections. For example, suppose you wanted to display a count of the number of encounters for each person. This can be accomplished by using the aggregate Count() function in the Foot P.Person_ID reportwriter section to set a local reportwriter variable, and then by using that variable as the source for a field in the layout.

10. Close the output and return to your source code file.
11. Switch to Layout Builder and click in the DetailSection to ensure it is active.
12. From the Edit menu, select Insert > Section to add a new section to the layout. A new layout section is inserted before the DetailSection.
13. Move the new section below the DetailSection by placing your cursor on the rule of the new section and drag the cursor to the middle of the DetailSection's ruler.
14. In the Properties Dialog box, modify the Name property to **FootPersonIDSection**.
15. Use the Label Tool Aa to add the following text to the FootPersonIDSection on your layout: **Number of encounters for this person**.
16. Use the Text Tool A to add a field to the FootPersonIDSection, and in the Properties dialog box modify the Name property to **Ecnt** and the Source to **Ecnt** (Ecnt is the name of a variable you will create in the Foot P.Person_ID reportwriter section of your source code file using the Count() function).
17. Use the vertical resize to move the bottom of the FootPersonIDSection up to just below the items you have added; leave a little blank space after these items to ensure a better display of the output. Your layout should look similar to the following example:



1_CCL_ADD_LAYT_TO_PRG* [FootPersonIDSection]

HeadReportSection

Add Layout To Program Example

Date	Time

HeadPageSection

PageOfPage

Person ID:	Name	Gender:	Age:	Encounter ID:	Type:

HeadPersonIDSection

Person_ID	Name	Gender	Age

DetailSection

Encntr_ID	Encounter_Type

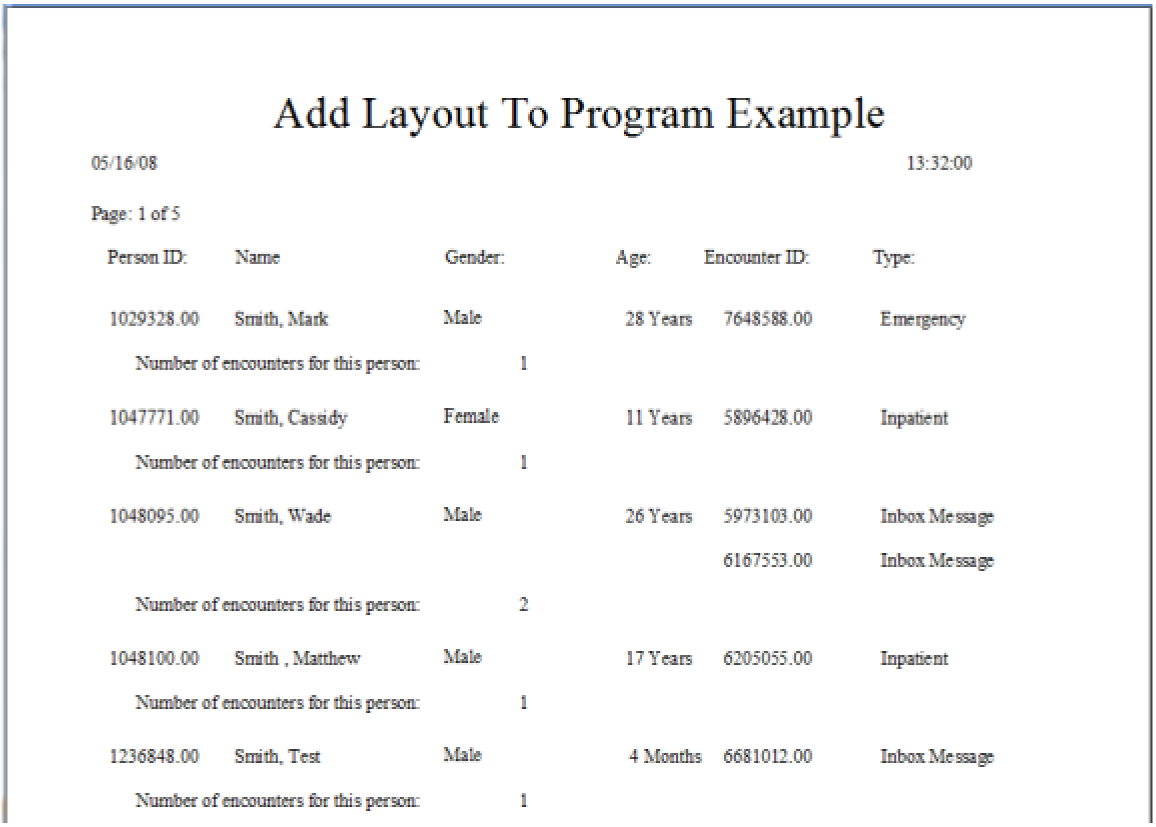
FootPersonIDSection

Number of encounters for this person:	Ecnt

18. Save the layout.
19. Add the following code after the Detail reportwriter section:

Notice in the example above that the Foot Page and Foot Report sections are added to conform to the *Discern Explorer* recommendation of having a matching foot reportwriter section for every head reportwriter section.
Your source code should resemble the following example:

20. Save the source code file and include/compile it.
21. Execute the program as a prompt program.
22. Use MINE as the output device and, when prompted, enter the last name of a person that you know has encounters. You can expect your output to be similar to the following screen:



```
                    Add Layout To Program Example
05/16/08                                                         13:32:00

Page: 1 of 5
 Person ID:   Name              Gender:         Age:    Encounter ID:    Type:

 1029328.00   Smith, Mark       Male          28 Years   7648588.00    Emergency
    Number of encounters for this person:    1

 1047771.00   Smith, Cassidy    Female        11 Years   5896428.00    Inpatient
    Number of encounters for this person:    1

 1048095.00   Smith, Wade       Male          26 Years   5973103.00    Inbox Message
                                                         6167553.00    Inbox Message
    Number of encounters for this person:    2

 1048100.00   Smith , Matthew   Male          17 Years   6205055.00    Inpatient
    Number of encounters for this person:    1

 1236848.00   Smith, Test       Male          4 Months   6681012.00    Inbox Message
    Number of encounters for this person:    1
```

# Executing Programs With Layouts

Up to this point, you have executed your program with a layout as a prompt program in DVDev. Most likely you will create programs with layouts that need to be executed by people who do not have access to DVDev. When you include/compile the source code that calls the subroutines you created using Layout Builder, a *Discern Explorer* program is created in the object library. This program can be executed from Explorer Menu (ExplorerMenu.exe) just like any other *Discern Explorer* program. Simply use the program name when adding a program item to the Explorer Menu.

# Moving Your Program With a Layout to Another Environment

Cerner recommends creating and testing all layout programs in a non-production environment. After the program has been created and tested, it can be moved into the production environment. Moving a layout program to a different environment requires you to export the layout program from a source environment (where the program is currently located) into a target environment (where you want the program to be moved).
If you have a front-end file share that can be accessed from both environments, an easy way to move the program with a layout is to open the *program_name* .PRG file in DiscernVisualDeveloper.exe (DVDev) and use the Export command from the File menu to export the file to the front-end file share. The export creates a *program_name*.DVT file. If you have a prompt form associated with the program, the export also creates a *program_name*.DPB file. You can then close DVDev and reopen it connected to the target environment. Use the Import command from the File menu to import the .PRG file from the common front-end file share. The import process uses the .DPB file to re-create the prompt form and the .DVT file to re-create the .DVL file automatically. You are prompted to save the prompt form. When the import completes, include/compile the .PRG file to create the object in the object library.

If you do not have a front-end file share that can be accessed from both environments, but do have access to a different front-end file share from each environment, the process is basically the same. You need to copy the .PRG file, the .DBP file, and the .DVT file to the front-end file share that can be accessed from the target environment before performing the import. The .DPB file is a binary file, so FTP must be used in binary mode when copying the file to the target environment.

If you cannot access front-end file shares, you can move the program and layout to the target environment using the back-end, end file structure. In addition to copying the .PRG file, this method also requires you to export and import the prompt form and the layout. To export the prompt form, select Transfer Objects from the Tools menu to open the Transfer Objects dialog box. Ensure that the Prompt Forms category is selected in the tree on the left side of the Transfer Objects dialog. Enter the name of your layout program in the Source Object: field. Use the Save to Backend... option on the Task menu to save the prompt form to a backend file with a .DPB extension. Use binary FTP to copy the .DPB file to a directory in the target environment. The .DPB file is a binary file, so FTP must be used in binary mode when copying the file to the target environment.

To export the layout, select Transfer Objects from the Tools menu to open the Transfer Objects dialog box. Ensure that the Associated Layouts category under Layouts is selected in the tree on the left side of the Transfer Objects dialog. Enter the name of your layout program in the Source Object: field. Use the Save to Backend... option on the Task menu to save the layout to a back-end file with a .DVT extension. Use ASCII FTP to copy the .DVT file to a directory in the target environment. Use ASCII FTP to copy the .PRG file to a directory in the target environment. Open DVDev in the target environment and select File > Open > Source to open the .DPB file. Opening the .DPB file opens Prompt Builder and imports the prompt form. Save the prompt form and close the Prompt Builder. Use File > Open > Source to open the .DVT file. Opening the .DVT file opens Layout Builder and imports the layout. Save the layout and close the Layout Builder.

Use File > Open > Source to open the .PRG file. Place the cursor inside the Create Program/End Go commands and use Tools > Layout Builder to open the layout. Save the layout. Include/Compile the .PRG file to create the program object in the new environment.

You have finished the last part of Use Discern Layout Builder 2007.18. For more information, see Appendix A.