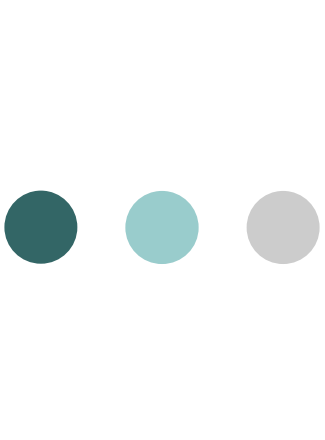


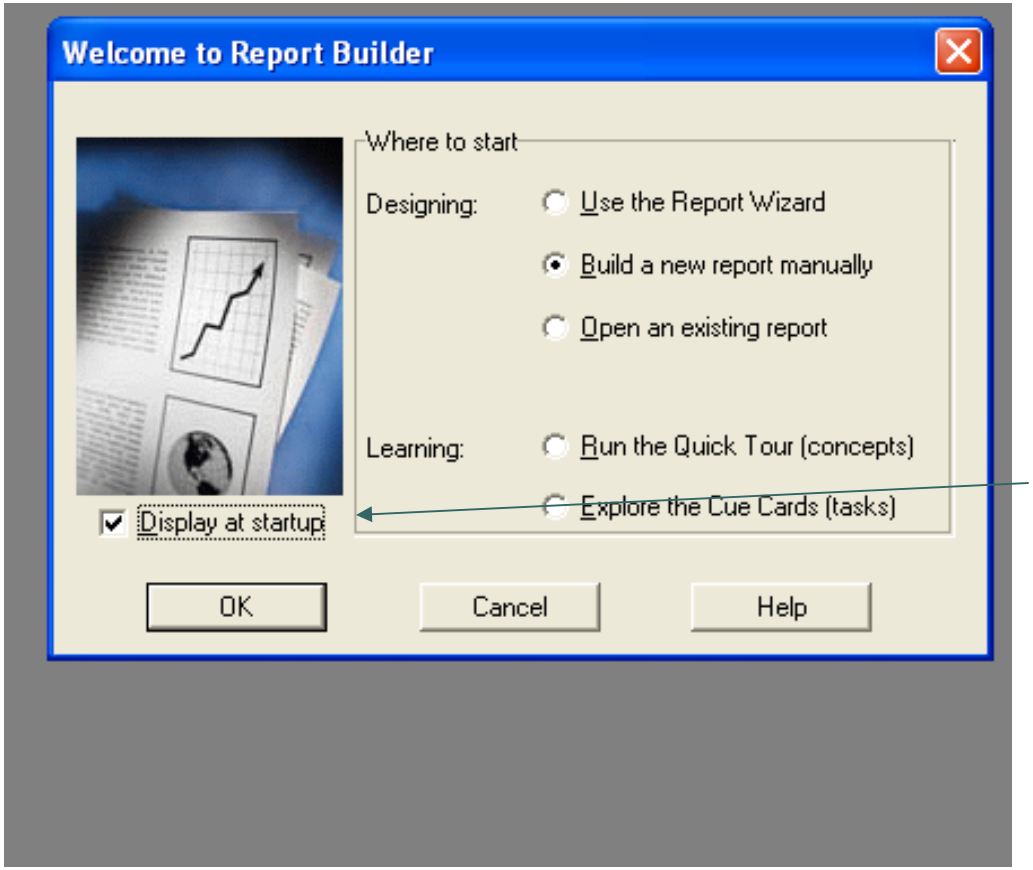
Reports 6i

Thomas Korbecki
oAppsNET Partners, LLC

Lab One specific topics:

- Generate reports in Character Mode for WYSIWYG
- How functions work within a report
- Are there pro's/con's to using multiple queries vs. a single query with joins in a report?
- How best to manage groups of data so that little or no intervention is needed to the report layout.
- Best approach to adding an additional field to a report layout.
- Layout navigation - Secrets to moving or centering objects.
- Layout (grouping, breaks, use of different formats, page setups)
- Creating and dropping temp tables in reports. Doing massive amounts of "setup" queries before the reports can be generated. Or, failing that, calling a report from PL/SQL.

Reports Welcome Dialog



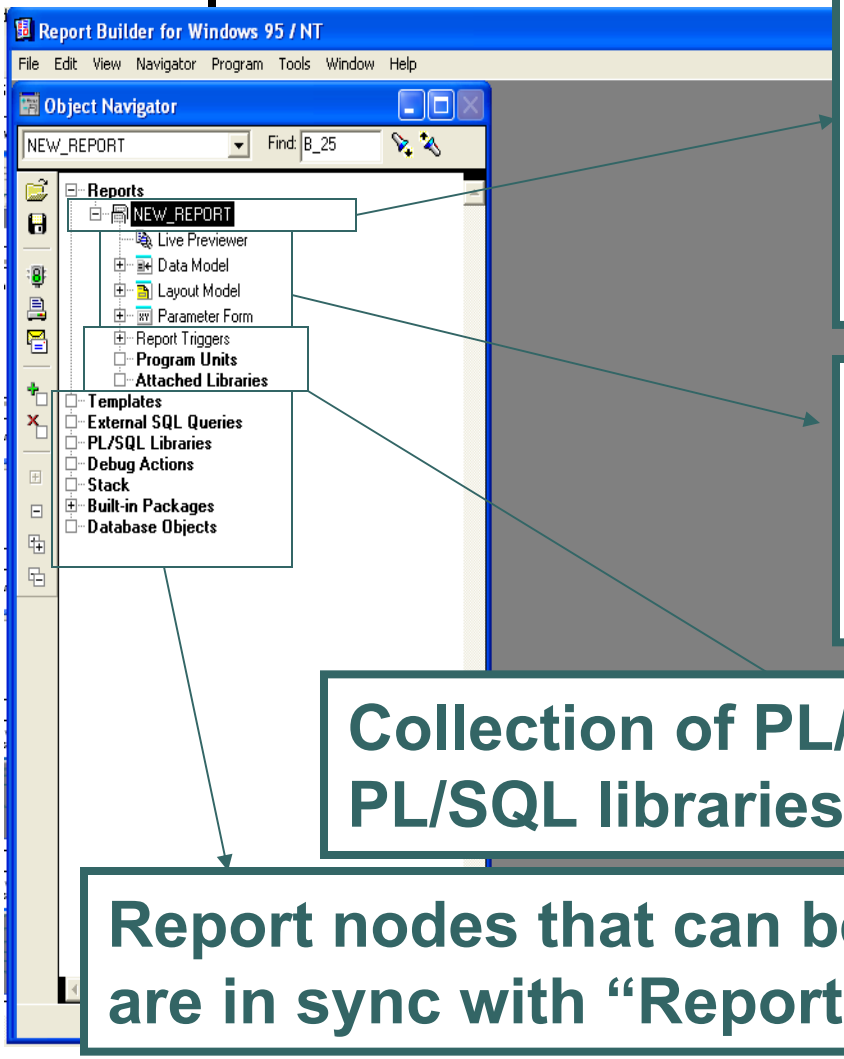
When you first open Reports 6i, the reports welcome dialog box opens. For most purposes, we will build our on report manually, or open an existing.

To suppress window:
Uncheck ‘Display at Startup’.

To display the welcome dialog box later:

1. Select Edit > Preferences
2. Select Wizards Tab
3. Select the Welcome Dialog check box.

Navigation Overview



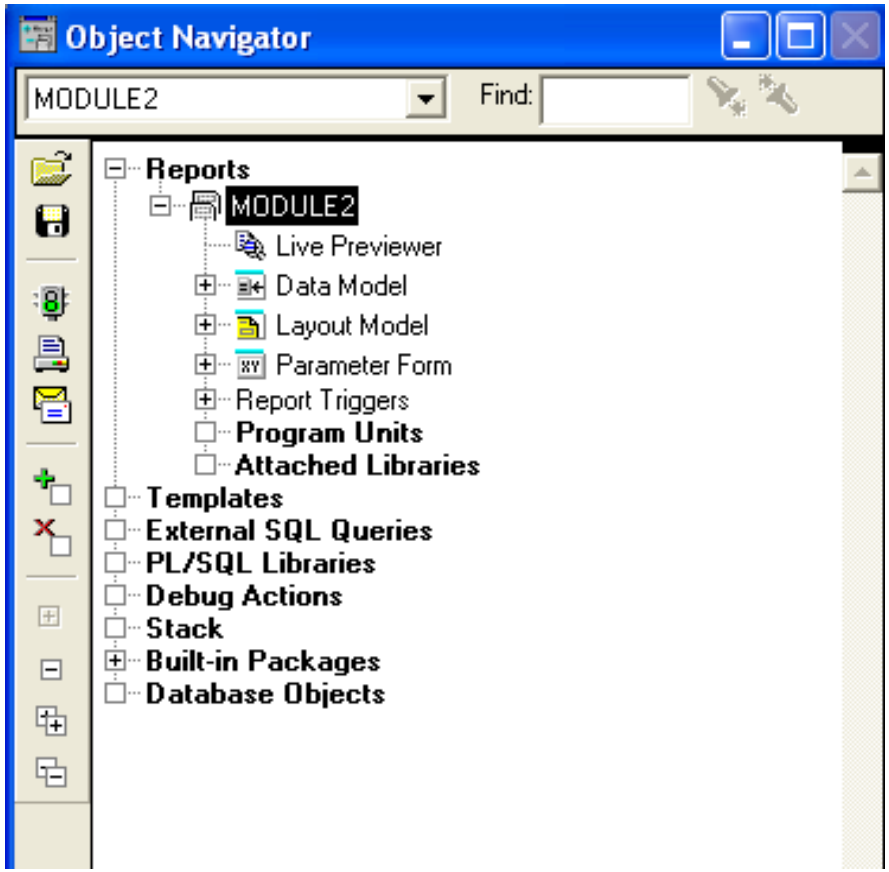
Modules: Top level objects that can be referenced by other modules (I.e., Oracle Forms)

Report Level Objects: Objects used to build the report but can not be referenced by other reports

Collection of PL/SQL source code and attached PL/SQL libraries specific to the report

Report nodes that can be utilized by any report…they are in sync with "Reports"

Object Navigator Properties

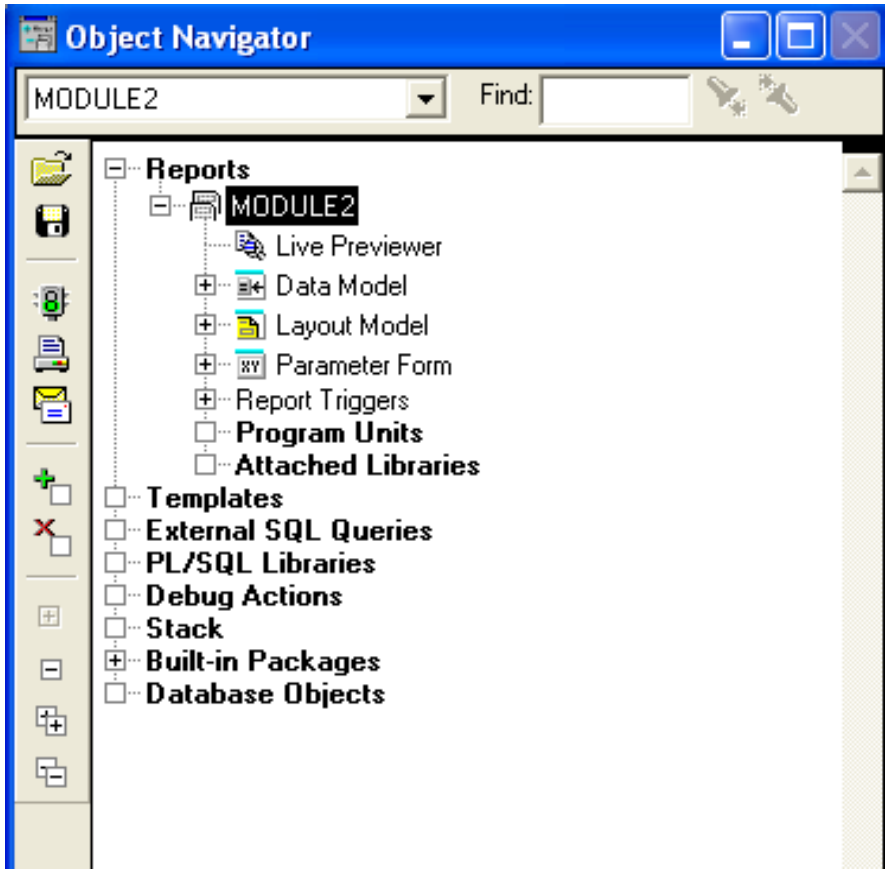


The Object Navigator shows a hierarchical relationship between all objects of the report.

The object navigator enables you to locate and manipulate objects quickly and easily.

Use the Find field to quickly locate items, searching forward and backward through any level.

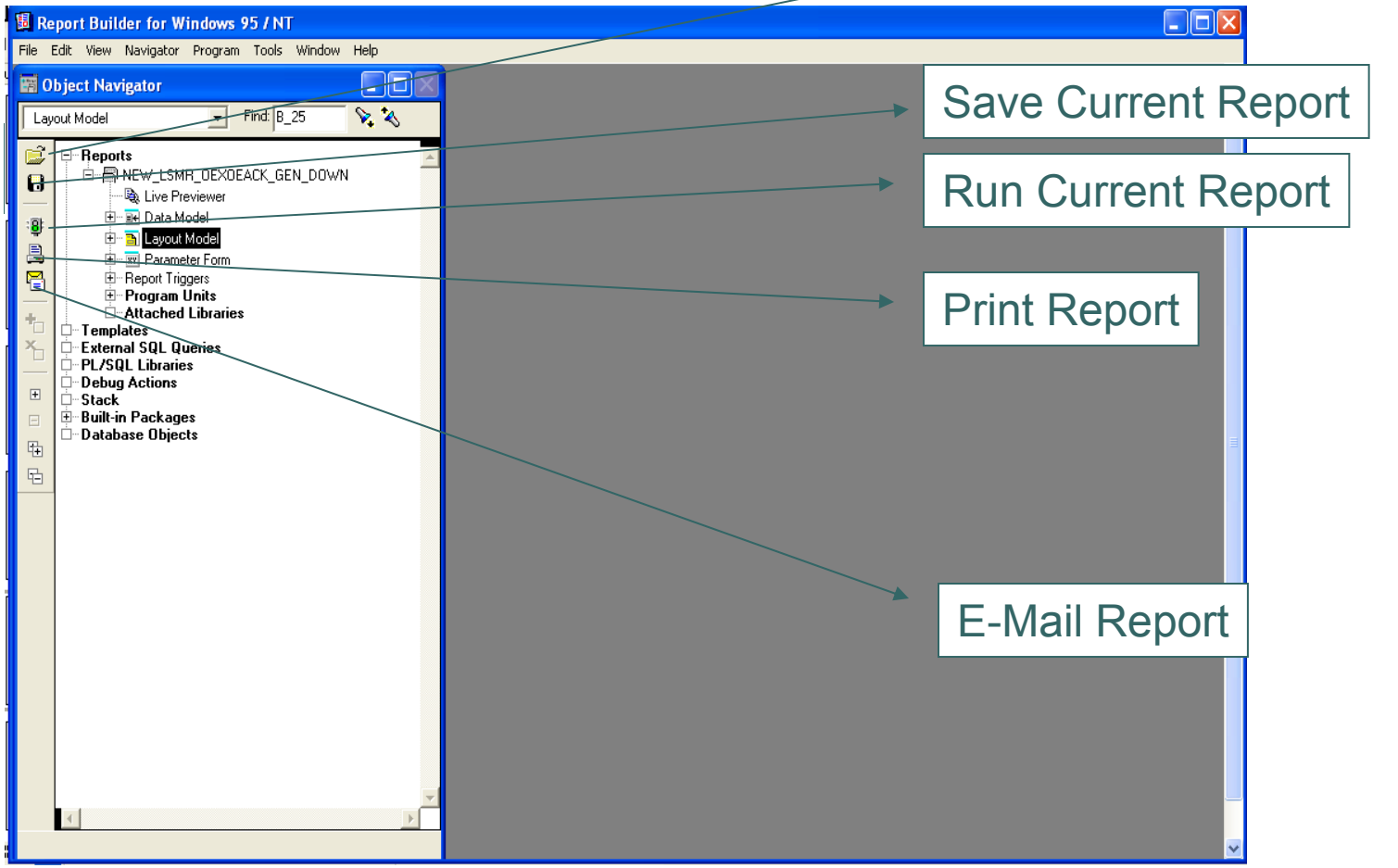
Object Navigator Properties



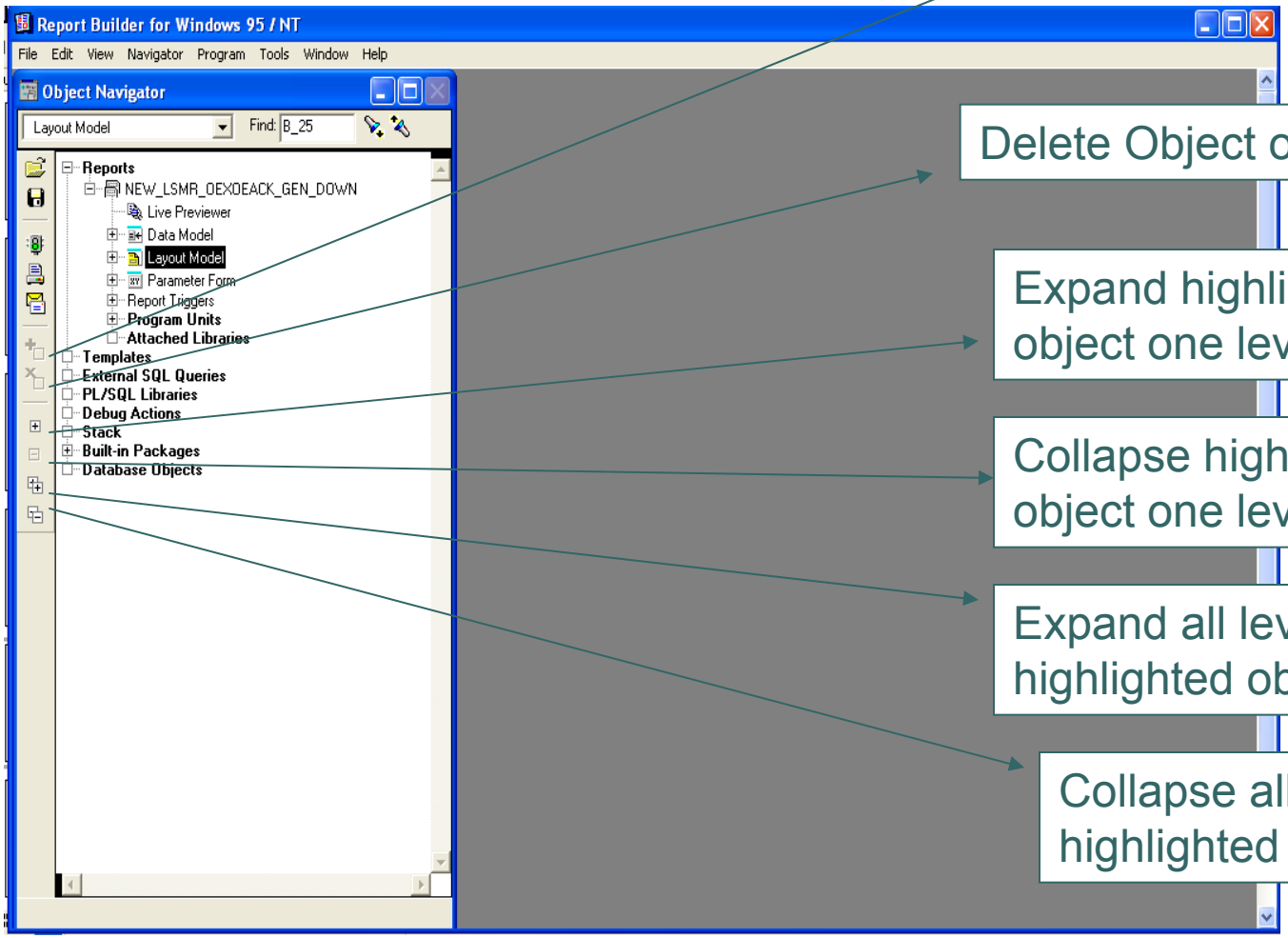
Just point and click:

‘+’ signs are expandable
‘-’ signs are collapsible
‘Boxes’ are lowest level objects and have no associated object below them

Object Navigator Properties



Object Navigator Properties



Add Object of highlighted type

Delete Object of highlighted type

Expand highlighted object one level

Collapse highlighted object one level

Expand all levels of highlighted object

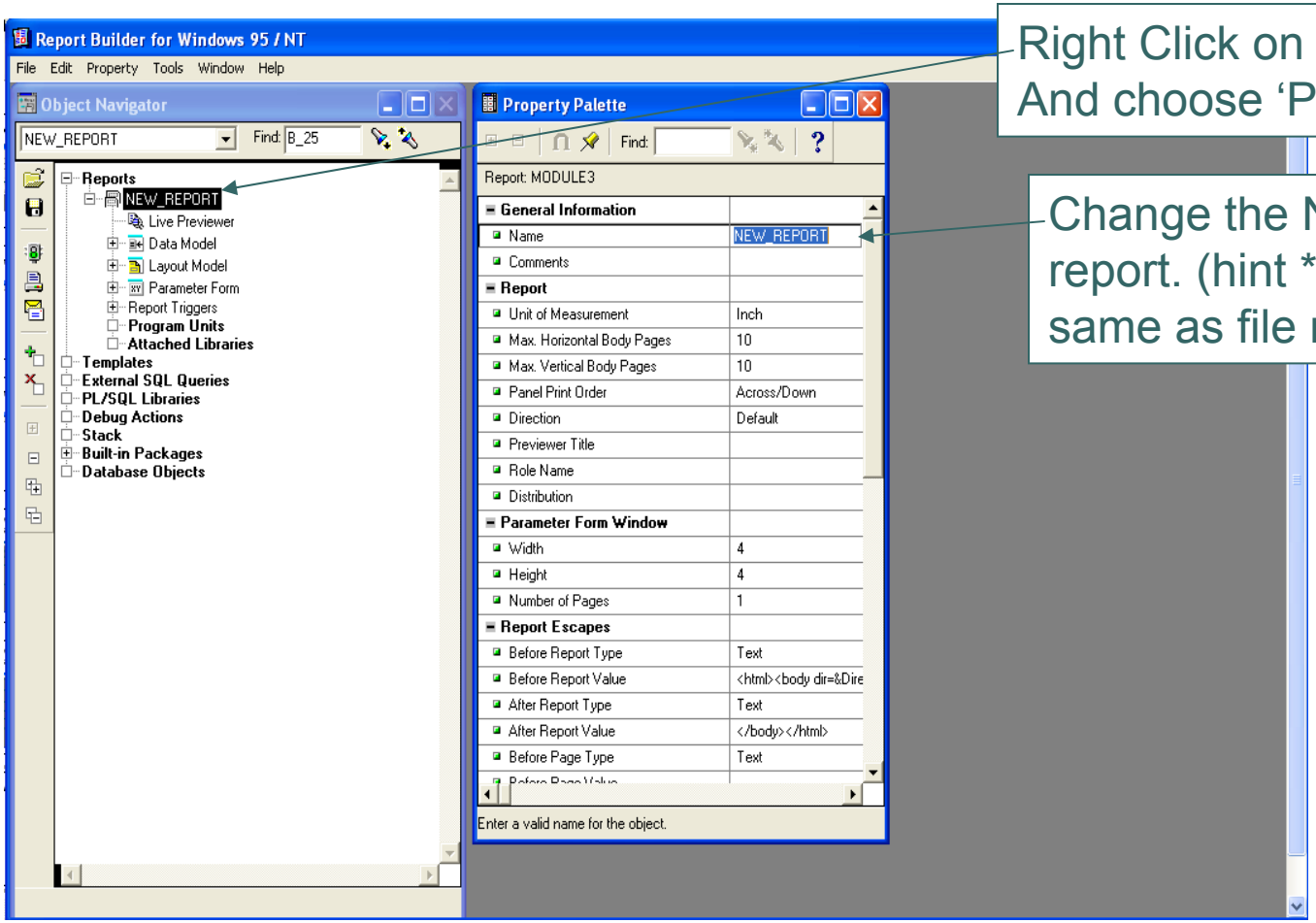
Collapse all levels of highlighted object

Hint

- Collapse branches as you get done with them. Oracle Reports contain MANY objects and the longer you wait to collapse objects the harder it will be to decipher the Object Navigator objects.

Object Navigator

Property Palette- Name

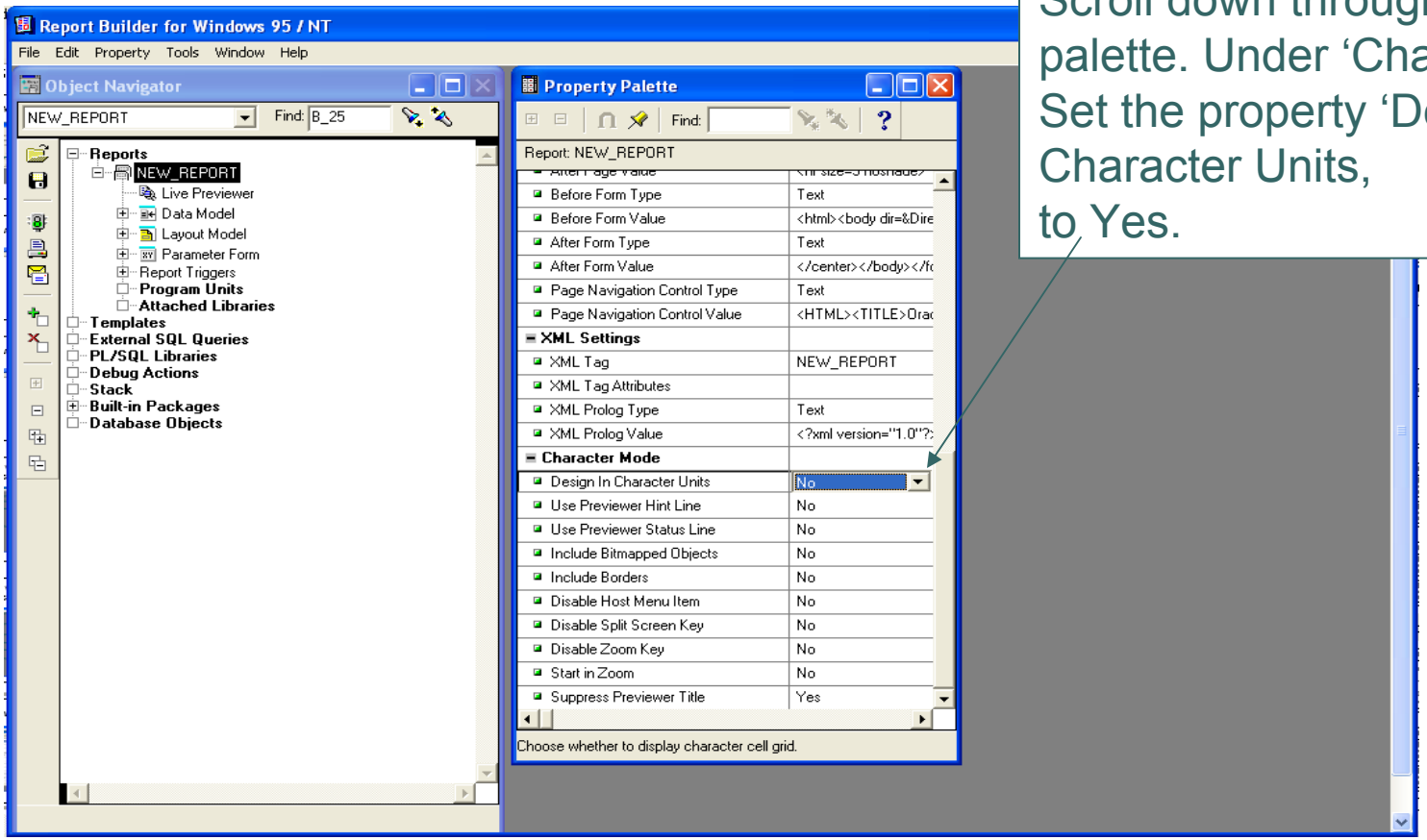


Right Click on ‘module name, And choose ‘Property Palette’

Change the Name of the report. (hint * name this the same as file name)

Object Navigator

Property Palette - Set Character Mode



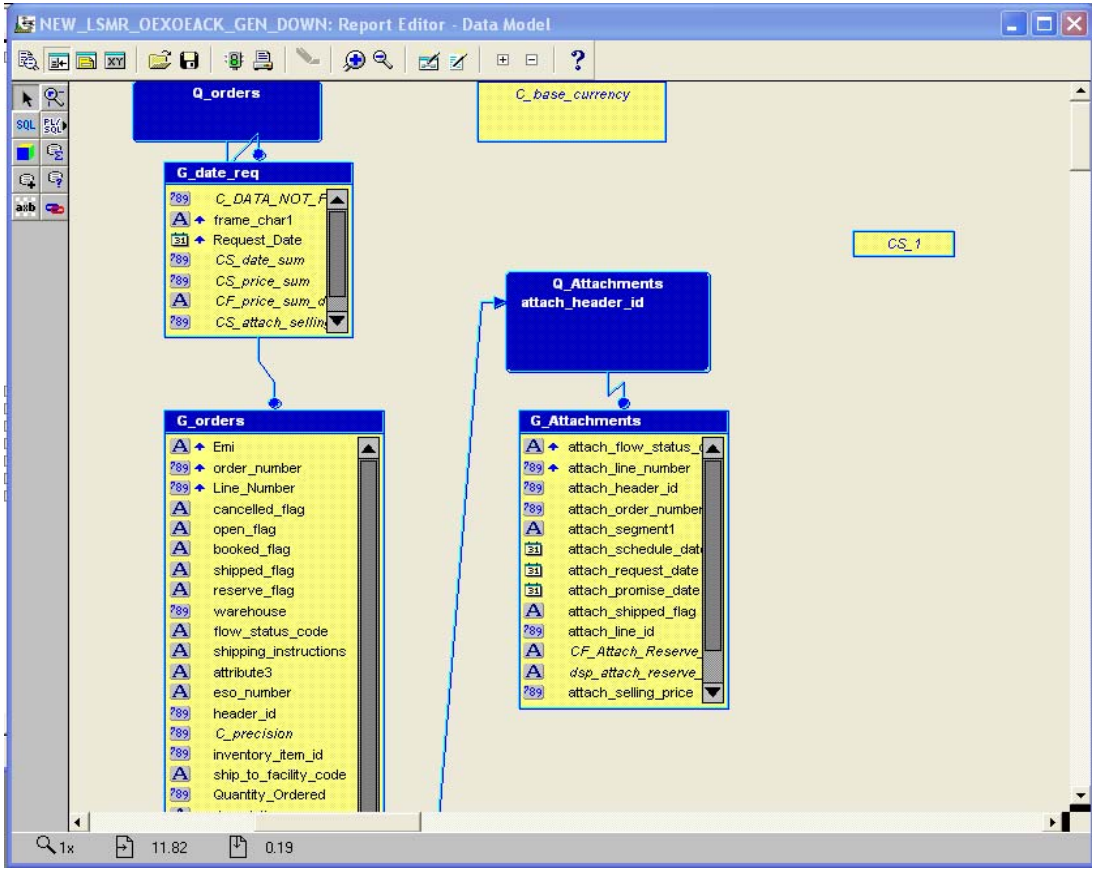
Scroll down through the property palette. Under ‘Character Mode’, Set the property ‘Design In Character Units,
to Yes.

Hint

Property Palette - Set Character Mode

Always set new reports to Character mode for Oracle Applications. It is best to do this before a layout is started, because once you have built a layout and change to character mode, you can run into problems.

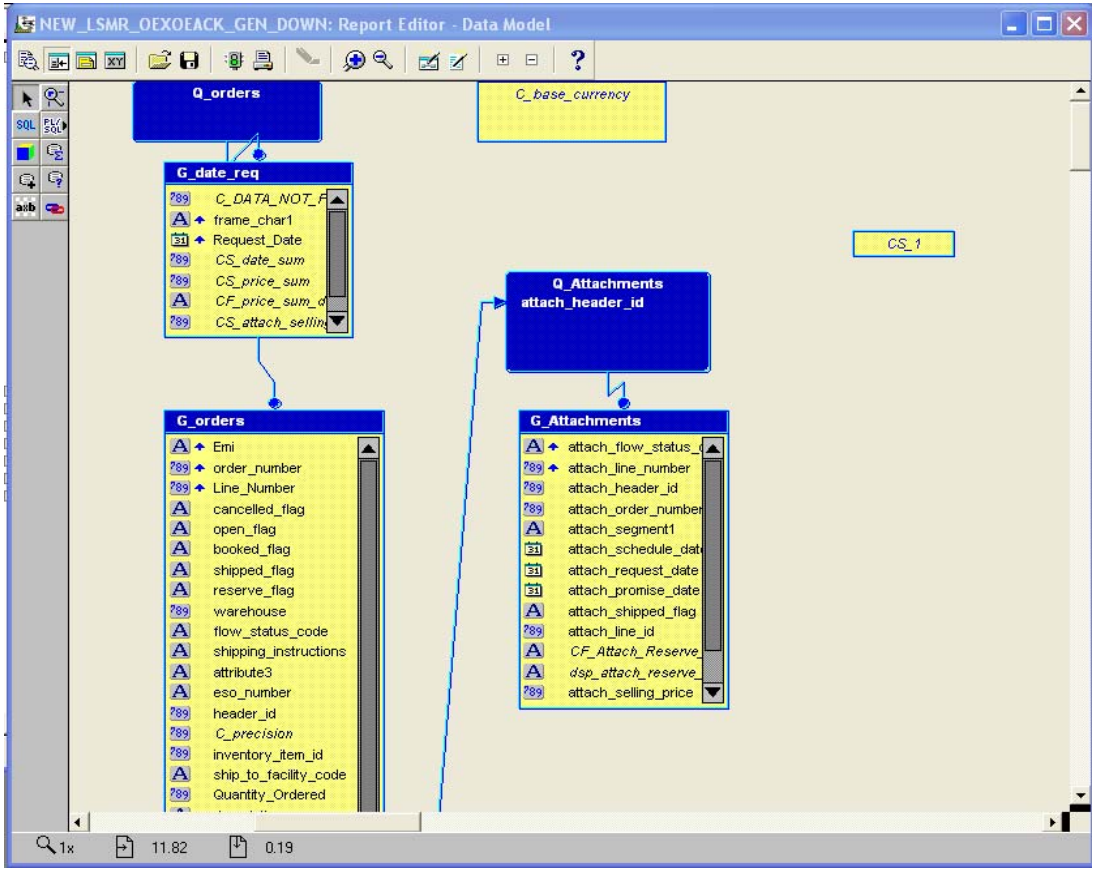
Data Model Overview



DATA MODEL OVERVIEW:

The Data Model editor is the workspace where you select and define data for your report. You are essentially calling in column and table definitions from the database

Data Model Overview



DATA MODEL OVERVIEW:

Using the tools in the tool palette, you can create queries, groups, columns and parameters. You can also link two separate queries together

Data Model Overview

Tool Palette

Select Object

Create Query

Express Query

Formula Column

Cross Product

Magnify

Ref Cursor Query

Summary Column

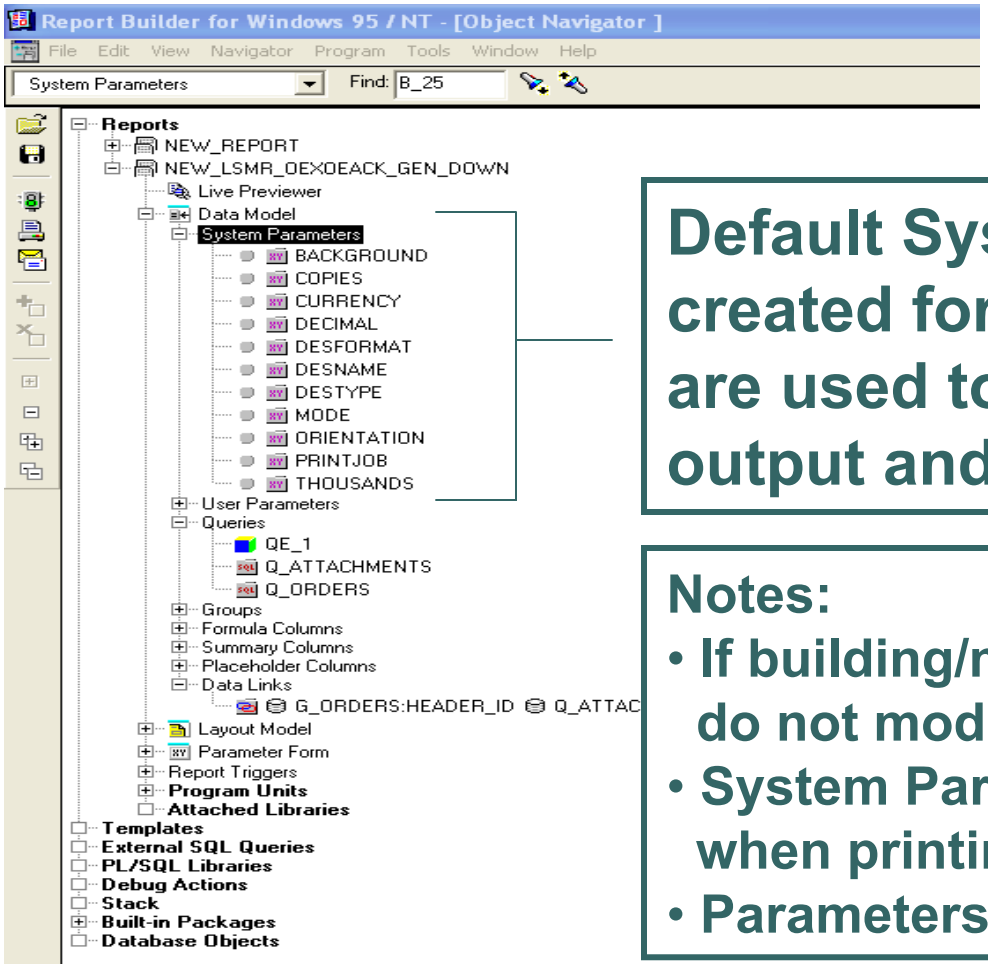
Placeholder Column

Data Link – to link two queries together

If Tool Palette is not visible in Data Model,
Go to:
View > Tool Palette

Data Model Overview

System Parameters



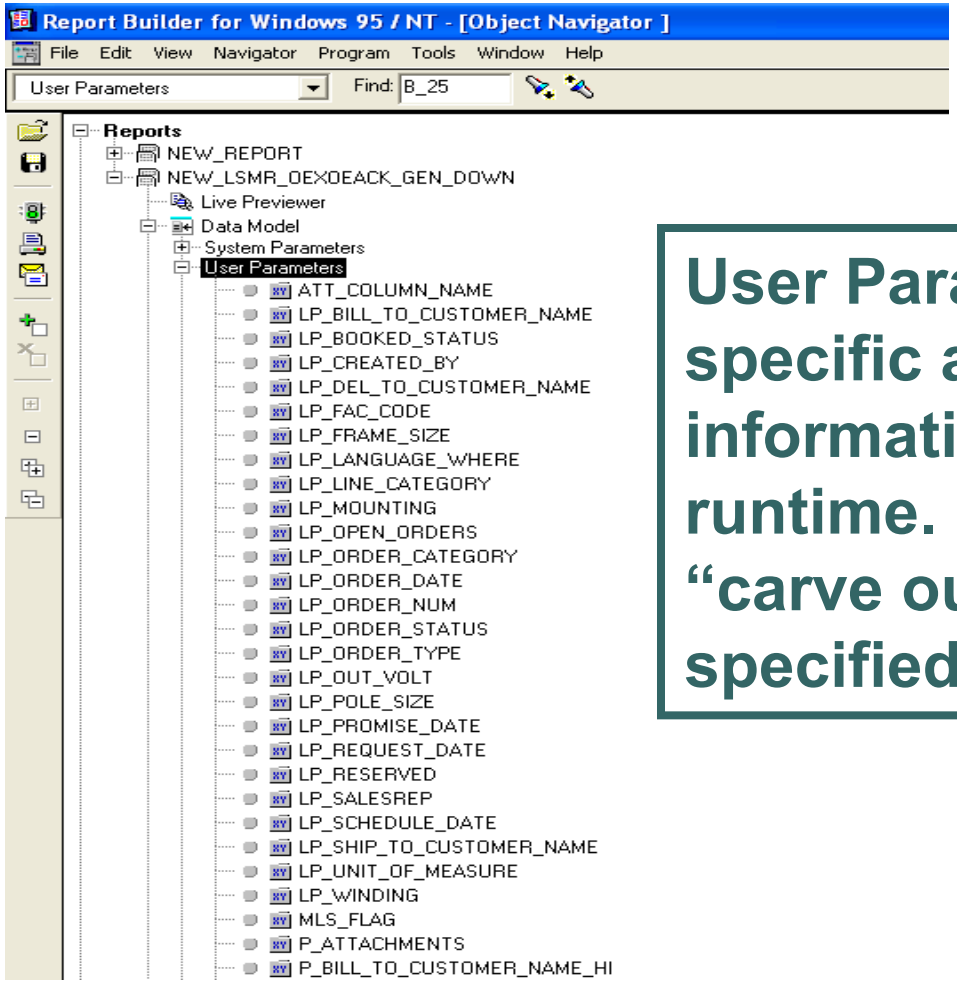
Default System Parameters are created for every report. They are used to modify/control report output and printing activity.

Notes:

- **If building/modifying reports for apps, do not modify in Reports 6i**
- **System Parameters come into play when printing from operating system**
- **Parameters can be modified at runtime**

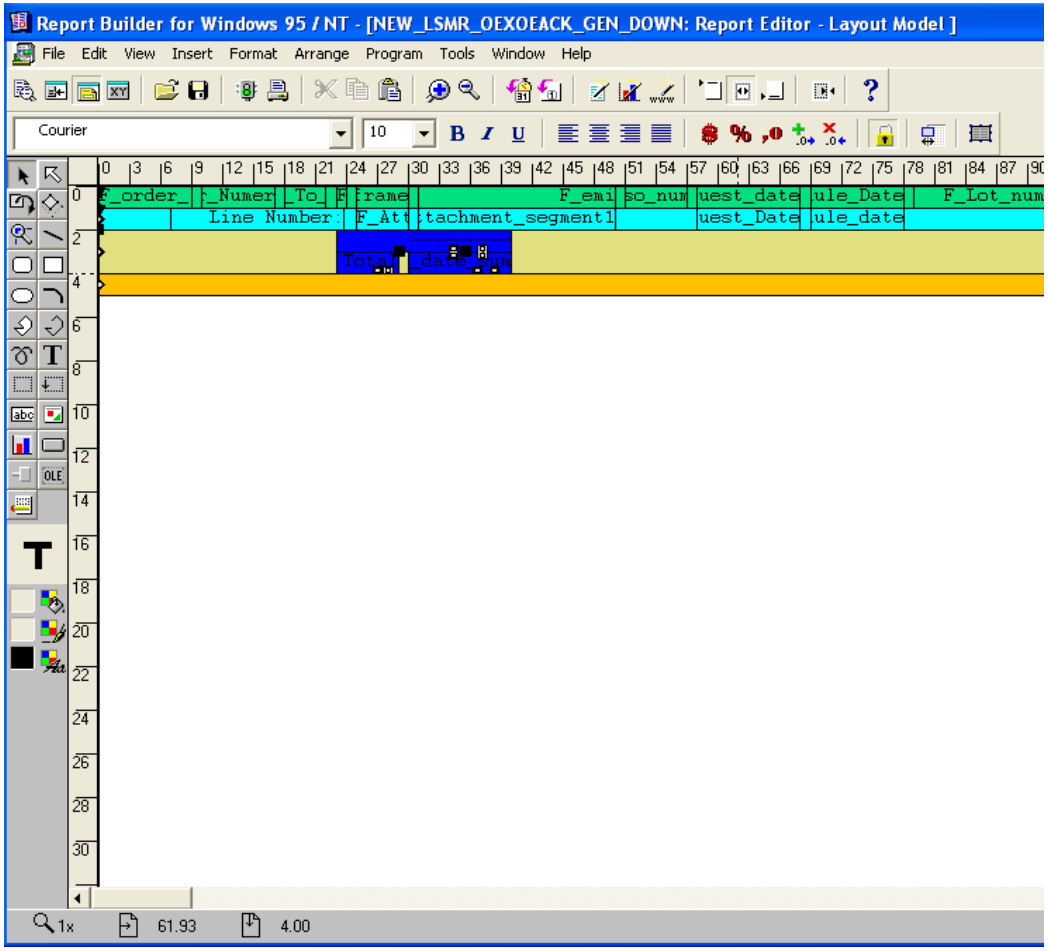
Data Model Overview

User Parameters



User Parameters are report specific and are used to pass information to a report at runtime. Parameters are used to "carve out" information as specified by the user.

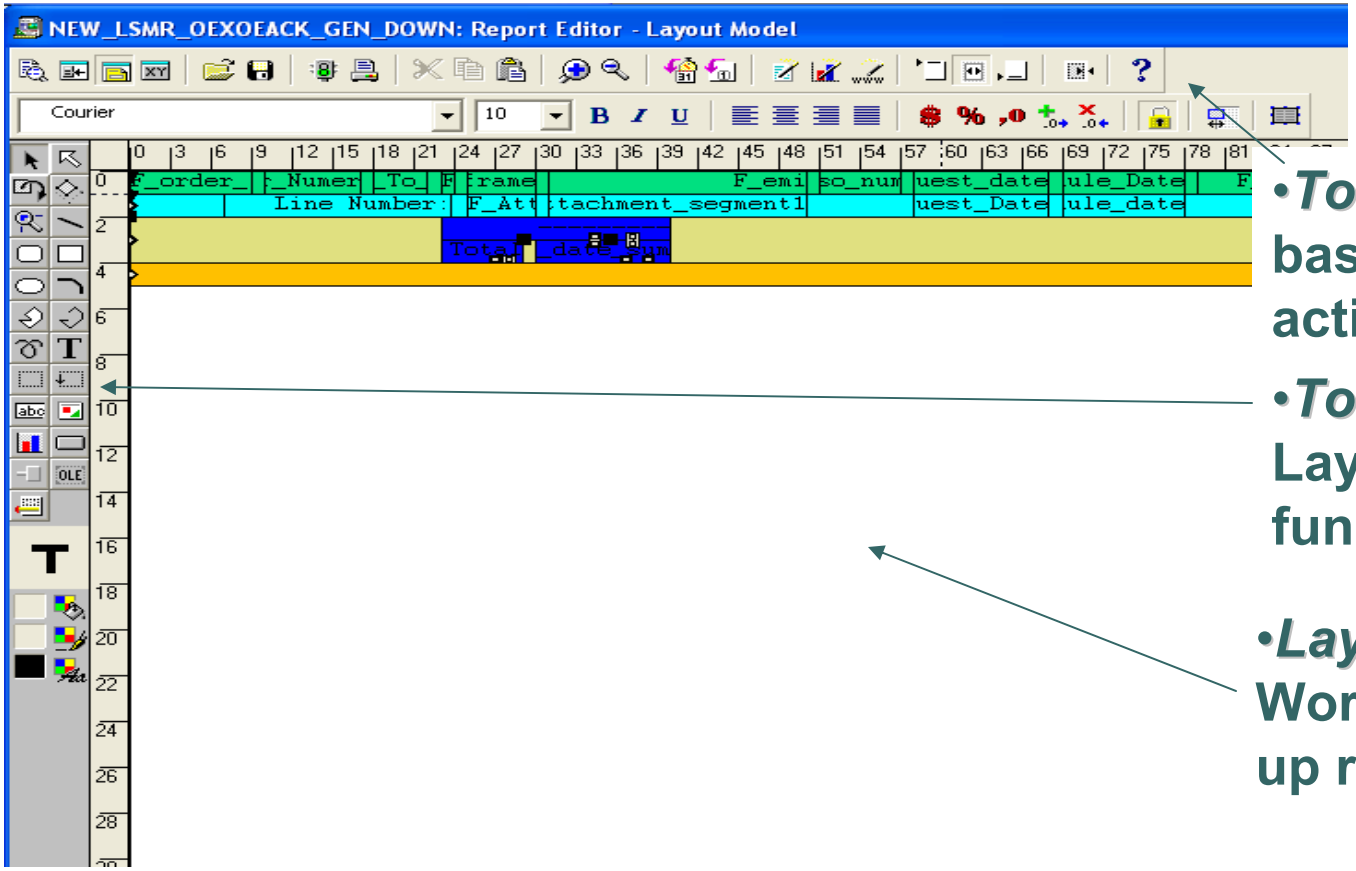
Layout Editor Overview



LAYOUT EDITOR OVERVIEW:

The Layout Editor is the workspace where you define the positions of the layout objects as they should appear in the report output.

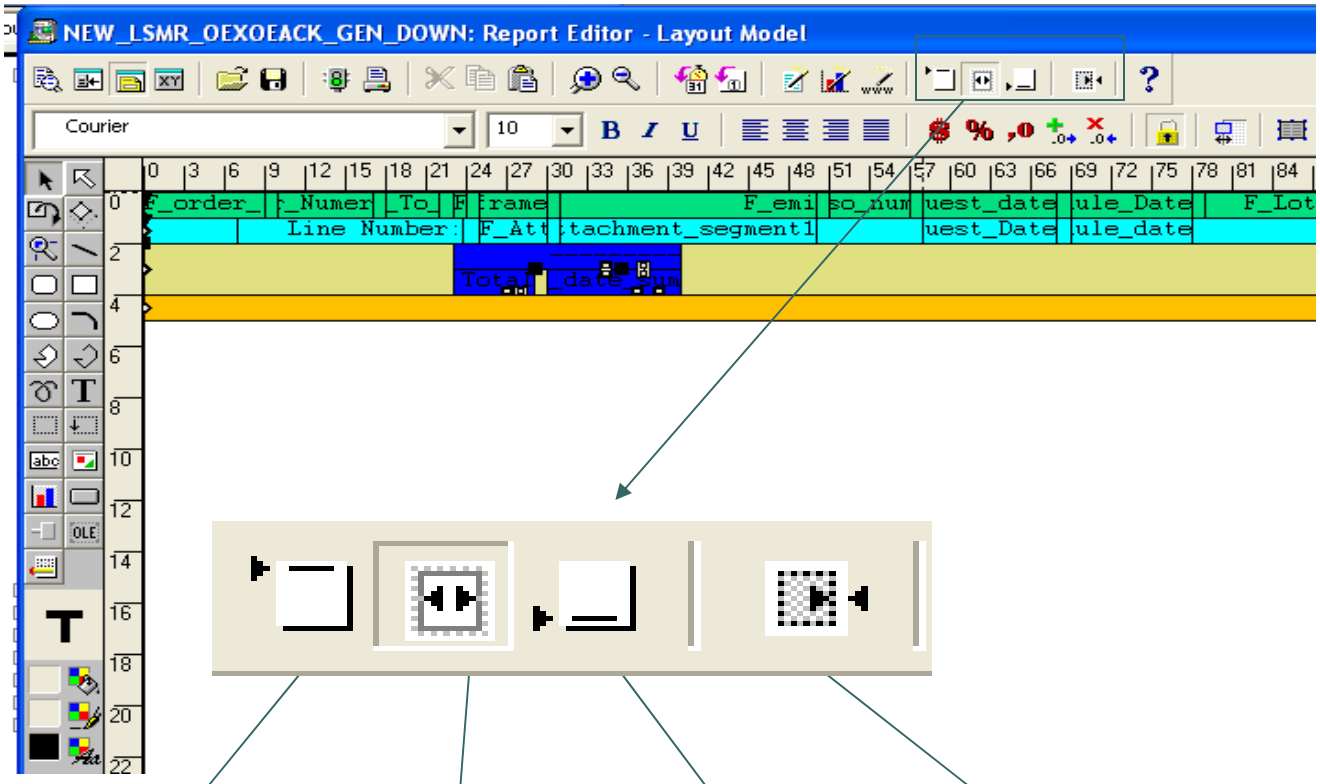
Layout Editor Overview



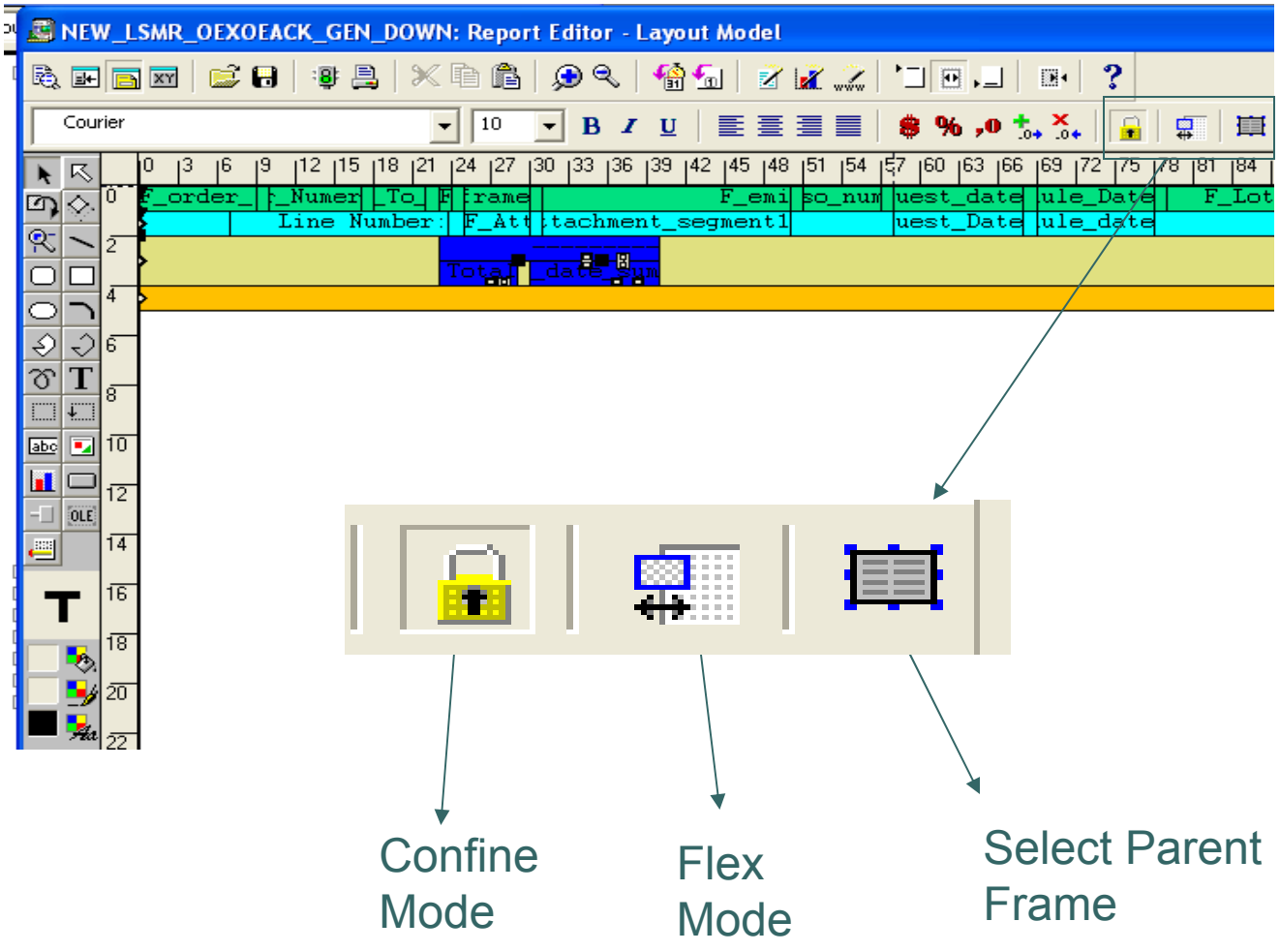
Layout Regions

- ***Tool Bar***: Perform basic report activities
- ***Tool Palette***: Perform Layout specific functions
- ***Layout Canvas***: Workspace for setting up report layout

Layout Editor Overview



Layout Editor Overview



Hint – Confine Mode

Use Confine Mode when moving objects to keep within groups and repeating Frames.

If confine mode is ‘Off’, you may accidentally move outside of the correct frame, and cause report to complete with error.

Confine Mode On/Off:

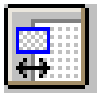
On

Off

When the confine mode is ‘On’, an object cannot be moved outside of or placed below its parent.

Flex Mode On/Off:

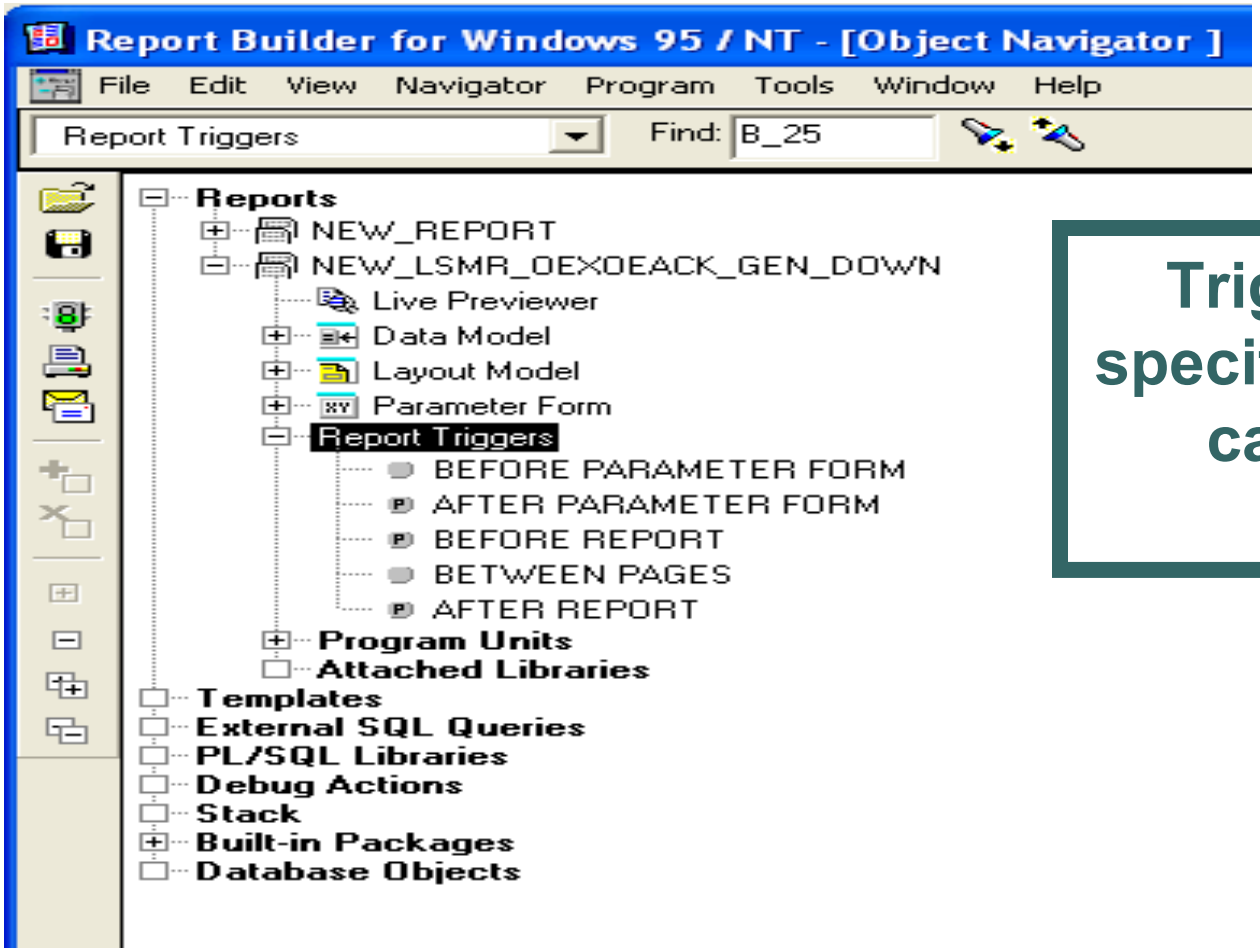
On



Off

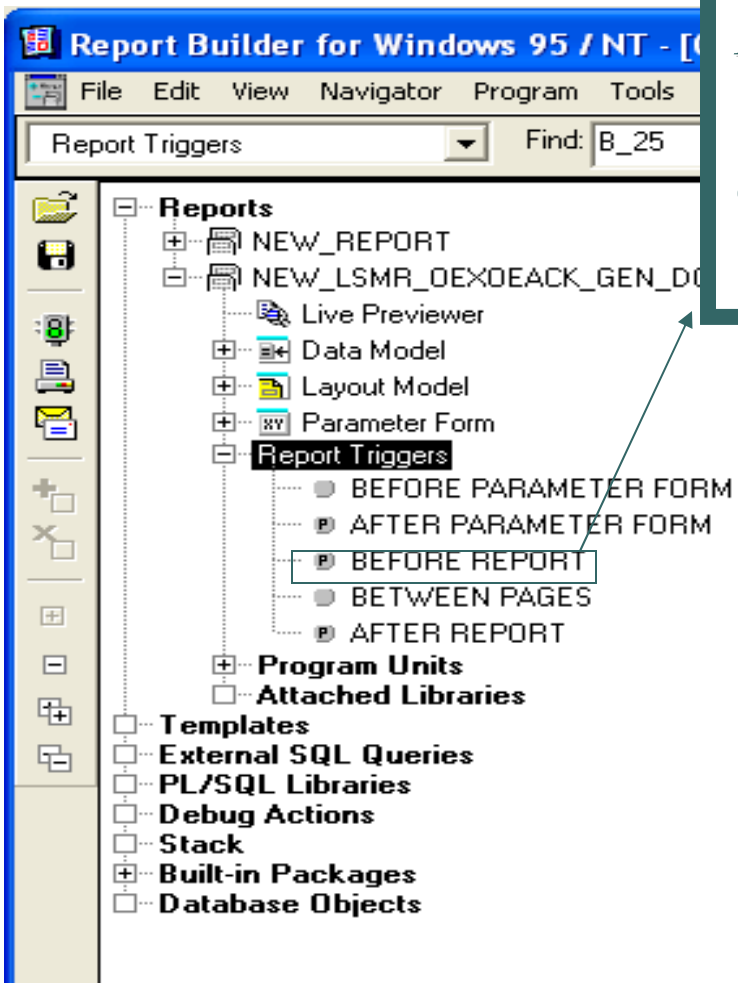
When the confine mode is ‘On’, the parent object is resized when a child object is resized.

Triggers Overview



Triggers fire at specific events and can execute PL/SQL

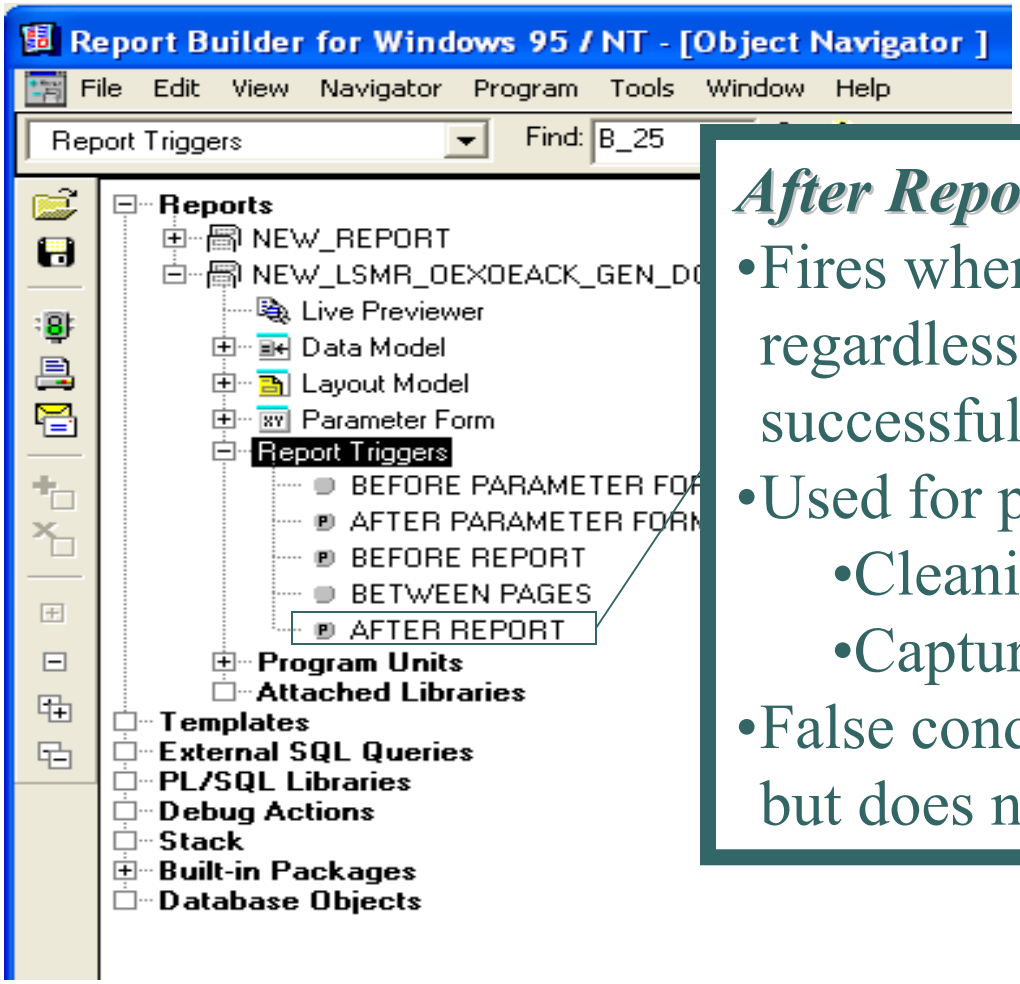
Triggers Overview



Before Report Trigger

- Fires after the Parameter form is executed, but before report is spooled
- False conditions stops processing

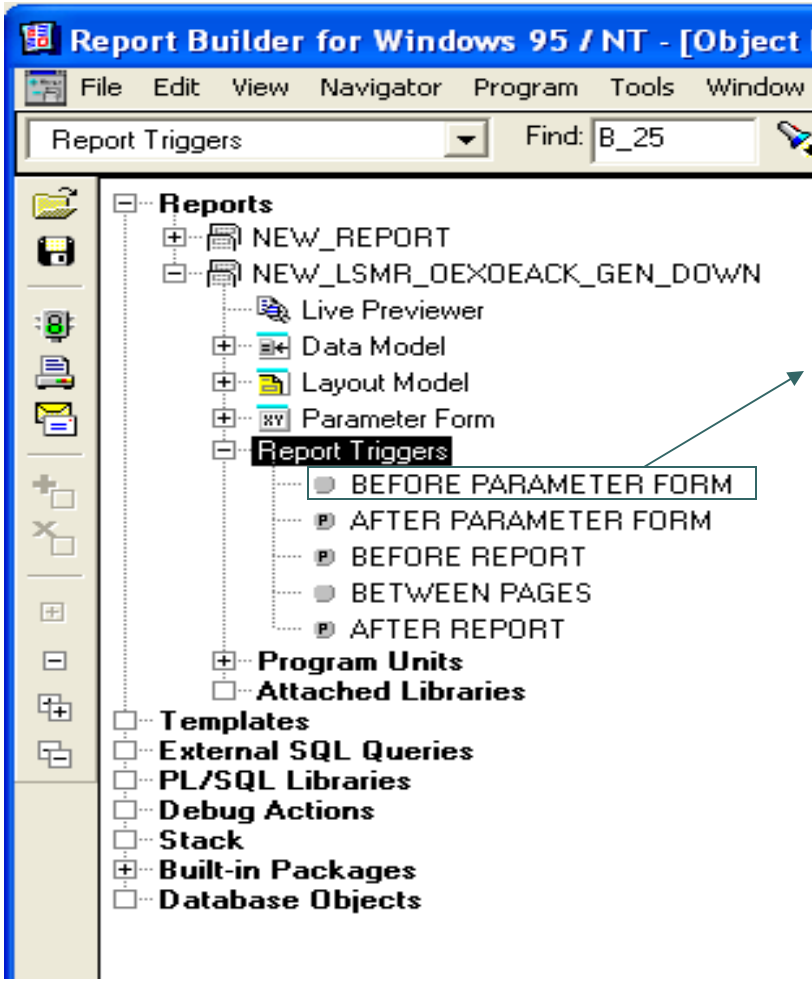
Triggers Overview



After Report Trigger

- Fires when report finishes spooling regardless if report completed successfully
- Used for post-report processing:
  - Cleaning up tables
  - Capturing statistics
- False conditions generates message but does not stop execution

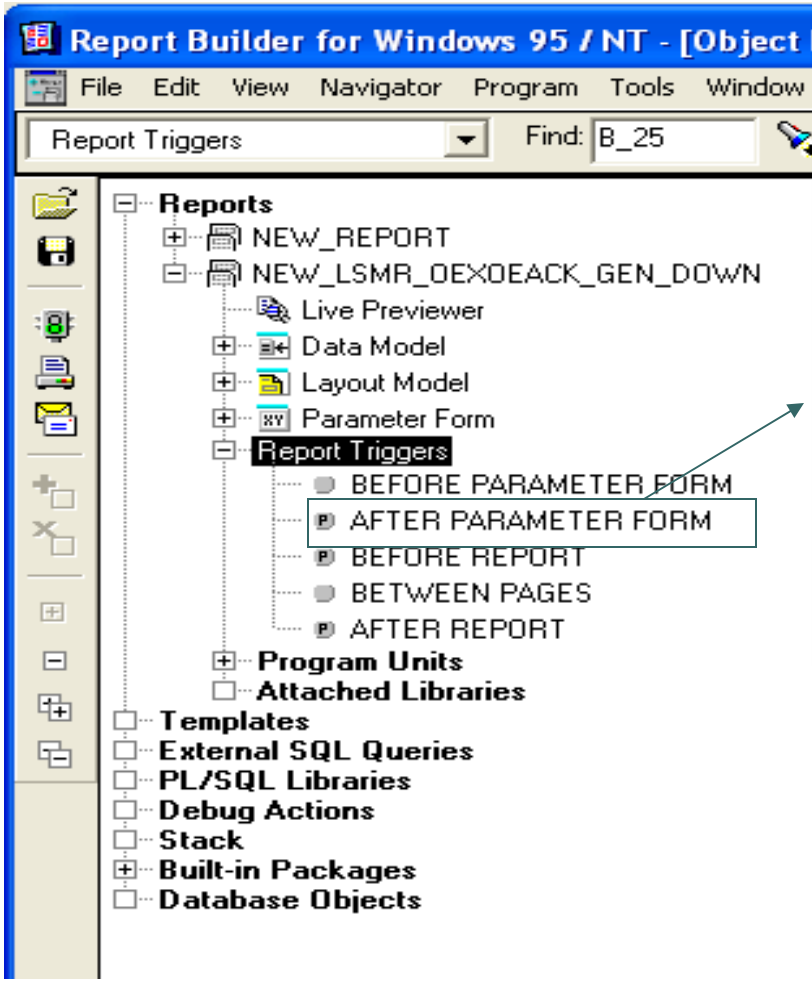
Triggers Overview



Before Parameter Form

- **Fires before parameter form is executed**
- **Used for preliminary processing**
  - Setting up tables
  - Formatting numbers or currency
- False condition stops processing

Triggers Overview

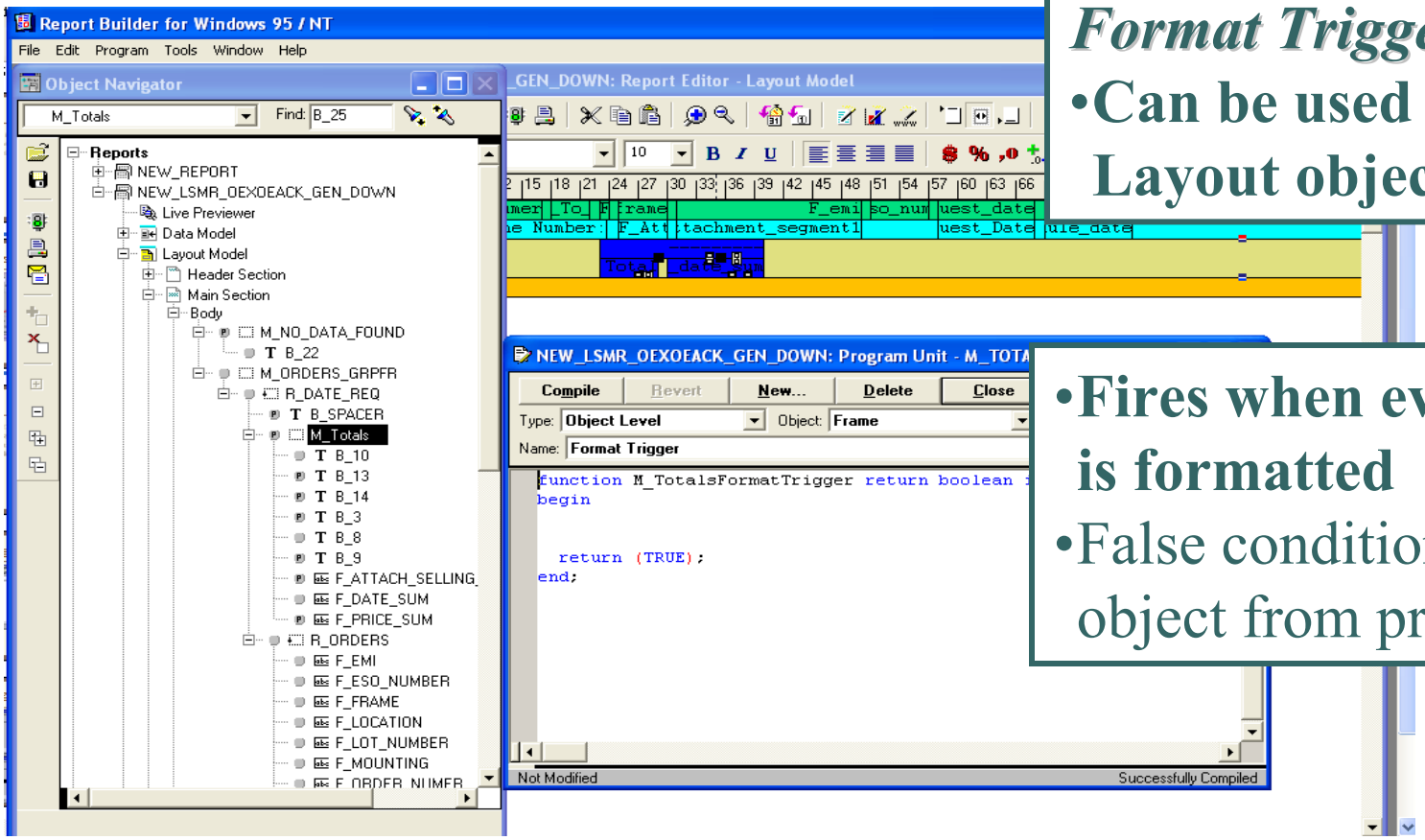


After Parameter Form

- **Fires after parameter form is executed**
- **Used to validate parameters**
- False condition sends you back to parameter for to re-enter parameters

Triggers Overview

Format Triggers



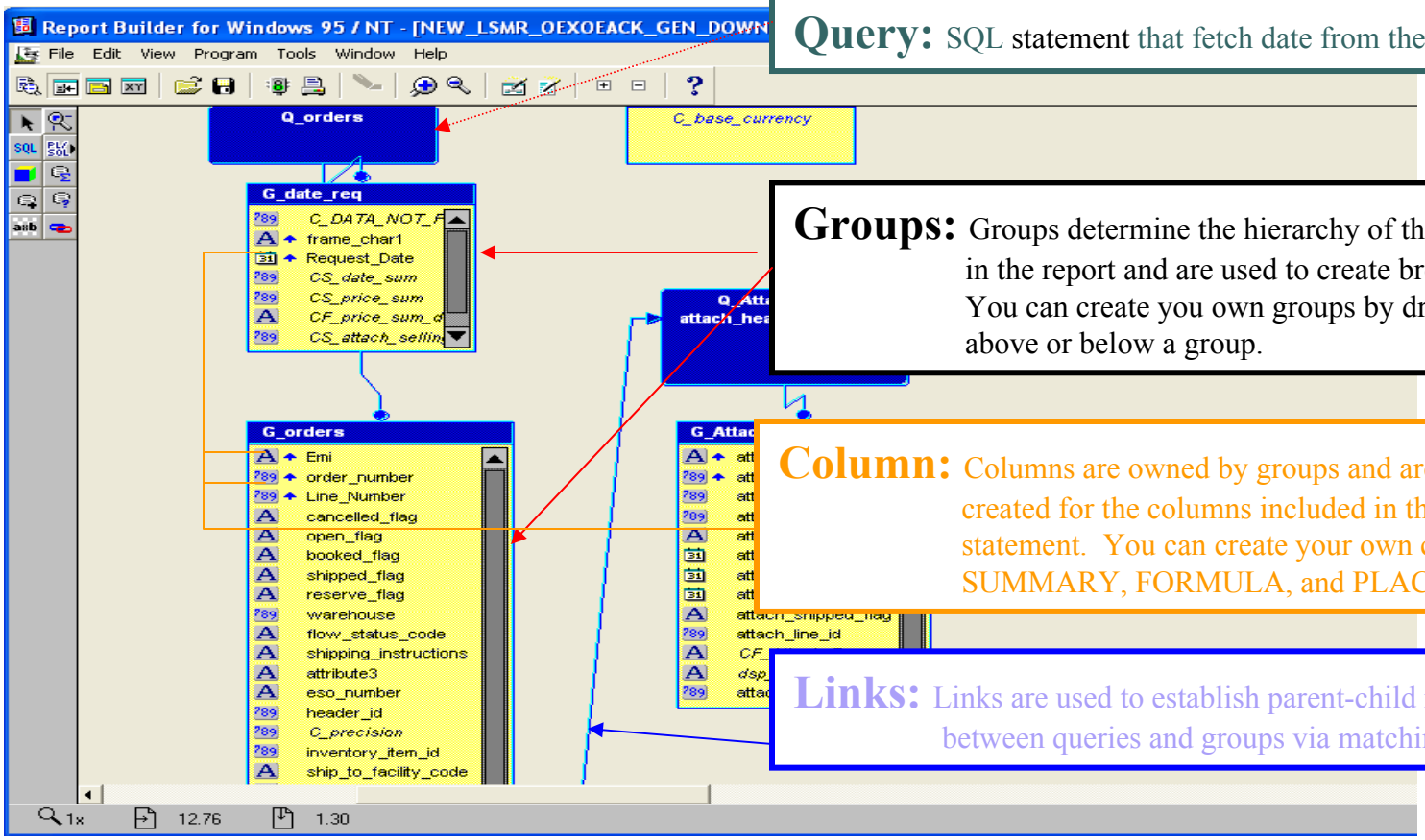
Format Trigger

- **Can be used in any Layout object**
- **Fires when every object is formatted**
- False condition prevents object from printing

Triggers Overview
Hint

- A FALSE condition for either a group or repeating frame will cause all enclosed objects such as fields and columns to inherit this condition and NOT print.

Working in the Data Model



Query: SQL statement that fetch date from the database

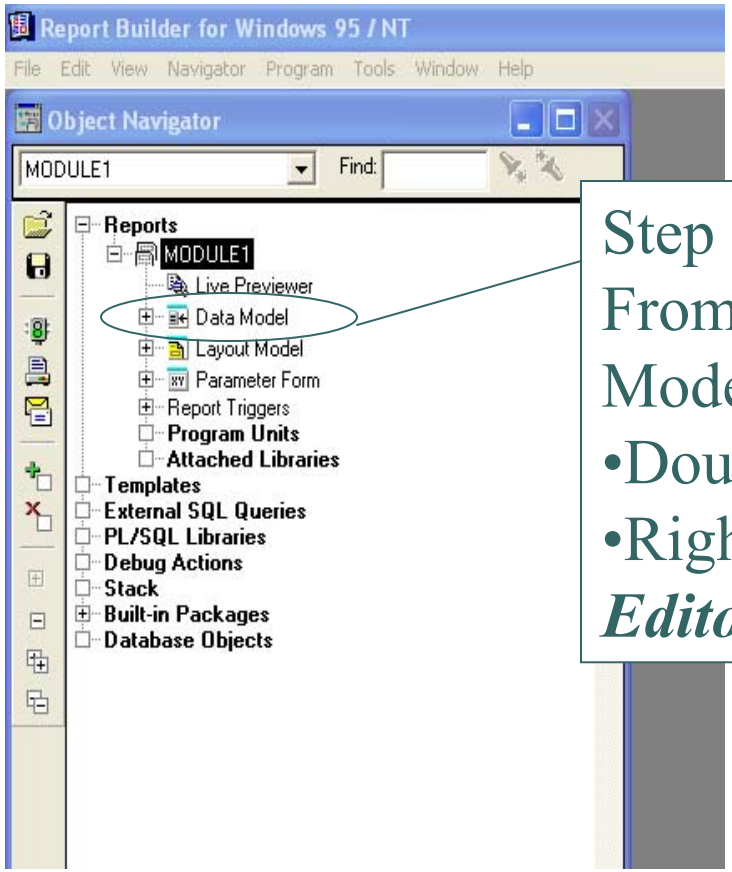
Groups: Groups determine the hierarchy of the data that appears in the report and are used to create breaks in the report. You can create you own groups by dragging a column above or below a group.

Column: Columns are owned by groups and are automatically created for the columns included in the select statement. You can create your own columns using the SUMMARY, FORMULA, and PLACE HOLER tools.

Links: Links are used to establish parent-child relationships between queries and groups via matching columns.

Parameters: Parameters allow you to change selection criteria (where clause) and calculations at run time.

Working in the Data Model
SQL

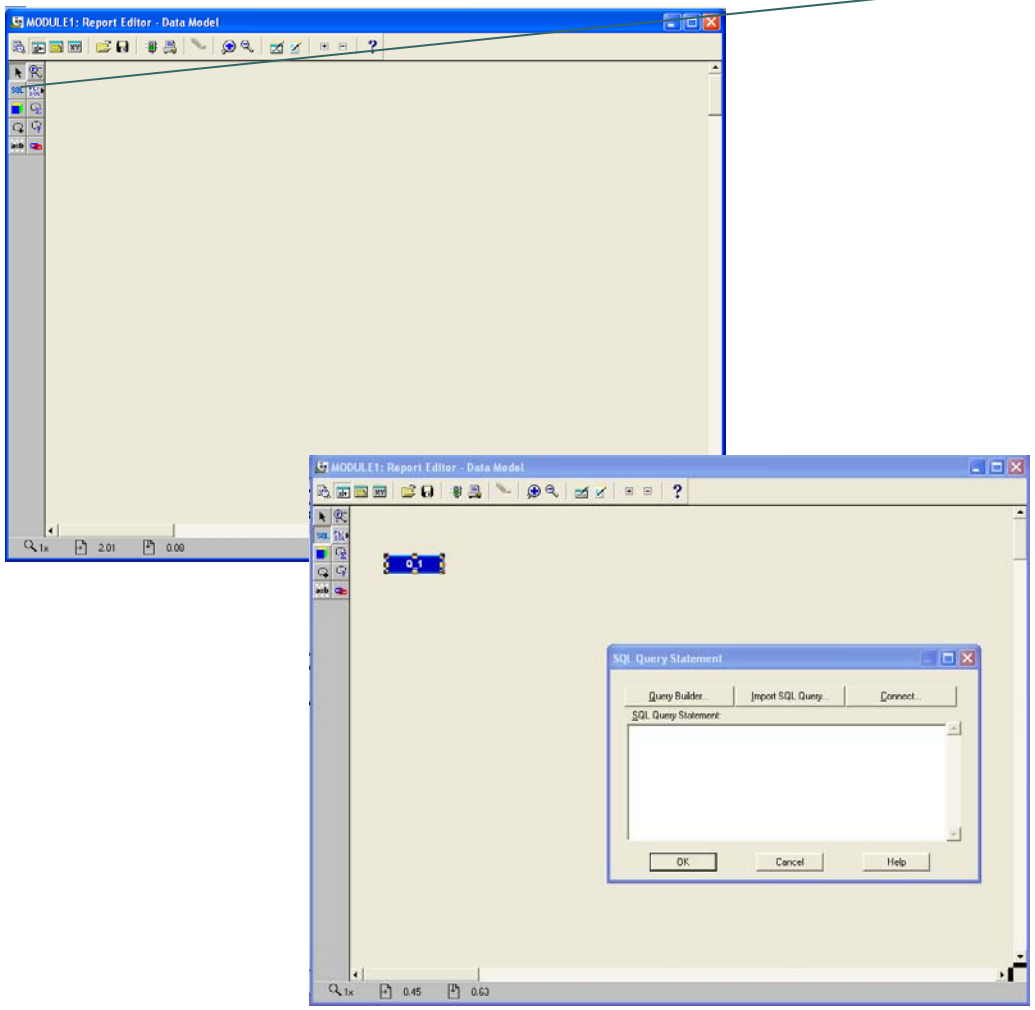


Building a SELECT Statement

Step 1:
From Object Navigator access Data Model

- Double click Data Model icon
- Right click Date Model and select ***Editor…***

Working in the Data Model
SQL



Click Sql Button.

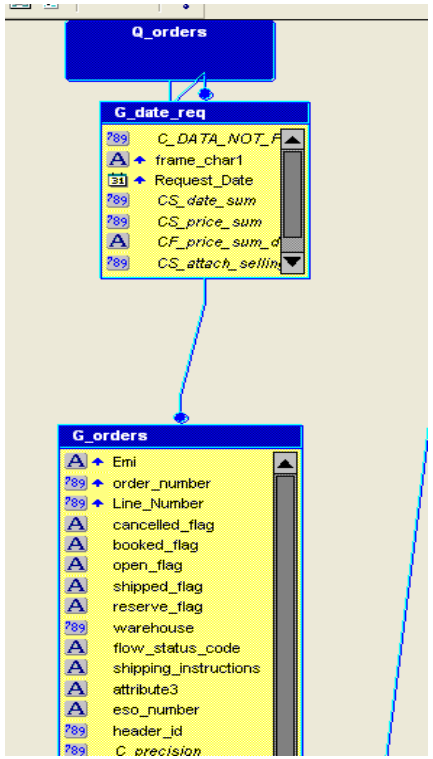
Place cursor anywhere in Data Model canvas and click.

This opens a SQL window, where you will enter your sql statement.

Note * You must be connected to the database to enter your sql, import sql, or use the Query Builder

Working in the Data Model

Groups



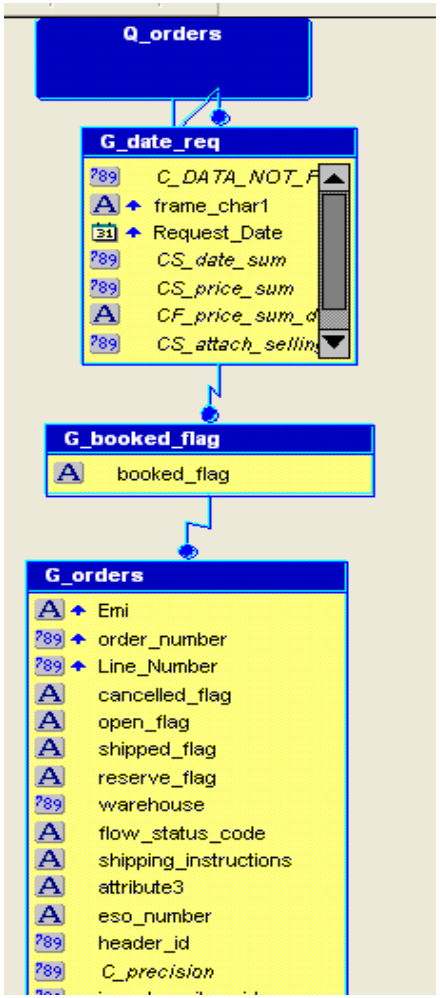
Use Break Groups

To create relationship in the data model you need to drag a column from G_ORDERS between G_ORDERS and G_date_req

Step 1: Point, click and hold down a column from G_ORDERS

Step 2: Drag booked_flag above G_ORDERS and below G_date_req

Working in the Data Model
Groups

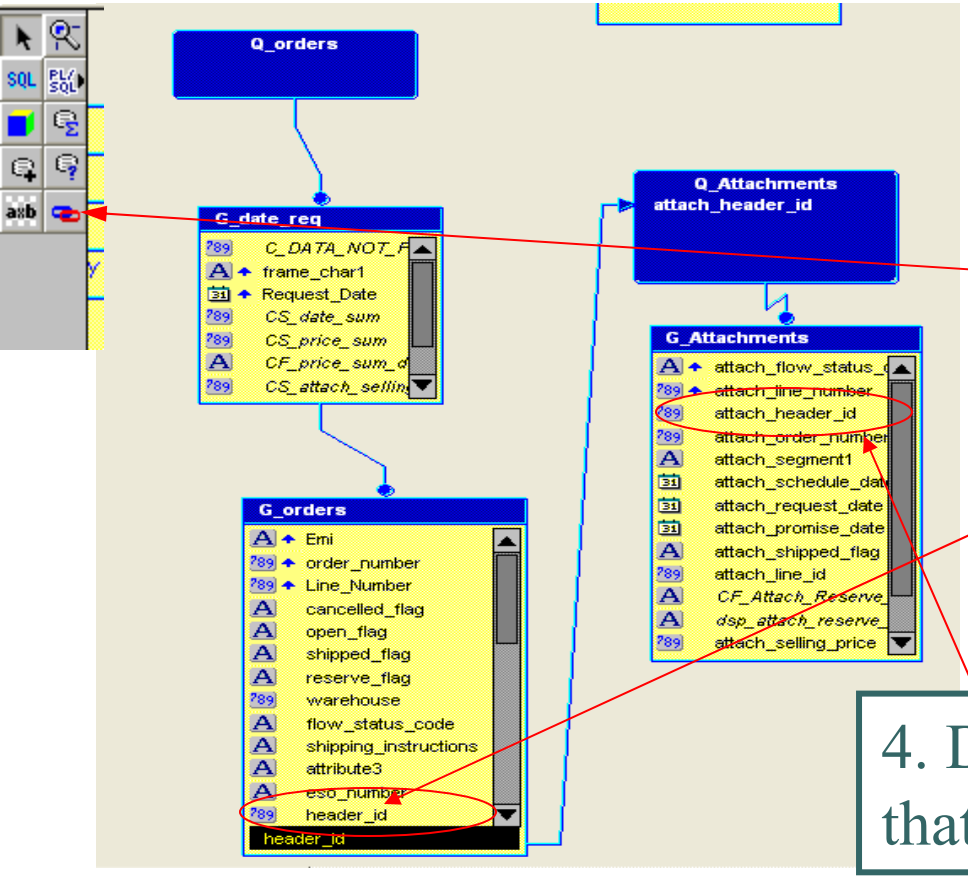


Use Break Groups

New break group is created with the name of 'G_<field_name>'

Working in the Data Model

Linking Queries



Use Linked Queries

1. Create second query

2. Point and click Link Icon

3. Point and click (and hold) on the column that defines the master relationship

4. DRAG the mouse to the column that defines the detail relationship

5. Release the mouse

Working in the Data Model

Data Groups vs. Linked Queries

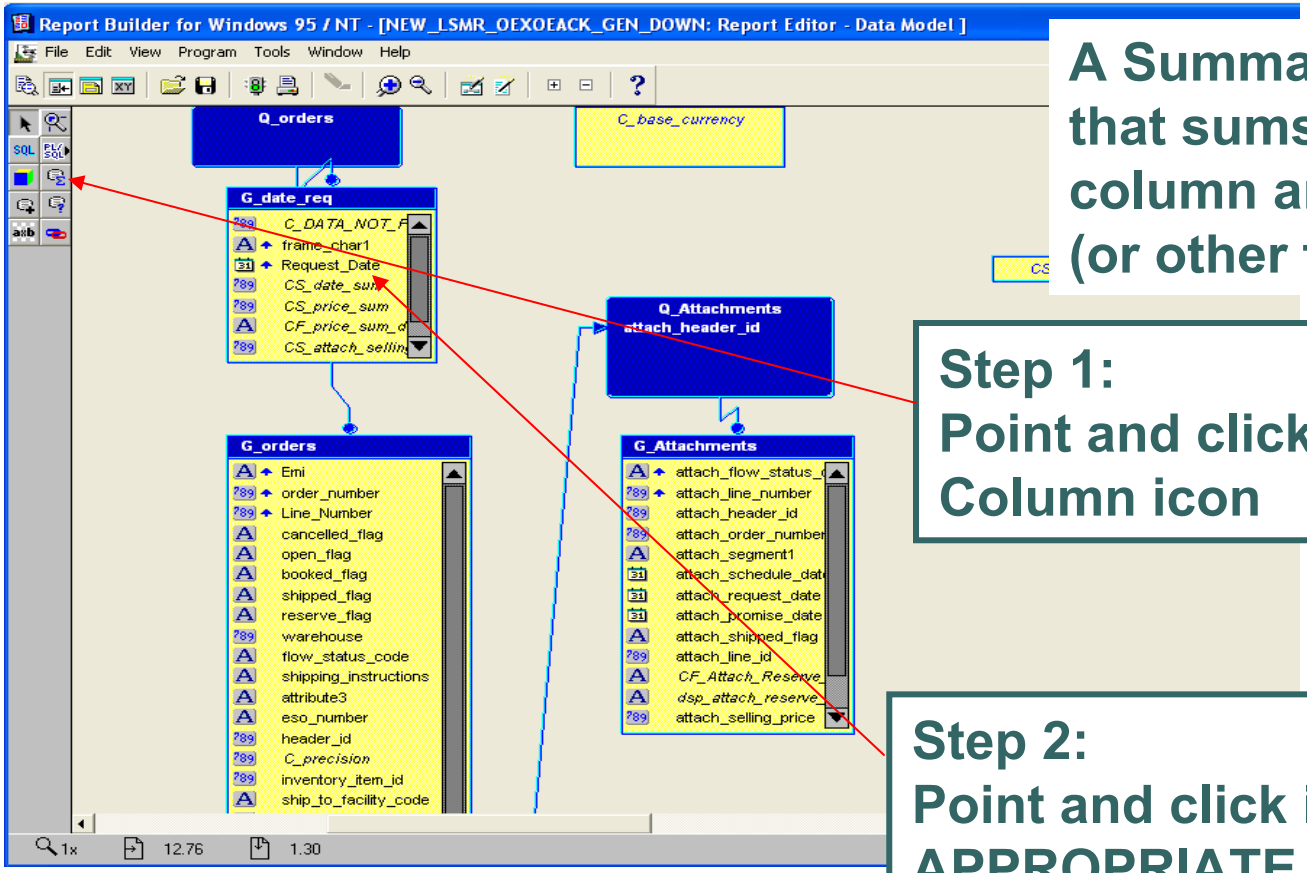
- Essentially, the groups and linked queries create the same relationship between data.

If SQL is already written, and performance is good, then one query should be fine. If performance seems to be a problem, may try and break up into smaller, more efficient queries and use a data link between queries. Remember, you need similar columns from both queries that will produce the link. It is helpful to give at least the second query a column alias on the field that will be used for the link.

If report is being modified and need to add more groups of data, may be easier to add additional query and use data link between queries.

Working in the Data Model

Summary Column



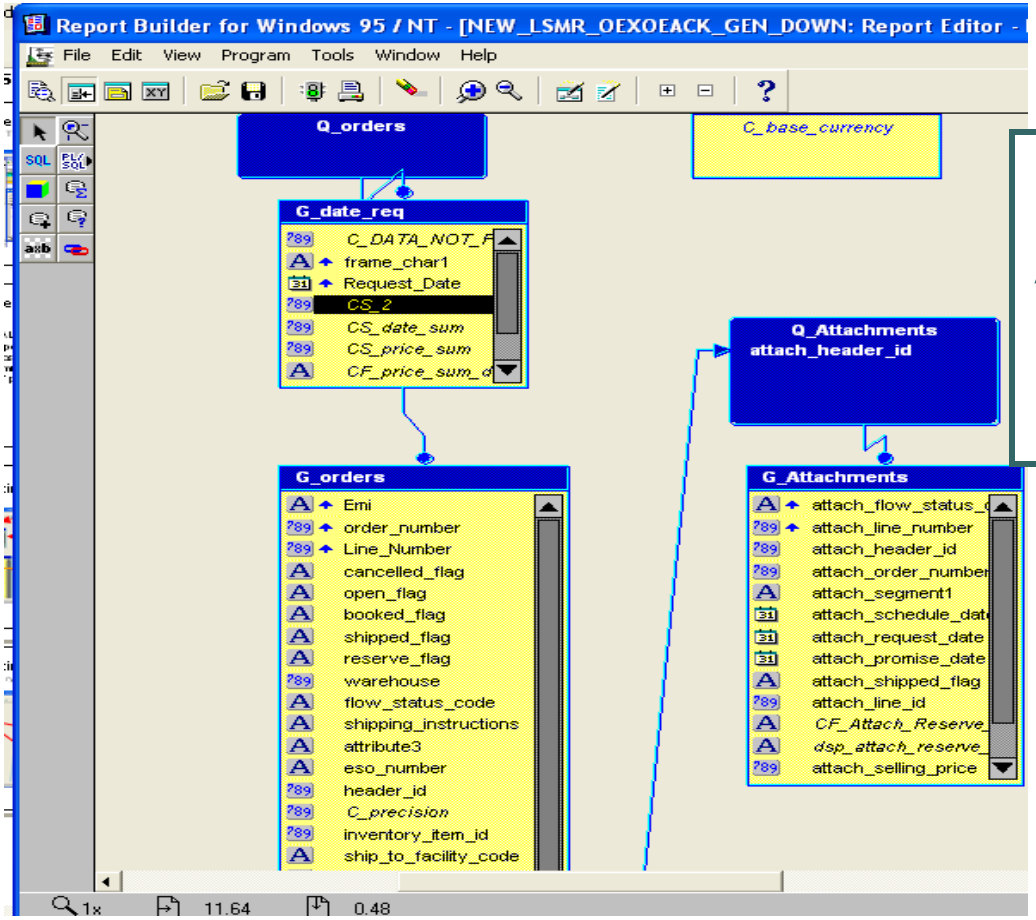
A Summary Column is a field that sums items in another column and generates a total (or other function)

Step 1:
Point and click on the Summary Column icon

Step 2:
Point and click in the APPROPRIATE area of the Data Model

Working in the Data Model

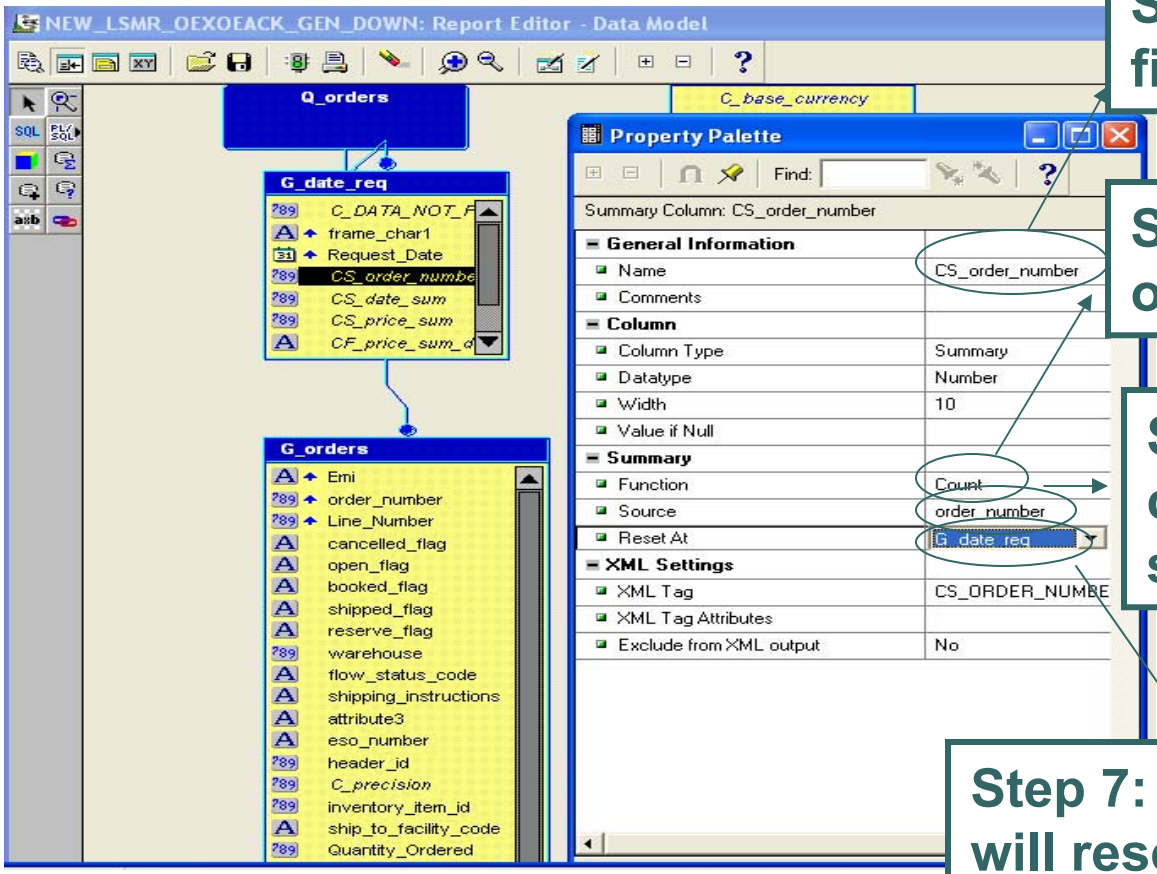
Summary Column



Step 3:
Access the Summary Column property palette

Working in the Data Model

Summary Column



Step 4: Rename the field, BUT KEEP 'CS_'

Step 5: Choose what type of summary function

Step 6: Identify a Source column to be summarized

Step 7: Select when the total will reset

Working in the Data Model

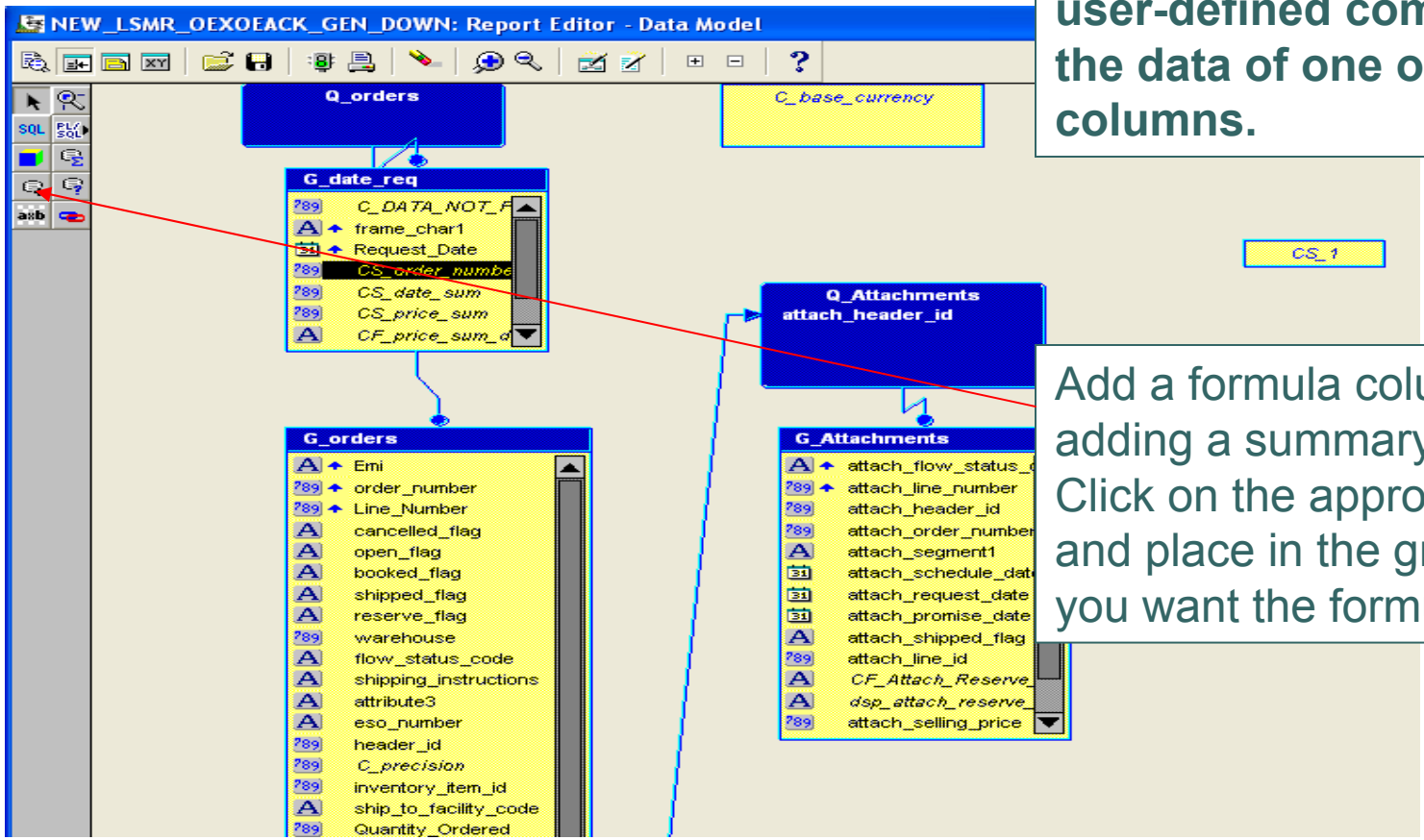
Summary Column - Tip

- Place your Summary Column within the group you want to total. The source does NOT have to be in that group

Working in the Data Model

Formula Column

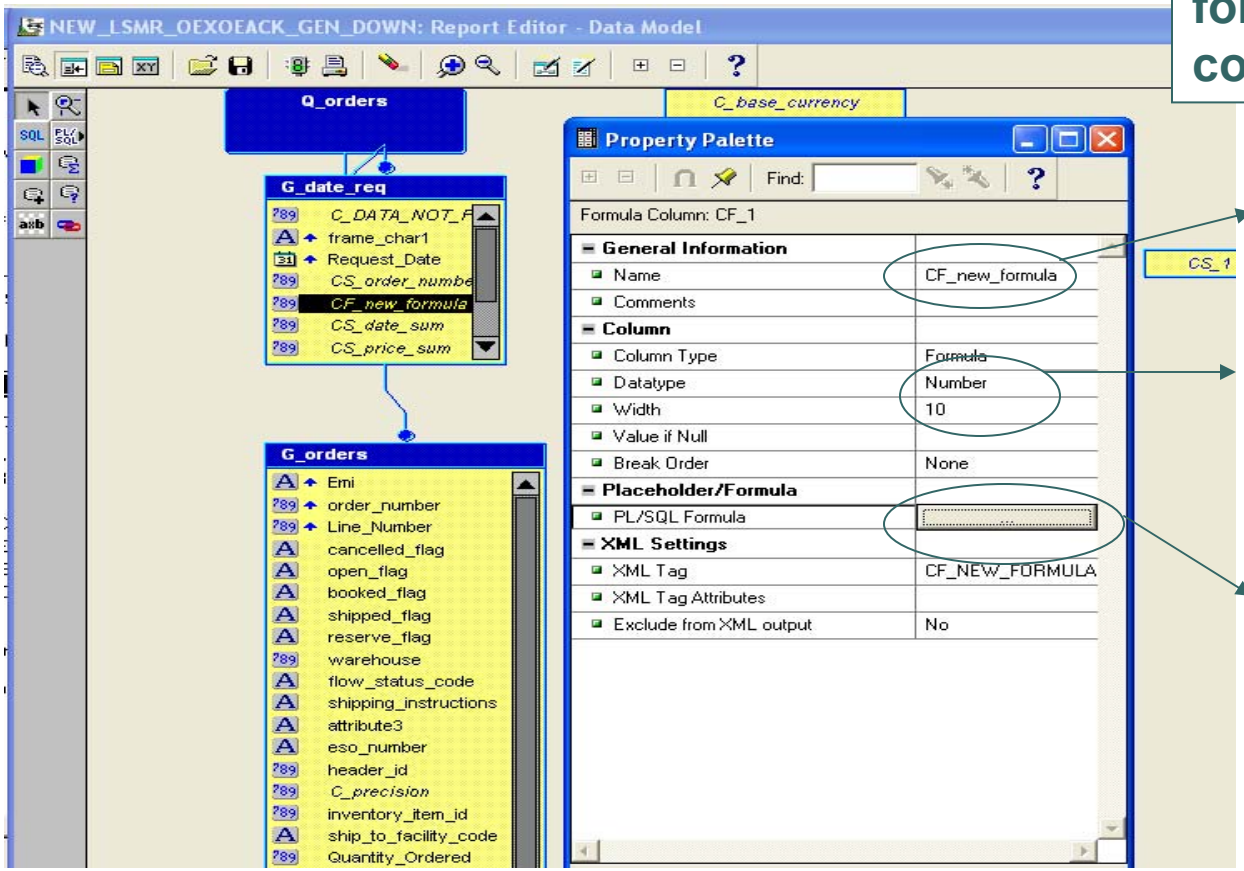
A Formula column performs a user-defined computation on the data of one or more columns.



Add a formula column similar to adding a summary column. Click on the appropriate icon, and place in the group where you want the formula column.

Working in the Data Model

Formula Column



Open Property Palette for new formula column

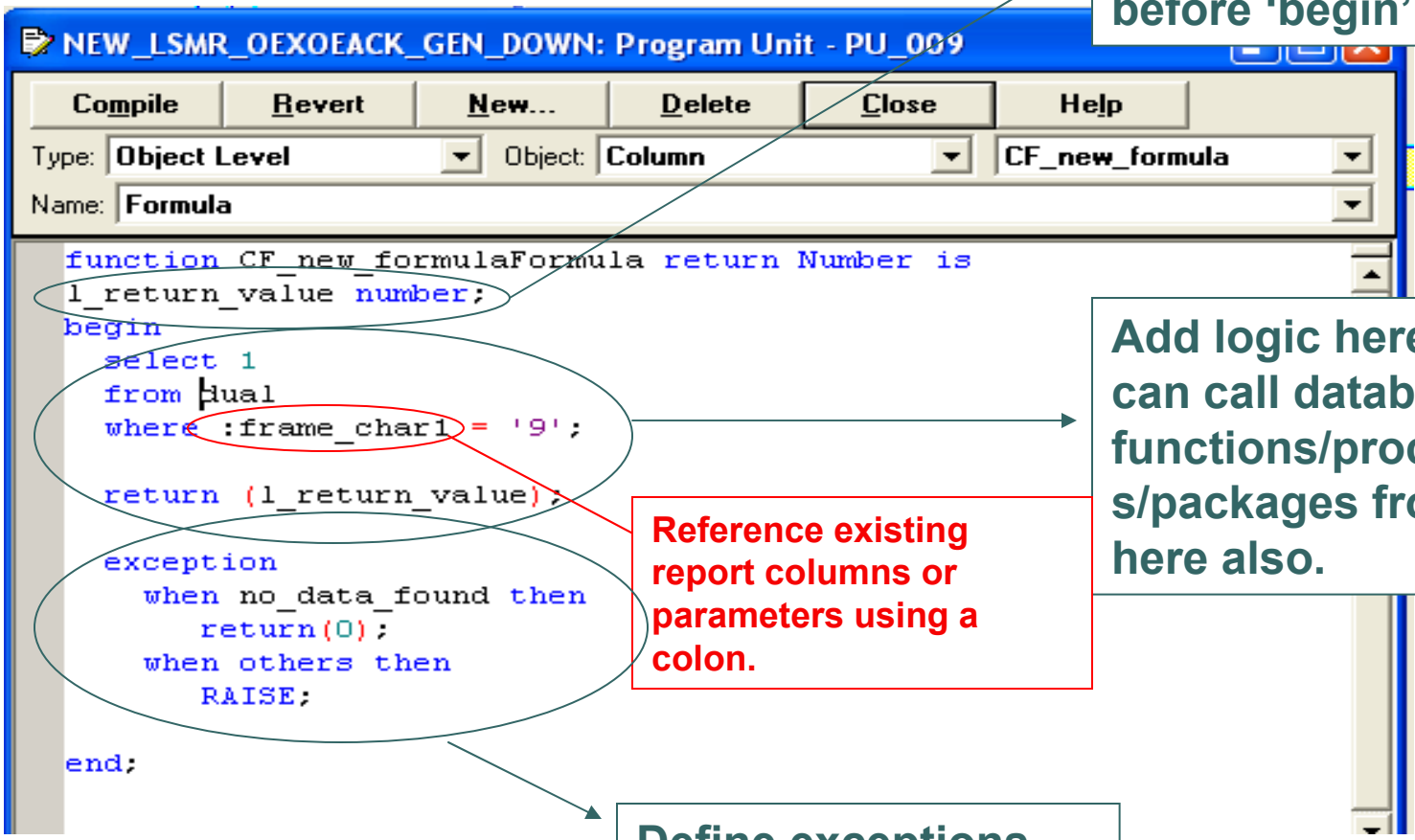
Rename the field, BUT KEEP 'CF_'

Choose Datatype and width

Click on PL/SQL Formula to add code to the formula column

Working in the Data Model

Formula Column



Define any variables before 'begin'

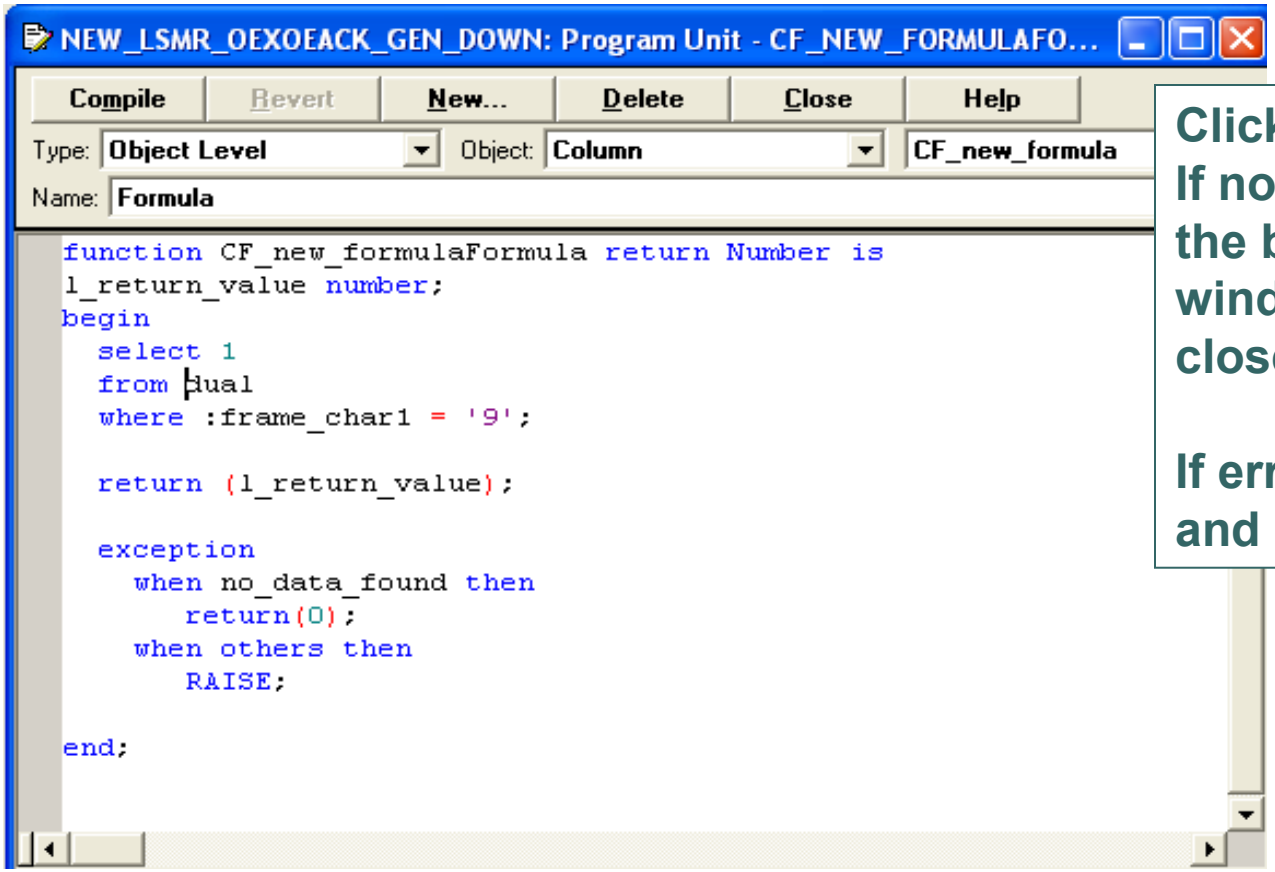
Add logic here. You can call database functions/procedures/packages from here also.

Reference existing report columns or parameters using a colon.

Define exceptions for function.

Working in the Data Model

Formula Columns



Click Compile.
If no errors show at the bottom of the window, you can close the window.

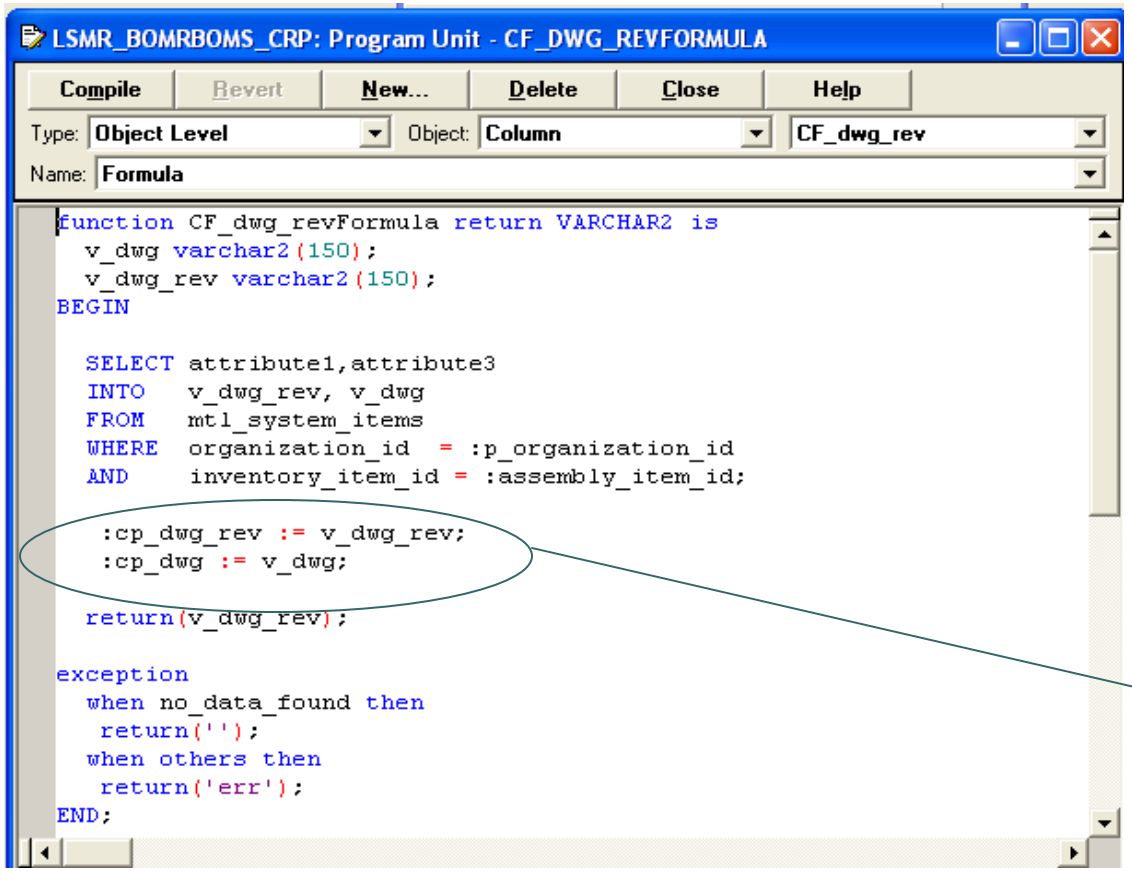
If errors – fix errors and compile again.

Working in the Data Model
Formula Columns

- Use Formula Columns when the data you need cannot be easily added to the query.
- Add formula columns to the same groups that the columns reside in you will be referencing in the formula column.
- Formula columns can be used in conjunction with placeholder columns.

Working in the Data Model

Formula Columns with Placeholder Columns



Create Placeholder columns in the same way as the formula and summary columns. Name will be 'CP_' (name)

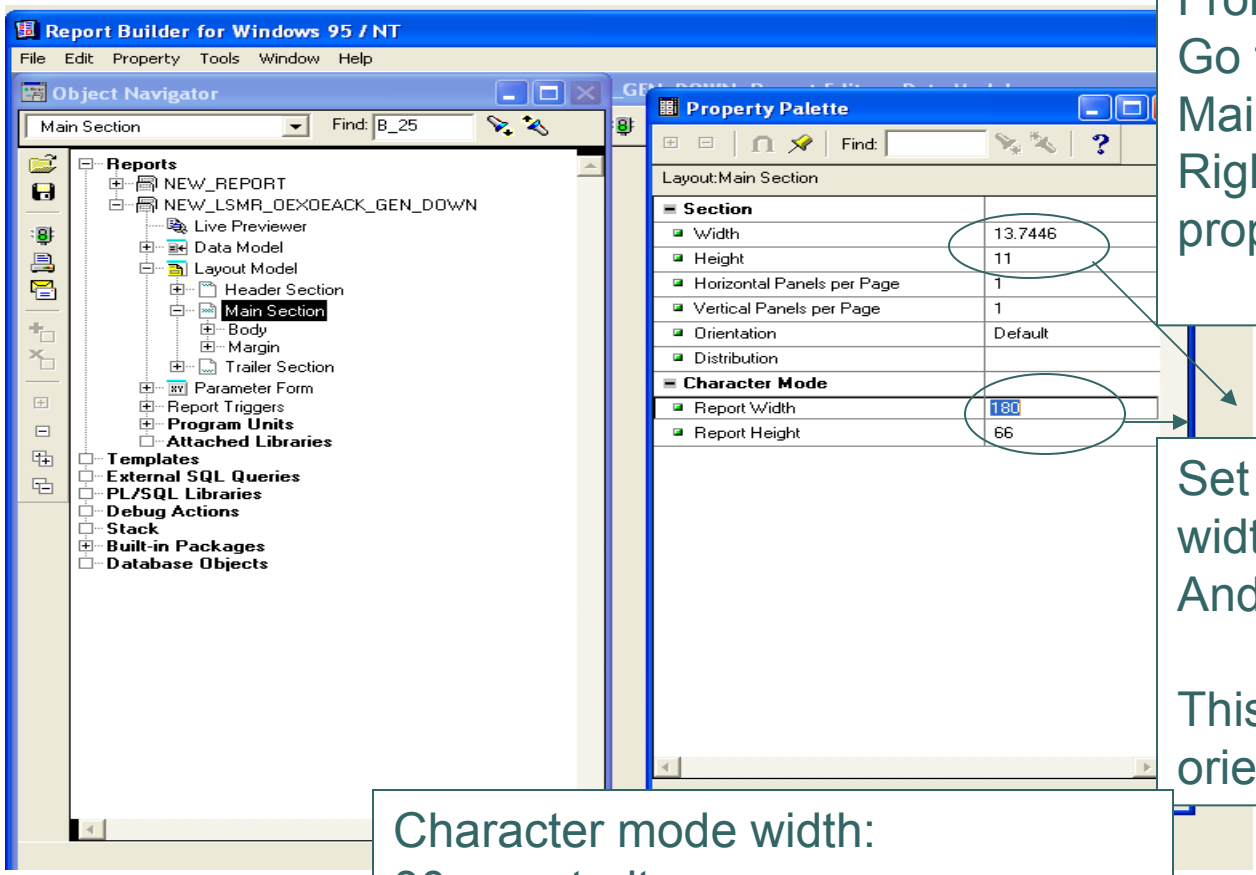
Create Formula column, and write code to populate placeholder columns. Reference the placeholder columns as you would any other report columns.

Working in the Layout Editor

- When building a new report, you can use the Report Wizard to create a new layout. ** Make sure Character Mode is set to Yes.
- If modifying existing report that only needs a few changes to layout, do not use the Report Wizard.

Working in Layout Editor

Property Palette (set page size)



From the object Navigator:
Go to Layout Model >
Main Section.
Right click and bring up property palette.

Set page size in inches for width and height,
And also for Character Mode.

This will also determine orientation.

Character mode width:
80 = portrait
132 = landscape
180 = landwide

Working in the Layout Editor

Set Rulers and Grid Snap

Check to make sure Grid and Snap to Grid are enabled:

View > Grid

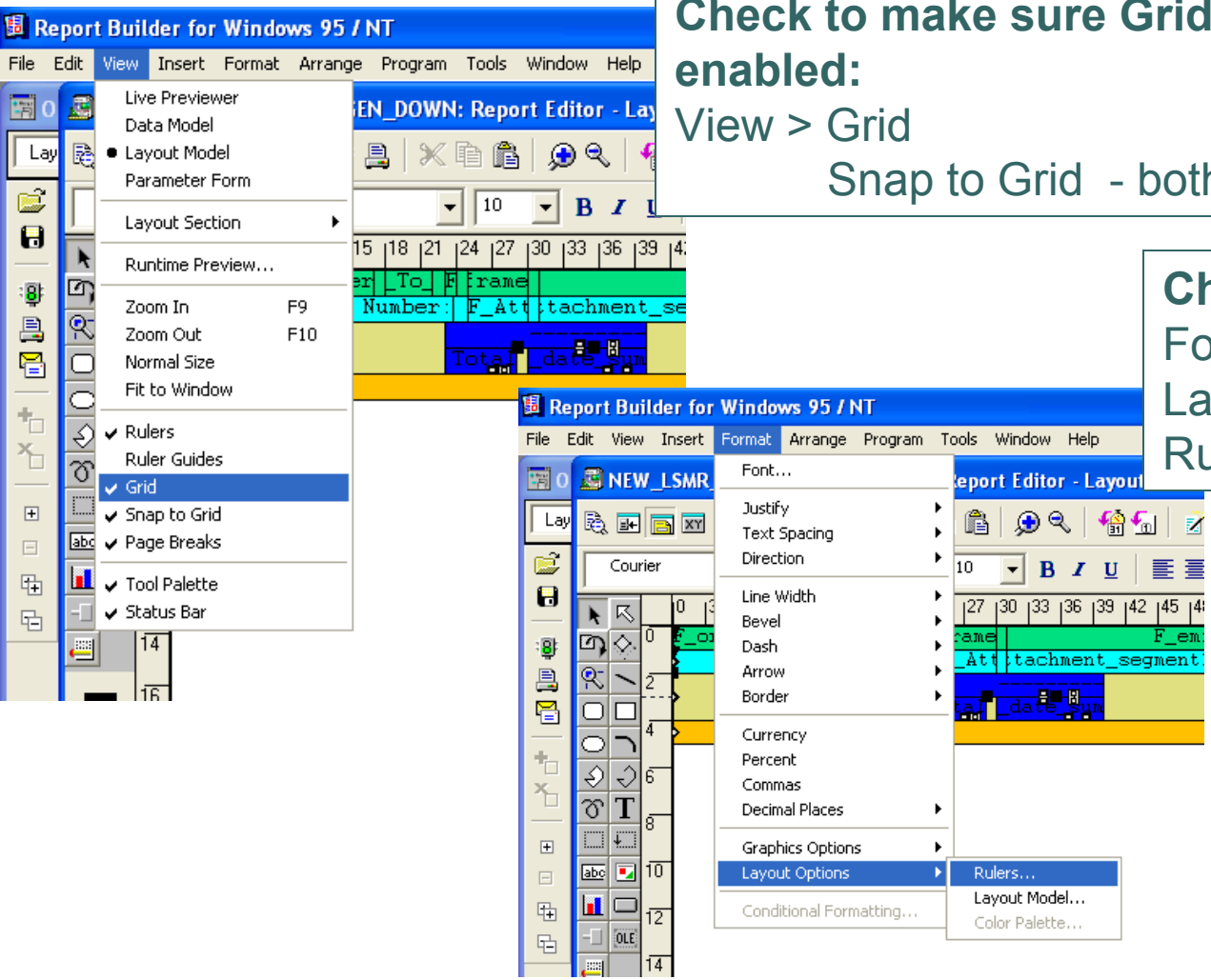
Snap to Grid - both should be checked.

Check Grid Space Options:

Format >

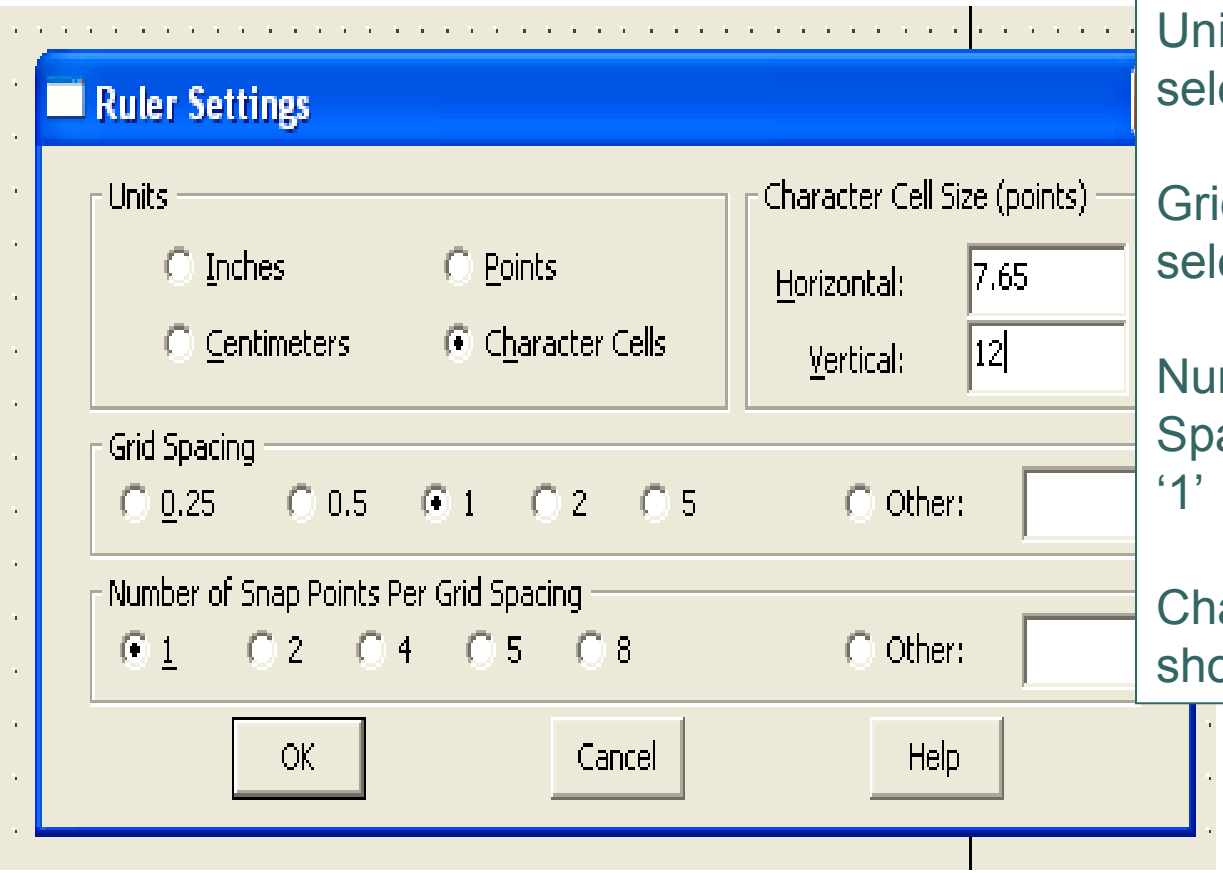
Layout Options >

Rulers



Working in the Layout Editor

Set Rulers and Grid Snap



Units: ‘Character Cells’ should be selected.

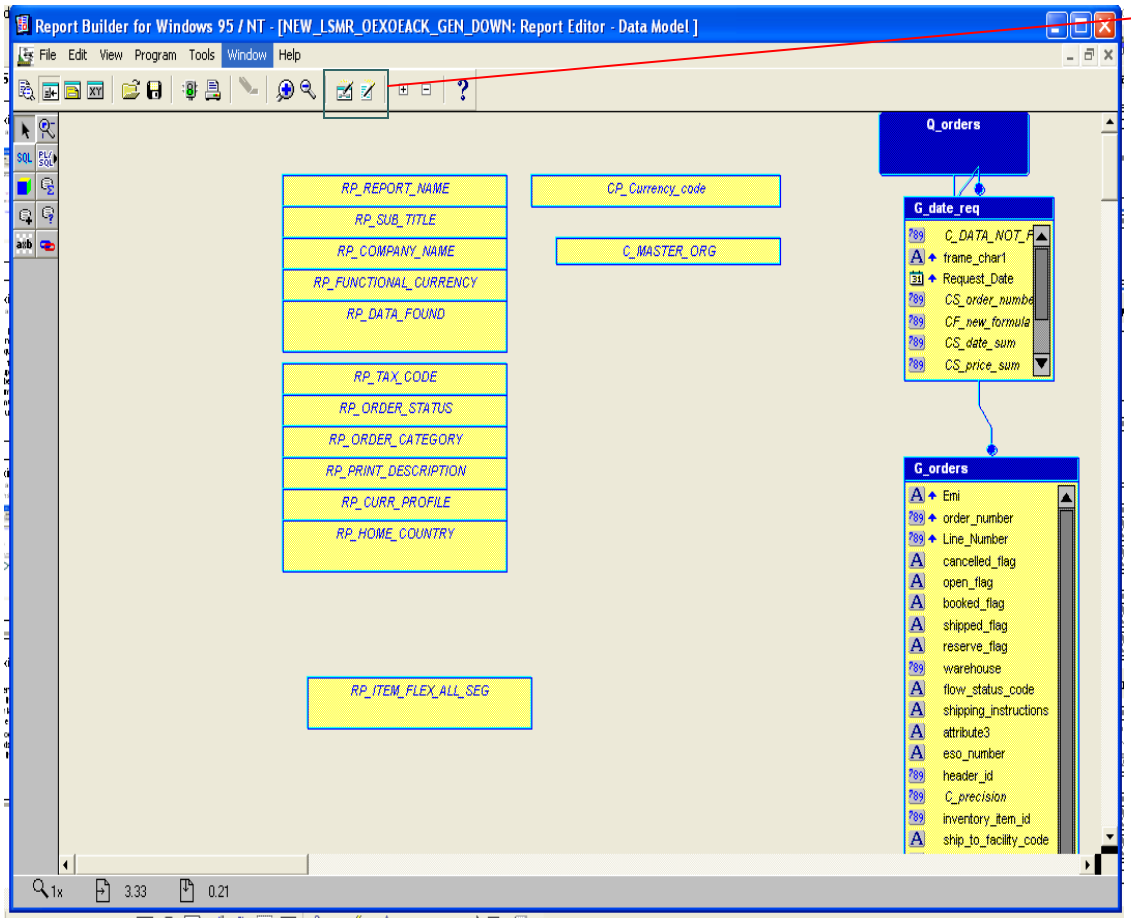
Grid Spacing: ‘1’ Should be selected.

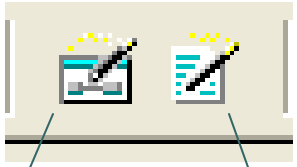
Number of Snap Points Per Grid Spacing:
‘1’

Character Cell Size(points) should default correctly.

Working in the Layout Editor

Report Wizard (from data model)





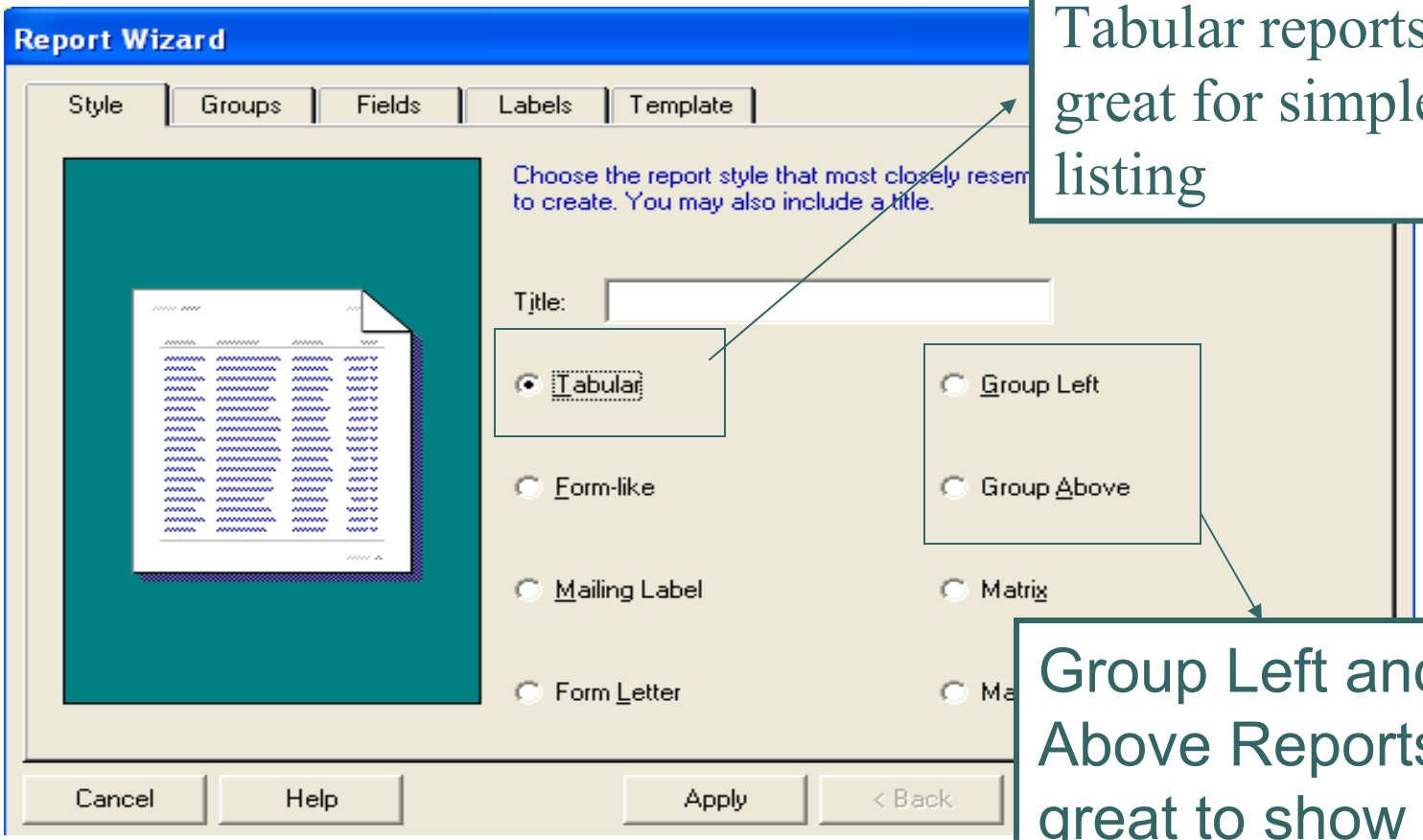
Data Wizard

Report Wizard

Click Report Wizard

Working in the Layout Editor

Report Wizard



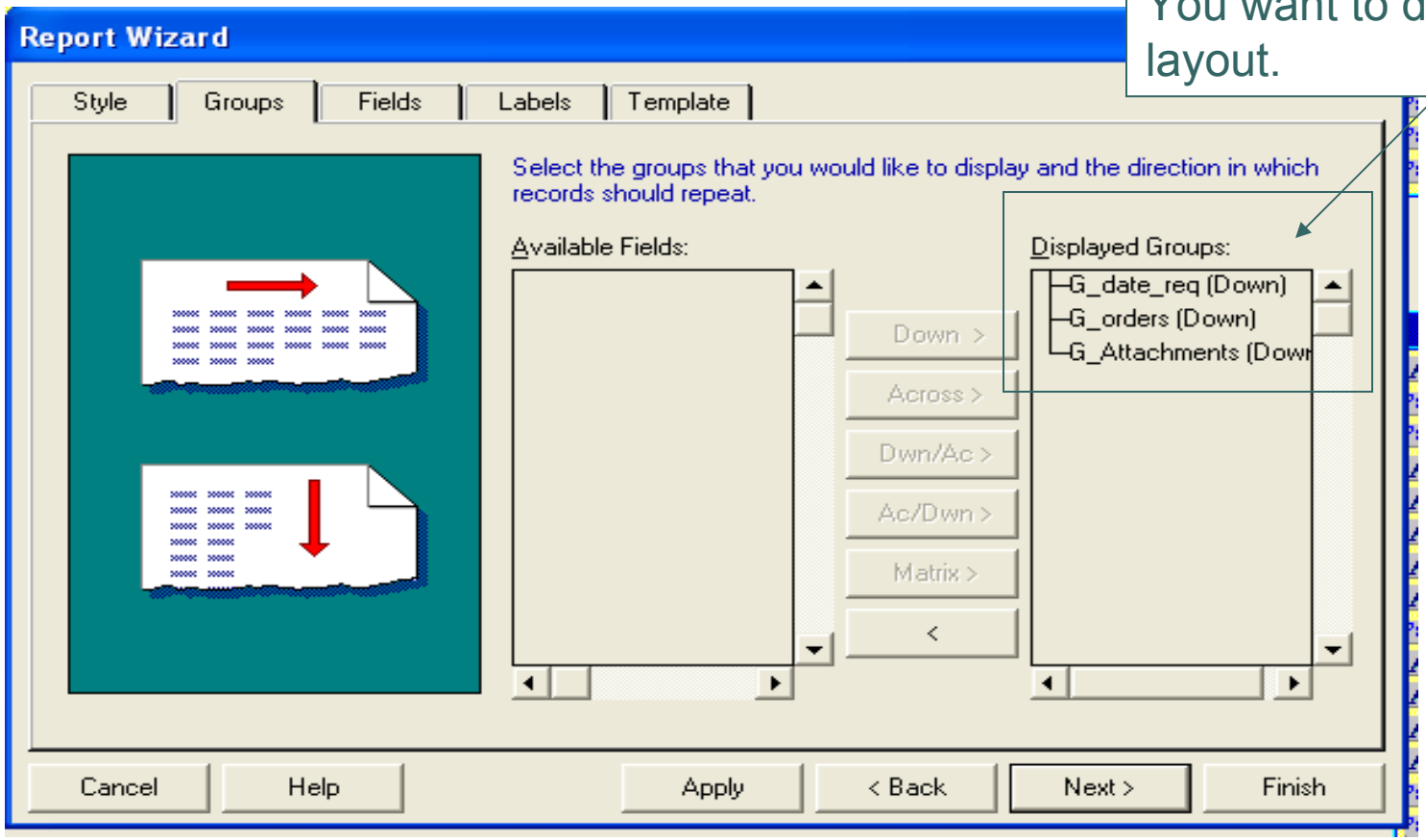
Tabular reports are great for simple listing

Group Left and Group Above Reports are great to show parent-child relationships

Working in the Layout Editor

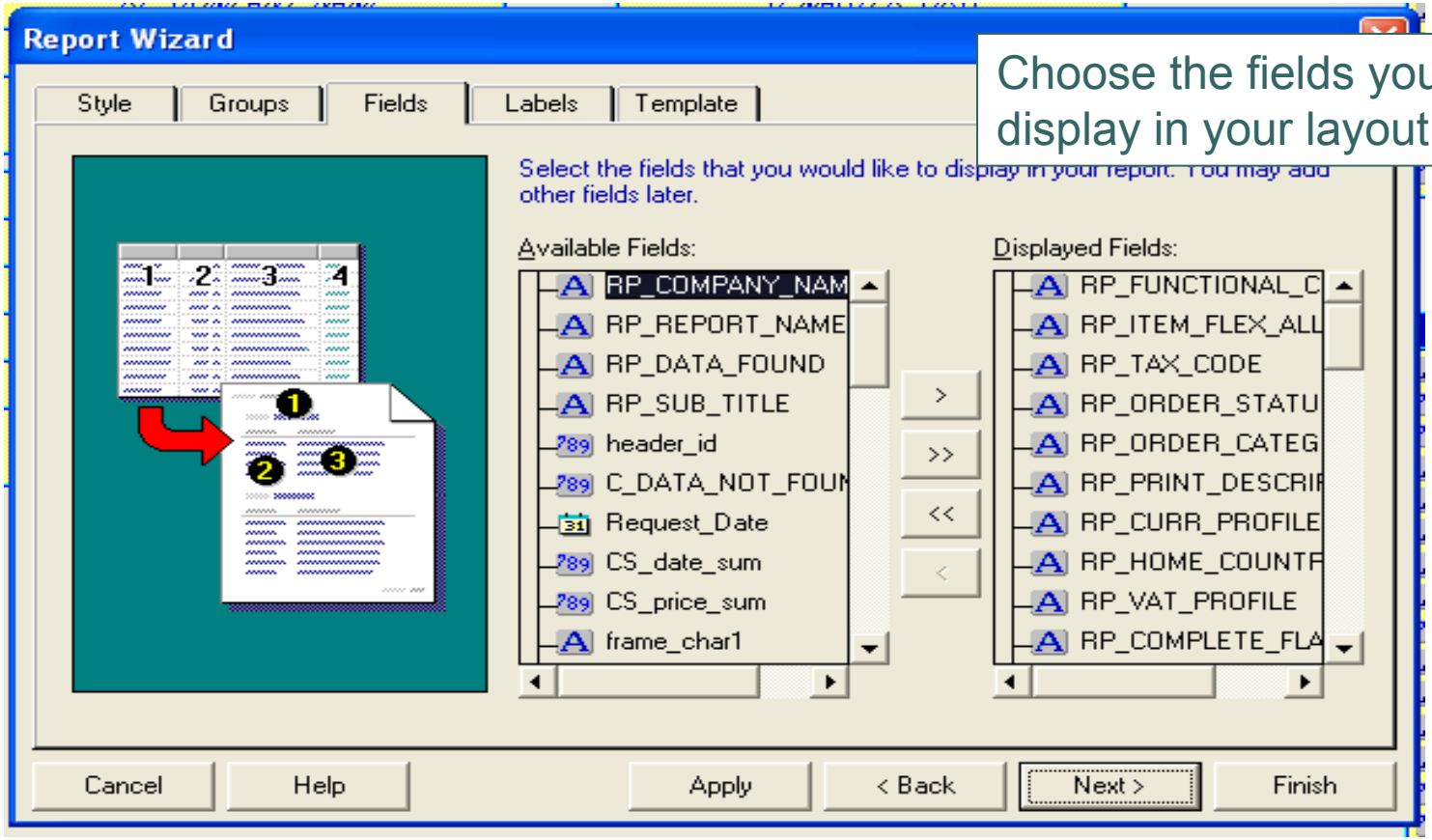
Report Wizard

Select
Report groups
You want to display in the layout.



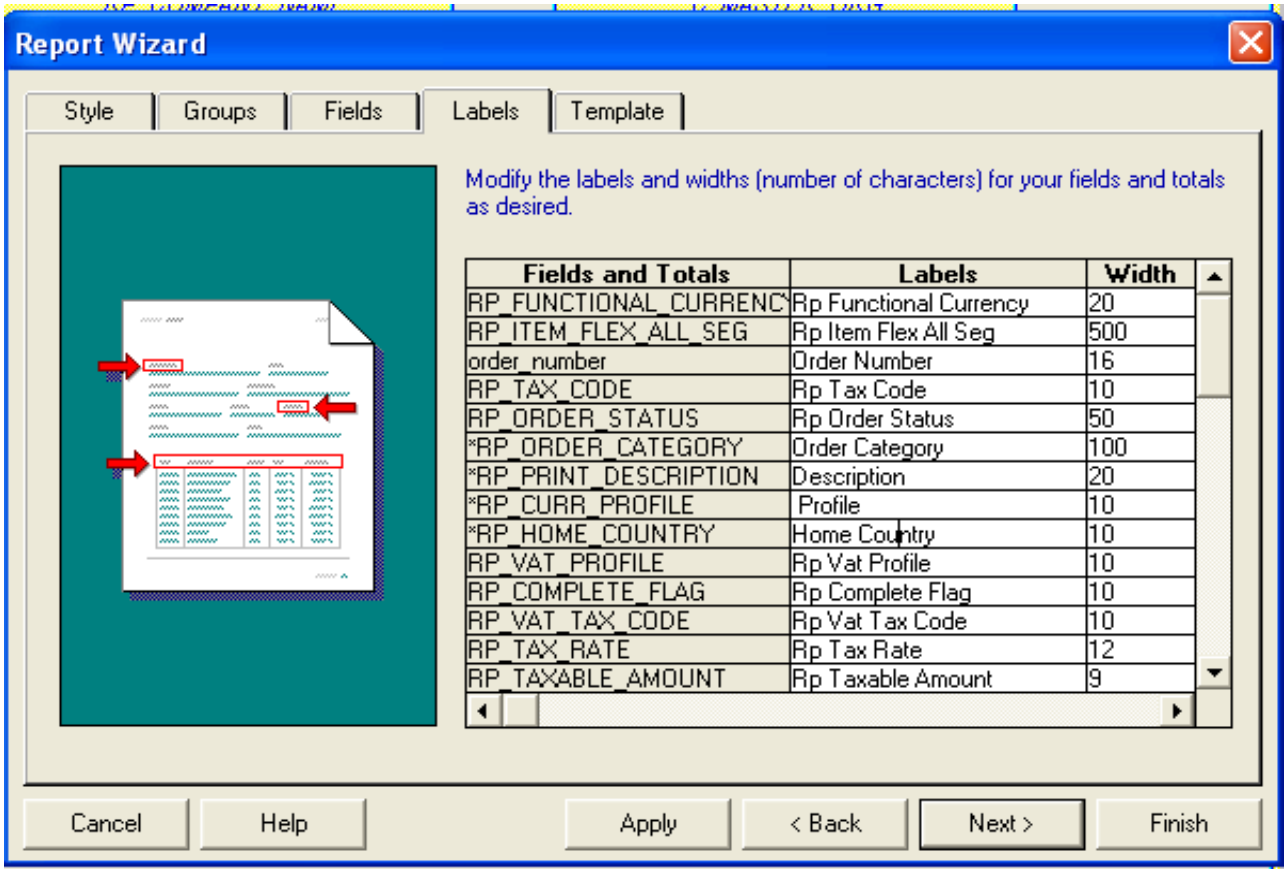
Working in the Layout Editor

Report Wizard



Choose the fields you want to display in your layout

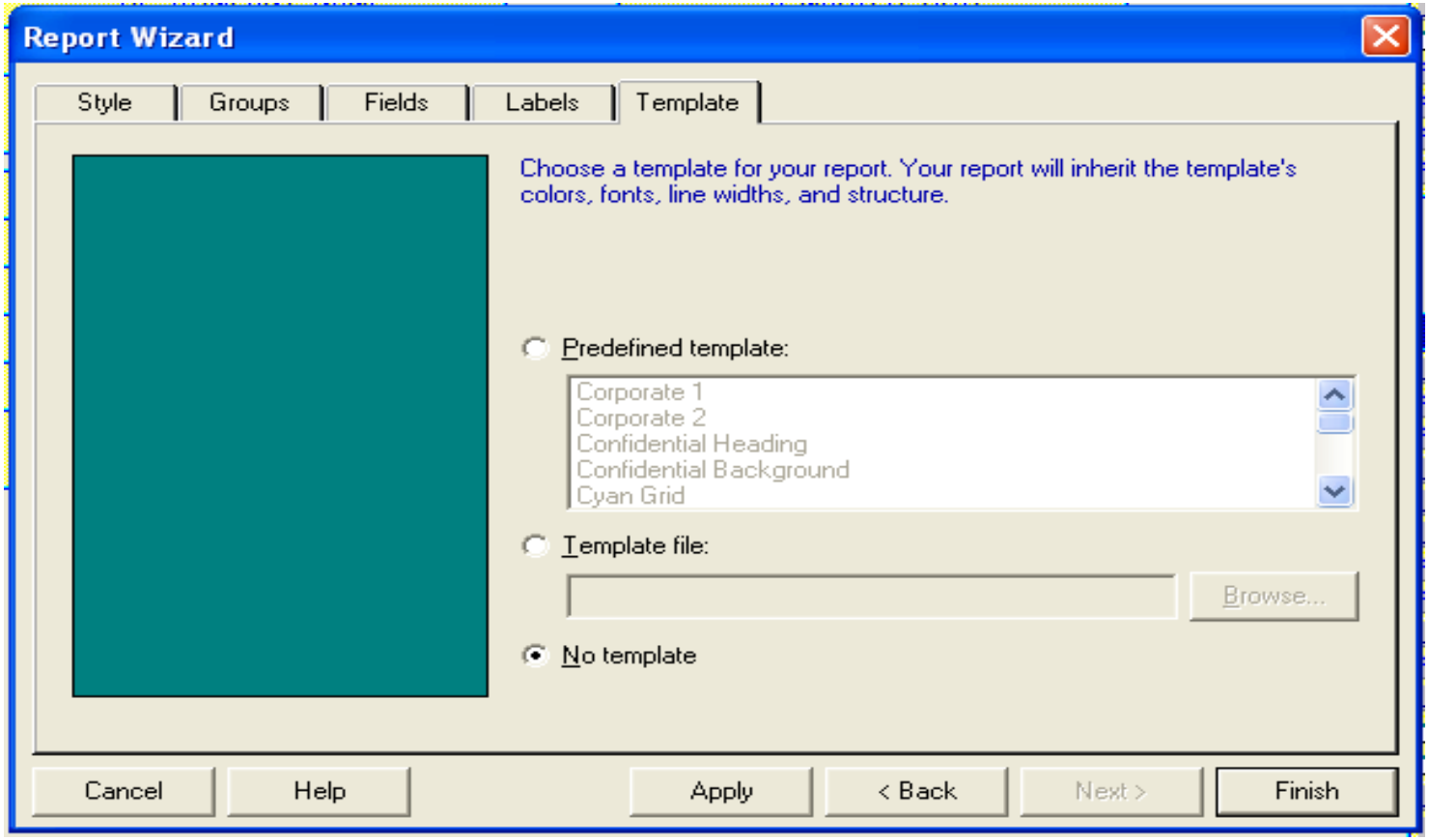
Working in the Layout Editor Report Wizard



Give each field a meaningful label

Working in the Layout Editor

Report Wizard

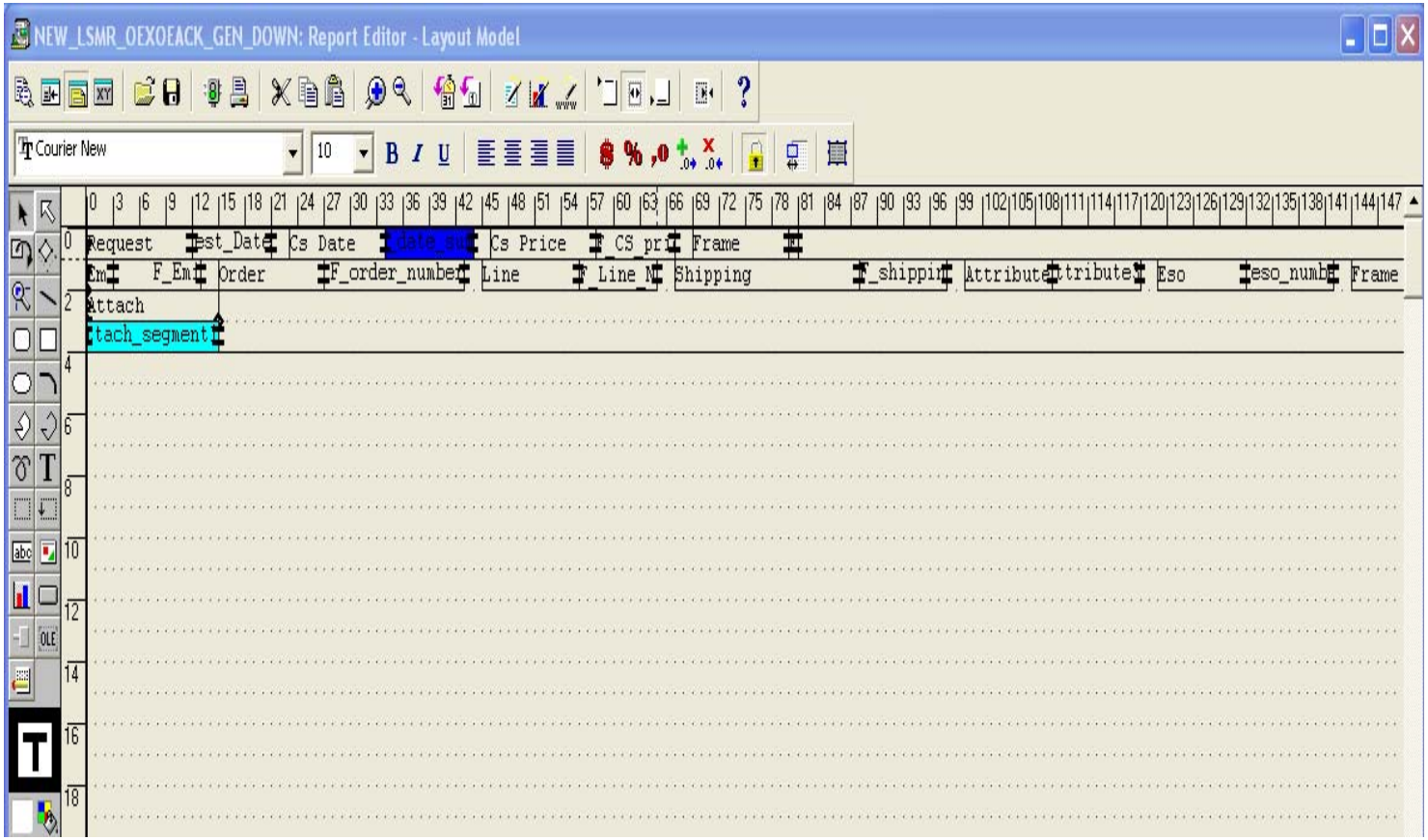


Working in the Layout Editor
Report Wizard

- Must be connected to the database to finish with the Report Wizard.
- Report Layout will then be ready to work with any fields that were chosen in the wizard steps.

Working in the Layout Editor

Report Wizard



Working in the Layout Editor
Frames

A frame protects the objects it encloses and controls the frequency with which those objects are formatted. Frames can enclose other frames.

Variable Group Frames

Reports generates one group frame around the entire layout when creating most layouts. Group frames are normally named with the convention 'G_<name>'

Fixed Frames

A fixed sized frame is normally created around boilerplate objects in the layout.

Working in the Layout Editor
Frames

Repeating Frames

Reports generates one repeating frame for each group when creating a default layout and places fields inside the repeating frame. Repeating frames can include other repeating frames. Also, they can control record level formatting. Usually named 'R_<name>'

Properties of Repeating Frames

- **Variable or Fixed**
- **Space between records**

Working in the Layout Editor
Frames

Repeating Frames - Con't

Relationship to Data Model

Each Repeating frame has a data group as its source. The repeating frame therefore prints once for each record in this source group.

In a mater/detail or control break hierarchy, the detail repeating frame must be nested inside the master repeating frame.

Working in the Layout Editor

Frames

Visual Frames Properties

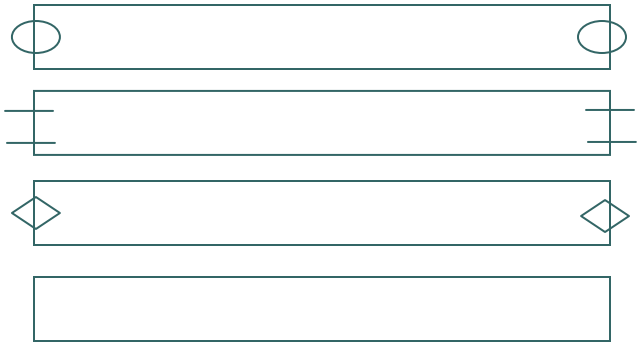
Frame Sizing

Contract

Expand

Variable

Fixed



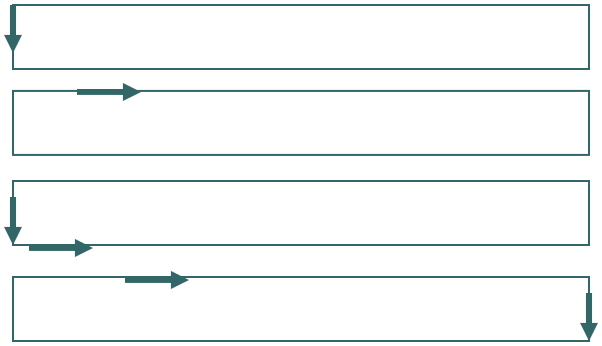
Print Direction (repeating frames)

Down

Across

Down/Across

Across/Down



Working in the Layout Editor
Fields

Source of Fields

- **System Variable such as a date or page number**
- **Column name from the query**
- **System Parameters, such as item number, revision**

Valid Source Columns

The source of the field must be in the relevant group for the repeating frame

Working in the Layout Editor
Fields

A field is a place holder for a column or a parameter, including page numbers, and current date. Reports creates one field for each column when you select the default layout, and places these fields in a repeating frame.

Relationship to Data Model

- **A column, variable, or parameter as defined in the report data model**
- **A system variable such as a current date, or page number**

Working in the Layout Editor
Anchors

An anchor determines vertical/horizontal positioning of the parent/child objects.

There are two types of anchors

- Implicit (not visible, Reports created)

  By Default, Reports implicitly anchors each object to its immediate enclosing object.

- Explicit (visible, Designer created)

Working in the Layout Editor
Anchors

Anchor Properties

Position

You can adjust the position of the anchor in the property palette.

Collapse

You can collapse the horizontal/vertical space between the anchor

Displaying Anchoring Information

Anchoring information can be displayed via the Object Navigator. Choose Navigator --> Navigator Options menu; the the Layout tab check the Anchoring Information Box. This allows you to see all anchors, both implicit and explicit.

- Red means parent object
- Blue means child object

Working in the Layout Editor
Pagination

Four Pagination Properties

- Page Break Before - force the object to be formatted on the next logical page after the one on which it would initially print
- Page Break After - force all external child objects to print on the next logical page
- Page Protect - cause the entire object and its enclosed objects to be kept together on the same logical page.
- Keep With Anchoring Object - cause the object and the object to which it's anchored to be kept together on the same logical page

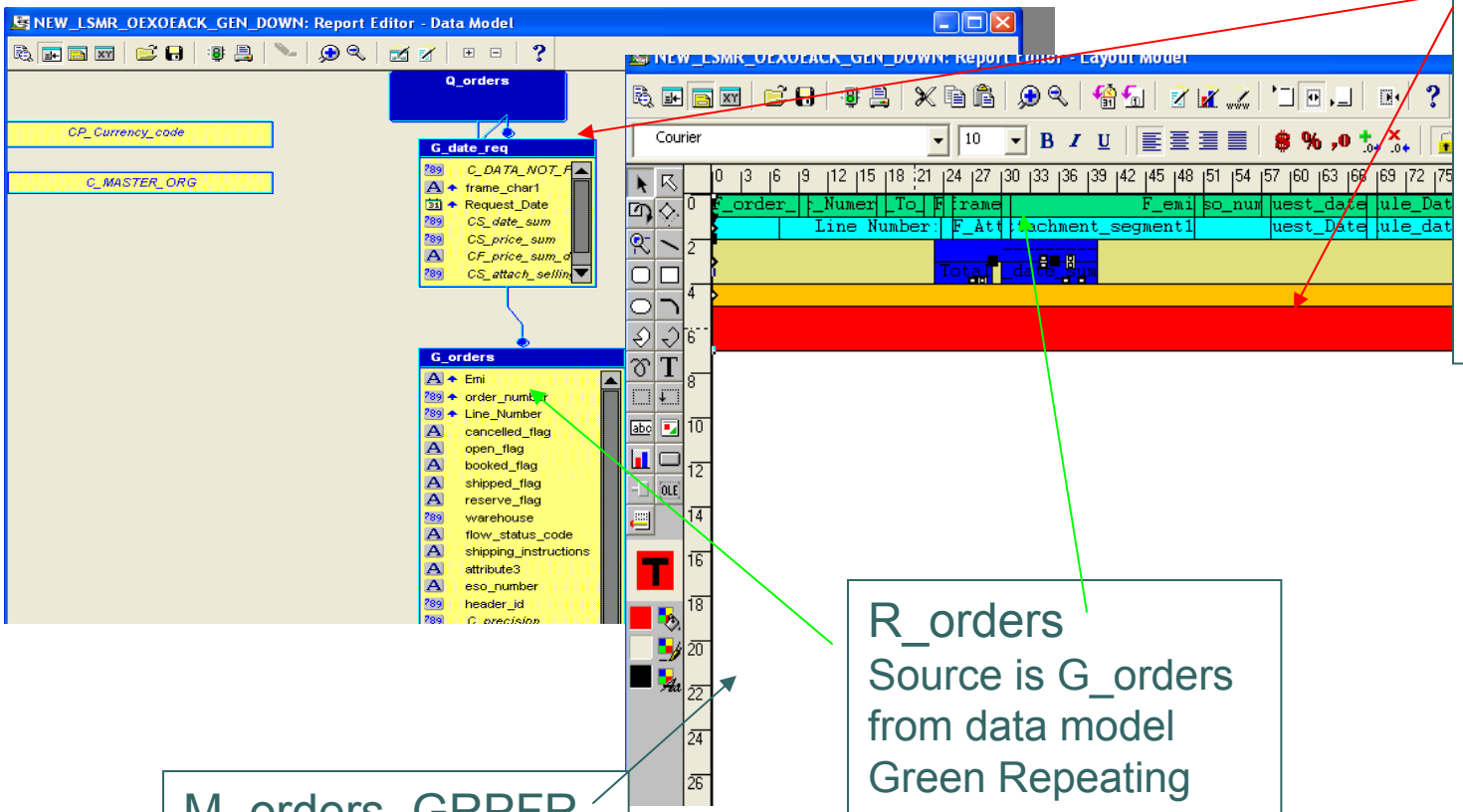
Working in the Layout Editor
Print Condition

Print Object On indicate how often the object in relation to another "parent" object.

Print Object On	Object
All	All logical pages of the parent
All But First/Last	All logical pages except first/last page
First/Last	The first/last page only.
Default	Reports sets this type, using an algorithm

Working in the Layout Editor

Data and Layout Relationship



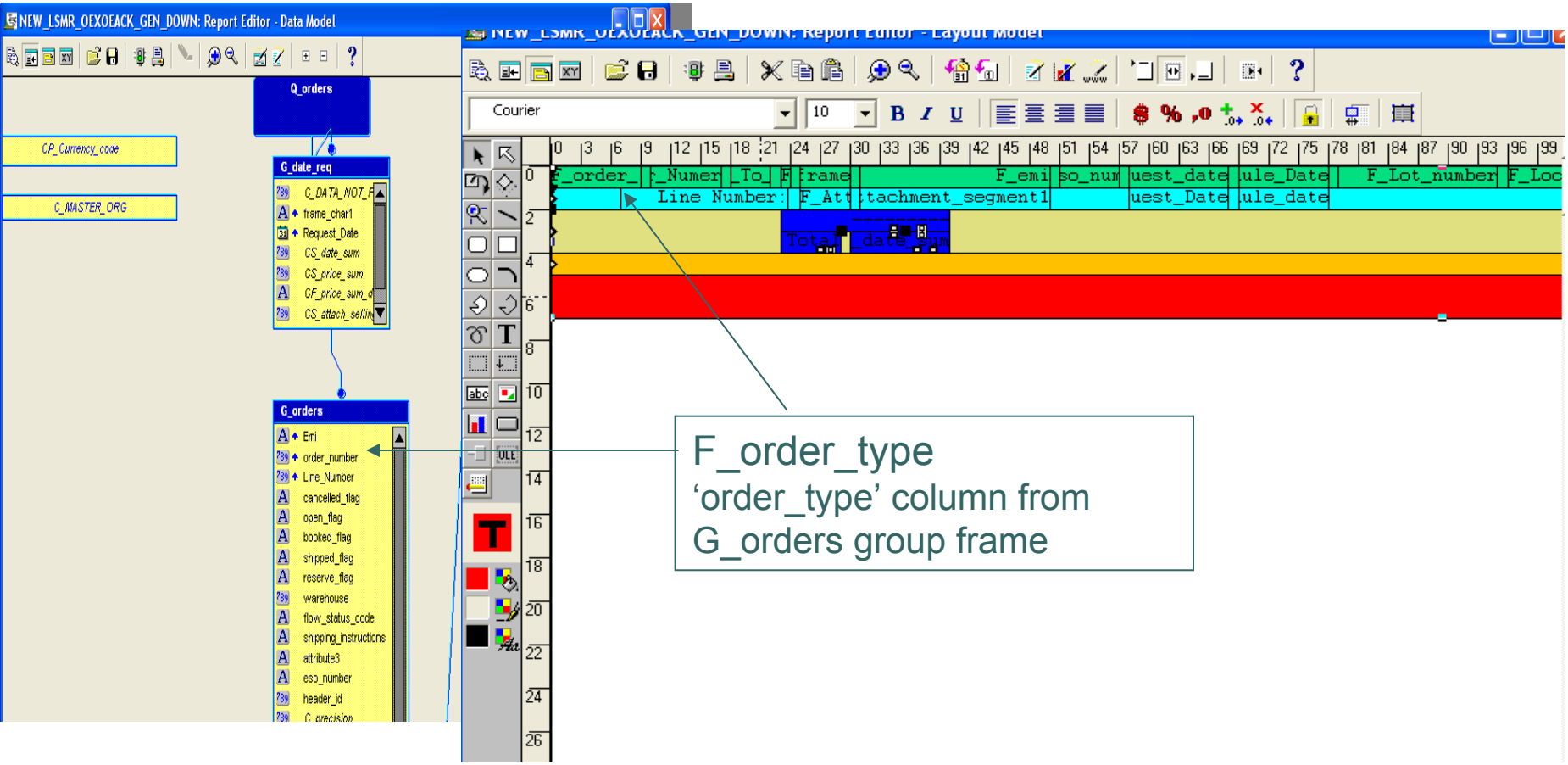
R_date_req
Source is G_date_req from data model. Red repeating frame that encloses all other frames (except M_orders_GRPFR)

R_orders
Source is G_orders from data model
Green Repeating Frame
(also surrounds turquois frame)

M_orders_GRPFR
White frame surrounding entire layout

Working in the Layout Editor

Data and Layout Relationship



Working in the Layout Editor

Modify Layout and Object Navigator

- It is best to work with the object navigator visible when working in layout editor…especially when adding new columns to your layout, or moving fields around. This allows you to see which groups you are in.

- When you click on a field in the report layout, you will see this field highlighted in the object navigator.

Working in the Layout Editor

Modify Layout - Tips

- Use colors on group frames and repeating frames to visually see what level you are working at.
- If users want do not want to see certain group of data, try to add a group frame around those objects and put a format trigger set to false. They may change their minds later.
- Try not to double-click or right click on an object. Reports is kind of quirky sometimes and field or frame may disappear. Instead, use Tools > Property Palette option.
- Use Arrow keys to move fields. Try not to move with mouse.

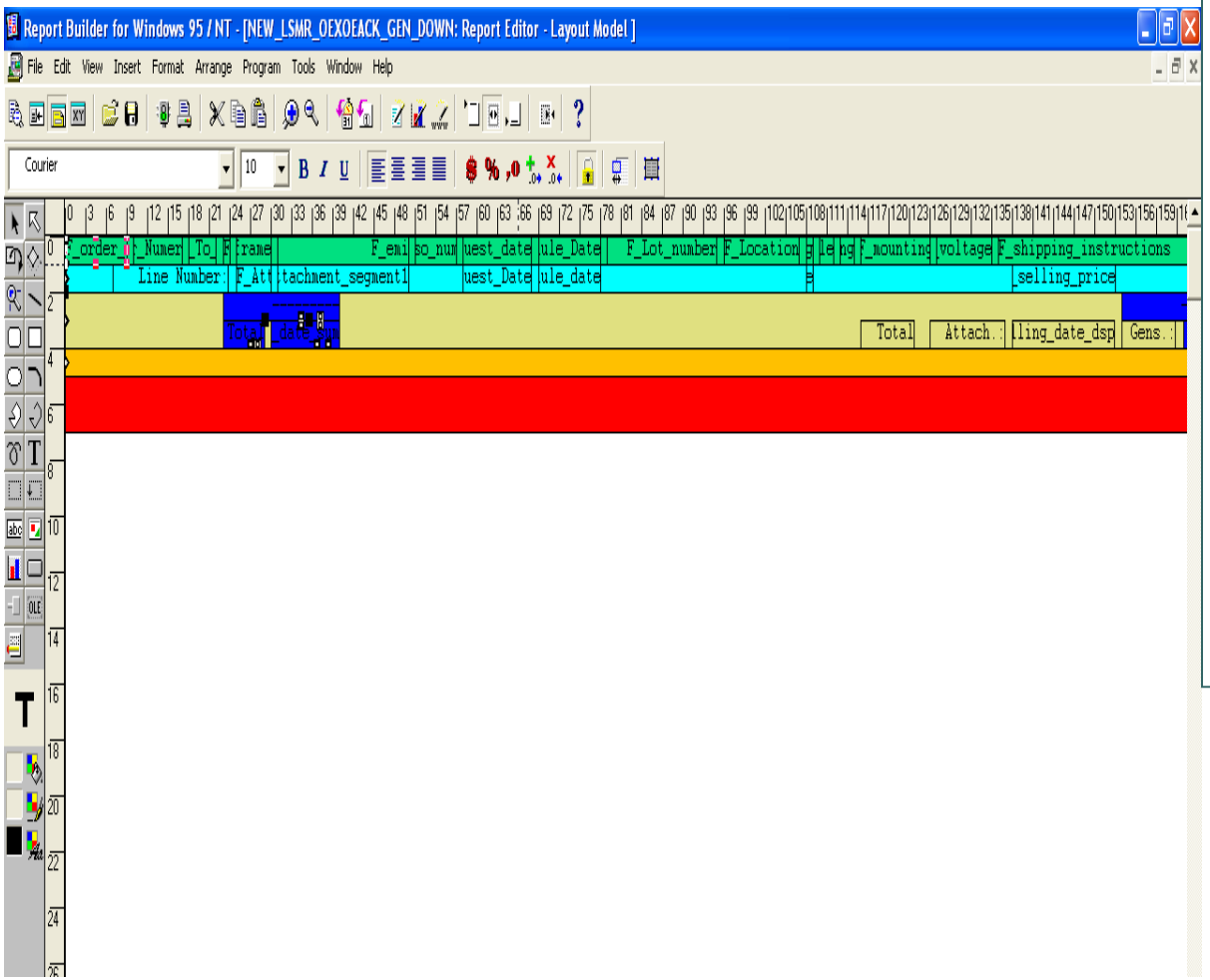
Working in the Layout Editor

Modify Layout - Tips

- SAVE OFTEN!!…along with different versions.
  - » Save after:
    - Interim milestones
    - Major milestones
    - Daily with description of changes from previous day

Working in the Layout Editor

Modify Layout

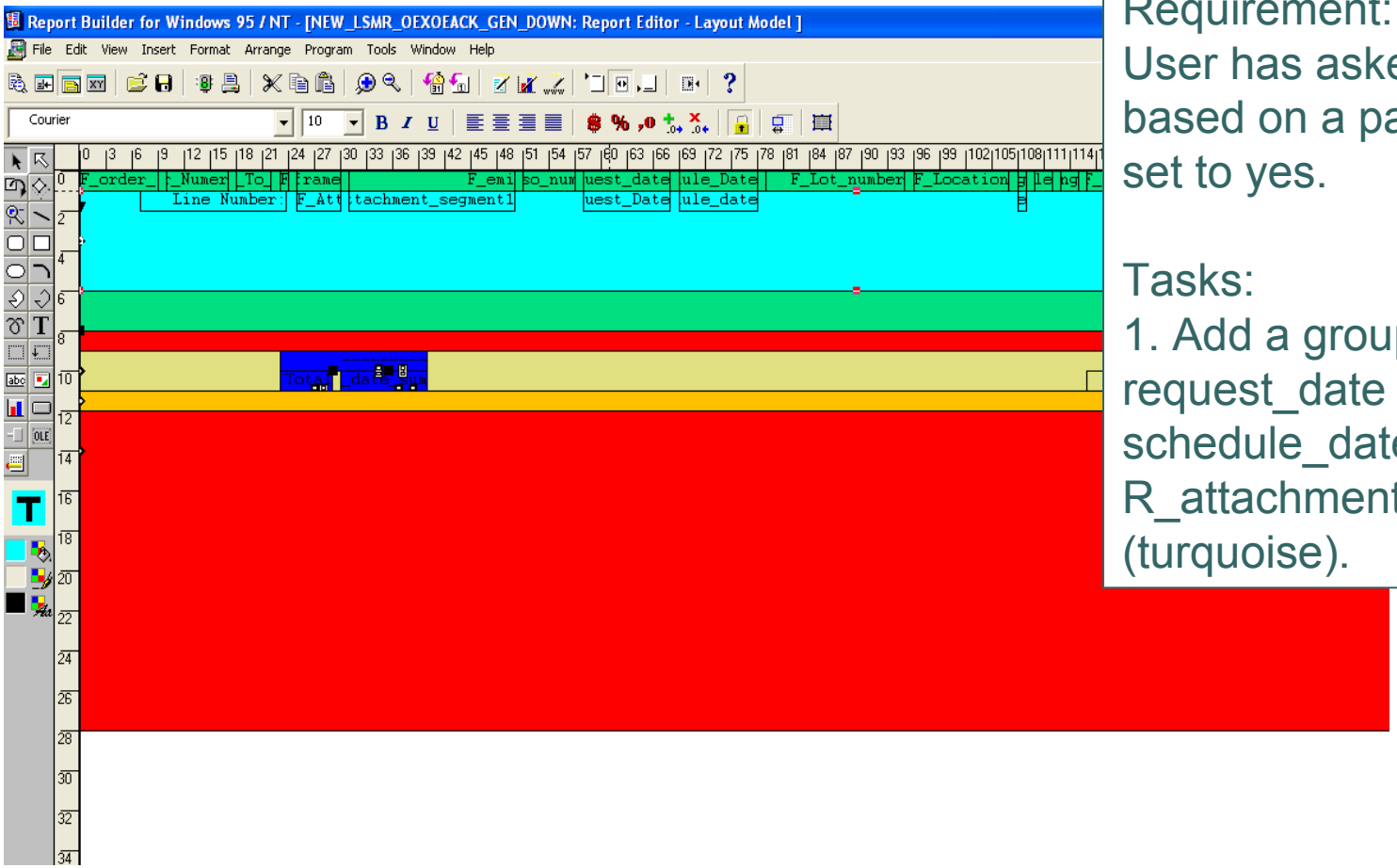


If you have to add a group frame or repeating frame, or several fields,
Add plenty of extra room so that you can work.

Do this by starting at the outer most frame, which can be found by looking in the object navigator.

Working in the Layout Editor

Modify Layout



Lab:

Requirement:

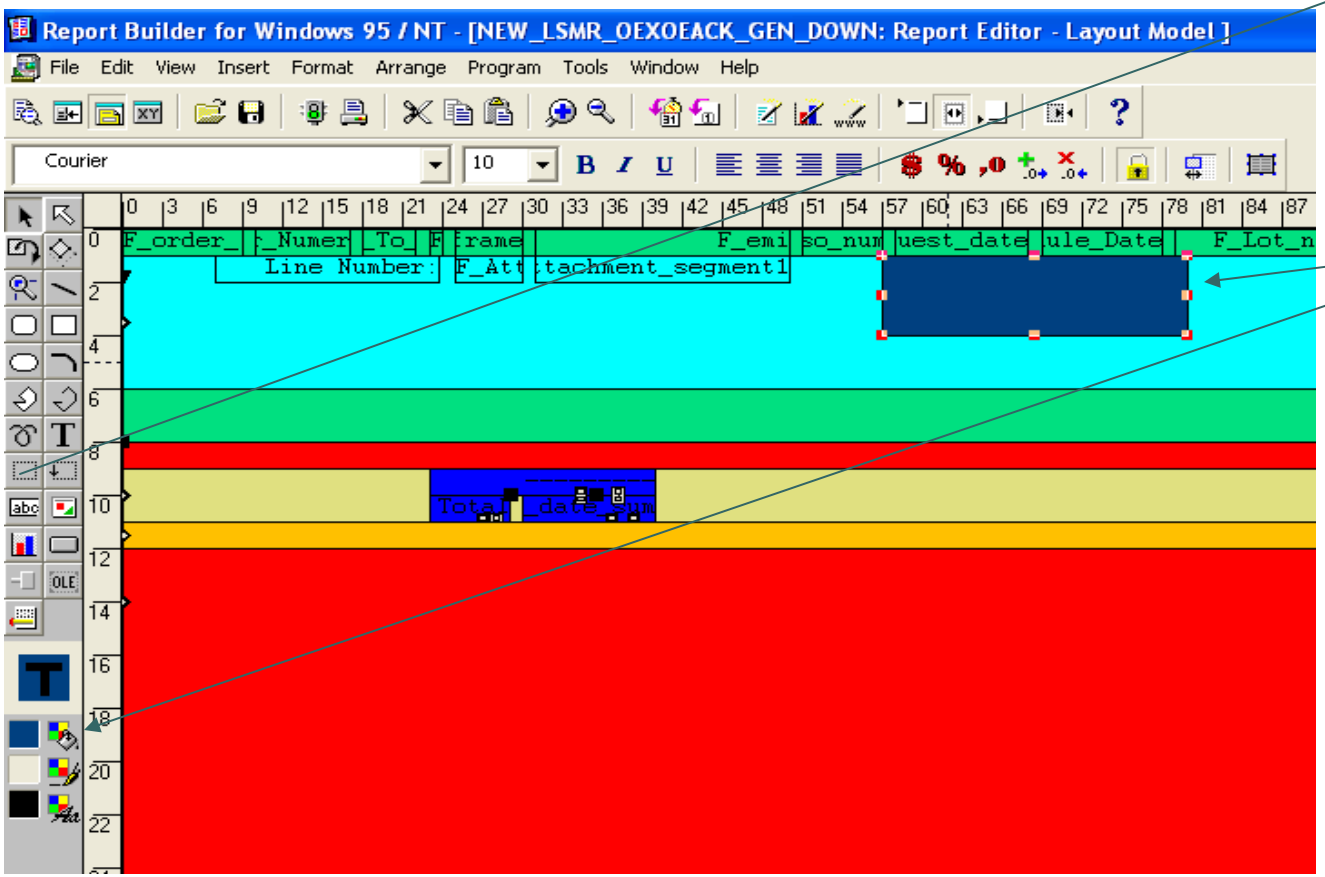
User has asked to see these based on a parameter being set to yes.

Tasks:

1. Add a group frame around request_date and schedule_date in the R_attachments group frame (turquoise).

Working in the Layout Editor

Modify Layout

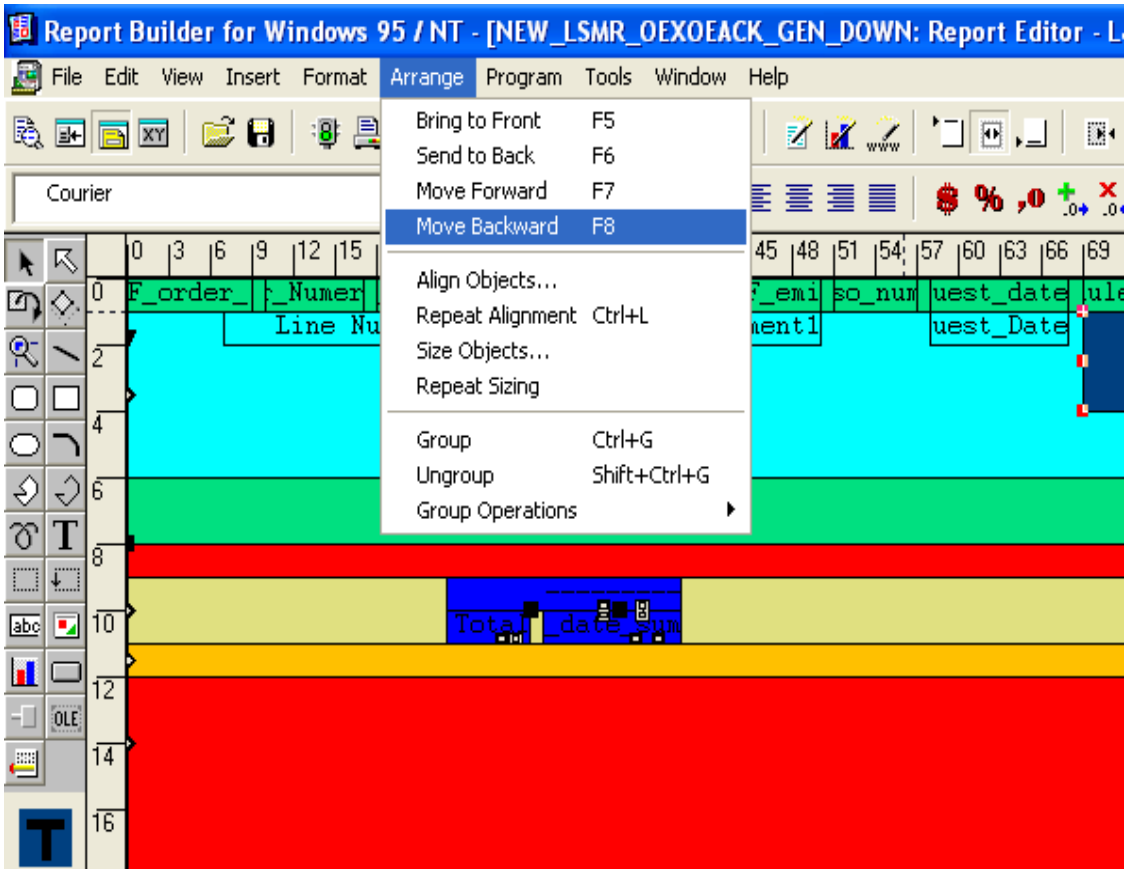


Click Icon to add new ‘Frame’

Draw frame over the date fields, and then change color of the frame.

Working in the Layout Editor

Modify Layout



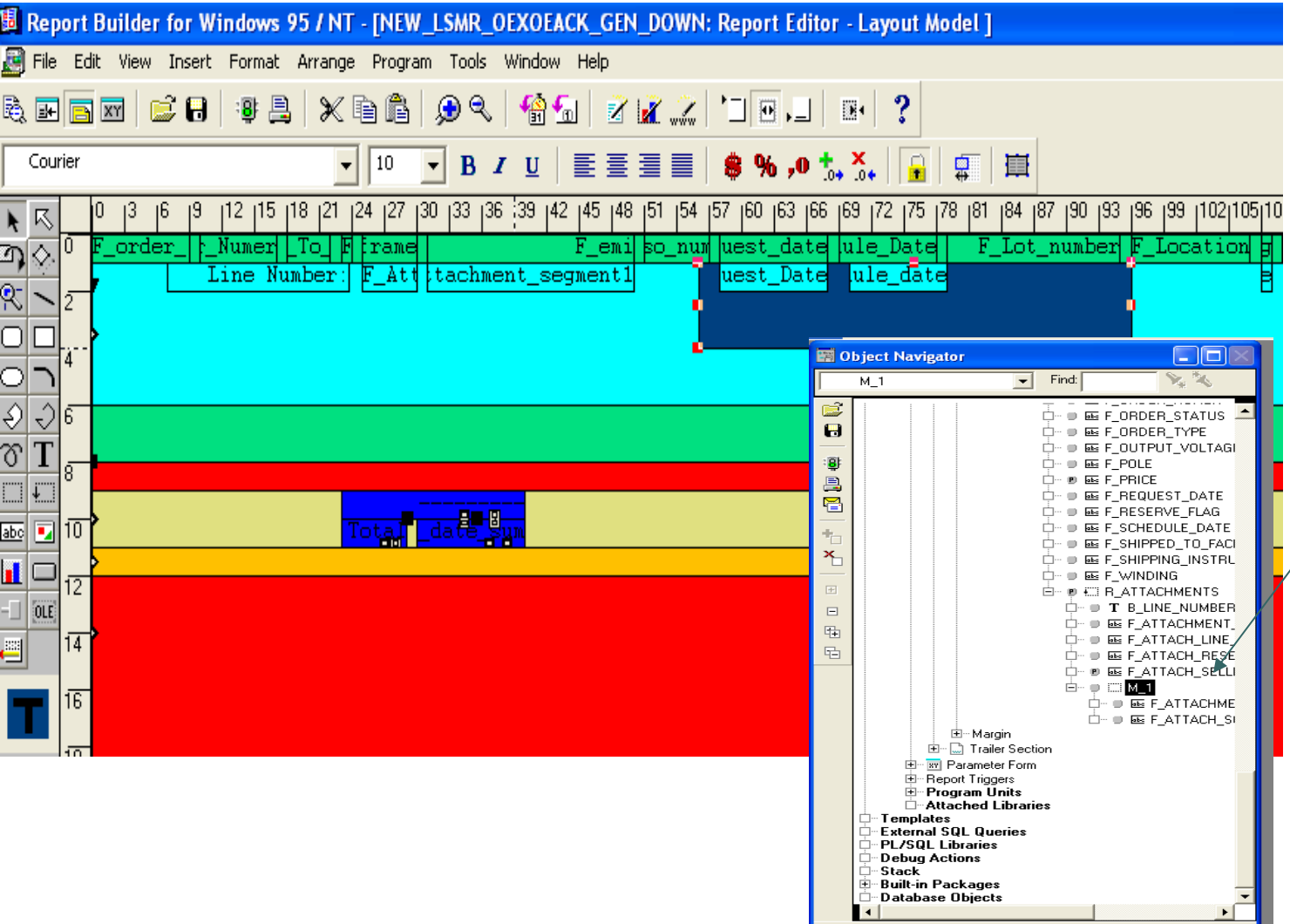
Highlight new frame.
Go to
Arrange > Move Backward

Shortcut for this is F8 key.

This will move the new group frame ‘behind’ the date fields. Note: the group frame must completely surround the frames you want to enclose.

Working in the layout Editor

Modify Layout

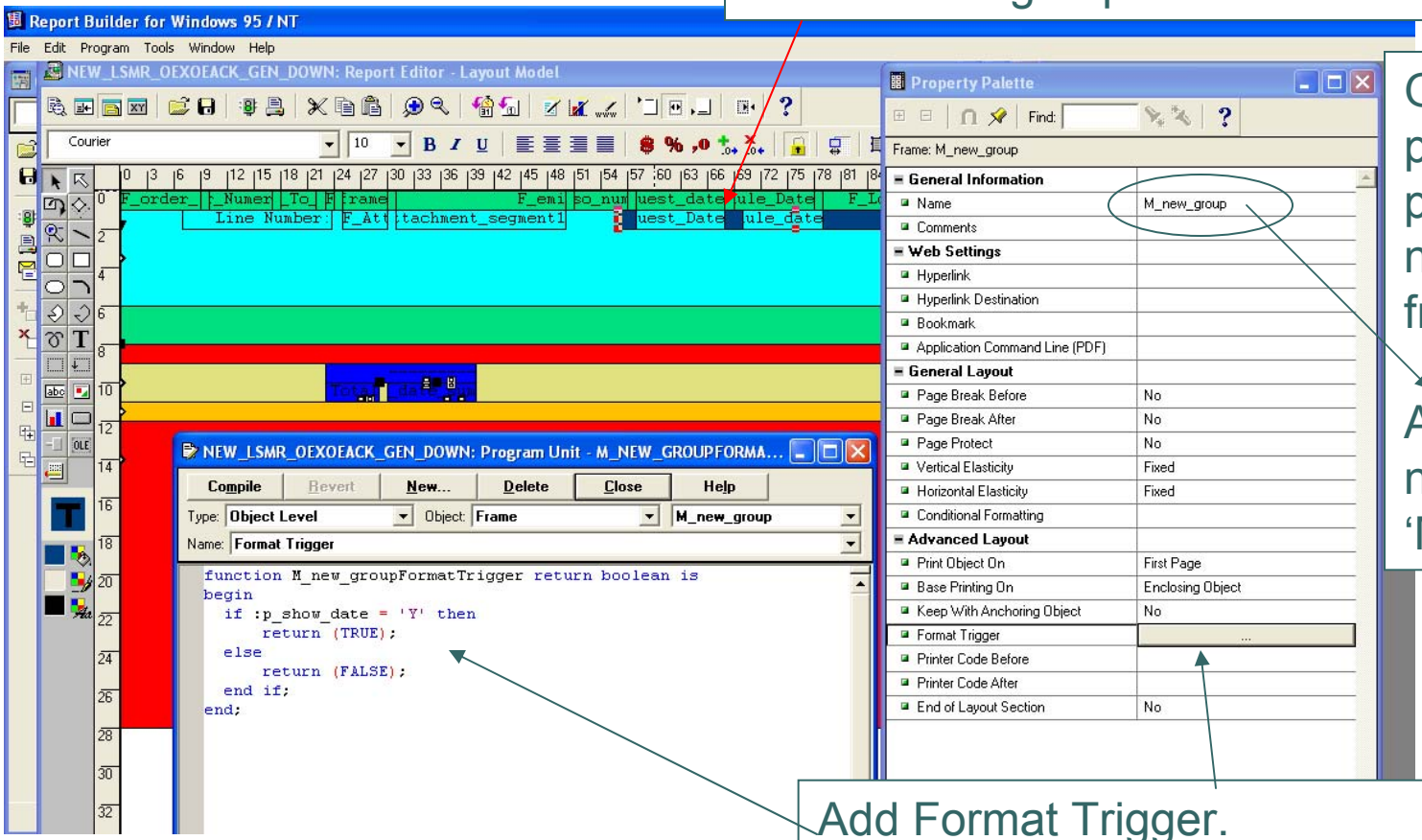


Date frames are now enclosed by group frame.

Verify with Object Navigator

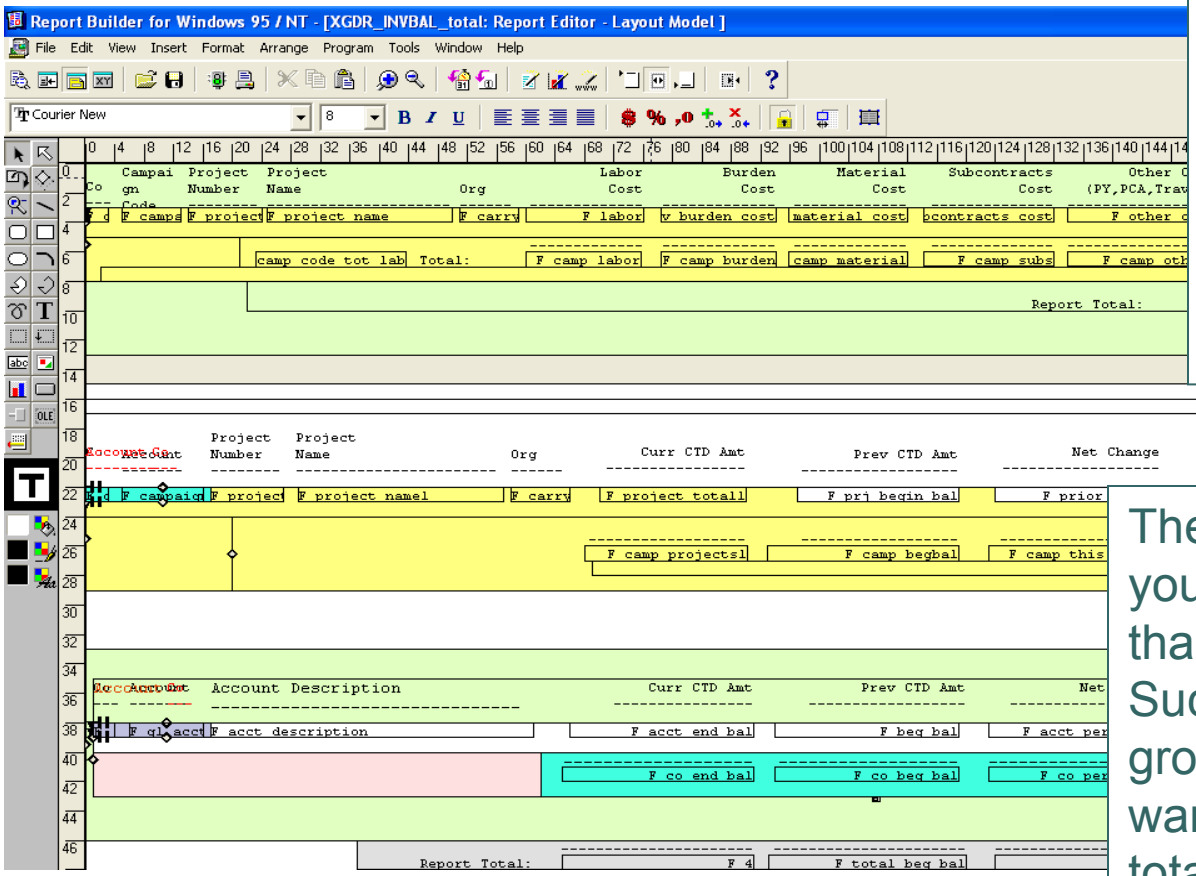
Working in the Layout Editor

Modify Layout



Working in the Layout Editor

More than one Layout



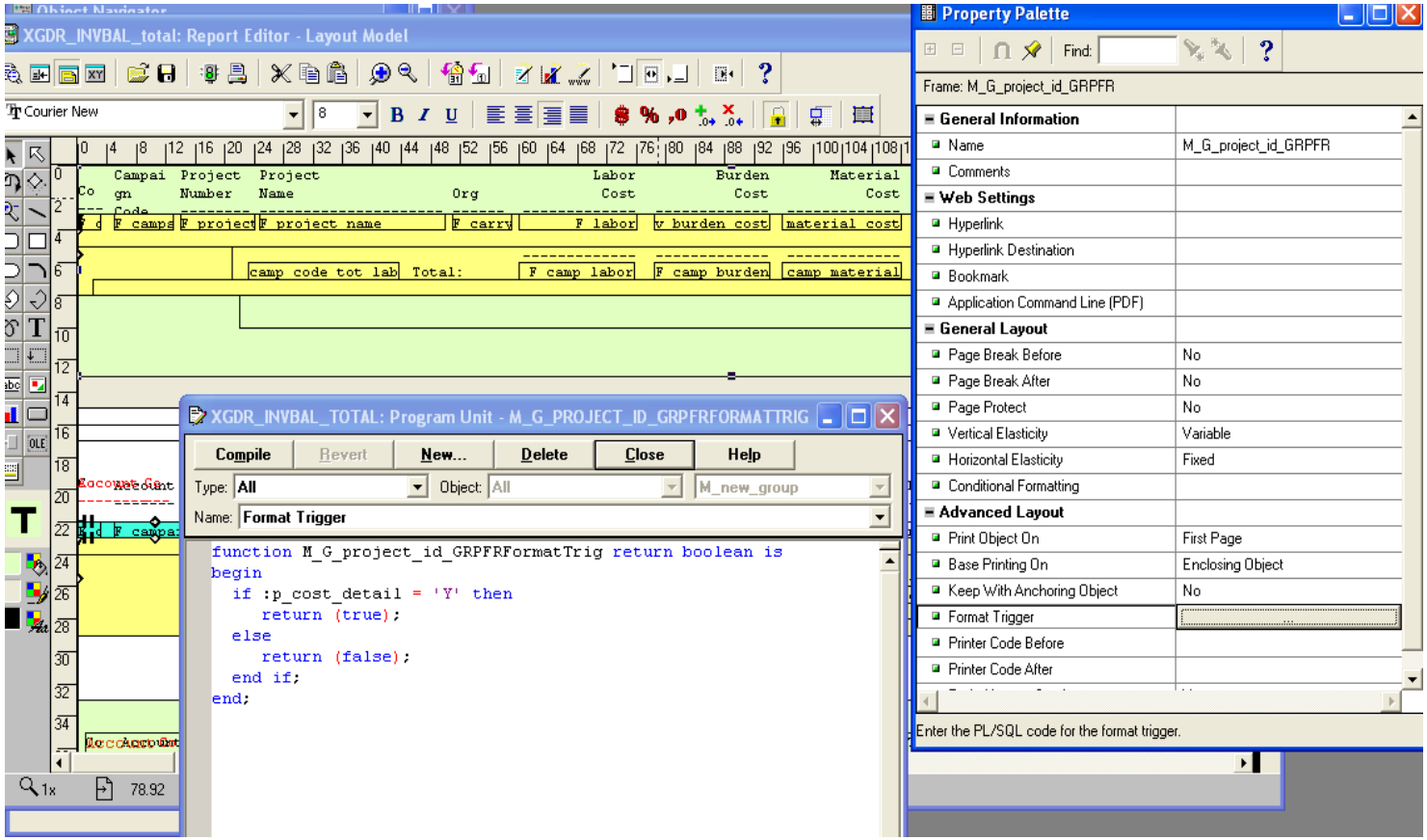
Build More than one layout from one Data Model.
After first layout, you must build manually. Copy and paste can work very carefully.
Save often.

There are several reasons you may want to build more than one layout,
Such as different sorts or groupings, different users want same data, but in totally different formats.

Working in the Layout Editor

More than one Layout

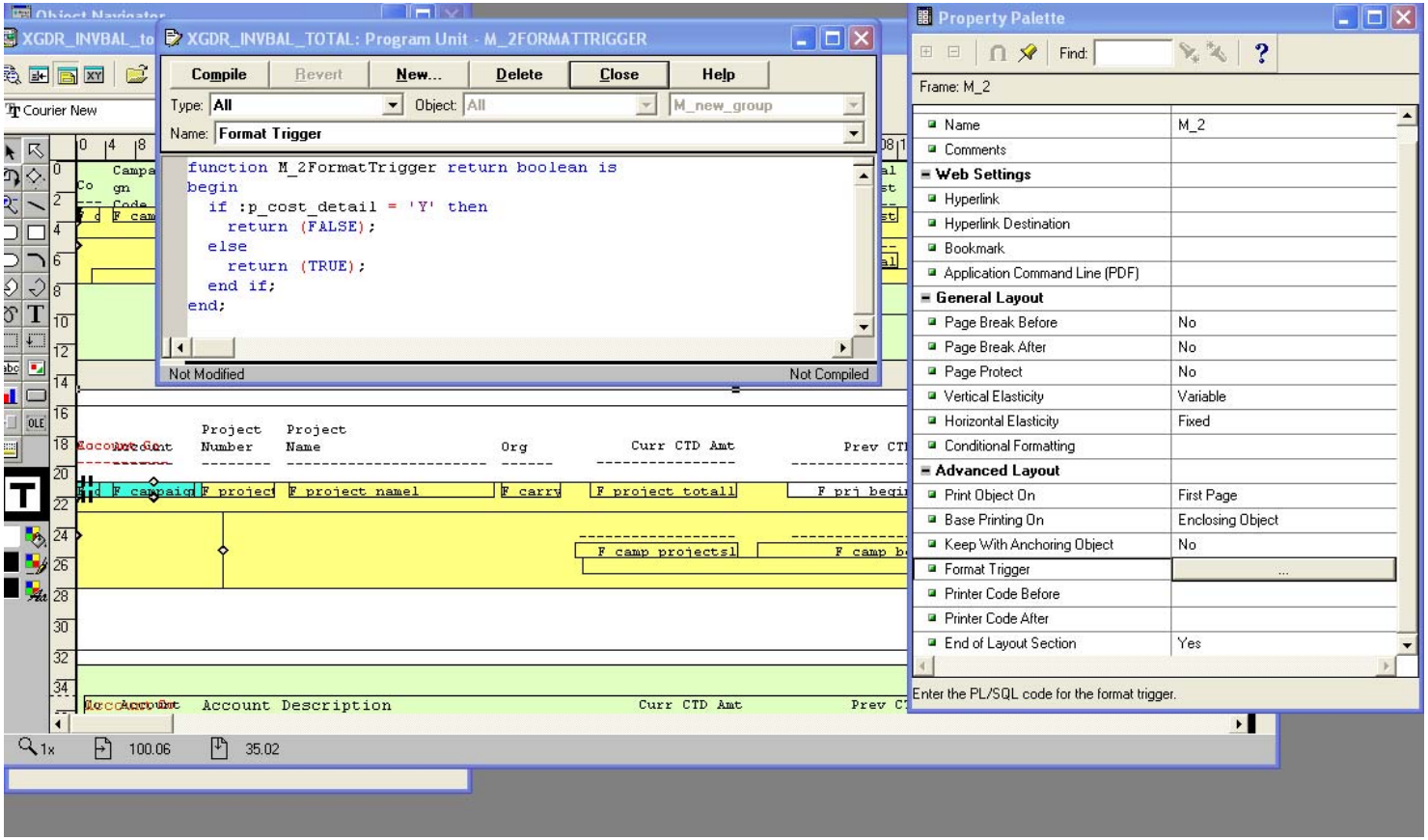
Format Trigger for outer most frame on first layout.
This one is based on parameter…could possibly be based on data:



Working in the Layout Editor

More than one Layout

Format Trigger for Second Layout based on parameter:



Advanced Topics

- Creating User Parameters
- Bind References
- Lexical Parameters
- Triggers
- Report Properties
- SRW.EXITS

Advanced Topics

Creating User Parameters

You can create your own parameters and use them to change the SELECT statement of your query at runtime.

A USER parameter is an object that you create to hold a value that users can change at runtime.

- Restrict values in the WHERE clause of the SELECT statement
- Substitute any part of the SELECT statement
- Substitute a single column or expression in the SELECT statement.
- Restrict printing in the Layout.

Relationship to AOL Parameters

There is a direct relationship between AOL and Reports Parameters.

Adavanced Topics

Referencing Parameters in a Report

There are two ways you can reference parameters in a query: use a bind reference or a lexical reference.

Bind Reference - replaces a single value or expression.

- To create a bind references in a query, reference the parameter with a colon (:).

Lexical Reference - replaces any part of a SELECT statement, such as column names, the FROM clause, the WHERE clause, or the ORDER BY clause.

- To create a lexical reference in a query, prefix the parameter name with an ampersand (&)

Advanced Topics

Comparing Bind and Lexical Parameters

Type	Prefix	Use to replace	Parameter Created by Default
Bind exist	:	WHERE, GROUP BY ORDER BY, HAVING CONNECT BY, START WITH	Yes, if it does not already
Lexical Object	&	any part of the SELECT statement	No, You must always create the parameter in the Navigator yourself. Data type must be **CHARACTER**

Advanced Topics

Bind References - Examples

SINGLE LITERAL VALUE:

```
SELECT order_number, customer_id
FROM          so_headers_all
WHERE         header_id = :P_HEADER_ID
```

VARIOUS REPORTS OPTIONS

```
SELECT order_number, customer_id
FROM          so_headers_all
SORT BY DECODE (:P_SORT, 1, order_number, 2, customer_id)
```

Advanced Topics

Lexical Parameters - Examples

REPLACING THE WHERE or ORDER BY CLAUSE:

```
SELECT order_number, customer_id
FROM    so_headers_all
&P_WHERE_CLAUSE
&P_ORDER_BY
```

Specify Column Names

```
SELECT &P_ORDER_NUMBER, customer_id
FROM          so_headers_all
SORT BY DECODE (:P_SORT, 1, order_number, 2, customer_id)
```

Advanced Topics

Building Lexical Parameters

Use the After Parameter Form report trigger to build a dynamic where clause depending on the value of a bind parameter that the user enters at runtime.

- EXAMPLE

```
FUNCTION AfterPForm RETURN BOOLEAN IS
BEGIN
IF :P_ORDER_NUMBER IS NULL THEN
        :P_WHERE_CLAUSE := '';
ELSE
        :P_WHERE_CLAUSE := 'where sha,order_number = :P_ORDER_NUMBER';
END IF;
RETURN (TRUE);
END;
```

Advanced Topics

- Enter a initial value for parameters that affect query validation when NULL
- Create lexical parameters explicitly in the Object Navigator
- For lexical parameters, define the a valid clause

Advanced Topics

Data Model Triggers - Formula

You can code a PL/SQL function in a formula column to populate other objects, such as placeholder columns, as well as the column itself.

EXAMPLE

```
FUNCTION set_bonus RETURN CHAR IS
BEGIN
IF :salary > 10000 THEN
        :bonus := salary * .10;
ELSE
        :bonus := salary * .05;
END IF;
RETURN (‘ ‘);  -- to populate the formula column
END;
```

Advanced Topics

Layout Triggers

Format triggers allow you to modify the display of objects dynamically at runtime, or to suppress display altogether.

- Frames
- Repeating Frames
- Fields
- Boilerplate Objects

Advanced Topics

Format Trigger - Example

```
FUNCTION hide_sales_order_number RETURN BOOLEAN IS
BEGIN
IF :P_ORDER_NUMBER IS NULL THEN
        return(false);
ELSE
        return(true);
END IF;
END;
```

Advanced Topics

SRW - Format Fields

The SRW references are used to modify objects into the report.

```
function c_extended_price_dspFormula return Char is
begin

 srw.reference (:currency1);
 srw.reference (:c_extended_price);
 srw.reference (:RP_CURR_PROFILE);
 srw.user_exit (
              'FND FORMAT_CURRENCY
               CODE=":currency1"
               DISPLAY_WIDTH="13"
               AMOUNT=":c_extended_price"
               DISPLAY=":c_extended_price_dsp"
               ');
return (:c_extended_price_dsp);

end;
```

Advanced Topics

SRW - Set Environment for Multi Org

Before Report Trigger - Sets the Views

```
BEGIN

  BEGIN
  SRW.USER_EXIT('FND SRWINIT');
  EXCEPTION
    WHEN SRW.USER_EXIT_FAILURE THEN
            SRW.MESSAGE (1000,'Failed in BEFORE REPORT trigger -
SRWINIT USER EXIT');
    return (FALSE);
  END;
```

Advanced Topics

SRW - Set Environment for Multi Org

After Report Trigger - Resets the Environment

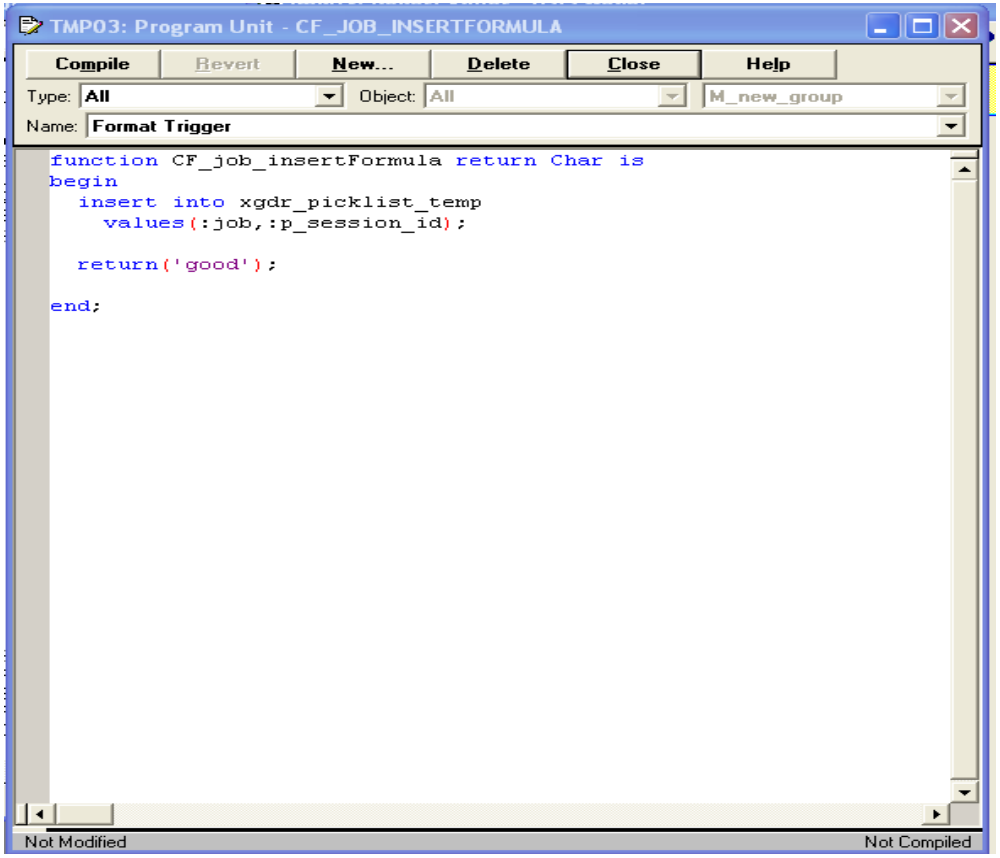
```
SRW.USER_EXIT('FND SRWEXIT');

EXCEPTION
   when srw.user_exit_failure then
     srw.message(1,'Failed in AFTER REPORT TRIGGER');
    return (FALSE);

END;  return (TRUE);
```

Advanced Topics

- Example to Populate Report table from a report formula column. Table exists on database:



Advanced Topics

FND_REQEUST.SUBMIT_REQUST

- Used this to call report from a report, form, or pl/sql. If this call is successful, l_request_id = concurrent manager request_id, else it will equal 0:

```
l_request_id := FND_REQUEST.SUBMIT_REQUEST('XGDR',
'XGDR_WIP_LABELS',
'',
'',
FALSE,
jobs.d_job,'','','','','','','','','',
'','','','','','','','','','',
'','','','','','','','','','',
'','','','','','','','','','',
'','','','','','','','','','',
'','','','','','','','','','',
'','','','','','','','','','',
'','','','','','','','','','',
'','','','','','','','','','',
'','','','','','','','','','' );
```

Advanced Topics