




Consolidated Income Statement



Consolidated Income Statement

The Consolidated Income Statement displays the actual, budgeted and respective variance amounts for all revenue and expense accounts in a typical Income Statement format. The report displays data for both the current month and for the year-to-date total.

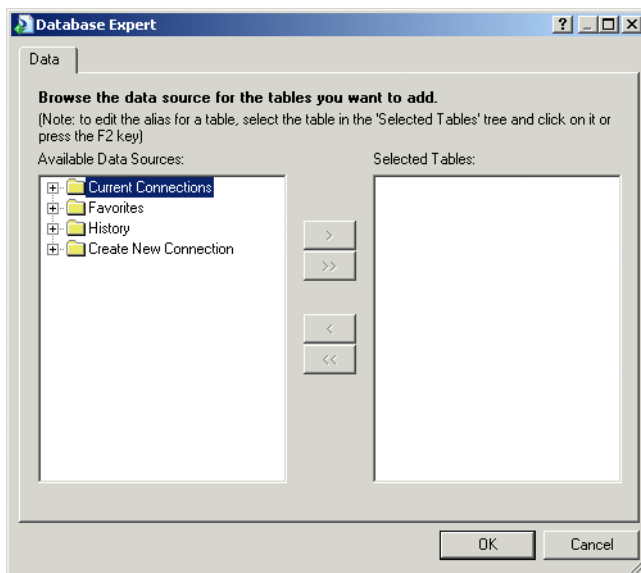
Introduction

The following tutorial assumes that you have completed the “Quick start for new users” tutorial in the *Crystal Reports XI User’s Guide*, which is located in the Docs folder of your product distribution.

Creating the report

On the Start Page in Crystal Reports, click Blank Report.

The Database Expert dialog box appears.



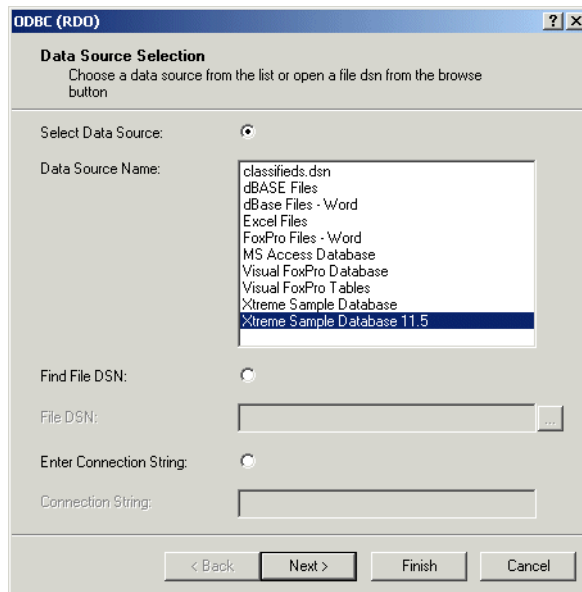
Selecting a database to use

When you create a report, the first thing that you must do is to select a database for the report to use.

► **To select a database**

1. In the Database Expert dialog box, expand the **Create New Connection** folder.
2. Double-click **ODBC (RDO)**.

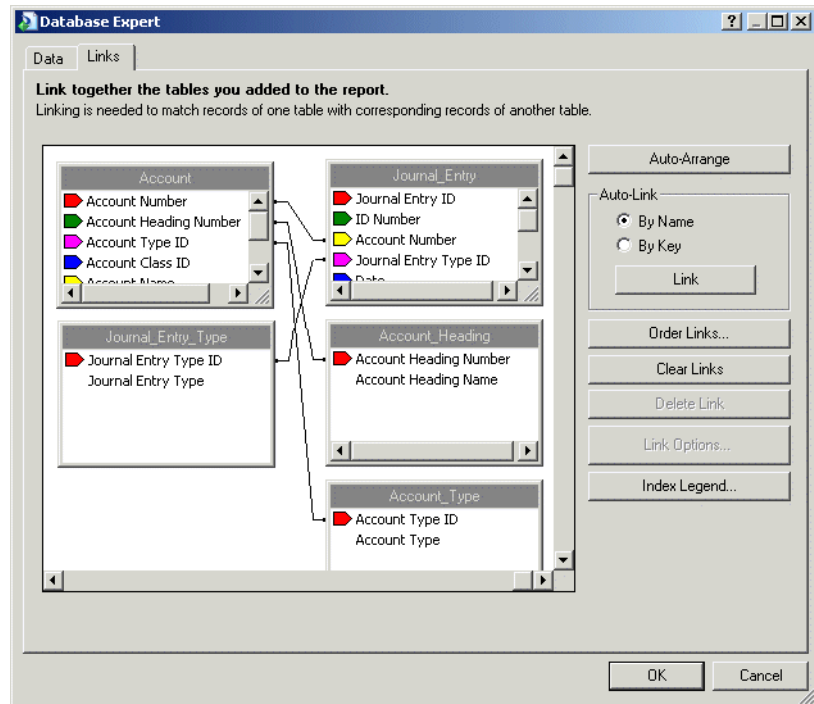
The ODBC (RDO) dialog box appears and displays a list of data sources.



3. From the list, select **Xtreme Sample Database 11.5**, and click **Finish**.
The Database Expert displays a list of available tables from the Xtreme Sample Database.
4. Select the **Account**, **Account Type**, **Account Heading**, **Journal Entry**, and **Journal Entry Type** tables, and click > to add them to the Selected Tables list.

Tip: You can use CTRL+click to select multiple tables simultaneously.

5. Click **OK**.
The Links tab is displayed.



6. Verify that the links between the tables are consistent with what is displayed in the screenshot above. If necessary, create the appropriate links by dragging fields from one table to corresponding fields in another table.

Note: Ensure that the following links exist:

- Account.Account Type ID to Account Type.Account Type ID
- Account.Account Number to Journal_Entry.Account Number
- Account.Account Heading Number to Account_Heading.Account Heading Number
- Journal_Entry.Journal Entry Type ID to Journal_Entry_Type.Journal Entry Type ID

7. Click **OK**.

Creating commands

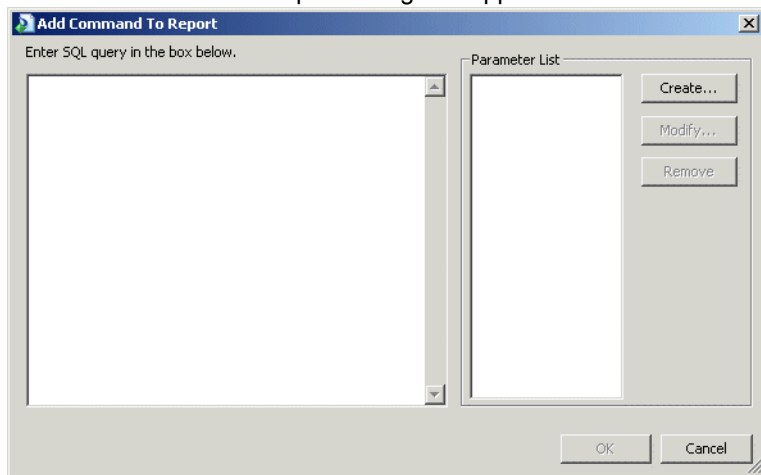
A command object is an SQL expression that is used to return data. The report makes use of two command objects.

► **To create a command**



1. Click **Database Expert**.
2. In the Available Data Sources area, under the **Xtreme Sample Database 11.5** connection, select **Add Command**, and click >.

The Add Command to Report dialog box appears.



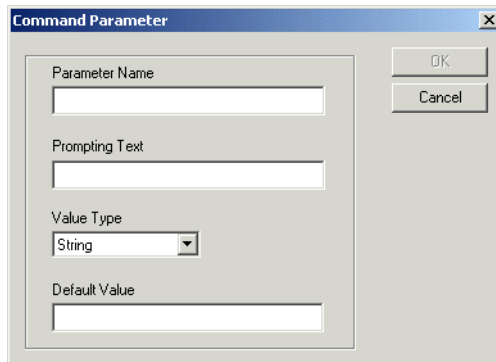
3. In the text area, type the following SQL query:

```
SELECT Sum(Amount) AS Year_BUDGET, [Account Number]
FROM [Monthly Account Budgets]
WHERE Year=Year({?End Date}) and Month <= Month({?End
Date})

GROUP BY [Account Number];
```

4. Click **Create**

The Command Parameter dialog box appears.



The Command Parameter dialog box is shown. It has a title bar with a close button. Inside, there are four input fields: 'Parameter Name', 'Prompting Text', 'Value Type' (a dropdown menu currently showing 'String'), and 'Default Value'. To the right of these fields are 'OK' and 'Cancel' buttons.

5. In the **Parameter Name** field, type End Date.

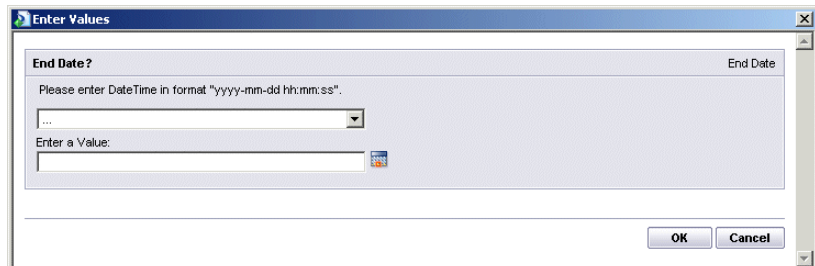
6. In the **Prompting Text** field, type End Date?.

7. From the **Value Type** list, select **DateTime**.

The program automatically enters the current date and time in the Default Value field.

8. Click **OK**, and click **OK** again.

The Enter Values window appears.

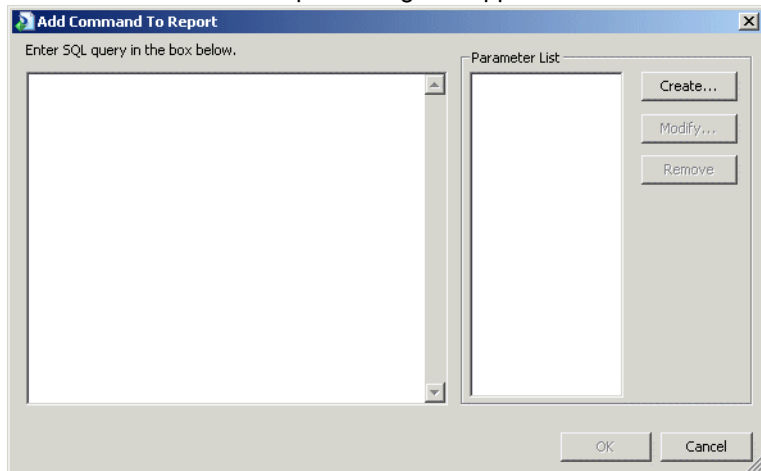


The Enter Values dialog box is shown. It has a title bar with a close button. Inside, there is a section titled 'End Date?' with a subtitle 'Please enter DateTime in format "yyyy-mm-dd hh:mm:ss"'. Below this is a dropdown menu showing '...' and an 'Enter a Value:' text box. To the right of the text box is a small calendar icon. At the bottom right are 'OK' and 'Cancel' buttons.

9. Provide a value for the End Date, and click **OK**.

10. In the Available Data Sources area, under the **Xtreme Sample Database 11.5** connection, select **Add Command**, and click **>**.

The Add Command to Report dialog box appears.

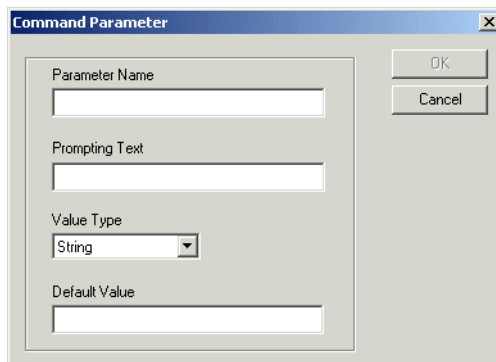


11. In the text area, type the following SQL query:

```
SELECT [Amount], [Account Number]
FROM [Monthly Account Budgets]
WHERE Year=?End Date) and Month=Month(?End
Date))
```

12. Click **Create**

The Command Parameter dialog box appears.



13. In the **Parameter Name** field, type End Date.

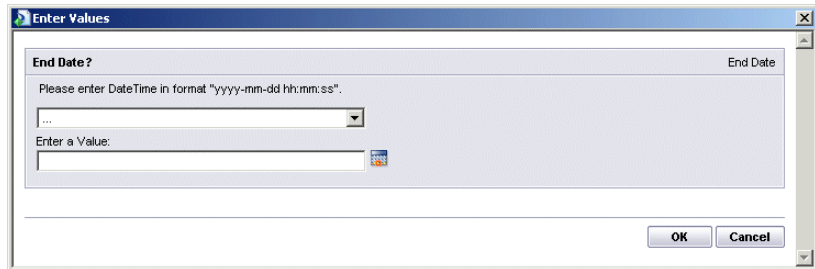
14. In the **Prompting Text** field, type End Date?.

15. From the **Value Type** list, select **DateTime**.

The program automatically enters the current date and time in the Default Value field.

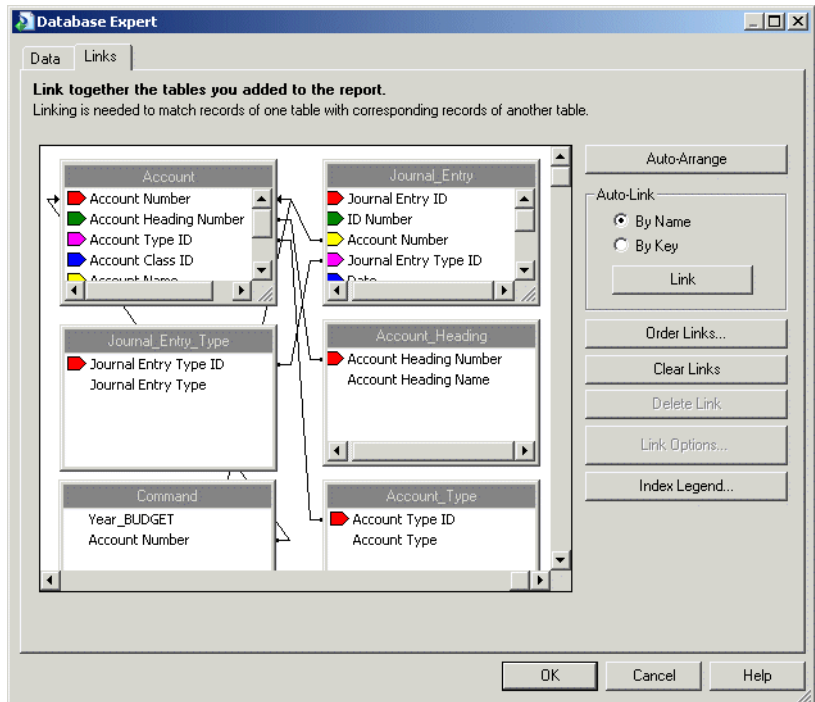
16. Click **OK**, and click **OK** again.

The Enter Values window appears.



17. Provide a value for the End Date, click **OK**, and click **OK** again.

The Links tab is displayed.



18. Verify that the links between the tables are consistent with what is displayed in the screenshot above. If necessary, create the appropriate links by dragging fields from one table to corresponding fields in another table.

Note: Ensure that the following links exist:

- Account.Account Type ID to Account Type.Account Type ID
- Account.Account Number to Journal Entry.Account Number
- Account.Account Heading Number to Account Heading.Account Heading Number
- Journal Entry.Journal Entry Type ID to Journal Entry Type.Journal Entry Type ID
- Command.Account Number to Account.Account Number
- Command_1.Account Account Number to Account.Account Number

19. Click **OK**.

Setting up the report date



When you created the command object, a new parameter field named End Date was also created. (Check the Parameter Fields in the Field Explorer to ensure that the field was made.)

Setting up selection criteria

A selection criteria restricts the range of journal entries that are included in the report. For this report, you need to restrict the journal entries to those that fall between January 1st of the End Date year (for YTD calculations) and the End Date parameter. For example, if the End Date parameter has a value of November 14, 2005, then the journal entries are restricted to those between January 1, 2005 and November 14, 2005.

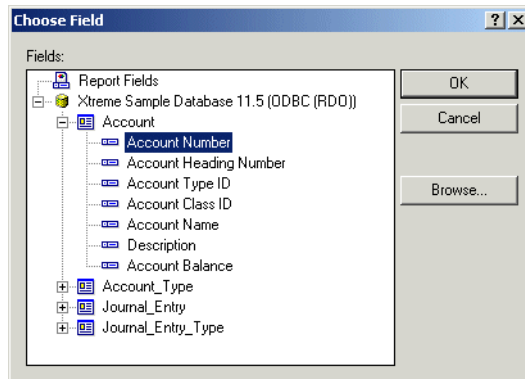
In addition to restricting journal entries based on dates, you need to restrict journal entries based on type. On January 1st of each year, the process of closing accounts is performed. These closing entries include the closure of all revenue and expense accounts and adjustments to Retained Earnings to reflect a net gain or net loss. Therefore, closing entries must not be included in the calculations; such entries must be restricted. You also need to restrict journal entries to display only Revenue and Expense accounts.



► To create selection criteria

1. Click **Select Expert**.

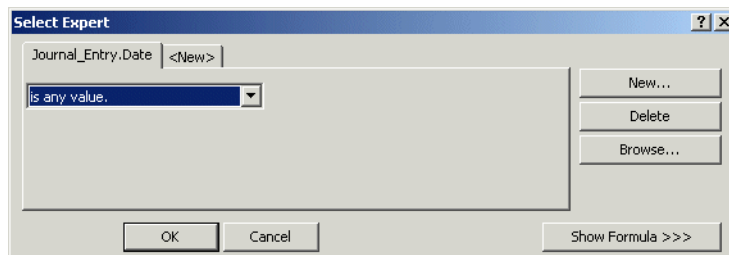
The Choose Field dialog box appears.



2. Expand the **Journal_Entry** table, and select **Date**.

3. Click **OK**.

The Select Expert dialog box appears.



4. Ensure that the **Journal_Entry.Date** tab is selected, and, from the list, choose **formula**:

5. To restrict journal entries, type the following formula in the text area:

```
{Journal_Entry.Date} <= {?End Date} and  
{Journal_Entry.Date} >= dateserial(year({?End Date}),1,1)
```

Note:

- **dateserial** is an SQL function that accepts three parameters (year, month, and day) and returns a valid date.
- **year** is an SQL function that accepts a datetime value and returns the year as an integer value.

6. Now that the journal entry dates are restricted by the formula, you need to add criteria to restrict the closing entries. In the Select Expert dialog box, click the <New> tab.
7. In the Choose Field dialog box, expand the **Journal_Entry_Type** table, and select **Journal Entry Type**.
8. Click **OK**.
9. Ensure that the Journal_Entry_Type.Journal Entry Type tab is selected, and from the list, choose **is not equal to**.
10. From the list that appears, select **closing entries**.
11. Click the <New> tab.
12. In the Choose Field dialog box, expand the **Account_Type** table, and select **Account Type**.
13. Click **OK**.
14. Ensure that the Account_Type.Account Type tab is selected, and, from the list, choose **is one of**.
15. From the list that appears, select **Revenue** and **Expense**.
16. Click **Show Formula >>>**.

The dialog box displays your selection criteria.

The screenshot shows the 'Select Expert' dialog box. The 'Journal_Entry_Type.Journal Entry Type' tab is selected, and the 'Account_Type.Account Type' tab is also visible. The 'is one of' dropdown is selected, and the list below it contains 'Expense' and 'Revenue'. The 'Add' button is visible. The 'New...' button is also visible. The 'OK', 'Cancel', and 'Help' buttons are at the bottom. The 'Hide Formula <<<' button is also visible. The 'Record Selection' radio button is selected, and the 'Group Selection' radio button is unselected. The 'Formula Editor...' button is visible. The formula text area contains the following text:

```
{Journal_Entry.Date} <= {?End Date} and
{Journal_Entry.Date} >= dateserial(year({?End Date}),1,1) and
{Journal_Entry_Type.Journal Entry Type} <> "closing entries" and
{Account_Type.Account Type} in ["Expense", "Revenue"]
```

17. Click **OK**.

Setting up groupings

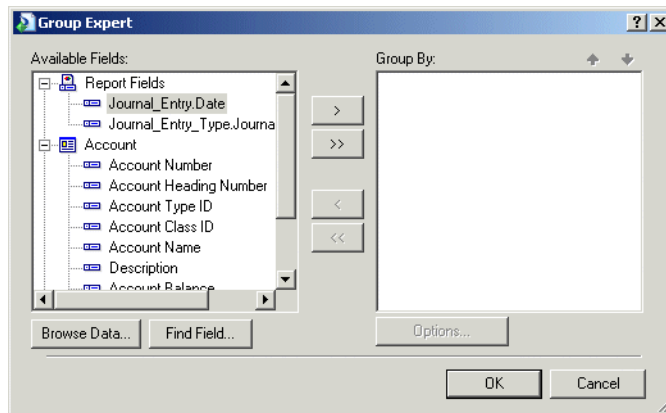
For this report, you must group records based on Account Type, Account Heading Name, and Account Name.

► **To create report groupings**



1. Click **Group Expert**.

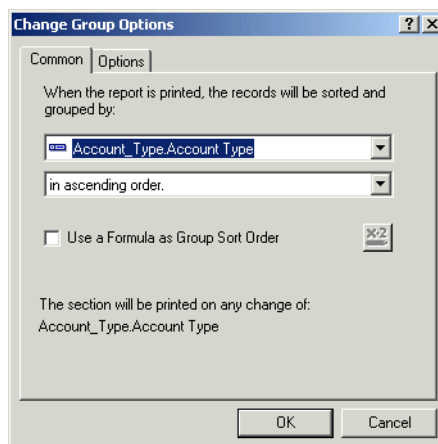
The Group Expert dialog box appears.



2. In the **Available Fields** area, expand the **Account_Type** table, select **Account Type**, and click >.

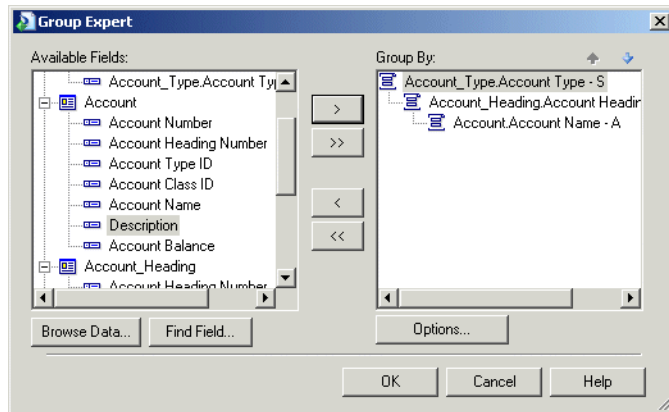
3. Click **Options....**

The Change Group Options dialog box appears.



4. From the second list, select in **specified order**.
5. From the Named Group: list, select the following values in order:
 - Revenue
 - Expense
6. Click **OK**.
7. Expand the **Account_Heading** table, select the **Account Heading Name** field, and click > to create a second group.
8. Expand the **Account** table, select the **Account Name** field, and click > to create a third group.

The dialog box displays your groupings.



9. Click **OK**.

Create formula fields

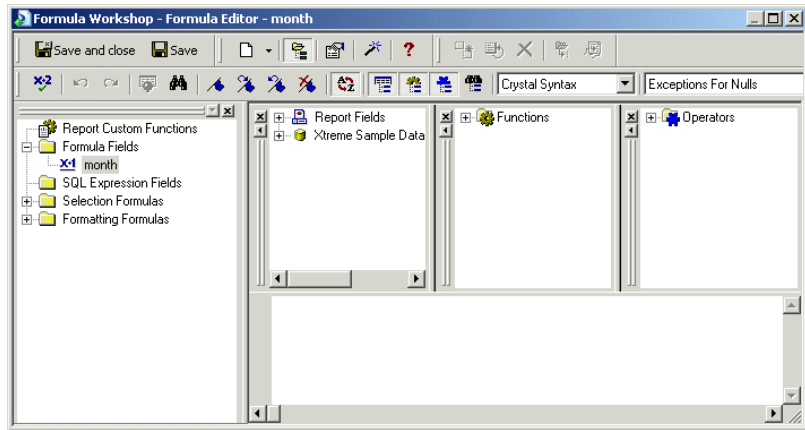
The Consolidated Income Statement uses several formula fields. You must create the necessary fields and add them to the report.

► To create formula fields



1. Click **Field Explorer**.
2. In the Field Explorer, right-click **Formula Fields**, and click **New**.
3. In the Formula Name dialog box, type `init`, and click **OK**.

The Formula Workshop appears.



4. Type the following formula:

```
whileprintingrecords;  
currencyvar month_budget_total :=0;  
currencyvar YTD_budget_total :=0;
```

This formula initializes the `month_budget_total` and the `YTD_budget_total` variables by setting them to '0'.



5. Click **Check** to verify that the syntax of the formula is correct.

6. Click **Save and close**.

7. Repeat steps 2 through 6 to create the following formula fields:

- **month:** This formula field displays the textual representation of the End Date month.
`monthname(month({?End Date}))`
- **month_balance:** This formula field reverses the sign of all the credit amounts for the reporting month.

```
if month({Journal_Entry.Date}) = month({?End Date})  
then  
(  
    if {Journal_Entry.Debit Or Credit} = 'Credit' then  
    (  
        {Journal_Entry.Amount} * -1  
    )  
    else  
    (  
        {Journal_Entry.Amount}  
    )  
)
```

- **month_account_variance:** This formula field compares the sum of the actual amounts versus the amounts that are budgeted for the month for each account.

```
if {Account_Type.Account Type} = 'Revenue' and
    {Account.Account Name} <> 'Sales Returns' then
(
    {Command.Amount} + Sum ({@month_balance},
    {Account.Account Name})
)
else
(
    {Command.Amount} - Sum ({@month_balance},
    {Account.Account Name})
)
```

- **month_budget:** This formula field displays the budgeted amount for the month for each account. It also maintains the running totals for each account type.

```
whileprintingrecords;
currencyvar month_budget_total;
if {Account.Account Name} = 'Sales Returns' then
(
    month_budget_total := month_budget_total -
    {Command.Amount}
)
else
(
    month_budget_total := month_budget_total +
    {Command.Amount}
);
{Command.Amount};
```

- **month_budget_net_income:** This formula field represents the month_budget_net_income variable.

```
whileprintingrecords;
currencyvar month_budget_net_income;
month_budget_net_income;
```

- **month_budget_type_total:** This formula field displays the total of the account budgets for a given account type for the month. It also maintains the running total for the budgeted net income, the value of which is stored in the month_budget_net_income variable.

```
whileprintingrecords;
currencyvar month_budget_net_income;
currencyvar month_budget_total;

if {Account_Type.Account Type} = 'Expense' then
(
    month_budget_net_income := month_budget_net_income
    - month_budget_total;
)
else
(
    month_budget_net_income := month_budget_net_income
    + month_budget_total;
);
month_budget_total;
```

- **month_net_income_variance:** This formula field adds the @month_budget_net_income field and the sum of the @month_budget_balance field.
- **month_type_total_variance:** This formula field adds the @month_budget_type_total field to the sum of the @month_balance field for revenue accounts. It also subtracts the @month_budget_type_total field from the sum of the @month_balance field for expense accounts.

```
if {Account_Type.Account Type} = 'Revenue' then
(
    {@month_budget_type_total} + Sum
    ({@month_balance}, {Account_Type.Account Type})
)
else
(
    {@month_budget_type_total} - Sum
    ({@month_balance}, {Account_Type.Account Type})
)
```


- type_total_heading:** This formula field sets heading to either Net Sales to Total Operating Expenses.


```
if GroupName ({Account_Type.Account Type}) =
  'Revenue' then
    "Net Sales"
  else
    "Total Operating Expenses"
```
- YTD_balance:** This formula field reverses the sign of all credit amounts for the reporting year.


```
if {Journal_Entry.Debit Or Credit} = 'Credit' then
(
  {Journal_Entry.Amount} * -1
)
else
(
  {Journal_Entry.Amount}
)
```
- YTD_budget:** This formula field displays the budgeted amount for the year for each account. It also maintains the running total for each account type.


```
whileprintingrecords;
currencyvar YTD_budget_total;
if {Account.Account Name} = 'Sales Returns' then
(
  YTD_budget_total := YTD_budget_total -
  {Command_1.Year_Budget};
)
else
(
  YTD_budget_total := YTD_budget_total +
  {Command_1.Year_Budget};
);
{Command_1.Year_Budget};
```
- YTD_budget_net_income:** This formula represents the YTD_budget_net_income variable.


```
whileprintingrecords;
currencyvar YTD_budget_net_income;
YTD_budget_net_income;
```

- **YTD_budget_type_total:** This formula field displays the total of the account budgets for a given account type for the year. It also maintains the running total for the budgeted net income.

```
whileprintingrecords;
currencyvar YTD_budget_net_income;
currencyvar YTD_budget_total;
if {Account_Type.Account Type} = 'Expense' then
(
    YTD_budget_net_income := YTD_budget_net_income -
    YTD_budget_total
)
else
(
    YTD_budget_net_income := YTD_budget_net_income +
    YTD_budget_total
);
YTD_budget_total;
```

- **YTD_net_income_variance:** This formula field adds the @YTD_budget_type_total field to the sum of @YTD_balance field for revenue accounts. It also subtracts the @YTD_budget_type_total field from the sum of @YTD_balance field for expense accounts.

```
{@YTD_budget_net_income} + Sum ({@YTD_balance})
```

- **YTD_type_total_variance:** This formula field adds the @YTD_budget_type_total field to the sum of @YTD_balance field for revenue accounts. It also subtracts the @YTD_budget_type_total field from the sum of @YTD_balance field for Expense accounts.

```
if {Account_Type.Account Type} = 'Revenue' then
(
    {@YTD_budget_type_total} + Sum ({@YTD_balance},
    {Account_Type.Account Type})
)
else
(
    {@YTD_budget_type_total} - Sum ({@YTD_balance},
    {Account_Type.Account Type})
)
```

- **YTD_account_variance:** This formula field compares the sum of the actual amounts versus the budgeted amounts for the year for each account.

```
if {Account_Type.Account Type} = 'Revenue' and
   {Account.Account Name} <> 'Sales Returns' then
(
   {Command_1.Year_Budget} + Sum ({@YTD_balance},
   {Account.Account Name})
)
else
(
   {Command_1.Year_Budget} - Sum ({@YTD_balance},
   {Account.Account Name})
)
```

Adding fields to the report

After you create the formula fields, you can begin to add fields to the report.

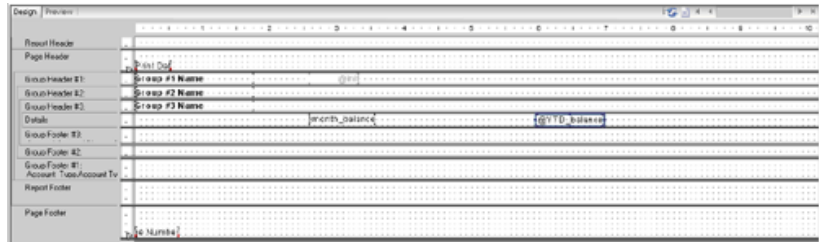
Note: Before you add the fields, adjust the page setup of the report to use legal size paper with a landscape orientation. This arrangement is required to accommodate the fields in the report.

► To add fields to the report

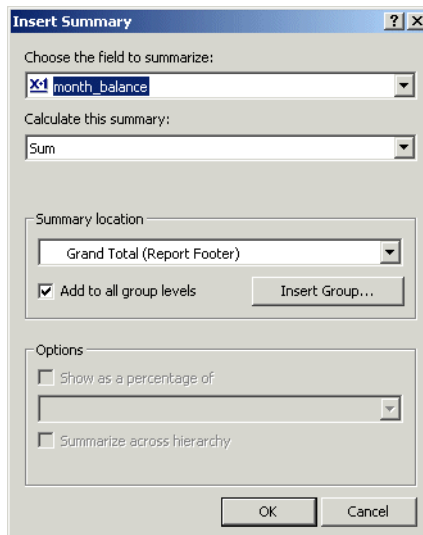
1. From the Field Explorer, drag the `YTD_balance` field to the **Details** section, and align the right edge of the field with the 8 inch mark.
2. Drag the `month_balance` field to the **Details** section, and align the right edge of the field with the 4.75 inch mark
3. Drag the `init` field to the **Group Header #1** section, and align it with the `month_balance` field.
4. Right-click the `init` field, and click **Suppress (No Drill-Down)**.

5. Right-click all of the headings that were automatically created in the **Page Header** section, and click **Delete**.

The report should now look similar to the following screenshot:

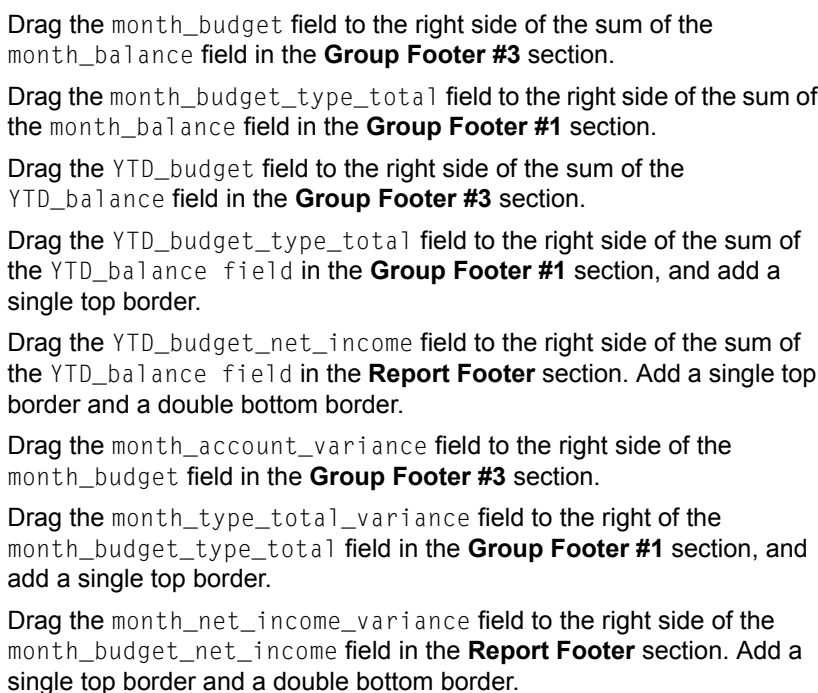


6. Right-click the month_balance field, and choose **Insert > Summary**. The Insert Summary dialog box appears.



7. For the **Summary Location**, select Group #1, and click **OK**.
8. Apply a single top border to the Group #1 summary field.
9. Right-click the month_balance field, and choose **Insert > Summary**.
10. For the **Summary Location**, select Group #3, and click **OK**.
11. Right-click the month_balance field, and choose **Insert > Summary**.
12. For the **Summary Location**, select Grand Total (Report Footer), and click **OK**.

- The report should now look similar to the following screenshot:



The report should now look similar to the following screenshot:

The screenshot shows the 'Manage Budgets' app in SAP Fiori. The table displays budget data for Group 03. The columns are: month, balance, and variance. The data is filtered for Group 03 and shows values for month 03, 04, and 05.

month	balance	variance
03	1000000	1000000
04	1000000	1000000
05	1000000	1000000

5. Create a text field in the **Report Footer** section, and align it horizontally with Group #1 Name. In this text field, type "Net Income".

The report should now look similar to the following screenshot:

Reviewing your work



To see how the report looks with the fields in place, click **Print Preview** to activate the Preview tab.

The report appears in preview mode.

	January			YTD		
	Actual	Budget	Variance	Actual	Budget	Variance
Revenue						
Sales Revenue						
Bike Sales - Competition	\$187,302.75	\$193,180.48	\$5,867.71	\$187,302.75	\$193,180.48	\$5,867.71
Bike Sales - Hybrid	\$15,764.61	\$1,367.66	(\$14,376.76)	\$15,764.61	\$1,367.66	(\$14,376.76)
Bike Sales - Kids	\$3,547.69	\$4,305.30	\$637.59	\$3,547.69	\$4,305.30	\$637.59
Bike Sales - Mountain	\$40,896.24	\$42,778.62	\$1,922.58	\$40,896.24	\$42,778.62	\$1,922.58
Sales Gloves	\$765.98	\$656.37	(\$109.61)	\$765.98	\$656.37	(\$109.61)
Sales Helmets	\$3,506.74	\$4,936.60	\$1,429.86	\$3,506.74	\$4,936.60	\$1,429.86
Sales Locks	\$741.25	\$1,353.32	\$652.06	\$741.25	\$1,353.32	\$652.06
Sales Returns	\$14,536.10	\$1,113.79	(\$13,222.31)	\$14,536.10	\$1,113.79	(\$13,222.31)
Sales Saddles	\$1,088.65	\$1,069.96	(\$18.70)	\$1,088.65	\$1,069.96	(\$18.70)
Net Sales	\$239,237.83	\$248,654.86	\$9,417.03	\$239,237.83	\$248,654.86	\$9,417.03
Expense						
Cost of Goods Sold						
Bikes (Competition) Cost	\$86,043.00	\$60,730.77	(\$25,305.03)	\$86,043.00	\$60,730.77	(\$25,305.03)
Bikes (Hybrid) Cost	\$7,166.95	\$9,949.37	\$2,782.42	\$7,166.95	\$9,949.37	\$2,782.42
Bikes (Kids) Cost	\$1,621.62	\$2,308.22	\$686.40	\$1,621.62	\$2,308.22	\$686.40

Note: Revenue accounts that are also credit accounts may display negative amounts. To display positive values, you need to apply conditional formatting on the sum fields for both the month_balance and the YTD_balance fields.

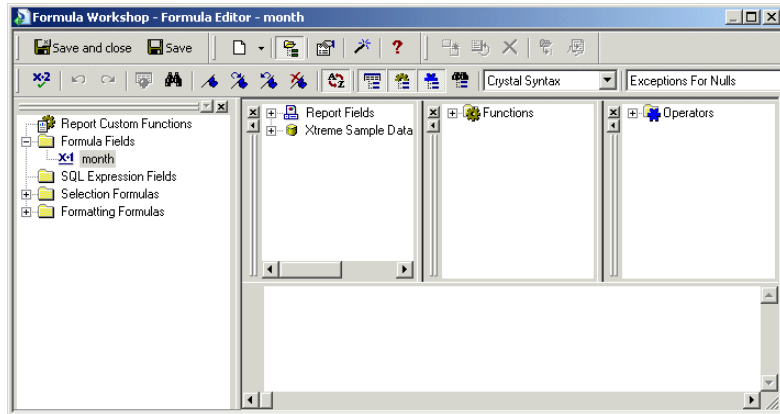
► To apply conditional formatting

1. In the **Group Footer #3** section, right-click sum of the month_balance field, and select **Format Field**.
2. On the Number tab, click **Customize....**



3. Select the **Reverse Sign for Display** option, and click the **Formula** button beside the option.

The Formula Workshop appears.



4. Type the following formula, and then click **Save and close**:
`{Account_Type.Account Type} = 'Revenue' and
{Account.Account Name} <> 'Sales Returns'`
5. In the **Group Footer #1** section, right-click the sum of the `month_balance` field, and select **Format Field**.
6. On the Number tab, click **Customize....**
7. Select the **Reverse Sign for Display** option, and click the **Formula** button beside the option.
8. In the Formula Workshop, type the following formula, and click **Save and close**:
`{Account_Type.Account Type} = 'Revenue'`
9. In the **Report Footer** section, right-click the sum of the `month_balance` field, and select **Format Field**.
10. On the Number tab, click **Customize....**
11. Select the **Reverse Sign for Display** option, and click **OK**.
12. Repeat steps 1 through 11 to apply the same formatting to all of the other columns.

After you finish reviewing the report, click the Design tab to correct any mistakes.

Completing the report

Now that the report has been built to display the correct data, you can add the report title, the report date, and the company logo.

► **To add a report title and report date**


1. Expand the **Page Header** section to be approximately 2 inches in height.
2. Create a new text field at the center of the **Page Header** section, and type `Xtreme Mountain Bikes`.
3. Create another text field directly underneath the `Xtreme Mountain Bikes` text field, and type `Consolidated Income Statement`.
4. Create another text field directly underneath the `Consolidated Income Statement` text field, and type `For the months ending`.
5. Drag the **End Date** parameter field to anywhere in the **Page Header** section.
6. Right-click the **End Date** field, and click **Format Field**.
The Format Editor appears.
7. Select **03/01/1999** as the Date and Time format, and click **OK**.
8. Drag the End Date field into the text field that contains **For the months ending**.



► **To add a company logo to the report**

1. Click **Insert Picture**.
2. Choose a bitmap (.bmp) logo file, and click **Open**.
3. Position the object frame in the upper right-hand corner of the **Report Header (RH)** section of the report.
4. Click the Preview tab.

The report should now look similar to the following screenshot:

<div>  </div>						
<div> Xtreme Mountain Bikes Consolidated Income Statement For the Month Ending 01/31/2005 </div>						
	January			YTD		
	Actual	Budget	Variance	Actual	Budget	Variance
Revenue						
Sales Revenue						
Bike Sales - Competition	\$187,302.76	\$193,100.46	\$5,867.71	\$187,302.76	\$193,100.46	\$5,867.71
Bike Sales - Hybrid	\$15,764.61	\$1,307.06	(\$14,376.76)	\$15,764.61	\$1,307.06	(\$14,376.76)
Bike Sales - Kids	\$3,547.69	\$4,385.28	\$837.59	\$3,547.69	\$4,385.28	\$837.59
Bike Sales - Mountain	\$40,996.24	\$42,770.82	\$1,722.58	\$40,996.24	\$42,770.82	\$1,722.58
Sales Gloves	\$765.98	\$656.37	(\$109.61)	\$765.98	\$656.37	(\$109.61)
Sales Helmets	\$3,536.74	\$4,936.60	\$1,429.86	\$3,536.74	\$4,936.60	\$1,429.86
Sales Locks	\$741.26	\$1,393.32	\$652.06	\$741.26	\$1,393.32	\$652.06
Sales Returns	\$14,236.10	\$1,113.79	(\$13,222.31)	\$14,236.10	\$1,113.79	(\$13,222.31)
Sales Saddles	\$1,006.66	\$1,006.66	(\$10.70)	\$1,006.66	\$1,006.66	(\$10.70)
Net Sales	\$279,237.83	\$248,654.86	\$9,417.03	\$279,237.83	\$248,654.86	\$9,417.03
Expense						
Cost of Goods Sold						
Bikes (Competition) Cost	\$86,043.80	\$50,736.77	(\$35,305.03)	\$86,043.80	\$50,736.77	(\$35,305.03)
Bikes (Hybrid) Cost	\$7,166.95	\$9,949.37	\$2,782.42	\$7,166.95	\$9,949.37	\$2,782.42
Bikes (Kids) Cost	\$1,621.82	\$2,306.22	\$686.40	\$1,621.82	\$2,306.22	\$686.40
Bikes (Mountain) Cost	\$18,719.30	\$27,731.69	\$9,012.39	\$18,719.30	\$27,731.69	\$9,012.39

Saving the report



► To save the report

1. Click **Save**.

Because this is the first time that you are saving the report, the Save As dialog box appears and displays the location where the file will be saved.

2. In the **File name** field, type Consolidated Income Statement.rpt, and click **Save**.

The report is saved to the location that you chose.