Parameter Fields and Prompts
Parameter and prompt overview

Parameters are Crystal Reports fields that you can use in a Crystal Reports formula. As a formula component, a parameter must have a value before the program can process the report. By using parameters in formulas, selection formulas, and in the report itself, you can create a single report that changes its behavior depending on the values entered by your users. Parameter fields can also be used in subreports.

Prompts are elements that help users set a value for report parameters. Prompts are different from parameters in several ways:

- Prompts are not used directly by a Crystal Reports formula.
- Prompts include user-interface settings that help you determine the appearance of the prompting dialog box that your users see.
- Prompts include an optional list of values that your users can choose from. This list of values can be a static list that is stored in each report, or a dynamic list that is retrieved from a database (this type of prompt includes multi-level cascading lists that are also retrieved from a database).

When your users select values in the prompting dialog box, they are setting values for prompts. The Crystal Reports prompting engine then assigns that value to the corresponding parameter, which is finally used by the report.

When you create and edit a parameter field, you work with one parameter and one or more prompts.

Parameter field considerations

There are a number of things to keep in mind when working with parameter fields:

- Parameter fields support the following data types:
  - Boolean: Requires a yes/no or true/false answer.
    Example: Include planned budget numbers in the summary?
  - Currency: Requires a dollar amount.
    Example: Display customers with sales over XXXXX.
  - Date: Requires an answer in a date format.
    Example: Enter the start and end dates of the quarter.
  - DateTime: Requires both date and time.
    Example: Display statistics for 07/04/1999 between 1:00pm-2:00pm.
- **Number:** Requires a numeric value.
  Example: Enter the customer identification number.
- **String:** Requires a text answer.
  Example: Enter the region.
- **Time:** Requires an answer using a time format
  Example: Display the total number of calls from 1:00pm-2:00pm.
- Parameter field prompting text can be up to four lines long with approximately 60-70 characters per line (depending on character width, up to the 254 character limit). Text over one line in length will automatically word wrap.
- You can create a list of values from which users can choose the parameter value rather than having them enter it manually. Lists of values can be static (embedded in each report), dynamic (refreshed from a database on demand), scheduled (refreshed from a database on a recurring schedule), or partially scheduled (portions of a list of values can be scheduled, while other portions remain on-demand). Scheduled and partially scheduled lists of values are administered through the Business View Manager. For more information, see the *Business Views Administrator’s Guide*.
- A parameter field does not have to be placed in a report in order to be used in a record or group selection formula. You create the parameter field and then enter it in your formula as you would any other field.

**Prompt considerations**

There are a number of things to keep in mind when working with prompts:

- Prompts can be static or dynamic. As well, a dynamic prompt can have a cascading list of values. For a description of each option, see these topics:
  - “Creating a parameter with a static prompt” on page 446.
  - “Creating a parameter with a dynamic prompt” on page 449.
  - “Creating a parameter with a cascading list of values” on page 451.
- Dynamic prompts contain lists of values that you can use for a particular report or for many reports. You share a list of values by adding it to the BusinessObjects Enterprise Repository.
- A list of values can be scheduled for automatic updates through the Business View Manager. For more information about scheduling a list of values, see the *Business Views Administrator’s Guide*.
A list of values can form a cascading prompt; that is, you can create a prompt whose final value is determined through a sequence of choices. For example, users might first be prompted to pick a country before the choices for region appear. The users might then need to pick a region before the choices for city appear, and so on.

### Understanding dynamic prompts

Both dynamic prompts and cascading lists of values are available in Crystal Reports. These features let you populate the lists of values associated with a prompt from a data source outside of your report. (Static lists of values that are stored in your report are also available in Crystal Reports.)

This dynamic capability is available to all Crystal Reports users—whether or not they have the added capabilities of BusinessObjects Enterprise. When Crystal report files are stored in BusinessObjects Enterprise, additional dynamic prompting features are available. The following table shows which features are available with each product.

**Note:** In the table, and elsewhere in this section, the term *managed report* is used to describe reports that are stored in a BusinessObjects Enterprise environment, while *unmanaged report* is used to describe reports that are stored outside such an environment.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Available when Crystal reports are stored outside of BusinessObjects Enterprise?</th>
<th>Available when Crystal reports are published to BusinessObjects Enterprise?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create dynamic prompts and cascading lists of values.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Reuse list of value definitions within a single report at design time.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Reuse list of value definitions within multiple reports at design time.</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Schedule lists of values.</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>
### Some components in the Crystal Reports and BusinessObjects Enterprise products support the execution of reports with dynamic prompts and cascading lists of values:

- All viewers except for the Java Applet viewer.

  **Note**: Reports that use dynamic prompts and cascading lists of values run in the Java Applet viewer; they will not, however, have pick lists where dynamic lists of values should appear.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Available when Crystal reports are stored outside of BusinessObjects Enterprise?</th>
<th>Available when Crystal reports are published to BusinessObjects Enterprise?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Populate lists of values from command objects.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Populate lists of values from report fields.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Populate list of values from Business Views.</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Cache lists of values at report run time, and share that list among multiple executing reports.</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Schedule lists of values to update themselves on a regular basis.</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Schedule portions of lists of values to update themselves on a regular basis, while the remaining portions retrieve their values from the database on demand.</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>
The supported viewers are as follows:

- ActiveX
- .NET Winform
- .NET Webform
- Java, COM, and JSF DHTML page viewers
- Java and COM Advanced DHTML Viewer
- The InfoView and Central Management Console scheduling interfaces in BusinessObjects Enterprise.
- The Report Designer Component (RDC).
- All editions of Crystal Reports 11.

These components support the design of reports with dynamic prompts and cascading lists of values.

- The integrated Visual Studio .NET designer.
- Business View Manager.

**Note:** While the dynamic capability is not available with Business Objects Web Intelligence or with full-client products, Crystal reports that are based on universes can use the dynamic prompting capabilities described throughout this section.

## Understanding lists of values

List-of-values objects describe how to return a set of values from a data source:

- They provide the values for prompts in the prompting dialog box.
- They can be either static (the values are stored in the report), or dynamic (the values are stored outside the report in a database).
- They are used for both single-level dynamic prompts, and multi-level cascading prompts.

For example, a list of countries could be a dynamic prompt, while a hierarchical list of countries, regions, and cities could be a dynamic prompt with a cascading list of values.

Crystal Reports supports three data sources for list-of-values objects:

- Report fields (only for unmanaged reports).
- Command objects (only for unmanaged reports).
- Business Views (only for managed reports).

A list of values can have one or more levels. If there are multiple levels in the list of values, then each level constrains the next.
You can think of a list of values as the definition of the data needed to populate all levels of a cascading list. In Crystal Reports, a cascading relationship is defined by a single list-of-values object, not by multiple queries that are linked together by a common key. Although the list of values is defined as a single entity, data is not necessarily fetched from the data source in a single query.

**Note:** List-of-values objects are not affected by report record selection or group selection formulas. These formulas affect the report data, not data used for prompt-time lists of values.

### List-of-values types

There are two different types of list-of-values objects:

- **Unmanaged lists of values** are stored within each report file.
  
  If you do not have BusinessObjects Enterprise, or if you never publish your reports to BusinessObjects Enterprise, you use an unmanaged list of values. Unmanaged list-of-values objects can use report fields or command objects as a data source.

- **Managed lists of values** are stored within BusinessObjects Enterprise.
  
  Every report that you store in BusinessObjects Enterprise uses managed list-of-values objects. All managed list-of-values objects are based on a Business View, even if the report itself does not use a Business View. Managed list-of-values objects have a number of features that are not available to unmanaged list-of-values objects.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Unmanaged list of values</th>
<th>Managed list of values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database query per list-of-value level.</td>
<td>Yes The database is queried for each level as the user selects values in the prompting dialog box.</td>
<td>No The database is queried for all values in all levels of the list at one time.</td>
</tr>
<tr>
<td>At design time, specify a filter for the list of values.</td>
<td>No All of the values in the report fields that are also in the list of values definition are used.</td>
<td>Yes Any filter that can be represented in SQL can be used.</td>
</tr>
<tr>
<td>Feature</td>
<td>Unmanaged list of values</td>
<td>Managed list of values</td>
</tr>
<tr>
<td>--------------------------------------------------</td>
<td>--------------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>Report fields</td>
<td></td>
<td></td>
</tr>
<tr>
<td>At report-view time, prompt for additional</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>information needed by the list of values.</td>
<td></td>
<td>If the command object</td>
</tr>
<tr>
<td></td>
<td></td>
<td>contains parameters,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>the value(s) are</td>
</tr>
<tr>
<td></td>
<td></td>
<td>prompted for at report-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>view time.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Display different values to different users.</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Because managed lists</td>
</tr>
<tr>
<td></td>
<td></td>
<td>of values are based on</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Business Views, they</td>
</tr>
<tr>
<td></td>
<td></td>
<td>inherit the view-time</td>
</tr>
<tr>
<td></td>
<td></td>
<td>security capabilities</td>
</tr>
<tr>
<td></td>
<td></td>
<td>of Business Views.</td>
</tr>
<tr>
<td>Schedule the list of values to update on a</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>recurring schedule.</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The scheduling capability</td>
</tr>
<tr>
<td></td>
<td></td>
<td>comes from the Business</td>
</tr>
<tr>
<td></td>
<td></td>
<td>View Manager, not the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Central Management</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Console (CMC) where</td>
</tr>
<tr>
<td></td>
<td></td>
<td>reports are scheduled.</td>
</tr>
<tr>
<td>Partially schedule the list of values to update</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>only certain portions of the list on a recurring</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>schedule.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Determining which list-of-values type to use

Different reporting problems require different prompting solutions, depending on the amount of data in the lists of values. This table provides a rough guide to which list-of-values design you should use.

<table>
<thead>
<tr>
<th>Description</th>
<th>Unmanaged list of values</th>
<th>Command objects</th>
<th>Managed list of values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single-level code tables. (A table that has hundreds of semi-static values in a single level.)</td>
<td>Not well suited. Because report fields cannot be filtered, any filtering of the list that you require must be done outside of Crystal Reports in a database view.</td>
<td>Well suited. You can define filtering within a command object. (That command objects return the data for all levels in the list is not a problem for single-level lists.)</td>
<td>Well suited. You can define filtering within the source Business View.</td>
</tr>
<tr>
<td>Multi level-cascading code tables. (A table that has hundreds of semi-static values in multiple levels.)</td>
<td>Well suited. Provided that the data filtering is done outside of Crystal Reports, report fields are well suited to multi-level cascading code tables because they retrieve data one level at a time.</td>
<td>Well suited. Provided that the total data volume for all levels in the list is not too large. If there is too much data, you could encounter performance issues because command-object-based lists of values retrieve data for all levels in one trip to the database.</td>
<td>Well suited. Business-View-based lists of values can support both filtering and level-by-level data retrieval.</td>
</tr>
<tr>
<td>Fact tables. (These tables tend to be very large, dynamic tables with millions of values in multiple levels.)</td>
<td>Well suited. Provided that the filtering is done outside of Crystal Reports in a database view, and provided that there is a multi-level hierarchy to the data.</td>
<td>Not well suited. Because command objects retrieve their data in one trip to the database, there could be performance issues when you use them against very large tables.</td>
<td>Well suited. Partially scheduled lists of values excel in this situation, where the semi-static part of the data can be scheduled, and the most dynamic part can be retrieved on-demand.</td>
</tr>
</tbody>
</table>
Lists of values and prompt groups contrasted

Lists of values are the data part of a prompt; the values from your data that your users will see and select from.

Prompt groups, on the other hand, are the presentation part of a prompt. Crystal Reports treats prompt groups as separate objects so that you can share the same list of values with different presentations. For example, you can have a Shipping City prompt, and a Customer City prompt. Perhaps you allow for multiple customer cities, but only a single shipping city in your report. You can design this report so that it uses a single city list of values, but with two different prompt groups (or presentations styles).

For more information about using prompt groups, see “Sharing common lists of values within a report” on page 453.

Creating a parameter with a static prompt

A static prompt is one that always contains the same values. For example, if your parameter prompts for a country value, you could create a static prompt because the country list represents a set of values that does not change often.

Use the following steps to create a parameter with a static prompt that lets users see a list of customers that they can select a specific country from.

This procedure is made up of two sets of steps. The first is creating the parameter, and the second is using the Select Expert to incorporate the parameter.

To create a parameter with a static prompt

1. Open the sample report called Group.rpt.

   Sample reports are located in the Crystal Reports directory under \Samples\En\Reports.

2. On the View menu, click Field Explorer.

   Tip: Another way to do this is to click the Field Explorer button on the Standard Toolbar.

   The Field Explorer dialog box appears.


   The Create New Parameter dialog box appears.

4. Enter a name for the parameter in the Name field (up to 255 alphanumeric characters).

   This example uses Country.
5. Select the appropriate **Type** from the list.
   This example uses String.
   
   **Note:** When creating a parameter whose Type is either Date or DateTime, you can change the date format to suit your needs. For details, see “Changing your default field formats” on page 262.

6. From the **Value Field** list, select **Country**.

7. Click **Actions** and select **Append all database values** to move all of the countries in the sample database to the **Values** area.
   
   This example will enable the user to choose from any of the countries. If you want to limit the selection, manually enter only the countries that you would like the user to choose from.

8. In the Options area, enter the desired prompting text in the **Prompt Text** field (up to 255 alphanumeric characters).
   
   This is the text that appears in the “Enter prompt values” dialog box when the report is refreshed. This example uses the default value.

9. Click **OK**.

10. Return to the Field Explorer dialog box, and drag the **Country** parameter into your report.
Parameter Fields and Prompts

Creating a parameter with a static prompt

**Note:** If you don’t want to see the parameter field you dropped in your report, place it in a section you can suppress, such as a report header or footer.

▶ **To incorporate the parameter into the record selection**

1. On the **Report** menu, click **Select Expert**.

   **Tip:** Another way to do this is to click the Select Expert button on the Expert Tools toolbar. The Choose Field dialog box appears.

2. Select **Country** from the Customer table, then click **OK**. The Select Expert appears.

3. Choose **is equal to** from the drop-down list.

4. Choose the parameter from the adjacent drop-down list.

   This example uses {?Country}.

5. Click **OK**.

   The “Enter prompt values” dialog box appears.

6. Select the country to base the report on.

   This example uses Brazil.

7. Click **OK**.
Creating a parameter with a dynamic prompt

A dynamic prompt is one that changes on a regular or irregular schedule. For example, if you are prompting for customer name, you could create a dynamic prompt because the names in your customer database probably change frequently. With a dynamic prompt, you are able to create or specify a list of values that you can schedule for updates as often as you like through the Business View Manager. For more information about scheduling a list of values, see the Business Views Administrator’s Guide.

Use the following steps to create a parameter that uses a dynamic prompt. As part of the procedure, you will create a list of values.

1. Open the sample report called Group.rpt. Sample reports are located in the Crystal Reports directory under \Samples\En\Reports.

2. On the View menu, click Field Explorer. The Field Explorer dialog box appears.


4. Enter a name for the parameter in the Name field (up to 255 alphanumeric characters). This example uses Region.

5. In the List of Values area, click Dynamic.

6. Ensure that New is selected in the Choose a Data Source area. Note: If your report already contains a list of values, or if you want to pick a list of values from your repository, you can select Existing and choose an existing list of values from the tree structure.

7. Click Insert, and from the Value list, select Region.

8. In the Options area, enter the desired prompting text in the Prompt Text field (up to 255 alphanumeric characters).
This is the text that appears in the “Enter prompt values” dialog box when the report is refreshed. This example uses “Select a Region.”

9. Click OK.
10. Return to the Field Explorer dialog box, and drag the Region parameter into your report.

Note: If you don’t want to see the parameter field you dropped in your report, place it in a section you can suppress, such as a report header or footer.

When you look at your dynamic prompt within Crystal Reports, it does not seem to be much different from a static prompt. In the background, however, Crystal Reports stores the information about the list of values that you created for this prompt. When you publish the report to BusinessObjects Enterprise, the list of values becomes a separate entity that you can schedule in the Business View Manager to ensure that the values are updated on a schedule that you choose. For example, if you knew that your sales force was rapidly expanding the regions that your company sells to, you might schedule your region list of values to be updated every night. Users who run a report that contains the region list of values will see the new values without you having to manually update the list (as you would have to do with a static prompt).
Creating a parameter with a cascading list of values

A dynamic prompt can include a cascading list of values, which adds additional levels of selection for your users. For example, if you are prompting for a city value, but you also need to know which country and region that city comes from, you could create a dynamic and cascading prompt. In this case, you first prompt for a country, and when that value has been selected, the program prompts for a region by showing only the regions that apply to the selected country. Finally, when a region value has been selected, the program prompts for a city by showing only the cities that apply to the selected region. In this way, you can be sure that your user picks the correct city (for instance, Vancouver, Washington, USA rather than Vancouver, British Columbia, Canada).

As with a dynamic prompt that has only one value, you are able to create or specify a cascading list of values that you can schedule for updates as often as you like through the Business View Manager. For more information about scheduling a list of values, see the Business Views Administrator’s Guide.

Use the following steps to create a parameter that uses a dynamic prompt and a cascading list of values.

► To create a parameter with a cascading list of values

1. Open the sample report called Group.rpt.
   Sample reports are located in the Crystal Reports directory under \Samples\En\Reports.

2. On the View menu, click Field Explorer.
   The Field Explorer dialog box appears.

   The Create New Parameter dialog box appears.

4. Enter a name for the parameter in the Name field (up to 255 alphanumeric characters).
   This example uses Supplier City.

5. In the List of Values area, click Dynamic.

6. Enter prompting text for your prompt group in the Prompt Group Text field.
   This example uses “Choose the City that your supplier is located in.”

   Note: Prompt groups let you use cascading list of values for other cascading prompts. For example, if you set up such a list of values for an address prompt, you might want to use the list for both shipping and billing addresses. In this case, you could create two prompt groups for the same list of values; you might call one prompt group Shipping Address and the other Billing Address.
7. Ensure that **New** is selected in the Choose a Data Source area.

   **Note:** If your report already contains a list of values, or if you want to pick a list of values from your repository, you can select Existing and choose an existing list of values from the tree structure.

8. Click **Insert**.

   The program automatically expands the Value list. You use this area to define the fields that make up your cascading list of values. This example uses a cascade of Country, Region, and City.

9. From the **Value** list, select **Country**.

10. Click the blank field under **Country** and select **Region**.

11. Click the blank field under **Region** and select **City**.

12. In the Parameter Binding area, click **Country** and **Region** to clear the binding (only the field that is associated with the **City** value should be bound).

   In this step, you are choosing the field in your cascading hierarchy that should be bound to the parameter field that you are creating. Because this particular parameter is intended to provide Crystal Reports with a city value, you bind the parameter to City.

13. In the **Prompt Text** field (in the Options area), enter additional prompting text for each of the fields in your cascading list of values.

   This is the text that appears in the “Enter prompt values” dialog box when the report is refreshed.
14. Click **OK**.

15. Return to the Field Explorer dialog box, and drag the **Supplier City** parameter into your report.

**Note:** If you don’t want to see the parameter field you dropped in your report, place it in a section you can suppress, such as a report header or footer.

---

**Working with lists of values**

Dynamic prompts use lists of values. You can create a list of values in either Crystal Reports or in the Business View Manager. Crystal Reports refreshes the data in a list of values when you open a report that contains a dynamic prompt; however, you can also refresh the data in a list of values by scheduling it in the Business View Manager. For more information, see the *Business Views Administrator's Guide*.

You can add lists of values to BusinessObjects Enterprise in several ways:

- You can create a list of values when you design a report, and then you can save the report to an Enterprise folder. For more information, see “Saving a report to an Enterprise folder” on page 396.
- You can create a list of values in the Business View Manager. For more information, see the *Business Views Administrator's Guide*.
- You can create a list of values when you design a report, and then you can add it to BusinessObjects Enterprise through the Central Management Console (CMC) or the Publishing Wizard. For more information, see the *BusinessObjects Enterprise Administrator's Guide*.
- You can migrate a list of values from one environment to another by using the Import Wizard. For more information, see the *BusinessObjects Enterprise Administrator's Guide*.

---

**Sharing common lists of values within a report**

You can use a single list-of-values object for many unrelated prompts within a single report. For example, in a report that shows shipments from suppliers to customers, shipments can go from a supplier in one city to a customer in another city. You would want your report to prompt for both supplier city and customer city; both have the same set of values. Because there are many hundreds of cities in the database, you can break up this long list of values into countries, regions, and cities. In that way, your users can navigate through hundreds of values by selecting from three shorter lists.
To share a common list of values within a report

1. Create a report that contains fields for Country, Region, and City.
2. In the Field Explorer, select Parameter Fields and click New.
   The Create New Parameter dialog box appears.
3. Enter a name for the parameter in the Name field.
   This example uses SupplierCity.
4. In the List of Values area, click Dynamic.
5. Enter prompting text for your prompt group in the Prompt Group Text field.
   This example uses “Choose the City that your supplier is located in.”
6. Click Insert, and from the Value list, select Country.
7. Click the blank field under Country and select Region, and then click the blank field under Region and select City.
8. In the Parameter Binding area, ensure that only the City value is bound.
9. Click OK.
10. Return to the Field Explorer dialog box, select Parameter Fields, and click New.
11. Enter a name for the second parameter in the Name field.
    This example uses CustomerCity.
12. In the List of Values area, click Dynamic.
13. Enter prompting text for your new prompt group in the Prompt Group Text field.
    This example uses “Choose the City that your customer is located in.”
14. Select Existing, and select the list of values that you created in steps 6 through 9.
15. Click OK.

You have created two prompts: a Country, Region, City hierarchy for the {?SupplierCity} parameter, and a second Country, Region, City hierarchy for the {?CustomerCity} parameter. By reusing the list-of-values object, you saved time and improved performance time in Crystal Reports.

Note: You can also use an existing list of values if you have another parameter field that should have a country value. For example, if you add a {?DivisionCountry} parameter to indicate the corporate division that you want to report on, you could use the same list-of-values object in your report. You would bind this parameter to the top-level value in your hierarchy—that is, the Country level of the Country, Region, City hierarchy.
Using separate value and description fields

It is common in relational databases to make use of code fields that represent values. These codes are often numeric or text strings that cannot be read by your users. For such cases, you can create separate value and description fields in your list of values definition. You set the value field to the parameter; the description field appears in the prompting dialog box. How the description field appears in the prompting dialog box is controlled by the prompt option “Prompt with Description Only.” When set to True, only the descriptions are visible; when set to False, both the values and descriptions are visible.

To set separate value and description fields
1. Open the sample report called Group.rpt.
   Sample reports are located in the Crystal Reports directory under Samples\En\Reports.
2. In the Field Explorer, select Parameter Fields and click New.
   The Create New Parameter dialog box appears.
3. Enter a name for the parameter in the Name field.
   This example uses Customer Name.
4. In the List of Values area, click Dynamic.
5. Click Insert and, in the Value field, select Customer ID.
6. In the Description field, select Customer Name.
7. In the Options area, set the Prompt with Description Only option to True.
8. Click OK.

When your users see this prompt in the prompting dialog box, they do not see values from the Customer ID field (the field on which the parameter is based), but rather, they see a list of customer names.

Using command objects as list-of-values data sources

Crystal Reports supports command objects as data sources for unmanaged lists of values. To use this feature, you first create a command object using the Database Expert to return the values that you want in your list-of-values object. For more information see “Defining an SQL Command” in the online help.

The query you create should return at least one database field per level in the new list-of-values object. For example, if you want to build a Country, Region, City list-of-values object, your query should return at least three columns from
your database: Country, Region, and City. You need to return all three values in a single select statement. There is no need for an order by clause, because the prompting engine sorts the values according to your specifications.

Note: It is not necessary to link your command object to your report fields. In general, command objects that you use only to populate lists of values do not need to be linked to your report. However, if you want to use a column from this command object in your report, you need to link the command object to the report.

All the command objects defined in your report appear in the value fields drop-down list, and in the description fields drop-down list in the Create New Parameter dialog box. You can then associate your list of values with the command object.

Null handling

Null values in the data returned by the list of values are displayed as “null” in the drop-down list for that prompt. Your users can select these values, and any parameters associated with that prompt receive the null value. You can test for this functionality in the Crystal Reports formula language with the IsNull function. For more information see “IsNull (fld)” in the online help.

Best practices for prompting

Unmanaged reports

Unmanaged reports are reports that you store outside of BusinessObjects Enterprise. These reports can use lists of values that you define within the report, or lists of values that you have stored in the BusinessObjects Enterprise Repository. (You create report lists of values in Crystal Reports when you use the Create New Parameter dialog box. You create repository lists of values in the Business Views Manager.)

Lists of values defined within a report have the following properties:

• They can be based on command objects or report fields.
• They are always stored within the .rpt file.
• They cannot be shared between reports.

If your unmanaged report also uses a repository-based list of values, the report will prompt you to log on to the repository whenever it is opened. This process is required so that Crystal Reports can locate the list-of-values object.
To maximize performance and scalability of unmanaged reports, you should reuse list-of-value objects within a single report wherever possible. For information about how to do this, see “Sharing common lists of values within a report” on page 453.

Managed reports

Managed reports are reports that you store within BusinessObjects Enterprise.

Lists of values used in managed reports have the following properties:
• They are always stored in the repository.
• They can be shared between reports.
• They are based on Business Views.

To maximize performance and manageability of your managed prompts, the following practices are recommended:
• Define your list-of-values objects in Business View Manager, not Crystal Reports.
• Build a separate Business View to provide lists of values for all of your reports. The only fields that you need in this Business View are those that you use for prompting. See the Business Views Administrator’s Guide for information about how to create a Business View.
• Create managed list-of-values objects for each dynamic prompt that you intend to use in your report. These objects are visible to Crystal Reports users when they design parameters and prompts. If you need a list of values for a Country > Region > City hierarchy, and a Country pick list, you can satisfy both needs with a single list-of-values object.

Defining your list-of-values objects in Business View Manager has the following benefits:
• You have control over the metadata that is used to define the prompting list of values.
• You can apply row-level security (if desired) to the Business View that is used for prompting.
• You can separate metadata design from report design. A metadata designer can be responsible for authoring the list-of-values definitions, and the report designer can be responsible for authoring the report.
• This method minimizes the number of metadata objects you need to create to support prompting.
Converting unmanaged reports to managed reports

You can convert an unmanaged report to a managed report in the following ways:

• Publish one or more reports with the Publishing Wizard.
• Save the report to BusinessObjects Enterprise using the Crystal Reports Save As command.
• Create a new report object from within the InfoView portal.
• Create a new report object from within the Central Management Console portal.

In all of these cases, these actions are carried out on the prompt object in your unmanaged report:

• List-of-values objects that are defined in the report are converted to repository lists of values. A Business View, Business Element, Data Foundation, and Data Connection object is created.
• If the same list-of-values object already exists in the repository, no duplicate object is created. Instead, your report references the existing list-of-values object.
• List-of-values objects that are repository-based are unchanged.
• Prompt groups that are defined in the report are converted to repository prompt groups.
• Prompt groups that are repository-based are unchanged.
• All of the new repository-based Business View objects, list-of-values objects, and prompt-group objects are created in the repository folder called Dynamic Cascading Prompts.

Deploying managed reports with dynamic prompts

Repository-based lists of values and prompt groups are repository objects like any other. When managed reports are migrated from one repository to another using the Import Wizard, the list-of-values and prompt-group objects are imported as well. This functionality makes it easy to migrate reports from system to system. BusinessObjects Enterprise tracks the objects that a report depends on, and imports these objects to the destination system if needed.

Business Views Manager has been enhanced to support the import and export of lists of values and prompt groups. For more information, see the Business Views Administrator’s Guide.
Deleting parameter fields

There are several methods for deleting parameters within a report. The type of parameter you are deleting determines the method you can use.

To delete a parameter that is not used in a formula
1. On the View menu, click Field Explorer.
   The Field Explorer appears.
2. Expand the Parameter Fields folder and click the parameter you want to delete.
3. Click Delete.
   If the parameter is used in your report, a Crystal Reports dialog box appears confirming whether you want to delete the parameter.
4. Click Yes.
   This parameter is removed from the Parameter Fields folder and from your report.

To delete a parameter used with the Select Expert
   Tip: Another way to do this is to click the Select Expert button on the Expert Tools toolbar.
   The Select Expert dialog box appears.
2. Choose the tab whose selection criteria uses the parameter you want to delete.
3. Click Delete.
4. Click OK to close the Select Expert.
5. On the View menu, click Field Explorer.
6. Expand the Parameter Fields folder and click the parameter you want to delete.
7. Click Delete.

To delete a parameter that is used in a formula
1. On the View menu, click Field Explorer.
   The Field Explorer appears.
2. In the Formula Fields folder, select the formula that contains the parameter you want to delete.
3. Click Edit and delete the parameter field from the formula.
Note: If the parameter is used in more than one formula, it must be deleted from each formula.

5. Expand the Parameter Fields folder and click the parameter you want to delete.
6. Click Delete.

Responding to parameter field prompts

Previewing a report for the first time

When you preview a report for the first time, the Enter Prompt Values dialog box appears, prompting you for a value.
• If you specified a default value when you created the parameter field, the program will use that value unless you specify a new one.
• If you did not specify a default value, the program will not refresh the data until you supply a new value.

Note: If the parameter is a string value type, without a default value, and the “Allow discrete values” field is True, an empty string appears.

Refreshing report data

When you refresh data from the Preview tab, the Refresh Report Data dialog box appears.

Select the “Use current parameter values” option to use the current parameter value.
Select the “Prompt for new parameter values” option to enter a new parameter value. When you select this option and click OK, the Enter Prompt Values dialog box appears.
• Enter String values exactly as they will appear in the field. If the parameter field allows multiple values, you can enter additional parameter values using the Add button.
• Enter Boolean values using the following format: TRUE or FALSE.
• Enter Number values exactly as they will appear in the field.
• Enter Currency values exactly as they will appear in the field.
• Enter Date values to match the format used on-screen. If the format is unspecified, enter as Date (Year, Month, Day). For example, Date (1997, 5, 21). To access the calendar, click the drop-down arrow beside the date.
• Enter Time values to match the format used on-screen. If the format is unspecified, enter as Time (Hour, Minutes, Seconds AM/PM). For example, Time (4:32:12 PM). You can also select the unit of time and then use the up and down arrows to scroll through the numbers.
• Enter DateTime values using the following format: Date (Year, Month, Day), Time (Hour, Minutes, Seconds AM/PM). For example, Date (1997, 5, 21), Time (4:32:12 PM). You can also enter DateTime values by using the associated drop-down arrow to access the calendar, and the up and down arrows to scroll through the time.

**Note:** The drop-down arrow to access the calendar for dates and the up and down arrows to scroll for times will only be available if you have the correct version (4.70 or later) of comctl32.dll.

To use the current value displayed, click OK, or, to use a different value from the one displayed, type a new value in the text box, and click OK.
• If the parameter field is range limited, then you can only enter values within a certain range. The range limit is specified in the Set Default Value dialog box when creating or editing a parameter.
• If the parameter is a string value type, you can limit the values the user is able to enter by using the length limit option or by using an edit mask to restrict the format.

The program now runs the report using the new value(s) you specified.

**Advanced parameter features**

There are a variety of ways that parameters can be used within a report. This section covers some of the advanced methods of using parameters:
• “Creating a parameter with multiple prompting values” on page 462.
• “Applying conditional formatting using parameter fields” on page 462.
• “Creating a report title using parameter fields” on page 463.
• “Specifying single or ranges of values” on page 464.
• “Incorporating a parameter into a formula” on page 465.
• “Defining sort order using parameter fields” on page 466.
• “Defining entry type and format using the Edit Mask” on page 467.

Creating a parameter with multiple prompting values

1. On the View menu, click Field Explorer.
2. Select Parameter Fields and click New.
3. Enter a Name for your parameter.
4. Select a value type from the Type list.
5. From the Value Field list, select the table and field that you want to use to obtain your prompting values.
6. Add the values from the field you selected by clicking Options and choosing Append all database values.
   You can also import values, or you can create your own in the Value area under the Options button.
7. In the Options area, select the Allow multiple values option and set its value to True to enable your users to enter more than one prompting value for this parameter.
   Both discrete and range type values are allowed individually or in combination.
   **Note:** For details on single or range values, see “Specifying single or ranges of values” on page 464.
8. Click OK.

Applying conditional formatting using parameter fields

Parameter fields can be used to create conditional formatting formulas. You can customize these formulas whenever you refresh the report data. A conditional formatting formula could be used for color-flagging data that meets certain conditions. For example:

- Sales representatives who sell more than 10% over quota.
- Customers who have not ordered in the last quarter.
- Inventory items that have not had any movement in the last month.

If the conditions under which you flag these items never change, you do not need to use parameter fields. You can just use formulas (for text flags) or conditional formatting (for border flags). However, to change the conditions from report to report, you need to use parameter fields in formulas and conditional formatting formulas.
To apply conditional formatting using parameter fields

1. Create the parameter field of the data type you need for the formula.
2. Create the formula and use the parameter field in place of the fixed value you would normally use.

For example, to be prompted for all the customers whose last year’s sales were over a certain value, and to print their names in red, select the Last Year’s Sales field and click Format from the Expert Tools toolbar.

The Format Editor appears.

3. Click the Conditional formula button next to the Color property on the Font tab, and format the field using a conditional formatting formula like this:

   If {customer.LAST YEAR’S SALES} > {?SalesTarget} Then
   Red
   Else
   Black

Now, when you refresh the data, the program will prompt you for the value that triggers the color flag (known as the threshold value). It then runs the report and flags all the customers that had sales last year above the threshold figure. You can change the figure each time you run the report and the program will flag a different set of Customer Names.

For more information see “Conditional formatting functions” in the online help.

Creating a report title using parameter fields

Crystal Reports allows you to use parameter fields to create a report title that can be changed each time the report is refreshed.

To creating a report title using parameter fields

1. On the View menu, click Field Explorer.
   The Field Explorer appears.

2. Select Parameter Fields and click New.
   The Create New Parameter dialog box appears.

3. Type a name for the parameter field in the Name field.
4. Select String from the Type list.
5. To specify a default title, click Click here to add item in the Value area and type the desired text.
   Continue adding titles as required.
6. Click OK.
7. Return to the Field Explorer, and then place the parameter field in the Page Header section of the report to have the title appear on every page, or in the Report Header section if you want the title to appear on only the first page of the report.

Now, when you refresh the data, the program will prompt you for a report title. If you wish, you can change the title each time you run the report.

Specifying single or ranges of values

You can create parameters that require users to enter a single (discrete) value, or a range of values. If you then include these parameters in your report’s record selection, you can help users find specific information. For example, in the case of a record selection parameter that uses a discrete value, a user might enter a single country name to see sales figures for only that country. In the case of a record selection parameter that uses a range of values, a user might enter a range of countries (for example, Canada and France) to see sales figures for all the countries in the range (in this case, Chili, China, Denmark, England, and so on).

To specify single or range values

1. Select the Allow discrete values option or the Allow range values option to specify whether the parameter field will accept a range of values.

   • If you select “Allow discrete values,” the parameter field will accept single values (rather than ranges of values).
   • If you select “Allow range values,” then when you are prompted for parameter values, you can enter a start value and an end value. For example, if you enter the values “5” and “10”, the range is 5-10, and a report that uses this parameter for record selection will display all records with values between 5 and 10. This also works for string parameter fields. With a start value of “A” and an end value of “E”, a report that uses this parameter for record selection will display all records within an alphabetical range of A-E.

   Note: If the “Allow multiple values” and the “Allow discrete values” options are selected, the parameter field will accept multiple single values. In this case, you can enter more than one value, but these values will be evaluated individually and will not be interpreted as a range. If the “Allow multiple values” and “Allow range values” options are selected, the parameter field will accept multiple ranges.
2. If applicable, select the **Min Length** and/or **Max Length** options to designate the length of the field.
   - For a Currency or Number parameter field, options for entering the “Min Value” and “Max Value” appear.
   - For a Date, DateTime, or Time parameter field, options for entering the “Start” and “End” values appear.

**Incorporating a parameter into a formula**

1. On the **View** menu, click **Field Explorer**.
   The Field Explorer appears.
2. Select **Parameter Fields** and click **New**.
   The Create New Parameter dialog box appears.
3. Create a parameter field and save it.
4. Select **Formula Fields** and click **New**.
   The Formula Name dialog box appears.
5. Enter the name of the formula, and then click **OK**.
6. Create a formula using the parameter field as you would any constant value. For example, rather than creating a formula that hard-codes the country name:
   {customer.COUNTRY} = “USA”
   Use a parameter field instead of “USA”.
   {customer.COUNTRY} = {?Country}
   To do this double-click the database field, press =, then double-click the parameter.
   **Tip:** Identify parameter fields easily by looking for (?).
7. Click **Close** on the Formula Workshop.
   The Field Explorer dialog box appears. The name of the formula you have just created is highlighted in the Formula list box.
8. Drag and drop the formula into the report.
9. Click **Refresh** to generate the report.
   A dialog box appears, prompting you for values.
Defining sort order using parameter fields

To set the sort order using parameter fields, you need to first create a formula that includes a parameter field and then sort based on that formula. For example, assume that you have a customer list report based on the Customer table. For each customer, you show the Customer Name, City, Region, Country, and Phone Number. You want to be able to sort the report by Country, by Region, or by City, depending on your needs at the time.

1. Create a parameter field and call it SortField.
2. Select String from the Type list.
   You may want to limit the number of characters the user can type to one.
3. To do this, type the numeral 1 into the Min Length and Max Length text boxes and click OK.
   Now the parameter field will only accept single-character values. The field will accept “C” as a value, but not “City.”
4. In the Prompting Text field, enter a prompt similar to this:
   Type R to sort by Region or C to sort by City; otherwise, data will be sorted by Country.
5. Create a formula similar to this and call it Sort:
   If {?SortField} = "C" Then {customer.CITY}
   Else
     If {?SortField} = "R" Then {customer.REGION}
     Else
       {customer.COUNTRY}
   This formula prompts for a value for the parameter field {?SortField}. If you enter “C”, the formula will sort by the City field. If you enter “R” it will sort by the Region field. If you enter anything else, or do not enter anything at all, the formula will sort by the Country field.
   For more information see “If statements” in the online help.
6. Place the formula in the Report Header section of the report and select Suppress (No Drill-Down) in the Section Expert so that it does not print.
7. Click Sort Records.
8. Choose your formula and click Add.
9. Click OK.
   Now when you run the report, the program will prompt you for a sort field, the formula will return a value based on your selection, and the sort facility will use that value as your sort field.
Defining entry type and format using the Edit Mask

1. On the View menu, click Field Explorer.
2. Select Parameter Fields and click New.
   The Create New Parameter dialog box appears.
3. Enter the Name and value Type.
4. For a string parameter field, you can choose to enter an edit mask in the Edit mask field, rather than specifying a range.

An edit mask can be any of a set of masking characters used to restrict the values you can enter as parameter values (the edit mask also limits the values you can enter as default prompting values).

You can enter any of the following masking characters, or any combination of them:

- “A” (allows an alphanumeric character and requires the entry of a character in the parameter value).
- “a” (allows an alphanumeric character and does not require the entry of a character in the parameter value).
- “0” (allows a digit [0 to 9] and requires the entry of a character in the parameter value).
- “9” (allows a digit or a space, and does not require the entry of a character in the parameter value).
- “#” (allows a digit, space, or plus/minus sign, and does not require the entry of a character in the parameter value).
- “L” (allows a letter [A to Z], and requires the entry of a character in the parameter value).
- “?” (allows a letter, and does not require the entry of a character in the parameter value).
- “&” (allows any character or space, and requires the entry of a character in the parameter value).
- “C” (allows any character or space, and does not require the entry of a character in the parameter value).
- “. , : ; - /” (separator characters). Inserting separator characters into an edit mask is something like hard coding the formatting for the parameter field. When the field is placed on the report, the separator character will appear in the field object frame, like this: LLLL/0000. This example depicts an edit mask that requires four letters followed by four numbers.
- “<“ (causes subsequent characters to be converted to lowercase).
• “>” (causes subsequent characters to be converted to uppercase).
• “\” (causes the subsequent character to be displayed as a literal). For example, the edit mask “\A” would display a parameter value of “A.” If the edit mask is “00\A00,” then a valid parameter value would consist of two digits, the letter “A,” and then two additional digits.
• “Password”. Allows you to set the edit mask to “Password,” you can create conditional formulas specifying that certain sections of the report become visible only when certain user passwords are entered.

**Note:** Some of the edit mask characters require that you enter a character in their place (when entering a parameter value), while others allow you to leave a space, if needed. For example, if the edit mask is 000099, you can enter a parameter value with four digits, five digits, or six digits, since the ‘9’ edit mask character does not require the entry of a character. However, since ‘0’ does require such an entry, you could not enter a parameter value with less than four digits.

5. Click **OK**.
6. Return to the Field Explorer, and drag the parameter into the report.