Overview

This document provides additional details beyond the product documentation on how to best use the new dynamic cascading prompting feature in Crystal Reports XI.

Contents

INTRODUCTION ........................................................................................................ 2
PROMPTING OVERVIEW .................................................................................... 2
DEFINITIONS .......................................................................................................... 2
WORKING WITH THE FRAMEWORK................................................................. 3
   Start Designing Prompts in Business Views ................................................. 3
   How to Start in Business Views ................................................................. 4
   What if I didn’t start designing in Business Views? ................................. 5
   Use Existing Prompt Objects .................................................................... 6
   Method #1: Sharing Existing Prompt Objects within the Report .......... 6
   Method #2: Sharing Existing Prompts within the Framework .......... 9
   Example of Sharing Existing Prompts within the Framework .......... 11
   Modifying Repository Prompt Objects .................................................... 15
   List of Values ......................................................................................... 15
   Prompt Groups ....................................................................................... 16
DYNAMIC PROMPTS, STORED PROCEDURES AND COMMAND OBJECTS....... 17
   Create a Data Source for the List of Values ........................................... 17
   Method #1: Using an Existing Data Source (Recommended) ............. 17
   Method #2: Creating a New Parameter Data Source ....................... 18
CONVERTING STATIC PROMPTS TO DYNAMIC PROMPTS .................. 21
   Static to Dynamic Conversion Tutorial ................................................ 21
Introduction

Dynamic cascading prompting is a powerful addition to the feature set of 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 document will help explain how to take advantage of these innovative capabilities.

Prompting Overview

Parameter prompting in Crystal Reports XI is separated into three functional objects: parameters, Lists of Values and Prompt Groups.

Parameters are formula language elements that receive their values from the prompting page (the Enter Values dialog box), and can optionally be associated with a static or dynamic pick list.

A List of Values encapsulates the data definition of a pick list. This includes the data source, and which fields in the data source should be used for the value field or description field of the prompt. The data source can be one of three types:

- Report table
- Command object
- Business View

Prompt Groups encapsulate the presentation of a pick list. This includes whether the prompt supports multiple values, discrete values, or range values.

Definitions

The table below contains the terms and corresponding descriptions used throughout this document.

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Framework</td>
<td>Either BusinessObjects Enterprise or Crystal Reports Server. Managed reports have been published to the framework.</td>
</tr>
<tr>
<td>Prompt Objects</td>
<td>New metadata objects added to the framework to support dynamic cascading prompting – these are List of Values and Prompt Groups. Both are editable from within Business Views.</td>
</tr>
<tr>
<td>Managed Report</td>
<td>A report that has been published to the framework</td>
</tr>
</tbody>
</table>
Working with the Framework

This section is designed to assist you in working with the framework. If you do not intend to publish your report to BusinessObjects Enterprise XI or Crystal Reports Server XI (the framework), you will not need this section.

Dynamic cascading prompts are very powerful when used in combination with the Business Views technology that is found in both BusinessObjects Enterprise and Crystal Reports Server.

In this section, we refer to a managed report as a report that has been published to the framework.

Start Designing Prompts in Business Views

To use this technology effectively, it is best to create reusable prompting objects in a Business View by using Business View Manager. Business View Manager is included with BusinessObjects Enterprise XI and Crystal Reports XI Server.

The main benefits of creating prompt objects in a Business View are:

- **Manageability.** This practice will help you to manage the objects in your metadata environment. Different prompt objects can be placed in different folders depending on who should be accessing them. This will also eliminate any potential for redundant metadata objects.

- **Designer productivity.** Defining your List of Values objects initially in Business Views allows report designers to add prompts to a Crystal report without having the report designer specify how each prompt gets its data. For example, this allows for a common List of Values for ‘Customer’ to be defined once in Business Views and that List of Values is visible in Crystal Reports.

- **Performance.** When multiple reports are sharing metadata-defined prompt objects, then a single List of Values is populated on the server and those values are used across all the reports that use them. For example, if you have 10 reports all using a list of customer values, then that list of customer values is populated once from the database and those values are shared by all reports that use it.

- **Advanced capabilities.** Defining a Business View for your prompt allows you to add view-time security filters so that you may filter pick list values to different users or groups. In addition, the ability to
define a refresh schedule for a List of Values is available within Business View Manager.

Finally, you should remember that your report data source is separate from your prompting data source. Your report data can still come from a conventional data source such as ODBC, or a native driver even if your prompts are defined in Business Views. This means your report can continue to use the data source that it is currently using.

**How to Start in Business Views**

Business Views are metadata objects that are managed with a program called Business View Manager. Business View Manager installs with Business Objects Enterprise and Crystal Reports Server. To start Business View Manager, click **Start > Programs > BusinessObjects XI > BusinessObjects Enterprise > Business View Manager.**

Create your metadata objects in the following order:

1. Data Connection
2. Data Foundation
3. Business Element
4. Business View
5. List of Values

Once complete, your List of Values objects will appear in Crystal Reports’ **Create New Parameter** dialog box, as shown in Figure 1.

![Create New Parameter](image)

**Figure 1**

Figure 1 shows four List of Value objects that were created in Business View Manager and are visible within the Crystal Reports designer. In this way, you can start using pre-built List of Values objects immediately without having to first design how their pick lists are populated.
What if I did not start designing in Business Views?

The guideline to start designing in Business Views is only a recommendation. The prompting feature will work correctly if you create prompt objects in Crystal Reports, and then publish to the framework.

If you choose to create a new List of Values objects using Crystal Reports’ Create New Parameter dialog box (see Figure 2), then when this report is published, new prompting objects and Business View objects are automatically created. If a duplicate or matching List of Values object already exists in the framework then a new one is not created.

![Figure 2](image)

This means that if you create your prompt objects in Crystal Reports and then publish your report to the framework, you can expect the following behavior:

- For each unique List of Values, four Business View objects and two prompt objects are created in the **Dynamic Cascading Prompts** folder of the BusinessObjects Enterprise Repository. The four Business View objects are Data Connection, Data Foundation, Business Element, and Business View. The two prompt objects are List of Values and Prompt Group.

- The framework creates default names for all these objects such that they all appear in a group within the Business View Manager Repository Explorer. Figure 3 is the result of publishing the Prompting.rpt sample report, included with Crystal Reports, to BusinessObjects Enterprise or Crystal Reports Server:
Duplicate List of Values objects are not created. For example, if the report designer modifies the *Prompting.rpt* sample report so that the *City* parameter allows range values and republishes the report, a new Prompt Group object is created, but the List of Values and Business View objects remain unchanged.

In the same way, if a report designer independently creates a List of Values object that matches an existing one in the repository, when that report is published it uses the pre-existing List of Values object instead of creating a duplicate.

**Use Existing Prompt Objects**

Whether you plan to publish to the framework or not, you will benefit from sharing prompt objects within a single report and between multiple reports.

**Method #1: Sharing Existing Prompt Objects within the Report**

If you do not plan to publish your report to the framework, you can still share prompt objects within a single report. This practice will improve performance because the List of Values is only populated once, even if it is used in pick lists for multiple parameters.

If you need two Lists of Values that have the same data but for different parameters, then you do not have to create two different data sources. For example, if you are authoring a report that shows transactions between two companies, you need two parameters (*Supplier* and *Consumer*) with matching pick list values.

You create the first parameter and dynamic List of Values as shown in Figure 4.
Now, when creating your second parameter, the existing dynamic List of Values that you used with the first parameter is visible in the **Existing** drop-down box, as shown in Figure 5.

The drop-down box in Figure 5 shows two items:

- The dynamic List of Values you created in this report.
- The Prompt Groups associated with each List of Values.

If you choose the Prompt Group object in the list, then you see the work that you did in configuring the first parameter (SupplierID). You would choose this if you want to associate the new ConsumerID parameter with a different level in the same Prompt Group.

If you choose the List of Values object for Country > Region > City > Customer ID, then you are reusing the same data definition that was created for the first SupplierID parameter. However, you are also creating a new Prompt Group. This means you can modify the presentation settings such as single, multiple, or range values, sort order, and prompt with description only. These values can be different from those used with the existing parameter. After clicking ‘OK’, create a new parameter called ConsumerID that shares the same pick list data as SupplierID, but, optionally, has different presentation settings. If you return to this dialog box after creating the new parameter, you can see that there are now two Prompt Groups using the same List of Values object (Figure 6).

![Figure 6](image)

When running the report, the Enter Values dialog box (Figure 7) shows two different sets of prompts; one for each parameter. The pick list values are shared between the two parameters and, therefore, overall performance is maximized.
Method #2: Sharing Existing Prompts within the Framework

If you use either BusinessObjects Enterprise or Crystal Reports Server, then you will benefit from sharing prompt objects between reports, and not just within a single report.

Sharing prompt objects between reports has several benefits.

- Improved system performance from reduced database access.
- Improved system scalability from shared lists of values.
- Improved designer productivity from reuse of existing prompt objects.

When logged on to the framework, an additional node, indicating the Central Management Server (CMS), appears in the Existing combo box of the Edit Parameter dialog box (Figure 8).
The table in Figure 9 explains the contents of the combo box.

<table>
<thead>
<tr>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Report" /></td>
<td><strong>Prompt objects</strong></td>
</tr>
<tr>
<td></td>
<td>Prompt objects under the Report node are stored in the report.</td>
</tr>
<tr>
<td></td>
<td>Prompt objects under the CMS node are stored in the framework</td>
</tr>
<tr>
<td></td>
<td>If the CMS node is not present, then you are not logged on to the framework</td>
</tr>
<tr>
<td><img src="image" alt="Customer ID By Geography" /></td>
<td><strong>List of Values</strong></td>
</tr>
<tr>
<td></td>
<td>List of Values can be either locally used in the report, or stored in the</td>
</tr>
<tr>
<td></td>
<td>framework. When it is stored in the framework, the framework folder is</td>
</tr>
<tr>
<td></td>
<td>identified with square brackets ([]). In this example, the Customer ID</td>
</tr>
<tr>
<td></td>
<td>By Geography List of Values is</td>
</tr>
<tr>
<td></td>
<td>stored in the Dynamic Cascading Prompts framework folder.</td>
</tr>
</tbody>
</table>
Selecting a List of Values object will create a new Prompt Group based on that List of Values.

**Employee ID - Prompt Group**

**Prompt Group**

Prompt Groups can be either locally used in the report, or stored in the framework.

Selecting a Prompt Group under the **Report** node results in displaying the Prompt Group that is already used in the report.

Selecting a Prompt Group under the repository node reuses that Prompt Group definition.

---

**NOTE**

If you are not connected to the framework, then the CMS node does not appear in the combo box. To connect to the framework, use the **Logon** button in the Repository Explorer.

---

**Example of Sharing Existing Prompts within the Framework**

If you followed Guideline #1 by creating your prompting objects in Business View Manager, then you will have a number of List of Values objects already defined in your framework. You can immediately start to use these objects in your reports and they can be shared with many different reports.

The following scenarios highlight the benefits of sharing existing prompts within the framework. In this example John and Susie work for the same company, and they will be sharing prompt objects.

John is designing a customer summary report, and he wants to prompt on customers by geography; therefore, he configures his Customer ID parameter as shown in Figure 10.
John uses the existing List of Values definition that is stored in the framework. It has no current Prompt Groups associated with it. By selecting the List of Values, he is automatically creating a new Prompt Group.

Because this is a customer summary report, he wants to allow multiple customers in multiple cities and regions to be selected. Therefore, he modifies the Prompt Group settings for the region, city, and customer level to allow multiple values.

The resulting prompt output is shown in Figure 11.
Susie is creating a customer detail report. Therefore, instead of reporting off multiple customers, she wants to report off a single customer.

For efficiency, Susie wants to reuse existing prompt objects, so she checks to see if there are prompting objects that meet her needs. As shown in Figure 12, she finds the existing Customer ID by Geography List of Values that now has the Prompt Group that John created for his report.
To save time and effort, Susie wants to see if she can use the Prompt Group created by John. By clicking on the Customer ID By Geography – Prompt Group entry, she can reuse John’s entire prompt configuration. She tries that but sees that it allows for multiple value prompts. She does not want multiple value prompts so she needs to create a new Prompt Group with her specific configuration. The framework will not allow Susie to modify the existing Prompt Group because it is used by John’s report.

By clicking on the Customer ID By Geography List of Values object, she can create a new Prompt Group that uses the same List of Values. She can then adjust the Prompt Group settings according to her preferences. The result is the prompt screen is shown in Figure 13.
The next time a report designer needs to use a customer prompt, they will now have available to them two predefined prompt configurations: John’s that allows multiple values for some levels, and Susie’s that allows only single values.

**Modifying Repository Prompt Objects**

Modification of repository based prompt objects is handled through Business View Manager. To start Business View Manager click **Start > Programs > BusinessObjects XI > BusinessObjects Enterprise > Business View Manager**.

**List of Values**

List of Values objects are created by clicking **File > New > List of Values**. Doing so launches the **Welcome to Business View Manager** dialog box, as shown in Figure 14.

![Welcome to Business View Manager](image)

Figure 14

List of Values objects can be modified if there are no Prompt Groups associated with them. Once a Prompt Group is associated with a List of Values object, that List of Values cannot be modified.

To see the Prompt Groups that are associated with a List of Values object, select the List of Values object in the Repository Explorer, and click **Show Dependent Objects** (Figure 15).

Guideline #3: Use Business View Manager to check dependencies between Prompt Groups and List of Values
Prompt Groups

Prompt Groups are created with the Crystal Reports designer. All Prompt Groups for managed reports are stored in the framework. These Prompt Groups can be edited within Business View Manager by double-clicking the Prompt Group within the Repository Explorer and modifying the settings (Figure 16).

Modifying the Prompt Group within Business View Manager affects all the reports that use this prompt. Therefore, if you want to make a global
Dynamic Prompts, Stored Procedures and Command Objects

Combining dynamic prompts with reports that are based on command objects and/or stored procedures is supported. The following section describes how to configure your report to enable this.

This procedure applies whether or not you plan to publish your report to the framework.

Create a Data Source for the List of Values

Method #1: Using an Existing Data Source (Recommended)

If you followed Guideline #1 by creating your prompting objects in Business View Manager, you already have a data source in the framework for your List of Values.

You can then edit an existing stored procedure or command object parameter in the Crystal Reports Field Explorer and modify it to use the dynamic prompt instead of the existing static prompt. To change it from static to dynamic, edit the parameter, click the Dynamic option button, and click Existing for the data source. Your list of framework based data sources should appear in the combo box as shown in Figure 17.

![Figure 17](cr_xi_dyncascading_prompting.pdf)

You can then select the List of Values or Prompt Group that you want to use with your parameter, and then your stored procedure or command...
Object parameter will use a dynamic List of Values. For systematic instructions, refer to the Converting Static Prompts to Dynamic Prompts section of this document.

Method #2: Creating a New Parameter Data Source

Use this method if you do not intend to publish your report to the framework, or if you prefer to create a new List of Values within the report instead of using an existing one.

Start by creating a data source with the Crystal Reports designer that will be used only for populating the List of Values for the parameter.

This report based data source can be either a report table or a command object/stored procedure. The Database Expert (Figure 18) is used to add the table or command object/stored procedure to the report. These new tables do not have to be linked to the report tables because they are only used to populate the List of Values. In this example, the report is based on the Microsoft SQL Server sample stored procedure called CustOrders.

![Database Expert](Figure 18)

The two tables in the Database Expert are CustOrders for the report, and CustomerList for the dynamic prompt. The CustOrders table is a stored procedure that uses the CustomerID parameter for the report data. The CustomerList command object (Figure 19) is used to populate the pick list for the CustomerID parameter.
The `CustomerListofValues` command object is straightforward (Figure 19). You do not need to link the command object and the stored procedure together in the links tab of the Database Expert. This is because the command object is only used for the prompt List of Values, and the stored procedure is only used for the report data. Crystal Reports may give you two warning messages about the tables not being linked. You can ignore these messages and continue designing your report.

In the Field Explorer (Figure 20), you can see the stored procedure’s parameter.

You can edit this parameter. Make this parameter Dynamic, and choose `OrderID` from the `OrdersListofValues` table as the value field.
By choosing the CustomerListOfValues as the table, you are using a different, dedicated data source to populate your List of Values that is separate from your command object.

You can then complete the configuration of the List of Values:

Click OK to complete the modification of the CustomerID prompt.
When you refresh your report with this configuration, you will initially see a prompt page that does not have a pick list associated with it. This behavior is a bug that occurs only when you are working with a report-based List of Values. When this report is published to the framework, this additional prompt will not occur. Clicking **OK** will dismiss this initial prompt page. The next prompt page will appear with the pick list showing the list of customers, as shown in Figure 23.

![Figure 23](image)

You can now publish your report to the framework.

**Converting Static Prompts to Dynamic Prompts**

This section describes how an existing report that uses static prompts can be converted to use dynamic prompts. This information applies to both managed and unmanaged reports.

This example assumes that you have an existing report that has three static parameters - Country, Region, and City. The goal is to change the static prompts to dynamic prompts.

**Static to Dynamic Conversion Tutorial**

1. In Crystal Reports Field Explorer, right-click the *City* parameter and then click **Edit**.
2. In the **Edit Parameter: City** dialog box (Figure 24), click the **Dynamic** option.
3. Click the **New** option button for the data source.
4. Click **Insert** and then select the appropriate report fields to define the List of Values hierarchy. In this example, the hierarchy is **Country**, **Region**, and then **City**.
5. In the **Parameters** column at the **City** field level, click **Click to create parameter**. Doing so binds the existing parameter to the **City** level in the List of Values.
6. Leave the **Country** and **Region** level unmapped and click **OK** to complete the editing of the **City** parameter.
7. In the Crystal Reports Field Explorer, right-click the Region parameter and then click Edit.

8. In the Edit Parameter: Region dialog box (Figure 25), click the Dynamic option button.

9. Click the Existing option for a data source and then select the Country > Region > City – Prompt Group that you created when editing the City parameter.

10. In the Parameters column at the Region field level, click Click to create parameter. Doing so binds the existing parameter to the Region level in the List of Values.

11. Leave the Country level unmapped and click OK to complete the editing of the Region parameter.

12. Complete the conversion from static to dynamic by repeating steps 7 through 11 for the Country (Figure 26) parameter.
13. To review the changes you have made, open the Country parameter for editing. Figure 27 is what you will see.
www.businessobjects.com

No part of the computer software or this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system, without permission in writing from Business Objects.

The information in this document is subject to change without notice. Business Objects does not warrant that this document is error free.

This software and documentation is commercial computer software under Federal Acquisition regulations, and is provided only under the Restricted Rights of the Federal Acquisition Regulations applicable to commercial computer software provided at private expense. The use, duplication, or disclosure by the U.S. Government is subject to restrictions set forth in subdivision (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at 252.227-7013.

The Business Objects product and technology are protected by US patent numbers 5,555,403; 6,247,008; 6,578,027; 6,490,583; and 6,289,352. The Business Objects logo, the Business Objects tagline, BusinessObjects, BusinessObjects Broadcast Agent, BusinessQuery, Crystal Analysis, Crystal Analysis Holos, Crystal Applications, Crystal Enterprise, Crystal Info, Crystal Reports, Rapid Mart, and WebIntelligence are trademarks or registered trademarks of Business Objects SA in the United States and/or other countries. Various product and service names referenced herein may be trademarks of Business Objects SA. All other company, product, or brand names mentioned herein, may be the trademarks of their respective owners. Specifications subject to change without notice. Not responsible for errors or omissions.

Copyright © 2007 Business Objects SA. All rights reserved.