



PUBLIC

SAP BusinessObjects Business Intelligence platform

Document Version: 4.3 Support Package 4 – 2023-12-07

Using Business Views

Content

1	Introduction to Business Views	7
1.1	About this documentation.	7
	What is Business Views?	7
	Who should read this guide.	7
2	Business Views Architecture	9
2.1	Business Views overview.	9
2.2	Architecture overview and diagram.	10
2.3	Client tier.	11
2.4	Business tier.	11
	Business View Manager.	12
	Business Views objects.	12
	Data Connection.	13
	Dynamic Data Connection.	13
	Data Foundation.	13
	Business Element.	13
	Business View.	14
2.5	Data tier.	14
2.6	Information flow.	14
	Usage scenario.	15
	Data Connection layer.	16
	Data Foundation layer.	18
	Business Element layer.	20
	Business View layer.	20
2.7	Architectural workflow.	20
3	Quick Start: Creating a Business View	23
3.1	Quick Start overview.	23
3.2	Before you begin.	23
	Sample data - Xtreme.mdb.	23
	Conventions.	24
	Tutorial scenario.	25
	What is a Business View?	25
3.3	Creating and configuring a Data Connection.	27
	Setting a Data Connection password.	31
	Saving and naming a Data Connection.	32
	Modifying the properties of a Data Connection.	33

	Exercise: Creating two more Data Connections.	34
3.4	Creating and configuring Dynamic Data Connections.	35
	Sorting Data Connections.	37
	Saving and naming a Dynamic Data Connection.	37
3.5	Creating and configuring a Data Foundation.	38
	Linking tables.	40
	Inserting a formula.	42
	Inserting an SQL expression.	44
	Inserting a parameter.	46
	Inserting a Business Filter.	48
3.6	Creating and configuring Business Elements.	50
	Creating a Business Element.	50
	Exercise: Creating additional Business Elements.	51
3.7	Creating and configuring a Business View.	53
	To create a Business View.	53
4	Using the Business View Manager.	54
4.1	Business View Manager overview.	54
4.2	Working with the Business View Manager.	54
	Navigating the Welcome to Business View Manager dialog box.	56
	Navigating within the Business View Manager.	57
4.3	Exporting and importing Business Views.	59
	To export Business Views.	59
	To import Business Views.	61
4.4	Using the Repository Explorer.	62
	Accessing the SAP BusinessObjects repository	63
	Adding folders to the repository.	64
	Deleting items from the repository.	65
	Installing sample repository objects.	66
4.5	SAP BusinessObjects repository security model.	67
	Applying security settings to folders in the repository.	68
	Viewing folder rights.	69
4.6	Creating a list of values.	69
	To create a list of values.	70
	To schedule a list of values.	72
4.7	Changing the Business Views Used by a List of Values.	73
	To change Business Views.	74
5	Creating Data Connections.	77
5.1	Data Connections overview.	77
5.2	Working with Data Connections.	77
	Data sources.	77

	Testing and verifying data connectivity and object dependency.	82
	Showing dependent and referenced objects.	82
	Modifying a Data Connection.	82
	Editing rights for a Data Connection.	85
6	Creating Dynamic Data Connections.	87
6.1	Dynamic Data Connections overview.	87
	Why use a Dynamic Data Connection?.	87
6.2	Working with Dynamic Data Connections.	88
	Verifying object dependency.	89
	Showing dependent and referenced objects.	89
	Modifying a Dynamic Data Connection.	89
	Editing rights for a Dynamic Data Connection.	94
7	Managing Data Foundations.	97
7.1	Data Foundations overview.	97
7.2	Working with Data Foundations.	97
	Creating a new Data Foundation.	97
	Verifying object dependency and verifying databases.	99
	Showing dependent and referenced objects.	100
	Modifying a Data Foundation.	100
	Linking tables.	100
	Inserting data tables	110
	Setting the Database Explorer Options.	113
	Inserting a formula.	115
	Inserting an SQL expression.	124
	Inserting a parameter.	127
	Inserting a filter.	132
	Importing a custom function.	136
	Using the Referenced Data Connections window.	139
	Using the Property Browser.	139
	Using the Object Explorer.	140
	Editing rights for a Data Foundation.	141
8	Managing Business Elements.	144
8.1	Business Elements overview.	144
8.2	Working with Business Elements.	144
	Verifying object dependency.	145
	Showing dependent and referenced objects.	145
	Modifying a Business Element.	146
	Inserting a filter.	147
	Inserting a parameter.	151

	Using the Referenced Data Foundation window.	153
	Using the Property Browser.	153
	Using the Object Explorer.	154
	Editing rights for a Business Element.	155
8.3	Using the Business Element Wizard.	157
	Creating and specifying multiple Business Elements	157
	Specifying a remaining option in the Wizard.	158
9	Dynamic Prompts and Cascading Lists of Values.	160
9.1	Parameters and prompts.	160
9.2	Dynamic prompts overview.	160
9.3	Supported components.	161
9.4	Understanding lists of values.	162
9.5	Lists of values and prompt groups.	162
	List-of-values types.	162
	Determining which list-of-values type to use.	163
	Using Separate Value and Description Fields.	164
	Null Handling.	165
9.6	Using dynamic prompts and cascading lists of values in Business Elements and Data Foundations	165
9.7	Using Business View Manager to manage prompting repository objects.	165
	Managing lists of values in Business View Manager.	165
	Managing security for lists of values.	169
	Scheduling lists of values in Business View Manager.	169
	Managing prompt groups in Business View Manager.	171
9.8	Best practices for prompting.	172
9.9	Converting unmanaged reports to managed reports.	173
9.10	Deploying managed reports with dynamic prompts.	173
10	Managing Business Views.	174
10.1	Business Views overview.	174
10.2	Working with Business Views.	174
	Importing and exporting Business Views.	174
	Showing dependent and referenced objects.	175
	Modifying a Business View.	175
	Overriding the linking in a Data Foundation.	177
	Editing rights for a Business View.	183
11	Business Views Security Concepts.	187
11.1	Security overview.	187
11.2	Business Views object rights overview.	187
11.3	Using the Edit Rights dialog box.	188
	Applying security settings.	188

11.4	Object and folder rights.	191
	Viewing Business Views object rights.	191
	Using inheritance to your advantage.	192
11.5	Security deployment.	193
11.6	Security considerations.	195
	Root folder.	195
	Reports based on Business Views.	195
	Importing and exporting Business Views.	196
12	User Scenarios.	197
12.1	User scenarios overview.	197
12.2	Data integration.	197
	Simplifying existing infrastructure through data abstraction.	198
12.3	Multiple data sources.	198
	Reporting off multiple data sources.	198
	Specifying locale: a global, multi-lingual deployment.	199
12.4	Security applications.	199
	Applying row security.	199
	Applying column security.	200
	Setting object security.	200

1 Introduction to Business Views

1.1 About this documentation

This help provides you with information and procedures that cover a wide range of administrative tasks. Procedures are provided for common tasks, including the use of the Business View Manager. Conceptual information and technical details are provided for all advanced topics such as the Business Views architecture, and security and deployment recommendations.

1.1.1 What is Business Views?

Business Views is a flexible and reliable multi-tier system that enables companies to build detailed and specific Business Views objects that help report designers and end users access the information they require.

Using Business Views, you can integrate data from disparate sources. You can also bring together data from multiple data collection platforms and application boundaries so that the differences in data resolution, coverage, and structure between collection methods are eliminated.

Business Views includes the Business View Manager, a thick-client application. This designer enables administrators to create and modify Business Views objects:

- Data Connections
- Dynamic Data Connections
- Data Foundations
- Business Elements
- Business Views

1.1.2 Who should read this guide

This help is intended for system administrators who are responsible for configuring, managing, and maintaining a Business Views installation. Familiarity with your operating system and your network environment is certainly beneficial, as is a general understanding of web server management, scripting technologies, and general security concepts. However, in catering to all levels of administrative experience, this guide aims to provide sufficient background and conceptual information to clarify all administrative tasks and features.

Business Views administrators should also be familiar with SAP BusinessObjects Business Intelligence platform, as Business Views is integrated with and is a part of the platform. For more information, consult the *SAP BusinessObjects Business Intelligence platform Master Guide*, and the *SAP BusinessObjects Business Intelligence platform Administrator Guide*.

Working and conceptual knowledge of SAP Crystal Reports and general reporting concepts will also be beneficial; for more information on Crystal Reports, see the *SAP Crystal Reports User's Guide*.

2 Business Views Architecture

This section provides a detailed overview of Business Views and its architecture, and it describes the different objects that you can create and modify using the Business View Manager.

2.1 Business Views overview

Business Views is a multi-tier system that enables companies to build comprehensive and specific Business Views objects that help report designers and end users access the information they require.

Business Views are useful as companies engage in data management to collect and organize operational data into databases. Companies often use data integration technology to build data warehouses or data marts to cleanse, aggregate, and store data for analytic use. However, data warehouses must serve the lowest common denominator and can—but often do not—provide the various views and granularity that different business units require in order to understand and analyze their businesses. In addition, these warehouses may not provide the level of detail necessary for some analytic operations.

Using Business Views, you can integrate data from disparate sources. You can bring together data from multiple data collection platforms across application boundaries so that the differences in data resolution, coverage, and structure between collection methods are eliminated.

In addition, Business Views enables you to add the necessary business context to these data islands as you link them into a single organized Business View for your organization. This view is more than just an integrated network of data. It can include consistent definitions of corporate hierarchy or customer information, and provide a variety of detailed or summarized viewpoints for the various information consumers in your business.

Administrators use the Business View Manager—a thick-client designer, which runs as a Microsoft Windows application. This designer provides a wide range of capabilities for creating Data Connections, Dynamic Data Connections, Data Foundations, Business Elements, and Business Views. The Business View Manager enables you to design relational views of information. This designer also enables you to set detailed column and row-level security for various objects in your report.

Note

Business Views objects consists of Data Connections, Dynamic Data Connections, Data Foundations, Business Elements, and Business Views.

Business Views supports the transition from one view to another at runtime, thereby enabling powerful analytics and rich information presentation through a broad range of SAP BusinessObjects client tools. The Business View Manager is designed to help organizations consolidate data by dynamically mapping back-end data into Business Views. These Business Views can then be secured at a granular level by administrators and used as the basis for reporting, analysis, and information delivery processes. This flexible approach enables organizations to use the data repository of their choice—including multiple repositories—for their views and to leverage all of these Business Views objects within their SAP BusinessObjects environment.

ⓘ Note

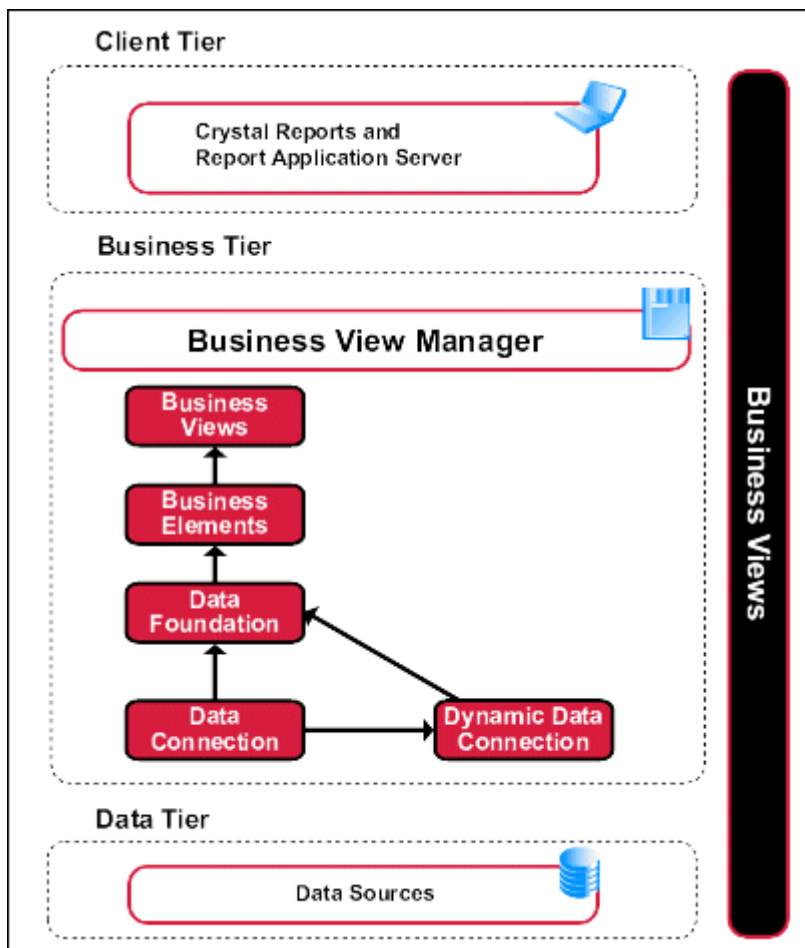
Business Views is intended for administrators who are familiar with SAP BusinessObjects Business Intelligence platform. For more information, see the *SAP BusinessObjects Business Intelligence platform Administrator Guide*.

ⓘ Note

Business Views is similar in concept to Crystal Dictionary files and Info Views, but provides numerous additional features and is designed to integrate seamlessly with SAP BusinessObjects Business Intelligence platform.

2.2 Architecture overview and diagram

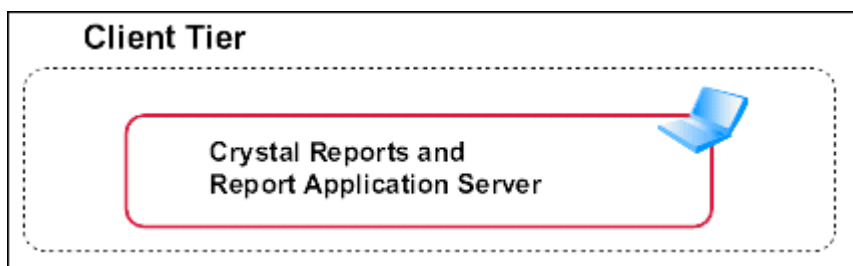
Business Views is a multi-tier system. The following diagram illustrates how each of the various components fits within this system.



Business Views is comprised of three tiers: the client tier, the business tier, and the data tier.

- [Client tier \[page 11\]](#)
The client tier consists of any SAP BusinessObjects client application that accesses the Business Views that are stored and organized in the business tier.
- [Business tier \[page 11\]](#)
Business Views objects—a collection that includes Data Connections, Dynamic Data Connections, Data Foundations, Business Elements, and Business Views—are a part of this tier. Using the Business View Manager, you specify the settings and conditions for the various objects so that the client tier has access only to specific information from the data tier.
- [Data tier \[page 14\]](#)
The data tier consists of data sources, such as multiple databases on different machines that provide the data for the business tier.

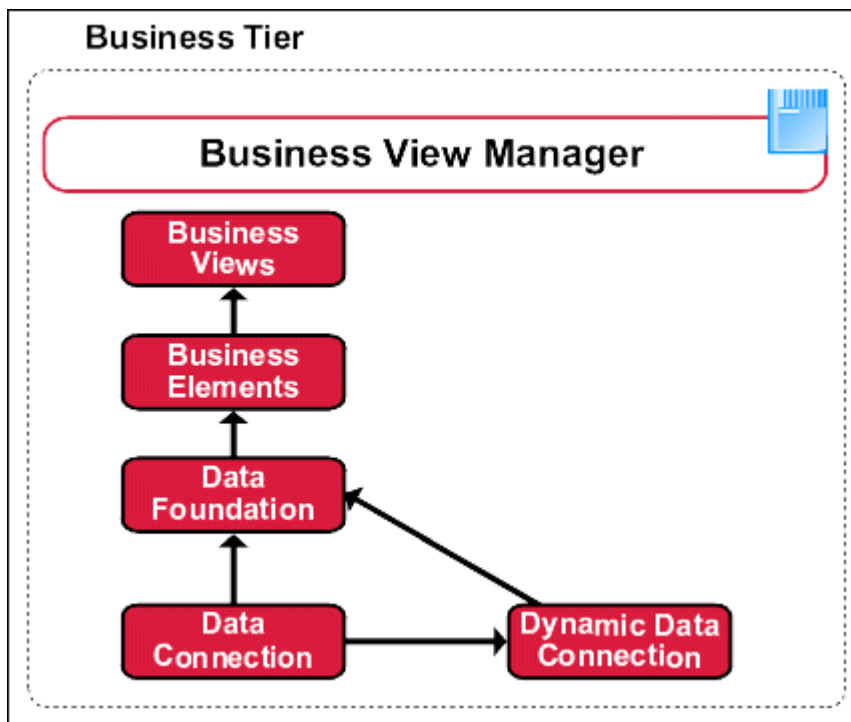
2.3 Client tier



The client tier includes SAP BusinessObjects applications (such as Crystal Reports and the Report Application Server) that access the Business Views that are stored and organized in the business tier. Business Views administrators control and define security and access to specific objects within the various Business Views.

Report designers using Crystal Reports see only the tables and fields that they have access to, as defined within the specific Business View. For example, a report designer in the company's sales department can access only regional sales data, and thus, design a report with sales-specific information, even though the data store also contains employee-specific information. Thus, depending on how the report is created and designed, a manager running the same report (or another report) has access to the additional information on employees within the company. All of the security and access information is handled by the Business View Manager.

2.4 Business tier



The business tier is the primary tier of the Business Views system. Administrators access this tier through the Business View Manager; they use this designer to create and modify Data Connections, Dynamic Data Connections, Data Foundations, Business Elements, and Business Views.

2.4.1 Business View Manager

The Business View Manager enables you to create and modify Business Views and the various objects that make up these Business Views. The administrator uses this designer to specify different Data Connections, to set security, and to control access to the data found within the different data sources in the data tier. This thick-client designer is the only part of Business Views that administrators interact with directly. End users access the data specified within the Business View Manager through their SAP BusinessObjects client application, such as SAP Crystal Reports (via the SAP BusinessObjects repository) or through the Report Application Server.

2.4.2 Business Views objects

Business Views objects include the following: Data Connections, Dynamic Data Connections, Data Foundations, Business Elements, and Business Views.

2.4.3 Data Connection

Data Connections specify and define the data sources for a Business View; they define how these data sources are made available to users. Thus, a Data Connection is a connection object, and an administrator can apply security to this object. Each Data Connection contains information that describes the physical data source, such as the server and data being accessed, the logon credentials, and the type of server being accessed.

2.4.4 Dynamic Data Connection

A Dynamic Data Connection is a collection of pointers to various Data Connections. An administrator or user is able to select which Data Connection to use through a parameter.

A typical scenario involves the migration of data from a development system to a test system, and finally, to a production system. In this scenario, a report is run against a development system, and then, when the data is migrated to a test system, the same report is run against the test system's data. The only change required is that the Dynamic Data Connection's settings must be updated so that it points to the test system's Data Connection. Finally, when the test system's data is migrated to the production system, the same report can again be run against the production system.

Note

When users refresh reports that are based on a Dynamic Data Connection, they are prompted to specify which of the available Data Connections to use.

2.4.5 Data Foundation

A Data Foundation consists of collections of tables and fields. Default table joins are defined at this level. Administrators can secure Data Foundations using the standard SAP BusinessObjects Business Intelligence platform security model, which includes View and Edit rights. The primary use of Data Foundations is for data abstraction: administrators control which tables and fields users can or cannot access when these users are designing or viewing a report.

2.4.6 Business Element

A Business Element is roughly equivalent to an OLAP dimension or logical view; that is, this object is a logically related collection of data fields that are based on a Data Foundation. These fields can be organized into a hierarchical structure within the Business Element. The most common example is a hierarchical structure that contains the following fields: Country, State or Province, and City. Business Fields in Business Elements can be aliased to support data abstraction and to facilitate report design. Administrators can secure Business Elements through the standard View and Edit rights.

2.4.7 Business View

A Business View is a logical collection of Business Elements. Users see Business Views as abstract database connections, and the contained Business Elements as virtual tables that, in turn, contain Business Fields. Administrators can secure Business Views through the standard View and Edit rights.

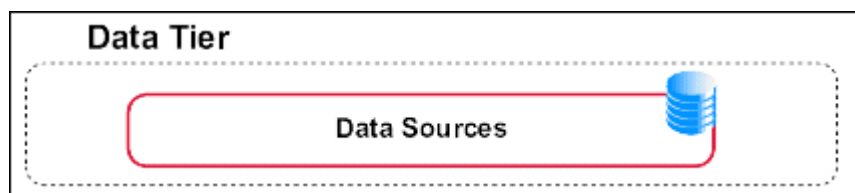
A Business View consists of the following objects:

- One or more Data Connections
- One optional Dynamic Data Connection
- One Data Foundation
- One or more Business Elements

Note

End users can access Business Views through applications such as SAP Crystal Reports and the Report Application Server.

2.5 Data tier



The data tier consists of multiple databases that contain the data used in the different views and objects, which are used in reports. Business Views supports a wide range of corporate databases. See the `release.html` file included with your product distribution for a complete list of tested database software and version requirements.

2.6 Information flow

This section describes a typical Business Views usage scenario. It demonstrates how report-processing is performed and how security access is defined and applied for the data in a report.

When creating Business Views, it is important to understand how each of the component objects relates to one another. The objects that make up a Business View are:

- Data Connections
- Dynamic Data Connections

- Data Foundations
- Business Elements

A Business View requires at least one of each type of object, except for Dynamic Data Connections, which are optional.

The objects that make up a Business View build on each other in a hierarchical fashion. You must first create a Data Connection or Dynamic Data Connection before you can create a Data Foundation. Then, after you have a Data Foundation, you can create a Business Element. After you finish creating a Business Element, you can then create a Business View—a view that report designers have access to.

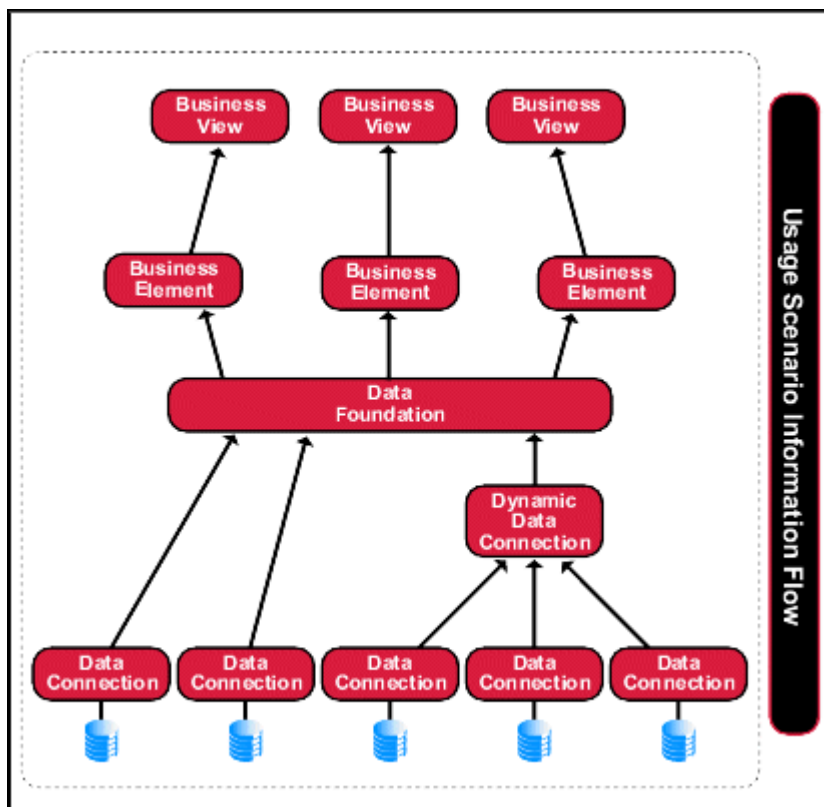
Ultimately, the Business View Manager allows administrators to integrate complex and disparate data sources seamlessly, effectively removing data silos in an organization. That is, using the Business View Manager, administrators can take data from several different sources and abstract it so that report designers see a single, unified, and logically organized data source.

2.6.1 Usage scenario

This section employs a hypothetical usage scenario to illustrate how the Business View Manager can be used to create Business Views from which to base a report.

In this scenario, a company has its data stored in three different databases. Its personnel data is stored in a Microsoft SQL Server. Its product information is stored in a DB2. Its sales data is stored in three different ORACLE databases: one for development, one for testing, and one for production.

The report designers want to create reports that show the performance of Sales representatives, including personnel information and product data. Users must be able to run reports based on any of the three ORACLE sales databases. Additionally, the company wants to specialize the reports for different regions. The East, Central, and West sales managers each want to receive the same reports, but want to see data from only their regions. All of the managers also have unique terminology preferences, and want their reports to reflect these preferences. To incorporate the different data sources and the three different sales region preferences, three different Business Views must be created.



2.6.2 Data Connection layer

The Data Connection layer is composed of one or more Data Connections. Dynamic Data Connections can also be part of the Data Connection layer, but are not integral to the creation of a Business View.

2.6.2.1 Data Connections

The first step in the process of building a Business View is to specify data sources for the Business View by creating Data Connections. Each Data Connection object connects to a single data source, such as a database, a data mart, a spreadsheet, and so on. In this scenario you will create five different data connections, one for each data storage system.

Because all objects in a Business View are located inside the SAP BusinessObjects repository, SAP BusinessObjects BI platform security applies to the objects. Consequently, as a connection property, you can set which groups have access to a data source through a specific Data Connection. (Users in the groups you specify are considered authenticated users.) For example, you may want only report designers and managers at a certain level to have Data Access rights to the Data Connection for the SQL Server database. Report designers must have access to the Data Connection in order to be able to create reports based on it. In this case, you can create two groups in SAP BusinessObjects BI platform, one for report designers and one for managers, and grant only those two groups Data Access rights for the Data Connection in the Business View

Manager. Note that by default the "Administrators" and "Everyone" groups have full rights to each object you create in the Business View Manager.

- For details about creating groups in SAP BusinessObjects BI platform, refer to the *SAP BusinessObjects Business Intelligence platform Administrator Guide*.
- For information on setting rights in the Business View Manager, see the related topic.

When you add a Data Connection, you can choose to store a user name and password for the Data Connection to use when accessing the data source. These credentials are stored in the SAP BusinessObjects repository and are used each time an authenticated user is designing or viewing a report based on a Business View that uses that Data Connection. If you prefer to have users log on to data sources individually, do not store a user name and password in the SAP BusinessObjects repository.

In the Property Browser for each Data Connection, you can change the name of the Data Connection. Taking the time to name each Data Connection properly can greatly enhance usability because Data Connection names are directly visible to report designers and end-users when reporting off a Business View that uses a Dynamic Data Connection. Report designers and end-users will be prompted to choose between the different Data Connections.

By default, at the time of creation, Data Connections are named Data Connection1, Data Connection2, Data Connection3, and so on. As in the usage scenario, you would name each Data Connection descriptively:

Data Connection		Name
1	Personnel data in SQL Server	Personnel
2	Product data in DB2	Product Info
3	Sales data in ORACLE (development database)	Sales Development
4	Sales data in ORACLE (testing database)	Sales Testing
5	Sales data in ORACLE (production database)	Sales Production

Related Information

[Business Views Security Concepts \[page 187\]](#)

2.6.2.2 Dynamic Data Connections

Once you have created two or more Data Connections, you can choose to create Dynamic Data Connections. These are pointers that allow administrators and/or users to choose between different data sources in the form of Data Connections. Note that each of the data sources that a Dynamic Data Connection points to must have similar schemas. Any tables, fields, stored procedures, stored procedure parameters, and other objects that are in the data sources must have identical structures, names, and content types.

When a Business View designer creates a Data Foundation that is based on a Dynamic Data Connection, this user is prompted to specify which Data Connection to use. Similarly, when a user refreshes a report based on a Dynamic Data Connection, that user is prompted to specify which Data Connection to use.

In the usage scenario, you would create one Dynamic Data Connection composed of the three ORACLE sales databases: Sales Development, Sales Testing, and Sales Production. A typical name for such a Dynamic Data Connection would be Dynamic Sales.

2.6.3 Data Foundation layer

After you have created the Data Connection layer (using Data Connections and possibly using Dynamic Data Connections), the next step is to create a Data Foundation. A Data Foundation is a component used for data access management, where you collect a list of data fields to be made available for use as Business Fields in Business Elements. A Data Foundation is an abstraction layer into which you can insert and join several types of objects (from different Data Connections):

- Table objects
 - Data tables
 - Views
 - Stored procedures
 - SQL command objects
- Formulas
- SQL expressions
- Filters
- Parameters
- Custom Functions

Note that views, stored procedures, and SQL command objects all appear as tables once you add them to the Data Foundation.

When you create a Data Foundation, you need to specify your data source, whether it be one or more Data Connections, or one or more Dynamic Data Connections. You can have any number or combination of Data Connections and Dynamic Data Connections for your Data Foundation. In the case of the usage scenario, the Data Foundation would be composed of two Data Connections—Personnel and Product Info—and one Dynamic Data Connection—Dynamic Sales.

The items you choose to include in the Data Foundation determine what fields will later be available for report designers to use when creating reports. That is, you, the administrator, can control which tables and columns users can or cannot access.

2.6.3.1 Using formulas and SQL expressions

At the Data Foundation level, you can use formulas to create data fields that will then be available to users working at the Business Element level (a Business Element is a collection of fields that are based on a Data Foundation). For example, the existing Data from the Sales Data Connection may list an employee's sales quota and actual sales, but not the percentage of the sales quota that employee reached. In this case, you could

write a formula that calculates this percentage. Later, at the Business Element layer, you could make this field available to report designers, but choose to keep the sales quota and actual sales fields hidden. This is a simple example of how you can use the Business View Manager to abstract and control user access to data.

SQL expressions are similar to formulas, but they are written in Structured Query Language (SQL). They are useful in optimizing report performance because the tasks they execute are always performed on the database server (as opposed to formulas, which are typically executed on the local machine).

2.6.3.2 Using filters

Filters provide row-level security for Business Views. You create your own filters to apply to your Data Foundation. These filters allow you to reference fields, formulas, SQL expressions, parameters, and other filters. You can use Boolean operators to create Business Filters that restrict access to certain information for specific users or groups. After you create a Business Filter, you can also assign to which user or group you want to apply this filter.

2.6.3.3 Using parameters

Parameters prompt the user of a report to enter information. Think of a parameter as a question that the user needs to answer before the report is generated from the Business View information. The information users enter—or the way they respond—determines what appears in the report. For example, in a report used by salespeople, there might be a parameter that asks the user to choose a region. The report would return the results for the specific region chosen by the user, instead of returning the results for all of the regions.

2.6.3.4 Using custom functions

Custom functions are procedures you create in SAP Crystal Reports to evaluate, to make calculations on, or to transform data. When you use a custom function in a formula, all of the operations in its definition are performed without the need to specify them individually in the formula itself. Thus, a custom function provides a way for you to share and reuse formula logic and makes it easier and less time-consuming for you and your users to create different Business Views objects and reports.

You create custom functions using SAP Crystal Reports and then save them in the repository. In the Business View Manager, you reference custom functions, and then include them in your formulas.

Note

A custom function cannot be placed directly into a Business Element; it must be used as a part of a formula at the Data Foundation level.

2.6.4 Business Element layer

A Business Element is an object that allows you to shape your data fields from a Data Foundation into components that make sense from a business perspective. It is important to understand that the organization of Business Fields in Business Elements does not need to reflect the shape of the underlying tables in the Data Foundation.

Whereas the Data Foundation layer is focused on the physical layout of the data sources, the Business Element layer allows you to create a hierarchical information landscape. You can reorganize the tables, fields, formulas, and SQL expressions from the Data Foundation into a logical view with multiple levels. A common example of a Business Element is a hierarchical structure that contains the following fields: Country, State or Province, and City.

When you insert data fields from the tables, formulas, and SQL expressions that you created in a Data Foundation into a Business Element, each of the items that you add appears as a Business Field. Using the Field Structure tab, you can restructure these Business Fields into a hierarchy.

In a Business Element, you can also create an alias for, and add a description to each of the Business Fields. As in the example, if the three sales managers for the East, Central, and West regions wanted to see similar reports but with different terminology for some of the fields, you could create three different Business Elements, each with the same structure and fields, but with different aliases for the field names.

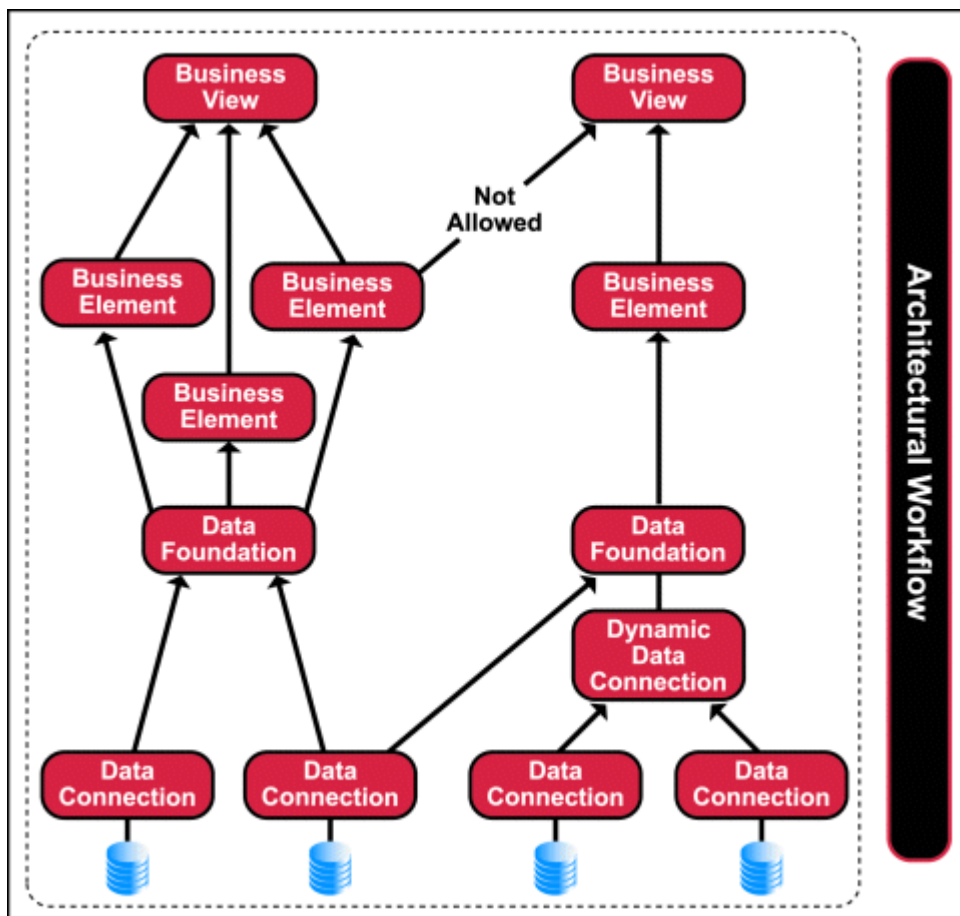
You can set rights for each Business Element so that certain groups and users do or do not have View rights for the object. Users who do not have View rights for that Business Element will not be able to create reports based on it. Column-level security can also be applied to Business Fields. If column-level security is applied, specified column contents are converted to null values during execution.

2.6.5 Business View layer

Once you have created one or more Business Elements, you can create a Business View. A Business View is a logical collection of Business Elements. Users see Business Views as an abstract database connection, and the contained Business Elements as virtual tables that contain Business Fields. End users access Business Views through the Ad Hoc application (or other applications designed using the Report Application Server SDK) and client applications such as SAP Crystal Reports. Administrators can secure Business Views through the standard View and Edit rights.

2.7 Architectural workflow

While the necessary, ordered structure of the Business Views objects allows for flexibility in the access, integration, and organization of data from different sources, there are limitations set by the relationships among these various components.



A Data Connection, once made, accesses and carries information specific to the data source to which it is linked. To access information from another data source, you must create another Data Connection or modify the existing one.

As an alternative, however, you can also make use of a Dynamic Data Connection, a collection of pointers to any number of individual Data Connections.

Depending on the user's needs, a Dynamic Data Connection may be preferable to using a single Data Connection.

Companies run the same reports based on data from different sources. Rather than recreating separate reports for each source, a Dynamic Data Connection allows you to point to a different Data Connection, obtain the information, and generate the same report based on the data from that source.

Once a connection is made, you then manage the information in a Data Foundation, an abstraction layer where the different objects (tables, fields, and so on) obtained from a data source can be added or joined. The items that you choose to include in a Data Foundation and the relationships you specify between these items determine what fields will later be available for report designers to use when creating reports.

You construct Business Elements from the various objects (tables, parameters, filters, and so on) situated in a Data Foundation. Because the information in a Business Element is defined by the information contained within a Data Foundation, it is important to remember that Business Elements are Foundation-specific. That is, you cannot create a Business Element with information derived from more than one Data Foundation. This constraint stems from the nature of the relationship between the Business Element and the Data Foundation.

At the Data Foundation level, users acquire and set specifications to the information from a data source. At the Business Element level, users take this information and arrange it into a structure that makes sense from a business perspective (often involving hierarchical levels and landscapes of data). In effect, the information in a Business Element is dependent on the information controlled by its parent Data Foundation.

Once you have created one or more Business Elements, you create a Business View. A Business View is a collection of Business Elements; it provides the highest level of data abstraction for end users. Users see Business Views as an abstract database connection, and the Business Elements contained within as virtual tables.

You can have a number of Business Elements that can be included in a single Business View. Similarly, you can have numerous Business Views that are based on a single Business Element. The only restriction between the components is from the relationship between the Business Element and the Data Foundation. Since Business Elements contain information provided by the parent Data Foundation, Business Views can include only Business Elements which stem from one Data Foundation. That is, a Business View cannot be composed of Business Elements that come from different Data Foundations.

3 Quick Start: Creating a Business View

This section contains a number of tutorials designed to teach new users how to create a Business View. The tutorials guide you through several basic steps to create a Business View using the Business View Manager.

3.1 Quick Start overview

Using Business Views, you can integrate data from disparate sources. You can bring together data from multiple data collection platforms across applications, to eliminate differences in data resolution, coverage, and structure between collection methods.

Administrators use the Business View Manager, a thick-client designer. This designer is a Microsoft Windows application that provides features, which enable you to design relational views of information. This designer also enables you to set detailed column and row-level security for various objects in your report.

This section contains a number of tutorials designed to teach new users how to create a Business View. Employing data from the Xtreme Sample Database (shipped with the product), the tutorials guide you through several basic steps to create a Business View using the Business View Manager:

- [Before you begin \[page 23\]](#)
- [Creating and configuring a Data Connection \[page 27\]](#)
- [Creating and configuring Dynamic Data Connections \[page 35\]](#)
- [Creating and configuring a Data Foundation \[page 38\]](#)
- [Creating and configuring Business Elements \[page 50\]](#)
- [Creating and configuring a Business View \[page 53\]](#)

3.2 Before you begin

This section describes the sample data and conventions used in the tutorials and the tutorial scenario. It explains what a Business View is and how to log on to the Business View Manager. To avoid confusion, it is best to follow this guide step-by-step without skipping over any sections.

3.2.1 Sample data - Xtreme.mdb

Business Views comes with Xtreme.mdb, a sample Microsoft Access database that you can use when learning how to operate the Business View Manager. All of the necessary drivers are included in the installation of Business Views.

Xtreme.mdb is a database that contains data for Xtreme Mountain Bikes, a fictitious manufacturer of mountain bikes and accessories.

The database includes these tables:

- Credit
Information from customer credit memos, such as credit authorization IDs and amounts.
- Customer
Data for the customers served by the company.
- Employee
Company-oriented data for the employees of Xtreme Mountain Bikes.
- Employee Addresses
Personal data for Xtreme Mountain Bikes employees.
- Financials
Financial data for Xtreme Mountain Bikes.
- Orders
Identity and tracking data for orders.
- Orders Detail
Line item data for orders.
- Product
Descriptive data for Xtreme Mountain Bikes products.
- Product Type
Category data for Xtreme Mountain Bikes products, including product pictures.
- Purchases
Identity and tracking data for product purchases.
- Supplier
Data for suppliers who serve Xtreme Mountain Bikes.
- Xtreme Info
Company data for the Xtreme Mountain Bikes company, including the company logo.

Xtreme.mdb also includes:

- List Totals (a Select query)
- Top Customers (a Select query)

3.2.2 Conventions

This section is organized into a number of tutorials and exercises. To create the sample Business View described in this section, you must complete each of the tutorials and exercises in succession.

Note

This tutorial was designed using Microsoft Windows 2000. If you are using a different platform, the screen shots may vary slightly.

3.2.3 Tutorial scenario

Your corporation uses a development, test, and production system model for housing its data. The data is initially stored on a development system, and then later migrated to a test system. After extensive testing, the data in the test system is ready to be used in the production system.

In this tutorial you will create these objects:

- Three Data Connections:
 - Xtreme Development
 - Xtreme QA
 - Xtreme Production
- One Dynamic Data Connection:
 - Xtreme Dynamic Connection
- One Data Foundation:
 - Xtreme Foundation
- Six Business Elements:
 - Customer
 - Employee
 - Orders
 - Orders Detail
 - Product
 - Supplier
- One Business View:
 - Xtreme Business View

3.2.4 What is a Business View?

A Business View is a hierarchical collection of several objects:

- Data Connections
- Dynamic Data Connections
- Data Foundations
- Business Elements

When you create Business Views, it is important to understand how the component objects relate to one another. Each object must be created in a specific order. For example, you must first create a Data Connection or Dynamic Data Connection before you can create a Data Foundation. Then, after you have a Data Foundation, you can create a Business Element. After you finish creating a Business Element, you can then create a Business View.

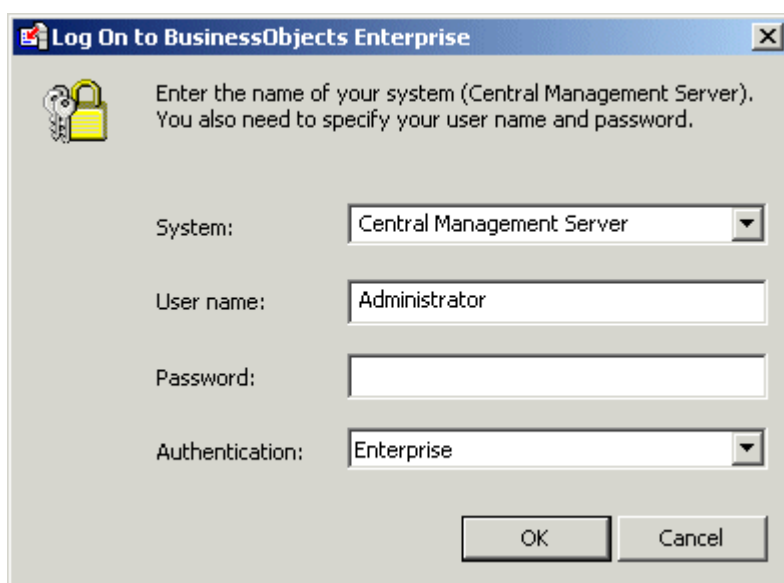
3.2.5 To log on to the Business View Manager

Every time you use the Business View Manager, you need to log on with the proper credentials. To log on to the Business View Manager, you must provide a Central Management Server (CMS) name, a User name, and Password for the appropriate CMS.

In this tutorial you will log on to the Business View Manager.

1. In Windows, click *Start > Programs > SAP BusinessObjects BI platform > SAP BusinessObjects BI platform Client Tools > Business View Manager*.

The Log On dialog box appears.

The image shows a Windows-style dialog box titled "Log On to BusinessObjects Enterprise". It contains a yellow padlock icon and the text: "Enter the name of your system (Central Management Server). You also need to specify your user name and password." Below this, there are four input fields: "System:" with a dropdown menu showing "Central Management Server", "User name:" with a text box containing "Administrator", "Password:" with an empty text box, and "Authentication:" with a dropdown menu showing "Enterprise". At the bottom right are "OK" and "Cancel" buttons.

2. Select the type of authentication in the *Authentication* list.
3. In the *System* list, select or type the appropriate CMS name.

Note

If SAP BusinessObjects Business Intelligence platform is installed on your local machine, the name of your CMS is the same as your machine name.

Note

If SAP BusinessObjects Business Intelligence platform is not installed locally, provide the name of the machine where the CMS is installed.

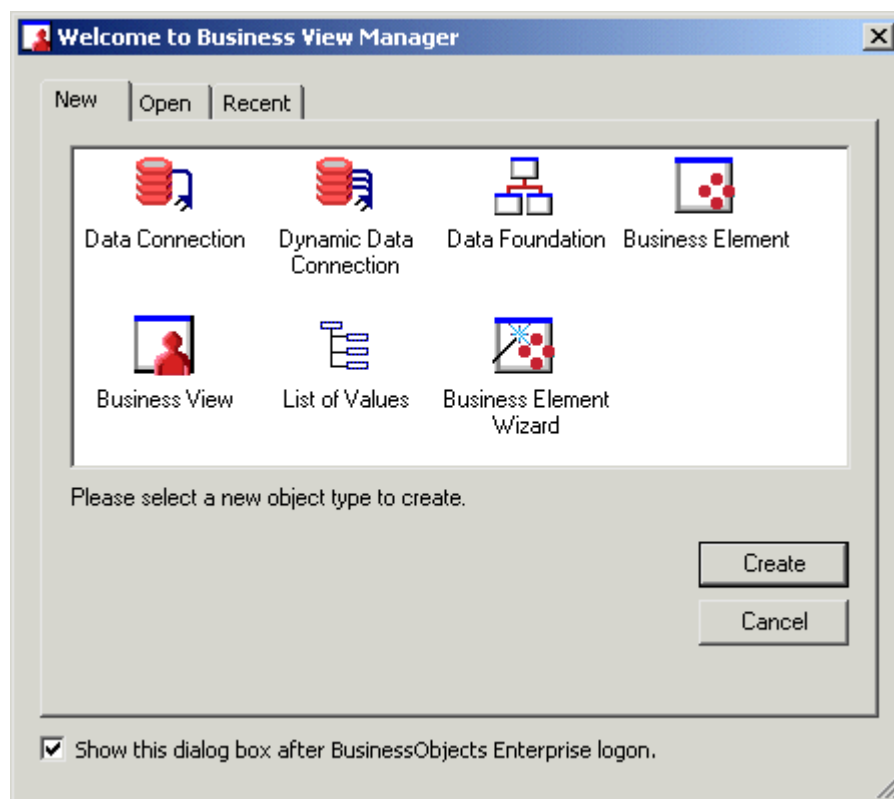
4. Type your *User name* and *Password*.

Note

If SAP BusinessObjects Business Intelligence platform is installed on your local machine, the default User name is administrator, without a password.

5. Click *OK*.

The "Welcome to Business View Manager" dialog box appears.



3.3 Creating and configuring a Data Connection

In the tutorials and exercise in this section, you will complete these tasks:

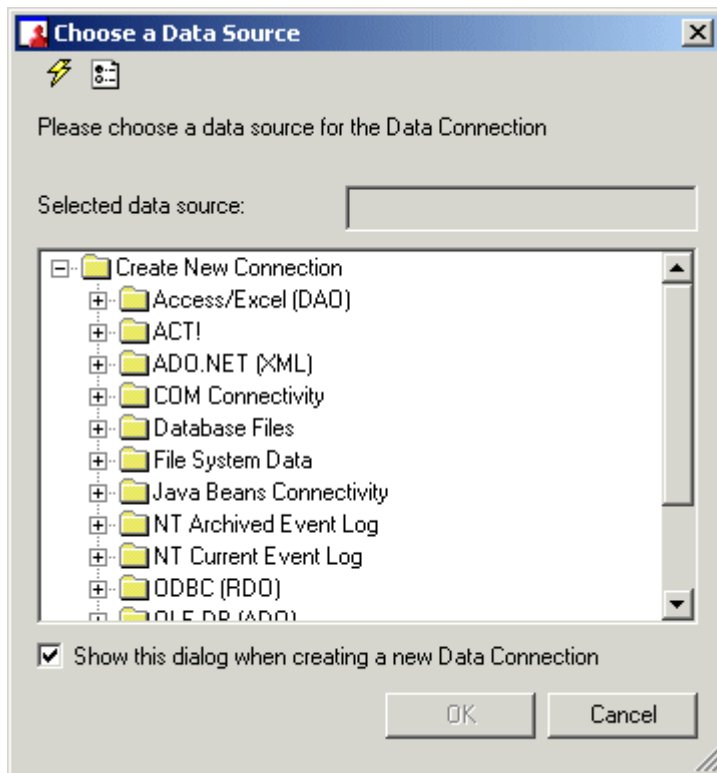
- Create three different Data Connections to the Xtreme Sample Database.
- Set passwords for each Data Connection.
- Save and name each Data Connection.
- Modify the properties of each Data Connection.

3.3.1 To create a Data Connection

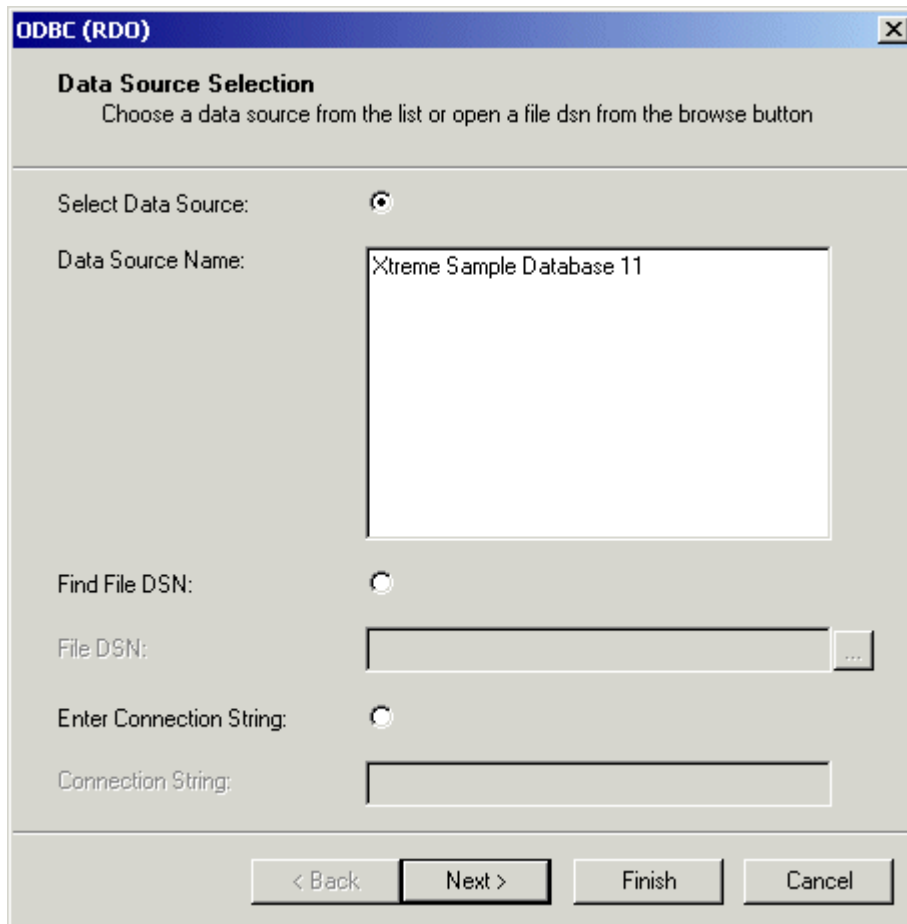
To connect to a data source, you must create a Data Connection.

1. Open a new Data Connection through one of the following ways:
 - If the "Welcome to Business View Manager" dialog box is open, click Data Connection, and then click Create.

- On the File menu, select New, and then click Data Connection. The "Choose a Data Source" dialog box appears.



2. Select ODBC (RDO) as the connection type, by double-clicking the [ODBC \(RDO\)](#) folder. The "Data Source Selection" dialog box appears.



3. In the Data Source Name list, select *Xtreme Sample Database XI*.
4. Click *Next*.

The "Connection Information" dialog box appears.

ODBC (RDO)

Connection Information
Provide necessary information to log on to the chosen data source

Data Source: Xtreme Sample Database 11

User ID:

Password:

< Back Next > Finish Cancel

5. In the "Connection Information" dialog box, you can specify the credentials that you want to use to log on to the data source.

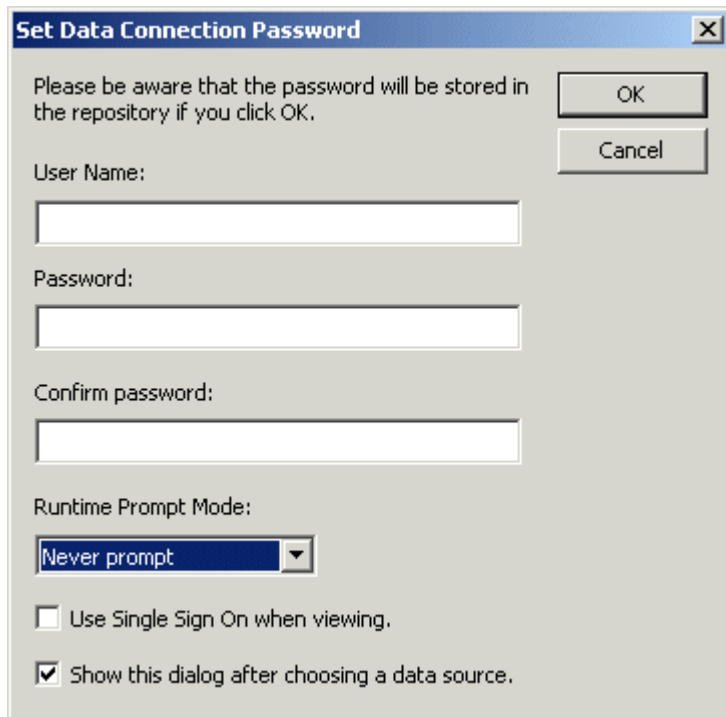
In this case, the Xtreme Sample Database XI does not require logon credentials, so leave the fields blank.

6. Click *Finish*.

You return to the "Choose a Data Source" dialog box.

7. Ensure that *Xtreme Sample Database XI* is selected, and then click *OK*.

The "Set Data Connection Password" dialog box appears.



Set Data Connection Password

Please be aware that the password will be stored in the repository if you click OK.

OK

Cancel

User Name:

Password:

Confirm password:

Runtime Prompt Mode:

Never prompt

☐ Use Single Sign On when viewing.

☒ Show this dialog after choosing a data source.

3.3.2 Setting a Data Connection password

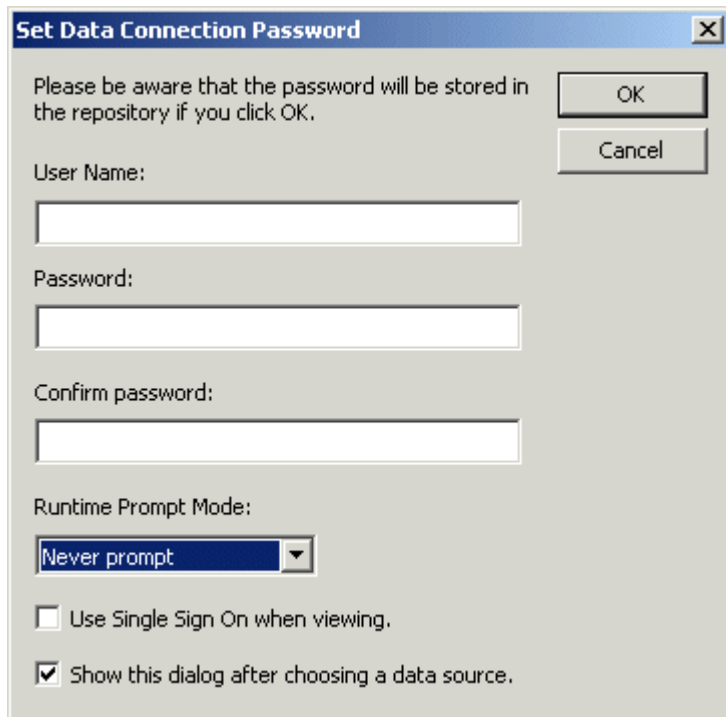
When you add a Data Connection, you can choose to store a user name and password. The Data Connection uses this information to access the data source. These credentials are stored in the SAP BusinessObjects repository. They are used each time an authenticated user designs or views a report that is based off a Business View that uses that Data Connection. If you choose to store data source logon credentials in the repository, you should set the Runtime Prompt Mode to Never prompt, so that authenticated users will not be asked to specify credentials.

If you prefer to have users log on to data sources individually, do not store a user name and password in the repository. Also, set the Runtime Prompt Mode to Always prompt, so that users are asked to specify credentials each time they design or run a report off a Business View that uses that Data Connection.

In this tutorial, you store a blank user name and password for the Data Connection object, and set the Runtime Prompt Mode to Never prompt, so that users are not asked to specify logon credentials.

3.3.2.1 To set the Data Connection password

1. In the "Set Data Connection Password" dialog box, leave each of the *User Name*, *Password*, and *Confirm password* fields blank.



Set Data Connection Password

Please be aware that the password will be stored in the repository if you click OK.

OK

Cancel

User Name:

Password:

Confirm password:

Runtime Prompt Mode:

Never prompt

☐ Use Single Sign On when viewing.

☒ Show this dialog after choosing a data source.

- In the *Runtime Prompt Mode* list, select *Never prompt*.

Note

When you store logon credentials for the Data Connection object, always select *Never prompt* as your Runtime Prompt Mode.

Tip

Select the "Use Single Sign On when viewing" check box to enable Single Sign On for the Data Connection. For information about the Single Sign On feature, see [Single Sign On \[page 81\]](#).

- Click *OK*.

3.3.3 Saving and naming a Data Connection

You must save a Data Connection object before you can use it to create other objects, such as Dynamic Data Connections and Data Foundations. It is important to choose a descriptive name for your Data Connection because this name is visible to users who design and view reports based on any Business View that uses the Data Connection.

In this tutorial, you will create a new folder called *Tutorial*, into which you will save the Data Connection object as *Xtreme Development*.

3.3.3.1 To save and name a Data Connection

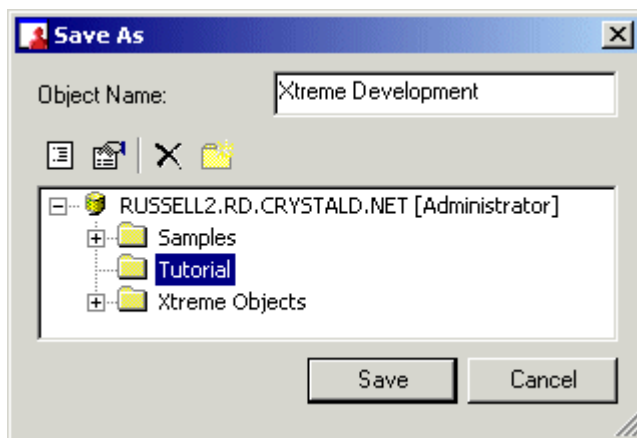
1. On the [File](#) menu, click [Save](#).


→ Tip



You can also click the Save button on the toolbar or press CTRL+S.

The "Save As" dialog box appears.



2. In the [Object Name](#) field, type Xtreme Development .
3.  Click the [Insert a new folder](#) button, and then name the new folder Tutorial .
4. Click the [Tutorial](#) folder to select it, and then click [Save](#).

3.3.4 Modifying the properties of a Data Connection

The Property Browser in the Business View Manager can be used to modify several properties of a Data Connection:

- Name
- Description
- Author
- User Name
- Password
- Use Single Sign On when viewing
- Connection
- Runtime Prompt Mode
- Use Owner
- Use Catalog

- Rights

Note

For complete descriptions of each of these properties, see [Modifying a Data Connection \[page 82\]](#).

In this tutorial, you will use the Property Browser to provide a description for the Data Connection.

3.3.4.1 To add a description using the Property Browser

1. If the Property Browser is not visible, on the [View](#) menu, click [Property Browser](#).

Property Browser	
Name	Xtreme Development
Description	
Author	Administrator
User Name	
Password	
Use Single Sign On ...	True
Connection	Xtreme Sample Data...
Runtime Prompt Mode	Never prompt
Use Owner	True
Use Catalog	True
Rights	...

2. In the Property Browser, click the cell next to the Description field and type `Development Environment`.
The description of the Data Connection is now Development Environment.
3. On the [File](#) menu, click [Save](#).

3.3.5 Exercise: Creating two more Data Connections

You now have one Data Connection object that has been named and saved. Using the same procedures (as described in [Creating and configuring a Data Connection \[page 27\]](#)), create and save two additional Data Connections in the Tutorial folder with these configurations:

Data Connection2

Name	Xtreme QA
Description	QA Test Environment
User Name	

Data Connection2

Password

Runtime Prompt Mode

Never prompt

Data Connection3

Name

Xtreme Production

Description

Production Environment

User Name

Password

Runtime Prompt Mode

Never prompt

3.4 Creating and configuring Dynamic Data Connections

After creating multiple Data Connections, you can create a Dynamic Data Connection. A Dynamic Data Connection is a collection of pointers to different Data Connections.

When users refresh a report that is based on a Dynamic Data Connection, they are prompted to specify which Data Connection to use, whether it be the connection information for the development data, the test data, or the production data. As long as the database schema is the same for all three databases, administrators can easily ensure that reports point to and use the proper data source as required.

In the tutorials and exercise in this section, you will complete these tasks:

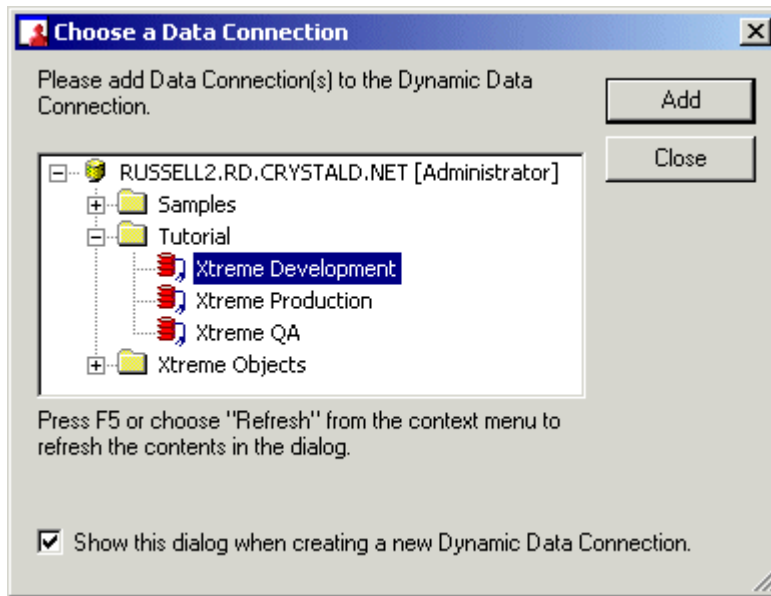
- Create a Dynamic Data Connection.
- Add a Data Connection to an existing Dynamic Data Connection.
- Sort Data Connections in the Dynamic Data Connections window.

3.4.1 To create a Dynamic Data Connection

In this tutorial, you will create one Dynamic Data Connection based on two of the Data Connections that you created in [Creating and configuring a Data Connection \[page 27\]](#).

1. On the *File* menu, select *New*, and then select *Dynamic Data Connection*.

The "Choose a Data Connection" dialog box appears.



2. Expand the *Tutorial* folder to show the available Data Connections.
3. Select the *Xtreme Development* Data Connection.
4. Click *Add*.

Note

Only Data Connections that do not always prompt for a user to log on to a data source can be used for a Dynamic Data Connection. You will receive an error message if you attempt to add a Data Connection that always prompts a user to log on.

5. Select and add the *Xtreme Production* Data Connection.
6. Click *Close*.

3.4.2 To add a Data Connection

In the main window of a Dynamic Data Connection, you can add new Data Connections or delete existing ones. In this tutorial you will add the Xtreme QA Data Connection to the Dynamic Data Connection that you created in [To create a Dynamic Data Connection \[page 35\]](#).

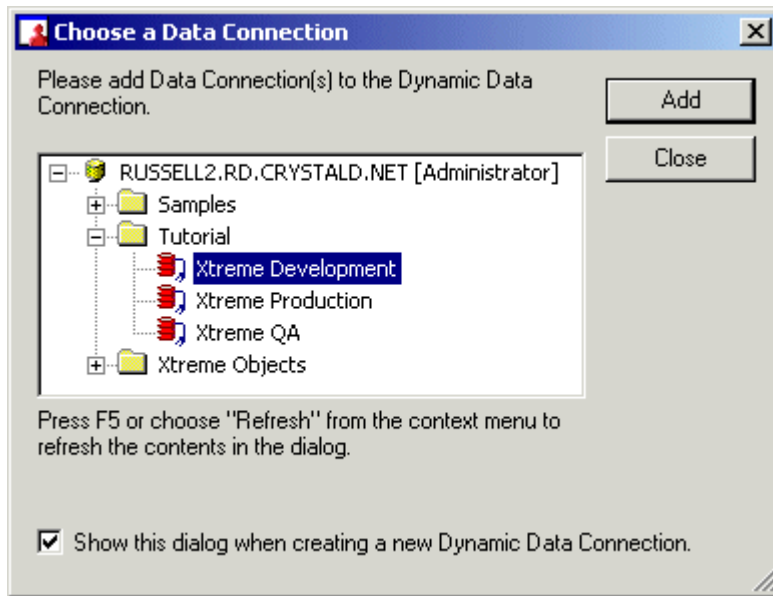
1. In the main window of the Dynamic Data Connection, click the *Add* button located at the bottom-left corner of the window.

→ Tip



You can also click the Add Data Connection button on the toolbar; alternatively, you can click Add Data Connection on the Edit menu.

The "Choose a Data Connection" dialog box appears.



2. Expand the *Tutorial* folder, and select *Xtreme QA*.
3. Click *Add*, and then click *Close*.

3.4.3 Sorting Data Connections

If you have a Dynamic Data Connection with numerous Data Connections, you can sort the listing of your Data Connections. To sort your Data Connections, you can choose from three options in the sort list located in the top right-hand corner of the Dynamic Data Connection window:

- Alphabetical ascending
- Alphabetical descending
- No Sort

You can also move a Data Connection object up or down the list by first selecting the object, and then clicking the up or down arrows next to the sorting list.

Use the arrows to sort the Data Connections so that they appear in this order:

- Xtreme Development
- Xtreme QA
- Xtreme Production

3.4.4 Saving and naming a Dynamic Data Connection

1. On the *File* menu, click *Save*.

→ Tip



You can also click the Save button on the toolbar or press CTRL+S.

The "Save As" dialog box appears.

2. In the *Object Name* field, type `Xtreme Dynamic`.
3. Click the *Tutorial* folder to select it, and then click *Save*.

3.5 Creating and configuring a Data Foundation

After you have created the Data Connection layer (using Data Connections and a Dynamic Data Connection), the next step is to create a Data Foundation. A Data Foundation is a component used for data access management, where you collect a list of data fields that are available for use as Business Fields in Business Elements. A Data Foundation is an abstraction layer into which you can insert and join several types of objects (from different Data Connections):

- Table objects
 - Data tables
 - Views
 - Stored procedures
 - SQL command objects
- Formulas
- SQL expressions
- Parameters
- Filters
- Custom Functions

Note

Views, stored procedures, and SQL command objects, all appear as tables once you add them to the Data Foundation.

The items you choose to include in the Data Foundation determine what fields are later available for report designers to use when creating reports.

In the tutorials in this section, you will complete these tasks:

- Create a Data Foundation.
- Link tables.
- Insert a formula.
- Insert an SQL expression.
- Insert two parameters.
- Insert a filter.

3.5.1 To create a Data Foundation

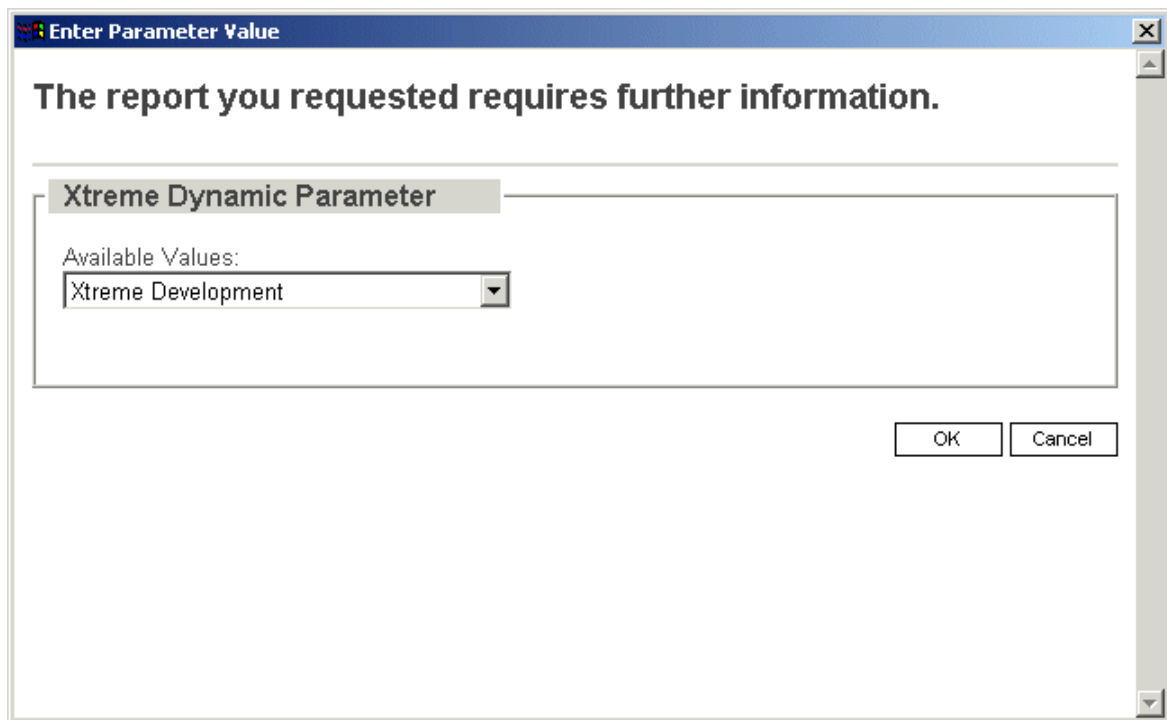
In this tutorial, you will create a Data Foundation called Xtreme Data Foundation, and you will add several tables to the Data Foundation.

1. On the *File* menu, select *New*, and then click *Data Foundation*.

The "Choose a Data Connection" dialog box appears.

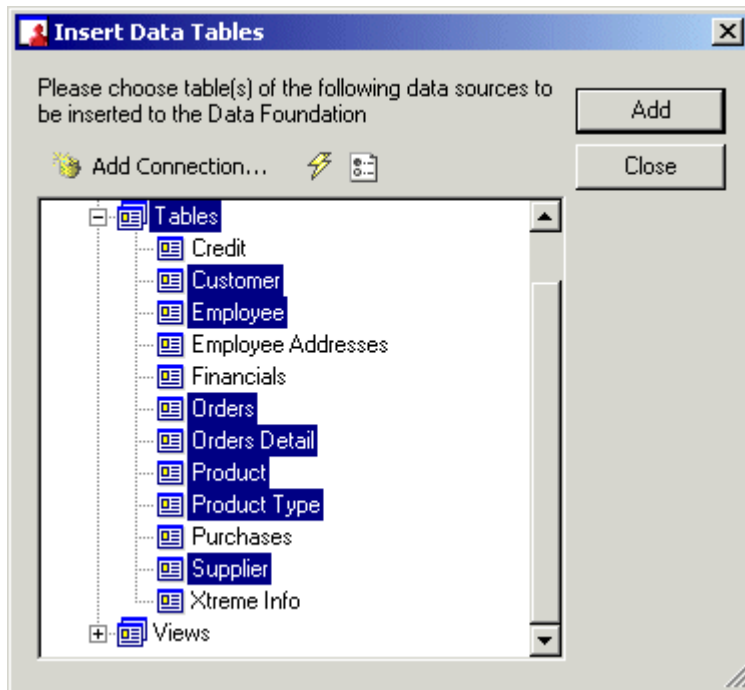
2. Expand the *Tutorial* folder, and select *Xtreme Dynamic*.
3. Click *OK*.

The "Enter Parameter Value" window appears.



4. In the Available Values list, select *Xtreme Development*.
5. Click *OK*.

The "Insert Data Tables" dialog box appears.



6. Expand the *Tables* node.

7. Select these tables:

- Customer
- Employee
- Orders
- Orders Detail
- Product
- Product Type
- Supplier

→ Tip

You can select multiple tables concurrently by holding down the CTRL key and clicking the tables that you want to select.

8. Click *Add*.

9. Click *Close*.

10.  Save the Data Foundation in the *Tutorial* folder as *Xtreme Foundation*.

3.5.2 Linking tables

In your Data Foundation, you link tables so that records from one table will match related records from another. For example, if you add an Orders table and a Customers table, you link the two tables, so that each order (from the Orders table) can be matched up with the customer (from the Customer table) who made the order.

When you link tables, you use a field that is common to both tables. Business Views uses the link to match up records from one table with those from the other. Note that in your Data Foundation, you can use different tables from different data sources.

In this tutorial, you will use the Smart Linking feature of the Business View Manager to link the tables in the Data Foundation that you created in [To create a Data Foundation \[page 38\]](#). You will also manually delete one link and add another.

3.5.2.1 To link tables in the Data Foundation

1. Right-click in the main Data Foundation window.

The shortcut menu appears.

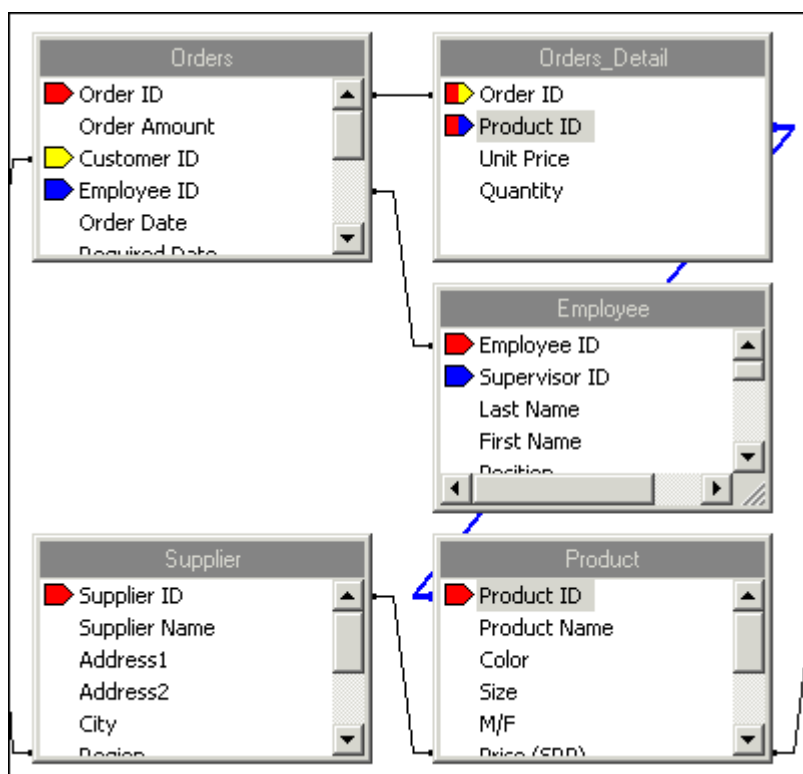
2. Select *Smart Linking By Name*.

The tables are automatically linked by field name.

3. Right-click the link between the Customer table and the Supplier table, and on the shortcut menu, select *Delete Link*.

4. From the Orders_Detail table, click and drag the *Product ID* field over to the Product ID field in the Product table.

A link appears.



For more information on linking tables, see [Linking tables \[page 100\]](#).

3.5.3 Inserting a formula

In many cases, the data needed for a Data Foundation object already exists in database fields. Sometimes, however, you need to put data in a Business Element (a Business Element is a collection of fields that are based on a Data Foundation) that does not exist in any of the data fields. In this situation, you need to create a formula. You must create and define your formulas at the Data Foundation level before you can add these formulas to your Business Elements.

To create a formula, use the Formula Editor. The Formula Editor contains four windows—three of which are situated under the Formula Editor toolbar—and the text window directly below.

Window	Description of contents
Report Fields	Report fields contains all the database fields that are accessible for your Business Element. They also contain any formulas or groups already created for the Business Element.
Functions	Functions are prebuilt procedures that return values. They perform calculations such as average, sum, count, sin, trim, and uppercase. Custom functions are also listed in this window.
Operators	Operators are the "action verbs" you use in formulas. They describe an operation or an action to take place between two or more values. Examples of operators include: add, subtract, less than, and greater than.
Formula text window	This is the area where you create a formula. You can also view your formula in the Property Browser window.

In this tutorial, you will use the Formula Editor to create a formula that calculates an employee's number of years of service.

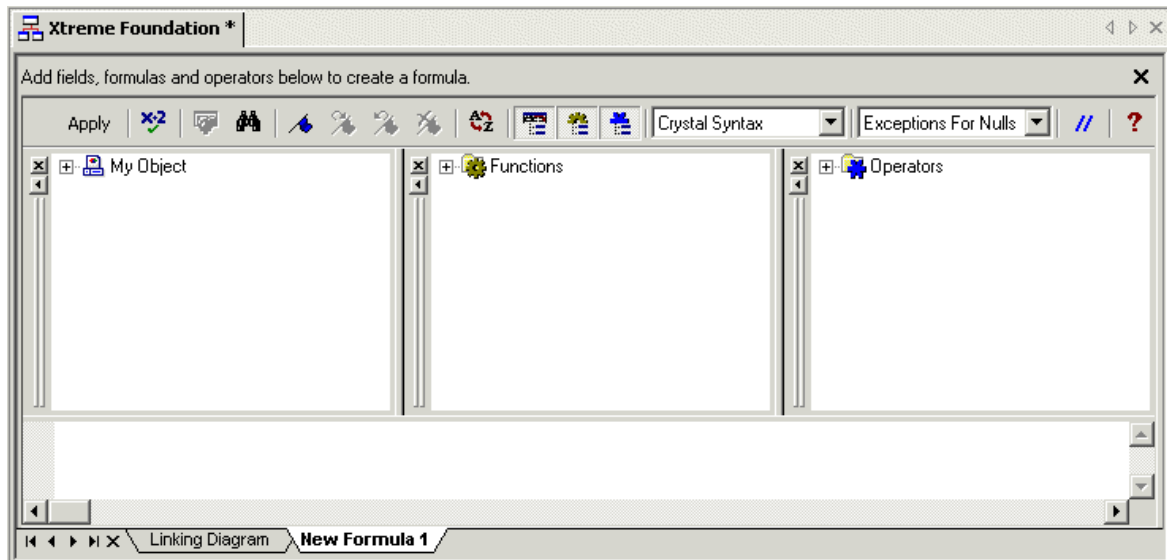
3.5.3.1 To create a formula

1. On the *Insert* menu, click *Insert Formula*.

→ Tip

In the Object Explorer, you can right-click Formulas, and click Insert Formula.


The Formula Editor appears.



2. From the first drop-down list on the Formula Editor toolbar, choose *Crystal Syntax*.
For more information on which syntax to use, see [Choosing the syntax \[page 120\]](#).
3. In the second drop-down list, specify one of the following values:
 - Choose *Exception for Nulls* to configure the formula to ignore null values.
 - Choose *Zeros for Nulls* to configure the formula to treat null values as zero values.
4. In the Functions window, expand *Functions > Date and Time > DateDiff*, and double-click *DateDiff (intervalType, startDateTime, endDateTime)*.
DateDiff (,) appears in the Formula text window.
5. In the Formula text window, type the necessary strings to fill in the formula so that it appears as `DateDiff ("YYYY", {Employee.Hire Date}, CurrentDate)`

Note

You can also select *Employee.Hire Date* and *CurrentDate* by expanding the Report Fields and Functions trees respectively and double-clicking the desired fields.

6.  On the Formula Editor toolbar, click *Check* to test if there are any errors in the formula.
7. Fix any syntax errors the Formula Checker identifies.

Note

For more information on formulas, see the SAP Crystal Reports Online Help.

8. Click *Apply* to save the formula.
9. In the Object Explorer, expand the *Formulas* node, and click *New Formula 1*. This is the new formula that you created.
10. In the Property Browser, rename the formula to `Years of Service`.

→ Tip

You change the name of the formula by clicking the cell next to the Name field, and then typing the desired name.

3.5.4 Inserting an SQL expression

SQL expressions are similar to formulas, but they are written in Structured Query Language (SQL). They are useful in optimizing report performance because the tasks they execute are always performed on the database server (as opposed to a regular formula, which is typically executed on the local machine). Like formulas, you must create and define your SQL expressions at the Data Foundation level before you can add these expressions to your Business Elements.

Much like the Formula Editor, the SQL expression Editor contains four windows.

Window	Description of contents
Report Fields	Report fields contains all the database fields that are accessible to your Business Element.
Functions	Functions are prebuilt procedures that return values. They perform calculations, including conversion calculations, numeric calculations, and so on.
Operators	<p>Operators are the "action verbs" you use in SQL expressions. They describe an operation or an action to take place between two or more values.</p> <p>Examples of operators include: add, subtract, less than, and greater than.</p>
SQL expression text window	This is the area where you create an SQL expression. You can also view your SQL expression in the Property Browser window.

In this tutorial, you will use the SQL expression Editor to create an SQL expression that returns an employee's full name.

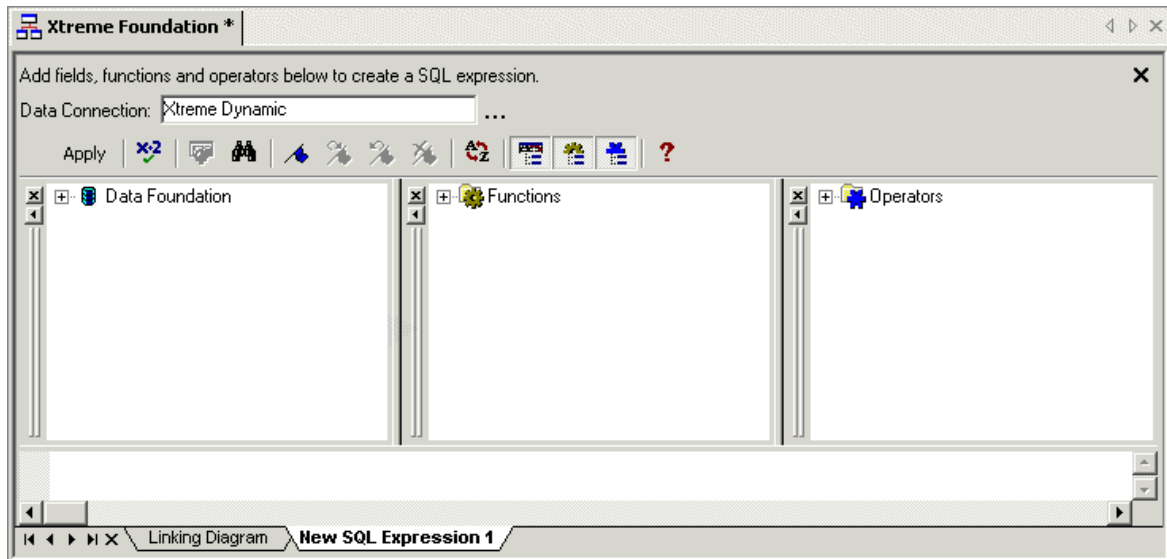
3.5.4.1 To create an SQL expression

1. On the *Insert* menu, click *Insert SQL Expression*.

→ Tip

In the Object Explorer, you can right-click SQL Expressions, and click Insert SQL Expression.

The SQL expression Editor appears.



2. In the Report Fields window, expand *Data Foundation*, and then *Employee*.
3. Double-click the *Last Name* field.
4. In the SQL expression text window, type + ' ', ' + next to 'Employee'. 'Last Name',


Note

That is, [space]+[space]','[space]'+

5. In the Report Fields window, double-click the *First Name* field.

The SQL expression should now read:

'Employee'. 'Last Name' + ' ', ' + 'Employee'. 'First Name'

6.  On the SQL expression Editor toolbar, click *Check* to test if there are any errors in the SQL expression.
7. Fix any syntax errors the SQL expression Checker identifies.

Note

For more information on SQL expressions, see the SAP Crystal Reports Online Help.

8. Click *Apply* to save the SQL expression.
9. In the Object Explorer, expand the *SQL Expressions* node, and click *New SQL Expression 1*. This is the new SQL expression that you created.
10. In the Property Browser, rename the SQL expression to *Employee Full Name*.

→ Tip

You can change the name of the SQL expression by clicking the cell next to the Name field, and then typing the desired name.

3.5.5 Inserting a parameter

Parameters prompt the user of a report to enter information. Think of a parameter as a question that the user needs to answer before the report is generated from the Business View information. The information that users enter—or the way they respond—determines what appears in the report. For example, in a report used by salespeople, there might be a parameter that asks the user to choose a region. The report would return the results for the specific region chosen by the user, instead of returning the results for all of the regions.

For detailed information on parameter fields and on advanced parameter features, see the "Parameter Fields" section of the SAP Crystal Reports Online Help.

3.5.5.1 Creating a new parameter field

You must create and define your parameter field in your Data Foundation before you can use it in formulas of fields. In this tutorial, you will create two new parameters—Order Start Date and Order End Date—that will prompt users to enter a date range for the order(s) that they want to view in a report. You will use these parameters to create a filter in the next exercise, [Inserting a Business Filter \[page 48\]](#).

3.5.5.1.1 To create a parameter field

1. On the [Insert](#) menu, click [Insert Parameter](#).

Note

In the Object Explorer, you can right-click Parameters, and click Insert Parameter.

The "Create Parameter" dialog box appears.

Create Parameter

Name: Type: List of Values:

Value Field: Description Field:

Value Options:

Option	Setting
Prompt Text	Enter My Parameter:
Prompt With Description Only	False
Allow custom values	True
Allow multiple values	False
Allow discrete values	True
Allow range values	False

OK Cancel Help

- In the *Name* field, type `Order Start Date`.

Note

By default, the name that you provide is automatically used in the parameter's Text field. (The text in this field appears when users are prompted to provide a value for the parameter.) You can use the given text or provide your own text for the prompt.

- From the *Type* list, select *Date*.
- From the *List of Values*, select *Static*. For more information on dynamic prompts and cascading lists, see [Dynamic Prompts and Cascading Lists of Values \[page 160\]](#)
- From the *Value Field*, select *Order Date*.
- Click the down arrow under the *Value* list. Navigate through the list to 01/01/2002 and click on it.
- From the *Value Options*, select *True* to *Allow discrete values*.
- Click *OK*.

Now that you have created one parameter, use the same steps to create another parameter with these properties:

New Parameter 2

Name	Order End Date
Prompt text	Enter end date of order history.
Parameter type	Date
Options	Discrete values
Default End Date	12/31/2002

3.5.6 Inserting a Business Filter

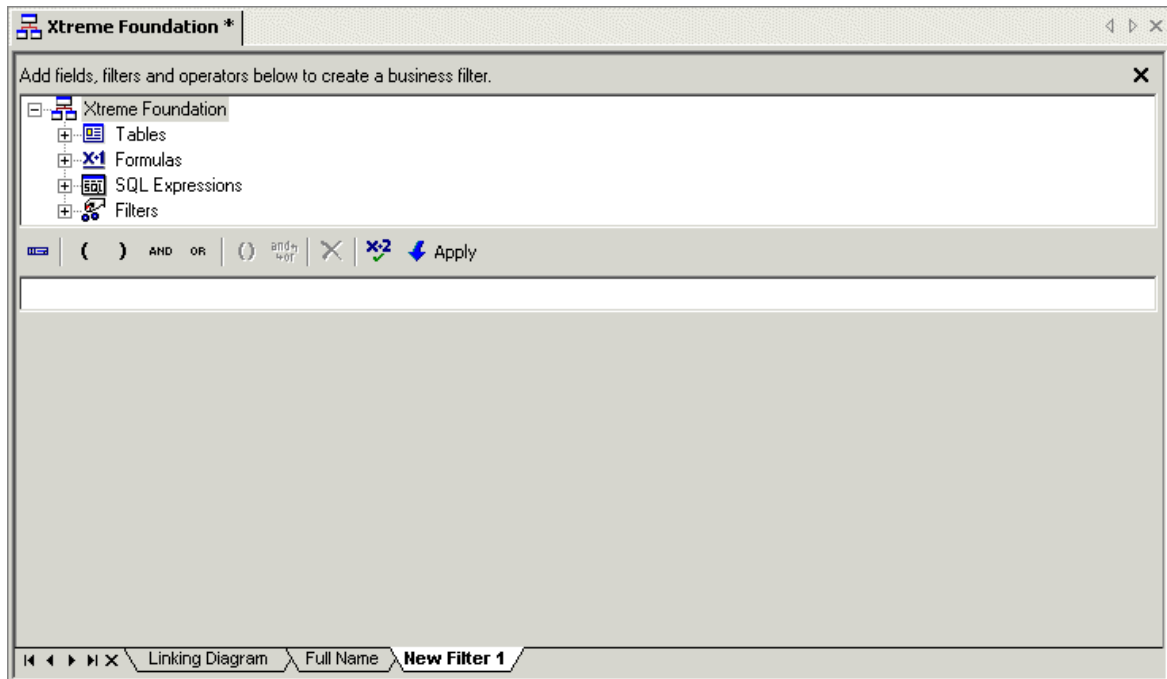
By default, a Full Data Access filter and a No Data Access filter are available for your Data Foundation. You can also create your own filters to apply to your Data Foundation. These filters allow you to reference fields, formulas, SQL expressions, parameters, and other filters. You can use Boolean operators to create Business Filters that restrict access to certain information for specific users or groups.

In this tutorial, you will create a Business Filter that uses the two order date parameters that you created in [Inserting a parameter \[page 46\]](#). The filter ensures that the records returned are those that fall within specific order start and end dates.

3.5.6.1 To create a Business Filter

1. In the Object Explorer, right-click *Filters*, and click *Insert Filter*.

The Filter Editor appears.



2. In the "Add fields, filters and operators below to create a Business Filter" area, navigate to ► **Tables** ► **Orders** and double-click **Order Date**.

Order Date is any value appears in the filters field.

3. Click **Order Date is any value**.

The Order Date area appears with a drop-down list.


4. From the drop-down list, select **is between**.

Two more drop-down lists appear.

5. From the two lists, select **{?Order Start Date}** AND **{?Order End Date}**.

Note

{?Order Start Date} and {?Order End Date} are the order date parameters you created in [Inserting a parameter \[page 46\]](#).

6.  In the Filter Editor toolbar, click **Check Filter Validity** to verify that there are no errors in the filter.
7. Click **Apply** to save your filter.
8. Use the Property Browser to change the name of the filter to **Order Date Filter**.
9. On the **File** menu, click **Save**.

Tip



You can also click the Save button on the toolbar or press CTRL+S.

3.6 Creating and configuring Business Elements

A Business Element is an abstraction object that allows you to shape your data fields from a Data Foundation into components that make sense from a business perspective. It is important to understand that the organization of Business Fields in Elements does not need to reflect the shape of the underlying tables in the Data Foundation.

In the tutorial and exercise in this section, you will complete these tasks:

- Create and configure six different Business Elements.
- Create an alias for a Business Field within a Business Element.

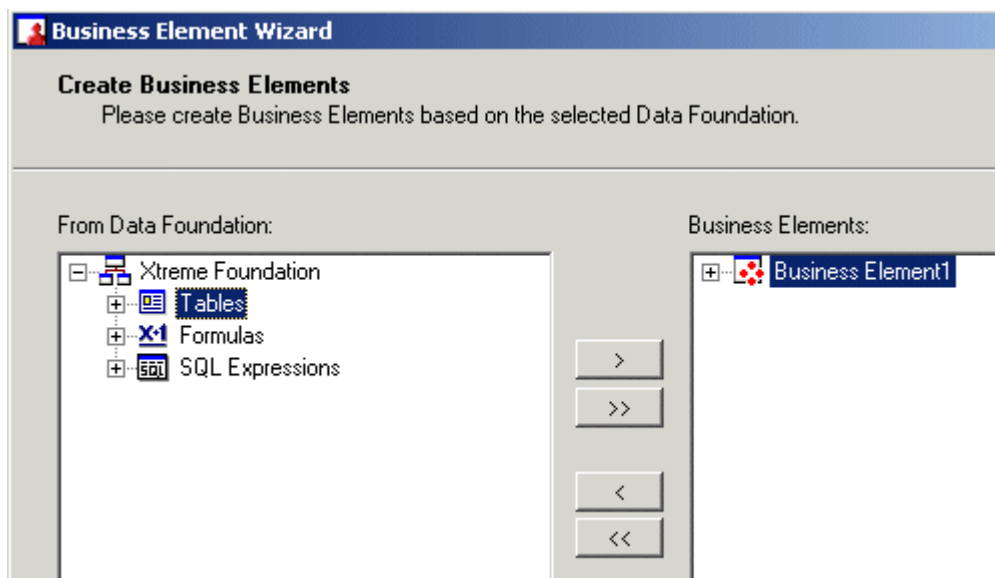
3.6.1 Creating a Business Element

You can create Business Elements in two ways. You can create each Business Element individually by choosing **File > New > Business Element**. However, using the Business Element Wizard is more efficient because it allows you to create several Business Elements at once.

In this tutorial, you will use the Business Element Wizard to create a Business Element called Customer.

3.6.1.1 To create a Business Element using the Business Element Wizard

1. On the **File** menu, click **New**, and then click **Business Element Wizard**.
2. Select **Xtreme Foundation** (located in the Tutorial folder) as your Data Foundation, and then click **Next**.
3. In the "Create Business Elements" dialog box, expand **Tables**, and select **Customer**.



- 4. Click the > arrow to move the entire Customer table to the Business Elements area.
- 5. In the Business Elements area, expand the *Customer* table, and select *Customer Credit ID*.
- 6. Click the < arrow to remove Customer Credit ID from the Business Elements area, and then click *Next*.

ⓘ Note

This field will not be visible to report designers who create reports based on Business Views that use this Business Element.

- The "Save to Repository" dialog box appears.
- 7. Select the *Tutorial* folder, and click *Next*.
- The "What to do next" dialog box appears.
- 8. Select *Create more Business Elements*, and then click *Finish*.
- The Business Element Wizard returns you to the first dialog box ("Choose a Data Foundation"), so that you can create another Business Element.

3.6.2 Exercise: Creating additional Business Elements

Now that you have created one Business Element, called Customer, use the Business Element Wizard to create five additional Business Elements as outlined in the tables below:

Business Element 2	
Table to add	Employee
Fields to remove	<ul style="list-style-type: none">• Home Phone• Extension• Photo• Notes• Emergency Contact First Name• Emergency Contact Last Name• Emergency Contact Relationship• Emergency Contact Phone
Fields to add	<ul style="list-style-type: none">• Years of Service• Employee Full Name
<div><div>ⓘ Note</div><div>These two fields are located under Formulas and SQL Expressions respectively.</div></div>	

Business Element 3

Table to add	Orders
Fields to remove	<ul style="list-style-type: none">Courier Website

Business Element 4

Table to add	Orders Detail
--------------	---------------

Business Element 5

Table to add	Product
Fields to add	<ul style="list-style-type: none">Product Type Name

Note

This field is located in the Product Type table.

Business Element 6


Table to add	Supplier
--------------	----------

Exit the Wizard once you have finished.

3.6.2.1 To create an alias for a Business Field

As a part of the abstraction that Business Elements facilitate, you can create an alias for a Business Field. This can make report design much easier because it allows you to make field names more accurate and descriptive.

In this tutorial, you will create an alias for a Business Field that is part of one of the Business Elements that you created in [Creating a Business Element \[page 50\]](#).

1. On the **File** menu, click **Open**, and then navigate to the Employee Business Element. Select it and click **Open**.
2. In the Object Explorer, expand the **Fields** node, and select **Salary**.
3. In the Property Browser, change the name to `Annual Salary`.
4.  Click **Save**.

3.7 Creating and configuring a Business View


Once you have created one or more Business Elements, you can create a Business View. A Business View is a logical collection of Business Elements. Users see Business Views as an abstract database connection, and the contained Business Elements as virtual tables that, in turn, contain Business Fields.

In this tutorial, you will create a Business View based on the Business Elements that you created in [Creating and configuring Business Elements \[page 50\]](#).

3.7.1 To create a Business View

1. On the *File* menu, click *New*, and then click *Business View*.
2. In the Object Explorer, right-click the *Business Elements* node, and click *Insert Business Elements*.

The "Insert Business Elements" dialog box appears.

3. Expand the *Tutorial* folder and select the *Customer* Business Element.
4. Click *Add*.
5. Individually select and add each of the remaining Business Elements in the Tutorial folder: Employee, Orders, Orders Detail, Product, and Supplier.
6. Click *Close*.
7.  Click *Save*.
8. In the *Object Name* field, type Xtreme Business View.
9. Select the *Tutorial* folder, and then click *Save*.

You have now created a Business View from which users can create reports. For information on setting security for each of the objects, see [Business Views Security Concepts \[page 187\]](#).

4 Using the Business View Manager

This section serves as an introduction to using the Business View Manager and details how to create and manage different Business Views objects. It also provides information on using the Repository Explorer through the Business View Manager.

4.1 Business View Manager overview

The Business View Manager is a designer for Business Views administrators that provides a wide range of capabilities for creating and modifying Data Connections, Dynamic Data Connections, Data Foundations, Business Elements, and Business Views. The Business View Manager enables you to design relational views of information. This designer also enables you to set detailed column and row-level security for various objects in your report. You can also use this designer to specify different Data Connections, to set security, and to control access to the data found within various data sources.

In addition, you can use the Business View Manager to apply detailed security settings to the SAP BusinessObjects repository. You can set security rights for users and groups for folders and objects found within the repository. As long as a user has the Business View Manager installed on his or her machine, he or she can use this designer; however, the user must have access rights to folders in the repository before he or she can save objects to these specific folders. All Business Views objects are saved to the repository. For more information on the security settings for the Repository, see the related link. .

This section briefly introduces to new Business Views administrators some of the features found within the Business View Manager. It also provides information on using the Repository Explorer, which allows you to navigate the repository.

Related Information

[SAP BusinessObjects repository security model \[page 67\]](#)

4.2 Working with the Business View Manager

You will use the Business View Manager (BVM) extensively to manage your Business Views system. This designer enables you to create and modify Data Connections, Dynamic Data Connections, Data Foundations, Business Elements, and Business Views. The BVM allows you to set in detail all of the properties and settings for these various objects, including the necessary security settings for each object.

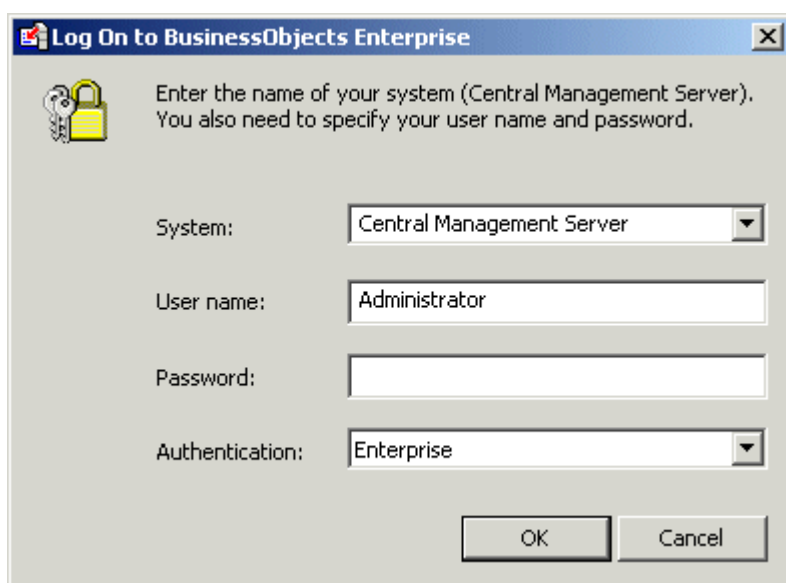
Any user with valid credentials to SAP BusinessObjects Business Intelligence platform can log on to the BVM and create and modify objects to which he or she has access. Because all of the objects found in Business

Views are saved to the SAP BusinessObjects repository, the Business Views user must have access to the repository before he or she can open, modify, and save different objects.

4.2.1 To log on to the Business View Manager

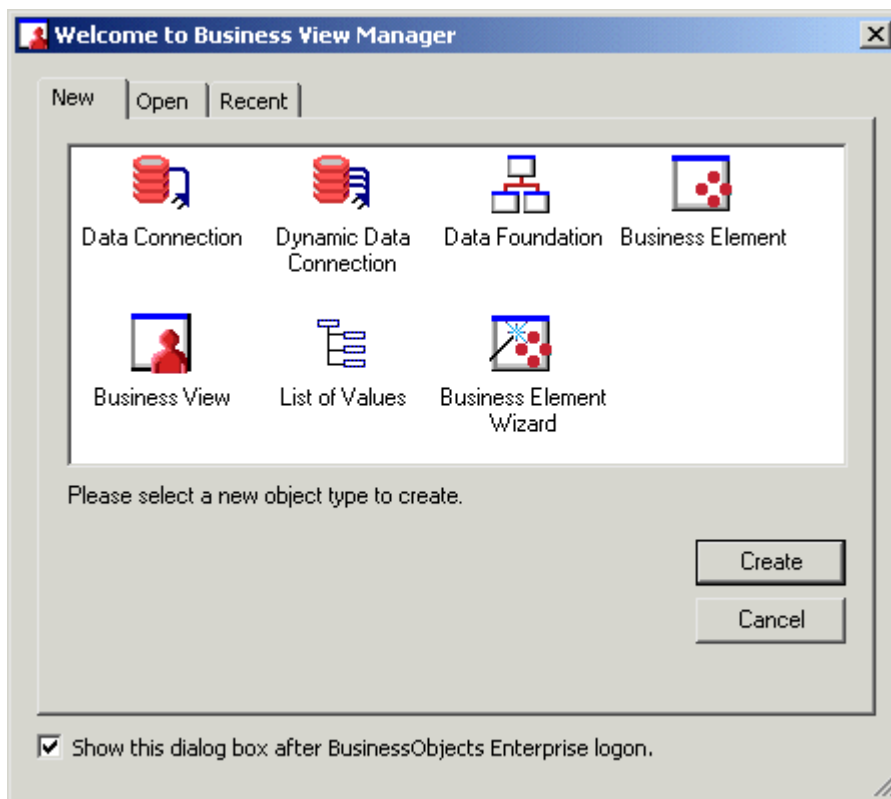
Every time you use the Business View Manager, you need to log on with the proper credentials. When you log on, you are logging on to the Central Management Server (CMS), which is a part of SAP BusinessObjects Business Intelligence platform.

1. In Windows, click *Start > Programs > BusinessObjects BI platform > BusinessObjects BI platform Client Tools > Business View Manager*.



2. When the Log On dialog box appears, select the type of authentication in the *Authentication* list.
3. In the *System* list, select the appropriate CMS.
4. Type your *User name* and *Password*.
5. Click *OK*.

The Welcome to Business View Manager dialog box appears.



4.2.2 Navigating the Welcome to Business View Manager dialog box

The Welcome to Business View Manager dialog box is your starting point for creating a new Business Views object or opening a saved or recently saved object. This dialog box allows you to navigate to any object and also to create and delete new folders and objects within the SAP BusinessObjects repository.

→ Tip

While in this dialog box, you can clear the "Show this dialog after BusinessObjects Enterprise logon" check box if you do not want this dialog box to appear after you log on to the Business View Manager.

→ Tip

This Welcome to Business View Manager dialog box also appears when you select the Open command on the File menu—you can select the "Show this dialog after BusinessObjects Enterprise logon" check box if you want to have the dialog box appear again every time you start the BVM.

This dialog box consists of three tabs:

- [New](#)
On the New tab, you can select any new object you want you create, including the following:
 - [Data Connection](#)

A Data Connection enables you to specify and define a data source. For more information on Data Connections, see [Data Connections overview \[page 77\]](#).

- **Dynamic Data Connection**
A Dynamic Data Connection is a collection of pointers to different Data Connections. For more information on this object, see [Dynamic Data Connections overview \[page 87\]](#).
- **Data Foundation**
This object is a collection of tables and fields—a Data Foundation enables tables from different data sources to be used together. For more information on Data Foundations, see [Data Foundations overview \[page 97\]](#).
- **Business Element**
A Business Element consists of a logically related collection of data fields that are based on a Data Foundation. For more information on Business Elements, see [Business Elements overview \[page 144\]](#).
- **Business View**
A Business View is a logical collection of Business Elements. Users see Business Views as abstract database connections, and the contained Business Elements as virtual tables that, in turn, contain Business Fields. Thus, end users can access Business Views through a client application such as SAP Crystal Reports. For more information on this object, see [Business Views overview \[page 174\]](#).
- **Business Element Wizard**
The Business Element Wizard guides you step-by-step through the process of creating multiple Business Elements directly from a Data Foundation. For more information on using this wizard, see [Using the Business Element Wizard \[page 157\]](#).
- **Open**
The Open tab displays the Repository Explorer. Use the Repository Explorer to find and open a previously saved object, to filter or change repository view settings, to delete an object or folder, and to create a new folder. For more information on using the Repository Explorer, see [Using the Repository Explorer \[page 62\]](#).
- **Recent**
In the Object Name column, select from one of the recently modified objects, and click Open to view and update this object.

4.2.3 Navigating within the Business View Manager

Each window within the Business View Manager can be set to be docked or to be in floating mode. You can manually dock each window anywhere in the designer. Logically related windows are typically grouped together: for example, the Property Browser is often grouped with the Object Explorer. You can switch between the different grouped windows by clicking the windows' appropriate tabs. Double-click the tab to remove the window from its associated group. You can also resize each window by dragging any of its edges with the Resizing cursor.

Within certain windows, you can click Auto Hide (the pin-shaped button located on the top-right corner of a window) to hide the window. The Business View Manager displays hidden windows as tabs. When you click a tab, the window reappears. You can dock windows on top of each other, which results in grouped windows that can be selected by their exposed tabs. You can also close a window by clicking the small "X" button, typically located in the top-right corner of the window. You can select which windows to open through the View menu.

The Business View Manager maintains all of your settings for each dockable window; thus, depending on how these windows appeared when you last used the Business View Manager, each window may be in docked or

in floating mode. In the screen shots that follow in this administrator's guide, some of your windows may be placed in different locations.

When you have multiple Business Views objects open, you can navigate between the different objects by clicking their respective tabs. These tabs are located on the top portion of the main window. You can also click the "Scroll Left" and "Scroll Right" buttons, located at the top-right corner of the main window to scroll between the various tabs.

4.2.4 To save an object within the Business View Manager

All of the objects found within Business Views are saved to the SAP BusinessObjects repository, which is contained in the Central Management Server (CMS) of SAP BusinessObjects Business Intelligence platform. For more information on the repository, see [Using the Repository Explorer \[page 62\]](#).

When you first save a Business Views object, you need to specify a name for the object and the location you want to save the object to.

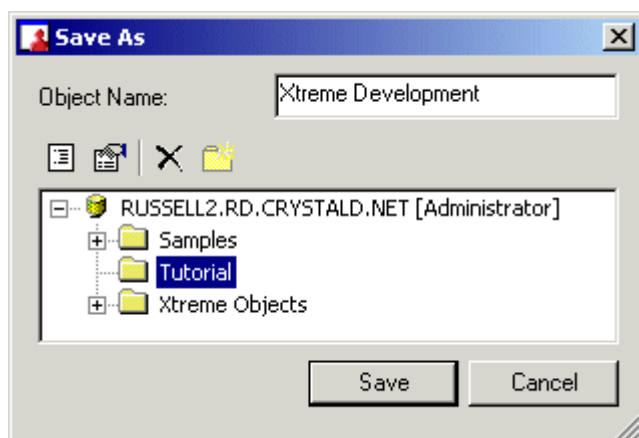
1. On the [File](#) menu, select [Save](#).

→ Tip



You can also click the Save (Ctrl+S) button or press CTRL+S.

The Save As dialog box appears.



2. In the [Object Name](#) field, specify the name of your object.
3. Select the folder you want to save your object to.

→ Tip



You can create new folders in the repository by clicking the "Insert a new folder" button.

4. Click [Save](#).

4.3 Exporting and importing Business Views

The import and export tools enable you to export and import Business Views and their associated objects (that is, the Data Connections, Dynamic Data Connections, Data Foundation, Business Elements, and any other objects that the Business Views reference).

When you export Business Views and their associated objects, all of this information is exported as an XML file. Depending on whether you choose to include security information or not, user and group rights are or aren't included in the XML file. Regardless, for security reasons, passwords for Data Connections are not exported.

The ability to export a complete set of Business Views objects allows you to develop different migration strategies. For example, you can migrate Business Views between CMS cluster installations. That is, you might have reporting systems with separate clusters for development, test, and production systems.

In addition, you are able to extract your Business Views data from within the SAP BusinessObjects repository. Because the information is saved as an XML file, this format is suitable for storage in a Code Asset Management tool (for example, Microsoft SourceSafe).

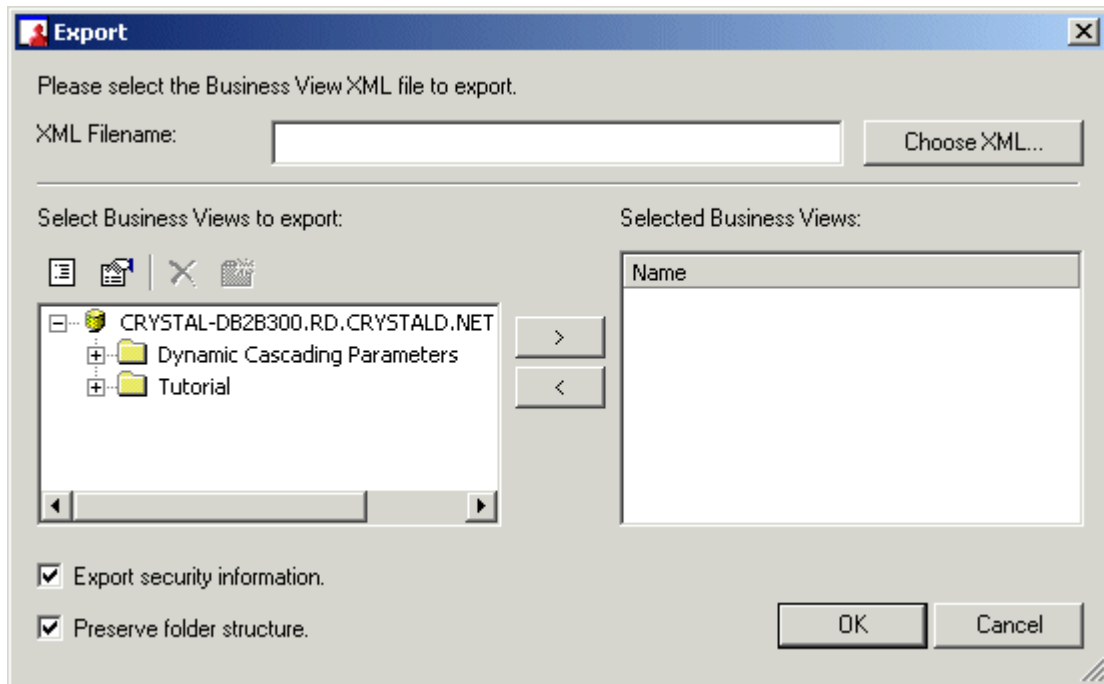
ⓘ Note

When you import an XML file to a repository, even if the file contains security settings, the security settings for the destination folder may override whatever settings you have in the XML file. Because Business Views uses a denial-based inheritance model, if rights are denied at the folder level, these rights will also be denied at the imported object level.

4.3.1 To export Business Views

1. On the *Tools* menu, select *Export*.

The Export dialog box appears.



2. In the *XML Filename* field, type the location and the name of the XML file you are exporting to, or click *Choose XML* to specify a location.
3. In the "Select Business Views to export" area, choose a Business View to export.

Note



Click the Change view settings button to bring up the View Settings dialog box. In this dialog box, you can select the item types you'd like to see, and also sort items by Name or by Type.

Note



Click the Advanced filtering button to filter items by text or by author. Click the button again to disable advanced filtering.

4. Click the > button to move the Business View to the "Selected Business Views" area.

Tip

You can export more than one Business View by selecting different Business Views using the CTRL key and clicking the > button to create a list of Business Views to export.

Tip

Click to < button to remove a Business View from the list of Business Views you want to export.

5. Select the *Export security information* check box to export the security rights that are associated with the Business View.

Note

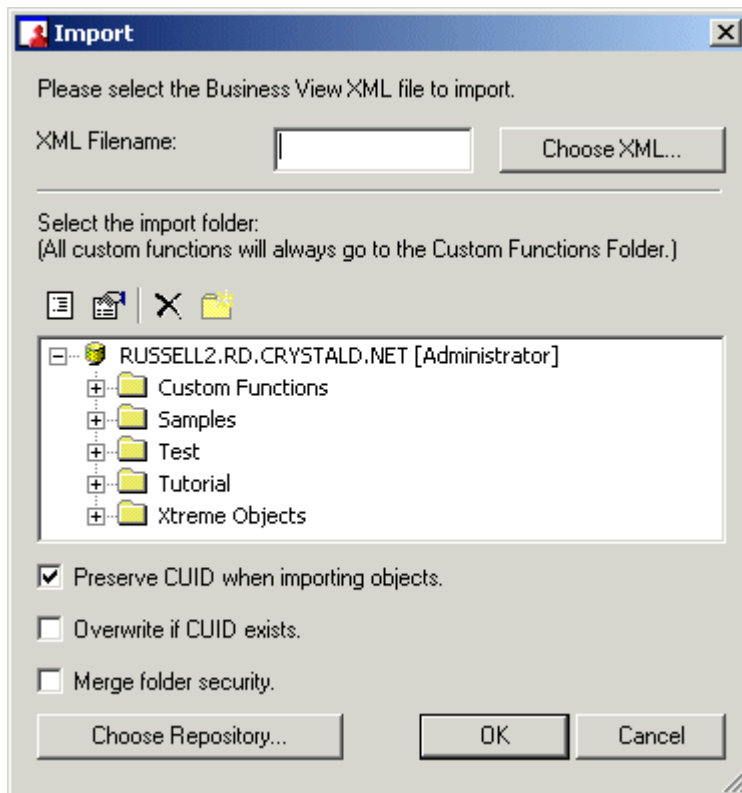
Passwords for Data Connections are not exported to the XML file.

6. Select the *Preserve folder structure* check box to include the folder hierarchy information of the Business View when it is exported.
7. Click *OK* to export the selected Business Views.



4.3.2 To import Business Views

1. On the *Tools* menu, select *Import*.

The Import dialog box appears.



2. In the *XML Filename* field, type the name of the XML file you are importing, or click *Choose XML* to find the XML file.
3. In the "Select the import folder" area, choose a folder to import the XML to.
 - Click the Change view settings button to bring up the View Settings dialog box. In this dialog box, you can select the item types you'd like to see, and also sort items by Name or by Type.
 - Click the Advanced filtering button to filter items by text or by author. Click the button again to disable advanced filtering.

-  Click the Delete button to delete items or folders in the list.
 -  Click the Insert a new folder button to insert a new folder.
4. Select the [Preserve CUID when importing objects](#) check box to ensure that each object's unique identifier is preserved during import.

Note

If this option is not selected, new CUIDs (an object's unique identifier) will be assigned to the imported objects.

Note

Reports that are based on Business Views reference the specific Business View's unique identifier; thus, if the unique identifier is not preserved, upon refresh, the report will fail.

Tip

You can select the Overwrite if CUID exists check box to replace any object in the folder that has same CUID as the object being imported.

5. Select the [Merge folder security](#) check box to combine the security settings of object's origin folder with the security settings of its destination folder. If there is a conflict, the security settings of the destination folder is used.
6. Click the [Choose Repository](#) button if you want to import the XML file to a different SAP BusinessObjects repository.

Specify the Central Management System name, the User name, Password, and Authentication type and click OK.
7. Click [OK](#) to import the selected XML file.

4.4 Using the Repository Explorer

The SAP BusinessObjects repository is the central location for you to store and manage your objects. Business Views users store objects such as Data Connections, Dynamic Data Connections, Data Foundations, Business Elements, and Business Views. All of the objects that are found within Business Views are stored in the repository.

The Repository that ships with Business Views is contained in the Central Management Server (CMS) of SAP BusinessObjects Business Intelligence platform. The CMS is installed when you install the platform and Business Views. Before publishing reports that reference repository objects, move your existing repository to the Central Management Server database. For more information on migrating the repository, see the *SAP BusinessObjects Business Intelligence platform Administrator Guide*.

While using the Repository Explorer, you can double-click a Business Views object to open and edit the object. You can also right-click the object to change its settings or view its properties. In addition, the Repository Explorer also displays the name of the current user and which CMS the user is connected to.

Note

The repository is also used by SAP Crystal Reports to store supported Crystal report object types, including text objects, bitmaps, custom functions, and commands (queries). For more information on the repository as it pertains to SAP Crystal Reports, see the SAP Crystal Reports Online Help.

4.4.1 Accessing the SAP BusinessObjects repository

The SAP BusinessObjects repository is set up when you install Business Views. You do not have to do anything additional before you use the repository.

4.4.1.1 To open the SAP BusinessObjects repository

1. In Business Views, under the [View](#) menu, select the [Repository Explorer](#).

The Repository Explorer appears.





2. Expand the top node to access the various folders and objects in the repository.

Note




The Repository Explorer might appear in a docked position in the Business View Manager depending on where it was when you last used Business Views. You can drag it where you like, or dock it elsewhere.

4.4.1.2 Toolbar

The Repository Explorer's toolbar consists of various buttons that allow you to add new folders, search for items, and so on:

-  [Change view settings](#)
Use this option to open the View Setting dialog box. Use this dialog box to limit the type of repository items displayed in the Repository Explorer. You can also select options to sort multiple items by name or by type.
-  [Advanced filtering](#)
Use this option to display filtering options at the bottom of the Repository Explorer. Use these filters to find specific items by words from the "Show items with this text in the name:" or the "Show items by this author:" fields.
-  [Delete the item/folder](#)
Use this option to remove the selected item or folder from the repository. When you delete a folder, you delete all the items it contains. For more information about deleting items from the repository, see [Deleting items from the repository \[page 65\]](#).
-  [Insert a new folder](#)

Use this option to add a new folder to the repository. For information on adding a new folder, see [Adding folders to the repository \[page 64\]](#).

-  **Check Dependent Integrity**
Use this option to test the Business Views objects that are dependent on the object that you have selected in the Repository Explorer.
-  **Show Dependent Objects**
Use this option to view a list of Business Views objects that depend on the object that you have selected in the Repository Explorer.
-  **Show Referenced Objects**
Use this option to view a list of Business Views objects that are referenced by the object that you have selected in the Repository Explorer.

4.4.2 Adding folders to the repository

You can organize the contents of the repository by creating folders and subfolders in the tree view.

Folders provide you with the ability to organize and facilitate content administration. They are useful when there are a number of Business Views objects that a department or area requires frequent access to, because you can set object rights and limits once, at the folder level, rather than setting them for each object within the folder.

By default, new objects that you add to a folder inherit the object rights that are specified for the folder. For more information, see [Applying security settings to folders in the repository \[page 68\]](#).

4.4.2.1 To add folders to the repository

1. In the Repository Explorer, right-click a node and click **New Folder** on the shortcut menu.

→ Tip



Alternatively, click the "Insert a new folder" button on the Repository Explorer's toolbar.

A new folder is added to the bottom of the repository tree. (If your repository items are sorted by type, the new folder is added alphabetically with the default name New Folder.)

2. Name your new folder and press the **Enter** key.
3. To add a subfolder, right-click your new folder and click **New Folder** on the shortcut menu.
4. Name your new subfolder and press the **Enter** key.

4.4.3 To rename a folder

You can rename any folder in the repository, as long as you have View and Edit rights for the folder.

1. In the Repository Explorer, right-click a folder and select [Rename](#).
2. Rename the selected folder and press the [Enter](#) key.

4.4.4 Deleting items from the repository

Any object you store in the repository can be deleted from that repository. Once you remove an object from the repository, it is removed for all users.

Note

The ability to delete repository objects is controlled by the permissions that are set in SAP BusinessObjects Business Intelligence platform. Once you confirm in Business Views that you want to delete the selected object and its dependent objects, you will receive an error message if you do not have adequate permission to delete the object from the repository.

4.4.4.1 To delete an object from the repository

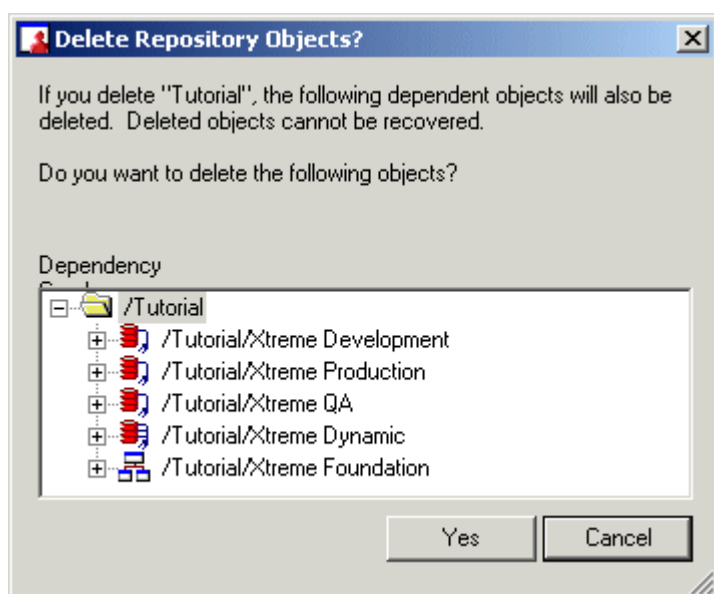
1. In the Repository Explorer, navigate to the appropriate folder, select the object you want to remove, and press the [Delete](#) key.

→ Tip



You can also delete a repository object by right-clicking it and selecting Delete from the shortcut menu. You can also click the "Delete the item/folder" button on the Repository Explorer's toolbar.

The "Delete Repository Objects?" dialog box appears.



2. Click [Yes](#) to confirm deletion of the selected object and its dependent objects.

The "Delete Repository Objects?" dialog box displays a Dependency Graph, which displays all of the dependent objects (child objects) that also need to be deleted as well. This dialog box also notes that once the selected object and its dependent objects are deleted, these objects are deleted permanently.

4.4.5 Installing sample repository objects

By default, the SAP BusinessObjects repository does not include any sample repository objects. You need to install the sample repository objects to the SAP BusinessObjects repository on your Central Management Server (CMS).

When you install the sample repository objects, the following folders and objects are added to the SAP BusinessObjects repository:

- **Commands**
This folder includes sample custom commands.
- **Custom Functions**
This folder includes a number of custom functions.
- **Images**
This folder contains sample logos.
- **Samples > Business Views > Xtreme**
This folder contains one Data Connection, one Data Foundation, five Business Elements, and one Business View.
- **Text Objects**
This folder contains sample text objects.

Note

Regardless of whether or not your product licensing includes the use of Business Views, you can install the sample repository objects through the Business View Manager.

Note

When you install the sample repository objects, you choose the language you want for the samples. For example, you can install Korean samples on an English SAP BusinessObjects Business Intelligence platform installation.

Note

Using the Business View Manager, you can install samples to any repository, regardless of whether the Central Management Server is located on a Windows or a UNIX machine.

Note

For more information on custom functions, commands, images, and text objects, see the SAP Crystal Reports User's Guide.

4.4.5.1 To install the sample repository objects

1. On the *Tools* menu, select *Install Repository Samples*.

The "Install Repository Samples" dialog box appears.

2. Select the language for your sample repository objects from the *Choose Language* list.
3. Click *OK*.

ⓘ Note

Any existing sample repository objects will be deleted before the new objects are installed.

4.5 SAP BusinessObjects repository security model

The security model for Business Views is based directly on the SAP BusinessObjects Business Intelligence platform security model; as such, this flexible security model allows you to be as general or as specific as you want in setting rights for users and groups. You can set these rights through the Repository Explorer, which determines whether users and groups can access and edit specific Business Views objects through the Business View Manager.

ⓘ Note

This section of the documentation details the SAP BusinessObjects security as it applies to the folders stored in the repository; it does not include detailed information on specific security information for the various Business Views objects—for this information on object security, refer to the specific section for each object, and also refer to [Security overview \[page 187\]](#).

Object rights are the base units for controlling users' access to folders and other objects within the repository. When granted, each right provides a user or group with permission to perform a particular action on an object. For any object, you can set security levels that affect individual users or entire groups.

To set object rights within the repository, you first locate the object in the Repository Explorer, and then you specify the rights for different users and groups. Each object right can be explicitly granted or denied. The object security model is designed such that, if a right is left "not specified," the right is denied by default. Additionally, if contradictory settings result in a right being both granted and denied to a user or group, the right is denied by default. This "denial based" design assists in ensuring that users and groups do not automatically acquire rights that are not explicitly granted.

By setting rights through group membership, you can deny or grant users of a certain group specific object rights. You can take advantage of the inheritance patterns recognized by the security model: users can inherit rights as the result of group membership; subgroups can inherit rights from parent groups; and both users and groups can inherit rights from parent folders. When you need to disable inheritance or to customize security levels for particular objects, users, or groups, you can disable these rights through the Business View Manager.

ⓘ Note

For more information on security as it applies to Business Views objects, see [Security overview \[page 187\]](#).

Note

For more information on the inheritance model of security as it applies to folders and Business Views objects, see [Using inheritance to your advantage \[page 192\]](#).

Note

For more information on security as it applies to SAP BusinessObjects Business Intelligence platform, see the *SAP BusinessObjects Business Intelligence platform Administrator Guide*.

4.5.1 Applying security settings to folders in the repository

Using the Repository Explorer, you can specify which user and/or group has access to specific folders in the repository. Folders provide you with the ability to organize and facilitate content administration. They are useful when there are a number of Business Views objects that a department or area requires frequent access to, because you can set object rights and limits once, at the folder level, rather than setting them for each object within the folder.

Note that folder rights follow the same inheritance model as object rights. Thus, if you set a right on a parent folder, all of its children folders inherit the same security rights. For more information on the inheritance model as it pertains to security, see [Using inheritance to your advantage \[page 192\]](#).

Before you can apply security rights to a folder, you must have Set Security rights to the folder. By default, members of the Administrators group have full access to all of the folders in the repository.

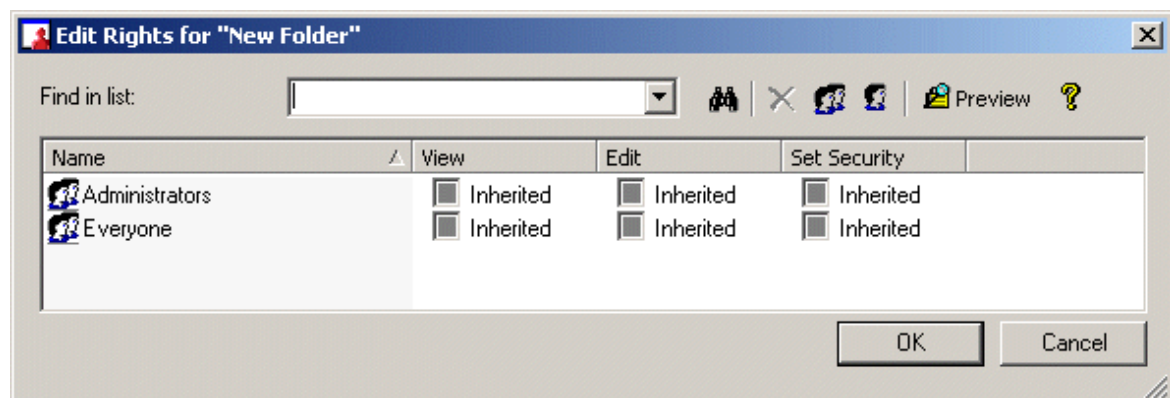
Note

For detailed information on the Edit Rights dialog box, see [Using the Edit Rights dialog box \[page 188\]](#).

4.5.1.1 To apply security settings to a folder

1. In the Repository Explorer, right-click a folder and select [Edit Rights](#).

The Edit Rights dialog box appears.

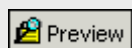


2. Set the following rights for either a user or a group by clicking the appropriate check box under each rights column.
 - View
This right specifies whether a user or group can or cannot view a folder.
 - Edit
This right specifies whether a user or group can or cannot edit the properties of a folder.
 - Set Security
This right specifies whether a user or group can or cannot modify the rights that are associated with a folder.



Note

A cleared check box indicates that the user or group is denied the right; a check mark in the check box indicates that the user or group is granted the right; a shaded check box indicates that the right is inherited.

Note



Click the Preview button to view the final security settings for the object with inheritance in effect. The Business View Manager resolves the net result of the current security settings and displays the net rights.

3.   To add another user or group, click the [Add Users](#) button or click the [Add Groups](#) button.
4. Assign rights to the new user or group as needed.
5. Click [OK](#).

4.5.2 Viewing folder rights

If a user or group does not have View rights to a folder, the user or users in that group will not be able to view the folder in the Business View Manager, nor will they be able to view the folder's security settings.

To view a folder's rights settings, in the Repository Explorer, right-click the folder and select Edit Rights.

4.6 Creating a list of values

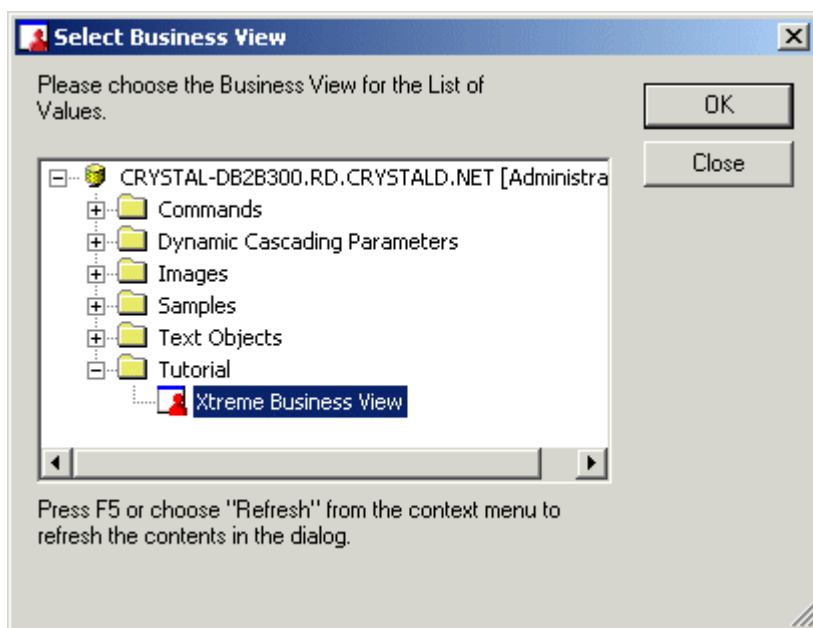
A list of values is an object that contains the values of specific fields in a Business View. Using the Business View Manager, you can create a list of values and use the values of the fields as values for dynamic and/or dynamic cascading parameters. (For more information about parameters, see [Inserting a parameter \[page 127\]](#) for parameters in Data Foundations and [Inserting a parameter \[page 151\]](#) for parameters in Business Elements.)

Lists of values are stored in the repository. Like a Crystal report, you can use SAP BusinessObjects Business Intelligence platform to schedule a list of values to obtain/update the values of the fields that it references from a Business View.

4.6.1 To create a list of values

1. From the *File* menu, select *New* and then select *Create List of Values Info*.

The Select Business View dialog box appears.



2. Expand the folders and select the Business View that contains the fields that you want for your list of values.
3. Click *OK*.

The Create List of Values dialog box appears.

Create List of Values

Name:

Available Fields:

- Database Fields
 - Customer
 - Customer ID
 - Customer Name
 - Contact First Name
 - Contact Last Name
 - Contact Title
 - Contact Position
 - Last Year's Sales
 - Address1
 - Address2
 - City
 - Region

List of Value Fields:

- Customer Name
- City
- Phone

☒ Sort by Value in Ascending Order
☐ Sort by Value in Descending Order

Description Field:

Oldest on-demand data given to a client (in seconds):

Default servers to use for viewing:

Referenced Business View:

List of Values Schedule Result:

4. In the *Name* field, specify a name for the list of values.
5. In the Available Fields area, expand the tables and select a field that you want to include in the list.
6. Click the right arrow to add the field to the list.
7. Continue to select and add all of the fields that you want to include in the list of values.
8. In the List of Value Fields area, select a field.
 - *To change the order of the fields in your list, click the Up or Down arrow.*

The order of the fields in the list establishes the order in which information is prompted for the fields in a parameter. For example, suppose that the fields are listed in the following order:

- Country
- Region
- City

A parameter that is based on the list of values will first prompt users for a country, then for a region within the country, and finally, for a city within the region.

Note

The order of the fields in a list of values is especially important if you plan to use the list of values for dynamic cascading parameters. If the parameter is not cascading, then it will always prompt for the first field in the list of values (which, in this case, is "Country").

- *To specify a description for the field, click the ... button beside the Description Field.*
When you click the button, a list that contains the tables and fields in the Business View appears. From the list, you can select a field. The values of the field that you choose become the descriptions for the values of the field that is selected in the List of Value Fields area.
For example, suppose that, in the List of Value Fields area, you select the field "Customer ID." When you click the ... button, you choose the field "Customer Name." When you create a parameter that uses the list of values (and configure the prompt to use Value and Description), the values that are entered for the parameter are the values of the "Customer ID" field; however, you specify these values by selecting a customer name. The description allows you to provide a value for that parameter without having to remember specific customer IDs. You only have to provide a name.
- *To sort the order of the values of the field, select Ascending by Value or Descending by Value in the Sort Order area.*
You can specify the order in which the values of the selected field appear in the list of values. When you create a parameter that uses the list of values, the order that you specified for the field is the order in which the field's values appear in the prompt.

9. Click *OK*.

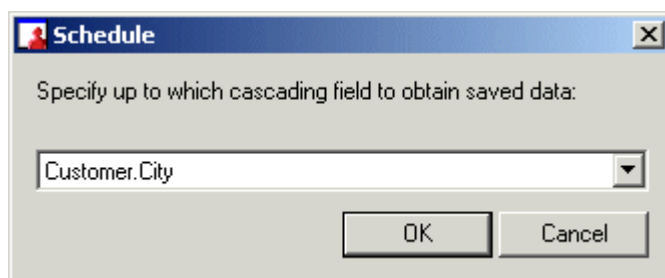
The Business View Manager creates the list of values and saves it in the top-level folder of the repository. To modify the list of values, right-click the object in the Repository Explorer and select Edit list of values.

For instructions on how to use the list of values for your parameters, see [Inserting a parameter \[page 127\]](#) for parameters in Data Foundations and [Inserting a parameter \[page 151\]](#) for parameters in Business Elements.

4.6.2 To schedule a list of values

1. In the Repository Explorer, right-click a list of values, and click *Schedule LOV*.

The Schedule dialog box appears.



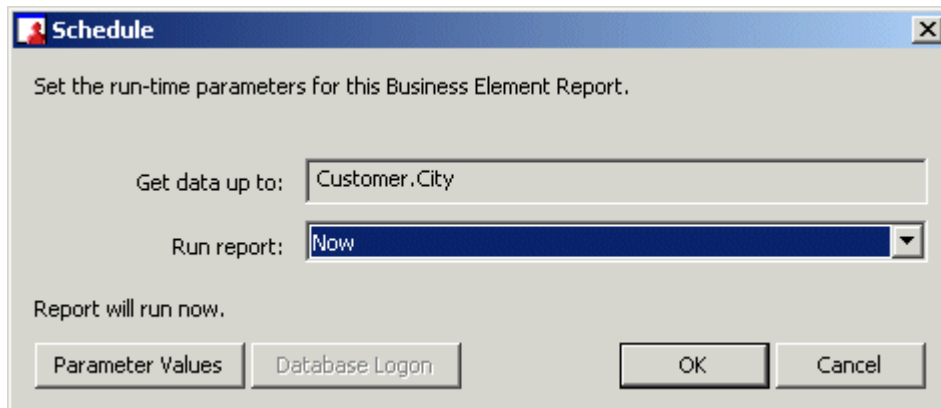
2. From the drop-down list, select a field.

Note

The field that you specify is the level up to which data is obtained/saved for dynamic cascading parameters. For example, if the list of values consists of "Country", "Region", and "City" levels, and you choose "Region", data is only obtained for the "Country" and "Region" fields; data is not obtained for the "City" field.

3. Click *OK*.

The second Schedule dialog box appears.



4. In the Run report list, choose the schedule in which the list of values will be run. (For more information about the different ways that you can schedule an object, see the *SAP BusinessObjects Business Intelligence platform Administrator Guide*).

Note

If your list of values is based on a Business View that contains parameters, click the Parameter Values button (at the bottom-left corner of the dialog box) to specify values for the parameters. The button is disabled if the Business View does not contain parameters.

Note

If your list of values is based on a Business View that requires database login, you must also click on the Database Logon button to connect to the Business View's data source. The button is disabled if the Business View does not require database login.

5. Click **OK**.

The list of values is scheduled in SAP BusinessObjects Business Intelligence platform.

4.7 Changing the Business Views Used by a List of Values

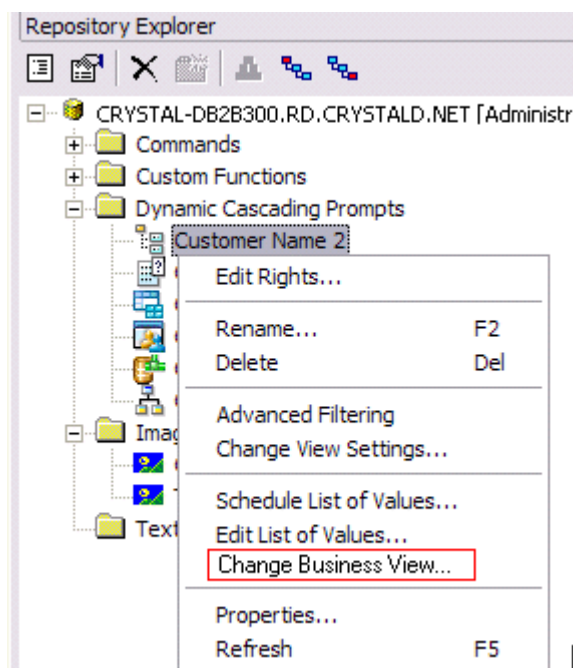
Modifying a List of Values object is useful in the following situations:

- Data sources can change over time, and you want to update the list of values objects to reflect the latest changes in a Business View. For example, additional fields might be added to a database table that you want to show in an existing list of values object.
- To reduce the number of Business Views you need to maintain, you can consolidate multiple Business Views into a single Business View used for List of Values objects. To implement this change you need to modify existing List of Values objects to use the new Business View.
- You want to modify the fields used by an existing List of Values object to use different fields in the same Business View.

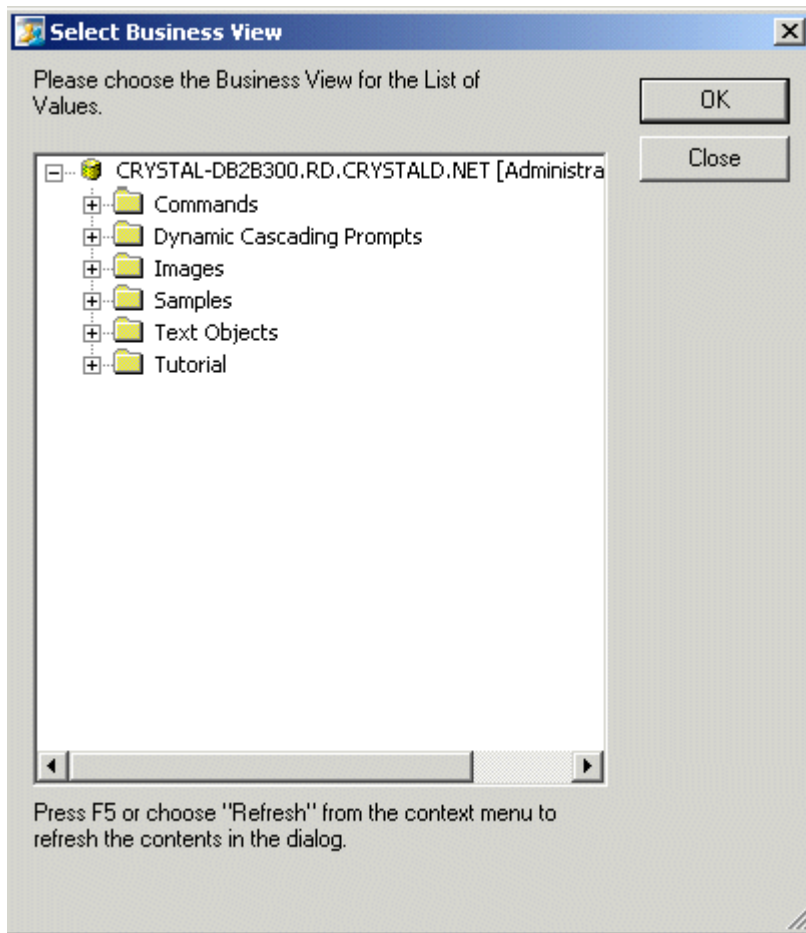
To make these changes, you can modify the Business View used by a List of Values. The new Business View to be used by the List of Values must be data type compatible with the existing Business View. The new Business View can be different from the current Business View, or it can be the same.

4.7.1 To change Business Views

1. In the Repository Explorer, right-click on the List of Values object and choose *Change Business Views*.



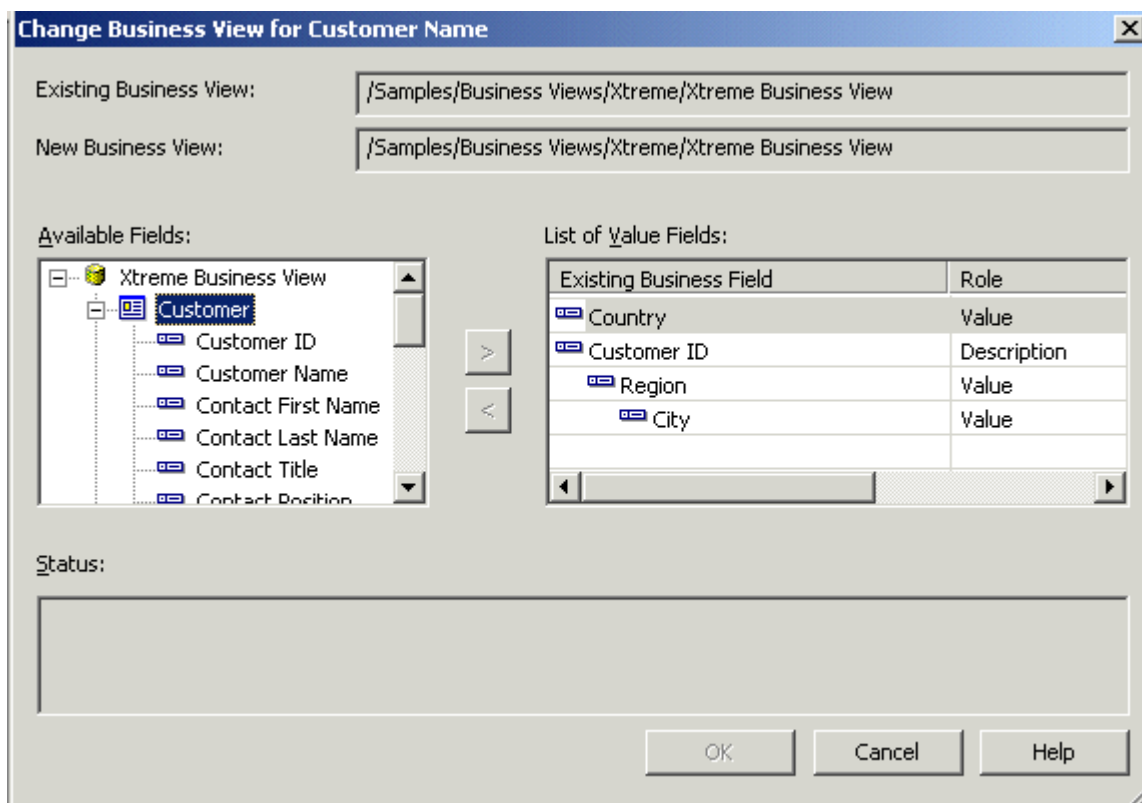
The Select Business View dialog box appears.



2. Expand the folders and select a new Business View for the List of Values.
3. Click [OK](#).
4. The next dialog shows you the new Business View, Business Element(s) and Business Field(s) on the left hand side in the 'Available Fields' list, and the existing List of Values fields on the right hand side. From the 'Available Fields' part of the dialog, select the desired field and click > and this new field will be mapped to the selected List of Value field on the right. Click < to unmap a selected List of Value field.

Note

The data type of the Business Fields and List of Value fields must be compatible. If the data types of the selections do not match, the > button will be greyed out. Description fields in the List of Values can only be modified if they were set in the original List of Values object. Each field can only be used once and description fields can only be modified if they were set in the old List of Values object.



- When you have completed choosing the new business fields, click **OK** to have the changes committed or click **Cancel** to leave the existing List of Values unchanged.

Note

The OK button will only be highlighted after all the fields have been mapped. If the List of Values has a scheduled instance then a message should appear on the dialog to the left of the OK button: "The List of Values object has a scheduled data. Changing the Business View will remove the scheduled data." Clicking OK will commit the changes to the CMS as well as delete the scheduled instance (if it exists). If all is configured accurately, all existing prompt groups and reports will continue to work.

5 Creating Data Connections

This section details how to create and modify Data Connections through the Business View Manager.

5.1 Data Connections overview

Data Connections specify and define the data sources for Business Views. They specify which physical data sources are made available to the system and how these data sources are made available. Each Data Connection contains information that specifies the physical data source and its relevant settings, which include the data being accessed, the logon credentials, the type of server being accessed, and so on.

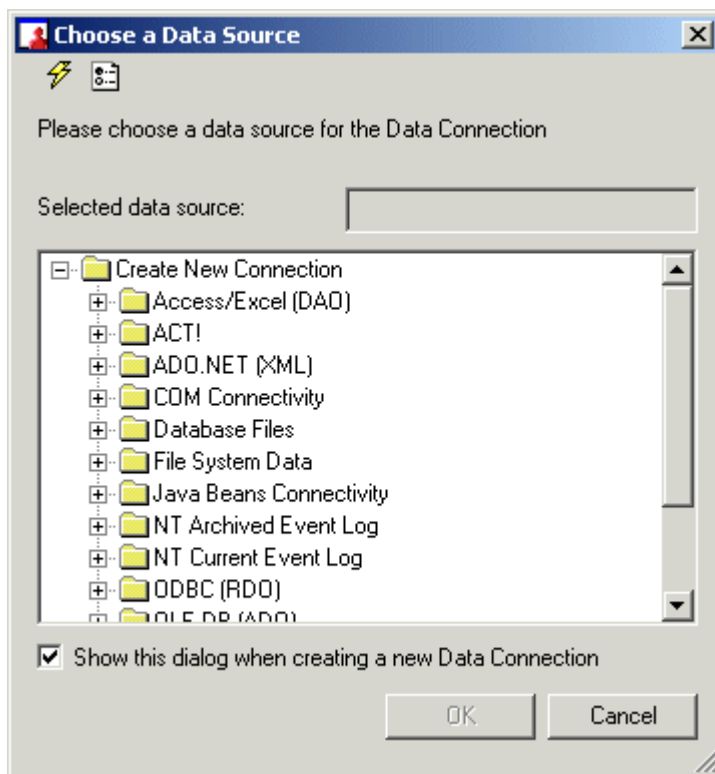
Administrators can apply security settings to Data Connections. Data Connections also pass on a user's credentials to the underlying data source. These user credentials are set when you create or modify a Data Connection.

5.2 Working with Data Connections

When you create a Data Connection, you need to specify and define a data source, along with the necessary connection information. This section details the different data sources that are available. It also details how to create a new Data Connection, how to set the Data Connection password, and how to modify a Data Connection. For more general information on databases, see the "Understanding Databases" section in the SAP Crystal Reports Online Help.

5.2.1 Data sources

Data sources can be chosen through the "Choose a Data Source" dialog box. This dialog box appears when you create a Data Connection from scratch, or when you choose Edit Connection from the Edit menu while you are working with a Data Connection.



The tree view in the "Choose a Data Source" dialog box lists all of the possible connections for your data source.

Note

The data source options available in the Create New Connection folder depend on the data access components selected during installation.

5.2.2 To create a Data Connection

When you create a new Data Connection, you specify the data source information for the Data Connection.

Note

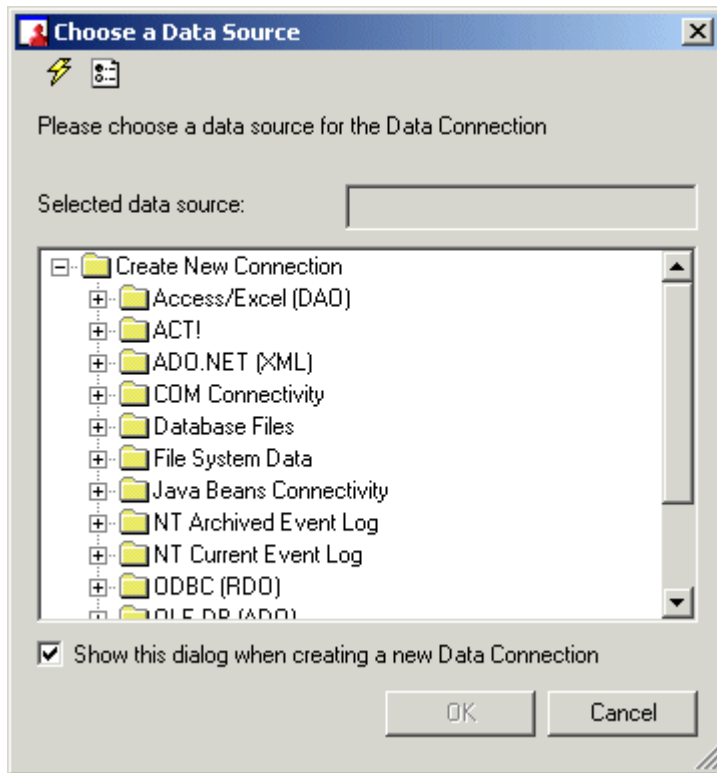


You can update this information in the Data Connection's Property Browser window. Alternatively, you can select Edit Connection on the Edit menu to update the data source information, or you can also click the Edit Connection button on the toolbar.

For more information on the different data sources that are available, see [Data sources \[page 77\]](#).

1. On the **File** menu, select **New**, and then select **Data Connection**.

The "Choose a Data Source" dialog box appears.



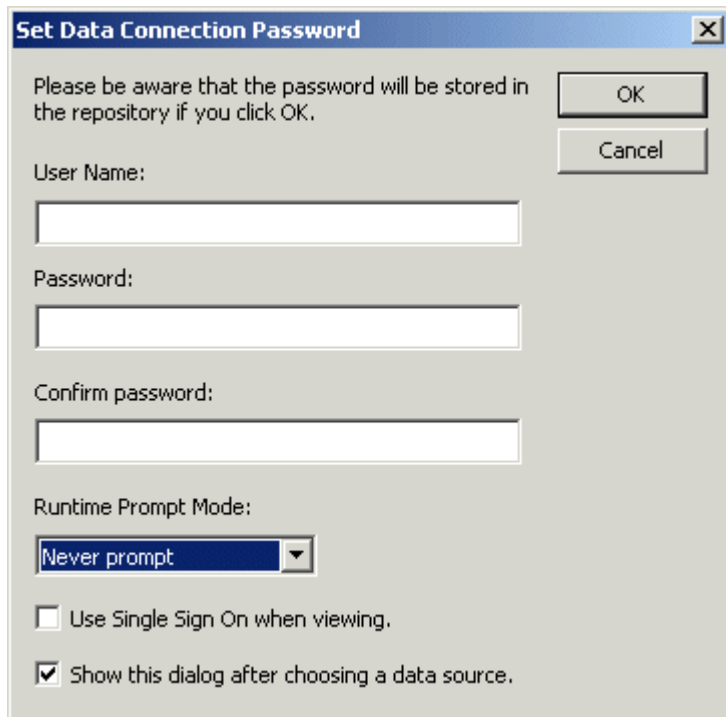
2. Expand the folders as necessary and select a data source.

Note

Depending on which data source you select, different dialog boxes that are related to the data source type appear. Navigate through these dialog boxes and provide the appropriate data source information as needed.

3. Click [OK](#) to continue.

The "Set Data Connection Password" dialog box appears.



Set Data Connection Password

Please be aware that the password will be stored in the repository if you click OK.

OK

Cancel

User Name:

Password:

Confirm password:

Runtime Prompt Mode:

Never prompt

☐ Use Single Sign On when viewing.

☒ Show this dialog after choosing a data source.

5.2.3 To set the Data Connection password

When you set the Data Connection user name and password, this information is stored in the SAP BusinessObjects repository.

Note

You can update this information in the Data Connection's Property Browser window. Alternatively, you can select Edit Password on the Edit menu.

1. In the "Set Data Connection Password" dialog box, type the appropriate *User Name* and *Password*. Type the password again in the *Confirm password* field.
2. In the *Runtime Prompt Mode* list, select one of the two Runtime Prompt Mode options:
 - Always prompt
Select Always Prompt if you want the user to be prompted for log on information at runtime.
 - Never prompt
Select Never Prompt if you have stored logon credentials in the repository (as in step 1). The user will not be prompted for the credentials at runtime. (The stored logon credentials will be used at runtime.)
3. If you want to enable Single Sign On for the Data Connection, select the "Use Single Sign On when viewing" checkbox. (For information about the Single Sign On feature, see [Single Sign On \[page 81\]](#).)
4. Click **OK**.

5.2.3.1 Single Sign On

Single Sign On enables users to access the data source of a Business View with the credentials that they use to log on to SAP BusinessObjects Business Intelligence platform.

By default, when users click to view a report that is based on a Business View, a logon prompt for the Business View's data source appears. The users must then provide a valid User Name and Password for the data source before the information in the report is displayed.

As the Business Views administrator, you can disable this prompt by configuring the Business View's Data Connection in one of the following ways:

- You can specify a User Name and a Password for the Data Connection, and set the Runtime Prompt Mode to "Never prompt."
- You can enable Single Sign On.

If you provide logon credentials for the Data Connection and choose "Never prompt" as the Runtime Prompt Mode (as instructed in [To set the Data Connection password \[page 80\]](#)), the User Name and Password that you specify are stored with the Data Connection in the repository. When users attempt to view a report that is based on a Business View that uses this Data Connection, the credentials that you provided are used to log on to the data source; users are never prompted for logon information.

Note

In this configuration, the rights that users have for the data source depend on the rights of the User Name and Password that you specified.

If, instead, you choose to enable Single Sign On, you configure the Data Connection to log on to the data source with the SAP BusinessObjects logon credentials of the users who try to view the report. If the users are successfully logged on to SAP BusinessObjects BI platform, they will not be prompted to log on to the data source when they click to view the report.

To enable Single Sign On, select the "Use Single Sign On when viewing" check box when you create the Data Connection. (To enable Single Sign On for a Data Connection that already exists, open the Data Connection in the Business View Manager and, in the Property Browser, set the "Use Single Sign On when viewing" property to "True".)

Note

In this configuration, the rights that users have for the data source depend on their rights in SAP BusinessObjects BI platform. The SAP BusinessObjects administrator can customize the data source access rights for each user.

Note

If you enable Single Sign On, you can leave the User Name and Password fields in the Data Connection empty.

In addition to configuring the Data Connections, you must provide connection information for the data sources in SAP BusinessObjects BI platform to enable Single Sign On.

Single Sign On works only when you view reports that are based on Business Views from within the SAP BusinessObjects BI platform environment (that is, when you are logged on to SAP BusinessObjects BI platform and you click to view the reports in InfoView). If you view the reports outside of SAP BusinessObjects BI

platform (for example, from within SAP Crystal Reports), you are prompted to log on to the data source, whether Single Sign On is enabled or not.

Note

Single Sign On also works for Data Connections that are referenced by Dynamic Data Connections. (For more information about Dynamic Data Connections, see [Dynamic Data Connections overview \[page 87\]](#).)

5.2.4 Testing and verifying data connectivity and object dependency



To verify your database connection, select Test Connectivity on the Tools menu. Business Views connects to the data source specified in your Data Connection and tests whether it can connect successfully to your data source or not. Alternatively, you can click the Test Connectivity button located on the toolbar.



To test any objects that are dependent on the Data Connection, select Check Dependent Integrity on the Tools menu. Alternatively, you can click the Check Dependent Integrity button located on the toolbar. For example, if you change the data source for your Data Connection, this change can affect a Data Foundation that uses the information found in the Data Connection—certain tables and fields that the Data Foundation is based on may not exist in the different data source.

5.2.5 Showing dependent and referenced objects

You can view the Business Views objects that depend on your Data Connection. Similarly, you can view the Business Views objects that are referenced by your Data Connection.

To view dependent or referenced objects, in the Tools menu, select either Show Dependent Objects or Show Referenced Objects. A dialog box appears; in this dialog box, click the Save To File button to save the list of dependent or referenced objects to a text file for future reference.

The list of dependent objects displays objects that are affected by the Data Connection (for example, Dynamic Data Connections, Data Foundations, Business Elements, and Business Views). The list of referenced objects displays objects that the Data Connection references. (Since a Data Connection is the first object you must create, it does not reference any other Business Views objects.) Expand the nodes to see all of the objects that depend on or are referenced by your Data Connection.

5.2.6 Modifying a Data Connection

After you create a Data Connection, the main window displays the data source information in a table. This window includes data source information such as the database's DLL, the DSN connection, and so on. Depending on the data source, the type and amount of information in this window will be different.

You can edit the data source and the connection user name and password information for a Data Connection. In addition, you can edit all of the properties that are associated with the Data Connection.

You can edit the user and group rights associated with the Data Connection. For more information on Data Connection security, see [Editing rights for a Data Connection \[page 85\]](#).

5.2.6.1 Using the Property Browser

The Property Browser lists all of the properties that you can edit or change for a Data Connection. You can specify some of these settings when you first create a Data Connection; alternatively, you can choose some of these settings through the Edit menu or by clicking the appropriate buttons on the toolbar.

- **Name**
The name of the Data Connection.
- **Description**
A description you enter for the Data Connection. This description appears when you point your mouse over the specific Data Connection in the Repository Explorer or when you have to choose a Data Connection. This information also appears when you right-click a Data Connection in the Repository Explorer and select Properties.
- **Author**
By default, this field contains the name of the user who created the Data Connection. The name of the author appears when you point your mouse over the specific Data Connection in the Repository Explorer or when you have to choose a Data Connection.
- **Parent Folder**
The repository folder in which the Data Connection is located. This is the only property that cannot be directly modified through the Property Browser.
- **User Name**
The user name that is passed to the data source. This user name is stored in the repository.
- **Password**
The password for the Data Connection. To change the password, ensure that the Password cell is selected and then click the ... button. For more information on the user's password, see [To set the Data Connection password \[page 80\]](#).
- **Use Single Sign On when viewing**
Allows users to use their SAP BusinessObjects BI platform User Name and Password to log on to the Data Connection's associated data source. For more information, see [Single Sign On \[page 81\]](#).
- **Connection**
The data source for the Data Connection. To change the data source, ensure that the Connection cell is selected and then click the ... button. For more information on the different data sources, see [Data sources \[page 77\]](#).
- **Runtime Prompt Mode**
Specifies whether or not the user is prompted for his or her user name and password at runtime. For more information, see [To set the Data Connection password \[page 80\]](#).
- **Use Owner**
Specifies whether the Data Connection uses the dbo portion of a table name (for example, pubs.dbo.authors) or not.
- **Use Catalog**

Specifies whether the Data Connection uses the pubs portion of a table name (for example, pubs.dbo.authors) or not. For example, if the Use Owner property is set to False, and the Use Catalog property is set to True, the Data Connection uses the following syntax: "pubs".""."authors".

Note

Changing this setting to False enables you to use Dynamic Data Connections that point to Data Connections with different schemas (not recommended).

- **Rights**
The user and group rights that are associated with the Data Connection. To update the rights, ensure that the Rights cell is selected and then click the ... button. For more information on editing rights, see [Editing rights for a Data Connection \[page 85\]](#).

You can also view some of the Data Connection's properties by right-clicking a Data Connection in the Repository Explorer and selecting Properties. The Properties dialog box displays the name of the object, the type of object, and the date the object was last saved. This dialog box also displays the object's description.

5.2.6.2 Using the Object Explorer

The Object Explorer displays the name of the Data Connection. By right-clicking the Data Connection, you can perform the following functions:

- **Edit Connection**
When you select Edit Connection, the "Choose a Data Source" dialog box appears. You can change the data source for your Data Connection here. For more information on choosing a data source, see [To create a Data Connection \[page 78\]](#). For more information on the different available data sources, see [Data sources \[page 77\]](#).
- **Edit Password**
Choose this function to change the login information for the Data Connection. You can also specify whether a user is prompted for information at runtime or not. For more information on the Data Connection login information, see [To set the Data Connection password \[page 80\]](#).
- **Edit Rights**
You can edit user and group rights for a Data Connection. For more information on editing rights, see [Editing rights for a Data Connection \[page 85\]](#).

5.2.7 To save a Data Connection object

Data Connections, like all of the other Business Views objects, are saved to the repository. The repository is contained in the Central Management Server (CMS) of SAP BusinessObjects BI platform. For more information on the repository, see [Using the Repository Explorer \[page 62\]](#).

When you first save a Data Connection, you need to specify a name for the object and the location you want to save the object to.

1. On the **File** menu, select **Save**.

→ Tip



You can also click the Save (Ctrl+S) button or press CTRL+S.

The Save As dialog box appears.

2. In the *Object Name* field, specify the name of your object.
3. Select the folder you want to save your object to.

→ Tip



You can create new folders in the repository by clicking the "Insert a new folder" button.

4. Click [Save](#).

5.2.8 Editing rights for a Data Connection

By editing the user and group rights for a Data Connection, you ensure that only specific users and groups can access the Data Connection. Since the Business Views security model is based on the SAP BusinessObjects BI platform security model, you have a flexible security model that you can work from. For example, if you explicitly deny a View right for a user or group for a specific Data Connection, this user or group, when creating a Dynamic Data Connection, or a Data Foundation, will not be able to view or select from any Data Connection that they do not have View rights to.

ⓘ Note

Before you can edit the security rights for a Data Connection, you must first save the Data Connection to the repository.

Business Views object rights are dependent on folder rights. If you set a right on a folder, all of the objects in this folder inherit the same security rights. For more information on the inheritance model as it pertains to security, see [Using inheritance to your advantage \[page 192\]](#).

Before you can apply security rights to an object, you must have Set Security rights to the object. By default, members of the Administrators group have full access to all of the folders and objects in the repository.

ⓘ Note

For detailed information on the Edit Rights dialog box, see [Using the Edit Rights dialog box \[page 188\]](#).

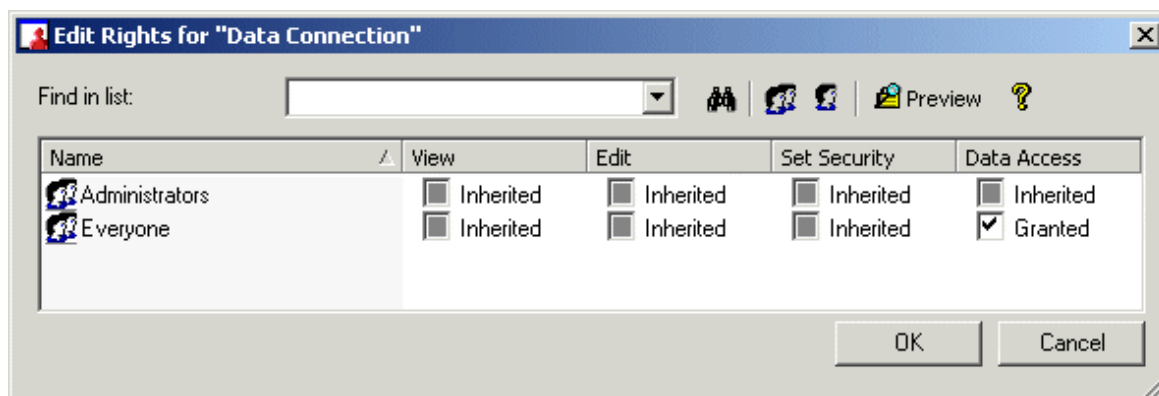
5.2.8.1 To apply security settings to a Data Connection

1. In the Object Explorer, right-click the Data Connection, and select [Edit Rights](#).

→ Tip

You can also select the Rights cell in the Property Browser and click the ... button. Alternatively, you can select Edit Rights on the Edit menu.

The Edit Rights dialog box appears.

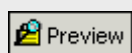


2. Set the following rights for either a user or a group by clicking the appropriate check box under each rights column.
 - View
This right specifies whether a user or group can or cannot view the Data Connection.
 - Edit
This right specifies whether a user or group can or cannot edit the properties of the Data Connection.
 - Set Security
This right specifies whether a user or group can or cannot modify the rights that are associated with the Data Connection.
 - Data Access
This right specifies whether a user or group can access the specified data source in the Data Connection. Note that this right cannot be inherited from other rights: a user or group either explicitly has data access or does not.



Note

A cleared check box indicates that the user or group is denied the right; a check mark in the check box indicates that the user or group is granted the right; a shaded check box indicates that the right is inherited. Also, note that a right may be shown as being granted, but may be denied once inheritance is taken into account.

Note



Click the Preview button to view the final security settings for the object with inheritance in effect. The Business View Manager resolves the net result of the current security settings and displays the net rights.

3.   To add another user or group, click the [Add Users](#) button or click the [Add Groups](#) button.
4. Assign rights to the new user or group as needed.
5. Click [OK](#).

6 Creating Dynamic Data Connections

This section details how to create and modify Dynamic Data Connections through the Business View Manager, and provides information on how Dynamic Data Connections point to specific Data Connections.

6.1 Dynamic Data Connections overview

Dynamic Data Connections consist of a collection of pointers to different Data Connections; through an exposed parameter, administrators or users can specify which Data Connection they wish to use.

6.1.1 Why use a Dynamic Data Connection?

When you create Business Views objects, you need to first specify your data source by creating a Data Connection. A Data Foundation then uses the information found in the data source's physical database properties (tables and fields, for example) for data abstraction.

A Dynamic Data Connection is a collection of pointers to different Data Connections—thus, if you need to switch between different data sources that contain the same database schema, you would use a Dynamic Data Connection. Then, when you create a Data Foundation, you can choose a Dynamic Data Connection as your data source connection and choose which Data Connection you want to use.

A Dynamic Data Connection enables you to change between data sources as needed, whereas a Data Connection is more static. As such, a Dynamic Data Connection provides additional flexibility in terms of specifying a data source, but is not a mandatory object when you create your Business View—you can create Data Foundations with or without using a Dynamic Data Connection. Note that you must first create more than one Data Connection before you can create a Dynamic Data Connection.

Many corporations use a development, test, and production system model for housing their data. That is, the data is initially stored on a development system and then later migrated to a test system. After extensive testing, the data in the test system is ready to be used in the production system.

Using a Dynamic Data Connection, you can specify three Data Connections—one for each system. When the data is moved from one system to another, administrators can test reports generated from each system simply by selecting the desired connection through the Dynamic Data Connection. As long as the database schema is the same for all three databases, administrators can easily ensure that reports point to and use the proper data source as required.

6.2 Working with Dynamic Data Connections

When you create a Dynamic Data Connection, you need to add and specify Data Connections. Since a Dynamic Data Connection represents a collection of physical data sources (from different Data Connections), the database schema for these data sources must be the same.

Note

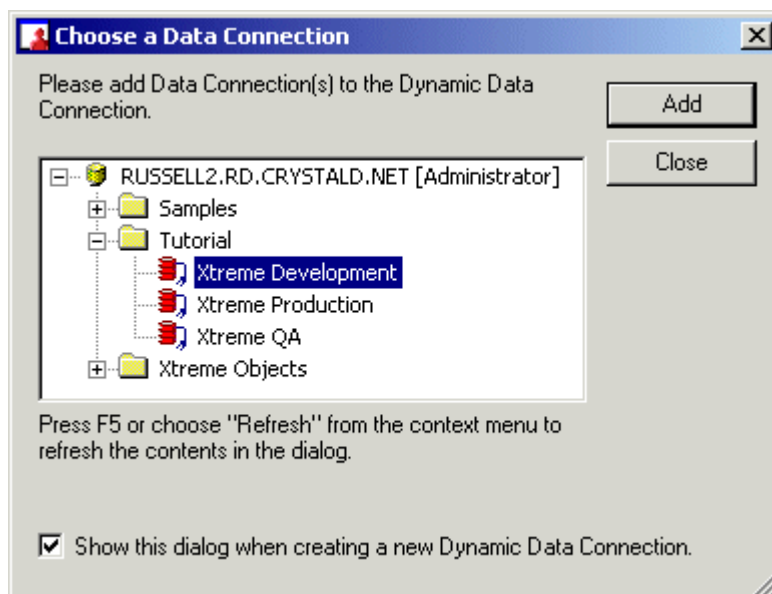
Only Data Connections that do not always prompt for a user to log on to a data source—that is, Data Connections that automatically pass the user name and password to the database—can be used for a Dynamic Data Connection. Thus, the Runtime Prompt Mode for the Data Connection needs to be set to Never prompt. For more information on the user information stored in a Data Connection, see [To set the Data Connection password \[page 80\]](#).

6.2.1 To create a Dynamic Data Connection

To create a new Dynamic Data Connection, you specify the different Data Connections that you want to include in your Dynamic Data Connection. You can add and delete different Data Connections at any time.

1. On the **File** menu, select **New**, and then select **Dynamic Data Connection**.

The "Choose a Data Connection" dialog box appears.



2. Expand the folders as necessary and select a Data Connection.
3. Click **Add**.

Note

Only Data Connections that do not always prompt for a user to log on to a data source can be used for a Dynamic Data Connection. You will receive an error message if you attempt to add a Data Connection that always prompts a user to log on.

4. Continue selecting and adding Data Connections as necessary; click [Close](#) when you are finished.

6.2.2 Verifying object dependency



You can verify the dependency of other objects that are affected by the settings you specify in the Dynamic Data Connection. Select Check Dependent Integrity on the Tools menu (or click the Check Dependent Integrity button located on the toolbar) to test the Dynamic Data Connection and the Business Views objects that depend on the settings specified in the Dynamic Data Connection.

Changes you make to the Dynamic Data Connection can affect Data Foundations, Business Elements, and Business Views.

6.2.3 Showing dependent and referenced objects

You can view the Business Views objects that depend on your Dynamic Data Connection. Similarly, you can view the Business Views objects that are referenced by your Dynamic Data Connection.

To view dependent or referenced objects, in the Tools menu, select either Show Dependent Objects or Show Referenced Objects. A dialog box appears; in this dialog box, click the Save To File button to save the list of dependent or referenced objects to a text file for future reference.

The list of dependent objects displays objects that are affected by the Dynamic Data Connection (for example, Data Foundations, Business Elements, and Business Views). The list of referenced objects displays objects that the Dynamic Data Connection references (Data Connections). Expand the nodes to see all of the objects that depend on or are referenced by your Dynamic Data Connection.

6.2.4 Modifying a Dynamic Data Connection

The main window of a Dynamic Data Connection displays all of the available Data Connections. In this window, you can add more Data Connections or delete existing ones. You can also sort how the Data Connections are displayed. You can edit the properties of a Dynamic Data Connection.

A Dynamic Data Connection, like all of the other Business Views objects, follows the SAP BusinessObjects security model. As such, you can edit user and group rights. For more information on security for Dynamic Data Connections, see [Editing rights for a Dynamic Data Connection \[page 94\]](#).

6.2.4.1 Adding and deleting Data Connections

In the main window of a Dynamic Data Connection, you can add new Data Connections or delete existing ones. Using the Repository Explorer, you can also add a new Data Connection by selecting it, and then dragging and dropping the Data Connection to the main window.

6.2.4.1.1 To add a Data Connection

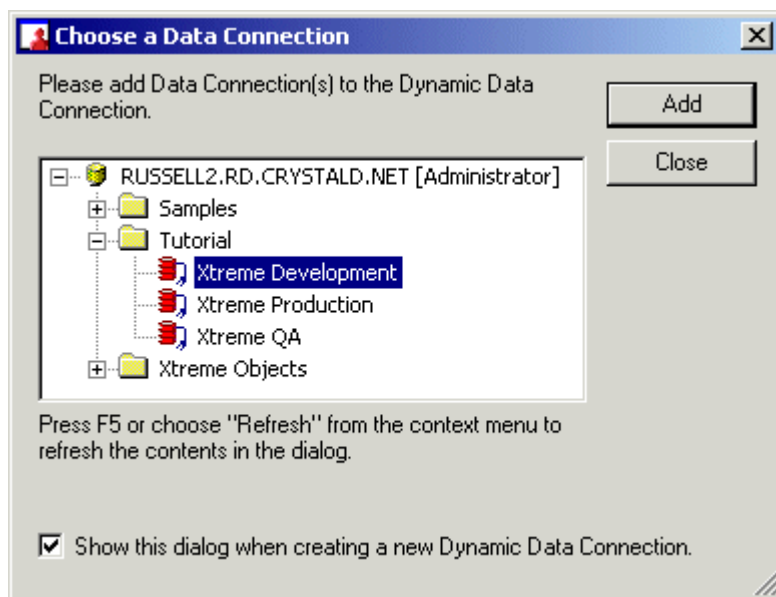
1. In the main window, click [Add](#).

→ Tip



You can also click the Add Data Connection button on the toolbar; alternatively, you can select Add Data Connection from the Edit menu.

The "Choose a Data Connection" dialog box appears.



2. Expand the folders as necessary and select a Data Connection.
3. Click [Add](#).

ⓘ Note

Only Data Connections that do not always prompt for a user to log on to a data source can be used for a Dynamic Data Connection. You will receive an error message if you attempt to add a Data Connection that always prompts a user to log on.

4. Continue selecting and adding Data Connections as necessary; click [Close](#) when you are finished.

6.2.4.1.2 To delete a Data Connection

1. In the main window, select a Data Connection.

Note

You can delete multiple Data Connections at one time by selecting more than one Data Connection.

2. Click *Delete*.

→ Tip



You can also click the Delete Data Connection button on the toolbar; alternatively, you can select Delete Data Connection on the Edit menu.

6.2.4.2 Sorting Data Connections

If you have a Dynamic Data Connection with numerous Data Connections, you can sort the listing of your Data Connections. You can also move a Data Connection up or down the list by first selecting the object and then clicking the up or down arrow located on the top right-hand corner of the main window.

To sort your Data Connections, you can choose from three options in the sort list:

- Alphabetical ascending
- Alphabetical descending
- No Sort

6.2.4.3 Using the Property Browser

The Property Browser displays all of the properties that you can edit or change for a Dynamic Data Connection object.

- *Name*
The name of the Dynamic Data Connection.
- *Description*
A description you enter for the Dynamic Data Connection. This description appears when you point your mouse over the specific Dynamic Data Connection in the Repository Explorer or when you have to choose a Dynamic Data Connection. This information also appears when you right-click a Dynamic Data Connection in the Repository Explorer and select Properties.
- *Author*
By default, this field contains the name of the user who created the Dynamic Data Connection. The name of the author appears when you point your mouse over the specific Dynamic Data Connection in the Repository Explorer or when you have to choose a Dynamic Data Connection.
- *Parent Folder*
The repository folder in which the Dynamic Data Connection is located. This is the only property that cannot be directly modified through the Property Browser.

- **Prompt Text**
This is the prompting text that appears when a user reports off of a Business View that references the Dynamic Data Connection.
- **Rights**
The user and group rights that are associated with the Dynamic Data Connection. To update the rights, ensure that the Rights cell is selected and then click the ... button. For more information on editing rights, see [Editing rights for a Dynamic Data Connection \[page 94\]](#).

You can also view some of the Dynamic Data Connection's properties by right-clicking a Dynamic Data Connection in the Repository Explorer and selecting Properties. The Properties dialog box displays the name of the object, the type of object, and the date the object was last saved. This dialog box also displays the object's description.

6.2.4.4 Using the Object Explorer

The Object Explorer displays the name of the Dynamic Data Connection. By right-clicking the Dynamic Data Connection, you can edit user and group rights for a Dynamic Data Connection. For more information on editing rights, see [Editing rights for a Dynamic Data Connection \[page 94\]](#).

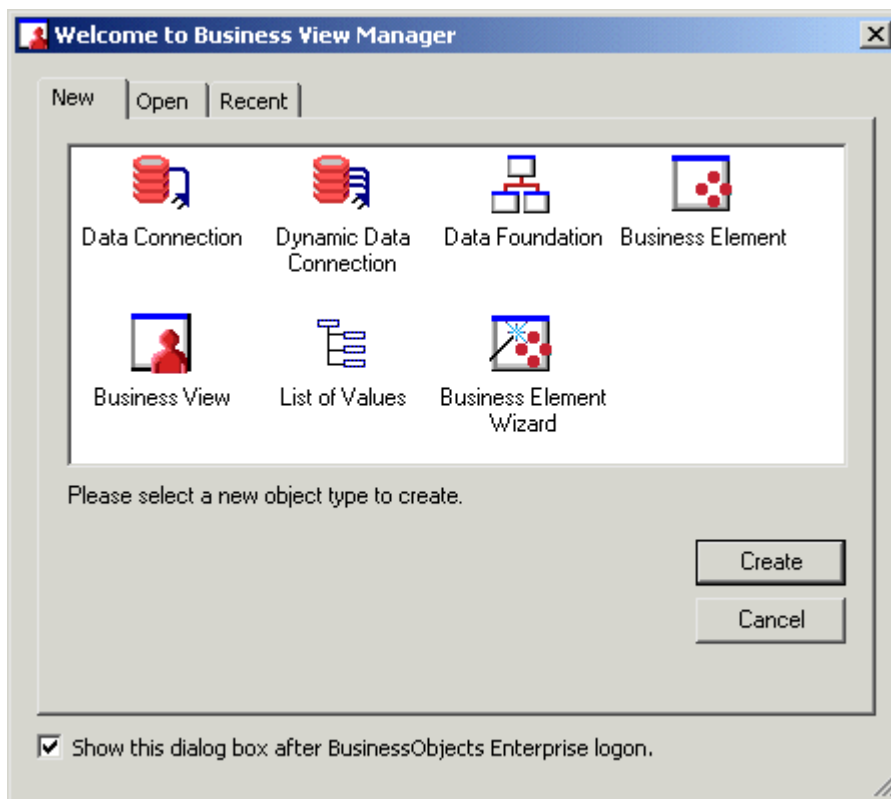
6.2.4.5 Enabling the Dynamic Data Connection to work in SAP Crystal Reports


If you want to work with reports that are based on Business Views that use a Dynamic Data Connection in SAP Crystal Reports, you need to modify the Use Owner and Use Catalog properties of all the Data Connections that are referenced by the Dynamic Data Connection. Otherwise, Crystal Reports will not be able to connect to the various data sources (Data Connections) that are specified in the Dynamic Data Connection.

6.2.4.5.1 To modify the Use Owner and Use Catalog properties

1. On the [File](#) menu, click [Open](#).

The Welcome to Business View Manager dialog box appears.



2. Click the [Open](#) tab and select a Data Connection that is referenced by the Dynamic Data Connection.
3. Click [Open](#).
4. In the Property Browser, set the value of the following properties to False:
 - Use Owner
 - Use Catalog
5.  Click [Save](#).
6. Repeat Steps 1-5 for all the Data Connections that are referenced by the Dynamic Data Connection.

If you do not set the values of the Use Owner and Use Catalog properties to False, then the Dynamic Data Connection that is used by the Business View essentially becomes a regular Data Connection in SAP Crystal Reports: when you refresh the report, SAP Crystal Reports ignores the parameter for selecting a different Data Connection and continues to report off the original data source (Data Connection) that is specified in the Dynamic Data Connection.

6.2.5 To save a Dynamic Data Connection

Dynamic Data Connections, like other Business Views objects, are saved to the repository. For more information on the repository, see [Using the Repository Explorer \[page 62\]](#).

When you first save a Dynamic Data Connection, you need to specify a name for the object and the location you want to save the object to.

1. On the [File](#) menu, select [Save](#).

→ Tip



You can also click the Save (Ctrl+S) button or press CTRL+S.

The Save As dialog box appears.

2. In the [Object Name](#) field, specify the name of your object.
3. Select the folder you want to save your object to.

→ Tip



You can create new folders in the repository by clicking the "Insert a new folder" button.

4. Click [Save](#).

6.2.6 Editing rights for a Dynamic Data Connection

User and group rights settings for a Dynamic Data Connection ensure that only specific users and groups can access the Dynamic Data Connection. Since the Business Views security model is based on the SAP BusinessObjects security model, you have a flexible security model that you can work from. For example, if you explicitly deny a View right for a user or group for a specific Dynamic Data Connection, this user or group, when creating a Data Foundation, will not be able to view or select from any Dynamic Data Connection that they do not have View rights to.

ⓘ Note

Before you can edit the security rights for a Dynamic Data Connection, you must first save the Dynamic Data Connection to the repository.

Business Views object rights are also dependent on folder rights. If you set a right on a folder, all of the Business Views objects in this folder inherit the same security rights. For more information on the inheritance model as it pertains to rights, see [Using inheritance to your advantage \[page 192\]](#).

Before you can apply security rights to an object, you must have Set Security rights to the object. By default, members of the Administrators group have full access to all of the folders and objects in the repository.

ⓘ Note

For detailed information on the Edit Rights dialog box, see [Using the Edit Rights dialog box \[page 188\]](#).

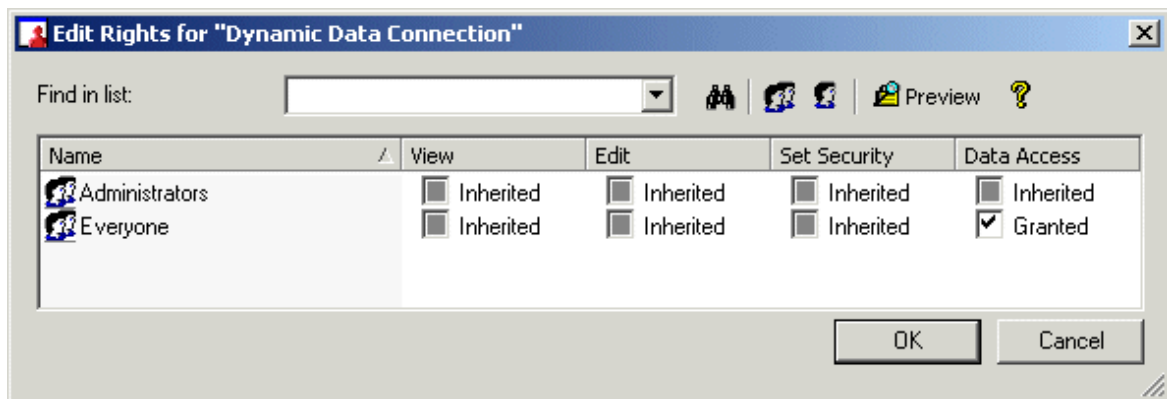
6.2.6.1 To apply security settings to a Dynamic Data Connection

1. In the Object Explorer, right-click the Dynamic Data Connection and select [Edit Rights](#).

→ Tip

You can also select the Rights cell in the Property Browser and click the ... button. Alternatively, you can select Edit Rights on the Edit menu.

The Edit Rights dialog box appears.



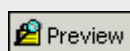
2. Set the following rights for either a user or a group by clicking the appropriate check box under each rights column.

- View
This right specifies whether a user or group can or cannot view the Dynamic Data Connection.
- Edit
This right specifies whether a user or group can or cannot edit the properties of the Dynamic Data Connection.
- Set Security
This right specifies whether a user or group can or cannot modify the rights that are associated with the Dynamic Data Connection.
- Data Access
This right specifies whether a user or group can access the specified data source in the Dynamic Data Connection. Note that this right cannot be inherited from other rights: a user or group either explicitly has data access or does not.


ⓘ Note

A cleared check box indicates that the user or group is denied the right; a check mark in the check box indicates that the user or group is granted the right; a shaded check box indicates that the right is inherited. Also, note that a right may be shown as being granted, but may be denied once inheritance is taken into account.

ⓘ Note



Click the Preview button to view the final security settings for the object with inheritance in effect. The Business View Manager resolves the net result of the current security settings and displays the net rights.

3.  To add another user or group, click the [Add Users](#) button or click the [Add Groups](#) button.
4. Assign rights to the new user or group as needed.
5. Click [OK](#).

7 Managing Data Foundations

This section details how to manage Data Foundations. It provides conceptual and procedural information on Data Foundations and their collections of tables and columns.

7.1 Data Foundations overview

A Data Foundation is a collection of tables and fields; it provides associated connections through default joins and aliases for tables that are from different Data Connections (or Dynamic Data Connections). The primary use of Data Foundations is for data abstraction; that is, administrators can control, from different Data Connections, which tables and fields users can or cannot access.

7.2 Working with Data Foundations

When you create a Data Foundation, you need to specify your data source, whether it be one or more Data Connections or one or more Dynamic Data Connections. You can have any number or combination of Data Connections and Dynamic Data Connections for your Data Foundation.

Data Foundations are flexible as they contain the physical database properties for tables and fields. Using Data Foundations, you can link different data tables. You can also insert additional data tables, and create formulas, SQL expressions, filters, parameters, and custom functions. In addition, you can set specific rights on all of these objects for your users and groups.

When you insert a new formula, SQL expression, filter, or parameter, a new tab appears at the bottom of the main Data Foundation window. Click between the different tabs to switch from the different objects that you inserted. When you insert a data table, the table is displayed on the Linking Diagram tab.

Note

When a Business View designer creates a Data Foundation that is based on a Dynamic Data Connection, this user is prompted to specify which Data Connection to use. (The Dynamic Data Connection specifies the list of Data Connections that the user is able to choose from.)

7.2.1 Creating a new Data Foundation

When you create a new Data Foundation, you first need to specify the tables you want to use (tables which are taken from the data source specified by a Data Connection or a Dynamic Data Connection). After you create the Data Foundation, you can link different tables. You can insert additional tables, formulas, SQL expressions, filters, parameters, and custom functions to your Data Foundation.

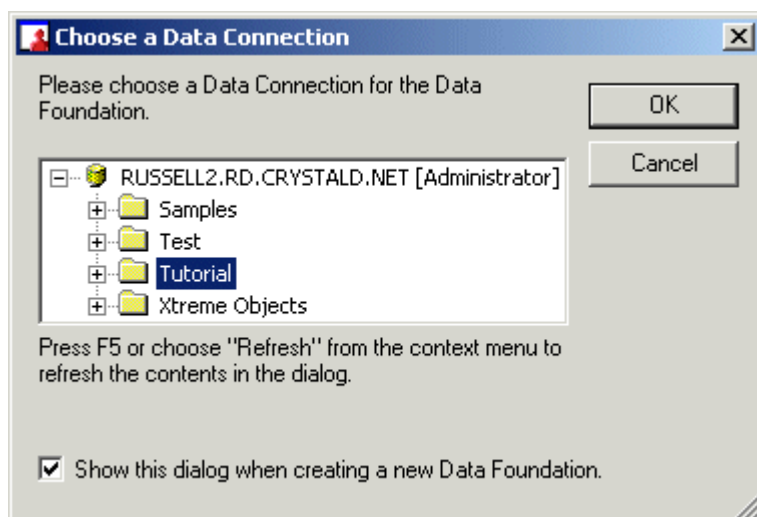
This section details how to create a new Data Foundation and specify your Data Connection(s) or Dynamic Data Connection(s). For more information on inserting formulas, filters, and so on, see the related topics below.

- [Linking tables \[page 100\]](#)
- [Inserting data tables \[page 110\]](#)
- [Inserting a formula \[page 115\]](#)
- [Inserting an SQL expression \[page 124\]](#)
- [Inserting a filter \[page 132\]](#)
- [Inserting a parameter \[page 127\]](#)
- [Importing a custom function \[page 136\]](#)

7.2.1.1 To create a Data Foundation

1. On the *File* menu, select *New*, and then click *Data Foundation*.

The "Choose a Data Connection" dialog box appears.

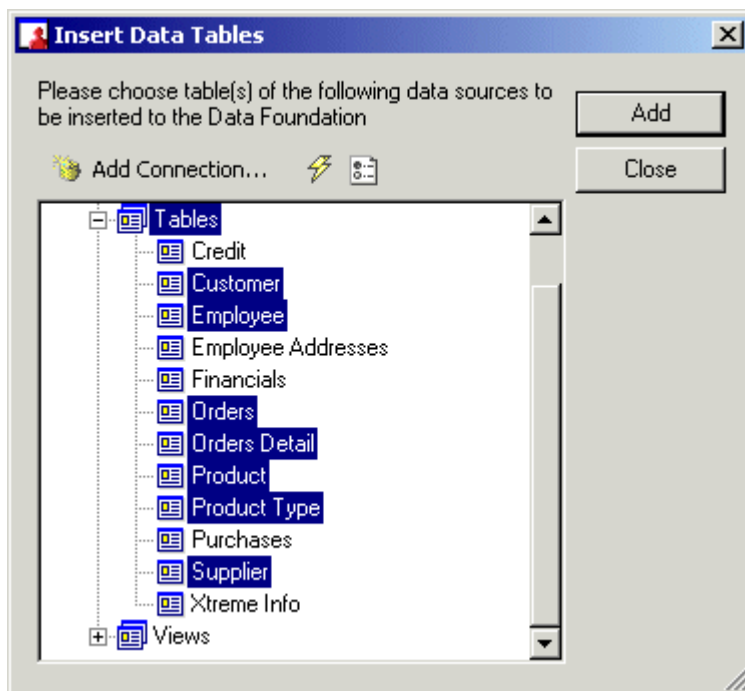


2. Select a Data Connection or a Dynamic Data Connection and click *OK*.

Note

If you choose a Dynamic Data Connection, you are prompted to choose from one of the Data Connections that the Dynamic Data Connection points to. Select the appropriate Data Connection from the Value list and click OK.

The "Insert Data Tables" dialog box appears.



3. If you want to add another Data Connection or Dynamic Data Connection, click [Add Connection](#).

Continue adding additional Data Connections and Dynamic Data Connections as required.

→ Tip



Click the Refresh button to refresh the data source information; click the Edit Options button to set how tables and fields are displayed in the "Insert Data Tables" dialog box. For more information on this dialog box, see [Setting the Database Explorer Options \[page 113\]](#).

4. Expand the various table nodes, select a table, and click [Add](#).
5. Continue adding tables as needed.
6. Click [Close](#).

7.2.2 Verifying object dependency and verifying databases



You can verify the dependency of other objects that are affected by the settings you specify in the Data Foundation. To test the Data Foundation and the Business Views objects that depend on the settings specified in the Data Foundation, select Check Dependent Integrity on the Tools menu or click the Check Dependent Integrity button located on the toolbar.



Changes you make to the Data Foundation can affect Business Elements and Business Views. To verify your data source(s), on the Tools menu, select Verify Database. This tool checks to see whether any data fields have changed. If any data fields have changed, you have to confirm the field mapping through a Map Fields dialog box. For more information on the Map Fields dialog box, see the "Map Fields dialog box" in the SAP Crystal Reports Online Help.

7.2.3 Showing dependent and referenced objects

You can view the Business Views objects that depend on your Data Foundation. Similarly, you can view the Business Views objects that are referenced by your Data Foundation.

To view dependent or referenced objects, in the Tools menu, select either Show Dependent Objects or Show Referenced Objects. A dialog box appears; in this dialog box, click the Save To File button to save the list of dependent or referenced objects to a text file for future reference.

The list of dependent objects displays objects that are affected by the Data Foundation (for example, Business Elements and Business Views). The list of referenced objects displays objects that the Data Foundation references (Data Connections and/or Dynamic Data Connections). Expand the nodes to see all of the objects that depend on or are referenced by your Data Foundation.

7.2.4 Modifying a Data Foundation

In a Data Foundation, you can update and modify your data source, whether it be one or more Data Connections or one or more Dynamic Data Connections.

You can also add and update additional data tables, formulas, SQL expressions, filters, parameters, and custom functions. In addition, you can set specific rights on these objects for your users and groups.

See the following topics for more information:

- [Linking tables \[page 100\]](#)
- [Inserting data tables \[page 110\]](#)
- [Setting the Database Explorer Options \[page 113\]](#)
- [Inserting a formula \[page 115\]](#)
- [Inserting an SQL expression \[page 124\]](#)
- [Inserting a filter \[page 132\]](#)
- [Inserting a parameter \[page 127\]](#)
- [Importing a custom function \[page 136\]](#)
- [Using the Referenced Data Connections window \[page 139\]](#)
- [Using the Property Browser \[page 139\]](#)
- [Using the Object Explorer \[page 140\]](#)
- [To save a Data Foundation \[page 141\]](#)
- [Editing rights for a Data Foundation \[page 141\]](#)

7.2.5 Linking tables

In your Data Foundation, you link tables so that records from one table will match related records from another. For example, if you add an Orders table and a Customers table, you link the two tables so that each order (from the Orders table) can be matched up with the customer (from the Customer table) who made the order.

When you link tables, you are using a field that is common to both tables. Business Views uses the link to match up records from one table with those from the other. Note that in your Data Foundation, you can have

different tables from different data sources. You can then use the Business View Manager to link these tables, even though they are not from the same data source.

For more information on linking tables, see the "Linking tables" section, located in the "Understanding Databases" section of the *Crystal Reports Online Help*.

7.2.5.1 Link from and link to

When you link two tables, you link from one table to another table. The "from" table is used as a primary table, while the "to" table acts as a lookup table, where records are looked up by the primary table. In a simple link, Business Views examines the first record in the primary table and finds all matching records in the lookup table. Once all matches have been found in the lookup table for the first record in the primary table, all matches in the lookup table for the next record in the primary table are found.

7.2.5.2 Join types

The join type expresses the nature of the relationship between two tables.

The join types are:

- Inner join
- Left Outer join
- Right Outer join
- Full Outer join

Inner join

An Inner join is the standard type of join. When a report is generated, the result set from an Inner join includes all the records in which the linked field value in both tables is an exact match.

Left Outer join

When a report is generated, the result set from a Left Outer join includes all the records in which the linked field value in both tables is an exact match. It also includes a row for every record in the primary (left) table for which the linked field value has no match in the lookup table.

Right Outer join

When a report is generated, the result set from a Right Outer join includes all the records in which the linked field value in both tables is an exact match. It also includes a row for every record in the lookup (right) table for which the linked field value has no match in the primary table.

Full Outer join

A Full Outer join is a bidirectional outer join where you can see all records in your linked tables. When a report is generated, the result set from a Full Outer join includes all the records in which the linked field value in both tables is an exact match. It also includes a row for every record in the primary (left) table for which the linked field value has no match in the lookup table, and a row for every record in the lookup (right) table for which the linked field value has no match in the primary table.

7.2.5.3 Enforced joins

Using the various join enforcement options can ensure that linked tables are included in a query when a report is generated in Crystal Reports, even when none of the fields in the table are used in the report.

The enforce join options are:

- Not Enforced
- Enforced From
- Enforced To
- Enforced Both

Not Enforced

When you select this option, the link you've created is used only if it's explicitly required. Your users can create reports based on the selected tables without restriction (that is, without enforcement based on other tables). This is the default option.

Enforced From

When you select this option, if the "to" table for the link is used, the link is enforced. For example, if you create a link from TableA to TableB using Enforce From and select only a field from TableB, the report will still include the join to TableA because it is enforced. Conversely, selecting only from TableA with the same join condition will not cause the join to TableB to be enforced.

Note

For an explanation of from and to tables, see [Link from and link to \[page 101\]](#).

Enforced To

When you select this option, if the "from" table for the link is used, the link is enforced. For example, if you create a link from TableA to TableB using Enforce To and select only a field from TableA, the join to TableB will be enforced, and the report that is generated will include the join to both tables.

Note

For an explanation of from and to tables, see [Link from and link to \[page 101\]](#).

Enforced Both

When you select this option, if either the "from" table or the "to" table for this link is used, the link is enforced.

7.2.5.4 Link relationships

When you link records from one table to another table, the records will typically fall under one of two relationship types:

- One-to-one
- One-to-many

One-to-one relationships

In a one-to-one relationship between records in two linked tables, for every record in the primary table, there is only one matching record in the lookup table (based on the linked fields). For example, in the Xtreme.mdb database, the Employee table can be linked to the Employee Addresses table based on the Employee ID field in each table. The Employee table contains information about employees at the company, the positions they hold, their salaries, hiring information, and so on. The Employee Addresses table contains each employee's home address. There is only one record for each employee in each of these tables. Thus, if the Employee table is linked to the Employee Addresses table, only one record will be found in the Employee Addresses table for each record in the Employee table—this is a one-to-one relationship.

One-to-many relationships

In a one-to-many relationship between records in two linked tables, for every record in the primary table, there may be more than one matching record in the lookup table, based on the linked fields. In the Xtreme.mdb database, the Customer table can be linked to the Orders table based on the Customer ID field in each table. The Customer table contains information about each customer that has placed an order with the company. The Orders table contains information about orders that customers have placed. Since customers can place more than one order, there may be more than one record in the Orders table for each customer record in the Customers table—this is a one-to-many relationship.

7.2.5.5 Link types

The link type expresses the nature of the relationship between two fields.

The link types are:

- Equal [=] link
- Greater Than [>] link
- Greater Than Or Equal [>=] link
- Less Than [<] link
- Less Than Or Equal [<=] link
- Not Equal [!=] link

Equal [=] link

The result set from an Equal link includes all the records where the linked field value in both tables is an exact match.

Greater Than [>] link

The result set from a Greater Than link includes all records in which the linked field value from the primary table is greater than the linked field value in the lookup table.

Greater Than Or Equal [>=] link

The result set from a Greater Than Or Equal link includes all records in which the linked field value in the primary table is greater than or equal to the linked field value in the lookup table.

Less Than [<] link

The result set from a Less Than link includes all records in which the linked field value in the primary table is less than the linked field value in the lookup table.

Less Than Or Equal [<=] link

The result set from a Less Than Or Equal link includes all records in which the linked field value in the primary table is less than or equal to the linked field value in the lookup table.

Not Equal [!=] link

The result set from a Not Equal link includes all records in which the linked field value in the primary table is not equal to the linked field value in the lookup table

7.2.5.6 To create a link

Select a field in a data table by clicking it, and then drag the mouse pointer to the field you want to link to.

Note

You will not be able to link two fields with incompatible data types.

Once you create your links between different tables, you can modify a link by right-clicking it. For more information on modifying links, see [Modifying a link \[page 105\]](#).

7.2.5.7 Modifying a link

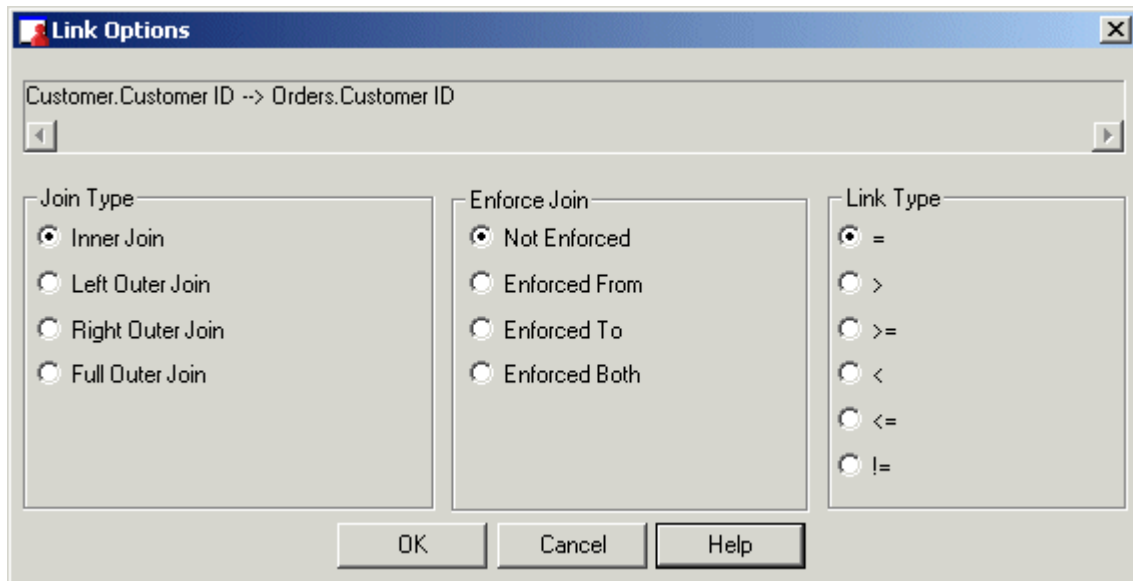
After you create links between different tables, you can modify a link by right clicking on it.

Note

To ensure that you are modifying a specific link, first click the link to select it. The selected link will be displayed as a blue link. Then, right-click the selected link.

7.2.5.8 Link Options

The Link Options dialog box allows you to specify join and link types.



- [Link display](#)
This box displays the selected link. It also shows the "from" table (used as a primary table) and the "to" table (which acts as a lookup table where records are looked up by the primary table). For more information on links, see [Link from and link to \[page 101\]](#).
- [Join Type](#)
Business Views enables you to specify the type of join you want it to use when joining tables. The join type expresses the nature of the relationship between two tables. For more information on the specific types of joins available, see [Join types \[page 101\]](#).
- [Enforce Join](#)
Business Views enables you to enforce the use of tables while you're specifying joins. For more information on the specific types of enforced joins available, see [Enforced joins \[page 102\]](#).
- [Link Type](#)
Business Views enables you to specify the type of link you want it to use when linking fields between tables. The link type expresses the nature of the relationship between two fields. For more information on the specific types of links available, see [Link types \[page 104\]](#).

7.2.5.9 Deleting a link

Click a link to select it, then right-click the link and select [Delete Link](#) to remove it.

7.2.5.10 Reverse a link

Click a link to select it, then right-click the link and select [Reverse Link](#) to reverse the "from" and "to" tables.

For more information on the "from" and "to" tables, see [Link from and link to \[page 101\]](#).

7.2.5.11 Remove All Links

Click a link to select it, then right-click the link and select [Remove All Links](#) to remove all of the links that are set in the "from" table.

7.2.5.12 To select Smart Linking By Key

When you select Smart Linking By Key, you are automatically linking tables by their foreign key information. Business Views scans through the data tables and links the fields with foreign keys that match.

On the Linking Diagram menu, select [Smart Linking By Key](#). Alternatively, on the Linking Diagram tab in the main Data Foundation window, right-click anything but an existing link and select Smart Linking By Key.

7.2.5.13 To select Smart Linking By Name

When you select Smart Linking By Name, you are automatically linking tables by their names. Business Views scans through the data tables and links the fields with names that match.

On the Linking Diagram menu, select [Smart Linking By Name](#). Alternatively, on the Linking Diagram tab in the main Data Foundation window, right-click anything but an existing link and select Smart Linking By Name.

7.2.5.14 To select Clear Links

When you select Clear Links, all existing links will be cleared. The Visual Linking dialog box asks you to confirm the deletion of the existing links.

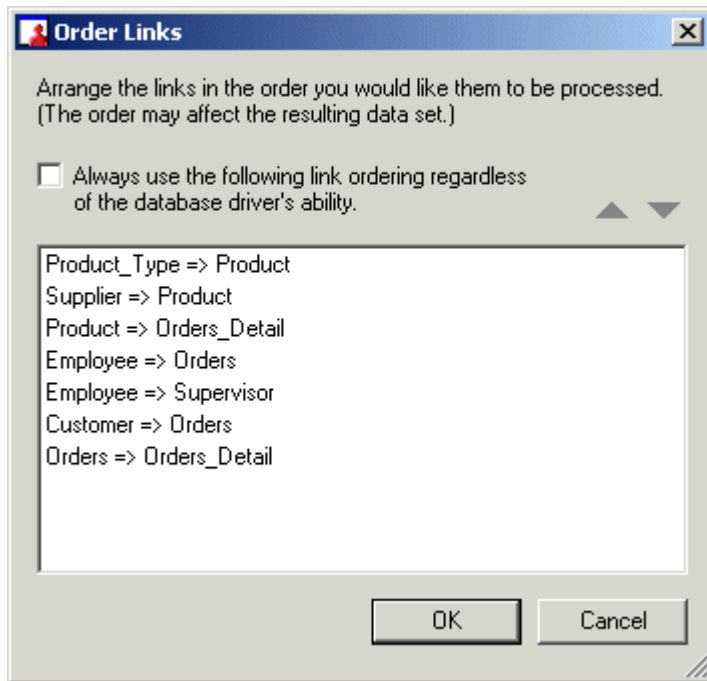
On the Linking Diagram menu, select [Clear Links](#). Alternatively, on the Linking Diagram tab in the main Data Foundation window, right-click anything but an existing link and select Clear Links.

7.2.5.15 To select Order Links

Using Order Links, you can establish the link processing order you want for the available linked tables. You must have more than two tables available for linking before you can order the links.

1. On the Linking Diagram menu, select [Order Links](#). Alternatively, on the Linking Diagram tab in the main Data Foundation window, right-click anything but an existing link and select Order Links.

The Order Links dialog box appears.



2. Use the arrow buttons in the Links Order dialog box to change the default link order.
3. Select the "Always use the following link ordering regardless of the database driver's ability" check box to enforce link ordering.

7.2.5.16 To remove a table

When you select Remove Table, the currently selected table is removed. Note that if a table is linked to other tables, you must clear these links first, before you can remove the table.

Right-click the table's title bar and select [Remove Table](#).

7.2.5.17 To select Locate Table

When you select Locate Table, a Locate Table dialog box appears, listing all of the tables displayed on the Linking Diagram tab of the main Data Foundation window. This function is useful when you have a number of tables in the Linking Diagram Tab and need to quickly locate a specific table. Select a table from the list and click the Done button. The Linking Diagram tab displays the selected table.



On the Linking Diagram menu, select [Locate Table](#). Alternatively, on the Linking Diagram tab in the main Data Foundation window, right-click anything but an existing link and select Locate Table. You can also click the Locate Table button on the toolbar.

7.2.5.18 To select Rearrange Tables

When you select Rearrange Tables, the data tables are rearranged by existing links.



On the Linking Diagram menu, select [Rearrange Tables](#). Alternatively, on the Linking Diagram tab in the main Data Foundation window, right-click anything but an existing link and select Rearrange Tables. You can also click the Rearrange Tables button on the toolbar.

7.2.5.19 To specify which tables are visible

When you click Select Visible Tables, the "Choose Visible Tables for Linking Diagram" dialog box appears. Using this dialog box, you can highlight which tables you want to be visible. The tables that are not highlighted will not appear on the Linking Diagram tab of the main Data Foundation window. You can click the Select All button to select all of the tables, or the Clear button to clear any tables that are highlighted.



On the Linking Diagram menu, click [Select Visible Tables](#). (You can also click the Select Visible Tables button on the toolbar.)

7.2.5.20 Fetch Table Indexes

When you select Fetch Table Indexes, the Business View Manager retrieves the indexes for the tables. See [To select Index Legend \[page 109\]](#) to view the key to the index indicators.

7.2.5.21 To select Index Legend

When you select Index Legend, the Index Legend dialog box appears. This dialog box displays a key to the index indicators that are used in the visible tables on the Linking Diagram tab area.

On the Linking Diagram menu, select [Index Legend](#). Alternatively, on the Linking Diagram tab in the main Data Foundation window, right-click anything but an existing link and select Index Legend.

7.2.5.22 To select Change Linking View

When you select Change Linking View, the view changes to display only the names of the tables.



On the Linking Diagram menu, select [Change Linking View](#). Alternatively, on the Linking Diagram tab in the main Data Foundation window, right-click anything but an existing link and select Change Linking View. You can also click the Change Linking View button on the toolbar.

7.2.6 Inserting data tables

When you first create a Data Foundation, you select which tables you want to include. After you have created your Data Foundation, you can continue adding and removing tables as needed. This section provides steps on inserting additional data tables. For more information on removing existing tables, see [To remove a table \[page 108\]](#).

After you insert a data table, you can browse the fields in the data table by right-clicking any of the fields in the table and selecting Browse Field. The Browse dialog box appears and lists the field type (and the field length, where appropriate) and displays all of the entries in the field.

7.2.6.1 To insert a data table

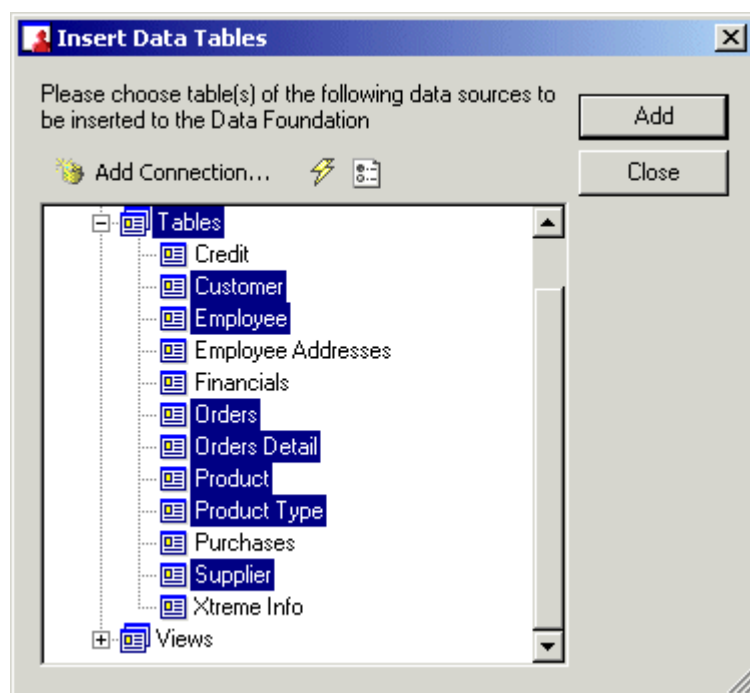
1. In the Object Explorer, right-click a table and select *Insert Data Tables*.

→ Tip



You can also click the Insert Data Tables button on the toolbar, or on the Insert menu, select Insert Data Tables.

The "Insert Data Tables" dialog box appears.



2. Expand the data connections, select a table and click *Add*.

→ Tip

If you want to add more Data Connections or Dynamic Data Connections from which to get tables, click Add Connection.

→ Tip



Click the Refresh button to refresh the dialog box; click the Edit Options button to set how tables and fields are displayed in the "Insert Data Tables" dialog box. For more information on this dialog box, see [Setting the Database Explorer Options \[page 113\]](#).

3. Continue adding tables as needed.
4. Click [Close](#).

7.2.6.2 Setting table location for data tables

Setting the table location allows you to do two things:

- Change the Data Connection that data tables, stored procedures, and command tables depend on.
- Redirect a data table to point to another table.

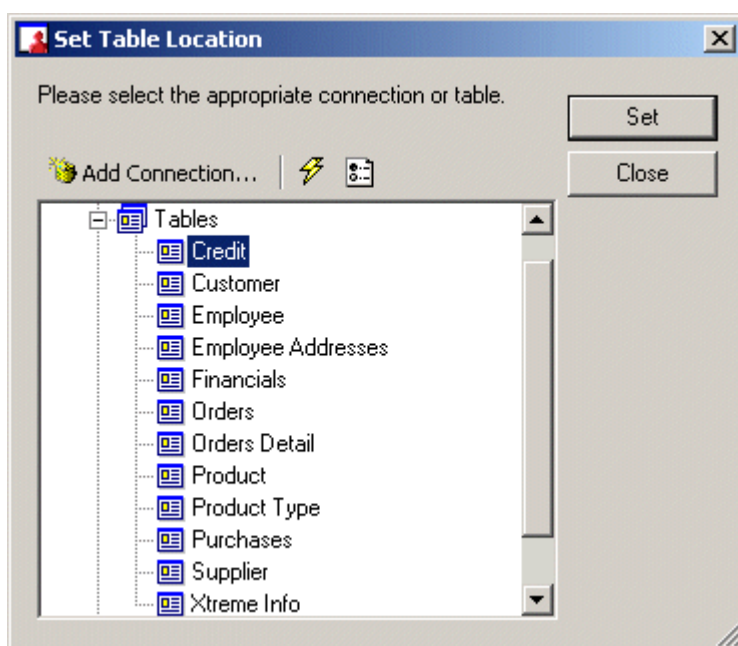


In the Object Explorer, select a table and click the "Set Table Location" button on the toolbar.

→ Tip

You can also select multiple tables using the [CTRL](#) key.

In the dialog box that appears, the Set button is enabled only when a Data Connection or table object is selected.



If a Data Connection is chosen, the designer checks (using the physical table name) to see if the table's counterpart exists in the connection's data source. If the table does not exist, an error message is returned. Otherwise, the table's connection is updated.

If a new table is selected instead, the original table is updated to point to this new table. Note that if the new table is based on a different Data Connection, the designer performs a check to see if the table exists in the connection's data source (using the physical table name). If the table does not exist, an error message is returned. Otherwise, the table and its connection are updated.

Note

If multiple tables are selected in the main Data Foundation window and set to point to a new table, the Business View Manager only updates the Data Connections of the multiple tables.

Note

Users cannot use Set Table Location to change the type of the original table. For example, a data table cannot be made to point to a stored procedure, nor can a command table be made to point to a physical database table. You can only make a table to point to another table of the same type.

7.2.6.3 Editing rights for a field

By editing the user and group rights for a field in a data table, you ensure that only specific users and groups can access specific fields in the Data Foundation.

Note

Before you can edit the security rights for a field, you must first save the Data Foundation to the SAP BusinessObjects repository.

Business Views object rights are dependent on folder rights. If you set a right on a folder, all of the objects in this folder inherit the same security rights. Similarly, all of the objects within the Data Foundation also inherit security rights from the rights set at the Data Foundation level. Thus, if a user does not have Set Security rights for a Data Foundation, he or she cannot edit the rights for a field. For more information on the inheritance model as it pertains to security, see [Using inheritance to your advantage \[page 192\]](#).

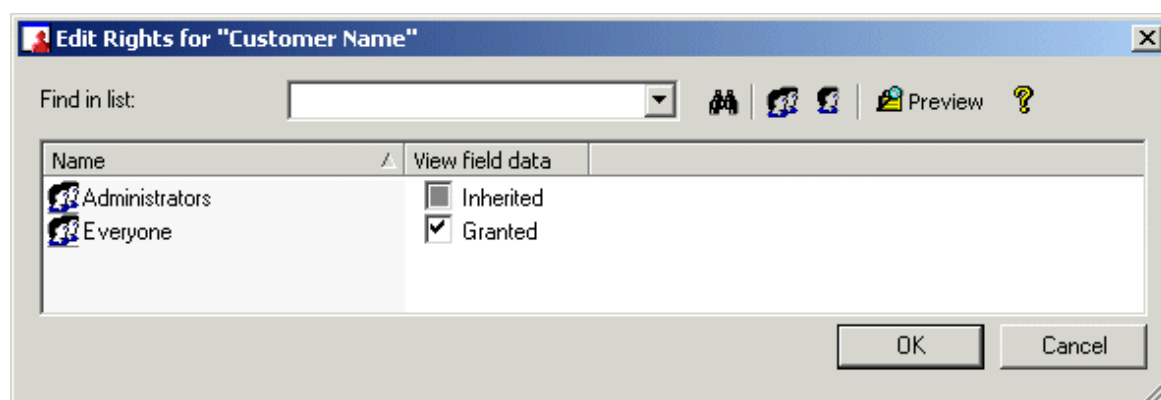
7.2.6.3.1 To apply security settings to a field

1. In the Object Explorer, select the field you want to apply security settings to and right-click the field and select [Edit Rights](#).

Tip

You can also select the Rights cell in the Property Browser and click the ... button. Alternatively, you can select Edit Rights on the Edit menu.

The Edit Rights dialog box appears.

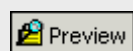


2. Set the right for either a user or a group by clicking the appropriate check box in the following column:
 - View field data
This right specifies whether a user or group can or cannot view this field and its associated data. You can explicitly grant or deny this right.



Note

A cleared check box indicates that the user or group is denied the right; a check mark in the check box indicates that the user or group is granted the right; a shaded check box indicates that the right is inherited. Also, note that a right may be shown as being granted, but may be denied once inheritance is taken into account.


Note

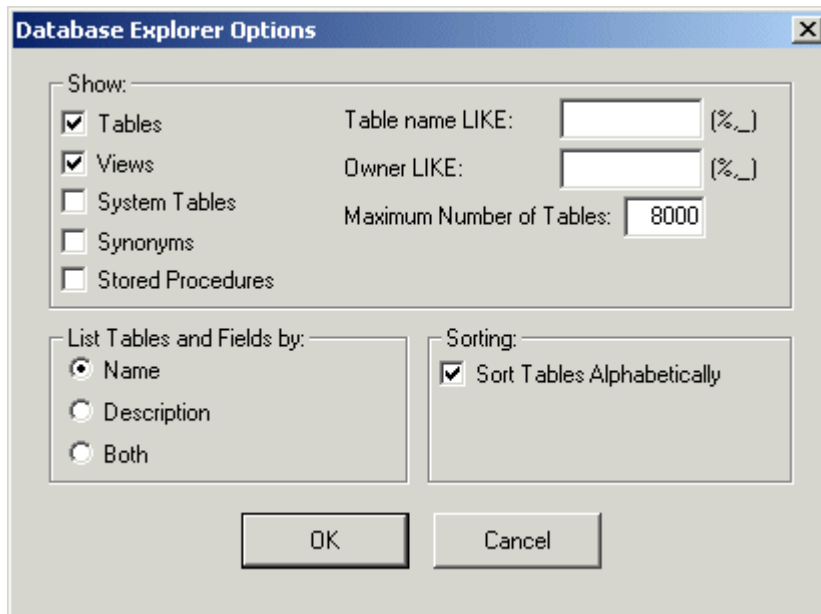


Click the Preview button to view the final security settings for the field with inheritance in effect. The Business View Manager resolves the net result of the current security settings and displays the net rights.

3.   To add another user or group, click the [Add Users](#) button or click the [Add Groups](#) button.
4. Assign rights to the new user or group as needed.
5. Click [OK](#).

7.2.7 Setting the Database Explorer Options

-  In the Insert Data Tables dialog box, click the [Edit Options](#) button to set what database options you want displayed, along with list and sort options.
The Database Explorer Options dialog box appears.



7.2.7.1 Show

- **Tables**
When you select this check box, your Data Foundation can report on database tables in your SQL/ODBC data sources. (This option is selected by default.)
- **Views**
When you select this check box, your Data Foundation can report on virtual tables in your SQL/ODBC data sources. (This option is selected by default.)
- **System Tables**
When you select this check box, your Data Foundation can report on system tables. These tables are typically used by the system administrator only but are available for use if other users have the appropriate permissions. (This option is not selected by default.)
- **Synonyms**
When you select this check box, your Data Foundation can report on virtual tables that are available on some hosts. (This option is not selected by default.)
- **Stored Procedures**
When you select this check box, your Data Foundation can report on the result sets from stored procedures if you are using SQL systems that support stored procedures. (This option is selected by default.)
- **Table name LIKE**
This field allows you to enter the SQL LIKE function to specify the kinds of table names you want to appear in the Insert Data Tables dialog box. You can use the underscore character (_) and the percent sign character (%) as wildcards with this function. The underscore character specifies any single character, while the percent sign signifies any character string. For example, DAV_ matches DAVE only, whereas DAV% matches both DAVE and DAVID. A C% in the Table name LIKE field would display only those tables that have a table name beginning with the letter C.
- **Owner LIKE**

This field works exactly like the Table name LIKE field except that the LIKE function here is used to select the Owner (or Creator or Alias) of the table, not the table name itself. For example, a C% in the Owner LIKE field would display only those tables that have an owner beginning with the letter C.

- *Maximum Number of Tables*

You specify the maximum number of tables you want displayed in the Insert Data Tables dialog box. By default, this number is set to 8000.

7.2.7.2 List Tables and Fields by

By setting the options in this section of the Database Explorer Options dialog box, you can specify the text that you want the program to use for tables and fields.

- *Name*

If you select this option, the program identifies tables and fields using the actual name (for example, "Customer" table and "Customer Name" field). This option is selected by default.

- *Description*

If you select this option, the program identifies tables and fields using the description you have specified (for example, "Our Clients" table and "The names of all our customers" field).

- *Both*

If you select this option, the program identifies tables and fields using both the name and the description you have assigned (for example, "Customer - Our Clients" table and "Customer Name - The names of all our customers" field).

7.2.7.3 Sorting

- *Sort Tables Alphabetically*

When you select this check box, tables are displayed throughout the program sorted in alphabetical order, instead of the order they appear in the database. This check box is selected by default.

7.2.8 Inserting a formula

In many cases, the data needed for a Data Foundation object already exists in database table fields. For example, to prepare an order list you would select and place the appropriate fields in a Business Element (a Business Element is a collection of fields that are based on a Data Foundation).

Sometimes, however, you need to put data in a Business Element that does not exist in any of the data fields. In this situation, you need to create a formula. For example, to calculate the number of days it takes to process each order for a customer, you need a formula that determines the number of days between the order date and the ship date.

To create formulas in Business View Manager, use the Formula Editor. You use the same formula components and syntax as when creating a formula in Crystal Reports.

7.2.8.1 Typical uses for formulas

There are many uses for formulas. If you have a need for specialized data manipulation, you can do it with a formula.

Creating calculated fields to add to your report

To calculate a price discounted 15%:

Crystal syntax example:

```
{Orders_Detail.Unit Price}*.85
```

Basic syntax example:

```
formula = {Orders_Detail.Unit Price}*.85
```

Formatting text on a report

To change all the values in the Customer Name field to uppercase:

Crystal syntax example:

```
UpperCase ({Customer.Customer Name})
```

Basic syntax example:

```
formula = UCase ({Customer.Customer Name})
```

Pulling out a portion, or portions, of a text string

To extract the first letter of the customer name:

Crystal syntax example:

```
{Customer.Customer Name} [1]
```

Basic syntax example:

```
formula = {Customer.Customer Name} (1)
```

Extracting parts of a date

To determine what month an order was placed:

Crystal syntax example:

```
Month ({Orders.Order Date})
```

Basic syntax example:

```
formula = Month ({Orders.Order Date})
```

Using a custom function

To convert \$500 from U.S. currency to Canadian:

Crystal syntax example:

```
cdConvertUSToCanadian (500)
```

Basic syntax example:

```
formula = cdConvertUSToCanadian (500)
```

7.2.8.2 Formula components and syntax

Formulas contain two critical parts: the components and the syntax. The components are the pieces that you add to create a formula while the syntax is the rules that you follow to organize the components.

7.2.8.2.1 Formula components

Creating a formula in Business View Manager is like creating one in any spreadsheet application. You can use any of the following components in your formula:

Fields

Example: {customer.CUSTOMER LAST NAME}, {customer.LAST YEAR'S SALES}

Numbers

Example: 1, 2, 3.1416

Text

Example: "Quantity", ":", "your text"

Operators

Example: + (add), / (divide), -x (negate)

Operators are actions you can use in your formulas.

Functions

Example: Round (x), Trim (x)

Functions perform calculations such as average, sum, and count. All functions available are listed with their arguments and are arranged by their use.

Custom functions

Example: cdFirstDayOfMonth, cdStatutoryHolidays

Custom functions provide a way to share and reuse formula logic. They can be stored in the repository and then added to a report. Once in the report, custom functions can be used in the Formula Expert when creating formulas.

Control Structures

Example: "If" and "Select", "For" loops

Group field values

Example: Average (fld, condFld), Sum (fld, condFld, "condition")

Group field values summarize a group. For example, you could use group field values to find the percentage of the grand total contributed by each group.

Other formulas

Example: {@GrossProfit}, {@QUOTA}

7.2.8.2.2 Formula syntax

Syntax rules are used to create correct formula. Some basic rules are:

- Enclose text strings in quotation marks.
- Enclose arguments in parentheses (where applicable).
- Referenced formulas are identified with a leading @ sign.

Crystal and Basic syntax

When creating formulas, you have the option of using either Crystal or Basic syntax. Almost any formula written with one syntax can be written with the other. Reports can contain formulas that use Basic syntax as well as formulas that use Crystal syntax.

Crystal syntax is the formula language included in all versions of Crystal Reports.

If you are familiar with Microsoft Visual Basic or other versions of Basic, then Basic syntax may be more familiar to you. In general, Basic syntax is modeled on Visual Basic except that it has specific extensions to handle reporting.

If you are already comfortable with Crystal syntax, you can continue to use it, and benefit from the new functions, operators and control structures inspired by Visual Basic.

Note

Record selection and group selection formulas cannot be written in Basic syntax.

Note

Report processing is not slowed down by using Basic syntax. Reports using Basic syntax formulas can run on any machine that Crystal Reports runs on.

Note

Using Basic syntax formulas does not require distributing any additional files with your reports.

For more information

- To learn about Basic syntax, see [Creating Formulas with Basic syntax](#) in the Crystal Reports online help.
- To learn about Crystal syntax, see [Creating Formulas with Crystal syntax](#) in the Crystal Reports online help.

7.2.8.3 The Formula Editor windows

The Formula Editor contains four main windows that are situated under the Formula Editor toolbar.

Window	Description of contents
Report Fields	Report fields contain all the database fields that are accessible for your Data Foundation. They also contain any formulas or groups already created for the Data Foundation.
Functions	Functions are prebuilt procedures that return values. They perform calculations such as average, sum, count, sin, trim, and uppercase. Custom functions are also listed in this window.
Operators	Operators are the "action verbs" you use in formulas. They describe an operation or an action to take place between two or more values. Examples of operators include: add, subtract, less than, and greater than.
Formula text window	This is the area where you create a formula. You can also view your formula in the Property Browser window.

7.2.8.4 Choosing the syntax

The Formula Editor toolbar contains the list where you choose either Crystal syntax or Basic syntax for the formula you are creating. By default, Crystal syntax is used.

Note

Changing the syntax from Crystal syntax to Basic syntax or vice versa will change the list of functions in the Functions window as well as the list of operators in the Operators window. The functions and operators differ from syntax to syntax. The available report fields, however, remain the same since the report fields are available to each syntax.

7.2.8.5 Entering formula components

The Report Fields, Functions, and Operators trees at the top of the Formula Editor contain the primary formula components. Double-click any component from these trees to add this component to your formula.

For example, if you set the syntax to Basic syntax and double-click the Operators > Control Structures > Do While c s Loop in the Operators tree, the following text is transferred to the Formula text window with the cursor between the Do While and Loop:

```
Do While
Loop
```

7.2.8.6 To create a formula

You must create and define your formulas at the Data Foundation level before you can add these formulas to your Business Elements.

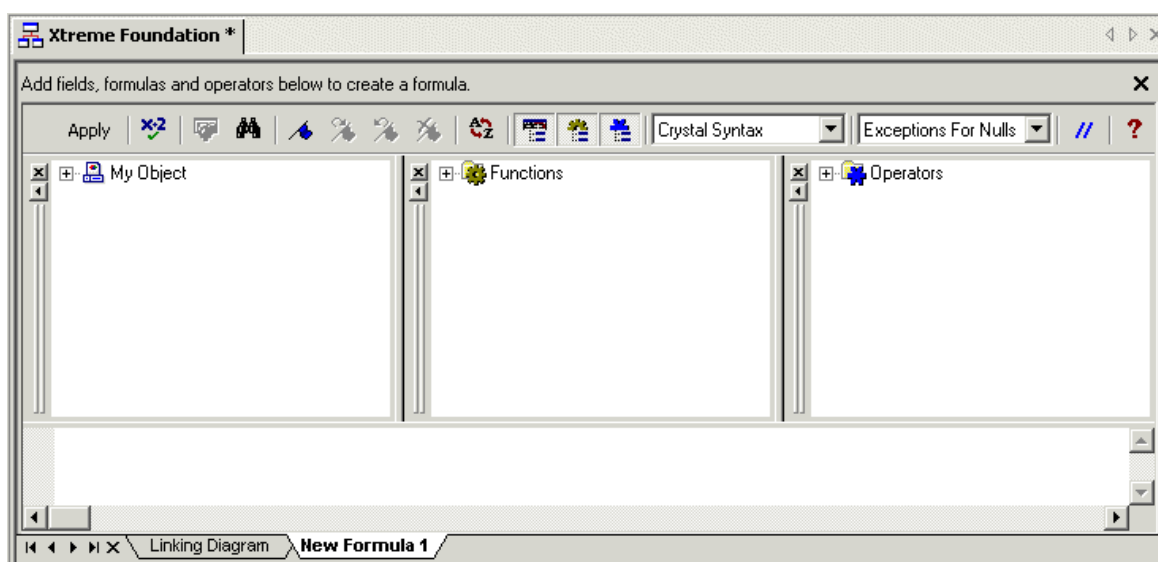
1. On the *Insert* menu, click *Insert Formula*.

Note



In the Object Explorer, you can right-click Formulas and select Insert Formula. Alternatively, you can click the Insert Formula button on the toolbar.


The Formula Editor appears.



2. On the Formula Editor Toolbar, choose either *Crystal Syntax* or *Basic Syntax*.

For more information on which syntax to use, see [Choosing the syntax \[page 120\]](#).

3. In the second drop-down list, specify one of the following values:

- Choose *Exception for Nulls* to configure the formula to ignore null values.
 - Choose *Zeroes for Nulls* to configure the formula to treat null values as zero values.
4. Enter the formula by typing in the components or by selecting them from the component trees.
 5.  Click *Check* to test if there are any errors in the formula.
 6. Fix any syntax errors the Formula Checker identifies.
 7. When the formula has the correct syntax, click *Apply* to save the formula.

When you create a Business Element, you can then select from a number of saved formulas in the Data Foundation.

7.2.8.7 Updating the formula through the Property Browser

The Property Browser displays all of the properties that you can edit or change for a formula you have selected in the Object Explorer.

- *Name*
The name of the formula.
- *Description*
A description you enter for the formula.
- *Field Type*
Displays the data field type.
- *Formula Syntax*
Displays whether the formula syntax is Crystal syntax or Basic syntax.
- *Formula Text*
Displays your formula.
- *Rights*
The user and group rights that are associated with the formula. To update the rights, ensure that the Rights cell is selected and then click the ... button. In the Edit Rights dialog box, you can specify whether users or groups can or cannot "View field data." This right is either explicitly granted or denied.

7.2.8.8 Setting rights for a formula

By editing the user and group rights for a formula in a Data Foundation, you ensure that only specific users and groups can access specific formulas in the Data Foundation. Thus, when a user creates a Business Element, he or she can view and add only formulas that he or she has access rights to.

Note

Before you can edit the security rights for a formula, you must first save the Data Foundation to the repository.

Business Views object rights are dependent on folder rights. If you set a right on a folder, all of the objects in this folder inherit the same security rights. Similarly, all of the objects within the Data Foundation also inherit security rights from the rights set at the Data Foundation level. Thus, if a user does not have Set Security rights

for a Data Foundation, he or she cannot edit the rights for a formula. For more information on the inheritance model as it pertains to security, see [Using inheritance to your advantage \[page 192\]](#).

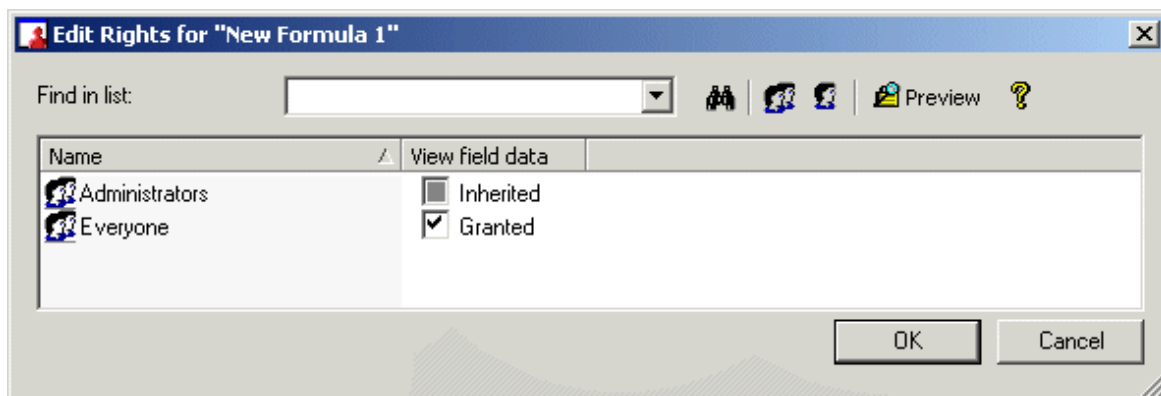
7.2.8.8.1 To apply security settings to a formula

1. In the Object Explorer, select the formula you want to apply security settings to, right-click the formula, and then select [Edit Rights](#).

→ Tip

You can also select the Rights cell in the Property Browser and click the ... button. Alternatively, you can select Edit Rights on the Edit menu.

The Edit Rights dialog box appears.

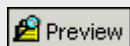


2. Set the right for either a user or a group by clicking the appropriate check box in the following column:
 - View field data
This right specifies whether a user or group can or cannot view this formula and its associated data.
You can explicitly grant or deny this right.

Note

A cleared check box indicates that the user or group is denied the right; a check mark in the check box indicates that the user or group is granted the right; a shaded check box indicates that the right is inherited. Also, note that a right may be shown as being granted, but may be denied once inheritance is taken into account.

Note



Click the Preview button to view the final security settings for the formula with inheritance in effect. The Business View Manager resolves the net result of the current security settings and displays the net rights.

3.   To add another user or group, click the [Add Users](#) button or click the [Add Groups](#) button.

4. Assign rights to the new user or group as needed.
5. Click [OK](#).

7.2.9 Inserting an SQL expression

SQL expressions are similar to formulas, but they are written in Structured Query Language (SQL). They are useful in optimizing report performance because the tasks they execute are always performed on the database server (as opposed to a regular formula, which is typically executed on the local machine).

For more information on the SQL language, see the "Understanding Databases" section of the *Crystal Reports Online Help*. For more information on formulas, see [Inserting a formula \[page 115\]](#).

7.2.9.1 The SQL expression Editor windows

The SQL expression Editor contains four main windows that are situated under the SQL expression Editor toolbar.

Window	Description of contents
Report Fields	Report fields contain all the database fields that are accessible for your Data Foundation.
Functions	Functions are prebuilt procedures that return values. They perform calculations including conversion calculations, numeric calculations, and so on.
Operators	<p>Operators are the "action verbs" you use in SQL expressions. They describe an operation or an action to take place between two or more values.</p> <p>Examples of operators include: add, subtract, less than, and greater than.</p>
SQL expression text window	This is the area where you create an SQL expression. You can also view your SQL expression in the Property Browser window.

7.2.9.2 To create an SQL expression

You must create and define your SQL expressions at the Data Foundation level before you can add these expressions to your Business Elements.

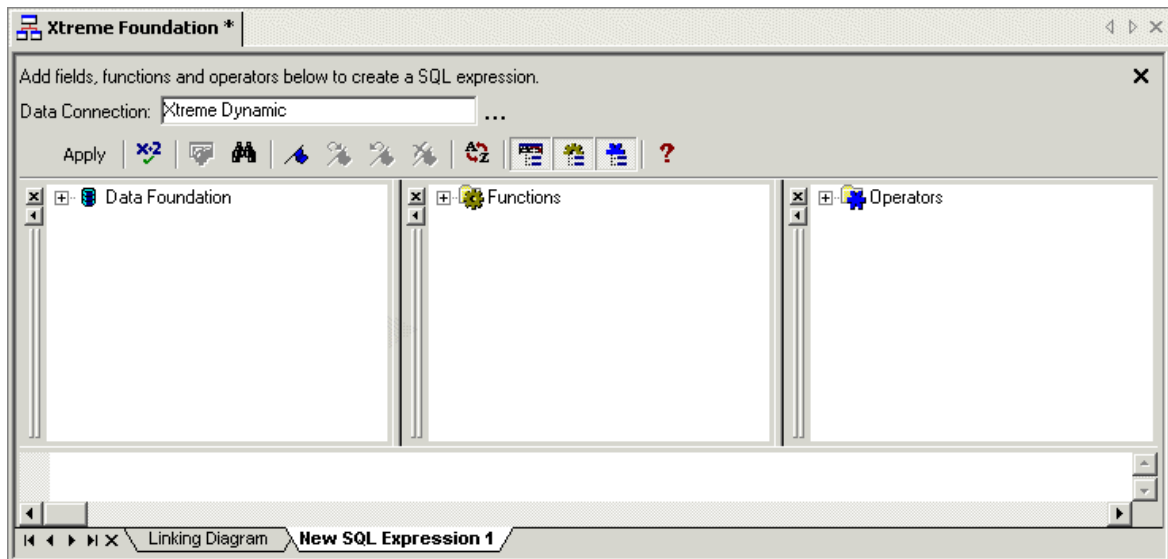
1. On the [Insert](#) menu, click [Insert SQL Expression](#).

→ Tip



In the Object Explorer, you can right-click SQL Expressions and select Insert SQL Expression. Alternatively, on the toolbar, you can click the Insert SQL Expression button.

The SQL expression Editor appears.



2. Enter the SQL expression by typing in the components or by selecting them from the component trees.



3. Click [Check](#) to test if there are any errors in the SQL expression.

4. Fix any syntax errors the SQL Expression Checker identifies.

5. When the expression has the correct syntax, click [Apply](#) to save the SQL expression.

When you create a Business Element, you can then select from a number of saved SQL expressions in the Data Foundation.

7.2.9.3 Updating the SQL expression through the Property Browser

The Property Browser displays all of the properties that you can edit or change for an SQL expression you have selected in the Object Explorer.

- [Name](#)
The name of the SQL expression.
- [Description](#)
A description you enter for the SQL expression.
- [Field Type](#)
Displays the data field type.

- [Formula Syntax](#)
Displays that the syntax used is SQL.
- [Formula Text](#)
Displays your formula.
- [Data Connection](#)
Specifies which Data Connection the SQL expression is using.
- [Show in Linking Diagram](#)
Select True or False from the list. If you select True, the SQL expression will be shown in the Linking Diagram.
- [Rights](#)
The user and group rights that are associated with the SQL expression. To update the rights, ensure that the Rights cell is selected and then click the ... button. In the Edit Rights dialog box, you can specify whether users or groups can or cannot "View field data." This right is either explicitly granted or denied.

7.2.9.4 Setting rights for an SQL expression

By editing the user and group rights for an SQL expression in a Data Foundation, you ensure that only specific users and groups can access specific SQL expressions in the Data Foundation. Thus, when a user creates a Business Element, he or she can view and add only SQL expressions that he or she has access rights to.

Note

Before you can edit the security rights for an SQL expression, you must first save the Data Foundation to the repository.

Business Views object rights are dependent on folder rights. If you set a right on a folder, all of the objects in this folder inherit the same security rights. Similarly, all of the objects within the Data Foundation also inherit security rights from the rights set at the Data Foundation level. Thus, if a user does not have Set Security rights for a Data Foundation, he or she cannot edit the rights for an SQL expression. For more information on the inheritance model as it pertains to security, see [Using inheritance to your advantage \[page 192\]](#).

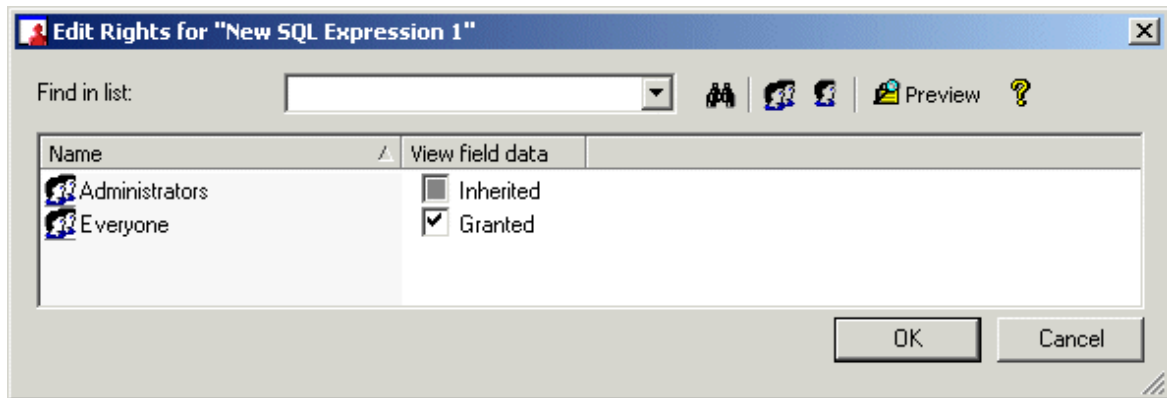
7.2.9.4.1 To apply security settings to an SQL expression

1. In the Object Explorer, select the SQL expression you want to apply security settings to, right-click the SQL expression, and select [Edit Rights](#).

→ Tip

You can also select the Rights cell in the Property Browser and click the ... button. Alternatively, you can select Edit Rights on the Edit menu.

The Edit Rights dialog box appears.

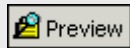


2. Set the right for either a user or a group by clicking the appropriate check box under the following column:
 - View field data
This right specifies whether a user or group can or cannot view the SQL expression and its associated data. You can explicitly grant or deny this right.

Note



A cleared check box indicates that the user or group is denied the right; a check mark in the check box indicates that the user or group is granted the right; a shaded check box indicates that the right is inherited. Also, note that a right may be shown as being granted, but may be denied once inheritance is taken into account.

Note



Preview

Click the Preview button to view the final security settings for the SQL expression with inheritance in effect. The Business View Manager resolves the net result of the current security settings and displays the net rights.

3.   To add another user or group, click the [Add Users](#) button or click the [Add Groups](#) button.
4. Assign rights to the new user or group as needed.
5. Click [OK](#).

7.2.10 Inserting a parameter

Parameters prompt the user of a report to enter information. Think of a parameter as a question that the user needs to answer before the report is generated from the Business View information. The information that users enter—or the way they respond—determines what appears in the report. For example, in a report used by salespeople, there might be a parameter that asks the user to choose a region. The report would return the results for the specific region chosen by the user, instead of returning the results for all of the regions.

For detailed information on parameter fields and on advanced parameter features, see the "Parameter Fields" section of the *Crystal Reports Online Help*.

7.2.10.1 Parameter field considerations

When working with parameter fields in a Data Foundation, there are a number of things to keep in mind:

- Parameter fields support the following data types:
 - Boolean: Requires a yes/no or true/false answer.
Example: Include regional sales numbers in the summary?
 - Currency: Requires a dollar amount.
Example: Display order amounts that are over \$50,000.
 - Date: Requires an answer in a date format.
Example: Enter the start and end dates of the fiscal quarter.
 - DateTime: Requires both date and time.
Example: Display statistics for 07/17/2003 between 5:00pm-7:00pm.
 - Number: Requires a numeric value.
Example: Enter the employee identification number.
 - String: Requires a text answer.
Example: Enter the region for the sales office.
 - Time: Requires an answer using a time format
Example: Display the total number of sales from 2:00pm-5:00pm.
- Parameter field prompting text can be more than a line 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 pick list for users to choose the parameter value rather than having them enter it manually.
- You can create a parameter field and then enter it in your formula as you would any other field.
- Like formulas, SQL expressions, and filters, you can set and apply user and group rights for specific parameters.

7.2.10.2 To create a parameter

You must create and define your parameter field in your Data Foundation before you can use it in formulas.

1. On the *Insert* menu, click *Insert Parameter*.

→ Tip



In the Object Explorer, you can right-click Parameters and select Insert Parameter. You can also click the Insert Parameters button on the toolbar.

The Create Parameter dialog box appears.

Create Parameter

Name: Type: List of Values:

Value Field: Description Field:

Value Options:

Option	Setting
Prompt Text	Enter My Parameter:
Prompt With Description Only	False
Allow custom values	True
Allow multiple values	False
Allow discrete values	True
Allow range values	False

OK Cancel Help

- Enter a name for the parameter in the [Name](#) field (up to 255 alphanumeric characters).
- Select the appropriate [Type](#) from the list.

Note

By default, the String value type is selected. For more information on value types, see [Parameter field considerations \[page 128\]](#).

- Choose your list-of-values type.

Note

By default, the Static list-of-values type is selected. For more information on dynamic prompts and cascading lists, see [Dynamic Prompts and Cascading Lists of Values \[page 160\]](#).

- From the [Value Field](#) list, select the field you wish to prompt on.
- Click [Actions](#) and select [Append all database values](#) to move all of the values of the select field in the sample database to the [Values](#) area.

This will enable the user to choose from any of the values. If you want to limit the selection, manually enter only the values that you would like the user to choose from.

- Click [OK](#).

7.2.10.3 Options for all parameter value types except Boolean

Prompt Text

Enter the text that you want the program to show your users as a prompt. For example, if you are prompting for a state value, you might want to enter a prompt like "Enter the state that you want sales data for."

Prompt With Description Only

Choose whether your user should be prompted with only a description (True) or both a value and its description (False). By default, the program sets this field to False.

Default Value

Add a value to this field if you want a default to appear when the program prompts your users to select a value.

Note

This option is available only for parameters with static prompts.

Allow custom values

If you set this option to True, your users can choose values that you entered or values of their own choice.

Allow multiple values

Use the "Allow multiple values" field to select more than one default value for your parameter field when you click the Refresh button on the Navigation Tools toolbar.

When this field is set to True, the prompt allows multiple values. When set to True, you can set both "Allow discrete values" and "Allow range values" to True. Otherwise, only one of these two fields can be set to True at any given time. The field defaults to False.

Allow discrete values

Use the "Allow discrete values" field to select only discrete, or singular, values for your parameter field. If you have "Allow multiple values" selected, you will be able to select more than one discrete value for your parameter field, though those discrete values will continue to function singularly as opposed to a range value.

When this field is set to True, the prompt allows only discrete values. The field defaults to True.

Allow range values

Use the "Allow range values" field to select a range of values for your parameter field. For example, if you've selected a currency field, your parameter field (with range value selected) could equal any number that falls between \$10,000 and \$100,000.

When this field is set to True, the prompt allows range values. The field defaults to False.

Min length

Use the "Min length" field to limit the minimum number of characters that can appear in your parameter field. For example, if you choose the number 4 as your minimum length limit, the name of customer Dan could not be used because it is fewer than 4 characters long.

Max length

Use the "Max length" field to limit the maximum number of characters that can appear in your parameter field. For example, if you choose the number 5 as your maximum length limit, the name of customer Margaret could not be used because it is more than 5 characters long.

Edit mask

Use the "Edit mask" field to create an Edit Mask for your parameter. The 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 your users can enter as prompting values).

Note

When you enter an Edit Mask, the length options become inactive.

7.2.10.4 Static Boolean parameter type

Prompt Text

Enter the text that you want the program to show your users as a prompt. For example, if you are prompting for a state value, you might want to enter a prompt like "Enter the state that you want sales data for."

Prompt With Description Only

Choose whether your user should be prompted with only a description (True) or both a value and its description (False). By default, the program sets this field to False.

7.2.10.5 Using the Property Browser for a parameter

The Property Browser displays all of the properties that you can edit or change for a parameter you have selected in the Object Explorer.

- *Name*
The name of the parameter. You can change the name of the parameter here.
- *Prompting text*
This is the prompting text that appears when a user reports off a Business View that references the selected parameter.
- *Field Type*
Displays the field type for the parameter.
- *Allow Multiple Value*
Displays whether multiple values are allowed or not.
- *Allow Null Value*
Displays whether null values are accepted or not.
- *Parameter Type*
Displays the parameter type.
- *Default Value*
Displays the default value for the parameter.

7.2.11 Inserting a filter

By default, a Full Data Access filter and a No Data Access filter are available for your Data Foundation. Apply the Full Data Access filter to a Data Foundation, field, formula, SQL expression, or other filter to give a user or group full data access. Apply the No Data Access filter to prevent data access for a user.

You can also create your own filters to apply to your Data Foundation. These filters allow you to reference fields, formulas, SQL expressions, parameters, and other filters. You can use Boolean operators to create business

filters that restrict access to certain information for specific users or groups. Business Filters enable you to apply row-level security to your data.

Note

Row-level security is also provided by processing extensions in SAP BusinessObjects BI platform. Filters in Business Views enable you to easily and quickly apply row-level security without using the SAP BusinessObjects BI platform SDK.

7.2.11.1 Key uses of filters

When you create a Business Filter, you can use this filter to specify the records you want included in your report—you select the field(s) that you want to apply selection conditions to and then you specify these conditions through dynamic drop-down lists that are contextually relevant to the select field(s).

Filters can be used to create simple record selection requests. For example:

- Customers from Alaska.
- Orders in the second quarter.
- Sales over \$50,000.

Filters can also be used to set up some sophisticated requests:

- Customers whose names start with "E," "N," "S," or "T."
- Sales representatives from California or Texas who sold units in June.

These are all range limit requests; that is, one or more constants define the range. Business Views compares the field value in each record to the constant(s) and rejects records with values outside the range. A report you create with these filters is limited to values within the range.

Besides row-level security, you can also apply filters to ensure that users, when they refresh a report, only request from the data source the relevant records. Instead of requesting that the database return all of the records for a field, a filter ensures that only certain records are returned. By using a filter, you can improve performance across an enterprise system by minimizing the number of records a database has to return.

7.2.11.2 To create a Business Filter

You must first create and define your filter at the Data Foundation level before you can apply this filter to a field, formula, SQL expression, or other filter.

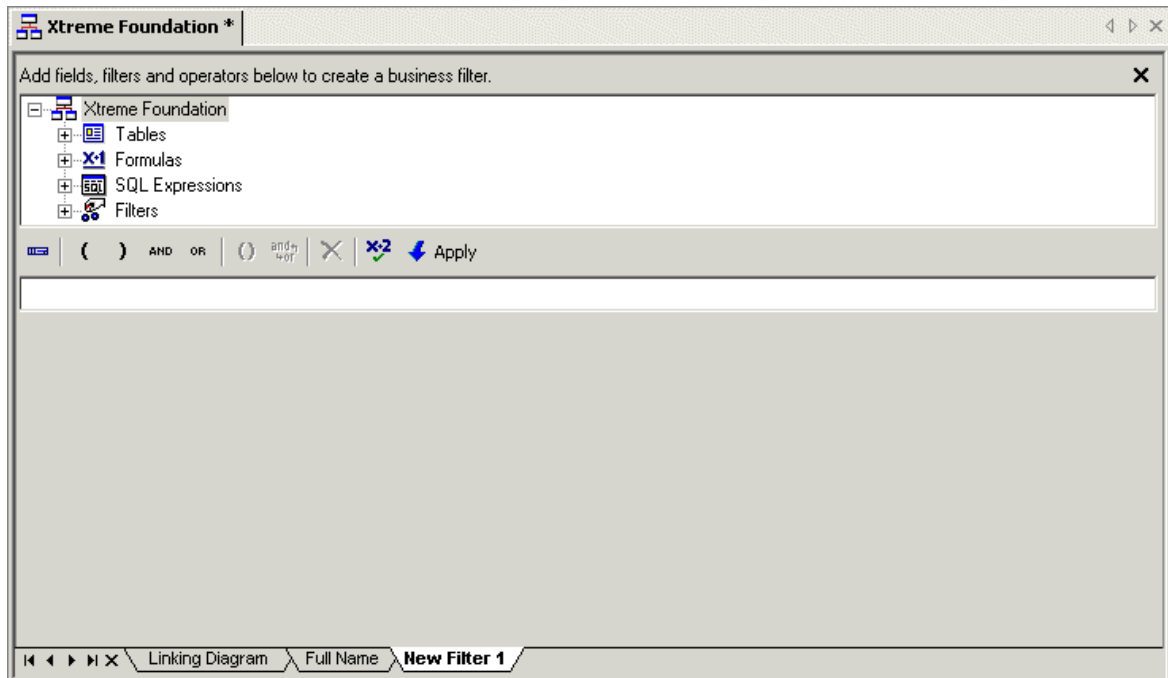
1. On the *Insert* menu, click *Insert Filter*.

→ Tip



In the Object Explorer, you can right-click Filters and select Insert Filter. You can also click the Insert Filter button on the toolbar.

The Filter Editor appears.




2. In the "Add fields, filters and operators below to create a business filter" area, navigate to a specific object and double-click the object.

→ Tip



You can also click the "Add Selected Tree Item to Filter" button that is located on the Filter Editor toolbar. Alternatively, you can drag and drop filters to the field below the Filter Editor toolbar.

3. Use the dynamic drop-down lists to enter your selection criteria for the selected field.
4. If needed, insert a Boolean operator by clicking a specific Boolean operator button on the Filter Editor toolbar.
5. Continue adding objects (and entering your selection criteria) and Boolean operators as required.
6.  Click the [Check Filter Validity](#) button to verify that there are no errors in the filter.
7. Click the [Apply](#) button to save your filter.

7.2.11.3 Using the Property Browser for a filter

The Property Browser displays all of the properties that you can edit or change for a filter you have selected in the Object Explorer.

- [Name](#)
The name of the filter.
- [Description](#)
A description you enter for the filter.

- [Rights](#)

The user and group rights that are associated with the filter. To update the rights, ensure that the Rights cell is selected and then click the ... button. In the Edit Rights dialog box, you can specify which users and groups to apply the filter to. This right for filters is either explicitly granted or denied.

7.2.11.4 Setting rights for a filter

By editing the user and group rights for a filter in a Data Foundation, you ensure that only specific users and groups can access specific filters in the Data Foundation. Thus, when a user creates an object that includes a filter, he or she can view and add only filters that he or she has access rights to.

ⓘ Note

Before you can edit the security rights for a filter, you must first save the Data Foundation to the repository.

Business Views object rights are dependent on folder rights. If you set a right on a folder, all of the objects in this folder inherit the same security rights. Similarly, all of the objects within the Data Foundation also inherit security rights from the rights set at the Data Foundation level. Thus, if a user does not have Set Security rights for a Data Foundation, he or she cannot edit the rights for a filter. For more information on the inheritance model as it pertains to security, see [Using inheritance to your advantage \[page 192\]](#).

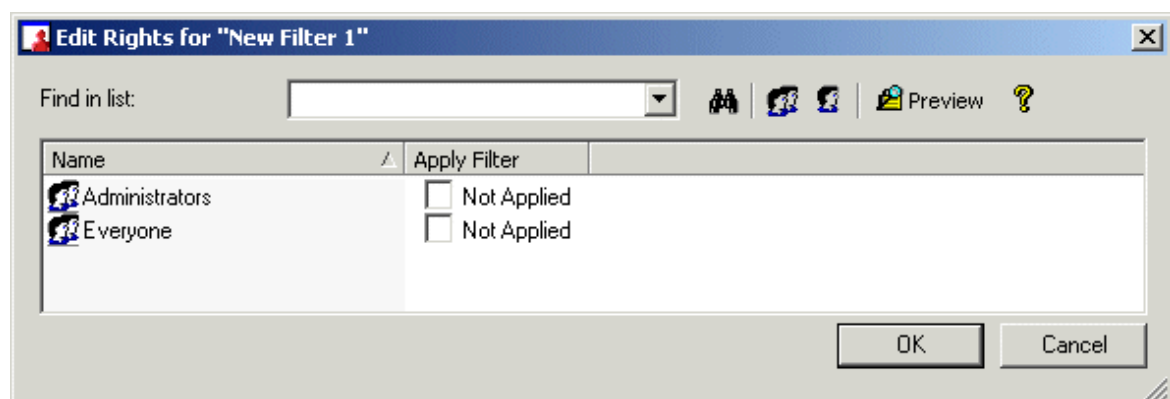
7.2.11.4.1 To apply security settings to a filter

1. In the Object Explorer, select the filter you want to apply security settings to and right-click the field and select [Edit Rights](#).

→ Tip

You can also select the Rights cell in the Property Browser and click the ... button. Alternatively, you can select Edit Rights on the Edit menu.

The Edit Rights dialog box appears.

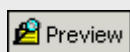


2. Set the right for either a user or a group by clicking the appropriate check box under the following column:
 - **Apply Filter**
This right specifies whether or not the specified filter is applied to a particular user or group. You can explicitly grant or deny this right.


Note

A cleared check box indicates that the user or group is denied the right; a check mark in the check box indicates that the user or group is granted the right; a shaded check box indicates that the right is inherited. Also, note that a right may be shown as being granted, but may be denied once inheritance is taken into account.

Note



Click the Preview button to view the final security settings for the filter with inheritance in effect. The Business View Manager resolves the net result of the current security settings and displays the net rights.

3.  To add another user or group, click the [Add Users](#) button or click the [Add Groups](#) button.
4. Assign rights to the new user or group as needed.
5. Click [OK](#).

7.2.12 Importing a custom function

Custom functions are procedures you create in SAP Crystal Reports to evaluate, to make calculations on, or to transform data. When you use a custom function in a formula, all of the operations in its definition are performed without you having to specify them individually in the formula itself. Thus, a custom function provides a way for you to share and reuse formula logic and makes it easier and less time consuming for you and your users to create different Business Views objects and reports.

You create custom functions using SAP Crystal Reports and then save the custom functions to the repository. In the Business View Manager, you reference custom functions and then include them in your formulas.

For more information on custom functions, see the "Using Custom Functions" chapter in the *SAP Crystal Reports Online Help*.

7.2.12.1 To import a custom function

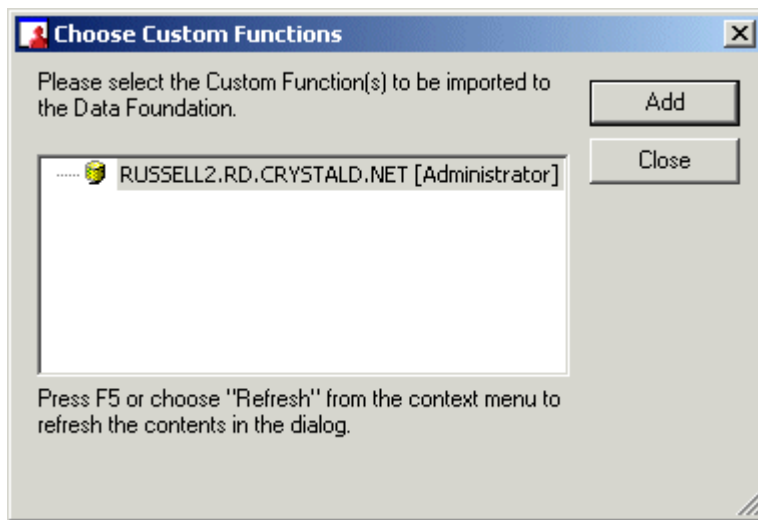
1. On the [Insert](#) menu, click [Import Custom Function](#).

→ Tip



In the Object Explorer, you can right-click Custom Functions and select Import Custom Function. You can also click the Import Custom Functions button on the toolbar.

The Choose Custom Functions dialog box appears.



2. Navigate to a folder that contains a custom function, select the custom function, and click [Add](#).
3. Click [Close](#) to continue.

When you create a formula, you can include any imported custom functions. In the Object Explorer, expand your Data Foundation node and select an imported custom function.

7.2.12.2 Using the Property Browser for a custom function

The Property Browser displays all of the properties for a custom function that you have selected in the Object Explorer.

- [Name](#)
The name of the custom function.
- [Description](#)
A description of the custom function.
- [Author](#)
The name of the user who created the custom function.
- [Formula Text](#)
Displays the formula for the custom function.
- [Category](#)
Displays what category the custom function is under.
- [ReturnType](#)
Specifies the data type that the selected custom function will return.

7.2.12.3 Setting rights for a custom function

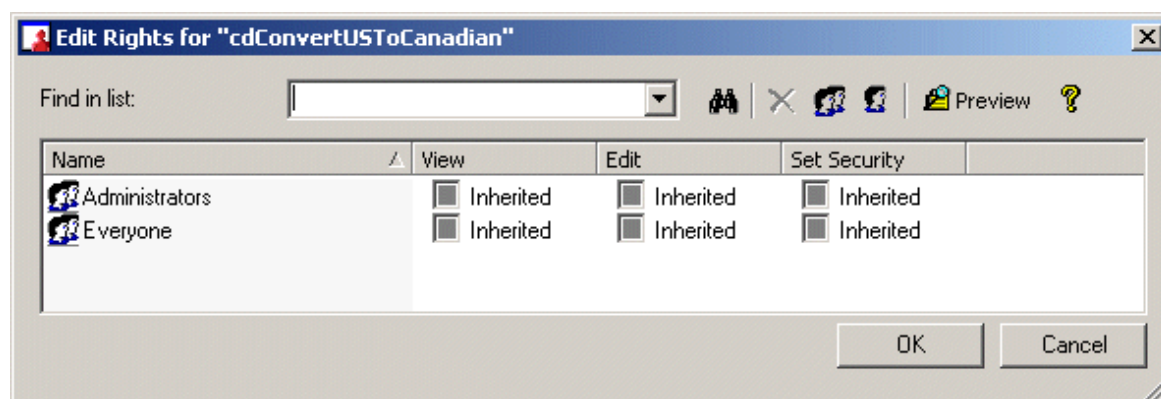
After importing a custom function, you can set user and group rights. Users can then only add or view custom functions that they have access to.

Because object rights are dependent on folder rights, you can also set folder rights for the Custom Functions folders. All of the custom functions then inherit these rights. For more information on the inheritance model as it pertains to security, see [Using inheritance to your advantage \[page 192\]](#).

7.2.12.3.1 To apply security settings to a custom function

1. In the Repository Explorer, expand the *Custom Functions* folder.
2. Select the custom function you want to apply security settings to, right-click the custom function, and select *Edit Rights*.

The Edit Rights dialog box appears.

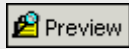


3. Set the following rights for either a user or a group by clicking the appropriate check box under each rights column.
 - View
This right specifies whether a user or group can or cannot view the custom function.
 - Edit
This right specifies whether a user or group can or cannot edit the properties of the specific object. Note that you can only add or delete custom functions through the Business View Manager.
 - Set Security
This right specifies whether a user or group can or cannot modify the rights that are associated with the custom function.



Note

A cleared check box indicates that the user or group is denied the right; a check mark in the check box indicates that the user or group is granted the right; a shaded check box indicates that the right is inherited. Also, note that a right may be shown as being granted, but may be denied once inheritance is taken into account.

Note



Click the Preview button to view the final security settings for the custom function with inheritance in effect. The Business View Manager resolves the net result of the current security settings and displays the net rights.

4.   To add another user or group, click the [Add Users](#) button or click the [Add Groups](#) button.
5. Assign rights to the new user or group as needed.
6. Click [OK](#).

7.2.13 Using the Referenced Data Connections window

You can use the Referenced Data Connections window to add additional data tables and stored procedures. You can also use this window to add a new Data Connection or Dynamic Data Connection. You can also refresh the Referenced Data Connections window and set the Database Explorer Options. When you double-click a data table or a stored procedure, this table or procedure is added to the Linking Diagram tab of the main Data Foundation window. Lastly, you can also add SQL commands through the Referenced Data Connections window.

To display the Referenced Data Connections window, on the View menu, select Referenced Data Connections. Note that the Referenced Data Connections window contains all of the same functionality as the Insert Data Tables dialog box.

- For more information on adding a data table, adding a new connection, or adding an SQL command, see [Inserting data tables \[page 110\]](#).
- For more information on setting the options found in the Database Explorer Options dialog box, see [Setting the Database Explorer Options \[page 113\]](#).

7.2.14 Using the Property Browser

The dynamic Property Browser displays all of the properties that you can edit or change for a Data Foundation. When you select a specific table, field, formula, SQL expression, filter, parameter, or custom function, the information in the Property Browser changes.

- [Name](#)
The name of the Data Foundation.
- [Description](#)
A description you enter for the Data Foundation. This description appears when you point your mouse over the specific Data Foundation in the Repository Explorer or when you have to choose a Data Foundation. This information also appears when you right-click a Data Foundation in the repository and select Properties.
- [Author](#)
By default, this field contains the name of the user who created the Data Foundation. The name of the author appears when you point your mouse over the specific Data Foundation in the Repository Explorer or when you have to choose a Data Foundation.

- [Parent Folder](#)
The repository folder in which the Data Foundation is located. This is the only property that cannot be directly modified through the Property Browser.
- [Allow Table Joins Override](#)
By default, this property is set to False. If you set it to True, then Business Views that are based on your Data Foundation will be able to override table joins, if necessary. For more information on overriding table joins, see [Overriding the linking in a Data Foundation \[page 177\]](#).
- [Rights](#)
The user and group rights that are associated with the Data Foundation. To update the rights, ensure that the Rights cell is selected and then click the ... button. For more information on editing rights, see [Editing rights for a Data Foundation \[page 141\]](#).

You can also view some of the Data Foundation's properties by right-clicking a Data Foundation in the Repository Explorer and selecting Properties. The Properties dialog box displays the name of the object, the type of object, the date the object was last saved. This dialog box also displays the object's description.

7.2.15 Using the Object Explorer

The Object Explorer displays the Data Foundation, tables and fields, formulas, SQL expressions, filters, parameters, and any custom functions that are imported. By right-clicking on any of these objects, you can perform different functions. Note that the shortcut menus are context-sensitive; some of the features are only available to certain objects.

- [Insert object](#)
You can insert data tables, formulas, SQL expressions, parameters, and filters. You can also import custom functions that have been saved to the repository.
- [Edit object](#)
When you edit the specific object, the object's relevant dialog box or editor appears.
- [Set Table Location](#)
Select Set Table Location to bring up the Set Table Location dialog box. You can update the table location for the current table.
- [Browse object](#)
You can browse a table or field.
- [Delete](#)
Select Delete to delete the object.
- [Edit Rights](#)
You can edit user and group rights for each object in the Object Explorer. For more information on rights for each object, refer to the help sections for each object:
 - [Editing rights for a field \[page 112\]](#)
 - [Setting rights for a formula \[page 122\]](#)
 - [Setting rights for an SQL expression \[page 126\]](#)
 - [Setting rights for a filter \[page 135\]](#)

7.2.16 To save a Data Foundation

Data Foundations, like all of the other Business Views objects, are saved to the repository. For more information on the repository, see [Using the Repository Explorer \[page 62\]](#).

When you first save a Data Foundation, you need to specify a name for the object and the location you want to save the object to.

1. On the [File](#) menu, select [Save](#).

→ Tip



You can also click the Save (Ctrl+S) button or press CTRL+S.

The Save As dialog box appears.

2. In the [Object Name](#) field, specify the name of your object.
3. Select the folder you want to save your object to.

→ Tip



You can create new folders in the repository by clicking the "Insert a new folder" button.

4. Click [Save](#).

7.2.17 Editing rights for a Data Foundation

You can set user and group rights settings for a Data Foundation to ensure that only specific users and groups can access the Data Foundation. Since the Business Views security model is based on the SAP BusinessObjects security model, you have a flexible security model that you can work from. For example, if you explicitly deny a View right for a user or group for a specific Data Foundation, this user or group, when creating a Business Element, will not be able to view or select from any Data Foundation to which they do not have View rights.

ⓘ Note

Before you can edit the security rights for a Data Foundation, you must first save the Data Foundation to the repository.

Business Views object rights are also dependent on folder rights. If you set a right on a folder, all of the objects in this folder inherit the same security rights. For more information on the inheritance model as it pertains to rights, see [Using inheritance to your advantage \[page 192\]](#).

Before you can apply security rights to an object, you must have Set Security rights to the object. By default, members of the Administrators group have full access to all of the folders and objects in the repository.

ⓘ Note

For detailed information on the Edit Rights dialog box, see [Using the Edit Rights dialog box \[page 188\]](#).

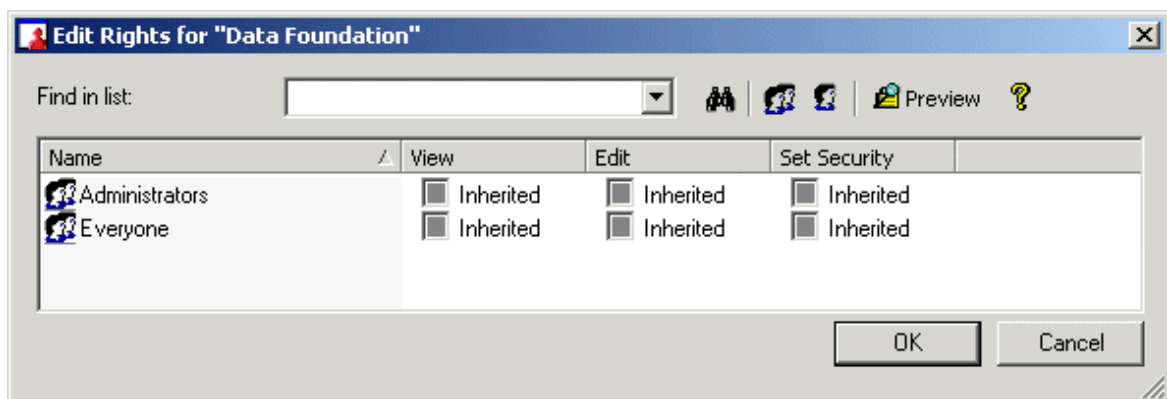
7.2.17.1 To apply security settings to a Data Foundation

1. In the Object Explorer, right-click the Data Foundation and select *Edit Rights*.

→ Tip

You can also select the Rights cell in the Property Browser and click the ... button. Alternatively, you can select Edit Rights on the Edit menu.

The Edit Rights dialog box appears.

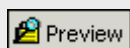


2. Set the following rights for either a user or a group by clicking the appropriate check box under each rights column.
 - View
This right specifies whether a user or group can or cannot view the Data Foundation.
 - Edit
This right specifies whether a user or group can or cannot edit the properties of the Data Foundation.
 - Set Security
This right specifies whether a user or group can or cannot modify the rights that are associated with the Data Foundation.

ⓘ Note

A cleared check box indicates that the user or group is denied the right; a check mark in the check box indicates that the user or group is granted the right; a shaded check box indicates that the right is inherited. Also, note that a right may be shown as being granted, but may be denied once inheritance is taken into account.

ⓘ Note



Click the Preview button to view the final security settings for the object with inheritance in effect. The Business View Manager resolves the net result of the current security settings and displays the net rights.

3.   To add another user or group, click the *Add Users* button or click the *Add Groups* button.

4. Assign rights to the new user or group as needed.
5. Click [OK](#).

8 Managing Business Elements

This section provides information on creating and modifying Business Elements.

8.1 Business Elements overview

A Business Element is roughly equivalent to an OLAP dimension or logical view. It is comprised of a logically related collection of data fields that are based on a Data Foundation. These fields can be organized hierarchically, resulting in different levels within a Business Element. A common example of a Business Element is a hierarchical structure that contains the following fields: Country, State or Province, and City.

You can change the alias names of the Business Fields in a Business Element to help make field names easier and less cryptic for a report designer. For example, if an employee field has the name Efield_0288, you can change the name, at the Business Field level, to Employee_Field so that a report designer can more easily select the correct field for a report. For more information on changing the field names, see [Using the Property Browser for a Business Field \[page 147\]](#).

Note

In Business Views, each object is interconnected with other objects—that is, objects must be created in a specific order. For example, you must first create a Data Connection or Dynamic Data Connection before you can create a Data Foundation. For a Dynamic Data Connection—since this connection represents a collection of Data Connections—you must first create more than one Data Connection. Then, after you have a Data Foundation, you can create a Business Element. You can create multiple Business Elements from a single Data Foundation. After you finish creating a Business Element, you can then create a Business View—a view that report designers have access to. For more information, see [Information flow \[page 14\]](#).

This section consists of two major parts:

- [Working with Business Elements \[page 144\]](#)
This section details information on creating and modifying a Business Element. It also provides information on setting user and group rights for a Business Element.
- [Using the Business Element Wizard \[page 157\]](#)
This section takes you step-by-step through the Business Element Wizard. Using the Wizard, you can create more than one Business Element at a time. When you are creating a Business Element for the first time, it is recommended that you use the Business Element Wizard.

8.2 Working with Business Elements

A Business Element consists of a logically related collection of data fields that are based on a Data Foundation. While adding fields from different tables, you can also add the formulas and SQL expressions that you have saved in the Data Foundation.

Note

When you add formulas and SQL expressions from a Data Foundation, they are added as Business Fields, and are considered fields in a Business Element.

It is recommended that you use the Business Elements Wizard to create your Business Element, as the wizard guides you through the process. For more information on the Business Elements Wizard, see [Using the Business Element Wizard \[page 157\]](#).

8.2.1 To create a Business Element

When you create a new Business Element, you need to specify the Data Foundation and then insert Business Fields from the Data Foundation.

1. On the **File** menu, select **New**, and then select **Business Element**.

The "Choose a Data Foundation" dialog box appears.

2. Expand the folders as necessary and select a Data Foundation.
3. Click **OK** to continue.

The "Insert Business Fields" dialog box appears.

4. Expand and select the Business Fields in tables, formulas, and SQL expressions; then, click **Add**.
5. Continue adding fields, formulas, and SQL expressions.
6. Click **Close**.

8.2.2 Verifying object dependency



You can verify the dependency of other objects that are affected by the settings you specify in the Business Element. Select Check Dependent Integrity on the Tools menu (or click the Check Dependent Integrity button located on the toolbar) to test the Business Element and the Business Views objects that depend on the settings specified in the Business Element.

Changes you make to the Business Element can affect Business Views. This verify tool checks the dependencies to ensure that any changes you make to the Business Element do not affect the other Business Views.

8.2.3 Showing dependent and referenced objects

You can view the Business Views objects that depend on your Business Element. Similarly, you can view the Business Views objects that are referenced by your Business Element.

To view dependent or referenced objects, in the Tools menu, select either Show Dependent Objects or Show Referenced Objects. A dialog box appears; in this dialog box, click the Save To File button to save the list of dependent or referenced objects to a text file for future reference.

The list of dependent objects displays objects that are affected by the Business Element (for example, Business Views). The list of referenced objects displays objects that the Business Element references (Data Connections, Dynamic Data Connections, and/or Data Foundations). Expand the nodes to see all of the objects that depend on or are referenced by your Business Element.

8.2.4 Modifying a Business Element

After you create a Business Element, the main window displays the Fields List tab. This tab lists all of the inserted Business Fields and also displays the field type and field length. You can sort the Business Fields by name, type, length, source, and description by clicking the heading bar for each column.

You can set the hierarchical structure of your Business Fields, insert additional Business Fields, insert a filter, insert a parameter, and change aliases for a Business Field through the Property Browser. For more information on the different modifications you can make to a Business Element, see the following:

- [To set or reset the field structure \[page 146\]](#)
- [To insert a Business Field \[page 147\]](#)
- [Inserting a filter \[page 147\]](#)
- [Inserting a parameter \[page 151\]](#)
- [Using the Referenced Data Foundation window \[page 153\]](#)
- [Using the Property Browser \[page 153\]](#)
- [Using the Object Explorer \[page 154\]](#)

8.2.5 To set or reset the field structure

The Field Structure tab, located on the main Business Element window, displays the current hierarchical structure of your Business Fields. On this tab, you can rearrange the hierarchical order of your Business Fields and also expand and minimize the different hierarchy views.

A common example of a hierarchical level is a Business Element that contains the following fields: Country, State or Province, and City.

→ Tip



You can reset the hierarchical order of your Business Fields by clicking the Reset Field Structure button on the toolbar. Alternatively, in the Object Explorer, right-click the Business Element and select Reset Field Structure.

1. Click the *Field Structure* tab on the main Business Element window.
2. Select a Business Field and drag it to a different field to move and change the hierarchy level of the selected field.
3. Continue selecting, dragging, and moving fields as necessary.

8.2.6 To insert a Business Field

You can insert additional Business Fields to your Business Element at any time. All Business Fields are listed on the Fields List tab of the main Business Element window.

1. On the *Insert* menu, select *Insert Business Fields*.

→ Tip



You can also click the Insert Business Fields button on the toolbar.

The "Insert Business Fields" dialog box appears.

2. Expand and select the Business Fields in tables, formulas, and SQL expressions; then, click *Add*.
3. Continue adding fields, formulas, and SQL expressions.
4. Click *Close*.

8.2.6.1 Using the Property Browser for a Business Field

The Property Browser displays all of the properties that you can edit or change for a Business Field you have selected in the Object Explorer.

- *Name*
The name of the Business Field. You can change the name of this field. By changing the name of the field, you are changing the alias name of the Business Field. By changing the name, you help give Business Fields relevant names that are useful to report designers. For example, if an employee field has the name Efield_0288, you can change the name, at the Business Field level, to Employee_Field so that a report designer can more easily select the correct field for a report.
- *Description*
A description you enter for the Business Field.
- *Field Type*
This field lists the field type of the Business Field.
- *Source Data Field*
Specifies the source of the data field.
- *Rights*
The user and group rights that are associated with the Business Field. To update the rights, ensure that the Rights cell is selected, and then click the ... button. In the Edit Rights dialog box, you can specify which users and groups to apply the filter to. For more information on editing rights, see [Editing rights for a Business Element \[page 155\]](#).

8.2.7 Inserting a filter

You can create your own filters to apply to your Business Element. You create business filters through the Business View Manager. These filters allow you to reference fields, formulas, SQL expressions, parameters, and other filters (from Business Elements and from Data Foundations). You can use Boolean operators to create

Business Filters that restrict access to certain information for specific users or groups. Business Filters enable you to apply row-level security to your data.

Note

Row-level security is also provided by processing extensions in SAP BusinessObjects BI platform. Filters in Business Views enable you to easily and quickly apply row-level security without using the SAP BusinessObjects BI platform SDK.

Note

When you add formulas and SQL expressions from a Data Foundation, they are added as Business Fields and are considered as fields in a Business Element. Thus, you can select and add formulas and SQL expressions to a filter, though formulas and SQL expressions will appear under the Fields category.

Note

Crystal reports that are based on Business Views that contain a filter now support the "Push Down Group By" option. This option is only available if a field that is used by a Business Element filter is also used for grouping. This field must be the only field that is used in the filter. For example, if you have a Business Element filter set to Country = "Canada," and your report is grouped on the country field, the push down will occur. However, if the filter is set to Country = "Canada" and Sales > 100 (Sales is not part of the grouping in the report), the push down will not happen. For more information on key uses of filters, see [Key uses of filters \[page 133\]](#).

8.2.7.1 To create a Business Filter

You must first create and define your filter at the Business Element level before you can apply this filter to a field, SQL expression, or other filter.

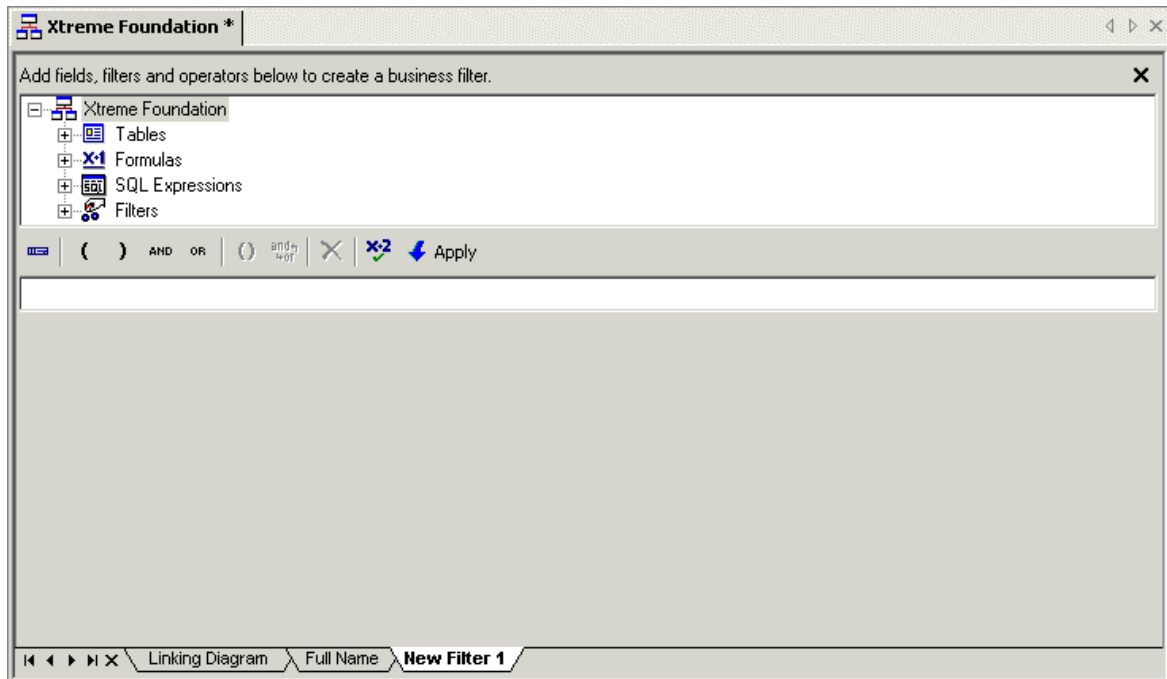
1. On the *Insert* menu, click *Insert Filter*.

→ Tip



In the Object Explorer, you can right-click Filters and select Insert Filter. You can also click the Insert Filter button on the toolbar.

The Filter Editor appears.




2. In the "Add fields, filters and operators below to create a business filter" area, navigate to a specific object and double-click the object.

→ Tip



You can also click the "Add Selected Tree Item to Filter" button that is located on the Filter Editor toolbar. Alternatively, you can drag and drop filters to the field below the Filter Editor toolbar.

3. Use the dynamic drop-down lists to enter your selection criteria for the selected field.
4. If needed, insert a Boolean operator by clicking a specific Boolean operator button on the Filter Editor toolbar.
5. Continue adding objects (and entering your selection criteria) and Boolean operators as required.
6.  Click the *Check Filter Validity* button to verify that there are no errors in the filter.
7. Click the *Apply* button to save your filter.

8.2.7.2 Using the Property Browser for a filter

The Property Browser displays all of the properties that you can edit or change for a filter you have selected in the Object Explorer.

- *Name*
The name of the filter.
- *Description*
A description you enter for the filter.
- *Rights*

The user and group rights that are associated with the filter. To update the rights, ensure that the Rights cell is selected and then click the ... button. In the Edit Rights dialog box, you can specify which users and groups to apply the filter to. This right for filters is either explicitly granted or denied.

8.2.7.3 To apply security settings to a filter

By editing the user and group rights for a filter in a Business Element, you can specify whether or not this filter can be applied to specific users and groups.

Note

Before you can edit the security rights for a filter, you must first save the Business Element to the repository.

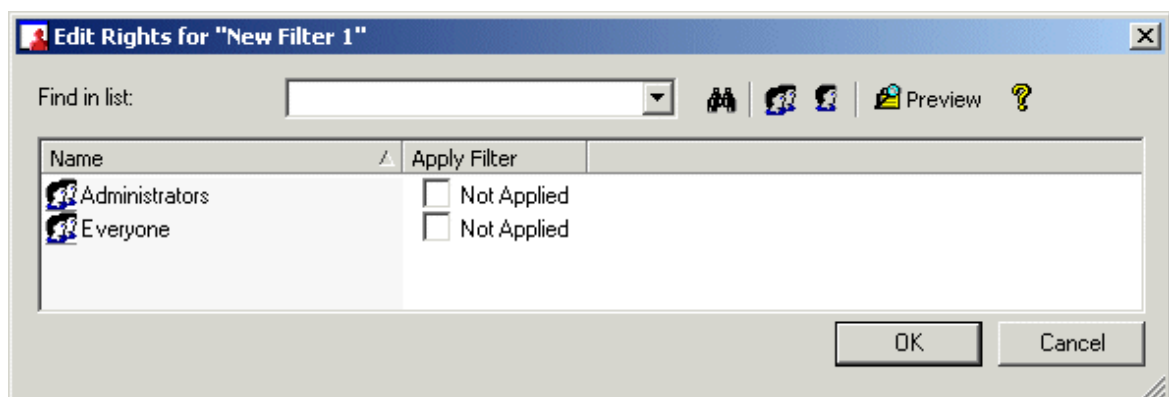
Business Views object rights are dependent on folder rights. If you set a right on a folder, all of the Business Views objects in this folder inherit the same security rights. Similarly, all of the objects within the Business Element also inherit security rights from the rights set at the Business Element level. Thus, if a user does not have Set Security rights for a Business Element, he or she cannot edit the rights for a filter. For more information on the inheritance model as it pertains to security, see [Using inheritance to your advantage \[page 192\]](#).

1. In the Object Explorer, select the filter you want to apply security settings to and right-click the filter and select [Edit Rights](#).



Tip

You can also select the Rights cell in the Property Browser and click the ... button. Alternatively, you can select Edit Rights on the Edit menu.

The Edit Rights dialog box appears.



2. Set the right for either a user or a group by clicking the appropriate check box under the following column:
 - Apply Filter
This right specifies whether or not the specified filter is applied to a particular user or group. You can explicitly grant or deny this right.

3.   To add another user or group, click the [Add Users](#) button or click the [Add Groups](#) button.
4. Assign rights to the new user or group as needed.
5. Click [OK](#).

8.2.8 Inserting a parameter

Parameters prompt the user of a report to enter information. Think of a parameter as a question that the user needs to answer before the report is generated from the Business View information. The information users enter—or the way they respond—determines what appears in the report. For example, in a report used by salespeople, there might be a parameter that asks the user to choose a region. The report would return the results for the specific region chosen by the user, instead of returning the results for all of the regions.

For detailed information on parameter fields and on advanced parameter features, see the "Parameter Fields" section of the *Crystal Reports Online Help*.

You can also add parameters to Data Foundations. Since parameters function in the same manner for both Data Foundations and Business Elements, see the following sections for more information on parameters:

- [Parameter field considerations \[page 128\]](#)
- [Options for all parameter value types except Boolean \[page 130\]](#)
- [Options for all parameter value types except Boolean \[page 130\]](#)
- [Static Boolean parameter type \[page 132\]](#)

8.2.8.1 To create a parameter field

You must create and define your parameter field in your Data Foundation before you can use it in formulas.

1. On the [Insert](#) menu, click [Insert Parameter](#).

→ Tip



In the Object Explorer, you can right-click Parameters and select Insert Parameter. You can also click the Insert Parameters button on the toolbar.

The Create Parameter dialog box appears

Create Parameter

Name: Type: List of Values:

Value Field: Description Field:

Value Options:

Option	Setting
Prompt Text	Enter My Parameter:
Prompt With Description Only	False
Allow custom values	True
Allow multiple values	False
Allow discrete values	True
Allow range values	False

OK Cancel Help

2. Enter a name for the parameter in the [Name](#) field (up to 255 alphanumeric characters).
3. Select the appropriate [Type](#) from the list.

Note

By default, the String value type is selected. For more information on value types, see [Parameter field considerations \[page 128\]](#).

4. Choose your list-of-values type.

Note

By default, the Static list-of-values type is selected. For more information on dynamic prompts and cascading lists, see [Dynamic Prompts and Cascading Lists of Values \[page 160\]](#).

5. From the [Value Field](#) list, select the field you wish to prompt on.
6. Click [Actions](#) and select [Append all database values](#) to move all of the values of the select field in the sample database to the [Values](#) area.

This will enable the user to choose from any of the values. If you want to limit the selection, manually enter only the values that you would like the user to choose from.

7. Click [OK](#).

8.2.8.2 Using the Property Browser for a parameter

The Property Browser displays all of the properties that you can edit or change for a parameter you have selected in the Object Explorer.

- *Name*
The name of the parameter. You can change the name of the parameter here.
- *Prompting text*
This is the prompting text that appears when a user reports off a Business View that references the selected parameter.
- *Field Type*
Displays the field type for the parameter.
- *Allow Multiple Value*
Displays whether multiple values are allowed or not.
- *Allow Null Value*
Displays whether null values are accepted or not.
- *Parameter Type*
Displays the parameter type.
- *Default Value*
Displays the default value for the parameter.

8.2.9 Using the Referenced Data Foundation window

You can use the Referenced Data Foundation window to view the Data Foundation that the Business Element references. The Referenced Data Foundation window is similar to an Object Explorer window for a Data Foundation: you can view the names of the various tables, formulas, SQL expressions, parameters, and filters that are saved to the Data Foundation.

Using the Referenced Data Foundation window, you can add Business Fields by dragging the Data Foundation table fields, formulas, or SQL expressions and dropping them to either the Fields List tab or the Fields node (in the Object Explorer).

→ Tip

You can quickly reuse a Data Foundation filter by selecting it and dragging and dropping the filter to the Filters node in the Object Explorer. This action creates a Business Element filter that has the same name as the Data Foundation filter (with all of the same filter settings).

To display the Referenced Data Foundation window, on the View menu, select Referenced Data Foundation.

8.2.10 Using the Property Browser

The dynamic Property Browser displays all of the properties that you can edit or change for a Business Element. When you select a specific Business Field, filter, or parameter, the information in the Property Browser changes. For more information on the Property Browser as it pertains to the different objects within a Business Element, see the following:

- [Using the Property Browser for a Business Field \[page 147\]](#)
- [Using the Property Browser for a filter \[page 149\]](#)
- [Using the Property Browser for a parameter \[page 153\]](#)

This list details the properties that are displayed when you select the Business Element object in the Object Explorer.

- *Name*
The name of the Business Element.
- *Description*
A description you enter for the Business Element. This description appears when you point your mouse over the specific Business Element in the Repository Explorer or when you have to choose a Business Element. This information also appears when you right-click a Business Element in the repository and select Properties.
- *Author*
By default, this field contains the name of the user who created the Business Element. The name of the author appears when you point your mouse over the specific Business Element in the Repository Explorer or when you have to choose a Business Element.
- *Parent Folder*
The repository folder in which the Business Element is located. This is the only property that cannot be directly modified through the Property Browser.
- *Data Foundation*
This field lists the Data Foundation that is being used for the Business Element.
- *Rights*
The user and group rights that are associated with the Business Element. To update the rights, ensure that the Rights cell is selected and then click the ... button. For more information on editing rights, see [Editing rights for a Business Element \[page 155\]](#).

You can also view some of the Business Element's properties by right-clicking a Business Element in the Repository Explorer and selecting Properties. The Properties dialog box displays the name of the object, the type of object, and the date the object was last saved. This dialog box also displays the object's description.

8.2.11 Using the Object Explorer

The Object Explorer displays the Business Element, Business Fields, filters, and parameters. By right-clicking on any of these objects, you can perform different functions. Note that the shortcut menus are context-sensitive; some of the features are only available to certain objects.

- *Insert object*
You can insert Business Fields, filters, and parameters.
- *Edit object*
When you edit the specific object, the object's relevant dialog box or editor appears.
- *Browse object*
You can browse through the data contained in a Business Field.
- *Delete*
Select Delete to delete the object.
- *Edit Rights*

You can edit user and group rights for each object in the Object Explorer. For more information on rights for each object, refer to the help sections for each object:

- [Editing rights for a Business Element \[page 155\]](#)
- [To apply security settings to a filter \[page 150\]](#)

8.2.12 To save a Business Element

Business Elements, like all of the other Business Views objects, are saved to the repository. For more information on the repository, see [Using the Repository Explorer \[page 62\]](#).

When you first save a Business Element, you need to specify a name for the object and the location you want to save the object to.

1. On the [File](#) menu, select [Save](#).

→ Tip



You can also click the Save (Ctrl+S) button or press CTRL+S.

The Save As dialog box appears.

2. In the [Object Name](#) field, specify the name of your object.
3. Select the folder you want to save your object to.

→ Tip



You can create new folders in the repository by clicking the "Insert a new folder" button.

4. Click [Save](#).

8.2.13 Editing rights for a Business Element

Set user and group rights settings for a Business Element to ensure that only specific users and groups can access the Business Element. Since the Business Views security model is based on the SAP BusinessObjects BI platform security model, you have a flexible security model that you can work from. For example, if you explicitly deny a View right for a user or group for a specific Business Element, this user or group, when creating a Business View, will not be able to view or select from any Business Element that they do not have View rights to.

ⓘ Note

Before you can edit the security rights for a Business Element, you must first save the Business Element to the repository.

Business Views object rights are also dependent on folder rights. If you set a right on a folder, all of the objects in this folder inherit the same security rights. For more information on the inheritance model as it pertains to rights, see [Using inheritance to your advantage \[page 192\]](#).

Before you can apply security rights to an object, you must have Set Security rights to the object. By default, members of the Administrators group have full access to all of the folders and objects in the repository.

Note

For detailed information on the Edit Rights dialog box, see [Using the Edit Rights dialog box \[page 188\]](#).

8.2.13.1 To apply security settings to a Business Element

1. In the Object Explorer, right-click the Business Element and select [Edit Rights](#).

→ Tip

You can also select the Rights cell in the Property Browser and click the ... button. Alternatively, you can select Edit Rights on the Edit menu.

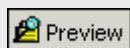
The Edit Rights dialog box appears.

2. Set the following rights for either a user or a group by clicking the appropriate check box under each rights column.
 - View
This right specifies whether a user or group can or cannot view the Business Element.
 - Edit
This right specifies whether a user or group can or cannot edit the properties of the Business Element.
 - Set Security
This right specifies whether a user or group can or cannot modify the rights that are associated with the Business Element.



Note

A cleared check box indicates that the user or group is denied the right; a check mark in the check box indicates that the user or group is granted the right; a shaded check box indicates that the right is inherited. Also, note that a right may be shown as being granted, but may be denied once inheritance is taken into account.

Note



Click the Preview button to view see the final security settings for the object with inheritance in effect. The Business View Manager resolves the net result of the current security settings and displays the net rights.

3.   To add another user or group, click the [Add Users](#) button or click the [Add Groups](#) button.
4. Assign rights to the new user or group as needed.
5. Click [OK](#).

8.3 Using the Business Element Wizard

Using the Business Element Wizard, you can quickly create multiple Business Elements directly from a Data Foundation. When you are creating a Business Element for the first time, it is recommended that you use the Business Element Wizard. For more information on Business Elements, see [Business Elements overview \[page 144\]](#).

The Business Element Wizard guides you through the steps needed to create multiple Business Elements.

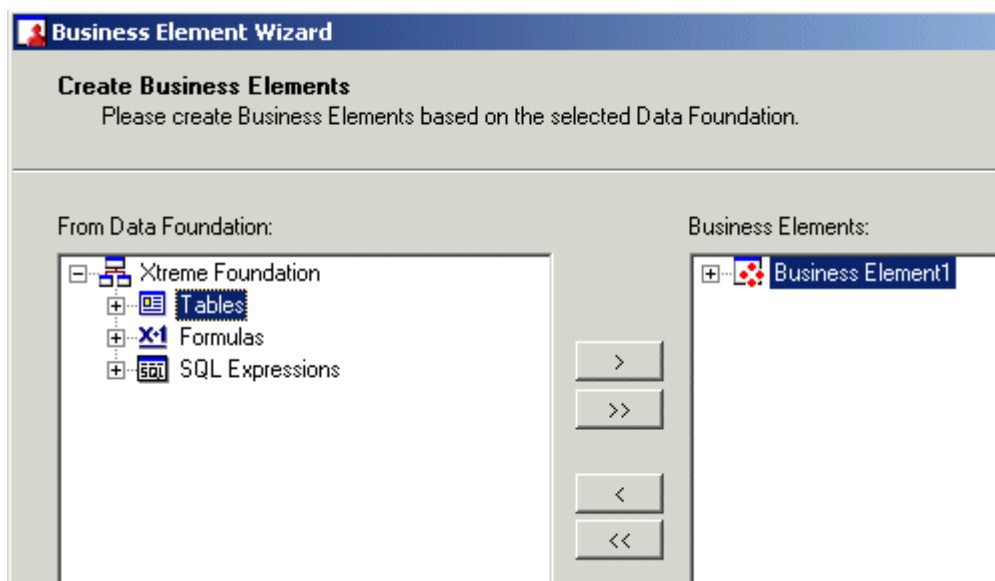
8.3.1 To specify a Data Foundation

The Business Element Wizard guides you through the steps needed to create multiple Business Elements.

1. On the *File* menu, select *New* and then select *Business Elements Wizard*.
The "Choose Data Foundation" dialog box appears.
2. Expand the folders as necessary and select a Data Foundation.
3. Click *Next* to continue.

8.3.2 Creating and specifying multiple Business Elements

In the "Create Business Elements" dialog box, you can create one or more Business Elements, rename the different Business Elements, fields, and so on.



When you select an entire table, the table is set as a Business Element. You can then delete or rename existing fields within the Business Element. You can also add additional fields from different tables (or even formulas and/or SQL expressions) to the Business Element.

→ Tip

The names you specify for the Business Elements are the names that will be saved to the repository. You cannot save Business Elements that have the same name in the same folder or subfolder.

8.3.2.1 To create Business Elements



1. Select the tables, fields, formulas, and SQL expressions that you want to add and click the > button to add them to the selected Business Element.

→ Tip

You can also add entire tables as Business Elements by selecting the table and clicking the > button.

→ Tip

You can add all of the tables, formulas, and SQL Expressions from the selected Data Foundation by clicking the >> button.

2.  To create a new Business Element, click the [New Business Element](#) button, located at the top-right corner of the Business Elements area.
3.  Rename any object in the Business Elements area by selecting the object and then clicking the [Rename the selected object](#) button.
4. Click [Next](#) to continue.
5. Specify a folder in the repository you want to save your Business Element(s) to; click [Next](#) to continue, or click [Finish](#) to exit the Wizard.

ⓘ Note

If you click Finish, the multiple Business Elements are grouped together in one Business View.

8.3.3 Specifying a remaining option in the Wizard

After you click Next, you are presented with four options. Select an option and click Finish.

- [Create a Business View](#)
Choose this option to create a new Business View that is based on the Business Elements that you created. For more information on Business Views, see [Working with Business Views \[page 174\]](#).
- [Create more Business Elements](#)

Select this option to create additional Business Elements using the Business Element Wizard. This option takes you back to the beginning of the Wizard. For more information on using the Wizard, see [Using the Business Element Wizard \[page 157\]](#).

- *[Edit the created Business Elements](#)*

Choose this option to use the Business View Manager to edit the Business Elements that you created. For more information on editing a Business Element, see [Working with Business Elements \[page 144\]](#).

- *[Exit](#)*

Choose this option to exit the Wizard. Alternatively, you can click the Cancel button.

9 Dynamic Prompts and Cascading Lists of Values

This section explains the basics of dynamic prompts and cascading lists of values.

9.1 Parameters and prompts

Parameters are Business View fields that you can use in a Business View formula in the Data Foundation and Business Element layer. Prompts work with dialog boxes that help users set a value for parameters in a report.

When used as a formula component, a parameter must have a value before the program can process the report. To use parameters in the Data Foundation and Business Element layer, you create a single Business View that can change its behavior depending on the values entered by the user.

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

- Prompts are not directly used by a Business View 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 the user sets a value in the prompting user interface, the Crystal Reports prompting engine assigns that value to the corresponding parameter, which is then used by the Data Foundation or Business Element.

9.2 Dynamic prompts overview

Dynamic prompts and cascading lists of values are now available in Crystal Reports and Business Views. These features let you populate the list of values associated with a prompt from a data source outside of your report, instead of a static list of values stored within each parameter.

Business Views has been modified to support dynamic prompts and cascading lists of values:

- Business Element and Data Foundation parameters can now use dynamic prompts and cascading lists of values.
For more information, see [Using dynamic prompts and cascading lists of values in Business Elements and Data Foundations \[page 165\]](#).
- Business View Manager is used to manage repository objects that support dynamic prompts and cascading lists of values in both Crystal Reports and Business Views.

These objects are lists of values and prompt groups, known as "prompting objects," which are explained later in this document. For more information, see [Using Business View Manager to manage prompting repository objects \[page 165\]](#).

The following prompting features are available to Business Element and Data Foundation prompts and are designed using Business View Manager. Crystal reports that use prompting objects that are defined within Business View Manager can also use these features:

- Create dynamic prompts and cascading lists of values.
- Reuse list of value definitions in many Business Elements or Data Foundations.
- Schedule lists of values.
- Populate lists of values from Business Views.
- Cache lists of values at report run time, and share that list among multiple executing reports.
- Schedule lists of values to update themselves on a regular basis.
- Schedule portions of lists of values to update themselves regularly, while the remaining portions retrieve their values from the database on demand.

9.3 Supported components

The following components in the Crystal Reports and SAP BusinessObjects BI platform products allow you to run 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, contain drop-down lists where dynamic lists of values should appear.

The supported viewers are as follows:

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

These components allow you to design reports with dynamic prompts and cascading lists of values:

- All editions of SAP Crystal Reports for Enterprise except for the Standard Edition.
- The integrated Visual Studio .NET designer.
- Business View Manager.

9.4 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 a 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.

Three data sources are supported for list-of-values objects:

- Report Fields (for report-based prompts).
- Command Objects (for report-based prompts).
- Business Views (for repository-based prompts).

Because this documentation focuses on repository-based prompts, it does not cover Report Fields or Command Objects that are used as a data source. For more information on Report Fields and Command Objects, see the *Crystal Reports User's Guide*.

A list of values can have one or more levels. If there are multiple levels in the list of values, each level constrains the next.

Unlike many other reporting products, 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.

9.5 Lists of values and prompt groups

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. They are prompts that have been split into separate objects, so that you can share the same list of values with different presentations.

For example, you may 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 list of values for cities but uses two different prompt groups (or presentation styles). One prompt group would prompt users for the supplier city and the other would prompt users for the customer cities.

9.5.1 List-of-values types

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

- Unmanaged lists of values, stored within each report file.
If you do not have SAP BusinessObjects BI platform, or if you never publish your reports to SAP BusinessObjects BI platform, you use unmanaged lists of values. Unmanaged list-of-values objects can use report fields or command objects as a data source.
- Managed lists of values, stored within SAP BusinessObjects BI platform.
Every report that you store in SAP BusinessObjects BI platform 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:

Feature	Managed list of values
Database query at the per list-of-value level.	Yes The database is queried for each level, as the user selects values in the prompting dialog box (provided the Business View is not based on a command object).
At design time, specify a filter for the list of values.	Yes Business Views support filters.
At report-view time, prompt for additional information needed by the list of values.	Yes If the Business View contains parameters, the value(s) are prompted for a report-view time.
Display different values to different users.	Yes Because managed lists of values are based on Business Views, they inherit the view-time security capabilities of Business Views.
Schedule the list of values to update on a recurring schedule.	Yes The scheduling capability comes from the Business View Manager, not the Central Management Console (CMC) where reports are scheduled.
Partially schedule the list of values to update only certain portions of the list on a recurring schedule.	Yes

9.5.2 Determining which list-of-values type to use


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

Description	Managed list of values
Single-level code tables. (A table that has hundreds of semi-static values in a single level.)	Well suited. You can define filtering within the source Business View.
Multi level-cascading code tables. (A table that has hundreds of semi-static values in multiple levels.)	Well suited. Business View-based lists of values can support both filtering and level-by-level data retrieval.
Fact tables. (These tables tend to be very large dynamic tables, with millions of values in multiple levels.)	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.

9.5.3 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.

9.5.3.1 To set separate value and description fields

1. Open the sample Business Foundation named *Xtreme Foundation*.
Sample data is located in the repository under \Samples\Business Views\Xtreme.
2. In the Object Explorer, right-click on *Parameters* and click *Insert Parameter*.
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. Under *Choose a Data Source*, choose *Select Business View*. The Select Business View dialog appears.
6. Select the Business View that you will use for your list of values, and then click *OK*.
7.  Click *Insert* and, in the *Value* field, select *Customer ID*.
8. In the *Description* field, select *Customer Name*.
9. In the Options area, set the *Prompt with Description Only* option to *True*.
10. 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.

9.5.4 Null Handling

Null values in the data that are returned by the list of values are displayed as "null" in the drop-down list for that prompt. If your users select these values, any parameters that are associated with that prompt receive the null value. You can test for this functionality with the IsNull function.

9.6 Using dynamic prompts and cascading lists of values in Business Elements and Data Foundations

Dynamic prompts and cascading lists of values can be used in Business Elements and Data Foundations to prompt for parameter values.

For more information on creating dynamic prompts and cascading lists of values in Business Elements and Data Foundations, please see [To create a list-of-values object \[page 165\]](#).

9.7 Using Business View Manager to manage prompting repository objects

Two repository objects are used to implement dynamic prompts and cascading lists of values:

- Lists of values
- Prompt groups

Although these objects are created and managed in Business Views, they can be used in Crystal Reports, Business Elements, and Data Foundation parameters.

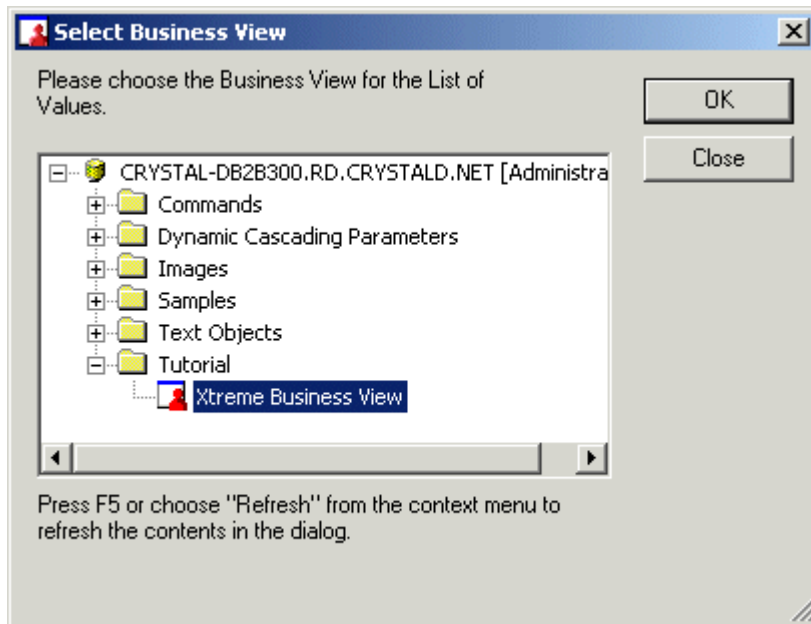
It is best to create your prompting objects first in Business Views, and then use them as needed in your Crystal Reports, Business Elements, or Data Foundation parameters.

9.7.1 Managing lists of values in Business View Manager

9.7.1.1 To create a list-of-values object

1. On the *File* menu, select *New*, then select *Create List of Values Info*.

The Select Business View dialog box appears.



2. Expand the folders and select the Business View that contains the fields that you want for your list of values.
3. Click [OK](#).

The Create List of Values dialog box appears.

Create List of Values

Name:

Available Fields:

- Database Fields
 - Customer
 - Customer ID
 - Customer Name
 - Contact First Name
 - Contact Last Name
 - Contact Title
 - Contact Position
 - Last Year's Sales
 - Address1
 - Address2
 - City
 - Region

List of Value Fields:

- Customer Name
- City
- Phone

☒ Sort by Value in Ascending Order
☐ Sort by Value in Descending Order

Description Field:

Oldest on-demand data given to a client (in seconds):

Default servers to use for viewing:

Referenced Business View:

List of Values Schedule Result:

4. In the *Name* field, specify a name for the list of values.
 5. In the Available Fields area, expand the tables and select a field that you want to include in the list.
 6. Click the right arrow to add the field to the list.
 7. Continue to select and add all of the fields that you want to include in the list of values.
 8. In the *List of Value Fields* area, select a field.
 - To change the order of the fields in your list, click the Up or Down arrow.
- The order of the fields in the list establishes the order in which information is prompted for the fields in a parameter. For example, suppose that the fields are listed in the following order:
- Country
 - Region
 - City

A parameter that is based on the list of values will first prompt users for a country, then for a region within the country, and finally, for a city within the region.

Note

The order of the fields in a list of values is especially important if you plan to use the list of values for dynamic cascading parameters. If the parameter is not cascading, it will always prompt for the first field in the list of values (which, in this case, is "Country").

- To sort the order of the values of the field, select "Sort by Value in Ascending Order" or "Sort by Value in Descending Order."

When you create a parameter that uses the list of values, the order that you specified for the field is the order in which the field's values appear in the prompt.

- To specify a description for the field, click an option from the Description Field list.
The list contains tables and fields from the Business View. From the list, you can select a field. The values of the field that you choose become the descriptions for the values of the field that is selected in the List of Value Fields area.

For example, suppose that in the List of Value Fields area, you select the field "Customer ID," then from the Description Field you choose the field "Customer Name." When you create a parameter that uses the list of values (and configure the prompt to use Value and Description), the values that are entered for the parameter are the values of the "Customer ID" field, which can now be specified by selecting a customer name. The description gives your users an easier way to provide a value for that parameter without having to remember specific customer IDs.

9. Modify the remaining options.

- Enter a value for *Oldest on-demand data given to a client (in seconds)*.
By default this field is set to 300 seconds. This means that the data that is retrieved on-demand from the database is held in memory for the specified amount of time and used by all reports that share the same list-of-values object, until this time limit expires. After the time limit is reached, the next access to this on-demand part of the list-of-values object incurs a database access, and is shared for another interval of time.

Note

If all levels of a list-of-values object are scheduled, then the "Oldest on-demand data given to a client (in seconds)" value has no effect.

- In the *Default servers to use for viewing* field, click the Report Application Server (RAS) server group that will be dedicated to processing your list-of-values object.
This allows you to define a separate RAS server group that is dedicated for list of values processing. This list shows server group names that are defined in the SAP BusinessObjects BI platform Central Management Console.

10. Click *OK*.

The Business View Manager creates the list of values and saves it in the top-level folder of the repository.

Note

No support is available for creating a list of values based on a Business View that depends on another list of values (for example, when a dynamic prompt parameter is used in the Data Foundation or Business Element). If this list of values is used for a parameter, the parameter prompting fails.

9.7.1.2 To edit a list-of-values object

Right-click the object in the Repository Explorer and select *Edit LOV*.

9.7.2 Managing security for lists of values

You can restrict the display of certain values to certain users. To do so, you must configure the security in the Business View that is used by the list-of-values object. For more information on how to configure security for an object, see [Object and folder rights \[page 191\]](#).

9.7.3 Scheduling lists of values in Business View Manager

List-of-values objects can be scheduled with Business View Manager.

By default, all new list-of-values objects execute "on-demand," which means that they access the database as needed to provide prompting values. If the Business View that the list of values references is based on a command object, all values in the list of values are retrieved at once. If the Business View that the list of values references is based on database fields, the values in the list of values are retrieved one level at a time.

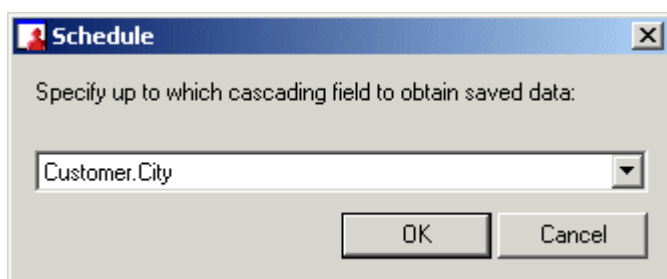
- Scheduled lists of values are useful for situations when you have some values that change on a time-driven basis or when the values change infrequently.
- Partially scheduled lists of values are useful when you want to prompt on a potentially large number of values.

9.7.3.1 To schedule a list of values

To schedule a list of values, you must log on to Business Views Manager with a user account that has the same data access rights (to all rows and columns) that all possible users of the list of values have.

1. In the Repository Explorer, right-click a list of values, and then click [Schedule LOV](#).

The Schedule dialog box appears.



From the drop-down list, select a field.

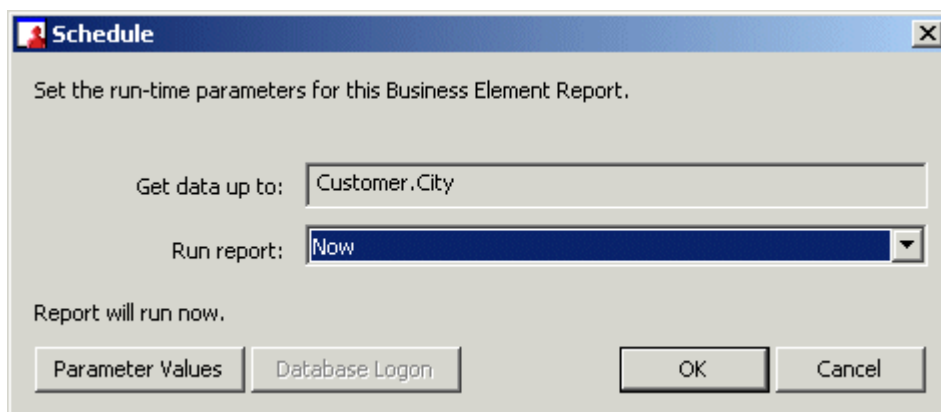
Note

The field that you specify is the level up to which data is obtained/saved. For example, if the list of values consists of "Country", "Region", and "City" levels, and you choose "Region", the data is obtained only for the "Country" and "Region" fields; data is not obtained for the "City" field and is therefore a

partially scheduled list of values. If you chose "City" though, it would become a fully scheduled list of values, and those levels that don't have scheduled data would access the database on demand.

2. Click [OK](#).

The second Schedule dialog box appears



3. In the [Run report](#) list, choose the schedule in which the list of values will be run. For more information on the different ways that you can schedule an object, see the *SAP BusinessObjects Business Intelligence platform Administrator Guide*.

Note

If your list of values is based on a Business View that contains parameters, click the Parameter Values button (at the bottom-left corner of the dialog box) to set the parameter values. The button is disabled if the Business View does not contain parameters.

Note

If your list of values is based on a Business View that requires database logon, you must also click on the Database Logon button to connect to the Business View's data source. The button is disabled if the Business View does not require database logon.

4. Click [OK](#).

The list of values is scheduled in SAP BusinessObjects BI platform.

Scheduled and partially scheduled lists of values can be converted back to on-demand lists of values.

9.7.3.2 To convert a scheduled list of values to an on-demand list of values

1. Right-click the object in the Repository Explorer and select [Edit LOV](#).
2. Click [Clear Instance](#).

The scheduled list of values instance is removed, and the data source is accessed for values at all levels.

9.7.4 Managing prompt groups in Business View Manager

Prompt groups are the presentation part of a prompt. These objects are used by Crystal Report, Business Element, and Data Foundation parameters. (For more information on creating prompt groups in Crystal Reports, see the *Crystal Reports User Guide*.) A single list of values can have multiple prompt groups. Each distinct prompt group results in a different presentation of the same underlying list of values.

9.7.4.1 To create a prompt group in Business View Manager

1. Open a sample Business Element or Data Foundation.

Sample data is located in the repository under \Samples\Business Views\Xtreme.

2. In the Object Explorer, select *Parameters*.
3. On the *Insert* menu, click *Insert Parameter*.

Note

If you are inserting a parameter into a Data Foundation that is based on a dynamic data connection, you are prompted to choose your parameter values.

The Create New Parameter dialog box appears.

4. Enter a name for the parameter in the *Name* field.

This example uses SupplierCity.

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."

7. Click *Select Business View*.

The Select Business View dialog appears.

8. Select the Business View that you will use for your list of values, and then click *OK*.

This example uses Xtreme Business View.

9.  Click *Insert*, and from the *Value* list, select *Country*.
10. Click the blank field under *Country* and select *Region*, and then click the blank field under *Region* and select *City*.
11. In the Parameter Binding area, ensure that only the *City* value is bound.
12. Click *OK*.

Save the prompt group to a folder in the repository.

13. Return to the Object Explorer dialog box, On the *Insert* menu, click *Insert Parameter*.
14. Enter a name for the second parameter in the *Name* field.

This example uses CustomerCity.

15. In the List of Values area, click *Dynamic*.

16. 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."

17. Select *Existing*, and select the list of values that you created in steps 8 through 11.

18. Click *OK*.

The Business View Manager creates the prompt group and saves it in the top-level folder of the repository.

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.

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.

9.7.4.2 To edit an existing Prompt Group

In the Repository Explorer, right-click on the prompt group and click *Edit Prompt Group*.

9.8 Best practices for prompting

Managed reports are reports that you store within SAP BusinessObjects BI platform.

Lists of values that are 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.
- 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 that you need to create to support prompting.

9.9 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 SAP BusinessObjects BI platform with 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.

9.10 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 with 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. SAP BusinessObjects BI platform tracks the objects that a report depends on, and imports these objects to the destination system if needed.

Business View Manager has been enhanced to support the import and export of lists of values and prompt groups. For more information on how to import and export lists of values and prompt groups, see [Exporting and importing Business Views \[page 59\]](#).

Note

When you export a list of values, all dependent objects are also exported, which includes all prompt groups, Business Views, Business Elements, Data Foundations, and Data Connections.

10 Managing Business Views

This section provides information on how to create and modify Business Views.

10.1 Business Views overview

A Business View is a logical collection of Business Elements. It provides the highest level of data abstraction for end users. Users see Business Views as abstract database connections, and the contained Business Elements as virtual tables that, in turn, contain Business Fields. End users access Business Views through client applications such as Crystal Reports or through the Report Application Server. Administrators can secure Business Views through the standard View and Edit rights.

10.2 Working with Business Views

When you create a Business View, you organize a collection of Business Elements and set user and group rights as required for report designers who access this abstracted data.

10.2.1 To create a Business View

When you create a new Business View, you choose the Business Element(s) to include in the Business View. After you select all of the Business Elements that you want to use, you can set user and group rights on the Business View for Business Views users and report designers.

1. On the *File* menu, select *New*, and then select *Business View*.
2. In the Object Explorer, right-click the *Business Elements* node, and click *Insert Business Elements*.

The "Insert Business Elements" dialog box appears.

3. Expand the folders as necessary and select a Business Element.
4. Click *Add*.
5. Continue selecting and adding Business Elements as required.
6. Click *Close* to continue.

10.2.2 Importing and exporting Business Views

After you create and save a Business View, you can export and import the Business View and its associated objects (that is, the Data Connections, Dynamic Data Connections, Data Foundation, Business Elements, and

any other objects that the Business View references). Business Views are exported as an XML file. During export, you can choose whether or not to include all of the security settings for your Business Views objects. For more information on the export and import tool, see [Exporting and importing Business Views \[page 59\]](#).

10.2.3 Showing dependent and referenced objects

You can view the Business Views objects that depend on your Business View. Similarly, you can view the Business Views objects that are referenced by your Business View.

To view dependent or referenced objects, in the Tools menu, select either Show Dependent Objects or Show Referenced Objects. A dialog box appears; in this dialog box, click the Save To File button to save the list of dependent or referenced objects to a text file for future reference.

The list of dependent objects displays objects that are affected by the Business View. (Since a Business View is the final object you must create, the other objects are not affected by the settings found in the Business View.) The list of referenced objects displays objects that the Business View references (Data Connections, Dynamic Data Connections, Data Foundations, and Business Elements). Expand the nodes to see all of the objects that depend on or are referenced your Business View.

10.2.4 Modifying a Business View

The main Business View window displays a list of all of the inserted Business Elements, along with the following information: author, name of the Data Foundation, and a description for each Business Element.

When you double-click a Business Element in the Object Explorer, the Business Element appears as a tab on the main Business View window. You can also right-click a Business Element and select Show Business Element Detail to display the details for that particular Business Element. On this tab, the different Business Fields for the Business Element are displayed. Also, the Property Browser displays the Name, Description, Author, and name of the Data Foundation for the selected Business Element.

10.2.4.1 To insert a Business Element

You can insert additional Business Elements into a Business View and then set the rights for users and groups for these Business Elements.

1. On the *Insert* menu, click *Insert Business Elements*.

→ Tip



Alternatively, in the Object Explorer, right-click Business Elements and select Insert Business Elements. You can also click the Insert Business Elements button on the toolbar.

→ Tip

You can also directly insert a Business Element by dragging a Business Element from the Repository Explorer and dropping it to the main window.

The "Insert Business Elements" dialog box appears.

2. Expand the folders as necessary and select a Business Element.
3. Click [Add](#).
4. Continue selecting and adding Business Elements as required.
5. Click [Close](#) to continue.

10.2.4.2 Using the Property Browser

The Property Browser displays all of the properties that you can edit or change for your Business View that you have selected in the Object Explorer.

- [Name](#)
The name of the Business View.
- [Description](#)
A description you enter for the Business View. This description appears when you point your mouse over the specific Business View in the Repository Explorer or when you have to choose a Business View. This information also appears when you right-click a Business View in the repository and select Properties.
- [Author](#)
By default, this field contains the name of the user who created the Business View. The name of the author appears when you point your mouse over the specific Business View in the Repository Explorer or when you have to choose a Business View.
- [Parent Folder](#)
The repository folder in which the Business View is located. This is the only property that cannot be directly modified through the Property Browser.
- [Business Element Filter Combination](#)
In a Business View, you can establish the type of relationship that occurs in the filters of the Business Elements. Using AND allows you to combine all of the conditions established by each filter, limiting the scope of the data made available in the end. Using OR produces a result that is a combination of all the outcomes that are generated by each filter.
- [Rights](#)
The user and group rights that are associated with the Business View. To update the rights, ensure that the Rights cell is selected and then click the ... button. When you set these rights, you are able to specify whether certain users or groups can or cannot have access to the Business View when they are creating a report. For more information on editing rights, see [Editing rights for a Business View \[page 183\]](#).

You can also view some of the Business View's properties by right-clicking a Business View in the Repository Explorer and selecting Properties. The Properties dialog box displays the name of the object, the type of object, and the date the object was last saved. This dialog box also displays the object's description.

10.2.4.3 Using the Object Explorer

The Object Explorer displays the Business View and Business Elements that are associated with the Business View. By right-clicking on any of these objects, you can perform different functions. Note that the shortcut menus are context-sensitive; some of the features are only available to certain objects.

- [Insert Business Elements](#)
You can insert additional Business Elements into your Business View.
- [Delete](#)
Select Delete to delete the object.
- [Show Business Element Detail](#)
You can view the details for each of the Business Fields that make up the Business Element.
- [Rights Test View](#)
Select Rights Test View to verify the Business View rights. For more information on viewing the set rights, see [Verifying Business View rights \[page 185\]](#).
- [Edit Rights](#)
You can edit user and group rights for the Business View object. For more information on rights, see [Editing rights for a Business View \[page 183\]](#).

10.2.5 Overriding the linking in a Data Foundation

You link tables in a Data Foundation so that records from one table will match related records from another. For example, if you add an Orders table and a Customers table, you link the two tables so that each order (from the Orders table) can be matched up with the customer (from the Customer table) who made the order.

You typically link your tables at the Data Foundation level; however, you can also override the linking through a Business View. In this case, the overridden link information is stored in the Business View. When you select Override Linking from the Linking Diagram menu, all of the current links for your existing tables are displayed in the Link Override tab (on the main Business View window).

In Data Foundations, once the tables (or table structures) have been added, any links and joins between those tables are defined. At times, when you are reporting off your data source, alternative joins between the tables may be desired.

For example, a software company may have a Business View on reported software bugs. In this Business View, the company has two tables: "Employee" and "Software Bugs." In the example, the "Employee" table has records for all of the staff for the company, including Developers, QA Testers, Documentation specialists, Managers, Finance Staff, and so on.

In a reporting view from the perspective of the Developer tracking the bugs they are fixing, the join between the two tables is from Employees.Employee ID to Software Bugs.Developer ID.

However, in a different Business View, you may want to view the bugs from the perspective of the QA Tester that identifies if the bugs are fixed. To do so, you can simply create a second Business View off the same Data Foundation. In this second Business View, you override the link between the tables so that it now goes from Employees.Employee ID to Software Bugs.QA Tester ID.

Thus, the Override Linking feature removes the need for users to add the same table to a Data Foundation multiple times and then create the same element multiple times as well.

Note

Before you can override the linking in your Business View, you must enable "Allow Table Joins Override" in the Data Foundation. For more information on this property, see [Using the Property Browser \[page 139\]](#).

Note

When you delete tables from the Data Foundation and you save the Data Foundation, the Business View Manager removes the links at the Business View level that are related to this deleted table.

Note

Initially, the Data Foundation tables and table joins required by the Business View are pre-populated. If a database table is added because a new Business Element is added, the table is shown in the Link Override tab, but the users have to manually define the links. If a database table is not needed anymore (because a Business Element is removed), then the table is removed from the Link Override tab, along with the links related to the table. In summary, even if the Data Foundation may contain more tables, the Link Override tab only shows the tables required by the current Business View. Users cannot select visible tables as they normally can in the Data Foundation Linking Diagram.

For information on linking as it relates to a Data Foundation, see [Linking tables \[page 100\]](#). For more information on linking tables, see the "Linking tables" section, located in the "Understanding Databases" section of the *Crystal Reports Online Help*.

10.2.5.1 Override Linking

You must first select Override Linking before you can update the table links at the Business View level.

To select Override Linking, on the Linking Diagram menu, select Override Linking.

Note

Before you can override the linking in your Business View, you must enable "Allow Table Joins Override" in the Data Foundation. For more information on this property, see [Using the Property Browser \[page 139\]](#).

10.2.5.2 Revert Linking

To revert the linking back to the table link settings found in the Data Foundation, select this option. You are asked to confirm that you want to revert the links. This option permanently removes all overridden table joins that you specified for the Business View.

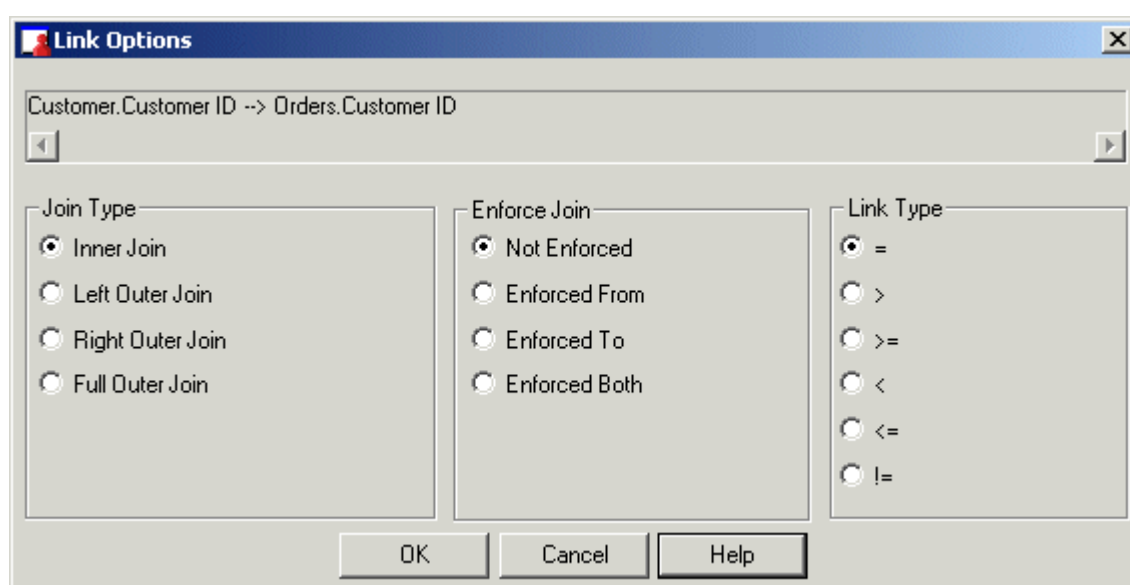
To select Revert Linking, on the Linking Diagram menu, select Revert Linking.

10.2.5.3 Modifying a link

You can modify a link by right-clicking it. To ensure that you are modifying a specific link, first click the link to select it—the selected link will be displayed as a blue link. Then, right-click the selected link and choose from the following:

Link Options

The Link Options dialog box enables you to specify join and link types.



- [Link display](#)
This box displays the selected link. It also shows the "from" table—used as a primary table—and the "to" table—which acts as a lookup table where records are looked up by the primary table. For more information on links, see [Link from and link to \[page 101\]](#).
- [Join Type](#)
Business Views enables you to specify the type of join you want it to use when joining tables. The join type expresses the nature of the relationship between two tables. For more information on the specific types of joins available, see [Join types \[page 101\]](#).
- [Enforce Join](#)
Business Views enables you to enforce the use of tables while you're specifying joins. For more information on the specific types of enforced joins available, see [Enforced joins \[page 102\]](#).
- [Link Type](#)
Business Views enables you to specify the type of link you want it to use when joining tables. The link type expresses the nature of the relationship between two fields. For more information on the specific types of links available, see [Link types \[page 104\]](#).

Delete Link

Right-click the link and select Delete Link to remove it.

Reverse Link

Right-click the link and select Reverse Link to reverse the "from" and "to" tables.

Remove All Links

Right-click the link and select Remove All Links to remove all of the links that are set in the "from" table. Note that this action removes the links at the Business View level only; it does not remove the links at the Data Foundation level.

10.2.5.4 Smart Linking By Key

When you select Smart Linking By Key, you are automatically linking tables by their foreign key information. Business Views scans through the data tables and links the fields with foreign keys that match.

To select Smart Linking By Key, on the Linking Diagram menu, select Smart Linking By Key. Alternatively, on the Linking Diagram tab in the main Business View window, right-click anything but an existing link and select Smart Linking By Key.

10.2.5.5 Smart Linking By Name

When you select Smart Linking By Name, you are automatically linking tables by their names. Business Views scans through the data tables and links the fields with names that match.

To select Smart Linking By Name, on the Linking Diagram menu, select Smart Linking By Name. Alternatively, on the Linking Diagram tab in the main Business View window, right-click anything but an existing link and select Smart Linking By Name.

10.2.5.6 Clear Links

When you select Clear Links, all existing links will be cleared. A dialog box prompts you to confirm your Clear Links action.

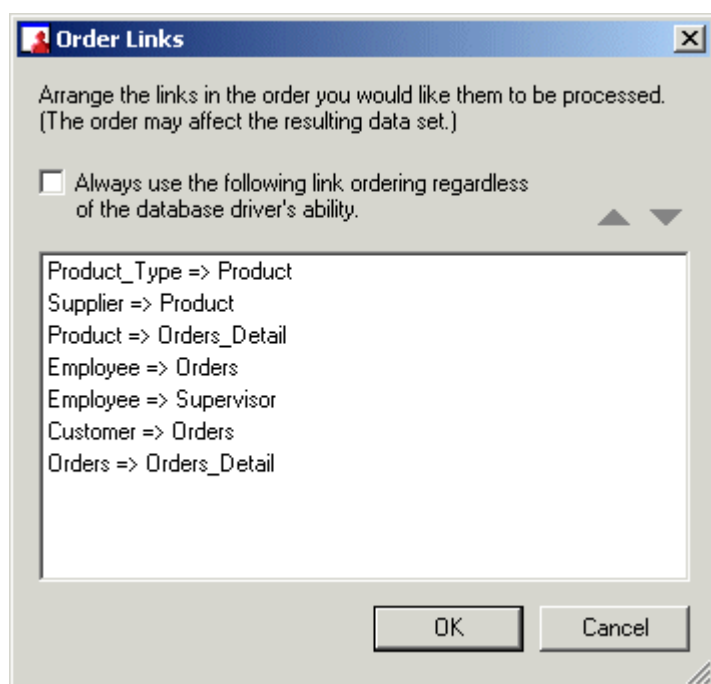
To select Clear Links, on the Linking Diagram menu, select Clear Links. Alternatively, on the Linking Diagram tab in the main Business View window, right-click anything but an existing link and select Clear Links.

10.2.5.7 Order Links

When you select Order Links, the Order Links dialog box appears. Using Order Links, establish the link processing order you want for the available linked tables.

To select Order Links, on the Linking Diagram menu, select Order Links. Alternatively, on the Linking Diagram tab in the main Business View window, right-click anything but an existing link and select Order Links.

The Order Links dialog box appears.



Use the Order Links dialog box to specify the order in which Business Views should process the table links. By default, the processing order matches the order in which the links appear on the Linking Diagram tab in the main Data Foundation window. Use the arrow buttons in the Links Order dialog box to change the default link order.

To enforce link ordering, select the "Always use the following link ordering regardless of the database driver's ability" check box.

Note

Different link orders may result in different data sets that are returned for use in your report.

10.2.5.8 Locate Table

When you select Locate Table, the Locate Table dialog box appears. This dialog box lists all of the tables displayed on the Linking Diagram tab of the main Business View window. This function is useful when you have

a number of tables and need to quickly locate a specific table. Select a table from the list and click the Done button. The Linking Diagram tab displays the selected table.



To select Locate Table, on the Linking Diagram menu, select Locate Table. Alternatively, on the Linking Diagram tab in the main Business View window, right-click anything but an existing link and select Locate Table. You can also click the Locate Table button on the toolbar.

10.2.5.9 Rearrange Tables

When you select Rearrange Tables, the data tables are rearranged by existing links.



To select Rearrange Tables, on the Linking Diagram menu, select Rearrange Tables. Alternatively, on the Linking Diagram tab in the main Business View window, right-click anything but an existing link and select Rearrange Tables. You can also click the Rearrange Tables button on the toolbar.

10.2.5.10 Fetch Table Indexes

When you select Fetch Table Indexes, the Business View Manager retrieves the indexes for the tables. You then select [Index Legend \[page 182\]](#) to view the key to the index indicators.

10.2.5.11 Index Legend

When you select Index Legend, the Index Legend dialog box appears; this dialog box displays a key to the index indicators that are used in the visible tables on the Linking Diagram tab area.

To select Index Legend, on the Linking Diagram menu, select Index Legend. Alternatively, on the Linking Diagram tab in the main Business View window, right-click anything but an existing link and select Index Legend.

10.2.5.12 Change Linking View

When you select Change Linking View, the view in the Link Override tab changes to display only the names of the tables.



To select Change Linking View, on the Linking Diagram menu, select Change Linking View. Alternatively, on the Linking Diagram tab in the main Business View window, right-click anything but an existing link and select Change Linking View. You can also click the Change Linking View button on the toolbar.

10.2.6 To save a Business View

Business Views, like all of the other Business Views objects, are saved to the repository. For more information on the repository, see [Using the Repository Explorer \[page 62\]](#).

When you first save a Business View, you need to specify a name for the object and the location you want to save the object to.

1. On the [File](#) menu, select [Save](#).

→ Tip



You can also click the Save (Ctrl+S) button or press CTRL+S.

The Save As dialog box appears.

2. In the [Object Name](#) field, specify the name of your object.
3. Select the folder you want to save your object to.

→ Tip



You can create new folders in the repository by clicking the "Insert a new folder" button.

4. Click [Save](#).

10.2.7 Editing rights for a Business View

You set user and group rights for a Business View to ensure that only specific users and groups can access the Business View. Since the Business Views security model is based on the SAP BusinessObjects BI platform security model, you have a flexible security model that you can work from. For example, if you explicitly deny a View right for a user or group for a specific Business View, this user or group, when using the Business View Manager, will not be able to preview the report.

ⓘ Note

Before you can edit the security rights for a Business View, you must first save the Business View to the repository.

Business View object rights are also dependent on folder rights. If you set a right on a folder, all of the Business View objects in this folder inherit the same security rights. For more information on the inheritance model as it pertains to rights, see [Using inheritance to your advantage \[page 192\]](#).

Before you can apply security rights to an object, you must have Set Security rights for the object. By default, all users of the Administrators group have full access to all of the folders and objects in the repository.

ⓘ Note

For detailed information on the Edit Rights dialog box, see [Using the Edit Rights dialog box \[page 188\]](#).

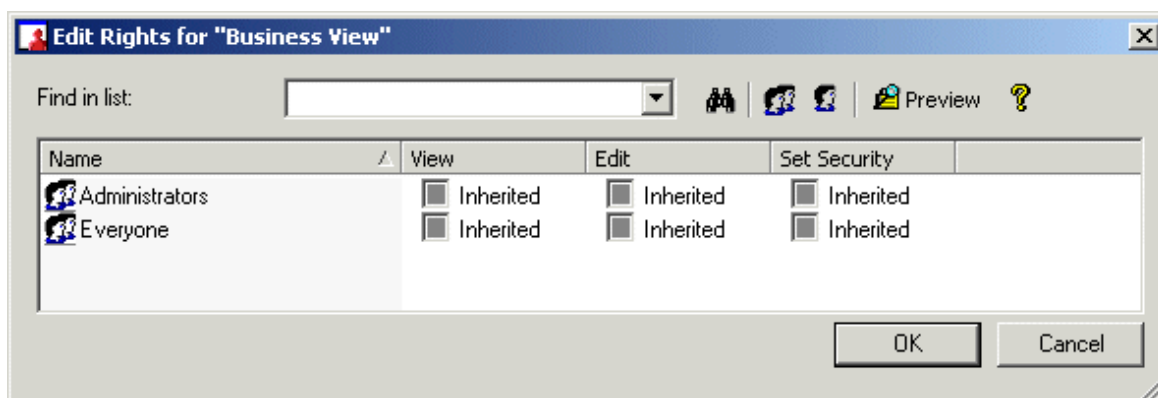
10.2.7.1 To apply security settings to a Business View

1. In the Object Explorer, right-click the Business View and select *Edit Rights*.

→ Tip

You can also click the ... button in the Rights row in the Property Browser. Alternatively, you can select Edit Rights on the Edit menu.

The Edit Rights dialog box appears.

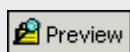


2. Set the following rights for either a user or a group by clicking the appropriate check box under each rights column.
 - View
This right specifies whether a user or group can or cannot view the Business View. Also, for a report based on the Business View, this right specifies whether a user can or cannot preview the report.
 - Edit
This right specifies whether a user or group can or cannot edit the properties of the Business View.
 - Set Security
This right specifies whether a user or group can or cannot modify the rights that are associated with the Business View.



Note

A cleared check box indicates that the user or group is denied the right; a check mark in the check box indicates that the user or group is granted the right; a shaded check box indicates that the right is inherited. Also, note that a right may be shown as being granted, but may be denied once inheritance is taken into account.

Note



Click the Preview button to see the final security settings for the object with inheritance in effect. The Business View Manager resolves the net result of the current security settings and displays the net rights.

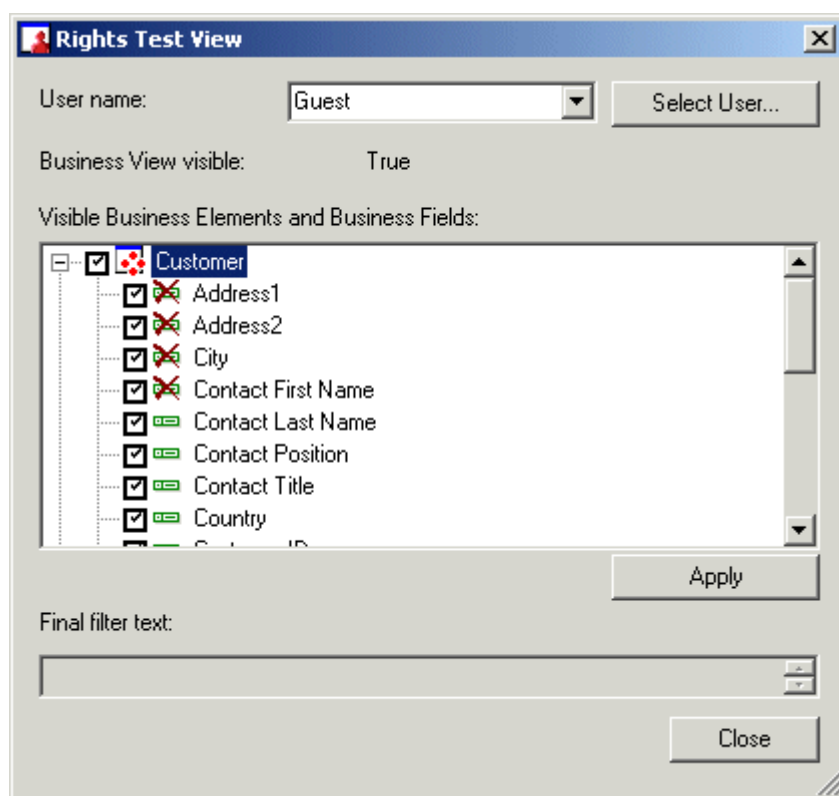
3.   To add another user or group, click the [Add Users](#) button or click the [Add Groups](#) button.
4. Assign rights to the new user or group as needed.
5. Click [OK](#).

10.2.7.2 Verifying Business View rights

The Rights Test View feature enables you to verify the Business View rights for a user or group. This feature is useful when you want to check whether a certain user or group can or cannot view a Business View, Business Elements, and Business Fields. You might want to verify users' rights to ensure that only specific users will have access to specific Business Views when they create reports based off of these Business Views.



To verify the Business View rights, on the Tools menu, select Rights Test View. (You can also click the Rights Test View button on the toolbar.)



In the Rights Test View dialog box, you select a user from the User name list. Click the Select User button to search for a user in the Add User dialog box. When you find the user whose rights you want to verify, click the Add button.

The Rights Test View dialog box specifies whether the current Business View is visible to the selected user or not. It also displays all of the visible Business Elements and Business Fields for the selected user. Additionally, for administrative purposes, it indicates any restricted Business Fields (that is, any field with column data access filters applied) with a red cross above the object's icon.

Beside each node in the "Visible Business Elements and Business Fields" area is a check box. Use this check box to specify the Business Elements and/or Business Fields that you want to include in the computation for the filter text information, then click Apply. The Rights Test View dialog box displays the filter text information in the "Final filter text" area.

11 Business Views Security Concepts

This section provides information on general and advanced security concepts and object security for Business Views. It also provides deployment suggestions and considerations regarding security settings.

11.1 Security overview

The Business Views architecture addresses the many security concerns that affect today's businesses and organizations with regards to securing data access from disparate data sources. Using the Business View Manager, administrators can design relational views of information. This designer also enables administrators to set detailed column and row-level security for various objects.

This section provides detailed conceptual and procedural information on both general and advanced security concepts and object security for Business Views. It also provides general deployment suggestions and considerations regarding security settings.

- For procedural information on using the Business View Manager to apply security settings, see [Using the Business View Manager \[page 54\]](#).
- For more information on row and column security, see [Security applications \[page 199\]](#).
- For information on SAP BusinessObjects BI platform security as it applies to the repository, see [SAP BusinessObjects repository security model \[page 67\]](#).
- For details on the SAP BusinessObjects BI platform security model, see the *SAP BusinessObjects Business Intelligence platform Administrator Guide*.

11.2 Business Views object rights overview

Object rights are the base units for controlling users' access to folders and other objects within the Business View Manager. When granted, each right provides a user or group with permission to perform a particular action on an object. For any object, you can set security levels that affect individual users or entire groups.

To set object rights within the Business View Manager, you first locate the object, and then you specify the rights for different users and groups who use that object. You specify folder rights through the Repository Explorer and you specify Business Views object rights either through the Repository Explorer as well, or by the Rights field in the Property Browser for the specific object. You can also set rights by selecting Edit Rights on the Edit menu.

❗ Note

This section provides information on rights as they pertain to Business Views objects. For information on setting folder rights for users and groups, see [Applying security settings to folders in the repository \[page 68\]](#).

Each Business View's right can be explicitly granted or denied (with the exception of the Data Access right for filters, which only have the option of being explicitly granted or remaining unspecified). The SAP BusinessObjects BI platform object security model is designed such that, if a right is left "not specified," the right is denied by default. Additionally, if contradictory settings result in a right being both granted and denied to a user or group, the right is denied. Inherited rights follow this model as well; if an inherited right is denied and granted in two different places, the right is denied. This "denial based" design helps to ensure that users and groups do not automatically acquire rights that are not explicitly granted (or rights that are denied).

Since the SAP BusinessObjects BI platform object security model is based on an inheritance structure, you can set rights through group membership. Thus, subgroups will inherit the rights of the group. The same model is applied to folders and subfolders.

Note

Grant and set rights for Business Views objects through the Business View Manager. For SAP BusinessObjects BI platform security tasks (setting up users, groups, and so on), use the Central Management Console. For details on managing users and groups, see the *SAP BusinessObjects Business Intelligence platform Administrator Guide*.

11.3 Using the Edit Rights dialog box

Through the Edit Rights dialog box, you can set user and group rights on a specific object. This section explains how to use the Edit Rights dialog box and the Add Groups and Add Users dialog boxes.

Note

By default, the Administrator user has full rights to every object within the repository. Also, users in the Administrator group are granted full rights to every object, unless otherwise specified (explicitly denied, for example).

11.3.1 Applying security settings

Through the Business View Manager, you apply security settings to the following objects:

- Data Connections
- Dynamic Data Connections
- Data Foundations
- Business Elements
- Business Views

You can also apply security settings to the following repository objects:

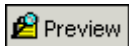
- Text objects
- Bitmaps
- Custom Functions

- Commands (queries)

In addition, you can also set the View field data right for a formula and/or SQL expression, Business Field, or a database field in a Data Foundation. You can also set the Apply Filter right for a filter.

Depending on the Business Views object, the Data Access right may or may not be available. For more specific information on security settings as it pertains to each object (Data Connection, Data Foundation, Business Element, and so on), refer to the relevant sections.

Before you edit the security rights for an object, you must first save the object to the repository. Also, because rights are inherited, note that any rights set at a folder or subfolder level is cascaded to the related subfolders.



Before you finish applying security settings to an object or folder, click the Preview button in the Edit Rights dialog box to ensure that the net rights (after inheritance) match your intended settings for the user or group. Explicit rights that are set in the Edit Rights dialog box can be overridden due to inheritance—thus, the Preview button enables you to preview the net rights.

→ Tip

Because Business Views uses the SAP BusinessObjects BI platform security model of inherited rights, when you set up your security settings, you might set the appropriate rights for users and groups at a folder level. Then, when you publish objects to the folder, the rights for users and groups will follow the settings found in the folder.

11.3.1.1 To apply security settings to a Business Views object

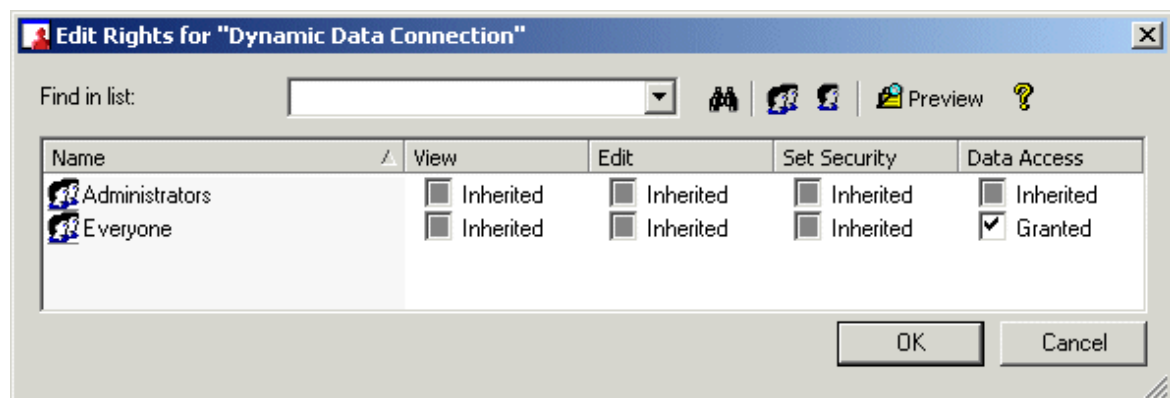
1. In the Object Explorer, right-click the object, and then select [Edit Rights](#).

You can also click the ... button in the Rights row in the Property Browser. Alternatively, you can select Edit Rights on the Edit menu.

→ Tip

If you want to apply security settings to a folder or subfolder, in the Repository Explorer, right-click the folder and select Edit Rights.

The Edit Rights dialog box appears.

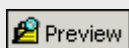


2. If you have a number of names in the [Name](#) column, enter a name in the [Find in list](#) field, and then click the [Search Groups and Users](#) button.
3. Set the following rights for either a user or a group by clicking the appropriate check box under each rights column.
 - [View](#)
This right specifies whether a user or group can or cannot view the object. Please note that while denying View rights prevents the user or group from viewing the object, the data is still potentially accessible. Data Access rights must be specified in order to prevent unwanted access.
 - [Edit](#)
This right specifies whether a user or group can or cannot edit the object.
 - [Set Security](#)
This right specifies whether a user or group can or cannot modify the rights that are associated with the object.
 - [Data Access](#)
This right specifies whether a user or group can access the specified data source in the object. Note that this right cannot be inherited from other rights: a user or group either explicitly has data access or does not.



Note

A cleared check box indicates that the user or group is denied the right; a check mark in the check box indicates that the user or group is granted the right; a shaded check box indicates that the right is inherited. Also, note that a right may be shown as being granted, but may be denied once inheritance is taken into account.

Note





Click the Preview button to see the final security settings for the object with inheritance in effect. The Business View Manager resolves the net result of the current security settings and displays the net rights.

4.   To add additional groups or users, click the [Add Groups](#) button or the [Add Users](#) button.
5. Assign rights to the new user or group as needed.
6. Click [OK](#).

11.3.2 To add and assign rights to a group or user

While in the Edit Rights dialog box, you can add additional groups and users, and then set rights for them.

1.   In the Edit Rights dialog box, click either the [Add Groups](#) button or the [Add Users](#) button.
Depending on which button you clicked, the Add Groups or Add Users dialog box appears.
2. Select a group from the Group Name column, or select a user from the Account Name column.

→ Tip

You can click the title bars of each column to sort your groups or users by name or description.

→ Tip

If you have numerous groups or users, click the First Page, Previous Page, Next Page, and Last Page buttons to navigate and find a specific group or user.

3. You can search for groups and users by selecting *Name* or *Description* in the *Search For* list.
4. In the adjacent list, select *is*, *is not*, *contains*, *does not contain*, or *starts with*; then type the search text in the *Text* field.
5. Click the *Search* button to search for the groups or users; click the *Clear Search* button to clear your search criteria.
6. Once you have selected a group or user to add to the Edit Rights dialog box, click *Add*.
7. Continue selecting groups and users, and then click the *Add* button; click the *Close* button once you are finished.

11.4 Object and folder rights

You set rights for Business Views objects and folders (and subfolders) through the Business View Manager. Because rights can also be inherited, changes to a folder with subfolders, for example, affect all of the subfolders below the folder.

ⓘ Note

Do not use the SAP BusinessObjects BI platform SDK to modify Business Views objects and folder rights.

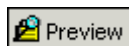
11.4.1 Viewing Business Views object rights

If a user or group does not have View rights to an object, the user or users in that group cannot view the object in the Business View Manager, nor can they view the object's security settings. Thus, if a user or group does not have View rights to an object, they will not be able to see the object at design time. From these users' perspective, they will not know that the object exists.

At runtime, if a user or group does not have View rights to a Business View, they will not see the Business View. However, even if they do not have View rights for a Business Element, Data Foundation, or Data Connection that the Business View references, as long as they have View rights for the Business View, they will be able to see the Business Element, Data Foundation, and Data Connection that are used by the Business View. Note that the user or group can only see objects that are referenced by the Business View.

For example, employees can create and view reports with salary information, even though they cannot view the salary data. The manager, running the same report (with different View rights), is able to view the salary data.

To view a folder's rights settings, right-click the object in the Repository Explorer and select Edit Rights. The Edit Rights dialog box shows only the specified rights (for example, Granted, Denied, and Inherited) before any inheritance settings are resolved.



In the Edit Rights dialog box, click the Preview button. By clicking the Preview button, the Business View Manager resolves any inheritance settings; it determines the actual net results of the rights that have been set for each user and group:

- Granted (Explicit)
- Granted (Inherited)
- Denied (Explicit)
- Denied (Inherited)
- Denied (Never Specified)

11.4.2 Using inheritance to your advantage

Regarding object rights, Business Views recognizes two types of inheritance: group inheritance and folder inheritance. By taking advantage of the ways in which object rights are inherited, you can reduce the amount of time it takes to secure the content that you have saved using the Business View Manager. Since Business Views uses the same security model as the one found in SAP BusinessObjects BI platform, you can set up SAP BusinessObjects BI platform to integrate new users and new content quickly and easily. For more information on the SAP BusinessObjects BI platform security model, refer to the *SAP BusinessObjects Business Intelligence platform Administrator Guide*.

→ Tip

By default, users or groups who have rights to a folder will inherit the same rights for any Business Views object that you subsequently publish to that folder, regardless of whether these users or groups have been assigned rights to the object. Consequently, the best strategy is to set the appropriate rights for users and groups at the folder level first. Then publish objects to that folder.

11.4.2.1 Group inheritance

Group inheritance enables users to inherit rights as the result of group membership. Group inheritance is especially useful when you organize all of your users into groups that coincide with your organization's current security conventions. For example, if you create a user called Sample User and add it to an existing group called Sales, then Sample User automatically inherits the appropriate rights for each of the folders and Business Views objects that the Sales group has been added to.

When group inheritance is enabled for a user who belongs to more than one group, the rights of both groups are considered when the system checks credentials. The user is denied any right that is explicitly denied in any group, and the user is denied any right that remains completely "not specified"; thus, the user is granted only those rights that are granted in one or more groups and never explicitly denied.

11.4.2.2 Folder inheritance

Folder inheritance allows users to inherit any rights that have been granted from an object's parent folder. Folder inheritance proves especially useful when you organize Business Views content into a folder hierarchy that reflects your organization's current security conventions. For example, suppose that you create a folder called Sales Reports, and you provide your Sales group with View access to this folder. By default, every user that has rights to the Sales Reports folder will inherit the same rights to the Business Views objects that you subsequently save to this folder in the repository. Consequently, the Sales group will have View access to all of the Business Views objects, and you need only set the object rights once, at the folder level.

Note

If you need to disable inheritance for a particular folder or object within your folder hierarchy, you can do so by explicitly granting or denying access to the folder or object.

11.5 Security deployment

The Business Views security model is flexible in that you can deny and grant rights at a granular level (to specific users) or at a general level (granting or denying rights to a group or folder) as necessary.

In your Business Views and SAP BusinessObjects BI platform deployment, you might have the following groups:

- **Business View administrators**
This group consists of administrators who manage Business Views through the Business View Manager.
- **Business View designers**
This group of users creates Business Views for report designers to use through a report client such as Crystal Reports. These users likely have database skills and are also familiar with Business Views.
- **Report designers**
Report designers create reports through a client such as Crystal Reports. These users may or may not be familiar with Business Views.
- **Report viewers**
These end users view the reports that report designers create.

These four groups must be granted Data Access rights (for a Data Connection or a Dynamic Data Connection).

For a Data Connection, the following access rights are recommended:

Group	View	Edit	Set Security
Business View administrators	Granted	Granted	Granted
Business View designers	Granted	Granted	Denied
Report designers	Denied	Denied	Denied

Group	View	Edit	Set Security
Report viewers	Denied	Denied	Denied

For a Data Foundation, the following access rights are recommended:

Group	View	Edit	Set Security
Business View administrators	Granted	Granted	Granted
Business View designers	Granted	Granted or Denied	Denied
Report designers	Granted or Denied (granted if designers need access to SQL queries)	Denied	Denied
Report viewers	Denied	Denied	Denied

For a Business Element, the following access rights are recommended:

Group	View	Edit	Set Security
Business View administrators	Granted	Granted	Granted
Business View designers	Granted	Granted	Denied
Report designers	Denied	Denied	Denied
Report viewers	Denied	Denied	Denied

For a Business View, the following access rights are recommended:

Group	View	Edit	Set Security
Business View administrators	Granted	Granted	Granted
Business View designers	Granted	Granted	Denied
Report designers	Granted	Denied	Denied
Report viewers	Granted	Denied	Denied

Note that report designers only need View rights on a Business Views object and Data Access rights to a Data Connection or Dynamic Data Connection. They do not need view rights for any other object in order to create reports that are based on a Business View.

11.6 Security considerations

This section covers security considerations for your Business Views deployment. It includes information about:

- [Root folder \[page 195\]](#)
- [Reports based on Business Views \[page 195\]](#)
- [Importing and exporting Business Views \[page 196\]](#)

11.6.1 Root folder

By default, the Everyone group is granted the View right for the repository root folder, and any Edit and Set Security rights that it has inherited.

If you change the View right to inherited, the right will be implicitly denied for every user unless they are explicitly granted the right elsewhere. If you deny View rights at the root folder for the Everyone group, all users are locked out of the Business Views system. But, as a security precaution, the Administrator user can still access the Business Views system and change rights.

11.6.2 Reports based on Business Views

After a report designer creates a report that is based on a Business View, the report designer can use the option called "Disconnect view security" (in the Save As dialog box). Selecting this option disconnects the report from the Central Management Server (CMS) so that users are not prompted for logon when they open the report. Once a report is disconnected from the CMS, you cannot reconnect to SAP BusinessObjects BI platform and its security features, nor can you refresh the report's data. Thus, this option is useful for when you might want to share data that is normally secured, but you don't want those who receive the report to be able to refresh the data later.

When you save a Crystal report, if users who view the report cannot connect to SAP BusinessObjects BI platform, they cannot view the report or refresh the data. Note that once users access a report, they can export the report (and the data they have access to) to a different format. When another user views the report in the exported format, security is no longer applied to the report.

It is the responsibility of the System Administrator, or the report designer who has Administrator rights, to save reports that are based on Business Views to a secure location. If the Administrator creates a report that is based on a Business View that has row- or column-level security, saves the report with saved data to a non-secure location, and then disconnects from SAP BusinessObjects BI platform, all users will have access to the report and its data.

11.6.3 Importing and exporting Business Views

When you import and export Business Views, regardless of whether or not you export security information, Data Connection passwords are not exported (this prevents users from deciphering the password in the XML file).

When you import a Business View with saved security information, inheritance still overrides the rights set at the Business View level.

Note that if your Business View includes a custom function, the custom function can only be imported to the Custom Functions folder in the repository. All custom functions must be saved to that folder.

For more information on importing and exporting Business Views, see [Exporting and importing Business Views \[page 59\]](#).

12 User Scenarios

This section details various user scenarios that demonstrate the features found in Business Views. These scenarios provide deployment guidelines and suggestions for managing your data.

12.1 User scenarios overview

This section provides deployment suggestions and guidelines on various user scenarios that demonstrate the features found in Business Views.

User scenarios are broken up into three categories. Each category contains either one or more user scenarios:

- [Data integration \[page 197\]](#)
 - [Simplifying existing infrastructure through data abstraction \[page 198\]](#)
 - [Multiple data sources \[page 198\]](#)
- [Multiple data sources \[page 198\]](#)
 - [Reporting off multiple data sources \[page 198\]](#)
 - [Specifying locale: a global, multi-lingual deployment \[page 199\]](#)
- [Security applications \[page 199\]](#)
 - [Applying row security \[page 199\]](#)
 - [Applying column security \[page 200\]](#)
 - [Setting object security \[page 200\]](#)
- This guide includes a tutorial on using Business Views. For more information on the tutorial, see [Quick Start overview \[page 23\]](#).
- For information on using the Business View Manager, see [Using the Business View Manager \[page 54\]](#)
- For details on the Business Views security model and security features, see [Business Views Security Concepts \[page 187\]](#).
- For details on scalability for SAP BusinessObjects BI platform, see the *SAP BusinessObjects Business Intelligence Administrator Guide*.

12.2 Data integration

This section provides recommendations on using Business Views to simplify data access for your end users and report designers. Business Views enables you to integrate your data from disparate sources. You can bring together data from multiple data collection platforms across application boundaries so that the differences in data resolution, coverage, and structure between collection methods are eliminated. In addition, you can add the necessary business context to these disparate data islands to provide consistent definitions of corporate hierarchy or customer information, and to provide a variety of detailed or summarized viewpoints for the various information consumers in your business.

Data integration enables you to simplify and consolidate your data so that end users and report designers see and access all of the necessary data from one source, even though the tables and fields may be residing on multiple data sources or machines. Business Views provides a central management point for data access.

12.2.1 Simplifying existing infrastructure through data abstraction

In this scenario, you have an existing data infrastructure that is extremely complex, and thus, difficult to manage. Users are unable to find the data they need, and when they do find it, they do not necessarily understand how to use this data. For instance, users may have a hard time locating data sources, as these sources may be hidden on the network or be on various local drives. Another instance is that field names may be cryptic or not obvious, and this problem is compounded when there are numerous field names in thousands of tables.

You can use Business Views to provide a central management point for data access.

You can use the abstraction capabilities of Business View Manager to join tables, alias fields, group information by related business units, and so on. In addition, you can resolve the issue in which there may be multiple definitions of a "customer" field from multiple data sources, so that users are not constrained in the flexibility they have due to existing data sources and their naming conventions. Thus, numerous fields can be labelled as the "customer" field, but these fields are given different aliases so that each field is clear and easily accessible to the end user or report designer.

For more information on table joins, see [Linking tables \[page 100\]](#). For information on alias fields, see [Modifying a Business Element \[page 146\]](#).

12.3 Multiple data sources

This section provides an overview of different scenarios in which an organization uses multiple data sources. Business Views enables users to report off disparate data sources without having to redesign new reports for each data source, or having to change the database settings for each report.

12.3.1 Reporting off multiple data sources

A typical scenario—for IT departments, web site designers, research and development departments, and so on—involves multiple data sources: a development system, a test system, and a production system.

For example, before rolling out a new system, a company will populate a development system with data and create applications that report off the data. When development is complete, the data is migrated to a test system, where a test deployment (for example, a test web site) is run against the test system's data. When the data and its related applications are ready to be put into production, the production database is populated with the relevant data.

Previously, report designers would have to redesign their reports and test these reports against the production database, or they would have to change the database settings in their reports. Business Views provides a quick

and simple means for the same reports that are run against the development system to be run later against the test system, and then run against the production system. All a report designer or administrator has to do is to change the Dynamic Data Connection to point from one Data Connection (which specifies the database setting) to another. Users refresh their reports in order to access the latest data from the specified database or data source.

12.3.2 Specifying locale: a global, multi-lingual deployment

Through the use of Dynamic Data Connections, administrators can handle a global, multi-lingual deployment without creating numerous reports for different locales. The Unicode features of Crystal Reports and SAP BusinessObjects BI platform are useful in enabling reports and systems to contain data in multiple code pages and languages, but this solution does not solve the issue of the displayed user interface (UI), report labels, and other information not being in a user's preferred language. By dynamically routing users to the correct language database and allowing the aliasing of field names, Business Views enables Crystal Reports to render the UI and other report objects and details in the correct language. Thus, the same report will be displayed in different languages, depending on the user—or on the location of the user—who accesses the report.

It is important to note that Data Connections that compose a Dynamic Data Connection must have the same schema. That is, the data sources for a Dynamic Data Connection must have the same structure. Also, any tables, fields, stored procedures, stored procedure parameters, and so on that are in the data sources must have identical structures, names, and content types. Often cultural differences may result in different names for a data field. For example, currency, date, and punctuation conventions differ between languages and cultures. It is important to watch for these differences before creating Dynamic Data Connections.

12.4 Security applications

Besides providing for different data migration scenarios, one of main strengths of Business Views is that it provides a comprehensive security model for users and administrators. You can be as general or as specific as you like in creating your security model through the Business View Manager. Although the security model can be extremely complex, end users and report designers only see and have access to the information that you grant to them—if they do not have access to, for example, employee salary information, you can ensure that they never see or have access to this data. At the same time, other users with the appropriate rights can easily access employee salaries. Business Views supports row and column-based security, along with object security for Business Elements, Business Views, and so on. Administrators use the Business View Manager to specify security settings for both their end users and report designers.

12.4.1 Applying row security

This type of security application is conceptually similar to the security that is currently provided by processing extensions in SAP BusinessObjects BI platform. Administrators use row security to control access to data based on a user's attribute: for example, the geographical location of an employee, the seniority of an employee (whether an employee is a manager or not, for example), an employee's external group membership (NT group, LDAP group, and so on), or any other attribute.

By using filters, an administrator can ensure that the data that is returned to the user is restricted to only specific information that the user has access to, and thus, can view. Business Views supports multiple filters at the same level, and also composite filters at multiple levels.

12.4.2 Applying column security

Column security is typically implemented to filter which fields are returned to the report user. A typical example involves a Business Element that contains employee information. Managers may need access to all of the fields, but employees do and should not have access to the salary and bonus fields. By applying column security, administrators can have as much granular control over the fields as needed.

12.4.3 Setting object security

Object security is closely related to column security, but it is applied on objects such as Business Elements and Business Views. In this scenario, an administrator uses Business Elements and Business Views to group related content. For example, a human resources (HR) Business View may contain the following Business Elements: an employee Business Element, a compensation Business Element, and an employee leave or vacation Business Element. These objects are all generated from different underlying Data Foundations.

By applying object security, an administrator can allow all users to access the HR Business View, and, based on group membership, filter the amount of information that can be seen. For example, only managers can access and see the compensation data, while all employees have access to the employee data.

You can also edit object security for all of the other Business Views objects. For example, when you set the user and group rights for a Data Connection, you ensure that only specific users and groups can access the Data Connection object. Thus, when you explicitly deny a View right for a user or group for a specific Data Connection, this user or group, when creating a Dynamic Data Connection, or a Data Foundation, will not be able to view or select from any Data Connection that they do not have View rights to.



In addition, object rights are dependent on folder rights. If you set a right on a folder, all of the Business Views objects in this folder inherit the same security rights. For more information on the inheritance model as it pertains to security, see [Using inheritance to your advantage \[page 192\]](#).

Important Disclaimers and Legal Information

Hyperlinks

Some links are classified by an icon and/or a mouseover text. These links provide additional information.

About the icons:

- Links with the icon  : You are entering a Web site that is not hosted by SAP. By using such links, you agree (unless expressly stated otherwise in your agreements with SAP) to this:
 - The content of the linked-to site is not SAP documentation. You may not infer any product claims against SAP based on this information.
 - SAP does not agree or disagree with the content on the linked-to site, nor does SAP warrant the availability and correctness. SAP shall not be liable for any damages caused by the use of such content unless damages have been caused by SAP's gross negligence or willful misconduct.
- Links with the icon  : You are leaving the documentation for that particular SAP product or service and are entering an SAP-hosted Web site. By using such links, you agree that (unless expressly stated otherwise in your agreements with SAP) you may not infer any product claims against SAP based on this information.

Videos Hosted on External Platforms

Some videos may point to third-party video hosting platforms. SAP cannot guarantee the future availability of videos stored on these platforms. Furthermore, any advertisements or other content hosted on these platforms (for example, suggested videos or by navigating to other videos hosted on the same site), are not within the control or responsibility of SAP.

Beta and Other Experimental Features

Experimental features are not part of the officially delivered scope that SAP guarantees for future releases. This means that experimental features may be changed by SAP at any time for any reason without notice. Experimental features are not for productive use. You may not demonstrate, test, examine, evaluate or otherwise use the experimental features in a live operating environment or with data that has not been sufficiently backed up.

The purpose of experimental features is to get feedback early on, allowing customers and partners to influence the future product accordingly. By providing your feedback (e.g. in the SAP Community), you accept that intellectual property rights of the contributions or derivative works shall remain the exclusive property of SAP.

Example Code

Any software coding and/or code snippets are examples. They are not for productive use. The example code is only intended to better explain and visualize the syntax and phrasing rules. SAP does not warrant the correctness and completeness of the example code. SAP shall not be liable for errors or damages caused by the use of example code unless damages have been caused by SAP's gross negligence or willful misconduct.

Bias-Free Language

SAP supports a culture of diversity and inclusion. Whenever possible, we use unbiased language in our documentation to refer to people of all cultures, ethnicities, genders, and abilities.

© 2024 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company. The information contained herein may be changed without prior notice.

Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies.

Please see <https://www.sap.com/about/legal/trademark.html> for additional trademark information and notices.