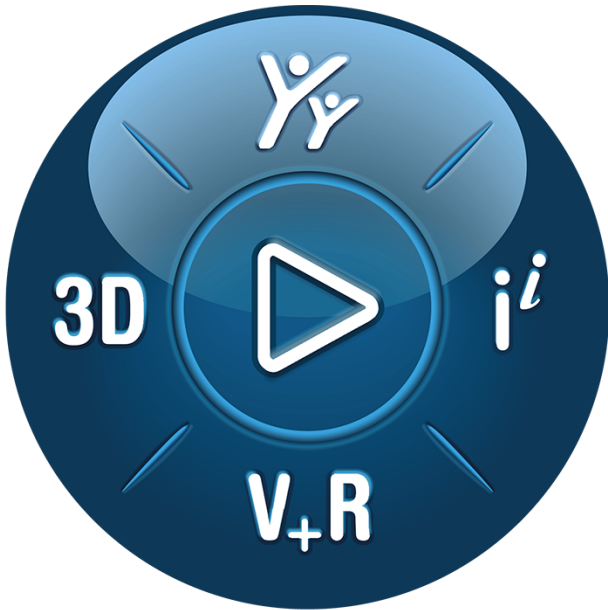


Reporting Framework - Crystal Reports

DELMIA Apriso 2021 Technical Guide



3DEXPERIENCE®

©2020 Dassault Systèmes. Apriso, 3DEXPERIENCE, the Compass logo and the 3DS logo, CATIA, SOLIDWORKS, ENOVIA, DELMIA, SIMULIA, GEOVIA, EXALEAD, 3D VIA, BIOVIA, NETVIBES, and 3DXCITE are commercial trademarks or registered trademarks of Dassault Systèmes or its subsidiaries in the U.S. and/or other countries. All other trademarks are owned by their respective owners. Use of any Dassault Systèmes or its subsidiaries trademarks is subject to their express written approval.

Contents

1 Introduction	5
1.1 Objective	5
1.2 Scope	5
1.3 Definitions, Acronyms, and Abbreviations	5
2 Framework Overview	6
2.1 Motivation	6
2.2 Features	6
2.3 Implementation	7
3 Usage	8
3.1 Key Aspects	8
3.2 Tools Installation and Prerequisites	8
3.3 Central Configuration	8
3.4 Configuring a Developer Machine for Designing Crystal Reports for DELMIA Apriso	9
3.5 Creating a Report Definition File	9
3.5.1 Creating a New Report	9
3.5.2 Defining a Grouping	10
3.5.3 Defining Filters for Reports	11
3.5.4 Defining Additional Parameters	11
3.5.5 Localization of Static Literals	11
3.5.6 Using UTC Conversion Functions	12
3.5.7 Making Culture-Dependent Date and Time Fields	13
3.6 Creating a Report Configuration File	14
3.6.1 General Properties of Reports	14
3.6.2 Filters	15
3.6.3 Parameters	16
3.6.4 Grouping	17
3.6.5 Discrete Values	18
3.6.6 Query Values	18
3.7 Using ADO.NET DataSets in Reports	19
3.7.1 Preparing a DataSet in C# Code	19
3.7.2 Using a DataSet in a Report	20
3.8 Performing Other Tasks	20
3.8.1 Writing Custom User-Defined Functions	20
4 Use Cases	23
4.1 Configuring an ODBC Connection	23
4.2 Creating Crystal Report	25
4.3 Modifying the Report	30
4.3.1 Adding Crystal Report Parameters	30
4.3.2 Adding Formula Fields	33
4.4 DELMIA Apriso Configuration	37

4.4.1 Adding DELMIA Apriso Parameters	37
4.4.2 Moving RPT and XML Files	38
4.4.3 Loading Report Administrator	38
4.4.4 Report Testing	40
4.4.5 Data Translation and Local Time Adjustment	41
5 Best Practices	46
5.1 Report Organization and Style	46
5.2 Standards	46
5.2.1 Paper Size and Fonts	46
5.2.2 Header	46
5.2.3 Footer	46
5.3 Common Mistakes	47
5.3.1 Binding Reports Statically to One Database	47
5.3.2 Using Views in Reports	47
5.3.3 Outer Joins	47
5.3.4 Selection Formulas	48
5.3.5 Using Sub-Reports	48
5.3.6 Page Setup – Optimization for Screen Display	48
5.4 Optimizing Reports for Exporting to Excel	50
6 Known Issues	51
7 References	54

Figures

Figure 1 FlexNetLocalizationLocalizeLiteral location in the Functions tree	12
Figure 2 Date Format Editor window	13
Figure 3 Configuring the report general properties	15
Figure 4 Configuring report filters	16
Figure 5 Configuring report parameters	17
Figure 6 Configuring report groupings	19
Figure 7 ODBC Data Source Administrator	23
Figure 8 Oracle ODBC Driver Configuration	24
Figure 9 Testing the data source connection	24
Figure 10 Creating a new report	25
Figure 11 Creating a new database connection	25
Figure 12 Selecting the data source	26
Figure 13 Entering the connection information	26
Figure 14 Data view	27
Figure 15 Selected Tables	27
Figure 16 Predefined table links	28
Figure 17 Adding the fields to display	28
Figure 18 Defining the grouping information	29
Figure 19 Report preview	29

Figure 20 Design view	30
Figure 21 Field Explorer window	30
Figure 22 ReportID parameter	31
Figure 23 LanguageID parameter	32
Figure 24 TimeZoneID parameter	32
Figure 25 Parameters added in Field Explorer	33
Figure 26 Selecting Formula Workshop	33
Figure 27 Adding a new formula	33
Figure 28 Editing the f_Title formula field	34
Figure 29 Library view	34
Figure 30 Placing the f_Title field in report header	35
Figure 31 Format field option	35
Figure 32 Font settings	35
Figure 33 Formatted report design	36
Figure 34 Report values	36
Figure 35 Report preview	37
Figure 36 Loading Report Administrator	39
Figure 37 Report Explorer	39
Figure 38 Adding report configuration and definition files	40
Figure 39 Opening Report Viewer	40
Figure 40 Viewing a report	41
Figure 41 Selecting the record editor	41
Figure 42 Record Selection Formula Editor	42
Figure 43 Adding a new formula field	42
Figure 44 Entering the formula name	43
Figure 45 f_LastUpdateOn formula text	43
Figure 46 Adding a formula to the report section	43
Figure 47 Specifying the report parameters	44
Figure 48 Final version of the report	44
Figure 49 Original data on the database server	45
Figure 50 No Printer option on the Page Setup screen	49
Figure 51 Trimmed report view	49

1 Introduction

1.1 Objective

The objective of this document is to gather all of the information required by the programmer to utilize the Reporting Framework using Crystal Reports. The knowledge included in this document should allow the programmer to use the framework without major prior knowledge of the subject.

1.2 Scope

The scope of this document describes the following core areas:

- ▶ The steps required to create a new DELMIA Apriso Reporting Services report
- ▶ Creating a report configuration file
- ▶ Using .NET methods in a report
- ▶ Framework limitations
- ▶ Best practice
- ▶ Known issues

1.3 Definitions, Acronyms, and Abbreviations

Report file – a file generated by Crystal Reports (Business Objects) that stores information about the report.

Report configuration file – a file used by the DELMIA Apriso Reporting Framework that contains information such as parameters, groupings, filters, etc.

2 Framework Overview

All of the Reporting Frameworks supported in DELMIA Apriso are presented for comparison in the following table:

Framework	Pros	Cons
Crystal Reports	<ul style="list-style-type: none"> ▶ Works on both Oracle and SQL databases ▶ Works on both 32-bit and 64-bit platforms ▶ Is the industry leader 	<ul style="list-style-type: none"> ▶ Requires a developer license to build the reports
SQL Server Reporting Services	<ul style="list-style-type: none"> ▶ Comes for free with the SQL Server ▶ No runtime or developer license is required ▶ Works on both 32-bit and 64-bit platforms 	<ul style="list-style-type: none"> ▶ Requires an SQL Server database, which is a problem for Oracle customers
XtraReports	<ul style="list-style-type: none"> ▶ Works on both Oracle and SQL databases ▶ No runtime license is required ▶ Works on both 32-bit and 64-bit platforms 	<ul style="list-style-type: none"> ▶ Requires a developer license to build the reports

2.1 Motivation

The motivation for developing the Crystal Reports Reporting Framework was to provide a flexible and easy way to print and display reports when retrieving data from a database, especially for those customers who wish to develop their own reports. It provides a means to easily integrate reports without additional development effort.

2.2 Features

The following features are characteristic of Crystal Reports:

- ▶ Users are provided with the ability to create new reports and add those reports to the DELMIA Apriso Report Viewer without programming
- ▶ Users can create report definition files containing groupings, filters, and parameters that can be applied to reports
- ▶ Only one definition file is generated, which can be used by both MS SQL Server and Oracle
- ▶ The framework allows for changing a database connection without programming and modifying the report definition files

- ▶ The framework supports the configuration of filters, parameters, and groupings without programming and changing the framework
- ▶ The framework makes it possible to create reports that can be localized
 - ▷ All static literals and literals from a database are localized, while UTC time conversions are made in the report
- ▶ Support for calling external .NET written functions for complicated calculations
- ▶ Support for reports obtaining data indirectly from a database but from .NET pre-generated data sets for very complicated reports that cannot be realized using SQL queries in the report

2.3 Implementation

The Web Reporting Framework is implemented as a configurable Web viewer using a report configuration file as an input. Based on an XML, a dynamic UI is created for each report. When using the UI, you can configure the report's parameters, which are passed on to the report engine by the framework.

3 Usage

3.1 Key Aspects

The main task of the Report Designer is to define new reports. To accomplish this, a report must be designed in Crystal Reports using standard Crystal rules as well as several rules unique to DELMIA Apriso. Corresponding configuration files are then created and the report must be registered in the DELMIA Apriso Report Administrator.

3.2 Tools Installation and Prerequisites

- ▶ To display Crystal Reports in the runtime, the SAP Crystal Reports runtime engine (Version 13, Service Pack 18) must be installed on the DELMIA Apriso server and the Central Configuration flag must be enabled (see [3.3 Central Configuration](#) for details).
- ▶ To design Crystal Reports, the Crystal Reports Designer or Crystal Reports for Visual Studio .NET ClickOnce (Version 13, Service Pack 18) must be installed on the developer's machine. This document describes the latter in use.

i Since the Express version of Visual Studio 2010 does not support plug-ins, it is required to have the Professional, Premium, or Ultimate version installed for use with Crystal Reports for Visual Studio (which is considered a plug-in).

- ▶ Microsoft Office with InfoPath (to ease report configuration file creation).

3.3 Central Configuration

Crystal Reports framework can be configured using keys located in the “Reporting” section of the Central Configuration file (for details, see [Central Configuration Documentation](#)). The Crystal Reports framework uses the following keys:

- ▶ `EnableCrystalReports` – the flag must be set to True for Crystal Reports to be used (by default it is set to False)
- ▶ `ReportLibraryPath` – points to the report location¹
- ▶ `LabelLibraryPath` – points to the label location¹
- ▶ `ReportHeaderString` – your company name

Custom connections allow the user to choose the database that will be used to generate the report:

- ▶ `CustomConnectionAlias1` – the name that will be displayed in Report Administrator
- ▶ `CustomConnectionType1` – Crystal Reports
- ▶ `CustomConnectionParameters1` – the database connection string

Export configuration:

¹When setting a location different than the original one, please make sure it has the proper access rights configured. The rights are “Read & execute,” “Read,” and “List folder contents” for the “<domain>\Users” user group.

- ▶ ReportExportDir – the directory where reports/labels are generated
- ▶ ReportExportUrl – the URL where reports/labels are generated

i Standard database connection configuration is performed with use of the DELMIA Apriso Configuration Manager (for details, refer to the [Configuration Manager Help](#)).

3.4 Configuring a Developer Machine for Designing Crystal Reports for DELMIA Apriso

The DELMIA Apriso Reporting Framework offers special features (like localization of UTC conversions) that can be used in reports. To enable these special features, the following configuration steps must be performed on the machine where reports will be designed:

1. Install the Crystal Reports Designer on the development machine.
2. Install .NET Framework on the development machine.
3. Install the report development files using the DELMIA Apriso Client Installer.

3.5 Creating a Report Definition File

3.5.1 Creating a New Report

To create a report file, the Crystal Report Designer application integrated with Visual Studio .NET or the standalone product from Crystal Decisions (Business Objects) must be used.

The best way to create a new report that can be used in DELMIA Apriso Report Viewer is to use one of the template files delivered with DELMIA Apriso (these can be found in the **File Packs** section of **Tools and File Packs** on the DELMIA Apriso Server Configuration page). Those templates include all of the standard parameters, a standard report header, and a page footer.

When creating a new report, the following rules must be obeyed:

- ▶ Create a new report file (*.RPT) and report configuration file (*.XML)
 - ▷ The name of the report definition file must be the same as the name of the report configuration file (excluding the extension). Usually the file name should look as follows:

```
FlexNet Rpt{Report Number} {Report Name}.{rpt | xml}
```

i The report configuration file must precisely match the report file. For more details, refer to [3.7 Using ADO.NET DataSets in Reports](#).

- ▶ Specify a data source for the report – generally, DELMIA Apriso supports two types of data sources:
 - ▷ ODBC with user authentication (not Windows)
 - ▷ ADO.NET (if a user wants to retrieve data programmatically using C# .NET components)
For more details, refer to [3.7 Using ADO.NET DataSets in Reports](#)

i If problems arise while using ODBC for Oracle DBMS, create new report files using the Oracle Server data source as described in [5.1 Report Organization and Style](#).

3.5.2 Defining a Grouping

Reports can be created that are displayable in different manners, depending on which grouping the user selects.

To create a report that can be grouped, the `GroupingID` parameter must be defined. This parameter is required to determine which grouping has been selected by the user (if a report is being created from a template, this parameter does not have to be defined). The grouping formula could resemble the following:

```
select {?GroupingID}
  case "WorkOrder":
    {WIP_ORDER.WipOrderNo}
  case "Product":
    {PRODUCT.ProductNo}
  case "WorkCenter":
    {WORK_CENTER.WorkCenter}
  default: ""
```

The above formulas verify which grouping has been selected by the user and return an adequate value from the variables used in the report.

Create a new group in the report and select the created formula as the "Grouped By" field. To make a grouping available to users, refer to [3.6.4 Grouping](#).

After completing the above steps, the user will have a grouping in their report that is modifiable.

If the user wants to have a report that is grouped by more than one formula, he or she needs to create adequate formulas for each grouping level (in Crystal Reports, groups implicate themselves, and there can only be one group on each level).

For example, assume a second formula has been created in the following way:

```
select {?GroupingID}
  case "WorkOrder":
    {PRODUCT.ProductNo}
  case "Product":
    {WIP_ORDER.WipOrderNo}
  case "WorkCenter":
    {WIP_ORDER.WipOrderNo}
  default: ""
```

In this case, if the report is grouped using both exemplary formulas and the user selects grouping by “Work Order,” data in the report will be grouped by WIP_ORDER.WipOrderNo and then by PRODUCT.ProductNo.

3.5.3 Defining Filters for Reports

If the user wants a report that can be filtered, then the only option is to add a filter section to the report configuration file. For details, refer to [3.6.2 Filters](#).

3.5.4 Defining Additional Parameters

To define new parameters, please refer to the Crystal Reports documentation. To make the parameters available to users, refer to [3.6.3 Parameters](#) for more details.

i Parameter names in sub-reports cannot be the same as parameter names in the main report.

3.5.5 Localization of Static Literals

i The functions described here may not work if the Crystal Reports were upgraded since DELMIA Apriso was installed. To resolve this problem, refer to [5.1 Report Organization and Style](#).

⚠ Make sure that the `ClientApplications.xml` file in Central Configuration directory on DELMIA Apriso server contains the `FlexNet.BusinessRules.Printing` client application entry. If not, it must be added manually in order for static literals localization to work correctly.

To localize static literals in a report, the user needs to use a formula that invokes the `FlexNetLocalizationLocalizeLiteral` function. This function can be selected from the Functions tree under Additional Functions.

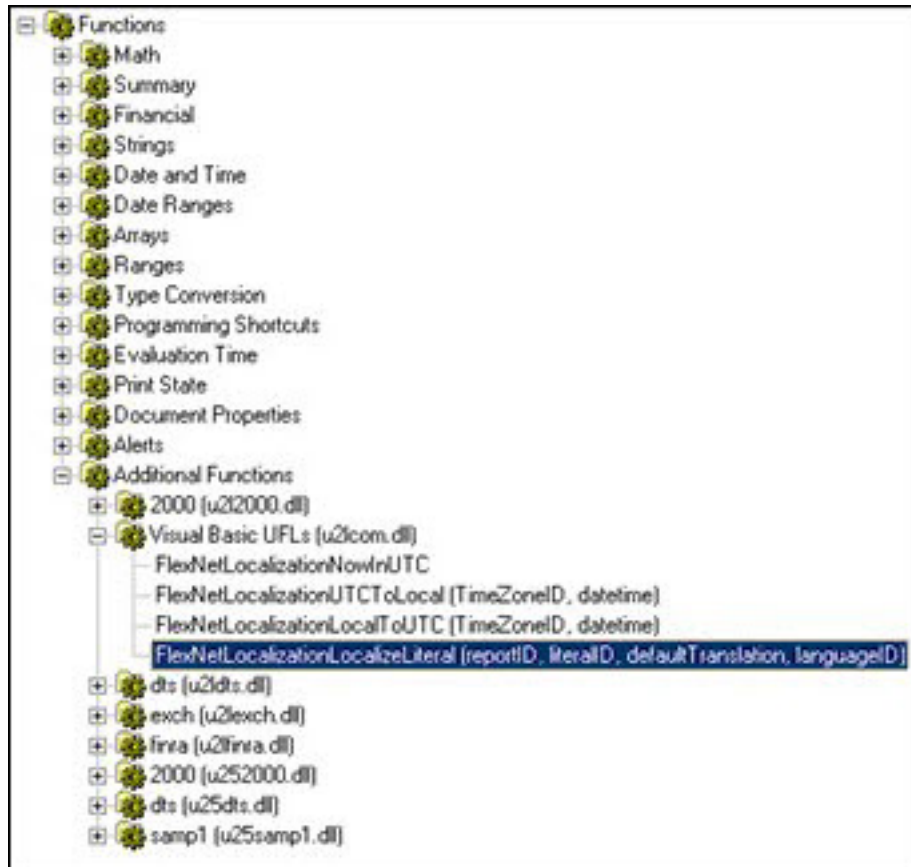


Figure 1 FlexNetLocalizationLocalizeLiteral location in the Functions tree

The Report Developer needs to pass to this formula the `ReportID` and `LanguageID` parameters as well as the default translation, which is the literal to be displayed and the `literalID` (that identifies which literal is used). The `literalID` should be a unique value within the report, and usually it is the default translation without spaces.

For example, if the user wants to display “Work Order”, then in the formula he or she calls:

```
FlexNetLocalizationLocalizeLiteral({?ReportID}, "WorkOrder", "Work Order", {?LanguageID});
```

Refer to the [Process of Translation Implementation Guide](#) for details on report localization.

3.5.6 Using UTC Conversion Functions

i The functions described here may not work when Crystal Reports are upgraded after DELMIA Apriso is installed. To resolve this problem, refer to section [5.1 Report Organization and Style](#).

When a date is in the UTC time zone, localization of the report’s `DateTime` value field can be accomplished by creating a formula that invokes either the `FlexNetLocalizationUTCtoLocal` or `FlexNetLocalizationUTCtoLocal11` function (which can be selected from the Functions tree under “Additional Functions”).

FlexNetLocalizationUTCToLocal11(int TimeZoneID,string datetime, int languageID) allows the user to localize DateTime values, regardless of employee personalization and server localization settings. By default, languageID should be taken from the LanguageID report parameter. This method should be used instead of FlexNetLocalizationUTCToLocal.

Conversion from local to UTC time can be done in a similar way through the use of the FlexNetLocalizationLocalToUTC OR FlexNetLocalizationLocalToUTC11 function.

The TimeZoneID and DateTime values (converted to string) must be passed to that function, which returns a localized DateTime string.

For example, if a user wants to convert WIP_ORDER.CreatedOn to local time in the dd-MMM-yy hh:mm format, then the formula should look like this:

```
local stringVar createdOnString;

createdOnString:= FlexNetLocalizationUTCToLocal({?TimeZoneID}, Cstr( {WIP_ORDER.CreatedOn}
) );

if IsDateTime(createdOnString) then
    ToText( DateTimeValue(createdOnString), "dd-MMM-yy hh:mm" );
```

3.5.7 Making Culture-Dependent Date and Time Fields

To make date-time fields culture dependent, a user needs to select in the **Format Editor** ► **Date** tab “System Default Long Format” or “System Default Short Format” for Date value fields and “System Default Time” for Time value fields (the date format will be retrieved from the system “Regional and Language Options” settings).



Figure 2 Date Format Editor window

3.6 Creating a Report Configuration File

Report Designers must create a configuration file in the XML format for every report. This describes the report's name, version, parameters, and the filters it will use. To create this file, Report Designers should use the `ReportDefinition.xsn` file as a template. This can be edited using MS InfoPath 2003. The configuration can also be created manually according to the `ReportDefinition.xsd` schema. This document describes only the first method.

To create the configuration file, users must fill in the template with the data they need in the report. Some sections are optional (a short Help is on the right-side of the panel).

Both the `ReportDefinition.xsn` and `ReportDefinition.xsd` files are available on the DELMIA Apriso server through the DELMIA Apriso Server Configuration page (in Tools and File Packs).

3.6.1 General Properties of Reports

Properties need to be defined for every report. These properties are at the top of the panel. Users have to set basic properties for each report, such as:

- ▶ **Report ID** (required value) – the report's unique identifier
 - ▷ This is used for localization and is set as the value for the `ReportID` parameter in the report
 - ▷ Typically, the report ID should resemble the following:

```
FlexNet.Reports.{TypeOfYourReport}.Rpt{ReportNumber}_{ReportNameWithoutWhitespaces}
```

For example:

```
FlexNet.Reports.Production.Rpt15_WorkOrderOperationPlannedVsActualDates
```

- ▶ **Report Name** (required value) – the name of the report that will be displayed to the user
 - ▷ Typically, the following template should be used:

```
FlexNet Rpt{ReportNumber} {Report Name}
```

For example:

```
FlexNet Rpt15 Work Order Operation Planned vs. Actual Dates
```

- ▶ **Report Version** (required value) – the version number of the report should be inserted here
- ▶ **Report Engine** (required) – for each Reporting Service there should always be a "ReportingServices" value
- ▶ **FilterType** – determines where filters will be performed, and the acceptable values are:
 - ▷ Crystal – filtering will be performed in the report, but the user cannot have a Report Generation section in the configuration file (this is the default value)

- ▷ SQL –filtering will be performed during report generation, but this can be used only if the report is generated; therefore, if it is set to SQL, then a Report Generation section must be added to the configuration file (see [3.7 Using ADO.NET DataSets in Reports](#))

Under **Description**, the report can be described in detail (not required).

Figure 3 Configuring the report general properties

3.6.2 Filters

If a user wants to enable filtering in their report, they must click “Click here to define filters for report.” Every defined filter field must be placed in that section. For each filter field, the user must define:

- ▶ **Filter Name** – the name of the filter field, which is usually the table name and field name used in the Report Definition, for example:

```
WIP_ORDER.WipOrderNo - for field from database used in report
@FormulaFieldName - for formula field
?ParameterName - for parameter
```

- ▶ **Value displayed to the user** – the value that will be displayed in the Filter Editor in Report Viewer
- ▶ **Type of filter field value** – the type of filter field that is being used in the report
 - ▷ Users cannot use every type that is available in Crystal Reports
 - ▷ The types available are:

Type	Description
Integer	The same as Number in Crystal Reports.
Decimal	The same as Decimal in Crystal Reports.
Char	The same as String in Crystal Reports and SQL.
DateTime	The same as DateTime in Crystal Reports.
Date	The same as Date in Crystal Reports.

Bool	The same as Bool in Crystal Reports. Available values are in Discrete Values, and there are two Discrete Values with Actual Values True and False.
------	--

Table 1 Available field types

Discrete Values or Query Values can also be defined for a filter by clicking on the appropriate link. For more details, refer to [3.6.5 Discrete Values](#) and [3.6.6 Query Values](#).

Figure 4 Configuring report filters

i Discrete Values and Query Values cannot be set together – only one of them can be used for values.

3.6.3 Parameters

Parameters are useful when a report needs to obtain some information from a user before it is displayed.

Besides standard parameters, each parameter that is defined in the report file must be located in the corresponding configuration file. To define a parameter, click “Click here to define parameter for report.” The following can be defined for each parameter:

- ▶ **Parameter Name** (required value) – the name of the parameter that is defined in the report
- ▶ **Value displayed to the user** – the value that will be displayed in the Parameters section in Report Viewer
 - ▷ The value does not have to be defined if the parameter is Internal or Hidden
- ▶ **Type of parameter value** – the type of parameter in the report
 - ▷ The available types are the same as for filters (see [3.6.2 Filters](#))
 - ▷ The value does not have to be defined if the parameter is Internal
- ▶ **Picked Value Required** – used when Discrete Values and Query Values are defined (see [3.6.5 Discrete Values](#) and [3.6.6 Query Values](#))

- ▶ **Value is Required** – if checked, the user does not have to pass the value for that parameter (by default this is selected)
- ▶ **Internal Parameter** – if selected, the parameter will be identified by the report file as internal – no value will be required and passed from it
 - ▷ Additionally, it will not be displayed to the user in the Report Viewer
- ▶ **Hidden for the user** – if selected, the parameter will not be displayed to the user in the Report Viewer, but it will still need to have a default value set that is to be used by the report (by default this is not selected)
- ▶ **Key Value Number** – when printing a report from a function, parameter values are saved to specific fields of the PRINT_REQUEST_HISTORY table
 - ▷ These fields are named Key1, Key2, Key3, etc.
 - ▷ The Key Value Number allows for assigning parameters to table fields (e.g., the Key1 field corresponds to Key Value Number "1")
- ▶ **Default Value for parameter** – the default value for the parameter, which will be displayed to the user
 - ▷ The default value is required only if the parameter is Hidden

Discrete Values and Query Values can also be defined for parameters by clicking the appropriate link. For more details, refer to [3.6.5 Discrete Values](#) and [3.6.6 Query Values](#).

Parameters

Parameter Name	Value displayed to the user
OperationsNotStarted	Operations Not Yet Started

Type of parameter value:

Picked Value Required Value is Required
 Internal Parameter Hidden for the user

Key Value Number:

Discrete Values

Rendered Literal	Value
No	False
Yes	True

[Click here to insert query for parameter](#)

Default Value for parameter:

Figure 5 Configuring report parameters

i Discrete Values and Query Values cannot be set together – only one of them can be used for values.

3.6.4 Grouping

In this section of the report configuration file, the available groups for a report are written. To add groups to a report, click “Click here to insert grouping options for report” and then input all

of the available groups that a user should be able to select for a report.

In “Value displayed to the user,” a value can be defined that will be displayed to the user in the grouping section in Report Viewer. In “Group,” a value can be set that will be passed to the `GroupingID` parameter of the report (this should be the group name defined in the report).

In “Default Group,” the report’s default group can be defined.

3.6.5 Discrete Values

Discrete Values are listed values that are available to the parameter or filter fields. If the user wants to insert a Discrete Value, then he or she has to define the value with the same type as a value selected from the drop-down. For every value there should be defined a rendered literal that will be displayed to the user instead of the value.

Picked Value Required can also be defined for filters. If selected, then the user can only select values displayed in the combo box. If cleared, then the user can also enter their own values. By default it is selected. Usually users set it as cleared when the values displayed to the user are the same as the values passed to the filter field. This happens when the Discrete Values rendered literal and the value are the same, or the Query Values rendered column and the value column are the same. If they are different, then it is preferable that the setting be selected.

3.6.6 Query Values

In this section a query can be defined that retrieves values for a parameter or filter field that are stored in the database. Users can insert a Query Values section into the configuration file by clicking “Click here to insert query for filter.”

To get those values, users can define:

- ▶ **Query** – this is the SQL query that retrieves data from the database
 - ▷ A query should be written that can be used on both SQL and Oracle servers

 This query can be parameterized by two variables: `@EmployeeID` and `@LanguageID`.

- ▶ **Rendered column** –the name of the column that will have its values displayed to the user
- ▶ **Value column** – the name of the column that will have its values passed to the parameter or filter field

Picked Value Required can also be defined for filters. For more information, refer to [3.6.5 Discrete Values](#).

Grouping options	
Value displayed to the user	Group
Grouping by Work Order	WorkOrder
Grouping by Product	Product
Grouping by Work Center	WorkCenter

Default group:

Figure 6 Configuring report groupings

3.7 Using ADO.NET DataSets in Reports

3.7.1 Preparing a DataSet in C# Code

It is possible to make a report that uses DataSets as a data source by following these steps:

1. Define the DataSet definition file (*.XSD) in Visual Studio .NET.
2. Write a class that implements the interface described below from the FlexNet.BusinessRules.Printing.dll assembly, which fills the DataSet that will be used as the data source for the report:

```
public interface IReportGenerator
{
    Outcome GenerateReport( string reportID, ReportPrintRequest profile, out DataSet
    dataSet, bool limitSize );
}
```

The following parameters are passed to GenerateReport:

- ▷ reportID – the unique ID of the report defined in the report configuration file
- ▷ limitSize – selected means that the size of the DataSet can be limited if it is too large and only part of the database data will be retrieved
- ▷ profile – the user profile for that report
 - ▷ It contains ParameterValues and values that were set by the user, the grouping that the user selected (if available), and Filter, which contains the filters that the user set for the report
 - ▷ The most important for generation of the report is *Filter* and occasionally *ParameterValues* (depending on the report)
 - ▷ From *Filter*, the user can retrieve the *FilterCondition* and *Expression* collection that can be used in the WHERE clause

3. In the report configuration file, the following section needs to be added:

```
<ReportGeneration>
<ClassName>Your class name</ClassName>
<AssemblyName>
Your assembly name without .dll extension
</AssemblyName>
</ReportGeneration>
```

3.7.2 Using a DataSet in a Report

To use a DataSet in a report, the user should set the DataSet defined as the report's data source. To set a DataSet as a data source, in Database Expert in the Data tab select "More Data Sources" and then ADO.NET (XML), and then browse to the DataSet's XSD file.

3.8 Performing Other Tasks

3.8.1 Writing Custom User-Defined Functions

User-defined functions can be designed and programmed in any language that supports the development of Windows DLLs. This means that an automation server can be created in any language environment that supports COM. Users must name the automation server with a CRUFL prefix and register it on a computer. Seagate Crystal Reports Formula Editor will then be accessible and make available any functions exposed by that automation server.

This is an example where user-defined functions are created using C# in Visual Studio .NET:

1. Create a C# Class Library project.
2. If the user wants to register the COM automatically, they must set:

Project ▶ Properties ▶ Configuration Properties ▶ Build ▶ Register for COM Interop = selected

This will automatically remove old functions and register the new one. This is a sample code with descriptions:

```

//the namespace must have CRUFL prefix
namespace CRUFLNamespace
{
//Here you must insert functions declarations
//(only this functions which you want to use in your reports).
//
//RESTRICTIONS:
//Interface must be public.
//
//Function parameter can't be decimal, it can be replace with float
//(function can't return dacimal as well).
//
//If you want to pass DateTime into function or if you want function //return DateTime
you must convert DateTime to string (string to //DateTime)
//
//You can't use out, ref parameters
//this number must be unique, you can create it using "uuidgen.exe"
//in .NET command prompt
[Guid("place your guid here")]
public interface IYourInterface
{
    string Method1(int param1,string param2);
    int Method2(string param1,float param2);
string Method3(string param1);
}

//This class must implement interface above.
//All functions implements this interface must be public
//This number must be unique you can create it using "uuidgen.exe"
//in .NET command prompt
[Guid("place your guid here")]
[ClassInterface(ClassInterfaceType.None)]
public class YourClassName : IYourInterface
{
    public string Method1(int param1,string param2)
    {
        //your code
    }
    public int Method2(string param1,float param2)
    {
        //your code
    }
    public string Method3(string param1)
    {
        //your code
    }
}
} //namespace CRUFLNamespace

```

3. If a user wants to use the newly created function in a report, he or she must:
 - a. Open the correct report.
 - b. Open the Formula Editor.
 - c. Go to: **Additional Functions ▶ Visual Basic UFLs**.
 - d. Choose the defined function.

The Crystal Reports construct function name uses the following rules:

- ▷ Takes Namespace (part after CRUFL)
- ▷ Appends Class name
- ▷ Appends function Name
- ▷ Eliminates any non-alphanumeric characters

This is a sample:

```
namespace CRUFLNamespace
{
    public class Class : IInterface
    {
        public string Method(string s)
        {
        }
    }
}
```

The function constructed will have the name: NamespaceClassMethod.

4 Use Cases

The following use cases provide guidance in the area of report configuration and creation from a blank page. Apart from the flows described in this chapter, it is also possible to use the report template shipped with DELMIA Apriso in order to create a report starting with default settings, like standard parameters, a standard report header, and a page footer.

i The use cases described in this chapter are based on an Oracle database and may vary for SQL scenarios.

4.1 Configuring an ODBC Connection

Create an ODBC (Open Data-Base Connection) link into Windows:

1. Go to the **Control Panel** and select **Administrative Tools**, then select **Data Sources (ODBC)**. Next, go to the **System DSN** tab and select **AprisoReports**.

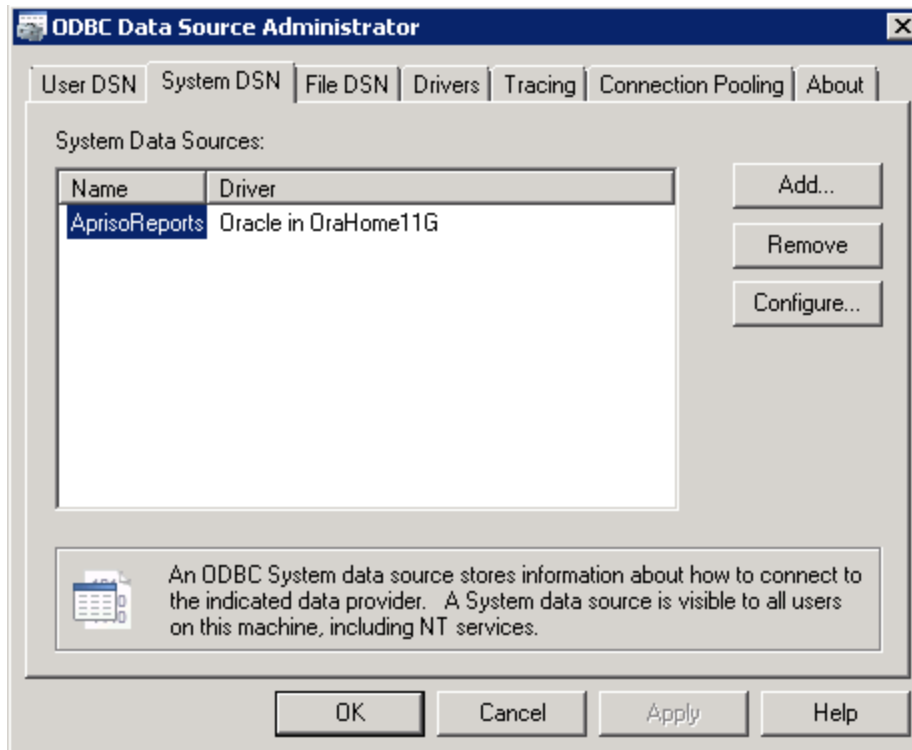


Figure 7 ODBC Data Source Administrator

2. If the AprisoReports data source does not exist, click **Add** and select the proper driver named **Oracle in [ORACLE_HOME]**. Enter the data source name **AprisoReports**.
3. Select the correct TNS service name and user ID to connect to DELMIA Apriso and click the **Test Connection** button.

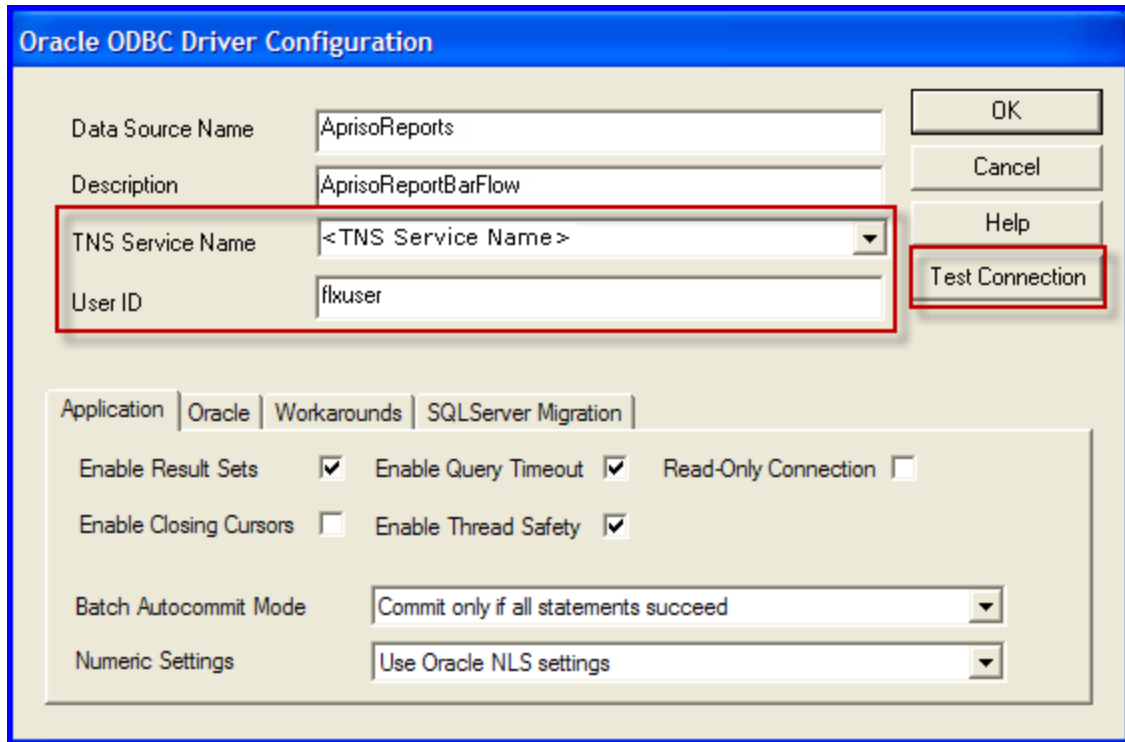


Figure 8 Oracle ODBC Driver Configuration

4. Test the database connection using the **flxuser** login.

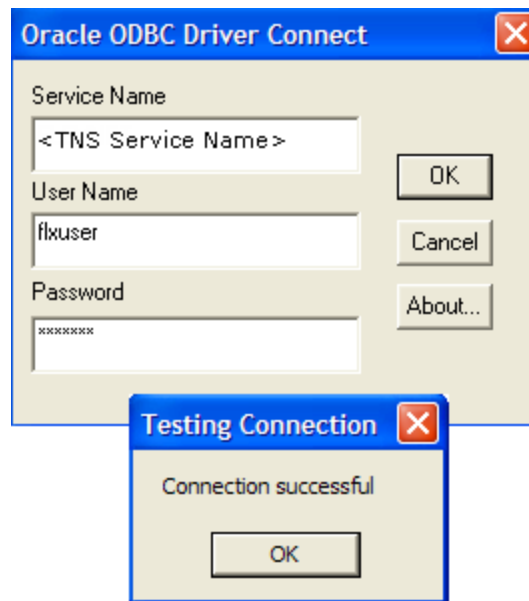


Figure 9 Testing the data source connection

4.2 Creating Crystal Report

1. Launch the Crystal Reports application and create a new report:

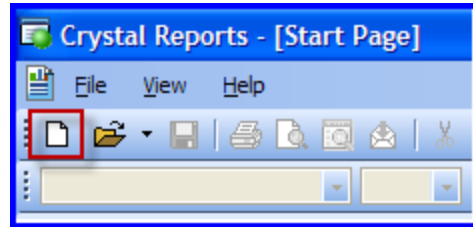


Figure 10 Creating a new report

2. Create a new database connection and double-click **ODBC (RDO)**.

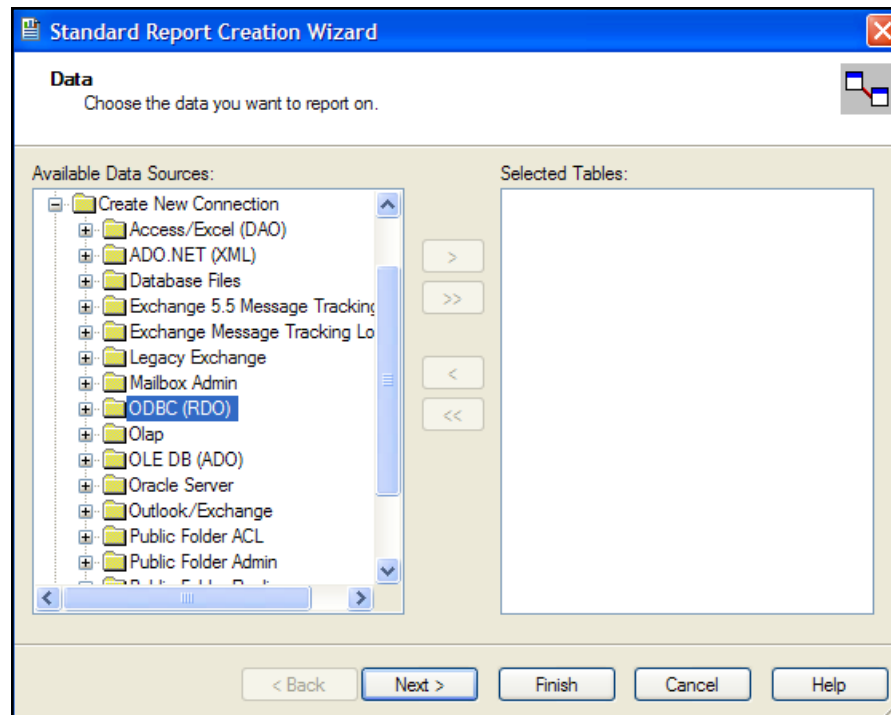


Figure 11 Creating a new database connection

3. Choose **AprisoReports** from the **Data Source** list:

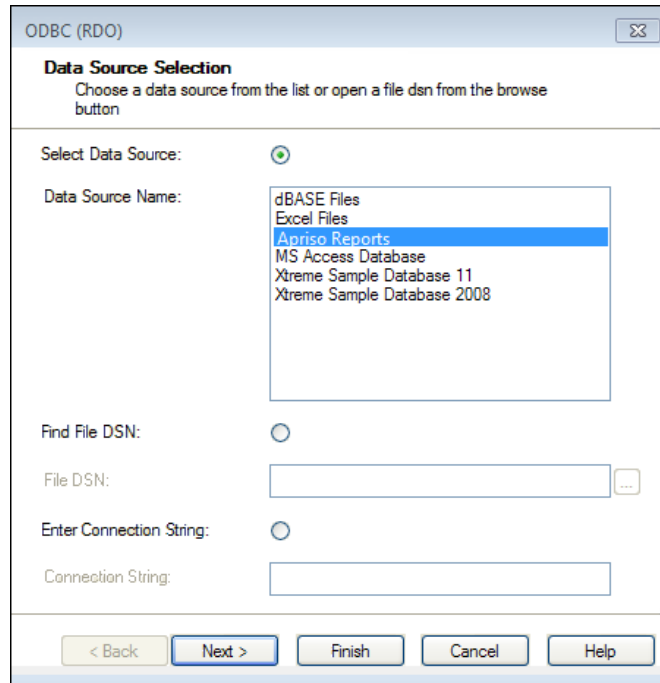


Figure 12 Selecting the data source

4. Provide the connection information.

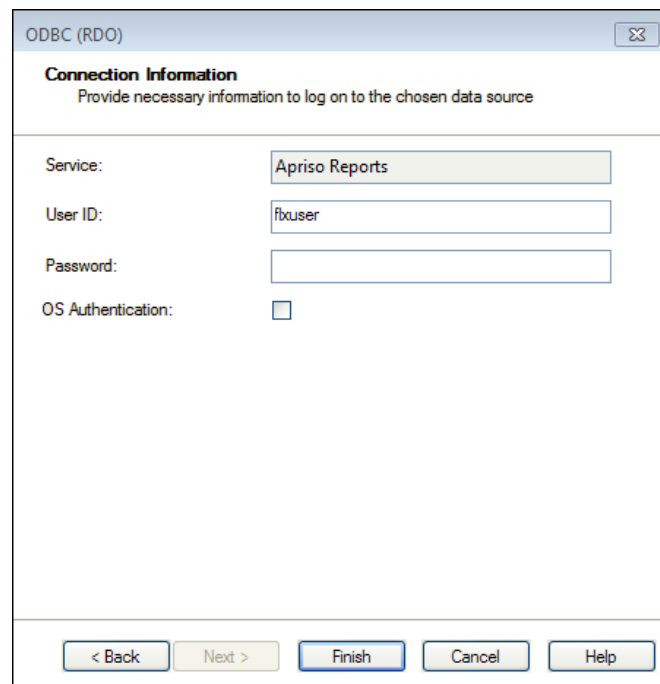


Figure 13 Entering the connection information

5. Choose the tables (expand Tables and then choose PRODUCT and TEXT_TRANSLATION).

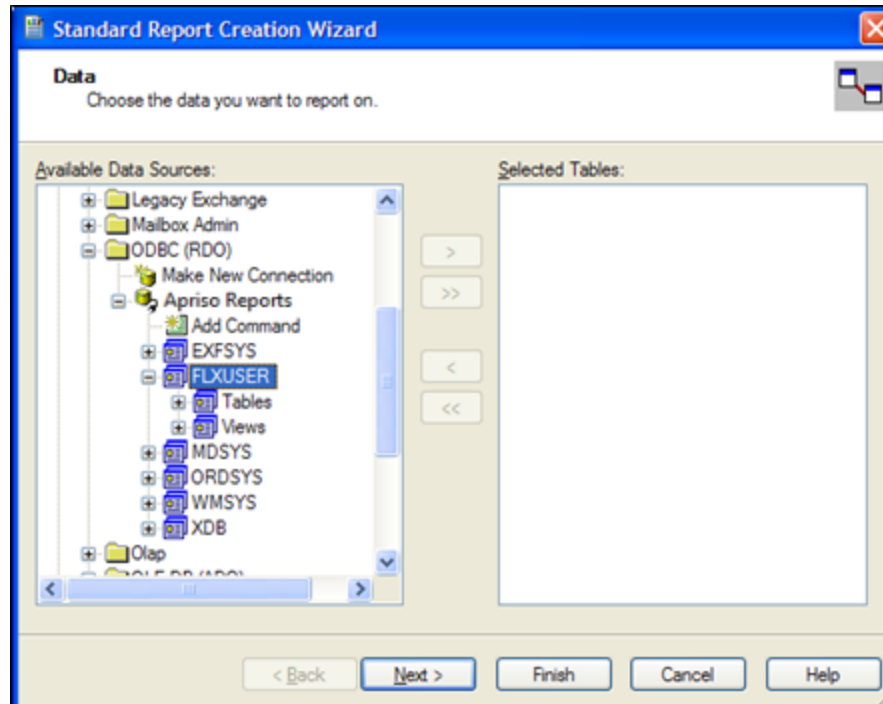


Figure 14 Data view

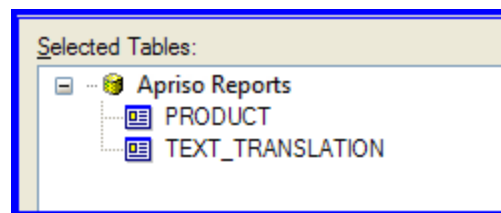


Figure 15 Selected Tables

6. Use the predefined links between these two tables.

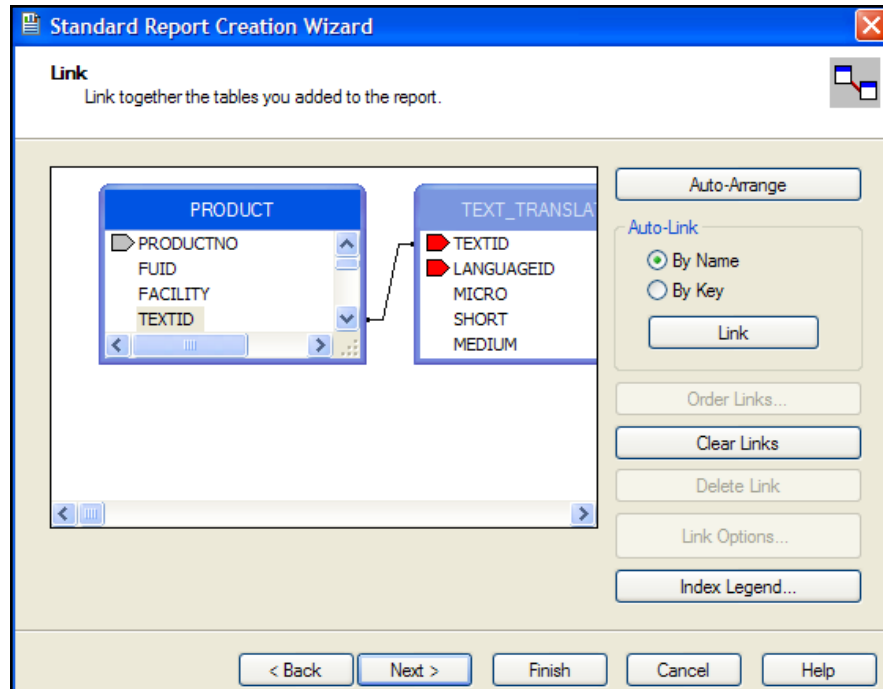


Figure 16 Predefined table links

7. Choose PRODUCTNO and MEDIUM as the fields to be displayed:

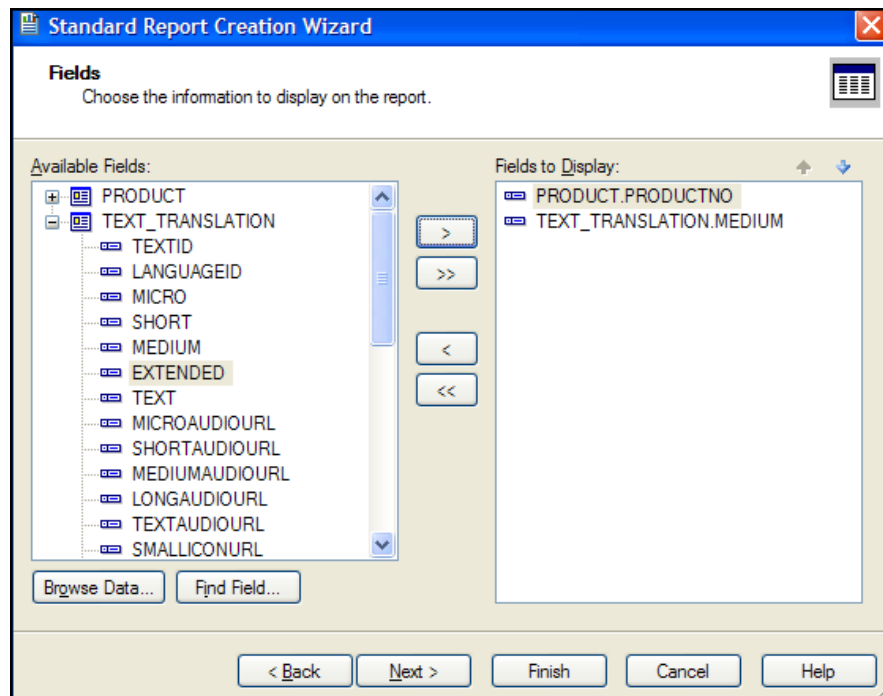


Figure 17 Adding the fields to display

8. Add a grouping by FACILITY:

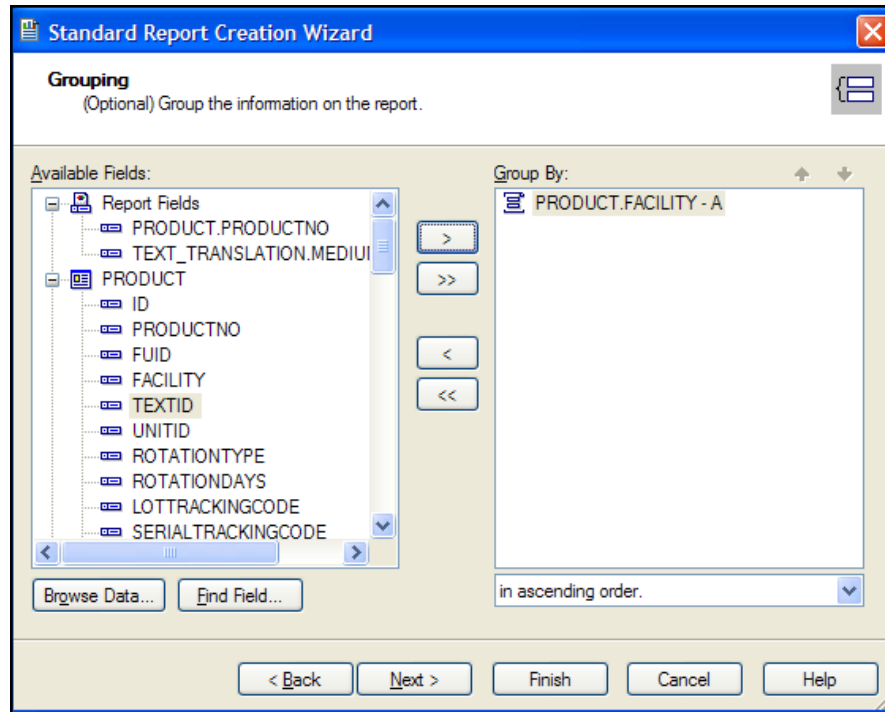


Figure 18 Defining the grouping information

The preview of the report is as follows:

22/05/2008		
<u>FACILITY</u>	<u>PRODUCTNO</u>	<u>MEDIUM</u>
C1P1		
C1P1	PC-TOWER	Tower PC
C1P1	APR_PRD_Skateboard	
C1P1	PC-TOWER	Tower PC
C1P1	PC-TOWER	Gabinete PC
C1P1	PC-RAM	RAM for PC
C1P1	PC-RAM	PC RAM
C1P1	PC-RAM	RAM do komputera PC
C1P1	PC-RAM	RAM para PC
C1P1	PC-PCB	PC Main Board
C1P1	PC-PCB	PC RAM
C1P1	PC-PCB	Płyta główna PC
C1P1	PC-PCB	Placa Mãe do PC
C1P1	PC-CASE	Case for PC
C1P1	PC-CASE	PC Case
C1P1	PC-CASE	Obudowa dla PC
C1P1	PC-CASE	Gabinete
C1P1	APR_PRD_Skateboard	
C1P1	APR_PRD_Skateboard	
C1P1	APR_PRD_Skateboard	
C1P1	PC-TOWER	PC
C1P1		

Figure 19 Report preview

The design view:

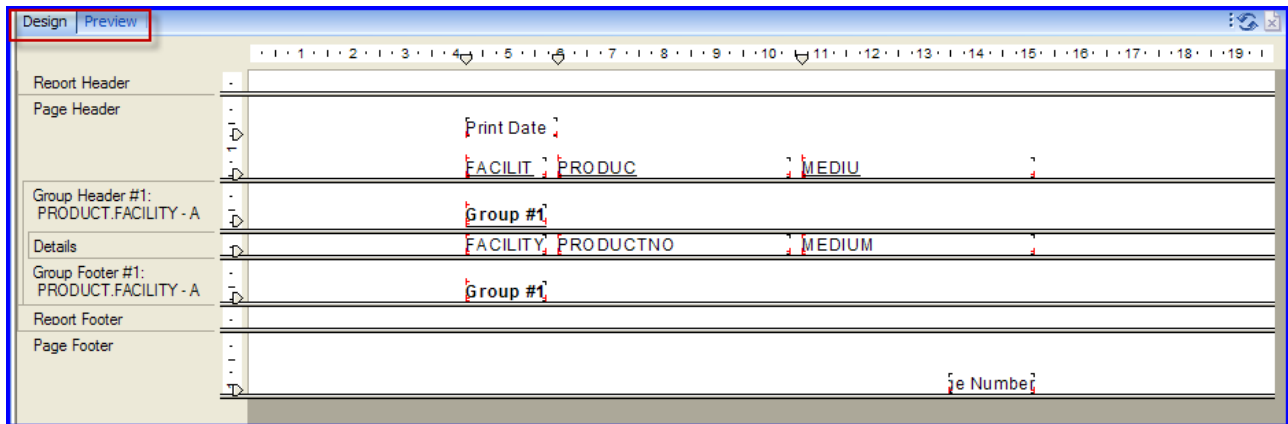


Figure 20 Design view

9. Finally, save as FlexNetListofProduct.rpt.

4.3 Modifying the Report

4.3.1 Adding Crystal Report Parameters

The parameters to be added:

- ▶ **ReportID**: String, Default value = 1
- ▶ **LanguageID**: Number, Default value = 1033 (US English)
- ▶ **TimeZoneID**: Number, Default value = 0

i The first two are mandatory for any kind of Crystal Reports with data translation.

The parameters can be added with the use of Field Explorer:

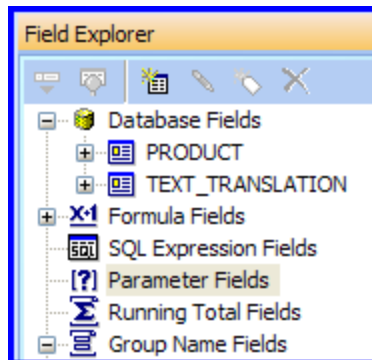


Figure 21 Field Explorer window

1. Add the ReportID parameter:

Edit Parameter: ReportID

Name: ReportID Type: String

List of Values: Static Dynamic

Value Field: (None) Description Field: (None)

Insert [X] [Up] [Down] Actions

Value	Description
Click here to add item	

Options:

Option	Setting
Prompt Text	Enter ReportID:
Prompt With Description Only	False
Default Value	1
Allow custom values	True
Allow multiple values	False
Allow discrete values	True

OK Cancel Help

Figure 22 ReportID parameter

2. Add the LanguageID parameter:

Edit Parameter: LanguageID

Name: LanguageID Type: Number

List of Values: Static Dynamic

Value Field: (None) Description Field: (None)

Options:

Option	Setting
Prompt Text	Enter LanguageID:
Prompt With Description Only	False
Default Value	1 033,00
Allow custom values	True
Allow multiple values	False
Allow discrete values	True

Figure 23 LanguageID parameter

3. Add the TimeZoneID parameter:

Edit Parameter: TimeZoneID

Name: TimeZoneID Type: Number

List of Values: Static Dynamic

Value Field: (None) Description Field: (None)

Options:

Option	Setting
Prompt Text	Enter TimeZoneID:
Prompt With Description Only	False
Default Value	0,00
Allow custom values	True
Allow multiple values	False
Allow discrete values	True

Figure 24 TimeZoneID parameter

After adding the three parameters, they should appear in the Field Explorer's parameter fields:

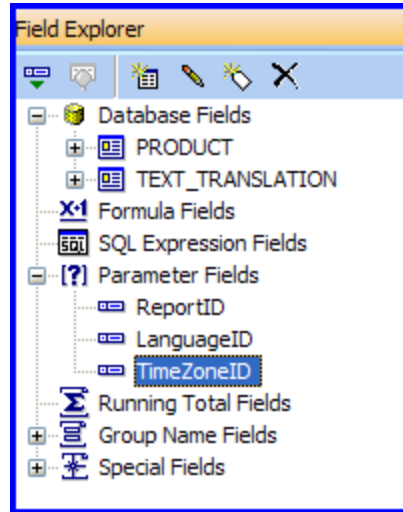


Figure 25 Parameters added in Field Explorer

4.3.2 Adding Formula Fields

1. Select **Formula Workshop** from the **Report** menu.

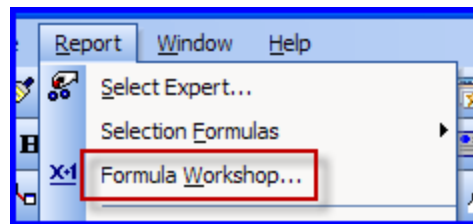


Figure 26 Selecting Formula Workshop

2. Add a new formula called f_Title.

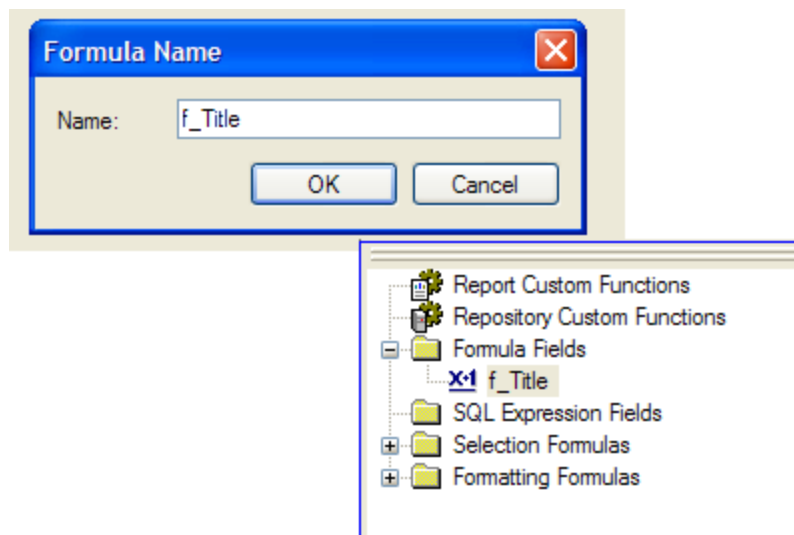


Figure 27 Adding a new formula

3. Add a title translation:

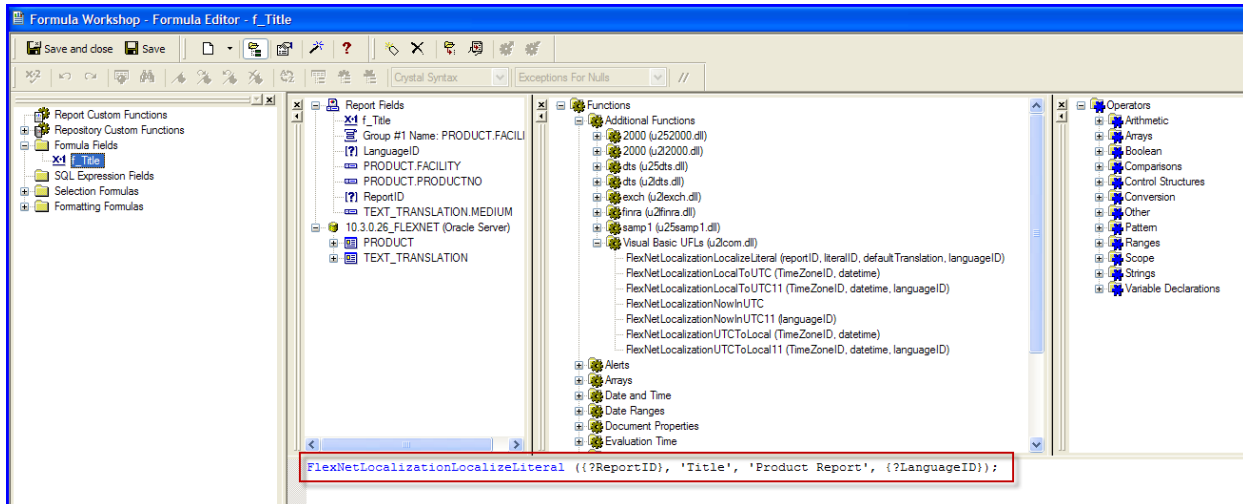


Figure 28 Editing the f_Title formula field

4. Double-click the library:

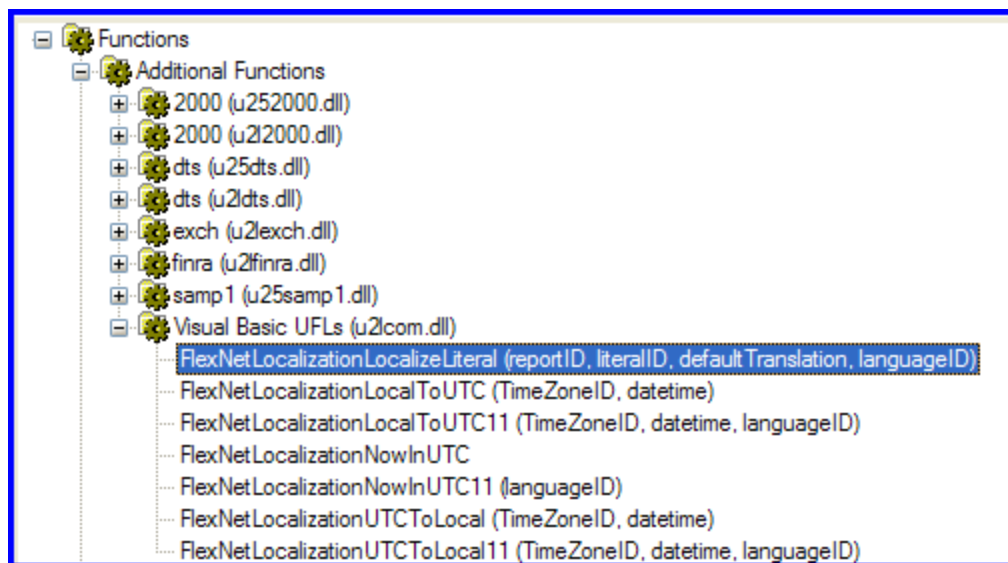


Figure 29 Library view

5. Next, modify the function parameters in the following way:

```
FlexNetLocalizationLocalizeLiteral ({?ReportID}, 'Title', 'Product Report',
{?LanguageID});
```

6. Save the formula.

7. Drag the f_Title data to the Report Header section.

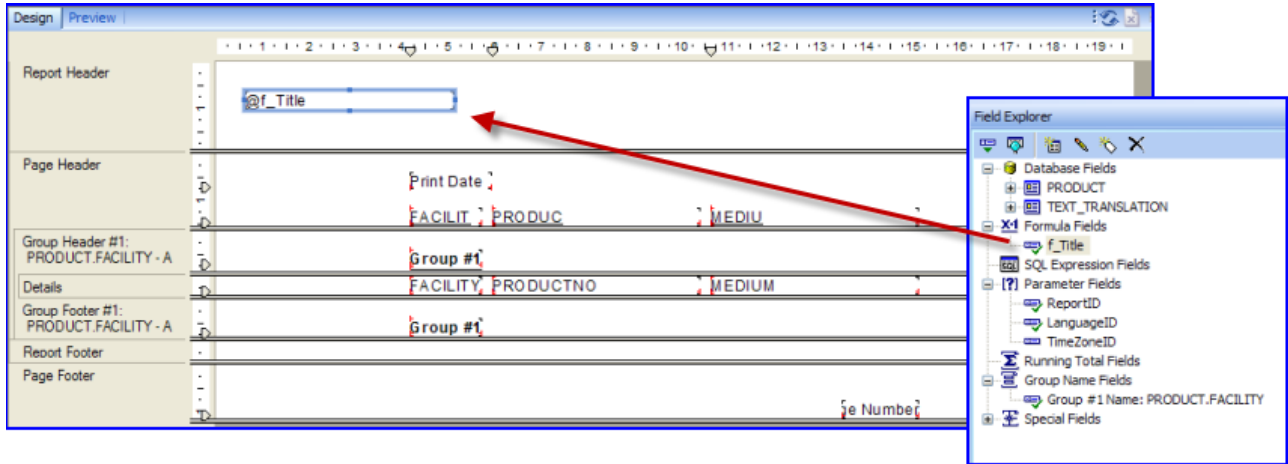


Figure 30 Placing the f_title field in report header

8. Format the title using the **Format Field** option from the right-click menu:

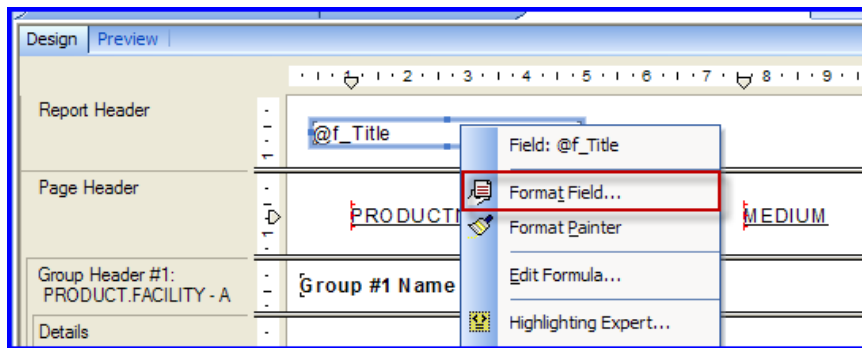


Figure 31 Format field option

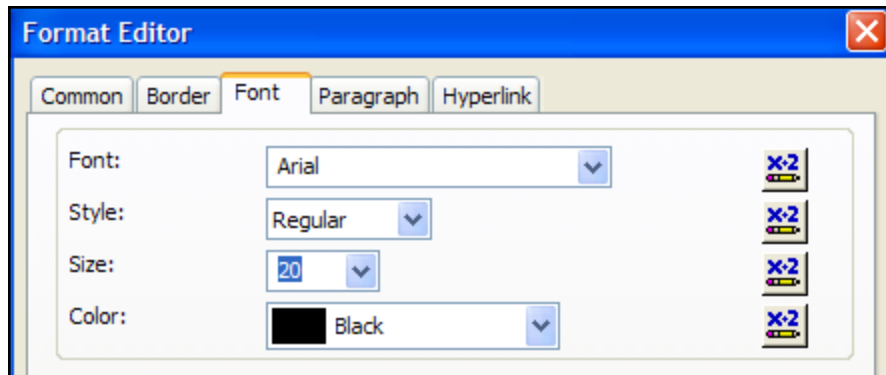


Figure 32 Font settings

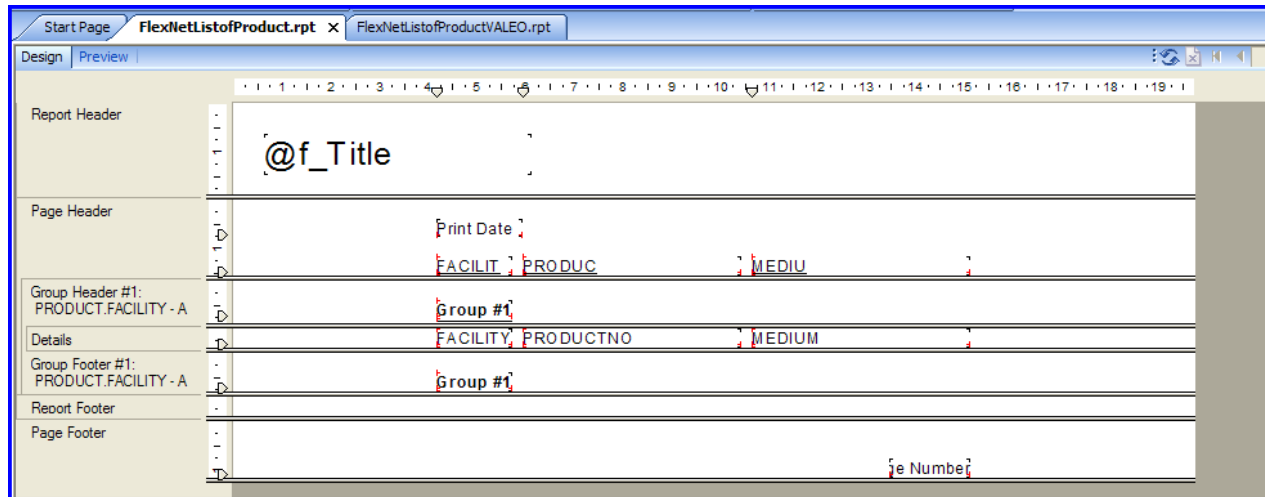


Figure 33 Formatted report design

9. Preview the report using the following values:

Enter ReportID:	ReportID
1	
Enter LanguageID:	LanguageID
1 033	

OK Annuler

Figure 34 Report values

After entering the connection information values, the report preview will be displayed:

Product Report		22/05/2008
<u>PRODUCTNO</u>	<u>MEDIUM</u>	
C1P1		
PC-TOWER	Tower PC	
APR_PRD_Skateboard		
PC-TOWER	Tower PC	
PC-TOWER	Gabinete PC	
PC-RAM	RAM for PC	
PC-RAM	PC RAM	
PC-RAM	RAM do komputera PC	

Figure 35 Report preview

i Crystal Reports returns all of the product codes found in the database in each language.

4.4 DELMIA Apriso Configuration

4.4.1 Adding DELMIA Apriso Parameters

The parameters to be added:

- ▶ **Report ID:** String, Default value = 1
- ▶ **Language ID:** Number, Default value = 1033 (English)

Create the **FlexNetListofProduct.xml** configuration file (within same folder as the RPT file) containing the following data:

```

<?xml version="1.0" encoding="utf-8" ?>
<ReportDefinition xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <ReportID>FlexNet.Reports.Production.FlexNetListofProduct</ReportID>
  <ReportName>FlexNetListofProduct</ReportName>
  <ReportVersion>1.0</ReportVersion>
  <Parameters>

    <Parameter>
      <Name>REPORTID</Name>
      <LiteralID>Report ID</LiteralID>
      <TypeString>Char</TypeString>
      <Required>>true</Required>
      <DefaultValue>1</DefaultValue>
    </Parameter>

    <Parameter>
      <Name>LANGUAGEID</Name>
      <LiteralID>Language ID</LiteralID>
      <TypeString>Integer</TypeString>
      <Required>>true</Required>
      <DefaultValue>1033</DefaultValue>
    </Parameter>

  </Parameters>
</ReportDefinition>

```

4.4.2 Moving RPT and XML Files

i By default, reports and labels are exported to the Report Export folder: <server name>\Program Files\Dassault Systemes\DELMIA Apriso 2021\WebSite\Portal\ReportExport\

Copy the FlexNetListofProduct.rpt and FlexNetListofProduct.xml files to the following directory on the server: <drive>\Program Files\Dassault Systemes\DELMIA Apriso 2021\Reports Definitions\

4.4.3 Loading Report Administrator

1. Load the Report Administrator by following this path: **Administration ▶ System Administration ▶ Report Administrator.**

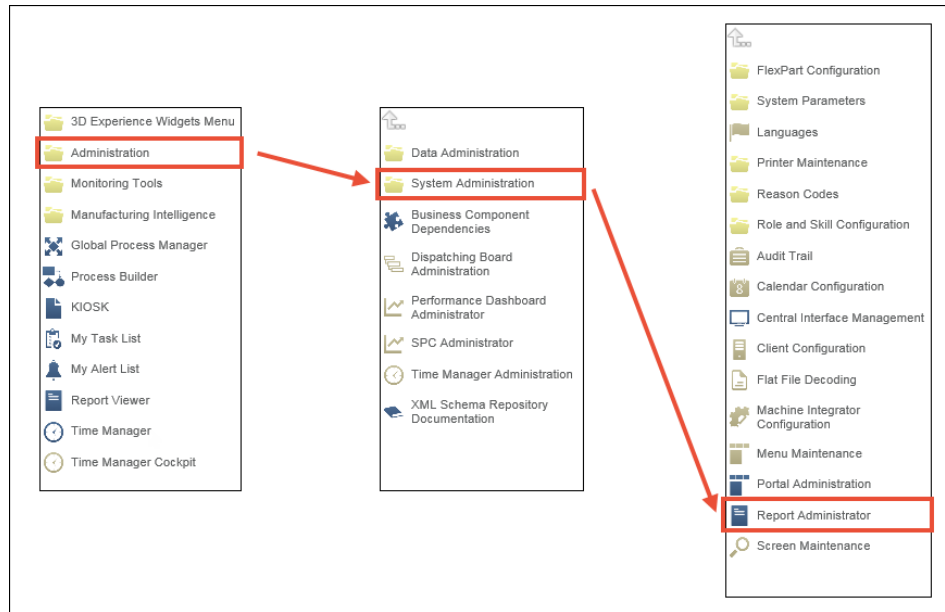


Figure 36 Loading Report Administrator

2. Choose **Reports** from the **Report type** drop-down list.

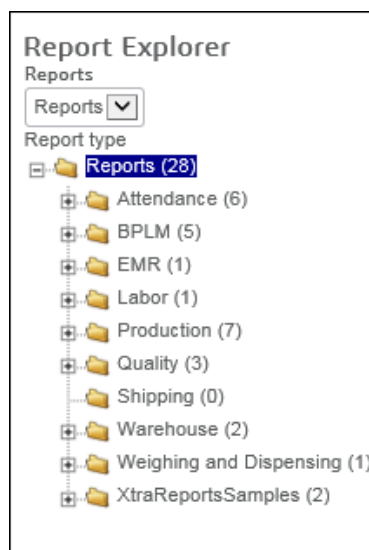


Figure 37 Report Explorer

3. Enter FlexNetListofProduct.xml in the **Report/label configuration file** field and FlexNetListofProduct.rpt in the **Report/label definition file** field, and then click **+** (Add).

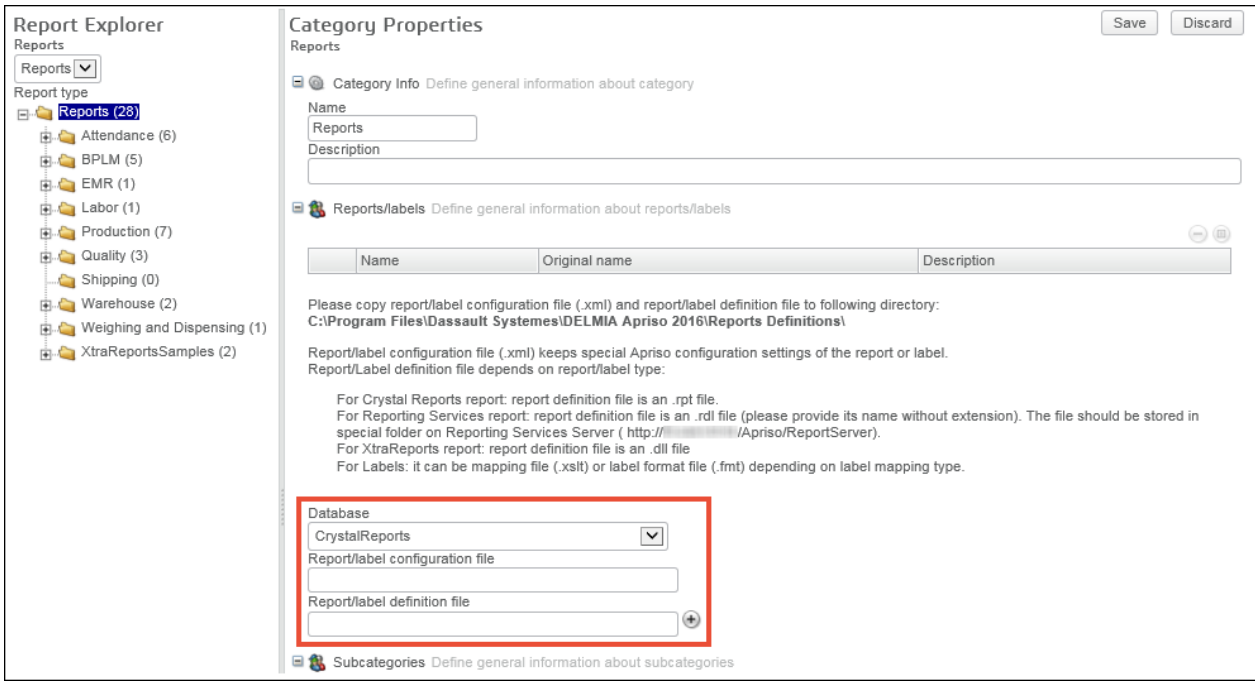


Figure 38 Adding report configuration and definition files

The report will be added to the system and displayed in the Report Explorer grid on the left.

4.4.4 Report Testing

1. Sign in to DELMIA Apriso and go to **Report Viewer**.

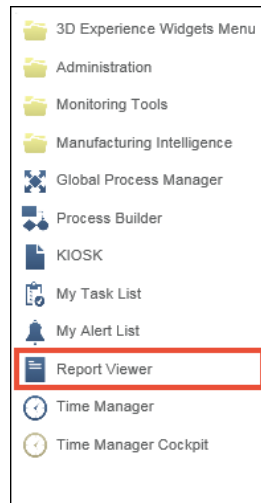


Figure 39 Opening Report Viewer

2. To view a report, select it with the use of Report Explorer on the left, choose the desired output format, and click **View** or **View in new window**:

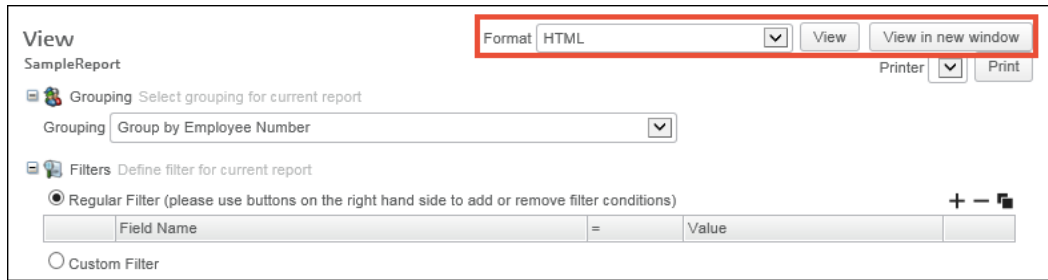


Figure 40 Viewing a report

4.4.5 Data Translation and Local Time Adjustment

1. To adjust items (list of products) on selected language (parameter), click **Report ▶ Selection Formulas ▶ Record**.

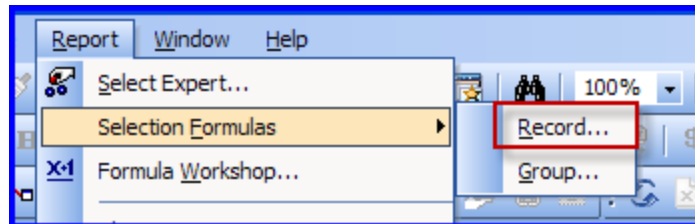


Figure 41 Selecting the record editor

2. Add a link between the XML language parameter and TEXT_TRANSLATION:

```
{TEXT_TRANSLATION.LANGUAGEID}={?LanguageID}
```

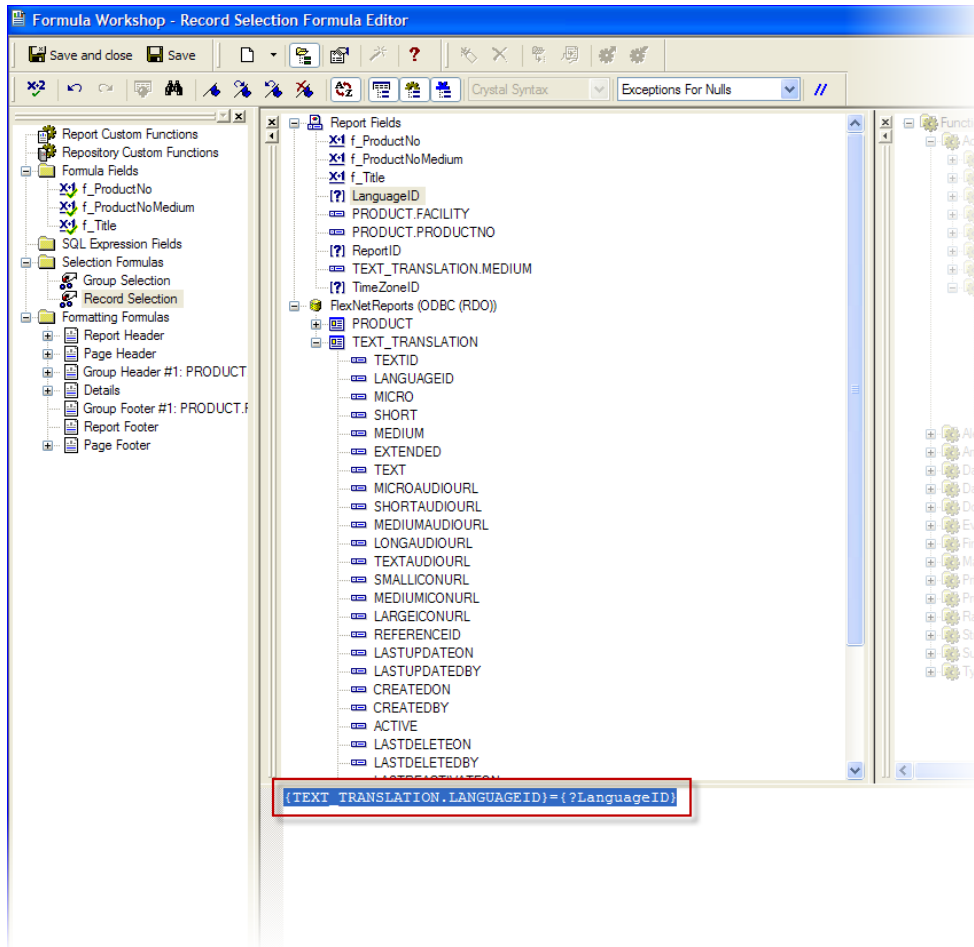


Figure 42 Record Selection Formula Editor

The last product update will be taken as an example to insert local time data (according to UTC time data).

3. Add the new **f_LastUpdate** formula field in Field Explorer:

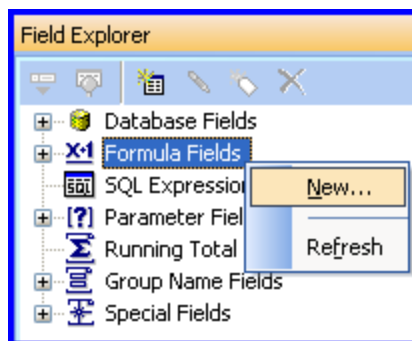


Figure 43 Adding a new formula field

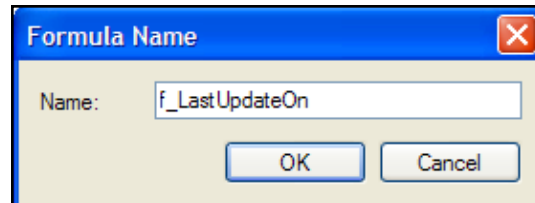


Figure 44 Entering the formula name

4. Enter the following formula text in Formula Editor:

```
FlexNetLocalizationUTCToLocal({?TimeZoneID},ToText({TEXT_TRANSLATION.LASTUPDATEON}));
```

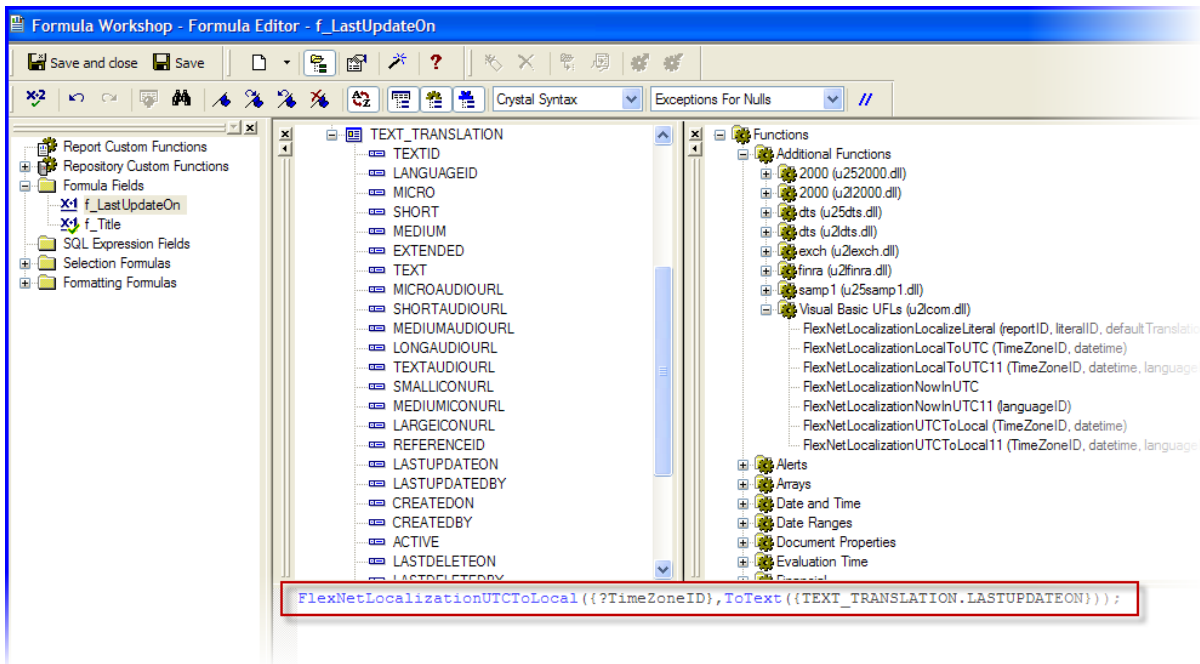


Figure 45 f_LastUpdateOn formula text

5. Drag the new formula field to the **Details** section:

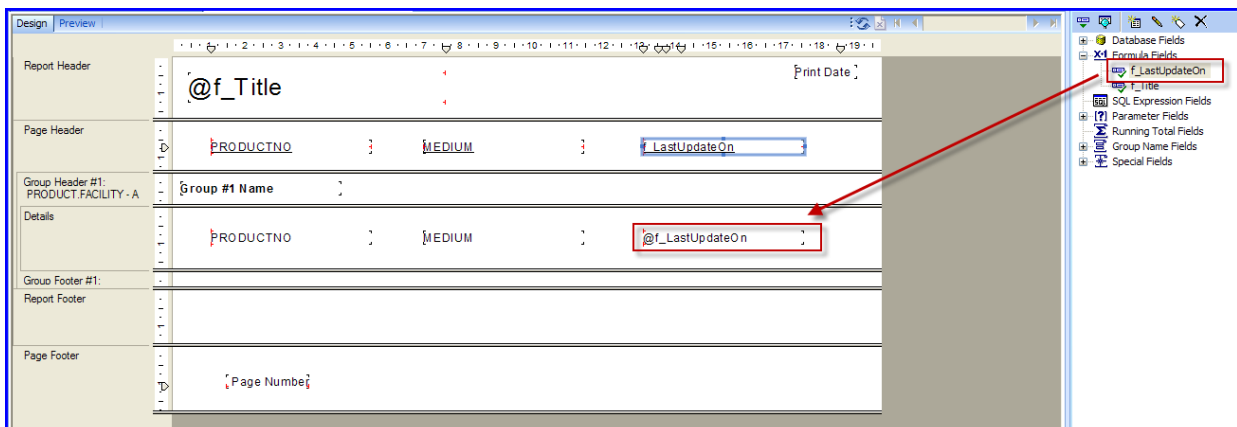


Figure 46 Adding a formula to the report section

6. Preview the report and specify the following parameters:

Figure 47 Specifying the report parameters

The final version of the report:

Product Report		23/05/2008
<u>PRODUCTNO</u>	<u>MEDIUM</u>	<u>f_LastUpdateOn</u>
C 1P1		
PC-RAM	RAM para PC	07/01/2005 06:45:19
PC-PCB	Placa Mãe do PC	07/01/2005 06:45:19
PC-TOWER	Gabinete PC	07/01/2005 06:45:20
PC-CASE	Gabinete	07/01/2005 06:45:19
APR_PRD_Skateboard		07/01/2005 06:45:18

Figure 48 Final version of the report

7. Compare the report with the original data recorded on the database server (at UTC time, in the Spanish language):

Row #	PRODUCTNO	FACILITY	LANGUAGEID	LASTUPDATEON	MEDIUM
1	PC-TOWER	C1P1	1046	7-janv.-2005 5:45:20	Gabinete PC
2	PC-RAM	C1P1	1046	7-janv.-2005 5:45:19	RAM para PC
3	PC-PCB	C1P1	1046	7-janv.-2005 5:45:19	Placa Mãe do PC
4	PC-CASE	C1P1	1046	7-janv.-2005 5:45:19	Gabinete
5	APR_PRD_Skateboard	C1P1	1046	7-janv.-2005 5:45:18	No_DescMed

Figure 49 Original data on the database server

5 Best Practices

5.1 Report Organization and Style

Reports are organized in subfolders:

- ▶ Attendance
- ▶ Production

When creating a new report, it is recommended that the templates delivered with the Web Reporting Framework installation be used. This is because all of the standard parameters, report headers, and page footers as well as the standard fonts used in the reports are already defined.

5.2 Standards

5.2.1 Paper Size and Fonts

Users should use **Letter Paper Size** in all their reports. The **Arial MS Unicode** font size 8, 9, or 10 is acceptable, based on the density of the report.

5.2.2 Header

The header shall be placed at the top of the report and contain two lines with the following format:

Line	Content	Style	Font
1	“Company:” + company name	Left, Justified	Arial MS Unicode Size 10
1	“Run Date:” + the date the report was run	Right, Justified	
2	“Report:” + report name	Left, Justified	

Table 2 Contents of report headers

Additionally, a header should be enclosed in a single-line box.

5.2.3 Footer

The footer shall be placed at the bottom of the page and contain:

Line	Content	Style	Font
1	Dassault Systèmes copyright information	Left, Justified	Arial MS Unicode Size 10
1	Page x of n	Center, Justified	
1	“Filtered” if a filter has been applied to the report or “Unfiltered” otherwise	Right, Justified	

Table 3 Contents of report footers

The footer should have a single-line top border.

5.3 Common Mistakes

5.3.1 Binding Reports Statically to One Database

Occasionally, a report can be displayed only when it is connected to the database that a user declared while creating the report by using the connection setting. This occurs because Crystal creates SQL queries that have a database name in them. To check if Crystal has created a correct SQL query, users can select Show SQL Query from the pop-up menu in the Database menu and then check if the database name is used in that query.

To correct this, users have to again set the location for the report (select Set Location from the Database menu), select a different ODBC connection, and then replace the old connection with the new one.

5.3.2 Using Views in Reports

When users use views in a report and they want to have the report displayed on the MS SQL server and Oracle databases, then they must create the report using an ODBC connection that is connected to an Oracle database. If the user has a report that uses views, then he or she must set the location for that report to an Oracle ODBC connection. If the user runs that report on an Oracle database, then it should also work fine on MS SQL.

5.3.3 Outer Joins

If a user links two tables using a field that could be null and he or she would like to have data returned if that field is null, then the left join instead of the inner join should be used. The reason for this is that, if an inner join is used and that particular field is null, then the query does not return a related record.

Another problem occurs when a user wants to filter the records using a field that could be null.

Example

A user has a FACILITY and TEXT_TRANSLATION left outer joined and wants to filter records using TEXT_TRANSLATION.LanguageID (TEXT_TRANSLATION.LanguageID = 1033). If the formula below is not used, the report will not display all of the records that have FACILITY without a related row in TEXT_TRANSLATION. Users should insert the following text into the selection formula:

```
IsNull( TEXT_TRANSLATION.LanguageID ) or TEXT_TRANSLATION.LanguageID = 1033
```

For other issues bounded with outer joins, refer to [5.3.4 Selection Formulas](#).

5.3.4 Selection Formulas

If users want to filter a particular field in their report and verify if it is null or equals a specified value, then the following should be used:

```
( IsNull( {TABLE.Field} ) or {TABLE.Field} = value )
```

instead of:

```
{TABLE.Field} = value
```

5.3.5 Using Sub-Reports

In specific circumstances when using sub-reports in a report, Apriso Report Viewer is unable to generate a report in any format other than HTML. This occurs when the generated sub-reports are supposed to be blank, but the Suppress Blank Sections option on a sub-report is not checked (or the formula that enables this option is not active).

If using sub-reports in a report, it is essential to link `LanguageID`, `ReportID`, or `TimeZoneID` to the sub-report parameters if they are needed in the sub-report.

i The parameter names in sub-reports cannot be the same as the parameter names in the main report.

i Sub-reports must use the same data source as the main report! Otherwise an error could occur during the report resave (first run on the DELMIA Apriso server), and the report would be unusable.

It is not possible to create a sub-report in a sub-report. Even attempting to add a report with sub-reports to another report will not be possible. In this case, the report must be composed in a different manner.

5.3.6 Page Setup – Optimization for Screen Display

In order for the reports to be correctly displayed in the Report Viewer, the No Printer option should be selected on the Page Setup screen ([Figure 50 No Printer option on the Page Setup screen](#)). Without this option set, the default printer setting will be used to render the report, which can result in displaying a trimmed report image ([Figure 51 Trimmed report view](#)).

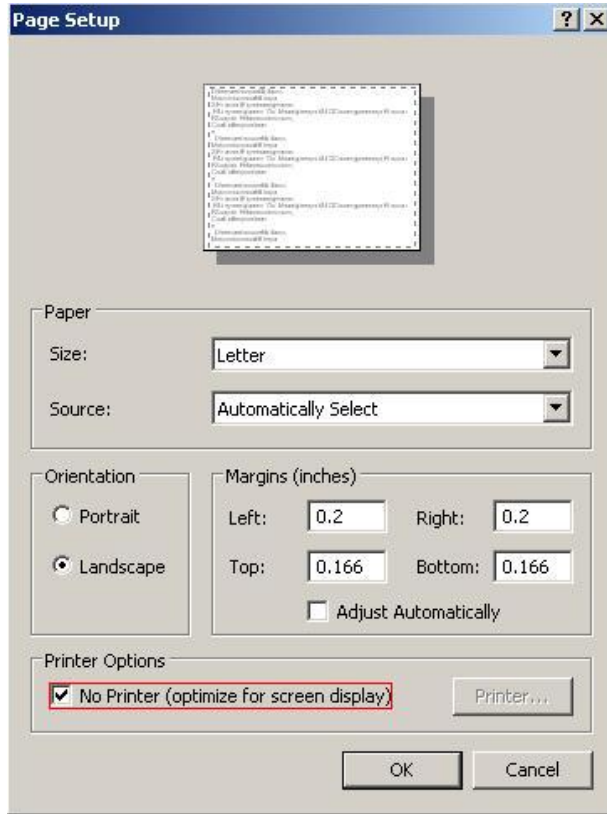


Figure 50 No Printer option on the Page Setup screen

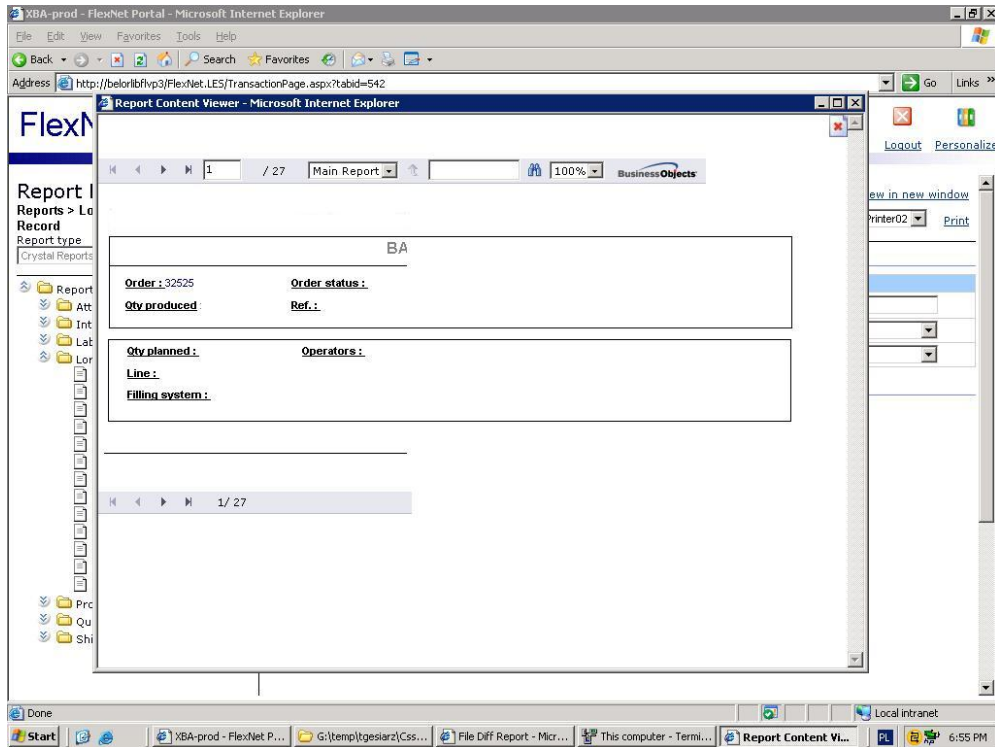


Figure 51 Trimmed report view

5.4 Optimizing Reports for Exporting to Excel

If the report is intended to be exported to Microsoft Excel data only, then certain optimization rules should be obeyed during development to ensure report compatibility and to maintain consistent report structures. There are no official guidelines on how to perform such optimization. However, many unofficial sources offer information on how to deal with the subject.

6 Known Issues

Issue with UFLs Being Unregistered upon Crystal Reports Upgrade (Third-Party Issue)

Problem:

During any Crystal Reports upgrade performed when DELMIA Apriso is already installed on the server, it may happen that the User Function Libraries (UFLs) are unregistered and stop functioning.

Solution:

1. Remove `FlexNet.BusinessRules.Printing.dll` from `<WindowsInstallationDirectory>\assembly` (by deleting it or selecting "uninstall" from the context menu on the file when open in Windows Explorer).
2. Execute the following from the command line (replacing the parts in "<>" with correct values):

```
<WindowsInstallationDirectory>\Microsoft.NET\Framework\v2.0.50727\]regasm.exe  
<FlexNetInstallationDirectory>\setup\gac\flexnet.businessrules.printing.dll
```

3. Add `Flexnet.businessrules.printing.dll` to `<WindowsInstallationDirectory>\assembly` (by dragging it from the Windows Explorer). The file is located as specified in step 2.

This will reregister the UFLs and resolve the problem.

Report Display Delay

Problem:

Reports executed in DELMIA Apriso Report Viewer in the HTML version are loaded and displayed after a 20-second delay.

Solution:

Add the development database machine name to the Windows hosts file (`<WindowsInstallationDirectory>\system32\drivers\etc\hosts`) and make it point to the production database machine on the production Web server. This should eliminate the 20-second delay as no name lookup occurs now.

Issues with ODBC connection on Oracle Database

Problem:

There may be problems with Crystal Reports 11 when using Oracle DBMS.

Solution:

1. Convert all of the report files to the new "Oracle Server" data source.
2. Modify the report connection parameters setting in DELMIA Apriso Configuration Manager (for details on Configuration Manager, refer to the [Configuration Manager Help](#)).
 - ▷ Former data source setting: <ODBC Source name>
 - ▷ Current data source setting: <TNSNAME>

Problem with Displaying Crystal Reports Localizable Label Text in the Report Viewer

Problem:

The Crystal Reports localizable label text is not displayed correctly in Report Viewer.

Solution:

Make the following changes in the server's control panel Regional Settings:

1. If the problem concerns East Asian languages, check "Install files for East Asian languages" on the Language tab.
2. Select the desired language in "Standards and formats" on the Regional Options tab (e.g., Chinese [PRC]).
3. Check "Default user account settings" on the Advanced tab.
4. Restart the computer.

If the issue persists, please install the **Arial Unicode MS** font on the DELMIA Apriso server and make sure that the **Default app pool** has access to this font.

Problem with Displaying Reports in Microsoft Word 2013 in Read Mode

Problem:

The problem may occur when using Report Viewer and displaying the Report in **Microsoft Word (doc)** Format. The report is displayed in Read Mode by default in Microsoft Word 2013 and it is not rendered correctly.

Solution:

Press ESC to exit Read Mode, the report will be displayed correctly. Alternatively, use a different version of Word.

Problem with displaying an Operation description in PDF reports (Japanese bold font)

Symptoms

The symptoms are visible only when the auto documentation feature is used in DELMIA Apriso Process Builder and the current language is Japanese. The descriptions of a Process and an Operation are not displayed correctly in the following reports: Detailed Operation

Definition and Detailed Process Definition. This only happens when the report is in the PDF format, the description contains a Unicode font (such as Japanese), and the text is bolded.

Problem Description

This is a third-party issue caused by bugs in the Crystal Reports framework.

Workaround

Modify the standard report so that it does not use the bold style.

Impacted Area

DELMIA Apriso Process Builder

Problem with Printing Crystal Reports from the Report Viewer When 'No Printer' Option is Enabled (Third-Party Issue)

Problem

When you enable the 'No Printer' option on the Page Setup screen in Crystal Reports for Visual Studio, even though the print job is sent to the printer, on certain printers the report may not be printed from the Report Viewer.

Solution

If the report is not printed, make sure the "No Printer" check box is unselected.

Blank Screen Appears when Trying to Access a Report

Problem

The reports are not displayed when trying to access them through a rewritten URL (i.e. Apriso/Portal/UI/<flexpart>).

Problem Description

This is a third-party issue caused by bugs in the Crystal Reports framework.

Workaround

Use `UIService.aspx?Alias=<flexpart>` URL.

7 References

Internal Documentation

1. ***Process of Translation Implementation Guide***

Provides an overview of the translation process as well as background information on translating DELMIA Apriso content and Cube literals. The guide focuses on the localization of operational data in DELMIA Apriso (through use of the DELMIA Apriso Localization Manager and DELMIA Apriso Translation Tool) as well as on translating the user interface data and MODEL data.

2. ***Configuration Manager Help***

Provides the background information necessary to use the DELMIA Apriso Configuration Manager tool and describes how to carry out common tasks.

3. ***Central Configuration Documentation***

Describes in detail all the keys of the Central Configuration (CC) file for DELMIA Apriso. Various sections group the keys for individual modules or distinct functional areas.

3DS Support Knowledge Base

If you have any additional questions or doubts not addressed in our documentation, feel free to visit the **3DS Support Knowledge Base** at <https://support.3ds.com/knowledge-base/>.