PRISMAdirect

Administration guide

Reporting

Copyright and Trademarks

Copyright

Copyright 2024 Canon Production Printing.

No part of this publication may be copied, modified, reproduced or transmitted in any form or by any means, electronic, manual or otherwise, without the prior written permission of Canon Production Printing. Illustrations and printer output images are simulated and do not necessarily apply to products and services offered in each local market. The content of this publication should neither be construed as any guarantee or warranty with regard to specific properties or specifications nor of technical performance or suitability for particular applications. The content of this publication may be subject to changes from time to time without notice.

CANON PRODUCTION PRINTING SHALL NOT BE LIABLE FOR ANY DIRECT, INDIRECT OR CONSEQUENTIAL DAMAGES OF ANY NATURE, OR LOSSES OR EXPENSES RESULTING FROM THE USE OF THE CONTENTS OF THIS PUBLICATION.

Language

Original instructions are in English.

Trademarks

Canon is a registered trademark of Canon Inc. ColorWave, DPConvert, PlotWave, PRISMA, PRISMAproduction, VarioPrint are trademarks or registered trademarks of Canon Production Printing Netherlands B.V.

Adobe, Acrobat, PostScript, and the Adobe logos are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Microsoft, Word, Excel, Office 365, Outlook, PowerPoint are trademarks or registered trademarks of Microsoft Corp. incorporated in the United States and/or other countries.

All other trademarks are the property of their respective owners and hereby acknowledged.

Data security

Introduction

Security is about safeguarding communication and data and keeping our online society and economy secure. It is critical for our customer's business and we take it seriously at Canon Production Printing. Security is incorporated in the early stages of our product development process. Our hardware and software products are developed according to industry security standard working methods, and equipped with the required features to protect our printing products and PRISMA workflow and application software against cyber security threats.

Policy / Guidelines / Rules

Canon's Secure Development Life Cycle (SDLC) process is an agile based development process that also models itself comparing with the National Institute of Standards and Technology (NIST) and Open Web Application Security Project (aka OWASP).

It uses the practice of sprints to take advantage of the continuous delivery and flexibility that an agile process enables.

Our back office support program (development of service tools and online services) is ISO/IEC 27001:2013 certified.

Regulation notices

For more detailed information on the security of our products, services and solutions please refer to your local Canon representative or to *https://cpp.canon/security* .

Contents

Contents

Chapter 1
Introduction

Software version

This documentation describes the functionality of PRISMAdirect v2.0.0.

Documentation

- The manuals available in the product might not contain the most recent help information. The latest manuals are available on *http://downloads.cpp.canon*
- The help files are available online

[Login]

Depending on the active options at the time of account creation, you may have to change your password at the first login.

Depending on the application configuration, it may be necessary to identify yourself. You have to enter your user name and password.

1. Enter your user name.
2. Enter your password.
3. Select [Remember my user name and password.] to log on next time, without entering your user name and password.
   You have to accept cookies from this application to support this option.
4. Click [Log in].
   For easy access, you are advised to add this page to your personal bookmarks.

Manage your profile

When you log on for the first time, the [Edit profile] dialog will appear. You must define your profile settings before you can continue. The email address is mandatory. Depending on the active options at the time of account creation, you may have to change your password at the first login.

The profile data of users can be stored on an LDAP server. For these users, the profile settings are automatically filled in with profile data from the LDAP server. These users cannot change their profile settings.

1. Click the pencil icon to edit your profile settings. The icon appears when you hover the mouse pointer over your username in the top-right corner of the workspace.
2. Change your profile.
3. Click [Save].

Chapter 2
Reporting

Configure the reporting environment

You have to configure the reporting environment before you can generate reports in workspace [Reporting].

1. You have to enable accounting if you want to collect payment information in your reports.
   • Go to: [Configuration]→[System]→[Accounting and reporting]→[Accounting]→[Accounting workflow].
2. You can generate and email the reports automatically.
   • Go to: [Configuration]→[System]→[Accounting and reporting]→[Reporting]→[Email accounting reports automatically].
3. You can select which reports you want to generate automatically.
   • Go to: [Configuration]→[System]→[Accounting and reporting]→[Reporting]→[Accounting reports].

   Dialog [Accounting reports] in workspace [Configuration] displays the report filenames (*.rpt). Workspace [Reporting] displays the report titles. When you create a new report, it is recommended that you use a similar report title and report filename. In this way, it is easier to identify the report between the two workspaces.
4. You have to configure the location to store the reports that are generated in workspace [Reporting].
   • Go to: [Configuration]→[System]→[Accounting and reporting]→[Reporting]→[Reports folder].

Download or save a generated report

You can generate the available reports manually. You have to prepare the reporting environment first. See *Configure the reporting environment* on page 12

Generate a report and download it

1. Select a report type.
   Each report type contains a number of items that you can use in a query.
2. Enable and configure the items that you want to use in the query.
3. Click [Generate report].
   The application generates the report and shows a preview of the report. You can export the report.
   1. Click [Export this report] in the preview of the report.
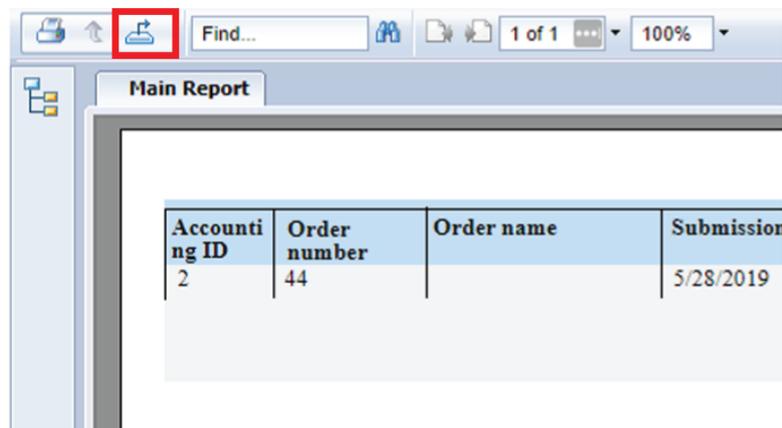
2. Select the file format.
3. Define which pages you want to export.
4. Click [Export].
   The report is generated and becomes available in the lower-left corner of the application.
   The report is stored in the [Downloads] folder of your computer.

Generate a report and save it on the PRISMAdirect server

1. Select a report type.
   Each report type contains a number of items that you can use in a query.
2. Enable and configure the items that you want to use in the query.
3. Click [Save report].
4. Select the file format.
   • The report is saved on the server. You have to configure the location to store
     the reports. Go to: [Configuration]→[System]→[Accounting and reporting]→[Reporting]→
     [Reports folder].
   • The old report will be overwritten when the [Overwrite existing report] option is enabled.
     For each report, this option is available in [Configuration] - [Accounting and reporting] -
     [Reporting] - [Accounting reports].

Download or save invoice information as an XML file

You can create XML files that contain the relevant invoice information. You can create an XML file for one invoice, or an XML file that contains the invoice information of multiple orders. You can use the XML files for integration with billing software. You have to configure the reporting environment before you can generate any invoice information. See *Configure the reporting environment* on page 12

1. Select [Single order invoice] to create an XML file for one invoice. Or select [Multiple orders invoice] to create an XML file that contains the invoice information of multiple orders.
2. Enable and configure the items that you want to use in the query.
3. Click [Search].
   The application shows for which orders invoice information is available.
   • [Download invoice]
     An XML file containing invoice information is generated and becomes available in the lower-left corner of the application. The XML file is stored in the [Downloads] folder of your computer.
   • [Save invoice]
     An XML file containing invoice information is generated and is saved on the server. You have to configure the location to store the XML files. Go to: [Configuration]→[System]→ [Accounting and reporting]→[Reporting]→[Reports folder].
     Each XML file has a unique timestamp.

Overview of the available invoice information

This topic describes the XML elements that contain invoice information of orders and jobs. Elements containing other elements are not described.

• **Information concerning your company in element** `<SupplierInformation>`

XML elements	Description
`<SupplierCompanyName>`	Your company name
`<SupplierAddress>`	The address of your company
`<SupplierVatID>`	The VAT percentage
`<SupplierCRN>`	Your [Sales tax ID number]
`<SupplierIBAN>`	Your International Bank Account Number (IBAN)
`<SupplierBIC>`	Your Bank Identifier Code (BIC)
`<SupplierPayment-Terms>`	The due date for payment of the invoice

• **Information concerning your customer in element** `<CustomerInformation>`

XML elements	Description
`<CustomerFirstName>`	First name
`<CustomerLastName>`	Last name
`<CustomerAddress>`	Address
`<CustomerCompanyName>`	Company name
`<CustomerVatID>`	The [Sales tax ID number] of your customer

XML elements	Description
`<CustomerDepartment>`	Department
`<CustomerEmailAd-dress>`	Email address

- **Cost and taxes in element** `<InvoiceAmounts>`

XML elements	Description
`<PayableAmount>`	Total price of the order
`<Shipping>`	The shipping cost
`<ManualLabor>`	The manual labor cost
`<Discount>`	The discount applied to the total price of the order The discount can be an amount or a percentage
`<TaxTotalAmount>`	The total amount of tax for the order
`<SubTotalAmount>`	The price of the order without tax

One or more taxes can be applied to an order. For each tax in element `<TaxAmount>`:

XML elements	Description
`<TaxAmount Category-Name="<custom tax code>">`	The custom tax code applied to the job
`<TaxableAmount>`	The price of the order without tax
`<TaxRate>`	The tax rate for the custom tax code
`<TaxAmount>`	The amount of tax

- **Information concerning the invoice in element** `<InvoiceInformation>`

XML elements	Description
`<InvoiceInformation>`	• type="InvoiceInformationJob The XML file contains the invoice information of one order • <no value> The XML file contains the invoice information of multiple orders
`<InvoiceNumber>`	The number of the invoice
`<InvoiceDate>`	The date that the invoice has been generated
`<Currency>`	The currency for the prices on the invoice
`<POnumber>`	The purchase order number This element receives a value when the order item [Purchase order number] is available in the product used to create the order with.
`<OrderNumber>`	The invoice applies to this order number

- [Single order invoice]
  Information for each job in the order in element `<LineItem>`
  A `<LineItem>` element contains the information of a job. An order can contain multiple jobs. The XML file contain a `<LineItem>` element for each job of the order.

XML elements	Description
`<Description>`	The name of the file of the job
`<ManualLabor>`	The manual labor cost
`<Discount>`	The discount applied to the total price of the order The discount can be an amount or a percentage
`<LineTotal>`	The price of the job including tax
`<LineTaxAmount>`	The total amount of tax for the job
`<JobNumber>`	The job number
`<UnitPrice>`	The `<LineTotal>` divided by `<Quantity>`
`<Quantity>`	The number of copies of the job
`<LineTaxRate>`	The tax rate applied to the job

- [Multiple orders invoice]
  Information for each order in element `<LineItem>`
  A `<LineItem>` element contains the information of an order. The XML file contain a `<LineItem>` element for each order.

XML elements	Description
`<ManualLabor>`	The manual labor cost
`<Discount>`	The discount applied to the total price of the order The discount can be an amount or a percentage
`<LineTotal>`	The price of the order including tax
`<LineTaxAmount>`	The total amount of tax for the order
`<POnumber>`	Purchase order number
`<OrderNumber>`	The order number
`<Shipping>`	The shipping cost

The available tables, stored procedures and views of the `PrintAccounting` **database**

When accounting is enabled, the `PrintAccounting` database of PRISMAdirect stores the accounting information when you finalize an order. In addition, the values of a number of ticket items are also stored, for example, information about the media and sheets. The tables of the `PrintAccounting` database and their content are described in detail in the section "The tables and their content" in this topic.

The `PrintAccounting` database offers a number of stored procedures and views.

Stored procedures and views

A stored procedure is prepared SQL code that you save so you can reuse the code over and over again. So, you can write a query and save it as a stored procedure. You can call the stored procedure to execute the SQL code that you saved as part of the stored procedure. In addition, you can pass parameters to the stored procedure. Now, the stored procedure acts based on the parameter values that were passed.

You cannot query the tables `OrderCustomItems` and `JobCustomItems` directly. You have to use the stored procedures that can query these two tables. You can query all other tables either directly or you can use stored procedures. The `PrintAccounting` database offers a number of default stored procedures:

1. `AnonymizeOrder`
   This stored procedure:
   - Anonymizes all finished/cancelled orders
   - Uses the following parameters:
     - `@NumberOfDays` - number of days (from configuration)
     - `@ProfileAttributes` - mapped profile items to be anonymized
     - `@JobCustomAttributes` - other items to be anonymized
     - `@AnonymousItem` - value for anonymized items (for instance, 'Anonymous User')
     - `@AnonymousEmail` - value for anonymized email
2. `getFinishingAction`
   This stored procedure:
   - Contains the finishing actions and corresponding `accounting_id`.
   - Uses the `@startDate` and `@endDate` as parameters.
   - Uses the view `uv_FinishingAction`.
3. `upsGetAllBundleItemsWithFilters`
   This stored procedure:
   - Contains all bundle items.
   - Uses the `@startDate` and `@endDate` as parameters.
4. `uspGetAccountingDataDump`
   This stored procedure:
   - Contains all information of the orders and jobs including the custom order items and the custom job items.
   - Uses the `@startDate` and `@endDate` as parameters.
   - Queries the tables: `OrderItems`, `OrderCustomItems`, `JobItems`, `JobCustomItems`, `User`, `MediaItem`, `FinishingAction`, `JobMediaSettings`, `ImageSettings`, `JobImageSettings`.
   - Uses the view `uv_JobImageSettings`.
5. `upsGetAllBundleItems`
   This stored procedure:
   - Contains all bundles information including bundle custom items.

- Uses the `@DumpDataInTempTable` parameter (true/false) to specify if the result will create the temporary table specified by parameter `@TempTable` for later selection of information.
- Queries the tables: `BundleItems`, `BundleCustomItems`.

6. `uspGetAllJobsItems`
   This stored procedure:
   - Contains all information of the jobs including custom job items.
   - Uses the `@DumpDataInTempTable` parameter (true/false) to specify if the result will create the temporary table specified by parameter `@TempTable` for later selection of information.
   - Queries the tables: `JobItems`, `JobCustomItems`.

7. `uspGetAllOrdersItems`
   This stored procedure:
   - Contains all information of the orders including the custom order items.
   - Uses the `@DumpDataInTempTable` parameter (true/false) to specify if the result will create the temporary table specified by parameter `@TempTable` for later selection of information.
   - Queries the tables: `OrderItems`, `OrderCustomItems`.

8. `upsGetBundleAndBundlePartsItems`
   This stored procedure:
   - Contains all bundle and bundle parts information.
   - Queries the tables `BundleItems`, `BundleCustomItems`, `JobItems`, `JobCustomItems`.

9. `upsGetJobAndBundleItems`
   This stored procedure:
   - Contains all jobs and bundles information.
   - Queries the tables `BundleItems`, `BundleCustomItems`, `JobItems`, `JobCustomItems`.

10. `upsGetOrderAndJobsBundleItems`
    This stored procedure:
    - Contains all order and jobs and bundle information including order and bundle custom items.
    - Queries the tables `OrderItems`, `JobItems`, `BundleItems`, `User`.

11. `upsGetOrderBundlesItems`
    This stored procedure:
    - Contains all orders and bundles information including order and bundle custom items.
    - Uses the `@DumpDataInTempTable` parameter (true/false) to specify if the result will create the temporary table specified by parameter `@TempTable` for later selection of information.
    - Queries the tables `OrderItems`, `OrderCustomItems`, `BundleItems`, `BundleCustomItems`, `User`.

12. `uspGetOrderJobsItems`
    This stored procedure:
    - Contains all information of the orders and jobs including the custom order items and the custom job items.
    - Uses the `@DumpDataInTempTable` parameter (true/false) to specify if the result will create the temporary table specified by parameter `@TempTable` for later selection of information.
    - Queries the tables: `OrderItems`, `OrderCustomItems`, `JobItems`, `JobCustomItems`, `User`.

13. `uspGetOrderJobsItemsDump`
    This stored procedure:
    - Contains all information of the orders and jobs including the custom order items and the custom job items.
    - Queries the tables: `OrderItems`, `OrderCustomItems`, `JobItems`, `JobCustomItems`, `User`.
    - Executes `uspGetOrderJobsItems` with parameters `@DumpDataInTempTable = True` and `@TempTable = ##ojdetails`. The name for `@TempTable` is hard-coded.

    📝 NOTE
    Both `uspGetOrderJobsItems` and `uspGetOrderJobsItemsDump` contain the same information.

A view is a SQL statement that is stored in the database with an associated name. A view is actually a composition of a table in the form of a predefined SQL query. A view can contain one or more rows from a table, or even all rows of a table. A view can be created from one or many tables which depends on the written SQL query to create a view.

The `PrintAccounting` database offers a number of default views:
1. `OrderBundlesItemsValues`
   This view:
   - Contains bundles information, including order information.
   - Queries the tables: `BundleItems, OrderItems, User`.
2. `OrderJobsItems`
   This view:
   - Contains all job items and all bundle items.
3. `OrderJobsItemsValues`
   This view:
   - Contains job items including order items and bundle items
   - Queries the tables: `JobItems, OrderItems, User, BundleItems`.
4. `uv_FinishingAction`
   This view:
   - Contains the finishing information and the key `OrderAccountingId`. The key `OrderAccountingId` is displayed as `FinishingAction_Accounting_Id`.
   - Queries the tables: `FinishingAction, JobItems, OrderItems`.
5. `uv_JobImageSettings`
   This view:
   - Contains the information of the images of the jobs and the key `JobAccountingId`. An image contains all digital information of the page, e.g. text, images, frames, etc.
   - Queries the tables: `JobImageSettings, ImageSettings`.

The tables and their content

The default name for the database containing the accounting information is "`PrintAccounting`". This database is created when you install PRISMAdirect. The following tables are always available:
- `FinishingAction`
- `ImageSettings`
- `InvoiceSettings`
- `JobCustomItems`
- `JobImageSettings`
- `JobItems`
- `JobMediaSettings`
- `MediaItem`
- `OrderCustomItems`
- `OrderItems`
- `Pricing`
- `Transactions`
- `User`
- `BundleItems`
- `BundleCustomItems`

Two additional tables can be created depending on the configuration of PRISMAdirect:
- `TransactionInfoes`
  Enable payment to create this table.
- `ShipmentPackagesHistories`
  Enable shipping to create this table.

To enable accounting, go to the [Configuration] workspace and select [System] - [Accounting and reporting] - [Accounting] - [Accounting workflow]. When you have enabled accounting, then the accounting information is stored in the database when you finalize an order. In addition, the values of a number of ticket items are also stored, for example, information about the media and sheets.
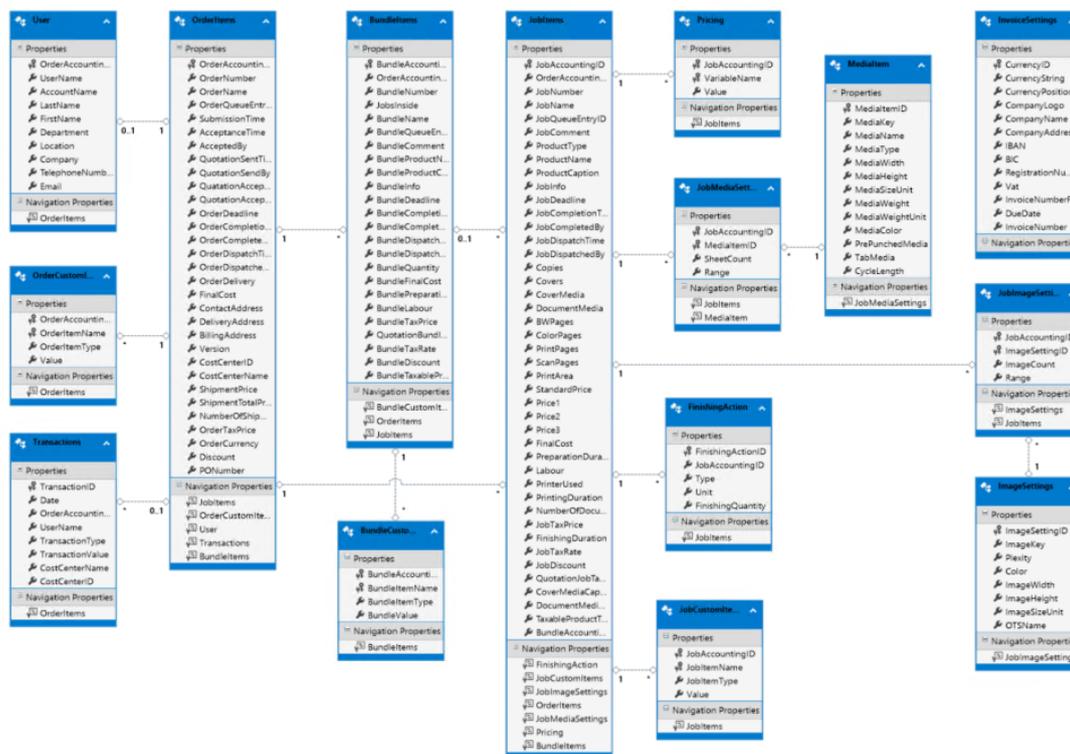
1. **Table:** `OrderItems`
   - This table contains a large part of the common order items. The `OrderCustomItems` table contains the remainder of the common order items and all custom order items.
   - One record per order.
   - You can either query this table directly or you can use a stored procedure that has access to this table.
   - The items `CostCenterName` and `CostCenterID` only receive a value when uniFLOW is paired with PRISMAdirect.

Column name	Item name	Item caption	Description
OrderAccountingID			Primary key
OrderNumber	OrderNumber	Order number	
OrderName	OrderName	Order name	
OrderQueueEntryID	OrderQEntryID		Non-public identification item. An order is identified by its `OrderQEntryID`, not by its `OrderNumber`.

Column name	Item name	Item caption	Description
`SubmissionTime`	`CreationDate`	Creation date	The date and time of order submission
`AcceptanceTime`	`AcceptanceTime`	Accepted on	Has a value only if item `AcceptedBy` has a value
`AcceptedBy`	`AcceptedBy`	Accepted by	The username of the operator who accepted the order
`QuotationSentTime`	`QuotationSentTime`	Quotation sent on	Has a value only if item `Quotation-SentBy` has a value
`QuotationSendBy`	`QuotationSentBy`	Quotation sent by	The username of the operator who sent the quotation
`QuatationAccep-tedTime`	`QuotationAccept-Time`	Quotation accepted on	Has a value only if item `QuotationAc-ceptedBy` has a value
`QuotationAccep-tedBy`	`QuotationAccep-tedBy`	Quotation accepted by	The username of the customer who accepted the quotation
`OrderDeadline`	`Date`	Delivery date	The requested delivery date and time of the order
`OrderCompletion-Time`	`FinalizedTime`	Finalized on	Has a value only if item `FinalizedBy` has a value
`OrderCompletedBy`	`FinalizedBy`	Finalized by	The username of the operator who finalized the order
`OrderDispatchTime`	`DispatchTime`	Dispatched on	Has a value only if item `DispatchedBy` has a value
`OrderDispatchedBy`	`DispatchedBy`	Dispatched by	The username of the operator who shipped the order
`OrderDelivery`			No value
`CostCenterID`	`CostCenterID`	Cost center ID	This item only receives a value when uniFLOW is paired
`CostCenterName`	`CostCenterName`	Cost center name	This item only receives a value when uniFLOW is paired

Column name	Item name	Item caption	Description
FinalCost	FinalCost	Total price	The total price of the order. A number without currency
ContactAddress	ContactAddress	Contact address	
DeliveryAddress	DeliveryAddress	Delivery address	
BillingAddress	BillingAddress	Billing address	
Version			Version of PRISMAdirect
ShipmentPrice	ShipmentPrice	Shipping price	The estimated price to ship the order. A number without currency. The price does not include tax. PRISMAdirect estimates the shipping price based on the requested delivery date upon order submission.
ShipmentTotal-Price	ShipmentTotal-Price	Total shipping price	The order manager determines the total shipping price. A number without currency. The price does not include tax.
NumberOfShipment-Packages	NumberOfShipment-Packages	Number of packages	The total number of packages required to send the order.
OrderTaxPrice	OrderTaxPrice	Tax	The amount of tax based on the estimated order price. A number without currency.
Discount			Sum of the value of item JobDiscount for all jobs
OrderCurrency	OrderCurrency	Currency	

2. **Table:** JobItems
   - This table contains a large part of the common job items. The JobCustomItems table contains the remainder of the common job items and all custom job items.
   - One record per job.
   - You can either query this table directly or you can use a stored procedure that has access to this table.

Column name	Item name	Item caption	Description
JobAccountingID			Primary key

Column name	Item name	Item caption	Description
OrderAccountingID			Foreign key of table OrderItems
JobNumber	JobNumber	Job number	
JobName	JobName	Job name	
JobQueueEntryID	QueueEntryID		Non-public identification item. A job is identified by its QueueEntryID, not by its JobNumber.
JobComment	Comments	Remarks	Remarks concerning the job.
ProductType			Job product type, for example, stationery
ProductName		Product name	Job product name, for example, monthly report
ProductCaption			Job product caption in the installation language of PRISMA-direct, for example, monthly report)
JobInfo	JobInfo	Job information	
JobDeadline	Date	Delivery date	The requested delivery date and time of the job
JobCompletionTime	FinalizedDate	Job marked 'Finalized' on	Has a value only if item FinalizedBy has a value
JobCompletedBy	FinalizedBy	Finalized by	The username of the operator who finalized the job
JobDispatchTime	DispatchTime	Dispatched on	Has a value only if item DispatchedBy has a value
JobDispatchedBy	DispatchedBy	Dispatched by	The username of the operator who shipped the job
Copies	Copies, or Quantity for stationery products	Copies / Quantity	
Covers	CoverPlace	Cover	None, FrontAndBack, Front, Back

Column name	Item name	Item caption	Description
CoverMedia	NewCoverMedia	Cover media	For example: ,S_A4,S_Plain, S_Red_249_37_94,1 20 g/ m2,False,False,Fa lse,1,False,, [0.1;0;Millime- ters],[2.5;0;Mil- limeters], [2.5;0;Millime- ters]
DocumentMedia	Media	Media	For example: ,S_A4,S_Plain, S_White_255_255_2 55,80 g/ m2,False,False,Fa lse,1,False,, [0.1;0;Millime- ters],[2.5;0;Mil- limeters], [2.5;0;Millime- ters]
BWPages	EstimatedBWPages	Number of B&W pages	
ColorPages	EstimatedColor- Pages	Number of color pages	
PrintPages			Not used
ScanPages			Not used
PrintArea			Not used
StandardPrice			Not used
Price1			Not used
Price2			Not used
Price3			Not used
FinalCost	FinalCost	Total price	The total price of the job. A number without currency
PreparationDura- tion	PrepareTime	Time to prepare job	Numeric value
FinishingDuration	FinishingTime	Time to finish job	Numeric value
Labor	ManualLabor	Labor cost	The operator types the number of hours, or the cost, to produce the job. Numeric value
PrinterUsed	PrinterName	Printer name	Name of the printer used to print the job

Column name	Item name	Item caption	Description
`PrintingDuration`	`PrintingDuration`	Time to print job	Numeric value
`NumberOfDocument-Sets`			`NumberOfCopies-PerSet` `* NumberOfBussi-nessCardsPerSet`
`JobTaxPrice`	`JobTaxPrice`	Tax	The amount of tax based on the estimated job price. A number without currency.
`QuotationJobTax-Price`	`QuotationJobTax-Price`	Tax	The amount of tax based on the quotation for the job. A number without currency.
`JobTaxRate`	`JobTaxRate`	Tax rate (%)	Numeric value
`JobDiscount`	`JobDiscount`	Discounts	The discount for the job based on the estimated job price. A number without currency.
`TaxableProduct-Type`	`TaxableProduct-Type`		Taxable Product Type
`BundleAccounting-ID`			`BundleItems` foreign key.

3. **Table:** `BundleItems`
   - This table important bundle information, one record per bundle job. All other ticket items like orientation and binding are saved in `BundleCustomItems` table
   - One record per job.
   - You can either query this table directly or you can use a stored procedure that has access to this table.

Column name	Bundle ticket item name	Bundle ticket item caption	Description
`BundleAccounting-ID`			Primary key
`OrderAccountingID`			Foreign key of table `OrderItems`
`BundleNumber`	`Number`	Bundle number	
`JobsInside`			Number of jobs inside of the bundle
`BundleName`	`Name`	Bundle name	
`BundleQueueEn-tryID`	`QueueEntryID`		Internal identity.

Column name	Bundle ticket item name	Bundle ticket item caption	Description
`BundleComment`	`Comments`	Remarks	Remarks concerning the bundle.
`BundleProductName`		Product name	Bundle product name, for example, Bundle-Product
`BundleProductCaption`		-	Bundle product caption in the installation language of PRISMA-direct, for example, BundleProduct)
`BundleInfo`	`Info`	Bundle information	
`BundleDeadline`	`Date`	Delivery date	The requested delivery date and time of the bundle.
`BundleCompletionTime`	`FinalizedDate`	Bundle marked 'Finalized' on	Has a value only if item `FinalizedBy` has a value
`BundleCompletedBy`	`FinalizedBy`	Finalized by	The username of the operator who finalized the bundle
`BundleDispatchTime`	`DispatchTime`	Dispatched on	Has a value only if item `DispatchedBy` has a value
`BundleDispatchedBy`	`DispatchedBy`	Dispatched by	The username of the operator who shipped the bundle
`BundleQuantity`	`Quantity`	Quantity	Quantity of the bundle
`BundleFinalCost`	`FinalCost`	Total price	Total price of the bundle without currency
`BundlePreparationDuration`	`PrepareTime`	Time to prepare bundle	Numeric value that indicates the time to prepare bundle
`BundleLabour`	`ManualLabor`	Labor cost	Number of hours or Cost used for the production of the corresponding bundle. It is a numeric value, the operator chooses what to use
`BundleTaxPrice`	`TaxPrice`	Tax	The tax value for bundle estimated price, without currency

Column name	Bundle ticket item name	Bundle ticket item caption	Description
QuotationBundle-TaxPrice	QuotationBundle-TaxPrice	Tax	The tax value for bundle total price, without currency
BundleTaxRate	TaxRate	Tax rate (%)	Numeric value representing the tax rate for bundle estimated price
BundleDiscount	Discount	Discounts	The discount value for bundle estimated price, without currency
BundleTaxableProductType	TaxableProduct-Type	Taxable Product Type	Bundle Taxable Product Type

4. **Table:** OrderCustomItems
   - This table contains the remainder of the common order items that are not in the OrderItems table. It also contains all custom order items.
   - One record per custom item.
   - You cannot query this table directly. You have to use a stored procedure that has access to this table.

Column name	Item name	Description
OrderAccountingID		Foreign key of table OrderItems
OrderItemName		Name of the custom order item
OrderItemType		For example: nvarchar, datetime, real. The types are stored as SQL item types.
Value		The item value is always stored as string.

5. **Table:** JobCustomItems
   - This table contains the remainder of the common job items that are not in the JobItems table. It also contains all custom job items.
   - One record per custom item.
   - You cannot query this table directly. You have to use a stored procedure that has access to this table.

Column name	Item name	Description
JobAccountingID		Foreign key of table JobItems
JobItemName		Name of the custom job item
JobItemType		For example: nvarchar, datetime, real. The types are stored as SQL item types.

Column name	Item name	Description
Value		The item value is always stored as string.

6. **Table:** BundleCustomItems
   - This table contains the remainder of the common job items that are not in the BundleItems table. It also contains all custom job items.
   - One record per custom item.
   - You cannot query this table directly. You have to use a stored procedure that has access to this table.

Column name	Item name	Description
BundleAccountingID		Foreign key of table BundleItems
BundleItemName		Name of the custom bundle item
BundleItemType		For example: nvarchar, datetime, real. The types are stored as SQL item types.
BundleValue		The item value is always stored as string.

7. **Table:** User
   - This table contains the information of the customer for a given order.
   - One record per customer.
   - You can either query this table directly or you can use a stored procedure that has access to this table.

Column name	Item name	Description
OrderAccountingID		Foreign key of table OrderItems
UserName	UserId	Name of the customer
AccountName	Account	
LastName	LastName	
FirstName	FirstName	
Department	Department	
Location	Location	
Company	Company	
TelephoneNumber	TelephoneNumber	
Email	EmailAddress	

8. **Table:** Transactions
   - This table contains the transactions per cost center.
   - One record per cost center.
   - You can either query this table directly or you can use a stored procedure that has access to this table.

Column name	Item name	Description
TransactionID		Primary key
Date		Date of transaction
OrderAccountingID		Foreign key of table OrderItems
UserName	UserId	Name of the customer
TransactionType		set / spent
TransactionValue	FinalCost	The total price of the order. A number without currency
CostCenterName	CostCenterName	
CostCenterID	CostCenterId	

9. **Table:** InvoiceSettings
   - This table contains information that is used for an invoice.
   - One record per invoice.
   - You can either query this table directly or you can use a stored procedure that has access to this table.

Column name	Item name	Description
CurrencyID		Primary key
CurrencyString		EUR, CHF, …
CurrencyPosition		TRUE if the currency is displayed in front of the amount
CompanyLogo		Company logo image
CompanyName		Company name
CompanyAddress		Company address

10. **Table:** Pricing
   - This table contains all items that are used in the formula, along with the calculated price for each item.
   - One record per item.
   - You can either query this table directly or you can use a stored procedure that has access to this table.

Column name	Item name	Description
JobAccountingID		Foreign key of table JobItems
VariableName		Formula variable name. For example, 800BW, Copies, FinishingPricing, DocumentMediaWeight.
Value		The price for the item as numeric value.

11. **Table:** FinishingAction
   - This table contains information about the selected finishing for a job, for example, binding, folding, punching.

- One record per job.
- You can either query this table directly or you can use a stored procedure that has access to this table.
- You can prepare a PDF file in PRISMAprepare. When you close PRISMAprepare, the values of the prepared items are stored in the PDF file itself. When you finalize the order, the finishing items are read from the PDF file and stored in the database. In this case, the values of the finishing items of PRISMAdirect are ignored. Reason: the values of the finishing items of PRISMAprepare are considered to be more correct.

Column name	Item name	Description
FinishingActionID		Primary key
JobAccountingID		Foreign key of table `JobItems`
Type		`TriFoldIn`, `SaddleStitchOrStapleSpine`, `Staples_4`, `Holes_3`, etc
Unit		`Copy`, `Set`, `Sheet`, or empty if the unit is unknown
FinishingQuantity		Numeric value

12. **Table:** `ImageSettings`
    - This table contains the unique image settings of all jobs. The database will not contain double entries for an image. An image contains all digital information of the page, e.g. text, images, frames, etc.
    - One record per image.
    - You can either query this table directly or you can use a stored procedure that has access to this table.
    - You can prepare a PDF file in PRISMAprepare. When you close PRISMAprepare, the values of the prepared items are stored in the PDF file itself. When you finalize the order, the image settings are read from the PDF file and stored in the database. In this case, the image settings used by PRISMAdirect are ignored.
      Reason: in PRISMAprepare you can use multiple image sizes in a file, for example, A4 and A3 images. In PRISMAdirect only one image size is possible. For correct accounting, all sizes must be taken into account.

Column name	Item name	Description
ImageSettingID		Primary key
ImageKey		This key is created by appending the five image items in combination with the "," separator. For example: `Duplex,Color,2100,2970,TenthOfMillimeter`
Plexity		`Simplex`, `Duplex`
Colour		`BlackAndWhite`, `Color`, `Mixed`
ImageWidth		
ImageHeight		

Column name	Item name	Description
ImageSizeUnit		For example, `Millimeters`, `Centimeters`, `TenthOfMillimeter`, `HundredthOfMillimeter`, `ThousandthOfMillimeter`
OTSName		Not used

13. **Table:** `JobImageSettings`
    - This table contains the number of images per job. An image contains all digital information of the page, e.g. text, images, frames, etc.
    - One record per job.
    - You can either query this table directly or you can use a stored procedure that has access to this table.

Column name	Item name	Description
JobAccountingID		Foreign key of table `JobItems`
ImageSettingID		Foreign key of table `ImageSettings`
ImageCount		Number of pages for each unique image
Range		Not used

14. **Table:** `MediaItem`
    - This table contains the unique media of all jobs. The database will not contain double entries for a media.
    - One record per media.
    - You can either query this table directly or you can use a stored procedure that has access to this table.
    - The following media items will be stored:
      1. When no PDF file is available, the media as defined in the ticket of PRISMAdirect is stored.
      2. When a PDF file is available, but the file is not prepared in PRISMAprepare, then:
         The media as defined in the ticket of PRISMAdirect is stored. The media used by the PDF file is ignored.
      3. When the PDF file is prepared in PRISMAprepare, then:
         You can prepare a PDF file in PRISMAprepare. When you close PRISMAprepare, the values of the prepared items are stored in the PDF file itself. When you finalize the order, the media items of PRISMAprepare are read from the PDF file and stored in the database. In this case, the media items used by PRISMAdirect are ignored.

Column name	Item name	Description
MediaItemID		Primary key
MediaKey		For example, `595x842 points`, `Tab(False)`, `Cyclic(1)`, `120 g/m2`, `Plain`, `Red`
MediaName		Media name
MediaType		For example, `Plain`, `Transparent`, `Envelope`, etc

Column name	Item name	Description
MediaWidth		Numeric value
MediaHeight		Numeric value
MediaSizeUnit		For example, `points`, `TenthOf-Millimeter`
MediaWeight		Numeric value
MediaWeightUnit		For example, g/m2
MediaColor		For example, White, Red
PrePunchedMedia		Media prepunched numbers (numeric value)
TabMedia		`True` or `False`
CycleLength		Numeric value

15. **Table:** `JobMediaSettings`
    - This table contains the number of sheets per job.
    - One record per job.
    - You can either query this table directly or you can use a stored procedure that has access to this table.

Column name	Item name	Description
JobAccountingID		Foreign key of table `JobItems`
MediaItemID		Foreign key of table `MediaItem`
SheetCount		Number of media sheets
Range		Not used

16. **Table:** `TransactionInfoes`
    - PRISMAdirect creates this table when payment is enabled. This table contains the information concerning the payment transactions.
    - One record per transaction.
    - You can either query this table directly or you can use a stored procedure that has access to this table.

Column name	Description
Id	Primary key
OrderNumber	Order number
OrderName	Order name
OrderId	Non-public identification item. An order is identified by its `OrderId`, not by its `OrderNumber`.
Provider	The payment provider
TransactionId	The payment transaction ID received from the provider
Timestamp	The timestamp of the payment transaction

Column name	Description
Status	Information about the payment transaction. For example, start, end, cancel, refund.
Amount	The price involved in the payment transaction
Currency	The currency used in the payment transaction
PayerName	Payer name
PayerEmailAddress	Payer email address
PayerPhoneNumber	Payer telephone number
PayerShippingAddress	Payer address
PayerStatus	Payer status received from the payment provider. It indicates whether the payment provider has verified the bank account of the customer, or not. For example: verified, unverified
PayerIP	IP host address of the payer
ServerName	Name of the computer where PRISMAdirect is installed
ServerIP	IP address of the computer where PRISMAdirect is installed
Tax	The tax involved in the payment transaction. The tax is included in the amount.

17. **Table:** `ShipmentPackagesHistories`
    - PRISMAdirect creates this table when shipment is enabled. This table contains the information concerning the sent packages.
    - One record per package.
    - You can either query this table directly or you can use a stored procedure that has access to this table.

Column name	Description
Id	Primary key
OrderNumber	Order number
OrderName	Order name
Description	Description of the package description. This field is mandatory for international shipping.
ShipmentProvider	The shipping provider
DeliveryDateUTC	The timestamp of the shipping transaction
Length	Length of the package
Width	Width of the package
Height	Height of the package
Unit	Unit of the package size in [inch] or [cm]
WeightValue	Weight of the package
WeightUnit	Unit of the package weight in [lb] or [kg]
ShipmentAddressFrom	The shipping address of the sender

Column name	Description
ShipmentAddressDestination	The shipping address of the recipient
InsuranceValue	Insurance value of the package
InsuranceCurency	The currency used for the insurance value of the package
ShipmentCostValue	The shipping cost for the package
ShipmentCostCurency	The currency used for the shipping cost

Overview of the default reports available in the [Reporting] workspace

1. **Available job items**
   This report contains the job information available in the `JobItems` table. The results of the report can be filtered by:
   • Job number
   • Job name
   • Product type
   • Product name
   • Total price
   • Deadline date
   • Completion date

2. **Available bundle items**
   This report contains the job information available in the `BundleItems` table. The results of the report can be filtered by:
   • Job number
   • Job name
   • Product name
   • Total price
   • Deadline date
   • Completion date

3. **Available order items**
   This report contains the order information available in the `OrderItems` table. The results of the report can be filtered by:
   • Order number
   • Order name
   • User name
   • Cost center name
   • Total price
   • Submission date
   • Completion date

4. **Color vs. Black and White pages per Cost Center**
   This report contains the total number of color pages and total number of black and white pages per cost center. It uses fields from the `JobItems` and `OrderItems` tables. The results of the report can be filtered by:
   • Cost center name
   • Submission date

5. **Color vs. Black and White pages per Day**
   This report contains the total number of color pages and total number of black and white pages per day. It uses fields from the `JobItems` and `OrderItems` tables. The results of the report can be filtered by:
   • Submission date

6. **Color vs. Black and White pages per Department**
   This report contains the total number of color pages and total number of black and white pages per department. It uses fields from the `JobItems` and `OrderItems` tables. The results of the report can be filtered by:
   • Department
   • Submission date

7. **Color vs. Black and White pages per Month**

This report contains the total number of color pages and total number of black and white pages per month. It uses fields from the `JobItems` and `OrderItems` tables. The results of the report can be filtered by:
- Submission date

8. **Color vs. Black and White pages per User**
   This report contains the total number of color pages and total number of black and white pages per customer. It uses fields from the `JobItems`, `OrderItems` and `User` tables. The results of the report can be filtered by:
   - Submission date

9. **Color vs. Black and White pages per Week**
   This report contains the total number of color pages and total number of black and white pages per week. It uses fields from the `JobItems` and `OrderItems` tables. The results of the report can be filtered by:
   - Submission date

10. **Cost report**
    This report contains the cost per order. It uses fields from the `OrderItems` and `User` tables. The results of the report can be filtered by:
    - Order number
    - User name
    - Cost center name
    - Submission date
    - Completion date

11. **Delivered packages**
    This report contains the shipping packages per order. It uses fields from the `ShipmentPackagesHistory` table. The results of the report can be filtered by:
    - Order number
    - Delivery date

12. **Jobs printed before vs. after the Delivery Date**
    This report contains the number of jobs printed before the delivery date and the number of jobs printed after the delivery date. It uses fields from the `JobItems` and `OrderItems` tables. The results of the report can be filtered by:
    - Submission date

13. **Number of orders and jobs per Day**
    This report contains the total number of orders and total number of jobs per day. It uses fields from the `JobItems` and `OrderItems` tables. The results of the report can be filtered by:
    - Submission date

14. **Number of orders and jobs per Month**
    This report contains the total number of orders and total number of jobs per month. It uses fields from the `JobItems` and `OrderItems` tables. The results of the report can be filtered by:
    - Submission date

15. **Number of orders and jobs per Week**
    This report contains the total number of orders and total number of jobs per week. It uses fields from the `JobItems` and `OrderItems` tables. The results of the report can be filtered by:
    - Submission date

16. **Overview of products of type Stationery**
    This report contains information about the stationery jobs. It uses fields from the `JobItems`, `OrderItems` and `User` tables. The results of the report can be filtered by:
    - Submission date

17. **Printing & price**
    This report contains the printing and price information per job. It uses fields from the `JobItems` and `OrderItems` tables, and the `uspGetAllJobsItems` stored procedure. The results of the report can be filtered by:
    - Job number

18. **Shipping cost report**

   This report contains the shipping cost per order. It uses fields from the `OrderItems` and `User` tables. The results of the report can be filtered by:
   • Order number
   • Submission date
   • Completion date
   • User name

19. **Total cost and number of orders per Cost Center**

   This report contains the total cost and total number of orders per cost center. It uses fields from the `OrderItems` table. The results of the report can be filtered by:
   • Submission date
   • Cost center name

20. **Total number of pages per Cost Center**

   This report contains the total number of pages per cost center. It uses fields from the `JobItems` and `OrderItems` tables. The results of the report can be filtered by:
   • Submission date
   • Cost center name

21. **Total number of pages per department**

   This report contains the total number of pages per department. It uses fields from the `JobItems`, `OrderItems` and `User` tables. The results of the report can be filtered by:
   • Submission date
   • Department

22. **Total number of pages per user**

   This report contains the total number of pages per customer. It uses fields from the `JobItems`, `OrderItems` and `User` tables. The results of the report can be filtered by:
   • Submission date
   • User name

23. **Usage per department**

   This report contains the used media per department. It uses fields from the `JobItems`, `OrderItems`, `MediaItem`, `JobMediaSettings` and `User` tables. The results of the report can be filtered by:
   • Submission date

24. **Usage per job**

   This report contains the used media per job. It uses fields from the `JobItems`, `OrderItems`, `MediaItem`, `JobMediaSettings` and `User` tables. The results of the report can be filtered by:
   • Submission date

25. **Usage per media**

   This report lists the used media. It uses fields from the `JobItems`, `OrderItems`, `MediaItem`, `JobMediaSettings` and `User` tables. The results of the report can be filtered by:
   • Submission date

26. **Usage per user**

   This report contains the used media per customer. It uses fields from the `JobItems`, `OrderItems`, `MediaItem`, `JobMediaSettings` and `User` tables. The results of the report can be filtered by:
   • Submission date

Retrieve the privacy-related information of your customers

The `PrintAccounting` database of PRISMAdirect stores privacy-related information about your customers. For example:
- Name, address, email address
- IP number
- Etc.

In addition, the information about the orders of your customers is stored in this database. You can download the information stored in the `PrintAccounting` database.

Retrieve the information stored in the `PrintAccounting` database

1. [Accounting database]
2. Specify a time span.
3. Click [Download].
   The application generates an Excel file that contains all information stored in the `PrintAccounting` database during the specified time span.
   The Excel file becomes available in the lower-left corner of the application. The Excel file is stored in the [Downloads] folder of your computer.

Chapter 3
Crystal reports

Crystal Reports application

PRISMAdirect offers a number of standard reports. These can be executed anytime.

You can create custom reports when you integrate Crystal Reports into PRISMAdirect. Crystal Reports is a separate product.

Create a custom Crystal report based on a default report of PRISMAdirect

This topic describes how you can create a custom report based on an existing report. For detailed information of the tables, stored procedures, and views that you can use, see "*The available tables, stored procedures and views of the PrintAccounting database* on page 17".

To create a report from scratch, see *Create a new Crystal report and add it to PRISMAdirect* on page 48.

1. Download a default report of PRISMAdirect
1. Open the [Configuration] workspace.
2. Click [System] - [Accounting and reporting] - [Accounting] - [Accounting workflow].
3. Enable the [Enable accounting] option.
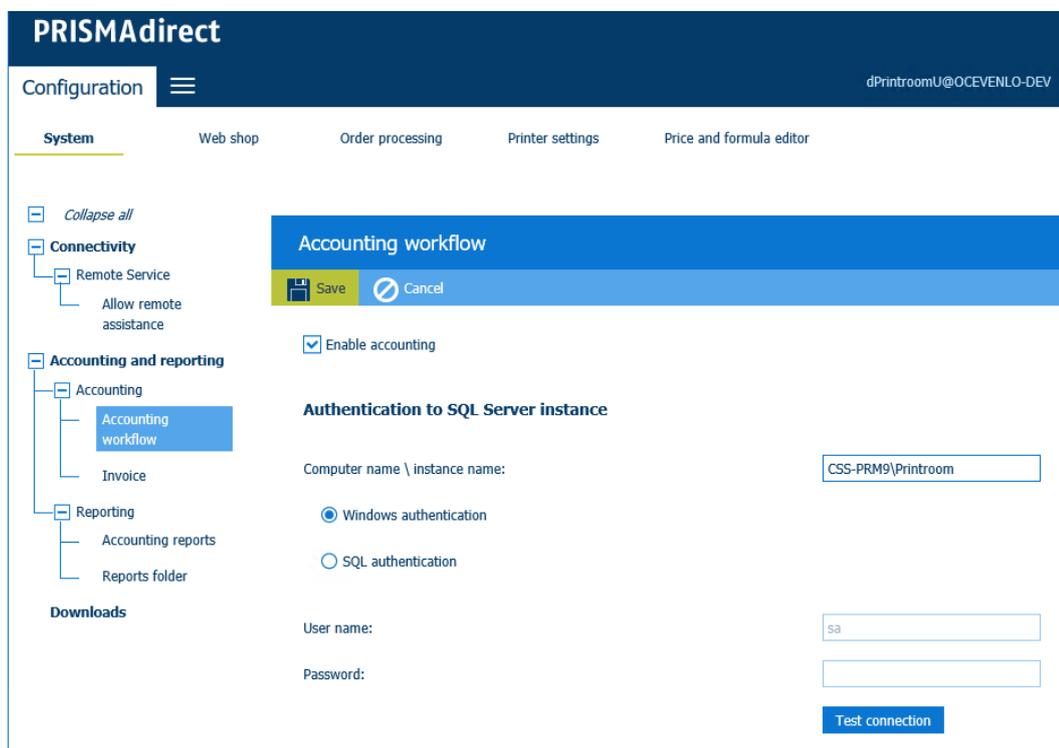
4. Click [System] - [Accounting and reporting] - [Reporting] - [Reports folder].
5. Type the path to the reports folder. For example, `C:\\MyReports`. You have to create the reports folder on the server of PRISMAdirect. The reports folder stores the reports that are generated in workspace [Reporting].
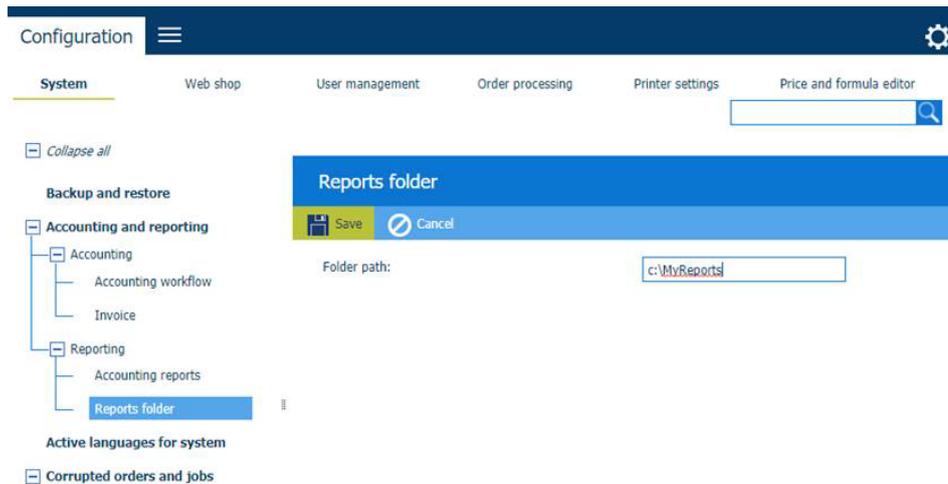
6. Open the [Order processing] workspace and finalize an order.
   When you finalize an order, any available accounting information is stored in the database along with the values of a number of ticket items. For example, information about the media and sheets.
7. Open the [Reporting] workspace.
8. Select a report that has the most in common with the custom report that you want to create. Click [Generate report].
9. Click [Save report] and select `Crystal Reports (RPT)`.
   The report is downloaded to the reports folder. In this example: `C:\\MyReports`.
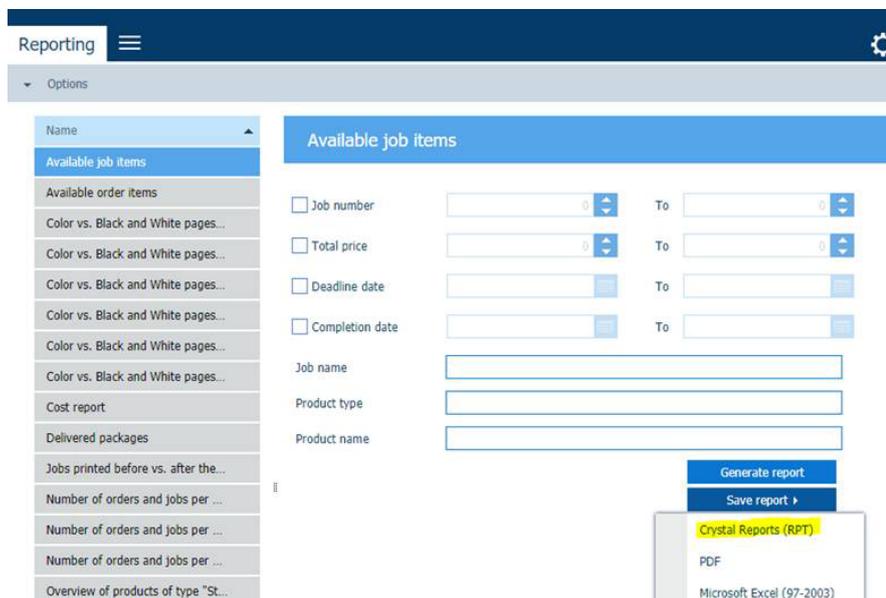
10. Go to the reports folder and change the name of the report. For example, you can rename the report to "Available job items extended.rpt".

2. Make the `PrintAccounting` database available to Crystal Reports

1. Open `Crystal Reports` and open the report.
2. Click "Database - Database Expert".
3. Double-click `<computer_name>\Printroom` to log in to the `PrintAccounting` database.
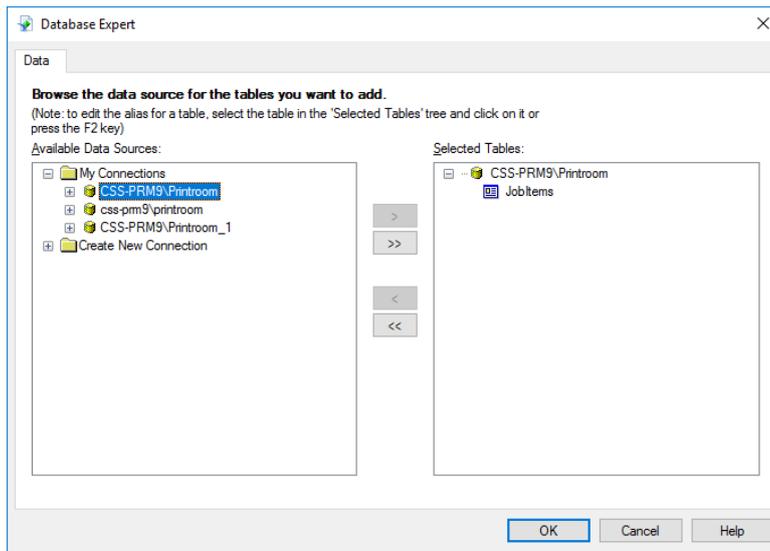
4. Type the password for user `sa` in the "OLE DB (ADO)" dialog that appears automatically. You have defined the password for user `sa` when you installed PRISMAdirect.
   Click [Finish].
   The `PrintAccounting` database is now available to `Crystal Reports`.
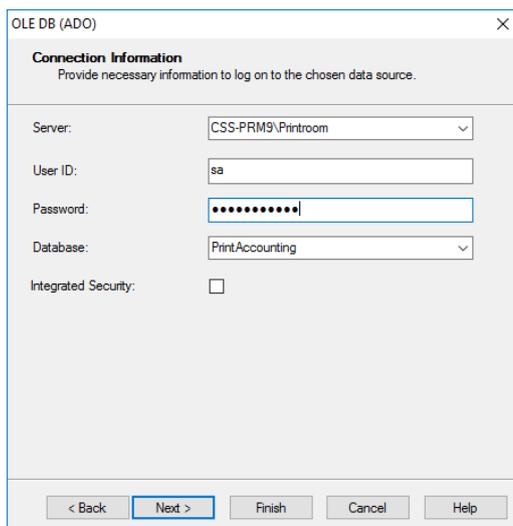
3. Customize the report with additional items from the `PrintAccounting` database

In the "Database Expert" dialog, you can see that the default report that you selected contains table `JobItems`. In this example, you want to add one or more custom job items to the customized report.

✏️ NOTE
Use only one stored procedure per report. Do not select an additional table or stored procedure when you already use a stored procedure for a report. More than one stored procedure in a report can lead to serious performance issues, especially for databases with many records. When you create a report, you can select tables or one stored procedure. Each table contains a limited set of ticket items. A stored procedure is a group of SQL-statements compiled into a single execution plan. You can select one of the following stored procedures per report:
- uspGetAllOrdersItems
  Contains all order ticket items including the custom ticket items at order level.
- uspGetAllJobsItems
  Contains all job ticket items including the custom ticket items at job level.
- uspGetOrderJobsItems
  Contains all order and job ticket items including the custom ticket items at order level and job level.

1. Select the table JobItems in the right-hand screen and click "<" to remove the table.
2. Go to the stored procedures, select uspGetAllJobsItems and click ">" to add it to the selected tables. This stored procedure contains all job items, both the default items and the custom items.
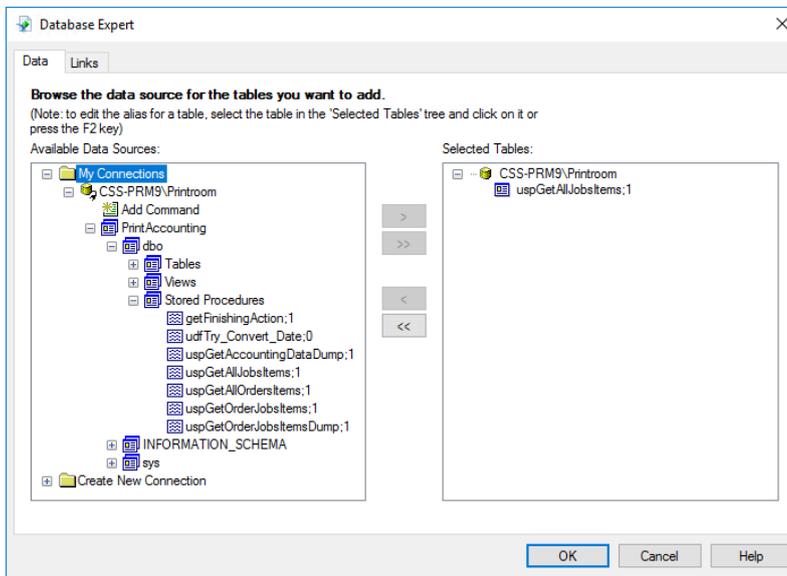
3. The "Enter Values" dialog appears automatically:
   1. Select False for @DumpDataInTempTable.
   2. Type a unique name for @TempTable, for example "table1".
   3. @selectedColumns:
      Later in this procedure, you can select which fields you want to display in the report. In that step, you can select from a set of available fields.
      - Enable option "Set to Null" if you want to display all available fields.
      - Fill this parameter with the desired fields names if you want to display only a subset of the available fields. Separate each field name with a comma.
   4. You can filter the results of the stored procedure using a valid WHERE clause. The WHERE clause is the condition of a SQL query. The name of the parameter that can contain the WHERE clause depends on the selected stored procedure. In this example, procedure uspGetAllJobsItems is selected and therefore the parameter name is @jobsFilter:
      - Type a valid WHERE clause in this parameter to filter the results of the stored procedure. For example, stored procedure uspGetAllJobsItems queries the tables JobItems and JobCustomItems. Table JobItems contains amongst others column JobNumber.

So, you can filter the results by using `JobNumber > 5 AND JobNumber < 20` in parameter `@jobsFilter`.

- Leave this parameter empty if you do not want to filter the results of the stored procedure. Do not select "Set to Null".

5. Click OK.

4. Click OK and close the "Database Expert" dialog.
   A message can appear ("A stored procedure has been used in this report. Please make sure ..."), click OK. A second message can appear ("Refresh report data?"). Click OK.
5. Click "View - Field Explorer".
6. Expand the stored procedure to find the item that you want to add to your custom report.
7. Right-click the item and select "Insert to report".
   The item is added to the report.
8. Configure the report to accommodate for the new custom job item.
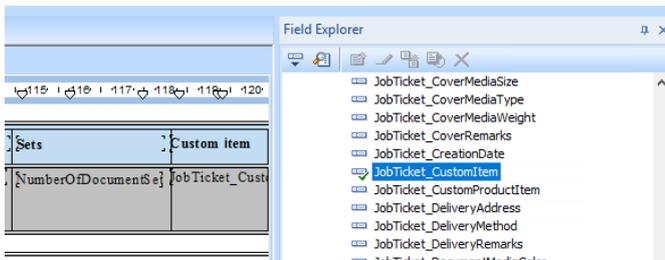
9. You can add and remove items to and from the report. Change the design of your report accordingly.
10. Click "Report - Refresh Report Data".
11. Select "Use current parameter values" and click OK.
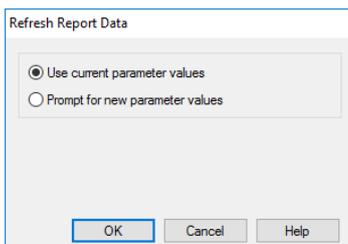
12. The preview of the report appears. The current values for the newly added items are displayed in the report. Check if your report is correct.

Print area	Price	Labour cost	Time to prepare job	Time to finish job	Time to print job	Printer name	Sets	Custom item
		0.00	0	0	0		1	
		0.00	0	0	0		1	
		0.00	0	0	0		1	Option2
		0.00	0	0	0		1	Option1

13. Click "File - Save".

4. Upload your custom report in PRISMAdirect

1. Open the [Reporting] workspace.
2. Click [Options] - [Upload custom reports].
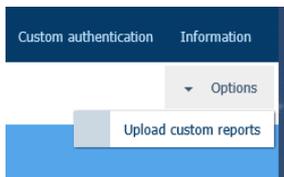
3. Browse to the RPT file in question.
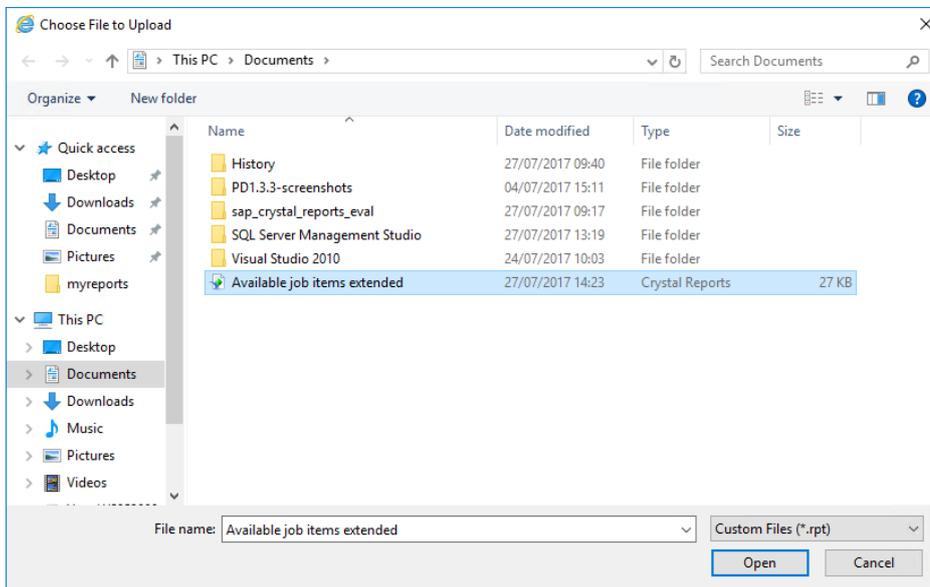
4. Click [Open].
   The report is added to the [Reporting] workspace.
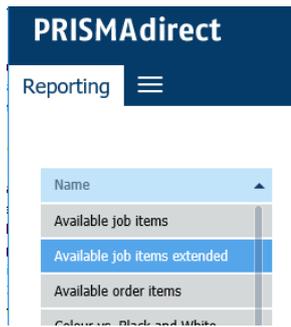
5. Now, you can generate the concerning report in PRISMAdirect. The report contains the new `Custom item`.

Create a new Crystal report and add it to PRISMAdirect

This topic describes how you can create reports using Crystal Reports which is a third party component. You have to upload the created reports in PRISMAdirect.

Procedure

1. Open Crystal Reports Editor and select "Report wizard" to create a report.

2. Select the data source for the report.

   1. Create a new connection.

   2. Select "OLE DB (ADO)".

   3. Select "Microsoft OLE DB Provider for SQL Server".
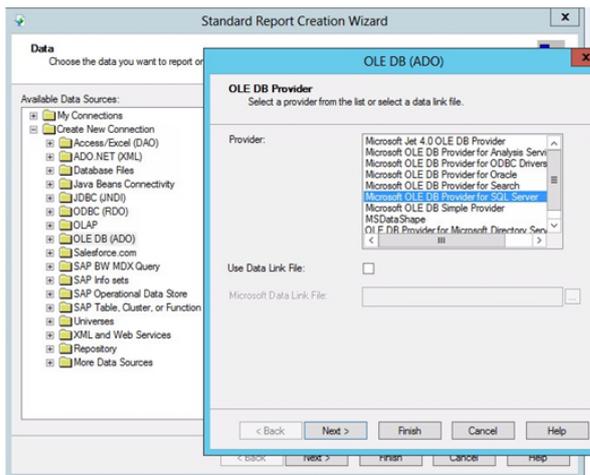
3. Type the logon credentials for the selected data source.

   1. Select the `PrintAccounting` database.

   2. User is `sa`.

   3. Type the password for user `sa`. You have defined the password for user `sa` when you installed PRISMAdirect.

   4. Click [Finish].
      The `PrintAccounting` database is now available to `Crystal Reports`.
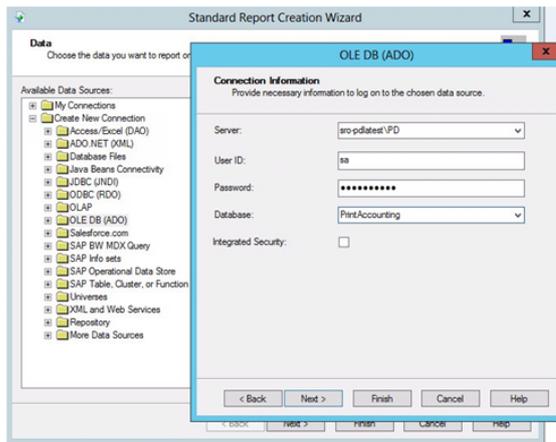
4. Select the data that you want to report on.
   You can select tables or a stored procedure. A stored procedure is a group of SQL-statements compiled into a single execution plan. Each table contains a limited set of ticket items.

   ✎ NOTE
   Use only one stored procedure per report. Do not select an additional table or stored procedure when you already use a stored procedure for a report. More than just one stored procedure in a report can lead to serious performance issues, especially for databases with many records.

   When you create a report, you can select tables or one stored procedure. Each table contains a limited set of ticket items. A stored procedure is a group of SQL-statements compiled into a single execution plan. You can select one of the following stored procedures per report:
   • uspGetAllOrdersItems
     Contains all order ticket items including the custom ticket items at order level.
   • uspGetAllJobsItems
     Contains all job ticket items including the custom ticket items at job level.
   • uspGetOrderJobsItems
     Contains all order and job ticket items including the custom ticket items at order level and job level.

5. For a stored procedure, you have to provide the input parameters. Tables do not have parameters.
   1. Select `False` for `@DumpDataInTempTable`.
   2. Type a unique name for `@TempTable`, for example "table1".
   3. `@selectedColumns`:
      Later in this procedure, you can select which fields you want to display in the report. In that step, you can select from a set of available fields.
      • Enable option "Set to Null" if you want to display all available fields.
      • Fill this parameter with the desired fields names if you want to display only a subset of the available fields. Separate each field name with a comma.
   4. You can filter the results of the stored procedure using a valid `WHERE` clause. The `WHERE` clause is the condition of a SQL query. The name of the parameter that can contain the `WHERE` clause depends on the selected stored procedure. In this example, procedure `uspGetAllJobsItems` is selected and therefore the parameter name is `@jobsFilter`:
      • Type a valid `WHERE` clause in this parameter to filter the results of the stored procedure. For example, stored procedure `uspGetAllJobsItems` queries the tables `JobItems` and `JobCustomItems`. Table `JobItems` contains amongst others column `JobNumber`. So, you can filter the results by using `JobNumber > 5 AND JobNumber < 20` in parameter `@jobsFilter`.
      • Leave this parameter empty if you do not want to filter the results of the stored procedure. Do not select "Set to Null".
   5. Click OK.

6. Select which fields you want to display in the report.
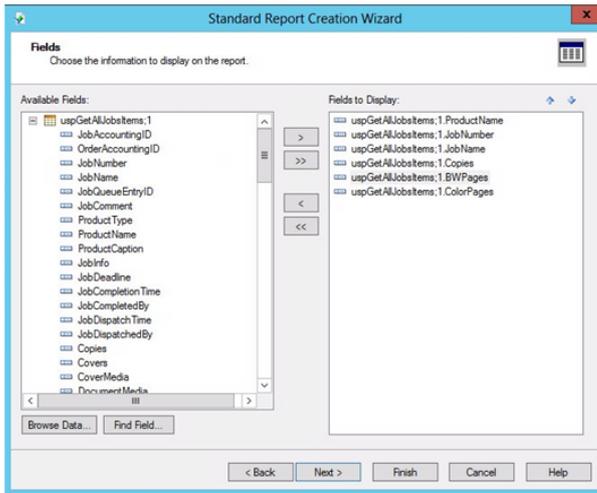
7. Optionally, you can select a field to group the data. Separating your data into groups often helps you to focus the information in the report.
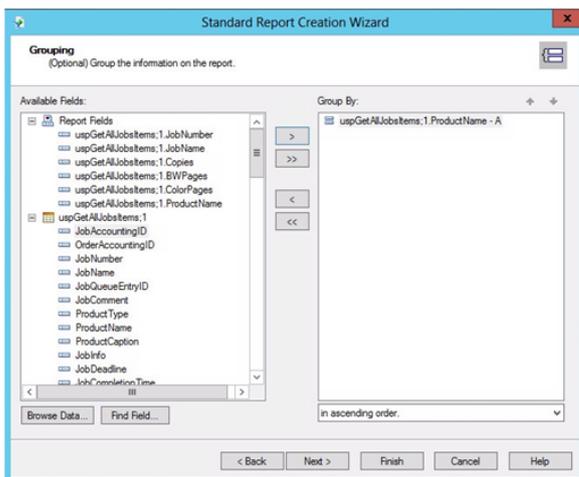
8. You can select a field to add summary information to the report. For example, select the field for colour pages to display a summary of the total number of color pages in the report.
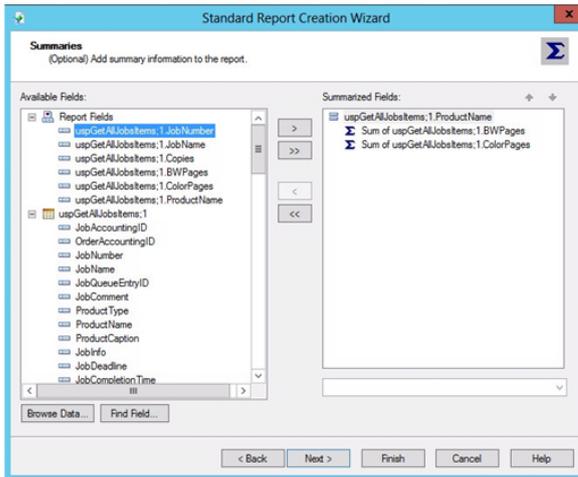
9. Select which chart you want to use for the report.
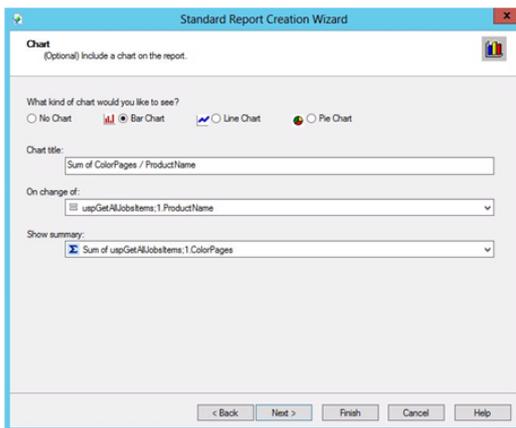
10. You can add new fields to an existing report.

   1. Expand the table or stored procedure in the "FieldExplorer" pane.

   2. Drag and drop the additional fields in section "Details" of the report.
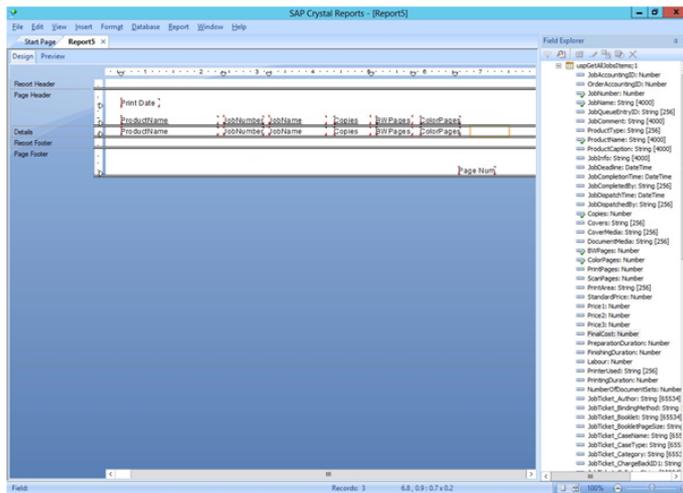
11. Add the created report to PRISMAdirect.

   1. Open the [Configuration] workspace of PRISMAdirect.

   2. Click [Options] - [Upload custom reports].

   3. Browse to the RPT file in question.

   4. Click [Open].
      The report is added to the [Configuration] workspace. Now, you can generate the concerning
      report in PRISMAdirect.

Index

56

Canon

Canon Inc.
www.canon.com

Canon U.S.A., Inc.
www.usa.canon.com

Canon Canada Inc.
www.canon.ca

Canon Europe Ltd
www.canon-europe.com

Canon Latin America Inc.
www.cla.canon.com

Canon Australia PTY. Ltd
www.canon.com.au

Canon China Co., Ltd
www.canon.com.cn

Canon Singapore PTE. Ltd
www.canon.com.sg

Canon Hongkong Co., Ltd
www.canon.com.hk