Acumatica

The Cloud ERP

S150 Report Designer (Part 1)

Johans Saavedra

Senior ERP Functional Consultant

Timing and Agenda

August 19, 2024 - 10:00-11:30 AM PT

Day 1

Lesson 1: Getting Started with the Report Designer

Lesson 2: Discovering DACs

Lesson 3: Creating a Report

August 20, 2024 - 10:00-11:30 AM PT

Day 2

Lesson 4: Getting Data from Multiple DACs

Day 1

Scott McLaughlin
Acumatica Sponsored Professional Race Car Driver

Company Structure

The SweetLife Fruits & Jams company is a midsize company located in New York City. The company consists of the following branches:

- SweetLife Head Office and Wholesale Center: This branch of the company consists of a jam factory and a large warehouse where the company stores fruit (purchased from wholesale vendors) and the jam it produces. Warehouse workers perform warehouse operations by using barcode scanners or mobile devices with barcode scanning support.

- SweetLife Store: This branch has a retail shop with a small warehouse to which the goods to be sold are distributed from the company's main warehouse. This branch is also planning on selling goods via a website created on an e-commerce platform to accept orders online. The e-commerce integration project is underway.

- SweetLife Service and Equipment Sales Center: This branch is a service center with a small warehouse where juicers are stored. This branch assembles, sells, installs, and services juicers, in addition to training customers' employees to operate juicers.

Operational Activity

The company has been operating starting in the 01-2023 financial period. In November 2023, the company started using Acumatica ERP as an ERP and CRM system and migrated all data of the main office and retail store to Acumatica ERP. The equipment center has begun its operations in 01-2024 in response to the company's growth.

The base currency of the company and its subsidiaries is the US dollar (USD). All amounts in documents and reports are expressed in US dollars unless otherwise indicated.

SweetLife Company Sales and Services

Each SweetLife company's branch has its own business processes, as follows:

- SweetLife Head Office and Wholesale Center: In this branch, jams and fruit are sold to wholesale customers, such as restaurants and cafes. The company also conducts home canning training at the customer's location and webinars on the company's website.

- SweetLife Store: In the store, retail customers purchase fresh fruit, berries, and jams, or pick up the goods they have ordered on the website. Some of the goods listed in the website catalog are not stored in the retail warehouse, such as tropical fruits (which are purchased on demand) and tea (which is drop-shipped from a third-party vendor).

- SweetLife Service and Equipment Sales Center: This branch assembles juicers, sells juicers, provides training on equipment use, and offers equipment installation, including site review and maintenance services. The branch performs short-term service provision.

The company has local and international customers. The ordered items are delivered by drivers using the company's own vehicle. Customers can pay for orders by using various payment methods (cash, checks, or credit cards).

Muffins & Cakes Company Sales and Services

The Muffins & Cakes branches have the following business processes:

- Muffins Head Office & Wholesale Center: In this branch, baked goods and products for baking are sold to wholesale customers, such as restaurants and cafes. The company also conducts baking classes at customer locations.

- Muffins Store: In the store, small retail customers purchase baked goods, or pick the goods ordered on the website.

Company Purchases

The company purchases fruits and spices from large fruit vendors for sale and for jam production. For producing jams and packing jams and fruits, the company purchases jars, labels, and paper bags from various vendors. For the internal needs of the main office and store, the company purchases stationery (printing paper, pens, and pencils), computers, and computer accessories from various vendors. The company also purchases juicers and juicer parts from large juicer vendors and either purchases the installation service for the juicers or provides the installation service on its own, depending on the complexity of the installation.

The Muffins & Cakes company also purchases stationery (printing paper, pens, and pencils) and advertising services

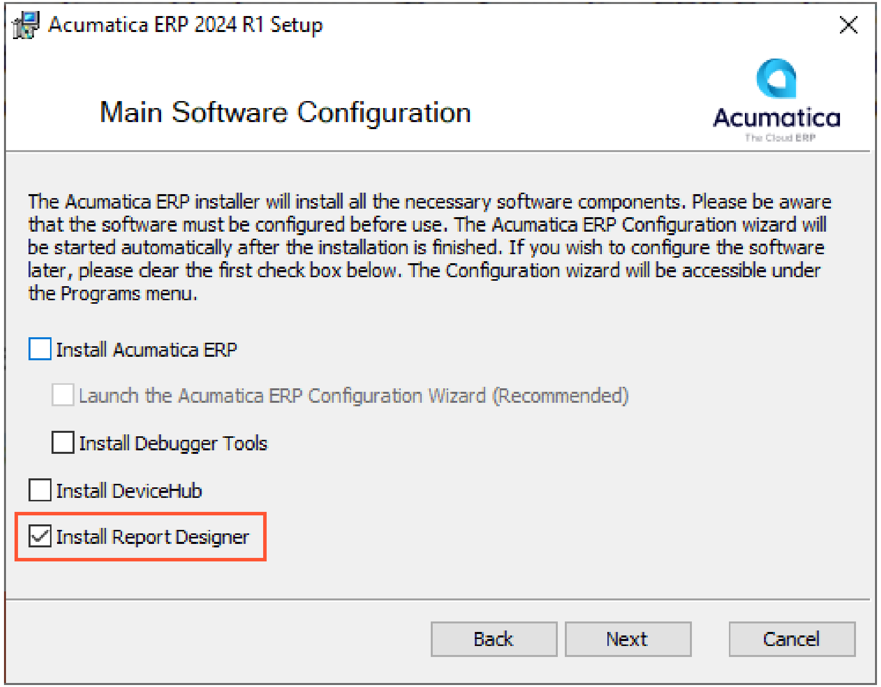
Lesson 1: Getting Started with the Report Designer

Learning Objectives

In this lesson, you will learn how to do the following:

- Install the Acumatica Report Designer
- Start the Acumatica Report Designer and explore its interface

Figure: The Acumatica ERP Installer wizard



Acumatica ERP 2024 R1 Setup

Main Software Configuration

The Acumatica ERP installer will install all the necessary software components. Please be aware that the software must be configured before use. The Acumatica ERP Configuration wizard will be started automatically after the installation is finished. If you wish to configure the software later, please clear the first check box below. The Configuration wizard will be accessible under the Programs menu.

☐ Install Acumatica ERP

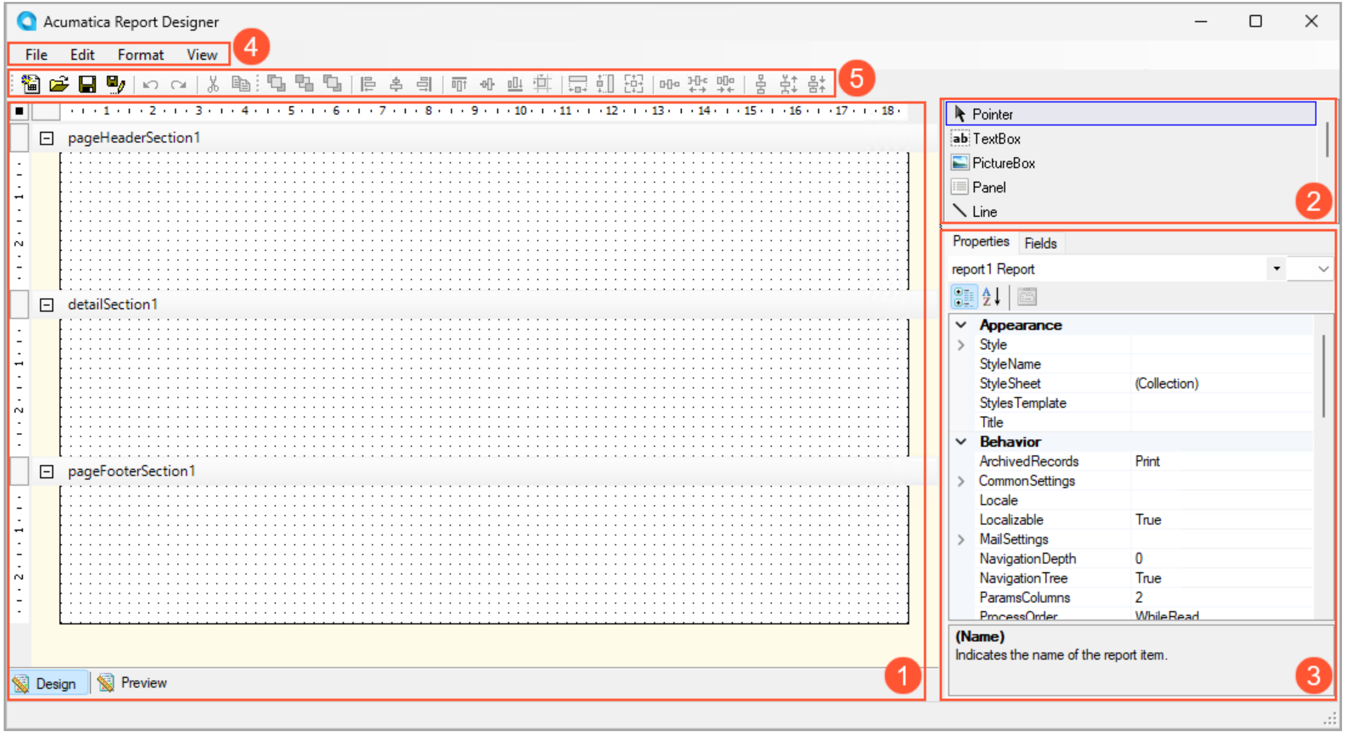
☐ Launch the Acumatica ERP Configuration Wizard (Recommended)

☐ Install Debugger Tools

☐ Install DeviceHub

☑ Install Report Designer

Figure: The main window of the Report Designer



report1 Report

Appearance	
Style	
StyleName	
StyleSheet	(Collection)
StylesTemplate	
Title	
Behavior	
ArchivedRecords	Print
CommonSettings	
Locale	
Localizable	True
MailSettings	
NavigationDepth	0
NavigationTree	True
ParamsColumns	2
ProcessOrder	WhileRead

(Name)
Indicates the name of the report item.

Report Designer: To Install the Acumatica Report Designer

Story

Suppose that you are a technical specialist in your company who is working on customizations. An accountant of the company has requested a number of reports that are not among the predefined reports. To develop reports based on the data of Acumatica ERP, you have decided to use the Acumatica Report Designer, which is included in the Acumatica ERP installation package.

Lesson 2: Discovering DACs

Learning Objectives

In this lesson, you will learn how to do the following:

- Inspect UI elements to find the underlying data fields.

DAC Discovery: General Information

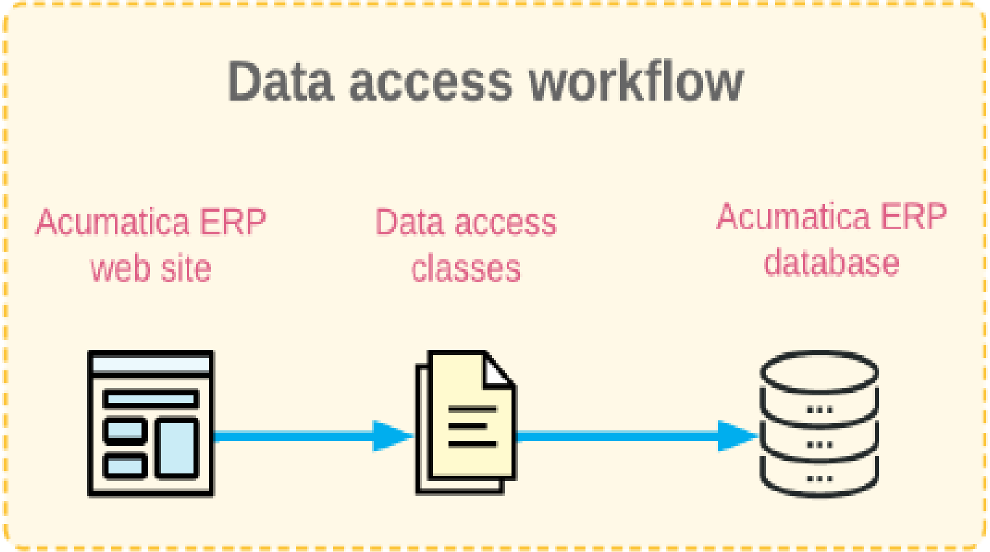
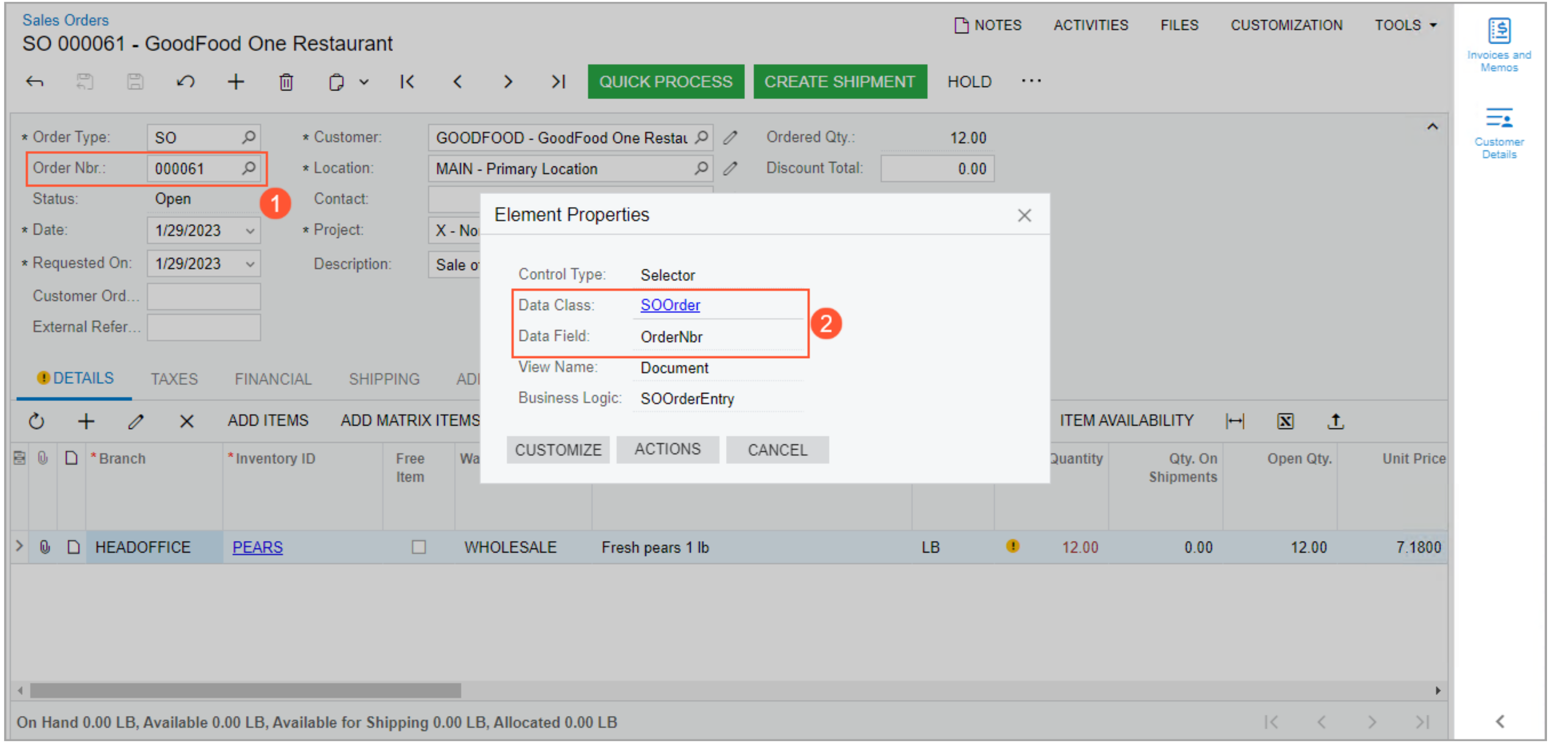


Figure: Form element inspection



SO 000061 - GoodFood One Restaurant

* Order Type:	SO	* Customer:	GOODFOOD - GoodFood One Restau	Ordered Qty.:	12.00
Order Nbr.:	000061	* Location:	MAIN - Primary Location	Discount Total:	0.00
Status:	Open	Contact:			
* Date:	1/29/2023	* Project:	X - No		
* Requested On:	1/29/2023	Description:	Sale o		
Customer Ord...					
External Refer...					

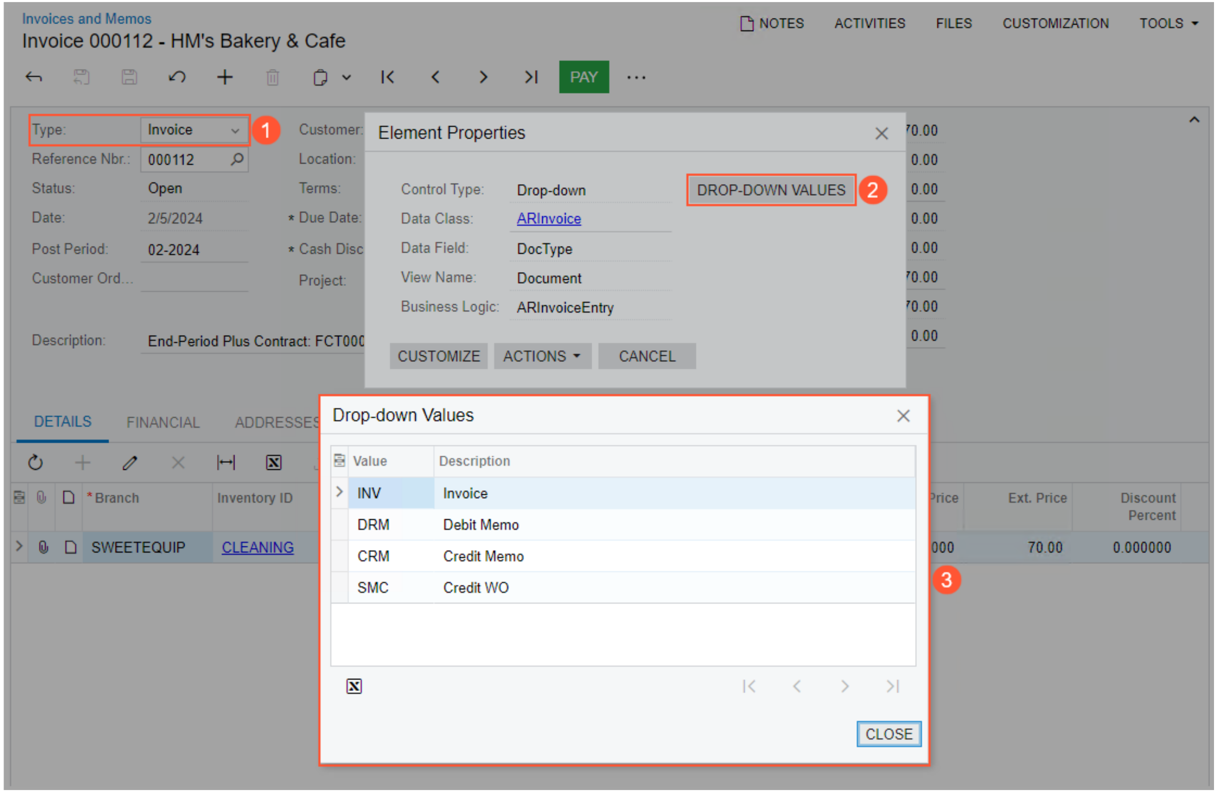
Element Properties

Control Type:	Selector
Data Class:	SOOrder
Data Field:	OrderNbr
View Name:	Document
Business Logic:	SOOrderEntry

* Branch	* Inventory ID	Free Item	Wa				Quantity	Qty. On Shipments	Open Qty.	Unit Price
HEADOFFICE	PEARS	□	WHOLESALE	Fresh pears 1 lb	LB	!	12.00	0.00	12.00	7.1800

On Hand 0.00 LB, Available 0.00 LB, Available for Shipping 0.00 LB, Allocated 0.00 LB

Figure: Inspection of a form element with a drop-down control



Invoices and Memos

Invoice 000112 - HM's Bakery & Cafe

Type:	Invoice
Reference Nbr.:	000112
Status:	Open
Date:	2/5/2024
Post Period:	02-2024
Customer Ord...	
Description:	End-Period Plus Contract: FCT000

Element Properties

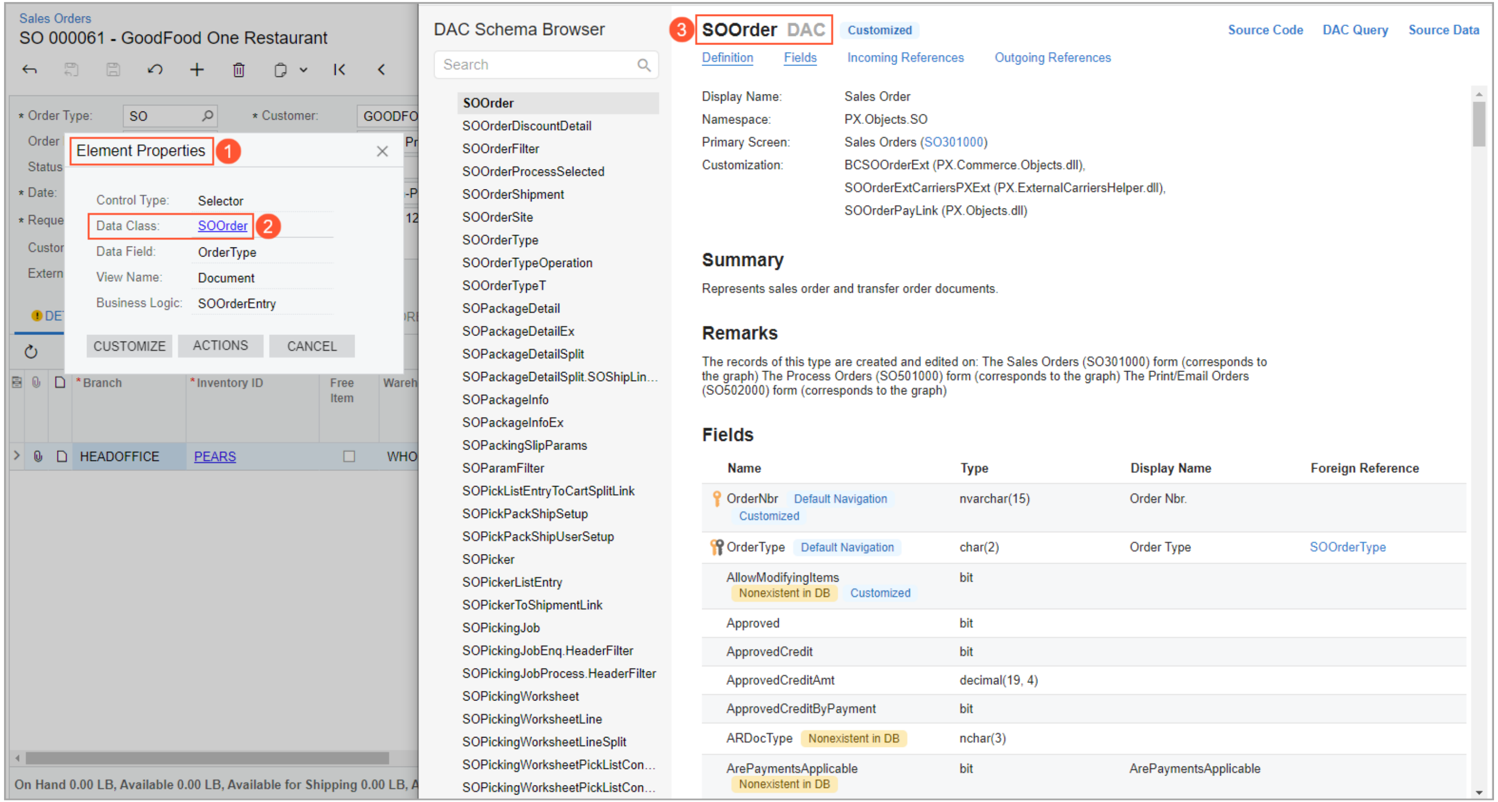
Control Type:	Drop-down
Data Class:	ARInvoice
Data Field:	DocType
View Name:	Document
Business Logic:	ARInvoiceEntry

Drop-down Values

Value	Description
INV	Invoice
DRM	Debit Memo
CRM	Credit Memo
SMC	Credit WO

*Branch	Inventory ID	Price	Ext. Price	Discount Percent
SWEETEQUIP	CLEANING	000	70.00	0.000000

Figure: Form element inspection



Sales Orders

SO 000061 - GoodFood One Restaurant

On Hand 0.00 LB, Available 0.00 LB, Available for Shipping 0.00 LB, A

Element Properties 1

Control Type:	Selector
Data Class:	SOOrder 2
Data Field:	OrderType
View Name:	Document
Business Logic:	SOOrderEntry

CUSTOMIZE ACTIONS CANCEL

DAC Schema Browser

Search

SOOrder
SOOrderDiscountDetail
SOOrderFilter
SOOrderProcessSelected
SOOrderShipment
SOOrderSite
SOOrderType
SOOrderTypeOperation
SOOrderTypeT
SOPackageDetail
SOPackageDetailEx
SOPackageDetailSplit
SOPackageDetailSplit.SOShipLin...
SOPackageInfo
SOPackageInfoEx
SOPackingSlipParams
SOParamFilter
SOPickListEntryToCartSplitLink
SOPickPackShipSetup
SOPickPackShipUserSetup
SOPicker
SOPickerListEntry
SOPickerToShipmentLink
SOPickingJob
SOPickingJobEnq.HeaderFilter
SOPickingJobProcess.HeaderFilter
SOPickingWorksheet
SOPickingWorksheetLine
SOPickingWorksheetLineSplit
SOPickingWorksheetPickListCon...
SOPickingWorksheetPickListCon...

3 SOOrder DAC Customized

Source Code DAC Query Source Data

Definition Fields Incoming References Outgoing References

Display Name:	Sales Order
Namespace:	PX.Objects.SO
Primary Screen:	Sales Orders (SO301000)
Customization:	BCSOOrderExt (PX.Commerce.Objects.dll), SOOrderExtCarriersPXExt (PX.ExternalCarriersHelper.dll), SOOrderPayLink (PX.Objects.dll)

Summary

Represents sales order and transfer order documents.

Remarks

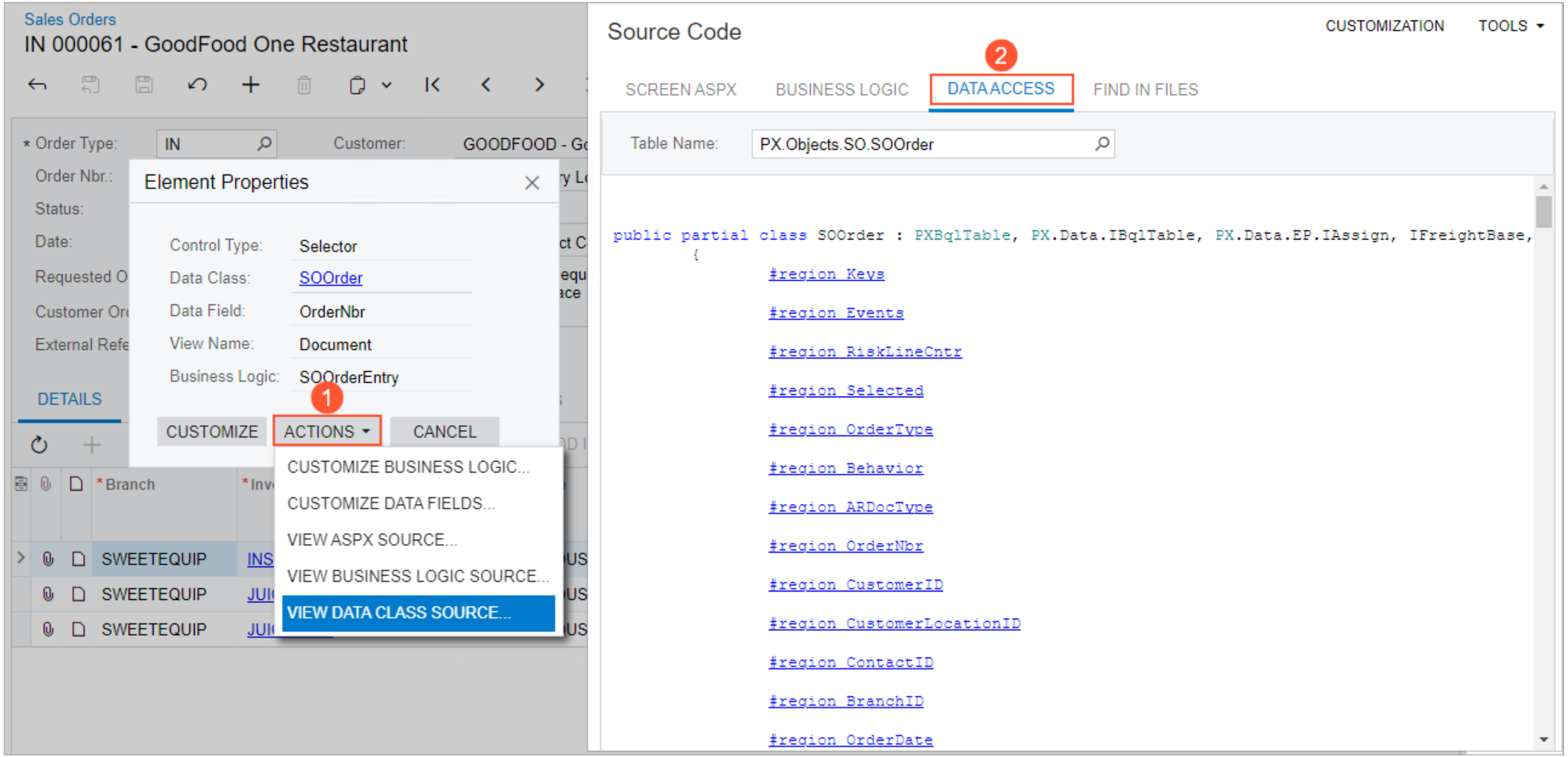
The records of this type are created and edited on: The Sales Orders (SO301000) form (corresponds to the graph) The Process Orders (SO501000) form (corresponds to the graph) The Print/Email Orders (SO502000) form (corresponds to the graph)

Fields

Name	Type	Display Name	Foreign Reference
OrderNbr Default Navigation Customized	nvarchar(15)	Order Nbr.	
OrderType Default Navigation	char(2)	Order Type	SOOrderType
AllowModifyingItems Nonexistent in DB Customized	bit		
Approved	bit		
ApprovedCredit	bit		
ApprovedCreditAmt	decimal(19, 4)		
ApprovedCreditByPayment	bit		
ARDocType Nonexistent in DB	nchar(3)		
ArePaymentsApplicable Nonexistent in DB	bit	ArePaymentsApplicable	

Acumatica
The Cloud ERP

Figure: Form element inspection



Sales Orders

IN 000061 - GoodFood One Restaurant

* Order Type:	IN	Customer:	GOODFOOD - Gc
Order Nbr.:			
Status:			
Date:			
Requested O			
Customer Ord			
External Refe			

Element Properties

Control Type:	Selector
Data Class:	SOOrder
Data Field:	OrderNbr
View Name:	Document
Business Logic:	SOOrderEntry

1
CUSTOMIZE
ACTIONS ▾
CANCEL
CUSTOMIZE BUSINESS LOGIC...
CUSTOMIZE DATA FIELDS...
VIEW ASPX SOURCE...
VIEW BUSINESS LOGIC SOURCE...
VIEW DATA CLASS SOURCE...
DETAILS

			* Branch	* Inv
			SWEETEQUIP	INS
			SWEETEQUIP	JUI
			SWEETEQUIP	JUI

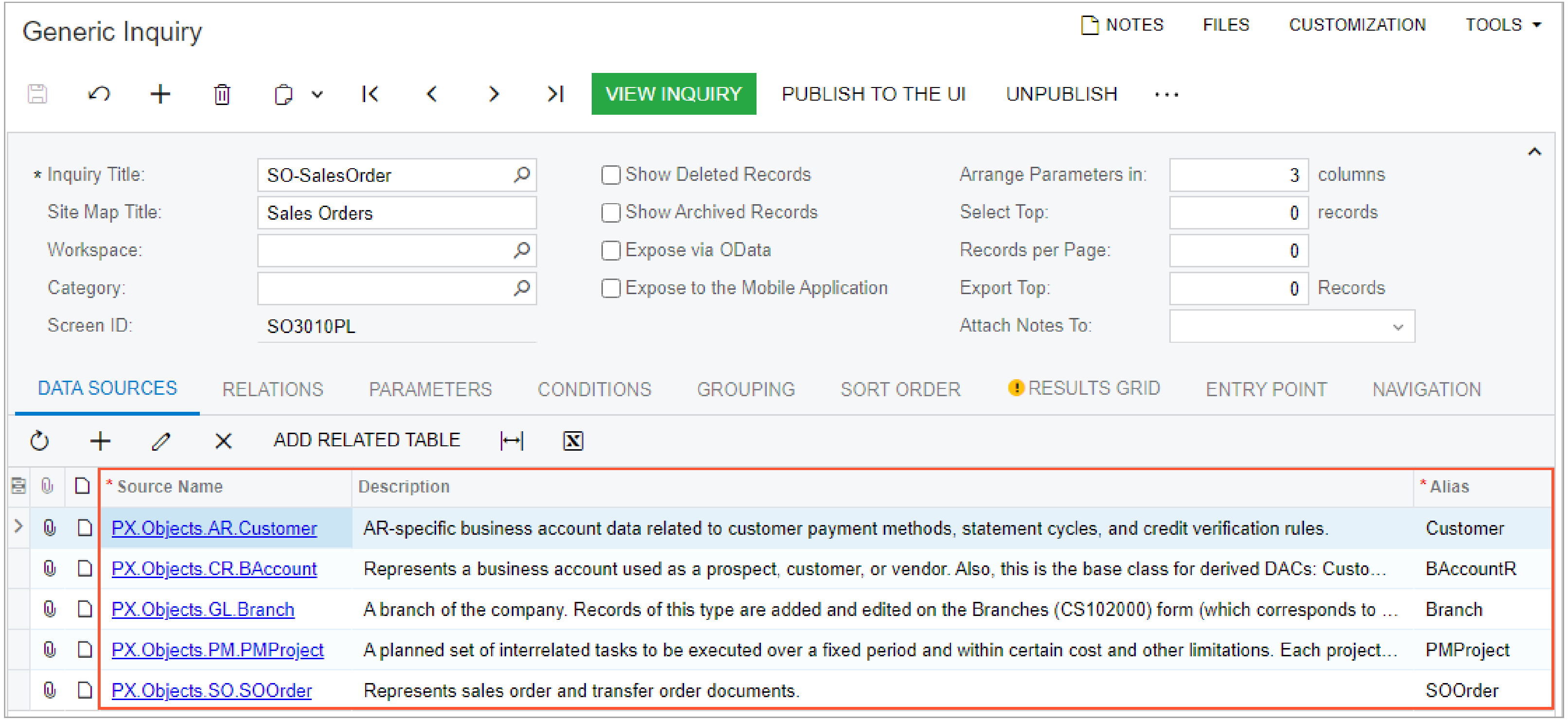
Source Code

Table Name: PX.Objects.SO.SOOrder

```
public partial class SOOrder : PXBqlTable, PX.Data.IBqlTable, PX.Data.EP.IAssign, IFreightBase,
        {
                #region Keys
                #region Events
                #region RiskLineCntr
                #region Selected
                #region OrderType
                #region Behavior
                #region ARDocType
                #region OrderNbr
                #region CustomerID
                #region CustomerLocationID
                #region ContactID
                #region BranchID
                #region OrderDate
```

Acumatica
The Cloud ERP

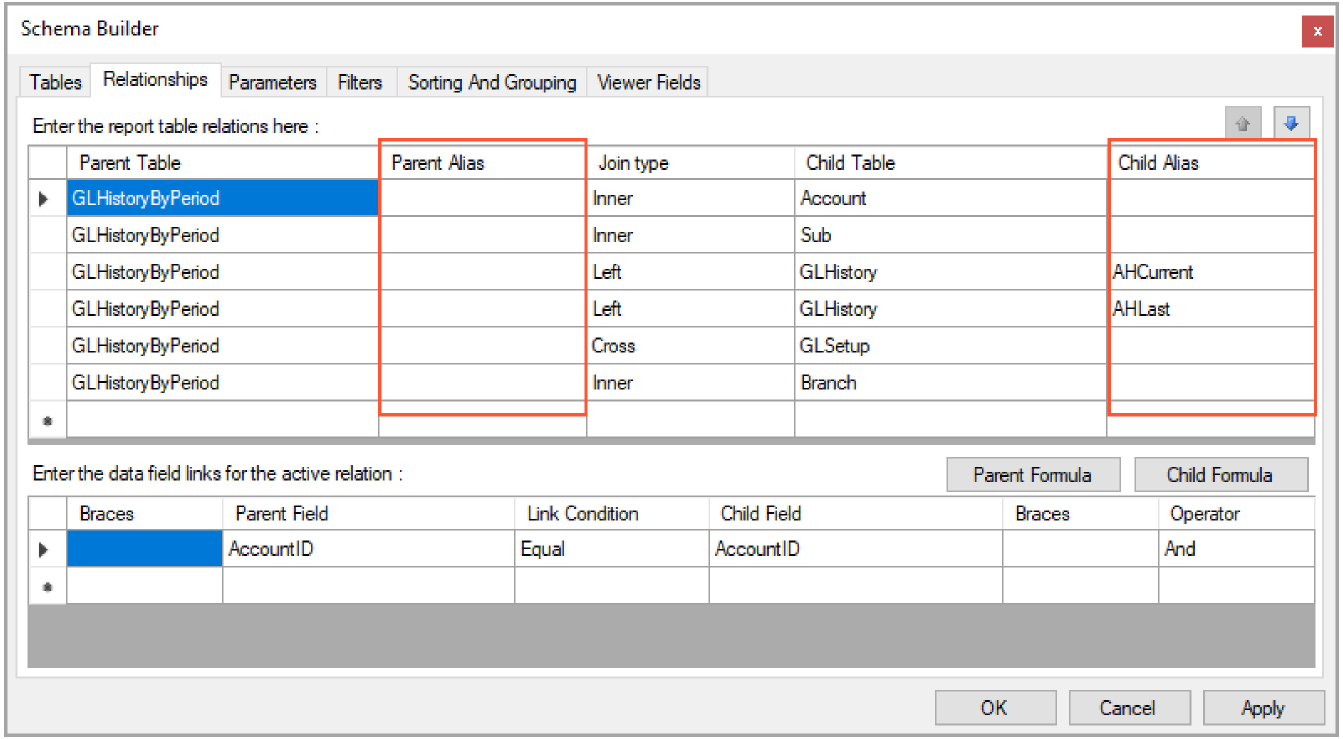
Figure: The list of data access classes and their aliases



* Inquiry Title:	SO-SalesOrder	□ Show Deleted Records	Arrange Parameters in: 3 columns
Site Map Title:	Sales Orders	□ Show Archived Records	Select Top: 0 records
Workspace:		□ Expose via OData	Records per Page: 0
Category:		□ Expose to the Mobile Application	Export Top: 0 Records
Screen ID:	SO3010PL		Attach Notes To:

* Source Name	Description	* Alias
PX.Objects.AR.Customer	AR-specific business account data related to customer payment methods, statement cycles, and credit verification rules.	Customer
PX.Objects.CR.BAccount	Represents a business account used as a prospect, customer, or vendor. Also, this is the base class for derived DACs: Custo...	BAccountR
PX.Objects.GL.Branch	A branch of the company. Records of this type are added and edited on the Branches (CS102000) form (which corresponds to ...	Branch
PX.Objects.PM.PMProject	A planned set of interrelated tasks to be executed over a fixed period and within certain cost and other limitations. Each project...	PMProject
PX.Objects.SO.SOOrder	Represents sales order and transfer order documents.	SOOrder

Figure: Aliases in the Schema Builder of the Report Designer



Enter the report table relations here :

	Parent Table	Parent Alias	Join type	Child Table	Child Alias
▶	GLHistoryByPeriod		Inner	Account	
	GLHistoryByPeriod		Inner	Sub	
	GLHistoryByPeriod		Left	GLHistory	AHCurrent
	GLHistoryByPeriod		Left	GLHistory	AHLast
	GLHistoryByPeriod		Cross	GLSetup	
	GLHistoryByPeriod		Inner	Branch	
*					

Enter the data field links for the active relation :

	Braces	Parent Field	Link Condition	Child Field	Braces	Operator
▶		AccountID	Equal	AccountID		And
*						

DAC Discovery: To Inspect UI Elements

Story

Suppose that you are a technical specialist in your company who is working on simple customizations. A sales representative has requested that you create a generic inquiry that lists sales orders. The inquiry should include the following data for each listed sales order: the sales order number, the order type, the status, the date, and the customer's name.

To get started in this effort, you need to inspect the relevant user interface elements on the Sales Orders (SO301000) form and the Customers (AR303000) form—the data entry forms on which sales orders and customers are created—in order to figure out which data access classes and data fields are related to them. (In this activity, you will only inspect the UI elements. You will not develop the generic inquiry.)

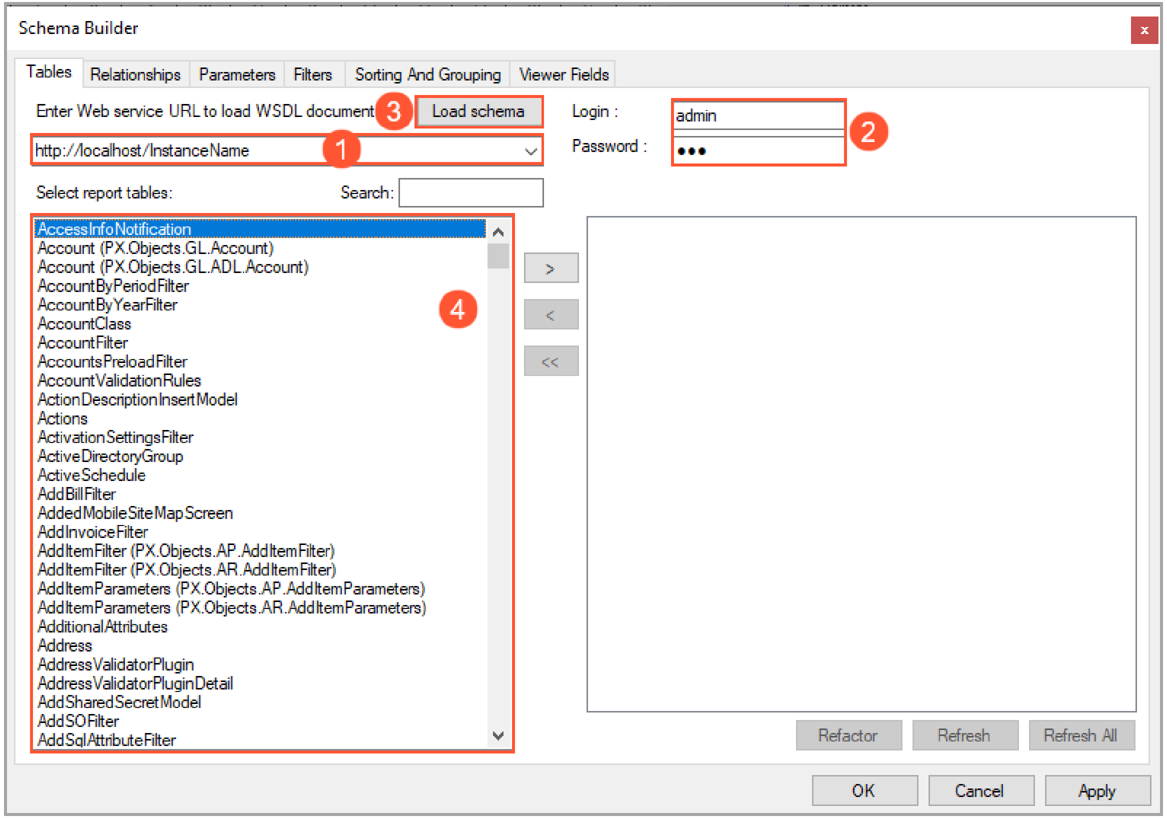
Lesson 3: Creating a Report

Learning Objectives

In this lesson, you will learn how to do the following:

- Open and view an existing report
- Copy an existing report
- Create a report from scratch
- Update the database schema for reports
- Publish and view a report

Figure: Loading of the schema of the data access classes



Schema Builder

Enter Web service URL to load WSDL document 3 Load schema

http://localhost/InstanceName 1

Login : admin 2

Password : ●●●

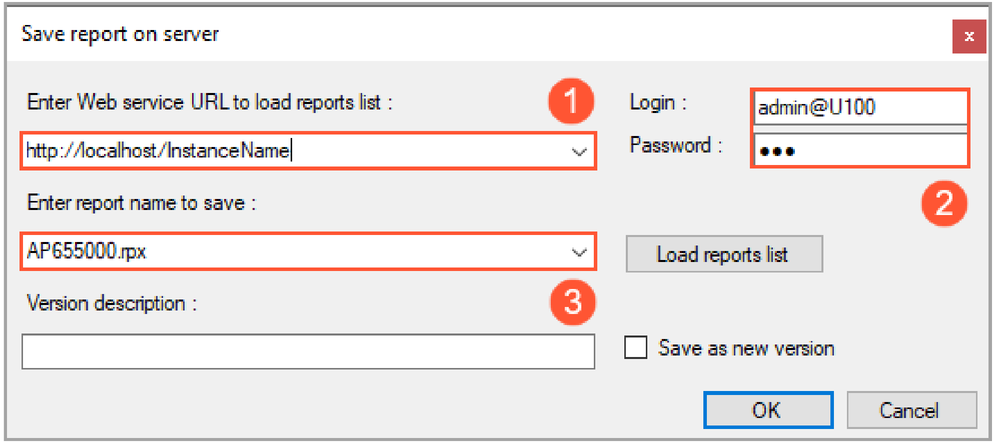
Select report tables:

Search:

4

- AccessInfoNotification
- Account (PX.Objects.GL.Account)
- Account (PX.Objects.GL.ADL.Account)
- AccountByPeriodFilter
- AccountByYearFilter
- AccountClass
- AccountFilter
- AccountsPreloadFilter
- AccountValidationRules
- ActionDescriptionInsertModel
- Actions
- ActivationSettingsFilter
- ActiveDirectoryGroup
- ActiveSchedule
- AddBillFilter
- AddedMobileSiteMapScreen
- AddInvoiceFilter
- AddItemFilter (PX.Objects.AP.AddItemFilter)
- AddItemFilter (PX.Objects.AR.AddItemFilter)
- AddItemParameters (PX.Objects.AP.AddItemParameters)
- AddItemParameters (PX.Objects.AR.AddItemParameters)
- AdditionalAttributes
- Address
- AddressValidatorPlugin
- AddressValidatorPluginDetail
- AddSharedSecretModel
- AddSOFilter
- AddSqlAttributeFilter

Figure: The Save Report on Server dialog box



Save report on server

Enter Web service URL to load reports list : http://localhost/InstanceName

Login : admin@U100

Password : ●●●

Enter report name to save : AP655000.rpx

Version description :

□ Save as new version

Figure: Sections of a report layout

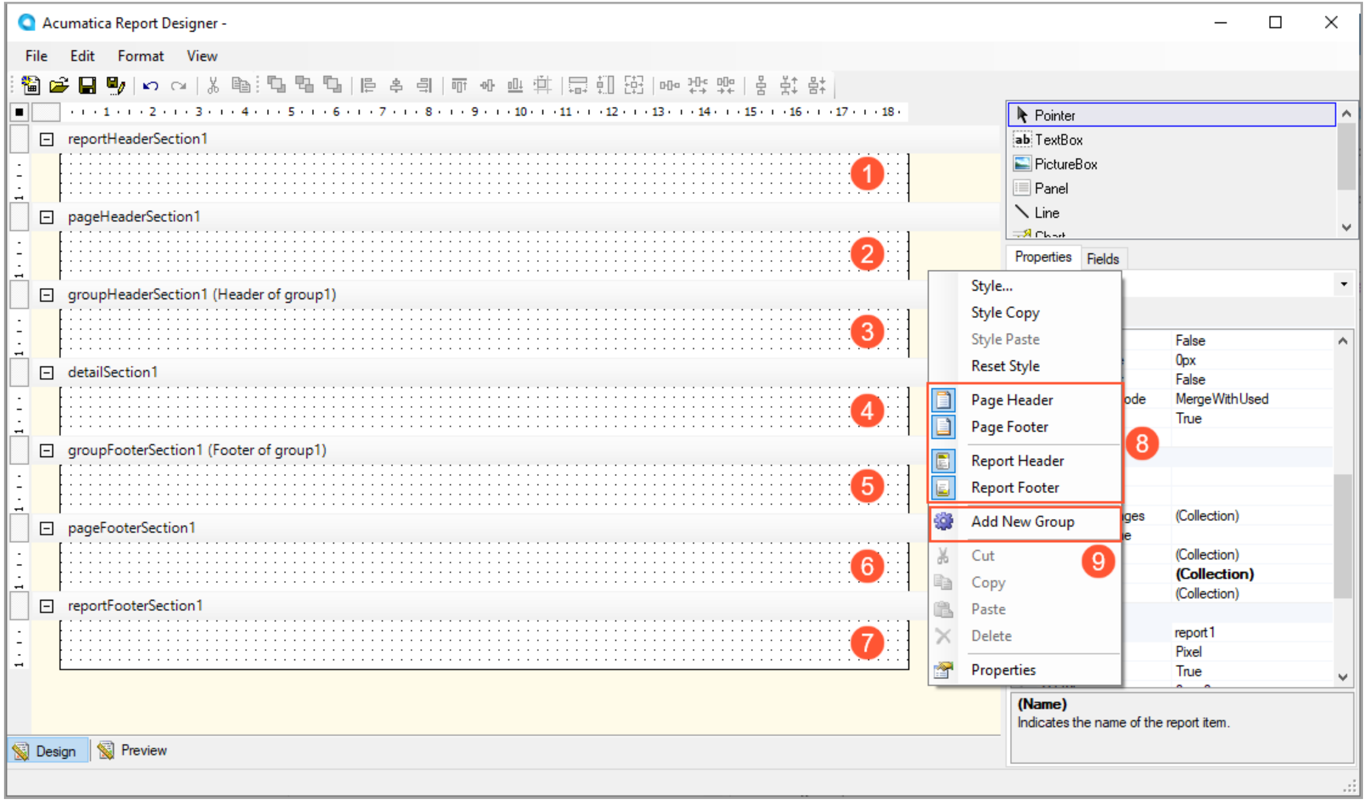
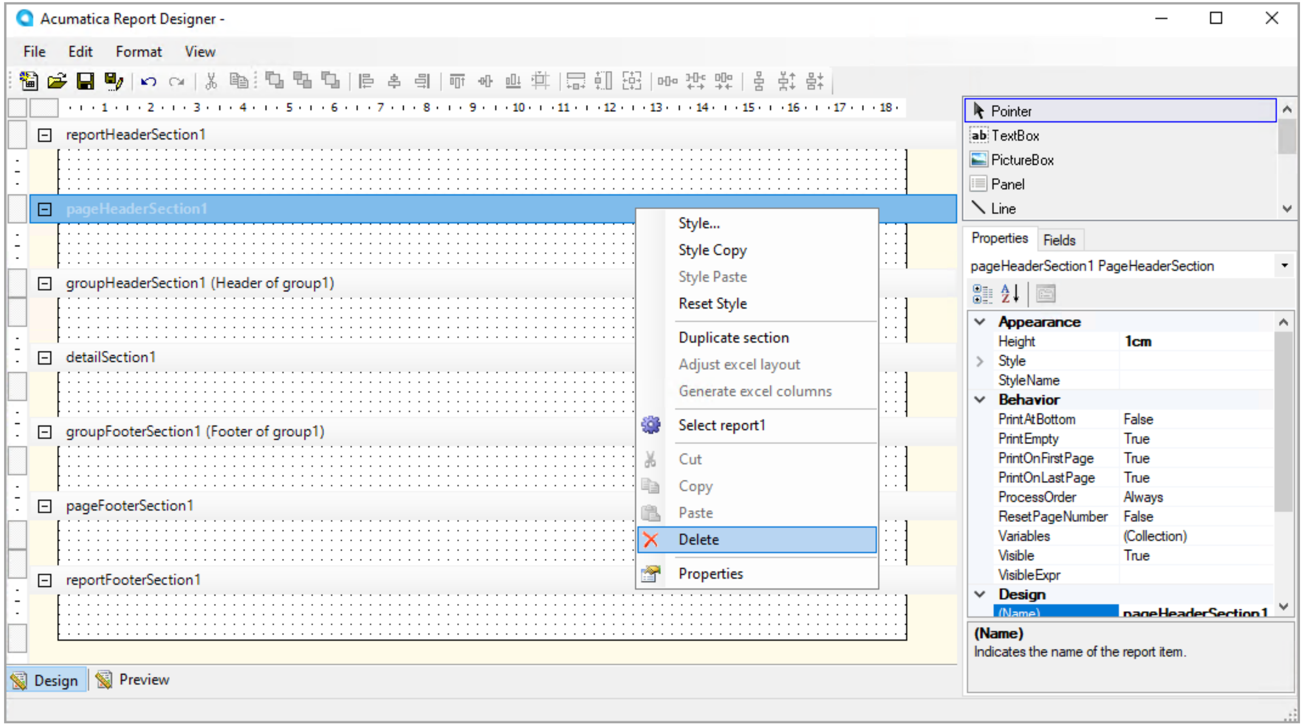


Figure: Deletion of a section



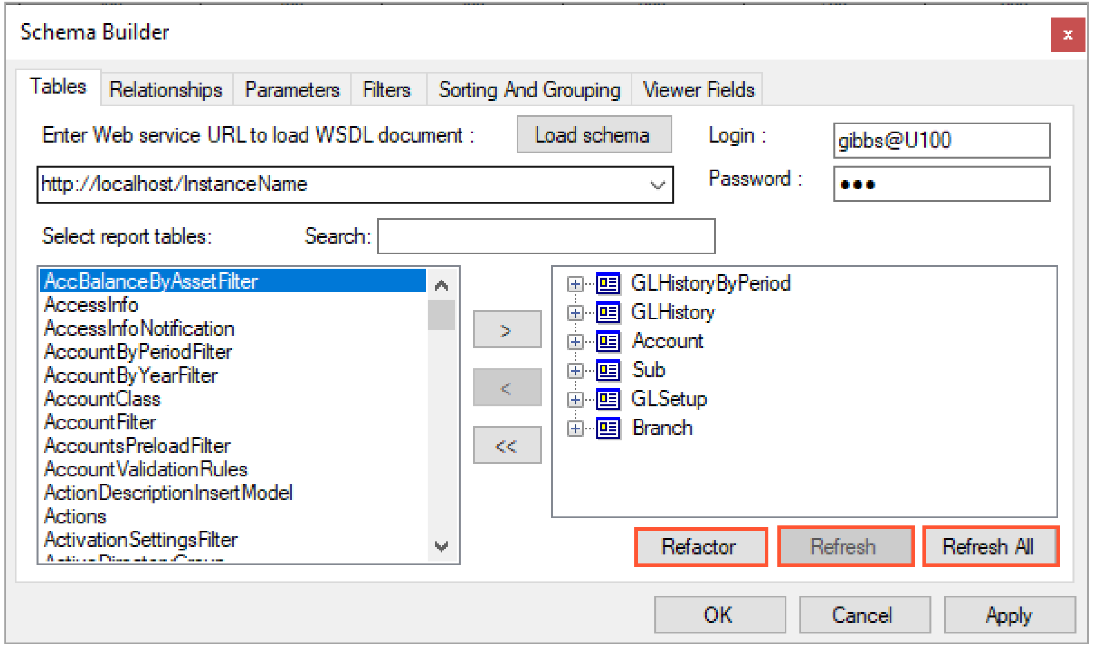
pageHeaderSection1 PageHeaderSection

Appearance	
Height	1cm
Style	
StyleName	
Behavior	
PrintAtBottom	False
PrintEmpty	True
PrintOnFirstPage	True
PrintOnLastPage	True
ProcessOrder	Always
ResetPageNumber	False
Variables	(Collection)
Visible	True
VisibleExpr	
Design	
(Name)	pageHeaderSection1

(Name)
Indicates the name of the report item.

Acumatica
The Cloud ERP

Figure: The buttons to update the database schema for a report



Schema Builder

Tables | Relationships | Parameters | Filters | Sorting And Grouping | Viewer Fields

Enter Web service URL to load WSDL document : http://localhost/InstanceName

Login : gibbs@U100

Password : ●●●

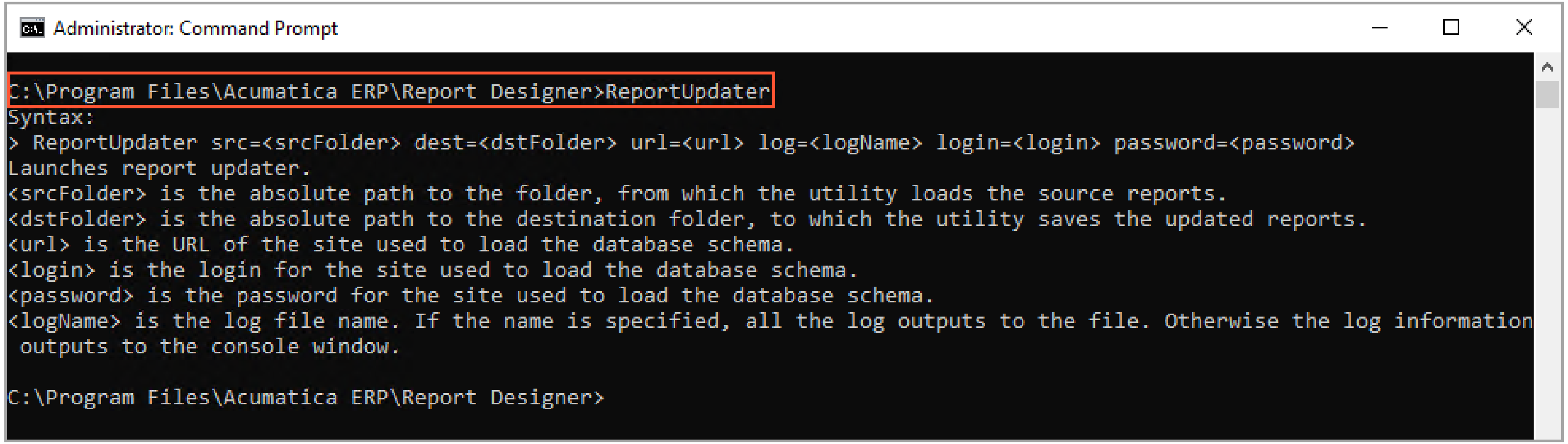
Select report tables:

AccBalanceByAssetFilter
AccessInfo
AccessInfoNotification
AccountByPeriodFilter
AccountByYearFilter
AccountClass
AccountFilter
AccountsPreloadFilter
AccountValidationRules
ActionDescriptionInsertModel
Actions
ActivationSettingsFilter

Search:

GLHistoryByPeriod
GLHistory
Account
Sub
GLSetup
Branch

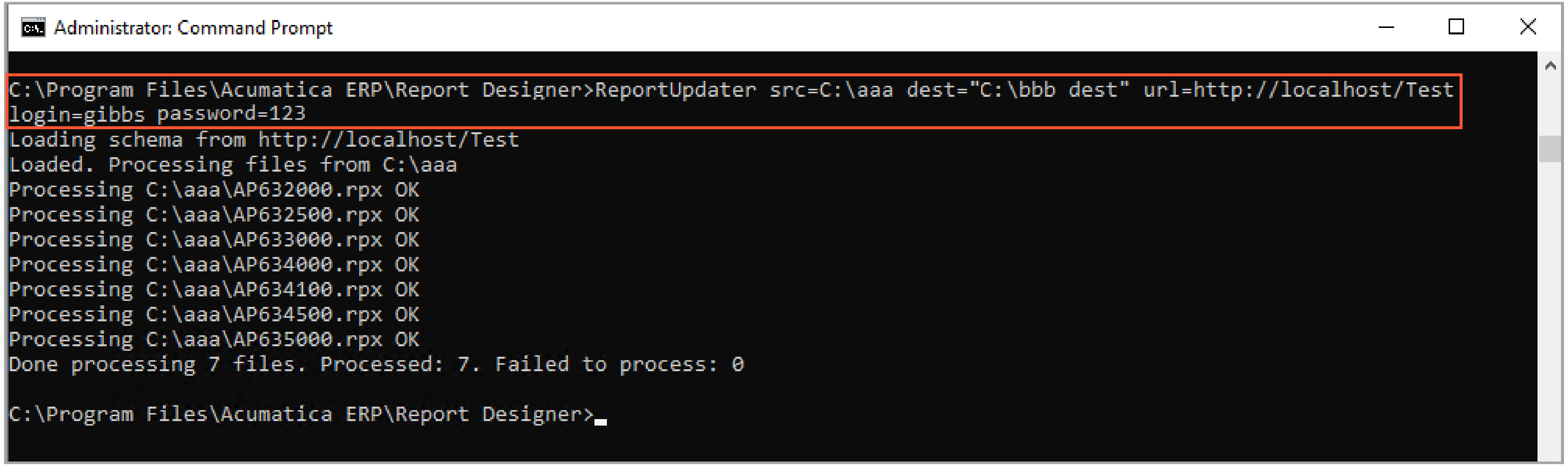
Figure: The ReportUpdater help information



```
C:\Program Files\Acumatica ERP\Report Designer>ReportUpdater
Syntax:
> ReportUpdater src=<srcFolder> dest=<dstFolder> url=<url> log=<logName> login=<login> password=<password>
Launches report updater.
<srcFolder> is the absolute path to the folder, from which the utility loads the source reports.
<dstFolder> is the absolute path to the destination folder, to which the utility saves the updated reports.
<url> is the URL of the site used to load the database schema.
<login> is the login for the site used to load the database schema.
<password> is the password for the site used to load the database schema.
<logName> is the log file name. If the name is specified, all the log outputs to the file. Otherwise the log information
 outputs to the console window.

C:\Program Files\Acumatica ERP\Report Designer>
```

Figure: The ReportUpdater utility in use



```
C:\Program Files\Acumatica ERP\Report Designer>ReportUpdater src=C:\aaa dest="C:\bbb dest" url=http://localhost/Test
login=gibbs password=123
Loading schema from http://localhost/Test
Loaded. Processing files from C:\aaa
Processing C:\aaa\AP632000.rpx OK
Processing C:\aaa\AP632500.rpx OK
Processing C:\aaa\AP633000.rpx OK
Processing C:\aaa\AP634000.rpx OK
Processing C:\aaa\AP634100.rpx OK
Processing C:\aaa\AP634500.rpx OK
Processing C:\aaa\AP635000.rpx OK
Done processing 7 files. Processed: 7. Failed to process: 0

C:\Program Files\Acumatica ERP\Report Designer>_
```

Report Creation: To Copy an Existing Report

Story

Suppose that you are a technical specialist in your company, and you are working on customizations. An accountant of the company has requested a report that displays the AR register. You have offered the accountant to use the AR Register (AR621500) report, which is a predefined report in Acumatica ERP, but the accountant has asked for modifications to the report. You have decided to make a copy of the report and change the copy in the Report Designer rather than directly modifying the predefined report. As a first step, you will create and save the copy of the report.

Day 2

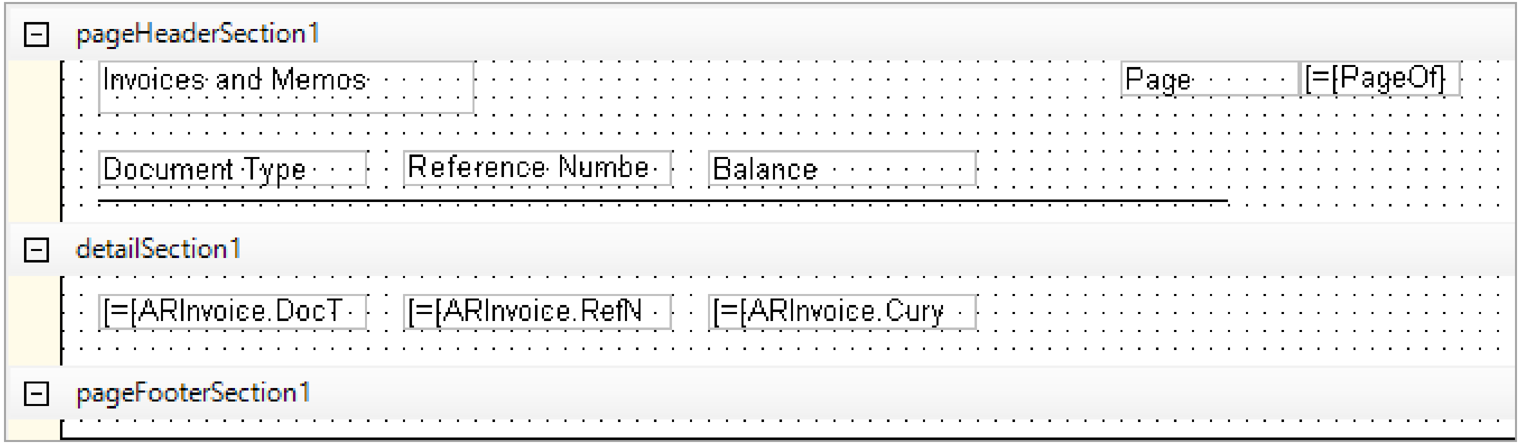
Jessica Korda
Acumatica Sponsored Professional Golfer

Report Creation: To Create a Report Based on One DAC

Story

Suppose that you are a technical specialist in your company who is working on simple customizations. An accountant of your company has requested a report that collects data about invoices and memos. The accountant wants a simple report that displays columns with the document type, the invoice reference number, and the balance of the invoice with that number.

Figure: The report designed to display data about invoices and memos



pageHeaderSection1

Invoices and Memos

Page [=[PageOf]

Document Type

Reference Numbe

Balance

detailSection1

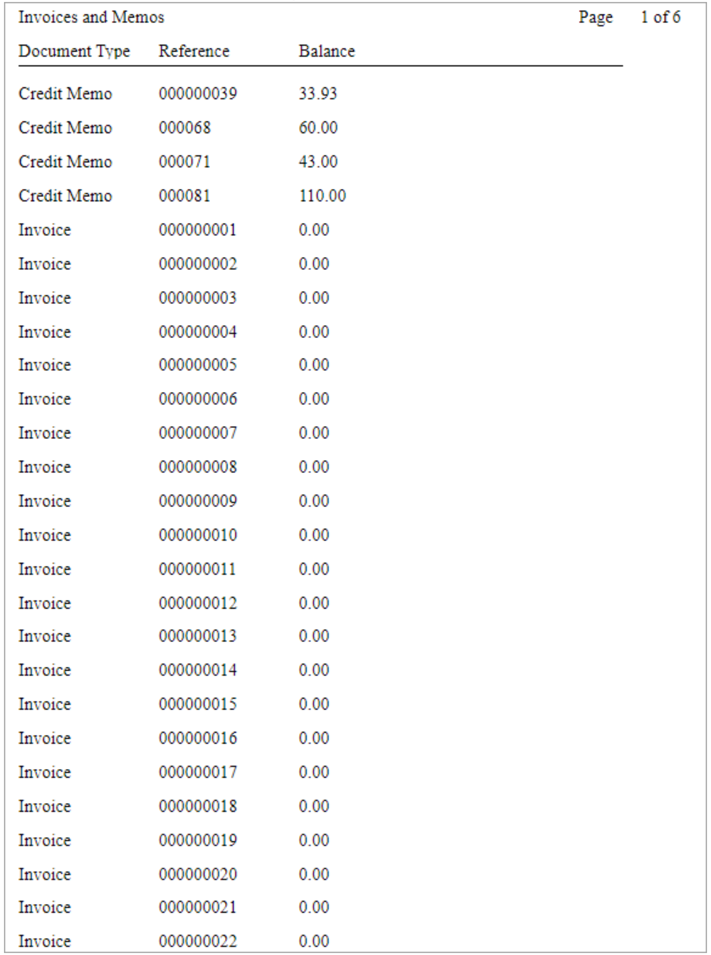
[=[ARInvoice.DocT

[=[ARInvoice.RefN

[=[ARInvoice.Cury

pageFooterSection1

Figure: The resulting report displaying information about invoices and memos



Invoices and Memos Page 1 of 6

Document Type	Reference	Balance
Credit Memo	000000039	33.93
Credit Memo	000068	60.00
Credit Memo	000071	43.00
Credit Memo	000081	110.00
Invoice	000000001	0.00
Invoice	000000002	0.00
Invoice	000000003	0.00
Invoice	000000004	0.00
Invoice	000000005	0.00
Invoice	000000006	0.00
Invoice	000000007	0.00
Invoice	000000008	0.00
Invoice	000000009	0.00
Invoice	000000010	0.00
Invoice	000000011	0.00
Invoice	000000012	0.00
Invoice	000000013	0.00
Invoice	000000014	0.00
Invoice	000000015	0.00
Invoice	000000016	0.00
Invoice	000000017	0.00
Invoice	000000018	0.00
Invoice	000000019	0.00
Invoice	000000020	0.00
Invoice	000000021	0.00
Invoice	000000022	0.00

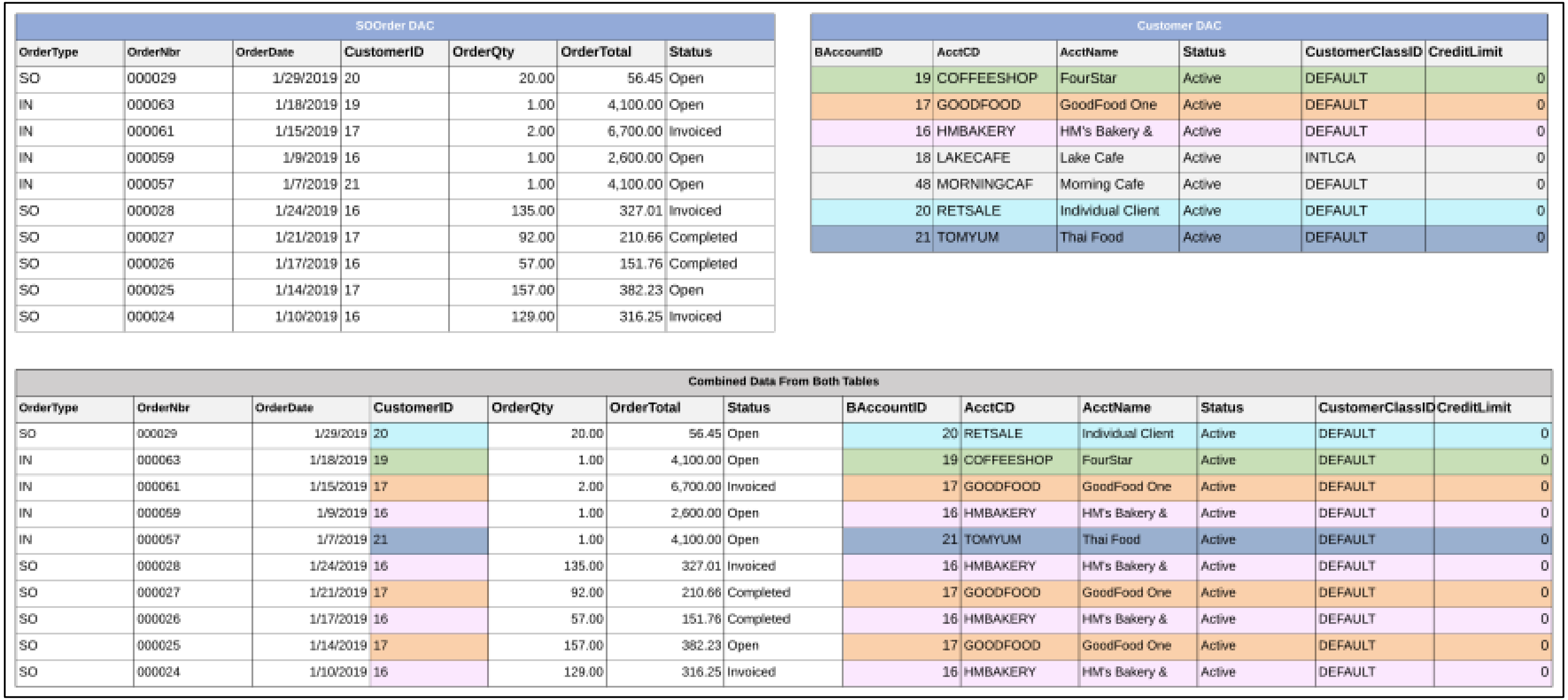
Lesson 4: Getting Data from Multiple DACs

Learning Objectives

In this lesson, you will learn how to do the following:

- Construct a data request to retrieve data from multiple data access classes.

Figure: Data combined from two tables

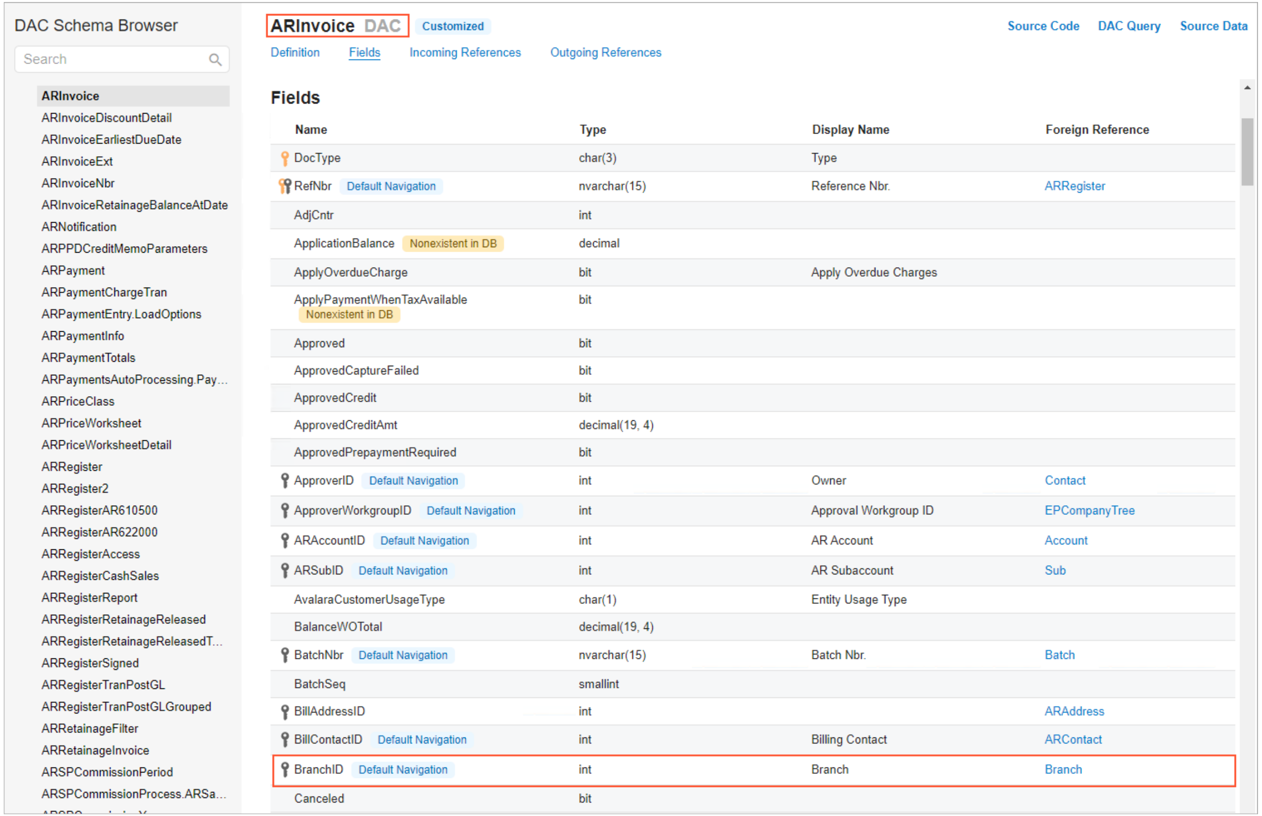


SOOrder DAC						
OrderType	OrderNbr	OrderDate	CustomerID	OrderQty	OrderTotal	Status
SO	000029	1/29/2019	20	20.00	56.45	Open
IN	000063	1/18/2019	19	1.00	4,100.00	Open
IN	000061	1/15/2019	17	2.00	6,700.00	Invoiced
IN	000059	1/9/2019	16	1.00	2,600.00	Open
IN	000057	1/7/2019	21	1.00	4,100.00	Open
SO	000028	1/24/2019	16	135.00	327.01	Invoiced
SO	000027	1/21/2019	17	92.00	210.66	Completed
SO	000026	1/17/2019	16	57.00	151.76	Completed
SO	000025	1/14/2019	17	157.00	382.23	Open
SO	000024	1/10/2019	16	129.00	316.25	Invoiced

Customer DAC					
BAccountID	AcctCD	AcctName	Status	CustomerClassID	CreditLimit
19	COFFEESHOP	FourStar	Active	DEFAULT	0
17	GOODFOOD	GoodFood One	Active	DEFAULT	0
16	HMBAKERY	HM's Bakery &	Active	DEFAULT	0
18	LAKECAFE	Lake Cafe	Active	INTLCA	0
48	MORNINGCAF	Morning Cafe	Active	DEFAULT	0
20	RETSALE	Individual Client	Active	DEFAULT	0
21	TOMYUM	Thai Food	Active	DEFAULT	0

Combined Data From Both Tables												
OrderType	OrderNbr	OrderDate	CustomerID	OrderQty	OrderTotal	Status	BAccountID	AcctCD	AcctName	Status	CustomerClassID	CreditLimit
SO	000029	1/29/2019	20	20.00	56.45	Open	20	RETSALE	Individual Client	Active	DEFAULT	0
IN	000063	1/18/2019	19	1.00	4,100.00	Open	19	COFFEESHOP	FourStar	Active	DEFAULT	0
IN	000061	1/15/2019	17	2.00	6,700.00	Invoiced	17	GOODFOOD	GoodFood One	Active	DEFAULT	0
IN	000059	1/9/2019	16	1.00	2,600.00	Open	16	HMBAKERY	HM's Bakery &	Active	DEFAULT	0
IN	000057	1/7/2019	21	1.00	4,100.00	Open	21	TOMYUM	Thai Food	Active	DEFAULT	0
SO	000028	1/24/2019	16	135.00	327.01	Invoiced	16	HMBAKERY	HM's Bakery &	Active	DEFAULT	0
SO	000027	1/21/2019	17	92.00	210.66	Completed	17	GOODFOOD	GoodFood One	Active	DEFAULT	0
SO	000026	1/17/2019	16	57.00	151.76	Completed	16	HMBAKERY	HM's Bakery &	Active	DEFAULT	0
SO	000025	1/14/2019	17	157.00	382.23	Open	17	GOODFOOD	GoodFood One	Active	DEFAULT	0
SO	000024	1/10/2019	16	129.00	316.25	Invoiced	16	HMBAKERY	HM's Bakery &	Active	DEFAULT	0

Figure: Details of the BranchID field



DAC Schema Browser

Search

- ARInvoice
- ARInvoiceDiscountDetail
- ARInvoiceEarliestDueDate
- ARInvoiceExt
- ARInvoiceNbr
- ARInvoiceRetainageBalanceAtDate
- ARNotification
- ARPPDCreditMemoParameters
- ARPayment
- ARPaymentChargeTran
- ARPaymentEntry.LoadOptions
- ARPaymentInfo
- ARPaymentTotals
- ARPaymentsAutoProcessing.Pay...
- ARPriceClass
- ARPriceWorksheet
- ARPriceWorksheetDetail
- ARRegister
- ARRegister2
- ARRegisterAR610500
- ARRegisterAR622000
- ARRegisterAccess
- ARRegisterCashSales
- ARRegisterReport
- ARRegisterRetainageReleased
- ARRegisterRetainageReleasedT...
- ARRegisterSigned
- ARRegisterTranPostGL
- ARRegisterTranPostGLGrouped
- ARRetainageFilter
- ARRetainageInvoice
- ARSPCommissionPeriod
- ARSPCommissionProcess.ARSa...

ARInvoice DAC Customized

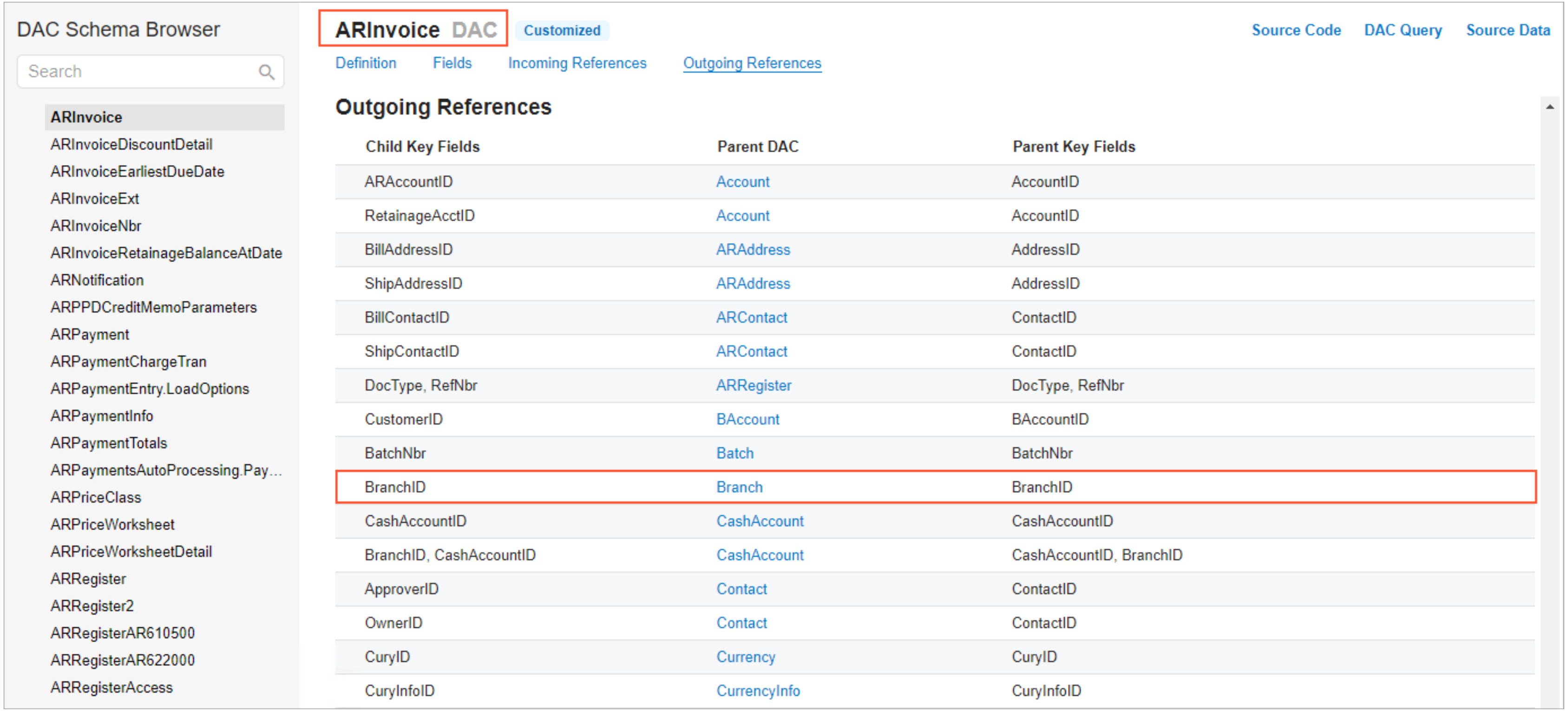
Source Code DAC Query Source Data

Definition Fields Incoming References Outgoing References

Fields

Name	Type	Display Name	Foreign Reference
DocType	char(3)	Type	
RefNbr Default Navigation	nvarchar(15)	Reference Nbr.	ARRegister
AdjCntr	int		
ApplicationBalance Nonexistent in DB	decimal		
ApplyOverdueCharge	bit	Apply Overdue Charges	
ApplyPaymentWhenTaxAvailable Nonexistent in DB	bit		
Approved	bit		
ApprovedCaptureFailed	bit		
ApprovedCredit	bit		
ApprovedCreditAmt	decimal(19, 4)		
ApprovedPrepaymentRequired	bit		
ApproverID Default Navigation	int	Owner	Contact
ApproverWorkgroupID Default Navigation	int	Approval Workgroup ID	EPCompanyTree
ARAccountID Default Navigation	int	AR Account	Account
ARSubID Default Navigation	int	AR Subaccount	Sub
AvalaraCustomerUsageType	char(1)	Entity Usage Type	
BalanceWOTotal	decimal(19, 4)		
BatchNbr Default Navigation	nvarchar(15)	Batch Nbr.	Batch
BatchSeq	smallint		
BillAddressID	int		ARAddress
BillContactID Default Navigation	int	Billing Contact	ARContact
BranchID Default Navigation	int	Branch	Branch
Canceled	bit		

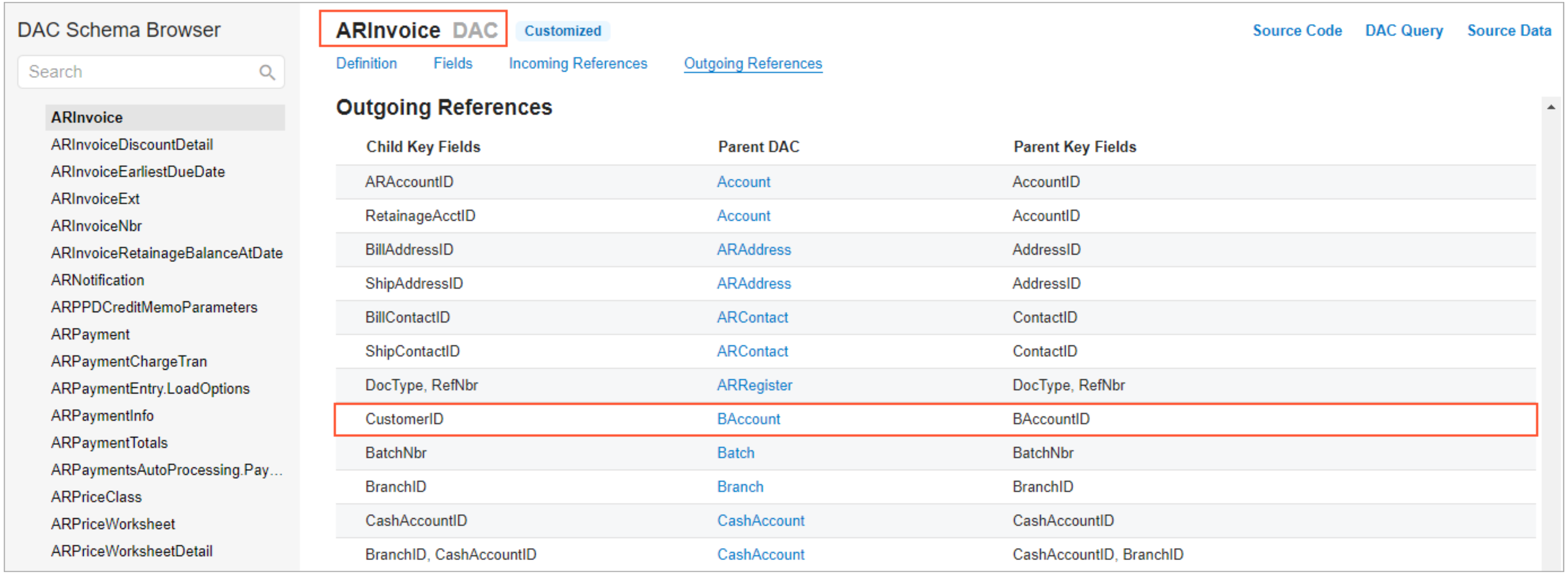
Figure: The information about the Branch DAC and its key field



Outgoing References

Child Key Fields	Parent DAC	Parent Key Fields
ARAccountID	Account	AccountID
RetainageAcctID	Account	AccountID
BillAddressID	ARAddress	AddressID
ShipAddressID	ARAddress	AddressID
BillContactID	ARContact	ContactID
ShipContactID	ARContact	ContactID
DocType, RefNbr	ARRegister	DocType, RefNbr
CustomerID	BAccount	BAccountID
BatchNbr	Batch	BatchNbr
BranchID	Branch	BranchID
CashAccountID	CashAccount	CashAccountID
BranchID, CashAccountID	CashAccount	CashAccountID, BranchID
ApproverID	Contact	ContactID
OwnerID	Contact	ContactID
CuryID	Currency	CuryID
CuryInfoID	CurrencyInfo	CuryInfoID

Figure: The information about BAccount DAC and its key field



Outgoing References

Child Key Fields	Parent DAC	Parent Key Fields
ARAccountID	Account	AccountID
RetainageAcctID	Account	AccountID
BillAddressID	ARAddress	AddressID
ShipAddressID	ARAddress	AddressID
BillContactID	ARContact	ContactID
ShipContactID	ARContact	ContactID
DocType, RefNbr	ARRegister	DocType, RefNbr
CustomerID	BAccount	BAccountID
BatchNbr	Batch	BatchNbr
BranchID	Branch	BranchID
CashAccountID	CashAccount	CashAccountID
BranchID, CashAccountID	CashAccount	CashAccountID, BranchID

Figure: Exploration of a data access class on the Source Code form

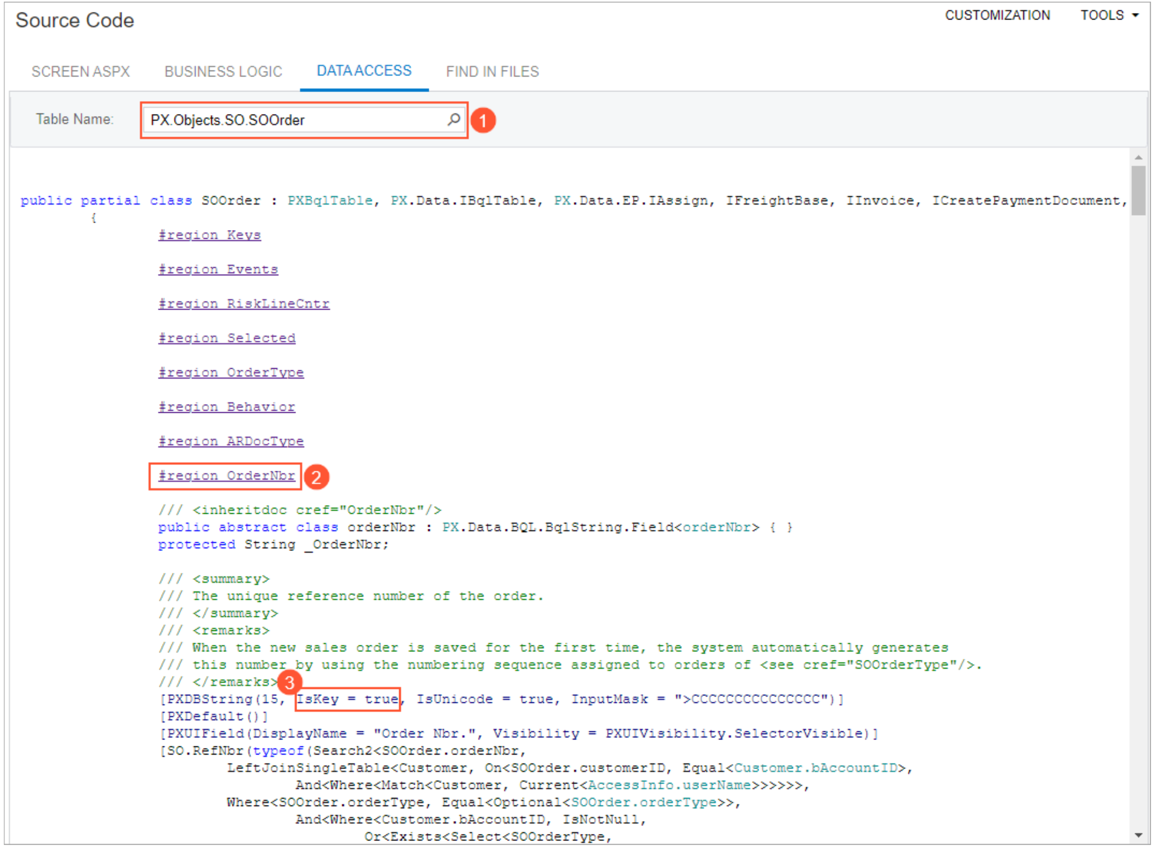


Table Name: PX.Objects.SO.SOOrder

```
public partial class SOOrder : PXBqlTable, PX.Data.IBqlTable, PX.Data.EP.IAssign, IFreightBase, IInvoice, ICreatePaymentDocument,
        {
                #region Keys

                #region Events

                #region RiskLineCntr

                #region Selected

                #region OrderType

                #region Behavior

                #region ARDocType

                #region OrderNbr

                /// <inheritdoc cref="OrderNbr"/>
                public abstract class orderNbr : PX.Data.BQL.BqlString.Field<orderNbr> { }
                protected String _OrderNbr;

                /// <summary>
                /// The unique reference number of the order.
                /// </summary>
                /// <remarks>
                /// When the new sales order is saved for the first time, the system automatically generates
                /// this number by using the numbering sequence assigned to orders of <see cref="SOOrderType"/>.
                /// </remarks>
                [PXDBString(15, IsKey = true, IsUnicode = true, InputMask = ">CCCCCCCCCCCCCCC")]
                [PXDefault()]
                [PXUIField(DisplayName = "Order Nbr.", Visibility = PXUIVisibility.SelectorVisible)]
                [SO.RefNbr(typeof(Search2<SOOrder.orderNbr,
                        LeftJoinSingleTable<Customer, On<SOOrder.customerID, Equal<Customer.bAccountID>,
                                And<Where<Match<Customer, Current<AccessInfo.userName>>>>>>,
                        Where<SOOrder.orderType, Equal<Optional<SOOrder.orderType>>,
                                And<Where<Customer.bAccountID, IsNotNull,
                                        Or<Exists<Select<SOOrderType,
```

Acumatica
The Cloud ERP

Thank you!

Johans Saavedra