



simplanova

Description of Simplanova
Report Converter Conversion
Logic

Contents

1.	Conversion of report properties, triggers and C/AL globals	4
1.1.	Conversion of report properties	4
1.2.	Conversion of report triggers and C/AL globals	8
2.	Conversion of Dataltem properties and triggers	10
2.1.	Conversion of Dataltem properties	10
2.2.	Conversion of Dataltem trigger	13
3.	Section Conversion	15
3.1.	Section Conversion and Grouping	15
3.1.1.	Header section	15
3.1.2.	TransHeader/TransFooter section	19
3.1.3.	GroupHeader/GroupFooter Section	19
3.1.4.	Body section	21
3.1.5.	Footer section	23
3.2.	Section Trigger Conversion	27
3.3.	Section Control Conversion	29
3.3.1.	Label Conversion	30
3.3.2.	Text Box Conversion	32
3.3.3.	Picture Box Conversion	38
3.3.4.	Image Conversion	41
3.3.5.	Shape Conversion	44
4.	Request Form Conversion	48
4.1.	Request Form Property Conversion	48
4.2.	Request Form Trigger Conversion	48
4.3.	Request Form Control Conversion	49
4.3.1.	Property Conversion	49
4.3.2.	Trigger	59
5.	Conversion of mostly used functions	62
5.1.	CurrReport.CREATETOTALS	62
5.1.1.	Sum() function in layout expressions	62
5.1.2.	“SRB_Total” C/AL global variables	63
5.2.	CurrReport.SHOWOUTPUT	65

5.3.	CurrReport.PAGENO	67
5.3.1.	CurrReport.PAGENO used in control SourceExpr	68
5.3.2.	CurrReport.PAGENO used with CurrReport.SHOWOUTPUT	68
5.4.	CurrReport.NEWPAGE.....	69
5.4.1.	CurrReport.NEWPAGE used with „DataItemTable“.“Field Name“	69
5.4.2.	CurrReport.NEWPAGE used on OnAfterGetRecord trigger without „DataItemTable“.“Field Name“	72
5.5.	CurrReport.NEWPAGEPERRECORD	75
5.6.	EXIT.....	76
5.7.	ISSERVICETIER	77
6.	Conversion Options.....	79

1. Conversion of report properties, triggers and C/AL globals

This section explains the main rules of report properties, triggers, C/AL global variables, text constants and functions conversion from Classic Dynamic NAV to RDLC format report.

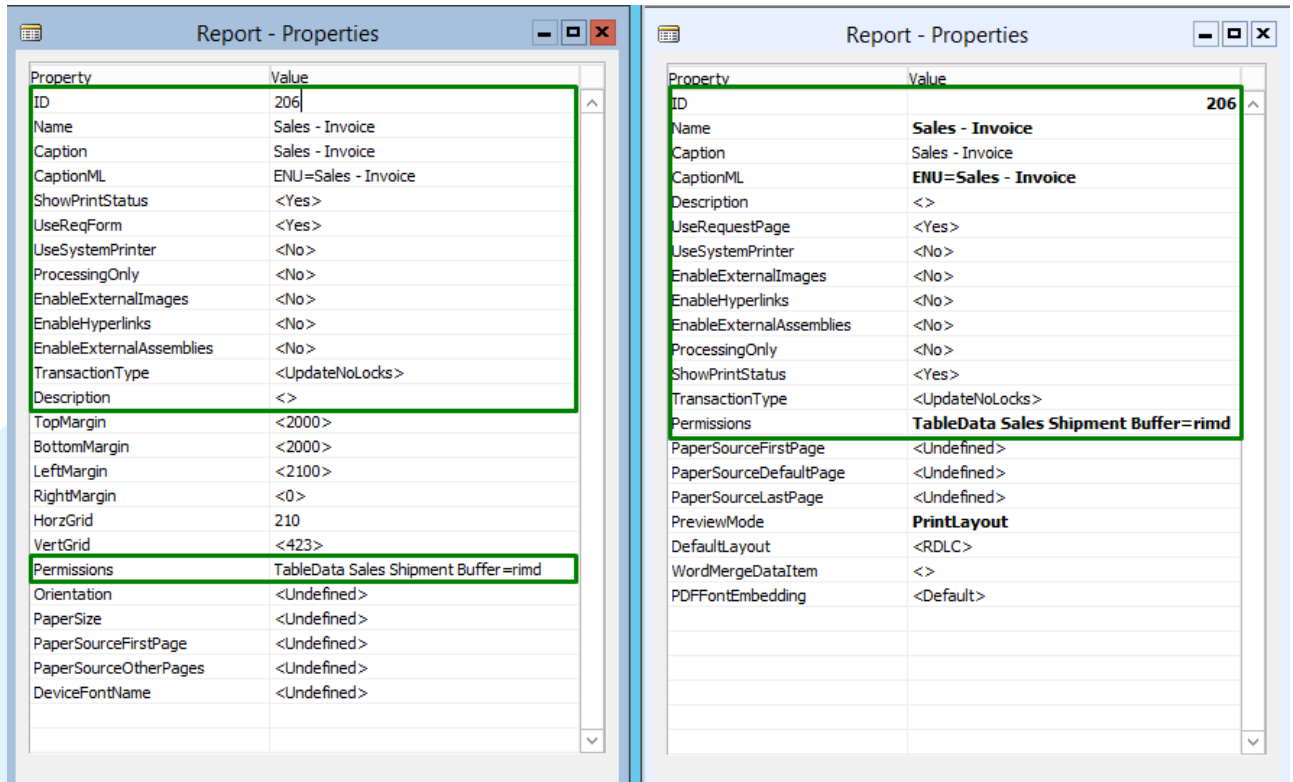
1.1. Conversion of report properties

If the same report property exist in both versions, the value of Classic Dynamic NAV report property is transferred to the same RDLC format report property with no changes.

List of Classic Dynamic NAV report properties which are transferred to RDLC format report properties:

Classic Report Property	RDLC Report Property
ID	ID
Name	Name
Caption	Caption
CaptionML	CaptionML
ShowPrintStatus	ShowPrintStatus
UseReqForm	UseRequestPage
UseSystemPrinter	UseSystemPrinter
ProcessingOnly	ProcessingOnly
EnableExternalImages	EnableExternalImages
EnableHyperlinks	EnableHyperlinks
EnableExternalAssemblies	EnableExternalAssemblies
TransactionType	TransactionType
Description	Description
Permissions	Permissions
PaperSourceFirstPage	PaperSourceFirstPage
PaperSourceOtherPages	PaperSourceDefaultPage and PaperSourceLastPage

Figure 1. Classic Dynamic NAV report properties which are transferred to RDLC format report properties.

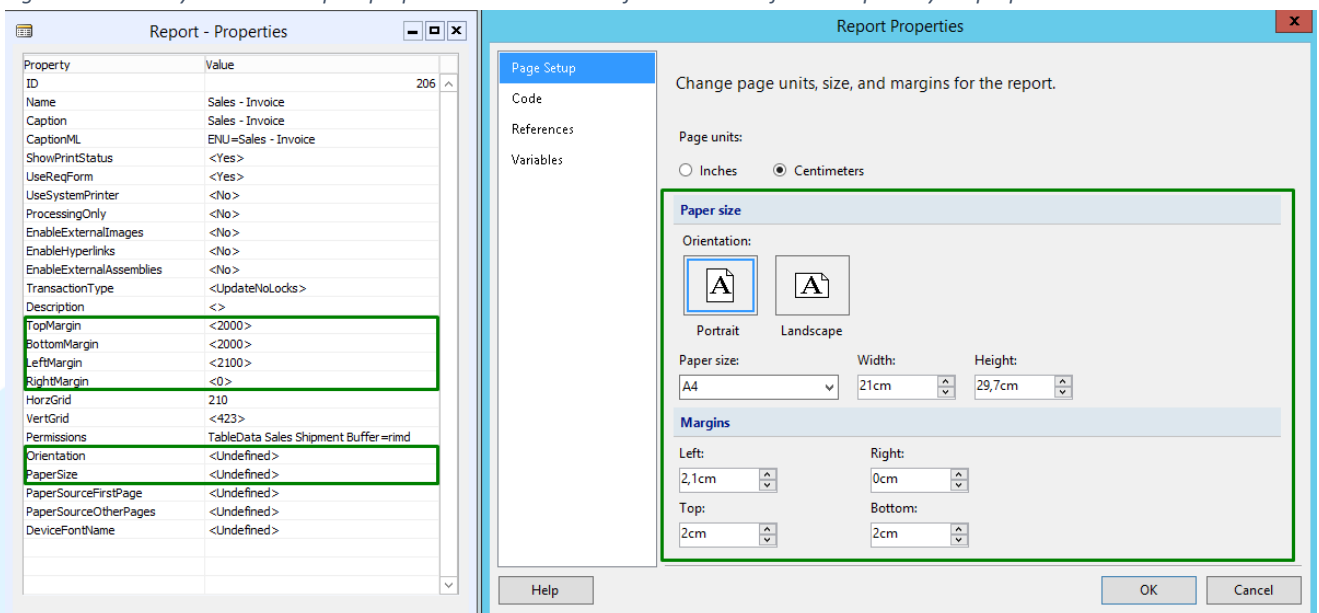


There are some Classic Dynamic NAV report properties which are transferred to RDLC format report layout, as RDLC format reports do not have such report properties, but have the same properties in layout part.

List of Classic Dynamic NAV report properties which are transferred to RDLC format report layout properties:

Classic Report Property	RDLC Report Layout Property
TopMargin	Top
BottomMargin	Bottom
LeftMargin	Left
RightMargin	Right
Orientation	Orientation
PaperSize	Paper size

Figure 2. Classic Dynamic NAV report properties which are transferred to RDLC format report layout properties.

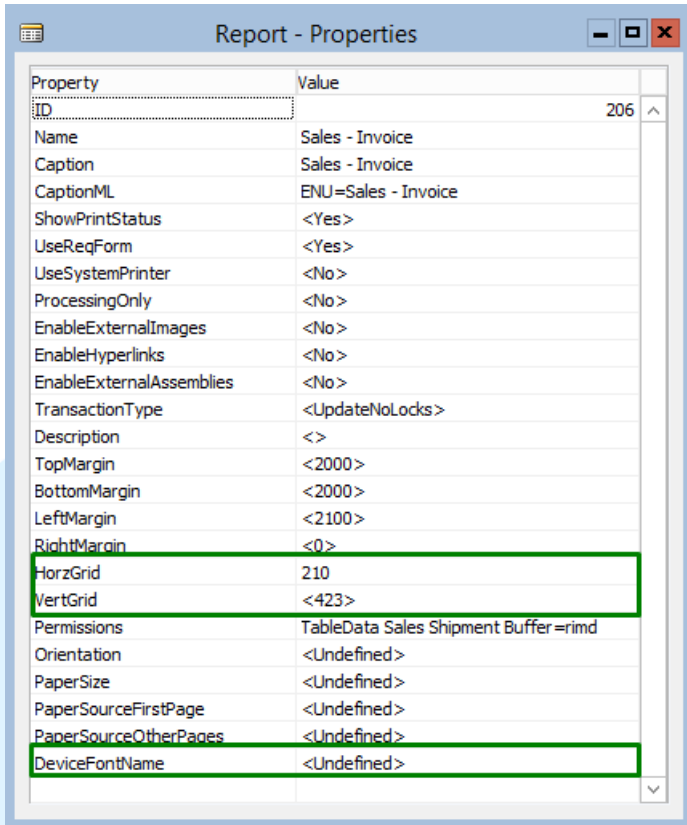


Some Classic Dynamic NAV report properties are no longer relevant, so they are not transferred to RDLC format report.

List of Classic Dynamic NAV report properties which are not transferred to RDLC format report:

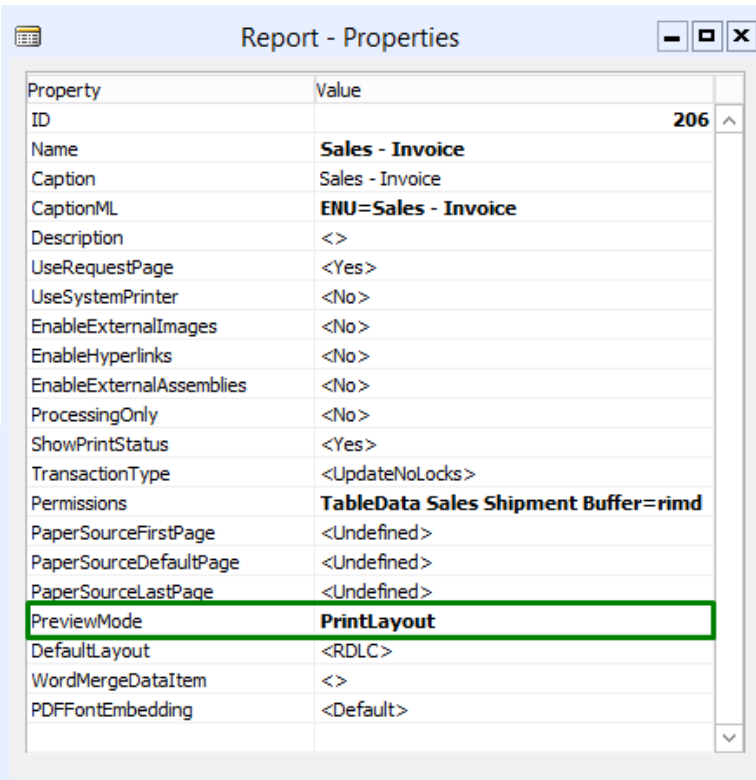
Classic Report Property
HorzGrid
VertGrid
DeviceFontName

Figure 3. Classic Dynamic NAV report properties which are not transferred to RDLC format report.



RDLC format reports have some properties which does not exists in Classic Dynamic NAV reports. Default values are assigned to all these properties after conversion, except PreviewMode property. Default value of PreviewMode property is "Normal", but during conversion the value is changed to "PrintLayout".

Figure 4. RDLC format report PreviewMode property after conversion.



1.2. Conversion of report triggers and C/AL globals

As performance of report triggers have not changed between Classic Dynamic NAV and RDLC format report versions, during the conversion all the code from Classic Dynamic NAV report trigger is transferred to the same RDLC format report trigger together with C/AL local variables. Classic Dynamic NAV report triggers OnCreateHyperlink() and OnHyperlink() are not supported on RDLC format reports, so all the code and C/AL local variables are not transferred to RDLC format reports.

List of Classic Dynamic NAV report triggers which are transferred to proper RDLC format report triggers:

Classic Report Trigger	RDLC Report Trigger
Documentation()	Documentation()
OnInitReport()	OnInitReport()
OnPreReport()	OnPreReport()
OnPostReport()	OnPostReport()
OnCreateHyperlink()	-
OnHyperlink()	-

All C/AL global variables, text constants and functions are transferred from Classic Dynamic NAV to RDLC format version exactly. If there are some variables which refer to obsolete fields or tables (for example, table ID 359 'Posted Document Dimension') after conversion variables still refer to the same fields and tables. All the obsolete functionality compilation errors have to be fixed manually.

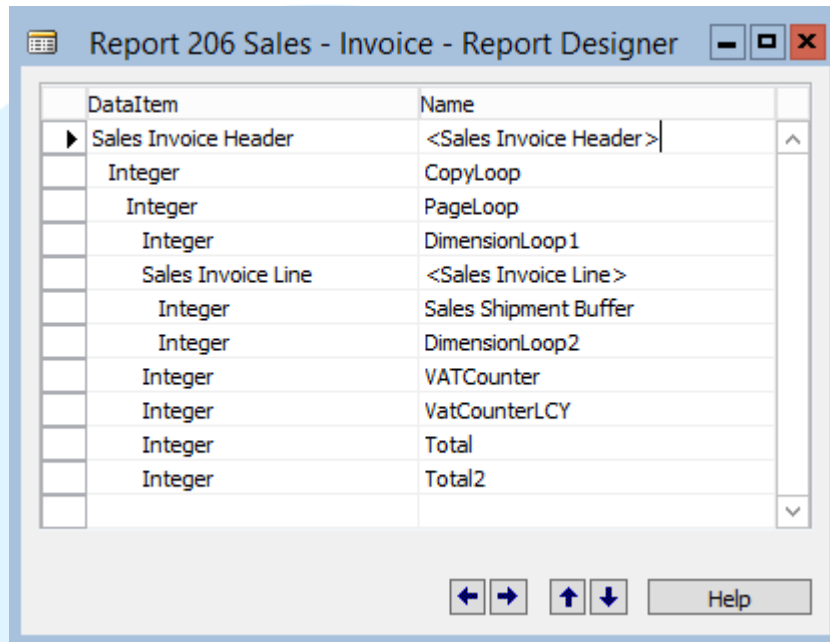
2. Conversion of DataItem properties and triggers

The following section explains the main rules of DataItem properties and triggers conversion from Classic Dynamics NAV to RDLC format report.

2.1. Conversion of DataItem properties

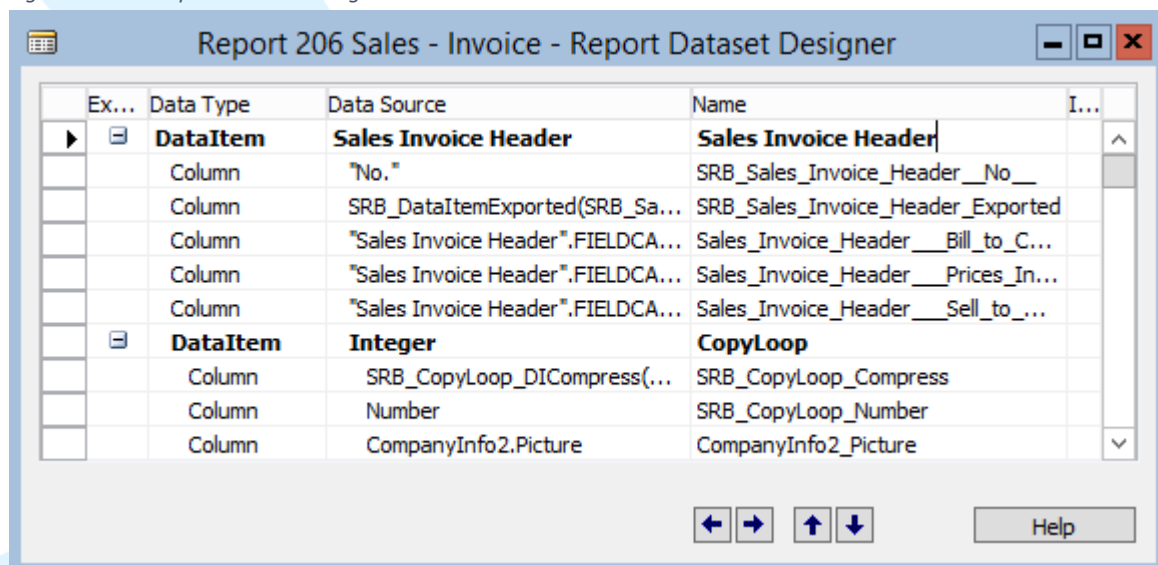
The structure of report designer where all the report DataItems are listed has changed. In Classic Dynamic NAV there is Report Designer, the place where only DataItems and their properties are defined.

Figure 5. Classic Report Designer.



In RDLC format reports there is Report Dataset Designer, the place where all DataItems, DataItem properties, columns and column properties are defined.

Figure 6. RDLC Report Dataset Designer.



The value of Classic Report Designer DataItem is transferred to “Data Source” where “Data Type” = DataItem in Report Dataset Designer (this is the same as DataItem property DataItemTable). The value of DataItem Name in Report Designer is transferred to Name where “Data Type” = DataItem in Report Dataset Designer.

If the same DataItem property exist in both versions, the value of DataItem property from Classic Dynamics NAV report is transferred to the same RDLC format report DataItem property with no changes.

List of Classic Dynamics NAV report DataItem properties which are transferred to RDLC format report DataItem properties:

Classic DataItem Property	RDLC DataItem Property
DataItemIndent	Indentation
DataItemTable	DataItemTable
DataItemTableView	DataItemTableView
DataItemLinkReference	DataItemLinkReference
DataItemLink	DataItemLink
ReqFilterHeading	ReqFilterHeading
ReqFilterHeadingML	ReqFilterHeadingML
ReqFilterFields	ReqFilterFields
CalcFields	CalcFields
MaxIteration	MaxIteration
DataItemVarName	Name
PrintOnlyIfDetail	PrintOnlyIfDetail

Figure 7. Classic Dynamics NAV report DataItem properties which are transferred to RDLC format report DataItem properties.

Property	Value
DataItemIndent	<0>
DataItemTable	Sales Invoice Header
DataItemTableView	SORTING(No.)
DataItemLinkReference	<Undefined>
DataItemLink	<Undefined>
NewPagePerGroup	<No>
NewPagePerRecord	Yes
ReqFilterHeading	Posted Sales Invoice
ReqFilterHeadingML	ENU=Posted Sales Invoice
ReqFilterFields	No.,Sell-to Customer No.,N...
TotalFields	<Undefined>
GroupTotalFields	<Undefined>
CalcFields	<Undefined>
MaxIteration	<0>
DataItemVarName	<Sales Invoice Header>
PrintOnlyIfDetail	<No>

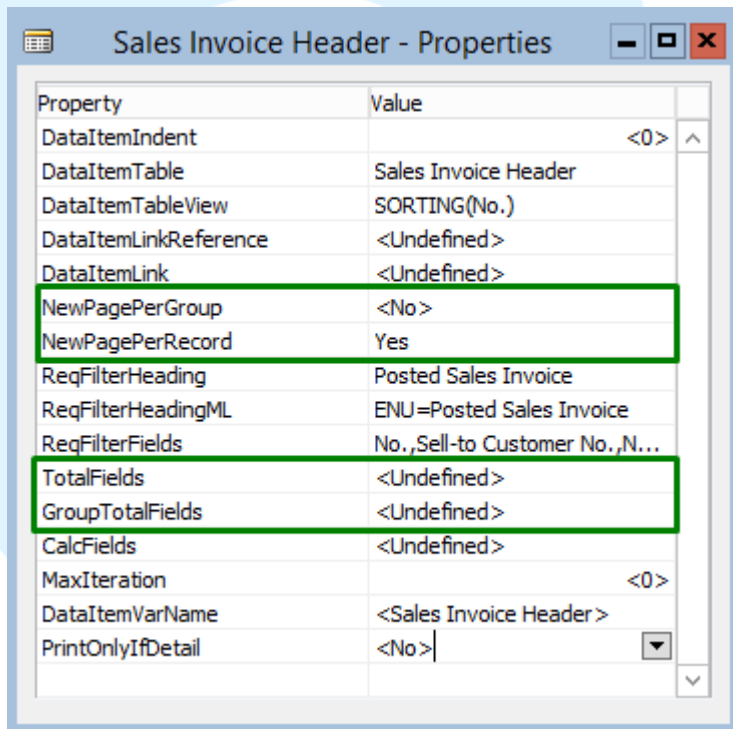
Property	Value
ID	9516001
Name	Sales Invoice Header
Indentation	0
DataItemTable	Sales Invoice Header
DataItemTableView	SORTING(No.)
DataItemLinkReference	<>
DataItemLink	<Undefined>
ReqFilterHeading	Posted Sales Invoice
ReqFilterHeadingML	ENU=Posted Sales Invoice
ReqFilterFields	No.,Sell-to Customer No.,No...
CalcFields	<Undefined>
MaxIteration	<0>
PrintOnlyIfDetail	<No>
Temporary	<No>

There are some Classic Dynamics NAV report Dataltem properties which are transferred to RDLC format report layout, as RDLC format reports do not have such properties, but have similar properties or functions in layout part.

List of Classic Dynamics NAV report Dataltem properties which are implemented in RDLC format report layout:

Classic Dataltem Property
NewPagePerGroup
NewPagePerRecord
TotalFields
GroupTotalFields

Figure 8. Classic Dynamics NAV report Dataltem properties which are implemented in RDLC format report layout.



NewPagePerGroup Dataltem property. This property specifies if page break is generated after each group defined in GroupTotalFields property. RDLC format reports does not have the same Dataltem property, but the same functionality is implemented in layout by using BreakLocation = Between (Group Properties -> PageBreaks -> BreakLocation) for a group defined in GroupTotalFields property.

NewPagePerRecord Dataltem property. This property specifies if page break is generated after each record. RDLC format reports does not have the same Dataltem property, but the same functionality is implemented in layout by using BreakLocation = Between (Group Properties ->

PageBreaks -> BreakLocation) for Dataltem body group (group where grouping is defined by DataltemTable primary key).

TotalFields Dataltem property. This property is used to specify the fields for which totals should be maintained. RDLC format reports does not have the same Dataltem property, but the same functionality is implemented by using Sum() function in RDLC format report layout for a field defined in TotalFields property or by adding global variables and performing calculations if Sum() function is not proper to use.

GroupTotalFields Dataltem property. This property specifies fields by which grouping should be performed in GroupHeader and GroupFooter sections. RDLC format reports does not have the same Dataltem property, but the same functionality is implemented by using fields listed in GroupTotalFields property as a grouping expressions in layout groups.

2.2. Conversion of Dataltem trigger

As performance of Dataltem triggers have not changed between Classic Dynamics NAV and RDLC format versions, during the conversion all the code from Classic Dynamics NAV report Dataltem trigger is transferred to the same RDLC format report Dataltem trigger together with C/AL local variables.

List of Classic Dynamics NAV report Dataltem triggers which code is transferred to the same RDLC format report Dataltem triggers:

Classic Dataltem Trigger	RDLC Dataltem Trigger
OnPreDataltem()	OnPreDataltem()
OnAfterGetRecord()	OnAfterGetRecord()
OnPostDataltem()	OnPostDataltem()

All the code from Classic Dynamics NAV report OnAfterGetRecord() trigger is added to BEGIN END sentence in order to avoid converted report compilation errors because of missing semicolon, which can occur as some implementation code (for example, function CurrReport.CREATETOTALS() implementation code) is usually added to the end of OnAfterGetRecord() trigger.

Figure 9. Classic Dynamics NAV report OnAfterGetReccord() trigger.

```
Sales Invoice Line - OnAfterGetRecord()  
PostedShipmentDate := 0D;  
IF Quantity <> 0 THEN  
    PostedShipmentDate := FindPostedShipmentDate;  
  
IF (Type = Type::"G/L Account") AND (NOT ShowInternalInfo) THEN  
    "No." := ''  
  
Sales Invoice Line - OnPostDataItem()
```

Figure 10. RDLC format report OnAfterGetRecord() trigger after conversion.

```

Sales Invoice Line - OnAfterGetRecord()
SRB_Sales_Invoice_Line_Body2_Visible := TRUE;
SRB_Sales_Invoice_Line_Body1_Visible := TRUE;
SRB_Sales_Invoice_Line_Exported := TRUE;
BEGIN
    PostedShipmentDate := 0D;
    IF Quantity <> 0 THEN
        PostedShipmentDate := FindPostedShipmentDate;
        IF (Type = Type::"G/L Account") AND (NOT ShowInternalInfo) THEN
            "No." := ''
        END;
    END;
    SRB_Sales_Invoice_Line_Amount_Total += Amount;
    SRB_Sales_Invoice_Line_Amount_Including_VAT_Total += "Amount Including VAT";
    Sales_Invoice_Line_Footer3_OnPreSection("Sales Invoice Line");
Sales Invoice Line - OnPostDataItem()
```

3. Section Conversion

This part explains the main rules about section, grouping, section trigger and control conversion from Classic Dynamics NAV to RDLC format report.

3.1. Section Conversion and Grouping

After conversion all Classic Dynamics NAV report sections are transformed to one RDLC format report layout tablix. Each section is converted to one tablix row and also rectangular control is added to the row as it would be possible to place text boxes and other controls on it. There are some different conversion features and rules depending on section type.

3.1.1. Header section

Classic Dynamics NAV report header type section can be converted to RDCL format report tablix row if this section has to be printed only on the first page (PrintOnEveryPage section property has NO value) or converted to report header if header section has to be printed on every page (PrintOnEveryPage section property has YES value).

- Example 1. Conversion of header section which PrintOnEveryPage property has NO value.

If Classic Dynamics NAV report header section's PrintOnEveryPage property has NO value,

Customer, Header (1)

Customer

=<COMPANYNAME>

=<FORMAT(TODAY,0,4)>

Page =...

=<USERID>

Customer, Header (2)

No.	Name	Amount
-----	------	--------

Customer, Body (3)

No.	Name	Amount
-----	------	--------

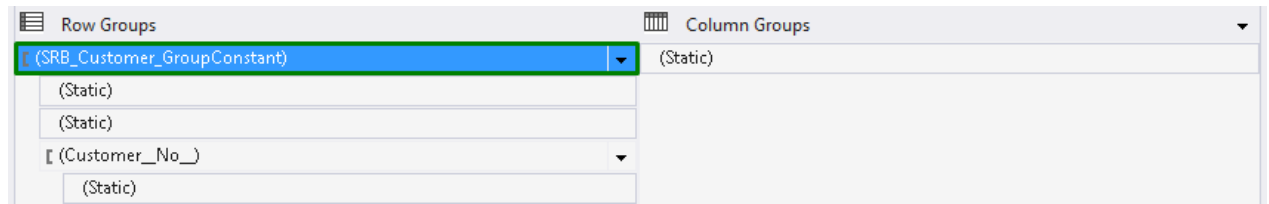
Customer, Header (1) - Properties

Property	Value
PrintOnEveryPage	No
SectionWidth	18150
SectionHeight	1269

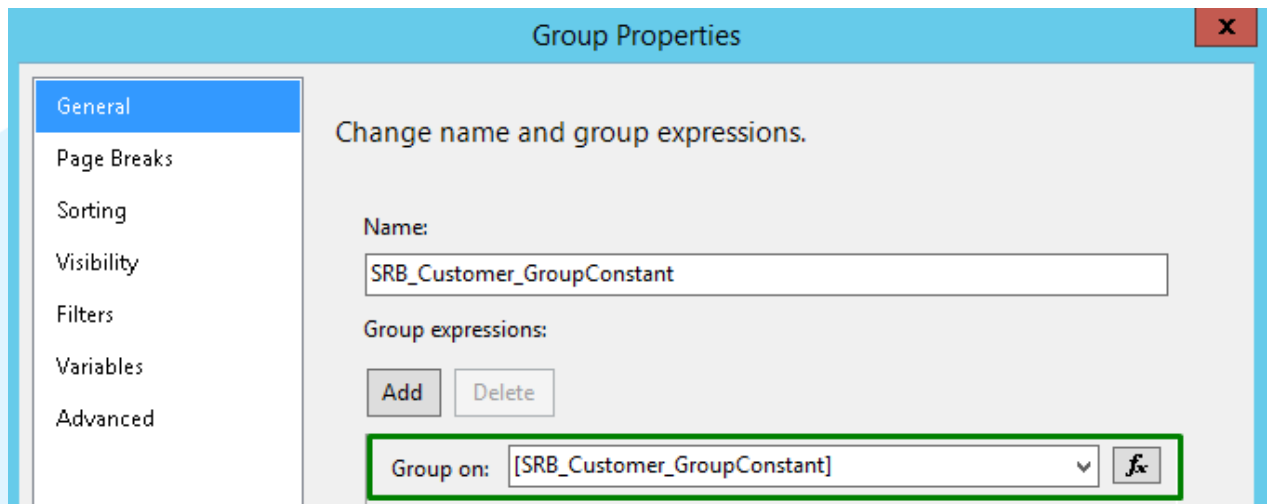
during the conversion there is column for header group created in Report Dataset Designer

Exp...	Data Type	Data Source	Name	I...
	DataItem	Customer	Customer	
	Column	'0'	SRB_Customer_GroupConstant	
	Column	"No."	SRB_Customer__No__	
	Column	Name	SRB_Customer_Name	

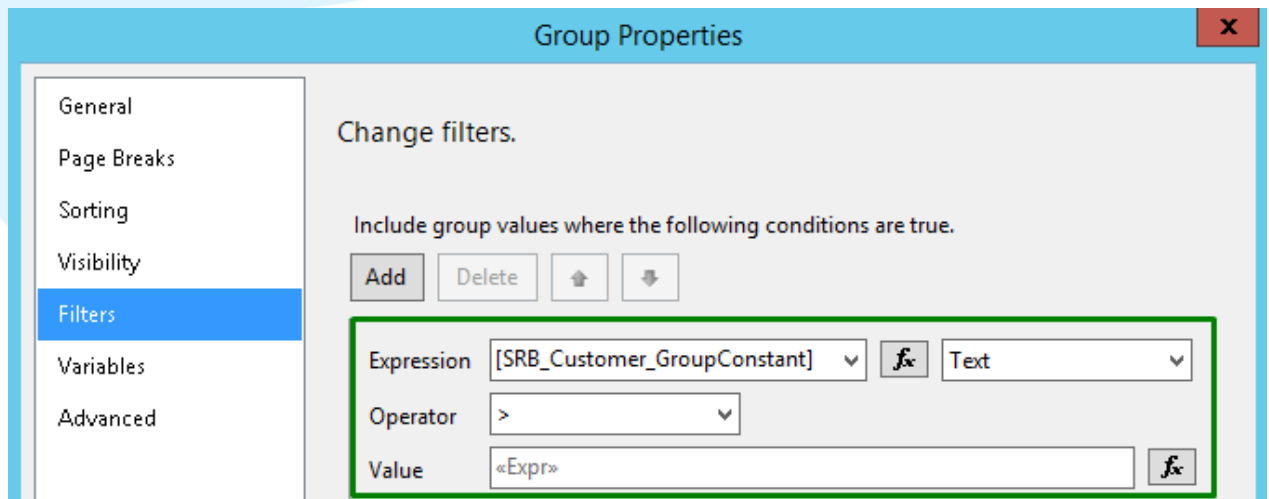
and header group created in Report Layout Designer.



Header group has grouping expression of dataset column with the value of constant '0'



and filter expression which prevents from showing empty values
(Fields!SRB_Customer_GroupConstant.Value >= "").



There is a tablix row created for header section and controls are placed on it. Header row has RepeatOnNewPage value set to false in order to be shown only on the first page. If CurrReport.PAGENO function is used in some text box SourceExpr property in Classic Dynamics NAV report header, during the conversion this function is changed to constant value of "1".

[CustomerCaption]		[&ExecutionTime]
Customer_COMPANYNAME]		[@Cus[Cu
		[&UserID]
@Custome	[@Customer_NameCaption]	[@Customer_f
Customer_	[Customer_Name]	Customer_Amount]

- Example 2. Conversion of header section which PrintOnEveryPage property has YES value.

If Classic Dynamics NAV report header section's PrintOnEveryPage property has YES value,

> > PageLoop, Header (1)

=<STRSUBSTNO(DocumentCaption...	
Page	=<CurrReport.PAGENO>
Invoice No.	=<"Sales In...
Posting Date	=<FORMAT(...
Due Date	=<FORMAT(...

> > > Sales Invoice Line, Header (1)

No.	Description	Quantity	Unit of Measure	Unit Price	Amount
-----	-------------	----------	-----------------	------------	--------

> > > Sales Invoice Line, Body (2)

=<Description>

PageLoop, Header (1) - Properties

Property	Value
PrintOnEveryPage	Yes
SectionWidth	18060
SectionHeight	2538

during the conversion there is column for header group created in Report Dataset Designer

Exp...	Data Type	Data Source	Name	I...
	DataItem	Integer	PageLoop	
	Column	'0'	SRB_PageLoop_GroupConstant	
	Column	Number	SRB_PageLoop_Number	
	Column	SRB_DataItemExported(SRB_PageLoop_Exported)	SRB_PageLoop_Exported	

and header group created in Report Layout Designer.

Row Groups

(SRB_PageLoop_GroupConstant)

(Static)

(PageLoop_Number)

(SRB_Sales_Invoice_Line_GroupConstant)

(Static)

(Sales_Invoice_Line_Document_No__Sal...

Column Groups

(Static)

Header group has grouping expression of dataset column with the value of constant '0'

Group Properties

General

Page Breaks

Sorting

Visibility

Filters

Variables

Advanced

Change name and group expressions.

Name:

SRB_PageLoop_GroupConstant

Group expressions:

Add Delete

Group on: [SRB_PageLoop_GroupConstant] fx

and filter expression which prevents from showing empty values (Fields!SRB_Customer_GroupConstant.Value >= "").

Group Properties

General

Page Breaks

Sorting

Visibility

Filters

Variables

Advanced

Change filters.

Include group values where the following conditions are true.

Add Delete Up Down

Expression [SRB_PageLoop_GroupConstant] fx Text

Operator >

Value = "" fx

There is a tablix row for header group created with red HeaderData1 text box in it for listing values which will be shown in report header.

<<Expr>>

<Expr>

&PageNumber

<Expr>

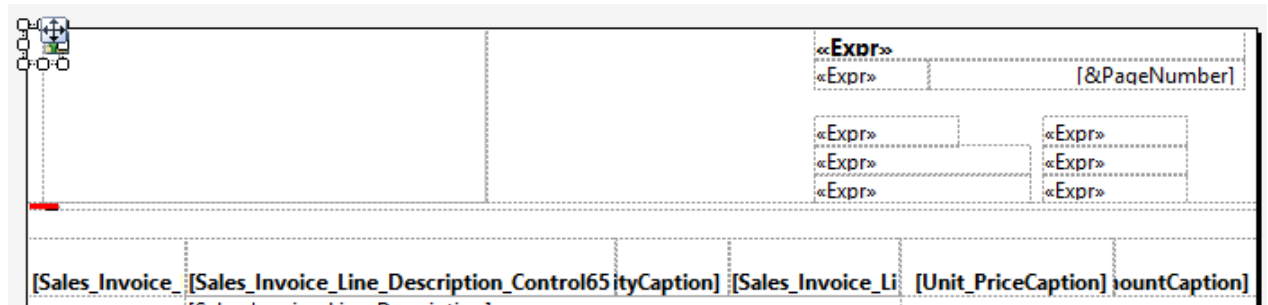
<Expr>

<Expr>

[Sales_Invoice_Line_Description_Control65ityCaption] [Sales_Invoice_Li] [Unit_PriceCaption] [ountCaption]

For red text box with header values listed there is setting text box HeaderData1SetTextBox created in report header's top left corner. Rectangular with header section controls is placed in report header and controls use GetData function to get values from HeaderData1 text box with

the help of HeaderData1SetTextBox text box. If CurrReport.PAGENO function is used in some text box SourceExpr property in Classic Dynamics NAV report, during the conversion this function is changed to Globals!PageNumber function.



3.1.2. TransHeader/TransFooter section

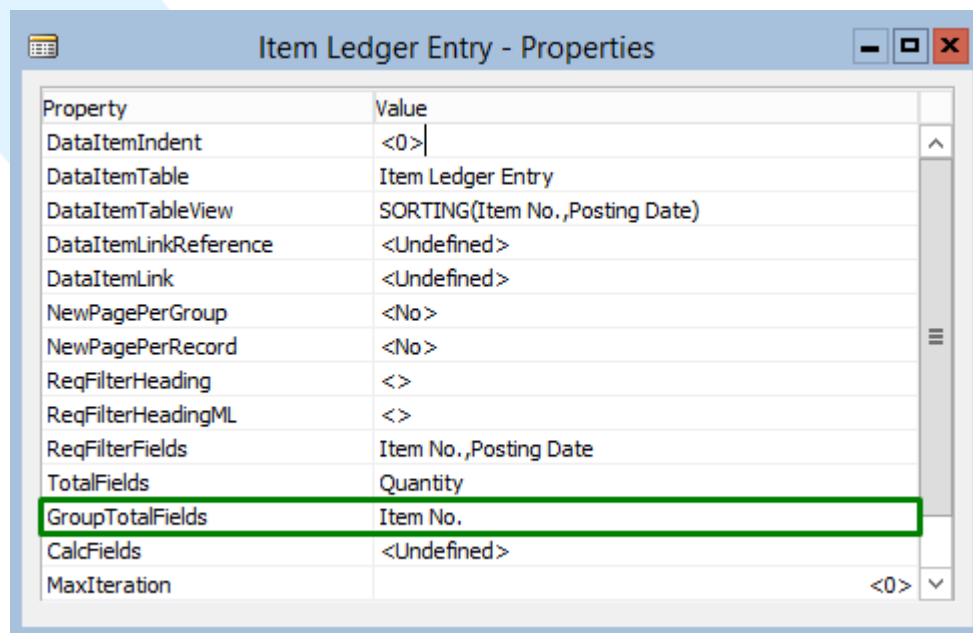
Conversion of TransHeader and TransFooter sections are not supported yet. No rows are created for these sections and no controls converted to RDLC format report layout during conversion.

3.1.3. GroupHeader/GroupFooter Section

Each GroupHeader/GroupFooter type Classic Dynamics NAV report section and its controls are converted to layout tablix row and controls which belongs to groups by fields specified in DataItem GroupTotalFields property.

- Example. Conversion of Classic Dynamics NAV report GroupHeader section.

If some Classic Dynamics NAV report DataItem has a field specified in GroupTotalFields property



and GroupHeader type section created,

Item Ledger Entry, Header (1)

Item Ledger Entry =<FORMAT(TODAY,0,4)>

=<COMPANYNAME> Page =...

=<USERID>

Item Ledger Entry, Header (2)

Item No.	Posting Date	E. T.	Source No.	Document No.	Description	Location Code	Quantity
----------	--------------	-------	------------	--------------	-------------	---------------	----------

Item Ledger Entry, GroupHeader (3)

Item No. =<Item No.>

Item Ledger Entry, Body (4)

Item No.	Posting Date	E. T.	Source No.	Document No.	Description	Location Code	Quantity
----------	--------------	-------	------------	--------------	-------------	---------------	----------

then during conversion there is column for GroupHeader implementation created with "Data Source" as specified in GroupTotalFields property

Exp...	Data Type	Data Source	Name	I...
▶	DataItem	Item Ledger Entry	Item Ledger Entry	^
	Column	'0'	SRB_Item_Ledger_Entry_GroupConstant	≡
	Column	"Entry No."	SRB_Item_Ledger_Entry_Entry_No__	
	Column	"Item No."	SRB_Item_Ledger_Entry_Item_No__	
	Column	"Posting Date"	SRB_Item_Ledger_Entry_Posting_Date__	
	Column	Quantity	SRB_Item_Ledger_Entry_Quantity	▼

and tablix group by field specified in GroupTotalFields property created in Report Layout Designer

Row Groups

- (SRB_Item_Ledger_Entry_GroupConstant) ▼
- (Item_Ledger_Entry_Item_No_) ▼
- (Item_Ledger_Entry_Entry_No_) ▼

with grouping expression

Group Properties ✕

General

Page Breaks

Sorting

Visibility

Filters

Variables

Advanced

Change name and group expressions.

Name:

Group expressions:

Group on:

Tablix row for GroupHeader section created and converted controls placed on it.

@Item_Led	@Item_L	@	@Item_Led	@Item_Led	@Item_Ledger_Entry_DescriptionCa	@Item_Led	@Item_Ledge
@Item_Ledger_Entry	Iti	@Item_Ledger_Entry					
Item_Ledge	Item_Led	Iti	Item_Ledge	Item_Ledge	Item_Ledger_Entry_Description	Item_Ledge	Entry_Quantity

If grouping is more complex and there are a few fields listed in GroupTotalFields property in Classic Dynamics NAV report, then there are more rows in RDLC format report tablix created, one for each value listed in GroupTotalFields property. Also for more complex cases there are some C/AL global variables and functions created to determine row visibility.

3.1.4. Body section

Each body type Classic Dynamics NAV report section is converted to one RDLC format report tablix row. Body type rows are added to the group by DataItemTable primary key. Sorting by the fields specified in DataItemTableView sorting property is applied. If DataItemTableView sorting is undefined, DataItemTable primary key is used for it.

- Example. Conversion of body type section.

If there is some Classic Dynamics NAV report DataItem with DataItemTable and DataItemTableView properties specified,

Property	Value
DataItemIndent	<0>
DataItemTable	Customer
DataItemTableView	SORTING(Name)
DataItemLinkReference	<Undefined>
DataItemLink	<Undefined>
NewPagePerGroup	<No>
NewPagePerRecord	<No>

and body type section created,

Customer, Header (1)		
Customer		=<FORMAT(TODAY,0,4)>
=<COMPANYNAME>		Page =...
		=<USERID>
Customer, Header (2)		
No.	Name	Amount
Customer, Body (3)		
=<No.>	=<Name>	=<Amount>

then during conversion there are columns with “Data Source” of DataItem table primary key and sorting fields created.

Exp...	Data Type	Data Source	Name	I...
	DataItem	Customer	Customer	
	Column	'0'	SRB_Customer_GroupConstant	
	Column	"No."	SRB_Customer__No__	
	Column	Name	SRB_Customer_Name	
	Column	SRB_DataItemExported(SRB_Customer_Exported)	SRB_Customer_Exported	
	Column	SRB_Customer_Header1_Visible	SRB_Customer_Header1_Visible	

Layout group created with grouping by DataItemTable table primary key (in this case it is table's Customer field "No.")

Row Groups

(SRB_Customer_GroupConstant)

(Customer_No__)

Group Properties

General

Page Breaks

Sorting

Visibility

Filters

Variables

Advanced

Change name and group expressions.

Name:

Customer_No__

Group expressions:

Add

Delete

Group on:

SRB_Customer_No__

f

and sorting by field specified in DataItemTableView property applied.

Group Properties

General

Page Breaks

Sorting

Visibility

Filters

Variables

Change sorting options.

Add

Delete

↑

↓

Column

Order

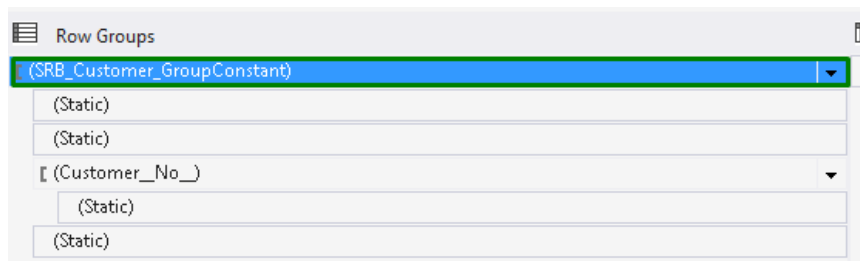
Sort by

SRB_Customer_Name

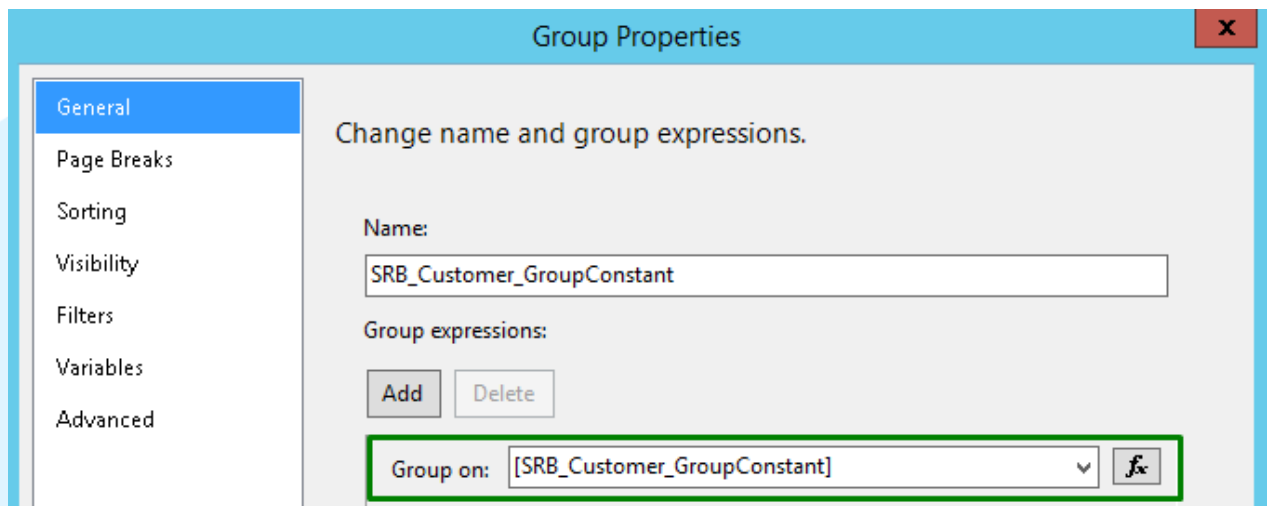
f

A to Z

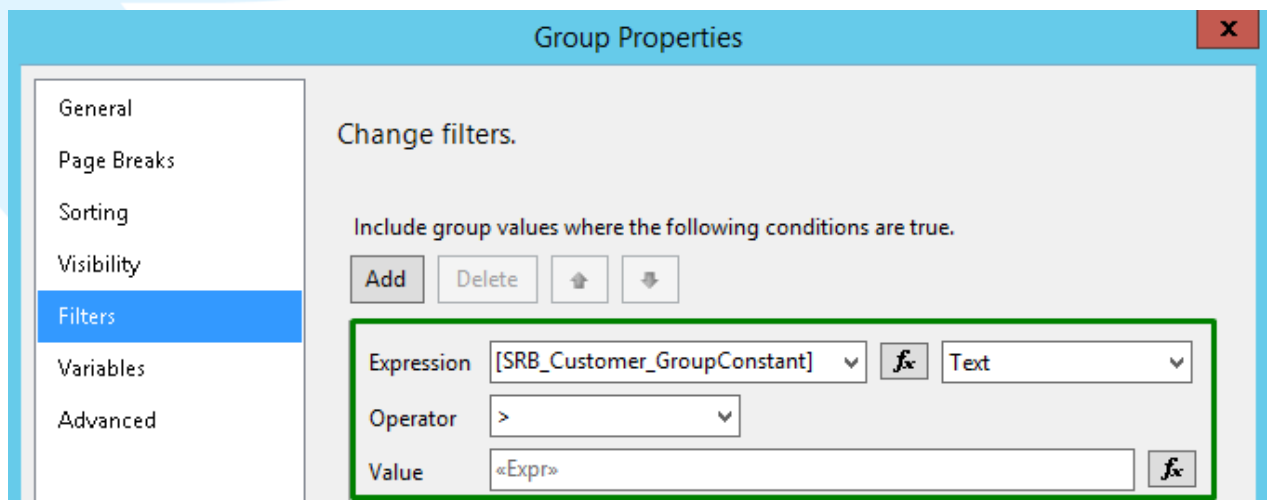
Tablix row for body section created and converted controls placed on it.



Footer group has grouping expression of dataset column with the value of constant '0'



and filter expression which prevents from showing empty values
(Fields!SRB_Customer_GroupConstant.Value >= "").



There is a tablix row created for footer section and controls are placed on it.

[<@CustomerCaption>]		[<ExecutionTime>]	
[<Extra>]		[<Customer>]	
[<@Customer>]	[<@Customer_NameCaption>]	[<@Customer_Amount>]	
[<Customer>]	[<Customer Name>]	[<Customer Amount>]	
[<@TotalCaption>]		[<Customer Amount>]	

- Example 2. Conversion of footer section which PrintOnEveryPage property has YES value.

If Classic Dynamics NAV report footer section's PrintOnEveryPage property has YES value,

Customer, Header (1)			
Customer		=<FORMAT(TODAY,0,4)>	
=<COMPANYNAME>		Page =...	
		=<USERID>	
Customer, Header (2)			
No.	Name	Amount	
Customer, Body (3)			
=<No.>	=<Name>	=<Amount>	
Customer, Footer (4)			
	Total	=<Amount>	

Customer, Footer (4) - Properties	
Property	Value
PrintOnEveryPage	Yes
PlaceInBottom	<No>
SectionWidth	18150

during the conversion there is column for footer group created in Report Dataset Designer

Exp...	Data Type	Data Source	Name	I...
	DataItem	Customer	Customer	^
	Column	'0'	SRB_Customer_GroupConstant	
	Column	"No."	SRB_Customer_No__	
	Column	Name	SRB_Customer_Name	

and footer group created in Report Layout Designer.

Row Groups	
(SRB_Customer_GroupConstant)	▼
(Static)	
(Static)	
[(Customer_No_)]	▼
(Static)	
(Static)	

Footer group has grouping expression of dataset column with the value of constant '0'

Group Properties

Change name and group expressions.

Name: SRB_Customer_GroupConstant

Group expressions:

Add Delete

Group on: [SRB_Customer_GroupConstant] fx

and filter expression which prevents from showing empty values (Fields!SRB_Customer_GroupConstant.Value >= "").

Group Properties

Change filters.

Include group values where the following conditions are true.

Add Delete Up Down

Expression [SRB_Customer_GroupConstant] fx Text

Operator >

Value «Expr» fx

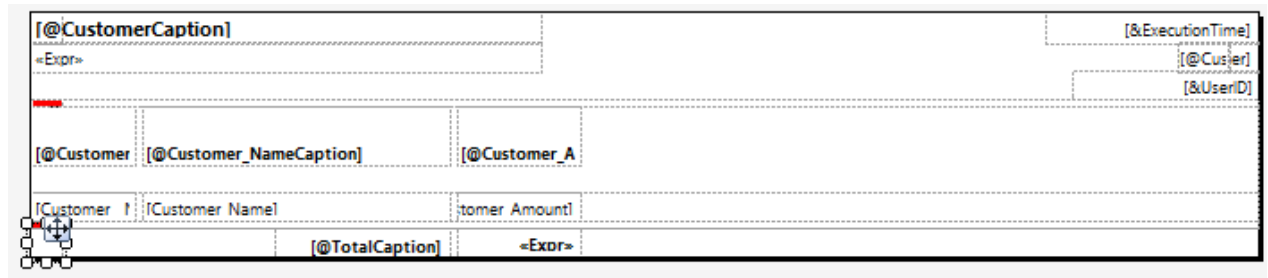
There is a tablix row for footer group created with red FooterData1 text box in it for listing values which will be shown in report footer.

Report Footer Section:

Customer Name	Amount
[Customer Name]	[Customer Amount]
[FooterData1] [FooterData2]	

For red text box with footer values listed there is setting text box FooterData1SetTextBox created in report footer's top left corner. Rectangular with footer section controls is placed in

report footer and controls use GetData function to get values from FooterData1 text box with the help of FooterData1SetTextBox text box. If CurrReport.PAGENO function is used in some text box SourceExpr property in Classic Dynamics NAV report, during the conversion this function is changed to Globals!PageNumber function.



3.2. Section Trigger Conversion

Classic Dynamics NAV report sections have two triggers - OnPreSection() and OnPostSection(). There are two ways of how code from OnPreSection() and OnPostSection() triggers can be transferred to RDLC format report during conversion. Code can be transferred to new C/AL functions or to RDLC format report layout row visibility in some cases if function CurrReport.SHOWOUTPUT used in section trigger code.

OnPreSection() and OnPostSection() trigger code is transferred to new C/AL functions if:

1. Section trigger contains more code than just CurrReport.SHOWOUTPUT function used;
2. More than one CurrReport.SHOWOUTPUT used in one section trigger;
3. CurrReport.SHOWOUTPUT used in both same section triggers (OnPreSection() and OnPostSection());
4. Symbols :: used in CurrReport.SHOWOUTPUT function. This is frequently used with option type variables (for example, CurrReport.SHOWOUTPUT(„Account Type“ = „Account Type“::Posting));
5. If some C/AL function is used in CurrReport.SHOWOUTPUT function (for example, GETFILTERS), except CurrReport.PAGENO;

In other cases OnPreSection() and OnPostSection() triggers code is transferred to RDLC format report layout row visibility.

- Example 1. OnPreSection() trigger code conversion to C/AL function.

If on OnPreSection() trigger there is code,

```
Integer, Body {1} - OnPreSection()
IF PrintToExcel THEN
    CurrReport.SHOWOUTPUT(FALSE)
ELSE
    CurrReport.SHOWOUTPUT("G/L Account"."Account Type" = "G/L Account"."Account Type"::Posting);
Integer, Body {1} - OnPostSection()
```

then during conversion there is C/AL global variable and function created (in this case boolean type variable `SRB_Integer_Body1_Visible` and function `Integer_Body1_OnPreSection()`).

The top screenshot shows the 'Variables' tab in the SAP IDE. It displays a table of global variables:

Name	DataType	Subtype	Length
DirectPostingTxt	Text		30
SRB_G_L_Account_Exported	Boolean		
SRB_G_L_Account_Header1_Visible	Boolean		
SRB_BlankLineCounter_Exported	Boolean		
SRB_Integer_Exported	Boolean		
SRB_Integer_Body1_Visible	Boolean		
SRB_Integer_Body2_Visible	Boolean		
SRB_Integer_Body3_Visible	Boolean		
SRB_PageGroupNo	Integer		

The bottom screenshot shows the 'Functions' tab in the SAP IDE. It displays a list of functions:

Name
SRB_DataItemExported
Integer_Body1_OnPreSection
Integer_Body2_OnPreSection
Integer_Body3_OnPreSection

Code from `OnPreSection()` trigger is transferred to C/AL function and `CurrReport.SHOWOUTPUT` function replaced with boolean variable.

```

Integer_Body1_OnPreSection(Integer : Record Integer)
WITH Integer DO BEGIN
    IF PrintToExcel THEN
        SRB_Integer_Body1_Visible := (FALSE)
    ELSE
        SRB_Integer_Body1_Visible := ("G/L Account"."Account Type" = "G/L Account"."Account Type"::Posting);
    END;
Integer_Body2_OnPreSection(Integer : Record Integer)

```

There is column with boolean variable as „Data Source“ created in Report Dataset Designer

Exp...	Data Type	Data Source	Name	
	DataItem	Integer	Integer	^
	Column	SRB_PageGroupNo	SRB_PageGroupNo	
	Column	Number	SRB_Integer_Number	
	Column	SRB_DataItemExported(SRB_Integer_Exported)	SRB_Integer_Exported	
▶	Column	SRB_Integer_Body1_Visible	SRB_Integer_Body1_Visible	
	Column	"G/L Account". "No."	G_L_Account__No__	
	Column	PADSTR("G/L Account".Indentation * 2)+"G/L ...	PADSTR____G_L_Account__...	▼

and used in Report Layout Designer as row visibility expression.

Set expression for: Hidden

```
=IIF(Max(Fields!SRB_Integer_Body1_Visible.Value),FALSE,TRUE)
```

- Example 2. OnPreSection() trigger code conversion to layout row visibility expression.

If on OnPreSection() trigger there is code,

```
Customer, Header (2) - OnPreSection()  
CurrReport.SHOWOUTPUT(PrintAmountsInLCY);
```

```
Customer, Header (2) - OnPostSection()
```

then during the conversion there is column with function's CurrReport.SHOWOUTPUT parameter as „Data Source“ created in Report Dataset Designer

Exp...	Data Type	Data Source	Name	I...
	DataItem	Customer	Customer	^
	Column	PrintOnlyOnePerPage	Customer_PrintOnlyOnePerPage	
	Column	'0'	SRB_Customer_GroupConstant	
	Column	"No."	SRB_Customer__No__	
▶	Column	PrintAmountsInLCY	Customer_PrintAmountsInLCY	
	Column	SRB_DataItemExported(SRB_Customer_Exported)	SRB_Customer_Exported	▼

and used as layout row visibility expression.

Set expression for: Hidden

```
=IIf((Fields!Customer_PrintAmountsInLCY.Value) = TRUE,FALSE,TRUE)
```

3.3. Section Control Conversion

This section explains the main rules of label, text box, picture box, image and shape type section control conversion from Classic Dynamics NAV to RDLC format report.

3.3.1. Label Conversion

Label type section controls are converted to RDLC format layout text boxes.

The main label conversion rules:

1. If function CurrReport.LANGUAGE is used somewhere in Classic Dynamics NAV report code, there is new C/AL text constant created for label conversion and values from Caption and CaptionML label properties are transferred to it, otherwise new label in Report Label Designer is created and Caption and CaptionML property values are transferred to it.
2. If property CaptionClass has a value, new Report Dataset Designer column is created with "Data Source"= CAPTIONCLASSTRANSLATE(CaptionClass property value) and used in layout.
3. Helvetica FontName is converted to Arial FontFamily by default.
4. If label DataSetFieldName property is specified, all new dataset columns, text boxes, labels and C/AL text constants created for label conversion are named by DataSetFieldName property value. If DataSetFieldName property is not specified, names are generated by parent control SourceExpr property value. If parent control is undefined, then names are generated by Caption property value.

Label control properties can be transferred to proper RDLC format report layout or dataset properties or not used after conversion if not relevant.

List of label properties which are transferred to RDLC format report layout properties:

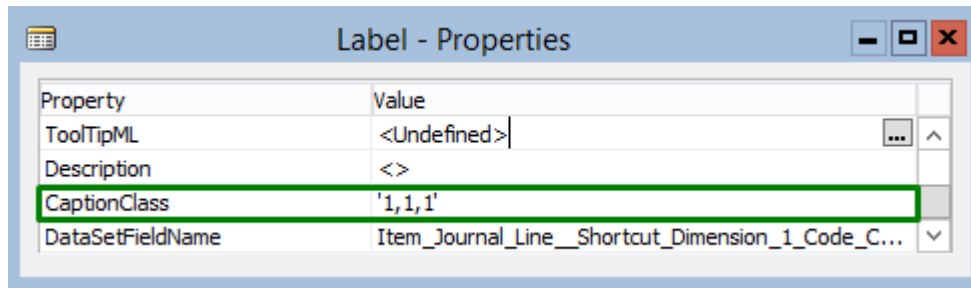
Classic Label Property	RDLC Text Box Property
XPos	Location -> Left
YPos	Location -> Top
Width	Size -> Width
Height	Size -> Height
Visible	Visibility -> Hidden
HorzAlign	Alignment -> TextAlign
VertAlign	Alignment -> VerticalAlign
FontName	Font -> FontFamily
FontSize	Font -> FontSize
FontBold	Font -> FontWeight -> Bold
FontItalic	Font -> FontStyle -> Italic
FontStrikethru	Font -> TextDecoration -> LineThrough
FontUnderline	Font -> TextDecoration -> Underline
LeaderDots	Border -> BorderStyle -> Bottom -> Dotted

List of label properties which are transferred to Report Dataset Designer ("Data Source" column):

Classic Label Property	RDLC Field "Data Source"
CaptionClass	CAPTIONCLASSTRANSLATE(CaptionClass)

- Example. Conversion of label control with CaptionClass property specified.

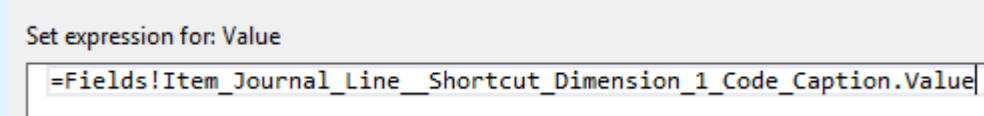
If label CaptionClass property is specified,



during conversion there is column created with CAPTIONCLASSTRANSLATE(CaptionClass) as "Data Source"

E...	Data Type	Data Source	Name	I...
	Column	ShowNote	PageLoop_ShowNote	
▶	Column	CAPTIONCLASSTRANSLATE('1,1,1')	Item_Journal_Line__Shortcut_Dimension_1_Code_Caption	
	Column	CAPTIONCLASSTRANSLATE('1,1,2')	Item_Journal_Line__Shortcut_Dimension_2_Code_Caption	
☐	DataItem	Item Journal Batch	Item Journal Batch	
	Column	"Journal Template Name"	SRB_Item_Journal_Batch__Journal_Template_Name_	
	Column	Name	SRB_Item_Journal_Batch_Name	

and used in layout text box as value expression.



List of label properties which are transferred to Report Label Designer label or C/AL text constant properties:

Classic Label Property	Label Or Text Constant Property
Caption	Caption
CaptionML	CaptionML

List of label properties which are not transferred after conversion:

Classic Label Property
ID
Name
HorzGlue
VertGlue
ParentControl
InFrame
InPage

InColumnHeading
ForeColor
BackColor
BackTransparent
Border
BorderColor
BorderStyle
BorderWidth
MultiLine
PadChar
ToolTip
ToolTipML
Description

3.3.2. Text Box Conversion

Text box type controls are converted to RDLC format report layout text boxes.

The main text box conversion rules:

1. If property CaptionClass has a value, new Report Dataset Designer column is created with "Data Source" CAPTIONCLASSTRANSLATE(CaptionClass property value) and used in child control text box expression.
2. Helvetica FontName is converted to Arial FontFamily by default.
3. If text box DataSetFieldName property is specified, all new dataset columns and text boxes created for text box conversion are named by DataSetFieldName property value. If DataSetFieldName property is not specified, names are generated by SourceExpr property value.

Text box control properties can be transferred to proper RDLC format report dataset field or layout text box properties and expressions or not used after conversion if not relevant.

List of text box properties which are transferred to RDLC format report layout properties:

Classic Text Box Property	RDLC Text Box Property
XPos	Location -> Left
YPos	Location -> Top
Width	Size -> Width
Height	Size -> Height
Visible	Visibility -> Hidden
HorzAlign	Alignment -> TextAlign
VertAlign	Alignment -> VerticalAlign
FontName	Font -> FontFamily
FontSize	Font -> FontSize

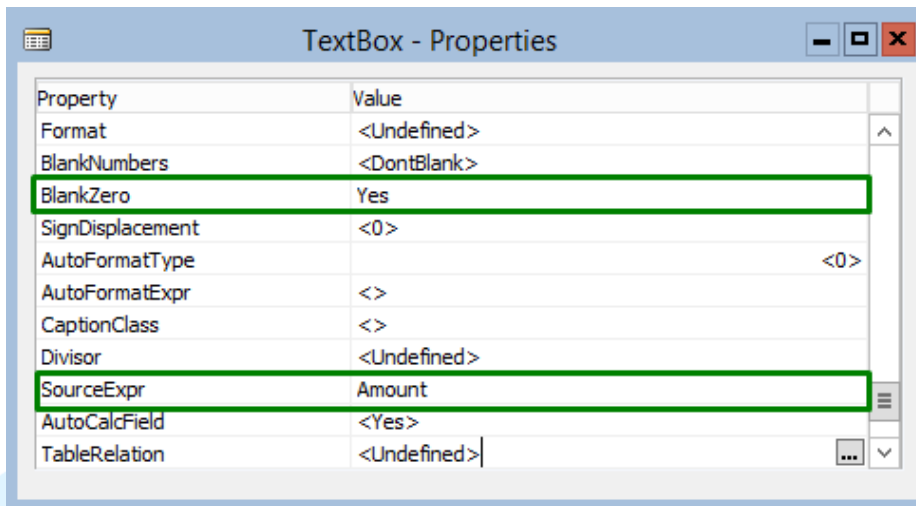
FontBold	Font -> FontWeight -> Bold
FontItalic	Font -> FontStyle -> Italic
FontStrikethru	Font -> TextDecoration -> LineThrough
FontUnderline	Font -> TextDecoration -> Underline

List of text box properties which are transferred to RDLC format report layout text box expression:

Classic Text Box Property	RDLC Text Box Expression
BlankNumbers	<p>Depending on the value of BlankNumbers property, there can be different layout functions used in text box expression</p> <ul style="list-style-type: none"> BlankNumbers = BlankNeg is converted to expression <code>=Code.BlankNeg(Value)</code> BlankNumbers = BlankNegAndZero is converted to expression <code>=Code.BlankNegAndZero(Value)</code> BlankNumbers = BlankZero is converted to expression <code>=Code.BlankZero(Value)</code> BlankNumbers = BlankZeroAndPos is converted to expression <code>=Code.BlankZeroAndPos(Value)</code> BlankNumbers = BlankPos is converted to expression <code>=Code.BlankPos(Value)</code>
BlankZero	<p>If BlankZero = Yes and there is decimal value in text box SourceExpr specified, after conversion this value is used with layout function BlankZero: <code>=Code.BlankZero(Value)</code></p> <p>If BlankZero = Yes and there is boolean value in text box SourceExpr specified, after conversion this value is used with expression: <code>=IIF(Value = "Yes",Value,Nothing)</code></p>

- Example 1. Conversion of text box with decimal value in SourceExpr property specified and BlankZero=Yes.

If there is text box with BlankZero=Yes and decimal value in SourceExpr property specified,



after conversion there is Report Dataset Designer column created with SourceExpr value in "Data Source"

Exp...	Data Type	Data Source	Name	I...
▶	DataItem	Customer	Customer	↑
	Column	'0'	SRB_Customer_GroupConstant	
	Column	"No."	SRB_Customer_No__	
	Column	SRB_DataItemExported(SRB_C...	SRB_Customer_Exported	
	Column	Amount	Customer_Amount	
	Column	FORMAT("Print Statements")	Customer__Print_Statements_	

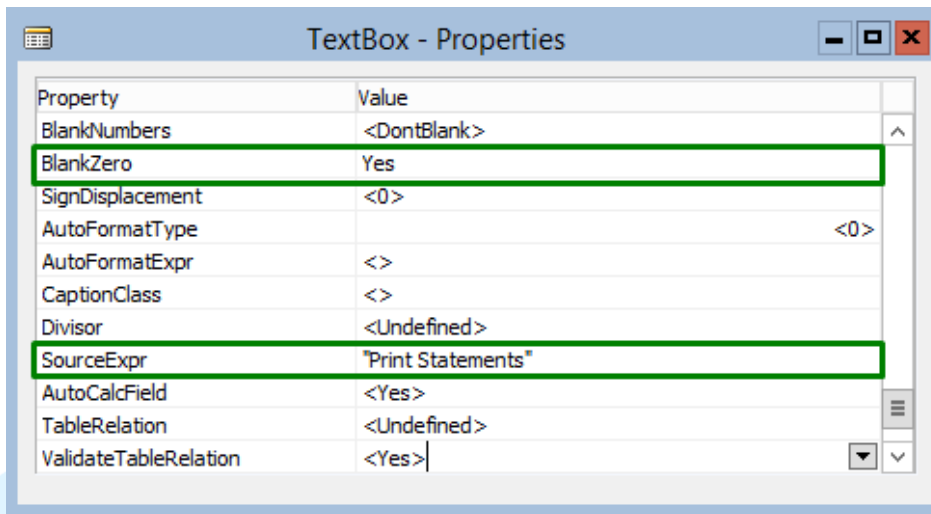
and layout text box created with expression:

Set expression for: Value

`=Code.BlankZero(Fields!Customer_Amount.Value)`

- Example 2. Conversion of text box with boolean value in SourceExpr property specified and BlankZero=Yes.

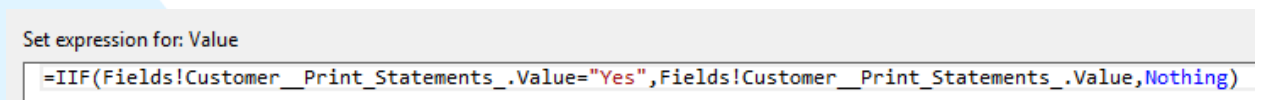
If there is text box with BlankZero=Yes and boolean value in SourceExpr property specified,



after conversion there is Report Dataset Designer column created with SourceExpr value in "Data Source"

Exp...	Data Type	Data Source	Name	I...
▶	DataItem	Customer	Customer	^
	Column	'0'	SRB_Customer_GroupConstant	
	Column	"No."	SRB_Customer_No__	
	Column	SRB_DataItemExported(SRB_C...	SRB_Customer_Exported	
	Column	Amount	Customer Amount	
	Column	FORMAT("Print Statements")	Customer__Print_Statements_	

and layout text box created with expression:



List of text box properties which are transferred to Report Dataset Designer column properties:

Classic Text Box Property	Dataset Column Property
OptionString	OptionString
OptionCaption	OptionCaption
OptionCaptionML	OptionCaptionML
DecimalPlaces	DecimalPlaces
AutoFormatType	AutoFormatType
AutoFormatExpr	AutoFormatExpr
SourceExpr	SourceExpr
AutoCalcField	AutoCalcField
Format	SourceExpr = FORMAT(SourceExpr,0,'FromatPropertyValue')

- Example 3. Conversion of text box with Format property specified.

If there is text box with Format property specified,

Property	Value
ClearOnLookup	<Yes>
Format	<Day,2><Filler Character, >. <Month Text,3> <Year4>
BlankNumbers	<DontBlank>
BlankZero	<No>
SignDisplacement	<0>
AutoFormatType	<0>
AutoFormatExpr	<>
CaptionClass	<>
Divisor	<Undefined>
SourceExpr	DateVar
AutoCalcField	<Yes>

during conversion there is column created with Format and SourceExpr properties values used in FORMAT function

E...	Data Type	Data Source	Name	I
	DataItem	Customer	Customer	
	Column	'0'	SRB_Customer_GroupConstant	
	Column	"No."	SRB_Customer_No__	
▶	Column	FORMAT(DateVar,0,'<Day,2><Filler Character, >. <Month Text,3> <Year4>')	Customer_DateVar	
	Column	SRB_DataItemExported(SRB_Customer_Exported)	SRB_Customer_Exported	

List of text box properties which are not transferred to RDLC format report:

Classic Text Box Property
ID
Name
HorzGlue
VertGlue
Enabled
Editable
Focusable
ParentControl
InFrame
InPage
InColumn
InMatrix
InMatrixHeading

InColumnHeading
ForeColor
BackColor
BackTransparent
Border
BorderColor
BorderStyle
BorderWidth
MultiLine
PadChar
LeaderDots
MaxLength
PasswordText
AutoEnter
ToolTip
ToolTipML
Lookup
DrillDown
AssistEdit
DropDown
PermanentAssist
Description
Title
MinValue
MaxValue
NotBlank
Numeric
CharAllowed
DateFormula
ClosingDates
ValuesAllowed
NextControl
ClearOnLookup
SignDisplacement
Divisor
TableRelation
ValidateTableRelation
LookupFormID

DrillDownFormID

3.3.3. Picture Box Conversion

Picture boxes are converted to RDLC layout images.

The main picture box conversion rules:

1. For picture box conversion there is RDLC layout image created with Sizing=FitProportional
2. Every RDLC layout image is added to rectangular in order to avoid location shifts if visibility expression used.
3. If picture box DataSetFieldName property is specified, image created for picture box conversion is named by DataSetFieldName property value. If DataSetFieldName property is not specified, name is generated by SourceExpr property value. Image container rectangular name is generated from image name and word "Wrapper" combination.

Picture box control properties can be transferred to proper RDLC format report layout image, rectangular or dataset properties or not used after conversion if not relevant.

List of picture box properties which are transferred to RDLC format report image's rectangular properties:

Classic Picture Box Property	RDLC Rectangular Property
XPos	Location -> Left
YPos	Location -> Top
Width	Size -> Width
Height	Size -> Height

List of picture box properties which are transferred to RDLC format report layout image properties:

Classic Picture Box Property	RDLC Image Property
XPos	Location -> Left
YPos	Location -> Top
Width	Size -> Width
Height	Size -> Height
Visible	Visibility -> Hidden
HorzAlign	Alignment -> TextAlign
VertAlign	Alignment -> VerticalAlign
DataSetFieldName	Name (if not empty)

List of picture box properties which are transferred to Report Dataset Designer column properties:

Classic Picture Box Property	Dataset Column Property
SourceExpr	SourceExpr

AutoCalcField	AutoCalcField
---------------	---------------

List of picture box properties which are not transferred to RDLC format report:

Classic Picture Box Property
ID
Name
HorzGlue
VertGlue
Enabled
Focusable
ParentControl
InFrame
InPage
InColumn
InMatrix
InMatrixHeading
Caption
CaptionML
BackColor
BackTransparent
Border
BorderColor
BorderStyle
BorderWidth
BitmapList
ToolTip
ToolTipML
Description
NextControl
CaptionClass

- Example. Conversion of picture box control.

If there is picture box control used in Classic Dynamics NAV report

> > PageLoop, Header (1)

		=<STRSUBSTNO(DocumentCaption...	
		Page =<CurrReport.PAGENO>	
		Invoice No.	=<"Sales In...
		Posting Date	=<FORMAT(...
		Due Date	=<FORMAT(...

> > > Sales Invoice Line, Header (1)

No.	Description	Quantity	Unit of Measure	Unit Price	Amount
-----	-------------	----------	-----------------	------------	--------

> > > Sales Invoice Line, Body (2)

with source expression,

PictureBox - Properties

Property	Value
SourceExpr	CompanyInfo.Picture
AutoCalcField	<Yes>
DataSetFieldName	CompanyInfo_Picture

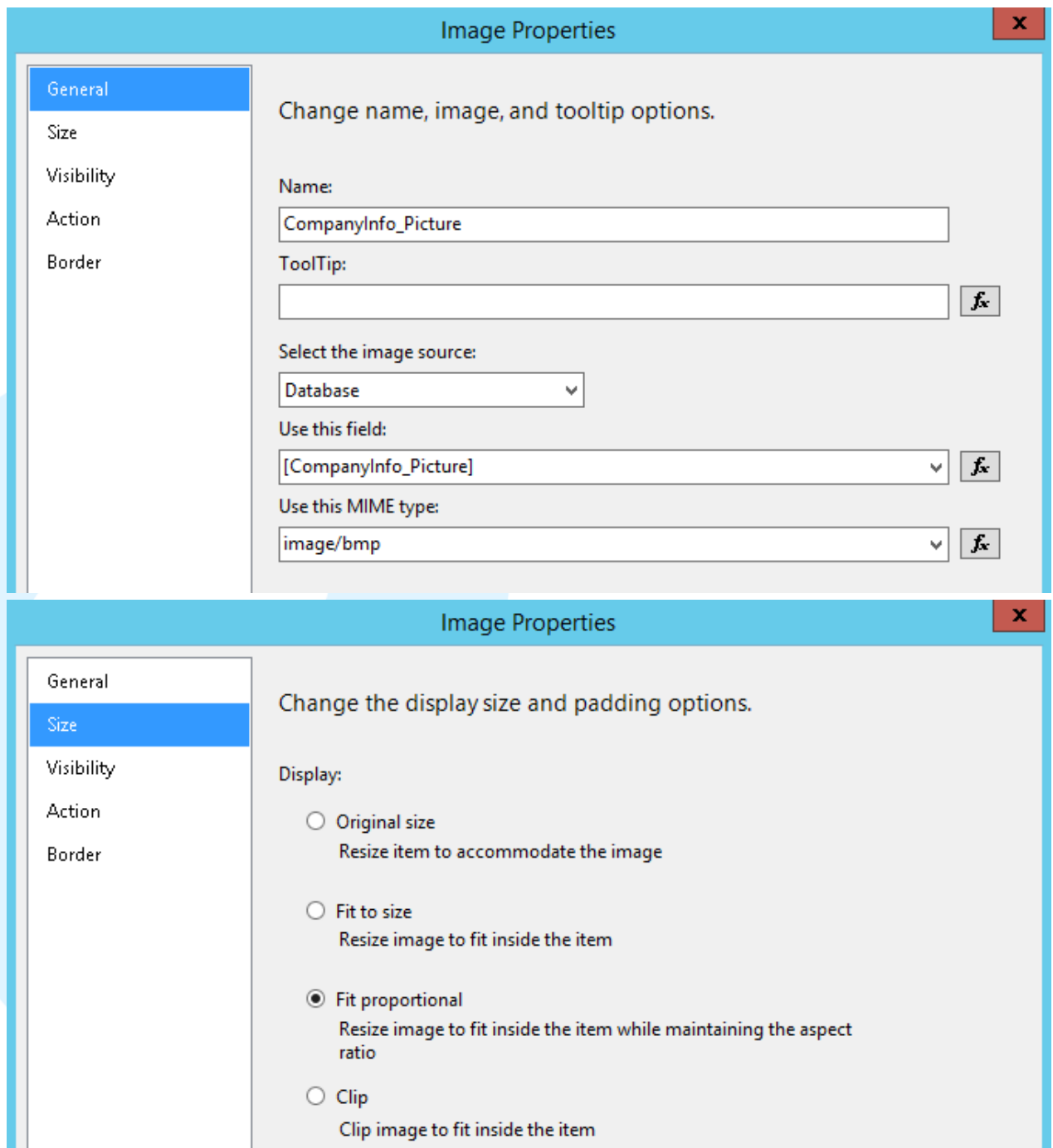
during conversion there is a column for SourceExpr value created,

Exp...	Data Type	Data Source	Name	I
	DataItem	Integer	PageLoop	
	Column	'0'	SRB_PageLoop_GroupConstant	
	Column	Number	SRB_PageLoop_Number	
	Column	SRB_DataItemExported(SRB_PageLoop_Exported)	SRB_PageLoop_Exported	
	Column	CompanyInfo.Picture	CompanyInfo_Picture	
	Column	SRB_PageLoop_Header1_Visible	SRB_PageLoop_Header1_Visible	

layout image and rectangular as image container created

		[STRSUBSTNO DocumentCaption Cc	
		[PageCapti][STRSUBSTNO Text005 FORI	
		Invoice No C	[Sales Invoice
		[Sales Invoice Head	[Sales Invoice
		[Sales Invoice Head	[Sales Invoice
[Sales_Invoice	[Sales_Invoice_Line_Description_Control	[Sales_Invoice_	[Unit_PriceCaption] ountCaption]
		[Sales_Invoice_Line_Description]	

with properties:



3.3.4. Image Conversion

Image type control is converted to RDLC format report layout image.

The main image conversion rules:

1. Image is converted to RDLC layout image with Source = External and Value = Bitmap properties values. After conversion user can change Source property value from External to Embedded and import picture file form the path specified in Value property.
2. Converted image has Sizing property set to FitProportional value.

3. Every RDLC layout image is added to rectangular in order to avoid location shifts if visibility expression used.
4. If image DataSetFieldName property is specified, converted image is named by DataSetFieldName property value. If DataSetFieldName property is not specified, name is generated by combination of ID property value and word "Image" (for example Image1000000033). Image container rectangular name is generated by combination of image name and word "Wrapper" (for example Image1000000033_Wrapper)

Image control properties can be transferred to proper RDLC format report layout image and rectangular properties or not used after conversion if not relevant.

List of image properties which are transferred to RDLC format report layout image's rectangular properties:

Classic Image Property	RDLC Rectangular Property
XPos	Location -> Left
YPos	Location -> Top
Width	Size -> Width
Height	Size -> Height
DataSetFieldName	Name (if DataSetFieldName is not empty)
ID	Name = "Image"+ ID + "Wrapper" (if DataSetFieldName empty)

List of image properties which are transferred to RDLC format report layout image properties:

Classic Image Property	RDLC Image Property
XPos	Location -> Left
YPos	Location -> Top
Width	Size -> Width
Height	Size -> Height
Visible	Visibility -> Hidden
Bitmap	General -> Value
DataSetFieldName	Name (if DataSetFieldName is not empty)
ID	Name = "Image"+ ID (if DataSetFieldName empty)

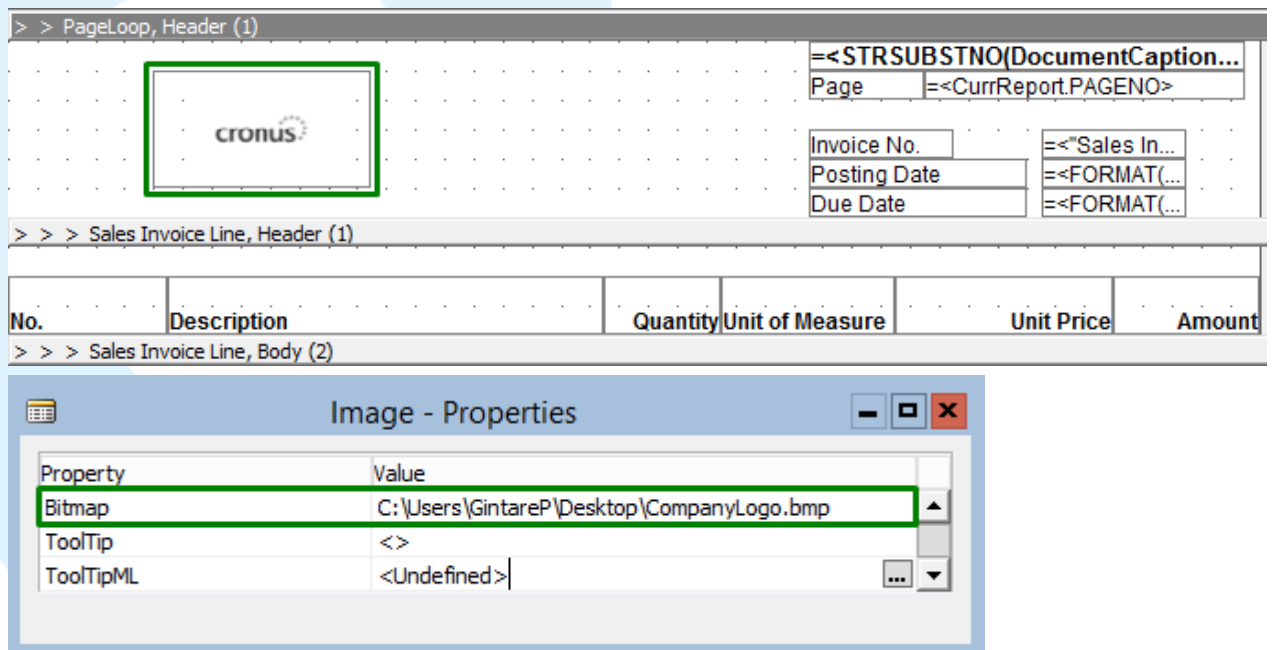
List of image properties which are not transferred to RDLC format report:

Classic Image Property
Name
HorzGlue
VertGlue
ParentControl
InFrame

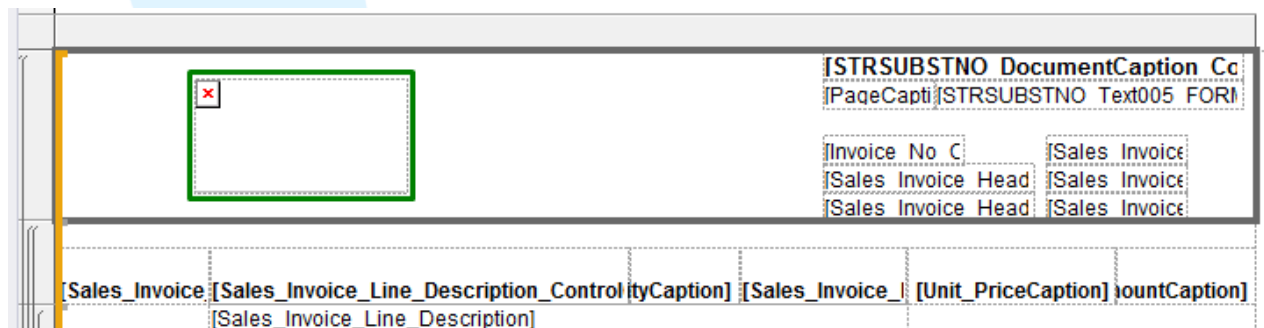
InPage
InMatrixHeading
BackColor
BackTransparent
Border
BorderColor
BorderStyle
BorderWidth
ToolTip
ToolTipML
Description

- Example. Conversion of image control.

If there is image control with Bitmap property specified used in Classic Dynamics NAV report,



then during conversion there is layout image and rectangular as image container created



with properties:

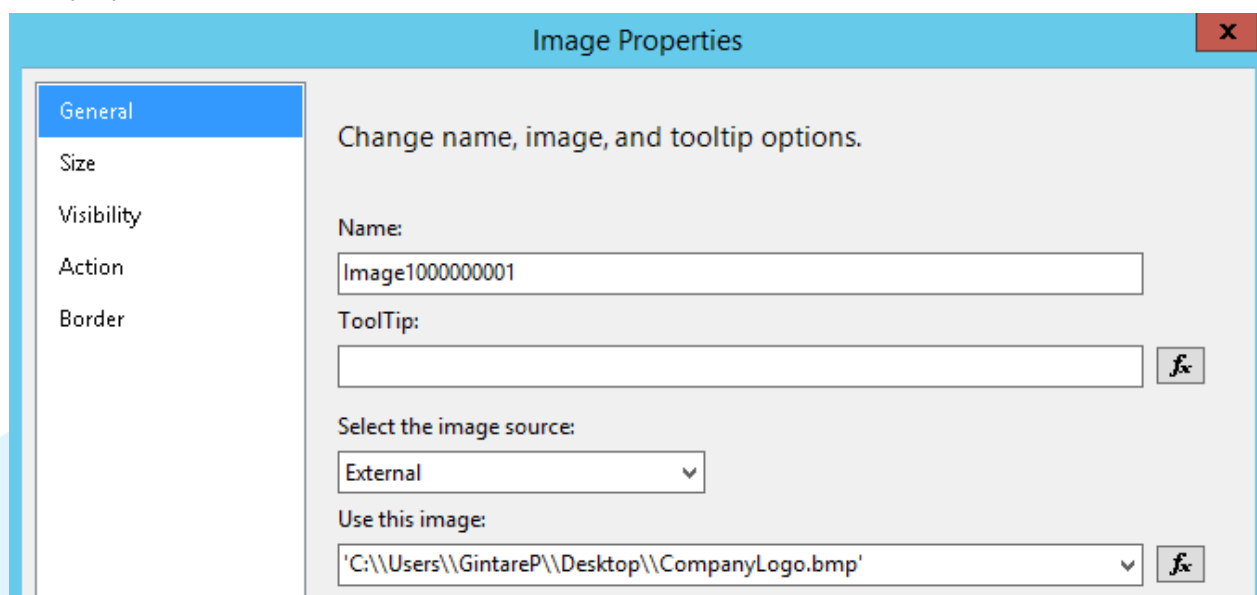


Image source can be changed from External to Embedded and proper image file imported.

3.3.5. Shape Conversion

Shape type control depending on ShapeStyle property value can be converted to RDLC layout rectangular, line or even image.

The main shape control conversion rules:

1. Shape type control is not converted if Border=No.
2. Converted shapes are named by the type of created RDLC format report control. If there is a shape with ShapeStyle = HorzLine, after conversion it has a name of word "Line" and some random number from seven digits combination (for example Line9516006)
3. Shape type control is not converted if ShapeStyle = Triangle.

There are different rules of shape control conversion depending on ShapeStyle property:

- Rectangular and RoundedRectangular

If Rectangular or RoundedRectangular value is specified in ShapeStyle property, this control is converted to RDLC format report layout rectangular.

List of shape properties which are transferred to RDLC format report layout rectangular properties:

Classic Rectangular or RoundedRectangular Property	RDLC Rectangular Property
XPos	Location -> Left
YPos	Location -> Top
Width	Size -> Width

Height	Size -> Height
Visible	Visibility -> Hidden
BorderWidth	Border -> BorderWidth

- Oval

If Oval value is specified in ShapeStyle property, this control is converted to image control with embedded image source of oval figure with width specified in BorderWidth property. Also image control is added to rectangular control which prevents from location shifts if some visibility expression used.

List of shape properties which are transferred to RDLC format report layout image's rectangular properties:

Classic Oval Property	RDLC Rectangular Property
XPos	Location -> Left
YPos	Location -> Top
Width	Size -> Width
Height	Size -> Height

List of shape properties which are transferred to RDLC format report layout image properties:

Classic Oval Property	RDLC Image Property
Width	Size -> Width
Height	Size -> Height
Visible	Visibility -> Hidden

- NW-Line

Shape with NW-Line value specified in ShapeStyle property is converted to RDLC format report layout line control.

List of shape properties which are transferred to RDLC format report layout line properties:

Classic NW-Line Property	RDLC Line Property
XPos	Location -> Left
YPos	Location -> Top
Width	EndPoint -> Horizontal = XPos + Width
Height	EndPoint -> Vertical = YPos + Height
Visible	Visibility -> Hidden
BorderWidth	Style -> LineWidth

- NE-Line

Shape with NE-Line value specified in ShapeStyle property is converted to RDLC format report layout line control.

List of shape properties which are transferred to RDLC format report layout line properties:

Classic NW-Line Property	RDLC Line Property
XPos	Location -> Left
YPos	EndPoint -> Vertical
Width	EndPoint -> Horizontal = XPos + Width
Height	Location -> Top = YPos + Height
Visible	Visibility -> Hidden
BorderWidth	Style -> LineWidth

- HorzLine

Shape with HorzLine value specified in ShapeStyle property is converted to RDLC format report layout line control.

List of shape properties which are transferred to RDLC format report layout line properties:

Classic NW-Line Property	RDLC Line Property
XPos	Location -> Left
YPos	Location -> Top = YPos + Height/2
Width	EndPoint -> Horizontal = XPos + Width
Height	EndPoint -> Vertical = YPos + Height/2
Visible	Visibility -> Hidden
BorderWidth	Style -> LineWidth

- VertLine

Shape with VertLine value specified in ShapeStyle property is converted to RDLC format report layout line control.

List of shape properties which are transferred to RDLC format report layout line properties:

Classic NW-Line Property	RDLC Line Property
XPos	Location -> Left = XPos + Width/2
YPos	Location -> Top
Width	EndPoint -> Horizontal = XPos + Width/2
Height	EndPoint -> Vertical = YPos + Height
Visible	Visibility -> Hidden
BorderWidth	Style -> LineWidth

Shape control properties are not transferred to RDLC format report if not relevant.

List of shape control properties which are not transferred to RDLC format report:

Classic Shape Property
ID
Name
HorzGlue
VertGlue
ParentControl
InFrame
InPage
InColumnHeading
BackColor
BackTransparent
BorderColor
ToolTip
ToolTipML
Description

4. Request Form Conversion

This section explains the main rules of Request Options Form Designer properties, triggers (C/AL code, local variables) and controls conversion to Request Options Page Designer elements.

4.1. Request Form Property Conversion

If the same property exists in both versions, the value of Classic Dynamics NAV report request form property is transferred to the same RDLC report request page property with no changes.

List of Classic Dynamics NAV report properties which are transferred to RDLC format report properties:

Request Form Property	Request Page Property
ID	ID
Name	Name
Caption	Caption
CaptionML	CaptionML
DataCaptionExpr	DataCaptionExpr
DataCaptionFields	DataCaptionFields
Editable	Editable
InsertAllowed	InsertAllowed
ModifyAllowed	ModifyAllowed
DeleteAllowed	DeleteAllowed
MultipleNewLines	MultipleNewLines
Description	Description
SourceTable	SourceTable
SourceTableView	SourceTableView
SourceTableTemporary	SourceTableTemporary
SaveValues	SaveValues
AutoSplitKey	AutoSplitKey
PopulateAllFields	PopulateAllFields
Permissions	Permissions
LinksAllowed	LinksAllowed

4.2. Request Form Trigger Conversion

Code and C/AL local variables from Classic Dynamics NAV report request form trigger are transferred to proper request page trigger. If proper request page trigger does not exist, code and variables are transferred to new C/AL function.

List of Classic Dynamics NAV report triggers which code and variables are transferred to RDLC report triggers:

Request Form Trigger	Request Page Trigger
OnInit()	OnInit()
OnOpenForm()	OnOpenPage()
OnCloseForm()	OnClosePage()
OnQueryCloseForm()	OnQueryClosePage()
OnActivateForm()	OnOpenPage()
OnFindRecord()	OnFindRecord()
OnNextRecord()	OnNextRecord()
OnAfterGetRecord()	OnAfterGetRecord()
OnAfterGetCurrRecord()	OnAfterGetCurrRecord()
OnNewRecord()	OnNewRecord()
OnInsertRecord()	OnInsertRecord()
OnModifyRecord()	OnModifyRecord()
OnDeleteRecord()	OnDeleteRecord()

List of Classic Dynamics NAV report triggers which code and variables are transferred to RDLC report C/AL functions:

Request Form Trigger	Function Name
OnDeactivateForm()	SRB_OnDeactivateForm
OnBeforePutRecord()	SRB_OnBeforePutRecord
OnTimer()	SRB_OnTimer
OnCreateHyperlink()	SRB_OnCreateHyperlink
OnHyperlink()	SRB_OnHyperlink

4.3. Request Form Control Conversion

Request form controls are converted to request page fields or groups. Label, text box, check box, option button, command button type controls are converted to request page fields, frame type controls are converted to request page groups. Menu button, image, picture box, shape, indicator, tab control, subform, table box and matrix box type controls are not converted to request page elements.

4.3.1. Property Conversion

- Label control property conversion

The main label type control conversion rules:

1. If ParentControl property not defined, then label type control is converted to request page field. Label Caption and CaptionML properties values are transferred to new C/AL text

constant Caption and CaptionML properties. C/AL text constant is used in request page field CaptionClass property.

2. If ParentControl property defined and parent control has a value in CaptionClass property, then label property values are not transferred.
3. If ParentControl property defined and parent control Caption property has a value, but there is no value in CaptionClass property, then label Caption and CaptionML properties values are transferred to new C/AL text constant which is used in converted parent control field CaptionClass property.
4. If ParentControl property defined and parent control Caption property has no value, then label Caption and CaptionML properties are transferred to parent control field Caption and CaptionML properties.

List of label control properties which are transferred to RDLC format report field properties:

Request Form Label Property	Request Page Field Property
ID	ID
Name	Name
XPos	-
YPos	-
Width	-
Height	-
HorzGlue	-
VertGlue	-
Visible	Visible
ParentControl	-
InFrame	-
InPage	-
InColumnHeading	-
Caption	Caption or C/AL text constant Caption property
CaptionML	CaptionML or C/AL text constant CaptionML property
HorzAlign	-
VertAlign	-
ForeColor	-
BackColor	-
BackTransparent	-
Border	-
BorderColor	-
BorderStyle	-
BorderWidth	-
FontName	-

FontSize	-
FontBold	Style=Strong
FontItalic	-
FontStrikethru	-
FontUnderline	-
MultiLine	MultiLine
PadChar	-
LeaderDots	-
ToolTip	ToolTip
ToolTipML	ToolTipML
Description	Description
CaptionClass	CaptionClass
DataSetFieldName	-

- Text box control property conversion

Request form text boxes are converted to request page fields.

List of text box control properties which are transferred to RDLC format report field properties:

Request Form Text Box Property	Request Page Field Property
ID	ID
Name	Name
XPos	-
YPos	-
Width	-
Height	-
HorzGlue	-
VertGlue	-
Visible	Visible
Enabled	Enabled
Editable	Editable
Focusable	-
ParentControl	-
InFrame	-
InPage	-
InColumn	-
InMatrix	-
InMatrixHeading	-
Caption	Caption

CaptionML	CaptionML
HorzAlign	-
VertAlign	-
ForeColor	-
BackColor	-
BackTransparent	-
Border	-
BorderColor	-
BorderStyle	-
BorderWidth	-
FontName	-
FontSize	-
FontBold	Style=Strong
FontItalic	-
FontStrikethru	-
FontUnderline	-
MultiLine	MultiLine
PadChar	-
LeaderDots	-
MaxLength	-
PasswordText	-
AutoEnter	-
ToolTip	ToolTip
ToolTipML	ToolTipML
Lookup	Lookup
DrillDown	DrillDown
AssistEdit	AssistEdit
DropDown	-
PermanentAssist	-
Description	Description
OptionString	-
OptionCaption	OptionCaption
OptionCaptionML	OptionCaptionML
DecimalPlaces	DecimalPlaces
Title	Title
MinValue	MinValue
MaxValue	MaxValue
NotBlank	NotBlank

Numeric	Numeric
CharAllowed	CharAllowed
DateFormula	DateFormula
ClosingDates	ClosingDates
ValuesAllowed	ValuesAllowed
NextControl	-
ClearOnLookup	-
Format	-
BlankNumbers	BlankNumbers
BlankZero	BlankZero
SignDisplacement	-
AutoFormatType	AutoFormatType
AutoFormatExpr	AutoFormatExpr
CaptionClass	CaptionClass
Divisor	-
SourceExpr	SourceExpr
AutoCalcField	-
TableRelation	TableRelation
ValidateTableRelation	-
LookupFormID	LookupPageID
DrillDownFormID	DrillDownPageID
DataSetFieldName	-

- Check box control property conversion

Request form check boxes are converted to request page fields.

List of check box control properties which are transferred to RDLC format report field properties:

Request Form Check Box Property	Request Page Field Property
ID	ID
Name	Name
XPos	-
YPos	-
Width	-
Height	-
HorzGlue	-
VertGlue	-
Visible	Visible

Enabled	Enabled
Editable	Editable
Focusable	-
FocusOnClick	-
ParentControl	-
InFrame	-
InPage	-
InColumn	-
InMatrix	-
InMatrixHeading	-
Caption	Caption
CaptionML	CaptionML
ShowCaption	-
HorzAlign	-
VertAlign	-
ForeColor	-
BackColor	-
BackTransparent	-
Border	-
FontName	-
FontSize	-
FontBold	Style=Strong
FontItalic	-
FontStrikethru	-
FontUnderline	-
BitmapPos	-
PushAction	-
Bitmap	-
ToolTip	ToolTip
ToolTipML	ToolTipML
Description	Description
MinValue	MinValue
MaxValue	MaxValue
ValuesAllowed	ValuesAllowed
NextControl	-
CaptionClass	CaptionClass
SourceExpr	SourceExpr
AutoCalcField	-

UpdateOnAction	-
RunObject	-
RunFormView	-
RunFormLink	-
RunFormLinkType	-
RunFormOnRec	-
RunCommand	-

- Option button control property conversion

Request form option buttons are converted to request page fields.

List of option button control properties which are transferred to RDLC format report field properties:

Request Form Option Button Property	Request Page Field Property
ID	ID
Name	Name
XPos	-
YPos	-
Width	-
Height	-
HorzGlue	-
VertGlue	-
Visible	Visible
Enabled	Enabled
Editable	Editable
Focusable	-
FocusOnClick	-
ParentControl	-
InFrame	-
InPage	-
InColumn	-
InMatrix	-
InMatrixHeading	-
Caption	Caption
CaptionML	CaptionML
ShowCaption	-
HorzAlign	-
VertAlign	-

ForeColor	-
BackColor	-
BackTransparent	-
Border	-
FontName	-
FontSize	-
FontBold	Style=Strong
FontItalic	-
FontStrikethru	-
FontUnderline	-
BitmapPos	-
PushAction	-
Bitmap	-
ToolTip	ToolTip
ToolTipML	ToolTipML
Description	Description
MinValue	MinValue
MaxValue	MaxValue
ValuesAllowed	ValuesAllowed
NextControl	-
OptionValue	-
CaptionClass	CaptionClass
SourceExpr	SourceExpr
AutoCalcField	-
UpdateOnAction	-
RunObject	-
RunFormView	-
RunFormLink	-
RunFormLinkType	-
RunFormOnRec	-
RunCommand	-

- Command button control property conversion

Request form command buttons are converted to request page fields.

The main command button conversion rules:

1. For each command button there is C/AL text constant created with Caption and CaptionML property values transferred. Text constant is used as a request page field SourceExpr.
2. Each new field created for command button conversion has Editable property value set to FALSE.

List of command button control properties which are transferred to RDLC format report field properties:

Request Form Command Button Property	Request Page Field Property
ID	ID
Name	Name
XPos	-
YPos	-
Width	-
Height	-
HorzGlue	-
VertGlue	-
Visible	Visible
Enabled	Enabled
Focusable	-
FocusOnClick	-
Default	-
Cancel	-
ParentControl	-
InFrame	-
InPage	-
Caption	C/AL text constant Caption property
CaptionML	C/AL text constant CaptionML property
ShowCaption	-
HorzAlign	-
VertAlign	-
ForeColor	-
BackColor	-
BackTransparent	-
Border	-
FontName	-
FontSize	-
FontBold	Style=Strong
FontItalic	-
FontStrikethru	-
FontUnderline	-
BitmapPos	-
AutoRepeat	-
PushAction	-

InvalidActionAppearance	-
Bitmap	-
Ellipsis	-
ToolTip	ToolTip
ToolTipML	ToolTipML
Description	Description
NextControl	-
CaptionClass	-
UpdateOnAction	-
RunObject	-
RunFormView	-
RunFormLink	-
RunFormLinkType	-
RunFormOnRec	-
RunCommand	-

- Frame control property conversion

Request form frames are converted to request page groups. All other controls which has frame ID value specified in ParentID property are added to frame group during conversion.

List of frame control properties which are transferred to RDLC format report group properties:

Request Form Frame Property	Request Page Group Property
ID	ID
Name	Name
XPos	-
YPos	-
Width	-
Height	-
HorzGlue	-
VertGlue	-
Visible	Visible
Enabled	Enabled
Editable	Editable
Focusable	-
ParentControl	-
InFrame	-
InPage	-
Caption	Caption

CaptionML	CaptionML
ShowCaption	-
ForeColor	-
BackColor	-
BackTransparent	-
Border	-
BorderColor	-
BorderStyle	-
BorderWidth	-
FontName	-
FontSize	-
FontBold	-
FontItalic	-
FontStrikethru	-
FontUnderline	-
TopLineOnly	-
ToolTip	-
ToolTipML	-
Description	Description
CaptionClass	-

4.3.2. Trigger

If same trigger exists, code and C/AL local variables from request form control trigger are transferred to proper request page field trigger, otherwise there is new C/AL function created and request from control trigger code and C/AL local variables transferred to it. C/AL function can be executed on some request page trigger or not executed at all if proper trigger does not exist.

There are some different trigger conversion rules applied according to the control type:

- Label trigger conversion

Label type control has only one trigger which is converted to C/AL function:

Label Trigger	Function Name
OnPush()	"SRB_" + ID + "OnPush"

- Text box trigger conversion

Text box type control triggers are transferred to proper request page field triggers or to new C/AL functions.

List of request form text box triggers which are transferred to proper request page field triggers:

Text Box Trigger	Request Page Field Trigger
OnValidate()	OnValidate()
OnLookup()	OnLookup()
OnDrillDown()	OnDrillDown()
OnAssistEdit()	OnAssistEdit()

List of request form text box triggers which are transferred to new C/AL functions:

Text Box Trigger	Function Name
OnActivate()	"SRB_" + ID + "OnActivate"
OnDeactivate()	"SRB_" + ID + "OnDeactivate"
OnFormat()	"SRB_" + ID + "OnFormat" (used in OnAfterGetRecord() trigger after actual OnAfterGetRecord() code)
OnBeforeInput()	"SRB_" + ID + "OnBeforeInput"
OnInputChange()	"SRB_" + ID + "OnInputChange"
OnAfterInput()	"SRB_" + ID + "OnAfterInput"
OnAfterValidate()	"SRB_" + ID + "OnAfterValidate" (used in OnValidate() trigger after actual OnValidate() code)

- Check box trigger conversion

Check box type control triggers are transferred to proper request page field triggers or to new C/AL functions.

List of request form check box triggers which are transferred to proper request page field triggers:

Check Box Trigger	Request Page Field Trigger
OnValidate()	OnValidate()

List of request form check box triggers which are transferred to new C/AL functions:

Check Box Trigger	Function Name
OnActivate()	"SRB_" + ID + "OnActivate"
OnDeactivate()	"SRB_" + ID + "OnDeactivate"
OnPush()	"SRB_" + ID + "OnPush" (used in OnValidate() trigger before actual OnValidate() code)
OnAfterValidate()	"SRB_" + ID + "OnAfterValidate" (used in OnValidate() trigger after actual OnValidate() code)

- Option button trigger conversion

Option button type control triggers are transferred to proper request page field triggers or to new C/AL functions.

List of request form option button triggers which are transferred to proper request page field triggers:

Option Button Trigger	Request Page Field Trigger
OnValidate()	OnValidate()

List of request form option button triggers which are transferred to new C/AL functions:

Option Button Trigger	Function Name
OnActivate()	"SRB_" + ID + "OnActivate"
OnDeactivate()	"SRB_" + ID + "OnDeactivate"
OnPush()	"SRB_" + ID + "OnPush" (used in OnValidate() trigger before actual OnValidate() code)
OnAfterValidate()	"SRB_" + ID + "OnAfterValidate" (used in OnValidate() trigger after actual OnValidate() code)

- Command button trigger conversion

Command button type control triggers are converted to C/AL functions:

Request Form Trigger	Function Name
OnActivate()	"SRB_" + ID + "OnActivate"
OnDeactivate()	"SRB_" + ID + "OnDeactivate"
OnPush()	"SRB_" + ID + "OnPush" (used in OnDrillDown() trigger)

5. Conversion of mostly used functions

5.1. CurrReport.CREATETOTALS

There are two main ways of how CurrReport.CREATETOTALS function is implemented during the conversion. Function's CurrReport.CREATETOTALS arguments are used with function Sum() in layout expressions or "SRB_Total" variables are created for the implementation.

5.1.1. Sum() function in layout expressions

Function's CurrReport.CREATETOTALS arguments are used with Sum() function in footer/groupfooter rows after conversion if there is no code with function's CurrReport.CREATETOTALS arguments used in footer/groupfooter sections triggers or arguments used only in CurrReport.SHOWOUTPUT function. After conversion function's CurrReport.CREATETOTALS arguments are used with scope parameter and few other functions, for example, instead of functions CurrReport.CREATETOTALS argument Amount used in SourceExpr property, after conversion there is layout text box created with expression - Sum(CDbl(IIF(Dataltem_Exported,Amount_DataSet_Name,0))), here:

Sum – function which returns sum of all specified values, evaluated in the given scope;

CDbl – function which converts an expression to type double;

Dataltem_Exported – variable which returns TRUE if record, which belongs to particular Dataltem is processed;

Amount_DataSet_Name – name of Report Dataset Designer column with Amount value in "Data Source".

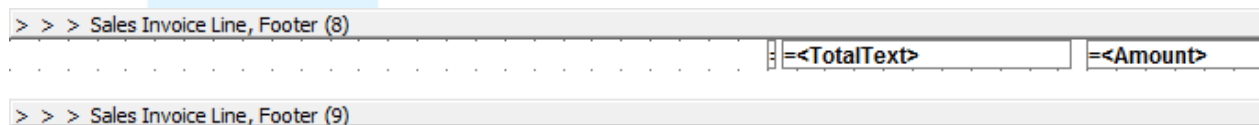
- Example. Conversion of footer text box which SourceExpr value is used in CurrReport.CREATETOTALS function.

If there is function CurrReport.CREATETOTALS used in some Dataltem's OnPreDataItem() trigger

```
Sales Invoice Line - OnPreDataItem()  
CurrReport.CREATETOTALS(Amount);
```

```
Sales Invoice Line - OnAfterGetRecord()
```

and function's CurrReport.CREATETOTLS argument is used in Dataltem's footer section control as a SourceExpr,



during conversion there is text box created with expression:

Set expression for: Value

```
=Sum(CDbl(IIF(Fields!SRB_Sales_Invoice_Line_Exported.Value,Fields!Sales_Invoice_Line_Amount.Value,0)))
```

Here Sales_Invoice_Line_Amount and SRB_Sales_Invoice_Line_Exported are 'Sales Invoice Line' DataItem columns

Column	SRB_DimensionLoop1_Body2_Visible	SRB_DimensionLoop1_Body2_Visible
DataItem	Sales Invoice Line	Sales Invoice Line
Column	'0'	SRB_Sales_Invoice_Line_GroupConstant
Column	SRB_DataItemExported(SRB_Sales_Invoice_Line_Exported)	SRB_Sales_Invoice_Line_Exported
Column	Amount	Sales_Invoice_Line_Amount
Column	Document No.	SRB_Sales_Invoice_Line__Document_No__

SRB_Sales_Invoice_Line_Exported is a boolean type variable which gets TRUE value on OnAfterGetReccord() trigger

```
SRB_Sales_Invoice_Line - OnAfterGetRecord()  
SRB_Sales_Invoice_Line_Exported := TRUE;  
  
SRB_Sales_Invoice_Line - OnPostDataItem()  

```

and its value is changed in C/AL function SRB_DataItemExported()

```
SRB_DataItemExported(VAR Exported : Boolean) : Boolean  
IF Exported THEN BEGIN  
    Exported := FALSE;  
    EXIT(TRUE);  
END ELSE  
    EXIT(FALSE);
```

5.1.2. "SRB_Total" C/AL global variables

If there are some code (not only function CurrReport.SHOWOUTPUT used one time) in some footer section trigger with function's CurrReport.CREATETOTALS arguments used, there are "SRB_Total" C/AL global variables created to implement function's CurrReport.CREATETOTALS functionality after conversion. Zero values are assigned for "SRB_Total" variables in OnPreDataItem() trigger, they are summed with function's CurrReport.CREATETOTALS arguments on OnAfterGetReccord() trigger and used in footer section trigger code instead of function's CurrReport.CREATETOTALS arguments after conversion.

- Example. Conversion of footer section trigger code if function's CurrReport.CREATETOTALS arguments used.

If there is function CurrReport.CREATETOTALS used in some DataItem's OnPreDataItem() trigger

```

Sales Invoice Line - OnPreDataItem()
CurrReport.CREATETOTALS(Amount,"Amount Including VAT");

Sales Invoice Line - OnAfterGetRecord()

```

and function's CurrReport.CREATETOTLS arguments used in DataItem's footer section trigger code

```

Sales Invoice Line, Footer (4) - OnPreSection()
IF (Amount <> "Amount Including VAT") THEN
    CurrReport.SHOWOUTPUT(TRUE)
ELSE
    CurrReport.SHOWOUTPUT(FALSE);

Sales Invoice Line, Footer (4) - OnPostSection()

```

after conversion "SRB_Total" variables are created for CurrReport.CREATETOTLS implementation.

Variables			
Text Constants			
Functions			
Name	DataType	Subtype	
FooterPrinted	Boolean		^
SRB_Sales_Invoice_Line_Exported	Boolean		
SRB_Sales_Invoice_Line_Header1_Visible	Boolean		
SRB_Sales_Invoice_Line_Header2_Visible	Boolean		
SRB_Sales_Invoice_Line_Body1_Visible	Boolean		
SRB_Sales_Invoice_Line_Footer1_Visible	Boolean		
SRB_Sales_Invoice_Line_Amount_Total	Decimal		
SRB_Sales_Invoice_Line_Amount_Including_VAT_Total	Decimal		
*▶			▼

Zero value is assigned to "SRB_Total" variables on OnPreDataItem() trigger.

```

Sales Invoice Line - OnPreDataItem()
CurrReport.CREATETOTALS(Amount,"Amount Including VAT");
SRB_Sales_Invoice_Line_Amount_Including_VAT_Total := 0;
SRB Sales Invoice Line Amount Total := 0;

Sales Invoice Line - OnAfterGetRecord()

```

Variables are summed with function's CurrReport.CRATETOTALS arguments on OnAfterGetReccord() trigger.


```

Sales Invoice Line - OnAfterGetRecord()
SRB_Sales_Invoice_Line_Body1_Visible := TRUE;
SRB_Sales_Invoice_Line_Exported := TRUE;
SRB_Sales_Invoice_Line_Amount_Total += Amount;
SRB_Sales_Invoice_Line_Amount_Including_VAT_Total += "Amount Including VAT";
Sales_Invoice_Line_Footer1_OnPreSection("Sales Invoice Line");

Sales Invoice Line - OnPostDataItem()

```

“SRB_Total” variables are used in footer section trigger function instead of function’s CurrReport.CREATETOTALS arguments after conversion.

```

Sales_Invoice_Line_Footer1_OnPreSection(SalesInvoiceLine : Record "Sales Invoice Line")
WITH SalesInvoiceLine DO BEGIN
IF (SRB_Sales_Invoice_Line_Amount_Total <> SRB_Sales_Invoice_Line_Amount_Including_VAT_Total) THEN
SRB_Sales_Invoice_Line_Footer1_Visible := (TRUE)
ELSE
SRB_Sales_Invoice_Line_Footer1_Visible := (FALSE);
END;

```

5.2. CurrReport.SHOWOUTPUT

There are two ways of how CurrReport.SHOWOUTPUT function is implemented during the conversion. The value of CurrReport.SHOWOUTPUT is transferred to the proper row visibility in RDLC format report layout or all the code from section trigger is added to new function, where CurrReport.SHOWOUTPUT is changed to new boolean type variable and after that used in proper row visibility in RDLC format report layout.

- Example 1. Function’s CurrReport.SHOWOUTPUT conversion to row visibility.

If function CurrReport.SHOWOUTPUT is used in OnPreSection() trigger,

```

Customer, Header (2) - OnPreSection()
CurrReport.SHOWOUTPUT(PrintAmountsInLCY);

```

```

Customer, Header (2) - OnPostSection()

```

during the conversion there is column with function’s CurrReport.SHOWOUTPUT parameter as „Data Source“ created in Report Dataset Designer

Exp...	Data Type	Data Source	Name	I...
	DataItem	Customer	Customer	^
	Column	PrintOnlyOnePerPage	Customer_PrintOnlyOnePerPage	
	Column	'0'	SRB_Customer_GroupConstant	
	Column	"No."	SRB_Customer__No__	
▶	Column	PrintAmountsInLCY	Customer_PrintAmountsInLCY	
	Column	SRB_DataItemExported(SRB_Customer_Exported)	SRB_Customer_Exported	▼

and used as row visibility expression.

Set expression for: Hidden

=IIf((Fields!Customer_PrintAmountsInLCY.Value) = TRUE,FALSE,TRUE)

- If function CurrReport.SHOWOUTPUT is used in OnPreSection() trigger,

```
Integer, Body (1) - OnPreSection()  
IF PrintToExcel THEN  
    CurrReport.SHOWOUTPUT(FALSE)  
ELSE  
    CurrReport.SHOWOUTPUT("G/L Account"."Account Type" = "G/L Account"."Account Type"::Posting);  
Integer, Body (1) - OnPostSection()
```

during the conversion there is C/AL global variable and function created (in this case boolean variable `SRB_Integer_Body1_Visible` and function `Integer_Body1_OnPreSection()`).

Variables				
Text Constants		Functions		
Name	DataType	Subtype	Length	
DirectPostingTxt	Text		30	^
SRB_G_L_Account_Exported	Boolean			
SRB_G_L_Account_Header1_Visible	Boolean			
SRB_BlankLineCounter_Exported	Boolean			
SRB_Integer_Exported	Boolean			
▶ SRB_Integer_Body1_Visible	Boolean			≡
SRB_Integer_Body2_Visible	Boolean			
SRB_Integer_Body3_Visible	Boolean			
SRB_PageGroupNo	Integer			▼

Code from OnPreSection() trigger is transferred to C/AL function and CurrReport.SHOWOUTPUT function replaced with boolean type variable.

```

Integer_Body1_OnPreSection(Integer : Record Integer)
WITH Integer DO BEGIN
    IF PrintToExcel THEN
        SRB_Integer_Body1_Visible := (FALSE)
    ELSE
        SRB_Integer_Body1_Visible := ("G/L Account"."Account Type" = "G/L Account"."Account Type"::Posting);
END;
Integer_Body2_OnPreSection(Integer : Record Integer)

```

There is column with boolean variable as „Data Source“ created in Report Dataset Designer

Exp...	Data Type	Data Source	Name
	DataItem	Integer	Integer
	Column	SRB_PageGroupNo	SRB_PageGroupNo
	Column	Number	SRB_Integer_Number
	Column	SRB_DataItemExported(SRB_Integer_Exported)	SRB_Integer_Exported
	Column	SRB_Integer_Body1_Visible	SRB_Integer_Body1_Visible
	Column	"G/L Account"."No."	G_L_Account__No__
	Column	PADSTR("G/L Account".Indentation * 2)+"G/L ...	PADSTR__G_L_Account__...

and used in Report Layout Designer as row visibility expression.

Set expression for: Hidden

```
=IIF(Max(Fields!SRB_Integer_Body1_Visible.Value),FALSE,TRUE)
```

The main cases when CurrReport.SHOWOUTPUT is replaced with boolean variable:

- Section trigger contains more code than just CurrReport.SHOWOUTPUT function used.
- More than one CurrReport.SHOWOUTPUT used in one section trigger.
- CurrReport.SHOWOUTPUT used in both same section triggers (OnPreSection() and OnPostSection()).
- If symbols :: used in CurrReport.SHOWOUTPUT function. This is frequently used with option type variables (for example, CurrReport.SHOWOUTPUT(„Account Type“ = „Account Type“::Posting)).
- If some C/AL function is used in CurrReport.SHOWOUTPUT function (for example, GETFILTERS), except CurrReport.PAGENO.

5.3. CurrReport.PAGENO

CurrReport.PAGENO is frequently used in Classic Dynamics NAV report code and control source expressions, but this function is not supported on RDLC format report layouts and always returns number 1. This makes CurrReport.PAGENO implementation difficult, but during conversion the most common cases of CurrReport.PAGENO usage are handled.

5.3.1. CurrReport.PAGENO used in control SourceExpr

If CurrReport.PAGENO function used in Classic Dynamics NAV report header or footer control SourceExpr property, this function is changed to Globals!PageNumber function in RDLC format report layout during conversion.

If header or footer section with CurrReport.PAGENO used have to be printed only on first/last page (PrintOnEveryPage = No), CurrReport.PAGENO is changed to constant value of number 1.

5.3.2. CurrReport.PAGENO used with CurrReport.SHOWOUTPUT

If (CurrReport.PAGENO = 1) expression used in some header sections trigger code with CurrReport.SHOWOUTPUT function, after report transformation sections are converted to tablix rows, which has RepeatOnNewPage = No (Static Properties -> Other -> RepeatOnNewPage). If there are some other same Dataltem header sections, which do not have CurrReport.SHOWOUTPUT(CurrReport.PAGENO = 1) code on section trigger and have to be repeated on every page (PrintOnEveryPage = Yes) in Classic report, during conversion different layout groups by constant values are created to separate rows which should be printed on every page from the ones which should be printed only on the first page.

- Example. Conversion of sections with code CurrReport.SHOWOUTPUT(CurrReport.PAGENO = 1) used.

If there are some header sections which triggers have code with CurrReport.SHOWOUTPUT(CurrReport.PAGENO = 1) used (marked in green) and all other header sections do not have such code and have to be repeated on every page,

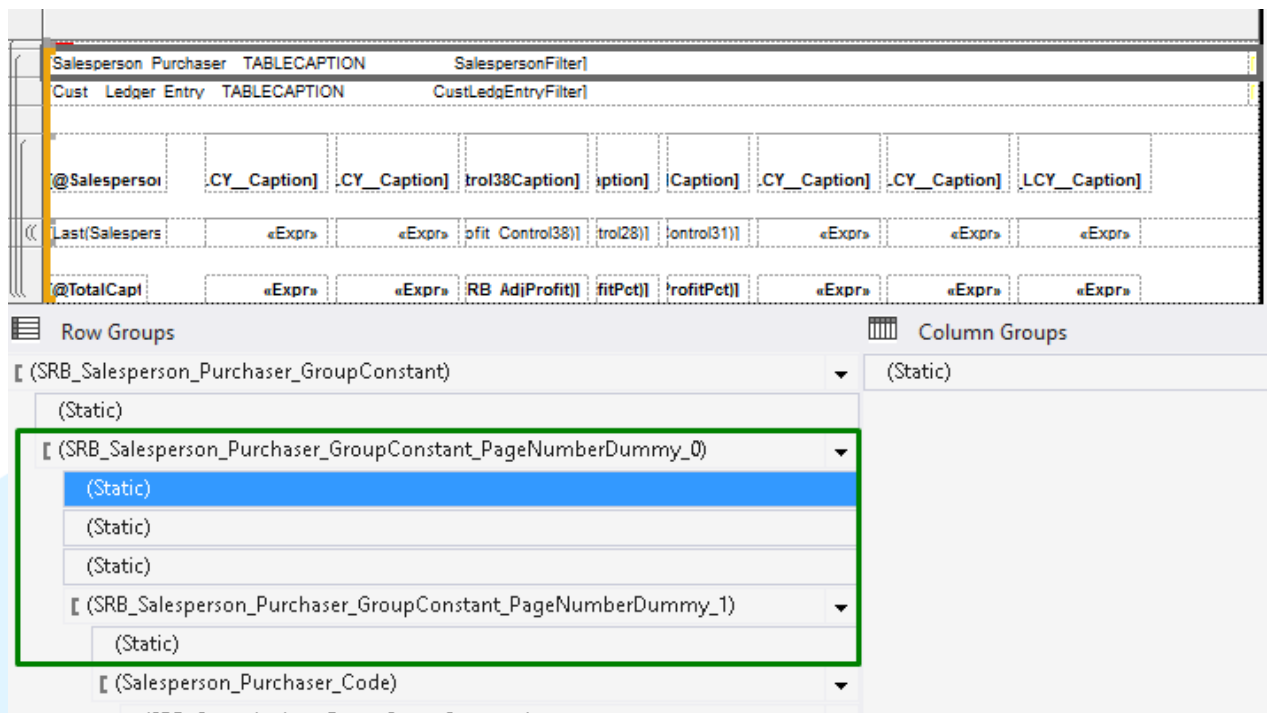
Salesperson/Purchaser, Header (1)									
Salesperson - Sales Statistics						=<FORMAT(TODAY,0,4)>			
=<STRSUBSTNO(Text000,PeriodText)>						Page =...			
=<COMPANYNAME>						=<USERID>			
All amounts are in LCY									
Salesperson/Purchaser, Header (2)									
=<"Salesperson/Purchaser".TABLECAPTION + ': ' + SalespersonFilter>									
Salesperson/Purchaser, Header (3)									
=<"Cust. Ledger Entry".TABLECAPTION + ': ' + CustLedgEntryFilter>									
Salesperson/Purchaser, Header (4)									
Salesperson/Purchaser, Header (5)									
Control24	Sales (LCY)	Profit (LCY)	Adjusted Profit (LCY)	Profit %	Adjusted Profit %	Invoice Disc. Amount (LCY)	Payment Disc. Given (LCY)	Pmt. Tolerance (LCY)	

Salesperson/Purchaser, Header (2) - OnPreSection()

CurrReport.SHOWOUTPUT((CurrReport.PAGENO = 1) AND (SalespersonFilter <> ''));

Salesperson/Purchaser, Header (2) - OnPostSection()

during conversion there are separate groups created with proper RepeatOnNewPage value for rows with CurrReport.SHOWOUTPUT(CurrReport.PAGENO = 1) and without it.



5.4. CurrReport.NEWPAGE

There are a few different implementations applied during report conversion depending on how CurrReport.NEWPAGE function is used.

5.4.1. CurrReport.NEWPAGE used with „DataItemTable“.“Field Name“

If CurrReport.NEWPAGE function is used on section trigger and is executed depending on „DataItemTable“.“Field Name“ value, new group by SRB_PageGroupNo with page break is created. SRB_PageGroupNo is integer type variable which is increased according to the same conditions as CurrReport.NEWPAGE function is executed in Classic Dynamics NAV report. Group by SRB_PageGroupNo is created as a parent group of „DataItemTable“ body group (group by „DataItemTable“ primary key).

- Example. Conversion of CurrReport.NEWPAGE function.

If there is Classic Dynamics NAV report with DataItems:

DataItem	Name	
G/L Account	<G/L Account>	^
Integer	BlankLineCounter	
Integer	<Integer>	

and CurrReport.NEWPAGE function used with „DataItemTable“.“Field Name“ condition

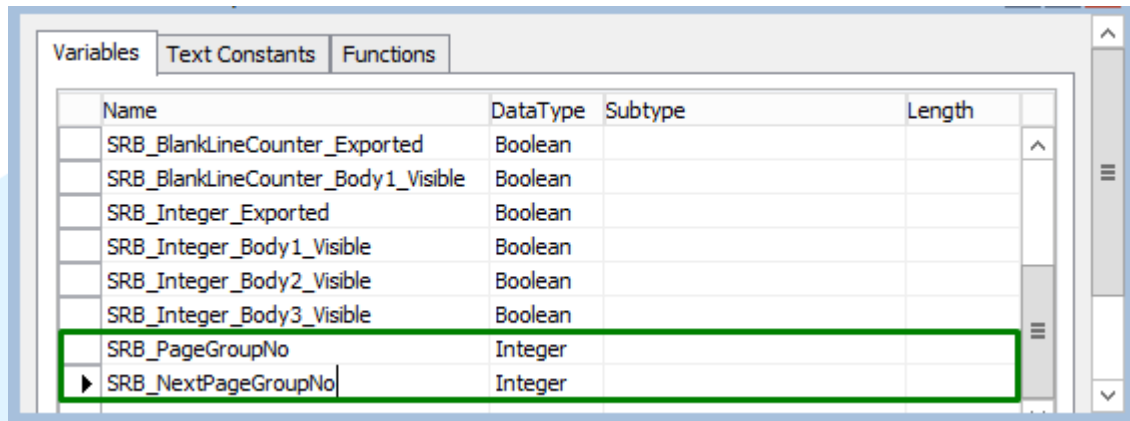
```
Integer, Body (3) - OnPreSection()
```

```
IF "G/L Account"."New Page" THEN
```

```
    CurrReport.NEWPAGE
```

```
Integer, Body (3) - OnPostSection()
```

during conversion there are two integer type variables added:



Name	DataType	Subtype	Length
SRB_BlankLineCounter_Exported	Boolean		
SRB_BlankLineCounter_Body1_Visible	Boolean		
SRB_Integer_Exported	Boolean		
SRB_Integer_Body1_Visible	Boolean		
SRB_Integer_Body2_Visible	Boolean		
SRB_Integer_Body3_Visible	Boolean		
SRB_PageGroupNo	Integer		
SRB_NextPageGroupNo	Integer		

and code added on „DataItemTable“ DataItem’s OnPreDataItem() trigger,

```
G/L Account - OnPreDataItem()
```

```
    SRB_PageGroupNo := 1;
```

```
    SRB_NextPageGroupNo := 1;
```

```
G/L Account - OnAfterGetRecord()
```

also code added on „DataItemTable“ DataItem’s OnAfterGetRecord() trigger.

```
G/L Account - OnAfterGetRecord()
```

```
    SRB_G_L_Account_Exported := TRUE;
```

```
    BEGIN
```

```
        SRB_PageGroupNo := SRB_NextPageGroupNo;
```

```
        IF "New Page" THEN
```

```
            SRB_NextPageGroupNo := SRB_PageGroupNo + 1;
```

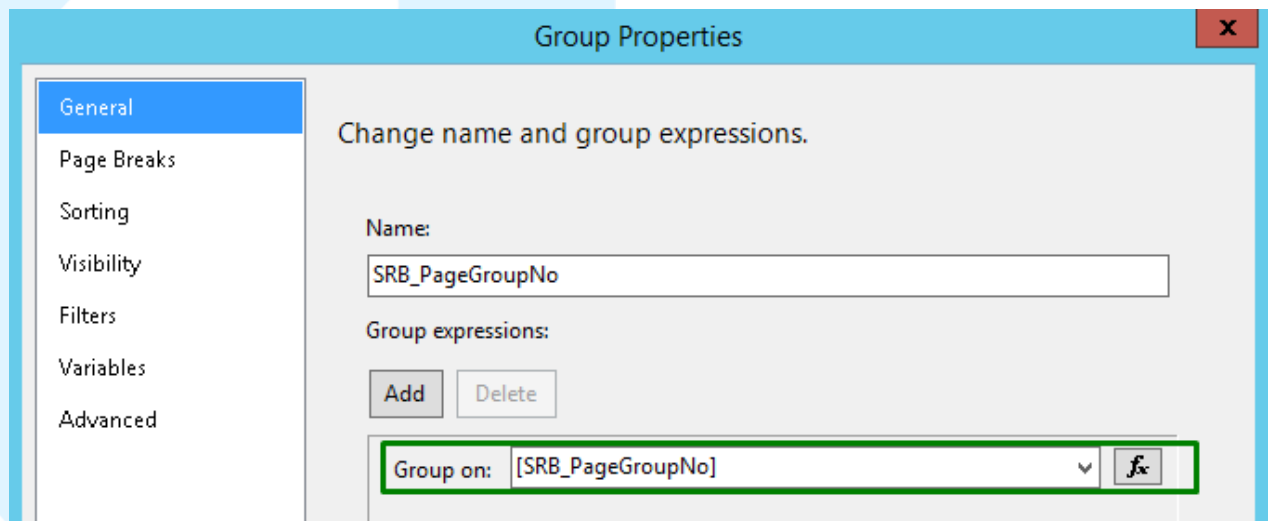
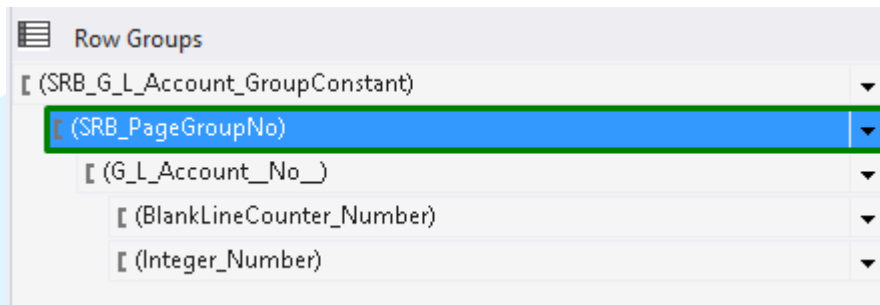
```
        END;
```

```
G/L Account - OnPostDataItem()
```

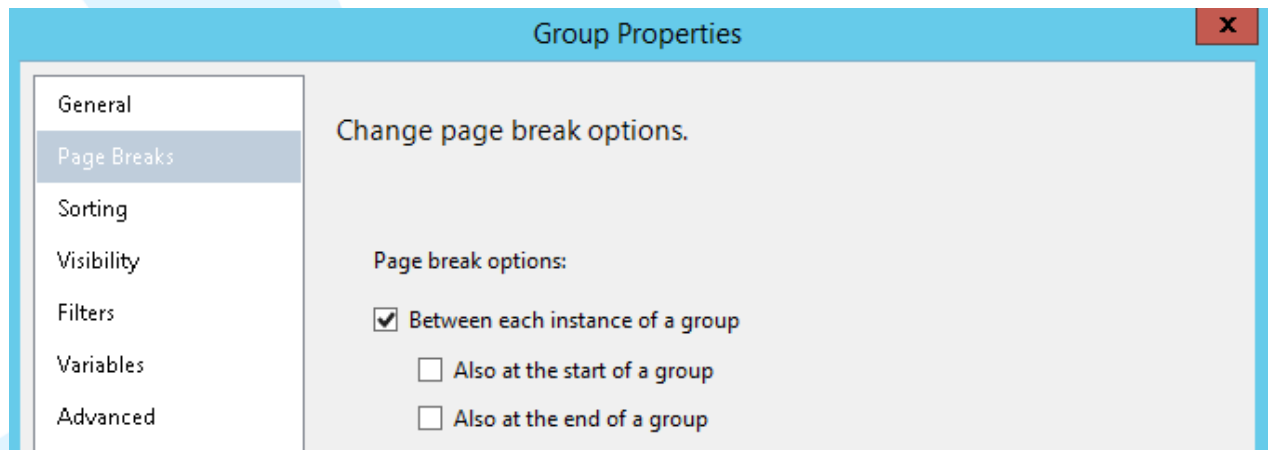
New column with SRB_PageGroupNo as „Data Source“ created

Exp...	Data Type	Data Source	Name	I...
	DataItem	G/L Account	G/L Account	
	Column	'0'	SRB_G_L_Account_GroupConstant	
	Column	"No."	SRB_G_L_Account_No_	
	Column	SRB_PageGroupNo	SRB_PageGroupNo	
	Column	SRB_DataItemExported(SRB_G_L_Account_Exported)	SRB_G_L_Account_Exported	
	Column	SRB_G_L_Account_Header1_Visible	SRB_G_L_Account_Header1_Visible	
	Column	COMPANYNAME	COMPANYNAME	

and new group by SRB_PageGroupNo value created



with page break option „Between each instance of a group“ selected.



5.4.2. CurrReport.NEWPAGE used on OnAfterGetRecord trigger without „DataltemTable“.“Field Name“

If function CurrReport.NEWPAGE is used in Classic Dynamics NAV report trigger without „DataltemTable“.“Field Name“ condition, then during conversion there is new C/AL integer type variable SRB_PageGroupNo created and increased instead of CurrReport.NEWPAGE code. Also group with page break by SRB_PageGroupNo is created as a parent group of „DataltemTable“ body group (group by „DataltemTable“ primary key) where function CurrReport.NEWPAGE is used.

- Example. Conversion of CurrReport.NEWPAGE function.

If function CurrReport.NEWPAGE is used in some Dataltem trigger,

```
G/L Entry - OnAfterGetRecord()
IF "Source Code" <> SourceCode.Code THEN
    IF NOT SourceCode.GET("Source Code") THEN
        SourceCode.INIT;
IF "No. Series" <> NoSeries.Code THEN
    IF NOT NoSeries.GET("No. Series") THEN
        NoSeries.INIT;

IF ("No. Series" <> LastNoSeriesCode) OR FirstRecord THEN BEGIN
    IF "No. Series" = '' THEN
        AddError(Text000)
    ELSE
        AddError(
            STRSUBSTNO(
                Text001,
                "No. Series",NoSeries.Description));
    IF NOT FirstRecord THEN
        CurrReport.NEWPAGE;
    NewPage := TRUE;
END ELSE BEGIN
    IF LastDocNo <> '' THEN
        IF NOT ("Document No." IN [LastDocNo,INCSTR(LastDocNo)]) THEN
            AddError(Text002)
        ELSE
            IF "Posting Date" < LastPostingDate THEN
                AddError(Text003);
    NewPage := FALSE;
END;
```

then after conversion there is new C/AL global variable SRB_PageGroupNo created

Variables				
Text Constants				
Functions				
Name	DataType	Subtype	Length	
SRB_G_L_Entry_Header3_Visible	Boolean			
SRB_ErrorLoop_Exported	Boolean			
SRB_ErrorLoop_Body1_Visible	Boolean			
SRB_ErrorLoop_Body2_Visible	Boolean			
SRB_GLEntry_Exported	Boolean			
SRB_GLEntry_Body1_Visible	Boolean			
SRB_PageGroupNo	Integer			

and its value is increased in the same place as function CurrReport.NEWPAGE was used.

```

G/L Entry - OnAfterGetRecord()
SRB_G_L_Entry_Exported := TRUE;
BEGIN
  IF "Source Code" <> SourceCode.Code THEN
    IF NOT SourceCode.GET("Source Code") THEN
      SourceCode.INIT;
  IF "No. Series" <> NoSeries.Code THEN
    IF NOT NoSeries.GET("No. Series") THEN
      NoSeries.INIT;
  IF ("No. Series" <> LastNoSeriesCode) OR FirstRecord THEN BEGIN
    IF "No. Series" = '' THEN
      AddError(Text000)
    ELSE
      AddError(
        STRSUBSTNO(
          Text001,
          "No. Series",NoSeries.Description));
    IF NOT FirstRecord THEN
      SRB_PageGroupNo := SRB_PageGroupNo + 1;
    NewPage := TRUE;
  END ELSE BEGIN
    IF LastDocNo <> '' THEN
      IF NOT ("Document No." IN [LastDocNo,INCSTR(LastDocNo)]) THEN
        AddError(Text002)
      ELSE
        IF "Posting Date" < LastPostingDate THEN
          AddError(Text003);
    NewPage := FALSE;
  END;

```

New column with SRB_PageGroupNo as „Data Source“ created

Exp...	Data Type	Data Source	Name	I...
	DataItem	G/L Entry	G/L Entry	
	Column	'0'	SRB_G_L_Entry_GroupConstant	
	Column	"Entry No."	SRB_G_L_Entry__Entry_No__	
	Column	SRB_PageGroupNo	SRB_PageGroupNo	
	Column	"Document No."	SRB_G_L_Entry__Document_No__	

and new group by SRB_PageGroupNo value created

Row Groups

- (SRB_G_L_Entry_GroupConstant)
 - (SRB_G_L_Entry_PageGroupNo)**
 - (G_L_Entry__Entry_No__)
 - (ErrorLoop_Number)
 - (GLEntry__Entry_No__)

Group Properties

General

Change name and group expressions.

Name: SRB_G_L_Entry_PageGroupNo

Group expressions:

Add Delete

Group on: [SRB_PageGroupNo] fx

with page break option „Between each instance of a group“ selected.

Group Properties

General

Page Breaks

Sorting

Visibility

Filters

Variables

Advanced

Change page break options.

Page break options:

☒ Between each instance of a group

☐ Also at the start of a group

☐ Also at the end of a group

5.5. CurrReport.NEWPAGEPERRECORD

If CurrReport.NEWPAGEPERRECORD function is used with assignment in DataItem's OnPreDataItem() trigger, new group by DataItem primary key is created which has BreakLocation=Between (Group Properties -> PageBreaks -> BreakLocation) and is executed if CurrReport.NEWPAGEPERRECORD returns TRUE.

- Example. Conversion of CurrReport.NEWPAGEPERRECORD function.

If there is function CurrReport.NEWPAGEPERRECORD used with assignment in some DataItem's OnPreDataItem() trigger,

```
Salesperson/Purchaser - OnPreDataItem()  
CurrReport.NEWPAGEPERRECORD := PrintOnlyOnePerPage;  
  
CurrReport.CREATETOTALS(  
    "Cust. Ledger Entry"."Sales (LCY)","Cust. Ledger Entry"."Profit (LCY)",  
    AdjProfit,ProfitCommissionAmt,AdjProfitCommissionAmt,SalesCommissionAmt);  
  
Salesperson/Purchaser - OnAfterGetRecord()
```

then during conversion there is column with function CurrReport.NEWPAGEPERRECORD value and DataItem table primary key as a „Data Source“ created.

E...	Data Type	Data Source	Name	I...
	DataItem	Salesperson/Purchaser	Salesperson/Purchaser	^
	Column	'0'	SRB_Salesperson_Purchaser_GroupConstant	
▶	Column	PrintOnlyOnePerPage	Salesperson_Purchaser_PrintOnlyOnePerPage	
	Column	Code	SRB_Salesperson_Purchaser_Code	
	Column	SRB_DataItemExported(SRB_Salesperson_Purchaser_Exported)	SRB_Salesperson_Purchaser_Exported	▼

Also group created with grouping expression:

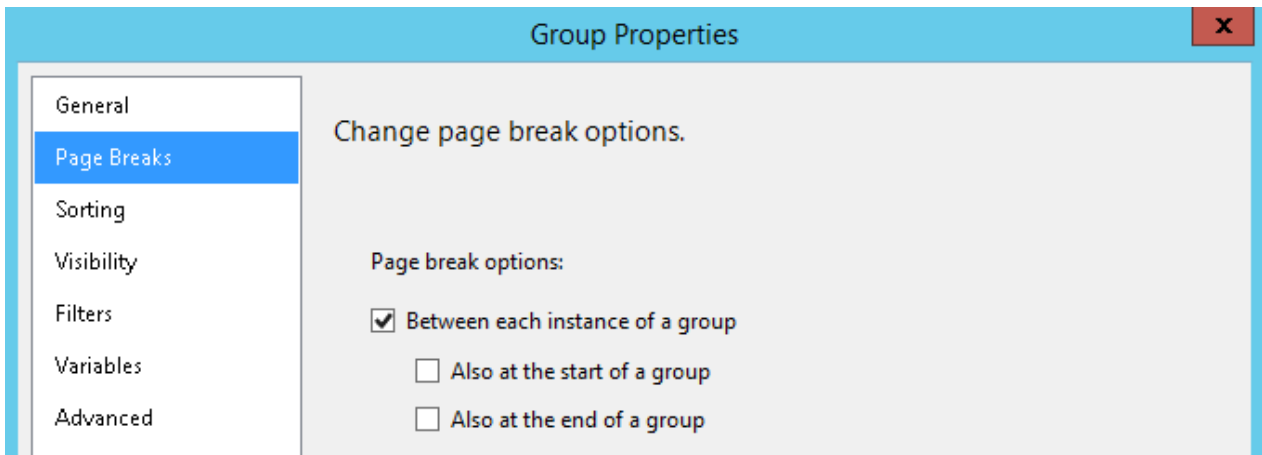
Row Groups
[(SRB_Salesperson_Purchaser_GroupConstant) ▼
[(Salesperson_Purchaser_PrintOnlyOnePerPageGroup) ▼
[(Salesperson_Purchaser_Code) ▼
[(SRB_Cust_Ledger_Entry_GroupConstant) ▼
[(Cust_Ledger_Entry_Entry_No_) ▼

Expression ✕

Set expression for: GroupExpression

=IIF(Fields!Salesperson_Purchaser_PrintOnlyOnePerPage.Value,Fields!SRB_Salesperson_Purchaser_Code.Value,true)

and page break option „Between each instance of a group“ selected.



5.6. EXIT

If EXIT function is used in some Classic Dynamics NAV report Dataltem OnAfterGetRecord() trigger, code from same Dataltem sections is still executed. If code from Dataltem sections triggers is added to C/AL functions during the conversion, these functions are executed before EXIT function so that the flow would be the same as in Classic. There can be also BEGIN END sentence added if it is necessary (for example, if EXIT used with IF THEN sentence without BEGIN END part).

- Example. Conversion of EXIT function.

If there is EXIT function used in some Dataltem's OnPreDataltem() trigger

```

DimensionLoop1 - OnAfterGetRecord()
IF Number = 1 THEN BEGIN
    IF NOT PostedDocDim1.FIND('-') THEN
        CurrReport.BREAK;
    END ELSE
        IF NOT Continue THEN
            CurrReport.BREAK;

    CLEAR(DimText);
    Continue := FALSE;
    REPEAT
        OldDimText := DimText;
        IF DimText = '' THEN
            DimText := STRSUBSTNO(
                '%1 %2',PostedDocDim1."Dimension Code",PostedDocDim1."Dimension Value Code")
        ELSE
            DimText :=
                STRSUBSTNO(
                    '%1, %2 %3',DimText,
                    PostedDocDim1."Dimension Code",PostedDocDim1."Dimension Value Code");
        IF STRLEN(DimText) > MAXSTRLEN(OldDimText) THEN BEGIN
            DimText := OldDimText;
            Continue := TRUE;
        EXIT;
        END;
    UNTIL (PostedDocDim1.NEXT = 0);

DimensionLoop1 - OnPostDataItem()

```

and DataItem (in this case DimensionLoop1) has some sections with code which are converted to new C/AL functions (for example, SRB_DimensionLoop1_Body1_OnPreSection() and SRB_DimensionLoop1_Body2_OnPreSection()), then section trigger functions are executed before EXIT execution.

```

DimensionLoop1 - OnAfterGetRecord()
SRB_DimensionLoop1_Body2_Visible := TRUE;
SRB_DimensionLoop1_Body1_Visible := TRUE;
SRB_DimensionLoop1_Exported := TRUE;
BEGIN
    IF Number = 1 THEN BEGIN
        IF NOT PostedDocDim1.FIND('-') THEN
            CurrReport.BREAK;
        END ELSE
            IF NOT Continue THEN
                CurrReport.BREAK;
        CLEAR(DimText);
        Continue := FALSE;
        REPEAT
            OldDimText := DimText;
            IF DimText = '' THEN
                DimText := STRSUBSTNO(
                    '%1 %2',PostedDocDim1."Dimension Code",PostedDocDim1."Dimension Value Code")
            ELSE
                DimText :=
                    STRSUBSTNO(
                        '%1, %2 %3',DimText,
                        PostedDocDim1."Dimension Code",PostedDocDim1."Dimension Value Code");
            IF STRLEN(DimText) > MAXSTRLEN(OldDimText) THEN BEGIN
                DimText := OldDimText;
                Continue := TRUE;
                DimensionLoop1_Body1_OnPreSection(DimensionLoop1);
                DimensionLoop1_Body2_OnPreSection(DimensionLoop1);
                EXIT;
            END;
        UNTIL (PostedDocDim1.NEXT = 0);
    END;
    DimensionLoop1_Body1_OnPreSection(DimensionLoop1);
    DimensionLoop1_Body2_OnPreSection(DimensionLoop1);
DimensionLoop1 - OnPostDataItem()

```

5.7. ISSERVICETIER

As function ISSERVICETIER always returns FALSE if Classic Dynamics NAV report is executed, after conversion this function is commented and replaced with FALSE value.

- Example. Conversion of ISSERVICETIER function.

If there is function ISSERVICETIER used somewhere in Classic Dynamics NAV report code,

```

CopyLoop - OnPreDataItem()
NoOfLoops := ABS(NoOfCopies) + Cust."Invoice Copies" + 1;
IF NoOfLoops <= 0 THEN
    NoOfLoops := 1;
CopyText := '';
SETRANGE(Number,1,NoOfLoops);
IF ISSERVICETIER THEN
    OutputNo := 1;

CopyLoop - OnAfterGetRecord()

```

then during conversion this function is commented and replaced with FALSE value.

```

CopyLoop - OnPreDataItem()
NoOfLoops := ABS(NoOfCopies) + Cust."Invoice Copies" + 1;
IF NoOfLoops <= 0 THEN
    NoOfLoops := 1;
CopyText := '';
SETRANGE(Number,1,NoOfLoops);
IF {ISSERVICETIER} FALSE THEN
    OutputNo := 1;

CopyLoop - OnAfterGetRecord()

```

6. Conversion Options

There are a few converter options which can be selected/unselected before conversion and have influence on conversion results:

- **Adjust Font Size** – reduces font size to fit in available text box height;
- **Substitute Fonts** – applies Segoe UI font for text boxes that do not have font family specified;
- **Add missing ENU Captions** – if ENU caption translation is not available, automatically creates it by copying the first available caption translation;
- **DataSet Name Style** – if „Prefix DataItem“ option choosed, Report Dataset Designer column name has syntax: „DataItemTable“ + field name,

Exp...	Data Type	Data Source	Name	I...
	DataItem	Sales Invoice Header	Sales Invoice Header	^
	Column	"No."	SRB_Sales_Invoice_Header__No__	
	Column	SRB_DataItemExported(SRB_...	SRB_Sales_Invoice_Header_Exported	
	Column	"Sales Invoice Header".FIELD...	Sales_Invoice_Header__Bill_to_Customer_No__Caption	

if „Postfix DataItem“ option choosed, Report Dataset Designer column name has syntax: field name + „DataItemTable“.

Exp...	Data Type	Data Source	Name	I...
	DataItem	Sales Invoice Header	Sales Invoice Header	^
	Column	"No."	SRB_No__Sales_Invoice_Header	
	Column	SRB_DataItemExported(SRB_...	SRB_Sales_Invoice_Header_Exported	
	Column	"Sales Invoice Header".FIELD...	Sales_Invoice_Header__Bill_to_Customer_No__Caption	

- **Add Visibility Expressions** – creates visibility expressions (recommended true).
- **Convert To AL** – converts report from classic to rdlc and to AL.
- **Add Horizontal Paddings** – sets 1pt left and right paddings on textboxes.
- **Add Documentation Trigger** – adds specified text in documentation trigger.
- **Add Version List** – sets version list for converted reports.
- **Create Copy of SRD Report** – creates SRD report for converted report.

Figure 11. Converter options.

The screenshot shows the 'Edit - Report Converter' dialog box with the following sections and options:

- License**
 - Total Complexity: 0
- Select Converter Options**
 - Adjust Font Size: ☒
 - Substitute Fonts: ☐
 - Add missing ENU Captions: ☒
 - DataSet Name Style: Prefix Dataltem (dropdown)
 - Add Visibility Expressions: ☒
 - Convert To Al: ☐
 - Add Horizontal Paddings: ☒
- Documentation**
 - Add Documentation Trigger: (empty dropdown)
 - Add Version List: (empty dropdown)
- Report Designer**
 - Create Copy of SRD Report: ☐
 - Last Designer Report ID used: 56000

Buttons: OK, Cancel

After complexity calculation where is a possibility to correct some of report header/footer conversion options. You can chose where to place your header/footer sections in body or in header/footer.

Convert To Options ⚙️ ^				
Find				
Source	Section	Print On Every ...	Place In B...	Convert To
Customer		<input type="checkbox"/>	<input type="checkbox"/>	
	Header (1)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Report Header
	Header (2)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Tablix Header
	Header (3)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Tablix Header
	Header (4)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Tablix Header
	Footer (1)	<input type="checkbox"/>	<input type="checkbox"/>	Tablix Footer