



# Avalara AvaTax for Microsoft Dynamics NAV 2013 RTM

---

## Specific Code Change Document

Version 01

Revision date: 9/30/16

Product release: Microsoft NAV 7.00.33781.06.04

*Avalara may have patents, patent applications, trademarks, copyrights, or other intellectual property rights governing the subject matter in this document. Except as expressly provided in any written license agreement from Avalara, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.*

*© 2013 Avalara, Inc. All rights reserved.*

*Avalara, AvaTax, AvaTax Calc, AvaTax Returns, AvaTax Certs, AvaTax Local, AvaLocal, AvaTax POS, AvaPOS, AvaRates, TrustFile, BPObridge, Laserbridge+, Sales TaxII, Sales TaxPC, SalestaxPC+, StatetaxII, and StatetaxPC are either registered trademarks or trademarks of Avalara, Inc. in the United States or other countries.*

*All other trademarks are property of their respective owners..*



Release Avalara AvaTax NAV 2013 RTM 7.00.33781.06.01 .....	3
Release Avalara AvaTax NAV 2013 RTM 7.00.33781.06.02 .....	56
Release Avalara AvaTax NAV 2013 RTM 7.00.33781.06.03 .....	60
Release Avalara AvaTax NAV 2013 RTM 7.00.33781.06.04 .....	61

## Release Avalara AvaTax NAV 2013 RTM 7.00.33781.06.01

### Details:

- Fixed an issue when posting on sales order, when warehouse management turned off.
- Fixed an issue with Page 14073520 (AVA Cust. Entity Class) is wrong 'Type'.
- Fixed an issue with OnRelease code in CU 14073320.

## Changes in Avalara CodeUnits

### Codeunit 14073301 AVA Tax Engine

#### Changes in Version List:

Old Value	AVANA7.00.06.00
New Value	AVANA7.00.06.01

#### Changes in Global Text Constant:

Text14073212	:	TextConst	'ENU=NAV 7.00.33781.06.00'
Text14073212	:	TextConst	'ENU=NAV 7.00    7.00.33781.06.01'

#### New Global Variables:

Name	DataType	Subtype	Length
InvoiceNotShip	Boolean		
InvoiceAndShip	Boolean		
AvaDocumentNo	Code		20

#### Changes in function → fnCalcTax

##### Old Code

Search for below 1<sup>st</sup> line for old code and replace with new code

```

IF STRLEN(AvaTempText)>1 THEN FromAddress += AvaAddrSeparator+TmpZip;
IF STRLEN(FromAddress)>0 THEN AvaAddrSeparator:=',';
{
AvaTempText := TmpCountry;
IF STRLEN(AvaTempText)>1 THEN FromAddress += AvaAddrSeparator+TmpCountry;
IF STRLEN(FromAddress)>0 THEN AvaAddrSeparator:=',';
}

AVAAddrLatLongFlag:=FALSE;
IF AvaReleaseStatusFlag THEN BEGIN
IF (Avaconfig."Release Quantity" = '3') THEN BEGIN
AvaCallGetTax := FALSE;
AvaHead.RESET; //AVANA7.00.06.00
AvaHead.SETFILTER("Document No.",'%1','*'+SalesHead."No."+'*');
AvaHead.SETRANGE("Document Type",FORMAT(SalesHead."Document Type"));
IF AvaHead.FINDFIRST AND (AvaHead."Company ID" <> '') THEN BEGIN

IF NOT DocTypeSetToInv THEN BEGIN
IF SalesHead."Document Type" = SalesHead."Document Type"::Quote THEN
AvaCallGetTax := TRUE;
IF AvaCallGetTax = FALSE THEN IF (SalesHead."Document Type" <> SalesHead."Document Type"::Quote) THEN
IF AvaHead."Document Status" = 0 THEN AvaCallGetTax := TRUE;

```

```

END
ELSE BEGIN
  IF SalesHead."Document Type" <> SalesHead."Document Type"::Quote THEN
    IF AvaHead."Document Status" = 0 THEN AvaCallGetTax := TRUE;
  END;

  IF AvaCallGetTax = FALSE THEN IF (AvaHead."Original Address" <> FromAddress) THEN AvaCallGetTax := TRUE;

  IF AvaCallGetTax = FALSE THEN IF (AvaHead."Destination Address" <> ToAddress) THEN AvaCallGetTax := TRUE;

  IF AvaCallGetTax = FALSE THEN IF (AvaHead."Customer No." <> SalesHead."Sell-to Customer No.") THEN
AvaCallGetTax := TRUE;
  IF AvaCallGetTax = FALSE THEN IF (AvaHead."Document Date" <>SalesHead."Document Date") THEN AvaCallGetTax
:= TRUE;
  IF AvaCallGetTax = FALSE THEN IF (AvaHead."Exemption No." <> ExemptionNo) THEN AvaCallGetTax:= TRUE;
  SalesHead.CALCFIELDS(SalesHead.Amount);
  IF AvaCallGetTax = FALSE THEN IF ((AvaHead."Total Taxable Amount" + AvaHead."Tax Exempt Amount") <>
SalesHead.Amount) THEN AvaCallGetTax:= TRUE;

  IF AvaCallGetTax = FALSE THEN BEGIN
    SalesLine.SETRANGE("Document Type",SalesHead."Document Type");
    SalesLine.SETRANGE("Document No.",SalesHead."No.");
    SalesLine.SETFILTER(Type,'<>%1',SalesLine.Type::" ");

    SalesLine.CALCSUMS("Inv. Discount Amount");
    IF AvaCallGetTax = FALSE THEN IF SalesLine."Inv. Discount Amount" <> AvaHead."Tax Discount Amount" THEN
AvaCallGetTax:= TRUE;

    //SalesLine.SETFILTER("Qty. to Invoice",'<>0'); //AVANA7.00.06.00
    SalesLine.SETFILTER("Qty. to Invoice",'>=0'); //AVANA7.00.06.00
    IF SalesLine.FINDFIRST THEN BEGIN
      REPEAT
        AvaLine.RESET;
        AvaLine.SETRANGE("Tax Header No.",AvaHead."No.");
        AvaLine.SETRANGE("Tax Line No.",FORMAT(SalesLine."Line No."));
        AvaLine.SETRANGE("Item No.",FORMAT(SalesLine."No."));
        IF AvaLine.FINDFIRST THEN BEGIN
          IF AvaCallGetTax = FALSE THEN IF (AvaLine."Entity No." <> EntityUseCode) THEN AvaCallGetTax:= TRUE;
          IF AvaCallGetTax = FALSE THEN IF (AvaLine."Tax Area" <> SalesLine."Tax Area Code") THEN AvaCallGetTax:=
TRUE;

          fnGetItemDetails(SalesLine.Type,SalesLine."No.",SalesLine."Tax Group
Code",Ref1,Ref2,"G\AccNo.",TaxibilityCode,UnitPrice);
          IF AvaCallGetTax = FALSE THEN IF (AvaLine."Tax Code" <> TaxibilityCode) THEN AvaCallGetTax:= TRUE;

          AvalLineDiffQtyInv := AvaLine."Quantity Inv." - SalesLine."Qty. to Invoice";
          IF AvaCallGetTax = FALSE THEN IF (AvalLineDiffQtyInv <> 0) THEN AvaCallGetTax:= TRUE;

          AvalLineDiffAmtInv := (AvaLine."Total Taxable Amount" + AvaLine."Tax Exempt Amount" + AvaLine."Tax
Discount Amount")- SalesLine."Line Amount";
          IF AvaCallGetTax = FALSE THEN IF (AvalLineDiffAmtInv <> 0) THEN AvaCallGetTax:= TRUE;
        END
      ELSE AvaCallGetTax := TRUE;
    UNTIL (SalesLine.NEXT = 0) OR (AvaCallGetTax = TRUE);
  END;
END;

```

```

    END;
  END
  ELSE AvaCallGetTax:= TRUE;
  IF AvaCallGetTax = FALSE THEN
    EXIT(TRUE);
  END;
END;

fnRemoveRecord(AvaHead."No.");
AvaWindowOpen(StrMessage+Text14073221);
CASE SalesHead."Document Type" OF
  0 : PassDocNo:='QUO' + SalesHead."No.";
  1 : PassDocNo:='ORD' + SalesHead."No.";
  2 : PassDocNo:='INV' + SalesHead."No.";
  3 : PassDocNo:='CRM' + SalesHead."No.";
  4 : PassDocNo:='BLC' + SalesHead."No.";
  5 : PassDocNo:='RET' + SalesHead."No.";
END;

IF Avaconfig.FINDFIRST THEN;
// AVANA7.00.06.00 start
//IF Avaconfig."Enable AsynchronousAvaTax Post" THEN BEGIN
// AvaHead.RESET;
// AvaHead.SETRANGE("Document Type",FORMAT(SalesHead."Document Type"));
// AvaHead.SETFILTER("Document No.",'%1','*'+SalesHead."No."+'*');
// IF AvaHead.FINDLAST THEN
// IF AvaHead."Status Code" = 'OK' THEN
//   PassDocNo := PassDocNo + '-' + FORMAT(AvaHead.COUNT)
// ELSE
//   PassDocNo := AvaHead."Document No.";
//END;
AvaAsynPosDoc(PassDocNo,SalesHead."Document Type",SalesHead."No."); //AVANA7.00.06.00
// AVANA7.00.06.00 end

IF Avaconfig."Source Code" = Avaconfig."Source Code"::"Sell-to Customer" THEN BEGIN
  CustCode := SalesHead."Sell-to Customer No.";

  IF CustCode = " THEN
    CustCode := SalesHead."Sell-to Customer Template Code"
  END
  ELSE BEGIN
    CustCode := SalesHead."Bill-to Customer No.";
    IF CustCode = " THEN
      CustCode := SalesHead."Bill-to Customer Template Code"
    END;

  LocCode := "";
  CompInfo.GET;
  LrSalesLine.SETRANGE("Document Type",SalesHeaderType);
  LrSalesLine.SETRANGE("Document No.",SalesHeaderNo);
  LrSalesLine.SETFILTER(Type, '<>%1', LrSalesLine.Type::" ");
  //LrSalesLine.SETRANGE("Line No.",10000);
  //LrSalesLine.SETFILTER("Location Code", '<>%1','');
  IF LrSalesLine.FINDFIRST THEN
    LocCode := LrSalesLine."Location Code";

```

```

//AVANA7.00.06.00 start
//ELSE
IF (LocCode = "") AND (SalesHead."Location Code" <> "") THEN
  LocCode := SalesHead."Location Code";
IF (LocCode = "") THEN
  AvaGetLocationCode(SalesHead."Responsibility Center",LocCode);
//AVANA7.00.06.00 END

fnGetTaxHdr(oAddFrom,
  oAddTo,
  ShipMethod,
  Avaconfig."Company ID",
  PassDocNo,
  //SalesHead."Document Date",//CONNECT-1905
  AvaTaxCalcDate,//CONNECT-1905
  iDocType,
  TotalDisc*DCoefficient,
  ExemptionNo,
  DtlLevel,
  CustCode,
  EntityUseCode,
  SalesHead."Salesperson Code",
  SalesHead."External Document No.",
  "",
  oGetTaxRequest,
  OutStatus,
  LocCode
);
IF SalesHead."Currency Code" <> "" THEN BEGIN
  Tgcurrencycode:=SalesHead."Currency Code";
  ExRate:= SalesHead."Currency Factor";

  //ExDate:= SalesHead."Document Date";//CONNECT-1905
  ExDate:= AvaTaxCalcDate;//CONNECT-1905
  fnGetTaxHdrV3(Tgcurrencycode,ExRate,ExDate);
END;
AvaWindowClose;

```

## New Code

```

IF STRLEN(AvaTempText)>1 THEN FromAddress += AvaAddrSeparator+TmpZip;
IF STRLEN(FromAddress)>0 THEN AvaAddrSeparator:=',';

CASE SalesHead."Document Type" OF
  0 : PassDocNo:='QUO' + SalesHead."No.";
  1 : PassDocNo:='ORD' + SalesHead."No.";
  2 : PassDocNo:='INV' + SalesHead."No.";
  3 : PassDocNo:='CRM' + SalesHead."No.";
  4 : PassDocNo:='BLC' + SalesHead."No.";
  5 : PassDocNo:='RET' + SalesHead."No.";
END;

AvaDocumentNo := SalesHead."No.";

AvaAsynPosDoc(PassDocNo,SalesHead."Document Type",SalesHead."No.");

```

```

AVAAddrLatLongFlag:=FALSE;
IF AvaReleaseStatusFlag THEN BEGIN
  IF (Avaconfig."Release Quantity" = '3') THEN BEGIN
    AvaCallGetTax := FALSE;
    AvaHead.RESET;
    AvaHead.SETFILTER("Document No.", '%1', '*'+SalesHead."No."+'*');
    AvaHead.SETRANGE("Document Type", FORMAT(SalesHead."Document Type"));
    IF AvaHead.FINDFIRST AND (AvaHead."Company ID" <> '') THEN BEGIN

      IF NOT DocTypeSetToInv THEN BEGIN
        IF SalesHead."Document Type" = SalesHead."Document Type"::Quote THEN
          AvaCallGetTax := TRUE;
        IF AvaCallGetTax = FALSE THEN IF (SalesHead."Document Type" <> SalesHead."Document Type"::Quote) THEN
          IF AvaHead."Document Status" = 0 THEN AvaCallGetTax := TRUE;
        END
      ELSE BEGIN
        IF SalesHead."Document Type" <> SalesHead."Document Type"::Quote THEN
          IF AvaHead."Document Status" = 0 THEN AvaCallGetTax := TRUE;
        END;

        IF AvaCallGetTax = FALSE THEN IF (AvaHead."Original Address" <> FromAddress) THEN AvaCallGetTax := TRUE;

        IF AvaCallGetTax = FALSE THEN IF (AvaHead."Destination Address" <> ToAddress) THEN AvaCallGetTax := TRUE;

        IF AvaCallGetTax = FALSE THEN IF (AvaHead."Customer No." <> SalesHead."Sell-to Customer No.") THEN
          AvaCallGetTax := TRUE;
        IF AvaCallGetTax = FALSE THEN IF (AvaHead."Document Date" <> SalesHead."Document Date") THEN AvaCallGetTax
:= TRUE;
        IF AvaCallGetTax = FALSE THEN IF (AvaHead."Exemption No." <> ExemptionNo) THEN AvaCallGetTax:= TRUE;
        SalesHead.CALCFIELDS(SalesHead.Amount);
        IF AvaCallGetTax = FALSE THEN IF ((AvaHead."Total Taxable Amount" + AvaHead."Tax Exempt Amount") <>
SalesHead.Amount) THEN AvaCallGetTax:= TRUE;

        IF AvaCallGetTax = FALSE THEN BEGIN
          SalesLine.SETRANGE("Document Type", SalesHead."Document Type");
          SalesLine.SETRANGE("Document No.", SalesHead."No.");
          SalesLine.SETFILTER(Type, '<>%1', SalesLine.Type::" ");

          SalesLine.CALCSUMS("Inv. Discount Amount");
          IF AvaCallGetTax = FALSE THEN IF SalesLine."Inv. Discount Amount" <> AvaHead."Tax Discount Amount" THEN
AvaCallGetTax:= TRUE;

          SalesLine.SETFILTER("Qty. to Invoice", '>=0');
          IF SalesLine.FINDFIRST THEN BEGIN
            REPEAT
              AvaLine.RESET;
              AvaLine.SETRANGE("Tax Header No.", AvaHead."No.");
              AvaLine.SETRANGE("Tax Line No.", FORMAT(SalesLine."Line No."));
              AvaLine.SETRANGE("Item No.", FORMAT(SalesLine."No."));
              IF AvaLine.FINDFIRST THEN BEGIN
                IF AvaCallGetTax = FALSE THEN IF (AvaLine."Entity No." <> EntityUseCode) THEN AvaCallGetTax:= TRUE;
                IF AvaCallGetTax = FALSE THEN IF (AvaLine."Tax Area" <> SalesLine."Tax Area Code") THEN AvaCallGetTax:=
TRUE;

```

```

        fnGetItemDetails(SalesLine.Type,SalesLine."No.",SalesLine."Tax Group
Code",Ref1,Ref2,"G\LAccNo.",TaxibilityCode,UnitPrice);
        IF AvaCallGetTax = FALSE THEN IF (AvaLine."Tax Code" <> TaxibilityCode) THEN AvaCallGetTax:= TRUE;

        AvalLineDiffQtyInv := AvaLine."Quantity Inv." - SalesLine."Qty. to Invoice";
        IF AvaCallGetTax = FALSE THEN IF (AvalLineDiffQtyInv <> 0) THEN AvaCallGetTax:= TRUE;

        AvalLineDiffAmtInv := (AvaLine."Total Taxable Amount" + AvaLine."Tax Exempt Amount" + AvaLine."Tax
Discount Amount")- SalesLine."Line Amount";
        IF AvaCallGetTax = FALSE THEN IF (AvalLineDiffAmtInv <> 0) THEN AvaCallGetTax:= TRUE;
        IF AvaCallGetTax = FALSE THEN BEGIN
            IF ((AvaLine."Tax Tax Amount" + AvaLine."Total Taxable Amount" + AvaLine."Tax Exempt Amount") <> 0)
THEN
                SalesLine."Amount Including VAT" := AvaLine."Tax Tax Amount" + AvaLine."Total Taxable Amount" +
AvaLine."Tax Exempt Amount";
                SalesLine.MODIFY;
                END;
            END
            ELSE AvaCallGetTax := TRUE;
            UNTIL (SalesLine.NEXT = 0) OR (AvaCallGetTax = TRUE);
            END;
            END;
            END
            ELSE AvaCallGetTax:= TRUE;
            IF AvaCallGetTax = FALSE THEN
                EXIT(TRUE);
            END;
            END;

fnRemoveRecord(AvaHead."No.");
AvaWindowOpen(StrMessage+Text14073221);

AvaDocumentNo := SalesHead."No.";

IF Avaconfig."Source Code" = Avaconfig."Source Code"::"Sell-to Customer" THEN BEGIN
    CustCode := SalesHead."Sell-to Customer No.";

    IF CustCode = " THEN
        CustCode := SalesHead."Sell-to Customer Template Code"
    END
    ELSE BEGIN
        CustCode := SalesHead."Bill-to Customer No.";
        IF CustCode = " THEN
            CustCode := SalesHead."Bill-to Customer Template Code"
        END;

    LocCode := "";
    CompInfo.GET;
    LrSalesLine.SETRANGE("Document Type",SalesHeaderType);
    LrSalesLine.SETRANGE("Document No.",SalesHeaderNo);
    LrSalesLine.SETFILTER(Type, '<>%1', LrSalesLine.Type::" ");
    IF LrSalesLine.FINDFIRST THEN
        LocCode := LrSalesLine."Location Code";

    IF (LocCode = "") AND (SalesHead."Location Code" <> "") THEN

```



```

LocCode := SalesHead."Location Code";
IF (LocCode = '') THEN
  AvaGetLocationCode(SalesHead."Responsibility Center",LocCode);

fnGetTaxHdr(oAddFrom,
  oAddTo,
  ShipMethod,
  Avaconfig."Company ID",
  PassDocNo,
  AvaTaxCalcDate,
  iDocType,
  TotalDisc*DCoefficient,
  ExemptionNo,
  DtlLevel,
  CustCode,
  EntityUseCode,
  SalesHead."Salesperson Code",
  SalesHead."External Document No.",
  "",
  oGetTaxRequest,
  OutStatus,
  LocCode
);
IF SalesHead."Currency Code" <> '' THEN BEGIN
  Tgcurrencycode:=SalesHead."Currency Code";
  ExRate:= SalesHead."Currency Factor";

  ExDate:= AvaTaxCalcDate;
  fnGetTaxHdrV3(Tgcurrencycode,ExRate,ExDate);
END;
AvaWindowClose;

```

## Old Code

Search for below 1<sup>st</sup> line for old code and replace with new code

```

IF NOT DocTypeSetToInv THEN BEGIN
  IF SalesHead."Currency Code" = '' THEN
    RecCurrency.InitRoundingPrecision
  ELSE BEGIN
    SalesHead.TESTFIELD("Currency Factor");
    RecCurrency.GET(SalesHead."Currency Code");
    RecCurrency.TESTFIELD("Amount Rounding Precision");
  END;

  IF NOT AvaReleaseStatisticsBln THEN BEGIN
    IF Avaconfig."Statistics Quantity" = '1' THEN BEGIN
      LineAmtPassed:=ROUND(ROUND(SalesLine.Quantity * UnitPrice,RecCurrency."Amount Rounding Precision") -
        ROUND((ROUND(SalesLine.Quantity * UnitPrice,RecCurrency."Amount Rounding Precision") *
SalesLine."Line Discount %")/100,RecCurrency."Amount Rounding Precision")
        ,RecCurrency."Amount Rounding Precision",<'); //AVANA7.00.06.00
      QtyPerLine:=SalesLine.Quantity;
    END ELSE IF Avaconfig."Statistics Quantity" = '2' THEN BEGIN
      IF (SalesLine."Document Type" = SalesLine."Document Type" :: "Return Order") OR (SalesLine."Document Type" =
SalesLine."Document Type" :: "Credit Memo") THEN BEGIN
        LineAmtPassed:=ROUND(ROUND(SalesLine."Return Qty. to Receive"*UnitPrice,RecCurrency."Amount Rounding
Precision") -

```

```

ROUND((ROUND(SalesLine."Return Qty. to Receive" * UnitPrice,RecCurrency."Amount Rounding
Precision") * SalesLine."Line Discount %")/100,RecCurrency."Amount Rounding Precision")
,RecCurrency."Amount Rounding Precision",<'); //AVANA7.00.06.00
QtyPerLine:=SalesLine."Return Qty. to Receive";
END ELSE BEGIN
LineAmtPassed:=ROUND(ROUND(SalesLine."Qty. to Ship"*UnitPrice,RecCurrency."Amount Rounding Precision") -
ROUND((ROUND(SalesLine."Qty. to Ship" * UnitPrice,RecCurrency."Amount Rounding Precision") *
SalesLine."Line Discount %")/100,RecCurrency."Amount Rounding Precision")
,RecCurrency."Amount Rounding Precision",<'); //AVANA7.00.06.00
QtyPerLine:=SalesLine."Qty. to Ship";
END;
END ELSE IF Avaconfig."Statistics Quantity" = '3' THEN BEGIN
LineAmtPassed:=ROUND(ROUND(SalesLine."Qty. to Invoice"*UnitPrice,RecCurrency."Amount Rounding Precision") -
-
ROUND((ROUND(SalesLine."Qty. to Invoice" * UnitPrice,RecCurrency."Amount Rounding Precision") *
SalesLine."Line Discount %")/100,RecCurrency."Amount Rounding Precision")
,RecCurrency."Amount Rounding Precision",<'); //AVANA7.00.06.00
QtyPerLine:=SalesLine."Qty. to Invoice";
END;
END ELSE BEGIN
IF Avaconfig."Release Quantity" = '1' THEN BEGIN
LineAmtPassed:=ROUND(ROUND(SalesLine.Quantity*UnitPrice,RecCurrency."Amount Rounding Precision") -
ROUND((ROUND(SalesLine.Quantity * UnitPrice,RecCurrency."Amount Rounding Precision") *
SalesLine."Line Discount %")/100,RecCurrency."Amount Rounding Precision")
,RecCurrency."Amount Rounding Precision",<'); //AVANA7.00.06.00
QtyPerLine:=SalesLine.Quantity;
END ELSE IF Avaconfig."Release Quantity" = '2' THEN BEGIN
IF (SalesLine."Document Type" = SalesLine."Document Type" :: "Credit Memo") OR (SalesLine."Document Type" =
SalesLine."Document Type" :: "Return Order") THEN BEGIN
LineAmtPassed:=ROUND(ROUND(SalesLine."Return Qty. to Receive"*UnitPrice,RecCurrency."Amount Rounding
Precision") -
ROUND((ROUND(SalesLine."Return Qty. to Receive" * UnitPrice,RecCurrency."Amount Rounding
Precision") * SalesLine."Line Discount %")/100,RecCurrency."Amount Rounding Precision")
,RecCurrency."Amount Rounding Precision",<'); //AVANA7.00.06.00
QtyPerLine:=SalesLine."Return Qty. to Receive";
END ELSE BEGIN
LineAmtPassed:=ROUND(ROUND(SalesLine."Qty. to Ship"*UnitPrice,RecCurrency."Amount Rounding Precision") -
ROUND((ROUND(SalesLine."Qty. to Ship" * UnitPrice,RecCurrency."Amount Rounding Precision") *
SalesLine."Line Discount %")/100,RecCurrency."Amount Rounding Precision")
,RecCurrency."Amount Rounding Precision",<'); //AVANA7.00.06.00
QtyPerLine:=SalesLine."Qty. to Ship";
END;
END ELSE IF Avaconfig."Release Quantity" = '3' THEN BEGIN
LineAmtPassed:=ROUND(ROUND(SalesLine."Qty. to Invoice"*UnitPrice,RecCurrency."Amount Rounding Precision") -
-
ROUND((ROUND(SalesLine."Qty. to Invoice" * UnitPrice,RecCurrency."Amount Rounding Precision") *
SalesLine."Line Discount %")/100,RecCurrency."Amount Rounding Precision")
,RecCurrency."Amount Rounding Precision",<'); //AVANA7.00.06.00
QtyPerLine:=SalesLine."Qty. to Invoice";
END;
END;
END
ELSE BEGIN
InvoiceFlag := Avaconfig.Invoice;
IF (SalesLine."Qty. Shipped Not Invoiced") <> 0 THEN BEGIN

```

```

IF InvoiceFlag THEN BEGIN
  IF SalesLine."Qty. Shipped Not Invoiced" < SalesLine."Qty. to Invoice" THEN BEGIN
    LineAmtPassed:=ROUND(ROUND(SalesLine."Qty. Shipped Not Invoiced"*UnitPrice,RecCurrency."Amount
Rounding Precision") -
      ROUND((ROUND(SalesLine."Qty. Shipped Not Invoiced" * UnitPrice,RecCurrency."Amount Rounding
Precision") * SalesLine."Line Discount %")/100,RecCurrency."Amount Rounding Precision")
      ,RecCurrency."Amount Rounding Precision",'<'); //AVANA7.00.06.00
    QtyPerLine:=SalesLine."Qty. Shipped Not Invoiced";
  END
  ELSE BEGIN
    LineAmtPassed:=ROUND(ROUND(SalesLine."Qty. to Invoice"*UnitPrice,RecCurrency."Amount Rounding
Precision") -
      ROUND((ROUND(SalesLine."Qty. to Invoice" * UnitPrice,RecCurrency."Amount Rounding Precision")
* SalesLine."Line Discount %")/100,RecCurrency."Amount Rounding Precision")
      ,RecCurrency."Amount Rounding Precision",'<'); //AVANA7.00.06.00
    QtyPerLine:=SalesLine."Qty. to Invoice";
  END;
END
ELSE BEGIN
  IF SalesLine."Qty. Shipped Not Invoiced" < SalesLine."Qty. to Invoice" THEN BEGIN
    LineAmtPassed:=ROUND(ROUND(((SalesLine."Qty. Shipped Not Invoiced" + SalesLine."Qty. to
Ship")*UnitPrice),RecCurrency."Amount Rounding Precision") -
      ROUND((ROUND((SalesLine."Qty. Shipped Not Invoiced" + SalesLine."Qty. to Ship")*
UnitPrice,RecCurrency."Amount Rounding Precision") * SalesLine."Line Discount %")/100,RecCurrency."Amount Rounding
Precision")
      ,RecCurrency."Amount Rounding Precision",'<'); //AVANA7.00.06.00
    QtyPerLine:=(SalesLine."Qty. Shipped Not Invoiced")+(SalesLine."Qty. to Ship");
  END
  ELSE BEGIN
    LineAmtPassed:=ROUND(ROUND(SalesLine."Qty. to Invoice"*UnitPrice,RecCurrency."Amount Rounding
Precision") -
      ROUND((ROUND(SalesLine."Qty. to Invoice" * UnitPrice,RecCurrency."Amount Rounding Precision")
* SalesLine."Line Discount %")/100,RecCurrency."Amount Rounding Precision")
      ,RecCurrency."Amount Rounding Precision",'<'); //AVANA7.00.06.00
    QtyPerLine:=SalesLine."Qty. to Invoice";
  END
  END;
  END;
  IF (SalesLine."Qty. Shipped Not Invoiced" = 0) AND (InvoiceFlag = FALSE) THEN BEGIN
    LineAmtPassed:=ROUND(ROUND(SalesLine."Qty. to Invoice"*UnitPrice,RecCurrency."Amount Rounding Precision")
-
      ROUND((ROUND(SalesLine."Qty. to Invoice" * UnitPrice,RecCurrency."Amount Rounding Precision") *
SalesLine."Line Discount %")/100,RecCurrency."Amount Rounding Precision")
      ,RecCurrency."Amount Rounding Precision",'<'); //AVANA7.00.06.00
    QtyPerLine:=SalesLine."Qty. to Invoice";
  END;
END;

IF (SalesLine."Document Type" = SalesLine."Document Type" :: "Credit Memo") OR (SalesLine."Document Type" =
SalesLine."Document Type" :: "Return Order") THEN BEGIN
  IF (SalesLine."Return Qty. Rcd. Not Invd.") <> 0 THEN BEGIN
    Avaconfig.FINDFIRST;
    InvoiceFlag := Avaconfig.Invoice;
    IF InvoiceFlag THEN BEGIN
      IF SalesLine."Return Qty. Rcd. Not Invd." < SalesLine."Qty. to Invoice" THEN BEGIN

```

```

LineAmtPassed:=ROUND(ROUND(SalesLine."Return Qty. Rcd. Not Invd."*UnitPrice,RecCurrency."Amount
Rounding Precision") -
    ROUND((ROUND(SalesLine."Return Qty. Rcd. Not Invd." * UnitPrice,RecCurrency."Amount Rounding
Precision") * SalesLine."Line Discount %")/100,RecCurrency."Amount Rounding Precision")
    ,RecCurrency."Amount Rounding Precision",<'); //AVANA7.00.06.00
QtyPerLine:=SalesLine."Return Qty. Rcd. Not Invd.";
END ELSE BEGIN
LineAmtPassed:=ROUND(ROUND(SalesLine."Qty. to Invoice"*UnitPrice,RecCurrency."Amount Rounding
Precision") -
    ROUND((ROUND(SalesLine."Qty. to Invoice" * UnitPrice,RecCurrency."Amount Rounding Precision")
* SalesLine."Line Discount %")/100,RecCurrency."Amount Rounding Precision")
    ,RecCurrency."Amount Rounding Precision",<'); //AVANA7.00.06.00
QtyPerLine:=SalesLine."Qty. to Invoice";
END;
END ELSE BEGIN
IF SalesLine."Return Qty. Rcd. Not Invd." < SalesLine."Qty. to Invoice" THEN BEGIN
LineAmtPassed:=ROUND(ROUND((SalesLine."Return Qty. Rcd. Not Invd."+SalesLine."Return Qty. to
Receive")*UnitPrice,RecCurrency."Amount Rounding Precision") -
    ROUND((ROUND((SalesLine."Return Qty. Rcd. Not Invd." + SalesLine."Return Qty. to Receive")*
UnitPrice,RecCurrency."Amount Rounding Precision")
    * SalesLine."Line Discount %")/100,RecCurrency."Amount Rounding Precision")
    ,RecCurrency."Amount Rounding Precision",<'); //AVANA7.00.06.00
QtyPerLine:=(SalesLine."Return Qty. Rcd. Not Invd.")+(SalesLine."Return Qty. to Receive");
END ELSE BEGIN
LineAmtPassed:=ROUND(ROUND(SalesLine."Qty. to Invoice"*UnitPrice,RecCurrency."Amount Rounding
Precision") -
    ROUND((ROUND(SalesLine."Qty. to Invoice" * UnitPrice,RecCurrency."Amount Rounding Precision")
* SalesLine."Line Discount %")/100,RecCurrency."Amount Rounding Precision")
    ,RecCurrency."Amount Rounding Precision",<'); //AVANA7.00.06.00
QtyPerLine:=SalesLine."Qty. to Invoice";
END
END;
END;
IF (SalesLine."Return Qty. Rcd. Not Invd." = 0) AND (InvoiceFlag=FALSE) THEN BEGIN
LineAmtPassed:=ROUND(ROUND(SalesLine."Qty. to Invoice"*UnitPrice,RecCurrency."Amount Rounding Precision")
-
    ROUND((ROUND(SalesLine."Qty. to Invoice" * UnitPrice,RecCurrency."Amount Rounding Precision") *
SalesLine."Line Discount %")/100,RecCurrency."Amount Rounding Precision")
    ,RecCurrency."Amount Rounding Precision",<'); //AVANA7.00.06.00
QtyPerLine:=SalesLine."Qty. to Invoice";
END;
END;
END;

IF TDocType=1 THEN BEGIN
IF QtyPerLine < 0 THEN
QtyPerLine:=QtyPerLine*-1;
END ELSE IF TDocType=5 THEN BEGIN
IF QtyPerLine < 0 THEN
QtyPerLine:=QtyPerLine*-1;
LineAmtPassed:= LineAmtPassed * -1;
END;

```

**New Code**

```
IF NOT DocTypeSetToInv THEN BEGIN
```

```

IF SalesHead."Currency Code" = " THEN
  RecCurrency.InitRoundingPrecision
ELSE BEGIN
  SalesHead.TESTFIELD("Currency Factor");
  RecCurrency.GET(SalesHead."Currency Code");
  RecCurrency.TESTFIELD("Amount Rounding Precision");
END;

IF NOT AvaReleaseStatisticsBln THEN BEGIN
  CASE Avaconfig."Statistics Quantity" OF
    '1' : QtyPerLine:=SalesLine.Quantity;
    '2' : IF (SalesLine."Document Type" = SalesLine."Document Type" :: "Return Order") OR (SalesLine."Document
Type" = SalesLine."Document Type" :: "Credit Memo") THEN
      QtyPerLine:=SalesLine."Return Qty. to Receive"
    ELSE
      QtyPerLine:=SalesLine."Qty. to Ship";
    '3' : QtyPerLine:=SalesLine."Qty. to Invoice";
  END;
END ELSE BEGIN
  CASE Avaconfig."Release Quantity" OF
    '1' : QtyPerLine:=SalesLine.Quantity;
    '2' : IF (SalesLine."Document Type" = SalesLine."Document Type" :: "Credit Memo") OR (SalesLine."Document
Type" = SalesLine."Document Type" :: "Return Order") THEN
      QtyPerLine:=SalesLine."Return Qty. to Receive"
    ELSE
      QtyPerLine:=SalesLine."Qty. to Ship";
    '3' : QtyPerLine:=SalesLine."Qty. to Invoice";
  END;
END;
END
ELSE BEGIN
  InvoiceFlag := Avaconfig.Invoice;
  IF ((SalesHead.Ship OR SalesHead.Receive) AND SalesHead.Invoice) THEN
    InvoiceAndShip := TRUE
  ELSE
    InvoiceAndShip := FALSE;

  IF (SalesLine."Qty. Shipped Not Invoiced") <> 0 THEN BEGIN
    IF InvoiceFlag THEN BEGIN
      IF SalesLine."Qty. Shipped Not Invoiced" < SalesLine."Qty. to Invoice" THEN BEGIN
        QtyPerLine:=SalesLine."Qty. Shipped Not Invoiced";
      END
    ELSE BEGIN
      QtyPerLine:=SalesLine."Qty. to Invoice";
    END;
  END
  ELSE BEGIN
    IF SalesLine."Qty. Shipped Not Invoiced" < SalesLine."Qty. to Invoice" THEN BEGIN
      QtyPerLine:=(SalesLine."Qty. Shipped Not Invoiced")+(SalesLine."Qty. to Ship");
    END
    ELSE BEGIN
      QtyPerLine:=SalesLine."Qty. to Invoice";
    END
  END;
END;
END;

```

```

IF (SalesLine."Qty. Shipped Not Invoiced" = 0) THEN
  IF (InvoiceNotShip OR InvoiceAndShip )THEN
    QtyPerLine:=SalesLine."Qty. to Invoice"
  ELSE
    QtyPerLine := 0;

  IF (SalesLine."Document Type" = SalesLine."Document Type" :: "Credit Memo") OR (SalesLine."Document Type" =
SalesLine."Document Type" :: "Return Order") THEN BEGIN
  IF (SalesLine."Return Qty. Rcd. Not Invd.") <> 0 THEN BEGIN
    InvoiceFlag := Avaconfig.Invoice;
    IF InvoiceFlag THEN
      IF SalesLine."Return Qty. Rcd. Not Invd." < SalesLine."Qty. to Invoice" THEN
        QtyPerLine:=SalesLine."Return Qty. Rcd. Not Invd."
      ELSE
        QtyPerLine:=SalesLine."Qty. to Invoice"
      ELSE
        IF SalesLine."Return Qty. Rcd. Not Invd." < SalesLine."Qty. to Invoice" THEN
          QtyPerLine:=(SalesLine."Return Qty. Rcd. Not Invd.")+(SalesLine."Return Qty. to Receive")
        ELSE
          QtyPerLine:=SalesLine."Qty. to Invoice";
        END;

    IF (SalesLine."Return Qty. Rcd. Not Invd." = 0) THEN
      IF (InvoiceNotShip OR InvoiceAndShip) THEN
        QtyPerLine:=SalesLine."Qty. to Invoice"
      ELSE
        QtyPerLine := 0;
      END;
    END;

    LineAmtPassed:=ROUND(ROUND(QtyPerLine * UnitPrice,RecCurrency."Amount Rounding Precision") -
      ROUND((ROUND(QtyPerLine * UnitPrice,RecCurrency."Amount Rounding Precision") * SalesLine."Line
Discount %")/100,RecCurrency."Amount Rounding Precision")
      ,RecCurrency."Amount Rounding Precision",'<');

    IF TDocType=1 THEN BEGIN
      IF QtyPerLine < 0 THEN
        QtyPerLine:=QtyPerLine*-1;
      END ELSE IF TDocType=5 THEN BEGIN
        IF QtyPerLine < 0 THEN
          QtyPerLine:=QtyPerLine*-1;
          LineAmtPassed:= LineAmtPassed * -1;
        END;

```

Search for below 1<sup>st</sup> line and make the changes in red colored lines

```

AvaWindowOpen(StrMessage+Text14073223);

//HdrID:=fnObjToTbl(oGetTaxRequest,oGetTaxResult,oGetTaxResult.DocCode,TDocType,DT2DATE(oGetTaxResult.DocDate)
,DocumentType);
HdrID:=fnObjToTbl(oGetTaxRequest,oGetTaxResult,oGetTaxResult.DocCode,TDocType,DT2DATE(oGetTaxResult.DocDate),D
ocumentType,AvaDocumentNo);

IF NOT DocTypeSetToInv THEN
  fnQtyAmtOrd(HdrID,SalesLine,FALSE)

```

```
ELSE
  fnQtyAmtOrd(HdrID,SalesLine,TRUE);
```

### Changes in function → fnCalcTaxBatch

Search for below 1<sup>st</sup> line and make the changes in red colored lines

```
IF Avaconfig."Source Code" = Avaconfig."Source Code"::"Sell-to Customer" THEN
  CustCode := SalesHead."Sell-to Customer No."
ELSE
  CustCode := SalesHead."Bill-to Customer No.";

AvaDocumentNo := SalesHead."No.";
IF Avaconfig.FINDFIRST THEN;
```

Search below 1<sup>st</sup> line for old code and replace with new code

Old Code

```
RecCurrency.TESTFIELD("Amount Rounding Precision");
END;

IF NOT AvaReleaseStatisticsBln THEN BEGIN //CONNECT-1596
  IF Avaconfig."Statistics Quantity" = '1' THEN BEGIN
    LineAmtPassed:=ROUND(ROUND(SalesLine.Quantity*UnitPrice,RecCurrency."Amount Rounding Precision") -
      ROUND((ROUND(SalesLine.Quantity * UnitPrice,RecCurrency."Amount Rounding Precision") *
SalesLine."Line Discount %")/100,RecCurrency."Amount Rounding Precision")
      ,RecCurrency."Amount Rounding Precision",'<'); //AVANA7.00.06.00
    QtyPerLine:=SalesLine.Quantity;
  END ELSE IF Avaconfig."Statistics Quantity" = '2' THEN BEGIN
    IF (SalesLine."Document Type" = SalesLine."Document Type" :: "Credit Memo") OR (SalesLine."Document Type" =
SalesLine."Document Type" :: "Return Order") THEN BEGIN
      LineAmtPassed:=ROUND(ROUND(SalesLine."Return Qty. to Receive"*UnitPrice,RecCurrency."Amount Rounding
Precision") -
        ROUND((ROUND(SalesLine."Return Qty. to Receive" * UnitPrice,RecCurrency."Amount Rounding
Precision") * SalesLine."Line Discount %")/100)
        ,RecCurrency."Amount Rounding Precision",'<'); //AVANA7.00.06.00
      QtyPerLine:=SalesLine."Return Qty. to Receive";
    END ELSE BEGIN
      LineAmtPassed:=ROUND(ROUND(SalesLine."Qty. to Ship"*UnitPrice,RecCurrency."Amount Rounding Precision") -
        ROUND((ROUND(SalesLine."Qty. to Ship" * UnitPrice,RecCurrency."Amount Rounding Precision") *
SalesLine."Line Discount %")/100,RecCurrency."Amount Rounding Precision")
        ,RecCurrency."Amount Rounding Precision",'<'); //AVANA7.00.06.00
      QtyPerLine:=SalesLine."Qty. to Ship";
    END;
  END ELSE IF Avaconfig."Statistics Quantity" = '3' THEN BEGIN
    LineAmtPassed:=ROUND(ROUND(SalesLine."Qty. to Invoice"*UnitPrice,RecCurrency."Amount Rounding Precision")
-
      ROUND((ROUND(SalesLine."Qty. to Invoice" * UnitPrice,RecCurrency."Amount Rounding Precision") *
SalesLine."Line Discount %")/100,RecCurrency."Amount Rounding Precision")
      ,RecCurrency."Amount Rounding Precision",'<'); //AVANA7.00.06.00
    QtyPerLine:=SalesLine."Qty. to Invoice";
  END;
END ELSE BEGIN
  IF Avaconfig."Release Quantity" = '1' THEN BEGIN
    LineAmtPassed:=ROUND(ROUND(SalesLine.Quantity*UnitPrice,RecCurrency."Amount Rounding Precision") -
      ROUND((ROUND(SalesLine.Quantity * UnitPrice,RecCurrency."Amount Rounding Precision") *
SalesLine."Line Discount %")/100,RecCurrency."Amount Rounding Precision")
```

```

        ,RecCurrency."Amount Rounding Precision",<'); //AVANA7.00.06.00
    QtyPerLine:=SalesLine.Quantity;
    END ELSE IF Avaconfig."Release Quantity" = '2' THEN BEGIN
    IF (SalesLine."Document Type" = SalesLine."Document Type" :: "Credit Memo") OR (SalesLine."Document Type" =
SalesLine."Document Type" :: "Return Order") THEN BEGIN
        LineAmtPassed:=ROUND(ROUND(SalesLine."Return Qty. to Receive"*UnitPrice,RecCurrency."Amount Rounding
Precision") -
            ROUND((ROUND(SalesLine."Return Qty. to Receive" * UnitPrice,RecCurrency."Amount Rounding
Precision") * SalesLine."Line Discount %")/100,RecCurrency."Amount Rounding Precision")
            ,RecCurrency."Amount Rounding Precision",<'); //AVANA7.00.06.00
        QtyPerLine:=SalesLine."Return Qty. to Receive";
    END ELSE BEGIN
    LineAmtPassed:=ROUND(ROUND(SalesLine."Qty. to Ship"*UnitPrice,RecCurrency."Amount Rounding Precision") -
            ROUND((ROUND(SalesLine."Qty. to Ship" * UnitPrice,RecCurrency."Amount Rounding Precision") *
SalesLine."Line Discount %")/100,RecCurrency."Amount Rounding Precision")
            ,RecCurrency."Amount Rounding Precision",<'); //AVANA7.00.06.00
        QtyPerLine:=SalesLine."Qty. to Ship";
    END;
    END ELSE IF Avaconfig."Release Quantity" = '3' THEN BEGIN
    LineAmtPassed:=ROUND(ROUND(SalesLine."Qty. to Invoice"*UnitPrice,RecCurrency."Amount Rounding Precision")
-
            ROUND((ROUND(SalesLine."Qty. to Invoice" * UnitPrice,RecCurrency."Amount Rounding Precision") *
SalesLine."Line Discount %")/100,RecCurrency."Amount Rounding Precision")
            ,RecCurrency."Amount Rounding Precision",<'); //AVANA7.00.06.00
        QtyPerLine:=SalesLine."Qty. to Invoice";
    END;
    END;
    //AVANA7.00.06.00 end
    END;

    IF TDocType=1 THEN BEGIN
    IF QtyPerLine < 0 THEN
        QtyPerLine:=QtyPerLine*-1;
    END ELSE IF TDocType=5 THEN BEGIN
    IF QtyPerLine < 0 THEN
        QtyPerLine:=QtyPerLine;
        LineAmtPassed:= LineAmtPassed * -1;
    END;

```

## New Code

```

    RecCurrency.TESTFIELD("Amount Rounding Precision");
    END;

    IF NOT AvaReleaseStatisticsBln THEN BEGIN
    CASE Avaconfig."Statistics Quantity" OF
        '1' : QtyPerLine:=SalesLine.Quantity;
        '2' : IF (SalesLine."Document Type" = SalesLine."Document Type" :: "Credit Memo") OR (SalesLine."Document Type"
= SalesLine."Document Type" :: "Return Order") THEN
            QtyPerLine:=SalesLine."Return Qty. to Receive"
        ELSE
            QtyPerLine:=SalesLine."Qty. to Ship";
        '3' : QtyPerLine:=SalesLine."Qty. to Invoice";
    END;
    END ELSE BEGIN
    CASE Avaconfig."Release Quantity" OF

```



```

'1' : QtyPerLine:=SalesLine.Quantity;
'2' : IF (SalesLine."Document Type" = SalesLine."Document Type" :: "Credit Memo") OR (SalesLine."Document Type"
= SalesLine."Document Type" :: "Return Order") THEN
    QtyPerLine:=SalesLine."Return Qty. to Receive"
    ELSE
    QtyPerLine:=SalesLine."Qty. to Ship";
'3' : QtyPerLine:=SalesLine."Qty. to Invoice";
END;
END;
END;

LineAmtPassed:=ROUND(ROUND(QtyPerLine*UnitPrice,RecCurrency."Amount Rounding Precision") -
    ROUND((ROUND(QtyPerLine * UnitPrice,RecCurrency."Amount Rounding Precision") * SalesLine."Line
Discount %)/100,RecCurrency."Amount Rounding Precision")
    ,RecCurrency."Amount Rounding Precision",'<');

IF TDocType=1 THEN BEGIN
    IF QtyPerLine < 0 THEN
        QtyPerLine:=QtyPerLine*-1;
    END ELSE IF TDocType=5 THEN BEGIN
        IF QtyPerLine < 0 THEN
            QtyPerLine:=QtyPerLine;
            LineAmtPassed:= LineAmtPassed * -1;
        END;
    END;

```

Search for below 1<sup>st</sup> line and make the changes in red colored lines

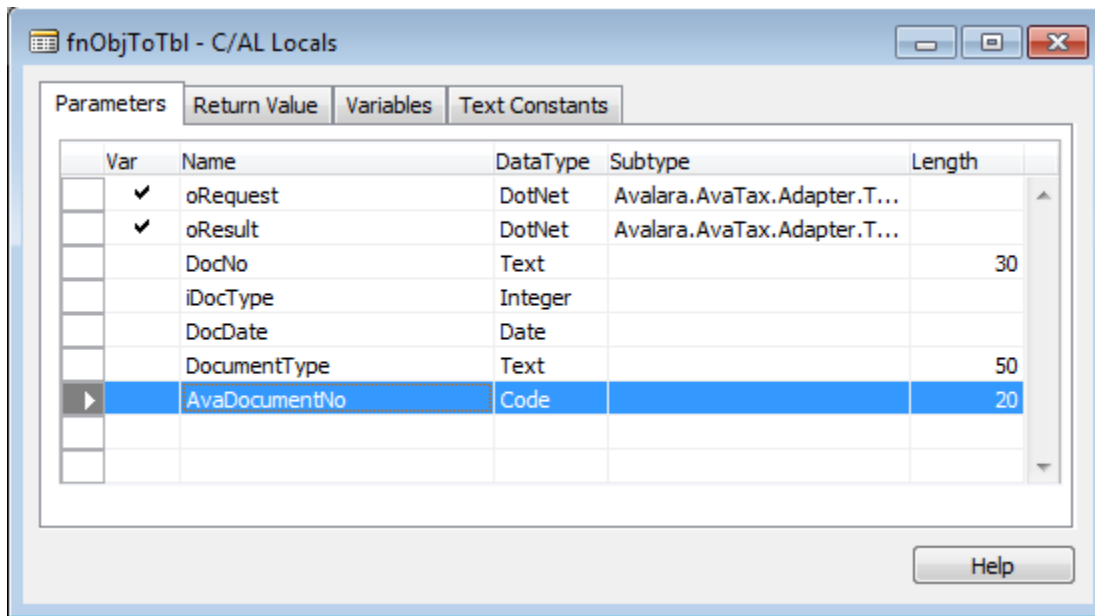
```

AvaWindowOpen(StrMessage+Text14073223);

//HdrID:=fnObjToTbl(oGetTaxRequest,oGetTaxResult,oGetTaxResult.DocCode,TDocType,DT2DATE(oGetTaxResult.DocDate)
,DocumentType);
HdrID:=fnObjToTbl(oGetTaxRequest,oGetTaxResult,oGetTaxResult.DocCode,TDocType,DT2DATE(oGetTaxResult.DocDate),D
ocumentType,AvaDocumentNo);
fnQtyAmtOrd(HdrID,SalesLine,FALSE);
IF NOT Avaconfig."Show Message Dialog" THEN
    Window.CLOSE

```

*Changes in function → fnObjToTbl  
Add new parameter*



Search for below 1<sup>st</sup> line and make the changes in red colored lines

```

IF NOT(COPYSTR(AvaHead."Document No.",1,3) = 'SER') THEN BEGIN
SalesOrderHdr.RESET;
//IF COPYSTR(AvaHead."Document No.",1,3) = 'QUO' THEN
//SalesOrderHdr.SETRANGE("Document Type",SalesOrderHdr."Document Type"::Quote);
//IF COPYSTR(AvaHead."Document No.",1,3) = 'ORD' THEN
// SalesOrderHdr.SETRANGE("Document Type",SalesOrderHdr."Document Type"::Order);
//IF COPYSTR(AvaHead."Document No.",1,3) = 'INV' THEN
//SalesOrderHdr.SETRANGE("Document Type",SalesOrderHdr."Document Type"::Invoice);
//IF COPYSTR(AvaHead."Document No.",1,3) = 'CRM' THEN
//SalesOrderHdr.SETRANGE("Document Type",SalesOrderHdr."Document Type"::"Credit Memo");
//IF COPYSTR(AvaHead."Document No.",1,3) = 'BLC' THEN
//SalesOrderHdr.SETRANGE("Document Type",SalesOrderHdr."Document Type"::"Blanket Order");
//IF COPYSTR(AvaHead."Document No.",1,3) = 'RET' THEN
//SalesOrderHdr.SETRANGE("Document Type",SalesOrderHdr."Document Type"::"Return Order");

CASE COPYSTR(AvaHead."Document No.",1,3) OF
'QUO' : SalesOrderHdr.SETRANGE("Document Type",SalesOrderHdr."Document Type"::Quote);
'ORD' : SalesOrderHdr.SETRANGE("Document Type",SalesOrderHdr."Document Type"::Order);
'INV' : SalesOrderHdr.SETRANGE("Document Type",SalesOrderHdr."Document Type"::Invoice);
'CRM' : SalesOrderHdr.SETRANGE("Document Type",SalesOrderHdr."Document Type"::"Credit Memo");
'BLC' : SalesOrderHdr.SETRANGE("Document Type",SalesOrderHdr."Document Type"::"Blanket Order");
'RET' : SalesOrderHdr.SETRANGE("Document Type",SalesOrderHdr."Document Type"::"Return Order");
END;

IF Avaconfig.FINDFIRST AND Avaconfig."Enable AsynchronousAvaTax Post" THEN
SalesOrderHdr.SETRANGE("No.",AvaDocumentNo)
ELSE
SalesOrderHdr.SETRANGE("No.",COPYSTR(AvaHead."Document No.",4,100));

IF SalesOrderHdr.FINDFIRST THEN BEGIN
AvaHead."Posted Date" := SalesOrderHdr."Posting Date";
AvaHead."Document Date" := AvaTaxCalcDate;
END;
END ELSE BEGIN

```

```

ServiceHeader.RESET;
// IF COPYSTR(AvaHead."Document No.",1,6) = 'SERQUO' THEN
// ServiceHeader.SETRANGE("Document Type",ServiceHeader."Document Type"::Quote);
// IF COPYSTR(AvaHead."Document No.",1,6) = 'SERORD' THEN
// ServiceHeader.SETRANGE("Document Type",ServiceHeader."Document Type"::Order);
// IF COPYSTR(AvaHead."Document No.",1,6) = 'SERINV' THEN
// ServiceHeader.SETRANGE("Document Type",ServiceHeader."Document Type"::Invoice);
// IF COPYSTR(AvaHead."Document No.",1,6) = 'SERCRM' THEN
// ServiceHeader.SETRANGE("Document Type",ServiceHeader."Document Type"::"Credit Memo");

CASE COPYSTR(AvaHead."Document No.",1,3) OF
'SERQUO' : ServiceHeader.SETRANGE("Document Type",ServiceHeader."Document Type"::Quote);
'SERORD' : ServiceHeader.SETRANGE("Document Type",ServiceHeader."Document Type"::Order);
'SERINV' : ServiceHeader.SETRANGE("Document Type",ServiceHeader."Document Type"::Invoice);
'SERCRM' : ServiceHeader.SETRANGE("Document Type",ServiceHeader."Document Type"::"Credit Memo");
END;

IF Avaconfig.FINDFIRST AND Avaconfig."Enable AsynchronousAvaTax Post" THEN
ServiceHeader.SETRANGE("No.",AvaDocumentNo)
ELSE
ServiceHeader.SETRANGE("No.",COPYSTR(AvaHead."Document No.",7,100));

IF ServiceHeader.FINDFIRST THEN BEGIN
AvaHead."Document Date" := AvaTaxCalcDate;
AvaHead."Posted Date" := ServiceHeader."Posting Date";
END;
END;

```

Search for below 1<sup>st</sup> line and make the changes in red colored lines

```

AvaLine."Tax Area" := 'AVATAX';

IF NOT(COPYSTR(AvaHead."Document No.",1,3) = 'SER') THEN BEGIN
//SalesLine.SETRANGE("Document No.",COPYSTR(AvaHead."Document No.",4,100));
SalesLine.SETRANGE("Document No.",SalesOrderHdr."No.");
SalesLine.SETFILTER(SalesLine."Line No.",oTaxLine.No);
SalesLine.SETFILTER("Document Type",DocumentType);
SalesLine.SETRANGE("Document Type",SalesOrderHdr."Document Type");
IF SalesLine.FINDFIRST THEN BEGIN
AvaLine."Tax Area" := SalesLine."Tax Area Code";
//SalesLine."Amount Including VAT" := AvaLine."Tax Tax Amount" + AvaLine."Total Taxable Amount";
IF ((AvaLine."Tax Tax Amount" + AvaLine."Total Taxable Amount" + AvaLine."Tax Exempt Amount") <> 0) THEN
SalesLine."Amount Including VAT" := AvaLine."Tax Tax Amount" + AvaLine."Total Taxable Amount" + AvaLine."Tax
Exempt Amount";
IF SalesOrderHdr."Currency Code" = "" THEN
RecCurrency.InitRoundingPrecision
ELSE BEGIN
SalesOrderHdr.TESTFIELD("Currency Factor");
RecCurrency.GET(SalesOrderHdr."Currency Code");
RecCurrency.TESTFIELD("Amount Rounding Precision");
END;
IF (SalesLine."Prepayment %" > 0) AND (AvaLine."Tax Tax Amount" > 0) THEN
SalesLine."VAT %" := AvaLine."Tax Rate" * 100;
SalesLine."Prepmt. Line Amount" := ROUND((SalesLine."Prepayment %" * SalesLine."Line Amount") /
100,RecCurrency."Amount Rounding Precision");

```

```

// PreLineAmt := AvaLine."Total Taxable Amount" + AvaLine."Tax Tax Amount" + AvaLine."Tax Exempt Amount" +
AvaLine."Tax Discount Amount";
PrepLineAmt := AvaLine."Total Taxable Amount" + AvaLine."Tax Tax Amount" + AvaLine."Tax Exempt Amount";
IF SalesOrderHdr."Prepmt. Include Tax" THEN
  SalesLine."Prepmt. Amt. Incl. VAT" := ROUND((SalesLine."Prepayment %" * PrepLineAmt) /
100,RecCurrency."Amount Rounding Precision")
ELSE
  SalesLine."Prepmt. Amt. Incl. VAT" := SalesLine."Prepmt. Line Amount";
SalesLine."Prepayment Amount" := SalesLine."Prepmt. Amt. Incl. VAT";
SalesLine.MODIFY;
END ELSE BEGIN
SalesInvLine.SETRANGE("Document No.",COPYSTR(AvaHead."Document No.",4,100));
SalesInvLine.SETFILTER("Line No.",oTaxLine.No);
IF SalesInvLine.FINDFIRST THEN
  AvaLine."Tax Area" := SalesInvLine."Tax Area Code"
ELSE BEGIN
  recObjPostedCrLine.SETRANGE("Document No.",COPYSTR(AvaHead."Document No.",4,100));
  recObjPostedCrLine.SETFILTER("Line No.",oTaxLine.No);
  IF recObjPostedCrLine.FINDFIRST THEN
    AvaLine."Tax Area" := recObjPostedCrLine."Tax Area Code"
  END
END
END ELSE BEGIN
//ServiceLine.SETRANGE("Document No.",COPYSTR(AvaHead."Document No.",7,100));
ServiceLine.SETRANGE("Document No.",ServiceHeader."No.");
ServiceLine.SETFILTER("Line No.",oTaxLine.No);
ServiceLine.SETRANGE("Document Type",ServiceHeader."Document Type");
IF ServiceLine.FINDFIRST THEN BEGIN
  AvaLine."Tax Area" := ServiceLine."Tax Area Code";
  //ServiceLine."Amount Including VAT" := AvaLine."Tax Tax Amount" + AvaLine."Total Taxable Amount";
  ServiceLine."Amount Including VAT" := AvaLine."Tax Tax Amount" + AvaLine."Total Taxable Amount" + AvaLine."Tax
Exempt Amount";
  ServiceLine.MODIFY;
END ELSE BEGIN
ServInvLine.SETRANGE("Document No.",COPYSTR(AvaHead."Document No.",7,100));
ServInvLine.SETFILTER("Line No.",oTaxLine.No);

```

### Changes in function → fnQtyAmtOrd

```

IF FORMAT(SalesLine."Line No.") = AvaLine."Tax Line No." THEN BEGIN
  AvaLine."Quantity Full":=SalesLine.Quantity;
  AvaLine."Quantity Inv.":=SalesLine."Qty. to Invoice";

  IF TDocType <> 5 THEN BEGIN
    IF (SalesLine."Qty. Shipped Not Invoiced" = 0) THEN
      IF (NOT AvalInvoiceType OR InvoiceAndShip) THEN
        AvaLine."Quantity Inv." := SalesLine."Qty. to Invoice"
      ELSE
        AvaLine."Quantity Inv." := 0;
    END ELSE BEGIN
      IF (SalesLine."Qty. Shipped Not Invoiced" = 0) THEN
        IF (AvalInvoiceType OR InvoiceAndShip) THEN
          AvaLine."Quantity Inv." := SalesLine."Qty. to Invoice"
        ELSE
          AvaLine."Quantity Inv." := 0;
      END;
    END;
  END;

```

```

IF TDocType = 5 THEN
  AvaLine."Quantity Shipped":=SalesLine."Return Qty. to Receive"
ELSE
  AvaLine."Quantity Shipped":=SalesLine."Qty. to Ship";

AvaLine."Amount Full":=ROUND(SalesLine."Line Amount"-SalesLine."Inv. Discount Amount"-AvaLine."Tax Exempt
Amount");
AvaLine."Discount Full":=ROUND(SalesLine."Inv. Discount Amount");

IF SalesLine.Quantity <> 0 THEN BEGIN
  //AvaLine."Amount Inv.":= ROUND((SalesLine."Line Amount"-SalesLine."Inv. Discount Amount"-AvaLine."Tax
Exempt Amount")/SalesLine.Quantity*SalesLine."Qty. to Invoice");
  AvaLine."Amount Inv.":= ROUND((SalesLine."Line Amount"-SalesLine."Inv. Discount Amount"-AvaLine."Tax
Exempt Amount")/SalesLine.Quantity*AvaLine."Quantity Inv.");

  IF TDocType = 5 THEN
    //AvaLine."Amount Shipped":=ROUND((SalesLine."Line Amount"-SalesLine."Inv. Discount Amount"-AvaLine."Tax
Exempt Amount")/SalesLine.Quantity*SalesLine."Qty. to Invoice")
    AvaLine."Amount Shipped":=ROUND((SalesLine."Line Amount"-SalesLine."Inv. Discount Amount"-AvaLine."Tax
Exempt Amount")/SalesLine.Quantity*AvaLine."Quantity Inv.")
  ELSE

    //AvaLine."Amount Shipped":= ROUND((SalesLine."Line Amount"-SalesLine."Inv. Discount Amount"-
AvaLine."Tax Exempt Amount")/SalesLine.Quantity*SalesLine."Qty. to Ship");
    //AvaLine."Discount Inv.":=ROUND((SalesLine."Inv. Discount Amount"/SalesLine.Quantity)*SalesLine."Qty. to
Invoice");
    AvaLine."Amount Shipped":= ROUND((SalesLine."Line Amount"-SalesLine."Inv. Discount Amount"-AvaLine."Tax Exempt
Amount")/SalesLine.Quantity*AvaLine."Quantity Inv.");
    AvaLine."Discount Inv.":=ROUND((SalesLine."Inv. Discount Amount"/SalesLine.Quantity)*AvaLine."Quantity
Inv.");

    AvaLine."Discount Shipped":=ROUND((SalesLine."Inv. Discount Amount"/SalesLine.Quantity)*SalesLine."Qty. to
Ship");

  END;
IF NOT AvaInvoiceType THEN BEGIN

```

### Changes in function → *fnGetTaxHistory*

Search for below 1<sup>st</sup> line and make the changes in red colored lines

```

IF (Result = 0) OR (Result = 1) THEN BEGIN
  TMessages:="";

  //fnObjToTbl(oGetTaxRequest,oGetTaxResult,oGetTaxResult.DocCode,DocType,DT2DATE(oGetTaxResult.DocDate),Docume
ntType);
  fnObjToTbl(oGetTaxRequest,oGetTaxResult,oGetTaxResult.DocCode,DocType,DT2DATE(oGetTaxResult.DocDate),Document
Type,AvaDocumentNo);
  AvaHead.RESET;
  IF (Result = 0) OR (Result = 1) THEN BEGIN

```

### Changes in function → *SetInvoiceFlag*

```

Avaconfig.FINDFIRST;
Avaconfig.Invoice := InvFlagStatus;
Avaconfig.MODIFY;
InvoiceNotShip := InvFlagStatus;

```

*Changes in function → fnServiceCalcTax*

## Old code

Search for below 1<sup>st</sup> line for old code and replace with new code

```

RecCurrency.TESTFIELD("Amount Rounding Precision");
END;

IF Avaconfig."Statistics Quantity" = '1' THEN BEGIN
    LineAmtPassed:=ROUND(ROUND(ServiceLine.Quantity*UnitPrice,RecCurrency."Amount Rounding Precision") -
        ROUND((ROUND(ServiceLine.Quantity * UnitPrice,RecCurrency."Amount Rounding Precision") *
ServiceLine."Line Discount %")/100,RecCurrency."Amount Rounding Precision")
        ,RecCurrency."Amount Rounding Precision",<'); //AVANA7.00.06.00
    QtyPerLine:=ServiceLine.Quantity;
END ELSE IF Avaconfig."Statistics Quantity" = '2' THEN BEGIN
    IF (ServiceLine."Document Type" = ServiceLine."Document Type" :: "Credit Memo") THEN BEGIN
        LineAmtPassed:=ROUND(ROUND(ServiceLine."Qty. to Invoice"*UnitPrice,RecCurrency."Amount Rounding
Precision") -
            ROUND((ROUND(ServiceLine."Qty. to Invoice" * UnitPrice,RecCurrency."Amount Rounding Precision")
* ServiceLine."Line Discount %")/100,RecCurrency."Amount Rounding Precision")
            ,RecCurrency."Amount Rounding Precision",<'); //AVANA7.00.06.00
        QtyPerLine:=ServiceLine."Qty. to Invoice";
    END ELSE BEGIN
        LineAmtPassed:=ROUND(ROUND(ServiceLine."Qty. to Ship"*UnitPrice,RecCurrency."Amount Rounding Precision") -
            ROUND((ROUND(ServiceLine."Qty. to Ship" * UnitPrice,RecCurrency."Amount Rounding Precision") *
ServiceLine."Line Discount %")/100,RecCurrency."Amount Rounding Precision")
            ,RecCurrency."Amount Rounding Precision",<'); //AVANA7.00.06.00
        QtyPerLine:=ServiceLine."Qty. to Ship";
    END;
END ELSE IF Avaconfig."Statistics Quantity" = '3' THEN BEGIN
    LineAmtPassed:=ROUND(ROUND(ServiceLine."Qty. to Invoice"*UnitPrice,RecCurrency."Amount Rounding Precision")
-
        ROUND((ROUND(ServiceLine."Qty. to Invoice" * UnitPrice,RecCurrency."Amount Rounding Precision") *
ServiceLine."Line Discount %")/100,RecCurrency."Amount Rounding Precision")
        ,RecCurrency."Amount Rounding Precision",<'); //AVANA7.00.06.00
    QtyPerLine:=ServiceLine."Qty. to Invoice";
END;
//AVANA7.00.06.00 end

IF TDocType=1 THEN BEGIN
    IF QtyPerLine < 0 THEN
        QtyPerLine:=QtyPerLine*-1;
    END ELSE IF TDocType=5 THEN BEGIN
        IF QtyPerLine < 0 THEN
            QtyPerLine:=QtyPerLine*-1;
        LineAmtPassed:= LineAmtPassed * -1;
    END;

```

## New Code

```

RecCurrency.TESTFIELD("Amount Rounding Precision");
END;

CASE Avaconfig."Statistics Quantity" OF
    '1' : QtyPerLine:=ServiceLine.Quantity;
    '2' : IF (ServiceLine."Document Type" = ServiceLine."Document Type" :: "Credit Memo") THEN
        QtyPerLine:=ServiceLine."Qty. to Invoice"

```

```

ELSE
  QtyPerLine:=ServiceLine."Qty. to Ship";
  '3' : QtyPerLine:=ServiceLine."Qty. to Invoice";
END;
LineAmtPassed:=ROUND(ROUND(QtyPerLine*UnitPrice,RecCurrency."Amount Rounding Precision") -
  ROUND((ROUND(QtyPerLine* UnitPrice,RecCurrency."Amount Rounding Precision") * ServiceLine."Line
Discount %")/100,RecCurrency."Amount Rounding Precision")
  ,RecCurrency."Amount Rounding Precision",'<');

IF TDocType=1 THEN BEGIN
  IF QtyPerLine < 0 THEN
    QtyPerLine:=QtyPerLine*-1;
  END ELSE IF TDocType=5 THEN BEGIN
  IF QtyPerLine < 0 THEN
    QtyPerLine:=QtyPerLine*-1;
    LineAmtPassed:= LineAmtPassed * -1;
  END;

```

Search for below 1<sup>st</sup> line and make the changes in red colored lines

```

ERROR(Text14073209 + Text14073204);
END;
END;
AvaWindowOpen(StrMessage+Text14073223);

//HdrID:=fnObjToTbl(oGetTaxRequest,oGetTaxResult,oGetTaxResult.DocCode,TDocType,DT2DATE(oGetTaxResult.DocDate)
,DocumentType);
HdrID:=fnObjToTbl(oGetTaxRequest,oGetTaxResult,oGetTaxResult.DocCode,TDocType,DT2DATE(oGetTaxResult.DocDate),D
ocumentType,AvaDocumentNo);
fnServiceQtyAmtOrd(HdrID,ServiceLine,FALSE);

```

Old code

Search for below 1<sup>st</sup> line for old code and replace with new code

```

RecCurrency.TESTFIELD("Amount Rounding Precision");
END;
IF (ServiceLine."Qty. Shipped Not Invoiced") <> 0 THEN BEGIN
  IF InvoiceFlag THEN BEGIN
    IF ServiceLine."Qty. Shipped Not Invoiced" < ServiceLine."Qty. to Invoice" THEN BEGIN
      LineAmtPassed:=ROUND(ROUND(ServiceLine."Qty. Shipped Not Invoiced"*UnitPrice,RecCurrency."Amount
Rounding Precision") -
        ROUND((ROUND(ServiceLine."Qty. Shipped Not Invoiced" * UnitPrice,RecCurrency."Amount Rounding
Precision") * ServiceLine."Line Discount %")/100,RecCurrency."Amount Rounding Precision")
        ,RecCurrency."Amount Rounding Precision",'<'); //AVANA7.00.06.00
      QtyPerLine:=ServiceLine."Qty. Shipped Not Invoiced";
    END ELSE BEGIN
      LineAmtPassed:=ROUND(ROUND(ServiceLine."Qty. to Invoice"*UnitPrice,RecCurrency."Amount Rounding
Precision") -
        ROUND((ROUND(ServiceLine."Qty. to Invoice" * UnitPrice,RecCurrency."Amount Rounding Precision")
* ServiceLine."Line Discount %")/100,RecCurrency."Amount Rounding Precision")
        ,RecCurrency."Amount Rounding Precision",'<'); //AVANA7.00.06.00
      QtyPerLine:=ServiceLine."Qty. to Invoice";
    END;
  END ELSE BEGIN
    IF ServiceLine."Qty. Shipped Not Invoiced" < ServiceLine."Qty. to Invoice" THEN BEGIN
      LineAmtPassed:=ROUND(ROUND(((ServiceLine."Qty. Shipped Not Invoiced")+(ServiceLine."Qty. to
Ship"))*UnitPrice,RecCurrency."Amount Rounding Precision") -

```

```

ROUND((ROUND((ServiceLine."Qty. Shipped Not Invoiced" + ServiceLine."Qty. to Ship") *
UnitPrice,RecCurrency."Amount Rounding Precision") * ServiceLine."Line Discount %")/100,RecCurrency."Amount Rounding
Precision")
,
RecCurrency."Amount Rounding Precision",'<'); //AVANA7.00.06.00
QtyPerLine:=(ServiceLine."Qty. Shipped Not Invoiced")+(ServiceLine."Qty. to Ship");
END ELSE BEGIN
LineAmtPassed:=ROUND(ROUND(ServiceLine."Qty. to Invoice"*UnitPrice,RecCurrency."Amount Rounding
Precision") -
ROUND((ROUND(ServiceLine."Qty. to Invoice" * UnitPrice,RecCurrency."Amount Rounding Precision")
* ServiceLine."Line Discount %")/100,RecCurrency."Amount Rounding Precision")
,RecCurrency."Amount Rounding Precision",'<'); //AVANA7.00.06.00
QtyPerLine:=ServiceLine."Qty. to Invoice";
END;
END;
END;

IF (ServiceLine."Qty. Shipped Not Invoiced" = 0) THEN BEGIN
LineAmtPassed:=ROUND(ROUND(ServiceLine."Qty. to Invoice"*UnitPrice,RecCurrency."Amount Rounding Precision")
-
ROUND((ROUND(ServiceLine."Qty. to Invoice" * UnitPrice,RecCurrency."Amount Rounding Precision") *
ServiceLine."Line Discount %")/100,RecCurrency."Amount Rounding Precision")
,RecCurrency."Amount Rounding Precision",'<'); //AVANA7.00.06.00
QtyPerLine:=ServiceLine."Qty. to Invoice";
END;

IF (ServiceLine."Document Type" = ServiceLine."Document Type" :: "Credit Memo") THEN BEGIN
LineAmtPassed:=ROUND(ROUND(ServiceLine."Qty. to Invoice"*UnitPrice,RecCurrency."Amount Rounding Precision")
-
ROUND((ROUND(ServiceLine."Qty. to Invoice" * UnitPrice,RecCurrency."Amount Rounding Precision") *
ServiceLine."Line Discount %")/100,RecCurrency."Amount Rounding Precision")
,RecCurrency."Amount Rounding Precision",'<'); //AVANA7.00.06.00
QtyPerLine:=ServiceLine."Qty. to Invoice";
END;
//AVANA7.00.06.00 end

IF TDocType=1 THEN BEGIN
IF QtyPerLine < 0 THEN
QtyPerLine:=QtyPerLine*-1;
END ELSE IF TDocType=5 THEN BEGIN
IF QtyPerLine < 0 THEN
QtyPerLine:=QtyPerLine;
LineAmtPassed:= LineAmtPassed * -1;
END;

```

## New Code

```

RecCurrency.TESTFIELD("Amount Rounding Precision");
END;

IF (ServiceLine."Qty. Shipped Not Invoiced") <> 0 THEN BEGIN
IF InvoiceFlag THEN BEGIN
IF ServiceLine."Qty. Shipped Not Invoiced" < ServiceLine."Qty. to Invoice" THEN BEGIN
QtyPerLine:=ServiceLine."Qty. Shipped Not Invoiced";
END ELSE BEGIN
QtyPerLine:=ServiceLine."Qty. to Invoice";

```



```

END;
END ELSE BEGIN
  IF ServiceLine."Qty. Shipped Not Invoiced" < ServiceLine."Qty. to Invoice" THEN BEGIN
    QtyPerLine:=(ServiceLine."Qty. Shipped Not Invoiced")+(ServiceLine."Qty. to Ship");
  END ELSE BEGIN
    QtyPerLine:=ServiceLine."Qty. to Invoice";
  END;
END;
END;

IF (ServiceLine."Qty. Shipped Not Invoiced" = 0) THEN BEGIN
  QtyPerLine:=ServiceLine."Qty. to Invoice";
END;

IF (ServiceLine."Document Type" = ServiceLine."Document Type" :: "Credit Memo") THEN BEGIN
  QtyPerLine:=ServiceLine."Qty. to Invoice";
END;

LineAmtPassed:=ROUND(ROUND(QtyPerLine*UnitPrice,RecCurrency."Amount Rounding Precision") -
  ROUND((ROUND(QtyPerLine* UnitPrice,RecCurrency."Amount Rounding Precision") * ServiceLine."Line
Discount %")/100,RecCurrency."Amount Rounding Precision")
  ,RecCurrency."Amount Rounding Precision",<');

IF TDocType=1 THEN BEGIN
  IF QtyPerLine < 0 THEN
    QtyPerLine:=QtyPerLine*-1;
  END ELSE IF TDocType=5 THEN BEGIN
    IF QtyPerLine < 0 THEN
      QtyPerLine:=QtyPerLine;
      LineAmtPassed:= LineAmtPassed * -1;
    END;
  END;

```

Search for below 1<sup>st</sup> line and make the changes in red colored lines

```

ERROR(Text14073209 + Text14073204);
END;
END;
AvaWindowOpen(StrMessage+Text14073223);
//HdrID:=fnObjToTbl(oGetTaxRequest,oGetTaxResult,oGetTaxResult.DocCode,TDocType,DT2DATE(oGetTaxResult.DocDate)
,DocumentType);
HdrID:=fnObjToTbl(oGetTaxRequest,oGetTaxResult,oGetTaxResult.DocCode,TDocType,DT2DATE(oGetTaxResult.DocDate),D
ocumentType,AvaDocumentNo);
ServiceLine.RESET;

```

### Changes in function → *fnServiceCalcTaxBatch*

Search for below 1<sup>st</sup> line for old code and replace with new code

```

RecCurrency.TESTFIELD("Amount Rounding Precision");
END;
IF NOT (SalesHead.Ship AND SalesHead.Invoice) THEN BEGIN
  IF NOT AvaReleaseStatisticsBln THEN BEGIN //CONNECT-1596
    IF Avaconfig."Statistics Quantity" = '1' THEN BEGIN
      LineAmtPassed:=ROUND(SalesLine.Quantity*UnitPrice,RecCurrency."Amount Rounding Precision") -
        ROUND((ROUND(SalesLine.Quantity * UnitPrice,RecCurrency."Amount Rounding Precision") * SalesLine."Line
Discount %")/100,RecCurrency."Amount Rounding Precision",<'); //AVANA7.00.06.00
      QtyPerLine:=SalesLine.Quantity;

```

```

END ELSE IF Avaconfig."Statistics Quantity" = '2' THEN BEGIN
  IF (SalesLine."Document Type" = SalesLine."Document Type" :: "Credit Memo") OR (SalesLine."Document Type" =
SalesLine."Document Type" :: "Return Order") THEN BEGIN
    LineAmtPassed:=ROUND(SalesLine."Return Qty. to Receive"*UnitPrice,RecCurrency."Amount Rounding Precision") -
      ROUND((ROUND(SalesLine."Return Qty. to Receive" * UnitPrice,RecCurrency."Amount Rounding Precision") *
SalesLine."Line Discount %")/100,RecCurrency."Amount Rounding Precision",'<'); //AVANA7.00.06.00
    QtyPerLine:=SalesLine."Return Qty. to Receive";
  END ELSE BEGIN
    LineAmtPassed:=ROUND(SalesLine."Qty. to Ship"*UnitPrice,RecCurrency."Amount Rounding Precision") -
      ROUND((ROUND(SalesLine."Qty. to Ship" * UnitPrice,RecCurrency."Amount Rounding Precision") *
SalesLine."Line Discount %")/100,RecCurrency."Amount Rounding Precision",'<'); //AVANA7.00.06.00
    QtyPerLine:=SalesLine."Qty. to Ship";
  END;
END ELSE IF Avaconfig."Statistics Quantity" = '3' THEN BEGIN
  LineAmtPassed:=ROUND(SalesLine."Qty. to Invoice"*UnitPrice,RecCurrency."Amount Rounding Precision") -
    ROUND((ROUND(SalesLine."Qty. to Invoice" * UnitPrice,RecCurrency."Amount Rounding Precision") *
SalesLine."Line Discount %")/100,RecCurrency."Amount Rounding Precision",'<'); //AVANA7.00.06.00
  QtyPerLine:=SalesLine."Qty. to Invoice";
  END;
END ELSE BEGIN
  IF Avaconfig."Release Quantity" = '1' THEN BEGIN
    LineAmtPassed:=ROUND(SalesLine.Quantity*UnitPrice,RecCurrency."Amount Rounding Precision") -
      ROUND((ROUND(SalesLine.Quantity * UnitPrice,RecCurrency."Amount Rounding Precision") * SalesLine."Line
Discount %")/100,RecCurrency."Amount Rounding Precision",'<'); //AVANA7.00.06.00
    QtyPerLine:=SalesLine.Quantity;
  END ELSE IF Avaconfig."Release Quantity" = '2' THEN BEGIN
    IF (SalesLine."Document Type" = SalesLine."Document Type" :: "Credit Memo") OR (SalesLine."Document Type" =
SalesLine."Document Type" :: "Return Order") THEN BEGIN
      LineAmtPassed:=ROUND(SalesLine."Return Qty. to Receive"*UnitPrice,RecCurrency."Amount Rounding Precision") -
        ROUND((ROUND(SalesLine."Return Qty. to Receive" * UnitPrice,RecCurrency."Amount Rounding Precision") *
SalesLine."Line Discount %")/100,RecCurrency."Amount Rounding Precision",'<'); //AVANA7.00.06.00
      QtyPerLine:=SalesLine."Return Qty. to Receive";
    END ELSE BEGIN
      LineAmtPassed:=ROUND(SalesLine."Qty. to Ship"*UnitPrice,RecCurrency."Amount Rounding Precision") -
        ROUND((ROUND(SalesLine."Qty. to Ship" * UnitPrice,RecCurrency."Amount Rounding Precision") *
SalesLine."Line Discount %")/100,RecCurrency."Amount Rounding Precision",'<'); //AVANA7.00.06.00
      QtyPerLine:=SalesLine."Qty. to Ship";
    END;
  END ELSE IF Avaconfig."Release Quantity" = '3' THEN BEGIN
    LineAmtPassed:=ROUND(SalesLine."Qty. to Invoice"*UnitPrice,RecCurrency."Amount Rounding Precision") -
      ROUND((ROUND(SalesLine."Qty. to Invoice" * UnitPrice,RecCurrency."Amount Rounding Precision") *
SalesLine."Line Discount %")/100,RecCurrency."Amount Rounding Precision",'<'); //AVANA7.00.06.00
    QtyPerLine:=SalesLine."Qty. to Invoice";
  END;
END;
//AVANA7.00.06.00 end

IF TDocType=1 THEN BEGIN
  IF QtyPerLine < 0 THEN
    QtyPerLine:=QtyPerLine*-1;
  END ELSE IF TDocType=5 THEN BEGIN
    IF QtyPerLine < 0 THEN
      QtyPerLine:=QtyPerLine;
    LineAmtPassed:= LineAmtPassed * -1;
  END;

```

```
END;
```

## New Code

```

RecCurrency.TESTFIELD("Amount Rounding Precision");
END;
IF NOT (SalesHead.Ship AND SalesHead.Invoice) THEN BEGIN
  IF NOT AvaReleaseStatisticsBln THEN BEGIN
    CASE Avaconfig."Statistics Quantity" OF
      '1' : QtyPerLine:=SalesLine.Quantity;
      '2' : IF (SalesLine."Document Type" = SalesLine."Document Type" :: "Credit Memo") OR (SalesLine."Document Type" =
SalesLine."Document Type" :: "Return Order") THEN
        QtyPerLine:=SalesLine."Return Qty. to Receive"
      ELSE
        QtyPerLine:=SalesLine."Qty. to Ship";
      '3' : QtyPerLine:=SalesLine."Qty. to Invoice";
    END;
  END ELSE BEGIN
    CASE Avaconfig."Release Quantity" OF
      '1' : QtyPerLine:=SalesLine.Quantity;
      '2' : IF (SalesLine."Document Type" = SalesLine."Document Type" :: "Credit Memo") OR (SalesLine."Document Type" =
SalesLine."Document Type" :: "Return Order") THEN
        QtyPerLine:=SalesLine."Return Qty. to Receive"
      ELSE
        QtyPerLine:=SalesLine."Qty. to Ship";
      '3' : QtyPerLine:=SalesLine."Qty. to Invoice";
    END;
  END;
END;

LineAmtPassed:=ROUND(QtyPerLine*UnitPrice,RecCurrency."Amount Rounding Precision") -
  ROUND((ROUND(SalesLine.Quantity * UnitPrice,RecCurrency."Amount Rounding Precision") * SalesLine."Line
Discount %")/100,RecCurrency."Amount Rounding Precision",<');

IF TDocType=1 THEN BEGIN
  IF QtyPerLine < 0 THEN
    QtyPerLine:=QtyPerLine*-1;
END ELSE IF TDocType=5 THEN BEGIN
  IF QtyPerLine < 0 THEN
    QtyPerLine:=QtyPerLine;
  LineAmtPassed:= LineAmtPassed * -1;
END;

```

Search for below 1<sup>st</sup> line and make the changes in red colored lines

```

AVABatchTaxFill(SalesHead,tResultMessage,Status);
EXIT(FALSE);
END;
END;
//HdrID:=fnObjToTbl(oGetTaxRequest,oGetTaxResult,oGetTaxResult.DocCode,TDocType,DT2DATE(oGetTaxResult.DocDate),Docume
ntType);

HdrID:=fnObjToTbl(oGetTaxRequest,oGetTaxResult,oGetTaxResult.DocCode,TDocType,DT2DATE(oGetTaxResult.DocDate),Document
Type,AvaDocumentNo);
fnQtyAmtOrd(HdrID,SalesLine,FALSE);

```

*Changes in function → fnServCalcTaxBatch*Search for below 1<sup>st</sup> line for old code and replace with new code

```

RecCurrency.TESTFIELD("Amount Rounding Precision");
END;
IF NOT AvaReleaseStatisticsBln THEN BEGIN //CONNECT-1596
  IF Avaconfig."Statistics Quantity" = '1' THEN BEGIN
    LineAmtPassed:=ROUND(ROUND(ServiceLine.Quantity*UnitPrice,RecCurrency."Amount Rounding Precision") -
      ROUND((ROUND(ServiceLine.Quantity * UnitPrice,RecCurrency."Amount Rounding Precision") *
ServiceLine."Line Discount %")/100,RecCurrency."Amount Rounding Precision"),
      RecCurrency."Amount Rounding Precision",<'); //AVANA7.00.06.00
    QtyPerLine:=ServiceLine.Quantity;
  END ELSE IF Avaconfig."Statistics Quantity" = '2' THEN BEGIN
    IF (ServiceLine."Document Type" = ServiceLine."Document Type" :: "Credit Memo") THEN BEGIN
      LineAmtPassed:=ROUND(ROUND(ServiceLine."Qty. to Invoice"*UnitPrice,RecCurrency."Amount Rounding Precision") -
        ROUND((ROUND(ServiceLine."Qty. to Invoice" * UnitPrice,RecCurrency."Amount Rounding Precision") *
ServiceLine."Line Discount %")/100,RecCurrency."Amount Rounding Precision")
          ,RecCurrency."Amount Rounding Precision",<'); //AVANA7.00.06.00
      QtyPerLine:=ServiceLine."Qty. to Invoice";
    END ELSE BEGIN
      LineAmtPassed:=ROUND(ROUND(ServiceLine."Qty. to Ship"*UnitPrice,RecCurrency."Amount Rounding Precision") -
        ROUND((ROUND(ServiceLine."Qty. to Ship" * UnitPrice,RecCurrency."Amount Rounding Precision") *
ServiceLine."Line Discount %")/100,RecCurrency."Amount Rounding Precision")
          ,RecCurrency."Amount Rounding Precision",<'); //AVANA7.00.06.00
      QtyPerLine:=ServiceLine."Qty. to Ship";
    END;
  END ELSE IF Avaconfig."Statistics Quantity" = '3' THEN BEGIN
    LineAmtPassed:=ROUND(ROUND(ServiceLine."Qty. to Invoice"*UnitPrice,RecCurrency."Amount Rounding Precision") -
      ROUND((ROUND(ServiceLine."Qty. to Invoice" * UnitPrice,RecCurrency."Amount Rounding Precision") *
ServiceLine."Line Discount %")/100,RecCurrency."Amount Rounding Precision")
        ,RecCurrency."Amount Rounding Precision",<'); //AVANA7.00.06.00
    QtyPerLine:=ServiceLine."Qty. to Invoice";
  END;

END ELSE BEGIN
  IF Avaconfig."Release Quantity" = '1' THEN BEGIN
    LineAmtPassed:=ROUND(ROUND(ServiceLine.Quantity*UnitPrice,RecCurrency."Amount Rounding Precision") -
      ROUND((ROUND(ServiceLine.Quantity * UnitPrice,RecCurrency."Amount Rounding Precision") *
ServiceLine."Line Discount %")/100,RecCurrency."Amount Rounding Precision")
        ,RecCurrency."Amount Rounding Precision",<'); //AVANA7.00.06.00
    QtyPerLine:=ServiceLine.Quantity;
  END ELSE IF Avaconfig."Release Quantity" = '2' THEN BEGIN
    IF (ServiceLine."Document Type" = ServiceLine."Document Type" :: "Credit Memo") THEN BEGIN
      LineAmtPassed:=ROUND(ROUND(ServiceLine."Qty. to Invoice"*UnitPrice,RecCurrency."Amount Rounding Precision") -
        ROUND((ROUND(ServiceLine."Qty. to Invoice" * UnitPrice,RecCurrency."Amount Rounding Precision") *
ServiceLine."Line Discount %")/100,RecCurrency."Amount Rounding Precision")
          ,RecCurrency."Amount Rounding Precision",<'); //AVANA7.00.06.00
      QtyPerLine:=ServiceLine."Qty. to Invoice";
    END ELSE BEGIN
      LineAmtPassed:=ROUND(ROUND(ServiceLine."Qty. to Ship"*UnitPrice,RecCurrency."Amount Rounding Precision") -
        ROUND((ROUND(ServiceLine."Qty. to Ship" * UnitPrice,RecCurrency."Amount Rounding Precision") *
ServiceLine."Line Discount %")/100,RecCurrency."Amount Rounding Precision")
          ,RecCurrency."Amount Rounding Precision",<'); //AVANA7.00.06.00
      QtyPerLine:=ServiceLine."Qty. to Ship";
    END;
  END ELSE IF Avaconfig."Release Quantity" = '3' THEN BEGIN

```

```

LineAmtPassed:=ROUND(ROUND(ServiceLine."Qty. to Invoice"*UnitPrice,RecCurrency."Amount Rounding Precision") -
    ROUND((ROUND(ServiceLine."Qty. to Invoice" * UnitPrice,RecCurrency."Amount Rounding Precision") *
ServiceLine."Line Discount %")/100,RecCurrency."Amount Rounding Precision")
    ,RecCurrency."Amount Rounding Precision",<'); //AVANA7.00.06.00
QtyPerLine:=ServiceLine."Qty. to Invoice";
END;
END;
//AVANA7.00.06.00 end

IF TDocType=1 THEN BEGIN
IF QtyPerLine < 0 THEN
QtyPerLine:=QtyPerLine*-1;
END ELSE IF TDocType=5 THEN BEGIN
IF QtyPerLine < 0 THEN
QtyPerLine:=QtyPerLine;
LineAmtPassed:= LineAmtPassed * -1;
END;

```

### New Code

```

RecCurrency.TESTFIELD("Amount Rounding Precision");
END;
IF NOT AvaReleaseStatisticsBln THEN BEGIN
CASE Avaconfig."Statistics Quantity" OF
'1' : QtyPerLine:=ServiceLine.Quantity;
'2' : IF (ServiceLine."Document Type" = ServiceLine."Document Type" :: "Credit Memo") THEN
QtyPerLine:=ServiceLine."Qty. to Invoice"
ELSE
QtyPerLine:=ServiceLine."Qty. to Ship";
'3' : QtyPerLine:=ServiceLine."Qty. to Invoice";
END;

END ELSE BEGIN
CASE Avaconfig."Release Quantity" OF
'1' : QtyPerLine:=ServiceLine.Quantity;
'2' : IF (ServiceLine."Document Type" = ServiceLine."Document Type" :: "Credit Memo") THEN
QtyPerLine:=ServiceLine."Qty. to Invoice"
ELSE
QtyPerLine:=ServiceLine."Qty. to Ship";
'3' : QtyPerLine:=ServiceLine."Qty. to Invoice";
END;
END;

LineAmtPassed:=ROUND(ROUND(QtyPerLine*UnitPrice,RecCurrency."Amount Rounding Precision") -
    ROUND((ROUND(QtyPerLine* UnitPrice,RecCurrency."Amount Rounding Precision") * ServiceLine."Line
Discount %")/100,RecCurrency."Amount Rounding Precision"),
    RecCurrency."Amount Rounding Precision",<');

IF TDocType=1 THEN BEGIN
IF QtyPerLine < 0 THEN
QtyPerLine:=QtyPerLine*-1;
END ELSE IF TDocType=5 THEN BEGIN
IF QtyPerLine < 0 THEN
QtyPerLine:=QtyPerLine;
LineAmtPassed:= LineAmtPassed * -1;

```

```
END;
```

Search for below 1<sup>st</sup> line and make the changes in red colored lines

```

  AVAServiceBatchTaxFill(ServiceHead,tResultMessage,Status);
  EXIT(FALSE);
  END;
  END;
  AvaWindowOpen(StrMessage+Text14073223);
  //HdrID:=fnObjToTbl(oGetTaxRequest,oGetTaxResult,oGetTaxResult.DocCode,TDocType,DT2DATE(oGetTaxResult.DocDate),DocumentType);
  HdrID:=fnObjToTbl(oGetTaxRequest,oGetTaxResult,oGetTaxResult.DocCode,TDocType,DT2DATE(oGetTaxResult.DocDate),DocumentType,AvaDocumentNo);
  fnServiceQtyAmtOrd(HdrID,ServiceLine,FALSE);
  IF NOT Avaconfig."Show Message Dialog" THEN

```

*Changes in function → AvaCheckTaxReqd*

Search for below 1<sup>st</sup> line and make the changes in red colored lines

```

AvaStatusNew:=TRUE;
Avaconfig.RESET;Avaconfig.FINDFIRST;
IF (Avaconfig."Disable Tax Calc.") THEN EXIT(FALSE);

CASE iSalesHeaderType OF
0 : IAvaHeadNo:='QUO' + iSalesHeaderNo;
1 : IAvaHeadNo:='ORD' + iSalesHeaderNo;
2 : IAvaHeadNo:='INV' + iSalesHeaderNo;
3 : IAvaHeadNo:='CRM' + iSalesHeaderNo;
4 : IAvaHeadNo:='BLC' + iSalesHeaderNo;
5 : IAvaHeadNo:='RET' + iSalesHeaderNo;
END;

//AvaAsynPosDoc(IAvaHeadNo,iSalesHeaderType,iSalesHeaderNo);

EXIT(AvaStatusNew);

```

*Changes in function → AvaAsynPosDoc*

Search for below 1<sup>st</sup> line and make the changes in red colored lines

```

DocCount := 0;
RecFound := FALSE;
AvaHead.RESET;
AvaHead.SETRANGE("AVA Document Type",DocType);
AvaHead.SETFILTER("Document No.",'%1',''+DocNo+'*');
AvaHead.SETFILTER("Status Code",'OK');
IF AvaHead.FINDLAST THEN BEGIN
  IF AvaHead."Status Code" = 'OK' THEN BEGIN
    //IF AvaHead.FINDLAST THEN
    //AvaHead.SETFILTER("Status Code",'OK');
    DocCount := DocCount + AvaHead.COUNT;
    RecFound := TRUE;
  END;
END;

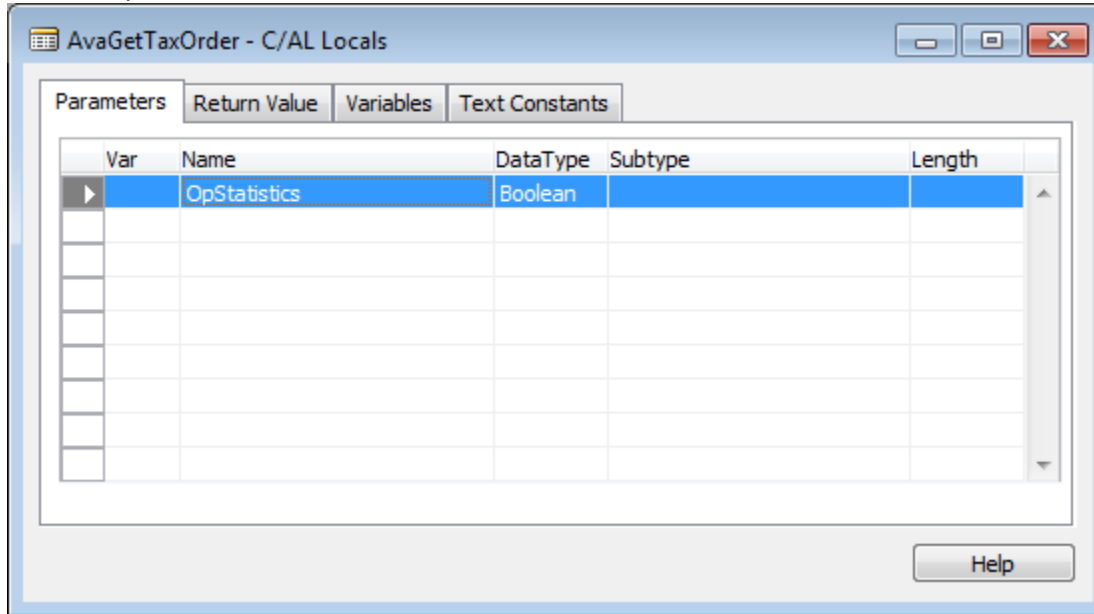
```

```
AvaHeadHst.RESET;
AvaHeadHst.SETFILTER("Document No.", '%1','*'+DocNo+'*');
IF AvaHeadHst.FINDLAST THEN BEGIN
  DocCount := DocCount + AvaHeadHst.COUNT;
  RecFound := TRUE;
END;

IF RecFound THEN
  IF DocCount >0 THEN
    PassDocNo := PassDocNo + '-' + FORMAT(DocCount)
  ELSE
    PassDocNo := AvaHead."Document No.";
```

**Codeunit 14073310 AVA Validate Call***Changes in Version List:*

Old Value	AVANA7.00.06.00
New Value	AVANA7.00.06.01

*Changes in function → AvaGetTaxOrder**Add new parameter*

```

IF NOT(AvaCheckInstStatus(0)) THEN
  EXIT(FALSE);
//IF NOT (SalesHead.Status = SalesHead.Status::Released) THEN BEGIN
IF ((NOT(SalesHead.Status = SalesHead.Status::Released) )OR OpStatistics) THEN BEGIN
  IF NOT(SALES_TO_POST) THEN BEGIN
    AvaTaxEngine.AvaSetReleaseFlag(TRUE); //AVANA7.00.06.00
    Result:=AvaTaxEngine.fnCalcTax(DocNum,DocType,AvaDocTypeTax,FALSE)
  END ELSE //AVANA7.00.06.00
    Result:=TRUE;

END ELSE BEGIN
  SalesLine.SETRANGE("Document Type",SalesHead."Document Type");
  SalesLine.SETRANGE("Document No.",SalesHead."No.");
  AvaSetDocNumTypeTable(SalesHead."No.",SalesHead."Document Type",1);
  IF NOT AvalsValidDocument2(SalesHead) THEN
    EXIT(FALSE);
  IF AvaGetTaxOrderPost THEN BEGIN
    CASE DocType OF
      0,1,2,4: BEGIN TDocType := 1; Coefficient:= 1; END;
      3,5: BEGIN TDocType := 5; Coefficient:= -1; END;
    END;
  IF SalesLine.FINDFIRST THEN BEGIN
    REPEAT
      LAvaHead.SETRANGE("Document No.",DocNumPre);
      LAvaHead.SETRANGE("AVA Document Type",TDocType); //AVANA7.00.06.00
      LAvaHead.SETRANGE("Document Type",FORMAT(SalesHead."Document Type")); //AVANA7.00.06.00
    
```



```

IF LAvaHead.FINDFIRST THEN BEGIN
  LAvaLine.SETRANGE("Tax Header No.",LAvaHead."No.");
  LAvaLine.SETRANGE("Tax Line No.",FORMAT(SalesLine."Line No."));
  IF LAvaLine.FINDFIRST THEN BEGIN
    IF AvaConfig.FINDFIRST THEN;
    IF (AvaConfig."Enable Tax Amt Adj.") AND (LAvaLine."Item No."=TextSalesTaxAdjust) THEN BEGIN
      SalesLine.Amount:=0;
      SalesLine."VAT Base Amount":=LAvaLine."Total Taxable Amount";
      IF ((LAvaLine."Tax Tax Amount" + LAvaLine."Total Taxable Amount" + LAvaLine."Tax Exempt Amount") <> 0) THEN
        //SalesLine."Amount Including VAT" := SalesLine.Amount + LAvaLine."Tax Tax Amount";
        SalesLine.VALIDATE("Amount Including VAT",LAvaLine."Tax Tax Amount" + LAvaLine."Total Taxable Amount" +
AvaLine."Tax Exempt Amount");
      SalesLine."VAT %":=0;
      SalesLine."Line Amount":=0;
      //END ELSE BEGIN
      // SalesLine.VALIDATE(Amount, SalesLine."Line Amount" - SalesLine."Inv. Discount Amount");
      // SalesLine.VALIDATE("VAT Base Amount" , SalesLine.Amount);
      // SalesLine."Amount Including VAT" := SalesLine.Amount + LAvaLine."Tax Tax Amount";
      // IF LAvaLine."Total Taxable Amount" = 0 THEN
      // SalesLine.VALIDATE("VAT %",0)
      // ELSE
      // SalesLine.VALIDATE("VAT %" , ROUND((LAvaLine."Tax Tax Amount"/LAvaLine."Total Taxable Amount")*100));
    END;
  END;
  SalesLine.MODIFY;

```

### Changes in function → OnRelease

```

IF NOT(AvaCheckInstStatus(0)) THEN
  EXIT(FALSE);

AvaConfig.RESET;
//IF AvaConfig.FINDFIRST THEN;
IF AvaConfig.FINDFIRST THEN BEGIN
  IF ((AvaConfig.Services MOD 100)/10) >= 1 THEN BEGIN
    AvaTaxEngine.AvaSetFlagStatReleaseFn;
    AvaDocLatLongSales(SalesHead1,FALSE);

    // IF SalesLines1.Type <> SalesLines1.Type::" " THEN
    // SalesLines1.SETRANGE("Document Type",SalesHead1."Document Type")
    // ELSE
    // SalesLines1.SETFILTER("Document Type", '<>%1',SalesLines1.Type::" ");
    // SalesLines1.SETRANGE("Document No.",SalesHead1."No.");
    // SalesLines1.SETFILTER(Type, '<>%1', SalesLines1.Type::" ");

    DocumentType:= FORMAT(SalesHead1."Document Type");
    AvaSetDocNumTypeTable(SalesHead1."No.",SalesHead1."Document Type",1);
    IF NOT AvalsValidDocument2(SalesHead1) THEN
      EXIT(FALSE);
    AvaDocTypeTax := 0;
    CASE DocType OF
      0,4: BEGIN AvaDocTypeTax := 0; TDocType := 1; END;
      1,2: BEGIN AvaDocTypeTax := 98; TDocType := 1; END;
      3,5: BEGIN AvaDocTypeTax := 99; TDocType := 5; END;
    END;

```

```

IF NOT AvaGetTaxOrder(FALSE) THEN //AVANA7.00.06.00
  EXIT(FALSE);
EXIT(TRUE);
END;
END;

```

### Changes in function → OnReleasePost

```

IF NOT(AvaCheckInstStatus(0)) THEN
  EXIT(FALSE);

AvaConfig.RESET;
//IF AvaConfig.FINDFIRST THEN;
IF AvaConfig.FINDFIRST THEN BEGIN

//IF SalesLines1.Type <> SalesLines1.Type::" " THEN
// SalesLines1.SETRANGE("Document Type",SalesHead1."Document Type")
//ELSE
// SalesLines1.SETFILTER("Document Type", '<>%1',SalesLines1.Type::" ");
// SalesLines1.SETRANGE("Document No.",SalesHead1."No.");
// SalesLines1.SETFILTER(Type, '<>%1', SalesLines1.Type::" ");

DocumentType:= FORMAT(SalesHead1."Document Type");
AvaSetDocNumTypeTable(SalesHead1."No.",SalesHead1."Document Type",1);
IF NOT AvalsValidDocument2(SalesHead1) THEN
  EXIT(FALSE);
IF AvaGetTaxOrderPost1 THEN BEGIN
  CASE DocType OF
    0,1,2,4: BEGIN TDocType := 1; Coefficient:= 1; END;
    3,5: BEGIN TDocType := 5; Coefficient:= -1; END;
  END;
  IF SalesLines1.FINDFIRST THEN BEGIN
    REPEAT
      LAvaHead.SETRANGE("Document No.",DocNumPre);
      LAvaHead.SETRANGE("AVA Document Type",TDocType);
      LAvaHead.SETRANGE("Document Type",FORMAT(SalesHead1."Document Type"));
      IF LAvaHead.FINDFIRST THEN BEGIN
        LAvaLine.SETRANGE("Tax Header No.",LAvaHead."No.");
        LAvaLine.SETRANGE("Tax Line No.",FORMAT(SalesLines1."Line No."));
        IF LAvaLine.FINDFIRST THEN BEGIN
          IF AvaConfig.FINDFIRST THEN;
          IF (AvaConfig."Enable Tax Amt Adj.") AND (SalesLines1."No." = TextSalesTaxAdjust) THEN BEGIN
            SalesLines1.Amount := 0;
            SalesLines1."VAT Base Amount":=SalesLines1.Amount;
            IF SalesLines1."Line Amount" <>0 THEN
              SalesLines1."Amount Including VAT":=LAvaLine."Tax Tax Amount";
            SalesLines1."VAT %":=0;
            SalesLines1."Line Amount":=0;
            SalesLines1."Inv. Discount Amount":=0;
          END ELSE BEGIN
            SalesLines1.Amount := SalesLines1."Line Amount" - SalesLines1."Inv. Discount Amount";
            SalesLines1.VALIDATE("VAT Base Amount",SalesLines1.Amount);
            SalesLines1.VALIDATE("Amount Including VAT", SalesLines1."Line Amount" + LAvaLine."Tax Tax Amount");
            IF LAvaLine."Total Taxable Amount" = 0 THEN
              SalesLines1."VAT %":=0
            ELSE

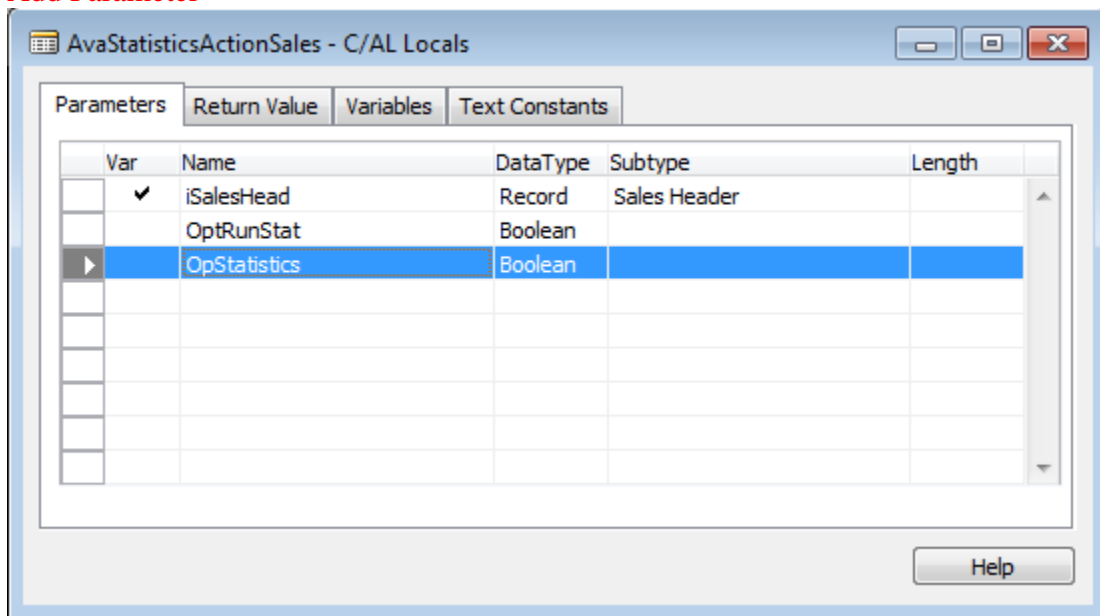
```

```

SalesLines1.VALIDATE("VAT %", ROUND((LAVAline."Tax Tax Amount"/LAVAline."Total Taxable Amount")*100));
END;
SalesLines1.MODIFY;
END;
END;
UNTIL SalesLines1.NEXT = 0;
END ELSE
EXIT(FALSE);
END ELSE
EXIT(FALSE);
EXIT(TRUE);
END;
    
```

*Changes in function → AvaStatisticsActionSales*

**Add Parameter**



```

AvaDocLatLongSales(iSalesHead,FALSE); //AVANA7.00.06.00
IF AvalsValidDocument(1) THEN BEGIN
SetPostingFlag(FALSE);
AvaSetReleaseFlag(FALSE);
//AvaGetTaxOrder;
AvaGetTaxOrder(OpStatistics);
COMMIT;
IF OptRunStat THEN //AVANA7.00.06.00
AvaCallStat;
    
```

**Changes in Avalara Pages**

**Page 14073213 AVA Mass Addr. Loc. Validation**

*Changes in Version List:*

Old Value	AVANA7.00.06.00
New Value	AVANA7.00.06.01

*Changes in Property → PageType*

Old Value	ListPart
New Value	CardPart

**Page 14073351 AVA Account Credentials***Changes in Version List:*

Old Value	AVANA7.00.06.00
New Value	AVANA7.00.06.01

*Changes in Global Text Constant:*

New Value	'ENU=NAV 7.00    7.00.33781.06.01'
-----------	------------------------------------

**Page 14073400 AVA Configuration Card***Changes in Version List:*

Old Value	AVANA7.00.06.00
New Value	AVANA7.00.06.01

*Changes in Global Text Constant:*

Text14073212	:	TextConst	'ENU=NAV 7.00.33781.06.00'
Text14073212	:	TextConst	'ENU=NAV 7.00    7.00.33781.06.01'

*Changes in trigger → OnAfterGetRecord**Modified Code (Search for 1st line / modified code marked in Rec color)*

```

//AvalInitiation;
AvaTaxEnableDisable;
AvaAddressEnableDisable;
//Validate Return Address Type
IF "Return Address" = '1' THEN
    AvaRetAddr := AvaRetAddr::Customer
ELSE IF "Return Address" = '2' THEN
    AvaRetAddr := AvaRetAddr::Invoice
ELSE IF "Return Address" = '3' THEN
    AvaRetAddr := AvaRetAddr::Shipping;

//Validate("Statistics Quantity");
IF "Statistics Quantity" = '1' THEN
    AvaQtyStats := AvaQtyStats::"Quantity (Order)"
ELSE IF "Statistics Quantity" = '2' THEN
    AvaQtyStats := AvaQtyStats::"Quantity to ship"
ELSE IF "Statistics Quantity" = '3' THEN
    AvaQtyStats := AvaQtyStats::"Quantity to invoice";

//Validate("Release Quantity");
IF "Release Quantity" = '1' THEN
    AvaQtyRelease := AvaQtyRelease::"Quantity (Order)"
ELSE IF "Release Quantity" = '2' THEN
    AvaQtyRelease := AvaQtyRelease::"Quantity to ship"
ELSE IF "Release Quantity" = '3' THEN
    AvaQtyRelease := AvaQtyRelease::"Quantity to invoice";

EntityUseCodeEnable := "Entity\Use";

```

*Changes in Field Trigger → chkEntityUseCode → OnValidate  
Modified Code (Search for 1st line / modified code marked in Rec color)*

```
//IF "Entity\Use" THEN
// EntityUseCodeEnable :=TRUE
// ELSE
// EntityUseCodeEnable :=FALSE;
EntityUseCodeEnable :="Entity\Use";
```

*Changes in function → AvaInitiation  
Modified Code (Search for 1st line / modified code marked in Rec color)*

```
IF NOT ISEMPTY THEN BEGIN
  AvaEnableCredentials(FALSE);//Changes

  //IF "Entity\Use" THEN
  // EntityUseCodeEnable :=TRUE
  // ELSE
  // EntityUseCodeEnable :=FALSE;

  EntityUseCodeEnable := "Entity\Use";

  "Connector Version":=Text14073211;
  IF "Tax Mapping" = "Tax Mapping"::"AvaTax Table" THEN
    TaxMappingVisible :=TRUE
  ELSE
    TaxMappingVisible :=FALSE;
  AvaTaxEnableDisable;
  AvaAddressEnableDisable;
END ELSE BEGIN
  AvaEnableCredentials(TRUE);//Changes
END;
```

**Page 14073520 AVA Cust. Entity Class**

*Changes in Version List:*

Old Value	AVANA7.00.06.00
New Value	AVANA7.00.06.01

*Changes in Property → PageType*

Old Value	Card
New Value	List

**Page 14073522 AVA Tax Update Utility**

*Changes in Version List:*

Old Value	AVANA7.00.06.00
New Value	AVANA7.00.06.01

*Changes in trigger → OnInit*

```
ItemRangeVisible := TRUE;
VendorRangeVisible := TRUE;
CustomerRangeVisible := TRUE;
CtrSleep := 100;
```

*Changes in Page Action → cmdUpdate**Changes in property*

Property Name	Old Value	New Value
Image	UpdateDescription	SetupList
ShortCutKey		Ctrl+T

**Old Code**

```

IF TaxGroupCode = " THEN
BEGIN
  MESSAGE ('Please enter Tax Group Code');
END ELSE
BEGIN
  CustomerRange:=SelectionRange;
  VendorRange:=SelectionRange;

  IF OpContentType = 0 THEN BEGIN
    UpdateCustomer;
    MESSAGE('Tax Area Code is updated for Customer/s %1', CustomerRange);
  END;
  IF OpContentType = 1 THEN BEGIN
    UpdateVendor;
    MESSAGE('Tax Area Code is updated for Vendor/s %1', VendorRange);
  END;

  END;
  TaxGroupCode:="";
  ItemRange:="";
  CustomerRange:="";
  VendorRange:="";
  CLEAR(SelectionRange);

```

**New Code**

```

IF TaxGroupCode = " THEN
  MESSAGE ('Please enter Tax Area Code')
ELSE BEGIN
  CASE OpContentType OF
    0 : BEGIN
      UpdateCustomer;
      UpdateResult := 'Tax Area Code is updated for Customer/s '+ SelectionRange;
    END;
    1 : BEGIN
      UpdateVendor;
      UpdateResult := 'Tax Area Code is updated for Vendor/s '+ SelectionRange;
    END;
  END;
  MESSAGE(UpdateResult);
END;
TaxGroupCode:="";
SelectionRange:="";
UpdateResult := "";
CurrPage.UPDATE;

```

*Changes in Field Trigger → optContentType → OnValidate*

```

IF OpContentType = 0 THEN

```

```

BEGIN
CustomerRangeVisible := TRUE;
//VendorRangeVisible := FALSE; VendorRange := "";
//ItemRangeVisible := FALSE; ItemRange := "";
VendorRangeVisible := FALSE; SelectionRange := "";
ItemRangeVisible := FALSE;
END ELSE
IF OpContentType = 1 THEN
BEGIN
VendorRangeVisible := TRUE;
//CustomerRangeVisible := FALSE; CustomerRange := "";
//ItemRangeVisible := FALSE; ItemRange := "";
CustomerRangeVisible := FALSE; SelectionRange := "";
ItemRangeVisible := FALSE;
END ELSE
IF OpContentType = 2 THEN
BEGIN
ItemRangeVisible := TRUE;
//VendorRangeVisible := FALSE; VendorRange := "";
//CustomerRangeVisible := FALSE; CustomerRange := "";
VendorRangeVisible := FALSE; SelectionRange := "";
CustomerRangeVisible := FALSE;
END;

IF NOT(AvaCurrentSelection=OpContentType) THEN BEGIN
SelectionRange:="";
AvaCurrentSelection:=OpContentType;
END;
    
```

*Delete below fields from Page:*

Type	SubType	SourceExpr	Name	Caption
Field		CustomerRange	cCustomerRange	Customer
Field		VendorRange	cVendorRange	Vendor
Field		ItemRange	cItemRange	Item

*Delete Global variables:*

Name	Data Type	Subtype	Length
CustRange	Text		250
VendRange	Text		30
ItemRange	Text		30
CustomerRange	Text		250
VendorRange	Text		250

*Add new Global variables:*

Name	Data Type	Subtype	Length
UpdateResult	Text		
CtrSleep	Integer		

*Change in function → UpdateCustomer*

Old Code

```

WITH Cust DO BEGIN
    
```

```

SETFILTER("No.",CustomerRange);
IF COUNT > 0 THEN BEGIN
    FIND('-');
    REPEAT
Window.OPEN('Updating Customer #1##### with Tax Area Code #2#####',"No.",TaxGroupCode);
"Tax Area Code" := TaxGroupCode;
MODIFY;
Window.CLOSE;
recCustShipTo.SETRANGE(recCustShipTo."Customer No.",Cust."No.");
IF recCustShipTo.FIND('-') THEN BEGIN
    REPEAT
        IF recCustShipTo.Code <> " THEN BEGIN
Window.OPEN('Updating Customer #1##### with Tax Area Code #2#####',"No.",TaxGroupCode);
recCustShipTo."Tax Area Code" := TaxGroupCode;
recCustShipTo.MODIFY;
Window.CLOSE;
            END;
        UNTIL recCustShipTo.NEXT = 0;
    END
    UNTIL NEXT = 0;
END;
END;

```

## New Code

```

WITH Cust DO BEGIN
SETFILTER("No.",SelectionRange);
IF FINDFIRST THEN BEGIN
    Window.OPEN('Updating Customer #1##### with Tax Area Code #2#####',"No.",TaxGroupCode);
    REPEAT
        Window.UPDATE;
        SLEEP(CtrSleep);
        "Tax Area Code" := TaxGroupCode;
        MODIFY;
        recCustShipTo.SETRANGE(recCustShipTo."Customer No.",Cust."No.");
        IF recCustShipTo.FINDFIRST THEN BEGIN
            REPEAT
                IF recCustShipTo.Code <> " THEN BEGIN
                    Window.UPDATE;
                    SLEEP(CtrSleep);
                    recCustShipTo."Tax Area Code" := TaxGroupCode;
                    recCustShipTo.MODIFY;
                END;
            UNTIL recCustShipTo.NEXT = 0;
        END
    UNTIL NEXT = 0;
    Window.CLOSE;
END;
END;

```

## Change in function → UpdateVendor

### Old Code

```

WITH Vend DO BEGIN
SETFILTER("No.",VendorRange);
IF COUNT > 0 THEN BEGIN

```



```

FIND('-');
REPEAT
  Window.OPEN('Updating Vendor #1##### with Tax Area Code #2#####',"No.",TaxGroupCode);
  "Tax Area Code" := TaxGroupCode;
  MODIFY;
  Window.CLOSE;
UNTIL NEXT = 0;
END;
END;

```

### New Code

```

  WITH Vend DO BEGIN
  SETFILTER("No.",SelectionRange);
  IF FINDFIRST THEN BEGIN

  Window.OPEN('Updating Vendor #1##### with Tax Area Code #2#####',"No.",TaxGroupCode);
  REPEAT
    Window.UPDATE;
    SLEEP(CtrSleep);
    "Tax Area Code" := TaxGroupCode;
    MODIFY;
  UNTIL NEXT = 0;
  Window.CLOSE;
  END;
  END;

```

### Change in function → UpdateItem

#### Old Code

```

WITH RecItem DO BEGIN
SETFILTER("No.",ItemRange);
IF COUNT > 0 THEN BEGIN
  FIND('-');
  REPEAT
    Window.OPEN('Updating Item #1##### with Tax Area Code #2#####',"No.",TaxGroupCode);
    SLEEP(1000);
    "Tax Group Code" := TaxGroupCode;
    MODIFY;
    Window.CLOSE;
  UNTIL NEXT = 0;
END;
END;

```

### New Code

```

  WITH RecItem DO BEGIN
  SETFILTER("No.",SelectionRange);
  IF FINDFIRST THEN BEGIN
  REPEAT
    Window.UPDATE;
    SLEEP(CtrSleep);
    "Tax Group Code" := TaxGroupCode;
    MODIFY;
  UNTIL NEXT = 0;
  Window.CLOSE;
  END;

```

```
END;
```

*Change in function → cCustomerRangeOnAfterValidate*

```
//Cust.SETFILTER(Cust."No.",CustomerRange);  
Cust.SETFILTER(Cust."No.",SelectionRange);
```

*Change in function → cVendorRangeOnAfterValidate*

```
//Vend.SETFILTER(Vend."No.",VendorRange);  
Vend.SETFILTER(Vend."No.",SelectionRange);
```

*Change in function → cItemRangeOnAfterValidate*

```
//RecItem.SETFILTER(RecItem."No.",ItemRange);  
RecItem.SETFILTER(RecItem."No.",SelectionRange);
```

## Changes in AvalaraX Pages

### Page 41 Sales Quote

#### Changes in Version List:

Old Value	NAVW17.00,NAVNA7.00,AVANA7.00.06.00
New Value	NAVW17.00,NAVNA7.00,AVANA7.00.06.01

#### Changes in Page Action → Statistics@OnAction

##### Add Code (Search for below first line and add colored line)

```

CalcInvDiscForHeader;
COMMIT;

IF "Tax Area Code" = " THEN
  PAGE.RUNMODAL(PAGE::"Sales Statistics",Rec)
ELSE BEGIN //Avalara
  //AvaValidCall.AvaStatisticsActionSales(Rec,TRUE);//Avalara
  AvaValidCall.AvaStatisticsActionSales(Rec,TRUE,TRUE);//Avalara
  PAGE.RUNMODAL(PAGE::"Sales Stats.",Rec);
END; //Avalara

```

#### Changes in Page Action → Print@OnAction

##### Add Code (Search for below first line and add colored line)

```

//AvaValidCall.AvaStatisticsActionSales(Rec,FALSE);//Avalara
AvaValidCall.AvaStatisticsActionSales(Rec,FALSE,FALSE);//Avalara
DocPrint.PrintSalesHeader(Rec);

```

### Page 42 Sales Order

#### Changes in Version List:

Old Value	NAVW17.00,NAVNA7.00,AVANA7.00.06.00
New Value	NAVW17.00,NAVNA7.00,AVANA7.00.06.01

#### Changes in Page Action → Statistics@OnAction

##### Add Code (Search for below first line and add colored line)

```

CalcInvDiscForHeader;
COMMIT;

IF "Tax Area Code" = " THEN
  PAGE.RUNMODAL(PAGE::"Sales Order Statistics",Rec)
ELSE BEGIN
  //AvaValidCall.AvaStatisticsActionSales(Rec,TRUE);//Avalara
  AvaValidCall.AvaStatisticsActionSales(Rec,TRUE,TRUE);//Avalara
  PAGE.RUNMODAL(PAGE::"Sales Order Stats.",Rec);

```

#### Changes in Page Action Added → Test Report

##### Add Code (Search for below first line and add colored line)

```

//AvaValidCall.AvaStatisticsActionSales(Rec,FALSE);//AVANA7.00.06.00
AvaValidCall.AvaStatisticsActionSales(Rec,FALSE,FALSE);
ReportPrint.PrintSalesHeader(Rec);

```

*Changes in Page Action Added → Prepayment & Test Report**Add Code (Search for below first line and add colored line)*

```
//AvaValidCall.AvaStatisticsActionSales(Rec,FALSE);//AVANA7.00.06.00  
AvaValidCall.AvaStatisticsActionSales(Rec,FALSE,FALSE);  
ReportPrint.PrintSalesHeaderPrepmt(Rec);
```

*Changes in Page Action Added → Post Prepayment & Invoice**Add Code (Search for below first line and add colored line)*

```
//AvaValidCall.AvaStatisticsActionSales(Rec,FALSE);//AVANA7.00.06.00  
AvaValidCall.AvaStatisticsActionSales(Rec,FALSE,FALSE);  
IF ApprovalMgt.PrePostApprovalCheck(Rec,PurchaseHeader) THEN  
SalesPostYNPrepmt.PostPrepmtInvoiceYN(Rec,FALSE);
```

*Changes in Page Action Added → Print Confirmation**Add Code (Search for below first line and add colored line)*

```
//AvaValidCall.AvaStatisticsActionSales(Rec,FALSE);//AVANA7.00.06.00  
AvaValidCall.AvaStatisticsActionSales(Rec,FALSE,FALSE);  
DocPrint.PrintSalesOrder(Rec,Usage::"Order Confirmation");
```

## Page 43 Sales Invoice

### Changes in Version List:

Old Value	NAVW17.00,NAVNA7.00,AVANA7.00.06.00
New Value	NAVW17.00,NAVNA7.00,AVANA7.00.06.01

### Change to trigger → Statistics@OnAction

#### Add Code (Search for below first line and add colored line)

```

CalcInvDiscForHeader;
COMMIT;

IF "Tax Area Code" = " THEN
  PAGE.RUNMODAL(PAGE::"Sales Statistics",Rec)
ELSE BEGIN
  //AvaValidCall.AvaStatisticsActionSales(Rec,TRUE);//Avalara
  AvaValidCall.AvaStatisticsActionSales(Rec,TRUE,TRUE);
  PAGE.RUNMODAL(PAGE::"Sales Order Stats.",Rec)

```

### Changes in Page Action Added → Test Report

#### Add Code (Search for below first line and add colored line)

```

//AvaValidCall.AvaStatisticsActionSales(Rec,FALSE);//Avalara
AvaValidCall.AvaStatisticsActionSales(Rec,FALSE,FALSE);
ReportPrint.PrintSalesHeader(Rec);

```

## Page 44 Sales Credit Memo

### Changes in Version List:

Old Value	NAVW17.00,NAVNA7.00,AVANA7.00.06.00
New Value	NAVW17.00,NAVNA7.00,AVANA7.00.06.01

### Change to trigger → Statistics@OnAction

Add Code (Search for below first line and add colored line)

```

CalcInvDiscForHeader;
COMMIT;

IF "Tax Area Code" = " THEN
  PAGE.RUNMODAL(PAGE::"Sales Statistics",Rec)
ELSE BEGIN
  AvaValidCall.AvaCheckSalesTaxAdjust("No.,"Document Type");//Avalara
  //AvaValidCall.AvaStatisticsActionSales(Rec,TRUE);//Avalara
  AvaValidCall.AvaStatisticsActionSales(Rec,TRUE,TRUE);
  PAGE.RUNMODAL(PAGE::"Sales Order Stats.",Rec);
  IF AvaValidCall.AvaAdjustSalesLines("Document Type", "No.") THEN//Avalara
    CurrPage.UPDATE;//Avalara
  END;

```

### Changes in Page Action Added → Test Report

Add Code (Search for below first line and add colored line)

```

AvaValidCall.AvaCheckSalesTaxAdjust("No.,"Document Type");//Avalara
//AvaValidCall.AvaStatisticsActionSales(Rec,FALSE);//Avalara
AvaValidCall.AvaStatisticsActionSales(Rec, FALSE, FALSE);
ReportPrint.PrintSalesHeader(Rec);

```

## Page 507 Blanket Sales Order

### Changes in Version List:

Old Value	NAVW17.00,NAVNA7.00,AVANA7.00.06.00
New Value	NAVW17.00,NAVNA7.00,AVANA7.00.06.01

### Change to trigger → Statistics@OnAction

Add Code (Search for below first line and add colored line)

```

CalcInvDiscForHeader;
COMMIT;

IF "Tax Area Code" = " THEN
  PAGE.RUNMODAL(PAGE::"Sales Order Statistics",Rec)
ELSE BEGIN
  //AvaValidCall.AvaStatisticsActionSales(Rec,TRUE);//Avalara
  AvaValidCall.AvaStatisticsActionSales(Rec,TRUE,TRUE);//Avalara
  PAGE.RUNMODAL(PAGE::"Sales Order Stats.",Rec)
END;

```

### Changes in Page Action Added → Print

Add Code (Search for below first line and add colored line)

```

//AvaValidCall.AvaStatisticsActionSales(Rec,FALSE);//Avalara
AvaValidCall.AvaStatisticsActionSales(Rec,FALSE,FALSE);//Avalara
DocPrint.PrintSalesHeader(Rec);

```

## Page 6630 Sales Return Order

### Changes in Version List:

Old Value	NAVW17.00,NAVNA7.00,AVANA7.00.06.00
New Value	NAVW17.00,NAVNA7.00,AVANA7.00.06.01

### Change to trigger → Statistics@OnAction

#### Add Code (Search for below first line and add colored line)

```

CalcInvDiscForHeader;
COMMIT;

IF "Tax Area Code" = " THEN
  PAGE.RUNMODAL(PAGE::"Sales Order Statistics",Rec)
ELSE BEGIN
  AvaValidCall.AvaCheckSalesTaxAdjust("No.", "Document Type");//Avalara
  //AvaValidCall.AvaStatisticsActionSales(Rec,TRUE);//Avalara
  AvaValidCall.AvaStatisticsActionSales(Rec,TRUE,TRUE);//Avalara
  PAGE.RUNMODAL(PAGE::"Sales Order Stats.",Rec);

```

### Changes in Page Action → Test Report

#### Add Code (Search for below first line and add colored line)

```

AvaValidCall.AvaCheckSalesTaxAdjust("No.", "Document Type");//Avalara
//AvaValidCall.AvaStatisticsActionSales(Rec,FALSE);//Avalara
AvaValidCall.AvaStatisticsActionSales(Rec,FALSE,FALSE);//Avalara
ReportPrint.PrintSalesHeader(Rec);

```

### Changes in Page Action → Print

#### Add Code (Search for below first line and add colored line)

```

AvaValidCall.AvaCheckSalesTaxAdjust("No.", "Document Type");//Avalara
//AvaValidCall.AvaStatisticsActionSales(Rec,FALSE);//Avalara
AvaValidCall.AvaStatisticsActionSales(Rec,FALSE,FALSE);//Avalara
DocPrint.PrintSalesHeader(Rec);

```

## Page 9300 Sales Quotes

### Changes in Version List:

Old Value	NAVW17.00,NAVNA7.00,AVANA7.00.06.00
New Value	NAVW17.00,NAVNA7.00,AVANA7.00.06.01

### Change to trigger → Statistics@OnAction

#### Add Code (Search for below first line and add colored line)

```

CalcInvDiscForHeader;
COMMIT;
// NA0001.begin
IF "Tax Area Code" = " THEN
  PAGE.RUNMODAL(PAGE::"Sales Statistics",Rec)
ELSE BEGIN //Avalara
  //AvaValidCall.AvaStatisticsActionSales(Rec,TRUE);//Avalara
  AvaValidCall.AvaStatisticsActionSales(Rec,TRUE,TRUE);//Avalara

```



```
PAGE.RUNMODAL(PAGE::"Sales Stats.",Rec);  
END; //Avalara  
// NA0001.end
```

*Changes in Page Action → Print*

*Add Code (Search for below first line and add colored line)*

```
//AvaValidCall.AvaStatisticsActionSales(Rec,FALSE);//Avalara  
AvaValidCall.AvaStatisticsActionSales(Rec,FALSE,FALSE);//Avalara  
DocPrint.PrintSalesHeader(Rec);
```

## Page 9301 Sales Invoice List

### Changes in Version List:

Old Value	NAVW17.00,NAVNA7.00,AVANA7.00.06.00
New Value	NAVW17.00,NAVNA7.00,AVANA7.00.06.01

### Change to trigger → Statistics@OnAction

#### Add Code (Search for below first line and add colored line)

```

CalcInvDiscForHeader;
COMMIT;
// NA0001.begin
//PAGE.RUNMODAL(PAGE::"Sales Statistics",Rec);
IF "Tax Area Code" = " THEN
    PAGE.RUNMODAL(PAGE::"Sales Statistics",Rec)
ELSE BEGIN //Avalara
    // AvaValidCall.AvaStatisticsActionSales(Rec,TRUE);//Avalara
    AvaValidCall.AvaStatisticsActionSales(Rec,TRUE,TRUE);//Avalara
    PAGE.RUNMODAL(PAGE::"Sales Order Stats.",Rec)
END; //Avalara
// NA0001.end
  
```

### Changes in Page Action → TestReport

#### Add Code (Search for below first line and add colored line)

```

//AvaValidCall.AvaStatisticsActionSales(Rec,FALSE);//Avalara
AvaValidCall.AvaStatisticsActionSales(Rec,FALSE,FALSE);//Avalara
ReportPrint.PrintSalesHeader(Rec);
  
```

## Page 9302 Sales Credit Memos

### Changes in Version List:

Old Value	NAVW17.00,NAVNA7.00,AVANA7.00.06.00
New Value	NAVW17.00,NAVNA7.00,AVANA7.00.06.01

### Change to trigger → Statistics@OnAction

#### Add Code (Search for below first line and add colored line)

```

CalcInvDiscForHeader;
COMMIT;
// NA0001.begin
//PAGE.RUNMODAL(PAGE::"Sales Statistics",Rec);
IF "Tax Area Code" = " THEN
    PAGE.RUNMODAL(PAGE::"Sales Statistics",Rec)
ELSE BEGIN //Avalara
    AvaValidCall.AvaCheckSalesTaxAdjust("No.,"Document Type");//Avalara
    //AvaValidCall.AvaStatisticsActionSales(Rec,TRUE);//Avalara
    AvaValidCall.AvaStatisticsActionSales(Rec,TRUE,TRUE);//Avalara
    PAGE.RUNMODAL(PAGE::"Sales Order Stats.",Rec);
    IF AvaValidCall.AvaAdjustSalesLines("Document Type", "No.") THEN//Avalara
        CurrPage.UPDATE;//Avalara
END; //Avalara
  
```

```
// NA0001.end
```

*Changes in Page Action Added → Test Report;*

*Add Code (Search for below first line and add colored line)*

```
AvaValidCall.AvaCheckSalesTaxAdjust("No.,"Document Type");//Avalara  
//AvaValidCall.AvaStatisticsActionSales(Rec,FALSE);//Avalara  
AvaValidCall.AvaStatisticsActionSales(Rec,FALSE,FALSE);//Avalara  
ReportPrint.PrintSalesHeader(Rec);
```

## Page 9303 Blanket Sales Orders

### Changes in Version List:

Old Value	NAVW17.00,NAVNA7.00,AVANA7.00.06.00
New Value	NAVW17.00,NAVNA7.00,AVANA7.00.06.01

### Change to trigger → Statistics@OnAction

Add Code (Search for below first line and add colored line)

```

CalcInvDiscForHeader;
COMMIT;

IF "Tax Area Code" = " THEN
  PAGE.RUNMODAL(PAGE::"Sales Order Statistics",Rec)
ELSE BEGIN
  //AvaValidCall.AvaStatisticsActionSales(Rec,TRUE);//Avalara
  AvaValidCall.AvaStatisticsActionSales(Rec,TRUE,TRUE);//Avalara
  PAGE.RUNMODAL(PAGE::"Sales Order Stats.",Rec)
END;

```

### Changes in Page Action Added → Print;

Add Code (Search for below first line and add colored line)

```

//AvaValidCall.AvaStatisticsActionSales(Rec,FALSE);//Avalara
AvaValidCall.AvaStatisticsActionSales(Rec,FALSE,FALSE);//Avalara
DocPrint.PrintSalesHeader(Rec);

```

## Page 9304 Sales Return Order List

### Changes in Version List:

Old Value	NAVW17.00,NAVNA7.00,AVANA7.00.06.00
New Value	NAVW17.00,NAVNA7.00,AVANA7.00.06.01

### Change to trigger → Statistics@OnAction

#### Add Code (Search for below first line and add colored line)

```

CalcInvDiscForHeader;
COMMIT;

IF "Tax Area Code" = " THEN
  PAGE.RUNMODAL(PAGE::"Sales Order Statistics",Rec)
ELSE BEGIN
  AvaValidCall.AvaCheckSalesTaxAdjust("No.", "Document Type");//Avalara
  //AvaValidCall.AvaStatisticsActionSales(Rec,TRUE);//Avalara
  AvaValidCall.AvaStatisticsActionSales(Rec,TRUE,TRUE);//Avalara
  PAGE.RUNMODAL(PAGE::"Sales Order Stats.",Rec);
END

```

### Changes in Page Action Added → Print;

#### Add Code (Search for below first line and add colored line)

```

AvaValidCall.AvaCheckSalesTaxAdjust("No.", "Document Type");//Avalara
//AvaValidCall.AvaStatisticsActionSales(Rec,FALSE);//Avalara
AvaValidCall.AvaStatisticsActionSales(Rec,FALSE,FALSE);//Avalara
DocPrint.PrintSalesHeader(Rec);

```

### Changes in Page Action Added → TestReport;

#### Add Code (Search for below first line and add colored line)

```

AvaValidCall.AvaCheckSalesTaxAdjust("No.", "Document Type");//Avalara
//AvaValidCall.AvaStatisticsActionSales(Rec,FALSE);//Avalara
AvaValidCall.AvaStatisticsActionSales(Rec,FALSE,FALSE);//Avalara
ReportPrint.PrintSalesHeader(Rec);

```

## Page 9305 Sales Order List

### Changes in Version List:

Old Value	NAVW17.00,NAVNA7.00,AVANA7.00.06.00
New Value	NAVW17.00,NAVNA7.00,AVANA7.00.06.01

### Change to trigger → Statistics@OnAction

Add Code (Search for below first line and add colored line)

```

CalcInvDiscForHeader;
COMMIT;
// NA0001.begin
//PAGE.RUNMODAL(PAGE::"Sales Order Statistics",Rec);
IF "Tax Area Code" = " THEN
    PAGE.RUNMODAL(PAGE::"Sales Order Statistics",Rec)
ELSE BEGIN
    //AvaValidCall.AvaStatisticsActionSales(Rec,TRUE);//Avalara
    AvaValidCall.AvaStatisticsActionSales(Rec,TRUE,TRUE);//Avalara
    PAGE.RUNMODAL(PAGE::"Sales Order Stats.",Rec)
END;
// NA0001.end
  
```

### Changes in Page Action Added → Test Report;

Add Code (Search for below first line and add colored line)

```

//AvaValidCall.AvaStatisticsActionSales(Rec,FALSE);//Avalara
AvaValidCall.AvaStatisticsActionSales(Rec,FALSE,FALSE);//Avalara
ReportPrint.PrintSalesHeader(Rec);
  
```

### Changes in Page Action Added → Order Confirmation;

Add Code (Search for below first line and add colored line)

```

//AvaValidCall.AvaStatisticsActionSales(Rec,FALSE);//Avalara
AvaValidCall.AvaStatisticsActionSales(Rec,FALSE,FALSE);//Avalara
DocPrint.PrintSalesOrder(Rec,Usage::"Order Confirmation");
  
```

### Changes in Page Action Added → Print;

Add Code (Search for below first line and add colored line)

```

//AvaValidCall.AvaStatisticsActionSales(Rec,FALSE);//Avalara
AvaValidCall.AvaStatisticsActionSales(Rec,FALSE,FALSE);//Avalara
DocPrint.PrintSalesOrder(Rec,Usage::"Work Order");
  
```

## Page 10026 Sales Order Shipment

### Changes in Version List:

Old Value	NAVNA7.00,AVANA7.00.06.00
New Value	NAVNA7.00,AVANA7.00.06.01

*Change to trigger → Statistics@OnAction*

*Add Code (Search for below first line and add colored line)*

```
SalesSetup.GET;
IF SalesSetup."Calc. Inv. Discount" THEN BEGIN
  CurrPage.SalesLines.PAGE.CalcInvDisc;
  COMMIT
END;
IF "Tax Area Code" = " THEN
  PAGE.RUNMODAL(PAGE::"Sales Order Statistics",Rec)
ELSE BEGIN
  //AvaValidCall.AvaStatisticsActionSales(Rec,TRUE);//Avalara
  AvaValidCall.AvaStatisticsActionSales(Rec,TRUE,TRUE);//Avalara
  PAGE.RUNMODAL(PAGE::"Sales Order Stats.",Rec);
END;
```

*Changes in Page Action Added → Test Report;*

*Add Code (Search for below first line and add colored line)*

```
//AvaValidCall.AvaStatisticsActionSales(Rec,FALSE);//Avalara
AvaValidCall.AvaStatisticsActionSales(Rec,FALSE,FALSE);//Avalara
ReportPrint.PrintSalesHeader(Rec);
```

## Page 10028 Sales Order Invoice

*Changes in Version List:*

Old Value	NAVNA7.00,AVANA7.00.06.00
New Value	NAVNA7.00,AVANA7.00.06.01

*Change to trigger → Statistics@OnAction*

*Add Code (Search for below first line and add colored line)*

```
SalesSetup.GET;
IF SalesSetup."Calc. Inv. Discount" THEN BEGIN
  CurrPage.SalesLines.PAGE.CalcInvDisc;
  COMMIT
END;

IF "Tax Area Code" = " THEN
  PAGE.RUNMODAL(PAGE::"Sales Order Statistics",Rec)
ELSE BEGIN
  //AvaValidCall.AvaStatisticsActionSales(Rec,TRUE);//Avalara
  AvaValidCall.AvaStatisticsActionSales(Rec,TRUE,TRUE);//Avalara
  PAGE.RUNMODAL(PAGE::"Sales Order Stats.",Rec)
END;
```

*Changes in Page Action Added → Test Report;*

*Add Code (Search for below first line and add colored line)*

```
//AvaValidCall.AvaStatisticsActionSales(Rec,FALSE);//Avalara
AvaValidCall.AvaStatisticsActionSales(Rec,FALSE,FALSE);//Avalara
ReportPrint.PrintSalesHeader(Rec);
```

## Release Avalara AvaTax NAV 2013 RTM 7.00.33781.06.02

### Details:

Fixed an issue with Entity/Use Codes list, revised the Entity/Use Codes list in AvaTax for Microsoft Dynamics NAV

### Changes in Table

#### Table 14073415 AVA Install

*Change in Version List details of objects:*

Old Value	AVANA7.00.06.00
New Value	AVANA7.00.06.02

*Changes in Function → EntityUseCodeTableData:*

*Delete Text Constants:*

Text14073212	N
Text14073213	Federal Government
Text14073214	State or Local Government
Text14073215	Tribal Government
Text14073216	Foreign Diplomat
Text14073217	Charitable Organization
Text14073218	Religious or Educational Organization
Text14073219	Resale
Text14073220	Agricultural Production
Text14073221	Agricultural Production
Text14073222	Direct Pay Permit
Text14073223	Multiple Points of Use
Text14073224	Direct Mail
Text14073225	Other

*Create New Text Constants:*

Name	ConstValue
Text14073212	N
Text14073213	P
Text14073214	Q
Text14073215	R
Text14073200A	Federal Government
Text14073201B	State Government
Text14073202C	Tribal Government
Text14073203D	Foreign Diplomat
Text14073204E	Charitable Organization
Text14073205F	Religious or Educational Organization
Text14073206G	Resale
Text14073207H	Agricultural Production
Text14073208I	Industrial Production/Manufacturing



Text14073209J	Direct Pay Permit
Text14073210K	Direct Mail
Text14073211L	Custom / Other
Text14073212N	Local Government
Text14073213P	Commercial Aquaculture

Old Code: Replace Old code with **New code**

```

AvaEntityUseCode.RESET;
AvaEntityUseCode.DELETEALL;

AvaEntityUseCode.INIT; AvaEntityUseCode."No.":=Text14073200;
AvaEntityUseCode.Description:=Text14073213; AvaEntityUseCode.INSERT;
AvaEntityUseCode.INIT; AvaEntityUseCode."No.":=Text14073201;
AvaEntityUseCode.Description:=Text14073214; AvaEntityUseCode.INSERT;
AvaEntityUseCode.INIT; AvaEntityUseCode."No.":=Text14073202;
AvaEntityUseCode.Description:=Text14073215; AvaEntityUseCode.INSERT;
AvaEntityUseCode.INIT; AvaEntityUseCode."No.":=Text14073203;
AvaEntityUseCode.Description:=Text14073216; AvaEntityUseCode.INSERT;
AvaEntityUseCode.INIT; AvaEntityUseCode."No.":=Text14073204;
AvaEntityUseCode.Description:=Text14073217; AvaEntityUseCode.INSERT;
AvaEntityUseCode.INIT; AvaEntityUseCode."No.":=Text14073205;
AvaEntityUseCode.Description:=Text14073218; AvaEntityUseCode.INSERT;
AvaEntityUseCode.INIT; AvaEntityUseCode."No.":=Text14073206;
AvaEntityUseCode.Description:=Text14073219; AvaEntityUseCode.INSERT;
AvaEntityUseCode.INIT; AvaEntityUseCode."No.":=Text14073207;
AvaEntityUseCode.Description:=Text14073220; AvaEntityUseCode.INSERT;
AvaEntityUseCode.INIT; AvaEntityUseCode."No.":=Text14073208;
AvaEntityUseCode.Description:=Text14073221; AvaEntityUseCode.INSERT;
AvaEntityUseCode.INIT; AvaEntityUseCode."No.":=Text14073209;
AvaEntityUseCode.Description:=Text14073222; AvaEntityUseCode.INSERT;
AvaEntityUseCode.INIT; AvaEntityUseCode."No.":=Text14073211;
AvaEntityUseCode.Description:=Text14073224; AvaEntityUseCode.INSERT;
AvaEntityUseCode.INIT; AvaEntityUseCode."No.":=Text14073212;
AvaEntityUseCode.Description:=Text14073225; AvaEntityUseCode.INSERT;

```

**New Code**

```

AvaEntityUseCode.RESET;
AvaEntityUseCode.DELETEALL;

AvaEntityUseCode.INIT; AvaEntityUseCode."No.":=Text14073200;
AvaEntityUseCode.Description:=Text14073200A; AvaEntityUseCode.INSERT;
AvaEntityUseCode.INIT; AvaEntityUseCode."No.":=Text14073201;
AvaEntityUseCode.Description:=Text14073201B; AvaEntityUseCode.INSERT;
AvaEntityUseCode.INIT; AvaEntityUseCode."No.":=Text14073202;
AvaEntityUseCode.Description:=Text14073202C; AvaEntityUseCode.INSERT;
AvaEntityUseCode.INIT; AvaEntityUseCode."No.":=Text14073203;
AvaEntityUseCode.Description:=Text14073203D; AvaEntityUseCode.INSERT;

```

```

AvaEntityUseCode.INIT; AvaEntityUseCode."No.":=Text14073204;
AvaEntityUseCode.Description:=Text14073204E; AvaEntityUseCode.INSERT;
AvaEntityUseCode.INIT; AvaEntityUseCode."No.":=Text14073205;
AvaEntityUseCode.Description:=Text14073205F; AvaEntityUseCode.INSERT;
AvaEntityUseCode.INIT; AvaEntityUseCode."No.":=Text14073206;
AvaEntityUseCode.Description:=Text14073206G; AvaEntityUseCode.INSERT;
AvaEntityUseCode.INIT; AvaEntityUseCode."No.":=Text14073207;
AvaEntityUseCode.Description:=Text14073207H; AvaEntityUseCode.INSERT;
AvaEntityUseCode.INIT; AvaEntityUseCode."No.":=Text14073208;
AvaEntityUseCode.Description:=Text14073208I; AvaEntityUseCode.INSERT;
AvaEntityUseCode.INIT; AvaEntityUseCode."No.":=Text14073209;
AvaEntityUseCode.Description:=Text14073209J; AvaEntityUseCode.INSERT;
AvaEntityUseCode.INIT; AvaEntityUseCode."No.":=Text14073210;
AvaEntityUseCode.Description:=Text14073210K; AvaEntityUseCode.INSERT;
AvaEntityUseCode.INIT; AvaEntityUseCode."No.":=Text14073211;
AvaEntityUseCode.Description:=Text14073211L; AvaEntityUseCode.INSERT;
AvaEntityUseCode.INIT; AvaEntityUseCode."No.":=Text14073212;
AvaEntityUseCode.Description:=Text14073212N; AvaEntityUseCode.INSERT;
AvaEntityUseCode.INIT; AvaEntityUseCode."No.":=Text14073213;
AvaEntityUseCode.Description:=Text14073213P; AvaEntityUseCode.INSERT;
AvaEntityUseCode.INIT; AvaEntityUseCode."No.":=Text14073214;
AvaEntityUseCode.Description:=Text14073214Q; AvaEntityUseCode.INSERT;
AvaEntityUseCode.INIT; AvaEntityUseCode."No.":=Text14073215;
AvaEntityUseCode.Description:=Text14073215R; AvaEntityUseCode.INSERT;
    
```

## Changes in Codeunits

### Page 14073500 AVA Entity\Use Code List

*Changes in Version List:*

Old Value	AVANA7.00.06.00
New Value	AVANA7.00.06.02

*Create new global variable:*

Name	Data Type	Subtype	Length
AVAIInstall	Record	AVA Install	

*Changes in Page Action Added → Restore default values → OnAction*

AVAIInstall.EntityUseCodeTableData;
-------------------------------------

### Page Properties

Name	Restore default values
CaptionML	Restore default values
Promoted	Yes
Image	Restore
Promoted Category	Process

**Page 14073503 AVA Entity\Use Code***Changes in Version List:*

Old Value	AVANA7.00.06.00
New Value	AVANA7.00.06.02

*Create new global variable:*

Name	DataType	Subtype	Length
AVAIInstall	Record	AVA Install	

*Changes in Page Action Added → Restore default values → OnAction*

AVAIInstall.EntityUseCodeTableData;
-------------------------------------

*Page Properties*

Name	Restore default values
CaptionML	Restore default values
Promoted	Yes
Image	Restore
Promoted Category	Process



## Release Avalara AvaTax NAV 2013 RTM 7.00.33781.06.03

### Details:

Fixed an issue with Reconciliation Utility




### Changes in Table

*Import list of below Table objects from AVANA7.00.06.03.Fob and replace*

Type	ID	Name	Modified	Version List
	14073522	AVA Reconciliation		AVANA7.00.06.03
	14073523	AVA Reconciliation Buffer		AVANA7.00.06.03
	14073524	AVA Tax Update Utility		AVANA7.00.06.03

### Changes in Pages

*Import list of below Page objects from AVANA7.00.06.03.Fob and replace*

Type	ID	Name	Modified	Version List
	14073523	AVA Reconciliation		AVANA7.00.06.03
	14073524	AVA Reconciliation Subform		AVANA7.00.06.03
	14073525	AVA Reconciliation Details		AVANA7.00.06.03

## Release Avalara AvaTax NAV 2013 RTM 7.00.33781.06.04

**Note:**

- In this release, Avalara made major changes storing data in Avalara tables, we suggest to direct import objects from Avalara.fob file.
- If you are being migrating from old Avalara version to 06.04, please use migration tool along with steps mentioned in migrate tool help file.
- [Upgrade Utility for Dynamics NAV](#)