



Avalara AvaTax for Microsoft Dynamics NAV 2013 RTM

Specific Code Change Document

Version 01

Revision date: 6/3/15

Product release: Microsoft NAV 7.00.33781.05.02

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..



Changes in Avalara objects 3

 Codeunit 14073301 AVA Tax Engine 3

 Changes in function → fnCalcTax..... 3

 Changes in function → fnCalcTax1..... 7

Changes in Avalara objects

Codeunit 14073301 AVA Tax Engine

Changes in function → fnCalcTax

Modified Code (Search for 1st Line) changes highlighted in red Color

```

IF STRLEN(FromAddress)>0 THEN  AvaAddrSeparator:=',';

AVAAddrLatLongFlag:=FALSE;
IF AvaReleaseStatusFlag THEN BEGIN
  IF (Avaconfig."Release Quantity" = '3') THEN BEGIN
    AvaCallGetTax := FALSE;
    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 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;

        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); //AVANA7.00.05.02
          SalesLine.SETFILTER("Qty. to Invoice",>=0); //AVANA7.00.05.02
          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;

```

Modified Code (Search for 1st Line) changes highlighted in red Color

```

Disc:=FALSE;
IF SalesLine."Inv. Discount Amount" <> 0 THEN
  Disc:=TRUE
ELSE
  Disc:=FALSE;

//AVANA7.00.05.00
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",<'); //AVANA7.00.05.00
    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
      //AVANA7.00.05.02 Start
      //IF SalesLine."Return Qty. to Receive" > 0 THEN
      // 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",<') //AVANA7.00.05.00
      //ELSE
      // 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",<'); //AVANA7.00.05.00
      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",<');
      //AVANA7.00.05.02 end
      QtyPerLine:=SalesLine."Return Qty. to Receive";
    END ELSE BEGIN
      //AVANA7.00.05.02 Start
      //IF SalesLine."Qty. to Ship" > 0 THEN
      // 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",<') //AVANA7.00.05.00
      //ELSE
      // 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",<'); //AVANA7.00.05.00
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",<');
//AVANA7.00.05.02 END
QtyPerLine:=SalesLine."Qty. to Ship";
END;
END ELSE IF Avaconfig."Statistics Quantity" = '3' THEN BEGIN
//AVANA7.00.05.02 Start
//IF SalesLine."Qty. to Invoice" > 0 THEN
// 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",<') //AVANA7.00.05.00
//ELSE
// 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",<'); //AVANA7.00.05.00
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",<');
//AVANA7.00.05.02 END
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",<'); //AVANA7.00.05.00
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
//AVANA7.00.05.02 start
//IF SalesLine."Return Qty. to Receive" > 0 THEN
// 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",<') //AVANA7.00.05.00
//ELSE
// 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",<'); //AVANA7.00.05.00
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",<');
        //AVANA7.00.05.02 end
        QtyPerLine:=SalesLine."Return Qty. to Receive";
    END ELSE BEGIN
        //AVANA7.00.05.02 start
        //IF SalesLine."Qty. to Ship" > 0 THEN
        // 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",<) //AVANA7.00.05.00
        //ELSE
        // 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",<); //AVANA7.00.05.00
        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",<);
        //AVANA7.00.05.02 end
        QtyPerLine:=SalesLine."Qty. to Ship";
    END;
    END ELSE IF Avaconfig."Release Quantity" = '3' THEN BEGIN
        //AVANA7.00.05.02 start
        //IF SalesLine."Qty. to Invoice" > 0 THEN
        // 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",<) //AVANA7.00.05.00
        //ELSE
        // 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",<); //AVANA7.00.05.00
        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",<);
        //AVANA7.00.05.02 END
        QtyPerLine:=SalesLine."Qty. to Invoice";
    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;
        END;
    END;

```

*Changes in function → fnCalcTax1*Modified Code (Search for 1st Line) changes highlighted in red Color

```
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; //AVANA7.00.05.02
      QtyPerLine:=QtyPerLine*-1; //AVANA7.00.05.02
    LineAmtPassed:= LineAmtPassed * -1;
  END;

  IF Disc THEN
  IF NOT (SalesLine.Quantity=0) THEN
    TotalDisc:=TotalDisc + ((SalesLine."Inv. Discount Amount"/SalesLine.Quantity)*QtyPerLine)
  ELSE
    TotalDisc:=TotalDisc + 0;

  AvaWindowOpen(StrMessage+Text14073225);
  fnUpdateTaxCode(SalesLine);
  fnGetTaxLine(oGetTaxRequest,
```