



Avalara AvaTax for Microsoft Dynamics NAV 2013 RTM

Specific Code Change Document

Version 01

Revision date: 1/27/16

Product release: Microsoft NAV 7.00.33781.05.06

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: Microsoft NAV 7.00.33781.05.01 3

Release: Microsoft NAV 7.00.33781.05.02 16

Release: Microsoft NAV 7.00.33781.05.03 21

Release: Microsoft NAV 7.00.33781.05.04 22

Release: Microsoft NAV 7.00.33781.05.05 25

Release: Microsoft NAV 7.00.33781.05.06 30

Release: Microsoft NAV 7.00.33781.05.01

Details

Fixed the warning message displayed during transaction posting when the **Quantity at Release** or **Quantity at Statistics** is set to the **Quantity to Invoice** and the transaction line **Unit Price** and the **Discount** have decimal values

Codeunit 14073301 AVA Tax Engine

Add new in Global variable:

RecCurrency : **Record** **Currency**

Changes in function → *fnCalcTax*

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
      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
      QtyPerLine:=SalesLine."Return Qty. to Receive";
    END ELSE BEGIN
      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
    QtyPerLine:=SalesLine."Qty. to Ship";
END;
END ELSE IF Avaconfig."Statistics Quantity" = '3' THEN BEGIN
    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
            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
                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
                    QtyPerLine:=SalesLine."Return Qty. to Receive";
                END ELSE BEGIN
                    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
QtyPerLine:=SalesLine."Qty. to Ship";
END;
END ELSE IF Avaconfig."Release Quantity" = '3' THEN BEGIN
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
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;

```

Changes in function → fnCalcTax1

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

```

//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;

Avaconfig.FINDFIRST;
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",<'); //AVANA7.00.05.00
QtyPerLine:=SalesLine."Qty. Shipped Not Invoiced";
END
ELSE BEGIN
LineAmtPassed:=ROUND(ROUND(SalesLine."Qty. to Invoice"*UnitPrice,RecCurrency."Amount Rounding
Precision") -

```



```

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",<'); //AVANA7.00.05.00
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",<'); //AVANA7.00.05.00
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",<'); //AVANA7.00.05.00
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;
LineAmtPassed:= LineAmtPassed * -1;
END;

```

Changes in function → fnCalcTaxBatch

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

```

UnitPrice:=SalesLine."Unit Price";
fnGetItemDetails(SalesLine.Type,SalesLine."No.",SalesLine."Tax Group
Code",Ref1,Ref2,"G\LAccNo.",TaxabilityCode,UnitPrice);

Disc:=FALSE;
IF SalesLine."Inv. Discount Amount" <> 0 THEN
Disc:=TRUE
ELSE
Disc:=FALSE;
IF NOT (SalesHead.Ship AND SalesHead.Invoice) THEN BEGIN
//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 //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",<'); //AVANA7.00.05.00
    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",<'); //AVANA7.00.05.00
      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",<'); //AVANA7.00.05.00
      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",<'); //AVANA7.00.05.00
    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",<'); //AVANA7.00.05.00
    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
      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
      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",<'); //AVANA7.00.05.00
      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",<'); //AVANA7.00.05.00
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;
LineAmtPassed:= LineAmtPassed * -1;
END;

```

Changes in function → fnObjToTbl

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

```

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

IF NOT(COPYSTR(AvaHead."Document No.",1,3) = 'SER') THEN BEGIN
SalesLine.SETRANGE(SalesLine."Document No.",COPYSTR(AvaHead."Document No.",4,100));
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";

//AVANA7.00.05.00 start
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");

PrepLineAmt := AvaLine."Total Taxable Amount" + AvaLine."Tax Tax Amount" + AvaLine."Tax Exempt
Amount" + AvaLine."Tax Discount 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";
//AVANA7.00.05.00 end
SalesLine.MODIFY;
END ELSE BEGIN

```

Changes in function → fnServiceCalcTax

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

```

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

//AVANA7.00.05.00
IF ServiceHead."Currency Code" = " THEN
  RecCurrency.InitRoundingPrecision
ELSE BEGIN
  ServiceHead.TESTFIELD("Currency Factor");
  RecCurrency.GET(ServiceHead."Currency 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",<'); //AVANA7.00.05.00
  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",<'); //AVANA7.00.05.00
  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",<'); //AVANA7.00.05.00
  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",<'); //AVANA7.00.05.00
  QtyPerLine:=ServiceLine."Qty. to Invoice";
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;

```

Changes in function → fnServiceCalcTax1

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

```

Disc:=FALSE;
IF ServiceLine."Inv. Discount Amount" <> 0 THEN
  Disc:=TRUE
ELSE
  Disc:=FALSE;
Avaconfig.FINDFIRST;
InvoiceFlag := Avaconfig.Invoice;
//AVANA7.00.05.00
IF ServiceHead."Currency Code" = " THEN
  RecCurrency.InitRoundingPrecision
ELSE BEGIN
  ServiceHead.TESTFIELD("Currency Factor");
  RecCurrency.GET(ServiceHead."Currency 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",<); //AVANA7.00.05.00
      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",<); //AVANA7.00.05.00
      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",<); //AVANA7.00.05.00
      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",<); //AVANA7.00.05.00
      QtyPerLine:=ServiceLine."Qty. to Invoice";
    END;
  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",<); //AVANA7.00.05.00
  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",<); //AVANA7.00.05.00
  QtyPerLine:=ServiceLine."Qty. to Invoice";
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;

```

Changes in function → fnServiceCalcTaxBatch

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 (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",<); //AVANA7.00.05.00
      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",<); //AVANA7.00.05.00
        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",<); //AVANA7.00.05.00
        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", '<'); //AVANA7.00.05.00
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", '<'); //AVANA7.00.05.00
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
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", '<'); //AVANA7.00.05.00
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", '<'); //AVANA7.00.05.00
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", '<'); //AVANA7.00.05.00
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;
LineAmtPassed:= LineAmtPassed * -1;
END;
END;

```

Changes in function → fnServCalcTaxBatch

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

```

Disc:=FALSE;
IF ServiceLine."Inv. Discount Amount" <> 0 THEN
Disc:=TRUE
ELSE
Disc:=FALSE;
//AVANA7.00.05.00
IF ServiceHead."Currency Code" = " THEN
RecCurrency.InitRoundingPrecision
ELSE BEGIN
ServiceHead.TESTFIELD("Currency Factor");
RecCurrency.GET(ServiceHead."Currency 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",<'); //AVANA7.00.05.00
  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",<'); //AVANA7.00.05.00
    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",<'); //AVANA7.00.05.00
    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",<'); //AVANA7.00.05.00
    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",<'); //AVANA7.00.05.00
    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",<'); //AVANA7.00.05.00
      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",<'); //AVANA7.00.05.00
      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",<'); //AVANA7.00.05.00  
    QtyPerLine:=ServiceLine."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;  
                LineAmtPassed:= LineAmtPassed * -1;  
            END;  
        END;
```

Release: Microsoft NAV 7.00.33781.05.02

Details

Fixed the warning message displayed during transaction posting when **At Order Release** was configured to **Quantity to Invoice** or **Quantity to Ship** and the transaction **Quantity** was 0.

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
//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;
```

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,
```

Release: Microsoft NAV 7.00.33781.05.03

Details

Fixed the warning message displayed during transaction posting when **At Order Release** was configured to **Quantity to Invoice** or **Quantity to Ship** and the transaction **Quantity** was 0.

OBJECT Page 6631 Sales Return Order Subform

Changes in Trigger → OnDeleteRecord

Old code

```
IF (Quantity <> 0) AND ItemExists("No.") THEN BEGIN
  COMMIT;
  IF NOT ReserveSalesLine.DeleteLineConfirm(Rec) THEN
    EXIT(FALSE);
  ReserveSalesLine.DeleteLine(Rec);
END;
AVATaxEngine.AvaDocNumSalesLine(Rec,FALSE)//Avalara
```

New code (Modified code marked in red color)

```
IF (Quantity <> 0) AND ItemExists("No.") THEN BEGIN
  COMMIT;
  IF NOT ReserveSalesLine.DeleteLineConfirm(Rec) THEN
    EXIT(FALSE);
  ReserveSalesLine.DeleteLine(Rec);
END;
AVATaxEngine.AvaDocNumSalesLine(Rec,TRUE)//Avalara
```

Release: Microsoft NAV 7.00.33781.05.04

Details

Fixed the issue related to Ship-to Code on Sales Credit Memo & Return Order, user will now able to use Ship to Address for transaction tax calculation.

Codeunit 14073301 AVA Tax Engine

Changes in function → *fnCalcTax*

New code (Search for below 1st line and Modified code marked in red color)

```

AvaWindowOpen(StrMessage);
"TmpL 1":=""; "TmpL 2":=""; "TmpL 3":=""; TmpCity:=""; TmpState:=""; TmpZip:=""; TmpCountry:=""; ResultText:="";
TmpTaxRegionID :=0;TmpLatitude:="";TmpLongitude:="";
TmpAddressSrc:=Text14073215;
IF (DocumentType = 'Credit Memo') OR (DocumentType = 'Return Order') THEN BEGIN
  IF SalesHead."Ship-to Code" = " THEN BEGIN //7.00.33781.05.04
    IF Avaconfig."Return Address" = '1' THEN BEGIN
      fnIsEmptyAddressRet(3,SalesHead,EntityUseCode);
      "TmpL 1":=SalesHead."Sell-to Address";
      "TmpL 2":=SalesHead."Sell-to Address 2";
      "TmpL 3":="";
      TmpCity:=SalesHead."Sell-to City";
      TmpState:=SalesHead."Sell-to County";
      TmpZip:=fnAVAPinCheck(SalesHead."Sell-to Post Code");
      TmpCountry:=FORMAT(SalesHead."Sell-to Country/Region Code");
    END ELSE IF Avaconfig."Return Address" = '2' THEN BEGIN
      fnIsEmptyAddressRet(2,SalesHead,EntityUseCode);
      "TmpL 1":=SalesHead."Bill-to Address";
      "TmpL 2":=SalesHead."Bill-to Address 2";
      "TmpL 3":="";
      TmpCity:=SalesHead."Bill-to City";
      TmpState:=SalesHead."Bill-to County";
      TmpZip:=fnAVAPinCheck(SalesHead."Bill-to Post Code");
      TmpCountry:=FORMAT(SalesHead."Bill-to Country/Region Code");
    END ELSE IF Avaconfig."Return Address" = '3' THEN BEGIN
      fnIsEmptyAddressRet(1,SalesHead,EntityUseCode);
      "TmpL 1":=SalesHead."Ship-to Address";
      "TmpL 2":=SalesHead."Ship-to Address 2";
      "TmpL 3":="";
      TmpCity:=SalesHead."Ship-to City";
      TmpState:=SalesHead."Ship-to County";
      TmpZip:=fnAVAPinCheck(SalesHead."Ship-to Post Code");
      TmpCountry:=FORMAT(SalesHead."Ship-to Country/Region Code");
    END;
  //7.00.33781.05.04 Start
  END
  ELSE BEGIN
    ShipToAddress.RESET;
    ShipToAddress.SETRANGE("Customer No.",SalesHead."Sell-to Customer No.");
    ShipToAddress.SETRANGE(Code,SalesHead."Ship-to Code");
    IF ShipToAddress.FINDFIRST THEN BEGIN
      "TmpL 1":=ShipToAddress.Address;
      "TmpL 2":=ShipToAddress."Address 2";
      "TmpL 3":="";
      TmpCity:=ShipToAddress.City;
      TmpState:=ShipToAddress.County;
      TmpZip:=fnAVAPinCheck(ShipToAddress."Post Code");
      TmpCountry:=FORMAT(ShipToAddress."Country/Region Code");
    END;
  END;
END;

```

```

END;
END;
//7.00.33781.05.04 End
END ELSE BEGIN
  IF fnIsEmptyAddress(1,SalesHead,EntityUseCode) THEN BEGIN
    TmpAddressSrc:=Text14073215;
  
```

Changes in function → *fnCalcTaxBatch*

New code (Search for below 1st line and Modified code marked in red color)

```

IF NOT Avaconfig."Show Message Dialog" THEN
  Window.OPEN(StrMessage)
ELSE
  AvaWindowOpen(StrMessage);
  "TmpL 1" := ""; "TmpL 2" := ""; "TmpL 3" := ""; TmpCity := ""; TmpState := ""; TmpZip := ""; TmpCountry := ""; ResultText := "";
  TmpTaxRegionID := 0; TmpLatitude := ""; TmpLongitude := "";
  TmpAddressSrc := Text14073215;

  IF (DocumentType = 'Credit Memo') OR (DocumentType = 'Return Order') THEN BEGIN
    IF SalesHead."Ship-to Code" = " THEN BEGIN //7.00.33781.05.04
      IF Avaconfig."Return Address" = '3' THEN BEGIN
        fnIsEmptyAddressRet(1,SalesHead,EntityUseCode);
        "TmpL 1" := SalesHead."Sell-to Address";
        "TmpL 2" := SalesHead."Sell-to Address 2";
        "TmpL 3" := "";
        TmpCity := SalesHead."Sell-to City";
        TmpState := SalesHead."Sell-to County";
        TmpZip := fnAVAPinCheck(SalesHead."Sell-to Post Code");
        TmpCountry := FORMAT(SalesHead."Sell-to Country/Region Code");
      END ELSE IF Avaconfig."Return Address" = '2' THEN BEGIN
        fnIsEmptyAddressRet(2,SalesHead,EntityUseCode);
        "TmpL 1" := SalesHead."Bill-to Address";
        "TmpL 2" := SalesHead."Bill-to Address 2";
        "TmpL 3" := "";
        TmpCity := SalesHead."Bill-to City";
        TmpState := SalesHead."Bill-to County";
        TmpZip := fnAVAPinCheck(SalesHead."Bill-to Post Code");
        TmpCountry := FORMAT(SalesHead."Bill-to Country/Region Code");
      END ELSE IF Avaconfig."Return Address" = '1' THEN BEGIN
        fnIsEmptyAddressRet(3,SalesHead,EntityUseCode);
        "TmpL 1" := SalesHead."Ship-to Address";
        "TmpL 2" := SalesHead."Ship-to Address 2";
        "TmpL 3" := "";
        TmpCity := SalesHead."Ship-to City";
        TmpState := SalesHead."Ship-to County";
        TmpZip := fnAVAPinCheck(SalesHead."Ship-to Post Code");
        TmpCountry := FORMAT(SalesHead."Ship-to Country/Region Code");
      END;
      //7.00.33781.05.04 Start
    END
  ELSE BEGIN
    ShipToAddress.RESET;
    ShipToAddress.SETRANGE("Customer No.",SalesHead."Sell-to Customer No.");
    ShipToAddress.SETRANGE(Code,SalesHead."Ship-to Code");
    IF ShipToAddress.FINDFIRST THEN BEGIN
      "TmpL 1" := ShipToAddress.Address;
    
```

```
"TmpL 2":=ShipToAddress."Address 2";
"TmpL 3":="";
TmpCity:=ShipToAddress.City;
TmpState:=ShipToAddress.County;
TmpZip:=fnAVAPinCheck(ShipToAddress."Post Code");
TmpCountry:=FORMAT(ShipToAddress."Country/Region Code");
END;
END;
//7.00.33781.05.04 End
END ELSE BEGIN

IF fnIsEmptyAddress(1,SalesHead,EntityUseCode) THEN BEGIN
    TmpAddressSrc:=Text14073215;
```


Release: Microsoft NAV 7.00.33781.05.05

Details

Fixed the issue related Credit Outstanding for Customer. On Sales Module, User can now able to receive messages / warning / error while using outstanding feature for customer on transaction

Codeunit 14073301 AVA Tax Engine

Changes in function → fnObjToTbl

New code (Search for below 1st line and Modified code marked in red color)

```

IF NOT(COPYSTR(AvaHead."Document No.",1,3) = 'SER') THEN BEGIN
  SalesLine.SETRANGE(SalesLine."Document No.",COPYSTR(AvaHead."Document No.",4,100));
  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";

    //AVANA7.00.05.00 start
    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 //AVANA7.00.05.05
    SalesLine."VAT %" := AvaLine."Tax Rate" * 100;
    SalesLine."Prepm. Line Amount" := ROUND((SalesLine."Prepayment %" * SalesLine."Line Amount") /
    100,RecCurrency."Amount Rounding Precision");

    PrepLineAmt := AvaLine."Total Taxable Amount" + AvaLine."Tax Tax Amount" + AvaLine."Tax Exempt
    Amount" + AvaLine."Tax Discount Amount";
    IF SalesOrderHdr."Prepm. Include Tax" THEN
      SalesLine."Prepm. Amt. Incl. VAT" := ROUND((SalesLine."Prepayment %" * PrepLineAmt) /
    100,RecCurrency."Amount Rounding Precision")
    ELSE
      SalesLine."Prepm. Amt. Incl. VAT" := SalesLine."Prepm. Line Amount";
    SalesLine."Prepayment Amount" := SalesLine."Prepm. Amt. Incl. VAT";
    //AVANA7.00.05.00 end

    //AVANA7.00.05.05 Start
    IF SalesLine."Document Type" IN [SalesLine."Document Type"::"Return Order",SalesLine."Document
    Type"::"Credit Memo"] THEN BEGIN
      SalesLine."Outstanding Quantity" := SalesLine.Quantity - SalesLine."Return Qty. Received";
      SalesLine."Outstanding Qty. (Base)" := SalesLine."Quantity (Base)" - SalesLine."Return Qty. Received (Base)";
      SalesLine."Return Qty. Rcd. Not Invd." := SalesLine."Return Qty. Received" - SalesLine."Quantity Invoiced";
      SalesLine."Ret. Qty. Rcd. Not Invd.(Base)" := SalesLine."Return Qty. Received (Base)" - SalesLine."Qty. Invoiced
      (Base)";
    END ELSE BEGIN
      SalesLine."Outstanding Quantity" := SalesLine.Quantity - SalesLine."Quantity Shipped";
      SalesLine."Outstanding Qty. (Base)" := SalesLine."Quantity (Base)" - SalesLine."Qty. Shipped (Base)";
      SalesLine."Qty. Shipped Not Invoiced" := SalesLine."Quantity Shipped" - SalesLine."Quantity Invoiced";
      SalesLine."Qty. Shipped Not Invd. (Base)" := SalesLine."Qty. Shipped (Base)" - SalesLine."Qty. Invoiced (Base)";
    END;
    SalesLine."Completely Shipped" := (SalesLine.Quantity <> 0) AND (SalesLine."Outstanding Quantity" = 0);
  
```

```

    AmountInclVAT := ROUND( (SalesLine."Line Amount" + (SalesLine."Line Amount" * AvaLine."Tax Rate")) *
SalesLine."Outstanding Quantity" / SalesLine.Quantity
    ,RecCurrency."Amount Rounding Precision");
    SalesLine.VALIDATE("Outstanding Amount",AmountInclVAT);
    //AVANA7.00.05.05 End

    SalesLine.MODIFY;
END ELSE BEGIN
    SalesInvLine.SETRANGE(SalesInvLine."Document No.",COPYSTR(AvaHead."Document No.",4,100));
    SalesInvLine.SETFILTER(SalesInvLine."Line No.",oTaxLine.No);
    IF SalesInvLine.FINDFIRST THEN
        AvaLine."Tax Area" := SalesInvLine."Tax Area Code"
    ELSE BEGIN
        recObjPostedCrLine.SETRANGE(recObjPostedCrLine."Document No.",COPYSTR(AvaHead."Document
No.",4,100));
        recObjPostedCrLine.SETFILTER(recObjPostedCrLine."Line No.",oTaxLine.No);
        IF recObjPostedCrLine.FINDFIRST THEN
            AvaLine."Tax Area" := recObjPostedCrLine."Tax Area Code"
        END
    END
END ELSE BEGIN
    ServiceLine.SETRANGE(ServiceLine."Document No.",COPYSTR(AvaHead."Document No.",7,100));
    ServiceLine.SETFILTER(ServiceLine."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";
        //AVANA7.00.05.05 start
        IF ServiceHeader."Currency Code" = " THEN
            RecCurrency.InitRoundingPrecision
        ELSE BEGIN
            ServiceHeader.TESTFIELD("Currency Factor");
            RecCurrency.GET(ServiceHeader."Currency Code");
            RecCurrency.TESTFIELD("Amount Rounding Precision");
        END;
        ServiceLine."VAT %" := AvaLine."Tax Rate" * 100;

        IF ServiceLine."Document Type" = ServiceLine."Document Type>::"Credit Memo" THEN BEGIN
            ServiceLine."Outstanding Quantity" := ServiceLine.Quantity;
            ServiceLine."Outstanding Qty. (Base)" := ServiceLine."Quantity (Base)";
        END ELSE BEGIN
            ServiceLine."Outstanding Quantity" := ServiceLine.Quantity - ServiceLine."Quantity Shipped";
            ServiceLine."Outstanding Qty. (Base)" := ServiceLine."Quantity (Base)" - ServiceLine."Qty. Shipped (Base)";
            ServiceLine."Qty. Shipped Not Invoiced" := ServiceLine."Quantity Shipped" - ServiceLine."Quantity Invoiced" -
ServiceLine."Quantity Consumed";
            ServiceLine."Qty. Shipped Not Invd. (Base)" := ServiceLine."Qty. Shipped (Base)" - ServiceLine."Qty. Invoiced
(Base)" - ServiceLine."Qty. Consumed (Base)";
        END;
        ServiceLine."Completely Shipped" := (ServiceLine.Quantity <> 0) AND (ServiceLine."Outstanding Quantity" = 0);
        AmountInclVAT := ROUND( (ServiceLine."Line Amount" + (ServiceLine."Line Amount" * AvaLine."Tax Rate"))
* ServiceLine."Outstanding Quantity" / ServiceLine.Quantity
        ,RecCurrency."Amount Rounding Precision");
        ServiceLine.VALIDATE("Outstanding Amount",AmountInclVAT);
        //AVANA7.00.05.05 End

        ServiceLine.MODIFY;
    END ELSE BEGIN
        ServInvLine.SETRANGE(ServInvLine."Document No.",COPYSTR(AvaHead."Document No.",7,100));
        ServInvLine.SETFILTER(ServInvLine."Line No.",oTaxLine.No);

```

```

IF ServInvLine.FINDFIRST THEN BEGIN
  AvaLine."Tax Area" := ServInvLine."Tax Area Code";
END ELSE BEGIN
  ServCrMemoLine.SETRANGE(ServCrMemoLine."Document No.",COPYPSTR(AvaHead."Document
No.",7,100));
  ServCrMemoLine.SETFILTER(ServCrMemoLine."Line No.",oTaxLine.No);
  IF ServCrMemoLine.FINDFIRST THEN
    AvaLine."Tax Area" := ServCrMemoLine."Tax Area Code"
  END;
END;
END;
IF BlnInsertLn THEN AvaLine.INSERT ELSE AvaLine.MODIFY;

```

Codeunit 14073310 AVA Validate Call

Changes in function → AvaGetTaxOrder

New code (Search for below 1st line and Modified code marked in red color)

```

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);
      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;
            //AVANA7.00.05.05 start
            //IF (AvaConfig."Enable Tax Amt Adj.") AND (LAvaLine."Item No."=TextSalesTaxAdjust) THEN BEGIN
            // SalesLine.Amount:=0;
            // SalesLine."VAT Base Amount":=LAvaLine."Total Taxable Amount";
            // SalesLine."Amount Including VAT" := SalesLine.Amount + LAvaLine."Tax Tax 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;
            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.VALIDATE("Amount Including VAT",LAvaLine."Tax Tax Amount" + LAvaLine."Total Taxable
Amount" + LAvaLine."Tax Exempt Amount");
              SalesLine."VAT %":=0;
              SalesLine."Line Amount":=0;
            END;

```

```

//AVANA7.00.05.05 end
END;
SalesLine.MODIFY;

```

Changes in function → AvaGetTaxOrderBatch

New code (Search for below 1st line and Modified code marked in red color)

```

IF AvaGetTaxOrderBatchPost() 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);
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
//AVANA7.00.05.05 start
//IF LAvaLine."Tax Tax Amount" = 0 THEN BEGIN
// SalesLine.VALIDATE(Amount , SalesLine."Line Amount" - SalesLine."Inv. Discount Amount");
// SalesLine.VALIDATE("VAT Base Amount" , SalesLine.Amount);
// SalesLine.VALIDATE("Amount Including VAT" , SalesLine.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 ELSE BEGIN
// SalesLine.VALIDATE(Amount, SalesLine."Line Amount" - SalesLine."Inv. Discount Amount");
// SalesLine.VALIDATE("VAT Base Amount" , SalesLine.Amount);
// SalesLine.VALIDATE("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;
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.VALIDATE("Amount Including VAT",LAvaLine."Tax Tax Amount" + LAvaLine."Total Taxable
Amount" + LAvaLine."Tax Exempt Amount");
SalesLine."VAT %":=0;
SalesLine."Line Amount":=0;
END;
//AVANA7.00.05.05 end
SalesLine.MODIFY;
END;

```


Release: Microsoft NAV 7.00.33781.05.06

Details

Fixed the issue related Sales Quotes, calculates tax will be calculate based on Quantity irrespective of AvaTax configuration setting for “At F7 Statistics” option.

Codeunit 14073301 AVA Tax Engine

Changes in function → fnCalcTax

New code (Search for below 1st line and Modified code marked in red color)

```

IF NOT AvaReleaseStatisticsBln THEN BEGIN
//AVANA7.00.05.06 start
IF SalesLine."Document Type" = SalesLine."Document Type" ::Quote 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",<');
    QtyPerLine:=SalesLine.Quantity;
END
ELSE BEGIN
//AVANA7.00.05.06 end
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; //AVANA7.00.05.06

END ELSE BEGIN
IF Avaconfig."Release Quantity" = '1' THEN BEGIN

```

Changes in function → fnCalcTaxBatch

New code (Search for below 1st line and Modified code marked in red color)

```

IF NOT (SalesHead.Ship AND SalesHead.Invoice) THEN BEGIN
//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 //CONNECT-1596
//AVANA7.00.05.06 start
IF SalesLine."Document Type" = SalesLine."Document Type" ::Quote 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",<');
QtyPerLine:=SalesLine.Quantity;

```

```

END
ELSE BEGIN
//AVANA7.00.05.06 end
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" :: "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",<'); //AVANA7.00.05.00
  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",<'); //AVANA7.00.05.00
  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",<'); //AVANA7.00.05.00
  QtyPerLine:=SalesLine."Qty. to Invoice";
END;
END; //AVANA7.00.05.06
END ELSE BEGIN
  IF Avaconfig."Release Quantity" = '1' THEN BEGIN

```

Changes in function → fnServiceCalcTax

New code (Search for below 1st line and Modified code marked in red color)

```

//AVANA7.00.05.00
IF ServiceHead."Currency Code" = " THEN
  RecCurrency.InitRoundingPrecision
ELSE BEGIN
  ServiceHead.TESTFIELD("Currency Factor");
  RecCurrency.GET(ServiceHead."Currency Code");
  RecCurrency.TESTFIELD("Amount Rounding Precision");
END;

//AVANA7.00.05.06 start
IF ServiceLine."Document Type" = ServiceLine."Document Type" ::Quote 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",<');
  QtyPerLine:=ServiceLine.Quantity;
END
ELSE BEGIN
//AVANA7.00.05.06 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",<'); //AVANA7.00.05.00
  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",<'); //AVANA7.00.05.00
    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",<'); //AVANA7.00.05.00
    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",<'); //AVANA7.00.05.00
  QtyPerLine:=ServiceLine."Qty. to Invoice";
END;

END; //AVANA7.00.05.06

IF TDocType=1 THEN BEGIN

```

Changes in function → fnServiceCalcTaxBatch

Remove complete function from codeunit

Changes in function → fnServCalcTaxBatch

New code (Search for below 1st line and Modified code marked in red color)

```

//AVANA7.00.05.00
IF ServiceHead."Currency Code" = " THEN
  RecCurrency.InitRoundingPrecision
ELSE BEGIN
  ServiceHead.TESTFIELD("Currency Factor");
  RecCurrency.GET(ServiceHead."Currency Code");
  RecCurrency.TESTFIELD("Amount Rounding Precision");
END;
IF NOT AvaReleaseStatisticsBln THEN BEGIN //CONNECT-1596

//AVANA7.00.05.06 start
IF ServiceLine."Document Type" = ServiceLine."Document Type" ::Quote 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",<');
QtyPerLine:=ServiceLine.Quantity;
END
ELSE BEGIN
//AVANA7.00.05.06 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",<'); //AVANA7.00.05.00
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",<'); //AVANA7.00.05.00
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",<'); //AVANA7.00.05.00
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",<'); //AVANA7.00.05.00
QtyPerLine:=ServiceLine."Qty. to Invoice";
END;
END; //AVANA7.00.05.06

END ELSE BEGIN
IF Avaconfig."Release Quantity" = '1' THEN BEGIN

```

Codeunit 14073310 AVA Validate Call

Changes in function → OnReleaseBatch()

New code (Search for below 1st line and Modified code marked in red color)

```

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

IF AvaConfig.FINDFIRST THEN BEGIN
IF ((AvaConfig.Services MOD 100)/10) >= 1 THEN BEGIN
//AvaTaxEngine.AvaSetFlagStatReleaseFn; //AVANA7.00.05.06

SalesLines1.SETRANGE("Document Type",SalesHead1."Document Type");
SalesLines1.SETRANGE("Document No.",SalesHead1."No.");

```