



Avalara AvaTax for Microsoft Dynamics NAV 2013 RTM

Specific Code Change Document

Version 01

Revision date: 1/21/15

Product release: Microsoft NAV 7.00.33781.04.01

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..... 2

 Object Table 14073510 AVA Release State Buffer 2

 Object Codeunit 14073301 AVA Tax Engine..... 2

 Changes in Global Text Constant: 3

 Changes in Global variables 3

 Create new function → AvaSetFlagStatReleaseFn 3

 Change to function → fnCalcTax 3

 Change to function → fnCalcTax1 5

 Change to function → OnServiceReleaseBatch 6

 Change in function → fnServiceCalcTax 7

 Change in function → fnServiceCalcTaxBatch..... 9

 Change in function → fnServCalcTaxBatch 9

 Changes in function → AvaSummarizeTaxLines 10

 Object Codeunit 14073310 AVA Validate Call..... 11

 Changes in Global variables 11

 Change to function → AvaGetTaxOrder 11

 Change to function → AvaGetTaxOrderPost 11

 Change to function → OnRelease..... 11

 Change to function → OnReleaseBatch 12

 Object Page 14073400 AVA Configuration Card 13

 Change in Version List of objects:..... 13

 Object OBJECT Page 14073523 AVA Reconciliation 13

 Change in Version List of objects:..... 13

 Change in function → FnConnect 13

 Object Page 14073531 Asynchronous AvaTax Posting..... 13

 Change in Version List of objects:..... 13

Changes in Avalara objects

Object Table 14073510 AVA Release State Buffer

Delete this Table object from Object designer

Object Codeunit 14073301 AVA Tax Engine

Change in Version List of objects:

Old Value	:	AVANA7.00.04.00
New Value	:	AVANA7.00.04.01

Changes in Global Text Constant:

Text14073212 :	TextConst	'ENU=NAV 7.00.33781.04.00'
Text14073212 :	TextConst	'ENU=NAV 7.00.33781.04.01';

Changes in Global variables

Add new variable:

AvaReleaseStatisticsBln – Boolean

Remove existing variable:

AvaTempReleaseStateBuf - Record - 14073510*Create new function → AvaSetFlagStatReleaseFn*

New Code

```
AvaReleaseStatisticsBln := TRUE;
```

Change to function → fnCalcTax

Search for below lines

```
TotalDisc:=0;
AvaWindowClose();
```

```
Disc:=FALSE;
SalesLine.SETRANGE(SalesLine."Document Type",SalesHeaderType);
SalesLine.SETRANGE(SalesLine."Document No.",SalesHeaderNo);
SalesLine.SETFILTER(SalesLine.Type, '<>%1', SalesLine.Type::" ");
```

Remove below Red colored lines

```
AvaTempReleaseStateBuf.RESET;
IF AvaTempReleaseStateBuf.FIND('-') THEN BEGIN
  AvaTempReleaseStateBuf.FINDFIRST;
END;//1361
```

```
IF SalesLine.FIND('-') THEN BEGIN
  REPEAT
    IF NOT((SalesLine."Qty. to Ship"=0) AND (SalesLine."Qty. to Invoice"=0)) OR
      ((Avaconfig."Statistics Quantity" = '1') AND NOT AvaReleaseStatisticsBln) THEN BEGIN//ERP CON-1361
      Ref1:="";Ref2:="";G\LAccNo.:="";TaxabilityCode:="";
      LineAmtPassed:=0;QtyPerLine:=0;
      ShipMethodLine:=ShipMethod;
      IF STRPOS(SalesLine."Tax Area Code",Text14073208) <> 0 THEN
        ShipMethodLine:=2;
```

Apply below code change

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

IF NOT AvaReleaseStatisticsBln THEN BEGIN**Remove below colored lines**

```
AvaTempReleaseStateBuf.RESET;
IF AvaTempReleaseStateBuf.FIND('-') THEN BEGIN
```

```

AvaTempReleaseStateBuf.FINDFIRST;
END;
IF AvaTempReleaseStateBuf."Release State" = FALSE THEN BEGIN

  IF Avaconfig."Statistics Quantity" = '1' THEN
  BEGIN
    LineAmtPassed:=ROUND((SalesLine.Quantity*UnitPrice) -
      ROUND((SalesLine.Quantity * UnitPrice * SalesLine."Line Discount %")/100));//CONNECT-918
    QtyPerLine:=SalesLine.Quantity;
  END ELSE IF Avaconfig."Statistics Quantity" = '2' THEN BEGIN
  IF (SalesLine."Document Type" = SalesLine."Document Type" :: "Return Order") OR
    (SalesLine."Document Type" = SalesLine."Document Type" :: "Credit Memo") THEN BEGIN
    LineAmtPassed:=ROUND((SalesLine."Return Qty. to Receive"*UnitPrice) -
      ROUND((SalesLine."Return Qty. to Receive" * UnitPrice * SalesLine."Line Discount
%/100));//CONNECT-918
    QtyPerLine:=SalesLine."Return Qty. to Receive";
  END ELSE BEGIN
    LineAmtPassed:=ROUND((SalesLine."Qty. to Ship"*UnitPrice) -
      ROUND((SalesLine."Qty. to Ship" * UnitPrice * SalesLine."Line Discount %")/100));//CONNECT-918
    QtyPerLine:=SalesLine."Qty. to Ship";
  END;
  END ELSE IF Avaconfig."Statistics Quantity" = '3' THEN BEGIN
  LineAmtPassed:=ROUND((SalesLine."Qty. to Invoice"*UnitPrice) -
    ROUND((SalesLine."Qty. to Invoice" * UnitPrice * SalesLine."Line Discount %")/100));//CONNECT-918
    QtyPerLine:=SalesLine."Qty. to Invoice";
  END;

  END ELSE BEGIN
  IF Avaconfig."Release Quantity" = '1' THEN
  BEGIN
    LineAmtPassed:=ROUND((SalesLine.Quantity*UnitPrice) -
      ROUND((SalesLine.Quantity * UnitPrice * SalesLine."Line Discount %")/100));//CONNECT-918
    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) -
      ROUND((SalesLine."Return Qty. to Receive" * UnitPrice * SalesLine."Line Discount
%/100));//CONNECT-918
    QtyPerLine:=SalesLine."Return Qty. to Receive";
  END ELSE BEGIN
    LineAmtPassed:=ROUND((SalesLine."Qty. to Ship"*UnitPrice) -
      ROUND((SalesLine."Qty. to Ship" * UnitPrice * SalesLine."Line Discount %")/100));//CONNECT-918
    QtyPerLine:=SalesLine."Qty. to Ship";
  END;
  END ELSE IF Avaconfig."Release Quantity" = '3' THEN BEGIN
  LineAmtPassed:=ROUND((SalesLine."Qty. to Invoice"*UnitPrice) -
    ROUND((SalesLine."Qty. to Invoice" * UnitPrice * SalesLine."Line Discount %")/100));//CONNECT-918
    QtyPerLine:=SalesLine."Qty. to Invoice";
  END;
  END;

  Remove below colored lines
  AvaTempReleaseStateBuf.DELETEALL;
  COMMIT;

```

```

CLEAR(AvaTempReleaseStateBuf);

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;

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,
    oAddFromLine,
    oAddFinalToLine,
    ShipMethodLine,
    SalesLine."Line No.",
    SalesLine."No.",
    SalesLine.Description,
    QtyPerLine,
    LineAmtPassed,
    Disc,
    ExemptionNo,
    Ref1,
    Ref2,
    "G\LAccNo.",
    TaxibilityCode,
    EntityUseCode,
    oTaxLines,
    OutStatus);
  AvaWindowClose(); ;
  END;//ERPCON-1316
UNTIL SalesLine.NEXT = 0;
END;

Remove below colored lines
AvaTempReleaseStateBuf.DELETEALL;
COMMIT;
CLEAR(AvaTempReleaseStateBuf);

IF TotalDisc <> 0 THEN

```

Change to function → fnCalcTax1

Search for below lines

```
AvaWindowClose();
```

```
TotalDisc:=0;
```

Remove below colored line

```
AvaTempReleaseStateBuf.RESET;
IF AvaTempReleaseStateBuf.FIND('-') THEN BEGIN
  AvaTempReleaseStateBuf.FINDFIRST;
END;
```

```
Disc:=FALSE;
```

Change to function → OnServiceReleaseBatch

Old Code

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

IF AvaConfig.FINDFIRST THEN BEGIN
  IF ((AvaConfig.Services MOD 100)/10) >= 1 THEN BEGIN
    IF AvaRelStateBuf.FIND('-') THEN BEGIN
      AvaRelStateBuf.DELETE;
      COMMIT;
      END;
      AvaRelStateBuf.INIT;
      AvaRelStateBuf.VALIDATE(AvaRelStateBuf."Release State", TRUE);
      AvaRelStateBuf.INSERT();

      ServiceLines1.SETRANGE(ServiceLines1."Document Type",ServiceHead1."Document Type");
```

New Code

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

IF AvaConfig.FINDFIRST THEN BEGIN
  IF ((AvaConfig.Services MOD 100)/10) >= 1 THEN BEGIN
    AvaTaxEngine.AvaSetFlagStatReleaseFn;
    ServiceLines1.SETRANGE(ServiceLines1."Document Type",ServiceHead1."Document Type");
```

Apply below code change

```
Disc:=FALSE;
IF SalesLine."Inv. Discount Amount" <> 0 THEN
  Disc:=TRUE
ELSE
  Disc:=FALSE;
IF NOT (SalesHead.Ship AND SalesHead.Invoice) THEN
  BEGIN
    IF NOT AvaReleaseStatisticsBln THEN BEGIN //AVANA7.08.05.01
    Remove below colored line
    IF AvaTempReleaseStateBuf.FIND('-') THEN BEGIN
      AvaTempReleaseStateBuf.FINDFIRST;
    END;
    IF AvaTempReleaseStateBuf."Release State" = FALSE THEN BEGIN
      IF Avaconfig."Statistics Quantity" = '1' THEN
```

```

BEGIN
  LineAmtPassed:=ROUND((SalesLine.Quantity*UnitPrice) -
    ((SalesLine.Quantity * UnitPrice * SalesLine."Line Discount %")/100));
  QtyPerLine:=SalesLine.Quantity;
END ELSE IF Avaconfig."Statistics Quantity" = '2' THEN BEGIN
  IF (SalesLine."Document Type" = SalesLine."Document Type" :: "Credit Memo") OR

```

Apply below changes

```

UNTIL SalesLine.NEXT = 0;
END;
Remove below colored line
AvaTempReleaseStateBuf.DELETEALL;
COMMIT;

IF TotalDisc <> 0 THEN
  oGetTaxRequest.Discount:=TotalDisc*Dcoefficient;

```

Change in function → fnServiceCalcTax

Apply below changes

```

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

Remove below colored line
IF AvaTempReleaseStateBuf.FIND('-') THEN BEGIN
  AvaTempReleaseStateBuf.FINDFIRST;
END;

IF Avaconfig."Statistics Quantity" = '1' THEN
  BEGIN
    LineAmtPassed:=ROUND((ServiceLine.Quantity*UnitPrice) -
      ((ServiceLine.Quantity * UnitPrice * ServiceLine."Line Discount %")/100));
    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((ServiceLine."Qty. to Invoice"*UnitPrice) -
      ((ServiceLine."Qty. to Invoice" * UnitPrice * ServiceLine."Line Discount %")/100));
    QtyPerLine:=ServiceLine."Qty. to Invoice";
  END ELSE BEGIN
    LineAmtPassed:=ROUND((ServiceLine."Qty. to Ship"*UnitPrice) -
      ((ServiceLine."Qty. to Ship" * UnitPrice * ServiceLine."Line Discount %")/100));
    QtyPerLine:=ServiceLine."Qty. to Ship";
  END;
END ELSE IF Avaconfig."Statistics Quantity" = '3' THEN BEGIN
  LineAmtPassed:=ROUND((ServiceLine."Qty. to Invoice"*UnitPrice) -
    ((ServiceLine."Qty. to Invoice" * UnitPrice * ServiceLine."Line Discount %")/100));
  QtyPerLine:=ServiceLine."Qty. to Invoice";
END;
Remove below colored line
AvaTempReleaseStateBuf.DELETEALL;
COMMIT;

```

```
IF TDocType=1 THEN BEGIN  
  IF QtyPerLine < 0 THEN
```


Change in function → fnServiceCalcTaxBatch

Apply below changes

```

Disc:=FALSE;
IF SalesLine."Inv. Discount Amount" <> 0 THEN
  Disc:=TRUE
ELSE
  Disc:=FALSE;
IF NOT (SalesHead.Ship AND SalesHead.Invoice) THEN
BEGIN
  IF NOT AvaReleaseStatisticsBln THEN BEGIN //AVANA7.08.05.01
Remove below colored line
  IF AvaTempReleaseStateBuf.FIND('-') THEN BEGIN
    AvaTempReleaseStateBuf.FINDFIRST;
  END;
  IF AvaTempReleaseStateBuf."Release State" = FALSE THEN BEGIN
  IF Avaconfig."Statistics Quantity" = '1' THEN
  BEGIN
    LineAmtPassed:=(SalesLine.Quantity*UnitPrice) -
      ROUND((SalesLine.Quantity * UnitPrice * SalesLine."Line Discount %")/100);

```

Apply below changes

```

  QtyPerLine:=SalesLine."Qty. to Invoice";
  END;
  END;
Remove below colored line
  AvaTempReleaseStateBuf.DELETEALL;
  COMMIT;
  END;

  IF TDocType=1 THEN BEGIN
  IF QtyPerLine < 0 THEN
    QtyPerLine:=QtyPerLine*-1;

```

Change in function → fnServCalcTaxBatch

Apply below changes

```

Disc:=FALSE;
IF ServiceLine."Inv. Discount Amount" <> 0 THEN
  Disc:=TRUE
ELSE
  Disc:=FALSE;
IF NOT AvaReleaseStatisticsBln THEN BEGIN //AVANA7.08.05.01
//IF AvaTempReleaseStateBuf."Release State" = FALSE THEN BEGIN
  IF Avaconfig."Statistics Quantity" = '1' THEN
  BEGIN
    LineAmtPassed:=ROUND((ServiceLine.Quantity*UnitPrice) -
      ((ServiceLine.Quantity * UnitPrice * ServiceLine."Line Discount %")/100));
    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((ServiceLine."Qty. to Invoice"*UnitPrice) -
      ((ServiceLine."Qty. to Invoice" * UnitPrice * ServiceLine."Line Discount %")/100));
    QtyPerLine:=ServiceLine."Qty. to Invoice";

```

```

END ELSE BEGIN
LineAmtPassed:=ROUND((ServiceLine."Qty. to Ship"*UnitPrice) -
    ((ServiceLine."Qty. to Ship" * UnitPrice * ServiceLine."Line Discount %")/100));
QtyPerLine:=ServiceLine."Qty. to Ship";
END;
END ELSE IF Avaconfig."Statistics Quantity" = '3' THEN BEGIN
LineAmtPassed:=ROUND((ServiceLine."Qty. to Invoice"*UnitPrice) -
    ((ServiceLine."Qty. to Invoice" * UnitPrice * ServiceLine."Line Discount %")/100));
QtyPerLine:=ServiceLine."Qty. to Invoice";
END;

END ELSE BEGIN
IF Avaconfig."Release Quantity" = '1' THEN
BEGIN
LineAmtPassed:=ROUND((ServiceLine.Quantity*UnitPrice) -
    ((ServiceLine.Quantity * UnitPrice * ServiceLine."Line Discount %")/100));
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((ServiceLine."Qty. to Invoice"*UnitPrice) -
    ((ServiceLine."Qty. to Invoice" * UnitPrice * ServiceLine."Line Discount %")/100));
QtyPerLine:=ServiceLine."Qty. to Invoice";
END ELSE BEGIN
LineAmtPassed:=ROUND((ServiceLine."Qty. to Ship"*UnitPrice) -
    ((ServiceLine."Qty. to Ship" * UnitPrice * ServiceLine."Line Discount %")/100));
QtyPerLine:=ServiceLine."Qty. to Ship";
END;
END ELSE IF Avaconfig."Release Quantity" = '3' THEN BEGIN
LineAmtPassed:=ROUND((ServiceLine."Qty. to Invoice"*UnitPrice) -
    ((ServiceLine."Qty. to Invoice" * UnitPrice * ServiceLine."Line Discount %")/100));
QtyPerLine:=ServiceLine."Qty. to Invoice";
END;
END;
//AvaTempReleaseStateBuf.DELETEALL;
//COMMIT;

IF TDocType=1 THEN BEGIN
IF QtyPerLine < 0 THEN

```

Changes in function → *AvaSummarizeTaxLines*

Changes in code highlighted in color

```

TempSalesTaxAmtLine.DELETEALL;

"AvaLine 1".RESET;
TaxExemptAmount:=0;
//IF Opt = 2 THEN //AVANA7.08.05.01
//"AvaLine 1".SETFILTER("Quantity Inv.", '>0'); //AVANA7.08.05.01
"AvaLine 1".SETFILTER("Item No.", '<>');

```

Object Codeunit 14073310 AVA Validate Call

Change in Version List of objects:

Old Value	:	AVANA7.00.04.00
New Value	:	AVANA7.00.04.01

Changes in Global variables

Remove existing variable:

AvaTempReleaseStateBuf - Record - 14073510

Change to function → AvaGetTaxOrder

Remove lines marked in Red

```

IF NOT (SalesHead.Status = SalesHead.Status::Released) THEN BEGIN
  IF NOT(SALES_TO_POST) THEN
    Result:=AvaTaxEngine.fnCalcTax(DocNum,DocType,AvaDocTypeTax)
  ELSE
    Result:=TRUE;
  IF AvaRelStateBuf.FIND('-') THEN BEGIN
    AvaRelStateBuf.DELETE;
    COMMIT;
    END;
  END ELSE BEGIN

```

Change to function → AvaGetTaxOrderPost

Remove lines marked in Red

```

IF NOT(SALES_TO_POST) THEN
  Result:=AvaTaxEngine.fnCalcTax(DocNum,DocType,TmpDocTypePost)
ELSE
  Result:=TRUE;
  IF AvaRelStateBuf.FIND('-') THEN BEGIN
    AvaRelStateBuf.DELETE;
    COMMIT;
    END;
  BEGIN
    AvaHead.SETRANGE(AvaHead."Document No.",DocNumPre);

```

Change to function → OnRelease

Old Code

```

AvaConfig.RESET;
IF AvaConfig.FINDFIRST THEN;

IF AvaConfig.FINDFIRST THEN BEGIN
  IF ((AvaConfig.Services MOD 100)/10) >= 1 THEN BEGIN
    IF AvaRelStateBuf.FIND('-') THEN BEGIN
      AvaRelStateBuf.DELETE;
      COMMIT;
      END;
    AvaRelStateBuf.INIT;
    AvaRelStateBuf.VALIDATE(AvaRelStateBuf."Release State", TRUE);
    AvaRelStateBuf.INSERT();

    IF SalesLines1.Type <> SalesLines1.Type::" " THEN

```

New Code

```

AvaConfig.RESET;
IF AvaConfig.FINDFIRST THEN;

IF AvaConfig.FINDFIRST THEN BEGIN
  IF ((AvaConfig.Services MOD 100)/10) >= 1 THEN BEGIN
    AvaTaxEngine.AvaSetFlagStatReleaseFn;
    IF SalesLines1.Type <> SalesLines1.Type::" " THEN

```

*Change to function → OnReleaseBatch***Old Code**

```

IF NOT(AvaCheckInstStatus) THEN
  EXIT(FALSE);

IF AvaConfig.FINDFIRST THEN BEGIN
  IF ((AvaConfig.Services MOD 100)/10) >= 1 THEN BEGIN
    IF AvaRelStateBuf.FIND('-') THEN BEGIN
    AvaRelStateBuf.DELETE;
    COMMIT;
    END;
    AvaRelStateBuf.INIT;
    AvaRelStateBuf.VALIDATE(AvaRelStateBuf."Release State", TRUE);
    AvaRelStateBuf.INSERT();

    SalesLines1.SETRANGE(SalesLines1."Document Type",SalesHead1."Document Type");

```

New Code

```

IF NOT(AvaCheckInstStatus) THEN
  EXIT(FALSE);

IF AvaConfig.FINDFIRST THEN BEGIN
  IF ((AvaConfig.Services MOD 100)/10) >= 1 THEN BEGIN
    AvaTaxEngine.AvaSetFlagStatReleaseFn;
    SalesLines1.SETRANGE(SalesLines1."Document Type",SalesHead1."Document Type");

```

Object Page 14073400 AVA Configuration Card*Change in Version List of objects:*

Old Value	:	AVANA7.00.04.00
New Value	:	AVANA7.00.04.01

Changes in Variable

Text14073211	:	TextConst	NAV 7.00.33781.04.01
--------------	---	-----------	----------------------

Object OBJECT Page 14073523 AVA Reconciliation*Change in Version List of objects:*

Old Value	:	AVANA7.00.04.00
New Value	:	AVANA7.00.04.01

Change in function → FnConnect

Old Value

oTaxSvc.Profile.Client:='NAV 7.10.35473.04.00';

New Value

oTaxSvc.Profile.Client:= 'NAV 7.00.33781.04.01';
--

Object Page 14073531 Asynchronous AvaTax Posting*Change in Version List of objects:*

Old Value	:	AVANA7.00.04.00
New Value	:	AVANA7.00.04.01