

## APPENDIX B: CUSTOMIZING ITEM TRACKING REPORTS

This appendix describes how to implement new record types in the generic Item Tracking Appendix report and how to change existing reports to show serial/lot numbers in Microsoft Dynamics® NAV 2009.

### Generic Report (Item Tracking Appendix)

Report 6521, Item Tracking Appendix, is used with reports called for the following record types (sources):

- Purchase Header
- Sales Header
- Sales Shipment Header
- Sales Invoice Header

Handling several record types (sources) can be implemented as follows.

1. Adjust the function **RetrieveDocumentItemTracking** in Codeunit 6500, Item Tracking Management, by:
  - Creating a new variable for the new record (header and line; for instance Sales Header and Sales Line)
  - Implementing a new procedure collecting Item Tracking for this record when the function is not implemented earlier
2. Complete the following steps in Report 6521:
  - Expand the global variable DocType with the new option string for the new document (such as Sales Quote, Sales Order, and so on.), which also will be shown on the request form
  - Adjust the code of the existing procedures handling the different sources:
    - SetRecordFilter
    - HandleRec
  - Create new procedures handling the new source such as:
    - HandleSales
    - FilterSalesHdr
  - Create a new procedure preparing the address of the new source such as:
    - AddressSalesHdr

The functions in the following table are created to collect item tracking data from the different Item Tracking tables and will also be available in Codeunit 6500, Item Tracking Management, as global functions.

Function Name	Function Description
FindReservEntries	Searches through item tracking data for a source document in Reservation Entry for Quote, Order, Invoice, and Credit Memo, which are not completely invoiced yet.
FindTrackingEntries	Searches through item tracking data for a source document in Tracking Specification for Quote, Order, Invoice, and Credit Memo, which are not completely invoiced yet.
FindItemEntries	Searches through item tracking data for a source document in Item Entry Relation for Posted Shipments/Receipts.
FindValueEntries	Searches through item tracking data for a source document in Value Entry Relation for Posted Invoices.

The functions in the following table are defined in Codeunit 6500, Item Tracking Management, as local functions.

Function Name	Function Description
FillTrackingSpecBufferFromILE	Creates an item tracking buffer line with information from Item Ledger Entries. (Called by FindItemEntries and FindValueEntries.)
FillTrackingSpecBuffer	Creates an item tracking buffer line.
ItemTrackingExistsInBuffer	Checks the temporary Item Tracking Buffer for existing records and creates a new one or sums the quantity if the record exists.
InitTrackingSpecBuffer	Initializes a new item tracking buffer record.
FilterReservEntries	Retrieves a data set of Reservation Entries.
FilterTrackingEntries	Retrieves a data set of Tracking Specification Entries.
RetrieveILEFromShptRcpt	Retrieves a data set of Item Ledger Entries (used for Posted Shipments/Receipts).
RetrieveILEFromPostedInv	Retrieves a data set of Item Ledger Entries (used for Posted Invoices).

---

## Appendix B: Customizing Item Tracking Reports

The functions in the following table define how the different source documents in the generic report are implemented.

Function Name	Function Description
SetRecordFilter()	Sets filters on the correct table, which are chosen by the user in the request form. Calls the following filter-procedures: <ul style="list-style-type: none"><li>• FilterSalesHdr</li><li>• FilterPurchHdr</li><li>• FilterSalesShip</li><li>• FilterSalesInv</li></ul>
HandleRec	Starts on MainRecord - OnAfterGetRecord() for every new source and will starting handling Item Tracking collecting for the different Record Types: <ul style="list-style-type: none"><li>• HandleSales()</li><li>• HandlePurchase()</li><li>• HandleShipment()</li><li>• HandleInvoice()</li></ul>
AddressSalesHdr AddressPurchaseHdr AddressShipmentHdr AddressInvoiceHdr	Sets up the address on the report for the Record Type.
IsStartNewGroup	A global function that delivers TRUE if a new group (item with Lot/Serial No.) starts. Used to assign a total quantity for all Lot/Serial No. for an item.

Using function **RetrieveDocumentItemTracking** for a source document which is not implemented will create an error message that states the table is not supported.

## Implementing Item Tracking on Existing Reports

These instructions assume that retrieving Item Tracking for a source is defined in the **RetrieveDocumentItemTracking** function in Codeunit 6500, Item Tracking Management.

Follow these steps to implement Item Tracking in an existing report. This functionality is standard in Report 208, Sales - Shipment.

1. Make changes on DataItem "PageLoop."

DataItem	Name
Sales Shipment Header	<Sales Shipment Header>
Integer	CopyLoop
Integer	PageLoop

### PageLoop - OnPreDataItem()

```
// Item Tracking:
IF ShowLotSN THEN BEGIN
    TrackingSpecCount := 0;
    OldRefNo := 0;
    ShowGroup := FALSE;
END;
```

2. Make changes on DataItem "Sales Shipment Line."

DataItem	Name
Sales Shipment Header	<Sales Shipment Header>
Integer	CopyLoop
Integer	PageLoop
Integer	DimensionLoop1
Sales Shipment Line	<Sales Shipment Line>

### Sales Shipment Line - OnPostDataItem()

```
// Item Tracking:
IF ShowLotSN THEN
    TrackingSpecCount :=
        ItemTrackingMgt.RetrieveDocumentItemTracking(TrackingSpecBuffer,
            "Sales Shipment Header"."No.",
            DATABASE::"Sales Shipment Header",0);
```

### 3. Provide new DataItems for Item Tracking.

DataItem	Name
Sales Shipment Header	<Sales Shipment Header >
Integer	CopyLoop
Integer	PageLoop
Integer	DimensionLoop1
Sales Shipment Line	<Sales Shipment Line >
Integer	DimensionLoop2
Integer	Total
Integer	Total2
Integer	ItemTrackingLine
Integer	TotalItemTracking

#### **ItemTrackingLine - OnPreDataItem()**

```

IF TrackingSpecCount = 0 THEN
  CurrReport.BREAK;
CurrReport.NEWPAGE;
SETRANGE(Number,1,TrackingSpecCount);
TrackingSpecBuffer.SETCURRENTKEY("Source ID","Source
Type","Source Subtype","Source Batch Name",
"Source Prod. Order Line","Source Ref. No.");
    
```

#### **ItemTrackingLine - OnAfterGetRecord()**

```

IF Number = 1 THEN
  TrackingSpecBuffer.FINDSET
ELSE
  TrackingSpecBuffer.NEXT;

ShowTotal := FALSE;
IF ItemTrackingAppendix.IsStartNewGroup(TrackingSpecBuffer)
THEN
  ShowTotal := TRUE;
    
```

### 4. Define the global variables.

Name	Data Type	Subtype	Length
TrackingSpecCount	Integer		
ShowLotSN	Boolean		
ShowTotal	Boolean		
ShowGroup	Boolean		
TotalQty	Decimal		
OldRefNo	Integer		
OldNo	Code		20

