

Microsoft Dynamics NAV

Creating XML Style Sheets for Data Export from Microsoft Dynamics NAV 5.0

Technical White Paper

Microsoft Dynamics™ NAV 5.0 contains a new feature which allows you to export data from Microsoft Dynamics NAV to other applications. A number of style sheets are provided on the product CD, for use when exporting data to Microsoft® Office Word or Excel®. This paper describes how you can modify the existing style sheets. It is assumed that the reader knows the AL programming language and is familiar with XML style sheets.

Date: January 2007

www.microsoft.com/dynamics/nav

Table of Contents

Introduction.....	3
Technical Description of the Feature.....	3
Creating a New Style Sheet	4
The Basic Steps to Modify an Existing Style Sheet	4
Examples	5
Adding Your Logo to a Style Sheet.....	5
Exporting Data That Is Not on the Current Form	6
Making Formatting Changes.....	8

Introduction

You can export data from all forms in Microsoft Dynamics NAV, with the exception of matrix forms, to other programs such as Microsoft® Office Word or Excel®. The data is exported as an XML file, and the file is formatted using a style sheet that has been imported into Microsoft Dynamics NAV.

There are default style sheets installed with Microsoft Dynamics NAV, which can be used to export all of the data on any form to Word or Excel. There are also style sheets for specific forms and specific purposes, such as the customer letter style sheet which you use from the customer card. You can modify the existing style sheets or create additional style sheets that are made especially for one particular form in Microsoft Dynamics NAV. You need to create a new style sheet if you want to export to a program other than Word or Excel.

The following style sheets are shipped with Microsoft Dynamics NAV:

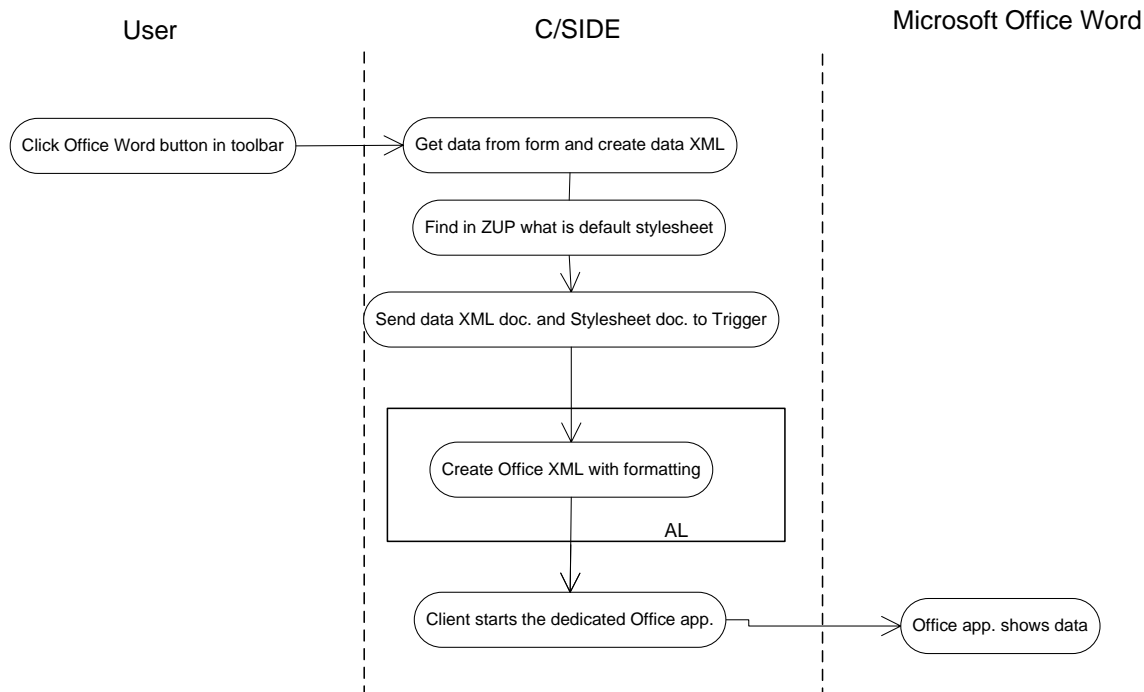
File Name	Purpose of Style Sheet	Applies to Form
NavisionFormToExcel	Creates a spread sheet in Excel that contains all of the data shown on the form.	All Forms
NavisionFormToWord	Creates a document in Word that contains a table with all of the data shown on the form.	All Forms
ContactLetterToWord.xslt	Creates a letter in Word with the contact address and your company's address filled in.	Contact Card
CustomerLetterToWord.xslt	Creates a letter in Word with the customer address and your company's address filled in.	Customer Card
SalesOrderConfirmationToWord.xslt	Creates a letter in Word with the contact address and your company's address filled in, as well as details of the sales order.	Sales Order
SalesQuoteToWord.xslt	Creates a letter in Word with the contact address and your company's address filled in, as well as details of the sales quote.	Sales Quote
VendorLetterToWord.xslt	Creates a letter in Word with the vendor address and your company's address filled in.	Vendor Card

You can find the files in a folder called \Stylesheets within the folder where your client is installed.

Technical Description of the Feature

When you click one of the Send buttons (Send to Microsoft Office Word, Send to Microsoft Office Excel, or Send Options), C/SIDE creates the *XML Data Document.xml* file. This file contains all of the data that is visible on the form you are exporting. This file is passed to codeunit 1 ApplicationManagement. Codeunit 1 triggers codeunit 403 ApplicationLaunchManagement, which generates the *XML Output Document.xml* file. C/SIDE then launches the relevant application (for example Word or Excel) which displays the XML output, formatted with the style sheet that the user has specified.

Exporting Data to Word:



Creating a New Style Sheet

In order to create a style sheet, you must first know:

1. what data is exported from Microsoft Dynamics NAV, and
2. what you want the final document to look like. This is represented by an instance document which you can create from scratch, or you can modify a document that you create using an existing style sheet.

The Basic Steps to Modify an Existing Style Sheet

You may have to create a new style sheet without any existing style sheet to base it on, for example if you are creating a style sheet for a form that does not already have a style sheet, or if you want to export to an application other than Word or Excel. Then you will have to create a new .xslt file. However, it is easier to start out with an existing style sheet and modify it. The rest of this document explains how to modify an existing style sheet.

This section gives an overview of the steps involved in creating a new XML style sheet based on the style sheets provided on the Microsoft Dynamics NAV product CD. Some of the steps will be explained in greater detail in the following sections.

I. Create an Instance Document

1. Decide which style sheet you want to modify.

In Microsoft Dynamics NAV, use the style sheet to export data from a form. This will create a document in the relevant application (for example, Word or Excel).

2. Define the layout for the document.
 - Format the document as you want it to appear.
 - Replace the CRONUS logo with the appropriate logo.
3. Save the document that you have created as XML. Then you can refer to the document to see how the transformed XML code should appear. We will refer to this as the instance document.

II. Prepare Data for Export

4. Prepare data in Microsoft Dynamics NAV for export.

If you want to export any data that is not shown on the form in Microsoft Dynamics NAV, then in codeunit 403 "Application Launch Management," add AL code to add the data to the *XML Data Document.xml* file.

You can also change text constants in codeunit 403.

III. Modify the Style Sheet

5. Locate the file for the style sheet that you used in step 1. The style sheet files are in the \Stylesheets folder, within the folder where your client is installed. Edit the style sheet in an XML editor, for example Visual Studio. (Remember to give the file a new name.)
 - To include data that you have added to the Application Launch Management codeunit, add the relevant code to the status sheet. (See example: Exporting Data That Is Not on the Current Form).
 - To replace the CRONUS logo, copy the code for the graphic from the XML document, and paste it into the style sheet. (See example: Adding Your Logo to a Style Sheet).
 - To change formatting, in Visual Studio, open the instance document (which you made in Step 3) and copy the relevant XML code to the .XSLT file. (See example: Making Formatting Changes).

Examples

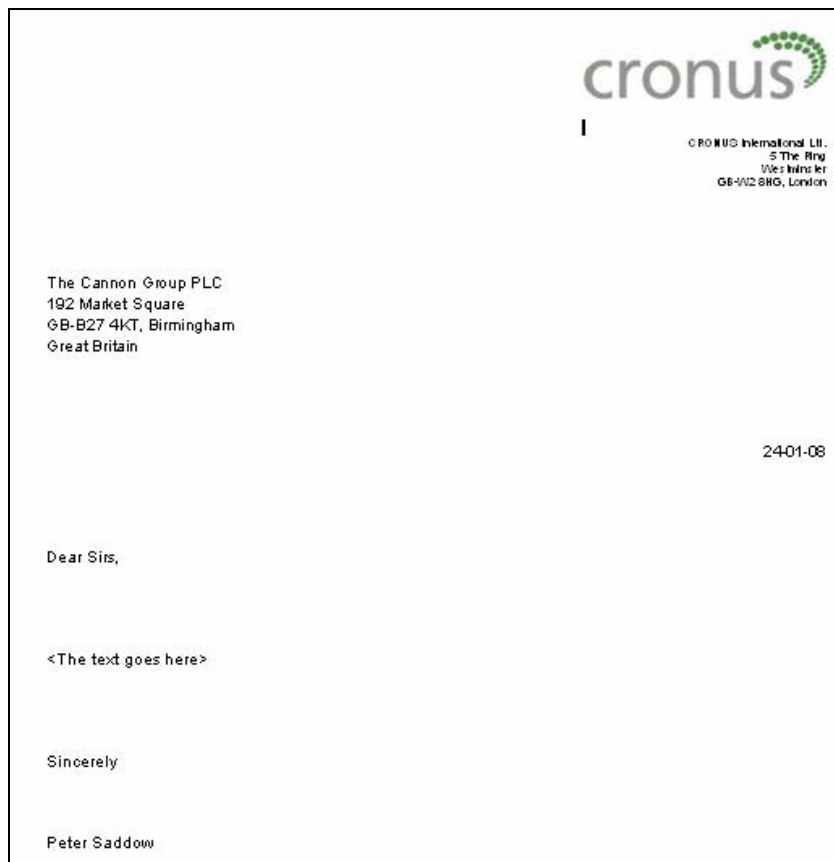
The following examples in the demonstration company illustrate some of the things that you can modify in a style sheet. The examples use the style sheet *CustomerLetterToWord.xslt*, which is one of the style sheets for the customer card that is provided with the application. These examples are for Word, however the same methods would also work to modify a style sheet for Excel.

Adding Your Logo to a Style Sheet

To replace the CRONUS logo in the *CustomerLetterToWord.xslt* style sheet with another logo, you would do as follows:

1. Open the customer card for customer number 10000 and click the Send to Microsoft Office Word button to create a Word document using the *CustomerLetterToWord.xslt* style sheet.

Word document created with CustomerLetterToWord.xslt



2. In Word, replace the CRONUS logo with the appropriate logo. Save the document as XML.
3. Open the XML file with Visual Studio. Select the text that defines the logo graphic. This is the text between the `<w:bindata>` and `</w:bindata>` tags. Copy the text.
4. Open the *CustomerLetterToWord.xslt* file in Visual Studio. Select the text between the `<w:bindata>` and `</w:bindata>` tags and paste.

Note:

The text must be entered exactly like the original – any differences, for example extra spaces, will cause the logo not to appear correctly. If a margin has been inserted, remove it.

5. Save the modified style sheet with a new name and import it into Microsoft Dynamics NAV, in the Manage Style Sheets window.

Exporting Data That Is Not on the Current Form

If you would like to include data that is not visible on the form being exported (either because it is a hidden field, or because it is from a related table) you can add it to the Application Launch Management codeunit.

In the *CustomerLetterToWord.xslt* style sheet, the salesperson's name (Peter Sadow) is included, even though this information is not on the customer card (only the salesperson code is).

To make this possible, the yellow highlighted lines have been added to codeunit 403:

Codeunit 403 modified to export Salesperson Name

```
AddCustomerLetter(VAR DataXML : Automation "Microsoft XML, v4.0'.DOMDocument40")
DocumentElement := DataXML.documentElement;

CustCode := GetAttributeValue(DataXML,'//Object/Control/Control/Row[1]/Control[2]','value');
AddCustomerInfo(DataXML,CustCode);
AddElement(DocumentElement,'WorkDate',FORMAT(WORKDATE),ChildNode);
AddElement(DocumentElement,'Salutation',FORMAT(Text003),ChildNode);
AddElement(DocumentElement,'LetterBody',FORMAT(Text004),ChildNode);
AddElement(DocumentElement,'ComplimentaryClose',FORMAT(Text005),ChildNode);
AddCompanyInfo(DataXML);

AddCustomerInfo(VAR DataXML : Automation "Microsoft XML, v4.0'.DOMDocument40";CustCode :
Code[10])
ParentNode := DataXML.documentElement;
AddElement(ParentNode,'Customer',"",ChildNode);

IF NOT Cust.GET(CustCode) THEN
    EXIT;

IF NOT Country.GET(Cust."Country/Region Code") THEN
    EXIT;

ParentNode := ChildNode;
AddElement(ParentNode,'Name',Cust.Name,ChildNode);
AddElement(ParentNode,'Address',Cust.Address,ChildNode);
AddElement(ParentNode,'PostCode',Cust."Post Code",ChildNode);
AddElement(ParentNode,'City',Cust.City,ChildNode);
AddElement(ParentNode,'Contact',Cust.Contact,ChildNode);
AddElement(ParentNode,'VATRegistrationNo',Cust."VAT Registration No.",ChildNode);
AddAttribute(ChildNode,'caption',Cust.FIELDCAPTION("VAT Registration No."));
AddElement(ParentNode,'CountryName',Country.Name,ChildNode);

IF NOT SalesPerson.GET(Cust."Salesperson Code") THEN
    EXIT;

AddElement(ParentNode,'SalespersonName',SalesPerson.Name,ChildNode);
AddAttribute(ChildNode,'caption',Text007);
```

The following lines appear in *CustomerLetterToWord.xslt* to include the salesperson name in the letter:

```
<!--Salesperson Name-->
<w:p>
  <w:pPr><w:pStyle w:val="BodyText"/></w:pPr>
  <w:r><w:t><xsl:value-of select="Object/Contact/SalespersonName"/></w:t></w:r>
</w:p>
</wx:sect>
```

Adding Data That Is Exported But Not Used in the Original Style Sheet

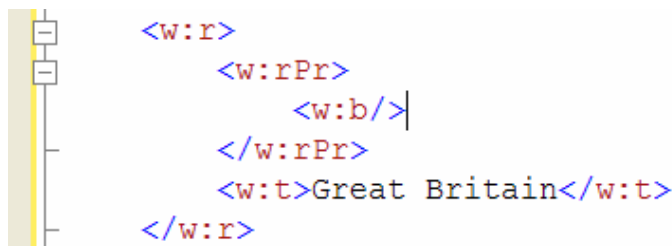
To add data that is included on the form being exported but not used by the style sheet you are modifying, you do not have to modify the Application Launch Management codeunit. Simply add lines to the .xslt file similar to those in the above snippet.

Making Formatting Changes

Even if you are not familiar with XML, you can figure out how to make minor formatting changes by making changes to your instance document and emulating them in the style sheet.

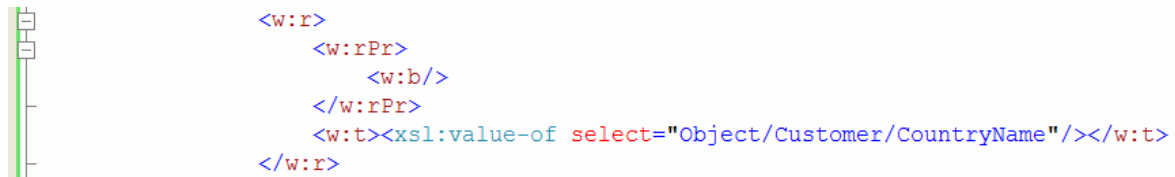
The following example illustrates how you would change the formatting of the name of the country/region in the customer address from normal to bold.

1. Open the customer card for customer number 10000 and click the Send to Microsoft Office Word button to create a Word document using the *CustomerLetterToWord.xslt* style sheet.
2. In the Word document, format Great Britain as bold, and then save the document as .xml.
3. In Visual Studio, open the XML instance document that you have created. Search for Great Britain and note how the XML tags around the text appear:



```
<w:r>
  <w:rPr>
    <w:b/>
  </w:rPr>
  <w:t>Great Britain</w:t>
</w:r>
```

4. Open the style sheet in Visual Studio and apply the same XML tags in the style sheet:



```
<w:r>
  <w:rPr>
    <w:b/>
  </w:rPr>
  <w:t><xsl:value-of select="Object/Customer/CountryName" /></w:t>
</w:r>
```

Hint:

Save a copy of the .XML instance document before you make any formatting changes. After you make the formatting changes, save the document under a different name. Compare the two documents in Visual Studio to find the differences in the XML code.

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, this document should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This White Paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

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

© 2007 Microsoft Corporation. All rights reserved.

Microsoft, The Microsoft Dynamics Logo, Excel and Visual Studio are either registered trademarks or trademarks of Microsoft Corporation or Microsoft Development Center Copenhagen ApS in the United States and/or other countries. Microsoft Development Center Copenhagen ApS is a subsidiary of Microsoft Corporation.

The Microsoft logo, consisting of the word "Microsoft" in a bold, sans-serif font.