



# IBM Company Challenge

Your challenge can be worth  
**€30,000**



Whatever the size of your company, you will always face challenges. Improving efficiency. Better customer service. Increasing market share. But what is the best solution?

Send IBM in the Netherlands your biggest company challenge and you could win €30,000 towards implementing a solution from an official IBM Business Partners (like LANSA).

IBM will chose the twenty biggest challenges and will for each of these challenges ask 3 business partners to propose a solution. If your challenge is selected as one of the twenty, you will have the opportunity to discuss the 3 proposed solutions with IBM and the respective business partners. You can then nominate the best solution to win the main prize. Then it's up to IBM's expert panel, complemented with a public jury, to select the winning case of the "IBM Company Challenge". If your case is selected, you will win €30,000 to spend on your nominated IBM business partner solution.

This competition is organized by IBM Nederland BV and is only open to Dutch companies. More information at [www.decompanychallenge.nl](http://www.decompanychallenge.nl). Challenges have to be submitted before 31 December 2010.

If you would like IBM in your country to organize a similar competition, send them the link to this page: <http://www.lansa.com/events/index.htm?tabitems=2> or use Google translate (<http://translate.google.com>) and paste the link <http://www.decompanychallenge.nl> in the translation area.

## In This Issue

<i>IBM Company Challenge</i>	<i>page 1</i>	<i>Intrinsic FieldAttributeAsString</i>	<i>page 10</i>
<i>Transaction Document Processing</i>	<i>page 2</i>	<i>DW_T01.DLL incorrectly recognized</i>	<i>page 11</i>
<i>Web Storage API</i>	<i>page 5</i>	<i>Force Landscape Printing in Windows</i>	<i>page 12</i>
<i>The Numeric "Enter" Key</i>	<i>page 6</i>	<i>Establish SSL in V12 Client/Server</i>	<i>page 19</i>
<i>Unable to load/locate VF_AC006</i>	<i>page 7</i>	<i>Services LANSA Integrator</i>	<i>page 21</i>
<i>Export list not created</i>	<i>page 9</i>		

---

---

# ***Transaction Document Processing LANSA Composer***

LANSA Composer's transport, transformation and process orchestration features provide an infinitely flexible and adaptable framework that can be used to solve almost any type of business process integration challenge.

However, many typical business process integration requirements follow a similar and familiar pattern. This pattern is characterized by:

- Transaction documents (such as EDI, XML or CSV) being exchanged with trading partners.
- Each transaction document may contain one or more discreet business transactions.
- The information in the transaction document is to be mapped from or to an application database.
- Application-specific functionality is executed to process the transactions.
- Acknowledgement of the transactions being sent to the originating trading partner.

In addition to being able to implement and orchestrate this type of processing, it is typically necessary to provide a document management capability that provides visibility into the inbound and outbound document flows.

While LANSA Composer has always provided the capabilities and tools to solve this type of business process integration challenge, LANSA Composer version 3.0 adds features that further realize that potential by providing:

- A transaction document processing framework that may be modified and extended to suit specific requirements.
- The new LANSA Composer Document Manager application.

## **Transaction Document Processing Framework**

LANSA Composer version 3.0 includes a near ready-to-use transaction document processing solution in the box. If you wish to implement solutions to the typical transaction document processing scenario described above, this framework can very quickly be modified and extended to:

- Handle the specific types of transaction documents in use.
- Take account of processing requirements specific to the customer's trading and processing environment.

This framework consists of:

- New Activities:  
**DISCOVER\_DOC**, Discovers the document type for a document file.  
**DISCOVER EDI**, Discover, catalogue and validate EDI document file.  
**EDI\_SPLIT**, Split EDI document file.  
**TXDOC\_ALLOCCTRL**, Allocate one or more control numbers for outbound transaction documents.

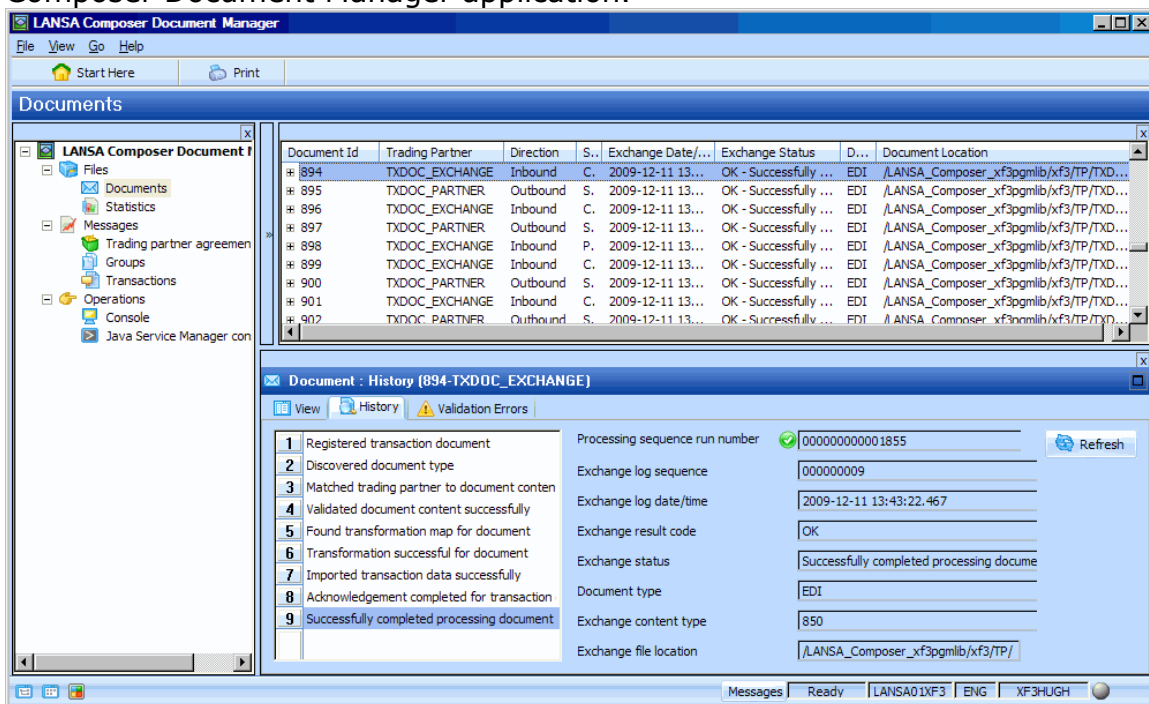
**TXDOC\_EXPORT and TXDOC\_IMPORT**, Export and import transaction data between staging files and transaction document formats.

**TXDOC\_REGISTER, TXDOC\_REGOUTBND and TXDOC\_STATUS**, Register and update the processing status of inbound and outbound transaction documents.

- Skeletal Processing Sequences.
- Pre-built Transformation Maps for selected EDI X12 transactions.
- Pre-built "staging" database tables and sample import and export processors for selected EDI X12 transactions.
- New Trading Partner and Transformation Map data interchange attributes and associated built-in variables.
- New Document types and Document standards definitions.

## LANSA Composer Document Manager

For transaction document processing solutions that make use of LANSAs Composer's built-in transaction document register (via the TXDOC\_REGISTER and TXDOC\_STATUS activities), LANSAs Composer 3.0 provides a completely new LANSAs Composer Document Manager application.



The LANSAs Composer Document Manager provides the ability to interrogate and monitor in-coming and out-going transaction document that are registered and processed using the new transaction document features, activities and other support.

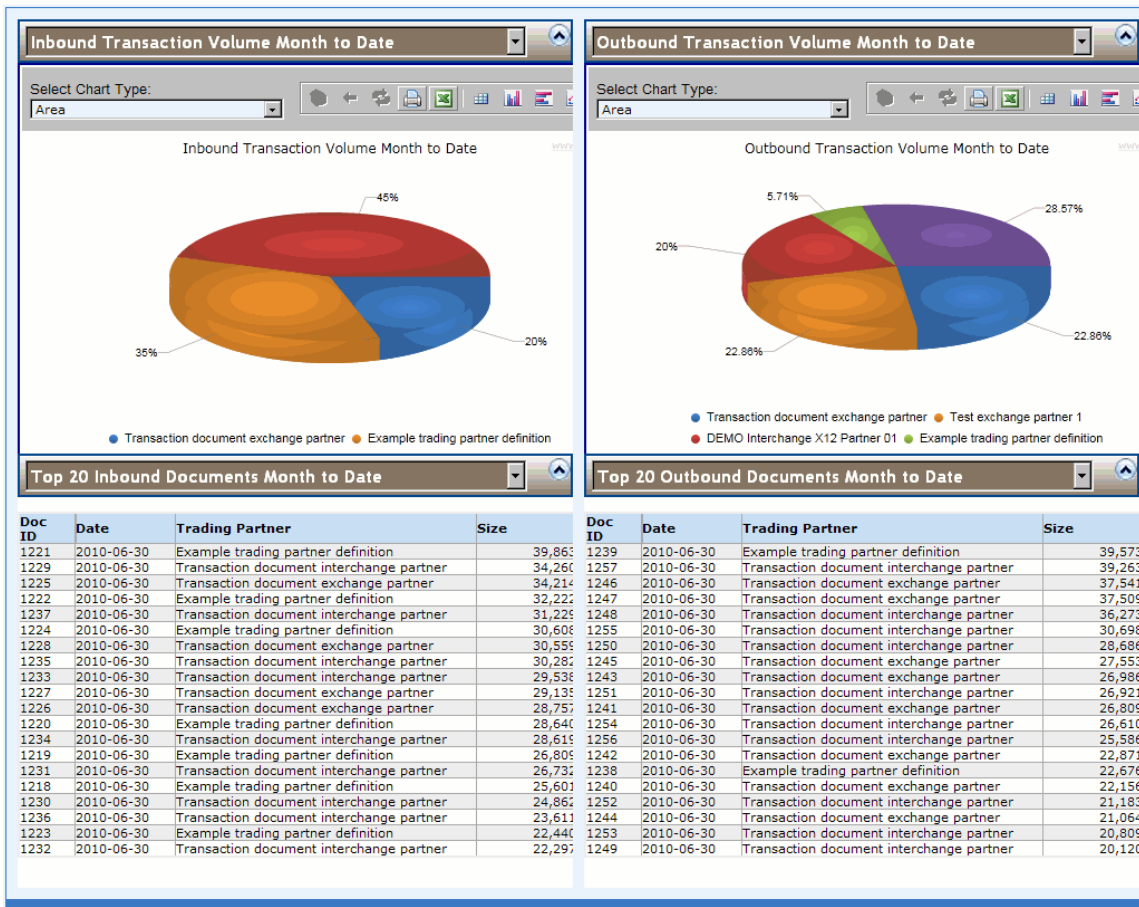
## Work with Transaction Document Statistics

LANSA Composer provides a web-browser based interface for graphically displaying management statistics about inbound and outbound transaction document flows.

To work with Transaction Document Statistics, expand Files in the Navigator and then select Statistics.

The Transaction Document Statistics page initially displays a selection of views in four panes. You can change the views by selecting from the drop-down list. You can collapse and expand the views by clicking the arrow buttons.

### Transaction Document Statistics



You can open the Transaction Document Statistics in your web browser (outside LANSAs Composer). To do this click the Open in Web Browser button. When opened in your web browser, you can add the console to your favorites or copy and send the URL to other users, such as operations staff.

---

---

# ***Web Storage API: More Security, Efficiency and Capacity than Cookies***

[http://www.webreference.com/authoring/Web\\_Storage/](http://www.webreference.com/authoring/Web_Storage/)

Web storage is a method by which a website can store data in a Web browser.

Currently, cookies are what all websites use to store client-related data. While cookies are definitely a good method for storing client-related information, they are not always efficient.

Web storage also offers a more secure method for storing client-related information.

Unlike a cookie, Web storage data is not carried over to the server via HTTP. So Web storage data can be accessed only via client-side scripts. This means the server cannot read or write directly to Web storage.

---

---

# ***The Numeric "Enter" Key can be customized to be an Enter or Tab key in V12***

V12 includes an enhancement to customize the behavior of the numeric Enter key when a user presses it.

Users may notice that in Version 12, the default behavior of the "Enter" key has changed.

This has been brought about by the fact that Visual LANSAs defaults the x\_run parameter **FLDX**, which determines whether the Numeric Keypad Enter key acts as a Field Exit key or not (i.e. treat this enter key as a Tab). In previous versions, this defaulted to Y (to be a tab key) however it has never behaved as expected. This has been rectified in V12.

## **Solution**

If you would prefer to have the Numeric Enter key behave similar to the Keyboard Enter (i.e. not to tab) then there is a simple way to customize this.

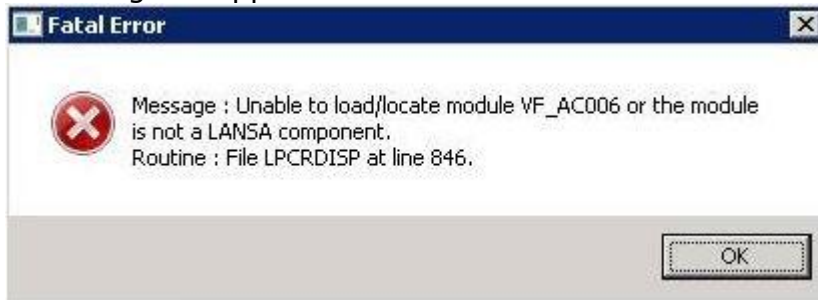
Change the x\_run parameter FLDX to N (FLDX=N) in the Start file or in the X\_LANSAs.PRO. This will give you the desired effect without any code changes or recompiling.

---

---

# ***Unable to Load/locate module VF\_AC006 error after deploying a VLF application in V12***

When deploying a Visual LANSAs Framework (VLF) application from Visual LANSAs V12, after a successful install, you may generate an error on some target PCs when launching the application:



This error can occur when the target PC does not have the Visual Studio 7 runtime executable available.

## **Note**

You may encounter variations of the error whereby other VLF DLLs may be referenced instead, for example, VF\_EXEC. Alternatively, if you have created your own VLF entry point form, this may be the component referenced in the error message.

Visual LANSAs V12 was built with Visual Studio 2008. But the version of VLF that is shipped in V12 requires Visual Studio 7.

When you create a package for an application using the Deployment Tool the VC90 runtime is included because the Visual LANSAs runtime and V12 generated components requires it.

However, when you deploy a VLF application from V12, the VLF application also requires the Visual Studio 7 runtime executables to be available on the target PCs.

On some PCs, the runtime executable is not there and this causes the VF\_AC006 error. The particular file that is being looked for is **msvcr71.dll**

## **Solution**

This Microsoft provided runtime library file, **msvcr71.dll** is freely available for download from many locations on the internet. It can even be copied from a target PC that doesn't generate the error above when launching the VLF application. In most cases it will be documented to copy the file into the C:\WINDOWS\system32 directory.

---

---

LANSAs recommendation is to place this file in the x\_win95\x\_lansa\execute directory instead of the C:\WINDOWS\system32 for the following reasons:

- To ensure that LANSAs runtime will resolve to this version of the file
- You won't copy a version of this DLL over any previous version of this file in C:\WINDOWS\system32.
- If some other software installs a version of this DLL into C:\WINDOWS\system32, the LANSAs runtime will not be affected.

### Note

As Visual LANSAs V12 was built with Visual Studio 2008, a new package setup option *Deploy with VC90 Runtime* is included in most templates as the Visual C++ 2008 Redistributable is required on the target system to perform the package installation.

A compatible version of msvc71.dll will be shipped with the next version of the VLF to avoid this situation.

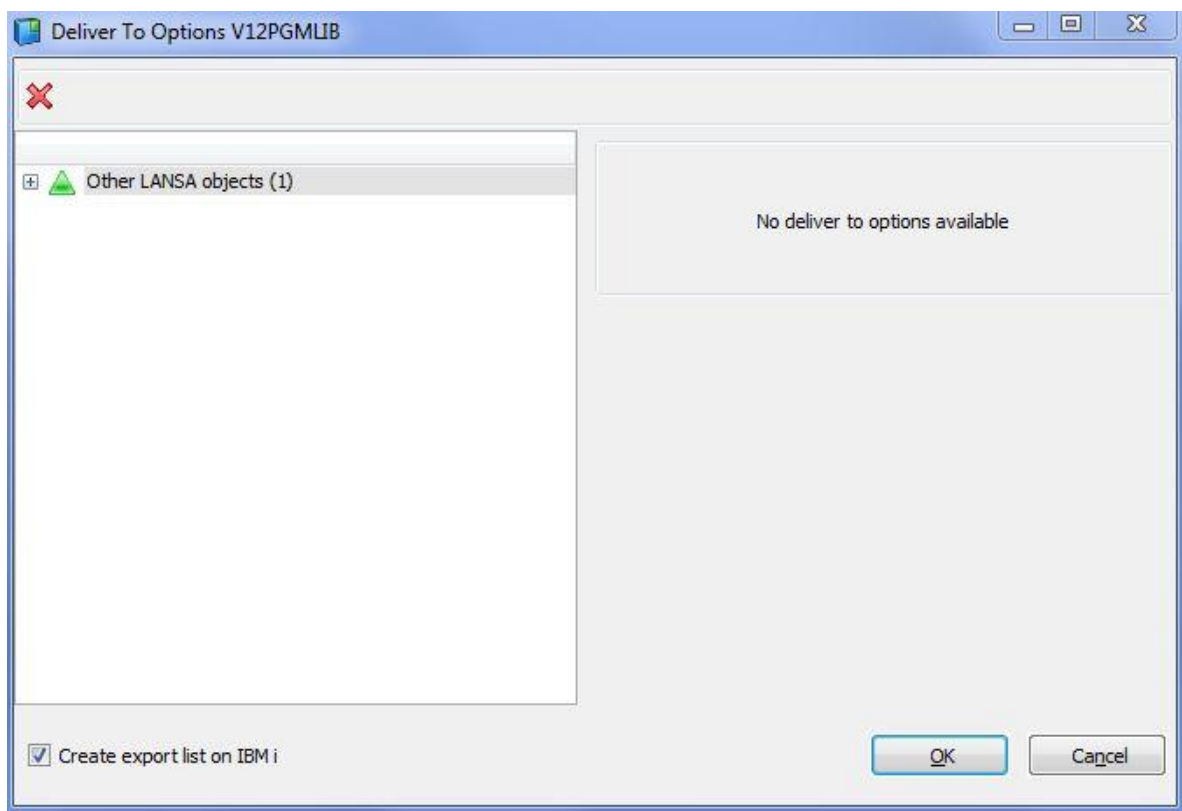


---

---

# ***Export list not created despite the option being selected in Deliver To screen***

One of the features of the Deliver To functionality in V12 is the option to create an export list on the IBM i server which contains the items that you have delivered to the remote system.



In some cases, even though the object is successfully delivered to the remote system, the export list is not created.

## **Solution**

This can occur if the remote system is not up to date with the latest PTFs. Applying the latest PTFs allows the export lists to be created.

---

---

# ***Intrinsic function FieldAttributeAsString***

Fields (PRIM\_FLD) provide a method to read various properties such as Description and Label.

The available properties include:

- Description
- Label
- Heading1
- Heading2
- Heading3
- HeadingSingleLine
- HeadingWithNewLines
- HeadingWithNewLinesNoBlanks
- EditCode
- EditWord

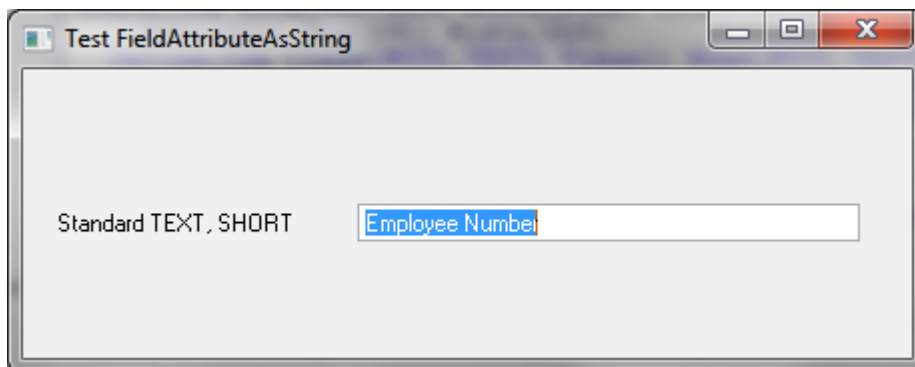
To read a property, supply the relevant property name as a parameter to the FieldAttributeAsString() method.

Example:

```
Function Options(*DIRECT)  
Begin_Com Role(*EXTENDS #PRIM_FORM) Clientheight(302) Clientwidth(653)  
Left(401) Top(195) Width(669)  
Define_Com Class(#STD_TEXTS.Visual) Name(#STD_TEXTS) Displayposition(1)  
Height(19) Left(17) Parent(#COM_OWNER) Tabposition(1) Top(67)  
Usepicklist(False) Width(401)
```

```
Evroutine Handling(#com_owner.Initialize)  
Set Com(#com_owner) Caption(*component_desc)  
#STD_TEXTS := #empno.FieldAttributeAsString( Description )  
Endroutine
```

```
End_Com
```



---

---

## ***VLF component DW\_T01.DLL is incorrectly recognized as an unwanted program by some anti-virus software***

Some anti-virus software, for example AVIRA's AntiVir, incorrectly flag a LANSAs shipped DLL (DW\_T01.DLL) as a potential virus or unwanted program during a Visual LANSAs installation. If this file is flagged and subsequently quarantined, deleted, renamed or denied access by the anti-virus software, this will cause VLF application failures.

### **Solution**

The *Installing LANSAs on Windows* Guide states the following in 2.2 Visual LANSAs General Checklist.

Before starting, you should have:

*Turned off your anti-virus software. Please leave it off while running the install/upgrade.*

To avoid potential install or execution failures, you should follow the recommendation above during a Visual LANSAs install. If the file has already been flagged and quarantined, you should follow the anti-virus software's instructions to release the file from quarantine.

---

---

# Force Landscape Printing in Windows

Using **WPPS** you can activate a Windows Printing Setup File which contains associated printer settings. These settings will be used to initialize the printer dialog. If the printer dialog is not displayed these settings will be automatically sent to the printer. The default printer settings on the PC will be used if WPPS is not specified. If WPPS is specified the default printer setting will be used for any parameters not specified in the file.

The following example shows how you can force a document to be printed Landscape (where normally Portrait is always the default value for printers). Changing the page orientation is just one of the settings you can use to initialize the printer. The end of this article shows the other options.

Let's have a look at this LANSA form. We want to print the filled list, and want to print it in Landscape:

The screenshot shows a Windows application window titled 'Verkoop merk per filiaal : (alle filialen)'. It contains several configuration panels and a data table.

**Configuration Panels:**

- Seizoenscode:** W10
- Doorlopend artikel
- 
- 
- Printer Selection:** PDFCreator (selected), WebEx Document Loader, PDFCreator, Kassabon Utrecht, Kassabon Tilburg, Kassabon Rotterdam, Kassabon Maastricht, Kassabon Heerlen, Kassabon Haarlem.
- Merk (Actief):** 100 % Hardcore, Adidas, Aktie barcode, Baby Phat, Bandana, Big And Tall, Billabong, Black top, Blend, Bling Bling, Bobby Fresh, C1rc, Cavello.
- Productlijn:** Ladies, ST, ST Ladies.
- Bepaal groep:**  Alle groepen,  Per groep(en). Groups:  Broeken,  Schoenen,  Tops.
- Bepaal artikelsoort:**  Alle artikelsoorten,  Per artikelsoort(en). Artikelsoort: (empty).
- Summary:** Totale voorraad: 417, Totaal ontvangen aantal: 3,230, Totaal aantal verkocht: 2,809, Gemiddelde doorverkoop %: 86.96.

**Data Table:**

	Aantal ontvangen	Aantal verkocht	Voorraad	Doorverkoop %
	327	280	47	85.62
	188	168	20	89.36

An important step is the selection of an existing printer IN this form:

This is a close-up of the printer selection dropdown menu from the previous screenshot. The dropdown is open, showing a list of printers. 'PDFCreator' is selected and highlighted in blue. Other visible printers include 'WebEx Document Loader', 'Kassabon Utrecht', 'Kassabon Tilburg', 'Kassabon Rotterdam', 'Kassabon Maastricht', 'Kassabon Heerlen', and 'Kassabon Haarlem'. The background shows parts of the form's configuration panels.

---

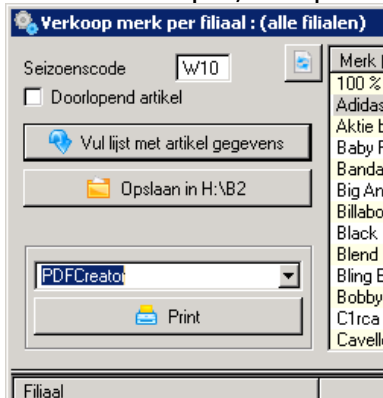
The reason for this, is that the LANSA function that creates the output, should know what printer should be initialized.

To fill a combo box (where the combo box contains field #STD\_QSEL) with a list of active printers, you can use the code:

*\* Fill combo box with active printers*

```
Define Field(#prn_name) Type(*char) Length(255)  
Define Field(#prn_loc) Type(*char) Length(1)  
Def_List Name(#PRNLIST) Fields(#PRN_NAME #PRN_LOC) Type(*WORKING)  
Use Builtin(LIST_PRINTERS) With_Args(A) To_Get(#PRNLIST #STD_CMPAR)  
Clr_List Named(#cmbx_1)  
Selectlist Named(#prnlist)  
#std_qsel := #prn_name  
Add_Entry To_List(#cmbx_1)  
Endselect
```

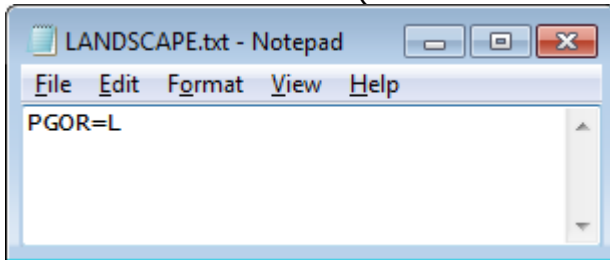
In this example, the printer PDFCreator is selected.



Using the Print button, a LANSA function is called to create the output (where field #STD\_QSEL contains the name of the selected printer):

```
Clr_List Named(#wlist2)  
* Read ListView in selected order and put data into workinglist  
For Each(#item) In(#LTVW_2.items)  
Change Field(#STD_NUM) To('#item.entry')  
Get_Entry Number(#STD_NUM) From_List(#LTVW_2)  
Add_Entry To_List(#WLIST2)  
Endfor  
* Exchange fields and CALL function  
Exchange Fields(#std_qsel)  
Call Process(*direct) Function(PR020) Pass_Lst(#wlist2)
```

To force landscape printing, we created in the partitions execute directory a text file called LANDSCAPE.TXT (but this can be any name):



The text file contains the page rotation printer setting PGOR, which can contain the values:

Value	Description
P	Portrait
L	Landscape

PGOR is just one of the available printer settings. The end of this article will show the other settings. So the text file we use (in this example LANDSCAPE.TXT, is not limited to just one printer setting, it can contain more values).

Let's go to the print function now.

The function will receive a filled workinglist to be printed and it will receive the name of the printer (in field #STD\_QSEL):

**Function Options(\*DIRECT) Rcv\_List(#wlist2)**

**Define Field(#prn\_name) Type(\*char) Length(255)**

*\* Specify text file name that contains the printer settings:*

**#prn\_name := (\*PART\_DIR\_EXECUTE + 'LANDSCAPE' + '.TXT')**

*\* Activate Windows Printing (Windows Printing Enabled, WPEN=Y):*

**Use Builtin(SET\_SESSION\_VALUE) With\_Args(WPEN Y)**

*\* Activate Windows Printing Print Dialog (WPPD=A, Automatic, means that the user is to be prompted only the first time that a report is to be printed):*

**Use Builtin(SET\_SESSION\_VALUE) With\_Args(WPPD A)**

*\* Activate Windows Printing Font Dialog (WPDF=A):*

**Use Builtin(SET\_SESSION\_VALUE) With\_Args(WPDF A)**

*\* Activate Windows Printing Default Font:*

**Use Builtin(SET\_SESSION\_VALUE) With\_Args(WPDF 'Verdana')**

*\* Activate Windows Printing Default Font Size:*

**Use Builtin(SET\_SESSION\_VALUE) With\_Args(WPDS '8')**

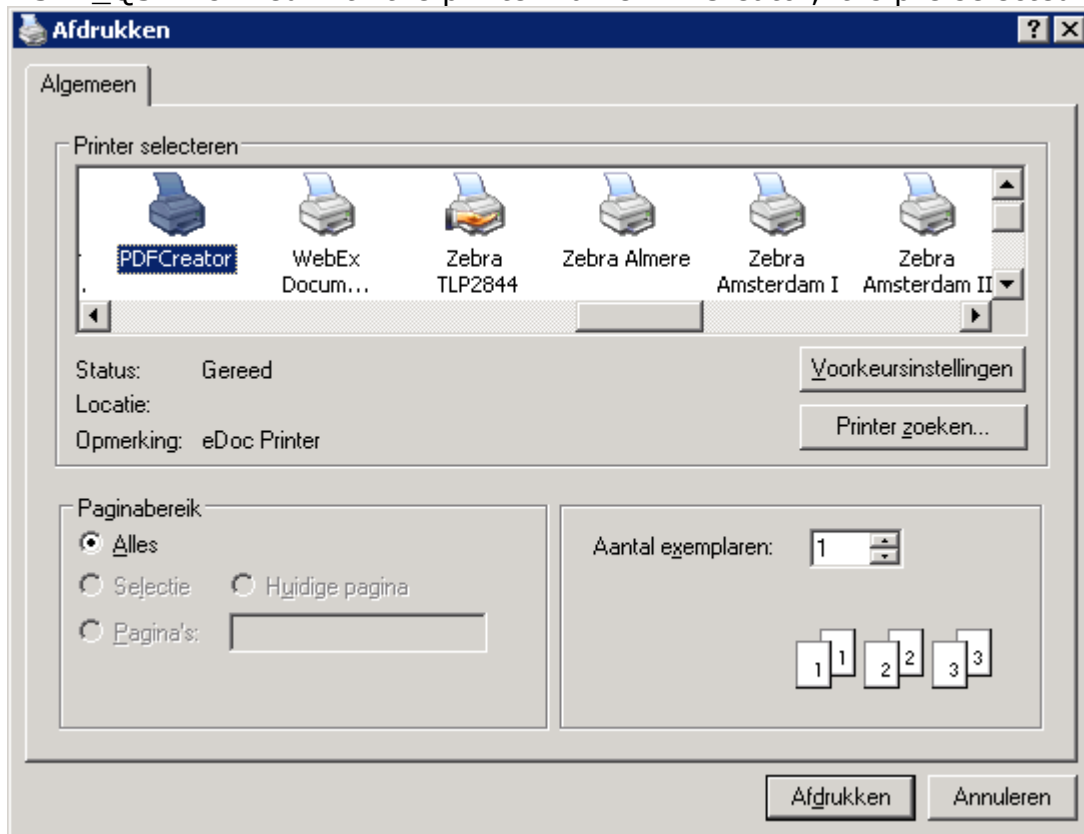
*\* Activate Windows Printing Printer Name:*

**Use Builtin(SET\_SESSION\_VALUE) With\_Args(WPPN #STD\_QSEL)**

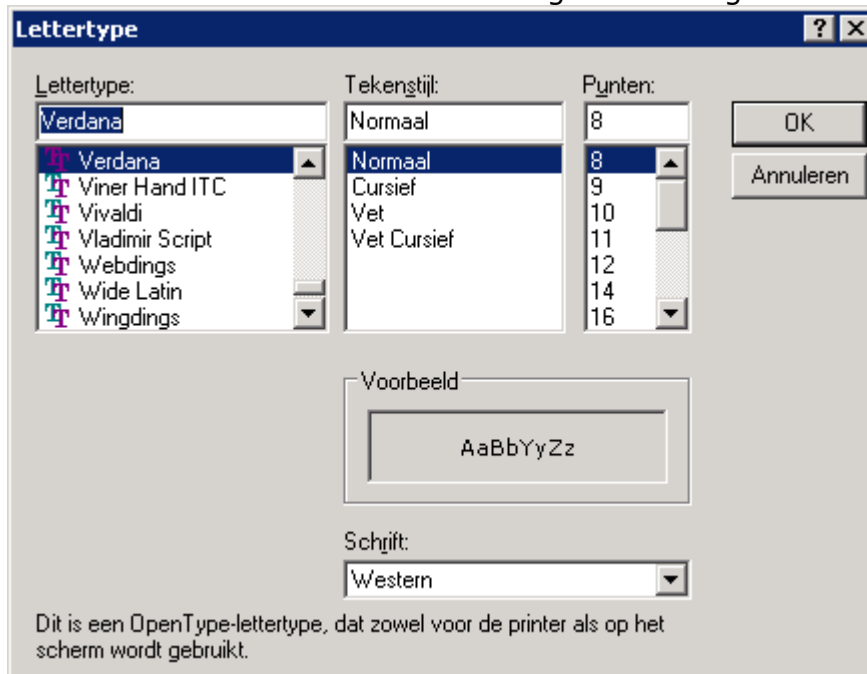
*\* Activate Windows Printing Setup File (without a filled WPPN Printer name value, the WPPS value will fail):*

**Use Builtin(SET\_SESSION\_VALUE) With\_Args(WPPS #PRN\_NAME)**

This print will result in a Windows Printing Print Dialog (WPPD). Because field #STD\_QSEL is filled with the printer name PDFCreator, it is pre selected:



It will also show the Windows Printing Font Dialog Window (WPFDF).



Because of value A (WPPD=A and WPFDF=A) these two windows will only be visible the first time that a report is to be printed. All other reports will use the settings that have been selected in these two dialogs.

---

---

In this example we did not select to print landscape. There is no need for this, because the printer setup file we used via WPPSS, forced this (PGOR=L). The final output will result in a report, printed in Landscape.

**PGOR** (Page Orientation) is just one of the settings that can be part of a Windows Printing Setup File.

The following other settings may be specified:

**DPXP= Duplex printing**

Value	Description
N	Normal (non-duplex)
H	Duplex over horizontal (flip over the long edge of the page)
V	Duplex over vertical (flip over the short edge of the page)

**COLR= Color printing**

Value	Description
C	Colour
M	Monochrome

**COPY= Number of copies**

**PSIZ= Paper Size**

Value	Description
LETTER	Letter, 8 ½ by 11-inches
LEGAL	Legal 8 ½ by 14 inches
A4	A4 sheet, 210 x 297 millimeters
A3	A3 sheet, 297 by 420 millimeters
xxx	xxx is a numeric value defined by the Microsoft Visual C runtime which represents a specific paper size. The complete list of values may be obtained from Microsoft's MSDN library at ( <a href="http://msdn.microsoft.com/library/default.asp?url=/library/en-us/intl/nls_Paper_Sizes.asp">http://msdn.microsoft.com/library/default.asp?url=/library/en-us/intl/nls_Paper_Sizes.asp</a> ).



---

---

**PSRC= Paper Source**

<b>Value</b>	<b>Description</b>
A	Auto
L	Lower tray
M	Middle tray
xxx	xxx is a numeric printer specific value. See below for instructions on how to retrieve printer specific values.

**QLTY= Print Quality**

<b>Value</b>	<b>Description</b>
H	High
M	Medium
L	Low
D	Draft
xxx	xxx is a numeric printer specific value which specifies the number of dots per inch (DPI). See below for instructions on how to retrieve printer specific values.

**COLL= Collation**

<b>Value</b>	<b>Description</b>
Y	Collate when printing multiple copies
N	Do not collate when printing multiple copies

**PTYP= Paper Type**

<b>Value</b>	<b>Description</b>
P	Plain paper
G	Glossy paper
T	Transparent film
xxx	xxx is a numeric printer specific value. See below for instructions on how to retrieve printer specific values.

---

---

**FNAM= Form Name**

<i>Examples of a printer setup file</i>	
<u>Example 1:</u> PGOR=P DPXP=V COPY=4	<u>Example 2:</u> PGOR=L DPXP=N COLR=C PSIZ=33

---

---

# ***How to establish that you are using SSL in your V12 Client Server connection***

LANSA V12 What's New contains the following item:

## *SSL Encryption*

---

*OpenSSL Encryption to IBM i for Client/Server communications is now available as a server-side option for secure encryption of network communications.*

To establish a SSL connection between LANSA on IBM i and Windows server using SuperServer connection, you need to do the following:

- change the Cryptographic algorithm from \*NONE to SSL on the "Add Communications Listener Record " screen
- restart the listener on IBM i.

But how do you establish that you are indeed using SSL in your client/server connection?

The following useful information will assist to test and ensure that you are using an SSL connection.

The first thing to keep in mind is that SSL is only supported in V12. So, one possible test to ensure that the SuperServer connection is using SSL is as follows:

## **Test 1**

- Stop the LANSA V12 listener.
- Set the socket type to be SSL for the listener record.
- Set on listener record tracing (on the same screen).
- Start the listener.
- Use the Execute a form shortcut in the VL V12 folder to execute VL\_DEM20 in local mode and then use the Connect menu option to connect to the V12 host.
- (assuming this connection is established successfully and the department/section etc. records are returned to the list from the IBM i server) Close the form.
- Use WRKLNK to go to the /LANSA\_V12pgmlib/tmp folder and open the lroute.trc.
- Search (F16) for SSL and you should see that this socket type is being used for the connection e.g. you should see the following messages in the lroute.trc:  
*Encryption Algorithm selection SSL*  
*Initialising SSL*  
*SSL ready to use*  
*Encryption Algorithm Agreed SSL*

---

---

**Test 2** (requires a V11 SP5 installation)

- use the Execute a form shortcut in a VL V11 SP5 folder to execute VL\_DEM20 in local mode and then use the Connect menu option to connect to the V12 host
- This connection should fail with a return code 6 (security not valid). The TP joblog will contain this message.

*Message . . . . : Cryptographic algorithm   < not supported.*

*Cause . . . . : Host selected cryptographic algorithm is   <. This algorithm is not supported by the client. Recovery . . . : Upgrade the clien to the same version as the host or select a cryptographic algorithm supported by the client and restart the listener.*

Both of these tests will confirm that you are making an SSL socket type connection. Please note that SSL does carry a processing overhead and may reduce transfer speeds.

---

---

# Services LANSA Integrator

LANSA Integrator enables the integration of Application-to-Application (A2A) and Business-to-Business (B2B) transactions through XML and Java services.

Using the shipped Services and the Java Service Manager, you can integrate with any system on any platform using your choice of B2B technologies such as SOAP, IBM MQ Series, XML Parsing, XSL Transformation, FTP, HTTP, HTTPS, SMTP, POP3 and AS2 protocol for HTTP. LANSA Integrator hides the complexities of interfacing with these technologies, letting you concentrate on the business requirements.

LANSA Integrator is based on a Java Service Framework that allows easy integration between Java programs and LANSA, C, RPG and COBOL applications.

Supported services at the moment are:

