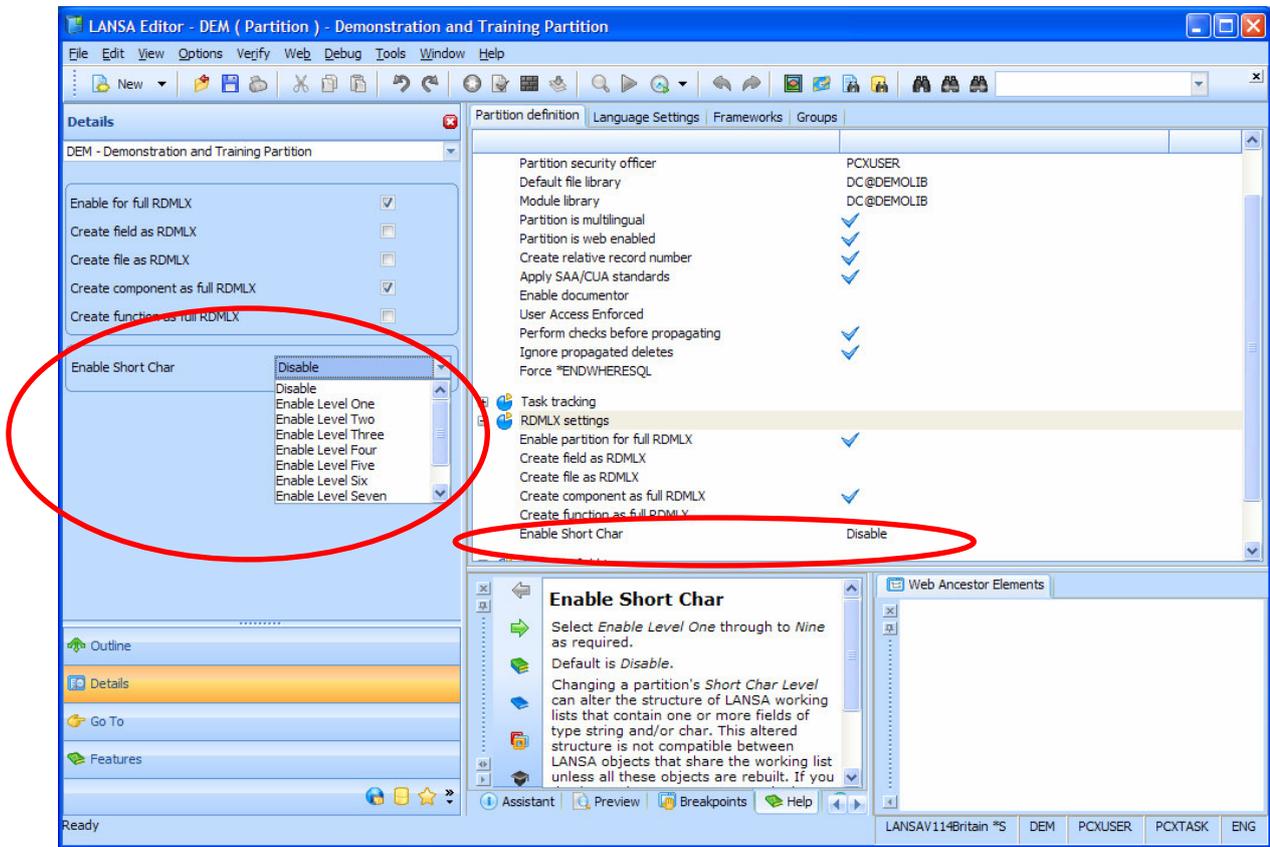


Enable short Char



This new setting at partition level indicates whether special Short Char handling should be generated to improve the runtime performance of Visual LANSAS applications using the Full RDMLX types of String and Char.

By default the Visual LANSAS runtime maintains the current value of a String/Char field by dynamically allocating a piece of memory long enough to store the field's current value.

In This Issue

Enable short Char	page 1	Firewall considerations EPC830	page 15
*ENDWHEREASQL	page 3	Unable to restore V11 SP5 objects	page 18
Field Visualisation in WAM's	page 4	Integrator licence on System i	page 19
Language CCSID mappings fetaure	page 8	*BLDNO error after applying EPC830	page 20
Problems with decimals	page 10	Support LANSAS V10 will end	page 22
Shipped defaults VLF	page 12	VLF.Net with iSeries Apache	page 23
ENCRYPT and DECRYPT BIFs	page 13	PrintScreen functionality VL	page 24

This is an efficient memory management mechanism when the length of a field's current value is generally much less than the field's defined length. On the down side, this mechanism does incur a performance overhead in order to manage the allocation and de-allocation of the piece of memory. This overhead can impact the performance of large working lists that include String/Char fields.

In order to minimize this overhead Visual LANSA has the facility to generate Short Char handling for String/Char fields. Short Char support is implemented by a single allocation of memory that can store the largest value allowed by the field's length. This feature saves the overhead of per value memory management, but this improved performance comes at the expense of a slightly larger memory allocation.

When the disabled setting is selected, Visual LANSA will treat all fields of type String and Char the same, irrespective of length.

To enable this setting, a level from one to nine is selected. Each level corresponds to a multiple of 32 such that the level multiplied by 32 derives a Short Char length. All Visual LANSA fields of type String or Char whose length is less than or equal to the Short Char length will be implemented as a Short Char. The most appropriate setting requires a judgment call that balances improved performance against increased memory usage. Further, the longer the String/Char field the greater the probability that much of the piece of memory allocated for the field's value will never be used. A reasonable balance can be achieved using a level around 2 to 4.

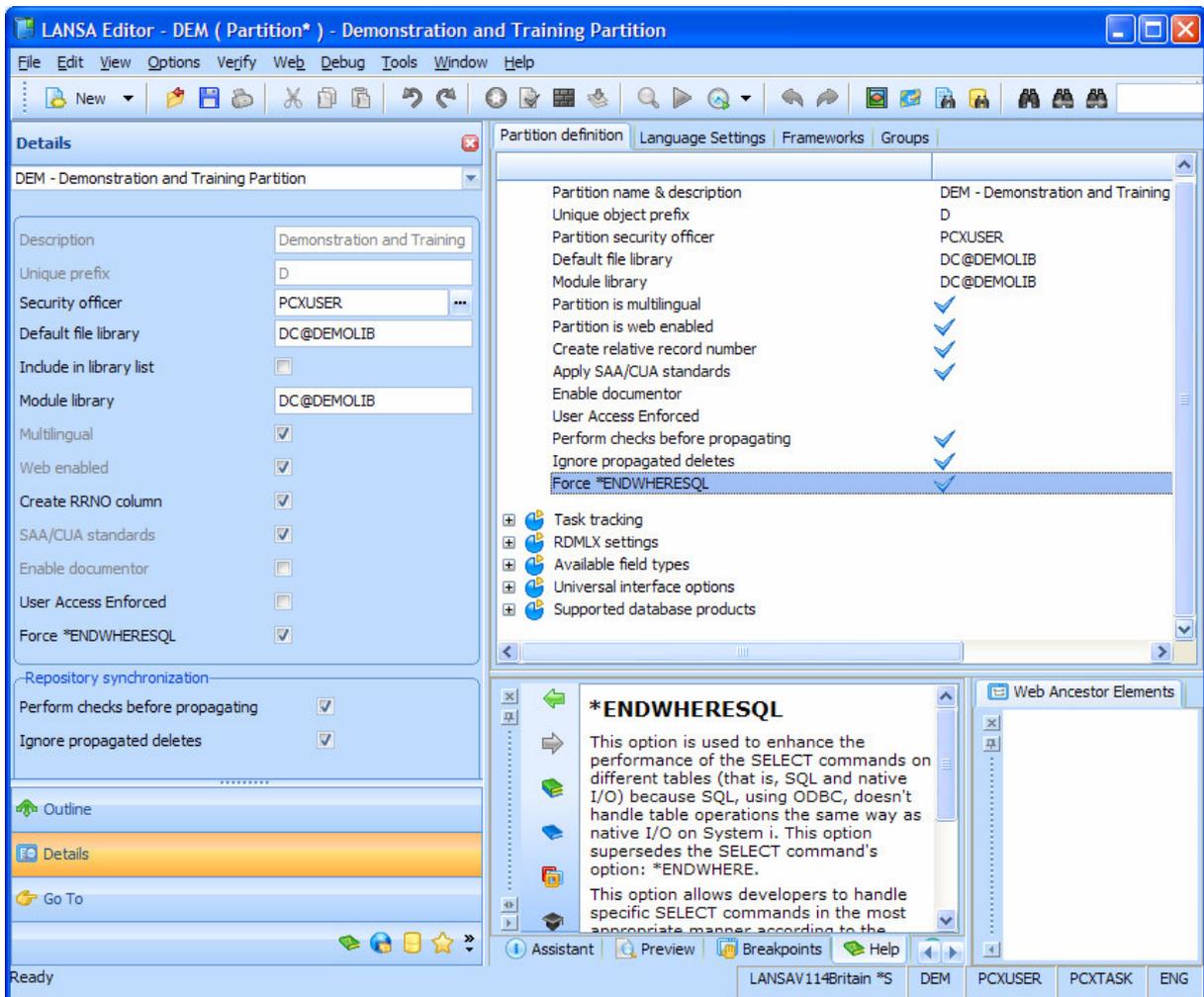
Since this setting changes the way String and Char fields are defined in generated code, this setting can impact the entry length of any working list that contains a String/Char field. Any changes to a working list's entry length will impact the compatibility of working lists that are passed to RDML Functions. Consequently, whenever this setting is changed all Visual LANSA Functions and Components should be rebuilt.

*ENDWHERE SQL

LANSA V11 SP5 has a new value *ENDWHERE SQL. This value can be used at partition level and is also a new OPTIONS parameter value of the SELECT command.

This option is used to enhance the performance of the SELECT commands on different tables (that is, SQL and native I/O) because SQL, using ODBC, doesn't handle table operations the same way as native I/O on System i.

*This option supersedes the SELECT command's option *ENDWHERE.*

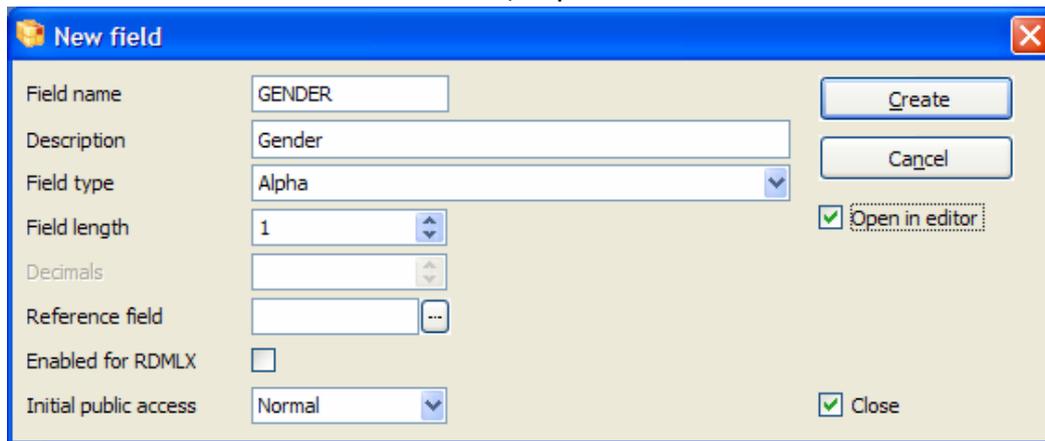


This option allows developers to handle specific SELECT commands in the most appropriate manner according to the WHERE condition. It signals the LANSAS system to interpret ALL *ENDWHERE options in SELECT commands as though *ENDWHERE SQL had been coded. It is recommended that, as code is updated or new code is written, that the SELECT commands are changed to use this new option where it is appropriate.

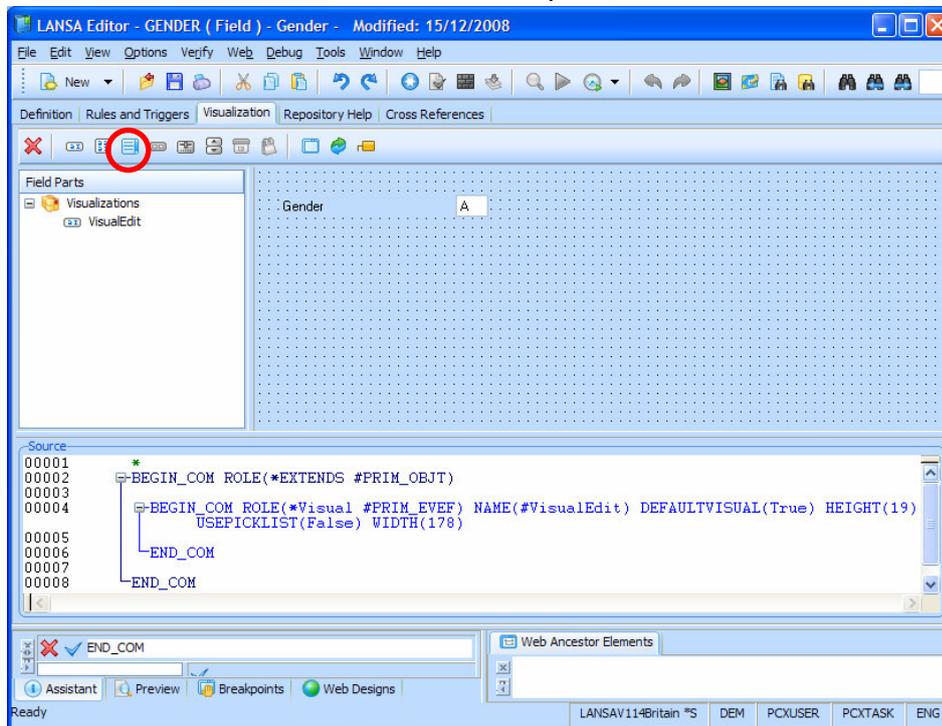
Field Visualization Picklists in WAM's

The example below shows how a field visualisation picklist can be used in a WAM.

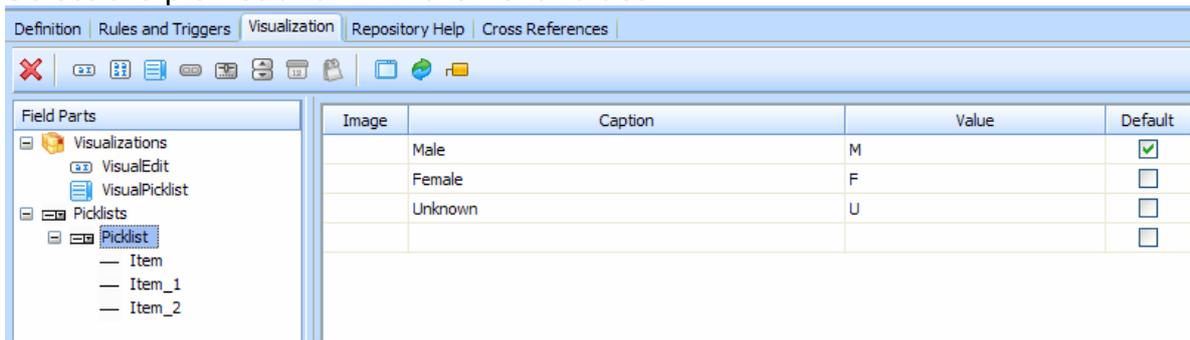
Create a new field called GENDER, Alpha 1.



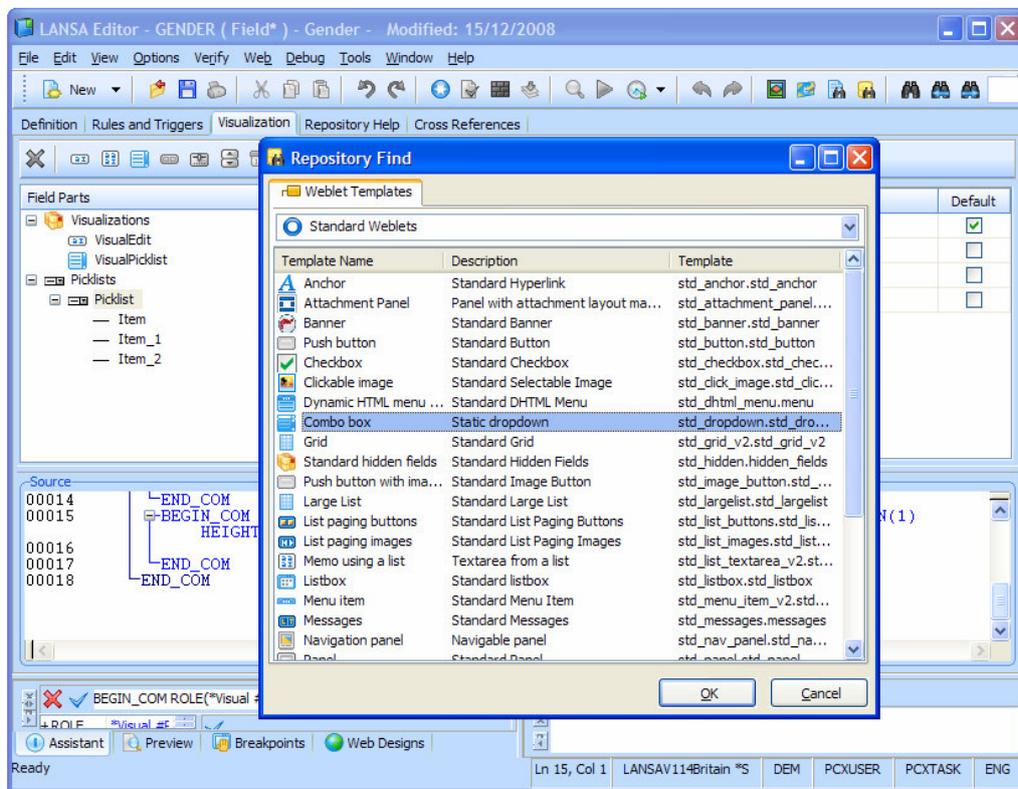
In the LANSAX IDE, open this new field GENDER and navigate to the *visualization* tabsheet. Click in the toolbar on the *picklist* button.



Select the picklist and fill in the next values:

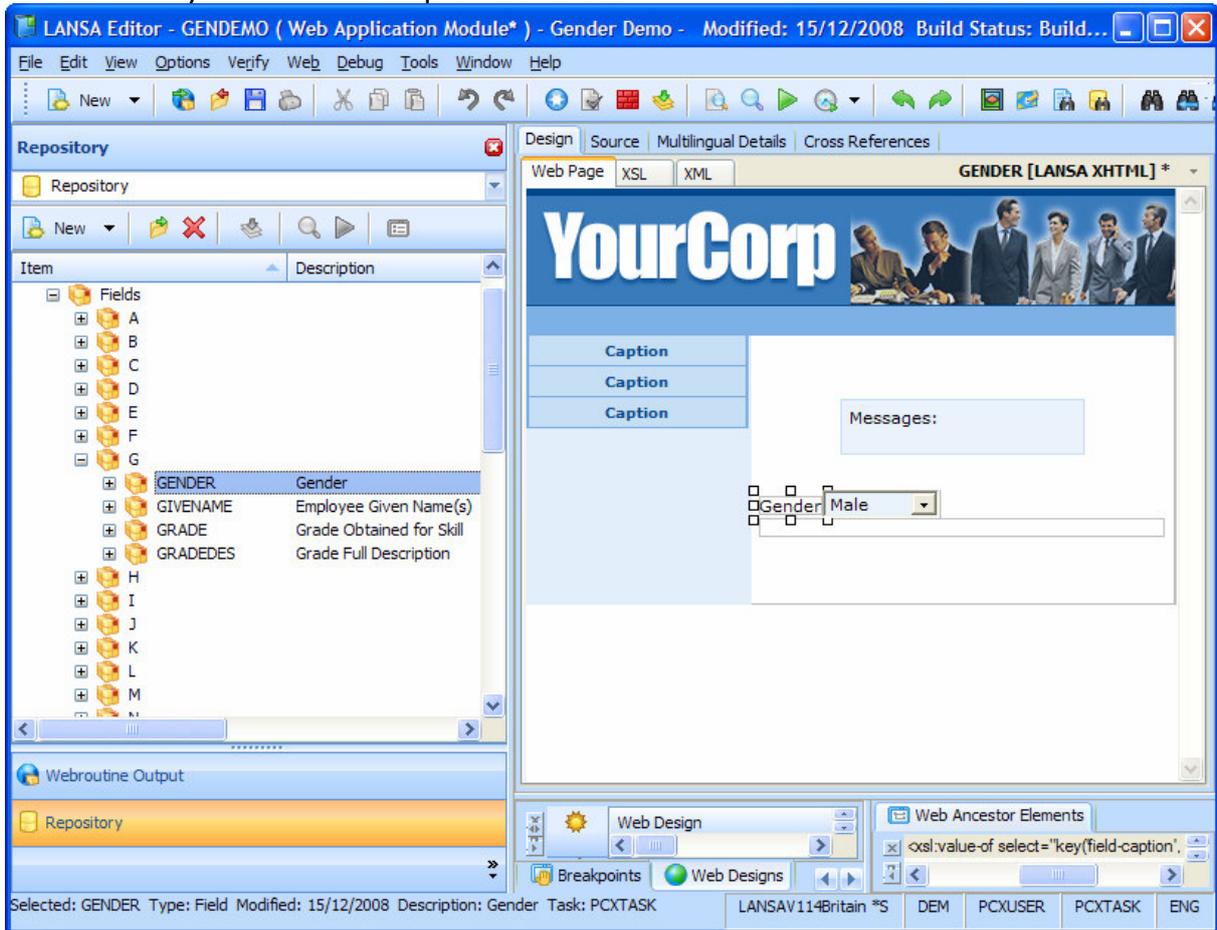


Use the *Select weblet* button now and select the weblet *Combo box*:

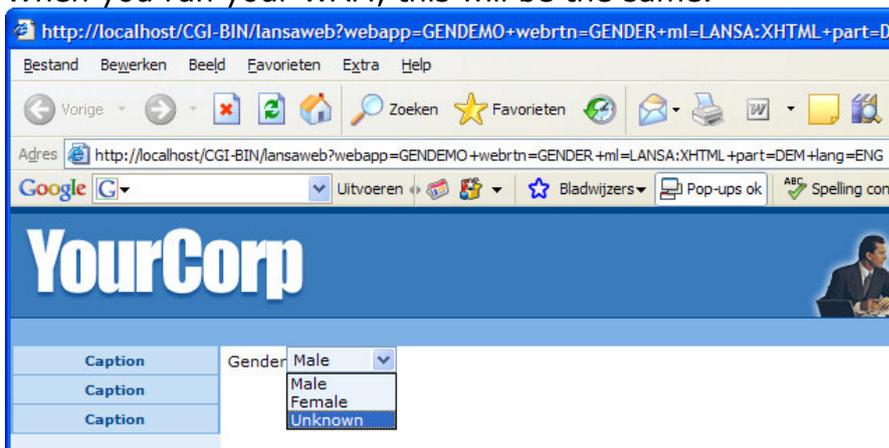


Save all changes.

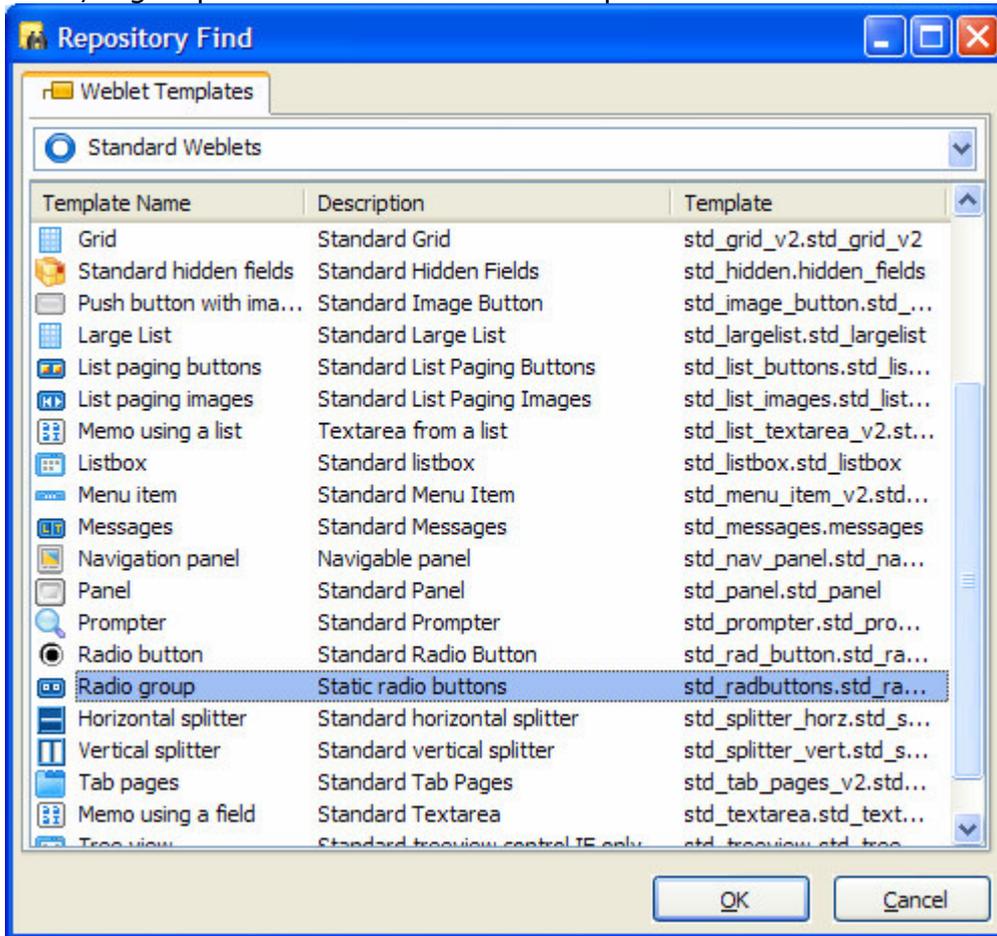
Create a new WAM now (fill in the name of your webroutine). Go to the Design tabsheet and drag field *GENDER* to the WAM. You will see that the combo box is automatically filled with the picklist values.



When you run your WAM, this will be the same.



Instead of a combo box, you can decide to show another visualisation in the WAM, a group of radio buttons for example:

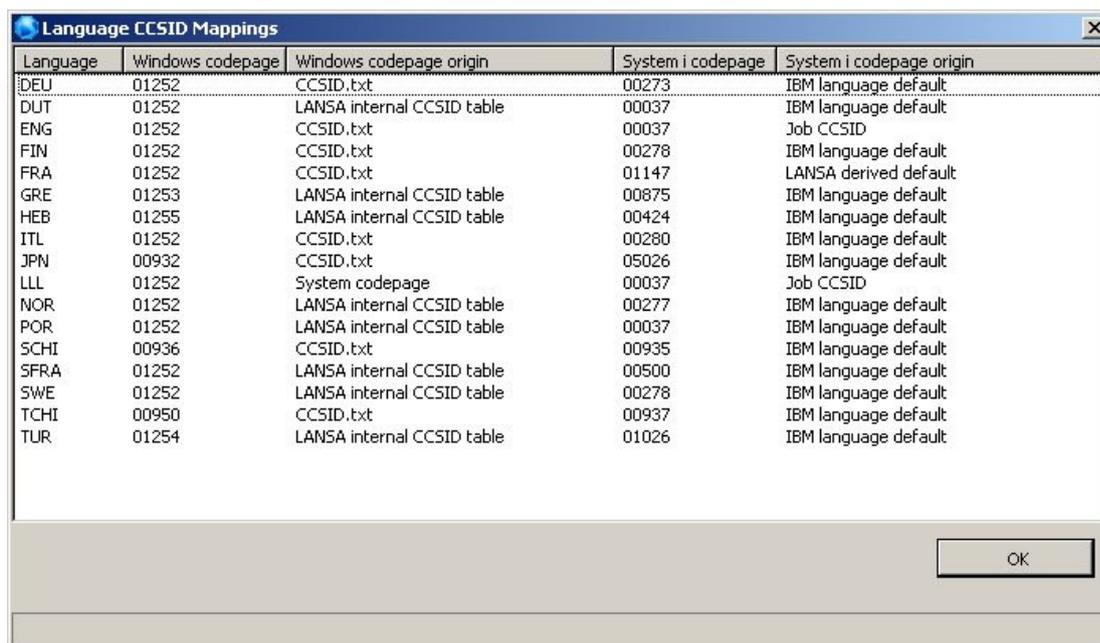


When you execute your WAM, it will look like this:



Detailed explanation of the Language CCSID Mappings feature

When a Partition Initialization is performed after updating the Partition Definition using System Initialization, the following dialog is displayed.



Language	Windows codepage	Windows codepage origin	System i codepage	System i codepage origin
IDEU	01252	CCSID.txt	00273	IBM language default
DUT	01252	LANSA internal CCSID table	00037	IBM language default
ENG	01252	CCSID.txt	00037	Job CCSID
FIN	01252	CCSID.txt	00278	IBM language default
FRA	01252	CCSID.txt	01147	LANSA derived default
GRE	01253	LANSA internal CCSID table	00875	IBM language default
HEB	01255	LANSA internal CCSID table	00424	IBM language default
ITL	01252	CCSID.txt	00280	IBM language default
JPN	00932	CCSID.txt	05026	IBM language default
LLL	01252	System codepage	00037	Job CCSID
NOR	01252	LANSA internal CCSID table	00277	IBM language default
POR	01252	LANSA internal CCSID table	00037	IBM language default
SCHI	00936	CCSID.txt	00935	IBM language default
SFRA	01252	LANSA internal CCSID table	00500	IBM language default
SWE	01252	LANSA internal CCSID table	00278	IBM language default
TCHI	00950	CCSID.txt	00937	IBM language default
TUR	01254	LANSA internal CCSID table	01026	IBM language default

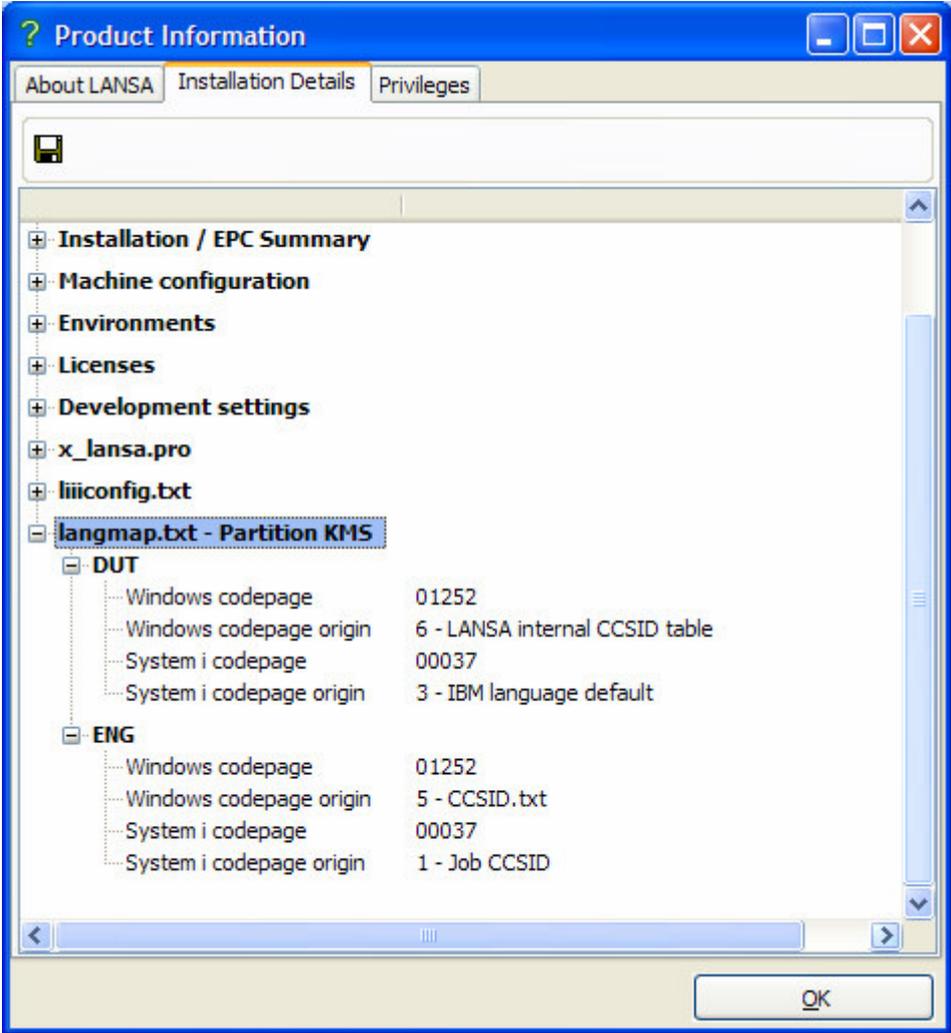
This dialog lists the codepages and CCSID that are used for multilingual text conversions. Principally this is used when exporting from iSeries to Windows and when using the Host Monitor, but it is also used when exporting from Windows to another Windows repository.

The purpose of the dialog is to:

1. make you aware that LANSAs has to make these decisions and,
2. precisely what decisions LANSAs has made in assigning codepages and CCSID to each language.

It is only shown the very first time that a Partition Initialization is performed after obtaining the Partition Definition using System Initialization.

If you need to see the information subsequently, it is listed in the Product Information (Installation Details in langmap.txt) accessible from the Visual LANSAs Editor.



We can see the same explanation in the following documentation. *Visual LANSAs Administrator Guide in 2.3 Partition Initialization*

Problems with decimals

Problems with decimals can be caused by inconsequent settings in your complete environment. When settings are not the same, you can get strange results in your programs.

Example

A number with 5 decimals will be changed with the Built-In-Function ROUND. The ROUND Built-In Function changes the number into a value with 2 decimals.

When this program is executed by a CALL PROCESS(*DIRECT), the field value will be changed from 943,85532 into 943,86.

As the same function is executed with CALL_SERVER_FUNCTION, they value will be changed from 943,85532 into 943.00.

The result can be that the decimal values are omitted at the implementation of the BIF on the iSeries.

Cause

The result can be different because the decimals are the result of settings on the iSeries itself, LANSA on de iSeries, the Windows server or PC settings, the Environment variables or startup parameters of the X_START parameter file (which uses the X_RUN.EXE command).

Things to check

1. Check the iSeries QDECFMT value?
This value controls the decimal sign character on the iSeries (comma or point).
2. Also check on the iSeries the LANSA dara area DC@A01. It also contains settings for the decimal sign character and this should be the same as the iSeries setting.
3. What are the "local" settings on your Windows machine?
4. What is the LANSA setting on the Windows site (compare this with the DC@A01 setting)? It is normally stored in the LANSA Internal file LX_F96.
5. What are the values in the X_LANSA.PRO file (in Windows) and the X_LANSA.PRO file (on the iSeries IFS)?
6. Which X_RUN - and Environment-variables are used by the start of your LANSA program (check the **X_DECIMAL_POINT_CHAR=.** or **X_DECIMAL_POINT_CHAR=,** values)?

-
-
7. Check the parameters in the X_START parameter file, or the parameter settings behind the icon you click on!
 8. Is a CALL_SERVER_FUNCTION involved in a Runtime Application or in the LANSa IDE (to test your application for example)?

So settings on the machine itself (both in Windows and iSeries), LANSa settings (both in Windows and iSeries) and parameter settings (X_RUN, X_START, X_LANSa.PRO, again on both machines) gives you flexibility, but can give different results if the settings are not all the same.

Possible Solution

To overrule different settings on your machines/environments, put in your **X_LANSa.PRO** file the value:

XENV=X_DECIMAL_POINT_CHAR=.

or

XENV=X_DECIMAL_POINT_CHAR=,

How to add an option in Framework application that allow end users to reset VLF screen layouts to 'as shipped' defaults

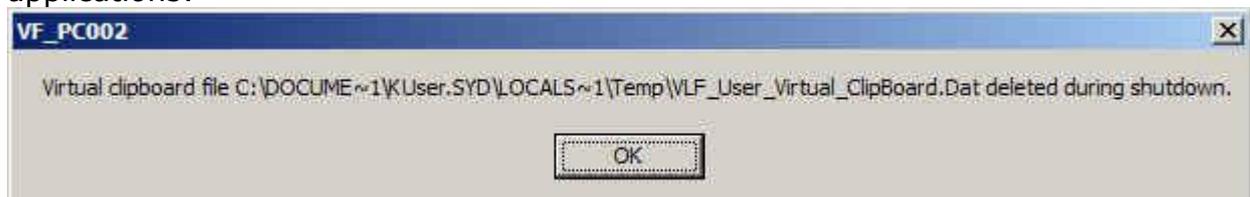
There might be times when you want to test your Framework application with VLF shipped default layouts (e.g. the main Framework window size, location of the command tab, layout).

Method

To do this you will need to:

1. Create a hidden command handler:
BEGIN_COM ROLE(*EXTENDS #VF_AC020)
MTHROUTINE NAME(uExecute) OPTIONS(*REDEFINE)
Invoke Method(#USystem.uSystemVClipboard.uDeleteFileonExit)
Invoke Method(#USystem.uUserVClipboard.uDeleteFileonExit)
Use message_box_show (ok ok info *Component 'Close and restart this application to use the originally shipped layout settings')
ENDROUTINE
END_COM
2. Create a command named "**Reset Default Layouts**" with a nice self-explanatory hint.
Set to display it e.g. on the File menu.
3. Enable this command "**Reset Default Layouts**" as a Framework Level command.
Ensure it is defined as type **Hidden** and snapped in with the hidden command handler (in Step 1).

The End User now has the option to delete their Virtual Clipboard Settings to restore their layout to 'as shipped' defaults when restarting their Framework applications:



You can also change the 'as shipped' defaults and deploy it to an end user machine (refer to the VLF guide "Deploying Clipboard Initial Values" for more information).

Simple Encrypting and Decrypting sample using the ENCRYPT and DECRYPT BIFs

LANSA provides the BIFs ENCRYPT and DECRYPT which are used to encode and decode text strings.

The following sample code takes a string INPUT and Encrypts the input and generates a encrypted text and then proceeds to decode the encrypted text.

Notes

In order to use the ENCRYPT and DECRYPT BIFs in RDMLX, a minimum of Version SP4 is required.

Due to potential issues in the usage of BINTOHEX and HEXTOBIN BIFs in conjunction with the ENCRYPT and DECRYPT BIFs, a recent enhancement (introduction of a 4th parameter) has been made to the ENCRYPT and DECRYPT BIFs. It is recommended that the 4th parameter of the ENCRYPT and DECRYPT BIFs is used instead of BINTOHEX and HEXTOBIN BIFs.

Sample code

```
FUNCTION OPTIONS(*DIRECT)

DEFINE FIELD(#KEY16) TYPE(*CHAR) LENGTH(016) DESC('Encryption key supplied')
DEFAULT('xG5LQPanEIXBhYkf')
DEFINE FIELD(#VAL32) TYPE(*CHAR) LENGTH(32) DESC('Value to decrypted')
DEFINE FIELD(#FP_EKEY) TYPE(*CHAR) LENGTH(32) DESC('Value to decrypted')
DEFINE FIELD(#HTTPUSREM) TYPE(*CHAR) LENGTH(32) DESC('Value to decrypted')
DEFINE FIELD(#ENC32) TYPE(*CHAR) LENGTH(32) DESC('Value in encrypted form')
DEFINE FIELD(#HEX64) TYPE(*CHAR) LENGTH(64) DESC('Encrypted value in Hex')
DEFINE FIELD(#ZZCHAR64) TYPE(*CHAR) LENGTH(64)
DEFINE FIELD(#ZZCHAR32) TYPE(*CHAR) LENGTH(32)
DEFINE FIELD(#DKEY16) TYPE(*CHAR) LENGTH(016) DEFAULT('xG5LQPanEIXBhYkf')
DEFINE FIELD(#DHEX64) TYPE(*CHAR) LENGTH(064) DESC('Encrypted value in hex')
DEFINE FIELD(#DVAL32) TYPE(*CHAR) LENGTH(032) DESC('Decrypted value returned')
DEFINE FIELD(#DENC32) TYPE(*CHAR) LENGTH(032)
DEFINE FIELD(#DLEN) TYPE(*DEC) LENGTH(005) DECIMALS(0)

BEGIN_LOOP
REQUEST FIELDS(#FP_EKEY)
CHANGE FIELD(#HTTPUSREM) TO(#FP_EKEY)
EXECUTE SUBROUTINE(ENCRYPT) WITH_PARMS(#HTTPUSREM #ZZCHAR64)

DISPLAY FIELDS(#ZZCHAR64)

REQUEST FIELDS(#ZZCHAR64)
EXECUTE SUBROUTINE(DECRYPT) WITH_PARMS(#ZZCHAR64 #ZZCHAR32)

DISPLAY FIELDS(#ZZCHAR32)
END_LOOP

RETURN
```

```
SUBROUTINE NAME(ENCRYPT) PARMS((#VAL32 *RECEIVED) (#HEX64 *RETURNED))
DEFINE FIELD(#LEN) TYPE(*DEC) LENGTH(5) DECIMALS(0)
CHANGE FIELD(#LEN) TO(32)
* Use ENCRYPT BIF to encrypt #VAL16 of length #LEN using
* #KEY16 to return encrypted value in #ENC16
USE BUILTIN(RIGHT) WITH_ARGS(#VAL32) TO_GET(#VAL32)
USE BUILTIN(ENCRYPT) WITH_ARGS(#VAL32 #LEN #KEY16 YES) TO_GET(#HEX64)
* USE BUILTIN(BINTOHEX) WITH_ARGS(#ENC32) TO_GET(#HEX64)
ENDROUTINE
```

```
SUBROUTINE NAME(DECRYPT) PARMS((#DHEX64 *RECEIVED) (#DVAL32 *RETURNED))
* Key to be used for the decryption. This must be the
* same key that was used for the encryption.
CHANGE FIELD(#DLEN) TO(32)
* convert input hex value to character
* USE BUILTIN(HEXTOBIN) WITH_ARGS(#DHEX64)
* TO_GET(#DENC32)
* Use DECRYPT BIF to decrypt character #DENC16 of length
* #DLEN using #DKEY16 to return decrypted value,#DVAL16
USE BUILTIN(DECRYPT) WITH_ARGS(#DHEX64 #DLEN #DKEY16 YES)
TO_GET(#DVAL32)
ENDROUTINE
```

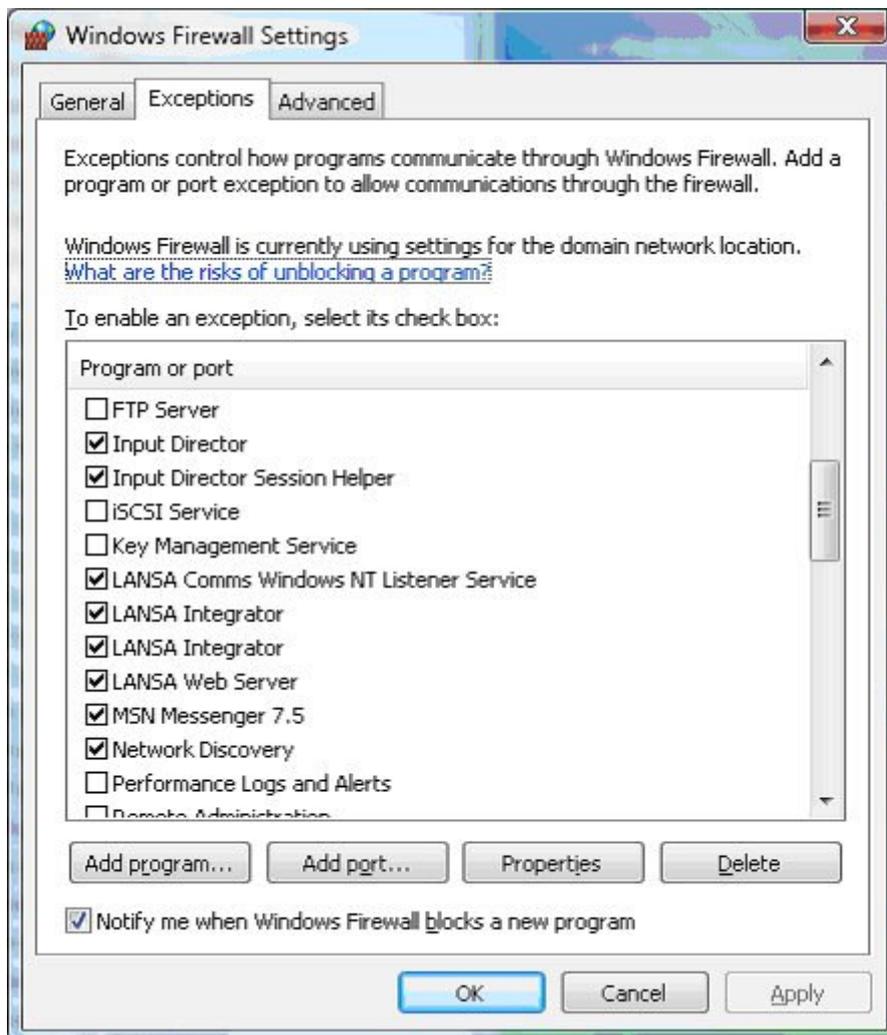
Firewall considerations for debugging in Visual LANSA after applying EPC830 to V11 SP5

From EPC830 for Visual LANSA V11 SP5, there is a new LANSA debugging mechanism. Refer to the LANSA Online Guides and the EPC830 documentation for further details. This new mechanism starts a debug listener service when starting up the Visual LANSA IDE. If you have Windows Firewall enabled, you might get the following screen when logging into Visual LANSA as the IDE is starting up.



You should elect to Unblock this program if you wish to debug within Visual LANSA.

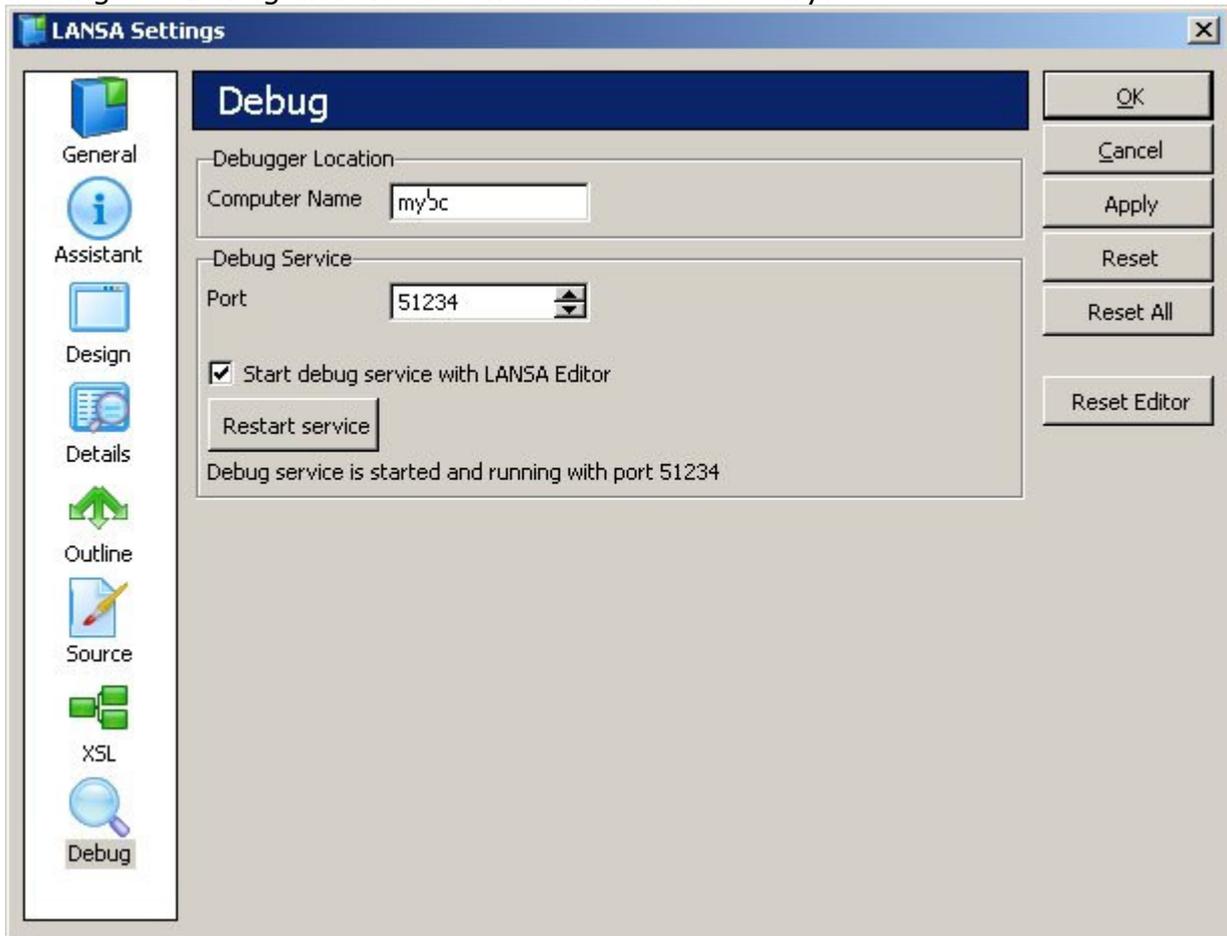
If you wish to avoid this security alert popping up every time you start the IDE, you can add this program as an exception in the Windows Firewall. This will allow the service to be automatically started every time the IDE is started.



Note

Depending on your firewall settings, you may not get a notification that the Windows Firewall has blocked the LANSAs debug service. In which case, the most obvious indicator of the debug listener service not being automatically started during login to the IDE is if you set a breakpoint in your LANSAs source code and the debug does not stop at the breakpoint.

In which case, you should check your debug setting under the menu Options -> Settings -> Debug and Restart the service if necessary.



Unable to restore V11 SP5 objects to QTEMP on V5R3 or V5R4

From V11 SP5, IBM i 6.1 is fully supported. LANSAs SP5 objects (EPCs, patches) are built on IBM i 6.1 so they can then be restored by customers on IBM i 6.1. PTFs are required for any objects saved on IBM i 6.1 before they can be restored on previous IBM i versions.

Sample error message for objects built on IBM i 6.1 but restore attempt on V5R4:

```
Message ID . . . . . : CPF3888
Date sent . . . . . : 09/11/08      Time sent . . . . . : 00:49:19
```

```
Message . . . . . : PGM DC@P8170 not restored to QTEMP.
```

Cause :

If the object is a database file, an unexpected failure occurred during the restore of a file member. The member was not restored.

If the object is not a database file, then the conversion needed to make the object compatible with the current system is not possible, so the object was not restored.

If the object is a program, module, service program, or SQL package, then either the object may not have the required creation data needed to perform the conversion, or there was an unexpected error. Conversion of these object types during restore is controlled by the Force Conversion on Restore (QFRCCVNRST) system value and the Force Object Conversion (FRCOBJCVN)

Resolution

Ensure these PTFs are applied:

V5R3

MF41354, delayed PTF to be applied at IPL
SI27297, apply immediate or delayed

V5R4

MF40520, delayed PTF to be applied at IPL
SI27294, apply immediate or delayed

V5R2

This is no longer an IBM supported Version. You are required to upgrade your OS version before applying V11 SP5.

Installation of 3rd party software affects LANSA Integrator license on System i

We have received a few reports where the installation (or upgrade) of 3rd party product on your system i can cause a working LANSA Integrator application to start generating a licence error. This can occur even though your LANSA Integrator licence is valid and no changes have been made to the LANSA Integrator application.

Reason

Some 3rd party software set environment variables at the system level, rather than at local level (within their software). This can affect the licence of the existing LANSA Integrator application because the system level environment variable overrides any other variables that are set and this causes LANSA Integrator to look in the incorrect location for its valid licence.

If your LANSA Integrator application suddenly (for no apparent reason) starts generating a licence error, you should check the following:

- Have you recently installed or upgraded any other software that uses similar licensing
- Type **WRKENVVAR LEVEL(*SYS)** at the command line and check if the following environment variable has been set.
LSFORCEHOST=<A server> where <A server> is the name of a server in your network.

If this environment variable has been set, then this is the cause of the LANSA Integrator application licence error.

Resolution

In the short term, especially if the LANSA Integrator application is a production application, you should immediately remove the environment variable LSFORCEHOST. You must also stop and restart the LANSA Integrator JSM subsystem for the change to take affect.

You should also contact the 3rd party software vendor that set the environment variable and establish why the LSFORCEHOST needs to be set at the system level where it can adversely affect other production software that was up and running working before it was set. You may also need to establish if removing the system level environment variable LSFORCEHOST adversely affects their product.

****BLDNO error when running host monitor or system Initialization after applying EPC830 to V11 SP5***

EPC830 for V11 SP5 introduced a new debugging mechanism. Refer to EPC830 for details. The description for the new debugger contains the following points:

- Special setup and administrative rights are no longer required for remote debugging. The distinction between local and remote debugging has been removed.
- X_RUN arguments DBRM, DBHU, DBHP and DBHS are no longer used or recognized and so must be removed from any X_RUN command line.
- The special Windows login XDBG_USER is no longer needed and is highly recommended to be removed.

After applying EPC830, you may encounter the following message while performing a system initialization or using the Host Monitor:

Fatal Error

Process : *BLDNO
***BLDNO**
Function : *BLDNO
***BLDNO**
Statement : 0
Message : (1073) - A communications error has occurred to server with LU Name DCXPGMLIB. Ensure the listener job is started on the server, check communications settings and job logs.
Routine : X_COM_CheckBuildNumber



Reason

One possibility for this error is that the X_LANSA.PRO on the system i IFS contains invalid parameters. For example, your X_LANSA.PRO might contain one of the parameters listed as needing to be removed i.e. DBRM=Y
Since this parameter is no longer required for remote server debugging, it is invalid and needs to be removed from the X_LANSA.PRO.

Note

On Windows, if you have an invalid parameter in your X_LANSA.PRO, you get the following error:



But for a connection to the System i, the reason isn't easily obvious from either the error on the client or the joblog generated. So, the X_LANSA.PRO should be checked as a possibility for this error.

Support for LANSA V10 will end

Support for LANSA Version 10.0 will end on June 30, 2009 or three months after the shipment of LANSA Version 12, whichever occurs first.

When you still have a LANSA V10 environment, please contact us as soon as possible so we can help you with your upgrade to the latest LANSA version.

For more details about this, have a look at:

<http://www.lansa.com/support/v11news/supportedversions.htm>

VLF.Net with iSeries Apache

If you use VLF.NET with iSeries Apache the following lines should be added into the web server configuration file:

```
AddType application/x-ms-application application  
AddType application/x-ms-manifest manifest  
AddType application/octet-stream deploy
```

You will have to stop and restart your web server to make these changes effective.

PrintScreen functionality in VL

With the code below, you can send any actual VL form to your printer:

```
EVTRoutine HANDLING(#PHBN_1.Click)
```

```
* Try this code...this is for a "printout". Otherwise you could open the
```

```
* .bmp in a local editor, or start an email with the .bmp as an attachment.
```

```
#SYS_APPLN.CaptureWindow( (*TEMP_DIR + 'LansaScreenShot.bmp') )
```

```
Use Builtin(SYSTEM_COMMAND) With_Args(X ('RUNDLL32.EXE
```

```
MSHTML.DLL,PrintHTML' + ' ' + (*TEMP_DIR + 'LansaScreenShot.bmp') + ''))
```

```
To_Get(#STD_NUM)
```

```
ENDROUTINE
```