LANSA NEWSLETTER

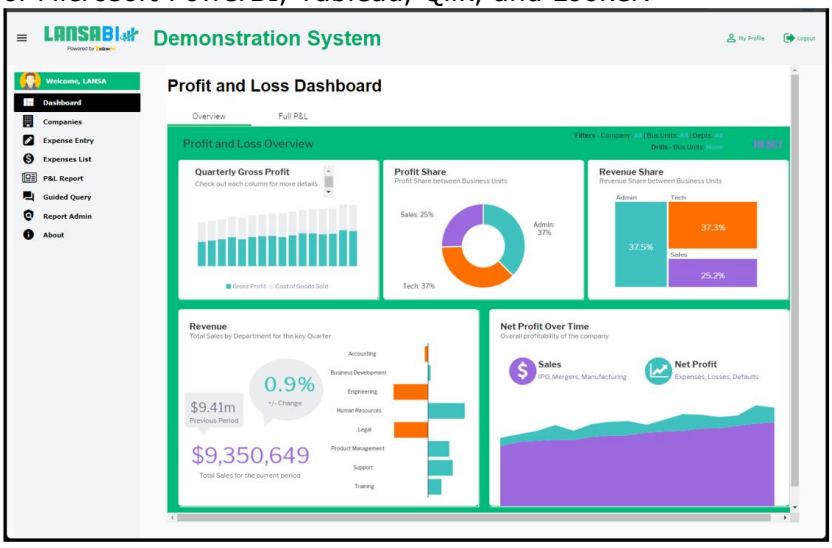
LANSA BI

This LANSA newsletter article is an excerpt of the following IT Jungle article:
https://www.itjungle.com/2023/03/29/lansa-developing-business-intelligence-tool/

LANSA is rolling out a new Web-based business intelligence tool. Dubbed LANSA BI, the new offering is based on an underlying analytics engine from Yellowfin and will be targeted predominantly at IBM i shops looking to give users greater access to actionable data.

In the past we offered customers some rudimentary query and reporting capability along the way in the form of LANSA Client, a Windows-based product that it included with other offerings. But up to this point, we have lacked a full-fledged business intelligence (BI) and analytics offering that can run with the *big boys*.

That gap in the product line-up will soon be filled by **LANSA BI**. The new offering, which is currently being tested by LANSA customers, will give LANSA a fully capable, Web-based analytics and BI suite that can compete with the likes of Microsoft PowerBI, Tableau, Qlik, and Looker.



INSIDE THIS ISSUE

LANSA BI 1

LANSA RDMLX Extended Library 4

"Old format or invalid type library" error when using Excel ActiveX component.... 6

End of support for IBM i 7.3 announced 7

Did you know? 8

Do the Opposite 8

Windows 11 and LANSA V14 & V15 - known issues 10

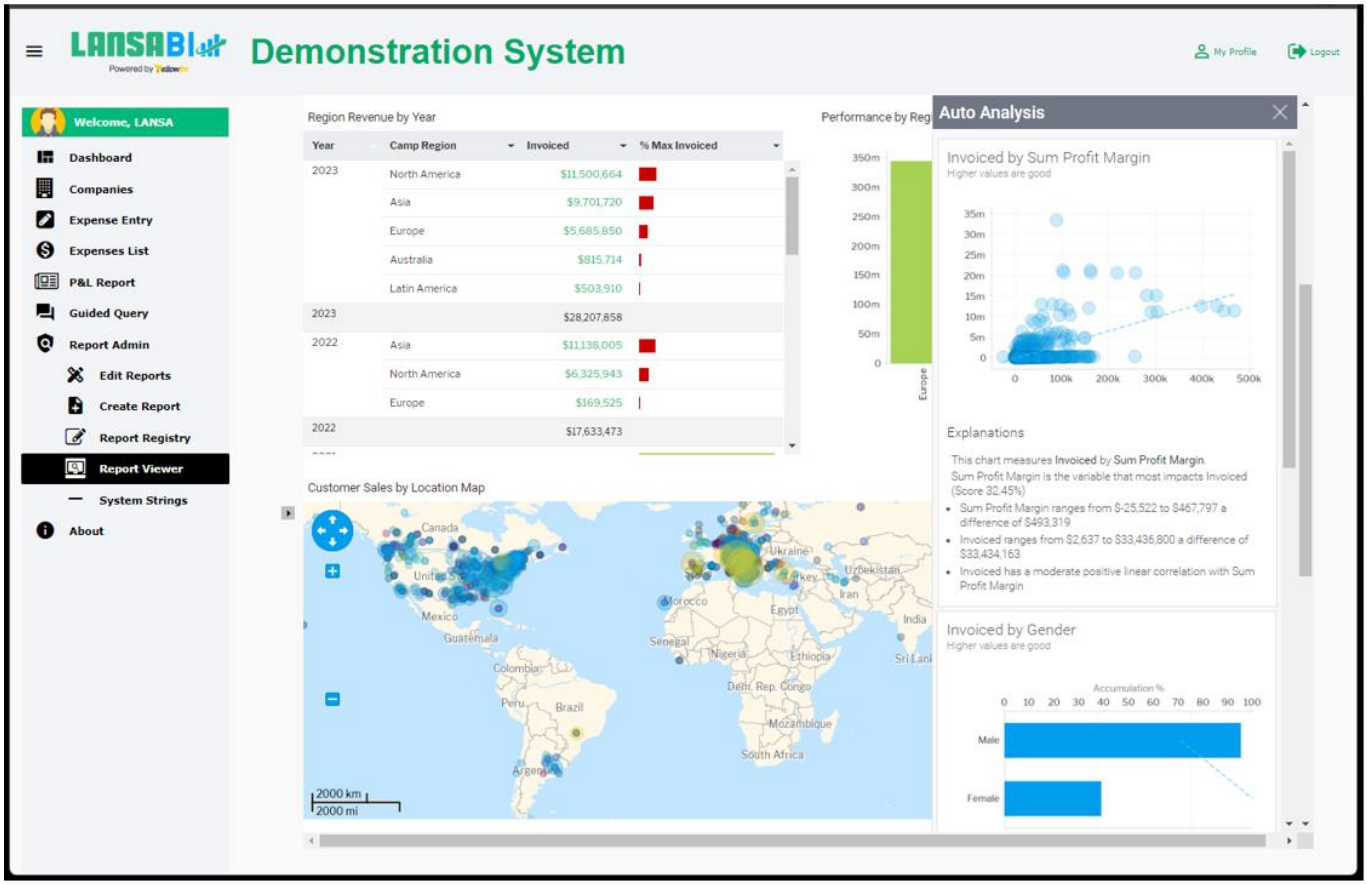
VL Web URL Parameters 11

Workstation Locks 13

Visual LANSA Editor on MS Surface..... 15

LANSA BI is based on a SQL query engine developed by Yellowfin, a Melbourne, Australia-based BI provider founded by Glen Rabie in 2003. Yellowfin developed a solid reputation for its platform and application-neutral BI product that wasn't owned and controlled by a tech giants.

Over the years, Yellowfin amassed thousands of customers in 75 countries around the world. All told, the BI product was used by 3 million end users. In early 2022, Yellowfin was sold to Idera, the same company that snapped up LANSA in 2019.

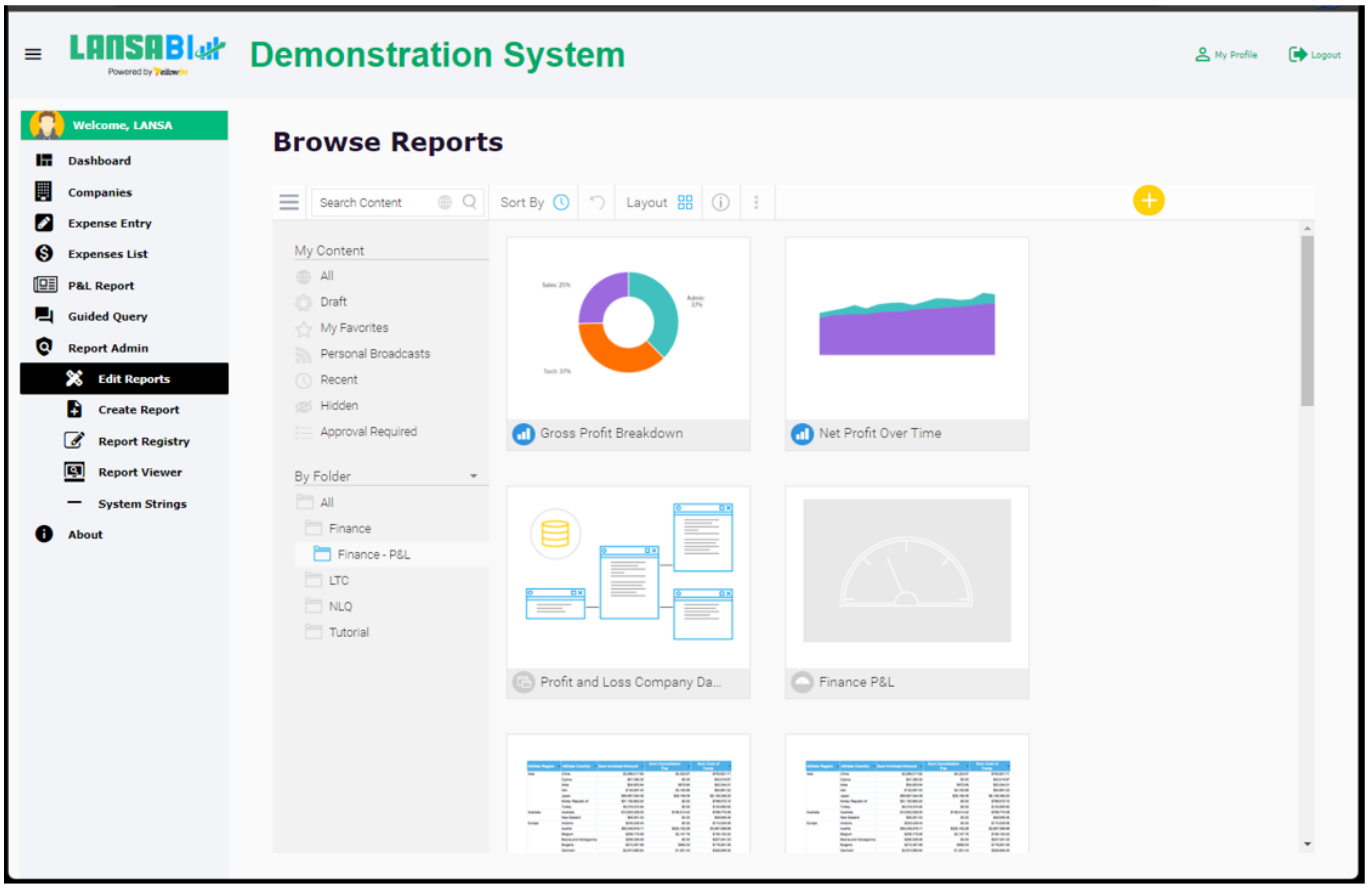


The acquisition turned out to be prescient for LANSA. Company leaders had been thinking they needed to improve their data analytic capabilities. The LANSA Client tool provided some functionality, but it lacked certain amenities. When we realized that Idera owned a BI and analytics offering via one of LANSA's sister companies, we decided to take a closer look.

Yellowfin's analytics offering was developed in Java and typically runs on Windows and Linux platforms (although LANSA will consider running it on IBM i). The product enables business analysts to conduct standard analytic functions, such as connecting to data sources, preparing data for analysis, and creating reports and dashboards. It also supports additional functionality, such as automated insights, threshold-based alerting, and "closed loop" analytics that allow users to take actions directly from dashboards. Yellowfin also supports embedded deployments, which will come in handy with the new LANSA integration.

Yellowfin forms the starting point for LANSA BI, which will feature Yellowfin functionality under the covers, but will be packaged up and integrated with other LANSA products.

The first proof of concept for LANSA BI involved one of LANSA's ERP Frameworks customers. By integrating LANSA BI with the customer's ERP software, the customer is able to access dashboards and create their own ad hoc reports against their own database directly from within the ERP software.



Yellowfin has all the basics covered when it comes to BI. Really exiting is its natural language query (**NLQ**) capability, which will enable users to construct queries in plain English as opposed to learning to program in SQL. NLQ is expected to be heavily adopted in IBM i shops that lack deep analytics expertise.

As an end user, someone who doesn't know anything about databases can say 'Hey, give me the gross profit for all my divisions for last quarter,' and it will just come up and give you a table of all that or a graph of all that.

Once that report is created via NLQ, it can be saved and built upon and customized later. That will give users the ability to extend and customize their analytics journey over time.

You can go back into that report . . . and say I want to extend this. Maybe you want to have some clickthrough capacity and say 'Hey, when I click on each quarter, I really want to go into the details of that quarter.' So you can connect that to another report that you may have created also.

Many LANSA users today are burdened by a reliance on manual data analytics methods. According to a regular survey, the organizations are relying heavily on things like Excel and manually building reports. The goal with LANSA BI is to automate more of that work and make the LANSA customers more productive.

The LANSA BI software will be formally announced in April at COMMON POWERUp 2023, with general availability expected later this year.

LANSA RDMLX Extended Library

LANSA RDML Extended Library provides various functions that are required in common situations (such as making an HTTP request), but haven't made their way to the core LANSA RDML Language. The majority of the functionality included in this library is the result of enhancement requests. Some of the features here may eventually be added to the core LANSA RDML Language.

The complete functionality of the RDMLX Extended Library is made available via LANSA supplied Reusable Parts. The name of these Reusable Parts starts with XPRIM_, for example:

XPRIM_HttpRequest

XPRIM_HttpRequestOptions

XPRIM_HttpResponse

To use the functionality in one of your programs, simply add a define_com in your source like:

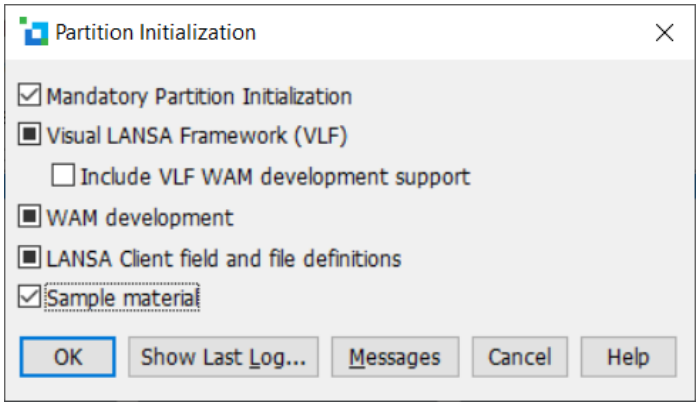
```
Define_Com Class(#XPRIM_HttpRequest) Name(#HttpRequest)
```

If you put a define_com like this in your program and the LANSA IDE shows…..

Class name #XPRIM_HttpRequest could not be found.

….. means that the LANSA RDMLX Extended Library is not loaded yet into your partition.

To load these, start on your LANSA development PC a partition Init:

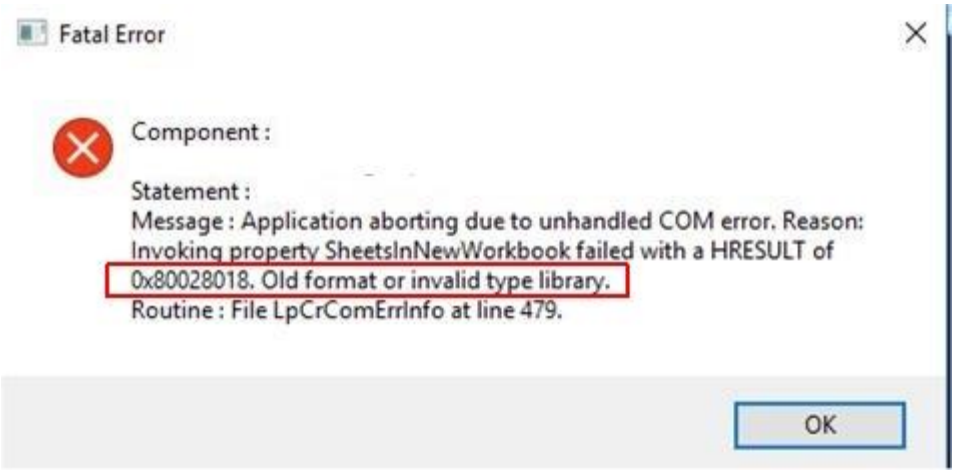


Important here is that you should checking the check boxes for 'Mandatory Partition Initialization' and 'Sample material' (instead of just accepting the grey out boxes).

The total content of the LANSA RDMLX Extended library is:

XPRIM_BaseObject	RDMLX Extended Library Base Class	Reusable Part
XPRIM_Binary	Binary Data	Reusable Part
XPRIM_Crypto_Hash	Cryptography - Hash Functions	Reusable Part
XPRIM_Crypto_HMAC	Cryptography - HMAC	Reusable Part
XPRIM_Crypto_Util	General Cryptography-related Utilities	Reusable Part
XPRIM_Directory	File System - Directory	Reusable Part
XPRIM_ErrorInfo	Error Details	Reusable Part
XPRIM_File	File System - File	Reusable Part
XPRIM_FileUtil	File Utilities	Reusable Part
XPRIM_HttpContent	HTTP Content	Reusable Part
XPRIM_HttpContentInfo	HTTP Request Content Info	Reusable Part
XPRIM_HttpRequest	HTTP Request	Reusable Part
XPRIM_HttpRequestOptions	HTTP Request Options	Reusable Part
XPRIM_HttpResponse	HTTP Response	Reusable Part
XPRIM_HttpResponseReadContentStatus	HTTP Response Read Content Status	Reusable Part
XPRIM_HttpUtil	HTTP Utility Functions	Reusable Part
XPRIM_Json	JSON	Reusable Part
XPRIM_JsonArray	JSON Array	Reusable Part
XPRIM_JsonBoolean	JSON Boolean	Reusable Part
XPRIM_JsonElement	JSON Element	Reusable Part
XPRIM_JsonNull	JSON Null	Reusable Part
XPRIM_JsonNumber	JSON Number	Reusable Part
XPRIM_JsonObject	JSON Object	Reusable Part
XPRIM_JsonPath	JSON Path	Reusable Part
XPRIM_JsonReader	JSON Reader	Reusable Part
XPRIM_JsonString	JSON String	Reusable Part
XPRIM_JsonWriter	JSON Writer	Reusable Part
XPRIM_OSUtil	Operating System Utility Functions	Reusable Part
XPRIM_RandomAccessJsonReader	Random Access JSON Reader	Reusable Part
XPRIM_StringEncodings	String Encoding Enumeration	Reusable Part
XPRIM_StringEncodingType	String Encoding Type	Reusable Part
XPRIM_StringEncodingUtil	String Encoding Utility	Reusable Part
XPRIM_StringUtil	String Utility Functions	Reusable Part
XPRIM_UriBuilder	URI Builder	Reusable Part

"Old format or invalid type library" error when using Excel ActiveX component



Symptoms

If you automate Microsoft Excel with Microsoft Visual Basic .NET, Microsoft Visual C# .NET, Microsoft Visual C++, or LANSA ActiveX, you may receive the following error :

```
Error: 0x80028018 (-2147647512)
Description: Old Format or Invalid Type Library
```

Cause

You receive this error calling an Excel method when the following conditions are true:

- The method requires an LCID (locale identifier).
- You run an English version of Excel. **However, the regional settings for the computer are configured for a non-English language.**

If the client computer runs the English version of Excel and the locale for the current user is configured for a language other than English, Excel will try to locate the language pack for the configured language. If the language pack is not found, the error is reported.

Solution

To work around this problem, you can Install the Multilingual User Interface Pack for your version of Office, or the language pack that matches your PC language, or set your PC to use language English.

See: https://support.microsoft.com/en-us/office/language-accessory-pack-for-office-82ee1236-0f9a-45ee-9c72-05b026ee809f

End of support for IBM i 7.3 announced

In an announcement dated September 27, 2022, IBM revealed that they will discontinue support for release IBM i 7.3 on September 30, 2023.

7.3 was released on April 15, 2016. With it going off support on September 30, 2023, it means that it will have been an active release for seven years and five and a half months.

You can read the full announcement, including the other product coming to end-of-life, here.

If you have a Power8 or higher and you are on IBM i 7.3 you need to start finalizing your plans to update your operating system to either 7.4 or 7.5.

Did you know?

Do the Opposite

Let's focus on example below. You present a button (or something else). In the program you check the Visible property, and, if the actual value is True, you want to change it to False. The opposite of course, if the actual value is False, you want to change it to True.

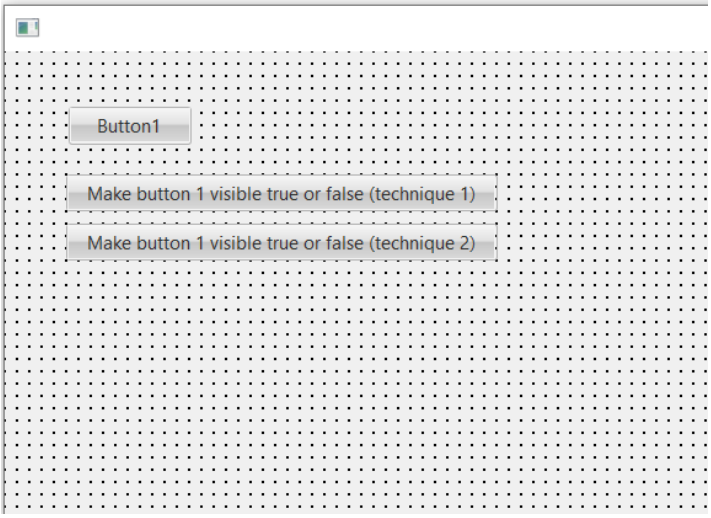
You could do this with this piece of code:

```
Evtroutine Handling(#Technique1.Click)
If (#Button1.Visible = True)
#Button1.Visible := False
Else
#Button1.Visible := True
Endif
Endroutine
```

But do it like this, will give you the same result:

```
Evtroutine Handling(#Technique2.Click)
#Button1.Visible := #Button1.Visible.Not
Endroutine
```

You can test this with this Visual LANSA Form. The source is available on the next page.



```
* ************************************************
*  COMPONENT:  STD_FORM
* ************************************************
Function Options(*DIRECT)
Begin_Com Role(*EXTENDS #PRIM_FORM) Clientwidth(640) Clientheight(301) Componentversion(2) Left(524)
Top(212)

Define_Com Class(#PRIM_PHBN) Name(#Button1) Caption('Button1') Displayposition(1) Left(42)
Parent(#COM_OWNER) Tabposition(1) Top(36)
Define_Com Class(#PRIM_PHBN) Name(#Technique1) Caption('Make button 1 visible true or false (technique 1)')
Displayposition(3) Left(40) Parent(#COM_OWNER) Tabposition(3) Top(80) Width(281) Height(24)
Define_Com Class(#PRIM_PHBN) Name(#Technique2) Displayposition(2) Left(40) Parent(#COM_OWNER)
Tabposition(2) Top(112) Width(281) Height(24) Caption('Make button 1 visible true or false (technique 2)')

Evtroutine Handling(#com_owner.CreateInstance)
Set Com(#com_owner) Caption(*component_desc)
Endroutine

Evtroutine Handling(#Technique1.Click)
If (#Button1.Visible = True)
#Button1.Visible := False
Else
#Button1.Visible := True
Endif
Endroutine

Evtroutine Handling(#Technique2.Click)
* Is exactly doing the same:
#Button1.Visible := #Button1.Visible.Not
Endroutine
End_Com
```

Windows 11 and LANSA V14 & V15 - known issues

There are a few known issues concerning LANSA V14 / V15 and Windows 11.

- **Installing LANSA IDE V15 from SPIN0340 with settings for UK fails on missing Translation Table**

Towards the end of the LansaInstallLog.txt (in the %temp% directory) you can see:

*** Exception raised *** in function <File::CopyFileW>, Error code <6> …….. Unable to copy file from
<C:\LANSAInstalls\LANSA\Open\TranslationTables\1146\lcoemesg.dat> to
<C:\LANSAInstalls\LANSA\Open\lcoemesg.dat> The handle is invalid.

Solution: Do a custom install. For VL have English-UK (1146), but for Open select 1140. Once the install has occurred, copy the missing file for 1146 from the 1140 directory to the LANSA Open directory

i.e
CD <install root>
Copy Open\TranslationTables\1140\LCOEMESG.DAT Open
Or
Copy "C:\Program Files (x86)\LANSA\Open\TranslationTables\1140\LCOEMESG.DAT" "C:\Program Files (x86)\LANSA\Open"

- **"System Initialization Failed" during initialization**
- **Web Execution generates a 404 error**
- **Communication: Native Error:10013 CPIC error: CM_PRODUCT_SPECIFIC_ERROR(20) in bind**

Cause: Windows 11 has some specific behavior that is different from Win10

Solution for these errors is
For LANSA V15: **apply EPC150050**
(or apply Hotfix: EPC150040HF_220202.zip when you are still on EPC150040 level.)

For LANSA V14 SP2: apply **EPC142070**
(or apply HotFix EPC142060HF_220202.zip when you are still on EPC142060 level.)

*** Server Modules (application and API) are failing**
*** Super Server connection failures**
*** LANSA Debug not working**

Cause: Windows 11 update 22H2 introduced a new Microsoft DLL, named lxutil.dll
(same name as a standard LANSA module !) which may be causing issues.

Solution:

Copy \x_win95\x_lansa\execute\lxutil.dll into \Connect
Copy \x_win64\x_lansa\execute\lxutil.dll into \Connect64

VL Web URL Parameters

The Web Application object allows access to the full URL and URL Parameters as name and value pairs. It also allows for monitoring of changes to the URL.

A Visual LANSA web URL is formatted like this:
http://[domain]:[port]/[configuration]/[partition]/[webpage].html?[parm1]=[value]&[parm2=value]

The URL for the current page is published as the URL property of SYS_WEB.
#Result := #sys_web.URL

This could return something like:
http://localhost:2014/tipvcs/ide/xvlwebtst.html?lang=eng&developer=yes&debug=yes,lansa14:51234

To process any parameters associated with the URL requires processing name/value pairs. The query string, that is any portion of the URL following the "?" (which by the way is optional), is parsed and made available as a collection of items, each with Name and Value properties. Each Name and Value is separated by a "&". If the query string is corrupt in any way, only those parameters before the corruption will be available.

This is an example with two parameters:

http://localhost:8080/lansa/dem/urlparame?lang=eng&developer=yes&name=Rene&city=Wageningen

I show the two parameters in screen fields:

The source

```
Begin_Com Role(*EXTENDS #PRIM_WEB) Theme(#SYS_THEME<MaterialDesignBlue>) Height(488)
Width(600)

Define_Com Class(#STD_TEXT.EditField) Name(#PARM1) Displayposition(1) Left(24)
Parent(#COM_OWNER) Tabposition(1) Top(40) Caption('First parameter') Captiontype(Caption)
Define_Com Class(#STD_TEXT.EditField) Name(#PARM2) Displayposition(2) Left(24)
Parent(#COM_OWNER) Tabposition(2) Top(135) Caption('Second parameter') Captiontype(Caption)

Evtroutine Handling(#Com_owner.Initialize)

If_Ref Com(#SYS_WEB.URLParameters<"name">) Is_Not(*Null)
#PARM1 := #SYS_WEB.URLParameters<"name">.Value
Endif

If_Ref Com(#SYS_WEB.URLParameters<"city">) Is_Not(*Null)
#PARM2 := #SYS_WEB.URLParameters<"city">.Value
Endif

Endroutine

End_Com
```

And…….
Because LANSA puts all URL parameters into a collection, you can also do this:

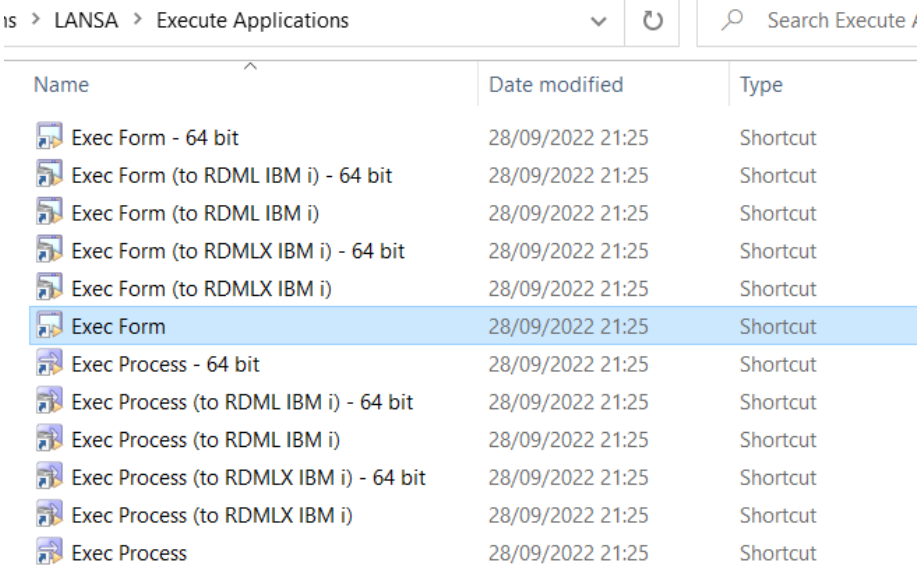
```
Evtroutine Handling(#Com_owner.Initialize)

For Each(#Parameter) In(#sys_web.UrlParameters)

If (#Parameter.Name.UpperCase = NAME)
#PARM1 := #Parameter.Value
Endif

If (#Parameter.Name.UpperCase = CITY)
#PARM2 := #Parameter.Value
Endif

Endfor

Endroutine
```

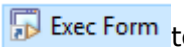
Workstation Locks

LANSA components can sometimes remain undesirably locked by a workstation.

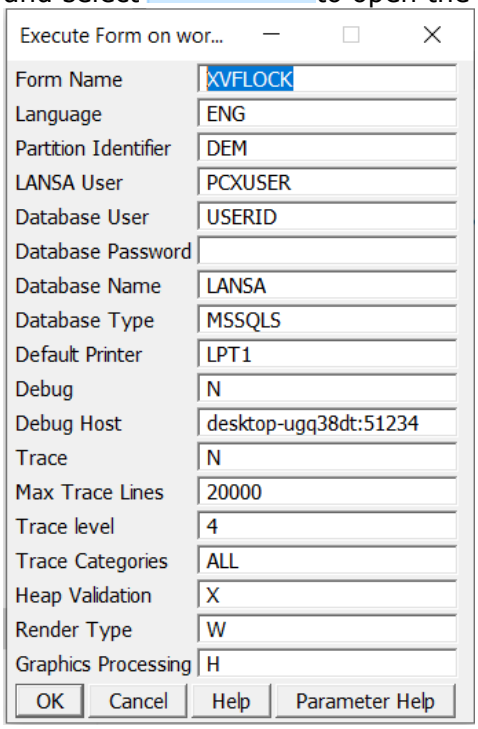
Execute the Workstation Locks **form**, **XVFLOCK**, to manage any workstation locks on the Repository. You must have administrator authority to display and clear all workstation locks.

To do this, go to the LANSA Folder, expand the Execute Applications folder:

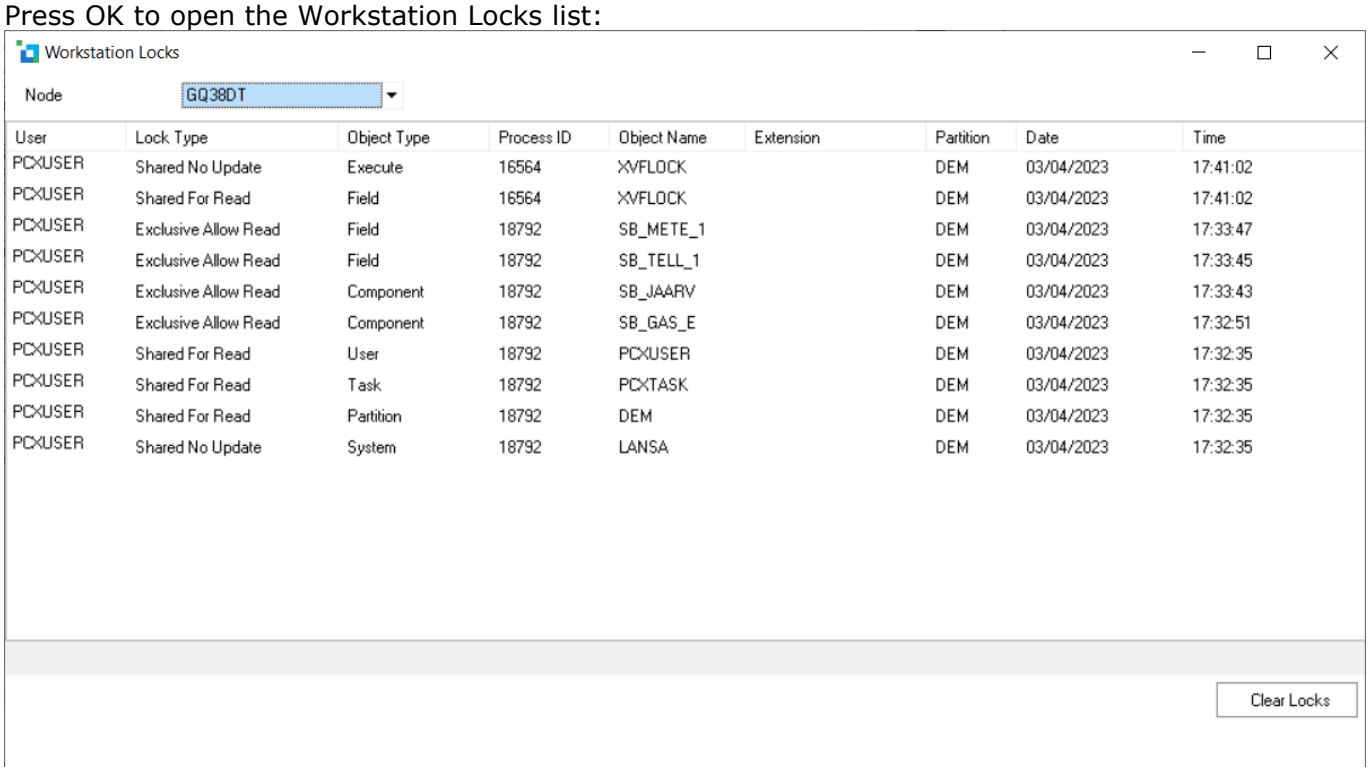


and select 

to open the Execute Form on workstation dialog:



Press OK to open the Workstation Locks list:



Select the desired workstation from the dropdown list to display the details of any locked objects for this workstation. Use the refresh button beside the dropdown to ensure you are viewing the latest details.

If you do not have administrator authority you will only see a list of the workstation locks related to your user profile. You cannot clear the workstation locks.

If you do have administrator authority you will see the details of all workstations and their related locks and will be able to clear the locks if the associated process is not active.

Visual LANSA Editor on MS Surface

A customer has installed the Visual LANSA environment on a MS Surface Pro 7 Plus.
Although, the installation was successful, the LANSA Editor frequently stopped processing.

The MS Surface had the following specifications and had sufficient performance.

- OS: Windows 10 Pro 64bit Edition
- CPU: i5 1135G7
- GPU: DirectX 12
- RAM: 8GB
- SSD: 128GB

Cause

The cause of the stopping was the resolution of the monitor.
The default resolution of MS Surface Pro 7 Plus is 2736 x 1824.

Solution

After changing the resolution to 1920 x 1080, the LANSA editor worked fine.

In particular, the dialog for creating fields seems to be affected.
Perhaps, the production application may be affected in the same way.