http://www.lansa.com

Newsletter
October 2009

aXes

Transform your 5250 green-screens into Rich Internet Applications without changing a Line of Code and *then Access them from Anywhere you can use a Web Browser.*

In This Issue

aXes automatically transforms your current 5250 green-screen business applications into modern Web browser-based applications – without changing any code.

Using the 'eXtensions' feature of aXes, you can enrich and add real business process value to your 5250 applications with a host of Windows user interface constructs – including dropdowns, checkboxes, radio buttons, calendars, charts, hyperlinks, images and a full color palette.

And there's more... straight out-of-the-box, aXes includes remote access to spooled print jobs, remote SQL access to IBM i databases and server administration.

Try the free evaluation version (http://www.axeslive.com/downloads/index.htm).

Auto-Deploy 5250 Apps in a Browser

Transform 5250 screens into Web pages on-the-fly right on your existing server. Access your IBM i, System i, iSeries or AS/400 green-screen applications securely via the Web from a browser, anywhere, at any time. It's a zero deployment, browser-based model that requires nothing to be installed on the client PC. Find out more (http://www.axeslive.com/products/web-enable.htm)

Modernise the User Experience

Use the 'eXtensions' feature of aXes to enhance applications with a host of standard Windows User Interface constructs – images, dropdowns, checkboxes, calendars, charts, tab panels, and hyperlinks – minimizing workflows and decreasing user training times.
Find out more (http://www.axeslive.com/products/modern-gui.htm)

Remote SQL Query

Easily extract and publish DB2/400 data in a browser or send query output to desktop applications like Microsoft Word and Excel.
Find out more (http://www.axeslive.com/products/sql-query-tool.htm)

Remote Spool Files Access

Provide point-and-click access to output queues and spool files, with print-ready documents available in PDF, XML, HTML, and text formats. Find out more (http://www.axeslive.com/products/spool-file-management.htm)

Monitor/Administrate Remote User Sessions

Give system administrators an online, zero client, browser-based method of managing user sessions, checking server statistics and monitoring the entire aXes system. Provide Helpdesk visibility to each user's screen experience via session "parking" and "shadowing", for ease of application and technical support.
Find out more (http://www.axeslive.com/products/axes-admin.htm)

Planned Support of Microsoft Windows 7 by LANSA

LANSA Software	Planned Support Date
32-bit Windows 7 Validation that the 32-bit LANSA development environment and 32-bit LANSA developed applications will operate under 32-bit Windows 7. **64-bit Windows 7** Validation that the 32-bit LANSA development environment and 32-bit LANSA developed applications will operate under 64-bit Windows 7	
Support Status: Uncertified	LANSA Version 11 SP5
Support Status: Maintained	As soon as possible after GA of LANSA Version 12
Support Status: Confirmed	As soon as possible after the GA of Windows 7 and LANSA Version 12.

Notes

1. LANSA is in the process of verifying the Windows 7 Beta with LANSA V11 SP5 and LANSA V12.0. Once Windows 7 is GA (current GA date from Microsoft is 22/10/2009 http://support.microsoft.com/lifecycle/), LANSA will complete its testing and update the Supported Platforms document.

2. LANSA will make all efforts to Confirm Windows 7 as soon as possible after its General Availability.

The support Status listed above are as per the LANSA Supported Platforms document available at Supported Versions (http://www.lansa.com/support/v11news/supportedversions.htm)

Status: Confirmed

1. Configuration has been comprehensively tested.
2. Response timeframes measured in accordance with the maintenance agreement.
3. Configuration components exist at Technical Support
4. Error fixes or workarounds provided for reported errors.

Status: Maintained

1. Configuration has not been comprehensively tested but substantial evidence suggests that it is expected to operate properly.
2. Response time frames may be delayed in some cases.
3. Configuration components exist at Technical Support
4. Error fixes or workarounds provided for reported errors.

Third Party software shipped with or an integral part of LANSA will also be subject to the same status points as the base LANSA software.

Planned Support of Microsoft Windows Server 2008 R2 by LANSA

LANSA Software	Planned Support Date
Windows Server 2008 R2 Validation that the 32-bit LANSA development environment and 32-bit LANSA developed applications will operate under 64-bit **Windows Server 2008 R2**.	
Support Status: Unsupported (1)	LANSA Version 11 SP5 (2)
Support Status: Maintained (3)	As soon as possible after GA of LANSA Version 12 (2)
Support Status: Confirmed (4)	As soon as possible after the GA of Windows Server 2008 R2 and LANSA Version 12.

Notes

1. There are no plans for Windows Server 2008 R2 support in LANSA V11 SP5.
2. Only LANSA Version 12.0 and later will have any level of support for Windows Server 2008 R2.
1. LANSA is in the process of verifying the Windows Server 2008 R2 Beta with LANSA V12.0. Once Windows Server 2008 R2 is GA (current GA date from Microsoft is 22/10/2009 http://support.microsoft.com/lifecycle/), LANSA will complete its testing and update the Supported Platforms document. There may be LANSA installation issues under Windows Server 2008 R2. Whilst the status is "Maintained", the installation of LANSA under Windows Server 2008 R2 may require a set of special instructions.
2. LANSA will make all efforts to Confirm Windows Server 2008 R2 as soon as possible after of its General Availability.

The support Status listed above are as per the LANSA Supported Platforms document available at Supported Versions (http://www.lansa.com/support/v11news/supportedversions.htm)

Status: Confirmed

1. Configuration has been comprehensively tested.
2. Response timeframes measured in accordance with the maintenance agreement.
3. Configuration components exist at Technical Support
4. Error fixes or workarounds provided for reported errors.

Status: Maintained

1. Configuration has not been comprehensively tested but substantial evidence suggests that it is expected to operate properly.
2. Response time frames may be delayed in some cases.
3. Configuration components exist at Technical Support
4. Error fixes or workarounds provided for reported errors.

Status: Unsupported

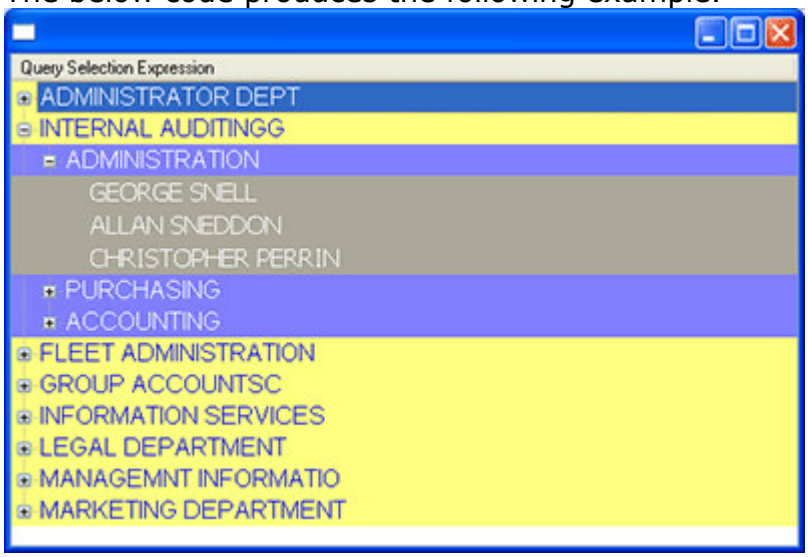
1. Configuration may or may not have been tested.
2. Configuration may or may not work.
3. No support provided.

Third Party software shipped with or an integral part of LANSA will also be subject to the same status points as the base LANSA software.

How to create a tree view with a different background colour for each level

In order to achieve this the ViewStyle property of the tree is used. When this is set to unlevelled, the effect is columnar trees as seen in the LANSA IDE.

The below code produces the following example.



Solution

The sample code below displays this effect. Simply copy and paste into a form compile and run.

```
Function Options(*DIRECT)
BEGIN_COM ROLE(*EXTENDS #PRIM_FORM) CLIENTHEIGHT(500)
CLIENTWIDTH(528) HEIGHT(534)
LAYOUTMANAGER(#ATLM_1) LEFT(499) TOP(151) WIDTH(536)

DEFINE_COM CLASS(#PRIM_ATLM) NAME(#ATLM_1)
DEFINE_COM CLASS(#PRIM_ATLI) NAME(#ATLI_1) ATTACHMENT(Center)
MANAGE(#Tree) PARENT(#ATLM_1)

DEFINE_COM CLASS(#PRIM_TRVW) NAME(#Tree) COLUMNBUTTONHEIGHT(18)
COMPONENTVERSION(2)
DISPLAYPOSITION(1) FULLROWSELECT(True) HEIGHT(500) LEFT(0)
PARENT(#COM_OWNER) TABPOSITION(1)
TOP(0) VIEWSTYLE(UnLevelled) WIDTH(528)

DEFINE_COM CLASS(#PRIM_TVCL) NAME(#TVCL_1) LEVEL(1) PARENT(#Tree)
SOURCE(#DEPTMENT) VISIBLE(False)
DEFINE_COM CLASS(#PRIM_TVCL) NAME(#TVCL_2) LEVEL(1) PARENT(#Tree)
SOURCE(#SECTION) VISIBLE(False)
DEFINE_COM CLASS(#PRIM_TVCL) NAME(#TVCL_3) LEVEL(1) PARENT(#Tree)
SOURCE(#EMPNO) VISIBLE(False)
```

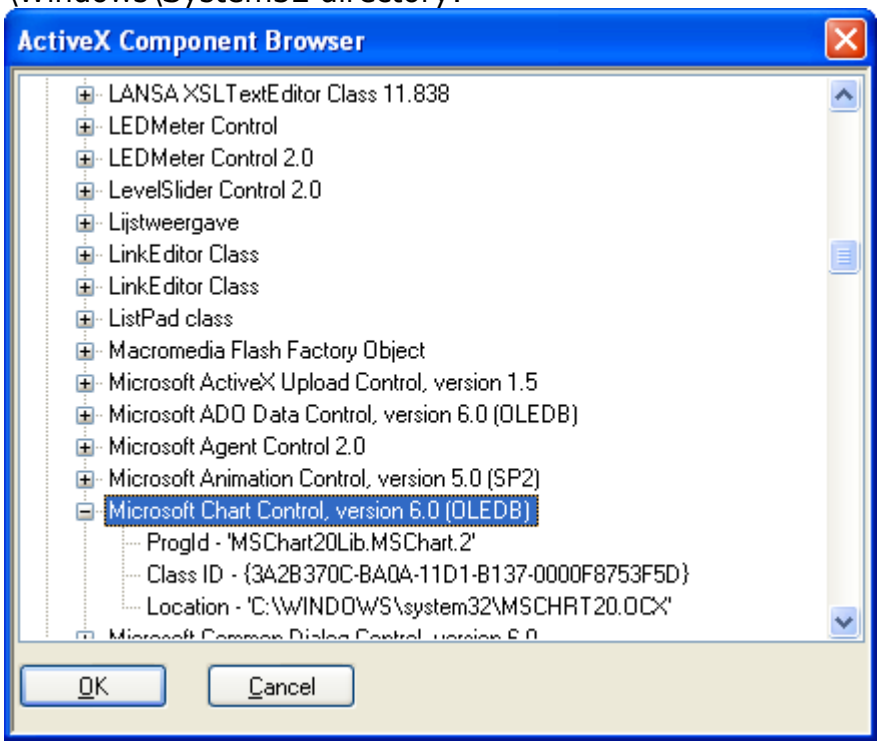
```
DEFINE_COM CLASS(#PRIM_TVCL) NAME(#TVCL_4) DISPLAYPOSITION(1)
LEVEL(1) PARENT(#Tree)
SOURCE(#STD_QSEL) WIDTHTYPE(Remainder)

Evtroutine Handling(#Com_owner.CreateInstance)
#Com_owner.AddDepartments
Endroutine

Mthroutine Name(AddDepartments)
Select Fields(*All) From_File(Deptab)
#Com_owner.AddEntry( #Deptdesc #UF_VS004 )
#Com_Owner.AddSections( #Tree.Currentitem )
Endselect
Endroutine

Mthroutine Name(AddSections)
Define_Map For(*Input) Class(#prim_tvit) Name(#ParentItem)
Pass(*by_reference)
Select Fields(*All) From_File(Sectab) With_Key(#Deptment)
#Com_owner.AddEntry( #Secdesc #UF_VS002 #ParentItem )
#Com_Owner.AddEmployees( #Tree.Currentitem )
Endselect
Endroutine

Mthroutine Name(AddEmployees)
Define_Map For(*Input) Class(#prim_tvit) Name(#ParentItem)
Pass(*by_reference)
Select Fields(*All) From_File(Pslmst1) With_Key(#Deptment #Section)
#Com_owner.AddEntry( (#GiveName + " " + #Surname) #VF_VS104
#ParentItem )
Endselect
Endroutine

Mthroutine Name(AddEntry) Access(*private)
Define_Map For(*Input) Class(#prim_alph) Name(#Text)
Define_Map For(*Input) Class(#prim_vs) Name(#VisualStyle)
Pass(*by_reference)
Define_Map For(*Input) Class(#prim_tvit) Name(#ParentItem)
Mandatory(*null) Pass(*by_reference)
Define_Map For(*Result) Class(#prim_tvit) Name(#Result) Mandatory(*null)
Pass(*by_reference)

#Std_qsel := #Text

Add_Entry To_List(#Tree)

#Tree.Currentitem.VisualStyle <= #VisualStyle
#Tree.Currentitem.ParentItem <= #ParentItem

#Result <= #Tree.Currentitem

Endroutine

End_Com
```

External Graphs in VL

(Thanks to Theo Mulder from Koninklijke Wagenborg, the Netherlands)

This example uses the Microsoft ActiveX control MSCHRT20.OCX.

1.
Create in the VL IDE a new ActiveX Control component. The name of the component is CHART2003. It should use the Microsoft MSCHRT20.OCX in the \Windows\System32 directory:



The source of the ActiveX component:

```
Begin_Com Role(*EXTENDS #PRIM_MCCO) Progid('MSChart20Lib.MSChart.2')
Typelibid('{65E121D4-0C60-11D2-A9FC-0000F8754DA1}2.0,0')
End_Com
```

2.
Create a new form called Graph (any name will do) and copy/paste source below into the form:

```
Function Options(*DIRECT)
Begin_Com Role(*EXTENDS #PRIM_FORM) Clientheight(384) Clientwidth(978) Height(418)
Layoutmanager(#ATLM_1) Width(986)
Define_Com Class(#CHART2003.MSChart) Name(#CHART2003) Displayposition(1) Height(336)
Left(4) Parent(#GPBX_1) Tabposition(1) Top(8) Width(970)
Define_Com Class(#PRIM_ATLM) Name(#ATLM_1)
Define_Com Class(#PRIM_ATLI) Name(#ATLI_1) Attachment(Center) Manage(#CHART2003)
Parent(#ATLM_1)
Define_Com Class(#PRIM_GPBX) Name(#GPBX_1) Displayposition(1) Height(348)
Layoutmanager(#ATLM_2) Left(0) Parent(#COM_OWNER) Tabposition(1) Tabstop(False) Top(0)
Width(978)
```

```
Define_Com Class(#PRIM_ATLI) Name(#ATLI_2) Attachment(Center) Manage(#GPBX_1)
Parent(#ATLM_1)
Define_Com Class(#PRIM_GPBX) Name(#GPBX_2) Displayposition(2) Height(36)
Layoutmanager(#FWLM_1) Left(0) Parent(#COM_OWNER) Tabposition(2) Tabstop(False) Top(348)
Width(978)
Define_Com Class(#PRIM_ATLI) Name(#ATLI_3) Attachment(Bottom) Manage(#GPBX_2)
Parent(#ATLM_1)
Define_Com Class(#PRIM_ATLM) Name(#ATLM_2)
Define_Com Class(#PRIM_ATLI) Name(#ATLI_4) Attachment(Center) Manage(#CHART2003)
Parent(#ATLM_2)
Define_Com Class(#PRIM_FWLM) Name(#FWLM_1)
Define_Com Class(#PRIM_RDBN) Name(#RDBN_1) Caption('2D Line') Displayposition(1) Left(4)
Parent(#GPBX_2) Tabposition(1) Top(8)
Define_Com Class(#PRIM_FWLI) Name(#FWLI_1) Manage(#RDBN_1) Parent(#FWLM_1)
Define_Com Class(#PRIM_RDBN) Name(#RDBN_2) Caption('2D Bar') Displayposition(2) Left(134)
Parent(#GPBX_2) Tabposition(2) Top(8)
Define_Com Class(#PRIM_FWLI) Name(#FWLI_2) Manage(#RDBN_2) Parent(#FWLM_1)
Define_Com Class(#PRIM_RDBN) Name(#RDBN_3) Caption('2D Area') Displayposition(3) Left(264)
Parent(#GPBX_2) Tabposition(3) Top(8)
Define_Com Class(#PRIM_FWLI) Name(#FWLI_3) Manage(#RDBN_3) Parent(#FWLM_1)
Define_Com Class(#PRIM_RDBN) Name(#RDBN_4) Caption('2D Pie') Displayposition(4) Left(394)
Parent(#GPBX_2) Tabposition(4) Top(8)
Define_Com Class(#PRIM_FWLI) Name(#FWLI_4) Manage(#RDBN_4) Parent(#FWLM_1)
Define_Com Class(#PRIM_RDBN) Name(#RDBN_5) Caption('3D Line') Displayposition(5) Left(524)
Parent(#GPBX_2) Tabposition(5) Top(8)
Define_Com Class(#PRIM_FWLI) Name(#FWLI_5) Manage(#RDBN_5) Parent(#FWLM_1)
Define_Com Class(#PRIM_RDBN) Name(#RDBN_6) Caption('3D Bar') Displayposition(6) Left(654)
Parent(#GPBX_2) Tabposition(6) Top(8)
Define_Com Class(#PRIM_FWLI) Name(#FWLI_6) Manage(#RDBN_6) Parent(#FWLM_1)
Define_Com Class(#PRIM_RDBN) Name(#RDBN_7) Caption('3D Area') Displayposition(7) Left(784)
Parent(#GPBX_2) Tabposition(7) Top(8)
Define_Com Class(#PRIM_FWLI) Name(#FWLI_7) Manage(#RDBN_7) Parent(#FWLM_1)

Define Field(#CNTR1) Type(*DEC) Length(3) Decimals(0)
Define Field(#CNTR2) Type(*DEC) Length(3) Decimals(0)
Define Field(#WRD1) Type(*DEC) Length(3) Decimals(0)
Define Field(#WRD2) Type(*DEC) Length(3) Decimals(0)
Define Field(#WRD3) Type(*DEC) Length(3) Decimals(0)

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

Endroutine

Evtroutine Handling(#CHART2003.Initialize) Options(*NOCLEARMESSAGES *NOCLEARERRORS)
```

*** Chart: lots of properties related to the visualisation can be set with the properties screen of the component = right mouse button on the component.**

*** When you move your mouse with your CTRL key pressed, you can rotate the 3D graphics.**

*** Column is vertical ; Row = horizontal ; possibility 'series in row' properties on chart-tab**

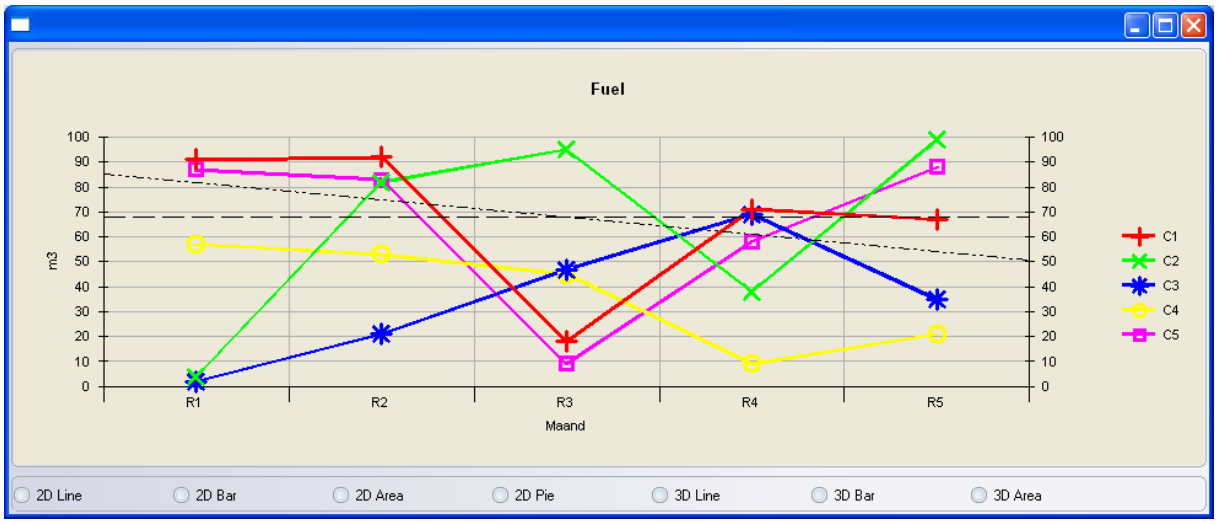
```
#CHART2003.ColumnCount := 3
#CHART2003.RowCount := 12
#CHART2003.chartType := CHART2003.VtChChartType2dLine
#RDBN_1.ButtonChecked := True
* VtChChartType provides chart type options.
* Constant Description
* VtChChartType3dBar = 3D Bar
* VtChChartType2dBar = 2D Bar
* VtChChartType3dLine = 3D Line
* VtChChartType2dLine = 2D Line
```

```
* VtChChartType3dArea = 3D Area
* VtChChartType2dArea = 2D Area
* VtChChartType3dStep = 3D Step
* VtChChartType2dStep = 2D Step
* VtChChartType3dCombination = 3D Combination
* VtChChartType2dCombination = 2D Combination
* VtChChartType2dPie = 2D Pie
* VtChChartType2dXY = 2D XY

#CHART2003.Plot.Axis<0>.AxisTitle := 'X-as'
#CHART2003.Plot.Axis<1>.AxisTitle := 'Linker Y-as'
#CHART2003.Plot.Axis<2>.AxisTitle := 'Rechter Y-as'

#CHART2003.TitleText := 'Fuel consumption'

* row
Begin_Loop Using(#CNTR1) To(12)
#CHART2003.Row := #CNTR1
* label row
Case Of_Field(#CNTR1)
When Value_Is('= 1')
#CHART2003.Rowlabel := JAN
#WRD1 := 10
#WRD2 := 10
#WRD3 := 50
When Value_Is('= 2')
#CHART2003.Rowlabel := FEB
#WRD1 := 70
#WRD2 := 40
#WRD3 := 120
When Value_Is('= 3')
#CHART2003.Rowlabel := MAR
#WRD1 := 40
#WRD2 := 30
#WRD3 := 50
When Value_Is('= 4')
#CHART2003.Rowlabel := APR
#WRD1 := 110
#WRD2 := 140
#WRD3 := 70
When Value_Is('= 5')
#CHART2003.Rowlabel := MAY
#WRD1 := 60
#WRD2 := 50
#WRD3 := 90
When Value_Is('= 6')
#CHART2003.Rowlabel := JUN
#WRD1 := 40
#WRD2 := 80
#WRD3 := 90
When Value_Is('= 7')
#CHART2003.Rowlabel := JUL
#WRD1 := 50
#WRD2 := 40
#WRD3 := 80
When Value_Is('= 8')
#CHART2003.Rowlabel := AUG
#WRD1 := 60
#WRD2 := 110
#WRD3 := 50
When Value_Is('= 9')
#CHART2003.Rowlabel := SEP
#WRD1 := 30
#WRD2 := 60
```

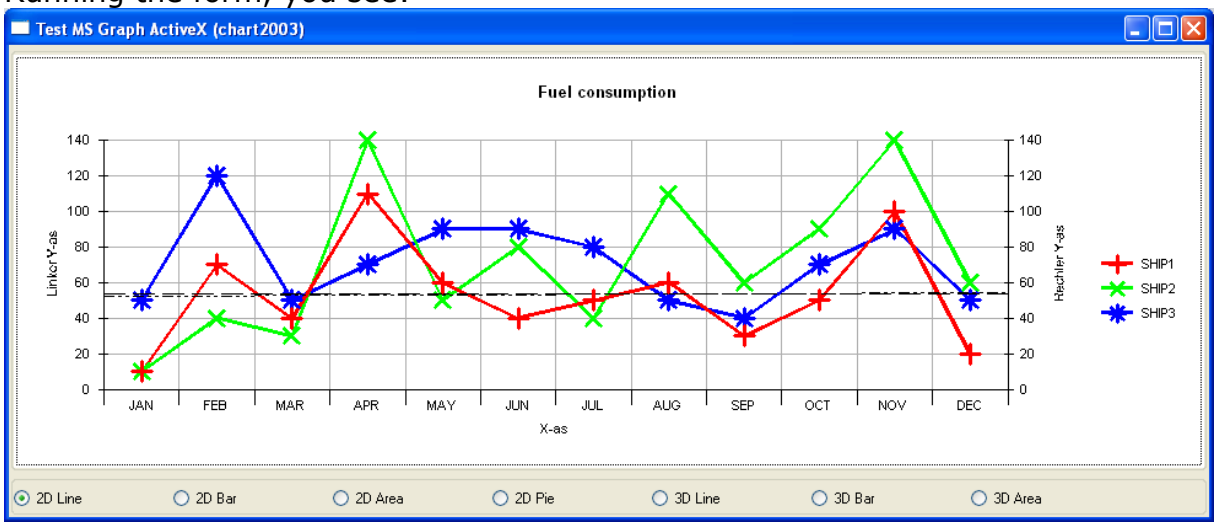
```
#WRD3 := 40
When Value_Is('= 10')
#CHART2003.Rowlabel := OCT
#WRD1 := 50
#WRD2 := 90
#WRD3 := 70
When Value_Is('= 11')
#CHART2003.Rowlabel := NOV
#WRD1 := 100
#WRD2 := 140
#WRD3 := 90
When Value_Is('= 12')
#CHART2003.Rowlabel := DEC
#WRD1 := 20
#WRD2 := 60
#WRD3 := 50
Endcase
* column (values)
Begin_Loop Using(#CNTR2) To(3)
#CHART2003.Column := #CNTR2
Case Of_Field(#CNTR2)
When Value_Is('= 1')
#CHART2003.Data := #WRD1.AsString
When Value_Is('= 2')
#CHART2003.Data := #WRD2.AsString
When Value_Is('= 3')
#CHART2003.Data := #WRD3.AsString
Endcase
End_Loop
End_Loop

* aantal schepen
#CHART2003.Column := 1
#CHART2003.Columnlabel := Ship1
#CHART2003.Column := 2
#CHART2003.columnlabel := Ship2
#CHART2003.column := 3
#CHART2003.Columnlabel := Ship3


Endroutine

* 2D Line
Evtroutine Handling(#RDBN_1.Click)
#CHART2003.chartType := CHART2003.VtChChartType2dLine
Endroutine
* 3D Bar
Evtroutine Handling(#RDBN_2.Click)
#CHART2003.chartType := CHART2003.VtChChartType2dBar
Endroutine
* 2D Area
Evtroutine Handling(#RDBN_3.Click)
#CHART2003.chartType := CHART2003.VtChChartType2dArea
Endroutine
* 2d Pie
Evtroutine Handling(#RDBN_4.Click)
#CHART2003.chartType := CHART2003.VtChChartType2dPie
Endroutine
* 2D Line
Evtroutine Handling(#RDBN_5.Click)
#CHART2003.chartType := CHART2003.VtChChartType3dLine
Endroutine
* 3D Bar
Evtroutine Handling(#RDBN_6.Click)
```

```
#CHART2003.chartType := CHART2003.VtChChartType3dBar
Endroutine
* 3d Area
Evtroutine Handling(#RDBN_7.Click)
#CHART2003.chartType := CHART2003.VtChChartType3dArea
Endroutine

End_Com
```

The VL editor looks like this:



Running the form, you see:



Install or upgrade can fail if the job message queue reaches maximum size

Upgrade of system creates a large joblog and subsequently fails with the following message.

```
CPF2526 Information    Message . . . . . .: Job Message queue for <job
name> can not be extended.  Job ended.
                       Cause . . . . . .: The size of the job message
queue for <job name> reached the maximum size. The maximum size is defined
in the JOBMSGQMX value of the job description used by the job. When the job
message queue becomes full, the job attribute, JOBMSGQFL, identifies
the action to take.
```

Solution

There are 2 things which can be done to ensure that the Upgrade does not fail due to a large JOBLOG.

1. Change the system value in order to increase the default Maximum size of joblogs - JOBMSGQMX. It can be set to a maximum of 64mb.
2. Change the system value which directs the action to be taken when joblog has reached maximum size (i.e Change the system value QJOBMSGQFL from *NOWRAP to *WRAP).

Altova MapForce will NOT work with recent IBM System i Access for Windows service level

Altova MapForce (any version) will be unable to successfully establish ODBC connections to your iSeries database if you install these recent service packs for IBM System i Access For Windows on the client computer on which you are using LANSA Composer and MapForce:

- Service level SI32504 for IBM System i Access for Windows V6R1M0
- Service level SI32972 for IBM System i Access for Windows V5R4M0

We believe this is a problem with the iSeries Access ODBC driver installed with these service packs. For SI32504 at V6R1M0, the driver version is 12.00.02.00. Note that other ODBC applications may function correctly, but this is simply because they use ODBC in a different way to Altova MapForce.

NOTE

If you are having problems establishing the ODBC connection to your System i database from Altova MapForce but it does not match this description, then please also refer to the article:

Latest DATABASE GROUP PTFs for i5/OS V6R1 may cause temporary/transient problem with Altova MapForce:
(http://www.lansa.com/support/notes/p0373.htm)

Corrected in

IBM have just released service level SI34289 for IBM System i Access for Windows V6R1M0. This service level updates the ODBC driver to version 12.00.05.00, which corrects the problem and works correctly with Altova MapForce.

We understand that IBM will shortly issue an update for V5R4M0 which we expect will correct the problem at that level.

What you should do

Do NOT install the following updates to IBM System i Access for Windows:

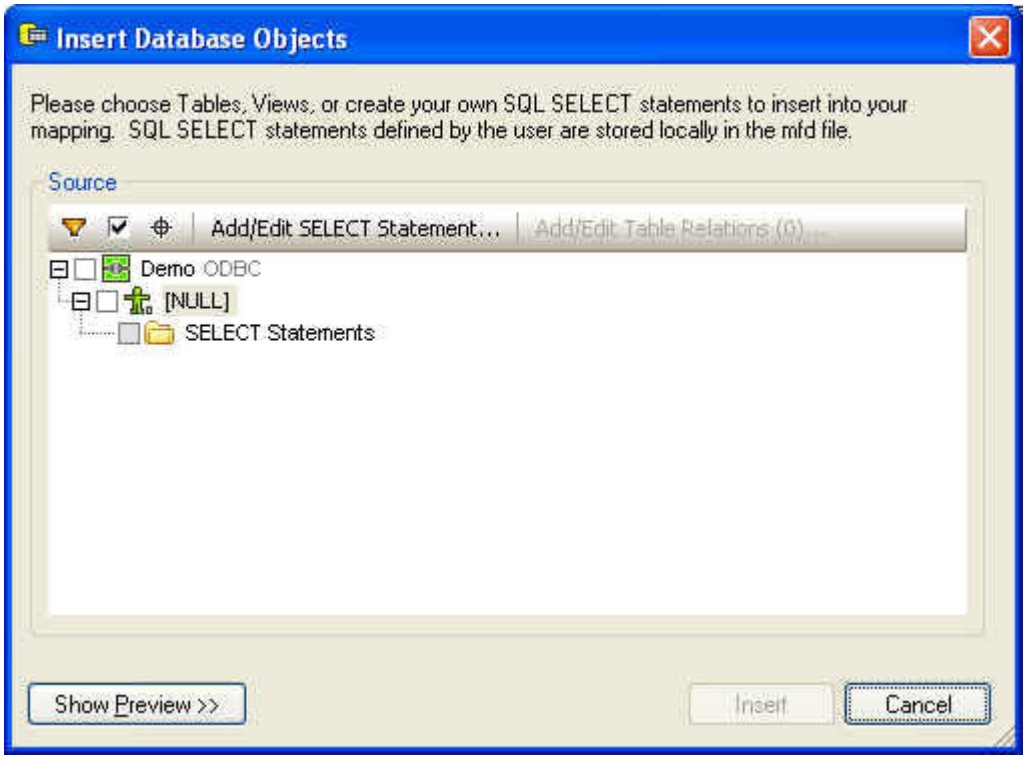
- service level SI32504 for IBM System i Access for Windows V6R1M0
- service level SI32972 for IBM System i Access for Windows V5R4M0

If you have already installed one of the above:
- if you are using IBM System i Access for Windows V6R1M0, update to the later service level SI34289
- if you are using IBM System i Access for Windows V5R4M0, either:
a) wait for the next IBM update we expect to be released shortly, OR
b) uninstall IBM System i Access for Windows and re-install at an earlier service level

Symptom

NOTE: The impact of this is limited to design-time. Existing transformation maps will continue to operate as before (The client PC is not involved and ODBC is NOT used at run-time).

On MapForce version 2009 SP1, the symptom is that, having established the ODBC connection, the *Insert Database Objects* displays *[NULL]* instead of the schema/library name (as shown below). On earlier versions of MapForce (eg: 2008R2SP2 as shipped with LANSA Composer version 2.0), the available library/schema names are simply not shown at all.



Changes to IBM SQL Engine affects performance in Logical Views keyed by numeric fields

Symptom

Recently IBM have introduced changes to the SQL engine so that it no longer supports unsigned numeric key fields for DDS created files. APAR MA36289 discusses this behavior and how it causes an issue when trying to use Logical Views keyed by numeric fields. In order to maintain some backwards compatibility, IBM have created a fix which creates SQL Indexes for these logical views 'on-the-fly' at runtime.

The updated SQL engine and subsequent compatibility fix is shipped in V6R1, and it is also delivered in PTFs for V5R4:
MF44150
MF44306

How this affects LANSA applications

It is important to note that the creation of the indexes at runtime will affect ANY iSeries application accessing these logical files using SQL. In addition, after applying the PTFs mentioned above or upgrading to V6R1, LANSA Applications making use of:

- SELECT_SQL commands
- LANSA Integrator SQL Service
- Certain LANSA Client queries (which use SQL behind the scenes)

Will have performance issues if they are used to access Logical Views containing numeric key fields with unsigned ordering.

Solution

The fastest solution is to rebuild the logical views so that they do not use unsigned ordering on the numeric key fields. In most cases, unsigned numeric ordering was not selected intentionally, in which case, changing the ordering type to Signed may not cause significant application behaviour changes. However, you should consider any impacts on your application behaviour prior to making this change.

There is a Visual LANSA IDE tool available from LANSA Support that can identify which logical files need to be updated and it can be supplied to customers who have many updates to make. Future releases of the Visual LANSA IDE will also make 'Signed' as the default ordering type for newly created numeric logical keys to prevent this being selected unless specifically required.

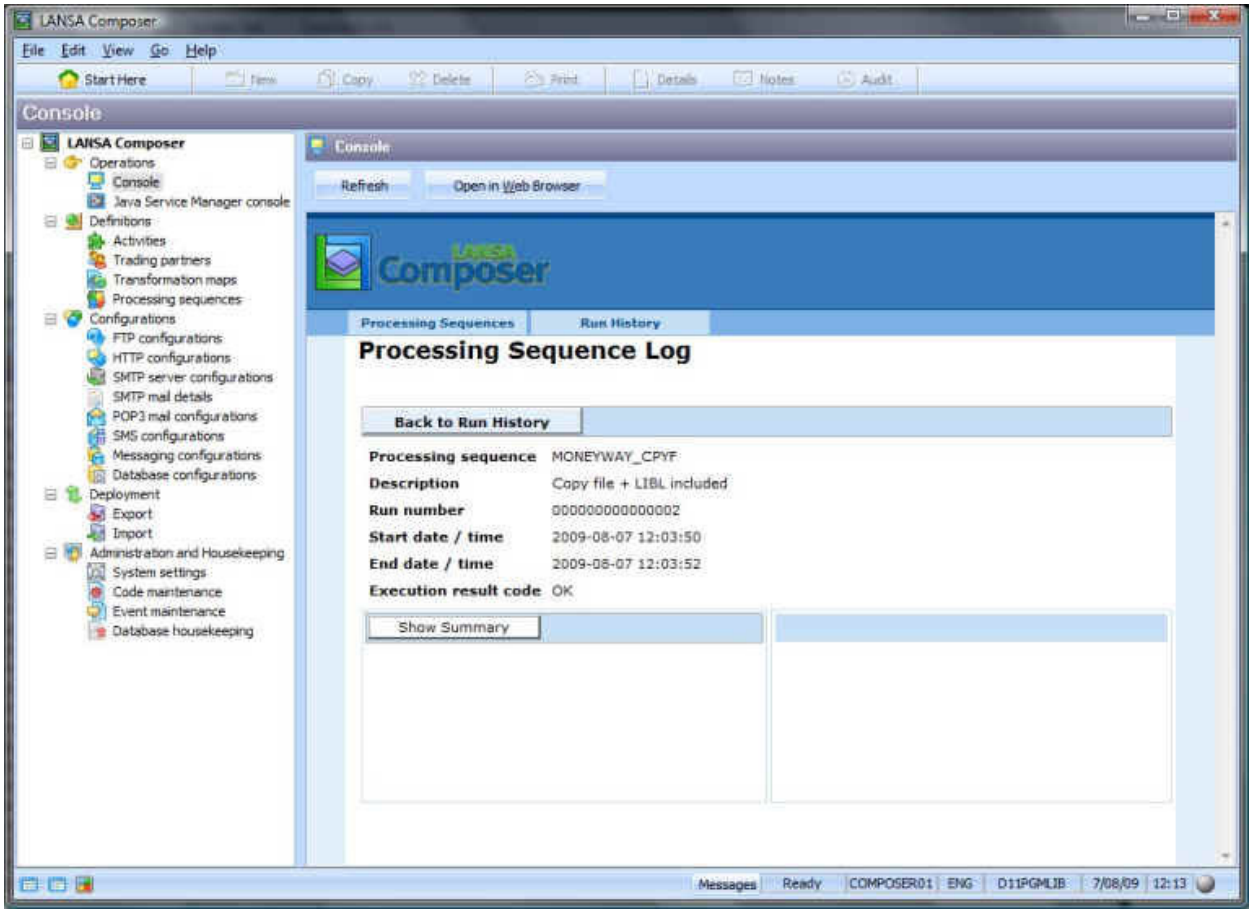
Please contact your local support agent for more information about either of these fixes.

WAM treeview issue introduced in EPC845 causes incomplete display of Processing Sequence Log in Console

Issue

EPC845 introduces an issue for the Processing Sequence log display in LANSA Composer's Operations Console. This affects both IBM i and Windows backend.

Specific WAM related changes in EPC845 causes TreeViews in WAMS to be parsed incorrectly. In LANSA Composer's Operations Console, the result is that the Processing Sequence Log display appears to be empty.



Solution

A fix for each Server type (Windows and IBM i) has been created which fixes the parsing issue. This can be issued upon request and will be shipped in the next EPC.

Note: If the symptoms still persist after applying the patch, then most likely, you will need to clear the cache on the Browser.

Additionally if the Server is Windows, then the following folders may need to be cleared as well:

- <Root Drive>:\<LANSA FOLDER>\WebServer\IISPlugin\xsl.cache
- <Root Drive>:\<LANSA FOLDER>\X_WIN95\X_LANSA\x_<PARTITION NAME>\web\document.cache

Note: This issue can potentially cause issues anywhere treeviews are used - i.e. it is not only restricted to LANSA Composer.