http://www.lansa.com

LANSA

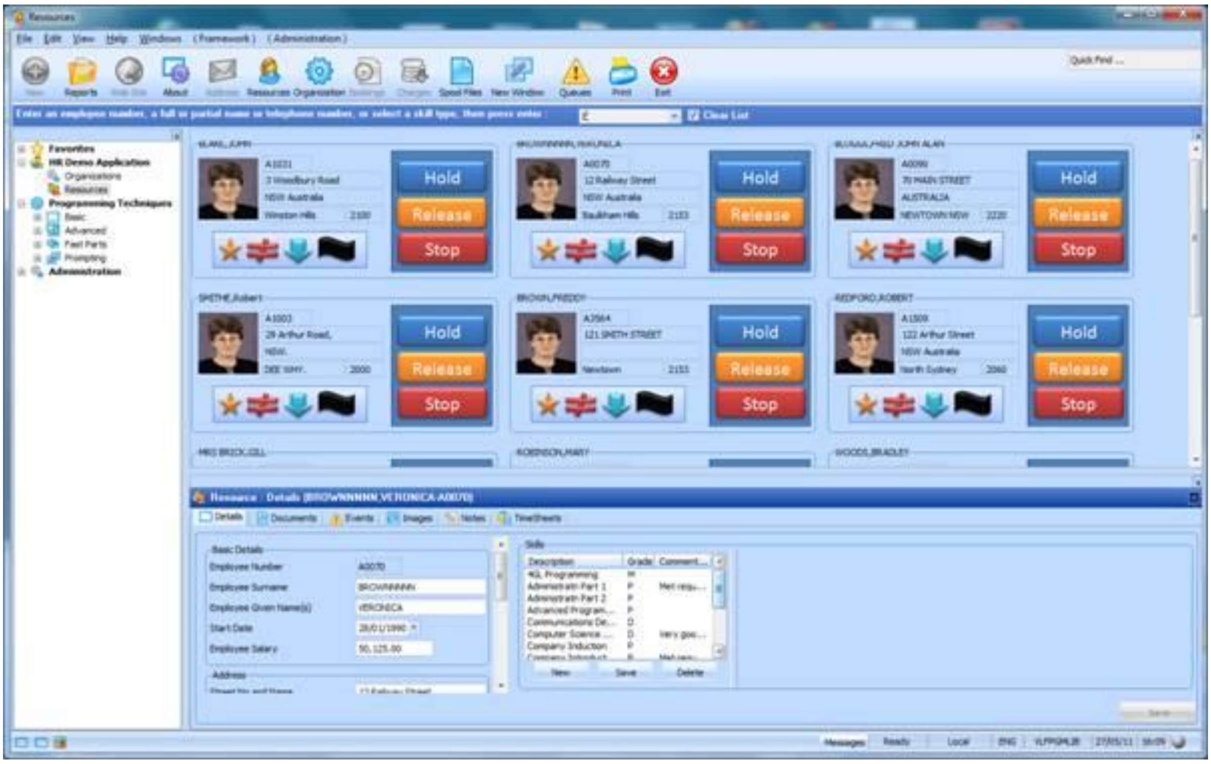
Newsletter
September 2011

Enhanced VLF Instance List Visualization

In the VLF the "instance list" concept tends to be taken very literally (in a visual sense) by many developers.
Being a keyed programmatic list of text strings like product numbers, names, addresses, etc it mostly ends up getting displayed that way in either a flat list or in a tree.
However an instance list is really meant to be an in memory list of keys and identifiers.

You can visualize in much more creative ways by using a custom instance list browser.

Look at the example below. It shows the VLF instance list completely different than the normal flat list or tree view. But it is the instance list!



Basic Details

Employee Number	A0070
Employee Surname	BROWNNNNN
Employee Given Name(s)	VERONICA
Start Date	28/01/1990
Employee Salary	50,125.00

Skills

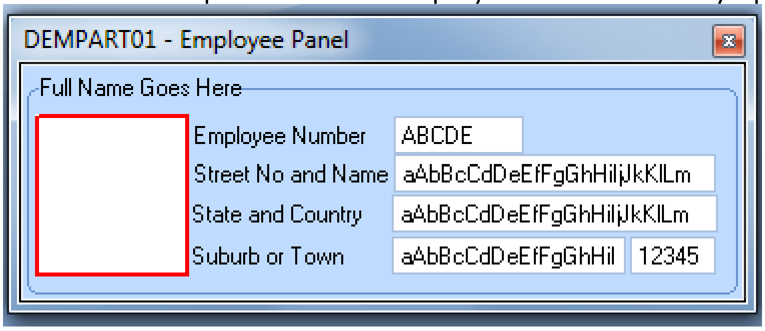
Description	Grade	Comment...
4GL Programming	M	
Administratn Part 1	P	Met requ...
Administratn Part 2	P	
Advanced Program...	P	
Communications De...	D	
Computer Science ...	D	Very goo...
Company Induction	P	

In This Issue

To create such an example, you should to this:

1. Create and compile VL reusable part DEMOPART01 – The Entry Visualizer

This is a small panel used to display the details of any specified employee.



DEMPART01 - Employee Panel

Full Name Goes Here

Employee Number ABCDE

Street No and Name aAbBcCdDeEfFgGhHiIjJkKlLm

State and Country aAbBcCdDeEfFgGhHiIjJkKlLm

Suburb or Town aAbBcCdDeEfFgGhHiI 12345

The source of this component:

```
Function Options(*DIRECT)
Begin_Com Role(*EXTENDS #PRIM_PANL) Displayposition(1) Height(116) Layoutmanager(#ATLM_1)
Left(0) Tabposition(1) Top(0) Width(359)
Define_Com Class(#PRIM_GPBX) Name(#GPBX_1) Caption('Full Name Goes Here') Displayposition(1)
Height(112) Left(0) Parent(#COM_OWNER) Tabposition(1) Tabstop(False) Top(4) Width(359)
Define_Com Class(#PRIM_ATLM) Name(#ATLM_1)
Define_Com Class(#PRIM_ATLI) Name(#ATLI_1) Attachment(Center) Manage(#GPBX_1) Margintop(4)
Parent(#ATLM_1)
Define_Com Class(#PRIM_IMGE) Name(#IMGE_1) Displayposition(2) Height(72) Parent(#GPBX_1)
Tabposition(1) Top(24) Width(70)
Define_Com Class(#Empno.Visual) Name(#Empno) Displayposition(1) Height(19) Left(85)
Marginleft(100) Parent(#GPBX_1) Readonly(True) Tabposition(2) Top(21) Usepicklist(False)
Width(165)
Define_Com Class(#ADDRESS1.Visual) Name(#ADDRESS1) Displayposition(3) Height(19) Left(85)
Marginleft(100) Parent(#GPBX_1) Readonly(True) Tabposition(3) Top(40) Usepicklist(False)
Width(261)
Define_Com Class(#ADDRESS2.Visual) Name(#ADDRESS2) Displayposition(4) Height(19) Left(84)
Marginleft(100) Parent(#GPBX_1) Readonly(True) Tabposition(4) Top(81) Usepicklist(False)
Width(217)
Define_Com Class(#ADDRESS3.Visual) Name(#ADDRESS3) Displayposition(5) Height(19) Left(84)
Marginleft(100) Parent(#GPBX_1) Readonly(True) Tabposition(5) Top(60) Usepicklist(False)
Width(263)
Define_Com Class(#POSTCODE.Visual) Name(#POSTCODE) Displayposition(6) Height(19) Left(303)
Marginleft(0) Parent(#GPBX_1) Readonly(True) Tabposition(6) Top(81) Usepicklist(False)
Width(43)
Define_Com Class(#PRIM_STTC) Name(#SelectedIndication) Displayposition(7) Drawtype(Rectangle)
Height(81) Left(6) Linewidth(2) Parent(#GPBX_1) Tabposition(7) Top(20) Visible(False)
Visualstyle(#VS_WARN) Width(77)

Define_Pty Name(FullName) Set(SetFullName)
Define_Pty Name(Address1) Set(*Auto #ADDRESS1)
Define_Pty Name(Address2) Set(*Auto #ADDRESS2)
Define_Pty Name(Address3) Set(*Auto #ADDRESS3)
Define_Pty Name(PostCode) Set(*Auto #POSTCODE)
Define_Pty Name(EmpNo) Get(*auto #Empno) Set(*auto #Empno)
Define_Pty Name(Department) Get(*auto #DEPTMENT) Set(*auto #Deptment)
Define_Pty Name(Section) Get(*AUTO #SECTION) Set(*auto #Section)
Define_Pty Name(VisibleImage) Set(SetImage)

Define_Com Class(#Prim_Boln) Name(#Selected)
Define_Pty Name(Selected) Get(*Auto #Selected) Set(SetSelected)

Define_Evt Name(PanelSelected)
Define_Evt Name(PanelUnSelected)

Ptyroutine Name(SetSelected)
Define_Map For(*input) Class(#Prim_Boln) Name(#State)
If (#State <> #Selected)
#Selected := #State
#SelectedIndication.Visible := #Selected
Endif
Endroutine
```

```
Ptyroutine Name(SetFullName)
Define_Map For(*input) Class(#Prim_alph) Name(#Value)
#GPBX_1.Caption := #Value
Endroutine

Ptyroutine Name(SetImage)
Define_Map For(*input) Class(#Prim_alph) Name(#Value)
#IMGE_1.FileName := #Value
Endroutine

Evtroutine Handling(#IMGE_1.Click)
If (#Selected)
Signal Event(PanelUnSelected)
Else
Signal Event(PanelSelected)
Endif
Endroutine

End_Com
```

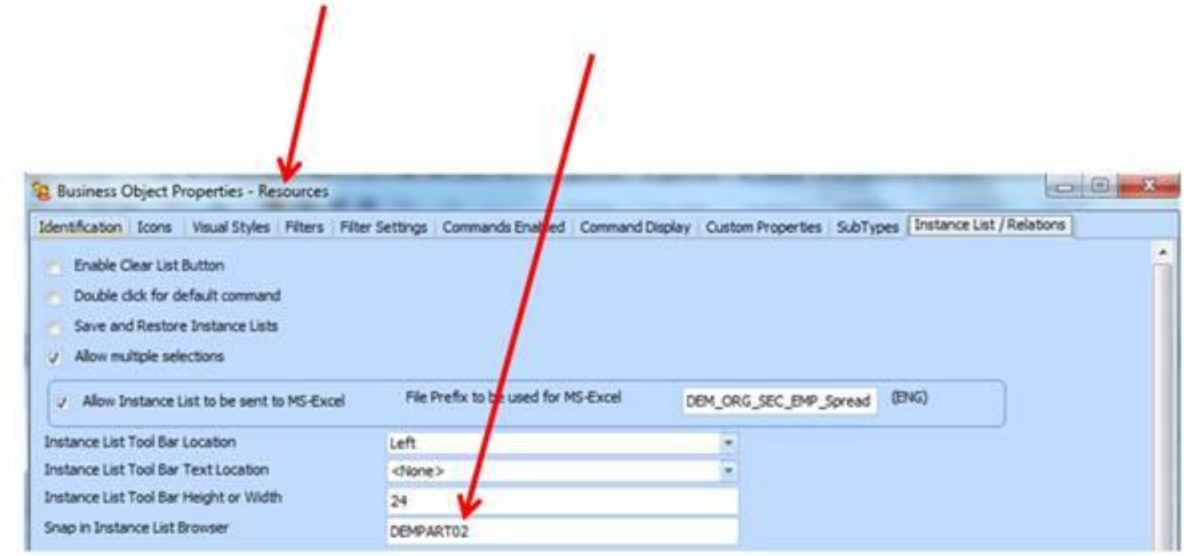
2. Create and compile VL reusable part DEMOPART02 – The Custom Instance List Browser

This is a custom instance list browser.

```
Function Options(*DIRECT)

Begin_Com Role(*EXTENDS #VF_AC012) Horizontalscroll(True) Layoutmanager(#FWLM_1)
Verticalscroll(True)

* Collection of employee details

Define_Com Class(#PRIM_KCOL<#DEMPART01 #EMPNO>) Name(#EmployeeListofPanels) Style(Collection)
Define_Com Class(#PRIM_ACOL<#PRIM_FWLI>) Name(#EmployeeLayoutItems)
Define_Com Class(#PRIM_FWLM) Name(#FWLM_1)

* -----------------------------------------------
* Redefine the standard uClearInstanceList method
* -----------------------------------------------

Mthroutine Name(uClearInstanceList) Options(*Redefine)

* Destroy all employee visualiser panels and layout items

For Each(#Gitem) In(#EmployeeLayoutItems)
#Gitem.Manage <= *null
Endfor

#EmployeeLayoutItems.RemoveAll
#EmployeeListofPanels.RemoveAll

Endroutine

* ------------------------------------------
* Redefine the standard uAddListEntry method
* ------------------------------------------

Mthroutine Name(uAddListEntry) Options(*Redefine)

Define_Com Class(#dempart01) Name(#NewEmp) Reference(*DYNAMIC)
Define_Com Class(#PRIM_FWLI) Name(#NewEmpLayoutItem) Reference(*DYNAMIC)

* Create the new employee visualiser panel and add to visible list ....

#Empno := #AKey3

If (#EmployeeListofPanels<#Empno> *Is *NULL)

* Create an employee visualizer panel and a layout item

Set_Ref Com(#NewEmp) To(*Create_as #DemPart01)
Set_Ref Com(#NewEmpLayoutItem) To(*Create_as #PRIM_FWLI)

* Set up the layout item

Set Com(#NewEmpLayoutItem) Manage(#NewEmp) Parent(#FWLM_1)

* Set up the employee visualiser panel (ie: the instance list entry visualization)

#NewEmp.Parent <= #COM_OWNER
#NewEmp.Department := #AKey1
#NewEmp.Section := #AKey2
```

```
#NewEmp.Empno := #AKey3
#NewEmp.FullName := #VisualId1
#NewEmp.Address1 := #AColumn1
#NewEmp.Address2 := #AColumn2
#NewEmp.Address3 := #AColumn3
#NewEmp.PostCode := #NColumn1
#NewEmp.VisibleImage := *part_dir_execute + DF_IM001.gif

* Track the employee visualiser panel and the layout item

#EmployeeListofPanels<#Empno> <= #NewEmp
#EmployeeLayoutItems.Insert Item(#NewEmpLayoutItem)

#NewEmp.Realize

Endif

* Finished

Endroutine

* ---------------------------------------------------------------------------
* Listen to the employee visualization panels and flick the events to the VLF
* ---------------------------------------------------------------------------

Evtroutine Handling(#EmployeeListofPanels<>.PanelSelected) Com_Sender(#SenderPanel)

For Each(#OtherPanel) In(#EmployeeListofPanels)
If ((#OtherPanel.Selected) And (#OtherPanel *IsNotEqualTo #SenderPanel))
#OtherPanel.Selected := false
Signal Event(DropCurrentInstance) Akey1(#OtherPanel.Department) Akey2(#OtherPanel.Section)
Akey3(#OtherPanel.EmpNo)
Endif
Endfor

#SenderPanel.Selected := True

Signal Event(SetCurrentInstance) Akey1(#SenderPanel.Department) Akey2(#SenderPanel.Section)
Akey3(#SenderPanel.EmpNo)

Endroutine

Evtroutine Handling(#EmployeeListofPanels<>.PanelUnSelected) Com_Sender(#SenderPanel)

For Each(#OtherPanel) In(#EmployeeListofPanels)
If ((#OtherPanel.Selected) And (#OtherPanel *IsNotEqualTo #SenderPanel))
#OtherPanel.Selected := false
Signal Event(DropCurrentInstance) Akey1(#OtherPanel.Department) Akey2(#OtherPanel.Section)
Akey3(#OtherPanel.EmpNo)
Endif
Endfor

#SenderPanel.Selected := False

Signal Event(SetCurrentInstance) Akey1(#SenderPanel.Department) Akey2(#SenderPanel.Section)
Akey3(#SenderPanel.EmpNo)

Endroutine

End_Com
```

3. Snap the custom browser into the shipped "Resources" business object


Business Object Properties - Resources

□ Enable Clear List Button

□ Double click for default command

□ Save and Restore Instance Lists

☑ Allow multiple selections

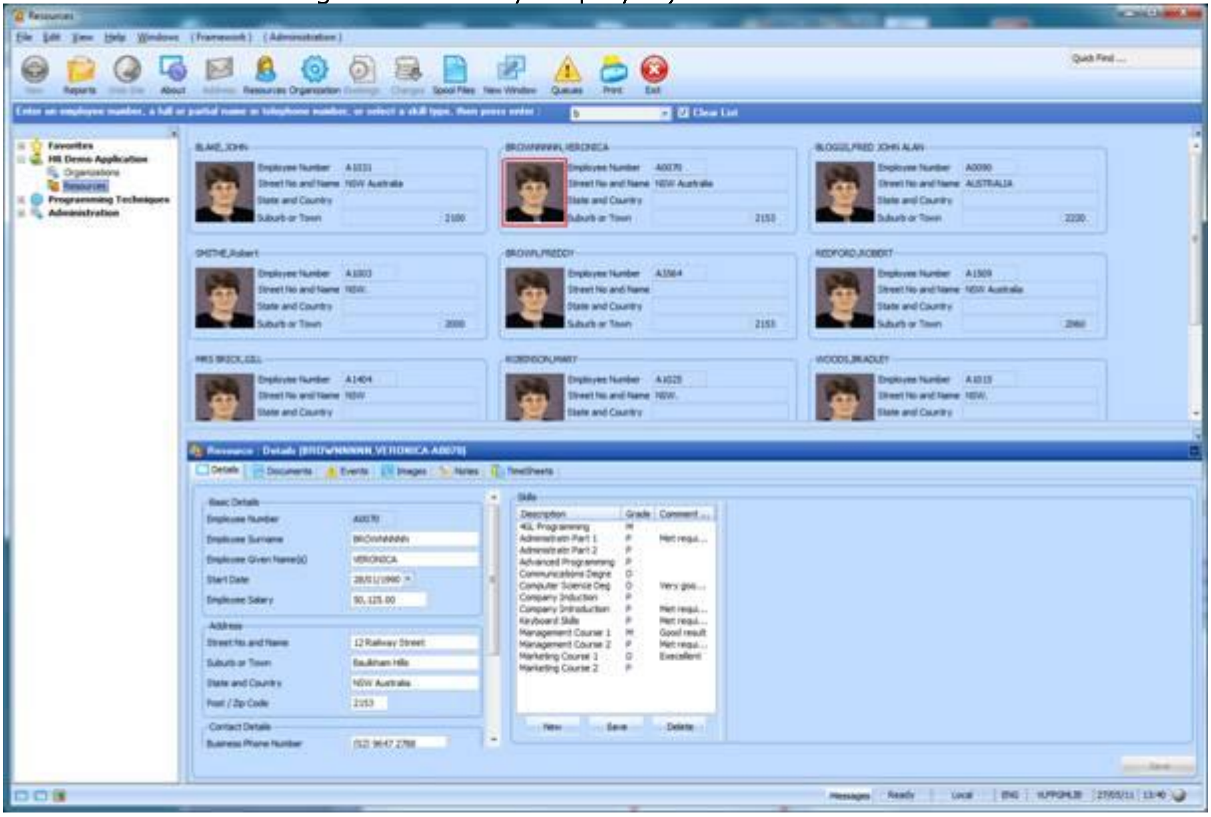
☑ Allow Instance List to be sent to MS-Excel   File Prefix to be used for MS-Excel: DEM_ORG_SEC_EMP_Spread (ENG)

Instance List Tool Bar Location	Left
Instance List Tool Bar Text Location	<None>
Instance List Tool Bar Height or Width	24
Snap in Instance List Browser	DEMPART02

Save your changes and restart the VLF application.

Execute the shipped demonstration

Display the Resources business object and search for employees.
The instance list should now be visualized something like this (noting that the example has used the same image file for every employee):



Enter an employee number, a full or partial name or telephone number, or select a skill type, then press enter : b

BLAKE, JOHN: Employee Number A1031; Street No and Name NSW Australia; Suburb or Town 2100

BROWNNNNN, VERONICA: Employee Number A0070; Street No and Name NSW Australia; Suburb or Town 2153

BLOGGS, FRED JOHN ALAN: Employee Number A0090; Street No and Name AUSTRALIA; Suburb or Town 2230

SMITHE, Robert: Employee Number A1003; Street No and Name NSW; Suburb or Town 2000

BROWN, FREDDY: Employee Number A1564; Suburb or Town 2153

REDFORD, ROBERT: Employee Number A1509; Street No and Name NSW Australia; Suburb or Town 2960

MRS BRICK, GILL: Employee Number A1404; Street No and Name NSW

ROBINSON, MARY: Employee Number A1025; Street No and Name NSW.

WOODS, BRADLEY: Employee Number A1015; Street No and Name NSW.

Resources : Details (BROWNNNNN, VERONICA A0070)

Basic Details

Employee Number	A0070
Employee Surname	BROWNNNNN
Employee Given Name(s)	VERONICA
Start Date	28/01/1990
Employee Salary	50,125.00

Address

Street No and Name	12 Railway Street
Suburb or Town	Baulkham Hills
State and Country	NSW Australia
Post / Zip Code	2153

Contact Details

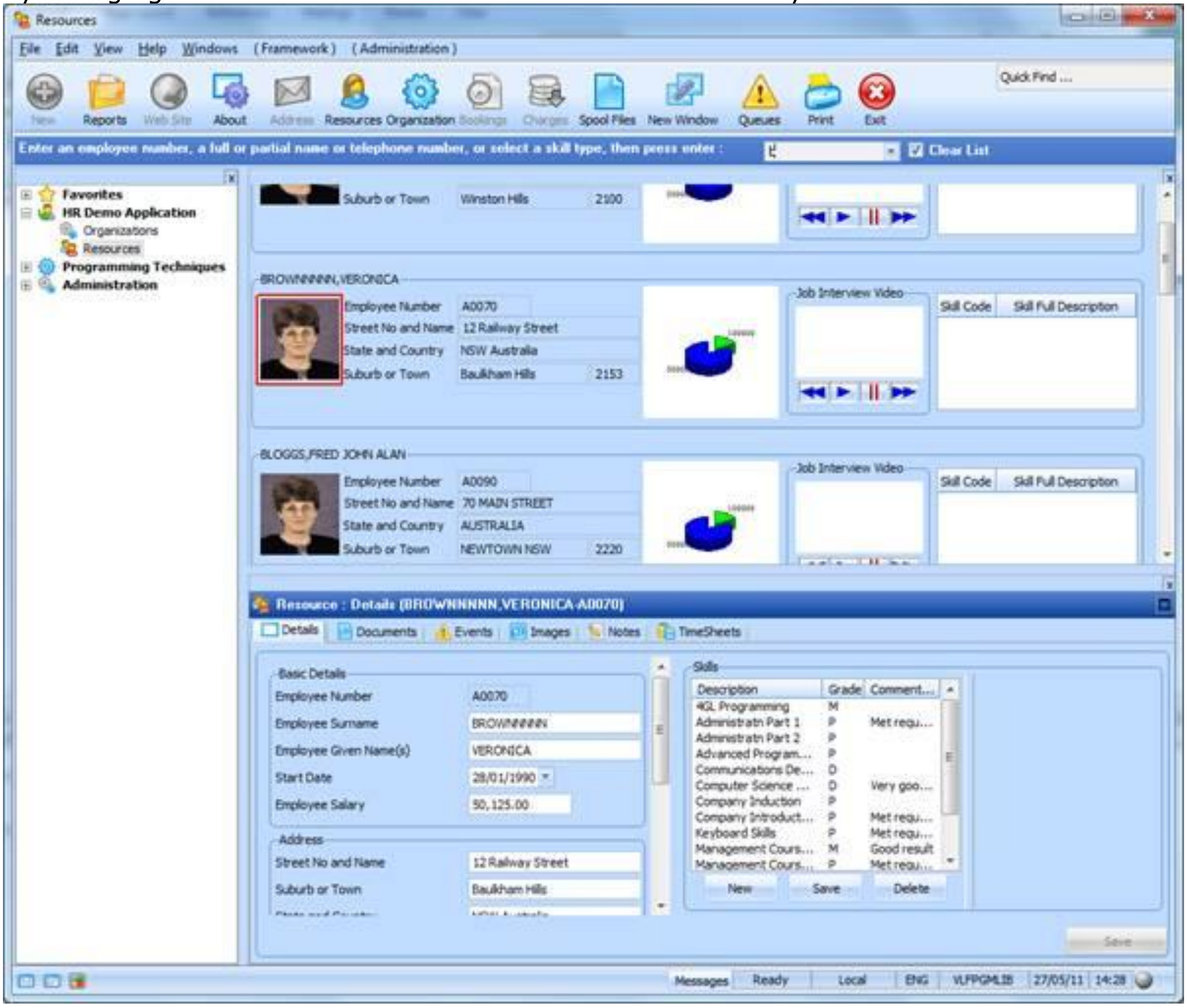
Business Phone Number	(02) 9647 2788

Skills

Description	Grade	Comment ...
4GL Programming	M	
Administratn Part 1	P	Met requi...
Administratn Part 2	P	
Advanced Programming	P	
Communications Degree	O	
Computer Science Deg	O	Very goo...
Company Induction	P	
Company Introduction	P	Met requi...
Keyboard Skills	P	Met requi...
Management Course 1	M	Good result
Management Course 2	P	Met requi...
Marketing Course 1	O	Excellent
Marketing Course 2	P	

The list is arranged by a VL layout manager and will automatically rearrange to fit the available space.

You can then start adding more and more things to the instance list "visualization" simply by changing the code in the DEMOPART01 instance list entry visualizer …..



Suburb or Town Winston Hills 2100

BROWNNNNN,VERONICA

Employee Number A0070
Street No and Name 12 Railway Street
State and Country NSW Australia
Suburb or Town Baulkham Hills 2153

BLOGGS,FRED JOHN ALAN

Employee Number A0090
Street No and Name 70 MAIN STREET
State and Country AUSTRALIA
Suburb or Town NEWTOWN NSW 2220

Resource : Details (BROWNNNNN,VERONICA-A0070)

Basic Details

Employee Number A0070
Employee Surname BROWNNNNN
Employee Given Name(s) VERONICA
Start Date 28/01/1990
Employee Salary 50,125.00

Address

Street No and Name 12 Railway Street
Suburb or Town Baulkham Hills

Skills

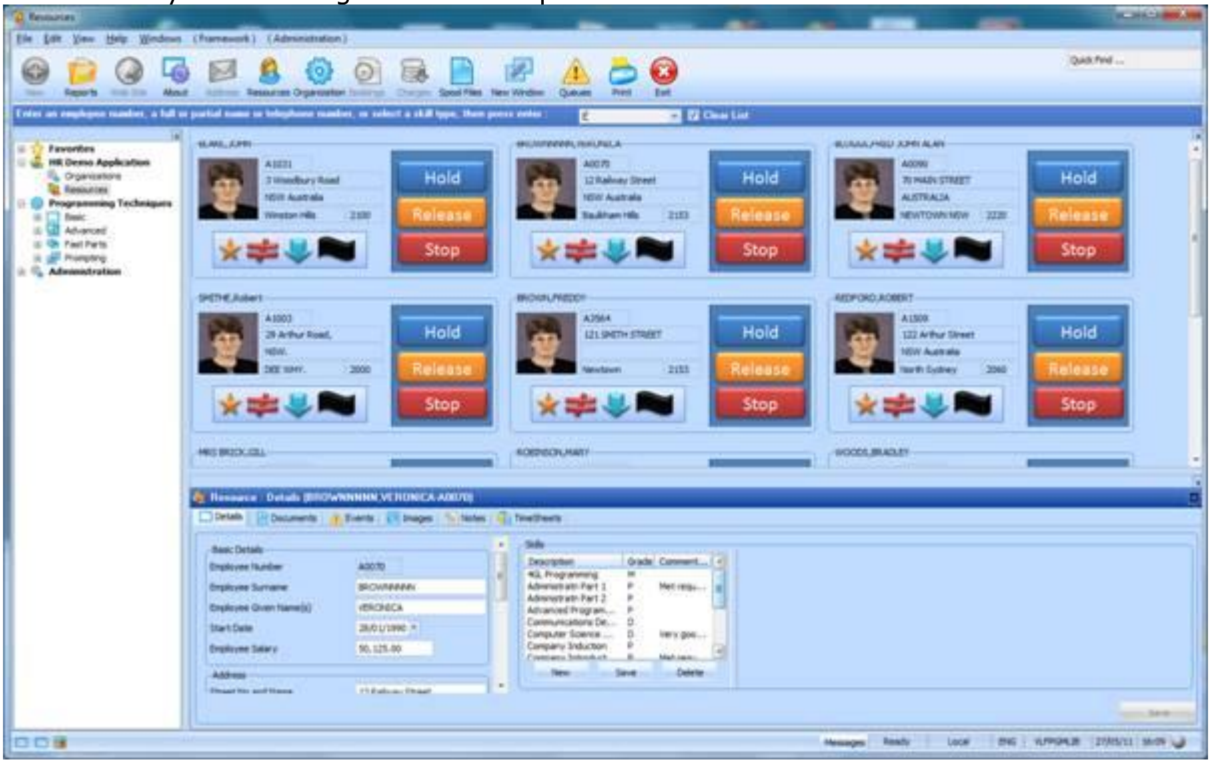
Description	Grade	Comment...
4GL Programming	M	
Administratn Part 1	P	Met requ...
Administratn Part 2	P	
Advanced Program...	P	
Communications De...	D	
Computer Science ...	D	Very goo...
Company Induction	P	
Company Introduct...	P	Met requ...
Keyboard Skills	P	Met requ...
Management Cours...	M	Good result
Management Cours...	P	Met requ...

Imagine that here each instance list entry visualization has a pie chart of the employees contribution to the group profit this month + a weekly progress review video + a grid of their skills.

The key concept is that the VLF's "instance list" is a boring list of key and identification numbers and strings that represent a single instance of a "business object" - such as an employee. There may be more interesting ways to visualize this business object that just a tree or list of strings and numbers.

Taking the preceding custom instance list browser further you can do a lot of visually impressive (if not always that tasteful) things with simple instance list visualizations changes.

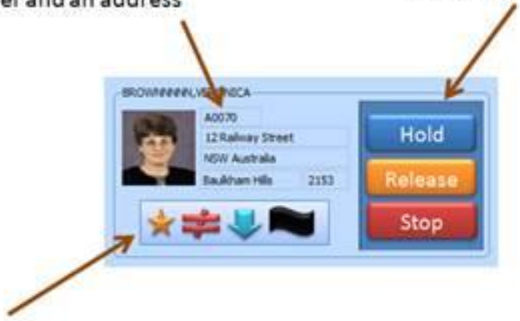
This is a very small change to the example:



Here

No labels – *everyone* knows it's an employee number and an address

Big and bold buttons for commonly used actions (to invoke command handlers)



Iconic/symbolic representation of state (eg: awarded employee, not a citizen, lives in the south and was once a pirate)

the button and state images are very easily assembled using tools as simple as MS-PowerPoint and then enrolled into the VL repository.

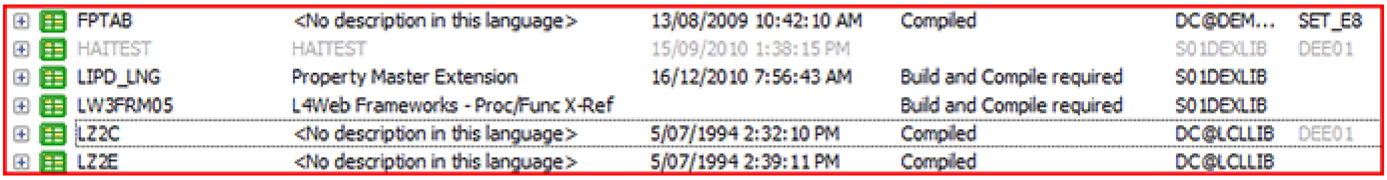
Why are my repository objects greyed out?

In V12 in a slave server environment, you will notice that there may be some objects in the Repository which are Greyed out. This is actually a new feature of V12.

In summary, the general rule is that **Gray = Master Black = local**.

In total the following three "STATUS" of the objects may be represented:

1. If an object is all black in the IDE (including task), this means that the object is stored locally and is available for editing.
2. If an object is all black but task greyed out means that the object has been checked out for read-only and they greyed out task is what its locked to on the server.
3. If an object is all grey means the object doesn't exist locally but is on the server only (and the task that it is locked to on the master is shown).



FPTAB	<No description in this language>	13/08/2009 10:42:10 AM	Compiled	DC@DEM...	SET_E8
HAITEST	HAITEST	15/09/2010 1:38:15 PM		S01DEXLIB	DEE01
LIPD_LNG	Property Master Extension	16/12/2010 7:56:43 AM	Build and Compile required	S01DEXLIB	
LW3FRM05	L4Web Frameworks - Proc/Func X-Ref		Build and Compile required	S01DEXLIB	
LZ2C	<No description in this language>	5/07/1994 2:32:10 PM	Compiled	DC@LCLLIB	DEE01
LZ2E	<No description in this language>	5/07/1994 2:39:11 PM	Compiled	DC@LCLLIB	

Taking the image above:

- File FPTAB is available locally and checked out for update. This should be available for editing.
- File HAITEST only exists on the master.
- File LZ2C has been previously checked out for read-only. Double clicking on this object would generate a not authorized message. DEE01 is the task that the object is locked to on the master system.

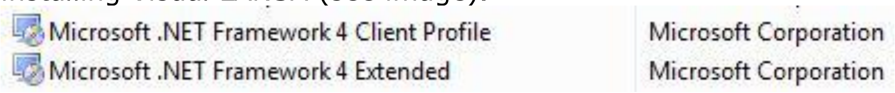
The main consideration is that an object in the IDE that is black except for the task has been checked out for read-only and the greyed out task is what its locked to on the master. This hold true even if you log in to the IDE with the task that the object is locked to on the master, its still in a read-only checkout state. If you then checkout the object for update, the task will now be black and the object is available for editing.

Visual LANSA - V12 SP1 install error

SQL Server 2008 SP2 is shipped on the V12 SP1 DVD. You may experience the following error when installing Visual LANSA where the install attempts to install or upgrade SQL Server 2008 to SP2.

Install ended in error.
Error <-3> creating database. Error message: SQL Server Network Interfaces: Error Locating Server/Instance Specified[xFFFFFFFF]. Microsoft SQL Server Native Client 10.0

If you have Microsoft .Net Framework 4 installed, it must be the full version i.e. both .Net Framework 4 Client Profile and .Net Framework 4 Extended must be installed prior to installing Visual LANSA (see image).



Microsoft .NET Framework 4 Client Profile	Microsoft Corporation
Microsoft .NET Framework 4 Extended	Microsoft Corporation

If the full version is not installed then the Microsoft SQL Server 2008 Express with SP2 install will fail. Refer to the Microsoft Support site article for full details: http://support.microsoft.com/kb/2500042

Note: The error will also occur if you attempt to install or upgrade to SQL Server 2008 SP2 manually (not as part of the Visual LANSA install).

Technique to prevent long running process windows from being "ghosted"

Starting with Windows Vista, Microsoft has included a feature called 'Window Ghosting', which involves 'ghosting' or replacing unresponsive windows with white/greyed out replacements, giving you the option to end unresponsive tasks. In some cases however, this ghosting is unwanted. An example might be a long running LANSA form that is generating a report, or a form that is waiting on the results of a submitted process.

Other symptoms of this ghosting might be an application that is unexpectedly minimized, or simply sent to the background. If this is happening in your application, it might be due to ghosting.

Prevention

This can be prevented in LANSA applications by the use of the following system command:
#Sys_Appln.DisableProcessGhosting

This will disable the process ghosting from being applied to your application. The command should be implemented in your startup form, but it only needs to be called once.

ZIPFileService does not support .gz file format

.gz files are not supported by the ZIPFileService.

LANSA Integrator uses the Java supplied Zip classes which are based on the InfoZip format: http://www.info-zip.org

A .gz file uses a different compressed file format. The extension .gz stands for GNU Zip.

GNU Zip or .gz uses a free GNU software implementation of a compression algorithm.

Hence the compression algorithms used by GNU Zip .gz are different from those used in Zip files and so you need different software to open .gz files.
See http://www.tech-pro.net/howto-open-gz-file.html

Unable to restore V12 objects to QTEMP on V5R3 or V5R4

When upgrading LANSA for iSeries to V12, you may be unable to restore V12 objects to QTEMP on V5R3 or V5R4. This is a known situation whereby PTFs are required for any objects saved on IBM i 6.1 before they can be restored on previous IBM i versions. Refer to Unable to restore V11 SP5 objects to QTEMP on V5R3 or V5R4 function/WAM/component on IBM i

From V11 SP5, IBM i 6.1 is fully supported. LANSA 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)
```

Solution

Ensure these PTFs are applied: (check IBM websites for other OS versions)

V5R3M0
MF41354, delayed PTF to be applied at IPL
SI27297, apply immediate or delayed

V5R4M0
MF40520, delayed PTF to be applied at IPL
SI27294, apply immediate or delayed

V5R2 all versions
This is no longer an IBM supported Version. You are required to upgrade your OS version before applying V11 SP5.

Optimization of JAR files is no longer required

The following note will be shipped in the next edition of the LANSA Integrator Guide:

It is no longer necessary to optimize JAR files. The new IBM Technology for Java JDKs do not use these optimized programs and the Classic JDKs have improved to such a point that these optimized programs do not offer much benefit.

Hence the option - Optimize Java Service Manager (OPTJSM) - is no longer necessary. It has, however, been maintained where customer may be using earlier versions of Java.

Unexpected DE2 behaviour after installing aXes 2.1 if a particular IBM PTF is not installed

There have been reports of unexpected behaviour when installing or using DE2 on V5R4M0.

For example:

1.
The following message may appear after installing aXes 2.1.

```
 Message ID  . . . . :   CPI32A6      Severity . . . . . . . . :   60
 Message Type . . :    Information
 Sent Date  . . . . . . . :   11/07/29      Sent Time . . . . . . : 05:08:34
 Message . . :    Request 1 ,  Type 2  Cross-Reference notification  8151
```

2.
DE2 screen freezes when attempting to display file columns and eventually an error XHR status 12031 is shown

Both these issues can be resolved by applying IBM PTF SI43347 for V5R4.

Refer to the following PTF Cover letter:
http://www-912.ibm.com/a_dir/as4ptf.NSF/b3cb9d42f672b70f86256739004afa0f/72793a5f460a5a1a8625787e005da981?OpenDocument

If you encounter the issues above or any other variations of unexpected behaviour using DE2 or aXes 2.1, the first troubleshooting step should be to get to the latest PTF level, and in particular ensure that PTF SI43347 has been applied.

This information will also be posted as a technical article on the aXes web site
http://www.axeslive.com/support/articles.htm

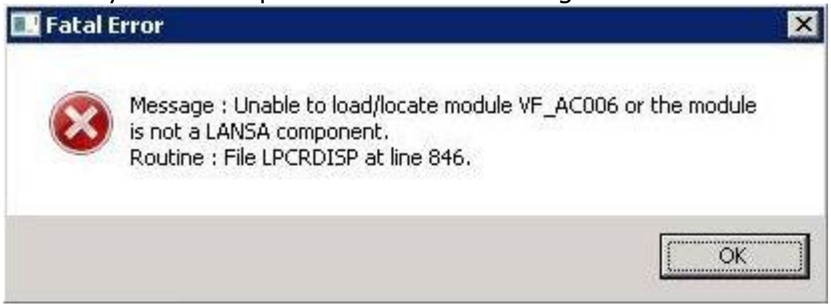
Missing C++ Runtime file MSVCR71.dll causes installation and runtime failures

Preface

Several issues that have arisen from missing C++ runtime file MVCR71.dll have been reported by LANSA customers. There are 2 existing tips regarding specific variations:
Unable to Load/locate module VF_AC006 error after deploying a VLF application in V12
Deploying an application compiled with Visual Studio .NET C++
This technical note aims to provide a central repository for information relating to missing C++ runtime DLLs and their effects on LANSA applications.

Description

On some PCs, a specific Microsoft C++ runtime DLL required by LANSA does not exist. This causes the LANSA runtime to generate an error reporting that the LANSA module/OAM can not be found. This is generated even though the object exists in the partition execute directory. An example of the error message is:



Fatal Error

Message : Unable to load/locate module VF_AC006 or the module is not a LANSA component.
Routine : File LPCRDISP at line 846.

During a deployment tool package installation, it can also cause the load of file data to fail. In this case the message "Version of [File name] cannot be determined" will appear (along with other errors) in the installation log dpinstal.log.

```
C Definition of File PSLMST has changed, rebuild required (16/03/11)
(12:53:58)
C Rebuild of File PSLMST successful (16/03/11) (12:53:58)
C Copy of file pslmst.CTD was successful (16/03/11) (12:53:59)
C Copy of file pslmst.DLL was successful (16/03/11) (12:53:59)
F Load of data for File PSLMST failed (16/03/11) (12:53:59)
 "C:\LANSA\X_WIN95\X_LANSA\X_DEM\SOURCE\PSLMST.CTD" - Invalid Path Specifie
 [Microsoft][SQL Server Native Client 10.0][SQL Server]The server principal
'LX_DTA' already exists when ex
 [Microsoft][SQL Server Native Client 10.0][SQL Server]User, group, or role
'LX_DTA' already exists in the cu
 Version of "PSLMST" cannot be determined.
```

Cause

As mentioned above, the root cause of this error is a missing C++ runtime file. In most cases it is the MS Visual Studio .Net 2003 C++ runtime object MSVCR71.DLL (objects created in Visual LANSA version 11 require this runtime DLL for execution). Previously, MSVCR71.DLL was shipped as standard in the System32 directory of Windows, so it was not included in the LANSA deployment package. However it is no longer shipped, and Microsoft have left it up to the user to download/install this themselves.

Objects created in Visual LANSA version 12 use the MS Visual Studio .Net 2008 runtime (MSVCR90.DLL), however the deployment tool now includes an option to ship the 2008 runtime with deployment packages (it is selected by default).

Diagnosis/Solution

If any of the symptoms above occur on a LANSA runtime environment then there is a chance that the required runtime DLL is missing. If any part of the application was built on Visual LANSA V11 with the standard compiler, then MSVCR71.DLL is required to be on the target PC, in a location in that is accessable by LANSA (i.e. c:\Windows\System32 or any path that is included in the %PATH% environment variable).

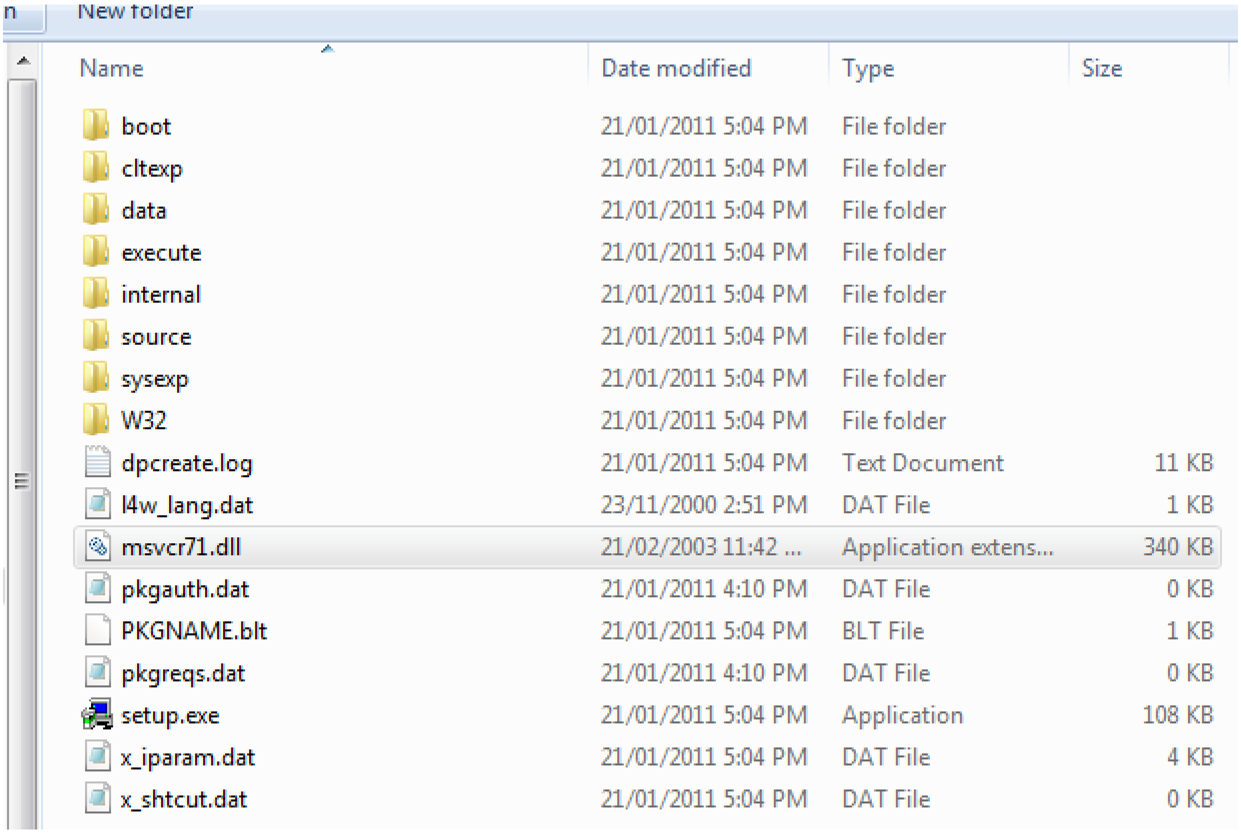
The following page provides a summary on which msvcrnn.dll is required for a given Visual Studio version:
http://support.microsoft.com/kb/326922

One recommended solution is to download or copy the MSVCR71.DLL from a working computer, and place it in the system execute directory (x_win95\x_lansa\execute) of the target PC. This allows LANSA programs to use the DLL without affecting other Windows applications.

For packages deployed using the Deployment Tool, the DLL can be included as a Non-LANSA object in your Deployment Tool Package (target location System Execute directory). However if the file data load error is being encountered during install, then this will not work, as the Non-LANSA Object file copy will occur after the error. In that case it is possible to manually include the DLL in your package and copy it as a before-installation command:

```
ALL   COPY   "%pkgd%[Your Application Name]\[Your Package Name]\msvcr71.dll"
"%dirl%execute\msvcr71.dll"
```

Once the package is built, just manually copy the msvcr71.dll in the main package directory (where the setup.exe is):



Name	Date modified	Type	Size
boot	21/01/2011 5:04 PM	File folder	
cltexp	21/01/2011 5:04 PM	File folder	
data	21/01/2011 5:04 PM	File folder	
execute	21/01/2011 5:04 PM	File folder	
internal	21/01/2011 5:04 PM	File folder	
source	21/01/2011 5:04 PM	File folder	
sysexp	21/01/2011 5:04 PM	File folder	
W32	21/01/2011 5:04 PM	File folder	
dpcreate.log	21/01/2011 5:04 PM	Text Document	11 KB
l4w_lang.dat	23/11/2000 2:51 PM	DAT File	1 KB
msvcr71.dll	21/02/2003 11:42 ...	Application extens...	340 KB
pkgauth.dat	21/01/2011 4:10 PM	DAT File	0 KB
PKGNAME.blt	21/01/2011 5:04 PM	BLT File	1 KB
pkgreqs.dat	21/01/2011 4:10 PM	DAT File	0 KB
setup.exe	21/01/2011 5:04 PM	Application	108 KB
x_iparam.dat	21/01/2011 5:04 PM	DAT File	4 KB
x_shtcut.dat	21/01/2011 5:04 PM	DAT File	0 KB

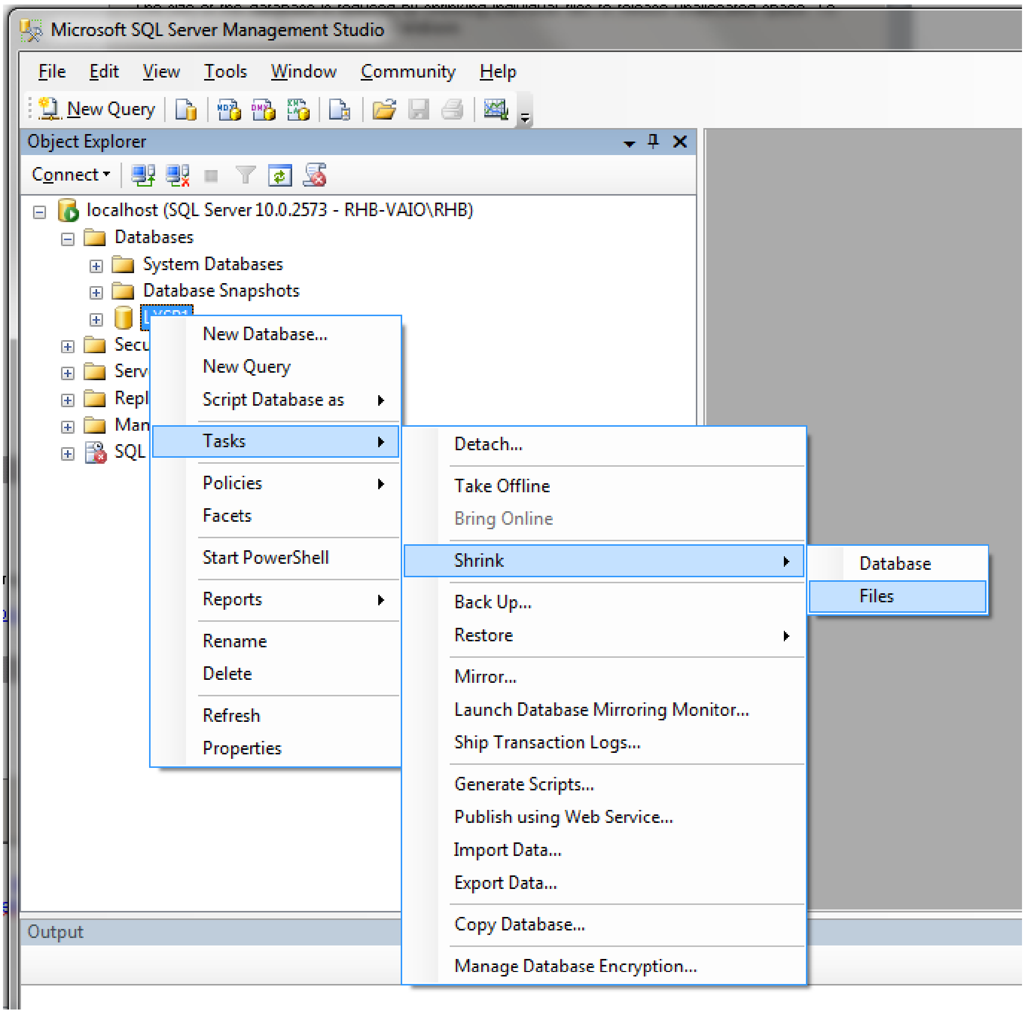
Reorganize log file MS SQL Server

When you use LANSA V12 on a development system with Microsoft SQL Server and you change something as a developer, this change will be collected in a log file. In fact, every change will be part of this log file. When many developers are active, this log file can become very big.

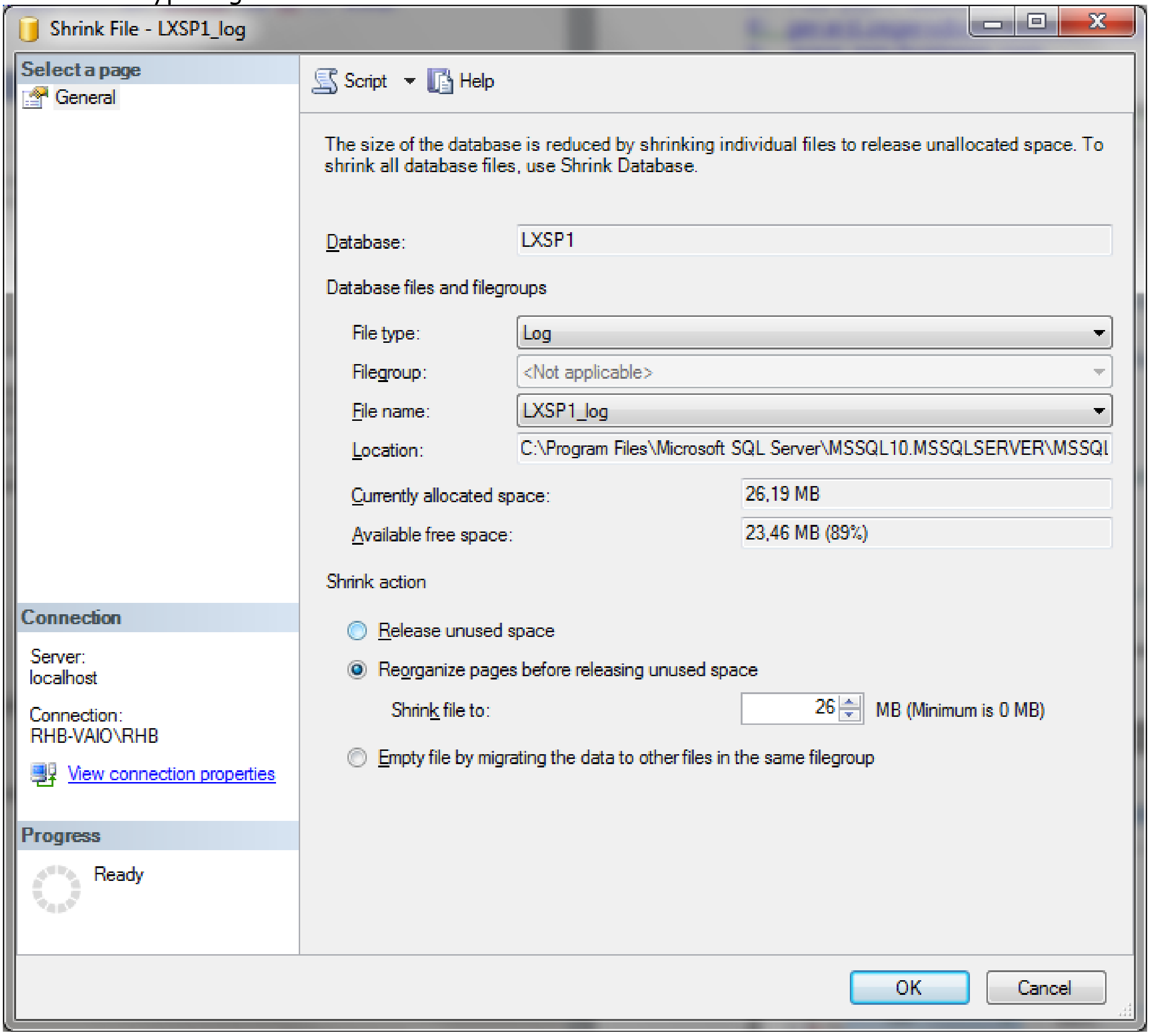
When it becomes too big, it needs to be reorganized.

This can be done as follows:

1. Open in Microsoft SQL Server Management Studio the database in the tree view.
2. Use your right mouse button to open Tasks, Shrink, Files:



Select File type log:



Shrink File - LXSP1_log

The size of the database is reduced by shrinking individual files to release unallocated space. To shrink all database files, use Shrink Database.

Database: LXSP1

Database files and filegroups

File type: Log

Filegroup: <Not applicable>

File name: LXSP1_log

Location: C:\Program Files\Microsoft SQL Server\MSSQL10.MSSQLSERVER\MSSQL

Currently allocated space: 26,19 MB

Available free space: 23,46 MB (89%)

Shrink action

○ Release unused space

● Reorganize pages before releasing unused space

Shrink file to: 26 MB (Minimum is 0 MB)

○ Empty file by migrating the data to other files in the same filegroup

Connection

Server:
localhost

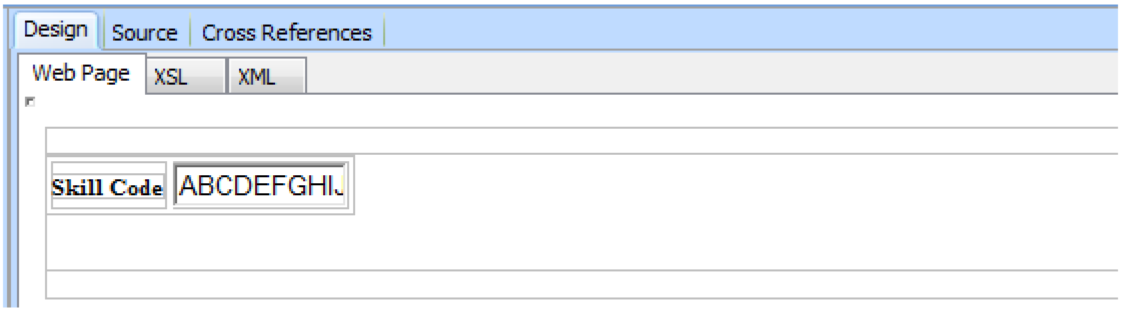
Connection:
RHB-VAIO\RHB

You can use option “Reorganize pages before releasing unused space”. With this option you can shrink the log file to a given numbers of Mb.

How to create a WebRoutine with no layout in V12 SP1

With the new WAM layouts in SP1, the method for displaying a WebRoutine with no layout/css has changed. Previously it was possible to select the layout weblet part of the design view, and then changing the property no_layout to true (and also show_title=false/show_messages=false).

In V12 SP1, to create a plain output you can now simply drag the supplied "std_blank_layout" weblet onto the webroutine. This replaces whatever layout it had previously and replicates the V12 no_layout=true behaviour.



*Cleaning up the LOB directory using the special process *LOBCLNUP*

When using BLOBs and CLOBs, temporary files are created in the LOB directory on the IBM i IFS. It's usage over time can build up to a large size. The LOB directory is typically located /LANSA_dc@pgmlib/x_lansa/lobuser/[username]/

To clear this temporary directory and reclaim space, execute the special process called *LOBCLNUP.

This can be called from the command line:
LANSA REQUEST(X_RUN) X_RUNADPRM('PROC=*LOBCLNUP')

The process needs to be run by a user with sufficient authority to remove files that may have been created by other users.

Consider using this special process as part of scheduled system maintenance.

PDFDocumentService error

Embedding certain images in PDFDocumentService gives error: stack trace: java.lang.UnsatisfiedLinkError: sun/awt/color/CMM

When using the PDFDocumentService to include images in a document being generated, using certain types of images will cause JSM to fail with sun/awt related errors.

For instance:
stack trace: java.lang.UnsatisfiedLinkError: sun/awt/color/CMM

Solution

Make sure that the JAVA awt feature is turned on.

Locate and open the SystemDefault.properties file in the JSM instance system directory
Ensure that the following line exists in the file.

Set java.awt.headless=true as per

```
#AllowOptions
#!<studio-project id="20000000-000000" name="lansa">
#
# Java system default properties
#
java.version=1.5
java.awt.headless=true
```

*Configuring JSM *PUBLIC access to files and directories*

On the iSeries, it is possible to configure the *PUBLIC authority of files and directories created by JSM.

The following System properties are available:

```
  os400.dir.create.auth
  os400.file.create.auth
```

These properties specify authorities assigned to files and directories.

Specifying the properties without any values or with unsupported values results in a public authority of *NONE.

You can specify os400.file.create.auth=RWX or os400.dir.create.auth=RWX, where R=read, W=write, and X=execute.

Edit the JSM SystemDefault.properties and add these properties with invalid values:

```
  /jsm/instance/system/SystemDefault.properties

     os400.dir.create.auth=---
     os400.file.create.auth=---
```

When a trace directory, a trace file, a file dragged/dropped from Studio and created by a service, the *PUBLIC authority will be *EXCLUDE.

```
/jsm/instance/trace

   Object . . . . . . . . . . . . :   /jsm/instance/trace/976026

     *PUBLIC     *EXCLUDE
     QOTHPRDOWN  *RWX         X     X     X     X
```

To remove existing *PUBLIC authority from all files and directories you can use the following OS/400 command:

```
CHGAUT OBJ('/jsm/instance') USER(*PUBLIC) DTAAUT(*EXCLUDE)
OBJAUT(*NONE) SUBTREE(*ALL)
```