http://www.lansa.com

LANSA

Newsletter
September 2007

System i Expo - Netherlands

Date: September 26 - 27, 2007
Location: Center Parcs De Eemhof in Flevoland
LANSA will be presenting at the event.

System iExpo is the biggest and most important, independent System i event in the Benelux in 2007. An expo that will be supported by IBM and various IBM Business Partners and ISV's. If you are a supplier of System i related products, than you don't want to miss this event!

Visit the System i Expo Web site (http://www.systemimagazine.nl/expo/) for further information.

Both on 26 and 27 September 2007, LANSA will organize workshops and demo's in the Barentz-zaal and in the lobby of the Business Center:

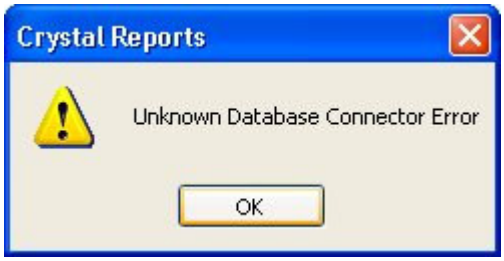
LANSA (26 September)	Snel en eenvoudig (Windows, Web, 5250)-applicaties ontwikkelen met Visual LANSA. http://www.systemimagazine.nl/expo/pagina.tpl?pagina=11891713961005230
LANSA (27 September)	Snel en eenvoudig Web Services ontwikkelen met LANSA Integrator. http://www.systemimagazine.nl/expo/pagina.tpl?pagina=11891713961005230

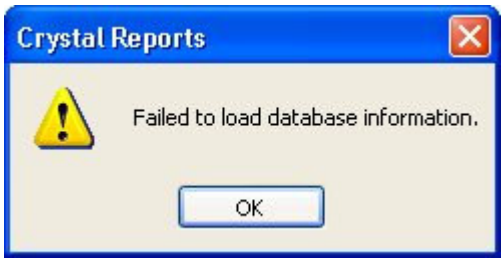
In This Issue

Crystal Reports XI cannot open DBF file created in LANSA Client 11.3

LANSA Client can output to file type dbf.
A DBF file created with LANSA Client V11.3 will not open when creating a report with Crystal Reports XI.

The error message generated in Crystal Reports XI is:





Cause

This error is generated due to a change in the way native database drivers are handled in Crystal Reports XI. DBF support is not installed by default by Crystal Reports XI. The following Crystal Reports article discusses this further:
http://support.businessobjects.com/library/kbase/articles/c2018235.asp

Resolution

The resolution is to open the database file as an xBase data source. However, xBase is not on the default list of database drivers and must be added on demand. In order to add it perform the following steps:

1. In Crystal Reports XI, under New Reports/Standard Reports Wizard, go to Create new connection/More data sources/xBase. You will then be prompted to insert the Crystal Reports installation CD.
2. Insert the LANSA Client 11.3 CD2.
3. After the setup has completed you will be able to open DBF files as normal.

xBase will now appear under Create new connection instead of being in More data sources.

In addition to the above, you can also use Create new connection/Database files to open DBF files (as well as Create new connection/xBase).

Correct use of the PositionOf and PositionIn Intrinsic functions

PositionOf and PositionIn are 2 useful intrinsic functions introduced in V11.0, which find the first occurance of one string within another string. They can be used to easily parse wordstrings or do simple string searching for validation. However when used with Alphanumeric (RDML) fields care must be taken with regards to 'trailing spaces'

Details

Firstly, it should be noted that PositionOf and PositionIn are interchangeable and either one can be used depending on the developer's preference. The difference is in the way that the code reads in the IDE (note there are optional arguments with these functions that are not specified below):

```
#FoundPos := #SearchString.PositionIn(#InputString)
vs.
#FoundPos := #InputString.PositionOf(#SearchString)
```

Where #InputString is the string to be searched, #SearchString is the string to be searched for, and #FoundPos is the location at which #SearchString is found in #InputString

Special Consideration when using Alpha (RDML) field types

It should be noted that one major distinction between Alpha and String field types, is the significance of trailing spaces. In Alpha fields, trailing spaces are insignificant, and are not used in processing. On the other hand, trailing spaces are significant in String fields

This is important if you are using PositionOf/PositionIn to search for " " blanks in a parsing algorithm. Setting an alpha field #SearchString to *blanks or " " is the same as setting it to nothing (""). Using this with PositionOf/PositionIn will not give the correct results.

The solution is to ensure #SearchString is defined as a String field. Alternatively you can manually code the search string as follows

```
#FoundPos := (" ").PositionIn( #InputString )
```

Usage examples

Parse an input string

```
#w_string := 'Hello World Goodbye'
#w_blank := ' '

#w_pos := #w_string.PositionOf( #w_blank )
dowhile ( #w_pos > 0 )
  * Loop until no more spaces are found
  #w_token := #w_string.LeftMost( (#w_pos - 1) )
  Use Ov_Message_Box #w_token

  #w_string := #w_string.Substring( (#w_pos + 1) )
  #w_pos := #w_string.PositionOf( #w_blank )
endwhile
* Get last word
#w_token := #w_string
Use Ov_Message_Box #w_string2
```

Validate an input string

```
* Crude Email Validity Checking
If Cond(#Email.PositionOf('@') = 0)
  Use Builtin(Ov_Message_Box) With_Args('Invalid Email Address')
Endif
```

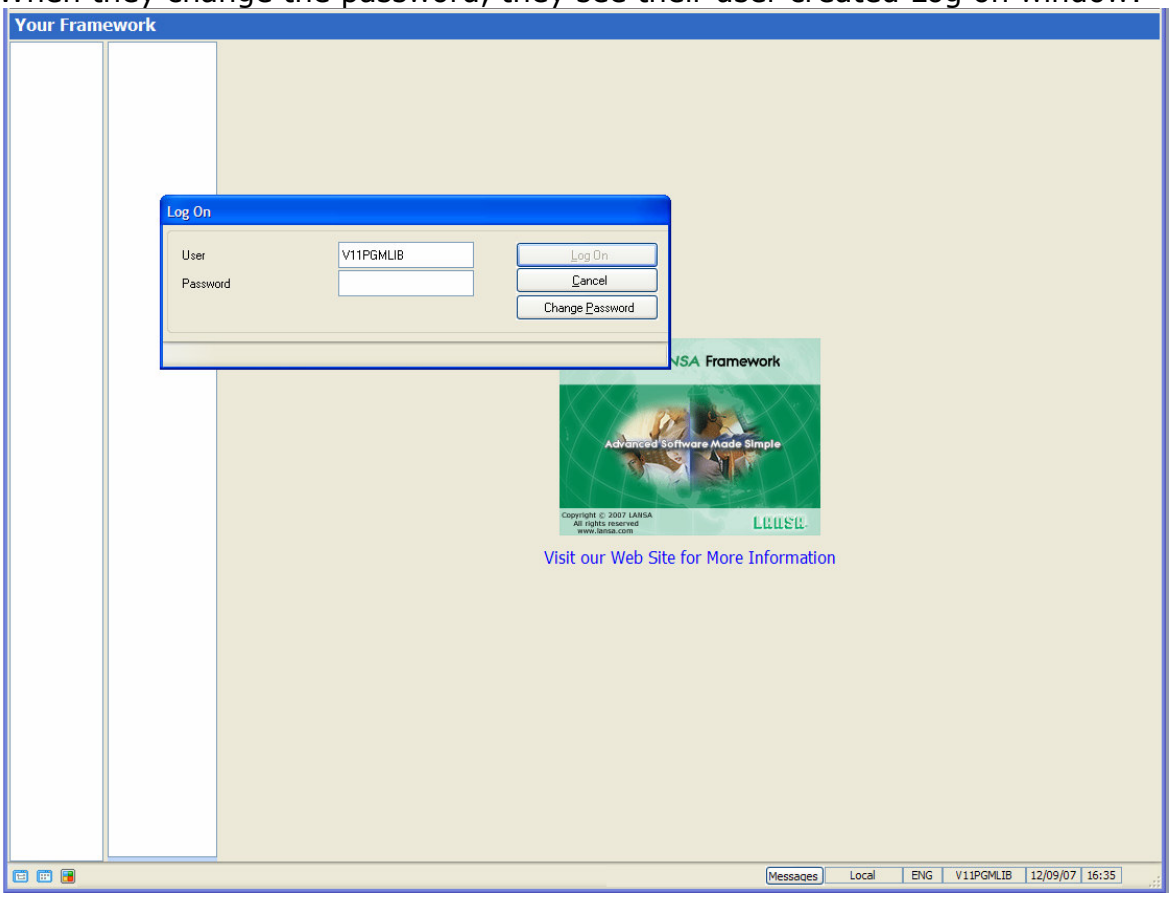
Note: Both code examples can be easily adapted to use PositionIn as noted above.

LOGON in Visual LANSA Framework

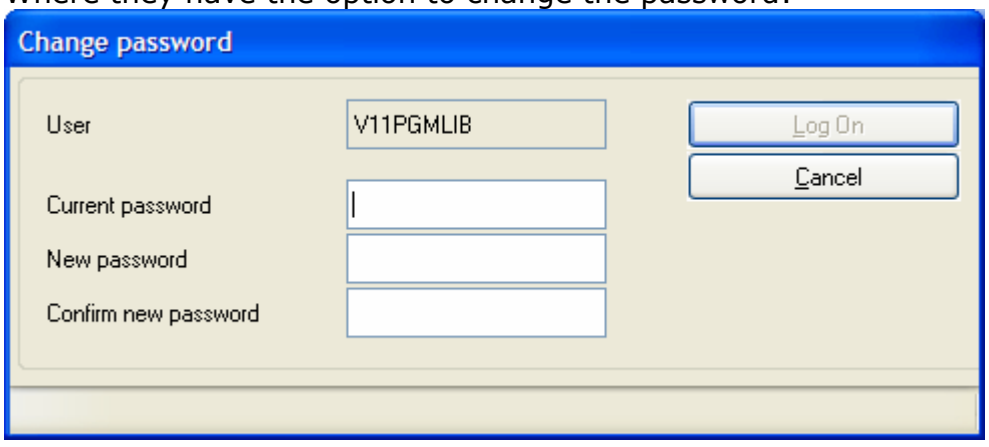
(Thanks to Tim De Porte from Fortis Commercial Finance – Belgium)

Fortis Commercial Finance in Belgium are developing VLF applications and the created their own mechanism to logon in the VLF and to change the password of a user.

When they change the password, they see their user created Log on window:



Where they have the option to change the password:



To make this possible, they created these programs (***all these programs can be downloaded when this newsletter is available on the LANSA web site***):

LANSA Functions:
CHGPWD
MAXSIGN
RTVUSRI

Visual LANSA Reusable Part:
UF_SYSFCF

Visual LANSA Forms:
CHGPWDWIN
DISPUSRINF
LOGONWIN

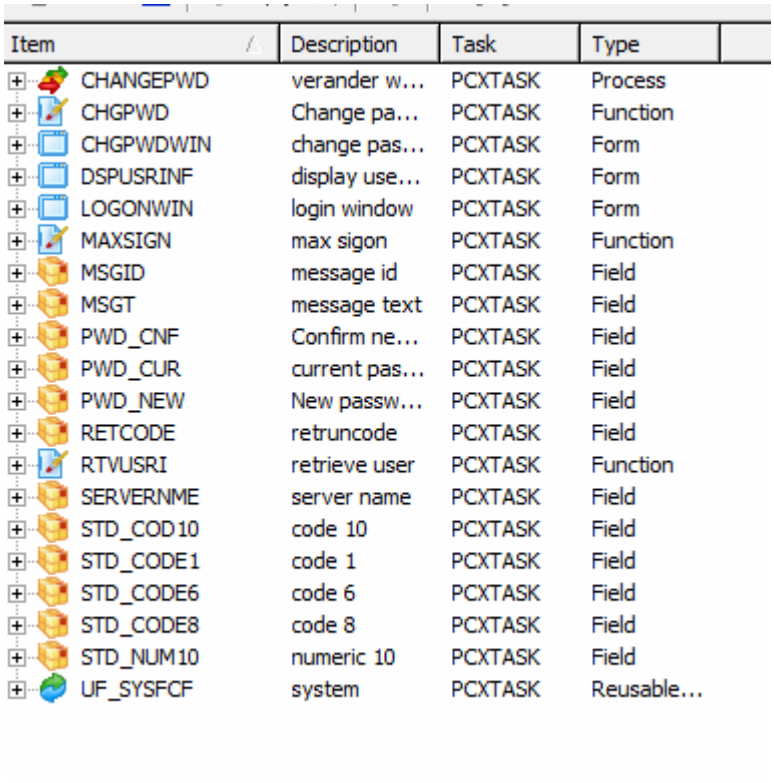
CL-programs:
CHGPWDC
RTVMAXSIGN
RTVUSRINFC

RPG program:
RTVUSRINFR

To make use of this tool, do the following:

1. Create all functions, forms and reusable parts in VL.
2. Create the attached CL and RPG programs.
3. Create about 10 new fields in the LANSA Repository (you need these fields in the forms and rup).
4. On the iSeries, create a new super user (u:SUPERUSER, pw:SUPERUSR).
5. In the VLF settings, select the VLF authority (connect to a remote server).
6. Create a correct Server definition in the VLF.
7. Set the User Imbedded Interface Point in the VLF to UF_SYSFCF.

The next page shows a list of objects with their defaults.



Item	Description	Task	Type
CHANGEPWD	verander w...	PCXTASK	Process
CHGPWD	Change pa...	PCXTASK	Function
CHGPWDWIN	change pas...	PCXTASK	Form
DSPUSRINF	display use...	PCXTASK	Form
LOGONWIN	login window	PCXTASK	Form
MAXSIGN	max sigon	PCXTASK	Function
MSGID	message id	PCXTASK	Field
MSGT	message text	PCXTASK	Field
PWD_CNF	Confirm ne...	PCXTASK	Field
PWD_CUR	current pas...	PCXTASK	Field
PWD_NEW	New passw...	PCXTASK	Field
RETCODE	retruncode	PCXTASK	Field
RTVUSRI	retrieve user	PCXTASK	Function
SERVERNME	server name	PCXTASK	Field
STD_COD10	code 10	PCXTASK	Field
STD_CODE1	code 1	PCXTASK	Field
STD_CODE6	code 6	PCXTASK	Field
STD_CODE8	code 8	PCXTASK	Field
STD_NUM10	numeric 10	PCXTASK	Field
UF_SYSFCF	system	PCXTASK	Reusable...

Field details:

Name	***Type***	***Length***
MSGID	Alphanumeric	10
MSGT	Alphanumeric	132
PWD_CNF	Alphanumeric	10
PWD_CUR	Alphanumeric	10
PWD_NEW	Alphanumeric	10
RETCODE	Alphanumeric	2
SERVERNMW	Alphanumeric	10
STD_COD10	Alphanumeric	10
STD_CODE1	Alphanumeric	1
STD_CODE6	Alphanumeric	6
STD_CODE8	Alphanumeric	8
STD_NUM10	Packed	10,0

Catering for requirements of mixed case in JSM commands

When building JSM commands, there may be instances where it is a requirement (from the Server) to build the command (a command is built up of arguments and keywords) in mixed case, i.e. not default to uppercase.

For example, there are applications that require to have the Username and Password passed to be in mixed case format. That is, the validation is Case Sensitive.

If a JSM Command is built as such:
CHANGE FIELD(#JSMCMD) TO('CONNECT DRIVER(DB2) DATABASE(JSMJDBC) USER(UserName) PASSWORD(Password)')

Then it will default to uppercase.

Therefore in order for the command to not convert the JSM command to uppercase, then it is suggested that the command is **enclosed with triple single quote** as per

CHANGE FIELD(#JSMCMD) TO(**'''**CONNECT DRIVER(DB2) DATABASE(JSMJDBC) USER(UserName) PASSWORD(Password)**'''**)

This ensures that the command is executed as typed without conversion.

UpdateListEntryData in VLF

If you have been coding your own snap in instance list browsers

AND

are using the new UpdateListEntryData method

Prior to EPC793, people generally used instance list method AddtoList to update an existing list entry.
AddtoList can be used to add a new entry or to update an existing entry.

In the EPC793 a new method was released to immediately update an existing instance list entry.
It's called **UpdateListEntryData**. It's more efficient than using AddtoList for instance list updates.

If you have started using it, and have your own snap in instance list browsers:

- ⇨ Using AddtoList against an instance list will cause method **uAddListEntry** to be invoked in your snap in browser.
- ⇨ Using UpdateListEntryData against an instance list will cause method **UpdateListEntryData** to be invoked in your snap in browser.

If you use F2=Feature Help in your VL editor when editing your snap in browser you can explore these methods and their parameters.

Some considerations when coding a **UpdateListEntryData** method into your snap in browser are:

- ⇨ It's only ever called to update an existing instance list entry, and only because a filter or handler has used UpdateListEntryData.
- ⇨ It does ***not*** receive the AKey1->AKey5, NKey1->NKey5 key identification parameters (like uAddListEntry does).
- ⇨ It does receive the uInstanceIdentifier parameter that uniquely identifies which instance list entry is being updated.
- ⇨ Using uInstanceIdentifier to locate an existing instance list entry is generally more efficient then having to compare (potentially) 5 alpha keys and 5 numeric keys.

Now:

- ⇨ When your instance list browser's uAddListEntry is called to create an instance list entry, it is passed a uInstanceIdentifier value.
- ⇨ When your instance list browser's UpdateListEntryData is called to update an instance list entry, it is passed a uInstanceIdentifier.

so you need to be able to associate them.

When your uAddListEntry method creates the thing that 'visualizes' the instance list entry, maybe in a tree, a grid, whatever, you need to record (ie: cross reference) the uInstanceIdentifier associated with the thing. You might do this as hidden column in a tree or grid, or even store a direct reference to the tree item or grid item created in a keyed collection (which is by far the fastest way to do this).

In your uUpdateListEntry method you will need to then find the correct item that needs to be updated, by using the passed uInstanceIdentifier as the key.

AutoTab property of a field

You can set the AutoTab property of a field to True if you want the cursor to move to the next input field (next TabPosition) when an input field has been filled.

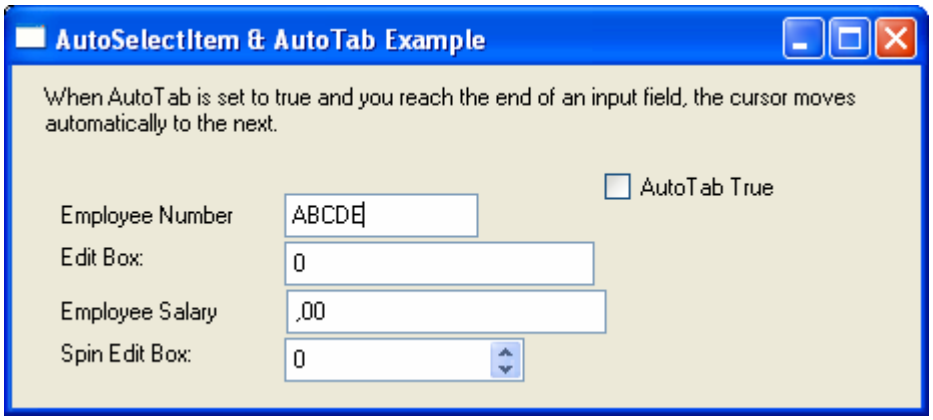
To see how AutoTab works, copy and paste the following code to a form, compile and execute it:

Source

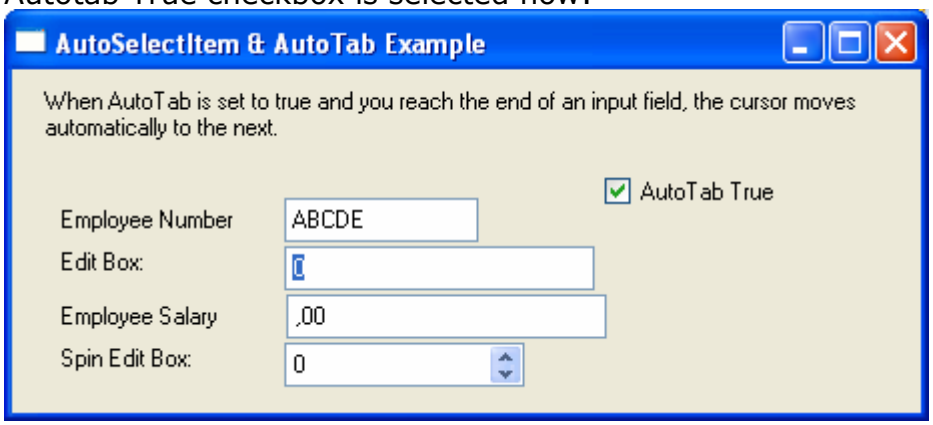
```
FUNCTION options(*DIRECT)

Begin_Com Role(*EXTENDS #PRIM_FORM) Caption('AutoSelectItem & AutoTab
Example') Clientheight(171) Clientwidth(454) Height(205) Left(336) Top(158)
Width(462)
Define_Com Class(#PRIM_CKBX) Name(#CKBX_AUTOTAB) Caption('AutoTab
True') Displayposition(5) Left(296) Parent(#COM_OWNER) Tabposition(6)
Top(48)
Define_Com Class(#EMPNO.Visual) Name(#EMPNO) Autoselect(False)
Caption('Employee Number') Displayposition(1) Height(22) Labeltype(Caption)
Left(24) Marginleft(112) Parent(#COM_OWNER) Tabposition(1) Top(64)
Usepicklist(False) Width(209)
Define_Com Class(#PRIM_EDIT) Name(#EDIT_1) Autoselect(False)
Displayposition(3) Height(22) Left(136) Maxlength(5) Parent(#COM_OWNER)
Tabposition(2) Top(88) Value('0') Width(155)
Define_Com Class(#SALARY.Visual) Name(#SALARY) Autoselect(False)
Displayposition(6) Height(22) Left(24) Marginleft(113) Parent(#COM_OWNER)
Tabposition(3) Top(112) Usepicklist(False) Width(273)
Define_Com Class(#PRIM_SPDT) Name(#SPDT_1) Displayposition(7) Height(22)
Left(136) Parent(#COM_OWNER) Tabposition(4) Top(136)
Define_Com Class(#PRIM_LABL) Name(#LABL_1) Caption('Edit Box:')
Displayposition(2) Height(19) Left(24) Parent(#COM_OWNER) Tabposition(7)
Tabstop(False) Top(88) Width(88)
Define_Com Class(#PRIM_LABL) Name(#LABL_2) Caption('Spin Edit Box:')
Displayposition(4) Height(25) Left(24) Parent(#COM_OWNER) Tabposition(5)
Tabstop(False) Top(136) Width(97)
Define_Com Class(#PRIM_LABL) Name(#LABL_3) Caption('When AutoTab is set
to true and you reach the end of an input field, the cursor moves automatically to
the next.') Displayposition(8) Height(33) Left(16) Parent(#COM_OWNER)
Tabposition(8) Tabstop(False) Top(8) Width(425)


EVTROUTINE handling(#CKBX_AUTOTAB.Click)

IF cond('#CKBX_AUTOTAB.ButtonState = Checked')
SET com(#EMPNO) AUTOTAB(TRUE)
SET com(#EDIT_1) AUTOTAB(TRUE)
SET com(#SALARY) AUTOTAB(TRUE)
SET com(#spdt_1) AUTOTAB(TRUE)
```

```
ELSE
SET com(#EMPNO) AUTOTAB(false)
SET com(#EDIT_1) AUTOTAB(false)
SET com(#SALARY) AUTOTAB(false)
SET com(#spdt_1) AUTOTAB(false)
ENDIF

ENDROUTINE

END_COM
```

Without the Autotab True checkbox selected:



Autotab True checkbox is selected now:



Writing to the IFS generates a 'Resource busy' error

Problem Description

stack trace: java.io.FileNotFoundException: Resource busy.
/jsm/instance/trace/479650/CLIENT00000023/XML_CONTENT000003.XML
at java.lang.Throwable.<init>(Throwable.java:195)
at java.lang.Exception.<init>(Exception.java:41)
at java.io.IOException.<init>(IOException.java:40)
at java.io.FileNotFoundException.<init>(FileNotFoundException.java:46)
...
...
...
Create exception message : Resource busy.
/jsm/instance/trace/479650/CLIENT00000023/XML_CONTENT000003.XML

Resolution

The problem can occur when LANSA Integrator tracing is enabled and is due to a 'lock' on a directory or file in the path specified. The FileOutputStream class is trying to create a file, but the underlying IFS API's cannot do it.

Note: Do not get confused by the FileNotFoundException in the Exception trail. This message is misleading.

Possible reasons for a lock:

There are potentially other programs using/holding folders/files in the same JSM directory. Because of this lock the current process in Integrator is failing to get exclusive access to write to the desired path. Programs known to have interfered with JSM processing are NETSERVER, MIMIX and other IFS/iSeries Monitoring programs.

There is also bug in IBM's NetServer to do with not removing locks.
OSP-INCORROUT RESOURCE BUSY OCCURS WHEN OPENING FILE
==========
Use the following the command to see if you have this PTF applied?
DSPPTF LICPGM(5722SS1) SELECT(SI26582)
==========
http://www-912.ibm.com/a_dir/as4ptf.nsf/a18db68aae4a7d81862566ba005d145c/15a430156b59187d8625728e005d7908?OpenDocument&Highlight=2,SI26582>

Temporary Workaround

To establish whether this error is caused by the combination of LANSA Integrator tracing enabled and either of the 2 reasons above. you can temporarily turn off the tracing. If this error is a result of tracing attempting to write a trace file, then turning off Tracing will bypass the issue.

Windows Help (.hlp) files no longer supported from Vista onwards

If you provide documentation with your deployed LANSA applications, please note:
As from Windows Vista, Microsoft will no longer supply the Windows Help reader. Refer to The Windows Help (WinHlp32.exe) program is no longer included with Windows operating systems starting with Windows Vista (http://support.microsoft.com/kb/917607) for a full discussion of this dropped support, including alternative help options as suggested by Microsoft.

If you have a LANSA developed application that uses Winhelp and you intend deploying the application to Vista, the help will not work as files with a file type of .hlp will no longer be viewable. You will need to change your application to use a help content that is supported by all currently supported Microsoft operating systems.

Furthermore, Microsoft have prohibited distributing winhelp viewer and have advised that users should instead download the viewer from the Microsoft download Centre.

LANSA's planned support for the Vista operating system is outlined in Vista support in LANSA (http://www.lansa.com/support/tips/t0423.htm).