New VLF version

Lots of new features in EPC132100 of the Framework!

The LANSA EPC132100 is completely related to the VLF and introduces many new features and enhancements. The highlights are described in this newsletter.
1. **Unicode Support**
The Framework can now handle Unicode data for certain parameters in certain methods. Unicode data can also be used in code tables, the Virtual Clipboard and in tracing.

2. **IBM i Password Management**
New options allow IBM i passwords to be managed from the Framework logon screen. IBM i user profile error checking (e.g.: wrong password, password expired) has also been improved.

**IBM i Password Management**
The Framework now has the ability to change IBM i passwords and to check their expiry date. The User Administration Settings tab in Framework Properties has these new settings:

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allow IBM password change</td>
<td>Check this box to allow password change.</td>
</tr>
<tr>
<td>Check Password Expiry</td>
<td>Check this box to compare the password’s expiry date with the current date.</td>
</tr>
<tr>
<td>Warn before (days)</td>
<td>Specify how many days before the expiry date to start issuing warnings.</td>
</tr>
</tbody>
</table>

**New IBM i Host Server Mapper properties**
In the Server Details tab you need to specify the IBM i server to connect to for password expiry checks and/or password change requests:

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name / IP address</td>
<td>Name or IP address of the IBM i Server Mapper. Supports full 40 character long IPV6 type addresses.</td>
</tr>
<tr>
<td>Port</td>
<td>Port of the IBM i Server Mapper to connect to. Defaults to IBM default.</td>
</tr>
</tbody>
</table>
IBM i Password Expiry Checking
To enable password expiry checking, you need to:

- Select Check Password Expiry in the User Administration Settings tab of the Framework properties.
- Specify a number of days greater than zero to start receiving warnings.
- Specify an IBM i Host Server Mapper name or IP address and the correct port.

IBM i Password Expiry Checking in Windows
During log on, when IBM i password checking is enabled, the Framework checks the expiry date of the password of the user logging on and compares it with the current date. If the difference is less or equal than the specified value in Warn before (days) a warning is issued.

For example:

![Password expiry check](image)

IBM i Password Expiry Checking in Web
In the web platform this feature has been added to the web IIP. Refer to the shipped function UFU0001 in process UF_SYSBR.
Note that fields have been added to the Exchange command.

During log on, the password’s expiry date is compared with the current date. If the difference is less or equal than the specified value for Warn before (days), a warning is issued.

For example:
Changing the IBM i Password
To enable IBM i password change you must check the Allow IBM i password change option in the User Administration Settings tab of Framework Properties.

Changing the IBM i Password in Windows
When IBM i password changing is allowed and the IBM i Host Server Mapper name or IP address has been specified, the Change IBM i Password button is displayed on the log on screen:

Clicking the Change IBM i Password button brings up the Change IBM i Password dialog:

Type in the old and the new password, and then click the Change Password button.
A message indicates if the change was successful:

Or unsuccessful:
Changing the IBM i Password in Web

On the web, if the server to connect to was saved as the Deployment Server and the IBM i Host Server Mapper name or IP address has been specified, the Change IBM i Password button is displayed in the log on panel:

Type in the old and the new password and press the Change button.

A message indicates if the change was successful:
To change the IBM i password on the web, the user must specify the IBM i server and port in the startup URL:

+IBMISERVER=<ibmiServer>+IBMIPORT=<ibmiPort>

Where:

<table>
<thead>
<tr>
<th>&lt;ibmiServer&gt;</th>
<th>is the ip of the server where the profile's password is to be changed</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;ibmiPort&gt;</td>
<td>is the port for changing IBM i profile passwords (usually 449)</td>
</tr>
</tbody>
</table>

For example:

+IBMISERVER=10.2.0.181+IBMIPORT=449
3. **Overriding Object Captions**
You can provide additional information to the end-user by temporarily overriding object captions.

4. **Indicate Development Status of Objects**
Developers can now attach development status indicators and notes to Framework objects. The indicators and notes are visible when the Framework is run in development mode.

5. **Improved Security**
VLF-WEB/NET temporary state files are now encrypted.
6. New Theme
The 2014 Clean theme is a low key, clean, crisp and flat theme that is designed to reflect the style used in later Windows products. The predominant colour is white and the predominant style is low key and flat.

Classical Windows features like bright colours, highlighting and 3D borders and colour gradients that are used for screen ornamentation have largely been removed.

7. Visual Styles can be Changed at Run Time
If you need a visual style to change at run time, you can swap in a new style or styles from any command handler, filter, or snap in instance list.

8. Option to Enable WEBEVENT Functions
The ability to use WEBEVENT filters or command handler is a now a deprecated feature.
If you have an existing Framework that already contains WEBEVENT filters or command handlers and you wish to add more, you can enable Framework for WEBEVENT functions on the on the Framework Properties tab.
9. Reusable Parts as Instance List Relationship Handlers

Instance list relationship handlers can now be created as reusable parts. The code for the reusable part version is simpler, and similar to the code used by filters to write to the instance list.

For example, this is the code used to write the sections for an organization into the instance list, in the shipped demo:

```plaintext
WHEN VALUE IS('DEMO_ORG')
* Expand Sections in a Department/Organization

#DEPARTMENT := #AKEY1
SELECT FIELDS(*ALL) FROM_FILE(SECTAB) WITH_KEY(#DEPARTMENT)

Signal uAddListItem AKey1(#DEPARTMENT) AKey2(#SECTION) VisualId1(#SECDESC) VisualId2(#SECTION) AColumn1(#SECADDR1) AColumn2(#SECADDR2) AColumn3(#SECADDR3) AColumn4(#SECADDR4) NColumn1(#SECPHASE) BusinessObjType(#TargetType)

ENDSELECT
```

The VLF looks for a component for the relationship handler (rather than a function) if the following property is checked:

Business object properties --> Instance List tab --> Relationship properties --> Use a Reusable part.

10. Tailor User/Authority reports

It is now possible to tailor the User/Authority reports produced by the Framework. The output can be a .csv file or it can be written out to a database. The report structure and content is fully customizable. This feature is designed for advanced developers.

Developers can write their own user/authority report programs for administrators. The program used for user/authority reporting is specified in the Framework properties --> User Administration Settings --> Authority Settings --> Report on Users – Imbedded Interface Point (Id). If unspecified, the standard user/authority report is produced.
11. Reusable parts as code table data handlers

Reusable parts can now be used as code table handlers. This simplifies coding because most default processing is done by the ancestor component.

Reusable part data handlers can also deal with Unicode data.

To use a Reusable part as a Code Table Data Handler tick the Use Reusable Part property check box and enter the identifier of the data handler component in the Reusable Part Data Handler (ID) property field.

UF_TDH01 is the default Reusable part Code Table Data handler and can be used as is, or as a starting point for a custom data handler.

12. The Programming Code Assistants feature has been updated

- All WEBEVENT code assistants have been removed.
- All images and screen shots have been updated to match the latest release of LANSA.
- The graphic quality of all images and screen shots has been improved.
13. Updated Organizations Business Object Instance List
The shipped Organizations Business Object Instance List provides improved summary and preview information when you hover over an item in the tree.

It does this by using the Pop up panel name (Id) option on the Instance List / Relations tab:

14. uQueryCanDeactivate reason codes
You can now find out what the user is trying to do when the uQueryCanDeactivate check occurs using the optional Reason Code.
15. **Shipped Resources Instance List Browser Updated**
The shipped Resources business object’s snap in instance list browser now uses (and requires) Direct-X features.

![Instance List Browser Screenshot]

16. **Improved demo images**
The demo images have been improved.

17. **Alternative find path algorithm in RAMP-TS**
Developers of large RAMP-TS systems using Axes 3.1 can now use an alternative Find Path algorithm, Find Path V2.

The new algorithm is much faster, but may occasionally result in RAMP finding a different path between screens than it did with the old algorithm.

*So thorough re-testing of the RAMP application is required if changing from one algorithm to the other.*

If users in large RAMP systems are experiencing delays the first time they go from one screen to another, it is possible that the number of RAMP nodes in the session are making it difficult for the Find Path algorithm to find a path between the screens. Using the new version of the algorithm may solve this problem.

To activate Find Path V2 use this statement in your login script:

```
GLOBAL_bUseFindPathV2 = true;
```

In the application trace, the line "Using Find Path V1" or "Using Find Path V2" will indicate which algorithm is being used.
## 18. Features No Longer Supported

The support of these features has been removed in EPC132100:

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>WEBEVENT functions deprecated</strong></td>
<td>The ability to use WEBEVENT filters or command handler is now a deprecated feature. If you have an existing Framework that already contains WEBEVENT filters or command handlers and you wish to add more, you can enable WEBEVENT functions on the Framework Properties tab.</td>
</tr>
<tr>
<td><strong>Auto Session Signoff</strong></td>
<td>Over time the auto session signoff feature has been rendered ineffective by pop-up blocking and browsers that do not allow secondary windows to be opened. Now when a VLF-WEB session is closed, an attempt is made to delete any associated temporary files it has in the temporary folder. If the close is initiated from a 'signoff' or 'exit' VLF command, the temporary files should be removed successfully. If the session close is initiated by closing the browser window, then whether the temporary files are deleted depends on the web browser being used: some browsers do not allow further requests to be sent from a closing window. To counter any build-up of temporary files, use the shipped VLF or OS features to regularly clear unwanted files from the temporary folder.</td>
</tr>
<tr>
<td><strong>Web Load Image</strong></td>
<td>The web load image feature has been removed.</td>
</tr>
<tr>
<td><strong>WAMTRANS=C</strong></td>
<td>The previously deprecated WAMTRANS=C (Client Side XML transformation) option has been removed from VLF-WEB and VLF.NET.</td>
</tr>
<tr>
<td><strong>Form Layout Assistant</strong></td>
<td>The previously deprecated FLA (Form Layout Assistant) feature has been removed from VLF-WEB and VLF.NET.</td>
</tr>
<tr>
<td><strong>Rich Text RAD-PADs</strong></td>
<td>Rich Text RAD-PADs are no longer supported.</td>
</tr>
<tr>
<td><strong>GZIP File Compression</strong></td>
<td>GZIP File Compression has been removed.</td>
</tr>
<tr>
<td><strong>VLF-WEB – All old Windows XP style themes have been removed</strong></td>
<td>The old XP theme set which was restricted to using Internet Explorer 9 (or earlier) have been removed. All browsers are now handled the same way with the same options. This is reflected in this old developer VLF-WEB launch form:</td>
</tr>
<tr>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
</tr>
<tr>
<td></td>
<td>All associated XXXXXXXX_WEB.HTM suffixed start up and launch files are no longer generated and can no longer be used.</td>
</tr>
</tbody>
</table>
CGICconvMode BINARY setting required for LANSAWEB CGI

As of V13, it is a mandatory requirement for the LANSAWEB CGI program to operate with the CGICconvMode BINARY setting in the HTTP Server configuration.

This requirement is primarily relevant to users who upgrade their existing LANSA for the Web system because the upgrade process does not alter existing HTTP Server configurations.

After upgrading, users may encounter unexpected web behaviours such as:
- Passing data as parameters in the URL returns empty values.
- Non-English characters are incorrectly displayed.

There are two ways to confirm that you have encountered this problem:
1. Run LANSA for the Web About page (http://mywebserver/cgi-bin/lansaweb?about)
   The CGIMODE and CGI_OUTPUT_MODE environment variables will contain the value: BINARY
   Another value other than BINARY will mean CGICconvMode is not correctly set.
2. On the IBM i, the HTTP Server CGI job will report the warning messages if not correctly set:
   WARNING: Support of non-binary CGICconvMode (EBCDIC) is now deprecated.
   WARNING: It is highly recommended to set CGICconvMode to BINARY.

If you encounter either situations, from your HTTP Server instance configuration:
1. Locate which COMLIB library is the lansaweb URL mapped to, for example:
   ScriptAliasMatch ^/cgi-bin/lansaweb(.*)/QSYS.LIB/DCXCOMLIB.LIB/LANSAWEB.PGM$1
2. Locate the corresponding COMLIB directory section and edit your configuration to set CGICconvMode BINARY, for example:
   <Directory /QSYS.LIB/DCXCOMLIB.LIB>
     . . .
     CGICconvMode BINARY
     . . .
   </Directory>
3. Restart your web server and run the LANSA Web clean up program for the new setting to take effect.

Note:
- If installing a new LANSA system, the HTTP Server configuration for your web instance created by LANSA will have this setting CGICconvMode BINARY created as part of the install process.
- If your HTTP Server configuration has CGICconvMode %%BINARY/BINARY%% listed, consider replacing this deprecated setting with the currently supported setting CGICconvMode BINARY.
Microsoft ‘Problem Steps Recorder’

You can use Problem Steps Recorder to automatically capture the steps you take on a computer, including a text description of where you clicked and a picture of the screen during each click (called a screen shot). Once you capture these steps, you can save them to a file that can be used by a support professional or someone else helping you with a computer problem.

It seems quite useful – if only because it is very simple to use and it looks like it is pre-installed on all Windows7 & 8 computers.

To record and save steps on your computer

1. Open Problem Steps Recorder by clicking the Start button ☰, and then typing psr. In the list of results, click psr.
2. Click Start Record. On your computer, go through the steps on your computer to reproduce the problem. You can pause the recording at any time, and then resume it later.
3. Click Stop Record.
4. In the Save As dialog box, type a name for the file, and then click Save (the file is saved with the .zip file name extension).

To view the record of the steps you recorded, open the .zip file you just saved, and then double-click the file. The document will open in your browser.

For all details, see:
You must have 64-bit JAVA to run JSM on a 64-bit PC

With V13 SP2, it is now mandatory to have 64-bit of JAVA on a 64-bit PC in order to use LANSA Integrator. Unlike previous versions of LANSA, where a warning message was issued if the correct JAVA version is not found, but the install continued, the V13 SP2 install now stops with the following message if the required JAVA version is not installed.

While the V13 SP2 install (on a 64-bit PC without 64-bit JAVA installed) will issue a message and not complete, if you are performing an upgrade to V13 SP2 you will experience unexpected behaviour.

The upgrade will continue to the end but you may see a corrupted message in the warning notes area, as per the image on the next page.
Solution
To finalize the upgrade of JSM to V13 SP2, you must:
- Download and install 64-bit JAVA.
- From the V13 SP2 DVD, select the Install LANSA option and take the option to Modify or Repair an existing install.
- Select the LANSA Integrator features to be Repaired.
LANSA LICENSE - F6 Add/Create and F22 Delete options not available

There are a few reasons why the F6 option to Add/Create licenses and the F22 option to delete a license are not available when running LANSA LICENSE.

The first possibility revolves around the QOTHPRDOWN user profile that is set up as part of a LANSA installation. You should check for either a disabled QOTHPRDOWN profile or not having QOTHPRDOWN as the LANSA owner. If QOTHPRDOWN is disabled, it needs to be re-enabled in order to be able to Add or Delete licenses. If you are not planning on using QOTHPRDOWN profile as the LANSA product owner, then you must perform the steps outline in the section “If not using QOTHPRDOWN as the Product Owner” in the Installing LANSA on IBM i guide.

Another possibility is that the user profile you are logged in with has the authority to view licenses but doesn't have the appropriate authority to Add/Delete licenses. Its worthwhile knowing the hierarchy of users that can Add/Delete licenses.

- If you log in as QSECOFR or a user that is part of the QSECOFR group, you will have F6/F22 available in LANSA Licensing (whether you specify a partition or not). For example LANSA LICENSE or LANSA LICENSE Partition(NNN) will both work OK.
- If you log in with the security officer for partition SYS and run LANSA Licence without specifying the partition, you will have F6/F22 available in LANSA Licensing. For example, LANSA LICENSE will work OK.
- If you log in with the security officer for partition SYS and run LANSA Licence specifying partition(SYS), you will have F6/F22 available in LANSA Licensing. For example, LANSA LICENSE Partition(SYS) will work OK.
- If you log in with the security officer for partition NNN and run LANSA Licence specifying partition(NNN) on the command, you will have F6/F22 available in LANSA Licensing. For example, LANSA LICENSE Partition(NNN) will work OK.
- If you log in with the security officer for partition NNN and run LANSA Licence without specifying partition(NNN), you will not have F6/F22 available in LANSA Licensing. For example, LANSA LICENSE will not work OK.
A further thing to check is whether the user profile running LANSA LICENSE has the correct authorities to its own profile. To explain this possibility, let’s say that we have set a user profile with very few rights (User: NORIGHTS) as the partition security officer for a partition AAA.

- When the user profile has *ALL rights to itself according to DSPOBJAUT OBJ(NORIGHTS) OBJTYPE(*USRPRF) and we execute LANSA LICENSE PARTITION(AAA) then the license screen does have “F6=Add/Create”.
- When the user profile has no rights to itself according to DSPOBJAUT OBJ(NORIGHTS) OBJTYPE(*USRPRF) (either covered by *PUBLIC *EXCLUDE or NORIGHTS *EXCLUDE) and we execute LANSA LICENSE PARTITION(AAA) then the license screen DOES NOT HAVE “F6=Add/Create”.

So, if you run the command

**DSPOBJAUT OBJ(NORIGHTS) OBJTYPE(*USRPRF)**

and you see

<table>
<thead>
<tr>
<th>Group</th>
<th>User Group Authority</th>
</tr>
</thead>
<tbody>
<tr>
<td>*PUBLIC</td>
<td>*EXCLUDE</td>
</tr>
<tr>
<td>NORIGHTS</td>
<td>*EXCLUDE</td>
</tr>
</tbody>
</table>

or something to that effect, then that also a likely reason.
**Using Custom Settings for V13 SP2 Deployments**

Visual LANSA deployments allow users to create an Application that install via MSI.

Introduced in V13 SP2 was the file CustomSettings.wxi that allows for some customisation of how the MSI will run. The location of this file is: `<lansa>\x_win95\x_lansa\x_apps\x_wix\source\CustomSettings.wxi`

This contains source for four different settings that can be turned on or off, simply by changing the number from 0 (off) to 1 (on) or vice versa. This has the benefit of not having to add the options to the command line when running the installation.

---

**A. ShowExecApp**

ShowExecApp sets whether the `<Start Application>` checkbox option is to be shown or not, after the installation has completed. It is very useful for giving the user the option as to whether they wish to run the application or not. Alternatively, you can take that option away from the end user and either have it run or not without prompting the user. This can be done in co-ordination with the next two settings. Note, if you have this turned on (set to 1), then by default, the check box will be selected regardless of the next two settings.
B. LaunchApplication
LaunchApplication is an option to have the application run straight after the install has completed, before the user has exited the install dialog. This is a good to use for client applications that have a high reliance on UI.

C. LaunchApplicationWhenFinishClicked
LaunchApplicationWhenFinishClicked will control whether or not the application is to be launched when the you click the Finish button on the exit dialog. Turning this off is useful when there is no UI for your application, for example, if you are deploying a Server installation.
D. SkipLicenseDlg
SkipLicenseDlg gives the option to pass through without displaying the End-User License agreement.

Note: Adding this option means that the user automatically agrees to the terms and conditions.

In all cases when you are trying various or new types of deployments it is always crucial to test thoroughly before distributing your package.
**WAM Editor not displaying design view on Windows 8.1 (32-bit only)**

Using Windows 8.1 (32-bit) may cause an issue with trying to view the Web design, where you will receive an error when trying to open the design view and the design will appear blank.
**Cause**
This is caused by the Visual LANSA not having authority to write to %temp% where the WAM Editor is trying to create temporary cache files.

**Solution**
The solution to this is to set your Visual LANSA to always run as administrator. This will mean that Visual LANSA has full permissions to read and write to the %temp%. 
Note

- This issue only seems to occur on 32-bit versions of Windows 8.1. However, if the same issue occurs with Windows 8.1 64-bit, the same solution can be attempted.
- If this does not solve your issue, WAM Editor logging can be enabled by creating a file called `wamed.def` in `<LANSA>\LANSA\WAMEditor` and adding the following script to it:

  ```plaintext
  [trace]
groups=*ALL
  ```

  Send the resulting log (found in `%Temp%\wamed\<006044>\trace\wamed.log`) to LANSA Support.
Using JSMGETENV program to access CGI variables

Often developers are required to access the CGI variables. JSM provides a JSMGETENV program to allow this.

The JSMGETENV program can be used in the JSMADMEXT, JSMPXYEXT, JSMDRTEXT and JSMLSAEXT programs to read CGI environment variables.

The program status code will have one of the following values:
OK, ERROR or NOVAR.

```assembly
DCL VAR(&ENVNAME) TYPE(*CHAR) LEN(128)
DCL VAR(&ENVVALUE) TYPE(*CHAR) LEN(1024)
DCL VAR(&ENVSTS) TYPE(*CHAR) LEN(5)
CHGVAR VAR(&ENVNAME) VALUE(HTTPS)
CALL PGM(JSMGETENV) PARM(&ENVNAME &ENVVALUE &ENVSTS)
IF COND(&ENVSTS *NE 'OK') THEN(DO)
   GOTO END
ENDDO
```
JSM jobs and threads explained

1. Suppose that 3 Java Classes are simultaneously running, as LANSA Integrator accepted 3 different requests for 3 different services it provides (for example, PdfSpoolFileService, PDFDocumentService and ExcelReadService).

Q: How are these 3 requests handled?
A: Each JSM_OPEN to run a service class has its own thread.
So there are 3 Threads running.
One thread for PDFSpoolFileService
One thread for PDFDocumentService
One thread for ExcelReadService

Q: Does LANSA Integrator handles these 3 simultaneous requests in the same job (JSMJOB01) and in the same JVM?
A: Yes.

2. Suppose that 1000 concurrent users use a Java class, either from a LANSA program or from an external source.

Q: How many JAVA Virtual Machine's (JVM's) does LANSA Integrator use in that case?
A: One.

Q: How many sessions are simultaneously open?
A: For 1000 requests, 1000 threads are spawned.

Q: Our customer is planning to use LANSA Integrator for the handling a lot of requests in our Applications. We would like to know how LANSA Integrator handles load balancing and how it interacts with the resources provided.
A: You can configure a POOL server and start JSM (STRJSM) in multiple JVM jobs.

JSMJOB01 QOTHPRDOWN BCH .0 CMD-RUNJSM TIMW <= STRJSM/RUNJSM CL program
JSMJOB01 QOTHPRDOWN BCI .0 JVM-com.lansa. TIMW <= One Java Virtual Machine

JSMJOB02 QOTHPRDOWN BCH .0 CMD-RUNJSM TIMW <= STRJSM/RUNJSM CL program
JSMJOB02 QOTHPRDOWN BCI .0 JVM-com.lansa. TIMW <= One Java Virtual Machine

JSMJOB03 QOTHPRDOWN BCH .0 CMD-RUNJSM TIMW <= STRJSM/RUNJSM CL program
JSMJOB03 QOTHPRDOWN BCI .0 JVM-com.lansa. TIMW <= One Java Virtual Machine

JSM_OPEN BIF call to the pool server will round-robin the request to one of the multiple JVM jobs. Using a POOL server, the service threads and the GC work load is spread across multiple JVM jobs.
**Q:** Can one JSM instance (One JSM subsystem with one JSMJOB and one JSMJOB01) handle 1000 concurrent requests?

**A:** Yes, but this is not recommended for reasons of performance and load.

Spreading the load across multiple JVM means that you have multiple Garbage Collectors (GC) running.

Each JVM has one Garbage Collector.

If you have 1000 threads running in one JVM, you only have one GC working.

If you have 333 threads running in each JVM, you have three GC working.

Therefore, the load on each JVM is reduced.

When considering this topic, the areas of processors, memory and other IBM work management considerations, i.e. private memory pools, should be included. [http://pic.dhe.ibm.com/infocenter/iseries/v7r1m0/topic/ryaha/ryaha.pdf](http://pic.dhe.ibm.com/infocenter/iseries/v7r1m0/topic/ryaha/ryaha.pdf)

3. Some other considerations

- In 32-bit mode, the Java object heap cannot grow much larger than 3 gigabytes. You will also be limited to running approximately 1000 threads.

- If your application requires more than 1000 threads or a Java object heap larger than 3 gigabytes use the 64-bit version of IBM Technology for Java.
**How to view mapped drives on Windows 8 during Visual LANSA Network Client Installation**

A Network Client to a Windows Server (Slave or Independent) is a thin client that has only shortcuts installed. In this model, all repository and LANSA processes are accessed from the Windows Server, meaning there is a smaller footprint of files left on the development PC's.

During the Visual LANSA Network Client installation process you can view mapped drives. On Windows 8 (and Windows 8.1), because of UAC policies, you are unable to view mapped drives during the installation process.

If you encounter this, you will need to add a registry setting that will allow you to view mapped drives so you can complete the Visual LANSA Network Client install.

4. Open Regedit.
5. Locate and then right-click the registry subkey `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System`.
6. Select **New**, and then click **DWORD Value**.
7. Type `EnableLinkedConnections`, and then press ENTER.
8. Right-click `EnableLinkedConnections`, and then click **Modify**.
9. In the **Value data** box, type `1`, and then click **OK**.
10. Exit Registry Editor, and then restart the computer.

This will enable you to the view mapped drives during the install.
After the install is complete is a good idea to change the EnableLinkedConnections value back to 0 and restart your PC again as this will disable the feature.

This will not affect your ability to run Visual Lansa as it is only the install that requires this setting to be on.
Garbage Cleanup of JVM using SERVICE_RECLAIM command

At times, application developers may need to capture HTTP request header information (trace) sent by a HTTP client or browser.

Built-in into the latest JSM is the SERVICE_RECLAIM command.

This command can be called on any loaded service. This command calls the System.gc () method to run a 'forced/explicit' Garbage Collection (GC) Cycle.

You can also enable Heap compaction on a this type of explicit GC call.

Xcompactexplicitgc

To trace this event, simply add the compaction keyword to the trace GC option.

-Xtgc:excessiveGC,compaction

A possible usage scenario

At the end of a batch run of JSM service processing, you can call SERVICE_RECLAIM to explicitly run the Garbage Collector and compact the heap.

This will return your JSM heap to the minimum number of live objects and the heap will be compacted. This heap compaction might also cause a heap shrinkage. Therefore heap will have a large contiguous block of memory to allocate new objects and will be ready for the next batch run of JSM services.

For more information refer to Java Diagnostics Guide - Heap expansion and Heap shrinkage.

The following are some key points to take note of:

- Heap shrinkage occurs after garbage collection while exclusive access of the virtual machine is still held.
- On initialization, the JVM allocates the whole heap in a single contiguous area of virtual storage.
- The amount that is allocated is determined by the setting of the -Xmx parameter.
- No virtual space from the heap is ever freed back to the native operating system.
- When the heap shrinks, it shrinks inside the original virtual space.
- Whether or not any physical memory is released depends on the ability of the native operating system.
- You never see and decrease in the amount of virtual storage that is used by the JVM.
- You might see physical memory free size increase after a heap shrinkage.
- The native operating system determines what it does with decommitted pages.

Where paging is supported, the GC allocates physical memory to the initial heap to the amount that is specified by the -Xms parameter. Additional memory is committed as the heap grows.
USE BUILTIN(JSM_OPEN) WITH_ARGS('LOCALHOST:7560') TO_GET(#JSMSTS #JSMMSG)

USE BUILTIN(JSM_COMMAND) WITH_ARGS('SERVICE_LOAD SERVICE(XXXService)') TO_GET(#JSMSTS #JSMMSG)

USE BUILTIN(JSM_COMMAND) WITH_ARGS(SERVICE_RECLAIM) TO_GET(#JSMSTS #JSMMSG)

USE BUILTIN(JSM_CLOSE) TO_GET(#JSMSTS #JSMMSG)

**SystemDefault.properties**

- #AllowOptions
- -verbose:sizes
- -Xgcpolicy:gencon
- -Xcompactexplicitgc
- -Xtgc:excessiveGC,compaction
- -Xverbosegclog:GC-%Y-%m-%d-%H%M%S.XML

**JSM job spooled file output**

- Xmca32K  RAM class segment increment
- Xmco128K  ROM class segment increment
- Xmnx10000K  initial new space size
- Xmnx512M  maximum new space size
- Xms40000K  initial memory size
- Xmos30000K  initial old space size
- Xmox2G  maximum old space size
- Xmx2G  memory maximum
- Xmr16K  remembered set size
- Xlp64K  large page size

Available large page sizes: 4K 64K

- Xmso256K  operating system thread stack size
- Xiss2K  java thread stack initial size
- Xssi16K  java thread stack increment
- Xss256K  java thread stack maximum size

ExcessiveGC: gcid="1" intimems="4.100" outtimems="815.502" percent="0.50"
averagepercent="0.03"

ExcessiveGC: gcid="2" intimems="8.253" outtimems="940.371" percent="0.87"
averagepercent="0.07"

ExcessiveGC: gcid="3" intimems="12.067" outtimems="1182.194" percent="1.01"
averagepercent="0.11"

Compact(1): reason = 4 (forced gc with compaction)
Compact(1): Thread 0, setup stage: 0 ms
Compact(1): Thread 0, move stage: handled 71336 objects in 6 ms, bytes moved 3658472.
Compact(1): Thread 0, fixup stage: handled 0 objects in 0 ms, root fixup time 0 ms.
Compact(1): Thread 1, setup stage: 0 ms.
A golden oldie VLF

Adding tracing options into deployed applications

The shipped demo framework ships with these end user tracing on/off options:

These are normal framework level commands.

They execute shipped demo VL part DF_DET46 - passing it parameter TRUE or FALSE to it ............

Note that DF_DET46 is a hidden command. DF_DET46 looks like this:

**Begin_Com** Role(*EXTENDS #VF_AC020)**

Mthroutine Name(uExecute) Options(*REDEFINE)

Invoke Method(#Com_Ancestor.uExecute)

Set Com(#AvFrameworkManager) Avtrace(#COM_OWNER.avAlphaArg1)

Endroutine

End_Com

If you hide away some tracing options like this in deployed frameworks it may make tracing out issues on end users PCs simpler.
Mobile Development with LongRange

Successful businesses are making mobile apps a natural extension to their businesses, such as for business partners ordering products and services, travelling sales reps accessing customer information, staff on the floor managing inventory, and many more scenarios where information needs to be available at the right time and location. Mobile apps allow users to access existing corporate information and capture new information at the source, including scanned barcodes, photos, videos, audio recordings, maps and geo-location. LANSA lets developers use their existing RPG, COBOL or LANSA skills to productively develop mobile apps and help companies achieve quick return on investment.

Have a look at:
http://www.longrangemobile.com/casestudies/index.htm

for some interesting case studies: