

LANSA Composer

Code-free Business Process Integration

"If you can click then you can connect with LANSAs Composer"

Practical Business Process Integration

Every business is under pressure to automate manual processes, better integrate internal systems and securely share data with external parties. Until now only large companies could afford the cost and complexity of Business Process Integration software. LANSAs Composer is a BPI solution designed and packaged for any size of business.



LANSA Composer is a highly visual and code-free tool that any business analyst or expert user can operate. This simplicity enables everyone to fully participate in the process, which leads to more predictable outcomes and less reliance on scarce technical resources.

LANSA Composer is the first BPI solution designed specifically for the heartland of the System i/iSeries community – the small or medium sized business. It is simple to install, easy to use and truly affordable.

In This Issue

LANSA Composer	page 1	JSM Service File paths	page 11
New VLF version available	page 6	Integrator Functions used properly	page 12
Unable to Transform Input XML WAMs	page 10	X-run params Client/Server environm.	page 13

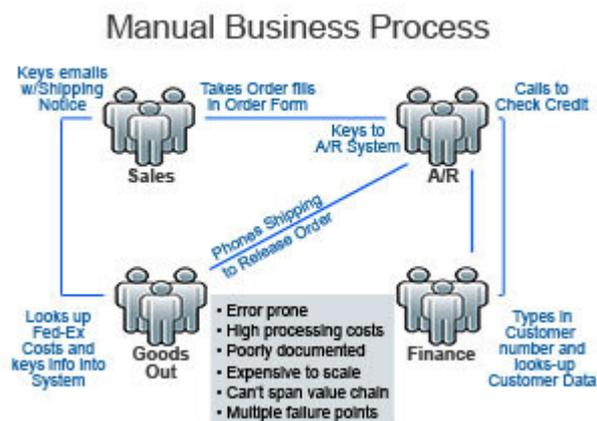
Introduction

Every business is under pressure to automate manual processes, better integrate internal systems and securely share data with external parties. But most Business Process Integration (BPI) tools are unsuited for quick and simple solutions to urgent business requirements. LANSA Composer delivers the power of BPI technology in a highly visual environment so designers and business analysts, rather than developers, can rapidly automate and integrate business processes.

What can LANSA Composer do for your business?

A Business Process Integration solution like LANSA Composer differs from other middleware because it takes a top-down, process-centric approach. The goal is to automate manual processes and eliminate re-keying of data by reducing the amount of paper, email, fax and human interaction required to complete a given business process.

LANSA Composer is the first BPI solution designed specifically for the heartland of the System i/iSeries community – the small or medium sized business. It is simple to install, easy to use and truly affordable.



Before LANSA Composer

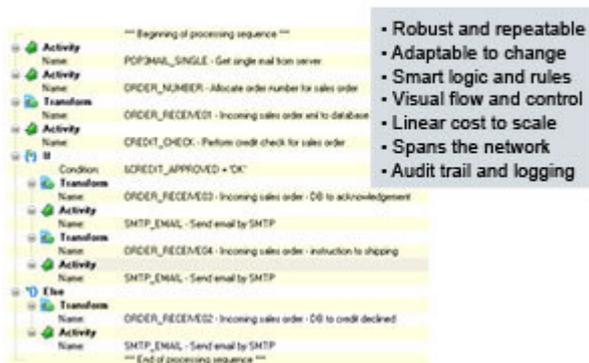
In many businesses, we have found that one of the main obstacles to achieving operational efficiency is actually knowing which tasks are performed when, by whom and under what circumstances? A lot of process expertise is carried around in people's heads, some is written down, but very little is properly documented or automated.

LANSA has been active in the application development and integration tools space for twenty years and have identified a set of common issues that seemed to hold true almost regardless of an organization's size, industry or geographic location:

- Businesses relies too much on error-prone manual process
- Existing data transfers are batch-oriented and rigid
- Application developers are over-burdened
- Implementing middleware products costs more than the software itself
- BPI technology is beyond the reach of most companies

LANSA Composer was designed to help companies overcome these issues and to bring the power of Business Process Integration technology to everyone.

Automated Business Process



After LANSACOMposer

With LANSACOMposer you can capture, model and implement a set of automated processes that, like robots on a factory floor, are more predictable and future-proof than their manual predecessors. LANSACOMposer can also be used to replace any batch processing or semi-automated file transfers with real-time interfaces between applications and databases.

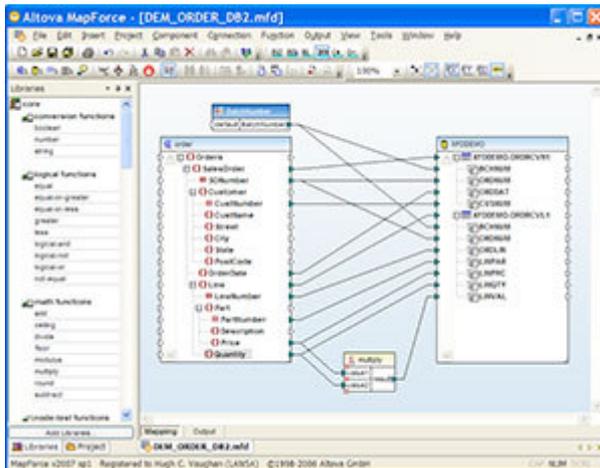
The end result is a more robust and efficient operating environment in which computer automation is, once again, making people and their employers more productive.

Typical Business solutions include:

- Streamline manual processes, such as sales order processing
- Replace batch processing with real-time communication
- Link back-office and ERP systems to new applications like CRM
- Easily share data between System i, Windows and Web applications
- Connect any application on any platform via XML Web Services

LANSA Composer is a highly visual and code-free tool that any business analyst or expert user can operate. This simplicity enables everyone to fully participate in the process, which leads to more predictable outcomes and less reliance on scarce technical resources.

Overview



Visually map data transformations using a drag-and-drop interface to transform data between different formats.

What is LANSA Composer?

LANSA Composer is a design and execution platform for integrating business activities involving transport and transformation of data and custom business processing. It satisfies the four key requirements of a Business Process Integration solution:

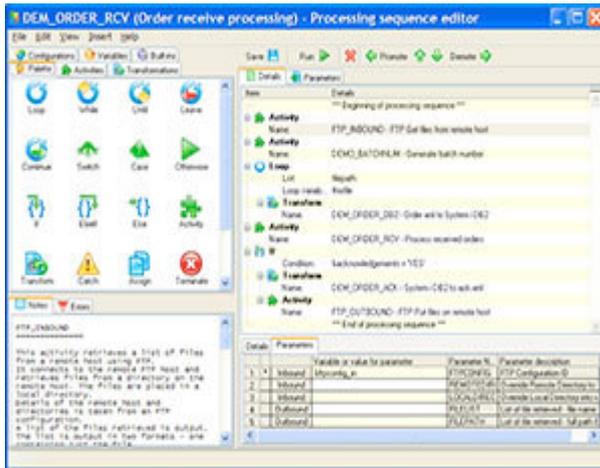
- **Transport** – moving data between source and target.
- **Transformation** – mapping data between many different formats.
- **Process Orchestration** – sequential and conditional execution of process flow.
- **Administration** – auditing, error-handling, logging, security and system operations.

A simple interface lets you wrap business logic as Activities and combine them with industry standard transformation and transport services in a single business process.

What can LANSA Composer Do?

In a graphical, drag-and-drop environment (without having to write program code), business analysts can do the following:

- Exchange business information and transactions in common and agreed formats with trading partners, internal business units and other business applications on the same or different computing platforms.
- Transform business information between XML, text and database formats.
- Invoke Web services, whether publicly available, published by trading partners or internal to your organization.
- Orchestrate transport, transformation and other Activities, pass variable data between them and apply conditional and structural directives to create multi-step business processes that can be executed and managed as a single unit.



Using the processing sequence editor you can easily define business activities without coding.

Who Should Use It?

LANSA Composer is intended to be used by business analysts to design and implement solutions to integration problems. It is not necessary to write program code to use LANSA Composer for solutions that use LANSA's standard transport and transformation Activities.

The implementation of the transport and transformation tasks and orchestrating them in business processes is all accomplished using highly visual graphical interfaces.

LANSA Composer can also be extended to encompass your custom business processing. You can define custom Activities that encapsulate the processing built by your developers. These custom Activities are orchestrated in the same way as the LANSA-supplied transport, transformation and other Activities.

LANSA Composer lends itself well to an environment in which the building of the custom business processing can be delegated to a services development group, while the combination and orchestration of the Activities is performed by business analysts.



New VLF version

Lot's of new features in the EPC826 of the Framework!

Web

- Open and save your Framework web applications before executing them to create new .HTM and .JS files!
- You can now create Framework-AJAX Applications to achieve optimal web performance with functionality close to that of a Windows rich client application.

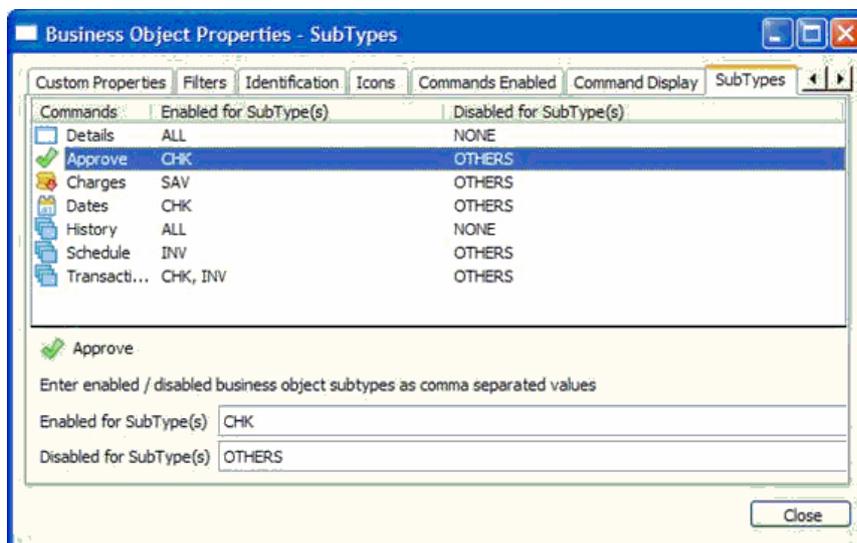
Demonstration Application

The Demonstration Application has been significantly revised and updated. To use it you need to open the new version of the shipped framework definition in file VF_Sy001_System_LastShipped.XML. See *Getting the Latest Version*. You need to remove the old demonstration application from any existing Framework versions as it may no longer function correctly.

Mini Filters

You can now specify the mini-filter as a panel to control all its display attributes. See *avMiniFilter* in the online guide. For example you can add combo boxes, drop downs, check boxes, and do instant editing on the panel.

Commands



Commands (and command tabs) can now be conditionally disabled or hidden depending on business object *SubTypes*.

Also, you can use the *Enable Child when Parent Selected* and *Enable Parent when Child Selected* properties to control command availability.

An option has been added to instance commands to allow a command never to be the default, even when the user used it on a previous instance (see *Default Command*).

Commands can now be executed without any user interface at all, that is, hidden from the user. See *Execute as Hidden Command* and *Hidden Command Handler Anatomy*.

Command Usage Tab

More details about the usage of command definitions are available on the Command Usage Tab:

- Default Command
- Hidden
- RAMP Destination

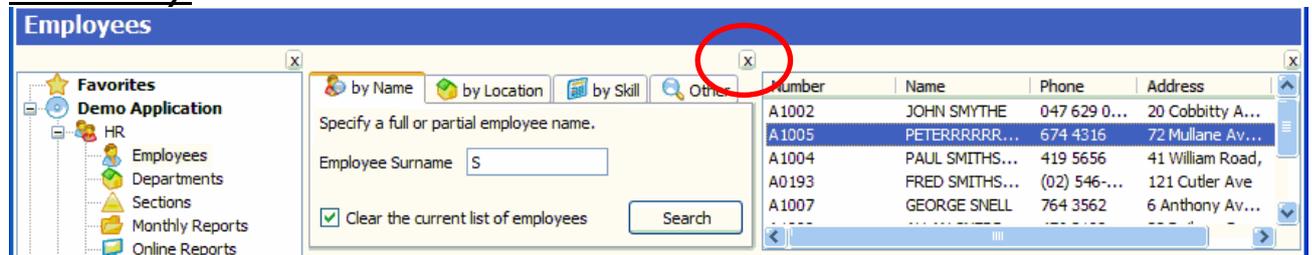
Performance

This version of the guide includes detailed information about *Assessing Performance in Framework web applications*.

HTML Startup Page

You can specify an HTML page as the startup page for the Framework or an individual application. You can provide links to other resources from this page. See *The Demonstration Application* for an example.

Usability



In Windows applications the main Framework panes can be shrunk and expanded to allow more efficient use of screen real estate. Use the *Allow Panes to be Shrunk and Expanded* option to enable this functionality.

Number	Name	Phone	Address	Zip
A1002	JOHN SMYTHE	047 629 0...	20 Cobbitty A...	2101
A1005	PETERRRRRR...	674 4316	72 Mullane Av...	2147
A1004	PAUL SMITHS...	419 5656	41 William Road,	2144
A0193	FRED SMITHS...	(02) 546-...	121 Cutler Ave	2001
A1007	GEORGE SNELL	764 3562	6 Anthony Av...	2164
A1008	ALLAN SNEDD...	476 2198	22 Railway Pa...	2160

You can use the new *Focused Input Field Style* property to highlight the input field that has focus with a different style in Framework web applications.

Security

In web applications it is now possible to replace the Framework's security model by specifying your own `avCheckAuth` method (this has always been an option in Windows applications).

To do this you need to supply a LANSA function and enroll it into the Framework. Refer to shipped example function `UFU0016` in process `UF_SYSBR` for more details. Also see [IIP Function to return web user details](#).

Web Deployment

It is now possible to use the Administrator interface in a deployed Framework web application to create users and groups, set authorities and custom property values.

This means that the Windows administrator components that were previously required in web deployment packages can now be omitted. See [Creating Web Interface for Maintaining Users and Authorities](#).

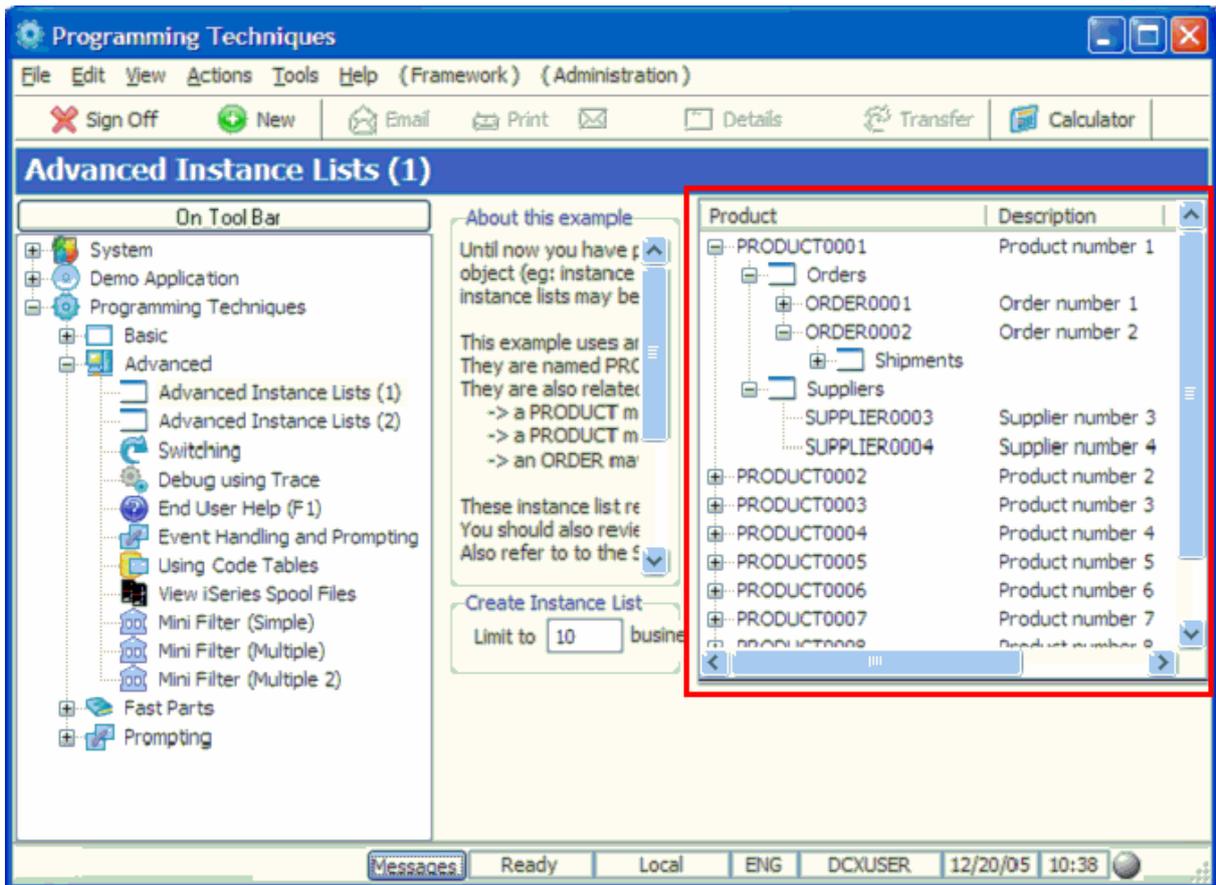
Instance List

[Instance Lists with Different Types of Objects](#) are now also available in Framework web applications.

Peer business objects are now shown with their icon to make it easier to distinguish different types of objects.

You can [Work with Hidden Child Objects](#) by double-clicking an instance list entry.

Instance lists now have a toolbar. You can set the [Instance List Tool Bar Location](#) and the [Instance List Tool Bar Height or Width](#) in the Instance List/Relations tab of the business object.



ActiveX instance lists are no longer supported.

In Windows applications, Instance lists that are saved and restored between VLF sessions now include the user object name/type of any current super-server connection in the saved file names. This means that different versions of a saved instance list can co-exist for example for Server1, Server2 and a Local DBMS.

Improved problem analysis

The Framework web application trace facility (+Trace=Y on URL) now displays the details of HTML pages arriving at the browser.

If an application detects a bad response from the web server, the user is given an option to analyze the nature of the error and display the problem HTML page. All details can be printed or copied to clipboard making it easier for end-users to provide helpful error information.

Position Menu option can be disabled

The Position menu option can now be turned off by designers so that users will not be able to change the relative position of any of the Framework display panes.

WAMS - Known reasons for Unable to Transform Input XML

WAM applications fail and the browser displays error message:

Unable to Transform Input XML

Known reasons for this error:

- WAM has not been compiled or the compile failed. A successful compile is indicated by a green tick next to the WAM in the repository viewer. If a green tick is not displayed then the WAM must be recompiled.
- A webroutine or weblet used by the WAM has not been checked in or has not been published
- When you create a WAM and compile, the objects are all created, but the layout is not included as part of a checkin/checkout by itself. The layouts need to be checked out as a separate object to the system running the WAM application.
- Standard weblets were not imported properly. Ensure the file std_keys.xsl exists in
{lansa_root}/x_win95/x_lansa/x_{partition}/web/lansa/{technology service}/{language}. If the file std_keys.xsl does not exist in this folder, you should re-import the weblets (via System Initialization)
- New languages - When creating new languages for a partition, ensure that a partition initialization is done prior to creating WAMs to ensure the standard weblets exist for the new language. This saves having to republish the WAMs.
- Ensure the xsl.cache directory in the IISPlugin subdirectory allows security rights for the IIS anonymous user to write there. If this is the case, to resolve, give the user (IIS default user - usually IUSR_<machine name> user) write and modify privileges for this xsl.cache directory.
- The XSL may need to be republished. Try republishing.
- The XSL has not been created in the correct directory. Ensure that the XSL files have been written to the language directory for the partition.

For example:

```
{lansa_root}\WebServer\IISPlugin\xsl.cache\x_{partition}\web\lansa\{technology service}\{language}
```

Note: If there are separate systems, the {system}_{port} part will look different for different users depending on their IIS Plug-in configuration.

For example:

```
{lansa_root}\WebServer\IISPlugin\xsl.cache\{system}_{port}\x_{partition}\web\{provider}\{technology service}\{language}
```

JSM service file paths

It is recommended to use the forward slash as the path separator and to avoid the use of the DOS drive designator.

The format of the path should be as follows:

For Windows you can specify:

Absolute path

For example:

/orders/order.xml

C:/orders/order.xml

C:\orders\order.xml

or

Relative path.

For example, orders/order.xml (note, no '/' at the start), in which case the order.xml document must reside in the orders directory under the JSM Instance directory on your server.

For the System i you can specify:

Absolute path.

For example:

/orders/order.xml

or

Relative path.

For example, orders/order.xml (note, no '/' at the start), in which case the order.xml document must reside in the orders directory under the JSM Instance directory on your server.

LANSA Integrator Functions used properly

Using work fields LANSA Integrator can have a big influence on the performance of your application. Fields used in a LANSA Integrator function will be converted into Java objects so they can be passed to the LANSA Integrator JSM. And the conversion costs time.

By default, all fields will be converted, so also work fields for example, and these work fields are not important at all for the end result. Let's say you want to read some file data and convert this into XML. Only the file fields are important to convert and not some possible existing work fields.

To ignore this conversion, you can give these (work)fields a name which begins with an Exclusion-prefix. This is JSM by default, so for example #JSMCOUNT or #JSMTMPA.

When having a complex function(s) with a lot of Database access (many fields/files), from which you eventually select/use only a few that are needed in the Transformation, it is better to do all the collecting, processing etc. in a *separate* function from the one where the actual JSM transactions will be triggered from (so separate the Integrator commands into a different function). The Data-set that you will need can be stored into a workinglist, and that workinglist can then be passed to the smaller Function that only handles the LANSA Integrator functionality, and has only uses a minimum number of fields. This will improve the processing time of LANSA Integrator.

X-RUN parameters in Client-Server environment

XENV

XENV determines the environment settings that LANSA uses to run an application.

Note: Contradict ional to what is being given sometimes by LANSA , you can NOT define multiple settings with this parameter ! As soon as you want more than one setting to be defined, you will need to use multiple statements, one XENV line for each setting.

Wrong samples:

```
XENV="X_DECIMAL_POINT_CHAR=. X_CURRENCY_SYMBOL=€"  
XENV=X_DECIMAL_POINT_CHAR=. X_CURRENCY_SYMBOL=€
```

Correct samples:

```
XENV=X_DECIMAL_POINT_CHAR=.  
XENV=X_CURRENCY_SYMBOL=€
```

To explain more about these features, see the next example that is commonly used in The Netherlands, Belgium, and maybe some more countries:

X DECIMAL POINT CHAR

This Environment Variable determines the sign that LANSA will use as a the decimal separator character.

XENV=X_DECIMAL_POINT_CHAR=,
will ensure that you can use the comma as the decimal separator sign in your application, e.g. when your PC's country settings is with the point, but you still want to use the comma.

If you have the opposite situation, you could use the same, but then like:

```
XENV=X_DECIMAL_POINT_CHAR=.
```

This setting can be specified on multiple locations:

In the X_RUN parameters behind the Icon on your desktop, or in the startup file that is used with the X_START command, but if this is applicable to all of your applications within the same partition, you can specify this in the file X_LANSA.PRO , that is located in the partition-directory (... \X_WIN95 \X_LANSA \X_ppp \), or when you use this for ALL your partitions, you can put this in the file X_LANSA.PRO and place that file into the LANSA runtime directory itself (... \X_WIN95 \X_LANSA \).

Be aware that when you work with Client-Server applications, you need to set the SAME sign always in the X_LANSA.PRO on the server ! Also, when you use this on the server , make sure you always use the correct setting for all your client applications that use CALL_SERVER_FUNCTION, or work from remote! (If either side is different , the ECHANGE FIELDS() command will notice an illegal character in your numeric field(s), and thus this will be discarded. So then the resulting value will be incorrect !)

This also implies that the visualisation in your application will be with the same decimal sign! This can NOT be used if you want to enter data, using the numeric keypad and the point, but want to see a comma as the decimal sign in your application!

X LANSA.PRO further explained

Most x_run parameter values can be permanently specified in a special profile file named x_lansa.pro.

A profile file can provide you with 3 benefits:

1. No need to type in most of the parameters every time you wish to use the x_run command.
2. Consistently specify all parameter values. Changing and forgetting parameters between different executions of x_run may produce differing results that confuse you.
3. Specify values for parameters that are not set up when you execute your application directly from within Visual LANSA.

This file can be created and edited by most standard source editors. You must observe these rules:

- If the first character of a line is a ";" then the line is ignored.
- Lines must be less than 256 characters long.
- As many parameters as can fit in 255 characters may be specified on a single line.
- Multiple parameters, specified on a single line, must be separated by space (i.e. blank) characters.

Note: X_LANSA.pro cannot be encrypted.

Note: The partition (PART=) and drive (DRIV=), if not the default drive, parameters must be specified on the command line to allow location of the profile file.

Note : If you directly specify an RPTH= parameter with your x_run command then x_lansa.pro should exist in the named RPTH= directory.

Note: For client-server applications , the x_lansa.pro profile file should be in the ..\x_win95\x_lansa\ or ..\x_win95\x_lansa\x_ppp partition directory .

Sample:

CMTH=C
CDLL=LCOMGR32.DLL
DBID=LX_LANSA
DBII=*NONE
DBUT=SQLANYWHERE
DBUS=DBA
GUSR=QPGMR
PSWD=SQL
XENV=X_DECIMAL_POINT_CHAR=.
XENV=X_CURRENCY_SYMBOL=€

Problem: The parameter settings from the file X_LANSA.PRO on the server , are NOT being used? Look at the settings of the parameter RPTH:

RPTH

RPTH (Remote Path) determines the location where LANSAs will use intermediate files , e.g. the Relative Record Numbers on Files.

For example if you always want to use a LAN drive for the RRN files and English:
RPTH=S:\RRNDIR\ LANG=ENG

Note: If you directly specify an RPTH= parameter with your x_run command then x_lansa.pro should exist in the named RPTH= directory.

Note: If you do not directly specify an RPTH= parameter on your x_run command, then x_lansa.pro should exist in the x_lansa\x_ppp partition directory (where "ppp" is the partition identifier) in the path containing Visual LANSAs (which is specified by DRIV= or PATH=).

It is recommended that you do not ever directly specify an RPTH= parameter on your x_run command. If you need to specify an RPTH= parameter, create an x_lansa.pro profile file in the x_lansa\x_ppp partition directory and include the RPTH= parameter value into it. This way you will not forget to include the correct RPTH= parameter.

Note: If the RPTH= parameter is not included, or if its value is changed from session to session, the file relative record numbers may be assigned from different areas, producing strange results.

As you can see, these were only a few of the many X_RUN parameters that can influence the behaviour of your application, and their (correct) usage will determine the way that your application will (or will not!) run!

This becomes more obvious when you want your application to run on another machine, without all your development environment, sources, database, etc.

Please read the LANSAs documentation for all the details on the X_RUN parameters and Client Server programming and deployment.