LANSA

Newsletter

August 2014

LANSA V13 SP2

The second Service Pack for LANSA Version 13 is now available with lots of new features!

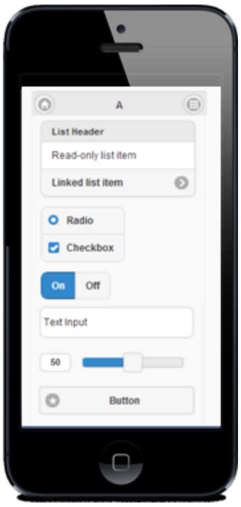
LANSA V13 SP2 introduces many new features and enhancements. The highlights are described in this newsletter.

1. Web Application Module Enhancements

This Service Pack 2 introduces many new features and enhancements for WAM's.

1a. jQuery Mobile WAM Enhancements

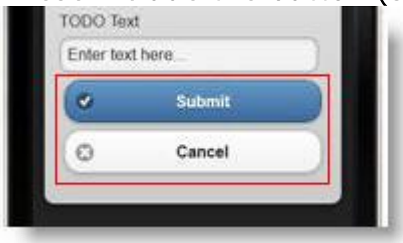
Commonly used jQuery Mobile weblets have been made easier to use and various new weblets have been introduced.



- **Easier to Design jQuery Mobile WAMs**

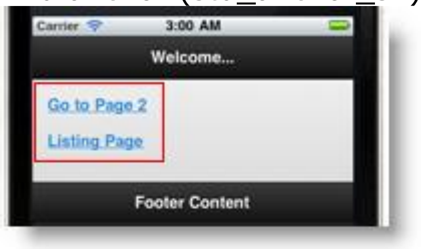
Simplified default versions of a number of jQuery Mobile Weblets have been introduced for ease of use.

These include the button (std_button_s1):



In This Issue

And anchor (std_anchor_s1) weblets:

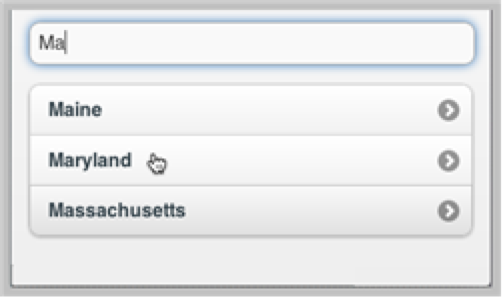


Use the previous versions of these weblets (std_button_v2 and std_anchor_v2) only if you need to add content to them.

- **New jQuery Mobile Weblets**

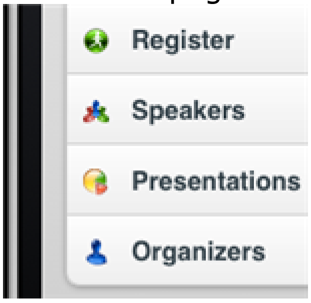
Autocomplete

The Autocomplete weblet provides suggestions while you type into the field.
The suggestions are provided by a webroutine using Ajax.



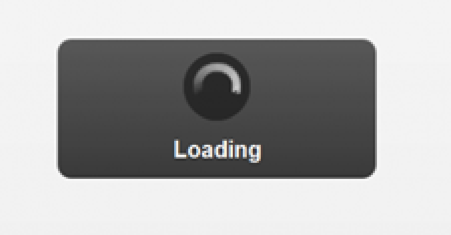
Image

The Image weblet displays an image.
The weblet has an option to load the image only when it comes into view, which helps render the page faster.



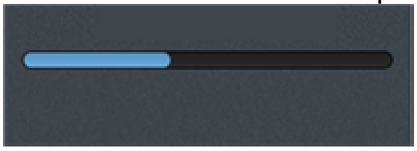
Loader

The Loader displays a small loading overlay when jQuery Mobile loads in content via AJAX, or when you want to perform an action that momentarily blocks user interaction.



Progress bar

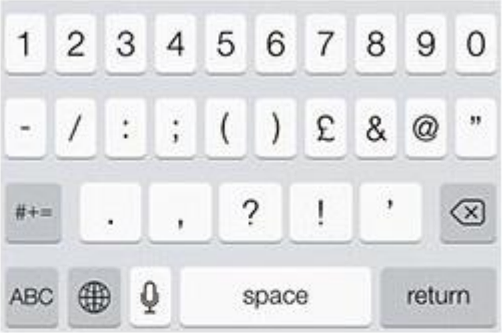
Progress bar displays the status of a determinate process.
It can also be used to display a value as a percentage of its maximum value.



- **Other improvements to jQuery Mobile**

Input Box Weblet

The Input Box weblet now supports input type=”number” to bring the correct keyboard to mobile devices.



Generated Lists for jQuery Mobile

Generated lists for jQuery Mobile are now similar in structure to the lists generated for Technology Service XHTML. You have access to columns by variable name.

Previously the XHTML and jQuery Mobile list methods were different, with the jQuery Mobile’s method giving more flexibility in adding content to the list entry at the expense of making it less easy to use in the WAM Editor.

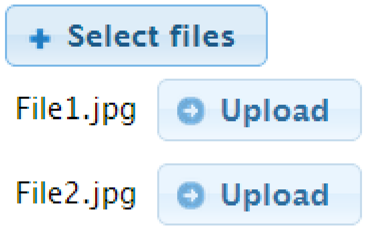
You can still use the more flexible and complex method by using the jQuery Mobile std_html_list weblet.

HTML5

- Default support for numbers in jQuery Mobile.
- Custom field visualisation support for new HTML5 input types.

1b. Support for File Uploads to a Webroutine

You can use the file upload weblet to select files to upload to the application server (into a temporary directory). The webroutine that receives the file upload can then manipulate the uploaded files as required.



The file upload uses multipart forms or Ajax. It uses BLOB capabilities inside LANSA to handle the upload of files. For more information, see the description of the XHTML File Upload (std_fileupload) weblet and the jQuery Mobile File Upload (std_fileupload) weblet.

1c. New XHTML Weblets

New XHTML Mobile weblets are available:

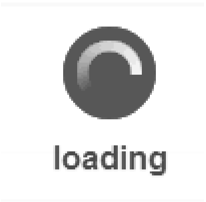
Image

The Image weblet displays an image.

The weblet has an option to load the image only when it comes into view, which helps render the page faster.

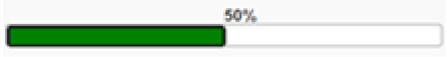
Loader

The Loader displays a small loading overlay when jQuery Mobile loads in content via AJAX, or when you want to perform an action that momentarily blocks user interaction.



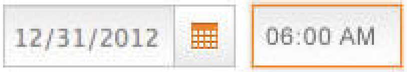
Progress bar

Progress bar displays the status of a determinate process.
It can also be used to display a value as a percentage of its maximum value.



1d. Upgraded Third-Party Libraries

Third-party libraries have been upgraded:

jQuery write less, do more.	jQuery Core 1.9.1
jQuery user interface	jQuery UI 1.10.3
jQuery mobile	jQuery Mobile 1.4.2
 12/31/2012 06:00 AM	jQuery Timepicker Plugin 1.4.3
CKEditor™	CKEditor 4.2.1
MOBISCROLL The customizable date and time picker for touch devices.	Mobiscroll 2.9.5

1e. WAM Editor Enhancements

The WAM Editor now inspects the design of a web routine for use of deprecated weblets and lets the user know if they are using deprecated weblets. This test is done when a web routine's design is opened in the WAM Editor.

Other enhancements:

- New location for scripts and styles:
  - Styles: Header and body-top.
  - Scripts: Header, body-top, body-bottom and asynchronously.
- Improved Undo.

1f. Usability and Performance

- jQuery Mobile weblets were designed to have their contents fully customizable.
- Downside is that they are too hard to design with the WAM Editor.
- Ship new simpler versions for commonly used weblets (anchor, button, list) that most users use without custom content.

2. Windows 64 bits Support

Visual LANSA now supports the generation and compilation of both Windows 32-bit and 64-bit LANSA applications.

- A 64-bit Visual LANSA runtime is provided in addition to the 32-bit runtime.
- 64-bit deployments are supported in addition to 32-bit deployments.

When Should Windows 64-bit Support be Enabled?

We recommended that Windows 64-bit support is only enabled when there is corporate requirement for it.

Windows 64-bit support should only be installed on a Build machine, not developer machines.

Drawbacks

Using Windows 64-bit support has some drawbacks:

- You must obtain your own 64-bit compiler, either Visual Studio 2010 Professional (or later) or Visual Studio 2012 Express for Desktop (or later).
- Compile times are longer because both 32-bit and 64-bit DLLs are always built.
- Functions which use DISPLAY, REQUEST or POPUP commands will fail to compile also in 32-bit DLLs.

Features not Supported

There are LANSA features that do not function or are not supported in 64-bit applications:

- Graphics Server
- Web Functions
- ZIP and specialized LANSA Built In Functions (BIFs)
- Explorer Component AutoRefresh Property

Installation Considerations

It is presumed that 64-bit support is only enabled on the Build machine and that developers do not enable it. As a consequence, when 64-bit support is enabled both the 32-bit and 64-bit compiles are performed and both MSI packages are built.

You cannot compile a function which contains DISPLAY, REQUEST and POPUP commands - even the 32-bit compile will fail. **This is why it is better not to enable 64-bit support on Developer's machines.** If developers need to work on both RDML functions and 64-bit applications, two systems which use the same repository can be installed on their machines

Compiler Installation

If a supported compiler is not installed before LANSA, the LANSA-shipped compiler is installed and enabled. If you install a 64-bit compiler later, you need to change this registry entry to disable the LANSA-shipped compiler:

On a 64-bit PC	HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\LANSA\MicrosoftCompiler\2010\Enabled to 0.
On a 32-bit PC	HKEY_LOCAL_MACHINE\SOFTWARE\LANSA\MicrosoftCompiler\2010\Enabled.

If Visual Studio is installed before LANSA, it is used as the compiler.

If the version of Visual Studio installed does not support 64-bit compiling, install a version that does. LANSA will detect it when it is next started.

Programming Considerations

There are no increases in the maximum size of any LANSA feature because the limits are considered sufficient. This ensures greater compatibility between 32-bit and 64-bit applications.

For example:

- The maximum size of an RDMLX List is still 2 billion rows, with each entry being 2 billion bytes long.
- The Built In Functions SND_TO_DATA_QUEUE and RCV_FROM_DATA_QUEUE may be used interchangeably.
- Job Queue Emulation can use either a 32-bit or 64-bit Job Queue Monitor and jobs may be submitted from either 32-bit or 64-bit. Note that the 64-bit Job Queue Monitor will execute the submitted job as 64-bit, no matter which platform submitted the job.

PC Other Files which are loaded using a 32-bit ODBC driver will need to create a 64-bit DSN with the same name as that used to load the file, or use CONNECT_SERVER when deployed to re-direct IO to a 64-bit driver.

An ActiveX included in LANSA RDML must be a registered 32-bit version. To execute the ActiveX, a version must be registered which is of the same processor architecture as the LANSA runtime. That is, if the LANSA runtime is 64-bit then the 64-bit ActiveX must be registered on the deployed PC.

32-bit and 64-bit Applications Accessing the Same Database

These considerations are particularly important when deploying an application into a production system:

Do not create mixed 32-bit and 64-bit applications

- To avoid complexity, it is recommended that applications are either 32-bit or 64-bit. For example, if you use both 32-bit and 64-bit clients when using SuperServer, only use a 64-bit server. Because the clients are not directly accessing the database, there is no complication (Presuming your app is deployed in multiple environments).
- Our client/server protocol is platform agnostic. It's had to be since the beginning with OS/2 and AS/400 differences (little-endian/big-endian and 32-bit v. 64-bit).
- Preferably deploy just one-type of client – choose either 32 or 64.

Auto-generate relative record numbers

- Assign relative record numbers using auto-generation.
- If relative record numbers are assigned using external files, duplicates will occur unless the RPTH parameter is assigned to the same path for both 32-bit and 64-bit applications.
- A file that is currently using external files can be changed to use auto-generation using the Upgrade tool feature *Convert Files to Use Identity Column*.

Database upgraded by first system upgraded

Table upgrades are identified by comparing the previous CTD file to the new CTD file being installed. Thus only the first system upgraded should upgrade the database. This is why database upgrade defaults to off during an MSI install and why per-user installs disable database upgrade.

Be consistent

- If an existing OAM is not there for 64-bit but is for 32-bit, and vice versa, the user needs to control which is the latest OAM.
- If 32-bit is the first environment to be installed, continue that way for all Upgrades and Patches. Once the 64-bit environment is at the same level, the Upgrade/Patch database change machine can be switched, but it is inadvisable.
- When you need to change to a different machine as the one to upgrade the database ensure the alarm bells go off! Thoroughly check that the machine's environment exactly matches the database.
- Be consistent and use one machine from the beginning.

Notable Environmental Differences

- The system directory for 32-bit applications is of the form x_win95\x_lansa. For 64-bit applications it is x_win64\x_lansa. Therefore system variables like *SYS_DIR return a different value.
- Visual LANSA is a 32-bit application. Hence interaction between Visual Lansa and 64-bit generated DLLs cannot occur.
- 32-bit OAMs are always built because Visual LANSA requires the 32-bit OAM to unload and load the data from the table. The 64-bit build command always skips the SQL table build, presuming that 32-bit has already done it.
- The Windows Installer has a known defect which converts the Target directory in a Shortcut from c:\program files to c:\program files (x86). Nonetheless, the shortcut still works correctly as if it was c:\program files. Even if the 32-bit version of the Application is installed in c:\program files (x86), it does not get executed, it is still the 64-bit version. See the MSDN forum post 32bit MSI on 64bit OS: Converting shortcut target path of 64bit app to 32 bit Path (*http://social.msdn.microsoft.com/Forums/windowsdesktop/en-US/a380e765-e062-4f84-89a3-d4072c46cbc4/32bit-msi-on-64bit-os-converting-shortcut-target-path-of-64bit-app-to-32-bit-path?forum=windowssdk*)
- A similar situation occurs with Windows\system 32. The shortcut looks OK but it does not find the object. It is not valid to create a shortcut that points to this directory.

64-bit Support – Deployment

Windows does automatic processing based on the directory you are installing into. Don’t mix the two up. The result is an inoperable application. On a 64-bit machine, never deploy a 32-bit app into Program Files or a 64-bit app into Program Files (x86).

Listener is always installed 64-bit

- On a 64-bit machine.
- Need to choose whether 32-bit or 64-bit server-side applications are used. Server configuration decides this.

Just-In-Time server may be either 32-bit or 64-bit

- Recommend 64-bit on a 64-bit machine.
- Both 32-bit and 64-bit MSIs may be served from either JIT Server.
- Client does not care. It’s just a port number.

3. IBM i User Profile Handling

IBM i user profiles can now be validated, and IBM i passwords can be changed using the IBM special APIs with interfaces to RDML and LANSA Open.

Using this functionality non-5250 users can find out when their password is going to expire, and, if the password has not expired or been disabled, to change it.

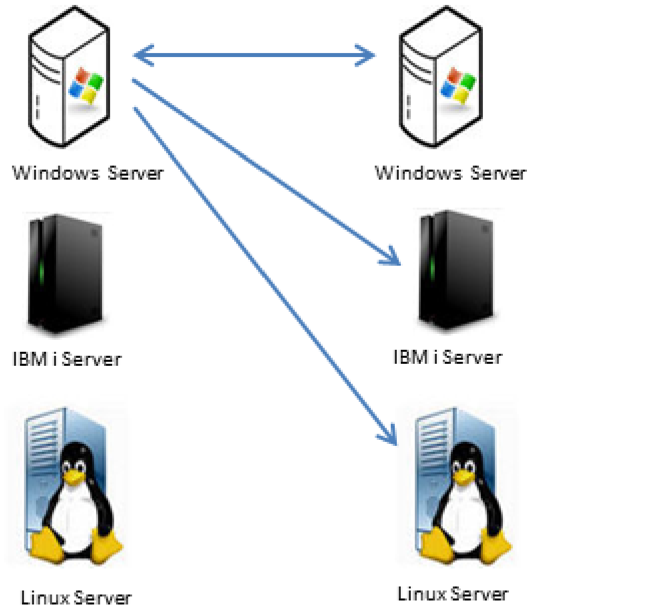
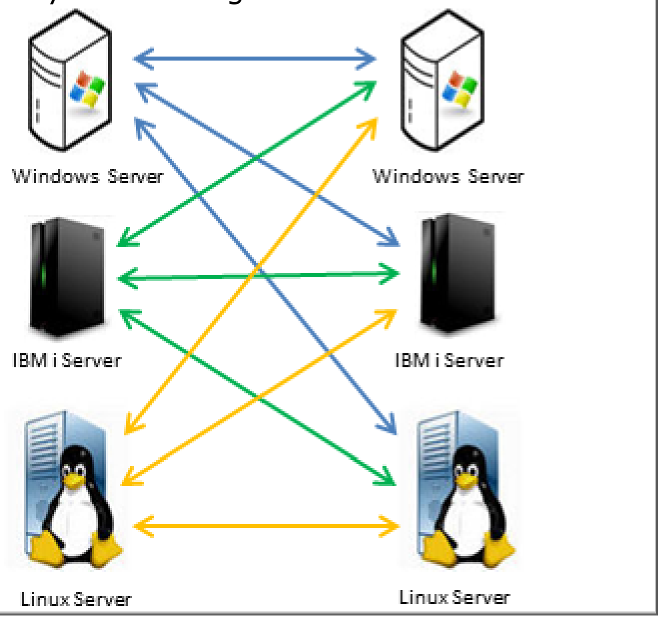
This functionality is implemented in Visual LANSA as Built-In Functions, and in LANSA Open as Lce APIs.

See:

- **CHECK_IBMI_SIGNON** and **CHANGE_IBMI_SIGNON** in the Technical Reference.
- **LceGetIBMiSignon** and **LceSetIBMiSignon** in the LANSA Open Guide.

4. SuperServer Enhancements

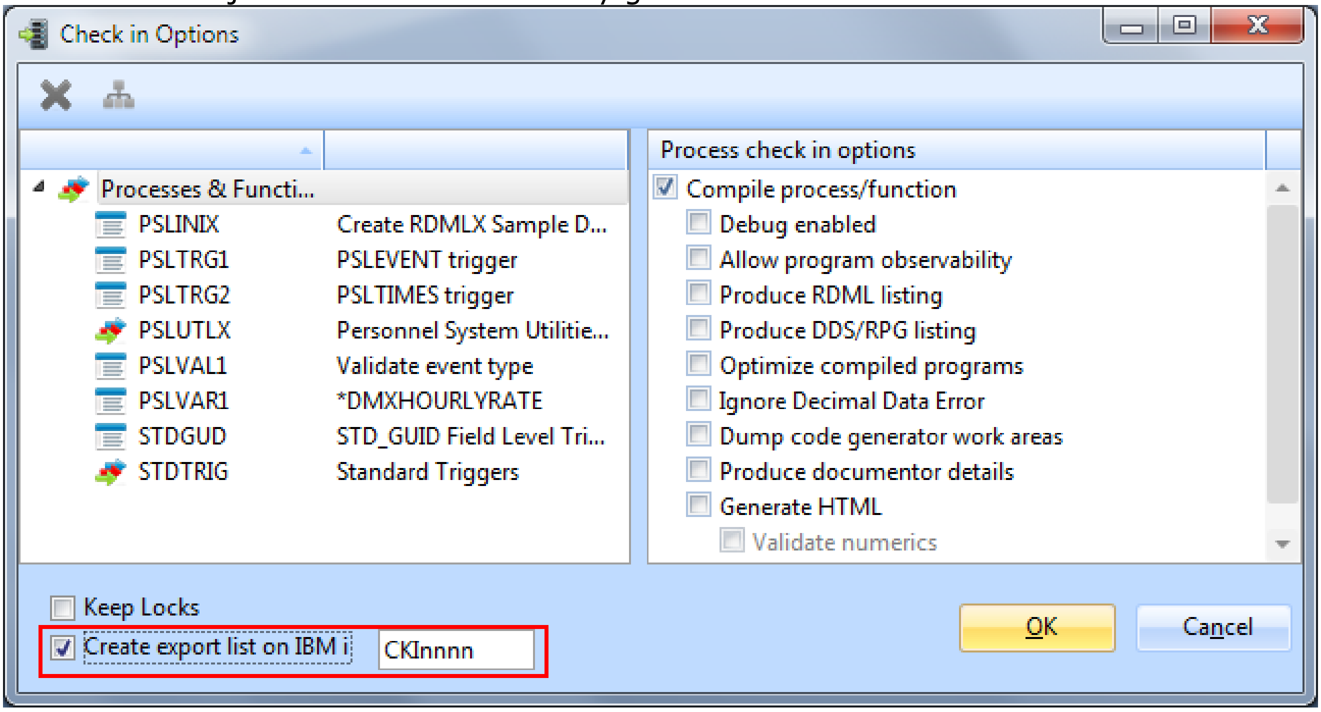
Interoperability between all LANSA platforms - Windows, IBM i & Linux

Until LANSA Version 13 Service Pack 2 the possibilities of establishing a SuperServer connection between different types of servers was limited.  Windows Server Windows Server IBM i Server IBM i Server Linux Server Linux Server	Now it is possible to establish a SuperServer connection between any two servers regardless of the operating system they are running.  Windows Server Windows Server IBM i Server IBM i Server Linux Server Linux Server

5. IDE Enhancements for IBM i Administrators

New Check-In Features

When objects are checked in or delivered to an IBM i system, an export list containing the selected objects can be automatically generated on the IBM i:



Check in Options

Processes & Functi...	
PSLINIX	Create RDMLX Sample D...
PSLTRG1	PSLEVENT trigger
PSLTRG2	PSLTIMES trigger
PSLUTLX	Personnel System Utilitie...
PSLVAL1	Validate event type
PSLVAR1	*DMXHOURLYRATE
STDGUD	STD_GUID Field Level Tri...
STDTRIG	Standard Triggers

Process check in options

- ☑ Compile process/function
  - ☐ Debug enabled
  - ☐ Allow program observability
  - ☐ Produce RDML listing
  - ☐ Produce DDS/RPG listing
  - ☐ Optimize compiled programs
  - ☐ Ignore Decimal Data Error
  - ☐ Dump code generator work areas
  - ☐ Produce documentor details
  - ☐ Generate HTML
    - ☐ Validate numerics

☐ Keep Locks

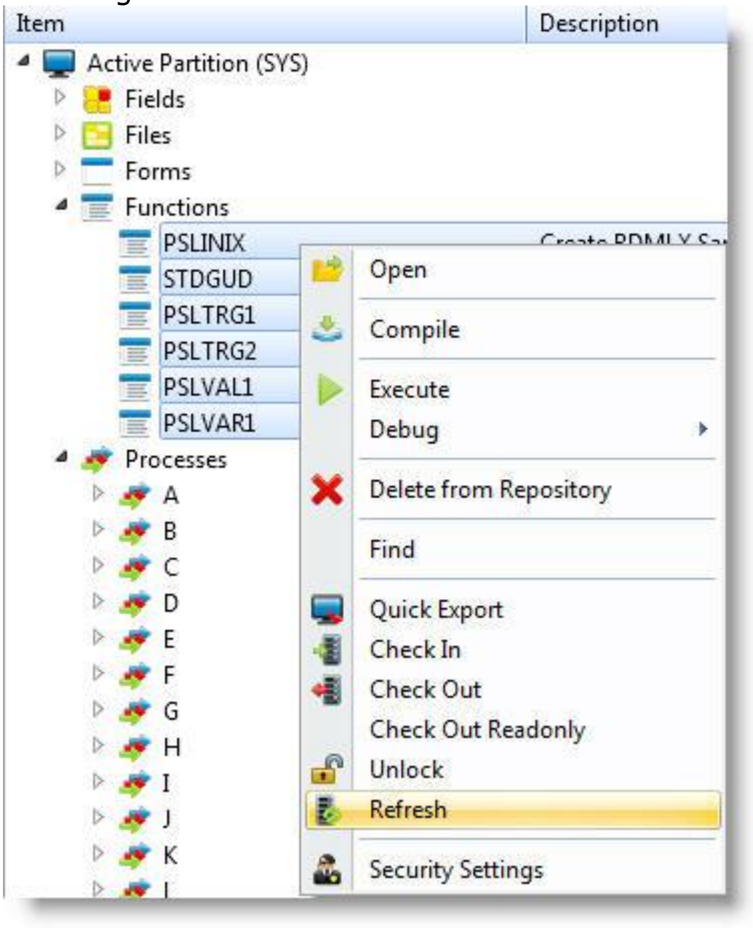
☑ Create export list on IBM i: CKInnnn

You can change the default name of the export list. If you specify the name of an existing export list, the new list will be appended to it.

For more information see Check In Options and Deliver To Options in the Visual LANSA Administrator's Guide.

Refresh Selected Objects

Instead of doing a full refresh of all the objects in the repository, you can refresh the details of selected objects which are already in the local repository by right-clicking and selecting Refresh:



Object data is downloaded from the Master in batches and then trickled into the local Repository to avoid impacting performance on the local machine. You can use F5 at any time to force the objects to be refreshed immediately.

For more information see Refresh Master Object List in the Visual LANSA Administrator's Guide.