Visual LANSA User Guide

- Getting Started with Visual LANSA
- Editing Objects
- Standard Editor Tabs
- Editor Features
- Executing Applications
- Debugging Applications
- Tracing Applications
- Editor Basics
- Product Information

Edition Date May 2, 2014

© 2014 LANSA

1. Getting Started with Visual LANSA

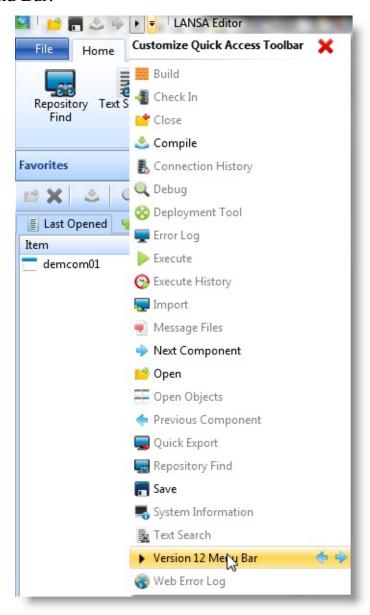
- 1.1 Finding Version 12 Commands in Version 13 IDE
- 1.2 How Do I Get Started?
- 1.3 Before You Begin Checklist
- 1.4 How Do I Populate the Repository?
- 1.5 Using LANSA Other Products
- 1.6 Setting up Your Workspace
- 1.6.1 Editor Options

1.1 Finding Version 12 Commands in Version 13 IDE

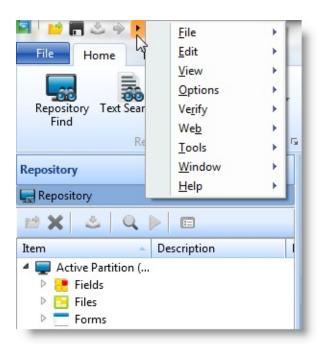
The Version 13 IDE has been redesigned to use a ribbon interface. We highly recommend you familiarize yourself with the ribbon to take advantage of the usability benefits it offers.

However, if you have problems finding a command, you can use the Version 12 menu options by adding them to the Quick Access Toolbar.

To do this use the Customize Quick Access Toolbar drop down and select Version 12 Menu Bar:



After this the Version 12 menus are accessible from the Quick Access Toolbar:



1.2 How Do I Get Started?

If you are a new LANSA developer, an excellent starting point is the Visual LANSA Tutorials and then the more advanced Visual LANSA Windows Applications Tutorials.

The LANSA Tutorials provide training for a variety of development topics. You will find training for all LANSA products in the online tutorials which are normally within the online Guide for the product.

Your first step in using Visual LANSA should be reviewing the 1.3 Before You Begin Checklist. This checklist identifies some of the items that you need to know about your Visual LANSA System.

Once you log on to Visual LANSA, you will be able to review the LANSA Repository. To learn about the LANSA repository, review the 1.4 How Do I Populate the Repository?

If you are an existing LANSA User, you may wish to review the list of new features in What's New?

Also See

VUI001 - Starting Visual LANSA in the Visual LANSA Fundamentals Tutorials Getting Started with LANSA Development in the *Developer Guide*

↑ 1. Getting Started with Visual LANSA

1.3 Before You Begin Checklist

Before you start using the Visual LANSA development environment, review the following checklist:

- Do you know the Visual LANSA System Type of your installation? You may
 be using a Visual LANSA Slave System with a LANSA for iSeries Master
 System, or you may be using an Independent Visual LANSA System. The
 system type influences the options available in the Visual LANSA Editor.
 For further information, in the *Administrator Guide*, refer to
 - Planning and Installing LANSA
 - Visual LANSA Independent System Administration
 - Visual LANSA Slave System Administration overviews.
- Are you using Visual LANSA with a LANSA for iSeries Master System? If you are, you should become familiar with Host Monitor Concepts as described in the *Administrator Guide*. In particular, the Example Development Cycle provides a good overview of starting development with Visual LANSA.
- Do you have a Visual LANSA user profile and password in order to logon? (Refer to Logon Parameters in the *Administrator Guide*.)
- Do you know the partition that you will be using when you logon?
- Do you know the Change Management strategy being used with Visual LANSA?
- Do you have a task ID assigned for completing your work in Visual LANSA? A task ID is required at logon.
- Do you know if Repository Synchronization is being used so that the Host Monitor should be started for Propagation?
- If you are planning to use the Logical Modeler or the Database Diagram Viewer, you will need to have Microsoft Visio 2003 (or higher) installed.
- You should also be familiar with the Visual LANSA interface 1.6 Setting up Your Workspace.

The *Administrator Guide* provides details to assist in the configuration of the Visual LANSA environment.

Also See

1.2 How Do I Get Started?

- 1.4 How Do I Populate the Repository?

1.4 How Do I Populate the Repository?

The contents of the Visual LANSA Repository are determined by the System Initialization and Partition Initialization. These actions are performed during the software installation and again each time you log on to Visual LANSA. For further information, refer to Logon Parameters in the *Administrator Guide*.

System initialization will import LANSA systems definition data into the repository along with user, security and other settings. Partition Initialization can import demonstration objects and objects used by LANSA products such as the Visual LANSA Framework. Other than these initialization objects, your Visual LANSA repository may have very few objects in it. As you begin development, you will add your own objects to the repository.

A Visual LANSA Slave repository can be populated from the LANSA for iSeries Master System. Objects may be exported or checked out from the master. The objects may be copied as read only (i.e. the slave systems cannot change the object definition) or the objects may be checked out for update.

An Independent Visual LANSA System can import application objects from another LANSA system. Also, the LANSA Deployment Tool can be used move objects between LANSA Systems.

Also See

Getting Started with LANSA Development in the *Developer Guide* Creating Application Databases in the *Developer Guide*

1. Getting Started with Visual LANSA

1.5 Using LANSA Other Products

LANSA is a family of application development tools. In addition to the LANSA for iSeries and Visual LANSA development environments, you can use other products including LANSA Client and LANSA Integrator to complete your projects.

1. Getting Started with Visual LANSA

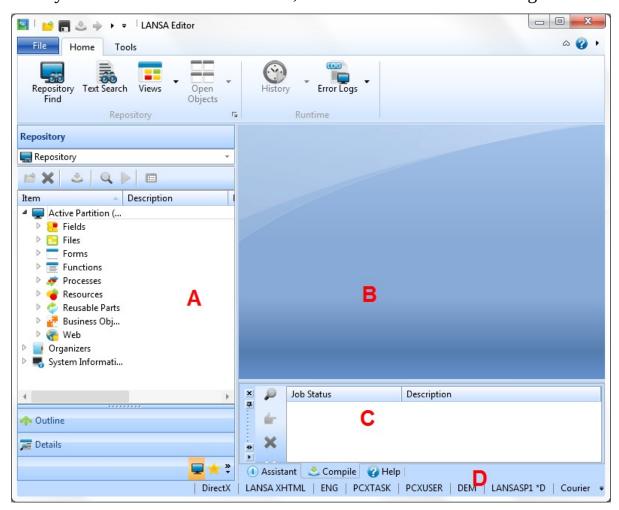
1.6 Setting up Your Workspace

Your work space in the LANSA Editor is very flexible. You can control your work space to suit what you are doing.

As well as setting up your work space, you can choose to change specific behaviors via the LANSA Settings. At the beginning, the selected defaults will probably suit you best but it is recommended that you become familiar with the time-saving options that are available.

If you have made some changes to the Editor's work space and wish to return to your original layout, press the Reset Editor button on the LANSA Setting's dialog.

When you first install Visual LANSA, the Editor will look something like this:

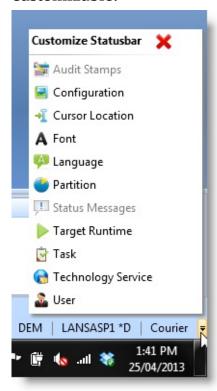


The Editor's window, by default, contains these main areas:

A - Left pane. This is a tab area and is, generally, used to locate information and

objects. Also in this area is the Details tab that you use to work with objects in the LANSA Repository.

- B Center pane which is the main work area and displays the currently opened object. You cannot close this center pane, change the location of its tabs or change its location.
- C Bottom pane. This is a supplementary work area, which is used for reporting the status of activities, such as compiles, as well as displaying Help text and the Prompter.
- D Status bar. The status bar shows active settings for the IDE. These settings are specified in the LANSA Settings dialog, but you can change some of them directly by clicking them in the status bar. The contents of the status bar are customizable:



There are a number of ways to rearrange, resize and close these areas and these are described in 1.6.2 Editor Tabs,1.6.3 Quickly Resize Tabs and 1.6.4 Autohide.

For further information about the use and default contents of the editor, refer to Editor Basics and Editor Tabs in this guide.

Also See

Editor Features

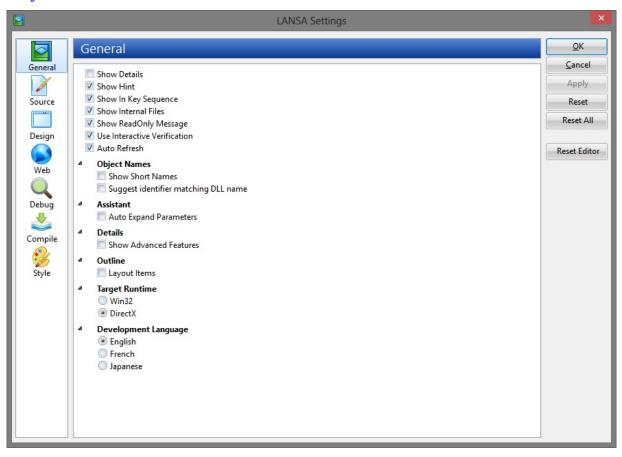
↑ 1. Getting Started with Visual LANSA

1.6.1 Editor Options

Setting the Editor Options General Design Source

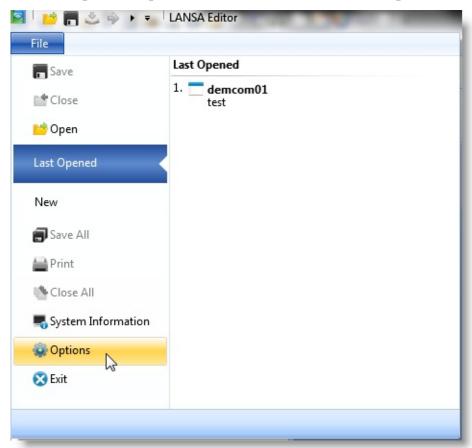
WAM - General tab WAM - Design tab Debug Compile

Style Reset Defaults



Setting the Editor Options

To set the Editor Options, open the File menu and choose Options:



The LANSA Settings dialog is opened. Use the options in this dialog to control the behavior of the editor.

General

Show Details	When selected, shows details about objects in the Outline tab and wherever else is applicable.
Show Hint	Shows information about components in a hint.
Show in Key Sequence	Shows fields in a file in key order. Fields are otherwise shown in sequence order.
Show Internal Files	Shows or hides internal files in the editor.
Show Read Only Message	Shows or hides a message about read-only objects.
Use Interactive Verification	Sets interactive verification to on or off. When objects are being edited in the LANSA Editor, a verification process is run to provide feedback to you about the state of the object being edited. For source objects, such as functions and components, verification occurs as the source changes, while for non-source objects such as fields and files, verifications are run every 1.5 seconds. This verification process produces the error/warning triangles and messages seen throughout the Editor.
Save Master Only Objects	Available on Slave systems only. Saves the master only state so it can be displayed in a new editor session. It is recommended that you run the Refresh Master Object List regularly to ensure the master-only state information is up to date.
AutoRefresh	Automatically refresh the list of repository objects.

Object Names

Short Names	to as identifier).
Suggest identifier matching DLL name	Identifier for it based on the name of the object. The identifier

Assistant

Auto Expand	Automatically expands all parameters in the	
Parameters	Command Assistant.	

Details

Show Advanced Features	Displays advanced features in the Editor.
-------------------------------	---

Outline

Layout Items	Displays Layout items in the Outline view. When using layout managers to manage the layout of a component, layout items are created to attach to the member
	components. Refer to Layout Managers in the Developer Guide.

Target Runtime

Win32	Sets the target runtime to Win32. Trying to use features specific to DirectX will cause warnings.

DirectX.
\Box

Object Locking

Off	In single user environments, object locking can be turned off.	
On	When object locking is on, locks are put on objects so that multiple users cannot edit them simultaneously.	
On (ReadOnly)	Setting object locking to Read only sets all objects in the partition to READ ONLY status.	
	Refer to Object Locking in the Administrator Guide.	

Development Language

language	Sets the language of the LANSA development environment. If multiple languages have been installed, you can use this option to change the language.
	Note that the language for the partition is changed when you log on the next time.

↑ 1.6.1 Editor Options

Source

Indentation	Indents commands in the Source tab.
Indentation Lines	Indentation lines assist you in navigating the source code in the Source tab
Line Numbers	Turns line numbering on and off in the Source tab.
Component Definitions	Displays DEFINE_COM statements in RDML source code.
Auto Display Messages	Displays all error and warning messages expanded in the Source tab.
Word	None - Turns word wrapping off.
Wrap	Simple - Line will break at any space.
	Smart - Line wrapping keeps parameters on the same line.
Auto Complete	Off: If you choose this option, you will get no prompting in your Source code window to assist you in completing your code. The Assistant tab is still available.
	Prompter : This is the most powerful of the options available to help you create our LANSA code. If you choose this option, a small dialog is open in the Source tab. This dialog offers items (commands, parameters) that appear to be close to those you are using.
	Inline: If you choose this option, the Auto Complete facilities offered are the opportunity to select the most likely option in your Source code. This option is shown as part of your current line, rather than as a separate dialog. Automatically completes text in the Source tab. Refer to Auto Complete.
	ShortCut: The keyboard shortcut for displaying Auto Complete can be either Ctrl + J or Ctrl + Space.
Audit	Name – displays the user name audit stamps in the Source

Stamps	tab.
	Date – displays audit stamps dates in the Source tab.
	Task Id – displays the audit stamps Task Id in the Source tab.
	Refer to Task Maintenance in the Administrator Guide.

Function Formatting

Commands	Format commands in:
	- Uppercase - Lowercase
	 Proper case – initial uppercasing, for example Begin_Com
Keywords	Format keywords using:
	- Uppercase - Lowercase - Proper case

Component Formatting

Automatically reformat statements	Use this option to automatically reformat statements.
Commands	Format commands in: - Uppercase - Lowercase
	- Proper case – initial uppercasing, for example Begin_Com.
Keywords	Format keywords using: - Uppercase
	- Lowercase - Proper case
Selectors	Format selectors using:

	- Uppercase - Lowercase - Proper case
Variables	Format variables:
	- As defined – exactly as in their definition.
	- Ignore – do not reformat variables.

\uparrow 1.6.1 Editor Options

Design

Design Displays design data inside controls. Design data helps you visualize how controls will look in the application.
 Refer to Design Tab.

Snap to Aligns components to the Design Tab grid (functions are always aligned to grid).

Show Shows a grid in the Design Tab which is used to align fields and components.

↑ 1.6.1 Editor Options

WAM - General tab

Technology Service	The Technology Service is a presentation or XML format used by a WAM to interact with a user agent or other XML-aware application.
-----------------------	--

Web Settings

Base URL	The Base URL is used during development time for all relative paths for images, CCS and JavaScript.
	Use the Test button to verify that the web server specified by the Base URL is accessible. If an http scheme is used, then the test determines if a successful connection can be made, ie, the web server returns HTTP status code 200. If a file scheme I used, then the test checks if the specific folder exists.
	Default value: http://localhost/
Application Base URL	The Application Base URL is used to launch a WebRoutine from the LANSA Editor.
	The CGI-BIN part of the URL is configurable using the Web Administrator.
	Use the Test button to verify that the URL is accessible. The test makes an attempt to load the LANSA for the web About page using the URL specified.
	Default value: http://localhost/CGI-BIN/lansaweb

Options

Number of	When using a LANSA list in a Web Page design, this setting
-----------	--

sample list rows	determines how many samples rows are shown. Minimum value is 1 and maximum is 5. Default value: 3
Number of Web design versions	This setting determines how many versions of the web designs are kept as history. Default value: 2

↑ 1.6.1 Editor Options

WAM - Design tab

Images virtual directory

This is the virtual directory for the location of images, CCS and JavaScript files used by this LANSA configuration. This setting is read-only and can only be changed using the Web Administrator.

Default value: /images

Mapped images folder

This setting is used in conjunction with the "Images virtual directory" when selecting an image using the image browser.

This is the physical folder that the "Images virtual directory" resolves to. If you are using IIS as Web Server and the LANSA for the Web Plug-in is running in Development mode it will resolve the "Images virtual directory" for you. Otherwise it is assumed that the images folder is located within the current configuration.

Default value: The value is worked out as per above.

Auto synchronize picklists for visualized fields

Some LANSA fields can be visualized with picklists. The picklist data is stored in the WebRoutine. Changes to the picklist data in the LANSA field definition will not automatically appear in a WebRoutine where the field is used. This happens when a WebRoutine is opened in the Editor, and only if this option is selected.

Default value: Enabled

Open Web Views when designing XSL

If this option is enabled, all Web related views will be opened when a Web design is loaded. This option is useful to get started with web development. Once they have been opened the Editor will remember them and their location and this option can be turned off.

Default value: Enabled

Delete
Orphan
WebRoutine
Designs when
closing Web
Application

If you delete a WebRoutine from a WAM's RDML source code you end up with an orphan Web Design. This option determines what happens with those orphan Web Designs when the WAM is closed. If this option is enabled they will be deleted, otherwise they will remain in the repository.

Modules

Default value: Enabled

Show deprecated weblets

Some LANSA supplied weblets have been deprecated and newer versions are available. We don't recommend that you use the deprecated weblets in new designs, but rather use the newer versions. Therefore, by default, the deprecated weblets are not included in the repository listing. Enable this option if

you need to view the deprecated weblets.

Default value: Disabled

Show Weblet XML

Typically, when editing a weblet, you won't need to change the weblet's XML. The XSL is therefore not shown. Enable this option if you need to change the XML.

Default value: Disabled

Save WebRoutine Field and List sample data

Enable this option if you need to save field and list sample data. This is useful if you have altered the sample data to use your own rather then the default from LANSA.

Note that saving field and list sample data can affect performance when opening Web Page designs.

Default value: Disabled

↑ 1.6.1 Editor Options

Debug

Computer Name

Name of the debugging computer used for a Visual LANSA debug session. (The debugging computer is where you will debug from if the Visual LANSA application to be debugged is running on a different computer.)

If you are debugging an application on the same computer, you do not need to change this name.

Note: This name is used as the TCP/IP network host name so you need to make sure that the computer, where the LANSA applications to be debugged is running, is able to resolve the name into a valid IP address. If this computer cannot resolve this name into a valid IP address, then you will need to enter the numeric IP address here. If you are debugging an application on the same computer, you can safely use this numeric IP address: **127.0.0.1**.

Default value: Name of your current computer.

Port

TCP/IP port number to be used by the Visual LANSA Debug Service.

The Visual LANSA development environment can choose an available port number between 51234 and 52233 (inclusive) automatically and so you don't usually need to change it here.

Note that if you have multiple Visual LANSA development environments running on the same computer, the port number for each of the Visual LANSA development environment must be unique. If you do not specify your own port number, Visual LANSA will resolve it correctly.

If you are in the very unusual situation that the whole range of port numbers are occupied, you need to select and enter a port number (recommended between 10000 and 65535 inclusive) here.

If the port number selected (either automatically by Visual LANSA or one specified here) is blocked by any kind of network firewall (e.g. ZoneAlarm), you need to either choose another port number or unblock the port by changing your network firewall's settings.

Note: Do not to confuse this port number with the port number for the Listener service. The port number here must not be in use by any Listener service running on the debugger computer.

Default value: 51234

Start the Visual LANSA Debug Service automatically when the Visual LANSA development environment starts

Visual LANSA development environment starts.

service If you do not want the Visual LANSA Debug Service to be started

with automatically, de-select this option.

LANSA Default value: Enabled **Editor**

Restart Press this button to start the Visual LANSA Debug Service manually if it is not started automatically or has stopped

abnormally.

Break at Select (ü) this option if you wish the debugger to stop at your first executable statement. If it is not selected, processing will continue until your first breakpoint is reached

executable until your first breakpoint is reached. **statement**

Default value: Selected.

Compile

The information specified in the Compile tab is generated into the DLL when an object is compiled. This information will be visible in the file properties of the DLL.

Product The name of the application.

Name

Product The version of the application.

Version

File The version of the file.

Version

Copyright Copyright notice.

Trade Trademark.

Mark

Comment A comment about the version. Note that this information is not

shown in the file properties in Windows Explorer.

Style

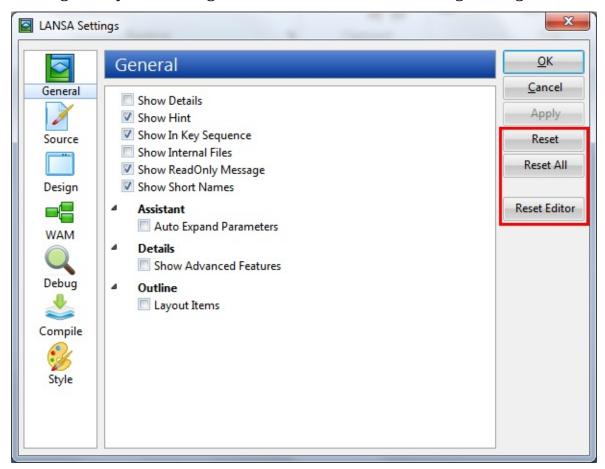
Choose the theme for the IDE using one of the styles provided.

You can also set the font used in the source view of the editor and the text and background colors for commands at different indent levels.

Also see 1.6.5 Font Options.

Reset Defaults

If you've got tangled up in changes you've made to the Settings and/or the Editor, it is sometimes quicker to reset them and then re-select some of the options you prefer, rather than to try to return the Setting or Editor to normal. Resetting is easy to do using the Reset buttons on the Settings dialog.



Reset Resets the options for the section in which you are working.

Reset All Resets all the settings to the system defaults.

Reset Editor Reset the Editor's window to the default layout.

Press OK to accept the reset.

1.6.2 Editor Tabs

To make the most of the tabs in the editor, review:

Tab Style

Docking and Undocking Tabs

Button Options

Add or Remove Favorites

Can't find a tab?

Change Tab Locations

Tab Style

The default style of tabs in the left pane is the **Navigator** bar style. You can choose to use the smaller (Standard) tab style by right clicking on the title bar of the active tab in the left pane and selecting the Tab View Style then **Standard** from the context menu.



Docking and Undocking Tabs

You can move tabs from the left (the default) to the right of the LANSA Editor's window, or you can undock them so that they float over the top of the open window.

To undock an individual tab:

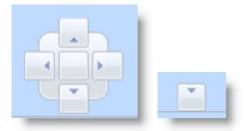
- If you are using the Navigator tab style, double click on the title bar to undock a tab, drag it to wherever you want it to go. If you move it near to the edge of the window, it will automatically dock there.
- If you are using the Standard tab style, grab the docking bar with your mouse and drag it away from it's current position.



While a tab is undocked, it can be displayed as a completely separate window floating over top of the editor. You can resize this window to display more information and you can close it and then re-display it.

To redock a tab to the main Editor window, double click on its Title bar to return it where it came from.

To dock it in another position, drag it to the left, right or bottom of the main window and use the docking guide diamond or the guide arrows to position the tab.



Button Options

If you are using the Navigator style of tab, then at the bottom right of the last tab shown in the left pane, is this button:

If you click on it you will see these options:

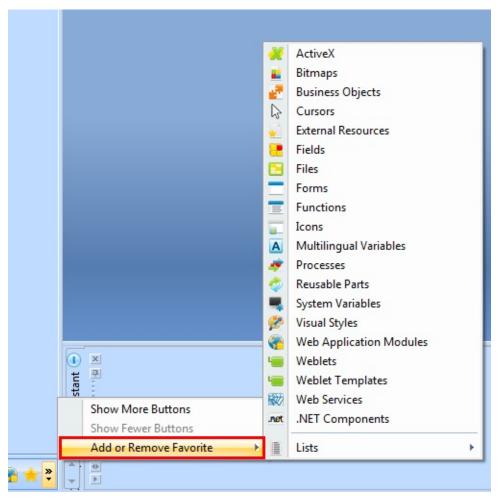


If you select Show Fewer Buttons, the Navigator bars (depending on space) are collapsed one by one to icons as shown here:



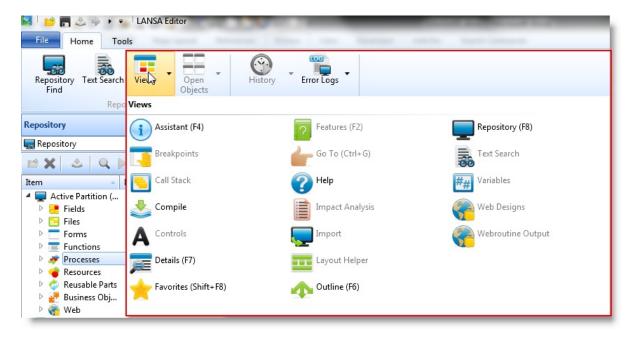
Add or Remove Favorites

If you select the Add or Remove Favorite option, a sub menu is displayed which you can use to select an object type to be added as a tab in the Favorites tab. You can also use the menu to remove the tab within the Favorites tab.



Can't find a tab?

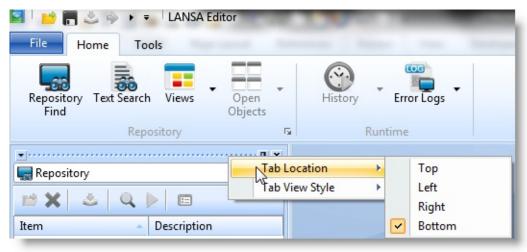
If you have moved your tabs around and can't see the tab you want, the easiest way to display it is by selecting it from the Views drop-down on the ribbon.



If you wish to return to the shipped default settings, select the Reset Editor button on the LANSA Settings dialog. This is described in Reset Defaults.

Change Tab Locations

When using the Standard tabs, you can specify the position of the tab labels by right-clicking on a tab to open a pop-up menu. Select the position that you want and the tab is immediately displayed in the new position.



↑ 1.6 Setting up Your Workspace

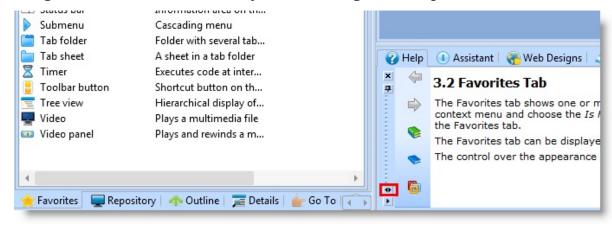
1.6.3 Quickly Resize Tabs

If you are using Standard tabs, at one end of the docking bar you may find some small arrows (as shown in the following example). You can use them to enlarge the tab in which you are working.

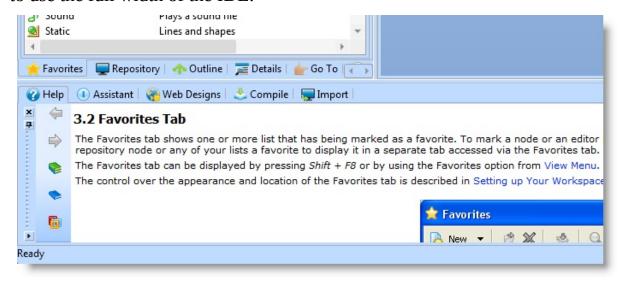


You may find this feature useful when you are working with Weblets.

Using the double arrow button, you can change the Help tab's size from this:



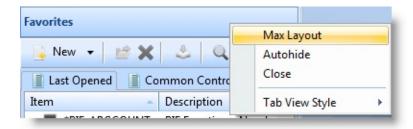
to use the full width of the IDE:



You don't need to return the tab to its original size, simply choose to enlarge another area, such as the left pane's tabs, and the original tab will automatically return to its default size.

If you're using the Navigator style tabs, you can enlarge the tab by right clicking on the title bar, then choosing Max Layout from the context menu as shown

here:



↑ 1.6 Setting up Your Workspace

1.6.4 Autohide

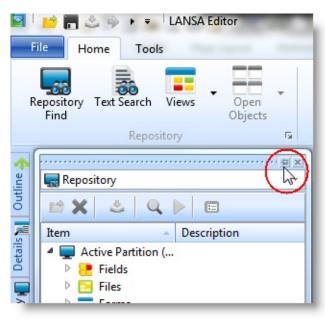
In order to obtain more room to work with the objects open in the Editor:

• for all tabs in **Navigator** tab style, right click on the tab's title bar and select Autohide from the context menu to hide the tab:



The tabs will immediately adopt the Standard tab style and will behave as follows for the Standard tab style.

• for all tabs in **Standard** tab style, you can click on the Autohide button to hide the tab:





When Autohide is set on, the tab folder is hidden but the tabs still remain visible as shown here:

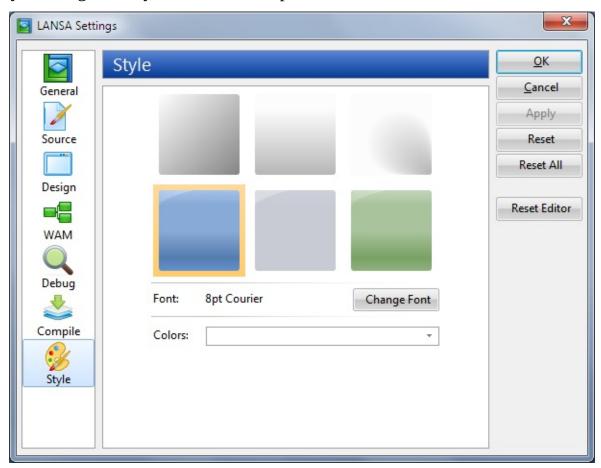
Click on the tab to re-display the contents.

To return the tab folder to its original position, display one of the tab folders and click on the Attach (*) button.

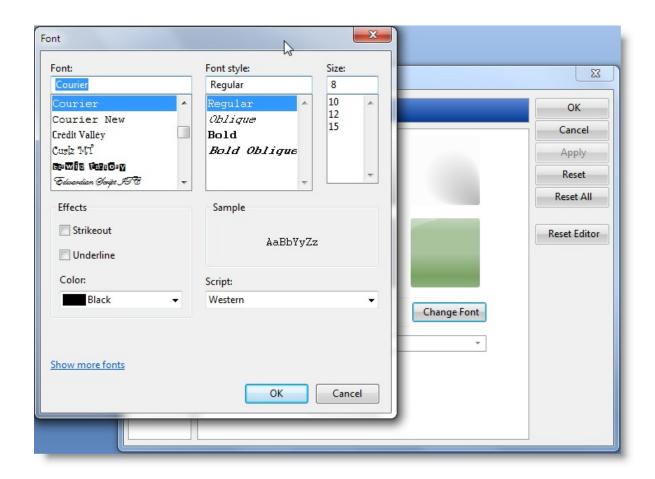
↑ 1.6 Setting up Your Workspace

1.6.5 Font Options

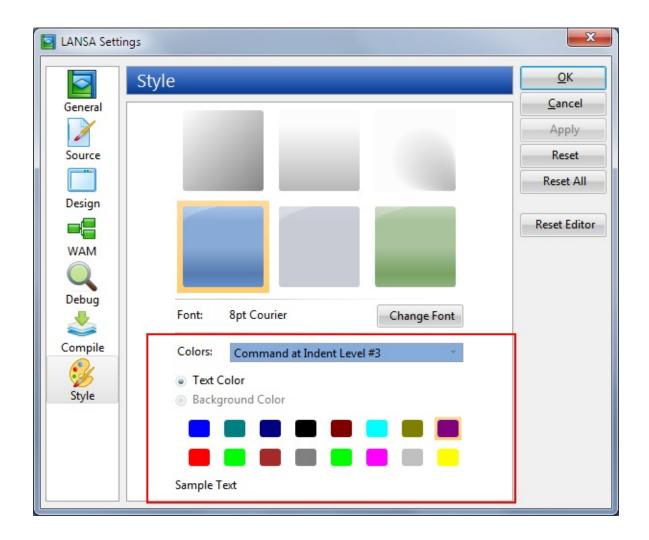
Use File -> Options or click the current font in the IDE status bar to display the Style dialog where you can set font options:



In this dialog you can change the font for the Source tab:



You can also set the colors for the different command indentation levels here:

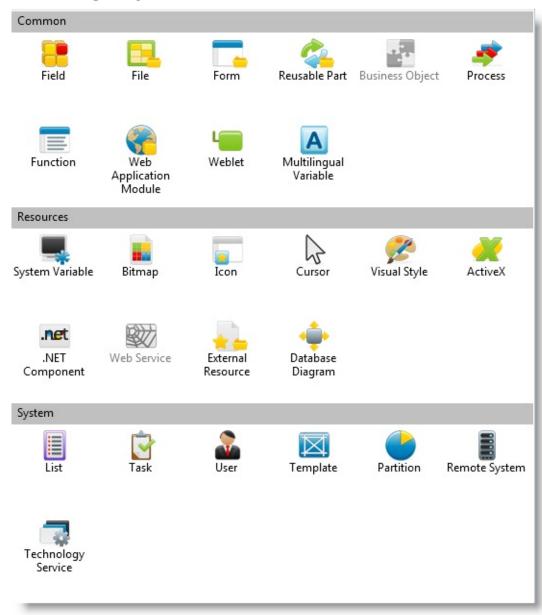


Also see Style.

↑ 1.6 Setting up Your Workspace

<

2. Editing Objects



In the Visual LANSA Editor you can create, change, compile (if applicable) and delete the following LANSA objects:

- 2.1 Fields
- 2.2 Files
- 2.3 Forms
- 2.4 Reusable Parts
- 2.5 Processes

- 2.6 Functions
- 2.7 Web Application Modules (WAMs)
- 2.8 Weblets
- 2.9 System Variables
- 2.10 Multilingual Variables
- 2.11 Bitmaps
- 2.13 Icons
- 2.12 Cursors
- 2.14 Visual Styles
- 2.15 ActiveX
- 2.16 .NET Components
- 2.17 External Resources
- 2.18 Lists
- 2.19 Database Diagrams
- 2.20 Tasks
- 2.21 User Definitions
- 2.22 Templates
- 2.23 Partition Definitions
- 2.24 Remote Systems
- 2.25 Technology Services

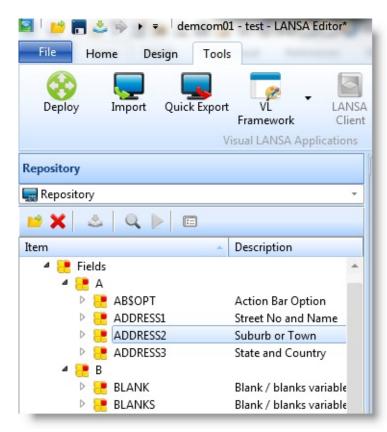
Also See

Editor Tabs

Editor Features

2.1 Fields

2.1.1 Create Fields
 2.1.2 Copy Fields
 2.1.3 Field Definition
 2.1.4 Field Rules and
 2.1.5 Field Visualization
 Tab



Fields are selected from the Repository tab or Favorites tab. Expand the tree view or double click on Fields to open the list of fields available for this partition.

Also See

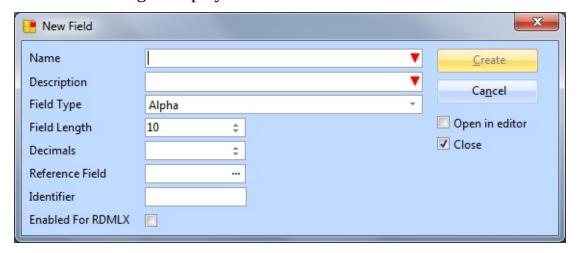
Fields in the Technical Reference Guide

↑ 2. Editing Objects

2.1.1 Create Fields

To create a field use the New command.

The New field dialog is displayed:



- Specify the Field Name, Description, and other details of the new field that is to be created. Refer to Field Definitions for details of these entries.
- Select the Open in editor option if you wish to specify additional field details.
- Select the Close option if you do not want the New Field dialog to reappear.
- When the required details have been specified, use the Create button to create the new field.

If you want the field that you are creating to permanently inherit the characteristics of another field, specify a Reference Field.

Fields must be defined in the Repository before they can be used in files, functions, forms or reusable parts.

Also See

2.1.2 Copy Fields

Fields in the Technical Reference Guide

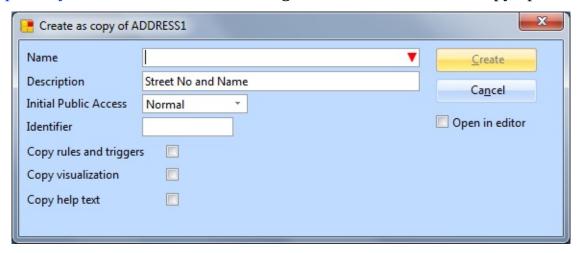
Field Definitions in the Technical Reference Guide

Developing with Fields in the Developer Guide

↑ 2.1 Fields

2.1.2 Copy Fields

To copy an existing field, select the field to be copied from the list in the Repository Tab or the Favorites Tab, right click and choose the Copy option.



The Create as copy of ... dialog allows you to specify if the rules and triggers, field visualization and help text are also copied.

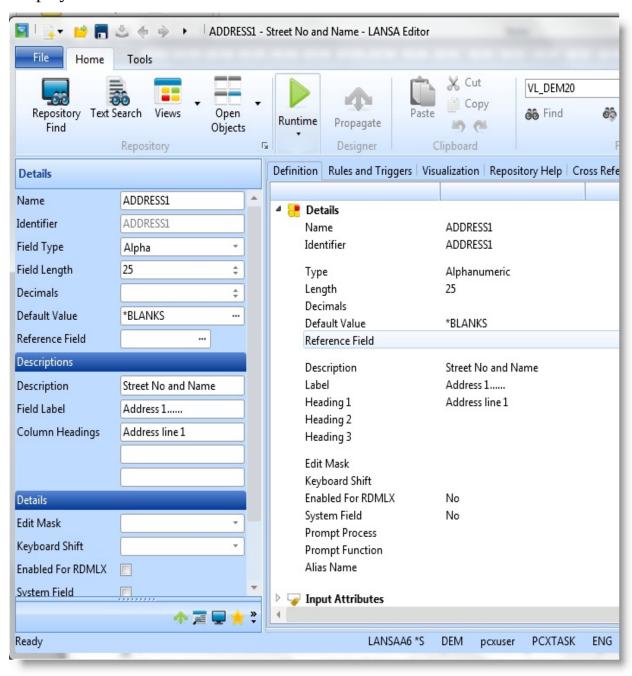
Also See

Field Rules and Triggers Tab
Field Visualization Tab
Repository Help Tab

↑ 2.1 Fields

2.1.3 Field Definition Tab

Double click on the field in the Repository to open the field's definition which is displayed in a number of tabs.



You can view the details in the various tabs, but you change them in the Details tab which you can open if you double click on an item in the tab. In addition to Details in the Definition tab there are:

Input and output attributes

To specify the field details, Input Attributes and Output Attributes for a field, double-click on the input or output

attribute list to display the relevant Details Tab.

Details

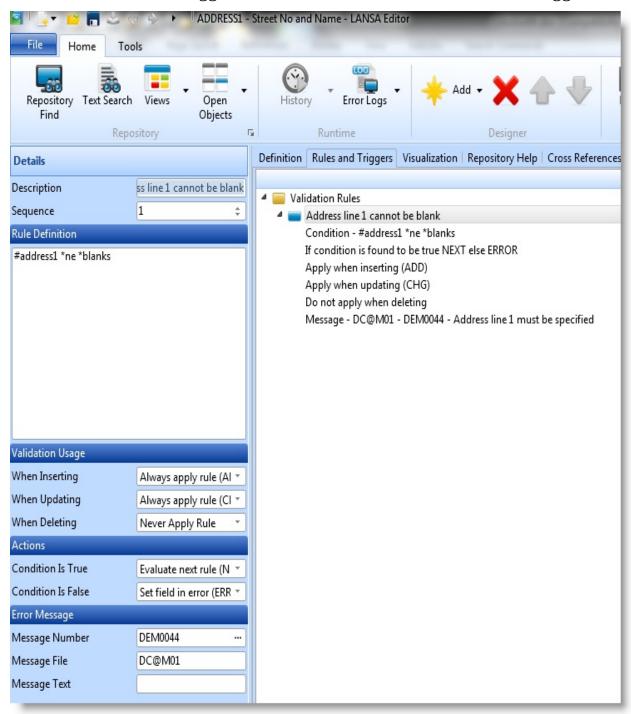
Multilingual Use the appropriate language tag to enter field descriptions for each language defined for the partition. Use the Propagate command to propagate the description, label and column headings for the focus language.

Also See

Field Definitions in the Technical Reference Guide ↑ 2.1 Fields

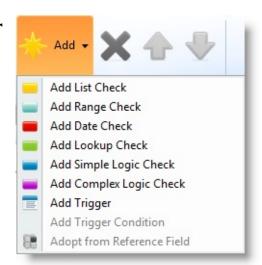
2.1.4 Field Rules and Triggers Tab

Select the Rules and Triggers tab to work with the Field Rules and Triggers.



If a reference field has been used, you can use the Adopt Field Rules and Triggers.

Add a rule or trigger



Use the Add command.

Edit a rule or trigger

Select the rule or trigger and doubleclick to display the Details tab.

Change the sequence of rules and triggers



Select the rule or trigger and use the Move Up and Move Down commands.

Remove rules and triggers



Select the rule or trigger and use the Delete command.

Also See

Rules and Triggers in the *Technical Reference Guide*Rule Definitions in the *Technical Reference Guide*Field Rules and Triggers Development in the *Developer Guide*↑ 2.1 Fields

Adopt Field Rules and Triggers

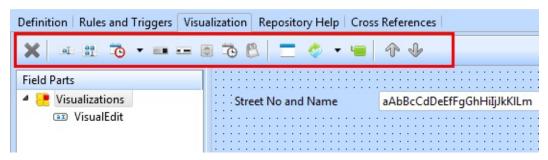
Use the Adopt rules and triggers dialog to adopt rules and triggers specified in the reference field for this field. Select the rules and triggers you want copied to the field then use the Adopt command.

↑ 2.1.4 Field Rules and Triggers Tab

2.1.5 Field Visualization Tab

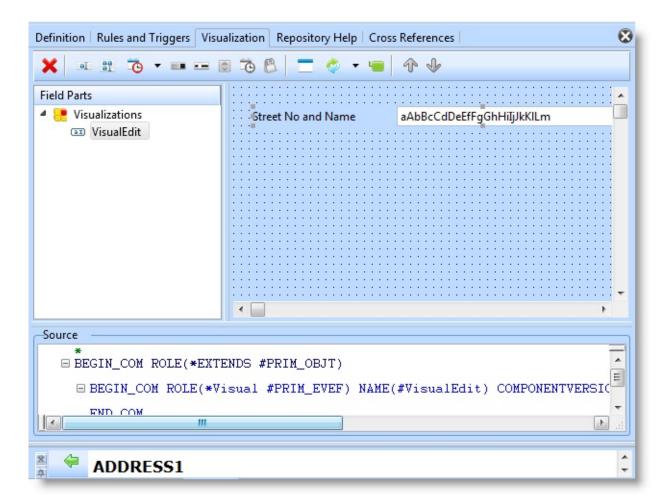
Select the Visualization tab to work with field visualizations.

Use the Visualization tab's toolbar to assign the visualization that you want.



Make the changes in the Details tab, in the Design tab or in the Source tab. To amend the properties of a field part, select the visualization in the in Field Parts list and double click on it to display the Details tab. View and amend details of the Field Parts in the Details tab, Visualization tab or Source tab. Context Help is available for visualizations by pressing F1.

The source code for the visualizations is shown at the bottom of the tab. For more information about using the Source tab refer to Source Tab.



Right click an object in the tab to open a context menu. The menu items will vary depending on the object which has the focus.

Also See

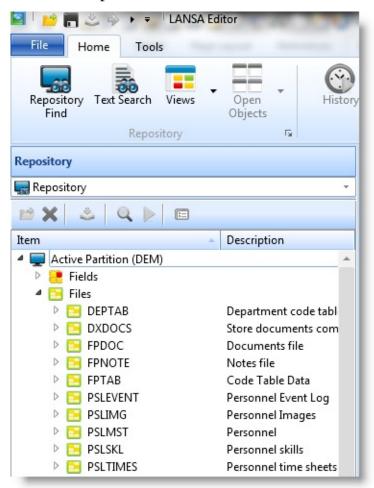
Field Visualizations in the Developer Guide

1 2.1 Fields

2.2 Files

2.2.1 Create Files	2.2.2 Copy Files	2.2.3 Fields in File Tab	2.2.4 Logical Views Tab
2.2.5 File Rules and Triggers Tab	2.2.6 Access Route Tab	2.2.7 Batch Control Tab	2.2.8 File Attributes Tab
2.2.9 Relationships Tab	2.2.10 Cross- References Tab	2.2.11 Virtual Derivation Tab	

Files are selected from the Repository tab or Favorites tab. Expand the tree view or double click on Files to open the list of files available for this partition.



Also See

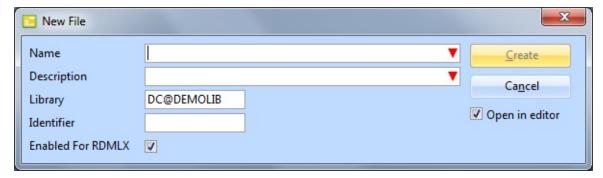
Files in the Technical Reference Guide

Developing Databases in the *Developer Guide*↑ 2. Editing Objects

2.2.1 Create Files

To create a file use the New command and choose LANSA file.

The New file dialog is displayed:



- Specify the File Name, Description, and other details of the new file that is to be created. Refer to File Definitions in the Technical Reference Guide for details of these entries.
- Select the Open in editor option if you wish to work with the file.
- When the required details have been specified, press the Create button to create the new file.

Also See

File Definition in the *Technical Reference Guide*Developing Databases in the *Developer Guide*↑ 2.2 Files

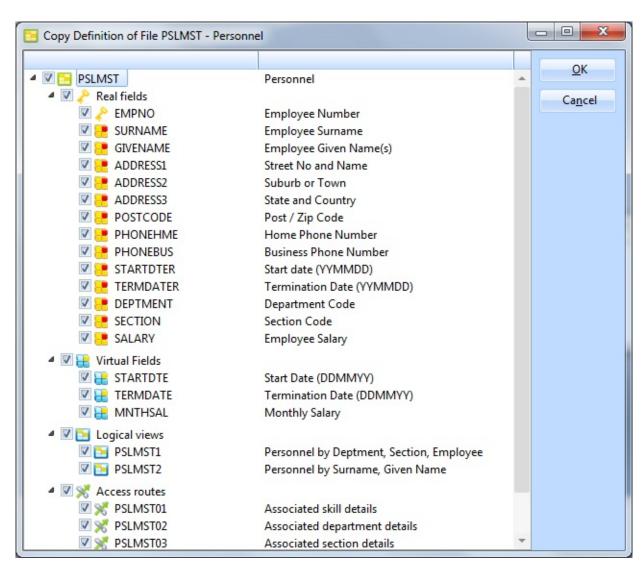
2.2.2 Copy Files

To copy an existing file, select the file in the Repository Tab or the Favorites Tab and use the Copy option from the context menu to open the Create as copy of <filename> dialog box.



Specify the Name and Description and other details of the new file.

Press the Create button, to open the Copy Definition of... dialog to select which fields and other options to be copied from the original file.



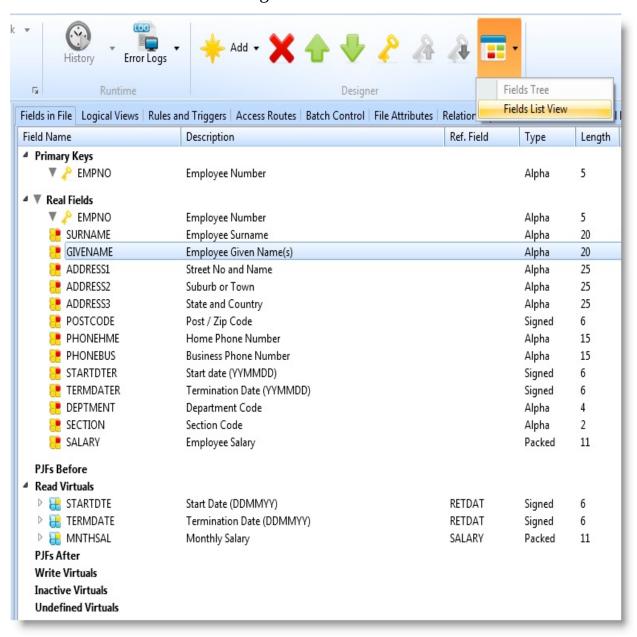
↑ 2.2 Files

2.2.3 Fields in File Tab

Select the Fields in File tab to add fields to the file and to work with the fields in the file, including virtual fields and predetermined join fields (PJFs). Use the View command to select the view in which you'd like display the fields. This can be in either Fields in File Tree View or Fields in File List View.

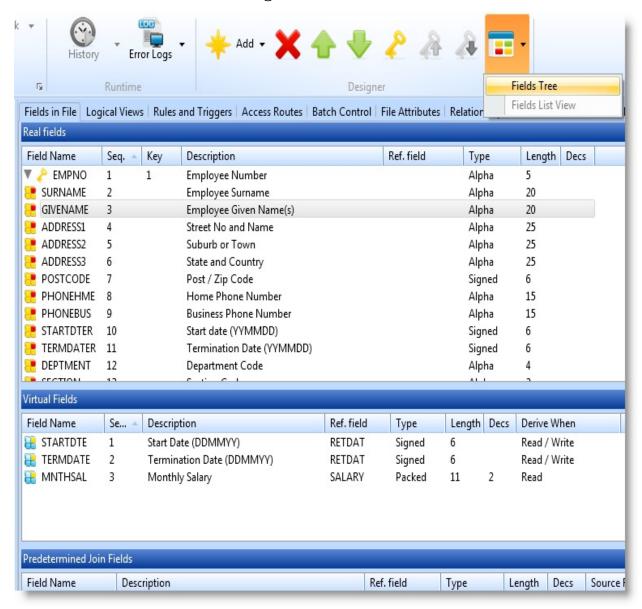
Fields in File Tree View

In this view, you can expand and collapse groupings to show or hide fields. Use the View command to change view to the sortable Fields List View.



Fields in File List View

In this view, you can sort the fields by column. You can also resize the lists. Use the View command to change views to a Fields Tree View.



Fields in File Commands

Add an existing field

Use the Add Real Field option to add a repository-defined field to the file. Refer to Real Fields in File in the *Technical Reference Guide*.

Add an existing virtual field



Add several new or existing fields or virtual fields Use the Add Virtual Field option to add new virtual field to the file. Double-click the new virtual field to display the field Details tab and to define the virtual field derivation. Refer to Virtual Fields in File in the Technical Reference Guide.

The field you add must already be defined in the Repository.

Use the Multi-Add Fields option to add new or existing fields to the file definition. The Add Fields to the File dialog is displayed. You can type in the field names or you can use the prompt command to locate the fields.

To make the field a virtual field, tick the checkbox in the Virtual column.

Click OK to add the fields to the file. If you specify new fields, you will be asked to confirm their creation.

When the fields have been added, use the Details tab to define the derivation of any virtual fields.

Delete



Remove a field, a virtual field or a predetermined join field (PJF).

Move



Change the sequence of fields, virtual fields and PJFs.

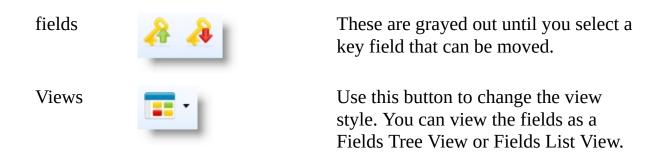
Set key field



Set a field as key field or to set it as a non-key field.

Move key

Change the sequence of key fields.



Also See

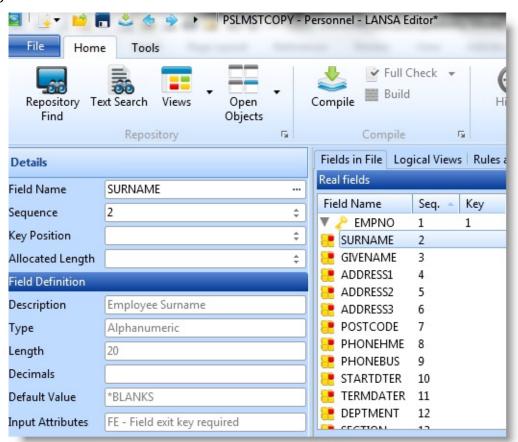
Real Fields in File in the *Technical Reference Guide*Virtual Fields in File in the *Technical Reference Guide*↑ 2.2 Files

Add Fields to a File

Add fields to the file using the Add command.



Select the Details tab to add or amend those details of the fields that can be changed in the file.



You can add existing fields to the file using the Add Real Field and Add Virtual Field commands:

- To add a field, type its name in the Field Name field or use the prompter button (with three dots) to display a Find dialog for locating existing fields.
- You can specify the key position the field will have in the file if it is to be a

key field.

Click OK to add the fields in the file.

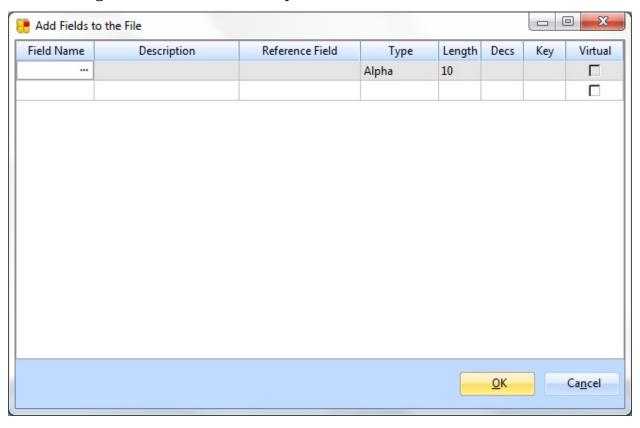
If you want to create a field to be added, use the Multi Add Fields to a File.

Also See

Real Fields in File in the *Technical Reference Guide*Virtual Fields in File in the *Technical Reference Guide*↑ 2.2.3 Fields in File Tab

Multi Add Fields to a File

This option gives you the opportunity to create a number of fields very quickly. It is best used when you only have to add the basic details and don't need to do a lot of changes to the fields once they have been created.



Field Name Enter the name of the field to be created. It can be a new field or one already created. For details of field names that can be used, refer to LANSA Object Names in the Technical Reference Guide.

Description If the field doesn't exist, enter the description here otherwise the description of an existing field, if selected, will be displayed.

Reference fieldIf you are basing your new field on a reference field, then enter the name of the reference field here.

Type If this is a new field, select the type of field (alpha, numeric, date, etc.) from the dropdown list.

Length Enter if this is a new field.

Decs Enter if this is a new field.

Key * Enter 1, 2, etc. to denote keys and their sequence.

Virtual * Tick (ü) if the field that you are entering is a virtual field.

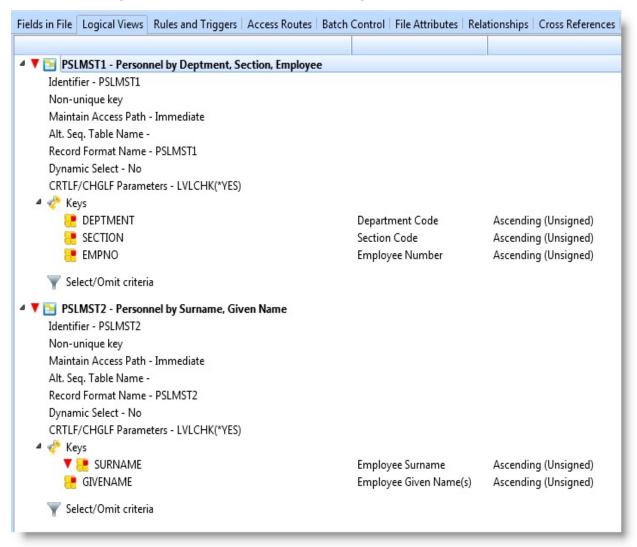
Press the OK button when you have entered your fields. If any of the fields are new, you are asked to confirm the field should be created.

↑ 2.2.3 Fields in File Tab

^{*} Do NOT forget to indicate which are the **Virtual** and **Key** fields.

2.2.4 Logical Views Tab

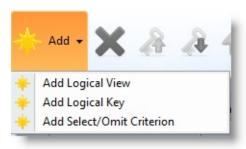
Select the Logical Views tab to work with logical views of the file.



For quick maintenance of logical keys, select the Keys item in the tree and use the grid in the Details tab.

You can carry out these actions on Logical Views:

Add a logical view



Use the Add Logical View option. An undefined logical view is added to the file.

Double-click the new logical view to display

Add keys to logical views

the Details tab and to define the logical view. To specify the key for the logical view, select the new key field and fill in the details in the Details tab.

Use the Add Logical Key option. Then select the new key field and specify it in the Details tab.

Add select/omit criteria

Use the Add Select/Omit Criterion option. Then select the new criterion and specify it in the Details tab.

Change logical view key sequence



Use the Move Key Up and Move Key Down commands.

Change select/omit criteria sequence



Use the Move Up and Move Down commands.

Delete keys, select/omit criteria and logical views



Highlight the logical view to be removed and use the Remove command to remove it.

Also See

Logical Views in the Technical Reference Guide
Logical View Development in the *Developer Guide*↑ 2.2 Files

2.2.5 File Rules and Triggers Tab

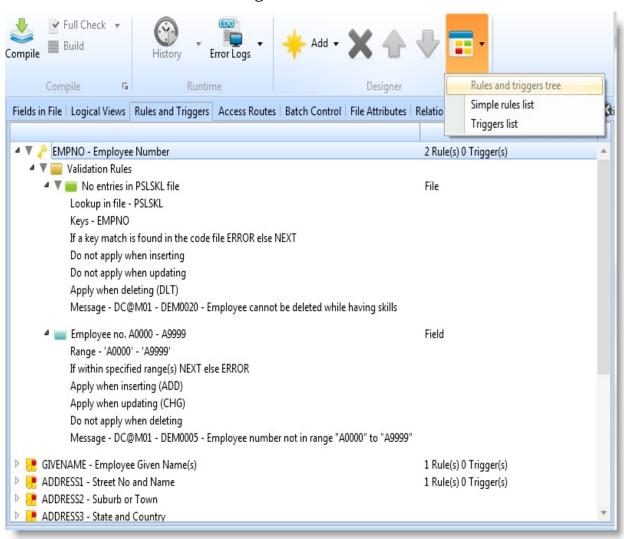
Select the Rules and Triggers tab to work with the file's rules and triggers for the fields in the file. Using the View command you can display this tab's contents as a:

File Rules and Triggers Tree View File Rules and Triggers List View Triggers List View

File Rules and Triggers Tree View

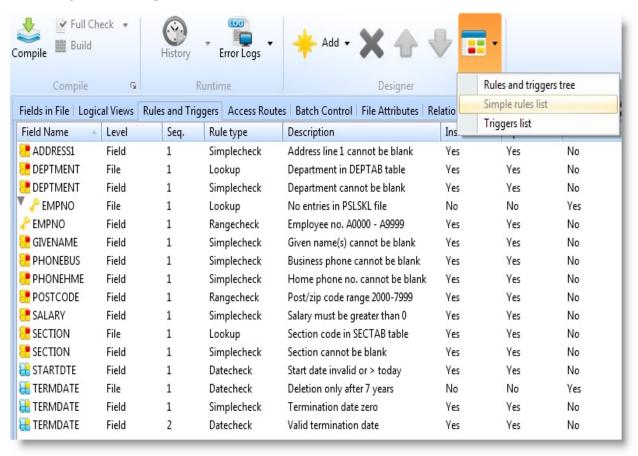
The tree view displays all the rules and triggers nested beneath the field they belong to.

Use the View command to change the Views.



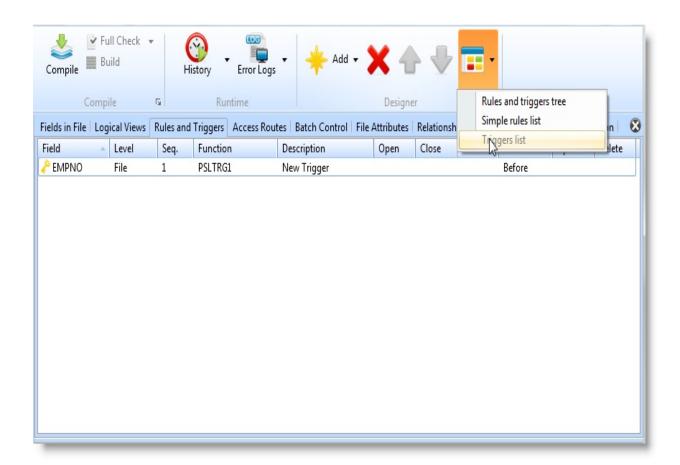
File Rules and Triggers List View

The list view displays the rules individually. The list can be sorted into sequence clicking on the required column's header.

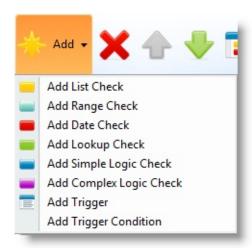


Triggers List View

The list view displays the triggers individually. The list can be sorted into sequence by clicking on the required column's header.



Add a rule or trigger



Select the field and use the Add command.

Edit a rule or trigger

Select the rule or trigger and double click on it to display the Details tab so that you can change it. Change the sequence of rules and triggers
Remove rules and triggers

View Style

Use the Move Up and Move Down commands.

Select the rule or trigger and use the Delete command.

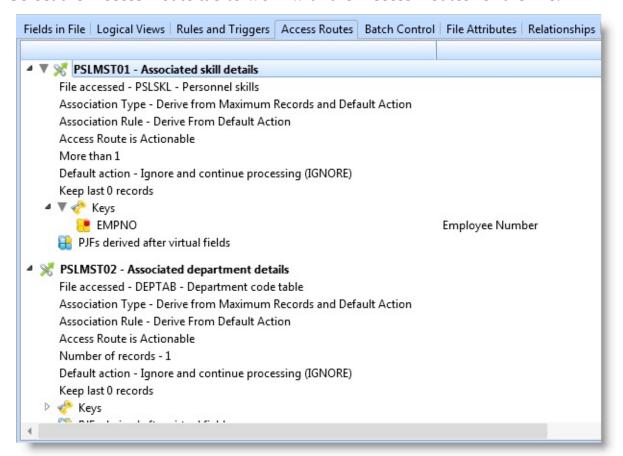
Change the view style so that you can see either the File Rules and Triggers Tree View or File Rules and Triggers List View.

Also See

Rules and Triggers in the *Technical Reference Guide*File Rules and Triggers Development in the *Developer Guide*1 2.2 Files

2.2.6 Access Route Tab

Select the Access Route tab to work with the Access Routes for the file.



Add an access route



Use the Add Access route option. An undefined access route is added to the file. Double-click the new access route to display the Details tab and to define the access route. To specify the key for the access route, select the new key field and fill in the details in the Details tab.

Add keys to access routes

Use the Add Key option. Specify the details of the key field in the Details tab.

Add PJFs to access routes

Select the access route and use the Add PJF option. Specify the details of the PJF it in the Details tab.

Change access route key sequence



Use the Move Key Up and Move Key Down commands.

Delete keys, PJFs and access routes



Use the Remove command.

Open Selected Field

Right-click to open the definition of the selected field.

Also See

Access Routes in the *Technical Reference Guide*Access Routes Development in the *Developer Guide*↑ 2.2 Files

2.2.7 Batch Control Tab

Select the Batch Control tab to work with the Batch Control Files.

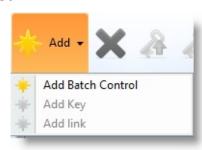
Batch control is a special feature of the LANSA Repository. Batch control logic is used to define how a numeric field in one file is totaled into the field of another file. This can be used as a means of verifying totals during, say, an order entry.

Batch control logic in files can improve performance by reducing the amount of database I/O.

To learn about batch control, refer to Batch Control Development in the Developer Guide.

Batch Control Toolbar

Add a batch control



Use the Add Batch Control option. An undefined batch control is added to the file. Double-click the new batch control to display the Details tab and to define the batch control.

Add keys to batch controls

Use the Add Key option. Specify the details of the key field in the Details tab.

Add batch control links

Select the batch control and use the Add Link option. Specify the details of the link in the Details tab.

Change batch control key sequence

Use the Move Key Up and Move Key Down commands.

Delete keys, links and batch control files

Use the Remove command.



Open Selected Field

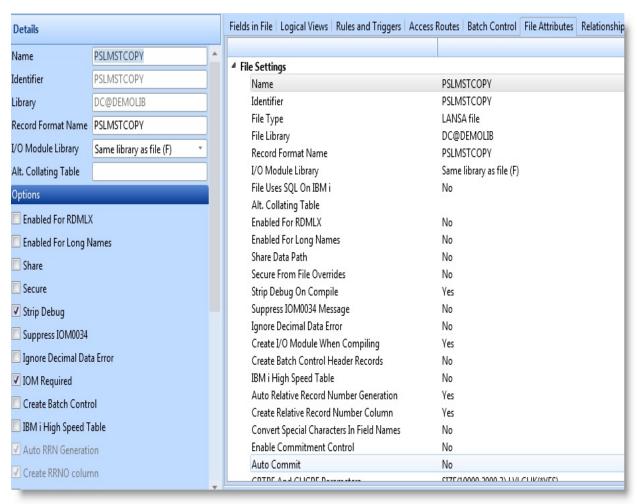
Right-click to open the definition of the selected field.

Also See

Batch Control in the *Technical Reference Guide*Batch Control Development in the *Developer Guide*↑ 2.2 Files

2.2.8 File Attributes Tab

Select the File Attributes tab to define the attributes of the file.



These two commands apply to this tab:



Enable a LANSA file for RDMLX. This option is only available for non-RDMLX LANSA files.



Convert an IBM i OTHER file to a LANSA file. This option is only available for IBM i OTHER files.

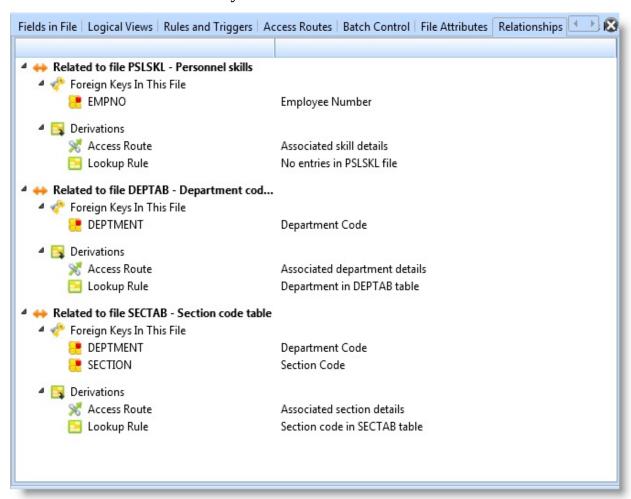
The attributes are changed in the Details Tab. Click on any item in the File Attributes tab to display the Details tab.

Also See

File Attributes in the Technical Reference Guide

2.2.9 Relationships Tab

Select the Relationships tab to view the relationships this file has to other files. This information is read-only.

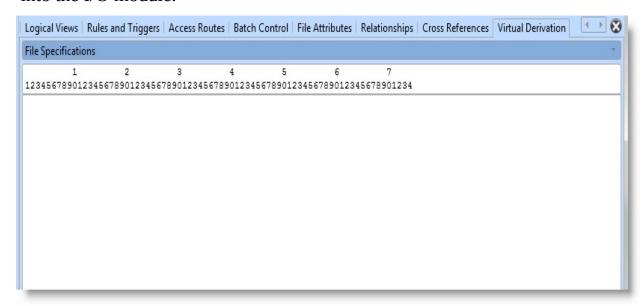


2.2.10 Cross-References Tab

See Cross-References Tab.

2.2.11 Virtual Derivation Tab

Select the Virtual Derivation tab to work with RPG or C code that is imbedded into the I/O module.



Use the dropdown list on the top of the tab to select the type of the virtual code you are entering.



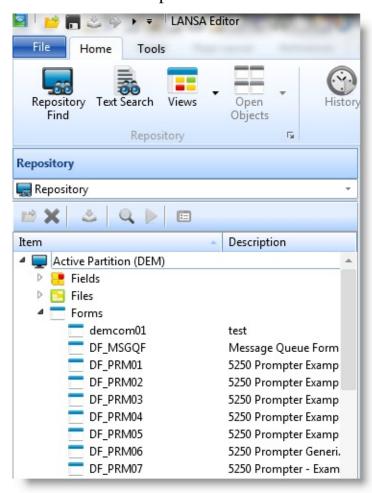
Also See

Virtual Derivation in the Technical Reference Guide

2.3 Forms

Source Tab Design Tab for Forms Repository Help Tab 2.3.1 Create Forms

Forms are selected from the Repository tab or Favourites tab. Expand the tree view or double click on Forms to open the list of forms available for this partition.



Also See

Creating Applications Using Components in the *Developer Guide*. Components in the *Technical Reference Guide*.

↑ 2. Editing Objects

2.3.1 Create Forms

To create a form use the New command.

The New form dialog is displayed:



- Specify the Name, Description and other details of the form that is to be created. Refer to Component Definitions in the *Technical Reference Guide*.
- When the required details have been specified, press the Create button to create the new form.

The form will be opened in the editor.

To copy an existing form, select the form in the Repository Tab or the Favorites Tab and use the Copy option from the context menu.

Also See

Creating Applications Using Components in the Developer Guide.

Components in the *Technical Reference Guide*.

↑ 2.3 Forms

2.4 Reusable Parts

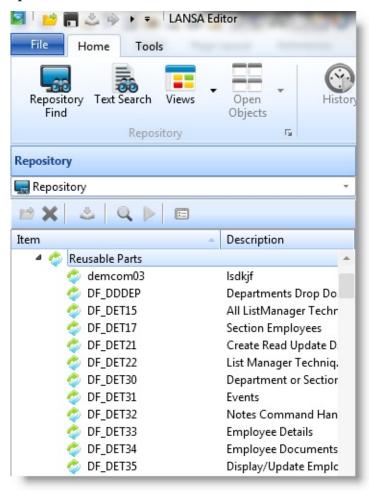
Source Tab

Other Objects in the Design Tab

Repository Help Tab

2.4.1 Create Reusable Parts

Reusable parts are selected from the Repository tab or Favorites tab. Expand the tree view or double click on Reusable Parts to open the list of reusable parts available for this partition.



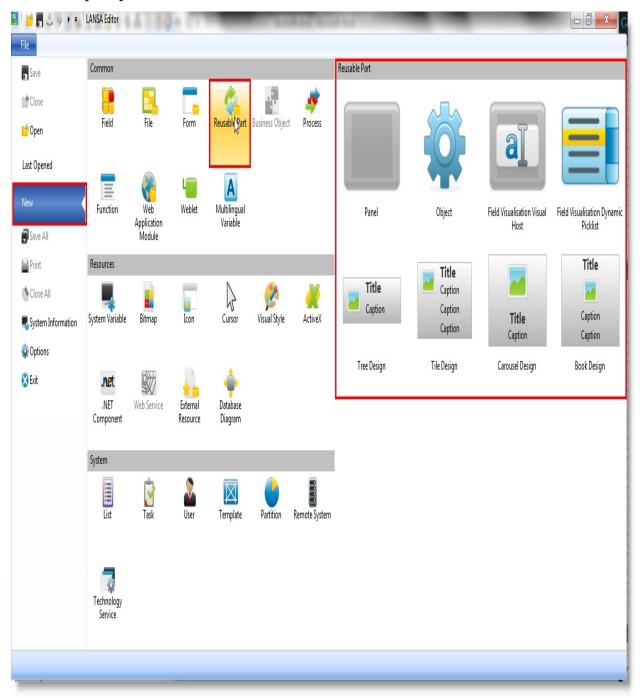
Also See

Components in the *Technical Reference Guide*.

↑ 2. Editing Objects

2.4.1 Create Reusable Parts

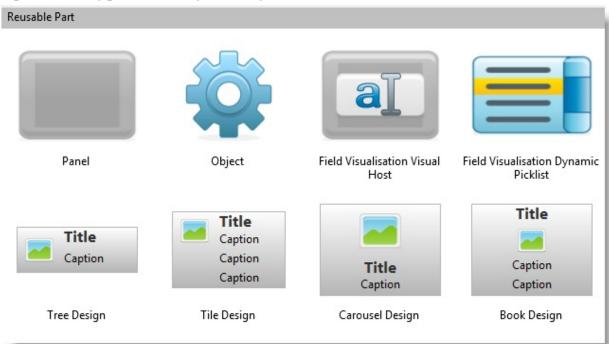
To create a reusable part use the New command. Then choose the type of reusable part you want to create:



There are several different types of reusable part. Some reusable parts are created with only the simplest code possible, while others have a number of

lines added to help simplify the development process.

All reusable parts ultimately inherit from the same primitive class and choosing a particular type in no way limits your choices after creation.



Panel

A simple component that inherits from a primitive panel (#Prim_panl) and is the simplest visible component possible. Only this type of reusable part was available prior to Version 13.

Object

A simple component that inherits from a primitive object (#Prim_Objt) and has no visible features.

Field Visualization Visual Host Similar to a panel but used as the editable portion of a repository field as part of Field Visualization.

When created the reusable part will implement the

#Prim_dc.iMonitorSubject

#Prim_dc.iContextualSubject interfaces and will include shell method routines and sample code as a

guide.

Field VisualizationDynamic Picklist

Similar to a Visual Host but used instead as the loader mechanism for a Field Visualization dynamic picklist. When created the reusable part will implement the #Prim_dc.iDynamicPicklist interface

and will include sample code as a guide.

Tree Design

Similar to a panel but used as the visible design portion of a Tree user designed control (#Prim_Tree). When created the reusable part will implement the #Prim_Tree.iTreeDesign interface and will include shell method routines for the interface features.

Tile Design

Similar to a panel but used as the visible design portion of a Tile user designed control (#Prim_Tile). When created the reusable part will implement the #Prim_Tile.iTileDesign interface and will include shell method routines for the interface features.

Carousel Design

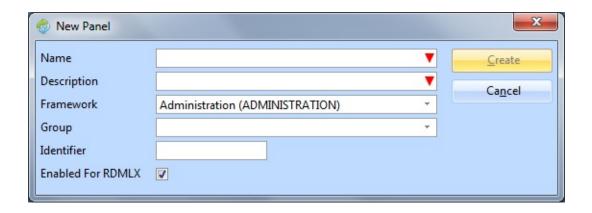
Similar to a panel but used as the visible design portion of a Carousel user designed control (#Prim_Caro).

When created the reusable part will implement the #Prim_Caro.iCarouselDesign interface and will include shell method routines for the interface features.

Book Design

Similar to a panel but used as the visible design portion of a Book user designed control (#Prim_Book).

When created the reusable part will implement the #Prim_Book.iBookDesign interface and will include shell method routines for the interface features.



- Specify the Name, Description and other details of the reusable part that is to be created. Refer to Component Definitions in the *Technical Reference Guide*.
- When the required details have been specified, press the Create button to create the new reusable part.

The reusable part will be opened in the editor.

To copy an existing reusable part, select the reusable part in the Repository Tab or the Favorites Tab and use the Copy option from the context menu.

Also See

Components in the *Technical Reference Guide*.

↑ 2.4 Reusable Parts

2.5 Processes

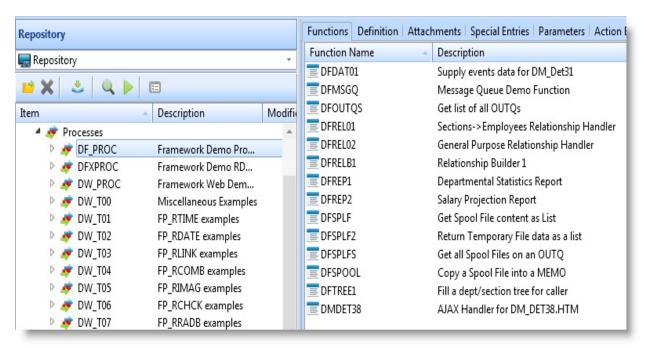
2.5.1 Create 2.5.2 Copy a 2.5.3 Functions Tab 2.5.4 Definition

Processes Process 2.5.7 Process Tab

2.5.5 Attachments 2.5.6 Special Parameters Tab 2.5.8 Action Bar

Tab Entries Tab Table Tab

2.5.9 Menu Structure Tab



Also See

Processes and Functions in the *Technical Reference Guide*Developing with Processes and Functions in the *Developer Guide*† 2. Editing Objects

2.5.1 Create Processes

To create a process use the New command.

The New process dialog is displayed:



- Specify the Name, Description and Identifier of the new process that is to be created. Refer to Process Definition in the Technical Reference Guide.
- Select the Open in editor option if you wish to specify additional process details.
- When the required details have been specified, press the Create button to create the new process.

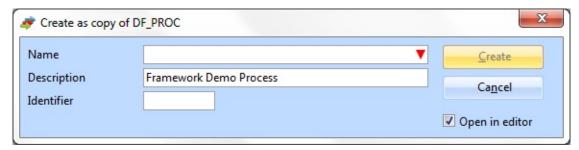
Also see

Processes and Functions in the Technical Reference Guide

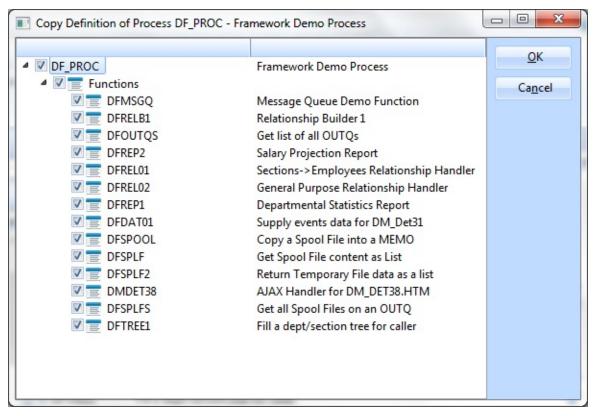
1 2.5 Processes

2.5.2 Copy a Process

To copy an existing process, select the process in the Repository Tab or the Favorites Tab and select the Copy option from the context menu to open the Create as a copy of..... dialog.



The Copy Definition of Process... dialog is used to select which functions you want to copy from the process being copied.

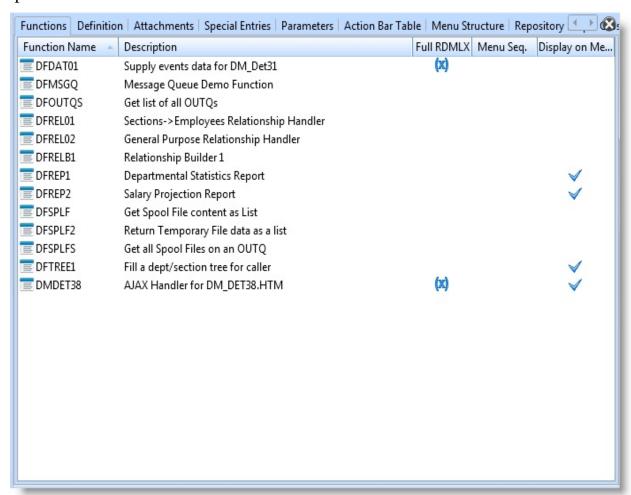


1 2.5 Processes

2.5.3 Functions Tab

The Functions tab lists the functions in the process. By clicking on a function, you open the Details tab, which allows you to specify how the function is to be used in this process. The Details tab provides the equivalent facilities to the Function Control Table in LANSA for iSeries.

You can add a function using the Functions tab, but any other actions, such as changes or to delete a function must be performed within the Functions editing process.



To make adjustments to the sequence or the next function to be added, double click on the function in the Functions tab to open the Details tab.

Add a function



Use the Add command to specify a function that is to be added to the process.

Double-click the function to display the Details tab

and to define its source code.

Full This column shows whether the function is enabled

RDMLX for full RDMLX.

Display on This column shows whether the function is displayed

menu on the program start dialog.

Sequence This column shows the order in which the functions

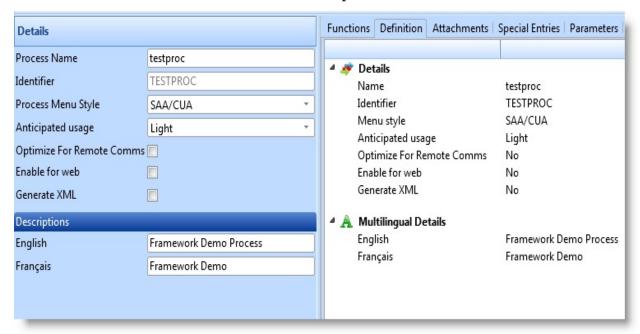
are shown in the process menu.

Also See

Function Control Table in the *Technical Reference Guide*Developing with Processes and Functions in the *Developer Guide*↑ 2.5 Processes

2.5.4 Definition Tab

Select the Definition tab to work with the process definition.



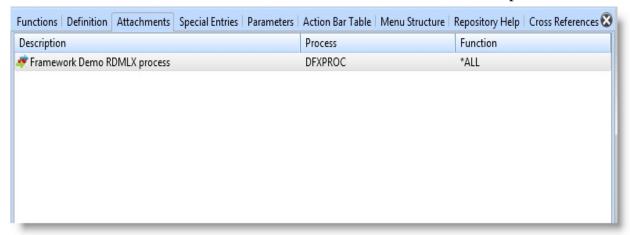
The attributes are edited in the Details tab.

Also See

Process Definition in the *Technical Reference Guide*Developing with Processes and Functions in the *Developer Guide*1 2.5 Processes

2.5.5 Attachments Tab

Select the Attachments tab to work with the Attachments for the process.



Add an attachment



Use the Add option from the Add command. An undefined attachment is added to the process.

Double-click the attachment to display the Details tab and to define its details.

Edit an attachment

Select an attachment and double-click to show the Details tab.

Change the sequence of attachments



Use the Move Key Up and Move Key Down commands.

Remove an attachment



Use the Remove command.

Also See

Attached Processes and Functions in the *Technical Reference Guide*Developing with Processes and Functions in the *Developer Guide*

↑ 2.5 Processes

2.5.6 Special Entries Tab

Select the Special Entries tab to work with the Special Entries associated with the selected LANSA process.

Special Entries are IBM i CL commands that have been defined to LANSA.

Add a special entry



Use the Add option from the Add command. An undefined special entry is added to the process.

Double-click the special entry to display the Details tab and to define its details.

Edit a special entry

Select a special entry and double-click to show the Details tab.

Change the sequence of special entries



Use the Move Key Up and Move Key Down commands.

Remove a special entry



Use the Remove command.

Also See

Special Entries in the *Technical Reference Guide*Developing with Processes and Functions in the *Developer Guide*↑ 2.5 Processes

2.5.7 Process Parameters Tab

Process Parameters should not be used, as they cannot be executed in Visual LANSA. They exist only for backward compatibility.

Also See

Process Parameters in the Technical Reference Guide.

↑ 2.5 Processes

2.5.8 Action Bar Table Tab

Select the Action Bar Table tab to work with the Action Bar Items and their pull-down options for the selected process.

Add an action bar item



Use the Add option from the Add command. An undefined action bar item is added to the process.

Double-click the action bar item to display the Details tab and to define its details.

Add a pulldown item

Select the action bar item with which to associate the pull-down item and then use the Add option from the Add command. An undefined pull-down item is added to the process.

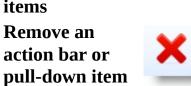
Double-click the action bar item to display the Details tab and to define its details.

Edit a action bar or pull-down item

Select the item and doubleclick to show the Details tab.

Change the sequence of action bar or pull-down items





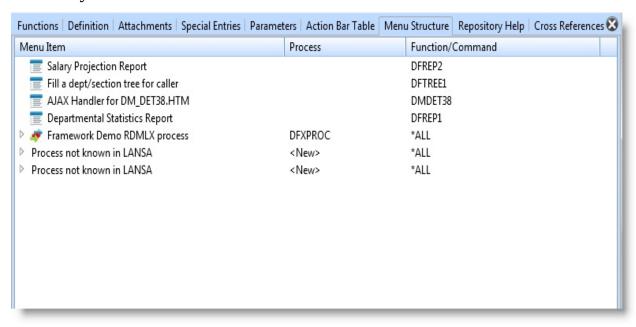
Use the Remove command.

Also See

Action Bar Table in the *Technical Reference Guide*Action Bar Concepts in the *Developer Guide*Developing with Processes and Functions in the *Developer Guide*↑ 2.5 Processes

2.5.9 Menu Structure Tab

Select the Menu Structure tab to view the menu structure of the process consisting of functions, special entries and attachments. This information is read-only.



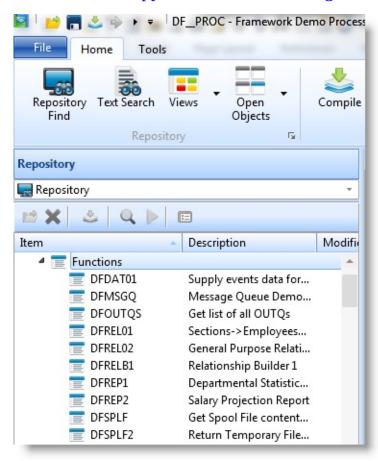
Also See

Processes and Functions in the Technical Reference Guide

↑ 2.5 Processes

2.6 Functions

2.6.1 Create Functions 2.6.2 Copy Functions 2.6.3 Design Functions



Also See

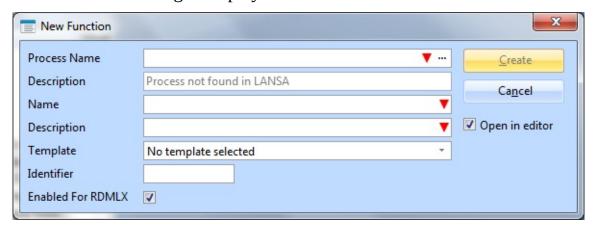
Function Definitions in the *Technical Reference Guide*Design Tab for Functions
Source Tab

↑ 2. Editing Objects

2.6.1 Create Functions

To create a function use the New command.

The New function dialog is displayed:

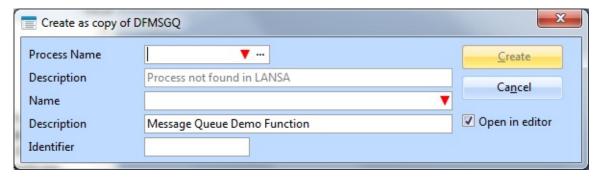


- Specify the process name, function name, function description, and other details of the new function that is to be created. Refer to Function Definitions in the *Technical Reference Guide*.
- If you want to use a template to create the function, select it from the dropdown list. When you use a template, you must select the Open in editor option or the template cannot execute.
- Select the Open in editor option if you wish to begin working with the RDML code in the function.
- When the required details have been specified, press the Create button to create the new function.

Developing with Processes and Functions in the *Developer Guide*Processes and Functions in the *Technical Reference Guide*1 2.6 Functions

2.6.2 Copy Functions

To copy an existing function, select the existing function from the Repository Tab or the Favorites Tab and use the Copy option from the context menu to open the Create as a copy of... dialog.



- Specify the Process name, Function name and Description of the new function that is to be created. Refer to Function Definitions in the *Technical Reference Guide*.
- Select the Open in editor option if you wish to begin working with the RDML code in the function.
- When the required details have been specified, press the Create button to create the new function.

1 2.6 Functions

2.6.3 Design Functions

2.6.4 Function Screen Designer 2.6.5 Function Report Designer

The Function Screen Designer and Function Report Designer provide control and flexibility in the design phase of your screens and reports. With them you can:

- Visualise screens as 5250 or Windows interfaces
- Modify the screen and report related parameters
- Provide easy access between the source and design tabs (i.e. no need to perform full function check).

Also see



FRM115 - Writing Reports in the Windows Fundamental Tutorials.

1 2.6 Functions

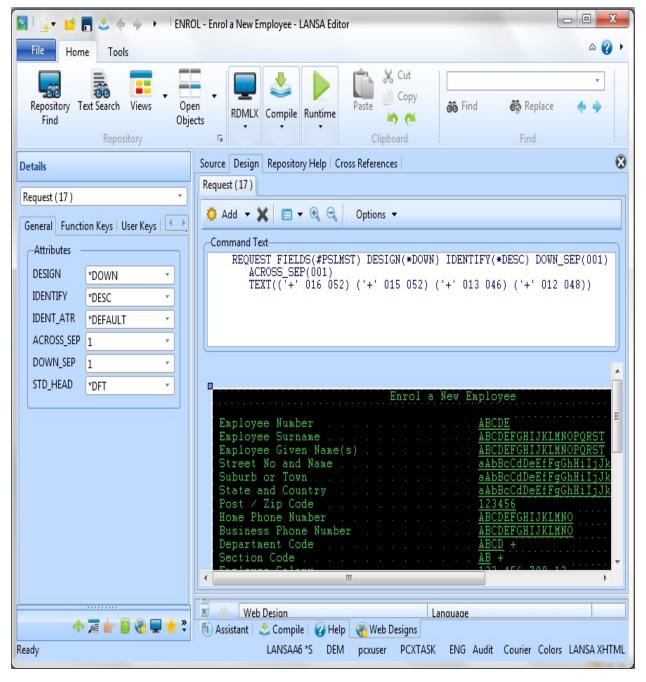
2.6.4 Function Screen Designer

Screen Representation General Function Keys User Keys Tab

Tab Tab

Fields, Groups and Lists Field Tab Add Text Create a Screen

Tab



The Design tab folder shows what the function you are working with will look

like at execution time. The tab folder contains a tab for every REQUEST, DISPLAY and POP_UP command in the function with a tab folder grouping all report commands used in the function.

If the function includes any report definitions, a single Report tab allows the report format (including DEF_HEAD, DEF_LINE, DEF_BREAK and DEF_FOOT commands) to be reviewed. Multiple reports can be reviewed from this tab.

↑ 2.6.3 Design Functions

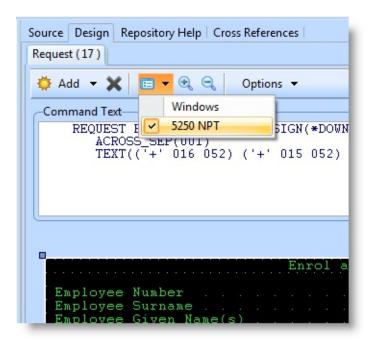
Screen Details Tab

The Details tab folder for the report design shows the REQUEST, DISPLAY or POP_UP command being designed. Focusing on different parts of the screen will change the tab focus accordingly.

↑ 2.6.4 Function Screen Designer

Screen Representation

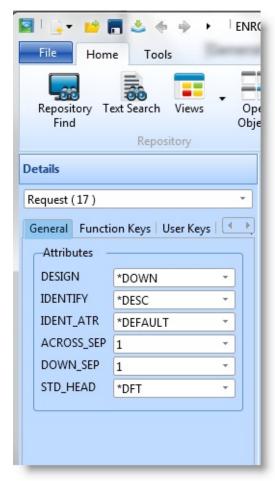
You have the option of displaying a screen representation of the command as it is displayed on a 5250 non-programmable terminal or as the screen is displayed in Windows:



↑ 2.6.4 Function Screen Designer

General Tab

The General tab shows the DISPLAY and REQUEST command parameters that govern the layout of the screen. Note that if specific attributes have been specified for a field, they override the general attributes. In this case, changing the general attributes has no effect.



Select from the dropdown lists, the options you require for the following:

DESIGN

The design/positioning method that should be used for fields that do not have specific position attributes associated with them.

IDENTIFY

The default identification method to be used for fields.

INDENT_ATR Displays attributes that are to be associated with identification text (labels, description, column headings and so on).

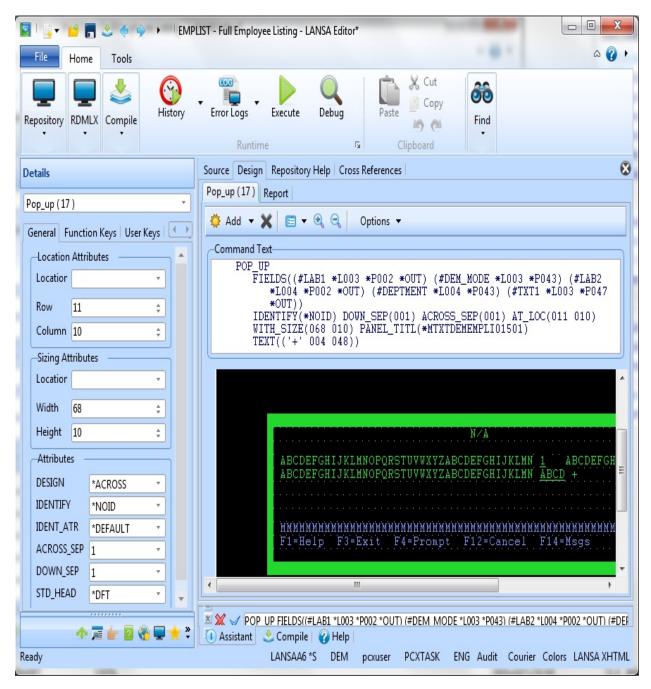
ACROSS_SEP The spacing between columns.

DOWN_SEP The spacing between rows.

STD_HEAD Specifies whether or not the standard LANSA design for the

screen heading lines should be used.

For POP_UP commands, the General tab will contain fields to allow you to specify the location and size of the pop-up window.

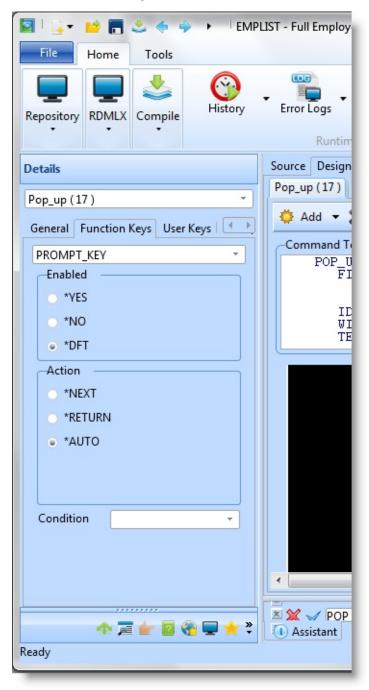


Also See

REQUEST command parameters in the *Technical Reference Guide*. DISPLAY command parameters in the *Technical Reference Guide*. POP_UP command parameters in the *Technical Reference Guide*. ↑ 2.6.4 Function Screen Designer

Function Keys Tab

Use the Function Keys tab to specify the settings for the standard function key parameters of the DISPLAY, REQUEST or POP_UP command:



The function keys are:

• PROMPT_KEY

- EXIT_KEY
- MENU_KEY
- ADD_KEY
- DELETE_KEY
- CHANGE_KEY

Also See

REQUEST command parameters, MENU_KEYS in the *Technical Reference Guide*.

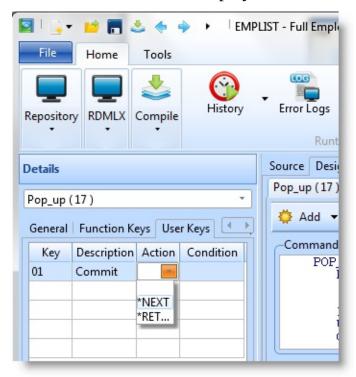
DISPLAY command parameters in the *Technical Reference Guide*.

POP_UP command parameters in the Technical Reference Guide.

↑ 2.6.4 Function Screen Designer

User Keys Tab

Use the User Keys tab to specify up to five additional user function keys that can be enabled when the screen format is displayed at the workstation:



Also See

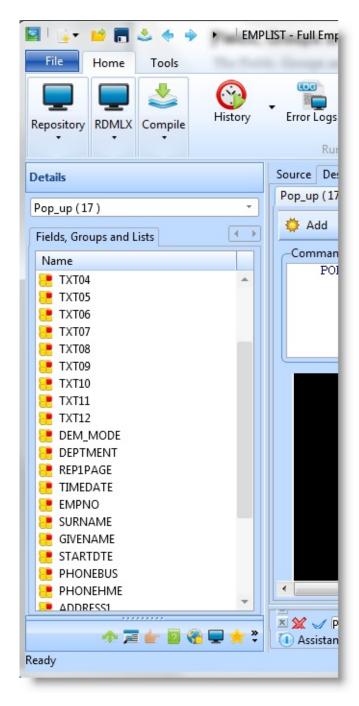
REQUEST command parameters, USER_KEYS in the Technical Reference Guide

DISPLAY command parameters in the Technical Reference Guide POP_UP command parameters in the Technical Reference Guide.

↑ 2.6.4 Function Screen Designer

Fields, Groups and Lists Tab

The Fields, Groups and Lists tab shows the fields, groups and lists defined in the function. You can drag and drop lists and groups (not individual fields) onto your screen representation to include them in your command provided are compatible with the command represented on the screen. You can also drag and drop fields from the Repository tab, provided the fields are compatible with the command represented on the screen.



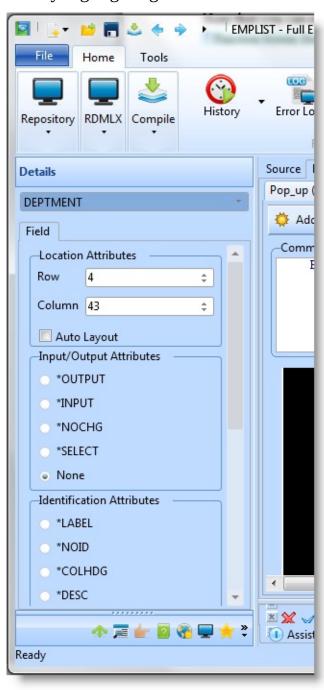
Note that you can also drag and drop fields from the Repository tab.

↑ 2.6.4 Function Screen Designer

Field Tab

If an individual field has been selected, the Details tab shows the attributes for that field.

You can specify attributes for the selected field which override any overall layout settings specified in the General tab for the command. List field attributes can also be maintained by highlighting the list on the screen representation.



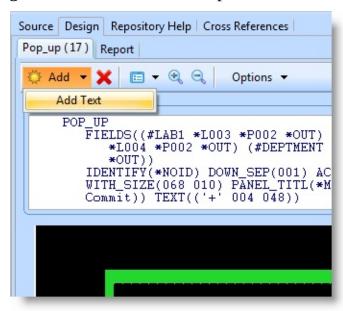
Also See

RDML Field Attributes and their Use in the Technical Reference Guide ↑ 2.6.4 Function Screen Designer

Add Text

You can specify up to 50 text strings that are to appear on the screen. Each text string has a maximum length of 20 characters.

To add a text string, use the Add Text Field option in the Add toolbar menu:

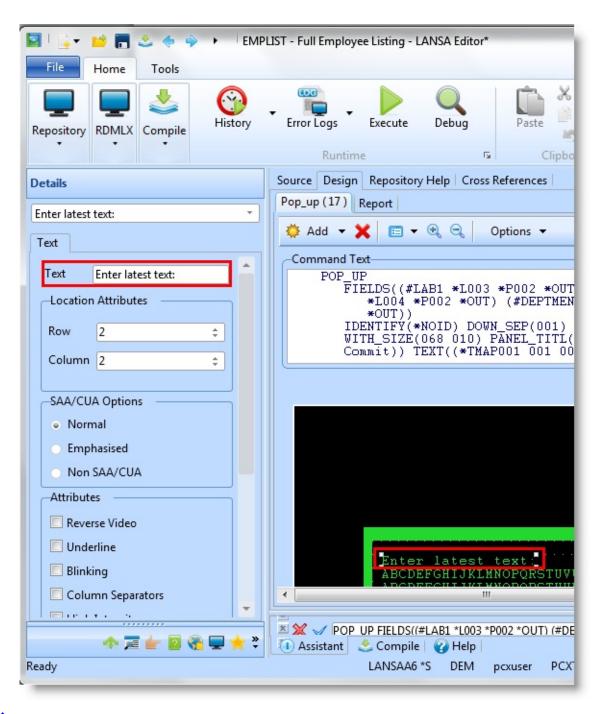


A new field will added to the window. First enter the text you require in the Text entry in the Design tab's Text tab and press Enter.

If you are using 5250 NPT Representation, you will set the location of the new text using the Location Attributes. The text will move to match the attributes as you select them.

If you are using the Windows Representation, then you can drag and drop your text using the mouse. The Location Attributes will change to reflect the final location of the text.

Using the Text tab, you can specify the SAA/CUA Options, if applicable and the Attributes and Color as required.

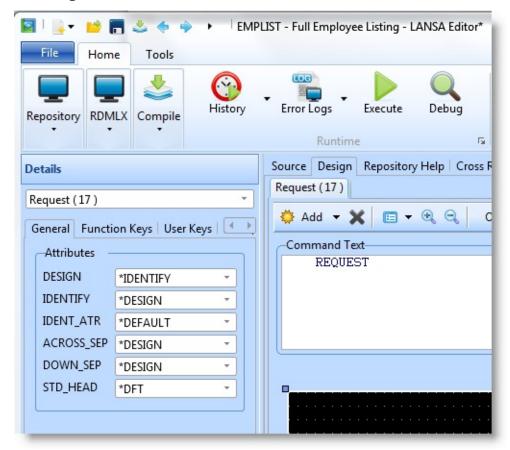


↑ 2.6.4 Function Screen Designer

Create a Screen

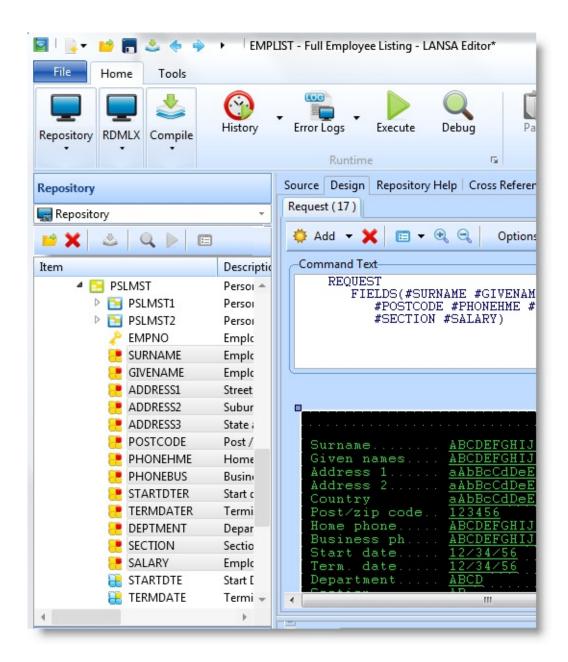
To create a screen, enter a DISPLAY, REQUEST or POP_UP command on a new line in the Source view, display the Design tab and then drag and drop the fields you want included either from the Fields, Groups and Lists tab in the Details tab folder or from the Repository tab.

For example, to create a screen that requests employee details from the Demonstration Personnel system, enter a REQUEST command on a blank line. Select the Design tab:



Select the Repository tab and expand the Files node. Locate the **PSLMST** file and expand it to display all the fields in the file.

Select all the fields in the file and drag and drop them to the screen representation.



Add a field to the command screen

You can display the Details tab and select the fields to be included on your screen

or

you can drag and drop them from the fields in the Repository tab

or

you can make a list of them which you can drag and drop into the screen.

Move a field

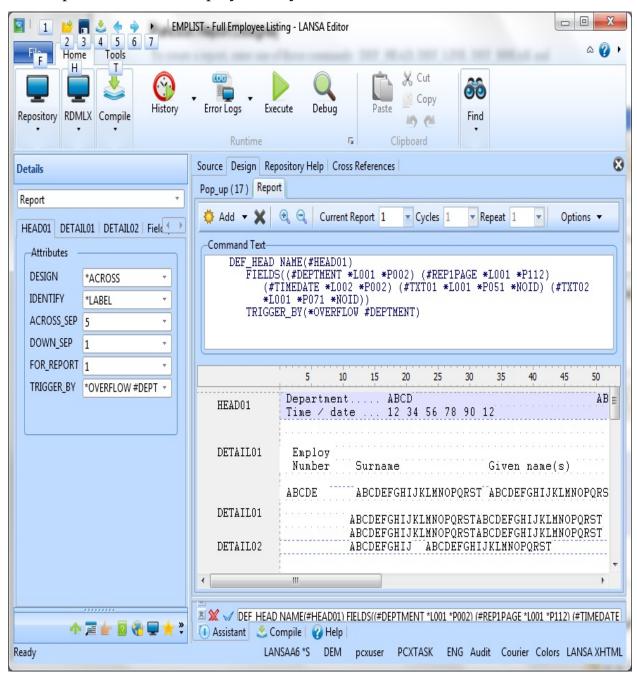
Select the field and drag-and-drop it to a new location.

Move a group of fields		Select the fields and drag-and-drop them to a new location.
Delete a field/text	×	Select the fields or text and press the Delete key or use the Delete command.
Add text	Add Text	Refer to Add Text
Showing a grid		Display the LANSA Settings dialog, then display the Design settings and select Show Grid option.
Increase or Decrease Text size	e e	Select the text to be increased or decreased and use whichever command is relevant.

↑ 2.6.4 Function Screen Designer

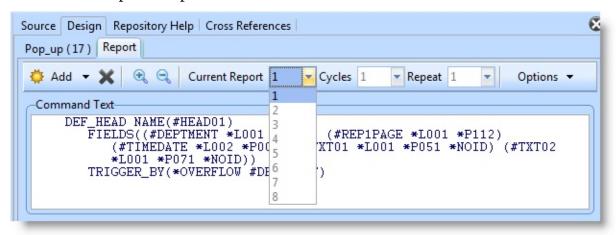
2.6.5 Function Report Designer

To create a report, enter one of these commands: DEF_HEAD, DEF_LINE, DEF_BREAK and DEF_FOOT in the Source Tab. Switch to the Design tab and the Report tab will be displayed for you to select.



All commands associated with a report can be reviewed in the Report tab by focusing on the appropriate line in the design.

Multiple reports can be designed by selecting the appropriate report number in the Current Report drop down list.



Also see



FRM115 - Writing Reports in the Windows Fundamental Tutorials.

1 2.6.3 Design Functions

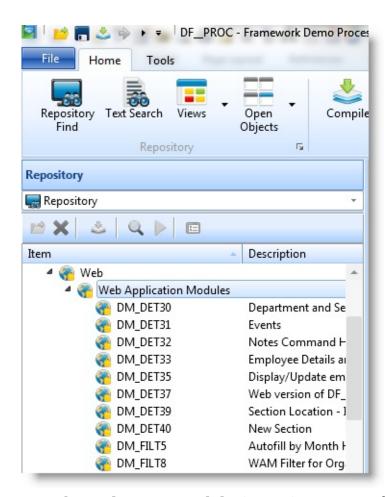
2.7 Web Application Modules (WAMs)

2.7.1 Create a 2.7.4 Details Tab 2.7.6 Create a Web Page WAM (Property Sheet) 2.7.7 Design Versions

2.7.2 Copy a WAM 2.7.5 Webroutine Output 2.7.8 Using External Resources

2.7.3 Outline tab – 2.7.3 Outline tab – with WAMs

WAMs WAMs



A Web Application Module (WAM) consists of one or more Webroutines. Each Webroutine will have at least one and maybe more than one Web Design. For the purpose of a Web Design, refer to The Presentation Layer in the Web Application Module Guide.

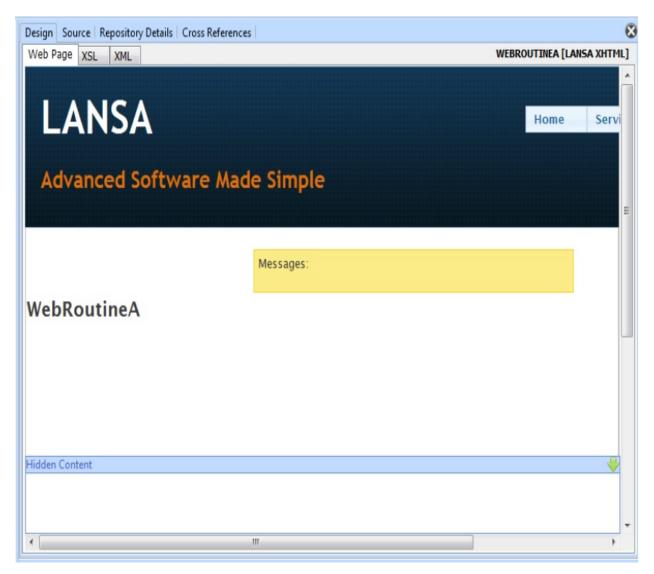
A Webroutine will have more than one Web Design when a different presentation is required for a different Technology Service Provider (TSP) or language. If you create WAMs for different languages, we recommend that you use Multilingual Text variables instead of creating multiple Web Designs, in

order to minimize your Web Design maintenance.

LANSA can save versions of a Web Design. The number of Web Design versions to be saved is set in the Number of saved Web Designs in the LANSA Editor's XSL - General tab. Keeping versions can be useful if you need to roll back changes to a Web Design.

You enter and edit the RDMLX WAM code in the Source Tab.

and create the Web Design in the Design tab:



The Design tab contains three tabs, the Web Page tab, the XSL tab and the XML tab.

- The **Web Page** tab shows the Web Design for a WebRoutine. This is where fields, lists, weblets and so on are dropped to create a design for the webroutine.
- The XSL tab allows text editing of the Web design's XSL source. We don't recommend that you work using this tab as errors and unexpected behavior can occur if you are not experienced in XSL and XPath.
- The XML tab displays a sample of the XML output by the Webroutine. This XML provides the input for the XSL transformation that produces the final HTML output. You can edit this XML to change the sample data displayed in the design. Note that changed sample data will only be saved in the repository if the Save Webroutine Field and List sample data setting is turned

on. Also note that the fields and list sections in the XML are generated by LANSA at runtime.

The name of the webroutine, the current Technology Service Provider and current language are shown in the upper right area of the Design tab. If there is a language specified after the Technology Service Provider then the design is for that language, otherwise it is for the partition's default language. In the example above the webroutine name is WR1, the Technology Service Provider is LANSA XHTML and the language is the partition's default language.

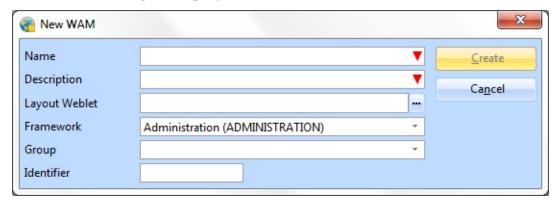
1 2. Editing Objects

2.7.1 Create a WAM

Note: WAMs can only be created in an RDMLX partition enabled for the web.

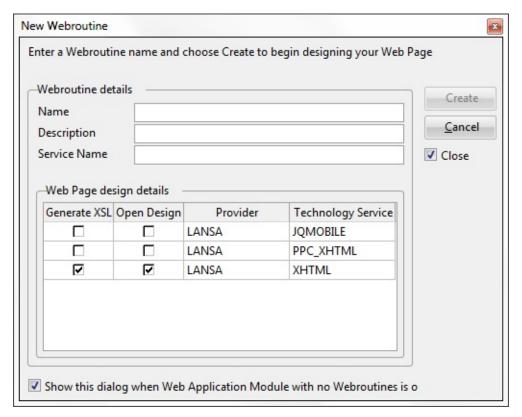
To create a WAM use the New command.

The New WAM dialog is displayed:



- Specify the Name, Description, Layout Weblet and Other details of the WAM that is to be created. For details of these entries, as well as Framework and Group, refer to Component Definition in the *Technical Reference Guide*.
- A Layout Weblet provides the basic HTML document structure (html, head, body, script, style, etc.) required by all web pages.
- Select the Framework with which you want to associate the WAM. The Framework is most likely the application to which the WAM will belong.
- Select the Group with which you want to associate the WAM.
 Groups and Frameworks are defined on the IBM i and imported to Visual LANSA.
- When the required details have been specified, press the Create button to create the WAM.

Once the WAM has been created the New Webroutine dialog is displayed – unless the option to show this dialog automatically has been unselected.



- Specify Name and optionally Description and Service Name of the Webroutine that is to be created.
- Select the Technology Service Providers you want to generate XSL for. For information about Technology Services, refer to Technology Services in the Technical Reference Guide.
- Select which Technology Service Provider you want to automatically open once the Webroutine is created.
- When these required details have been specified, press the Create button. The RDML is created and the Web Design tab is automatically opened.
- Cancel will return you to the RDML Source tab.

Once in Design tab you can start to design the look and feel of your web page by dragging and dropping items from the repository onto the Design.

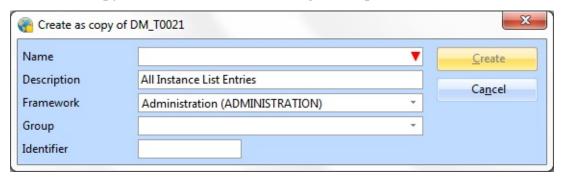
Additional webroutines can be created from the Source tab by right mouse clicking in the WAM and selecting New Webroutine... from the context menu.

1 2.7 Web Application Modules (WAMs)

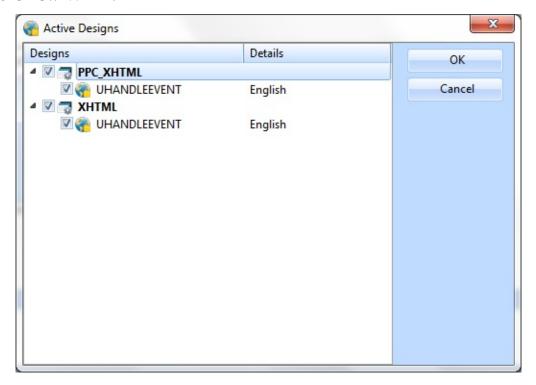
2.7.2 Copy a WAM

To create a WAM by copying from an existing one, select the WAM to be copied in either the Repository or Favorites tab, right click to open the context menu and choose the Copy command.

The Create as copy of <WAM name> dialog will open.



Enter the new WAM's name and other details and press the Create button. The Active Designs dialog will open for you to select the web designs to be used with the new WAM.

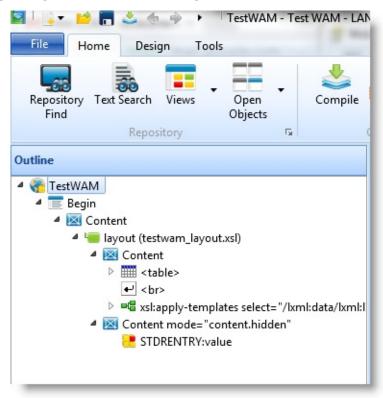


1 2.7 Web Application Modules (WAMs)

2.7.3 Outline tab – WAMs

When working with a WAM, the Outline tab lists the open WAM and its webroutines. It shows items (fields, lists, descriptions, labels, headings, weblets, multilingual variables, system variables) used in a webroutine's Web Page.

This tab is useful for switching between different open documents, selecting layouts, and opening Weblets for editing.



To expand and collapse the item, click on the plus or minus sign to the left of the name in the Outline.

The Outline tab and Web Page are synchronized so that if you select an item in one the item will be selected, if possible, in the other as well.

The items on the context menu will vary depending on the context in which it is opened.

↑ 2.7 Web Application Modules (WAMs)

2.7.4 Details Tab (Property Sheet)

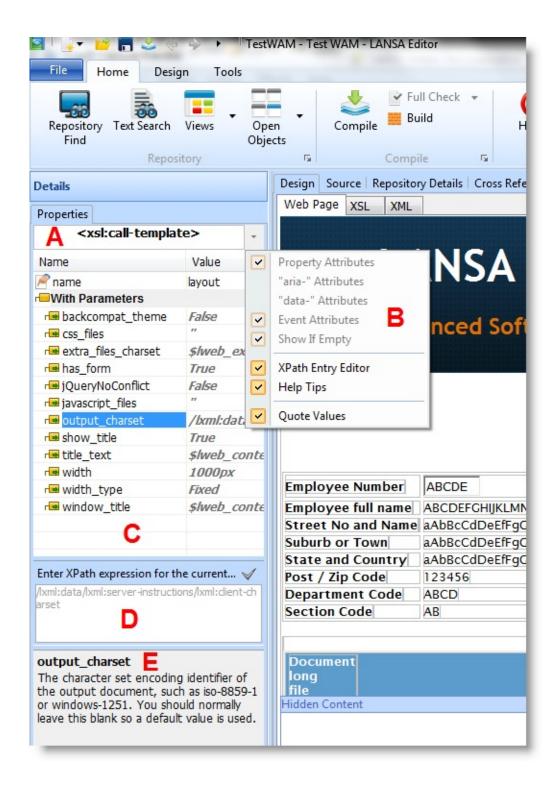
One of the most used tabs when working with WAMs is the Details tab. It allows you to change properties, which affect the look and feel of elements on your web design.

The Details tab removes a lot of the guesswork of filling in details, i.e., when you need quotes around values and when you don't. You don't need to specify the quotes around values, they are implied.

When you are working with a webroutine's presentation layer or web design, the currently selected item in the design view has its properties visible in the Details tab.

Note that some values are in gray italics. They indicate that no value has been entered or selected and hence the property has no value. In this case a default value (the value shown in gray italics) are used.

The Details tab's Property Sheet is divided into several sections as shown in this screen capture:



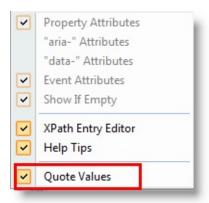
- **A** Name: At the top of the Properties Sheet is the name of markup tag.
- **Property Sheet Menu Options**: The gray button, at the top right of the Details tab is a drop down menu. This menu shows Properties, Events,

XPath Entry Editor and Quote Values. Properties and Events toggle display of respective attribute groups in the Properties area for HTML attributes. XPath Entry Editor shows or hides the entry area for XPath expressions. Quote Values toggles automatic quoting of values.

- **Properties**: The available properties for the currently selected markup tag are then listed in the Properties Sheet. Each weblet and each HTML tag will have a unique set of properties.
- **Npath Entry Editor**: This is where XPath expressions for a property are entered. Select the property and enter the XPath expression here. Don't enter XPath expressions directly into the property field. Use this editor which you open from the Property Sheet Menu.
- **Help**: Shows a short description of the currently selected property. In the example below, the help shows a short description of the std_char weblet's name parameter.

If the HTML tag is visualizing a LANSA field there is also a property for setting the field's sample value. Setting the sample value is not possible for fields inside inline lists. If you change the sample value it will not be saved in the XML unless the Save Webroutine Field and List sample data option is turned on in the Settings dialog. Turning on this option will impact performance when loading a web design.

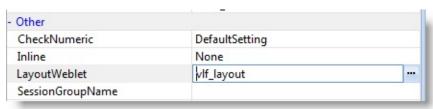
Unselecting Quote Values will put the responsibility for making sure values are properly quoted over to the user. Note that after unselecting Quote Values some values in the Details tab are shown with quotes. It is recommend that you select the Quote Values setting.



You cannot enter XPath expressions directly in the field immediately to the right of a property name. You must use the XPath Entry Editor to enter XPath expressions. If you don't intend to use XPath expressions, it is recommended that the XPath Entry Editor is not shown.

Some properties are enumerated, i.e., they can have a set number of predefined values. In such cases the Property Sheet will show a drop down and its values can be selected or the value can just be typed in directly. The last item in the drop down is shown in gray and italics. This is the default value. If the default value is empty then
blank> will be shown in its place in the drop down.

Most weblet properties and some HTML attributes have prompters, which assist you in filling in details or select values. If a property shows an Ellipse button (...) then it has a Prompter/Designer/Editor for your assistance.

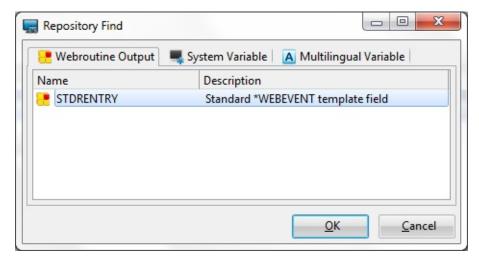


In many cases it is still possible to type into the value field without using the Ellipse ... button for assistance.

Each weblet may have custom property editors of its own.

Some common Property Prompters/Designers/Editors:

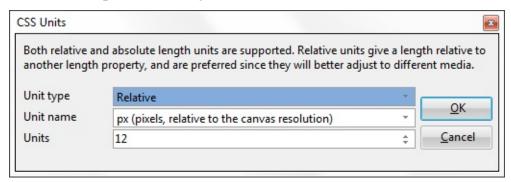
Variable Prompter: Either chose a LANSA field which is in one of the
web_maps available for this webroutine, a System variable or a Multilingual
variable. This can be useful if the property's value can vary at runtime.
Webroutine output lists all webroutines available to the current webroutine.
System Variable and Multilingual Variable list all available for the current
partition.



WAM Name Prompter: In addition to picking a LANSA field, a System

variable or a Multilingual variable, a WAM's name can be picked from a list of available WAMs. Webroutine output lists all webroutines available to the current webroutine. System Variable, Multilingual Variable and Web Application Module list all available for the current partition. If the drop down is used instead of this prompter, all WAMs which are currently open in the LANSA Editor are listed.

- Webroutine Name Prompter: In addition to picking a LANSA field, a System variable or a Multilingual variable, a webroutine's name can be picked from a list of available webroutines. Webroutine output lists all webroutines available to the current webroutine. System Variable, Multilingual Variable and Webroutines list all available for the current partition. If the drop down is used instead of this prompter all webroutines, which are available in the current or selected WAM are listed.
- **Length Prompter:** This prompter can be used to specify width or height values and is helpful to clarify units available to use.

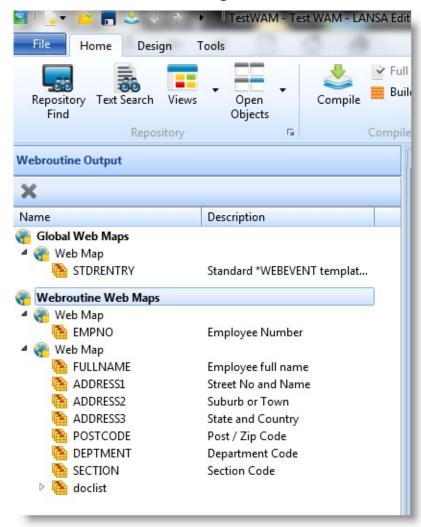


- **Image Prompter**: This prompter is useful when selecting an image. The HTML tag and several weblets use this prompter for image selection.
- **Color Prompter**: This prompter allows you to choose a color. Alternatively you can either type standard css color names or #rrggbb in the property's value field.



2.7.5 Webroutine Output

When working with a WebRoutine, the Webroutine Output tab displays the contents of the WAM's Web Maps (Global Web Maps) and the current Webroutine's Web Maps. Web maps are the interfaces for the exchange of data between a Webroutine and its Web Design.



This tab is useful for adding fields and lists to a Webroutine as there is no need to switch back and forth between the RDML source and the Web Page tab to manually add fields and lists. Once you are happy with your Web Design you can go back to the RDML source and write your supporting RDML code.

To add fields to a Web Map, drag fields from the repository and drop them onto an existing Web Map. To add a list to a Web Map, right mouse drag fields from the repository and drop them, either as an inlined list or a normal list, on any existing Web Map. To create a Web Map, drop on the Global Web Maps or the

Webroutine Web Maps entry.

You can delete entries from a Web Map by selecting a field or list and using the delete key. You can rearrange the Web Maps by dragging from one Web Map and to another. If you drop on the "Global Web Maps" or the "Webroutine Web Maps" entry a new Web Map is created.

You can drag and drop the fields/lists from the Webroutine Output tab to your Web Page. You can also drag and drop fields from the repository directly to the Web Page. In this case the Webroutine Output tab will be updated as well. An advantage of dropping a list on the Webroutine Output tab first is that you can manipulate the list, ie, add/delete/re-order fields, before you drop the list on your Web Page.

The RDML source is automatically when you drag and drop items. Note that when you drop a list, the DEF_LIST command is created in the RDML code, but is not deleted when the list is removed from the Webroutine Output tab.

To expand and collapse a Web Map or its contents, click on the plus or minus sign in front to the left the name in the Webroutine Output tab.

Note that changing Webroutine output details after fields and/or lists have been placed on the design will affect the integrity of the generated XSL and can produce errors in your design.

1 2.7 Web Application Modules (WAMs)

2.7.6 Create a Web Page

The following steps indicate how to make a new Web Design for a Webroutine based on an existing Web Design.

A different Web Design is useful when you need a specific Design for a different language or Technology Service. Note, though, that creating language-specific Web Designs for different languages isn't recommended as it increases your maintenance work. For different languages, use Multilingual Text variables instead as then only one Web Design will need to be maintained.

A new Web Design will always be based on the existing web design in the default language. You can then choose which webroutine, which Technology Service Provider to use and which the target language is.

You can create a web design based on an existing one by performing the following steps:

- 1. Open the WAM containing the webroutine that you want to create the new web page for.
- 2. In the Source tab, locate the WebRoutine command you wish to copy and click on the
 Webroutine Design glyph to the right of WebRoutine command.

```
Design Source Repository Details | Cross References |

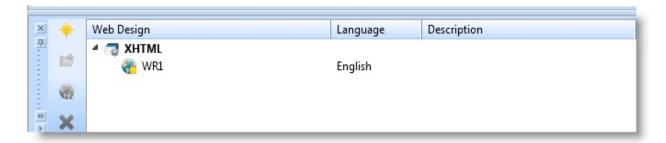
Function Options(*DIRECT)

■ BEGIN_COM ROLE(*EXTENDS #PRIM_WAM)
Web_Map For(*BOTH) Fields((#stdrentry *hide Group_By Name(#empdata) Fields(#EMPNO #SUR)

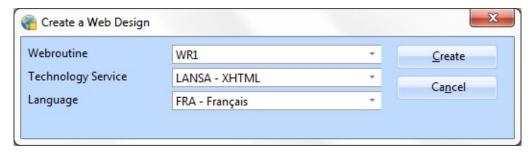
Def_List Name(#doclist) Fields(#df_elfnam)

■ WEBROUTINE NAME(WR)
```

The Web Design tab for the WebRoutine is displayed. If you have not previously created a Web Design for this WebRoutine, one will be created for you.



3. Go to the Web Design tab and use the Add command to open the Create a Web Design dialog.



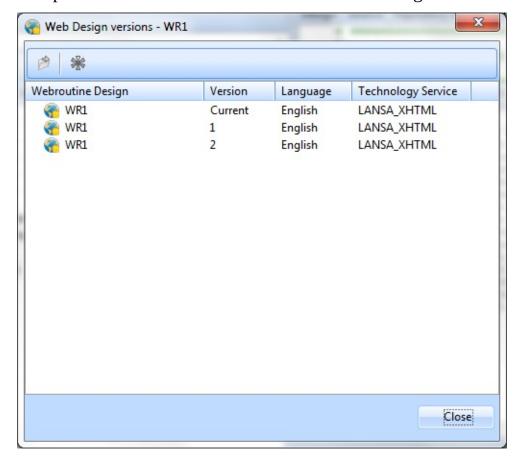
- 4. In the Create a Web Design dialog, select the WebRoutine for which you wish to create a web design, the Technology Service and target Language.
- 5. Press the Create button to create the web page.
- 6. You new web design will be added to the Web Design tab.
- ↑ 2.7 Web Application Modules (WAMs)

2.7.7 Design Versions

When working with a WAM, you may need to track the changes made to a web design. To see all versions for a web design, open a WAM, go to the Web Designs tab and select a design.



Select the Web Design you are interested in and press Show Web Design versions to open a list of versions for the selected Web Design.



Version

Each Web Design has a version number in the Version column. The current design is identified as **Current**. Older designs are identified with a number. The higher the version number the older the web design.

Press the Open Design icon to view a selected version or to change it in the Editor.

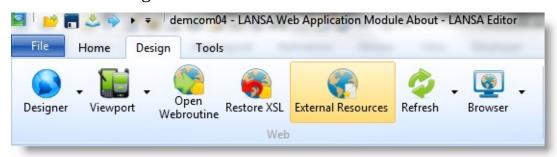
If you want to promote a previous version to become the current one, select it and press the Rollback icon. Note that if you rollback a version, it means adopting the selected version and discarding all later versions. For example, if you select version 2 and select Rollback, version 1 and the current version will be discarded.

↑ 2.7 Web Application Modules (WAMs)

2.7.8 Using External Resources with WAMs

When working with webroutines or weblets, you may need to use external resources in your web design.

To manage external resources in your web design, use the External Resources command in the Design tab of the ribbon:



In the Manage External Resources dialog box you can add, re-sequence or delete either Style or Script external resources for the WAM you have open in the Designer.

See the Web Application Modules (WAM) Guide for more information about using external resources in WAMs.

For information about how to enroll External Resources with Visual LANSA, refer to:

- Register Individual External Resources or
- Register Multiple External Resources in the Developer Guide.

Further Information

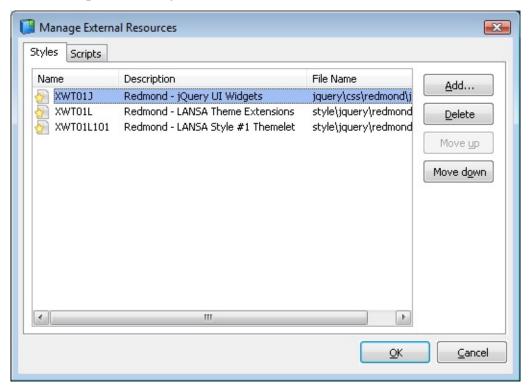
Style External Resources with WAMs Script External Resources with WAMs

Style External Resources with WAMs

In the Manage External Resources dialog box, select the Styles tab to see the current list of Style external resources planned for the WAM that you have open in the Designer.

Important Note:

The sequence in which the external resources are listed is the order in which they will be used when loaded in the resulting web page. The nature of Cascading Style Sheets makes the order particularly important as a style loaded later overrides a previous style.



Add... Opens the Repository Find dialog to show all available **Style** external resources for you to select file(s) to be added to this Design.

Delete Removes the selected Style external resource(s). If you select them all, then all will be deleted.

Move Moves the selected style external resource up in the list. **up**

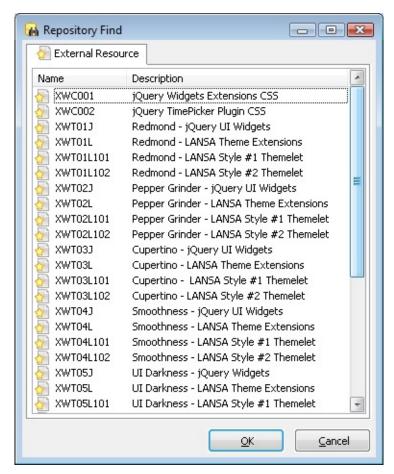
Move Moves the selected style external resource down in the list. **down**

OK Press when your list is complete and you are happy with the sequence of the external resources listed.

Cancel Makes no changes to an existing list.

Select Style External Resources

The Repository Find dialog shows a list of the External Resource that have been recorded as described in and External Resources.



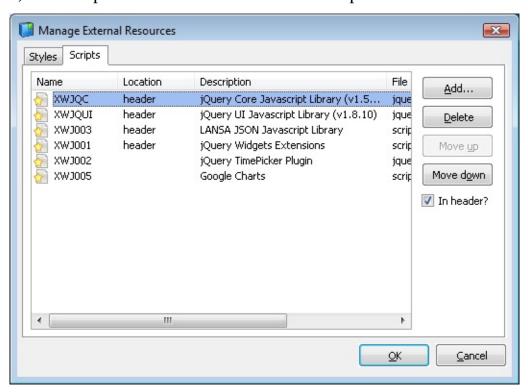
Select the desired Style external resources and press OK to go return to the Manage External Resources dialog.

Script External Resources with WAMs

In the Manage External Resources dialog box, select the Script tab to see the current list of Script external resources required for the WAM that you have open in the Designer.

Important Note:

The order in which the external resources are listed is the order in which they will be loaded in resulting web page. If script B depends on the contents of script A, then script A needs to be listed before script B.



Add... Opens the Repository Find dialog to show all available **Script** external resources for you to select files to be added to this Design.

Delete Removes the selected Script external resource. If you select them all, then all will be deleted.

Move Moves the selected script external resource down in the list.

Move Moves the selected script external resource up the list. **up**

In If a script is to be in the header, highlight it and select (tick) this

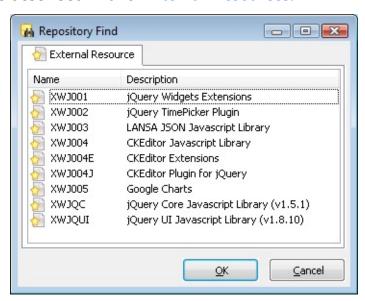
header? option. If a script is already defined as a header script, this indicator will be ticked. If you select more than one external resource, and some are defined as headers and some are not, this indicator will display intermittently.

OK Apply the changes to the current web design, which will be refreshed to reflect the changes.

Cancel Don't apply any changes.

Select External Resource Script

The Repository Find dialog shows a list of the External Resources that have been recorded as described in and External Resources.



Select the desired Script external resources and click OK to return to the Manage External Resources dialog.

2.8 Weblets

2.8.1 Create 2.8.3 Create Weblet 2.8.4 Replace Visualization Weblet Weblet Design 2.7.8 Using External Resources with WAMs
Weblet

Weblets are reusable components for common HTML functions and can be dragged and dropped onto your Web Designs.

WAMs are shipped with a set of standard Weblets for the Technology Services supplied. These Weblets provide powerful building blocks for you to start creating your WAM based applications.

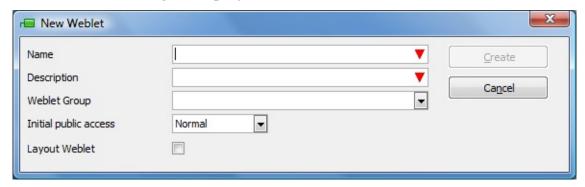
1 2. Editing Objects

2.8.1 Create Weblet

Note: Weblets can only be created in an RDMLX partition enabled for the web.

To create a Weblet use the New command.

The New Weblet dialog is displayed.



Specify the Weblet's Name, Description, its Weblet Group and whether it is a Layout Weblet. For further information of these entries, refer to Weblets in the Technical Reference Guide.

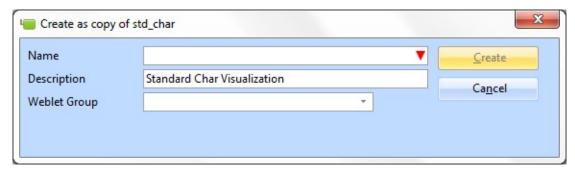
A Weblet Group provides a convenient way to group similar Weblets. A Weblet can be a member of more then one group. You can create additional groups and make Weblets a member of these new groups. When viewing a list of Weblets in the Repository, you can choose which groups to view. This will limit the list to only those Weblets belonging to that group and hence making it easier to get an overview of the available Weblets for a particular task. If a Weblet Group is not selected the new weblet will be added to the 'Unassigned' group.

Select the WAM Layout option if creating a layout weblet. You will be provided with the basic HTML document structure (html, head, body, script, style, etc.) to create your weblet.

1 2.8 Weblets

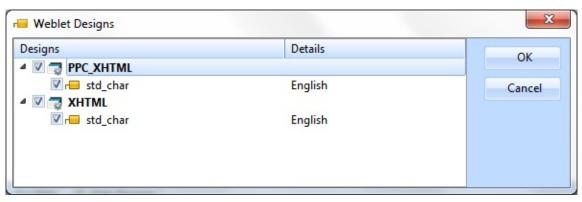
2.8.2 Copy Weblet

To copy a weblet, select an existing weblet in the Repository tab or the Favorites tab, right click to open the context menu and select Copy.



The Create as copy of ... dialog allows you to specify the Name, Description, and Weblet Group for the new weblet.

After you press the Create button, the Weblet Designs dialog will appear for you to select which designs you want to copy to the new weblet. Note that you always need to have a copy of a weblet in the default language. Therefore when you make a copy of a weblet, you have to copy the Web Design for the default language for the current Technology Service Provider.



1 2.8 Weblets

2.8.3 Create Weblet Design

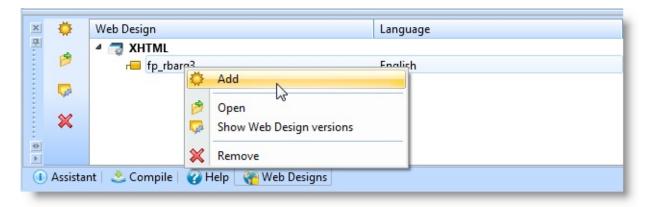
The following steps indicate how to create a Web Design for a Weblet.

A different Weblet Design is useful when you need a specific Web Design for another language. However, creating language-specific Web Designs is **not** recommended. Instead, use Multilingual Text variables for different languages as this will greatly reduce the maintenance work as only one Web Design will need to be maintained.

A new Web Design's default language will always be based on the existing web design. You can then choose which webroutine, which Technology Service Provider to use and which target language to use.

To create a web design based on an existing one, do the following:

Open the Weblet that is to be the basis for the new weblet. Use the Add command to open the Create a Web Design dialog.



3. Select the Technology Service and target Language for the new Web Design.



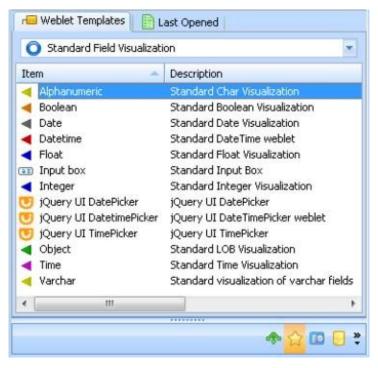
4. Press the Create button to create the Web Design. You can now either continue to work with the current Web Design or select to work with the newly created Web Design by opening it from the Web Designs tab.

↑ 2.8 Weblets

2.8.4 Replace Visualization Weblet

In a web design, LANSA fields can be represented as either an HTML <input>tag or be visualized as a weblet. If you choose to use visualization using a weblet, then you can choose from several field visualization Weblets.

At times you might want to replace a visualization weblet with another field visualization weblet, and retain the value of properties with a matching name between the original field visualization weblet and the new weblet.



The easiest way to replace one visualization weblet with another one is:

- 1. Select the weblet and open the context menu (right click) and select Replace with Input Field. This operation will replace the field visualization weblet with an input field, and remember the weblet properties of the previous weblet.
- 2. Drop the new field visualization weblet on top of the input field. This operation will use the remembered weblet properties from step 1, but only those properties with a matching name.
- 3. You are now ready to use your new field visualization weblet.

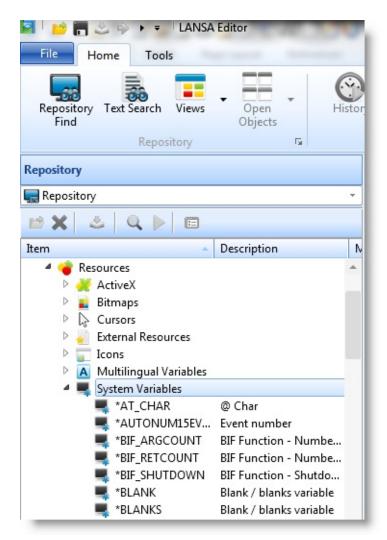
The WAM Editor will remember the weblet properties for a field as long you do not leave the web design and as long as the field created in step 1 remains unchanged.

↑ 2.8 Weblets

2.9 System Variables

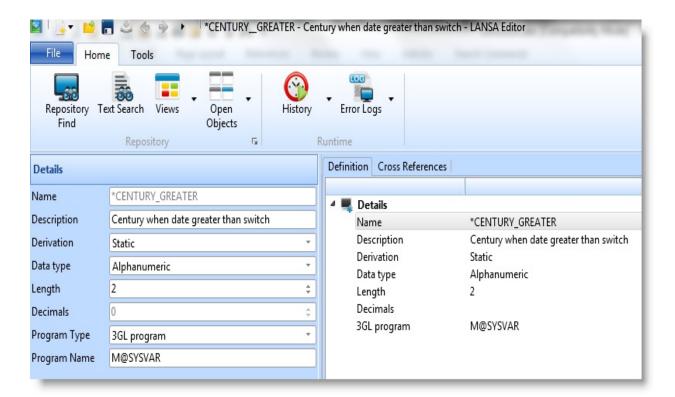
2.9.1 Create System Variables

System Variables are listed under the Resources heading in the Repository tab. Each system variable exists in all LANSA Partitions. If the value of a system variable is changed in one LANSA partition, then all partitions will use the new value.



To update a system variable:

- 1. Select the system variable in the Repository tab and open it in the editor.
- 2. Click on one of the items in the Definition tab, and the Details tab will be displayed so that you can amend the details.



For the specification of each entry in a System Variable, refer to System Variable Definitions in the Technical Reference Guide.

Also See

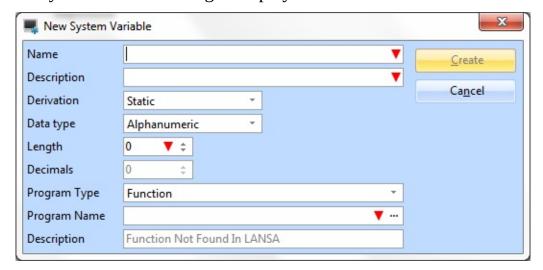
System Variable Concepts in the *Developer Guide*.

System and Multilingual Variables in the *Technical Reference Guide*1 2. Editing Objects

2.9.1 Create System Variables

To create a system variable use the New command.

The New system variable dialog is displayed:



Refer to System Variable Definition in the Technical Reference Guide, for details of the entries required on this dialog.

When the required details have been specified, press the Create button to create the new system variable.

When created, the new System Variable will be available to all LANSA partitions.

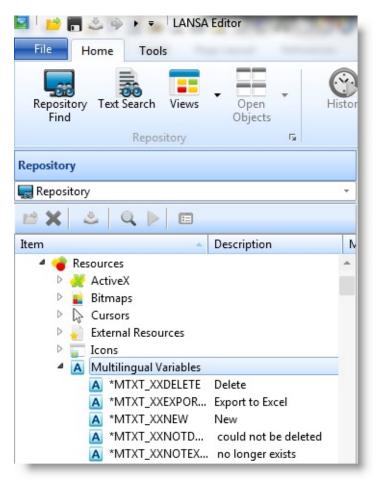
1 2.9 System Variables

2.10 Multilingual Variables

2.10.1 Create Multilingual Variables 2.10.2 Copy Multilingual Variables

Multilingual variables are specific to each partition and must be defined for each language in the required partition.

Multilingual Variables are listed under the Resources heading in the Repository tab.



Also See

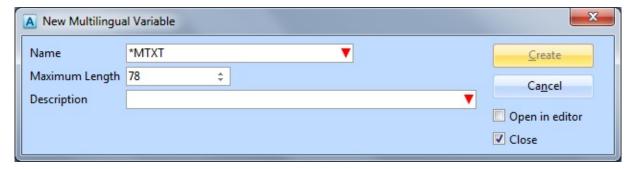
Multilingual Text Variable Concepts in the Developer Guide System and Multilingual Variables in the *Technical Reference Guide* Multilingual Variables in the Multilingual Application Design Guide.

1 2. Editing Objects

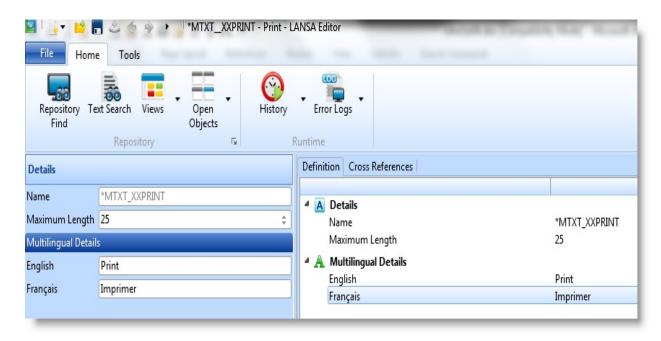
2.10.1 Create Multilingual Variables

To create a multilingual variable use the New command.

The New multilingual variable dialog is displayed.



- Enter the Name, Maximum Length, Description. For details of these fields, refer to Multilingual Text Variables in the Technical Reference Guide.
- Select the Open in editor option if you wish to specify additional details.
- Select the Close option if you do not want to create another multilingual text variable.
- When the required details have been specified, press the Create button to create the new multilingual variable. It will appear in the Multilingual list under the Resources heading in the Repository tab. You will find your Multilingual variables in the sequence of the first letter following the *MTXT.
- Single click on an item in the Definition tab to display the Details tab. In the Details tab you can add a description for each of the languages defined for the partition.
- The initial description will be used for each language in the Partition.



↑ 2.10 Multilingual Variables

2.10.2 Copy Multilingual Variables

To copy an existing multilingual variable, select one in the Multilingual list under Resources in the Repository or Favorites tab, right click to open the context menu and select the Copy option.



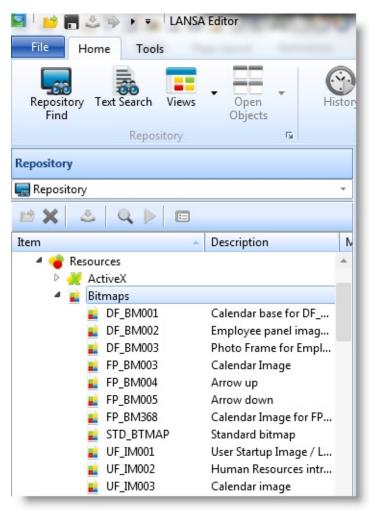
The Create as copy of ... dialog allows you to specify the Name, Length and Description for the new variable.

↑ 2.10 Multilingual Variables

2.11 Bitmaps

Source Tab Other Objects in the Design Tab 2.11.1 Create Bitmaps

You will find the Bitmap listed under the Resources heading in the Repository tab.



↑ 2. Editing Objects

2.11.1 Create Bitmaps

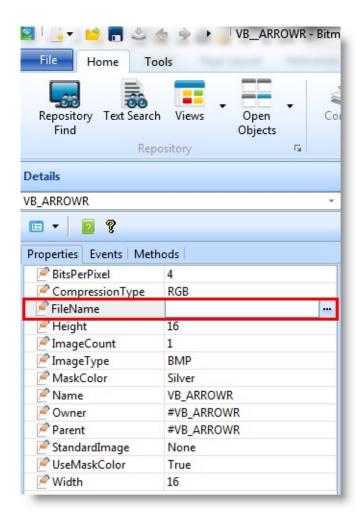
To register a new bitmap use the New command.

The New bitmap dialog is displayed:



Enter the Name by which you will be referring to your bitmap, and a Description. For details, refer to Component Definitions in the *Technical Reference Guide*.

You do not create a bitmap using Visual LANSA, but record the location of the bitmap in the Details tab. To open the Details tab for bitmaps, use the Views command in the ribbon and select Details.



Enter the name and location of your bitmap. The bitmap will be displayed in the Design tab.

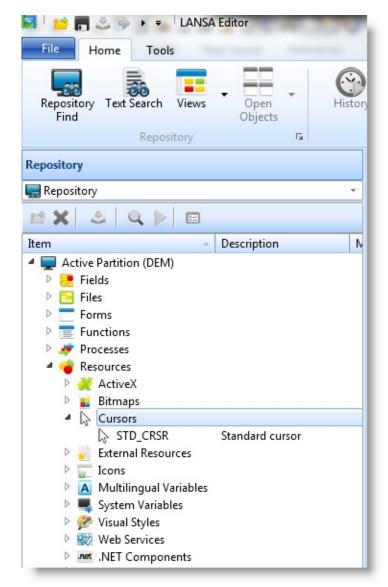
Also See

Components in the *Technical Reference Guide*Enrolling Bitmaps and Icons in the *Developer Guide*↑ 2.11 Bitmaps

2.12 Cursors

2.12.1 Create Cursors

You will find the Cursors listed under the Resources heading in the Repository tab.



A cursor cannot be edited in Visual LANSA. You are simply registering the definition in the Repository.

Also See

Source Tab

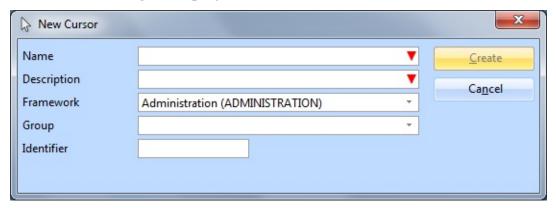
Other Objects in the Design Tab

↑ 2. Editing Objects

2.12.1 Create Cursors

To register a cursor use the New command.

The New cursor dialog is displayed:



- Specify the Name and Description of the cursor that is to be created. For details of these entries, refer to Component Definitions in the *Technical Reference Guide*.
- When the required details have been specified, press the Create button to create the new cursor.

The cursor will be opened in the editor.

You do not create a cursor using Visual LANSA, but record the location of the cursor in the Details tab. To open the Details tab for cursors, use the Views command in the ribbon and select Details.

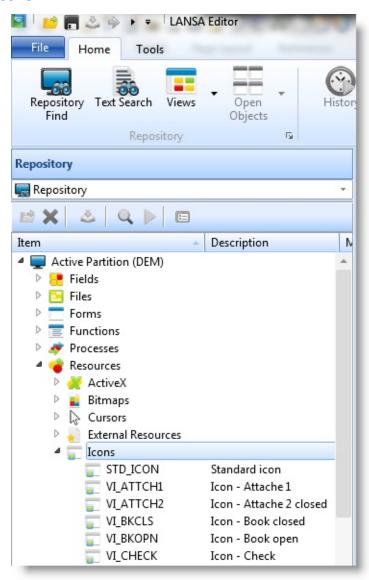
Also See

Components in the *Technical Reference Guide*Enrolling Bitmaps and Icons in the *Developer Guide*

1 2.12 Cursors

2.13 Icons

2.13.1 Create Icons



Also See

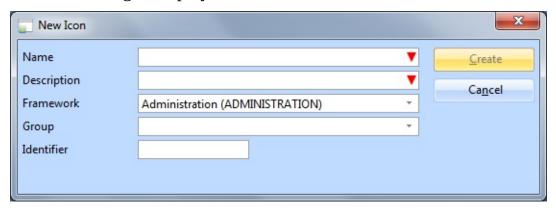
Source Tab
Other Objects in the Design Tab

↑ 2. Editing Objects

2.13.1 Create Icons

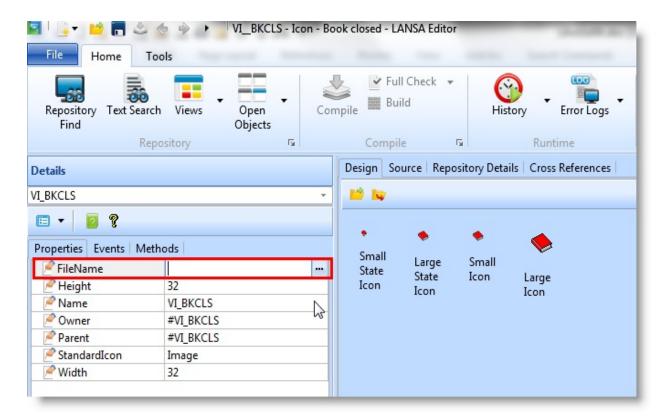
To register an icon use the New command.

The New icon dialog is displayed:



- Specify the Name and Description of the icon that is to be registered. For details, refer to Component Definitions in the *Technical Reference Guide*.
- When the required details have been specified, press the **Create button** to create the new icon.

You do not create an icon using Visual LANSA, but record the location of the icon in the Details tab. To open the Details tab for icons, use the Views command in the ribbon and select Details.



Also See

Components in the Technical Reference Guide.

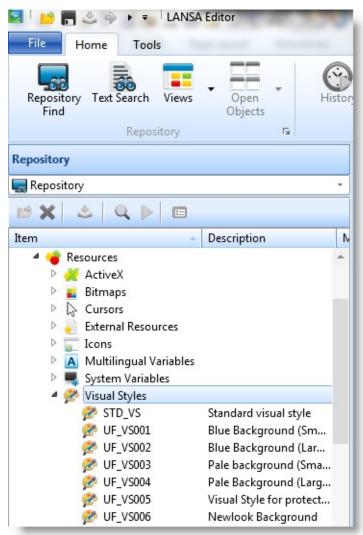
Enrolling Bitmaps and Icons in the Developer Guide.

↑ 2.13 Icons

2.14 Visual Styles

2.14.1 Create Visual Styles

You will find the Visual Styles listed under the Resources heading in the Repository tab.



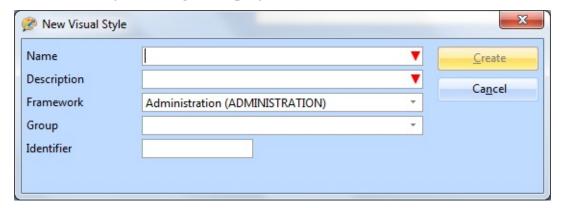
Also See

Source Tab
Other Objects in the Design Tab
Visual Styles in the *Developer Guide*↑ 2. Editing Objects

2.14.1 Create Visual Styles

To create a visual style use the New command.

The New visual style dialog is displayed:



- Specify the Name and Description of the visual style that is to be created.
 For details, refer to Component Definitions in the *Technical Reference Guide*.
- When the required details have been specified, press the Create button to create the new visual style.

The visual style will be displayed in the Design tab of the editor. Press the Details navigator bar to open the Details tab which shows details of your visual style. Visual styles do not have Repository help

Also See

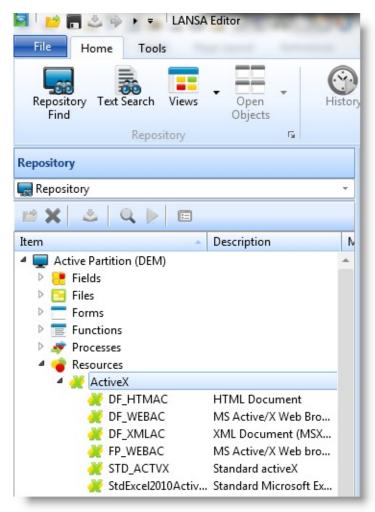
Visual Styles in the Developer Guide

↑ 2.14 Visual Styles

2.15 ActiveX

2.15.1 Register ActiveX Controls

You will find the ActiveX objects under the Resources heading in the Repository tab.



An ActiveX control is not edited in Visual LANSA. You are simply registering its definition in the Repository.

Also See

Source Tab

Other Objects in the Design Tab

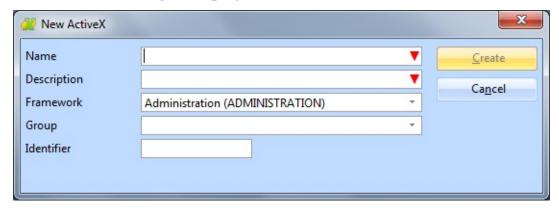
ActiveX Controls in the Developer Guide

↑ 2. Editing Objects

2.15.1 Register ActiveX Controls

To register an ActiveX control use the New command.

The New ActiveX dialog is displayed:



- Specify the Name and Description of the ActiveX control that is to be registered. For details, refer to Component Definitions in the *Technical Reference Guide*.
- When the required details have been specified, press the Create button to create the new ActiveX.

For details of how to register an ActiveX control, refer to ActiveX Controls in the *Developer Guide*.

Also See

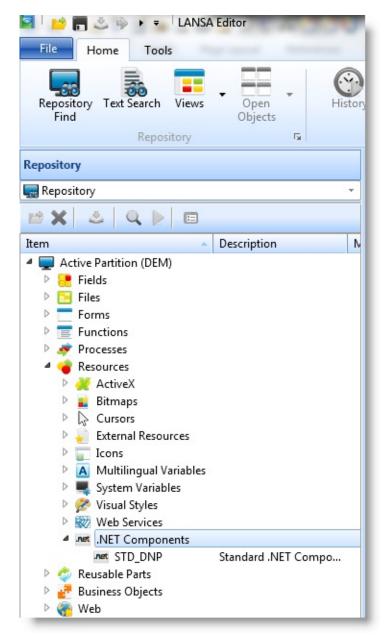
Components in the Technical Reference Guide

↑ 2.15 ActiveX

2.16 .NET Components

2.16.1 Register .NET Components

You will find.NET Components under the Resources heading in the Repository tab.



The .NET component is not edited in Visual LANSA. You are simply registering its definition in the Repository.

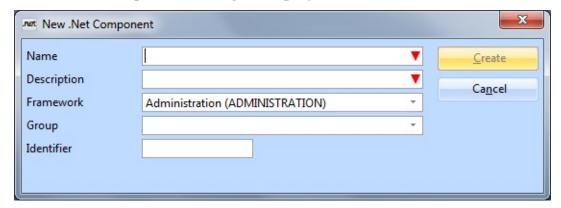
Also See

Source Tab
Other Objects in the Design Tab
.NET Components in the *Technical Reference*↑ 2. Editing Objects

2.16.1 Register .NET Components

To register a .NET component use the New command.

The New .NET Component dialog is displayed:



- Specify the Name and Description of the .NET component control that is to be registered. For details, refer to Component Definitions in the *Technical Reference Guide*.
- When the required details have been specified, press the Create button to create the .NET component.

For details of how to register a .NET component, refer to .NET Components in the *Developer Guide*.

Also See

.NET Components in the Technical Reference Guide

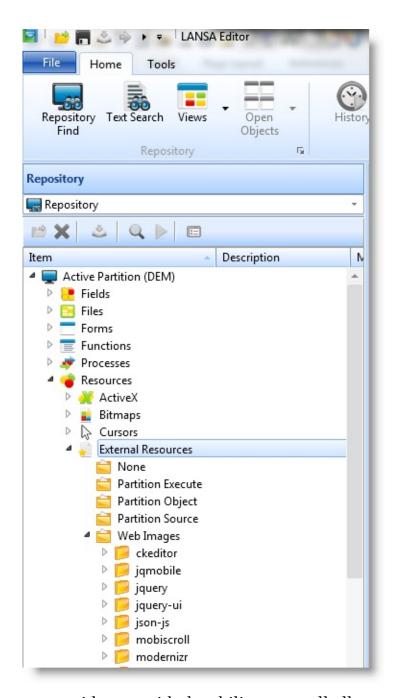
1 2.16 .NET Components

2.17 External Resources

2.17.1 Register Individual External Resources

2.17.2 External
Resource Definition
Tab

2.7.8 Using External Resources with WAMs



External Resources provide you with the ability to enroll all resources associated

with an application in the Repository. This makes it easier for you to package completed applications for deployment as well as to use external resources in your applications.

External resources include:

- Images
- Cascading Style Sheet files
- JavaScript files for a Web application
- Visual LANSA Framework /RAMP screen definition xml files

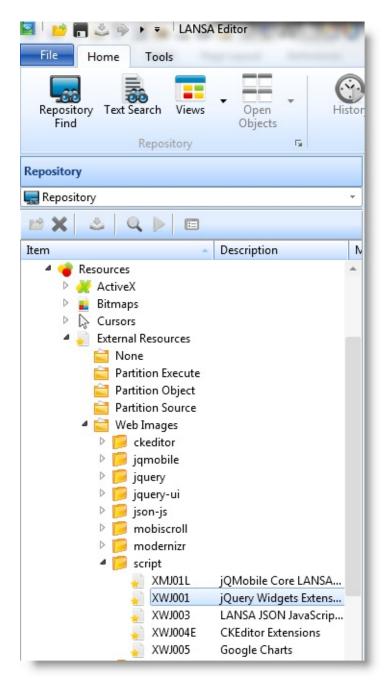
External Resource objects are stored internally within LANSA as BLOBs. They can be checked in and checked out, task tracked and deployed.

The Visual LANSA IDE detects when the physical file has changed so it can be re-saved, and it provides the ability to publish/recover files from the Repository to the file system.

External resources are used with LANSA for the Web to ship:

- Themelets
- jQuery and jQuery UI Libraries
- Cascading Style Sheets and JavaScript files associated with the jQuery Weblets.

External Resources are selected from the Repository tab or Favorites tab. Expand the tree view or double click on External Resources to open the list of External Resources available for this partition.



Note that you can view External Resources by their location on disk. Just right-click to open the context menu and select Group by Location.



Also See

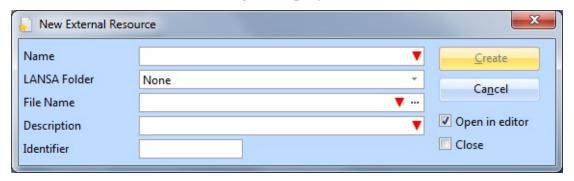
External Resources in the Technical Reference Guide

↑ 2. Editing Objects

2.17.1 Register Individual External Resources

To register an individual External Resource use the New command, and choose External Resource.

The New External Resource dialog is displayed:



- An External Resource needs a Name and Description.
- The LANSA Folder and File Name specify the file you wish to register as an External Resource in the LANSA repository. Refer to External Resource External Resource Definitions for details of these entries.
- Select the Open in editor option if you wish to specify additional External Resources.
- Select the Close option if you do not want the New External Resources dialog to reappear.
- When the required details have been specified, press the Create button to register the new External Resource.

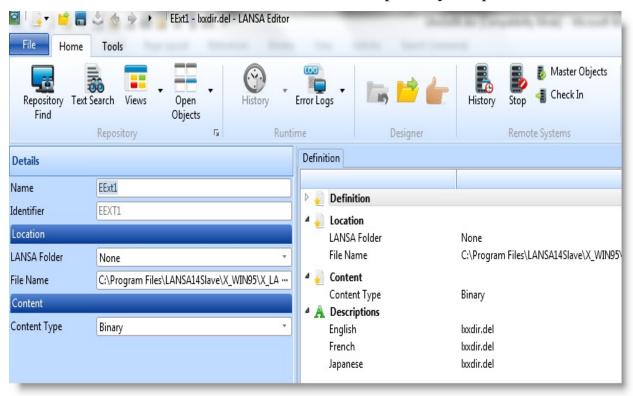
Also See

External Resources in the Technical Reference Guide
Register Multiple External Resources in the Developer Guide

1 2.17 External Resources

2.17.2 External Resource Definition Tab

Double click on the External Resource in the Repository to open its definition.



The details in the Definition tab are split into these groups:

Location

The location of the External Resource file. Specify a path relative to a known LANSA folder or choose None to select any file. Note that it only really makes sense to register files within the known LANSA folders if the goal is to be able to share or deploy them.

Content

Indicates whether the file is Binary or Text. For Text files it also describes the encoding. See the Technical Reference Guide for more details.

Descriptions The multilingual descriptions of this External Resource

The following commands can be used with external resources:

Extract



Replace the file on disk with the version stored in the repository.

Open file in External Editor



Use Windows to open an Editor for this file.

Go to file location



Open Windows Explorer to show the file on disk.

Note on editing External Resources

Opening an External Resource in the LANSA Editor means you're editing its Repository definition. You must open the file in an external editor to edit its contents.

After you save your changes in the external editor, the file will be show as **modified (*)** in the Definition Tab.



You can now click Save to save the modified contents to the LANSA Repository.

Alternatively, if the version on disk is different from the one stored in the Repository, you can choose Extract to replace it with the Repository definition.

Also See

External Resource Definitions in the *Technical Reference Guide*1 2.17 External Resources

2.18 Lists

2.18.1 List Types 2.18.2 Create Static 2.18.3 Create Lists Dynamic Lists

2.18.4 Save Results of 2.18.5 Create Impact

Search as List Analysis Lists

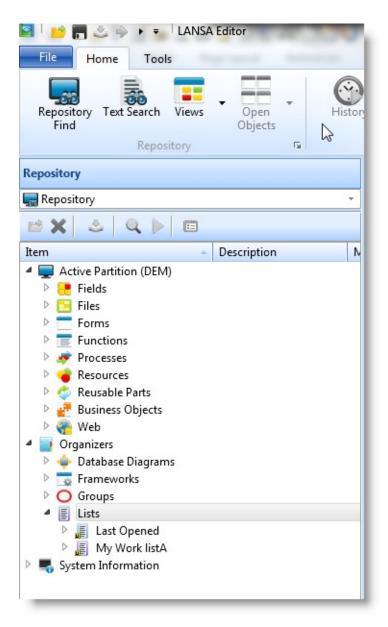
You can create a List:

• Using the New command

• Using Save results following a Find operation

There is also an automatically generated list the Last Opened, which you will find as a tab in the Favorites folder.

You will find your Lists under Organizers in the Repository tab.



Each list is represented by an icon.

- The blue icon indicates a Static list
- The purple icon indicates a Dynamic list.
- The orange icon indicates an Impact list.

A small **u** in the bottom left corner of the icon indicates a **User** list. To delete a list, highlight it in the Repository tab's list and use the Delete command. A list must be closed before it can be deleted.

2.18.2 Create Static Lists

- 2.18.3 Create Dynamic Lists
- 2.18.4 Save Results of Search as List
- ↑ 2. Editing Objects

2.18.1 List Types

Static List

A static list will only contain the objects you specifically include in it.

Dynamic List

A dynamic list will include any existing and new items that satisfy the inclusion criteria.

For example if you have a dynamic list which contains all fields starting with letter A, and you create a field ADDRESS4, the field will be automatically added to the list. Similarly, a dynamic list may contain all objects that belong to a Task ID or a user, or all objects that are created after a specified date.

You can add as many criteria as you want. Objects that correspond to any of the criteria you specify will be included. In other words the criteria are joined with an OR, not with an AND.

Impact List

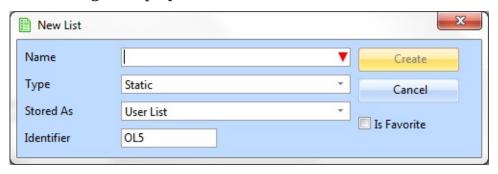
You use Impact Analysis (after creating an Impact List) to examine the consequences of a particular change to a LANSA object or objects in the LANSA Repository. You do this using one or more searches using specific search criteria. When you execute the search, a list will be produced selecting the objects that may be impacted by a change. The lists can be exported to an Excel or CSV file.

2.18.2 Create Static Lists

A static list only contains the objects you specifically include in it..

You create a list using the New command or using the Save Results command in the Find Results toolbar.

The New List dialog is displayed:



Specify the

Name

The name of this list must be unique.

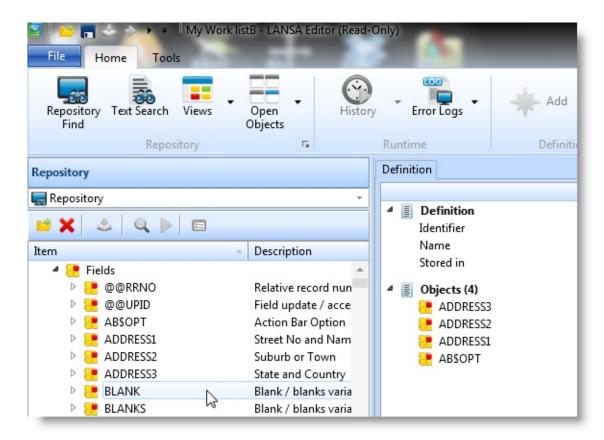
List

Select **Static. Type**

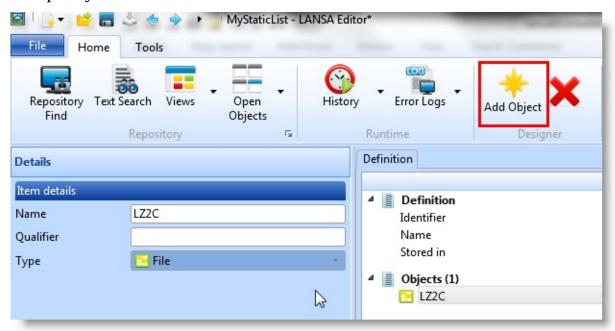
Stored as User list (specific to the user and the computer) or repository list (available to any user accessing this repository).

ü Select this option if you wish it to be added to the Favorites tab. **Is Favorite**

Press Create to create the list. You can either drag and drop objects from the Repository tab to the list:



Or you can use the Add Object command to add individual objects to the list and specify their details in the Details tab:



Note that the Qualifier applies only to some objects such as the library of a file, the second half of a long mtxt name, a function's process and so on.

↑ 2.18 Lists

2.18.3 Create Dynamic Lists

Once a Dynamic list has been created, any new objects that fit its selection criteria will be automatically added to the list.

Create the list using the New command. The New List dialog is displayed:



Specify the

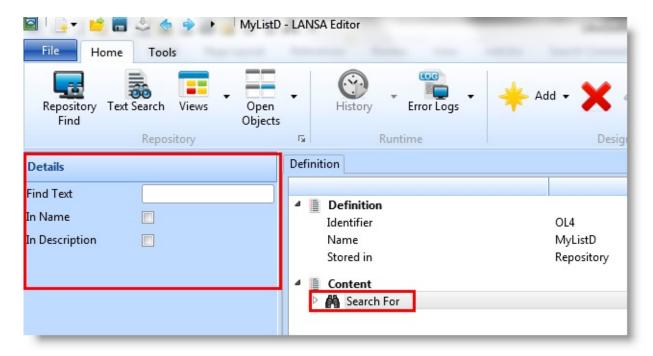
The name of this list must be unique. **Name**

Type Select **Dynamic.**

Stored as Repository list or portable user list.

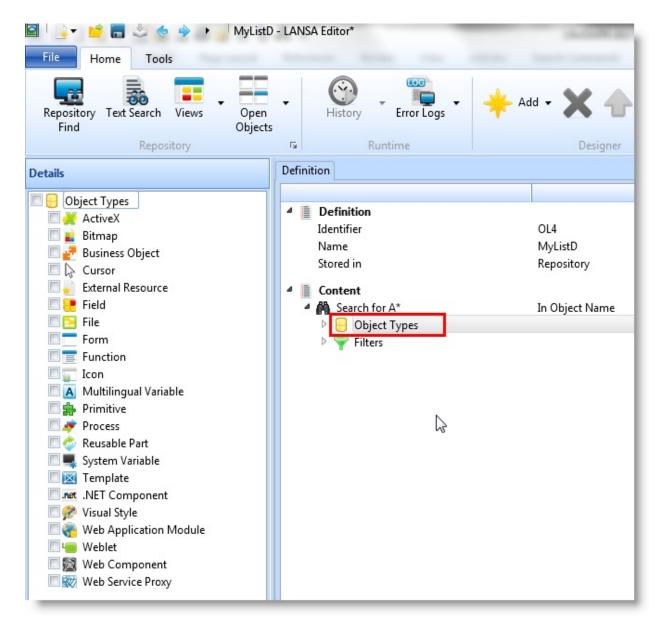
Is Favorite ü Select this option to add the list to the Favorites tab.

After the list is created, you specify selection criteria for the list. In the open list double-click Search For to display the Details tab:

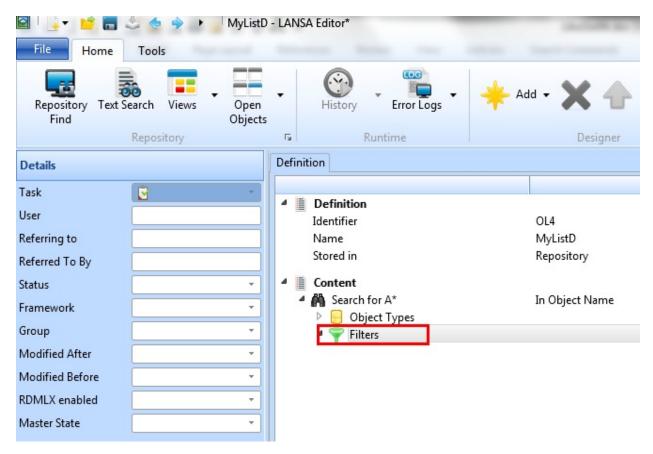


Specify the text to be used as the search criteria in the Find Text field. The text is searched in the object name and/or the object description.

Expand Search For and double-click Object Types to display the Details tab to specify the object types to be included:



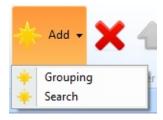
If you want to add filters to the list, double-click Filters:



Save the list.

For details of the possible selection criteria, refer to Repository Find.

Use the Add command to add additional searches or groupings in the search:

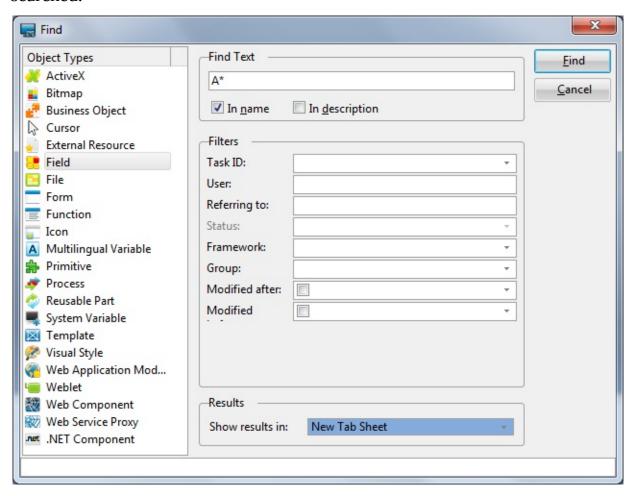


Groupings are sub lists to an existing list, which allow you to group selected items separately. For example, you may wish to separately track all the fields, files and forms for a particular Task ID.

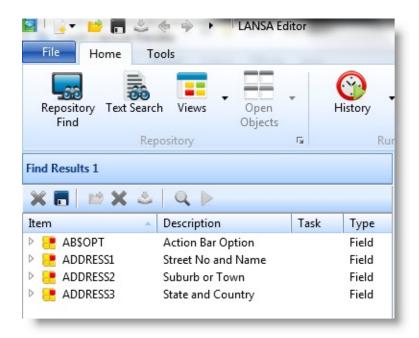
↑ 2.18 Lists

2.18.4 Save Results of Search as List

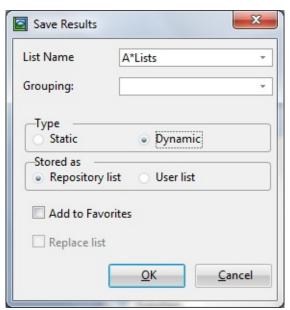
You can save the results of a search to a list. Here fields starting with A* are searched.



And the results are shown in Find Results:



When you click Save results, the Save Results dialog will open so that you can specify the details of the new list:



Specify the

List The name of this list.

Name

Grouping You can create subgroups in the list by specifying a grouping

Type Refer to 2.18.1 List Types for details of this entry.

Stored as Repository lists are available for any user accessing the same

Repository, a user list is specific for the user and the computer on

which it is created.

Add to Ü Select this option if you wish it to be added to the Favorites tab. **Favorites**

Replace ü Select this option if you want to replace an existing list with this

list list's contents.

Press OK to create the list.

↑ 2.18 Lists

2.18.5 Create Impact Analysis Lists

To create an impact list (see 2.18.1 List Types) use the New command, and then List. In the New List dialog, enter the name of the list, select **Impact** as the type, and press the Create button.



Specify the:

Name The name of the list must be unique.

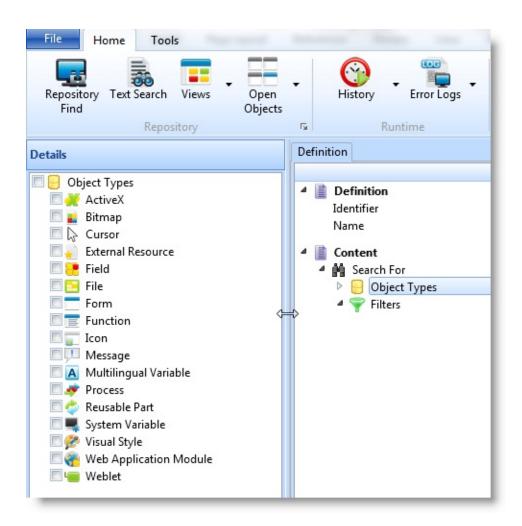
Type Select Impact. For more information about this list type, refer to 2.18.1 List Types.

Stored as Impact lists are always stored as repository lists (available to any user accessing this repository).

The object's short name. Use for reference only. **Identifier**

Is Is Favoriteü Select this option if you wish it to be added to the Favorites tab, otherwise leave it blank.

After the list is created, you specify selection criteria for it:



Also see

Add and Remove searches
Change the Impact List's Definition

↑ 2.18 Lists

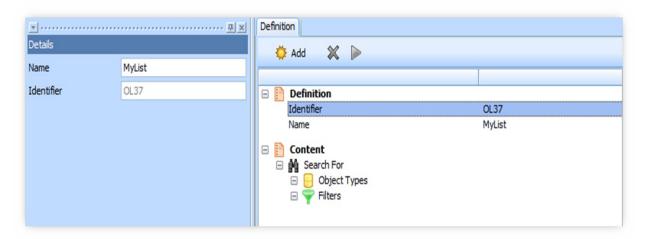
Add and Remove searches

You can add as many searches as you want by using the Add Object command. You can remove searches by selecting the search to remove and pressing the Delete key or using the Delete command.

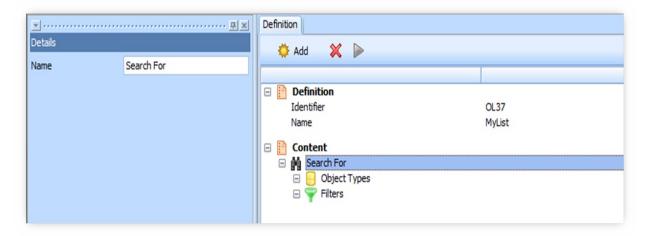
↑ 2.18.5 Create Impact Analysis Lists

Change the Impact List's Definition

When created, the list is displayed in the LANSA Editor. By clicking on various parts of the list the details are displayed in the Details tab where they can be changed. Here the Identifier details are displayed.



Click on Search For to change the name of your search.



Also see

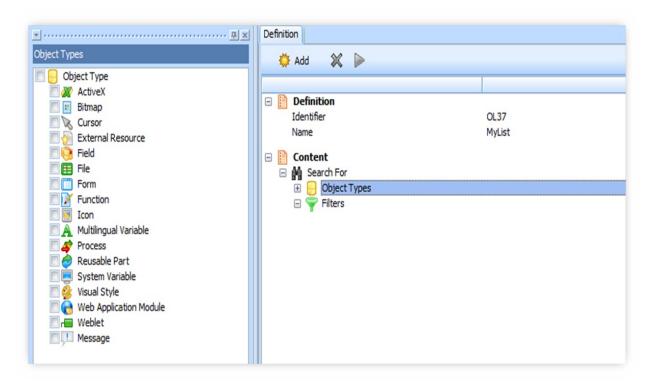
Add Object Types
Specify Filters for your Objects
Save and Run an Impact List
Review the Results
Export the Impact List

↑ 2.18.5 Create Impact Analysis Lists

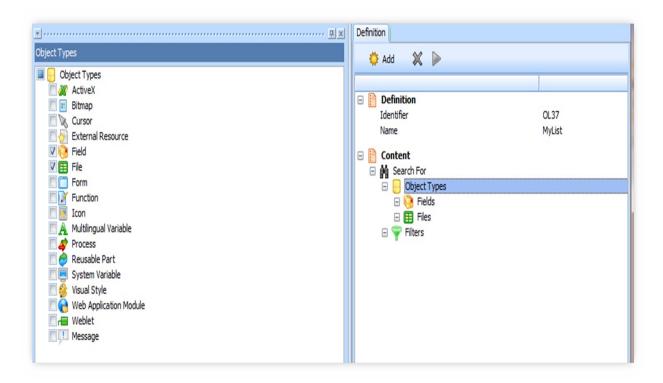
Add Object Types

Specify the objects to be included in the search.

Double click on Object Types. A list of Object Types will be displayed in the left pane. Tick the box for each Object Type to be included in your search. The logical relationship between all the object types you select is a logical *OR



For example if you tick Fields and Files, the Impact Analysis will find all the fields and all the files match in your criteria. These are listed in the Definition:



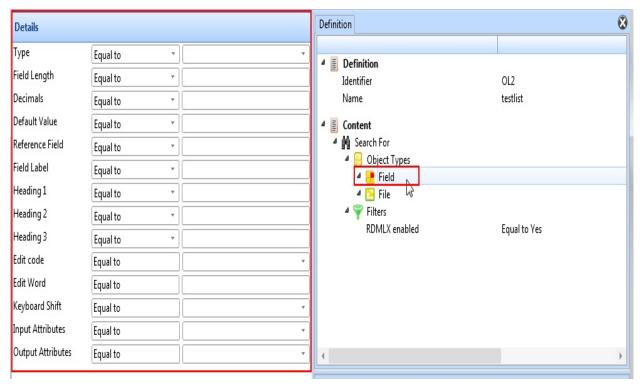
↑ Change the Impact List's Definition

Specify Filters for your Objects

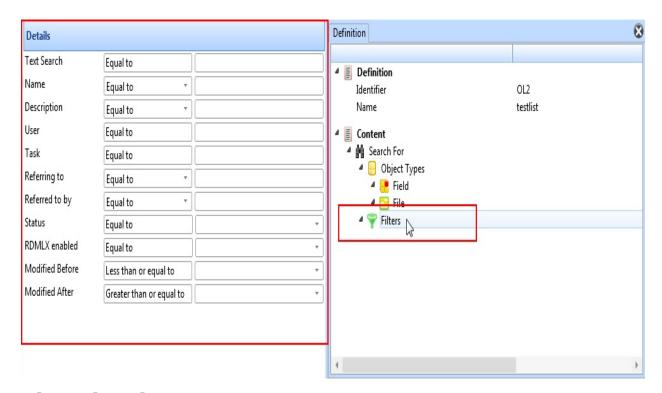
Filters are used to define inclusion criteria for the impact list.

Some filters apply to all the object types selected, others only to a single object type, such as the File Type filter which only applies to files.

For filters that are specific to an object type, select the Object Type in the Definition and the available filters will be displayed:



For filters that are common to one or more object type, such as Name and Description, select the Filters node:

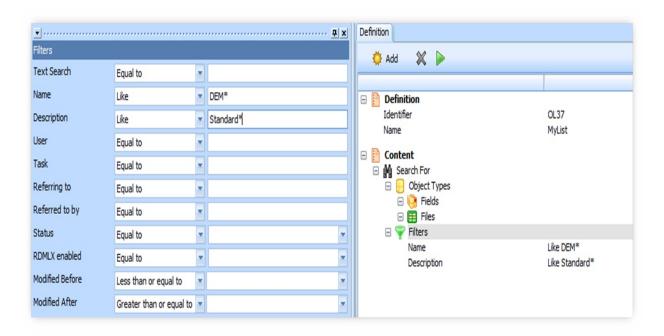


About the Filters

- The fields with a blank value are inactive. Thus to deactivate a filter, remove any value and/or assign its drop down value to *blank.
- The wildcard to use with the Like operator is *.

 For example if filter Name is like DEM, the Impact Analysis would produce all the fields starting with DEM.
- The operators Equal to, Greater than, Less than, Less than or equal to, Greater than or equal to treat * as a character and the value strictly equal for all filters except Text Search.
 - For example if filter Name is equal to MYTEXT, the Impact Analysis would only produce the object called MYTEXT. If filter name Description is equal to My Test Field *, the Impact Analysis would only return the object with a description exactly the same as My Test Field *, otherwise nothing is returned.
- The operator Equal to for filter Text Search will remove all objects which do not contain the value specified.
 - For example if filter Text Search is equal to **Define** for the Form Object Type, the Impact Analysis will return all the forms in which RDML code lines contain Define.

The logical relationship between filters is a logical AND.
 For example, as shown below, if the Filters Name is DEM* and the Description is Standard*, the Impact Analysis List will include all the fields and files with the name starting with DEM, and with a Description starting with Standard*.



The exception to these rules is the Text Search filter which will return all objects containing any lines of source code with the search value.

- For the object type Multilingual Variables:
 - Filter Text Search will search in every Multilingual Variable
 - Filter Name will filter the search by Multilingual Variable Name
 - Filter Description will filter the search by looking in the Multilingual Variable description, but only in the current language.
- For the object type Message:
 - Filter Text Search will search for message in every language,
 - Filter Name will filter the search by Message ID
 - Filter Description will search in the message text, but only in the current language.
- The Virtual Filter in Files searches for all files which have a:
 - Virtual field and this Virtual field is equal to the one specified in the filter when operator is Equal To.

 Virtual field and this Virtual field is not equal to the one specified in the filter when operator is Not Equal To. 	

Save and Run an Impact List

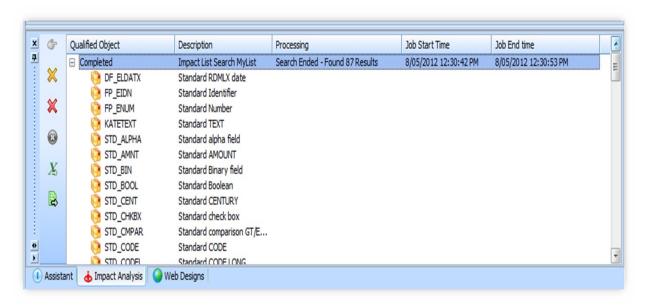
To **Save the impact list**, use the Save command. Alternatively, if you run the impact list it will be automatically saved.

Note that the Run command is only enabled when there is at least one object type selected and one filter active.

↑ Change the Impact List's Definition

Review the Results

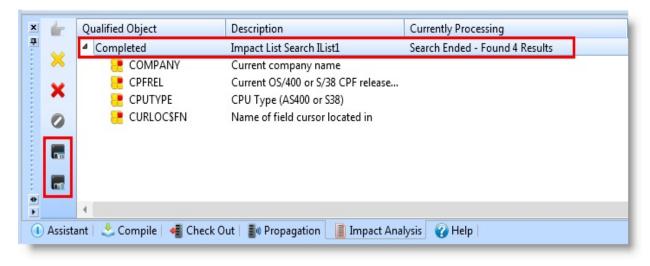
The results are displayed in the Impact Analysis output view.



↑ Change the Impact List's Definition

Export the Impact List

To export the impact list, select the job heading In the Impact Analysis output view:



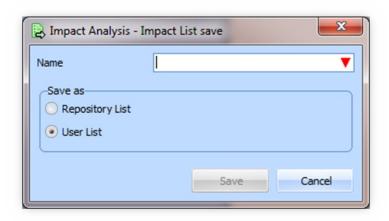
You can export the list of objects produced by the impact analysis. You can:

• Export to Excel

Click on the export to Excel icon. The list of LANSA objects will be exported to a CSV file.

• Export to a Static List

Click on the Export as Static List option. A dialog is opened where you can choose the name and the type of list.



↑ Change the Impact List's Definition

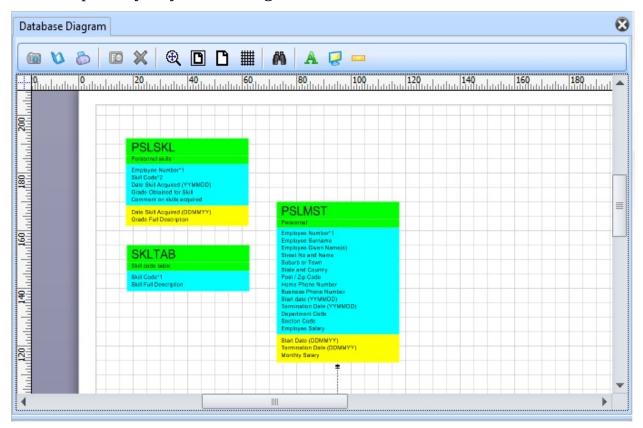
2.19 Database Diagrams

2.19.1 Create a Diagram Visual LANSA Logical Modeler

Prerequisite: In order to use database diagrams you need Microsoft Visio 2010 or later.

Database Diagrams provide an up-to-date graphical view of the file definitions that are stored in your LANSA Repository.

The diagrams are stored in the LANSA directory on your PC. Therefore, database diagrams that you create are not visible to others who may be using the same Repository as you are using.



You can place any physical file anywhere on the diagram by dragging and dropping from the editor's Repository tab. The diagrams show relationships between files defined by access routes.

You have the option of adding related files when manually adding files to a diagram.

To open the definition of a file shown on the diagram, right-click and choose

Open file for edit from the context menu.

You can save diagrams for later viewing or as a separate, standalone Visio document for use in documents and presentations.

Files in Database Diagrams

The Database Diagram shows physical files that reside in the LANSA Repository. This is how the Personnel Skills file PSLSKL from LANSA Demonstration Personnel Application is shown on the diagram:

Each file has three portions:

- At the top, a green section which shows the name and description of the file
- In the middle, a blue section shows the descriptions of all real fields belonging to the file. The asterisks after the first two field descriptions indicate that the fields are primary keys of the file. In addition, the number after the asterisk shows the key positions of the fields in the file.
- At the bottom, a yellow section shows the descriptions of all virtual fields belonging to the file.

In addition, the diagram shows the relationships between physical files as defined by access routes defined for the files. These are depicted by lines which join the file shapes. The lines have different shapes at their ends according to the properties of the access routes, which specify the type of relationship. For example:



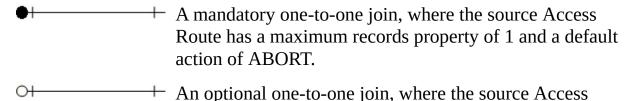
The example shows the relationship between the Personnel and Personnel Skills files. Note that the Personnel end of the line has a single line through it and the Personnel Skills end of the line has three prongs. This denotes that there is a one-to-many relationship between the Personnel and Personnel Skills files, as defined by their access routes.

The different types of line end shapes are discussed later in this guide.

Access Routes in Database Diagrams

Access route definitions are the means by which the diagram viewer determines that files are related. These relationships are shown as lines between file shapes on the diagram.

The end of the connecting line shows the type of relationship depicted:



Route has a maximum records property of 1 and a default action of IGNORE.

A one-to many, parent/child relationship, where the source Access Route has a maximum records property of 9999.

A relationship with Access Route properties that differ from those above.

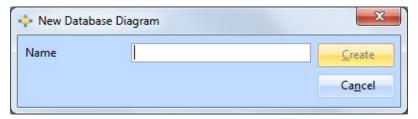
Note that when Access Route definitions in both the source and target files of the relationship match, the connecting line is colored black on the diagram. This denotes a fully defined relationship between the two files.

When there is not a matching Access Route, the connecting line is colored pink on the diagram. This denotes a partially defined relationship between the two files.

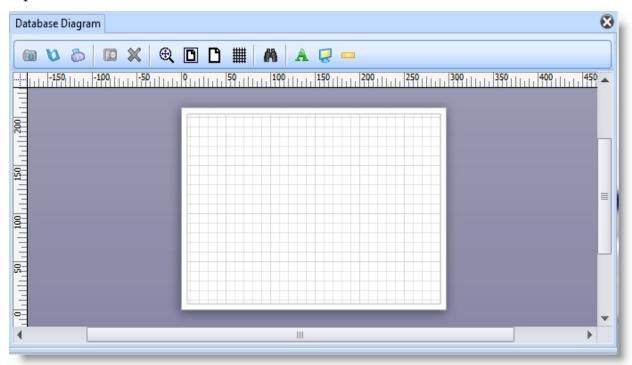
1 2. Editing Objects

2.19.1 Create a Diagram

To create a database diagram, use the New command. The Create Database Diagram will appear:



Enter a name for the diagram and press the OK button. A blank diagram is opened:

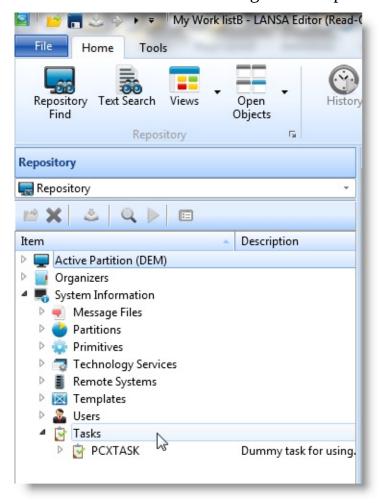


↑ 2.19 Database Diagrams

2.20 Tasks

2.20.1 Create a Task Tracking Configuration

You will find tasks under the Resources heading in the Repository tab:



The concept of task tracking is to create a Task Identifier that is used to control and record changes to individual objects in the LANSA development environment. Once a developer logs on with an assigned Task ID, the Task ID can be used to check authority to objects, lock objects, log of activity on objects, etc. Refer to Task Tracking Concepts in the Administrator's Guide for information about defining task tracking for your site.

1 2. Editing Objects

2.20.1 Create a Task

A new task can only be created if the Visual LANSA System Type is a master system. To create a task, use the New command and choose Task.

The New Task dialog is displayed:



Specify the task Name and Description as described in Task Tracking Configuration in the Administrator's Guide.

Also see

Task Tracking in the Administrator's Guide.

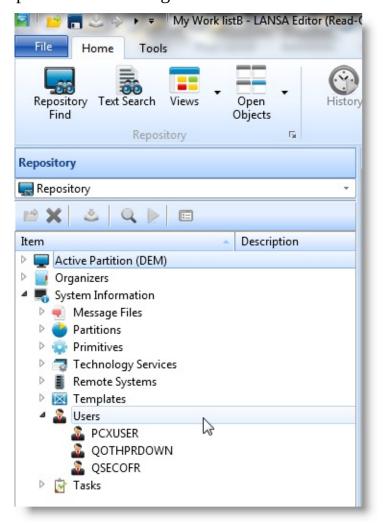
↑ 2.20 Tasks

2.21 User Definitions

2.21.1 New User 2.21.2 User Privileges Tab

You will find the Users listed under the System Information heading in the Repository tab.

Before you set up a number of users, it is recommended that you set up some user groups so that the privileges adopted by the new users can be obtained from a user group rather than having to be entered for each user.



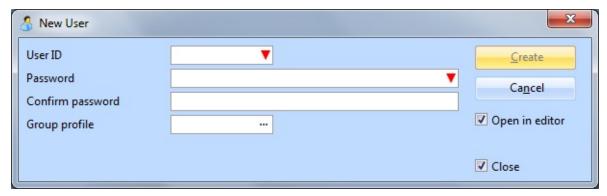
Also See

User and Security Maintenance in the *Administrator Guide*↑ 2. Editing Objects

2.21.1 New User

A new user can only be created if the Visual LANSA System Type is a master system. To create a user use the New command.

The New User dialog is displayed:



User ID Enter the User Identifier.

Password and Confirm Password

Enter the password for the User Identifier. The password can be up to 256 characters and numbers. Note that the case you use is retained.

Group Profile

Enter the Group to which this user will belong. See Group Profile in the *Administrator Guide*.

Open in editor

Select this option if you wish to specify the user's privileges before you continue with the next user.

- Allocating a user to a Group will save time when setting up the user's privileges.
- When the required details have been specified, press the Close button, and the new user will be displayed in the Repository's User list.
- Select a user in the list to display that user's details. If you double click on one of the items in the Definition tab, the Details tab will open so that you can make changes if required.



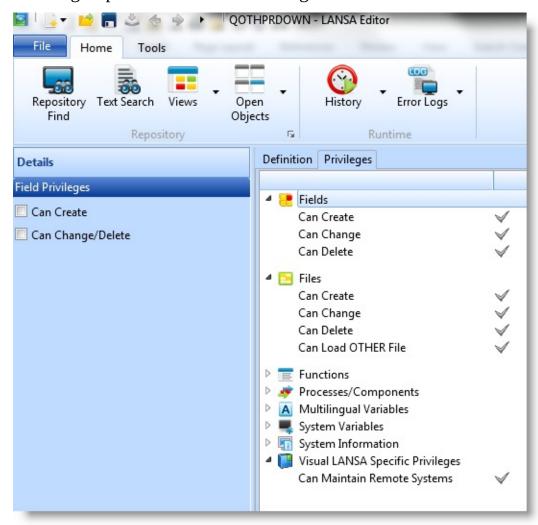
Also See

Object Maintenance Concepts in the Administrator Guide.

↑ 2.21 User Definitions

2.21.2 User Privileges Tab

Select the Privileges tab to specify privileges the user has for different object types and operations in the Visual LANSA environment. The default selection is based on the group to which the user belongs.



Select each object type in the Privileges tab one by one, double click to open the Details tab where you can change the privileges.

Also See

Object Maintenance Concepts in the Administrator Guide

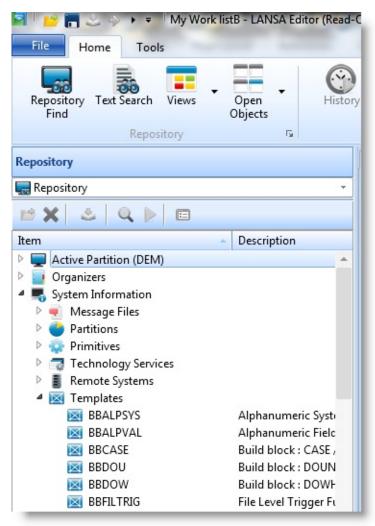
↑ 2.21 User Definitions

2.22 Templates

2.22.1 Create a Template 2.22.2 Copy a Template 2.22.3 Template Help

LANSA application templates are one of its most powerful features. A template is a fraction of code. When executed, templates will automatically create standard blocks of RDML code based on a question and answer session. Templates are extremely important as they ensure consistency in function design, operation and structure. Templates increase productivity and reduce coding errors.

You will find existing Templates listed under System Information in the Repository tab.



↑ 2. Editing Objects

2.22.1 Create a Template

To create a template use the New command.

The New Template dialog is displayed.



Specify the new template's Name, Description and Extended Description.

For details of these entries, refer to Template Definition in the Administrator's Guide.

Press the Create button when the entries are complete. The new template will be listed under System Information in the Repository tab.

Also see

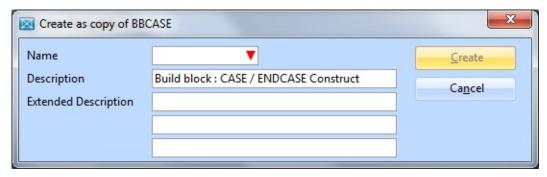
Tips for Template Programming in the Technical Reference Guide.

Application Templates in the Developer Guide.

↑ 2.22 Templates

2.22.2 Copy a Template

Select the Application Template that you wish to copy in the Templates list in the Repository tab. Right click to open the context menu to display the Create as copy of <template name> dialog.



Enter the Name, Description and Extended Description, then press the Create button to create the new template. For details of these entries, refer to Template Definition in the Administrator's Guide.

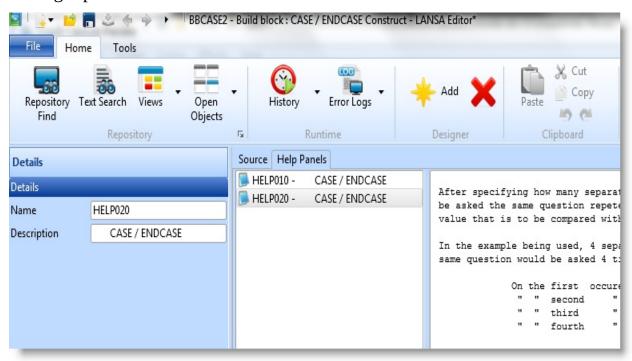
Press Create to create the new template. It is displayed in the Editor.

↑ 2.22 Templates

2.22.3 Template Help

Use the Help Panels Tab to create the help text that is displayed on request when the template is executed.

In the Details tab, enter a unique name for the new help text so that you can easily access it again, enter a Description and then type in the actual help text in the right pane.



For more information, refer to the HELPIDS parameter on the @@QUESTION command.

↑ 2.22 Templates

2.23 Partition Definitions

2.23.1 Create 2.23.2 Partition Definition 2.23.3 Language Settings

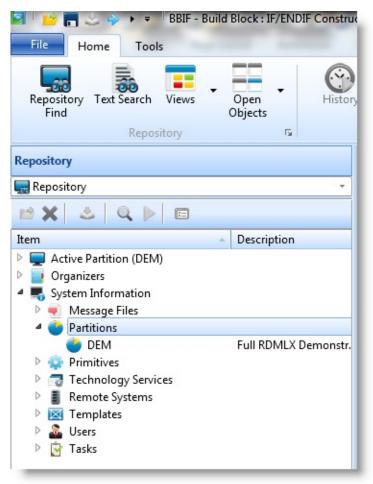
Partitions Tab Tab

2.23.4 Frameworks 2.23.5 Groups Tab

Tab

A system partition is a means of dividing up or "partitioning" one LANSA system. Each partition is completely separate from other partitions. The Partition Definition can only be edited if the Visual LANSA System Type is a master system.

You will find the Partitions listed under the System Information heading in the Repository tab.



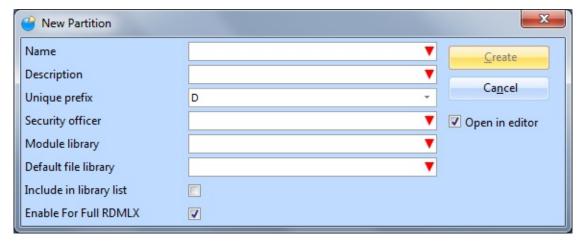
For details about Partitions, refer to Partitions in the Administrator Guide.

↑ 2. Editing Objects

2.23.1 Create Partitions

A new partition can only be created if the Visual LANSA System Type is a master system. To create a partition use the New command.

The New partition dialog is displayed:



- Specify the Name, Description and all other details required for the partition that is to be created. For a description of the entries on this dialog, refer to Partitions in the *Administrator Guide*.
- When all the details have been specified, press the Create button to create the new partition.
- Use the Partition Definition tab, Language Settings tab, Frameworks tab, and Groups tab to set the partition characteristics.

For a description of all the entries in this dialog, refer to Partition Definition Tab in the *Administrator Guide*.

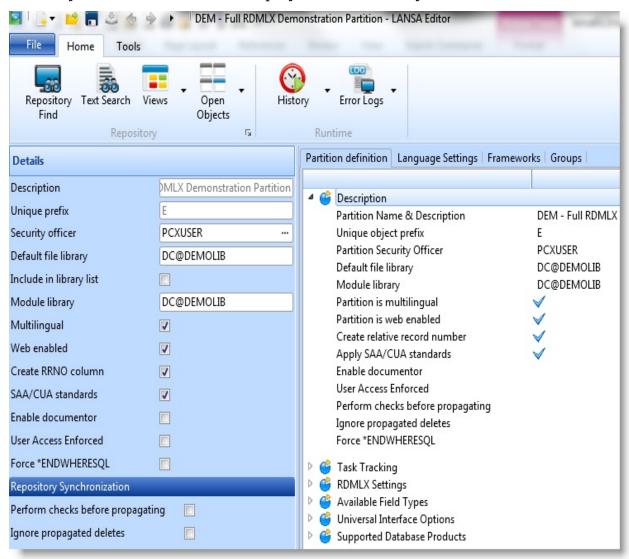
1 2.23 Partition Definitions

2.23.2 Partition Definition Tab

Select the Partition definition tab to review the partition's characteristics including:

- Description
- Available field types
- Task tracking
- Universal interface options
- RDMLX settings
 Supported database products.

Only Master or Independent systems allow partition details to be changed. For Slave systems, the details are displayed in Read-Only mode.



For a master or independent system, open the Details tab to set individual

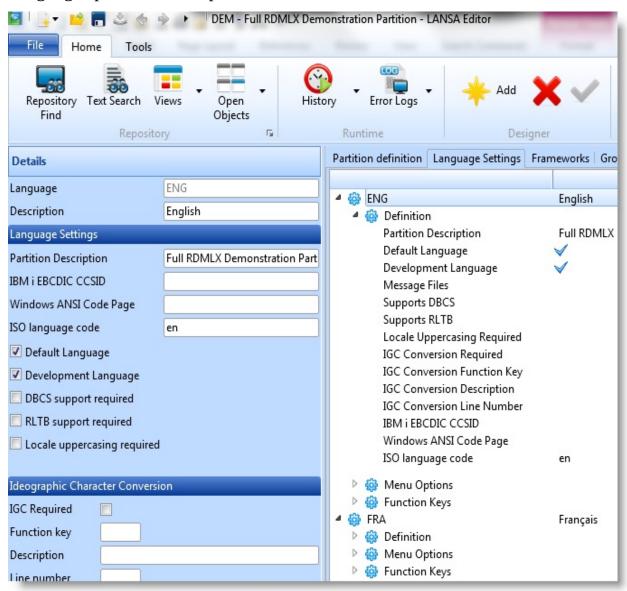
characteristics. To do this, double click on an entry in the group to be changed. For ***ENDWHERESQL**, refer to the **SELECT command's Options parameter** in the Technical Reference Guide for how to use this option.

Also See

Partition Definition Tab in the *Administrator Guide*↑ 2.23 Partition Definitions

2.23.3 Language Settings Tab

Select the Language Settings tab to work with the Languages settings for each language specified in the partition.



Add a language



Use the Add command.

Edit a language

Select the language and double click on it to display the Details tab (double-click) containing the options you can change.

Set a language as the default Remove a language



Select the Set as Default command.

Select the language and use the Delete command.

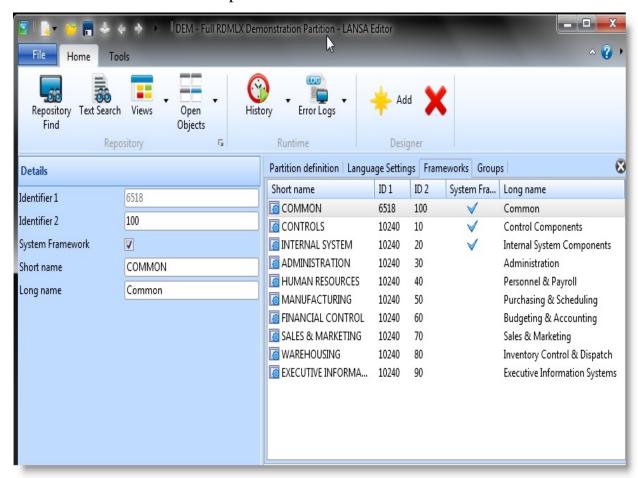
Also See

Language Settings in the Administrator Guide.

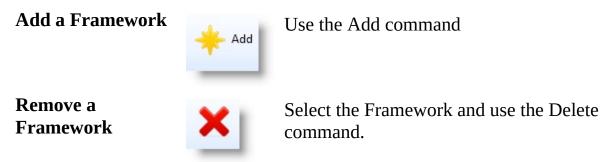
↑ 2.23 Partition Definitions

2.23.4 Frameworks Tab

Select the Frameworks tab to work with the Frameworks. You can add and remove Frameworks in the partition.



Please note that Frameworks are user-defined groupings of objects. Frameworks and the Visual LANSA Framework are not the same.



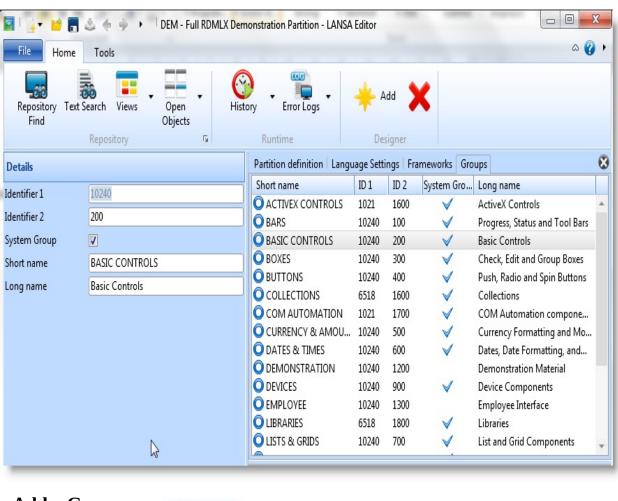
Also See

Frameworks in the Administrator Guide

↑ 2.23 Partition Definitions

2.23.5 Groups Tab

Select the Groups tab to work with the Groups. You can add and remove Groups in the partition.



Add a Group



Use the Add command

Remove a Group



Select the Group and use the Delete command.

Also See

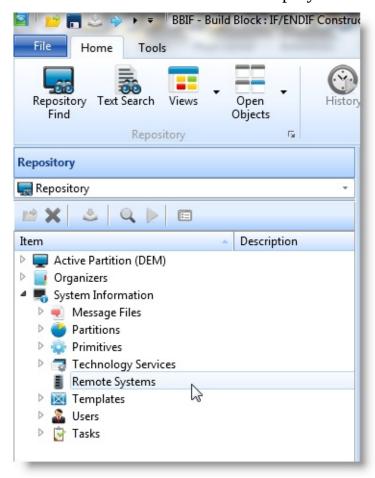
Groups in the *Administrator Guide*

↑ 2.23 Partition Definitions

2.24 Remote Systems

2.24.1 Create Remote System 2.24.2 System Definition Tab

A Remote System identifies a LANSA master or deployment environment.



- A Remote System defined as a master system is used for check in, check out and other host monitor related activities. The master Remote System details are automatically generated during installation of the Visual LANSA system.
- A Remote System defined as a deployment system is used to deliver objects to the remote system.

The Remote System name relates directly to the LANSA Communications Administrator LU Partner.

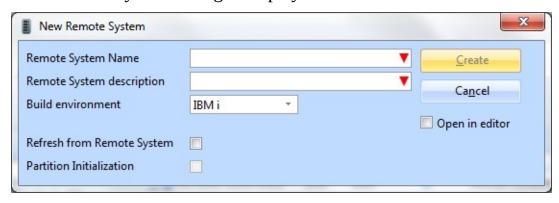
↑ 2. Editing Objects

2.24.1 Create Remote System

If this is a slave system, an entry for the IBM i Master is automatically added during system initialization.

You can manually create a Remote System for Deliver To using the New command.

The New Remote System dialog is displayed:



Specify the:

Remote This must correspond with the LU Partner defined in the

System LANSA Communications Administrator for this system. This

Name is not validated.

Remote A description to help identify the Remote System.

System Description

Build Either IBM i or Linux

Environment

System

Refreshü Select this option to refresh workstation names from a **from Remote**remote system of type IBM i (only). If you don't do this remote system of type IBM is (only).

remote system of type IBM i (only). If you don't do this refresh at the same time as creating the remote system, simply right click on the actual Remote System in the Repository tab of the LANSA Editor (to open the context menu) and select Refresh.

Partition Initialization

ü For a Linux Remote System, select this option to initialize the partition. The partition must be initialized before the remote system can be used.

If you don't initialize the partition at the same time as creating the remote system, right click on the actual Remote System in the Repository tab of the LANSA Editor (to open the context menu) and select Initialize.

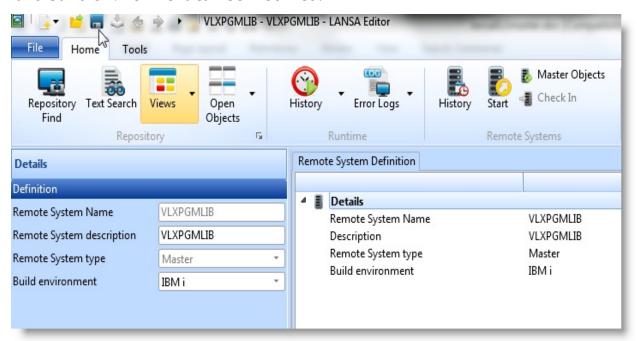
When you press Create, the Remote System definition will be saved. The Remote System will only be available to deliver to the Repository after restarting Visual LANSA.

2.24.2 System Definition Tab

Select the System Definition tab to view the definition of a Remote System.

The Remote System definition for a Master installation is for display only.

For a Remote System defined as a Deployment installation only the description and build environment can be modified.



Also See

Remote Systems in the Administrator Guide

1 2.24 Remote Systems

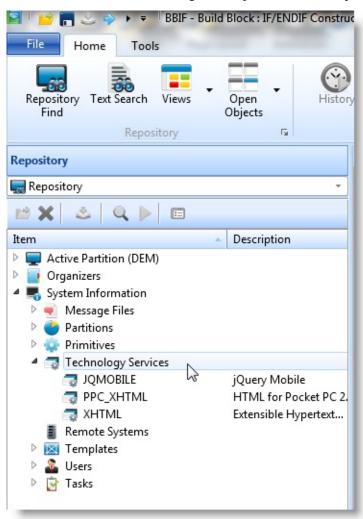
2.25 Technology Services

2.25.1 Technology Services Definition Tab Add and Remove searches

A Technology Service is a presentation or XML format used by a WAM to interact with a user agent or other XML-aware application. LANSA supplies Technology Services for XHTML and Pocket PC HTML.

Note: To update a Technology Service you must be in an RDMLX partition and be authorized to maintain System Settings.

Technology Services are listed in the Repository tab under System Information.



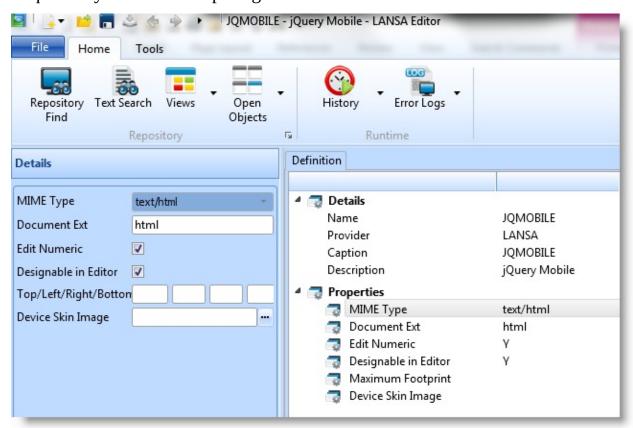
Double click on a Technology Service to view its definition in the Definition tab.

Also See

Technology Services in the *Technical Reference Guide*↑ 2. Editing Objects

2.25.1 Technology Services Definition Tab

Click on an item in the Definition tab to display the Details tab. In the Details tab you can define the presentation requirements of the technology services required by different computing devices.



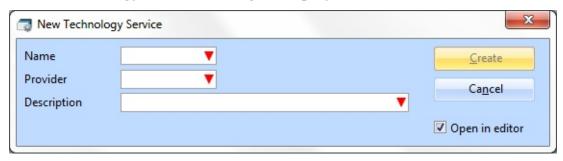
↑ 2.25 Technology Services

2.25.2 Create Technology Service

Note: To create a Technology Service, you must be in an RDMLX partition and be authorized to maintain System Settings.

To create a technology service, use the New command and choose Technology Service.

The New Technology Service dialog is displayed:



Specify the technology service Name, Provider name and Description required.

Also See

Technology Services in the *Technical Reference Guide* for details of the entries required.

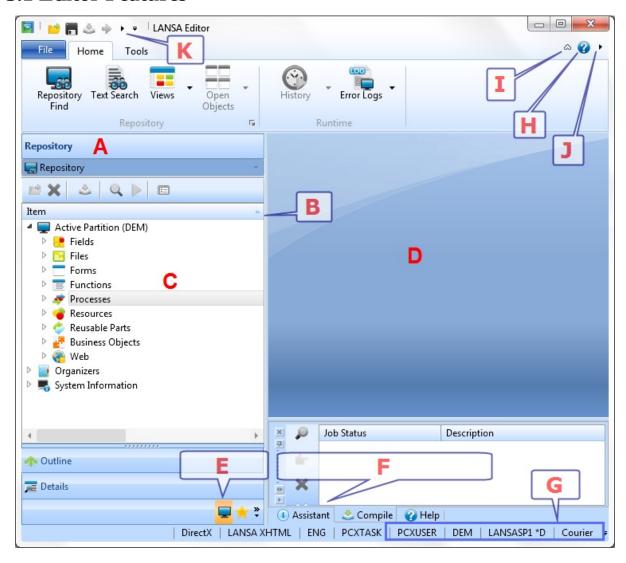
↑ 2.25 Technology Services

3. Standard Editor Tabs

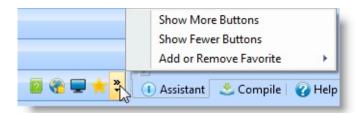
3.1 Editor Features	Editor Basics		
3.2 Repository Tab	3.7 Details Tab	3.11 Debug Tabs	3.15 Propagation Tab
3.3 Favorites Tab	3.8 Source Tab	3.12 Compile Tab	3.16 Cross References Tab
3.4 Outline Tab	3.8.5 Assistant Tab	3.13 Check In Tab	3.6 Features Tab
3.5 Go To Tab	3.9 Design Tab	3.14 Check Out Tab	3.17 Web Design Tab
3.10 Repository Help Tab			

[↑] Visual LANSA User Guide

3.1 Editor Features



- **A Title bar**. Use it to move or undock tab.
- **B** Control the order of items listed in the tab.
- C Press F1 to obtain Help for an item in the tab.
- **Work area.** Format depends on the opened object.
- Views. Views are shown as buttons (as Outline, Details and Go To in the image above) or collapsed into icons. You can control how many buttons are shown:



- **F** Context-sensitive tabs. These tabs show information about the object being worked with.
- G Edit settings in the status bar. Click on these setting to change them.
- **H** Help buttons.
- I Hide/show ribbon.
- J Help menu:



K Version 12 menus:



Also See

Setting up your Workspace

↑ Visual LANSA User Guide

3.2 Repository Tab

3.2.1 Repository Tab 3.2.2 Information about 3.2.3 View Object

Contents Objects Properties

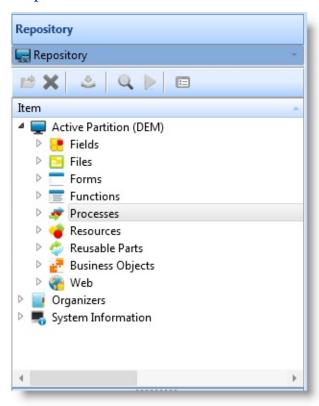
3.2.4 Alphabetical 3.2.5 Object Cross-

Groupings References

The Repository tab is used to maintain objects in the current repository and all the objects within the active development partition.

The Repository tab can be displayed by pressing F8 or by using the View command on the ribbon.

Control over the appearance and location of the Repository tab is described in Setting up Your Workspace.



Repository Toolbar

Open Selected object.
Refer to Editing Objects.

Delete fromDelete selected object from the repository.

Refer to Delete Objects.

Repository

Compile Compile selected executable object.

Refer to Compile Options. This gives selectable compile

options.

Debug Debug selected executable object.

Refer to Debugging Applications.

Execute Execute selected executable object.

Refer to Executing Applications.

ChangeToggles the display of the Repository tab contents between alphabetic grouping, grouping in dropdown and all objects.

alphabetic grouping, grouping in dropdown and all objects.

Only applies when a specific object type (not the whole

repository) is selected. Refer to 3.2.1 Repository Tab

Contents.

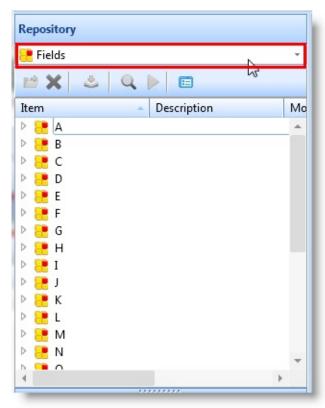
1 3. Standard Editor Tabs

view

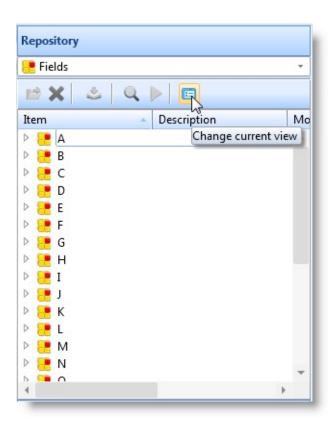
3.2.1 Repository Tab Contents

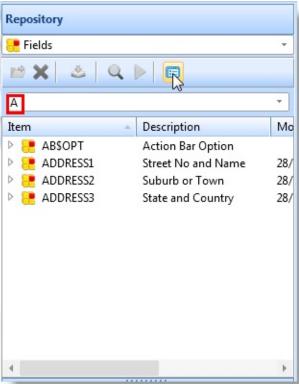
By default all objects in the repository are shown in the Repository tab.

You can choose to only display objects of a particular type using the object type drop-down in the tab:



When a single object type is displayed, you can use the Change current view button in the toolbar to toggle the repository view between alphabetic grouping, grouping in dropdown and all objects:



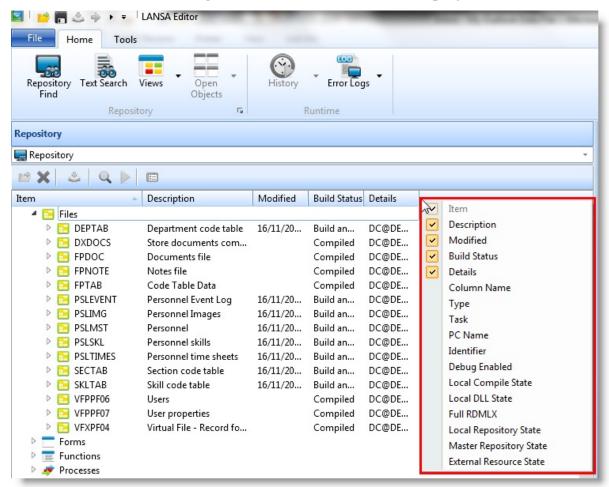


3.2.2 Information about Objects

The object information displayed will depend on the type of Object selected and the Visual LANSA System Type of your installation. For example, a Visual LANSA Slave System will include object information pertaining to the status of the LANSA for iSeries Master System.

You can sort by a specific column by clicking on the column's header. By default, objects are sorted by the Item name.

You can show and hide columns by right-clicking a column header to open a context menu and selecting which columns are to be displayed:



You can rearrange the columns by clicking on a column header and then dragging and dropping the column to the position you want.

Also See

Lists

↑ 3.2 Repository Tab

3.2.3 View Object Properties

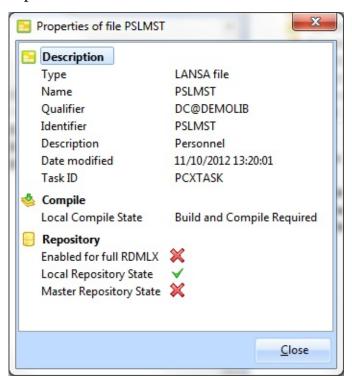
To view the definition of an object:

You can open it in the editor

or

• You can use the quicker way of looking at it in the Properties dialog.

To open the Properties dialog, select the object, right click to open the context menu and select Properties.

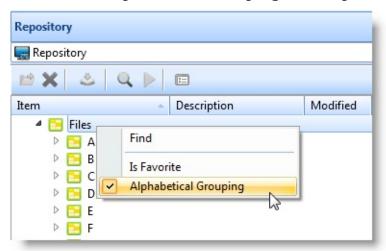


↑ 3.2 Repository Tab

3.2.4 Alphabetical Groupings

By default, specific types of objects in the Repository tab are grouped alphabetically to control the access to potentially large list of objects. For example, you can list objects in the Active Partition in alphabetic groupings because there may be thousands of them.

To turn off the alphabetical grouping, select a type of object in the tree list, right-click and deselect the Alphabetical Grouping menu option.



The objects in the selected node are then displayed as one list without alpha groupings.

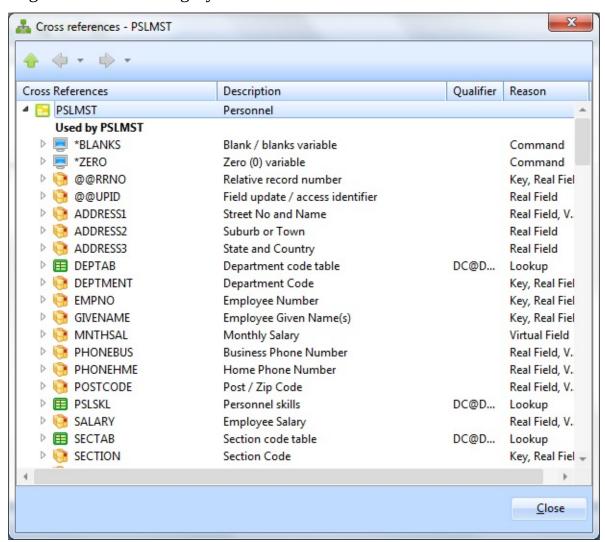
Notes:

- Turning off the alphabetical groupings for a node means that there may be a delay the very first time that node is opened.
- This option only applies to nodes that have alphabetical groupings. For example Groups, Frameworks and lists do not contain alphabetical groupings.
- An alternate way of viewing a selected group of objects is to define an Editor List with specific criteria for inclusion of objects.

↑ 3.2 Repository Tab

3.2.5 Object Cross-References

The Cross-references dialog box shows all objects related to the selected object. Select the object for which you wish to see the related items, select Cross References from the context menu to open the Cross References dialog box. In this list, all objects used by this object are listed under **Used by** category, and all using it under **Uses** category.

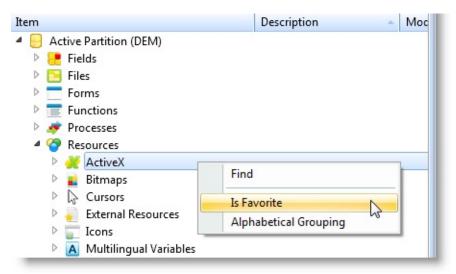


You can use this dialog box to see cross references for every object listed on it. To do this, select an object, and then either click the plus sign to see all objects either **Used by** or **Uses**, or press the on the toolbar to show both **Used by** and **Uses** for the object being selected.

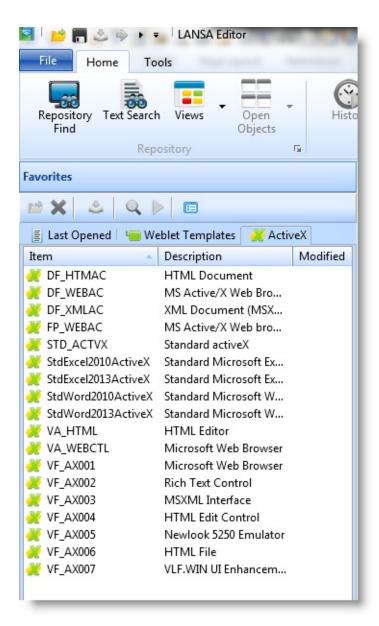
↑ 3.2 Repository Tab

3.3 Favorites Tab

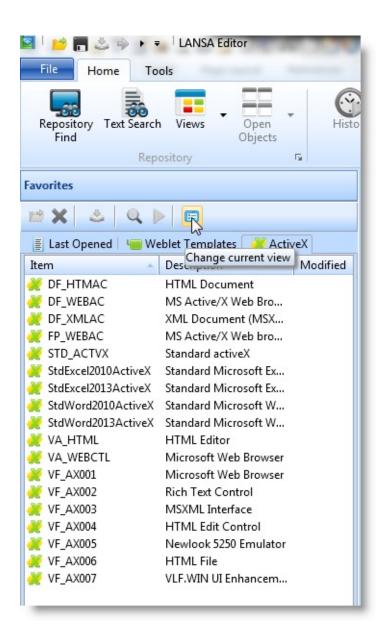
The Favorites tab shows one or more lists that have being marked as a favorite. To mark a group or a list as a favorite, select it, right-click to open the context menu and choose the Is Favorite option. You can make any repository node or any of your lists a favorite to display it in a separate tab accessed via the Favorites tab.



The Favorites tab can be displayed by pressing Shift + F8 or by using the Views command on the ribbon. By default the Favorites tab always contains the Last Opened tab.



When a particular object type is made a favorite, you can use the Toggle current view to toggle the Favorites tab between alphabetic grouping, grouping in dropdown and all objects:



Also See

Editing Lists

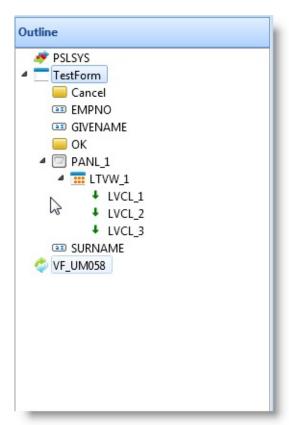
↑ 3. Standard Editor Tabs

3.4 Outline Tab

The Outline tab lists the objects open in the editor and shows the components inside the objects. You can use this view to move easily between the objects and components.

You can undock this tab it and put it somewhere on your desktop.

The Outline tab can be displayed by pressing F6 or by using the Views command.

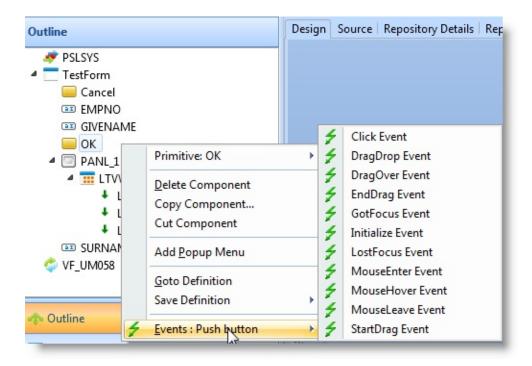


An item which has focus in the Source or Design tab is highlighted in the Outline.

Outline Tab Context Menu for Components

Some components are only visible in the Outline, not in the Design tab. For example, components such as Visual LANSA timers and sound components are only visible in the Outline. Also when you include a form in another form in multi-form applications, the member forms are only shown in the Outline.

A context menu is displayed if you right click on a component:



The context menu options vary depending on the type of component. These are some of the menu items that may be displayed:

Add/Detach Detaches context menu from the selected component.

Popup Menu Refer to Menus.

Goto Definition Moves focus to the definition of the selected component.

Refer to 3.5 Go To Tab.

Save Definition Saves the definition of the selected component.

Refer to Saved Definitions.

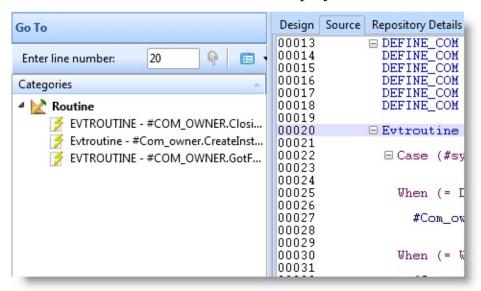
↑ 3.4 Outline Tab

3.5 Go To Tab

The Go To tab shows all the events and methods of the Source Code open in the editor in the Source tab. The type of object being edited determines the information that is shown in this tab. For example it may show errors in a file, or methods and event routines in a form or a reusable part.

The Go To tab can be displayed using Views command in the ribbon or rightclicking on an object in the Design or Source tabs to open the context menu and then choosing Goto Definition.

Double-click on an item in the Go To tab to display it in the Editor.

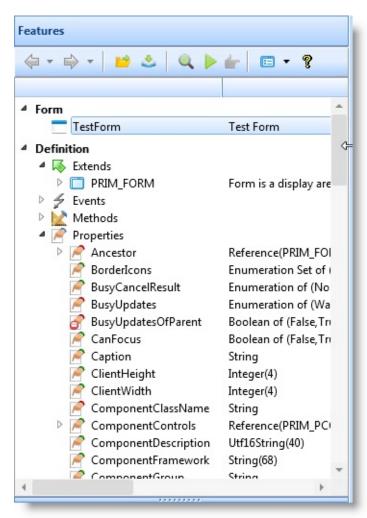


If you wish to jump to a particular line in the Source code, enter the line number in Enter line number at the top of this tab.

1 3. Standard Editor Tabs

3.6 Features Tab

The Features tab displays the events, methods and properties of the object being edited. The information displayed is dependent upon the type of LANSA object selected.



To find out more about the type of help provided for a selected feature and how to access it, refer to Help Text.

↑ 3.4 Outline Tab

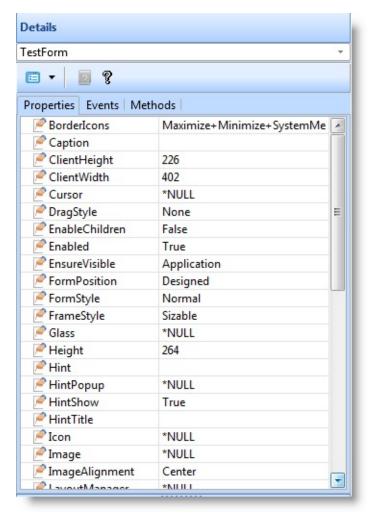
3.7 Details Tab

3.7.1 Form Component Details Tab

This tab is where you view and change the characteristics of Visual LANSA objects.

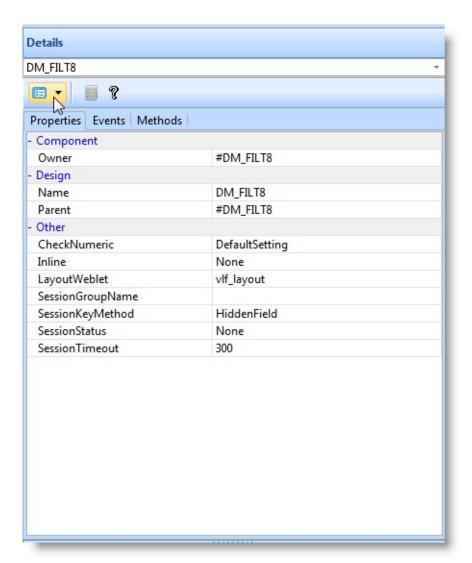
For example when editing a file, if you click on any field in the right-hand pane, the Details tab will open. For instance if you select a field attribute, the corresponding attribute is immediately displayed in the Details tab.

The Details tab can also be displayed by pressing F7 button or by using the Views command in the ribbon.



The Details Tab may contain other tabs depending upon the type of information being edited.

The content of the Details tab can be displayed in categories using the Switch between available views button:

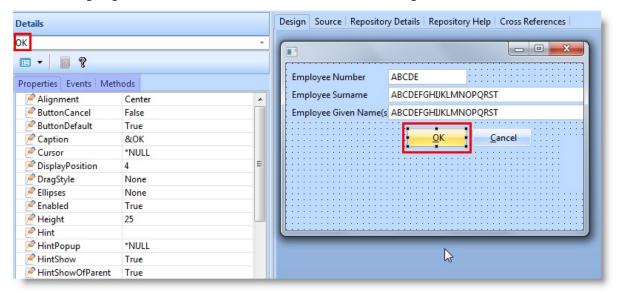


↑ 3. Standard Editor Tabs

3.7.1 Form Component Details Tab

When a form is open in the editor, you can use the Details tab to view or edit the properties, events and methods for the component currently being worked with. You can change the current component by selecting it from the dropdown list on the top of this tab or by selecting it in the 3.4 Outline Tab or in the 3.9 Design Tab.

When you select a component (for instance a button on a form), the Details tab shows the properties, events and methods of the component.



When editing a form, the Details tab for components is divided into three tabs:

Details - Properties tab

The Details Properties tab shows the attributes of the component which is being edited. You can use properties to control things such as the size and color of a component, its caption, and whether it should be visible or not.

When you change the properties of a component using the Design view, you are making design-time changes. You can also change the properties of components at run time using the SET command.

Some values such as for example Caption have to be typed in; predefined values can be selected from a list.

Note that when you change the value of a property, this change is automatically reflected in the other parts of the editor. For example if you change the Height and Width properties of a component, the component is simultaneously resized in the Design tab. Its DEFINE_COM statement in the 3.8 Source Tab is also

updated.

Details - Events tab

The Details Events tab is used to create and view event routines for the component being edited.

The Events tab is synchronized with the Source tab:

- If you double-click on an event in the Events tab, an EVENTROUTINE with a matching ENDROUTINE command is inserted into the 3.8 Source Tab.
- When the source code of the component contains a routine for an event, the icon in front of the event name in the Events tab is changed.

You can locate an existing event routine by double-clicking its name in the Events tab.

Details - Methods tab

The Details Methods tab is used to create and view methods for the component being edited.

When you double-click on a method, a statement invoking this method is inserted into the source of your component in the position where the cursor is in the 3.8 Source Tab.

So if you double-click the MaximizeForm method for a form and your cursor is in the Click event of #PHBN_1, a statement invoking the MaximizeForm method of the form is inserted in the Click event:

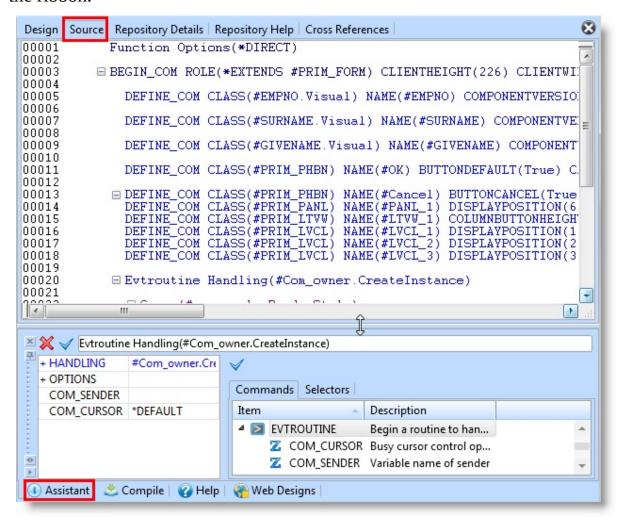
EVTROUTINE HANDLING(#PHBN_2.Click)
INVOKE METHOD(#COM_OWNER.MaximizeForm)
ENDROUTINE

↑ 3.7 Details Tab

3.8 Source Tab

The Source tab is only available when an object that contains code (such as a form, function or WAM) is open in the Editor. The Source tab allows you to view and edit the source code for your object. The 3.9 Design Tab and 3.10 Repository Help Tab are often displayed with the Source Tab.

The Source tab can be displayed by pressing F9 or by using Views command in the ribbon.



Also See

3.8.1 Rules for the Editor

Following are the general rules for code syntax in LANSA:

- The object type determines the structure of the source code.
- The object type determines the commands allowed.
- You can enter one statement or command per line.
- The first character of a command line must not be blank.
- Blank characters are not allowed between a parameter name and its first bracket.

For example, FUNCTION OPTIONS(*DIRECT) is correct. FUNCTION OPTIONS (*DIRECT) is incorrect and will be flagged as an error.

- The editor is not case sensitive.
- You do not need to end statements with a semicolon.
- To comment out a line, use an asterisk (*).
- A comment line in RDMLX code can be up 2048 bytes in length. The limit of 55 characters still applies for RDML code.
- The syntax of the commands is checked as you type.
- You cannot have blank lines in an RDML function, but blank lines are allowed in RDMLX code.

For Visual LANSA components note that:

- A component definition must be enclosed inside BEGIN_COM and END_COM statements.
- With the exception of DEFINE_MAP, all definition statements must be placed before event routines. This means that DEFINE_COM, DEFINE_EVT and DEFINE_PTY statements are always defined in the beginning of the component. DEFINE_MAP must either follow the DEFINE_EVT statement it belongs to or it has to be placed inside a PTYROUTINE or a MTHROUTINE statement.
- All code, except for definition statements, must be written inside an event routine or a subroutine.

3.8.2 Find

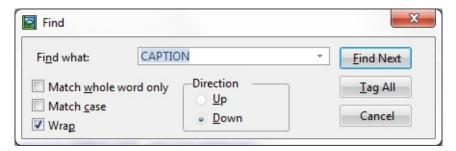
You can find text in the Source tab by entering the string you wish to locate in the Find entry field in the ribbon and pressing Enter:



Or by pressing the Find button:

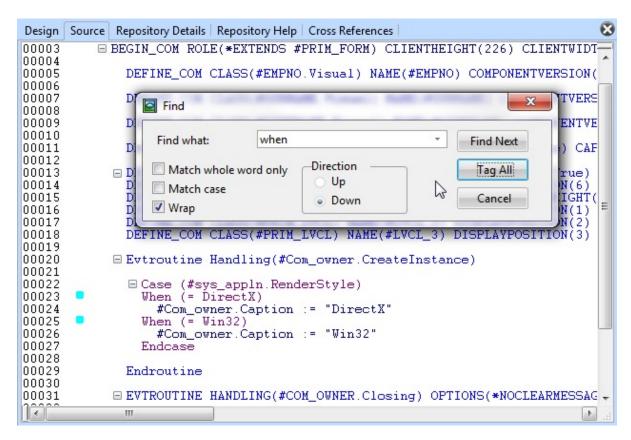


Or using Ctrl + F keys to open the Find dialog:



The Find Dialog gives you the option of matching whole words only and/or of matching case when searching. (Reminder: code is not case sensitive.)

The Tag All option will highlight all lines in the editor that match the search criteria. You can use the Tag All option in combination with the Tags in the 3.5 Go To Tab to quickly navigate within your code:



Also see

Find and Replace keyboard shortcuts.

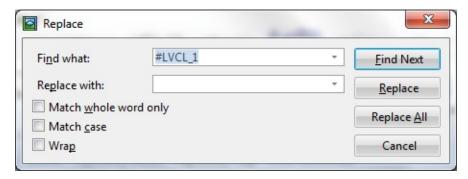
↑ 3.8 Source Tab

3.8.3 Replace

You can replace text in the Source tab by pressing the Ctrl + H keys or by using the Replace command in the ribbon:



To open the Replace dialog box:



Also See

Find and Replace keyboard shortcuts.

↑ 3.8 Source Tab

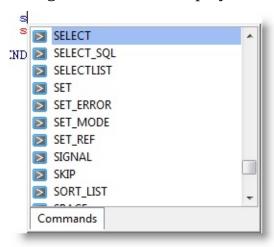
3.8.4 Auto Complete

The Auto Complete feature helps you complete commands and parameters as you type. For example, the command name is incomplete.

The type of help you get from auto complete is controlled using the Auto Complete option in the Source Settings.

There are three auto complete options:

- **Off** which means you need to type every command. If you have this option turned off, you can quickly start it by pressing Ctrl + Space bar.
- **Prompter** if selected gives you a list of commands starting with the first character you type. For example, if you type the letter S on a blank line, a list of commands starting with S will be displayed:



You can use the Up and Down arrow keys to move up and down the list of choices. Press Enter to apply the selection to a statement.

Prompter is the default when you install a new LANSA system. The prompter also automatically adds related enclosing commands, for example:

- Selecting BEGIN_LOOP will automatically add END_LOOP
- Selecting DOWHILE will automatically add ENDWHILE.

It also adds any required parameters, for example selecting IF will also include the COND() in the resulting statement.

Pressing Ctrl + Space also brings up the prompter (you can change this shortcut to Ctrl + J in the Editor Options).

• **Inline**, if selected gives you a and you a best guess of the command you are entering. For example if you type S on an empty line, the remainder of the

command is displayed, in the following example, this is ELECT (for the SELECT command).

```
00012

00013

00014▼ sELECT

s is not a valid LANSA Command.

00015
```

Also See

3.8.1 Rules for the Editor

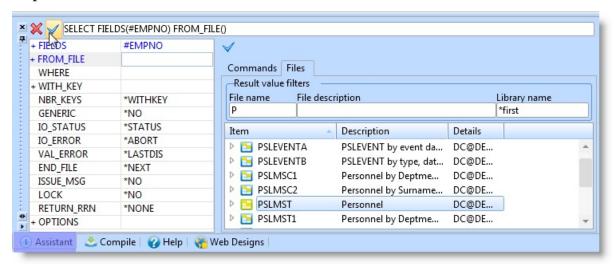
3.8.5 Assistant Tab

↑ 3.8 Source Tab

3.8.5 Assistant Tab

The purpose of the Assistant tab is to help you specify complex commands while working within the 3.8 Source Tab.

The Assistant helps you to enter new commands and to work with existing RDML/X commands with 3.8.4 Auto Complete helping you to speedily specify simple commands.



The Assistant tab shows a structured view of a command's parameters and values. Refer to List of Commands, Specify Parameters, The Last Saved Command and Specify Attribute Values for Parameters.

The Assistant can be displayed by pressing F4 when focus is placed on a command in the Source tab. The contents of the tab are automatically updated based on the command that has focus.

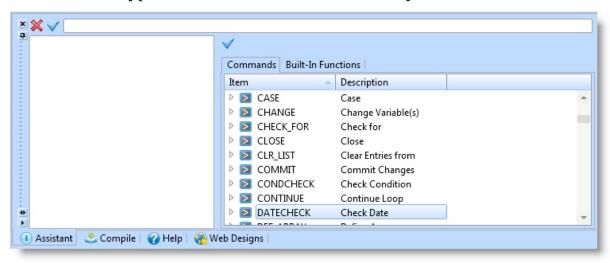
Control over the appearance and location of the Command Assistant tab is described in Setting up Your Workspace.

1 3. Standard Editor Tabs

List of Commands

When you invoke the Assistant from the Source tab, the selected command line is displayed. If your cursor is on a blank line, a list of all LANSA commands are displayed.

To quickly locate a command in the list, set the focus on the list of the commands and type the first letters of the command you want to enter.



Choose a Command

Press Enter when you have highlighted the command that you want to use. In this example, the command DATECHECK has been highlighted. The Assistant will show the parameters and then the values available for use with that command.

You can display the parameters for a command by double-clicking the command. To automatically display them whenever you select a command, select the Auto Expand Parameters option in LANSA Settings.

You can obtain help about a command by pressing F1 when you have highlighted the command for which you want information.

Also See

The Technical Reference Guide, for a full list of the:

RDML Commands

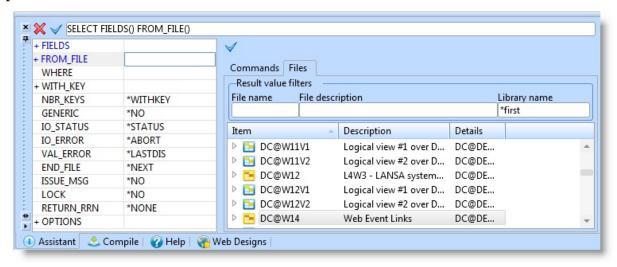
RDMLX Commands

↑ 3.8.5 Assistant Tab

Specify Parameters

In the displayed parameters, you can use the up and down arrow keys to move around the list of the command's parameters.

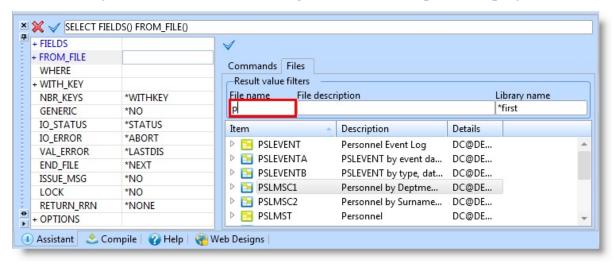
In the following example, showing the SELECT command, the FROM_FILE parameter has been selected:



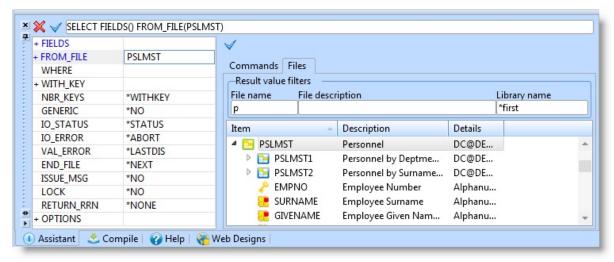
The potential values for the parameters listed are displayed in the right side of the Assistant tab.

You can reduce the number of values listed using the filters above the lists. You can select a tab and then double-click on the selected item, or press Enter or click on the tick above the list. To select multiple values (when appropriate), hold down the Shift key while making your selection.

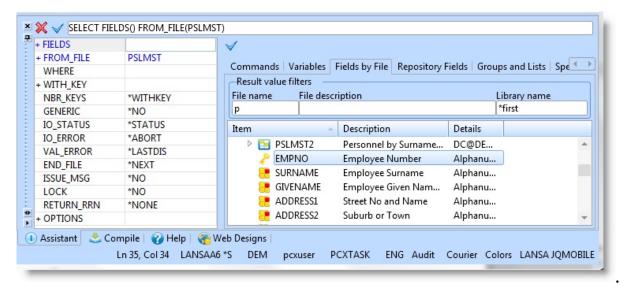
In the following example, the letter 'p' has been entered in the filters File Name field and only files with names starting with the letter 'p' are displayed.



Select the required file from the list and then select the fields:



To select one or more fields move the cursor to the Fields parameter and choose the fields:



↑ 3.8.5 Assistant Tab

The Last Saved Command

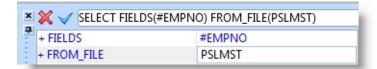
You can use the Cancel ★ and Check ✓ buttons at the top of the Assistant tab to save the command and display it in the Source tab. You can make further changes to the command, and if you don't save them, you can restore the last saved command if required.

For example:

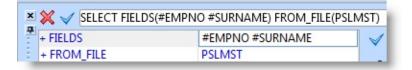
1. Save the command that has just been entered by clicking on the check ✓ button. The saved command is displayed at the top of the Command Assistant.



2. Delete the #SURNAME field from the Fields parameter:



3. Press the Cancel **x** button. The command reverts to the last saved format:

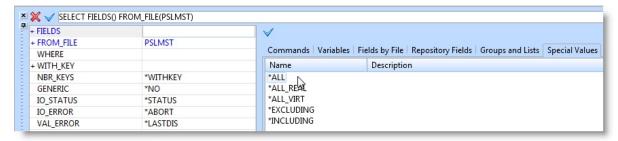


↑ 3.8.5 Assistant Tab

Specify Attribute Values for Parameters

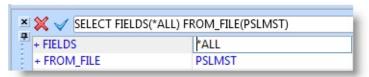
You can use the Command Assistant to enter parameter values that are not found in the Repository.

For example, if you want to make the value of the Fields() parameter of a SELECT command *ALL, place the cursor in the Fields() parameter. Display the Special Values tab.



Highlight *ALL and either double-click on it, press the ✓ button or press the Enter key to move it into the field's attributes.

The Command Assistant enters the *ALL attribute as the value of the Fields parameter:



↑ 3.8.5 Assistant Tab

3.9 Design Tab

The Design tab is used to display the layout of the object open in the Editor.

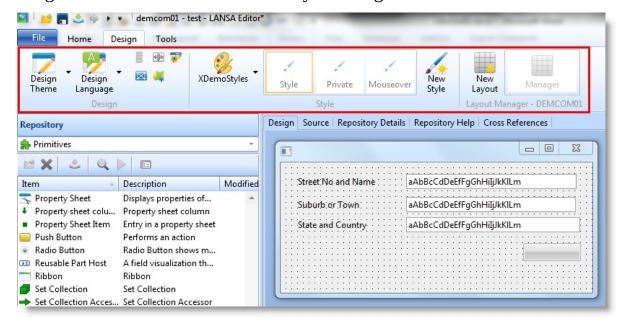
The Design tab can be displayed by clicking on the tab or pressing F10.

The Design tab cannot be closed or repositioned in the editor. The settings for Design tab are controlled using the Editor Options.

↑ 3. Standard Editor Tabs

3.9.1 Design Tab for Forms and Reusable Parts

The Design tab shows what the form or reusable part will look like at execution time. It is used to build the layout of the object. Use the commands on the Design tab in the ribbon to work with your design.:



Also see Creating Applications Using Components in the Developers Guide.

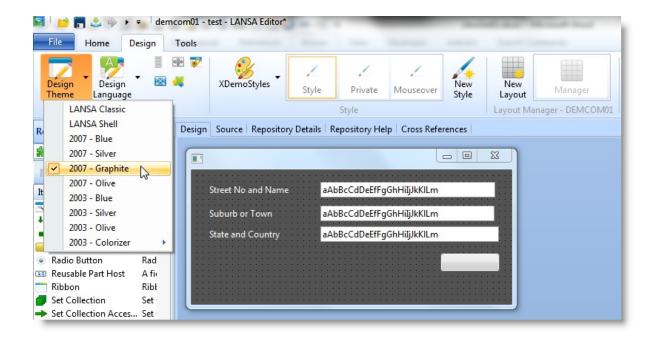
Also See

Forms

↑ 3.9 Design Tab

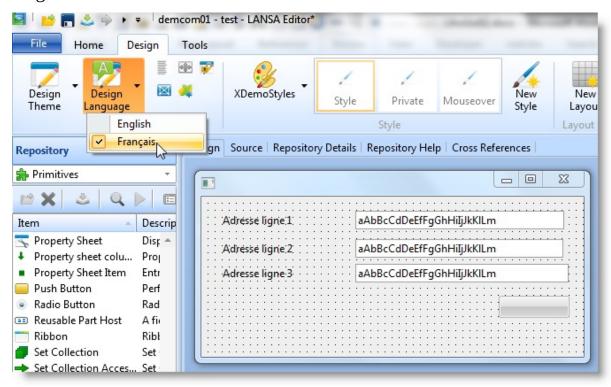
Design Theme

Use the Design Theme command to specify the theme of the object being designed :



Design Language

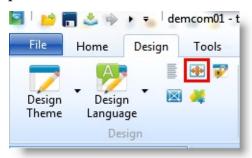
Use the Design Language command to select the language of the object being designed:



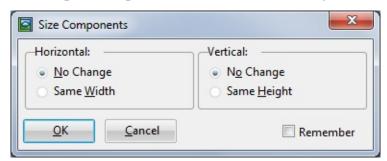
Size Command

Use the Size command to make two or more selected components have the same width and/or height on the form.

Select two or more components and use the Size command:



The order of selection on the Design tab is important. The first component selected is considered the anchor. All other objects will be sized to match this object. (To select multiple components, hold the Shift key down.)



If you select the Remember check box, the width and height selections you make in this dialog are remembered the next time you display it. If you do not select Remember, the selections will be set to 'No Change'.

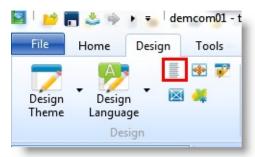
Also See

Layout Managers in the Developer Guide.

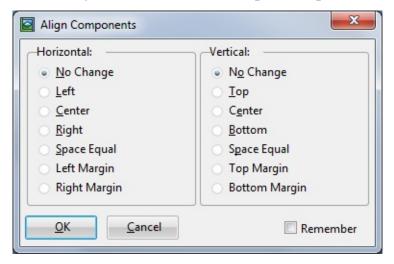
Align Command

Use the Align Components dialog to align two or more selected components in the Design tab. You can specify both the horizontal and the vertical alignment.

The Align Dialog is displayed by selecting the Align command once two or more components have been selected:



The order of selection on the Design tab is important. The first component selected is considered the anchor. All other objects will be aligned to match this object. (Hold the Shift key down to select multiple components.)



If you select the Remember check box, the alignment selections you make in this dialog are remembered the next time you display it. If you don't select it, the selections will be set to 'No Change'.

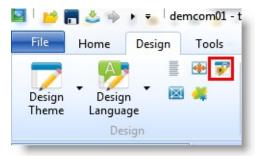
Also See

Layout Managers in the Developer Guide.

Lock Design

Use the Lock Design option to prevent changes being saved when you are prototyping changes to a design in the Design tab. When you lock a design, all component definitions are saved. Once locked, the layout of components can be changed in the Design tab, but the changes are not saved. When you unlock the design, all component layouts will be restored to the original saved settings.

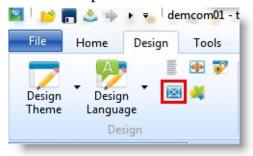
Select the Lock Design command from the Design tab on the ribbon:



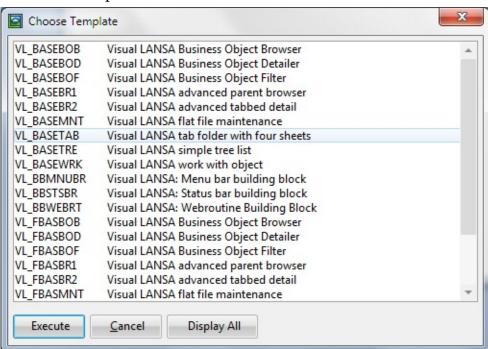
When a design is locked, no new components can be added. Any components that are altered will appear will appear greyed out to indicate that they have been changed.

Use Templates

Select the Use Templates command to run a template which generates all or part of the component interface:



Choose a template to execute:



Depending on the template, you may need to specify information such as file and field names etc.

Expose as ActiveX

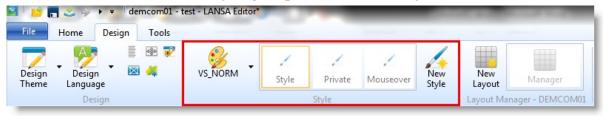
Use the Expose as ActiveX command to set the attributes required to expose the component as an ActiveX control:



Refer to ActiveX Controls in the Developer Guide.

Style

Use the commands in the Style group to work with styles.

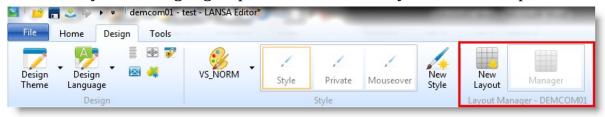


See Visual Styles in the Developer Guide.

To learn more about DirectX styles, see the document http://www.lansa.com.au/support/tips/t0588.htm at the LANSA Support web site.

Layout Manager

Use the Layout Manager group to work with the layout of the component:

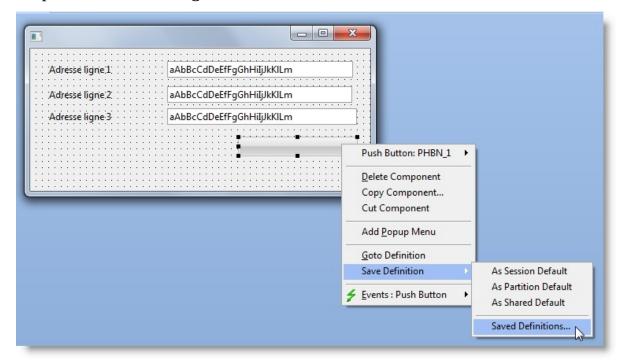


See Layout Managers in the Developer Guide.

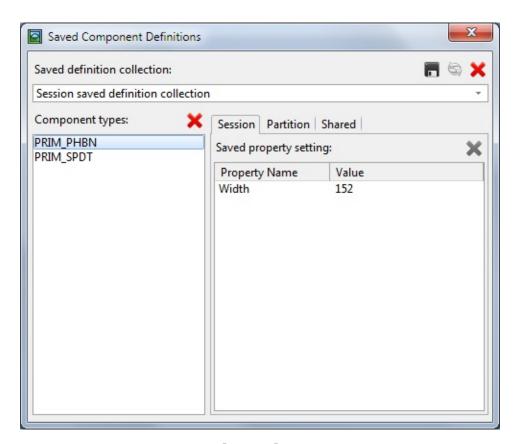
Saved Definitions

Use the Saved Component Definitions dialog to view and delete saved default definitions for different types of components.

The Saved Component Definitions dialog is displayed by right-clicking a component and choosing Saved Definitions from the context menu:



The dialog shows the definitions for different component types:



The saved definitions are stored in lansa\lansa\

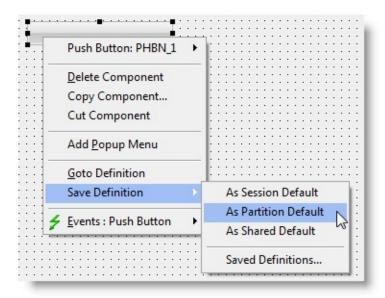
<system/configuration>\liiy1*.dat.

Note that when you are using layout managers, the saved height or width of a component may be changed by the layout manager. For more information, refer to Layout Managers in the *Developer Guide*.

Saving Definitions

When you save a default definition for a type of component, other instances of this type of component assume the characteristics of the saved default when you add them to a form or other container. The attributes which are saved depend on the type of component, but most default definitions include Height or Width or both. Note that some components, for example context menus, do not have any default definitions which can be saved.

To set a default definition for a type of component, place a component of the type required on a form in the Design tab and set its attributes. Right-click to open the context menu, and select the Save Definition option:



Use the sub menu to choose the level of save you want. The levels are:

- **As session default**. The definition is lost when the editor is closed.
- **As partition default**. The definition is saved for the current partition.
- **As shared default.** The definition is shared by all partitions.

In the case of partition and shared defaults the Saved Definitions are not written to disk until the editor shuts down, so if you have a repository on a server, you may want the modified Saved Definitions saved immediately so that they are available to other users. You can also save the definitions to a file for backup while making changes and then reload them from the file.

The session definitions are searched first for a match. If it is found, the saved definition is used. If it is not found, the partition definitions are searched. If not found in both the session and partition defaults, the shared defaults is searched. If no match is found in any of the saved definitions, the component is created as its system default.

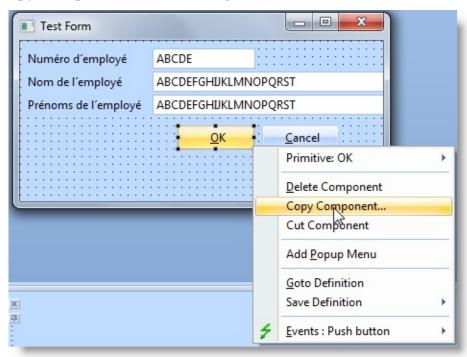
Note that if the definition you are trying to save is the same as the system default, no values are shown in the Saved Definitions dialog. Only if the definition is different from the system default can it be saved.

For a field you can also save the definition of its ancestor.

When adding additional fields that inherit from the primitive entry field component, they will be initialized with that definition, provided a definition of the field itself is not found.

Copy Component

You can copy components in the Design tab:



When you copy a component the Include dialog is displayed so that you can specify whether the existing event routines and children of the component are copied.

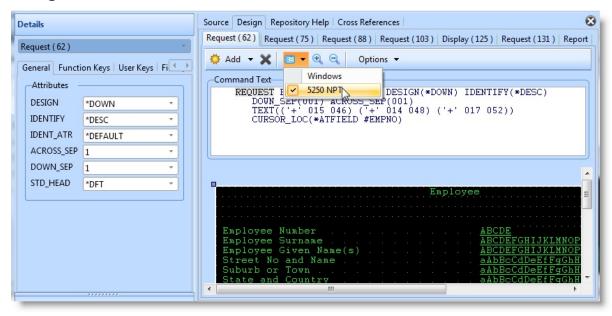


Also See

Copy a Component in the Visual LANSA Developer Guide

3.9.2 Design Tab for Functions

The Design tab shows what the function will look like at execution time. The tab folder contains a tab for every REQUEST, DISPLAY and POP_UP command in the function. Following is an example of the Design tab when editing a function:



Also See

Editing Functions

↑ 3.9 Design Tab

3.9.3 Other Objects in the Design Tab

The Design tab is shown for many types of LANSA objects, for example:

Bitmaps If you are working on a bitmap, the Design tab will show how the bitmap will be displayed on a toolbar or push-button.

If you are using a bitmap that contains separate images for the different button states, the first button view shows the bitmap on a button which is not selected (button state Up), the second view shows it on a disabled button, the third view shows it on a button that has been pressed down and the fourth view shows the image on button which kept pressed down (Checked). The bitmap has an ImageCount property used to indicate how many images are contained in the bitmap.

If you use a bitmap with a single image, all the four views in the Design tab show this image.

Icons When you are working on an icon, the Design tab will show how the icon will be displayed in different contexts. For more information refer to the online Feature help.

Cursors When you are working on a cursor, the Design tab will show how the cursor will be displayed. For more information refer to the online Feature help.

ActiveX When you are working on an ActiveX control, the Design tab will show how the control will be displayed if it has a visualization. For more information refer to the online Feature help.

Visual StylesWhen you are working on a visual style, the Design tab will show what the style will look like on various controls. With visual styles you can control visual attributes such as colors, fonts, font sizes and 3D effects.

Also See

3.9.1 Design Tab for Forms and Reusable Parts3.9.2 Design Tab for FunctionsEditing Objects

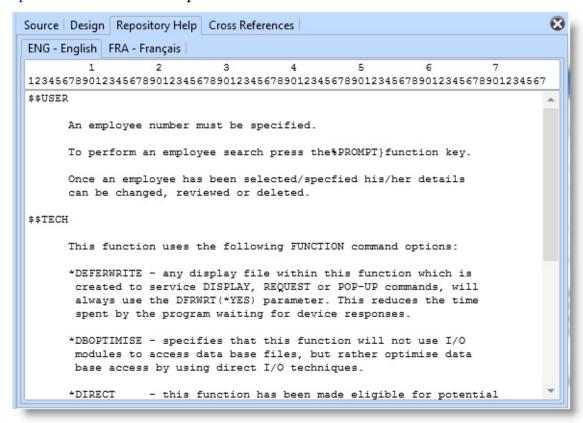
↑ 3.9 Design Tab

3.10 Repository Help Tab

The Repository Help tab is used to enter help text for the opened object. It cannot be closed or repositioned in the editor. The help text that you enter here is displayed in the Help text pane at the bottom of the LANSA Editor's window. Not all object types support Repository Help.

A separate tab is displayed for each of the 3.10.1 Languages specified in the partition.

For a complete list of rules for repository help editing, refer to the Repository Help Editor in the *Developer Guide*.



For fields, you can insert dummy formatted text or import the help text from its reference field using the Default Help Text and Import From commands in the ribbon:



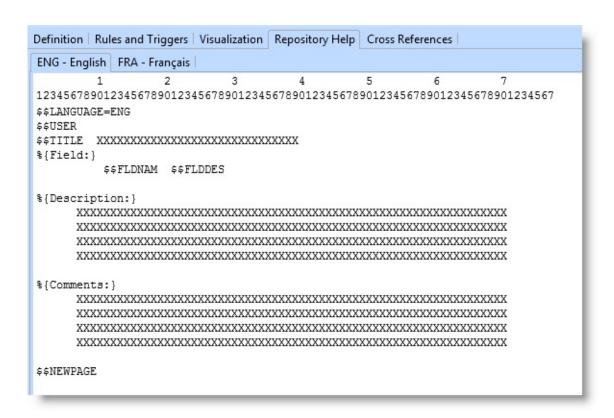
Import Applicable to fields only: Imports the help text of the reference field of this field . Refer to Field Help Text in the *Technical*

Reference Reference Guide.

Field

Default Help Inserts some dummy formatted text for you to replace with the

Text actual text. See example following.

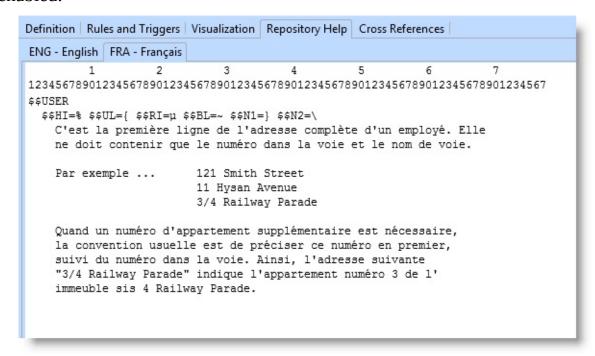


Also See

Field Help Text in the *Technical Reference Guide*Component Help Text in the *Technical Reference Guide*Process Help Text in the *Technical Reference Guide*Function Help Text in the *Technical Reference Guide*Repository Help Text Development in the *Developer Guide*↑ 3. Standard Editor Tabs

3.10.1 Languages

A help text tab is available for all the languages for which the partition has been enabled:



Also See

Field Help Text in the Technical Reference Guide
Component Help Text in the Technical Reference Guide
Process Help Text in the Technical Reference Guide
Function Help Text in the Technical Reference Guide
Repository Help Text Development in the Developer Guide
Introduction in the Multilingual Application Design Guide

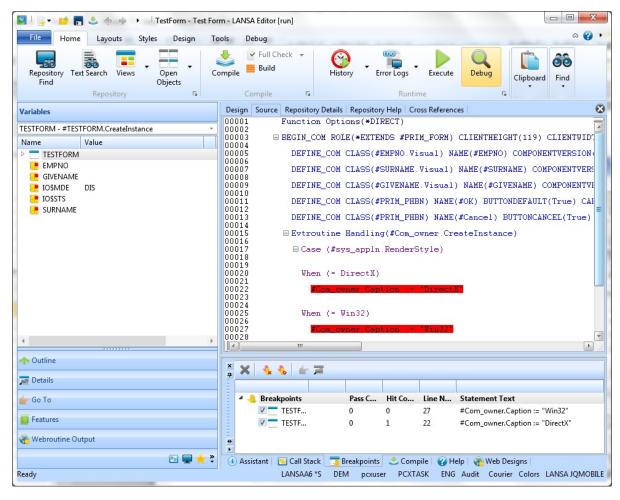
1 3.10 Repository Help Tab

3.11 Debug Tabs

The Visual LANSA Debugger uses the following tabs:

- 3.11.1 Debug Breakpoints Tab
- 3.11.2 Debug Variables Tab
- 3.11.3 Debug Call Stack Tab

The debug tabs can be displayed at any time using the Views command on the ribbon. These tabs will be blank unless a program object (such as a form, function or WAM) is being executed in debug mode or has been opened in the editor.



After you have started debugging, the Debug tab is displayed on the ribbon:



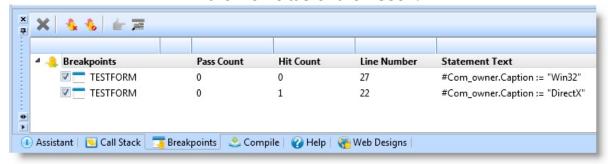
Also See

Debugging Applications

↑ 3. Standard Editor Tabs

3.11.1 Debug - Breakpoints Tab

The Breakpoints tab can be displayed at any time by using the Views command in the Home tab of the ribbon.



For information, refer to Breakpoints.

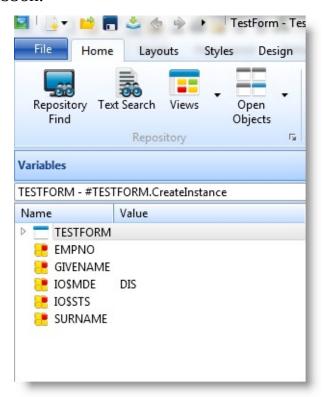
Also See

Debugging Applications

↑ 3.11 Debug Tabs

3.11.2 Debug - Variables Tab

The Variables tab can be displayed at any time using the Views command in the Home tab of the ribbon.



For information, refer to Variables.

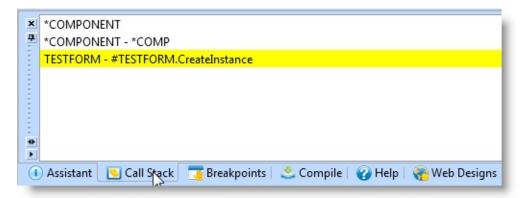
Also See

Debugging Applications

↑ 3.11 Debug Tabs

3.11.3 Debug - Call Stack Tab

The Call Stack tab can be displayed using the Views command in the Home tab of the ribbon:



For information, refer to Call Stack.

The control over the appearance and location of the debug tabs is described in Setting up Your Workspace.

Also See

Debugging Applications

↑ 3.11 Debug Tabs

3.12 Compile Tab

The Compile tab shows the status of any submitted compile jobs.

Objects can be submitted for compile using the Compile command:

- In the Ribbon.
- On any of the context menus.



If you submit several objects for compilation at the same time, this will create one compile job.

To see details about the compile output, display the 3.12.2 Compile Details Dialog.

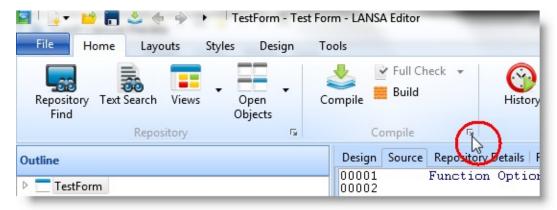
Compile Toolbar

P	Show detailed output	Refer to 3.12.2 Compile Details Dialog
ė,	Go To	If a compile fails with syntax errors, Go To opens the component and positions at the selected line.
×	Remove job	Removes the selected job details from the compile tab
×	1 0	Removes all pending jobs and completed job details from the compile tab
0	Stop	Stop the compile job.

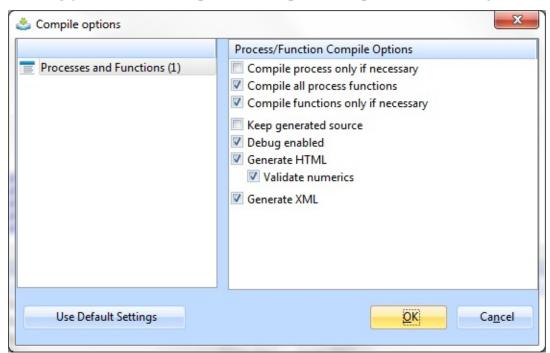
↑ 3. Standard Editor Tabs

3.12.1 Compile options

To display the Compile options dialog, expand the Compile group on the ribbon:



In the dialog you can set the specific compilation options for the object.



The Use Default Setting button will reset the compile options to their default values.

For information about the Compile options, refer to the following:

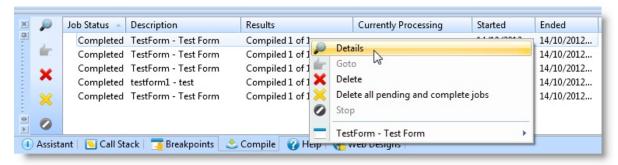
- File Compile Options in the Technical Reference Guide
- Component Options in the *Technical Reference Guide*.

3.12.2 Compile Details Dialog

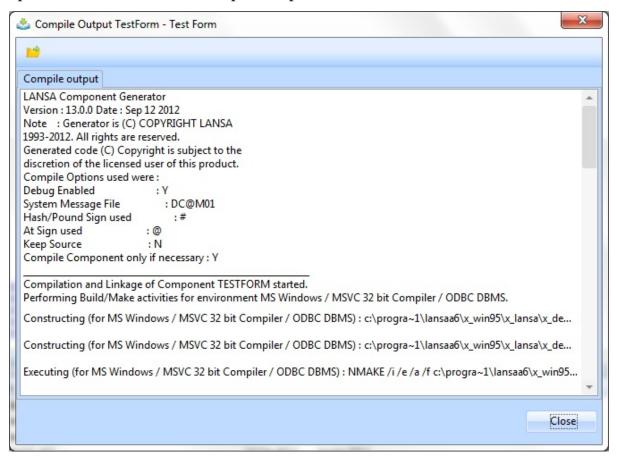
↑ 3.12 Compile Tab

3.12.2 Compile Details Dialog

To see the details of a completed compile job, double-click the compile message or right-click and select Details:



Use the Compile Details dialog to view detailed information about the output of the selected compile job. For forms and reusable parts, the information is separated into Build and Compile output.



Press the Save button on the toolbar to display a Save As dialog. You can save the compile details as a text file in order to send this information to LANSA

Support if necessary.

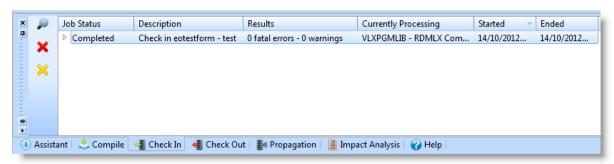
3.12.1 Compile options

↑ 3.12 Compile Tab

3.13 Check In Tab

The Check In tab shows the status of Check in jobs. Objects can be checked in using the context menu from the 3.2 Repository Tab.

The Check In tab is only displayed if the Visual LANSA System Type is a slave system.



If you Check In several objects into the master repository at the same time, this will create one Check In job. To see details about the Check In, display the 3.13.1 Check In Details Dialog.

If it is not already started, the Host Monitor will be started automatically in the background when you Check In objects.

Messages shown in red indicate an error, messages in brown indicate a warning.

Check In Toolbar

View detailed messages	View the detailed messages for the selected check in job. Refer to 3.13.1 Check In Details Dialog.
Remove job	Removes the job from the list and release the Windows resources.
Delete all pending and complete jobs	Removes all pending jobs and completed job details

Also See

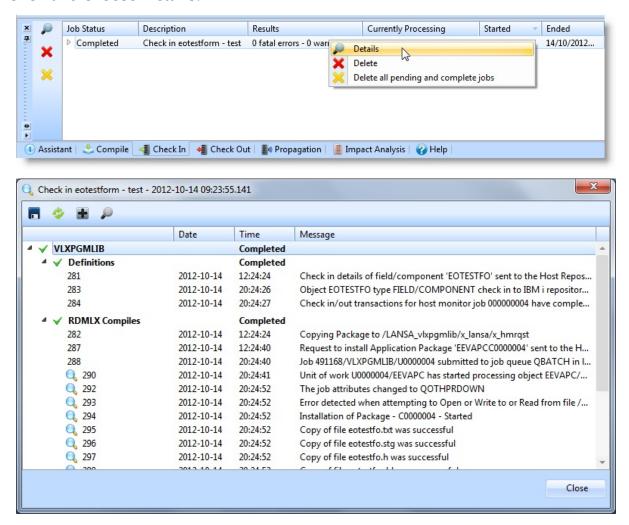
1 3. Standard Editor Tabs

Check In in the *Administrator Guide*Host Monitor Concepts in the *Administrator Guide*Host Monitor in the *Administrator Guide*

3.13.1 Check In Details Dialog

Use the Check In Details dialog to view detailed information about the messages for the Check in job.

To display the details double-click the check-in completion message or right-click and choose Details:

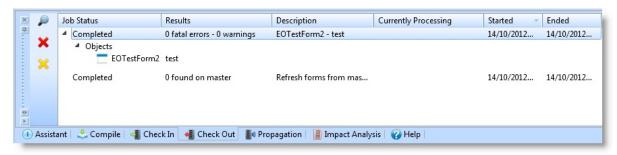


↑ 3.13 Check In Tab

3.14 Check Out Tab

The Check Out tab shows the status of Check Out jobs and messages issued by the 3.14.2 Refresh Master Object List and Refresh of selected objects. Objects can be checked out using the context menu from the 3.2 Repository Tab.

The Check Out tab is only displayed if the Visual LANSA System Type is a slave system.



If you Check Out several objects from the master repository at the same time, this will create one Check Out job. To see details about the Check Out, display the 3.14.1 Check Out Details Dialog.

If it is not already started, the Host Monitor will be started automatically in the background when you Check Out objects.

Messages shown in red indicate an error, messages in brown indicate a warning.

Check Out Toolbar

	View detailed messages	View the detailed messages for the selected check out job. Refer to 3.14.1 Check Out Details Dialog.
×	Remove job	Removes the job from the list and release the Windows resources
×	Delete all pending and complete jobs	Removes all pending jobs and completed job details

Also See

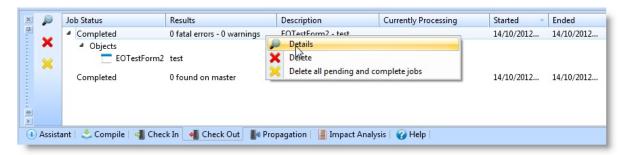
Check Out in the *Administrator Guide*Host Monitor Concepts in the *Administrator Guide*Host Monitor in the *Administrator Guide*.

1 3. Standard Editor Tabs

3.14.1 Check Out Details Dialog

Use the Check Out Details dialog to view detailed information about the messages for the Check out job.

To display the details double-click the check-out completion message or right-click and choose Details:

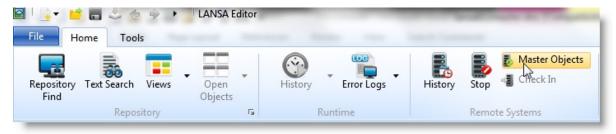


You can save and refresh the messages being displayed using the toolbar buttons. You can also show or hide message details and show the IBM i job log.

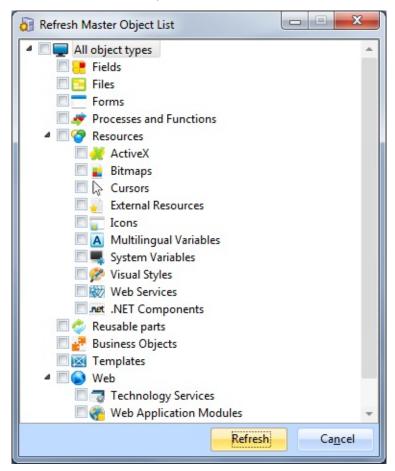
↑ 3.14 Check Out Tab

3.14.2 Refresh Master Object List

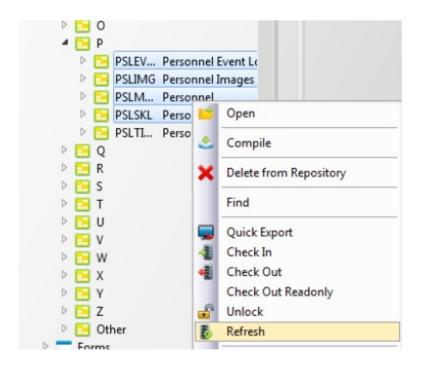
Use the Master Objects command to obtain a complete set of data from the master system for the selected object types or all object types:



Use the Refresh Master Object List dialog to specify the types of objects that will be retrieved from the master system:



To Refresh a selection of objects, select the objects in the repository and use the Refresh command in the right click menu.



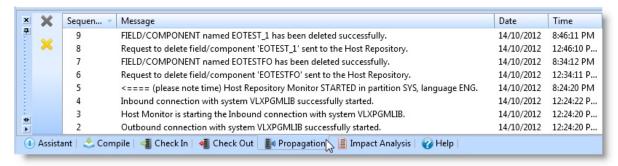
Also See

Refresh Master Object List in the Visual LANSA Administrator Guide † 3.14 Check Out Tab

3.15 Propagation Tab

The Propagation tab shows messages about the Host Monitor status and the Propagation of objects between LANSA Systems.

The Propagation tab is only displayed if the Visual LANSA System Type is a slave system.



The messages in the Propagation tab are for information only. Details are not displayed for these messages. Objects will be propagated as changes made to the Master System are reflected in the connected Slave Systems. The Host Monitor must be active in order for propagation to occur.

To manually start the Host Monitor, refer to the Editor Toolbar.

Propagation Toolbar



Remove one line Removes one line of information



Delete all lines

Removes all information

Also see

3.15.1 Connection History Dialog

In the Administrator Guide:

Propagation

Host Monitor

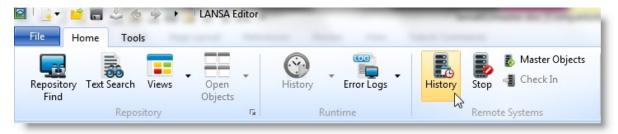
Host Monitor Concepts

1 3. Standard Editor Tabs

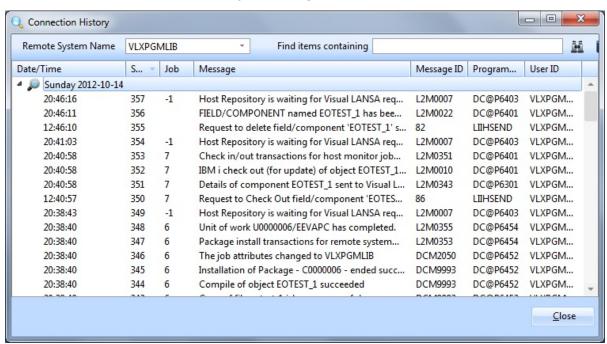
3.15.1 Connection History Dialog

The Connection History dialog displays any messages relating to connections to another LANSA System. Including the Host Monitor. Seven days of messages are available for the current workstation and partition, and a find option is available to search for specific information in the messages.

The Connection History dialog can be displayed using the History command in the Ribbon:



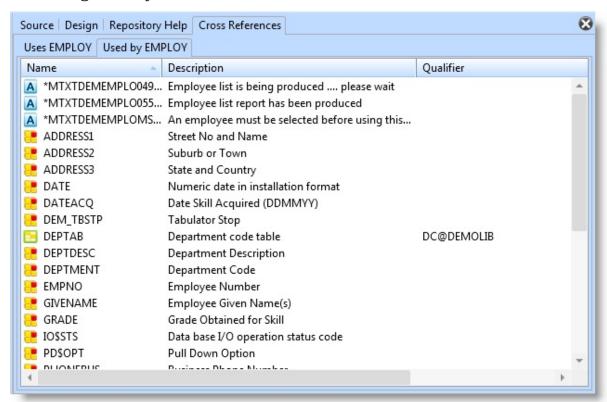
You can sort the details listed by clicking on column headers:



↑ 3.15 Propagation Tab

3.16 Cross References Tab

The Cross References tab shows the objects used by the opened object and all those using this object.



You can open a context menu by selecting one or more objects and rightclicking. From the context menu, you can select Open, Compile, or any of the other options that are available.

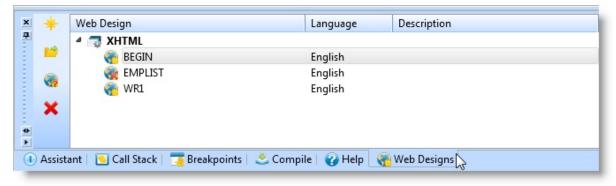
1 3. Standard Editor Tabs

3.17 Web Design Tab

The Web Design tab is designed for Weblet and WAM development.

It lists all designs for the current weblet open in the editor. Every design is in a specific language using particular Technology Service.

It can also list all webroutines for the current WAM open in the editor. Every webroutine on the tab represents a web page design using certain technology in certain a Language. Refer to Web Application Modules for further information.



↑ 3. Standard Editor Tabs

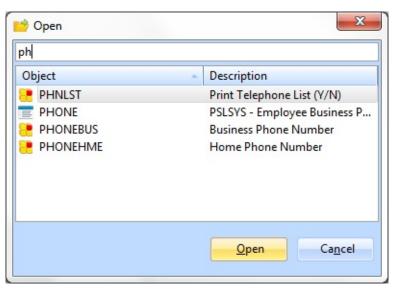
4. Editor Features

There are a number of features in Visual LANSA that are common to all objects and/or processes. These are:

- 4.1 Open an Object
- 4.2 Delete Objects
- 4.4 Search Facilities
- 4.5 Interface Indicators
- 4.6 Printing in Visual LANSA
- 4.7 Help Text
- 4.3 LANSA Import Facility

4.1 Open an Object

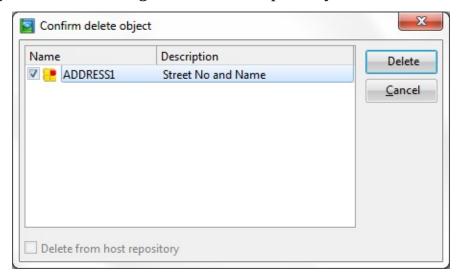
In the LANSA Editor, if you press Ctrl + O, you will activate the Open dialog box. This can also be activated from the File Menu using the Open option. Any objects corresponding to the name you are typing are listed and can be selected to open:



↑ 4. Editor Features

4.2 Delete Objects

You delete an object by selecting the object in one of the tab views like the Repository Tab and choosing Delete from Repository.



You can delete the object from the host repository as well as the local repository by selecting Delete from Host Repository. Note that:

• if the Host Monitor is not currently running, the request will be queued in the local database and will be actioned the next time the Host Monitor is executed with the current task from the current PC.

When the deletion is actioned on the host, confirmation messages will be listed in the Propagation Tab. If a different user signs on to Visual LANSA on the PC with the appropriate task and starts the Host Monitor, then that user will be involved in deleting the object on the host. This will not affect the ability to delete the object on the host but may lead to confusion. To avoid this confusion ensure that Host Monitor is running before requesting to delete an object on the host repository.

• if your system is a stand-alone system, then the Delete from host repository is disabled.

↑ 4. Editor Features

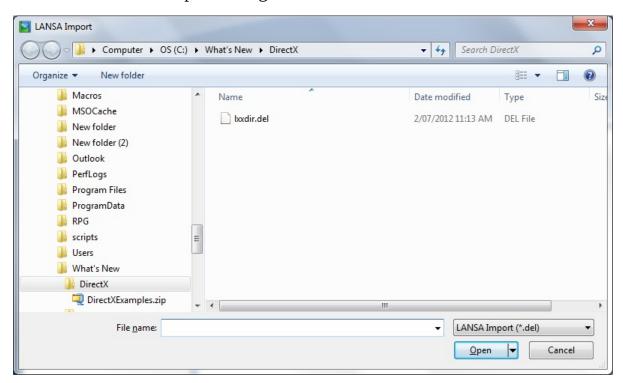
4.3 LANSA Import Facility

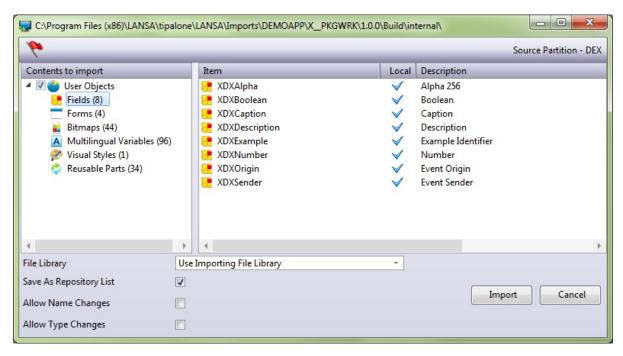
The Import facility is designed so that you can look at the contents of the import before actioning it. This may take some time, so if you have a large import it is more efficient to use the Import shortcut in the LANSA Administration folder.

1. To start the import, select the LANSA Import command in the ribbon:



2. Select the location of the package to be imported into the current partition from the LANSA Import dialog.





Objects in the package will be put into one of four categories:

- Partition Definitions Partition Settings, Frameworks and Groups
- System Definition System Settings and Commands
- System Objects Users and other relevant settings
- User Objects including Multilingual Variables, Fields, Files, Components, Files, Processes, Functions, WAMs, Web Components, Technology Services, Weblets, System Variables, Tasks, BIFs and Templates.

You can select the categories you require to be imported. By default, only the User Objects category is selected.

It is possible to execute an import which does not contain objects, for example it may be a series of commands to be processed. In this case no verification of the import contents is provided and the user is responsible for understanding the actions of proceeding with an import.

Importing Partition Definitions, System Definition or System Objects may change your current Visual LANSA settings, so please BE SURE that the change is desired before proceeding.

✓ Use this option to verify the import before actually executing it. Any messages, including errors, will be reported in the Import tab

Press the Import button to submit the import job. The submitted job will import objects based on your selections.

File Library

Use Importing Use the library information supplied with the file definition

File Library and import into the same library locally.

Default.

Library

Save as Repository List

Create a static list of all the objects as they are imported.

Allow Name Changes

All objects in the Repository have a unique identifier. It is this identifier and the object type, sometimes in combination with a qualifier, which is used by the import to determine if an object already exists in the repository.

With the introduction of names, every object effectively has an extra attribute, i.e. its (long) name, and like other attributes the name may be changed as a result of an import. However, changing an object's name could have serious implications, for example the name may be referred to in a file or field, or it may be directly referenced in RDMLX code.

Select this option to proceed with the import and apply any name changes to existing objects.

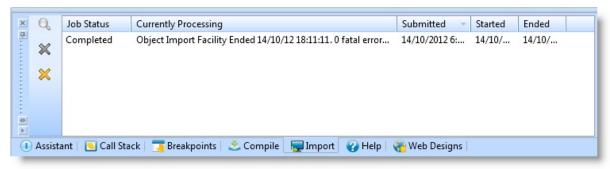
Allow Type Changes

It is not uncommon to import an object that already exists in your repository; after all, you are just replacing one definition with another but it is less likely that you're replacing say, a Field with a Process, or an Icon with a Form, so to prevent you from accidently deleting object definitions, the import will not automatically proceed if it would mean the same identifier will end up being used for a different type of object.

If it's an intentional change however, you can select this option and proceed with the import. In this case the existing object will be deleted and the new object definition will be imported.

4.3.1 Import Tab

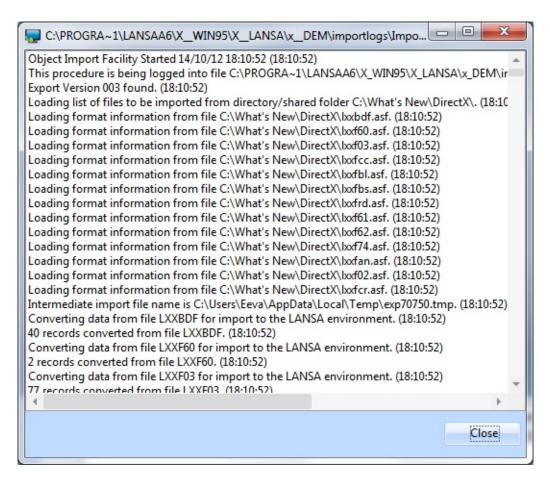
The Import tab shows the status of any import jobs, such as when they were submitted and finished, or pending, indicating that you have started using the 4.3 LANSA Import Facility.



Use the toolbar's icons to control the contents of this tab. They are:

- Show import log
- **X** Remove the selected job details from the import tab.
- Remove all pending and complete job details.

You can open the import log file by either double-clicking the job directly or by selecting the job in the Import tab's list and pressing the Show import log button: 4.



You will find the log files in the importlogs folder of the current partition directory.

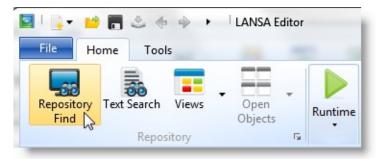
1 4.3 LANSA Import Facility

4.4 Search Facilities

The LANSA Editor contains a number of facilities to enable you to find what you need.

These are:

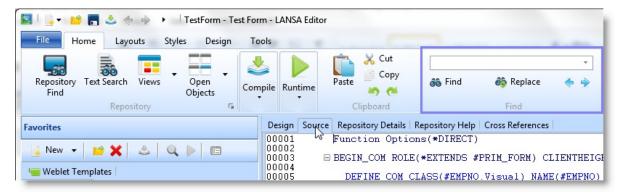
- 4.4.1 Repository Find which can be opened with:
 - The Repository Find command in the ribbon:



- Using the Find command in the context menu that opens when you right-click one of the Repository items.
- 4.4.2 Text Search which you open using the Text Search command:



• 4.4.3 Source Code Find and Replace using the Find commands in the ribbon:



↑ 4. Editor Features

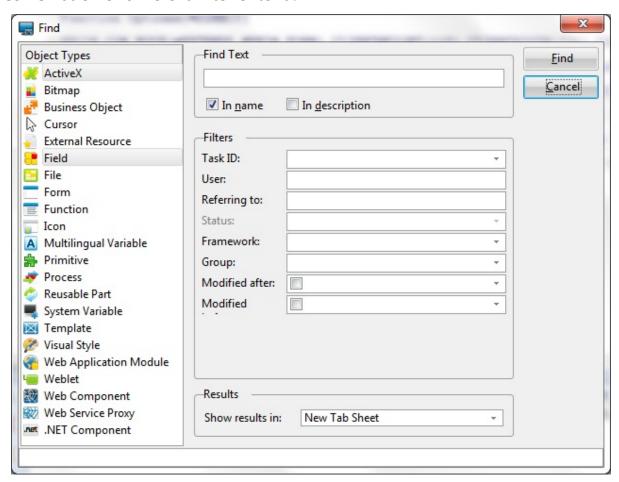
4.4.1 Repository Find

Repository Find can be opened with the Repository Find command in the ribbon:



Or using the Find command in the context menu that opens when you rightclick one of the Repository items.

Use the Find dialog to search though single or multiple object types, using a combination of different filter criteria:



Object Type

In the Object Types list, select the object types that will be included in the search. One or more object types may be selected from the list - press the Shift key while you click on the object types to be included in your search.

The specified criteria in the Find dialog will be applied to the object types selected. The criteria that can be entered will be determined by the object type(s) selected. For example, status may only be used with objects that are compiled and Framework may only be used with objects stored in Frameworks.

Find Text

Enter a whole or partial string. For example:

- ABC searches for any string containing ABC, (for example *ABC*)
- ABC* searches for any name that starts with ABC
- *ABC searches for any name that ends with ABC

Use the In name and In description check boxes to specify whether the string is searched in the name or the description of the selected Object Types or both.

Filters

Specify the filters that will be used in the search. All specified criteria will be combined using AND logic.

Task ID

Choose a task ID you want to use as search criteria. All tasks defined for the current partition can be used.

You can use the System Initialization options to refresh the task list if using a LANSA for iSeries System.

User

Specify a user ID to use as search criteria.

You can use the System Initialization options to refresh the user list if using a LANSA for iSeries System

Referring To

Specify the name of an object for which you want to find cross-reference information.

Note: Objects that can be compiled must be compiled for their cross-reference information to be returned.

Status

Select the object status of the selected Object Type to be

searched. More than one status can be selected.

This criteria can only be specified if the selected object can be compiled.

Framework Select the Framework you want to use as search criteria. More than one Framework can be selected.

> This criteria can only be specified if the selected object is stored in a framework.

Refer to Frameworks in the Administrator Guide.

Group

Select the Group you want to use as search criteria. More than one Group can be selected.

This criteria can only be specified if the selected object is stored in a group.

Refer to **Groups** in the *Administrator Guide*.

Modified After

Select the first modified date of the objects to be searched, i.e. all objects that have been modified on or after this date.

Modified Before

Select the last modified date of the objects to be searched, i.e. all objects that have been modified on or before this date.

Results

Using the Show results In option, specify the tab in which you want the search results to be displayed.

By default:

- the result will appear in a tab in the left-hand tab folder group
- the first search results will be placed in a tab called Find Results 1.

Specify **New Tab Sheet** in Show results in as the location of the find results if you don't want them added to the end of the last results found.

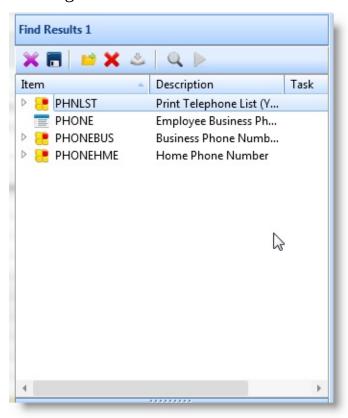
Also see

Find Results

↑ 4.4 Search Facilities

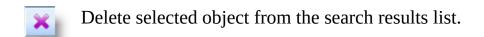
Find Results

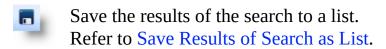
In the Results output tab, you can work with any of the found objects in the list. For example, you can open or delete, or depending on the type of object, compile, execute or debug:



You can save the search results as a list which is described in Lists.

Find Toolbar





Open the object that you have selected in the list.

Delete selected object from the repository. You will be asked to confirm the deletion.

- Compile selected executable object. Refer to Compile Options.
- Execute selected executable object Refer to Executing Applications.
- Debug selected executable object Refer to Debugging Applications.

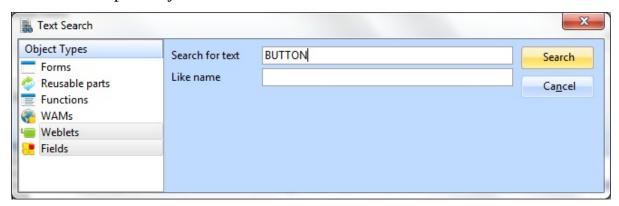
↑ 4.4.1 Repository Find

4.4.2 Text Search

Text Search is opened using the Text Search command:



The Text Search facility allows you to search within all the LANSA Source code in the Repository.



Object Types

You must select at least one Object Type in the list.

Search for text

Enter the text that you wish to search for.

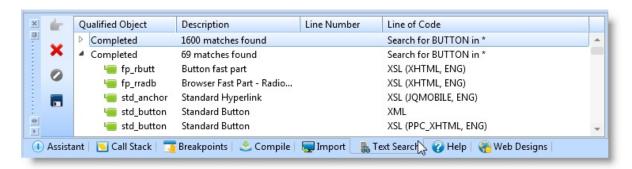
Like name

Enter a whole or partial string. For example:

- ABC searches for any string containing ABC, (for example *ABC*)
- ABC* searches for any name that starts with ABC
- *ABC searches for any name that ends with ABC

Press the Search button to start the search.

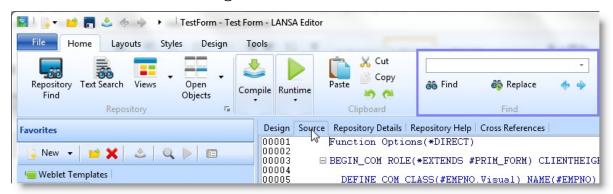
The results are displayed in the Text search tab:



↑ 4.4 Search Facilities

4.4.3 Source Code Find and Replace

Search code is searched using the Find commands in the ribbon:



These Find commands are only available for use with the Source tab.

You can find the string you wish to locate by entering it in the Find entry field in the toolbar and pressing Enter.

You can open the Find dialog, which is similar to the standard Windows text Find dialog, by:

- using the Find command in the ribbon
- pressing the Ctrl + F buttons

You can browse backwards or forwards through your search results using the forward and back buttons.

To use the Replace facility:

- use the Replace command in the ribbon
- press the Ctrl + H buttons.

For further information, refer to the Source tab's Find and Replace.

1 4.4 Search Facilities

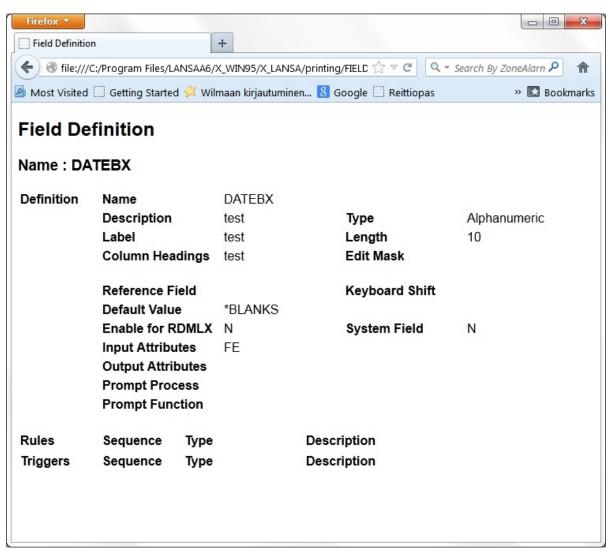
4.5 Interface Indicators

- A red triangle vindicates an error or that an entry is compulsory.
- A gray/brown ▼ triangle indicates a warning.

4.6 Printing in Visual LANSA

When you print object definitions in LANSA, note that there are two different ways of doing it:

- Objects with source code (function, reusable part, form etc.) can be printed by selecting Print from the File menu when the object is open in the editor.
- Field, File and Process (objects without source code) can be printed using the context menu in the Repository Tab once you have selected the object to be printed. The requested information will be displayed in a browser window where the Print Option on the File menu can be selected to print the information.



Also See

- 4.6.1 Print Dialog
- ↑ 4. Editor Features

4.6.1 Print Dialog

Use the options in the Print dialog to print the LANSA object definitions of the selected object.

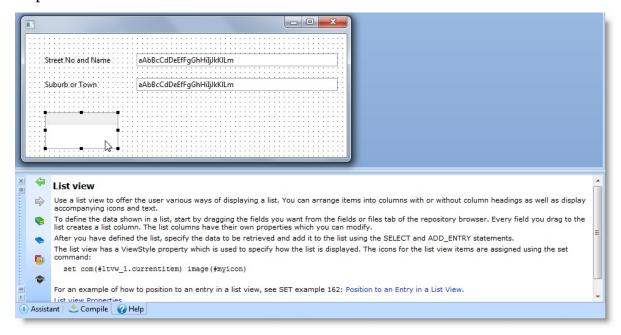
If you are printing plain text on a local printer, the printer must be configured to LPT1. Or if you are using LPT2, you must add PRTR=LPT2 to your x_lansa.pro file in the X_LANSA directory.

When you print formatted text, the printing is handled either by Microsoft Word or your browser.

1 4.6 Printing in Visual LANSA

4.7 Help Text

Help text provides information about the object you are working with. To see the help text, select an object and press F1. The information is displayed in the Help tab:



Online Help Toolbar

The toolbar is used for navigation, but it also provides you with access to all the LANSA documentation.

In the toolbar with the F1 help:

- Go back to a previous topic that has been displayed. (Can only be used after more than page has been displayed.)
- Go forward to a topic that has been displayed. (Can only be used if the Back button has been used.)

In the toolbar with both the full guides and the F1 help:

Browse

Opens a new window displaying **all** the LANSA guides so that you can search for a topic in all the LANSA documentation.

Opens a new window showing the full guide that contains the topic Contents you are currently viewing. The guide opens with a table of contents. Based on the example above, pressing the Contents button will open the Technical Reference Guide with the File Name topic

displayed.

Opens a new window displaying the SET (Samples, Examples and Templates) documentation.

Opens a new window displaying a list of all the Tutorials supplied with LANSA. From the opened window, you can select and view any LANSA tutorial.

If you wish to print a page, right click with the mouse and select the Print option from the context menu.

Also See

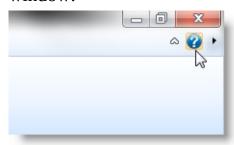
4.7.1 LANSA Online Documentation

4.7.1 LANSA Online Documentation

LANSA online documentation is installed as part of the LANSA software. A full set of the LANSA Guides are installed.

Note that the file format used, CHM (Compressed or Compiled HTML) does not always display from a network drive without some changes to the Registry. For best results, these files should be stored and viewed on the same PC.

You can access the help text using the help button in the top right of the editor window:



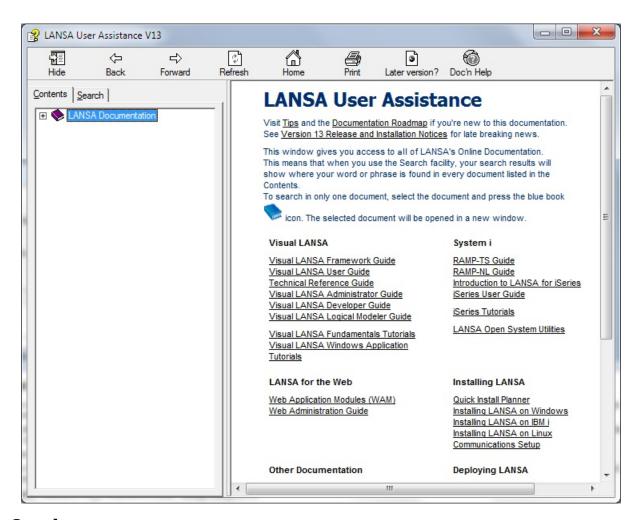
The Global Guide will give you access to all LANSA guides and from that guide you can open an individual guide using the icon, if you wish.



Press the Later version? button if you wish to check if there are any updates to the documentation on the LANSA web site at www.lansa.com/support. If there are any new guide versions, they will be listed. To check whether a version is more recent than your documentation, compare the Edition Date on your guide. Place any downloaded guides in the same directory in which you have all the other LANSA online documentation.



The Doc'n Help button will open the Tips guide, which contains helpful hints to enable you to use the LANSA documentation more effectively. Of particular interest is the section on Searching, which has quite extensive features. Try it!



See also

Tips for Using the Online Documentation for time saving tips when using the online documentation, especially the extensive Search facilities.

↑ 4.7 Help Text

5. Executing Applications

5.1 Execute	5.2 Execute Function	5.3 Execute	5.4 Execute
Process		Form	WebRoutine
5.5 Execute	5.6 Additional Execution	5.7 Fatal	
Options	Parameters	Errors	

Applications are created as functions in a process, as forms or as WAM webroutines.

To execute an application from within Visual LANSA, you compile it and then execute it using one of the following:

• the Execute command in the ribbon:



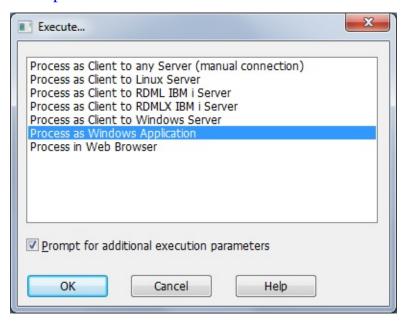
• the Execute option from a context menu in the Repository Tab or the Favorites Tab

Also See

Debugging Applications

5.1 Execute Process

Choose the appropriate execution option for your application. For details, refer to the 5.5 Execute Options.

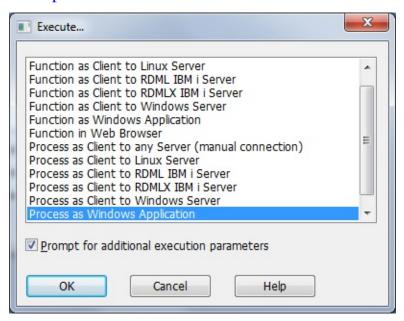


Select Prompt to choose or vary 5.6 Additional Execution Parameters.

↑ 5. Executing Applications

5.2 Execute Function

Choose the appropriate execution option for your application. For details, refer to the 5.5 Execute Options.

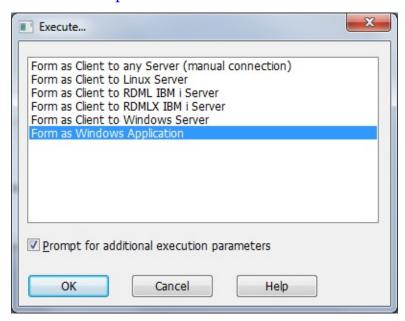


Select Prompt to choose or vary additional execution options.

↑ 5. Executing Applications

5.3 Execute Form

Choose the appropriate execution option for your application. For details, refer to the various 5.5 Execute Options.



Select Prompt to choose or vary 5.6 Additional Execution Parameters.

↑ 5. Executing Applications

5.4 Execute WebRoutine

A Webroutine can be executed from within the Editor in one of the following ways:

- **In the Source tab:** Right mouse click on the green arrow to the right of the webroutine statement. From the menu choose Execute and the Technology Service Provider you wish to use.
- **In the Source tab:** Right mouse click on the webroutine statement. From the menu choose Webroutine: <name>, Execute and the Technology Service Provider you wish to use.
- **In the Design tab:** Choose the Execute command on the ribbon.
- **In the Design tab:** Press Ctrl+Shift+E.

Example of execute URL:

http://localhost/CGI-BIN/lansaweb? webapp=MYWAM+webrtn=MYWR+ml=LANSA:XHTML+part=DEX+lan

Note that only saved files are executed. The WAM being edited is not saved to the repository by executing it. You must save the WAM first if you wish to execute the WAM with your most recent changes to the Web Design.

1 5. Executing Applications

5.5 Execute Options

Choose the appropriate execution option for your application. Note that the type of application determines what execution options are available.

- 5.5.1 As Client to Any Server (manual connection)
- 5.5.2 As Client to RDML IBM i Server
- 5.5.3 As Client to RDMLX IBM i Server
- 5.5.4 As Client to Linux Server
- 5.5.5 As Client to Windows Server
- 5.5.6 As Windows Application
- 5.5.7 In Web Browser
- 5.5.8 XML Java Client
- 5.5.9 Prompt for additional execution Parameters
- ↑ 5. Executing Applications

5.5.1 As Client to Any Server (manual connection)

Use this facility to execute a LANSA application in client/server mode with support for RDMLX. This means that all your RDML and RDMLX level database access requests will be automatically re-routed into your remote server database.

Technical Note 1: Your application will not know that a remote database is being used. If your application has been designed so that it is capable of connecting and disconnecting itself from the server database, you should NOT start it via this interface. Use the Execute a process on the workstation facility instead.

Technical Note 2: Special provisions apply to running batch style applications from this interface. Refer to the documentation of the CONNECT_FILE built-in function before attempting to run batch style programs that contain SELECT-UPDATE WITH_RRN()-ENDSELECT or SELECT-DELETE WITH_RRN()-ENDSELECT loops.

Also See

5.6 Additional Execution Parameters

5.5.2 As Client to RDML IBM i Server

Use this facility to execute a LANSA application in client/server mode with support for RDML only. This means that all your RDML level database access requests will be automatically re-routed into your remote server database.

Technical Note 1: Your application will not know that a remote database is being used. If your application has been designed so that it is capable of connecting and disconnecting itself from the server database, you should NOT start it via this interface. Use the Execute a process on the workstation facility instead.

Technical Note 2: Special provisions apply to running batch style applications from this interface. Refer to the documentation of the CONNECT_FILE built-in function before attempting to run batch style programs that contain SELECT-UPDATE WITH_RRN()-ENDSELECT or SELECT-DELETE WITH_RRN()-ENDSELECT loops.

Also See

5.6 Additional Execution Parameters

5.5.3 As Client to RDMLX IBM i Server

Use this facility to execute a LANSA application in client/server mode with support for RDMLX. This means that all your RDML and RDMLX level database access requests will be automatically re-routed into your remote server database.

Technical Note 1: Your application will not know that a remote database is being used. If your application has been designed so that it is capable of connecting and disconnecting itself from the server database, you should NOT start it via this interface. Use the Execute a process on the workstation facility instead.

Technical Note 2: Special provisions apply to running batch style applications from this interface. Refer to the documentation of the CONNECT_FILE built-in function before attempting to run batch style programs that contain SELECT-UPDATE WITH_RRN()-ENDSELECT or SELECT-DELETE WITH_RRN()-ENDSELECT loops.

Also See

5.6 Additional Execution Parameters

5.5.4 As Client to Linux Server

Use this option to execute a LANSA application under Windows but with the files on a non-IBM i data server. It is similar to the LANSA Folder shortcuts "Exec Process (CS Mode)" & "Exec Form (CS Mode)", except that the connection is to a Linux Server and the parameters that the development environment can deduce are not shown.

The Linux Host must be configured to support this execute option and the files and remote functions must be compiled on the Linux host.

Note that this is only one method of executing an application that uses SuperServer. If an INIT function is used to connect to a Server then a custom execute option will be required. If the application uses Built-In Functions such as DEFINE_OS_400_SERVER then the Client to any Server execute option should be chosen.

Also See

5.6 Additional Execution Parameters

5.5.5 As Client to Windows Server

This executes an application under Windows but with the Files on a Windows data server. It is similar to the LANSA Folder shortcuts "Exec Process (CS Mode)" & "Exec Form (CS Mode)", except that the connection is to a Windows Server and the parameters that the development environment can deduce are not shown.

The Windows Host must be configured to support this execute option and the files and remote functions must be compiled on the Windows host.

If you are using a Built In Function such as DEFINE_OS_400_server, or if you are using an INIT parameter, you should choose the Windows Application execute option instead of this one.

Also See

5.6 Additional Execution Parameters

5.5.6 As Windows Application

Use the Windows Application option to execute the application entirely in Windows. If the Prompt is selected then another dialog box is displayed which will allow you to choose or vary:

- The default Printer
- To run in Debug mode
- To run with Trace on.

These options have exactly the same meaning as in the Exec... shortcuts in the LANSA folder on the Desktop. The only difference is that all the parameters that the development environment already knows about are not shown.

When OK is pressed, the application is executed, unless the dialog is displayed from the Editor, in which case the options chosen are saved for use when the Execute or Execute Debug option is selected on the Editor's toolbar. Note that the Debug value will be based on whether Execute or Execute Debug was selected, not on what was chosen in the last execution.

When the Prompt is not selected, the Prompt dialog is not shown and the application is executed as it was the last time. Note that edit boxes of the same name in different dialogs are saved with the same values. Therefore, if a function was executed in debug mode (debug value set to 'Y'), then a form was executed next, the Debug value would also be set to 'Y'.

Also See

5.6 Additional Execution Parameters

5.5.7 In Web Browser

This executes a function or process in a Web Browser on a LANSA Web Server. The LANSA Web Server must be configured to support this execute option and all objects must be compiled on the LANSA Web Server and have HTML generated.

Also See

5.6 Additional Execution Parameters

5.5.8 XML Java Client

This executes an object in the LANSA Java Client application on a LANSA Web Server.

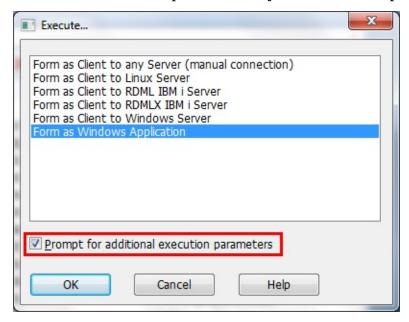
The IBM i Host must be configured to support this execute option and all objects must be compiled on the IBM i and have XML generated.

Also See

5.6 Additional Execution Parameters

5.5.9 Prompt for additional execution Parameters

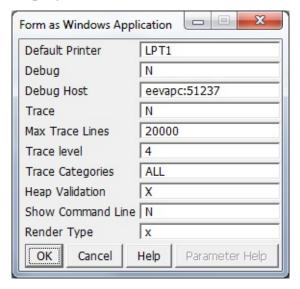
Select this option to select further options to vary the execution process.



The 5.6 Additional Execution Parameters dialog will open after you have pressed the OK button.

5.6 Additional Execution Parameters

Be selecting the Execute dialog option to Prompt for additional parameters, a further dialog will be displayed:



These parameters can be set:

- 5.6.1 Default Printer
- 5.6.2 Debug
- 5.6.3 Debug Host
- 5.6.4 Trace
- 5.6.5 Max Trace Lines
- 5.6.6 Trace Level
- 5.6.7 Trace Categories
- 5.6.8 Heap Validation
- 5.6.9 Show command line
- 5.6.10 Render Type
- 5.6.11 Graphics Processing

Also See

- **5.5** Execute Options
- ↑ 5. Executing Applications

5.6.1 Default Printer

This parameter specifies the printer port through which reports are printed. Typically it is specified as LPT1 or LPT2.

This parameter maps to the Visual LANSA X_RUN parameter PRTR.

5.6.2 Debug

This parameter specifies whether the application is to be started in debug mode either locally or on the server.

Specify Y to start in Debug or Remote Debug mode or N to start in normal mode.

Refer to Debugging Applications for details of the Debug facilities provided with LANSA.

5.6.3 Debug Host

The address of the machine from which you wish to debug. This name/address will be obtained from the Debug options of the LANSA Settings but it can be amended here.

Also See

Debug in the LANSA settings.

Debugging Applications

5.6.4 Trace

This parameter specifies whether the application is to produce a trace file. Specify Y to produce a trace file or N to not produce a trace file. Trace files are named X_tracennn.txt. The highest nnn suffix indicates the newest trace file. The production of trace files severely impacts application performance. This parameter maps to the Visual LANSA X_RUN parameter ITRO.

Also See

Trace Parameters in the Technical Reference Guide

5.6.5 Max Trace Lines

This parameter specifies the maximum number of lines in the trace file. The maximum number you can enter is 999,999,999.

This parameter maps to the Visual LANSA X_RUN parameter ITRM.

Also See

Trace Parameters in the Technical Reference Guide

5.6.6 Trace Level

This parameter specifies the level of trace required. Valid values are 0 to 9, where 0 provides the lowest level of detail and 9 the highest level of detail. This parameter maps to the Visual LANSA X_RUN parameter ITRL.

Also See

Trace Parameters in the Technical Reference Guide

5.6.7 Trace Categories

This parameter specifies the trace categories. It allows you to restrict the areas of LANSA that will generate trace messages. This should not be changed unless requested by your Product Vendor. Use of this value is described in Microsoft Exception. Multiple values can be specified at a time as one string, e.g. DBMUIM.

This parameter maps to the Visual LANSA X_RUN parameter ITRO.

ALL All categories

DBM Database only

UIM User Interface only

FUN Standard Function only

PIM Printer functions only

COM Communications only

PDF Platform Dependent Functions only

BIF Built-In Functions only

PRO Reserved

RDM RDML only

RDX RDMLX only

HEP Heap Validation only

Also See

Trace Parameters in the Technical Reference Guide

5.6.8 Heap Validation

This parameter specifies the heap validation level. This should not be changed unless requested by your Product Vendor. Use of this value is described in Microsoft Exception. Note that it does not require ITRO=Y in order to validate the heap. Setting ITRO=Y will just add trace messages to the heap validation. They are often used in conjunction in order to provide detailed diagnostic information for use by your Product Vendor.

This parameter maps to the Visual LANSA X_RUN parameter ITHP.

- X Use default as set in code (N for GA versions, G for internal debug versions)
- N No heap validation
- G Guard bytes and validate pointer
- P Validate pointer only
- H P + validate whole of heap that the pointer is in
- A H + validate all heaps
- T H + trace validations
- Z A + trace validations

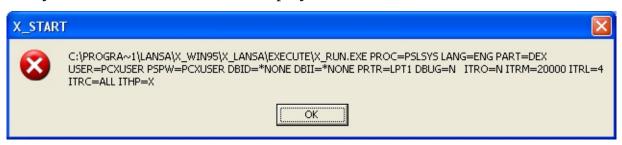
Also See

Trace Parameters in the Technical Reference Guide

5.6.9 Show command line

To see the command line created from your execute options, enter Y here, otherwise enter N.

If Y, your command line will be displayed like this:



↑ 5.6 Additional Execution Parameters

5.6.10 Render Type

Specify here the render engine for the Visual LANSA GUI. Valid values are:

W Win32 Traditional Windows applications.

X DirectX rendering engine.

M Application Individual components can specify DirectX

defined. rendering.

5.6.11 Graphics Processing

Specify here whether the Graphics Processing Unit (GPU) will be used to render the graphics when running as DirectX.

- H Hardware the GPU will be used if the PC has one.
- S Software do not use the GPU. Choose this option if the grapics processing results in unusual screen painting.

5.7 Fatal Errors

When you run your application, you will be asked to select the additional parameters you require.

If any Fatal errors are found, a dialog box will display the details of the error. You can note these details for later investigation, or you can press OK and review the errors later.

Also see

5.7.1 Visual LANSA Error Log Viewer

5.7.2 Web Runtime Error Log Viewer

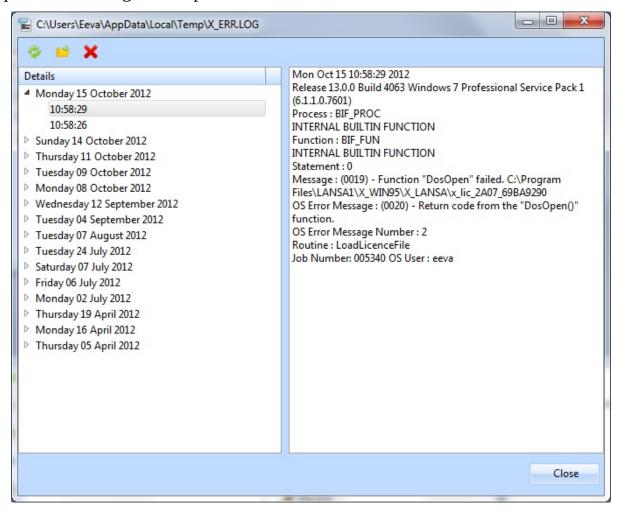
↑ 5. Executing Applications

5.7.1 Visual LANSA Error Log Viewer

Use the Error Logs command to display and choose Local to view Visual LANSA error logs:



The time and date of the error you are looking for are shown on the left Details pane, and the right hand pane shows the related details:



The Details pane lists the most recent errors sorted by date. Only the latest 250 messages are shown in the dialog. If you wish to view a complete list of all errors, open the text file containing previous messages using the Open icon

on the toolbar.

Control the contents of this dialog using these toolbar icons:

- Update the viewer with the latest error messages if there are any that have not yet reached the list.
- This icon opens the actual X_Error Log file in text format so that you can see the messages that are no longer displayed in this dialog.
- Press this icon to delete the message that is highlighted in the Details pane or to delete all the messages listed.

You can keep this modeless dialog open when developing.

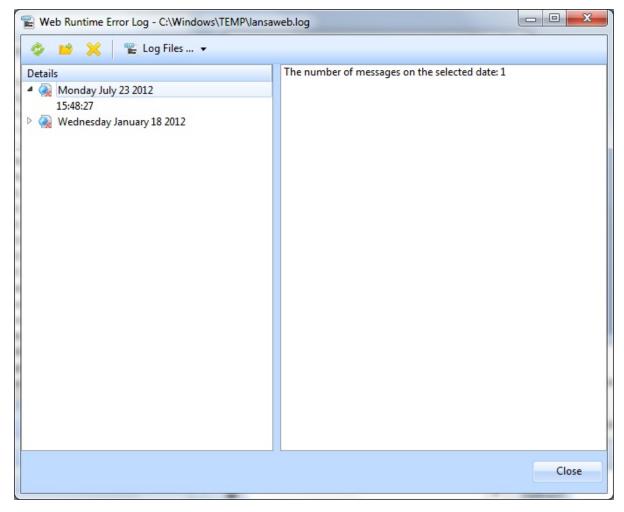
↑ 5.7 Fatal Errors

5.7.2 Web Runtime Error Log Viewer

Use the Error Logs command to display and choose Web to view web runtime error logs:



The Web Runtime Error Log Viewer is designed to help you to locate fatal errors in your code when you are developing web-related applications.



The left pane lists the latest errors sorted by date and time, and the right pane shows details for the error selected in the left pain.

Update the viewer with the latest error









messages - if there are any that have not yet reached the list.

This icon opens the actual Web Runtime Error Log file in text format, if that is the one you are accustomed to using.

Press this icon to delete the message that is highlighted in the Details pane or to delete all the messages listed.

Two options available:

Data/application Server: This is where the LANSA application runs. Look here for application errors.

Web server: Process requests from user agents (browsers). Look here for problems connecting to the Data/application server or XSL transformations (when transforming on the web server).

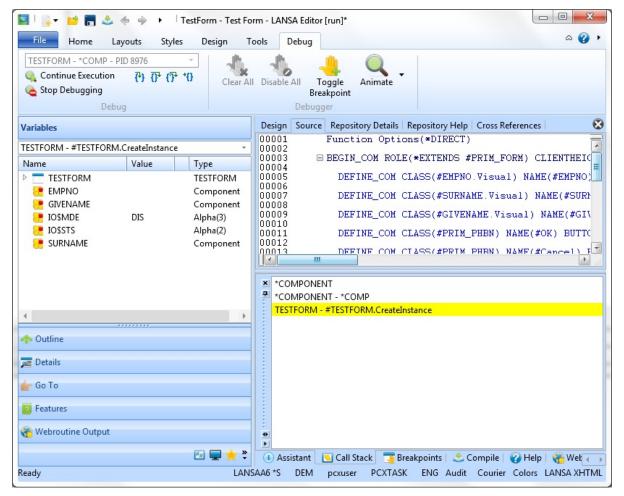
You can keep this modeless dialog open when you are developing.

↑ 5.7 Fatal Errors

6. Debugging Applications

6.1 Start Debugging 6.2 Set Breakpoints 6.3 Variables 6.4 Call Stack

6.5 Run Parts of the 6.6 Exclude/Include 6.7 Animate 6.8 Debug Application Applications Application Execution WAMs



From the LANSA Editor you can debug Forms, Functions, WAMs or Web Functions running locally or on an IBM i or Windows Data/Application Server. The server can be a machine different from the one running the Development Environment.

From the LANSA editor you can:

- Single step through code to examine how the logic flows.
- Set break points in code to stop at places of interest on demand.
- Watch/modify field values on the fly to identify and rectify problems

temporarily or change the logic flow for testing purposes.

The object you are debugging must be compiled first with the relevant debug option enabled. (Refer to Compile Tab.) If you are making changes to the source code, make sure you have recompiled it to ensure that the latest changes are present.

When the source code is opened during a debug session, it can be edited. However, once you have completed editing the source code you should immediately terminate the debug session. Do not continue to use the debugger once the source code has been edited as your source code is no longer synchronized with the debug session.

You can view information about the code you are debugging by displaying additional debug tabs:

- 6.2.1 Breakpoints
- 6.3 Variables
- 6.4 Call Stack

To display these tabs, use the Windows command on the ribbon.

Also See

Interactive Debugging in the Web Administrator Guide.

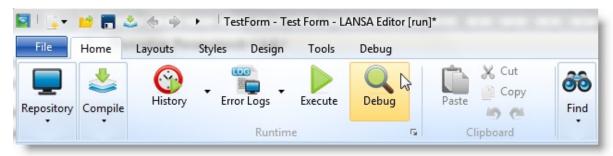
Additional Execution Parameters

↑ 6. Debugging Applications

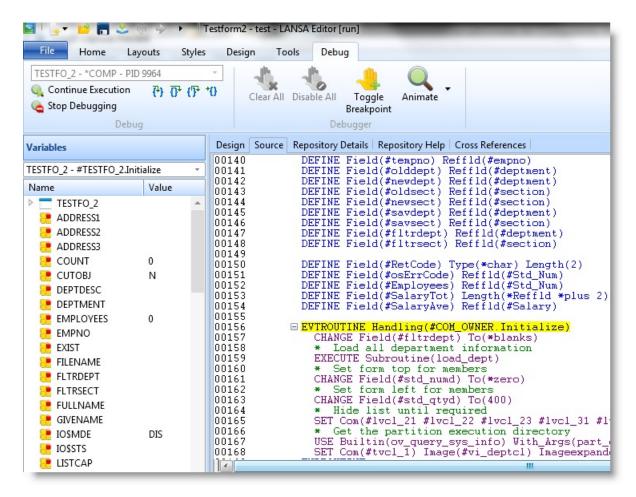
6.1 Start Debugging

You can start the debug of a compiled LANSA program using one of the following:

• Use the Debug command in the ribbon:



- the debug and start option from a context menu in the Repository Tab or the Favorites Tab
- the debug parameter when specifying the Additional Execution Parameters When the object to be debugged is executed, it is opened in the editor. If no breakpoints are set, the debug session immediately stops at the first executable line so you can set breakpoints or start stepping through the code.



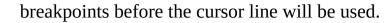
By default, lines with enabled breakpoints are shown in red and the active debug line is shown in yellow. Disabled breakpoints are shown in gray.

Debugger Commands

Continue Execution	Continue execution of the program.
Stop Debugging	The debug session will be terminated and the program execution cancelled.
(*)	Step Into. Refer to 6.5 Run Parts of the Application.
<u>O</u> +	Step Over. Refer to 6.5 Run Parts of the Application.
T	Go back to the calling procedure. Refer to 6.5 Run Parts of the Application.
*0	Execute the program and break at the current cursor line even if not marked as a breakpoint. Note: All



Disable All



Insert a breakpoint on the selected line, or if a breakpoint already exists, remove it from the line.

Clear all breakpoints.

Disable all breakpoints.



Animate. See 6.7 Animate Application Execution.

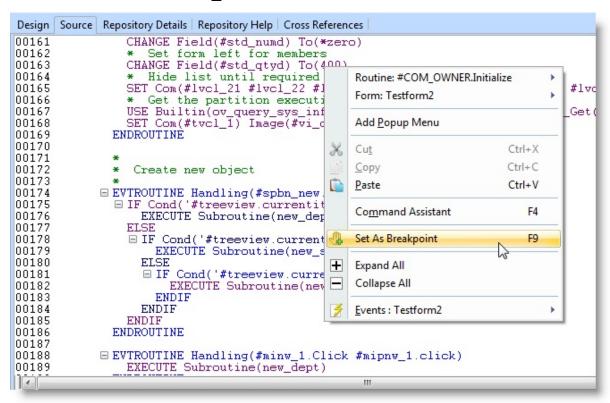
↑ 6. Debugging Applications

6.2 Set Breakpoints

A breakpoint tells the debugger to halt just before executing a specific line of code. Refer to 6.2.1 Breakpoints and 6.2.2 Breakpoint Properties.

You can mark breakpoints in the code at any time. The debugger does not have to be active to mark the breakpoints. The breakpoints are saved as part of the source code definition.

To set a breakpoint, put the cursor on a statement, use the Toggle Breakpoint command in the ribbon or right-click and choose Set As Breakpoint from the context menu. You can also press the F9 key. Only lines with executable commands may have breakpoints. For example, you cannot set a breakpoint on a comment line or GROUP BY command.



You can open other objects for editing during the debug session and set breakpoints where required.

If an object is not already open in the editor and a breakpoint has been set on it, the object will be opened automatically when the breakpoint is encountered.

When you start debugging an application, the execution will stop at the statement which has the first breakpoint set. When the application is halted, you can examine the application's current state by referring to the 6.3 Variables and

6.4 Call Stack.

To continue debugging, either use the Continue Execution command (F5), or use the Step Into (F8) and Step Over (Shift + F8) commands to move forward one statement at a time.

To disable or remove breakpoints, use the Toggle Breakpoint command or the Breakpoints Toolbar in the 6.2.1 Breakpoints tab.

Also See

6.2.1 Breakpoints

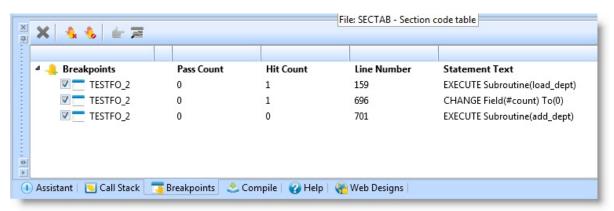
6.2.2 Breakpoint Properties

↑ 6. Debugging Applications

6.2.1 Breakpoints

The Breakpoints window displays a list of all lines that have been marked as breakpoints. The window displays debug statistics and allows you to reset the program breakpoints.

To display the Breakpoints tab, use the Views command in the ribbon. By default, the Breakpoints window will appear as a docked tab in the bottom pane of the editor.



The checkboxes are used to enable or disable breakpoints already added for a specific line. To add breakpoints to a new line, use the F9 key.

Breakpoints Toolbar

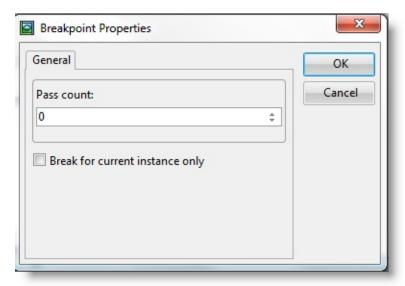
- **X** Delete a selected breakpoint in the list.
- Clear all breakpoints set in the open application.
- Disable or enable all breakpoints in the open application.
- Display the selected statement in the Source view.
- Display the 6.2.2 Breakpoint Properties dialog.

↑ 6.2 Set Breakpoints

6.2.2 Breakpoint Properties

Use the Breakpoint Properties dialog to set the pass count for the breakpoint.

To display the Breakpoint Properties dialog, press the Breakpoints icon in the Toolbar.



The pass count is the number of times a line with a breakpoint is to be executed before the debug facility stops the LANSA application running and displays the line.

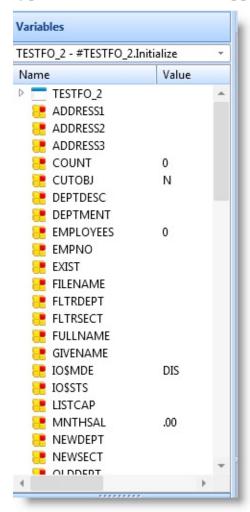
For example if the pass count is 3, the LANSA application must execute the command on the specified line twice, and on the third time the debug facility will halt the LANSA application and display the line on which the breakpoint is set.

↑ 6.2 Set Breakpoints

6.3 Variables

Use the Variables window to work with the variables in the application being debugged. Refer to 6.3.1 Include/Exclude Fields and 6.3.2 Changing Values.

You can view the value of variables at any point during execution. The Window shows the value and the type of the variables in the application being debugged.



Variables Context Menu

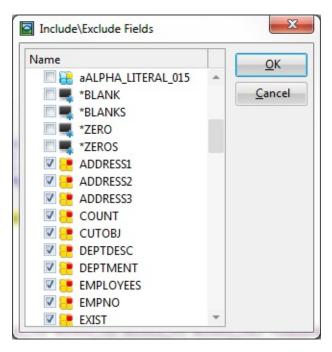
The items on this context menu will vary depending on the context in which it is opened.

You can view the variables of another function, process or component by using the 6.4 Call Stack.

6.3.1 Include/Exclude Fields

Use the Include/Exclude Fields dialog to specify (by a tick) which individual variables are shown on the first level of nesting in the Variables tab.

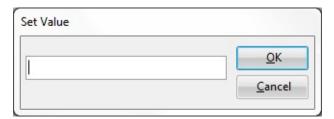
To display the Include/Exclude Fields dialog, use the Include/Exclude option from the context menu when you have selected a variable in the 6.3 Variables tab.



↑ 6.3 Variables

6.3.2 Changing Values

Use the Set Value dialog to specify the new value for the selected variable. The current program value of the variable will be initially displayed in the dialog. To display the Change Value dialog, double-click on the variable in the 6.3 Variables Window or select the Set Value option from the Variables context menu.



It is important to remember that breakpoints occur before the command line is executed. For example, if you set a breakpoint on an INSERT command and change the value of a variable, the changed value will be used by the INSERT command when the program executes.

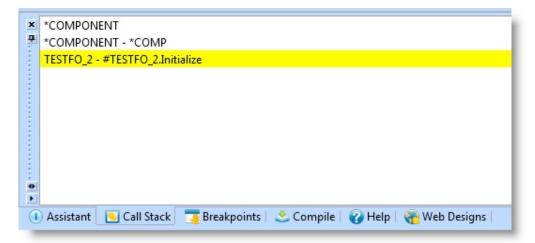
↑ 6.3 Variables

6.4 Call Stack

The Call Stack tab shows a list of the processes, functions and components that have been invoked in order to get to the current function or component.

To display the Call Stack tab, use the Views command in the ribbon.

By default, the Call Stack tab will appear as a docked tab in the bottom pane of the editor.



The Call Stack tab helps you trace the operation of an application as it calls other objects.

You can view the variables of another function, process or component in the call stack by double clicking on it. The line will be highlighted in green and the 6.3 Variables will change to show the variables in the program selected in the Call Stack window.

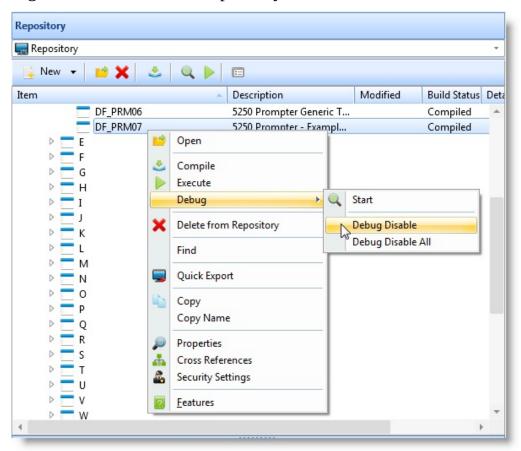
6.5 Run Parts of the Application

If you only know the general area of the code causing an error, you can use a breakpoint to isolate the problem area, and then use toolbar icons or function keys to step into (F8) and step over (Shift + F8) to view each statement.

- Step Executes the current statement and breaks at the next line Into You can use Step Into to execute code one statement at a time. After (F8) stepping through each statement, you can see its effects by looking at the Variable window. **○** Step Execute the procedure called by the current line and break at the line following the current line Over (Shift + Step Over is identical to Step Into except when the current statement F8) contains a call to a procedure, Step Over executes this procedure as a unit and then steps to the next statement in the current procedure. (F Step Go back to the calling procedure. If you are using Step Into and have been moved to a called procedure, you can automatically run the Out current procedure, and return to the calling procedure.
- ↑ 6. Debugging Applications

6.6 Exclude/Include Applications

You can exclude and include individual or all applications to be debugged using the Debug context menu in the Repository tab:



Debug DisableDisable Disable Disa

Debug Enables debugging for all objects in the repository that can be debugged.

DebugDisables debugging for all objects in the repository that can be debugged.

You can also select Start from the context menu to start debugging.

6.7 Animate Application Execution

To view application execution in slow motion, use the Animate and Animate All commands. The speed at which the application executes is set using the Delay option.

Animate In animate mode, the debug automatically steps from one command

to the next while showing the line currently being executed.

You can set the delay between commands using the Delay

command..

Animate

All

This option is otherwise the same as Animate, except that it also executes functions or components which have been marked as not

debuggable.

You can set the delay between commands using the Delay

command.

Delay Use this option to specify the delay between the execution of

commands. The delay can be:

50 milliseconds

500 milliseconds

1 second

2 seconds.

6.8 Debug WAMs

A Webroutine can be executed for debugging from within the Editor in the following ways:

- **In the Source tab:** Right mouse click on the green arrow to the right of the webroutine statement. From the menu choose Debug and the Technology Service Provider you wish to use.
- **In the Source tab:** Right mouse click on the webroutine statement. From the menu choose Webroutine: <name>, Debug and the Technology Service Provider you wish to use.
- **In the Design tab:** Choose the Debug toolbar button.
- **In the Design tab:** Press Ctrl+Shift+D.

Example of debugging URL:

http://localhost/CGI-BIN/lansaweb?

webapp=MYWAM+webrtn=MYWR+ml=LANSA:XHTML+part=DEX+lan

Note that Debug uses saved files. The WAM being edited is not saved to the repository by debugging it. You must save the WAM first if you wish to debug the WAM with your most recent changes to the Web Design.

Also see

Interactive Debugging in the Web Administration Guide.

7. Tracing Applications

Visual LANSA provides a tracing feature that you can embed in your own RDML applications.

Tracing must be designed and integrated into your application during development with the inclusion of tracing statements in your RDML code at critical points of the application processing.

Use tracing as an alternative approach to debugging. It is especially useful for analyzing problems in deployed applications as a switch mechanism can be built into your application to allow end users to activate the tracing without any technical knowledge of the application.

To utilize Visual LANSA tracing in your application:

- 7.1 Create a Trace Handler
- 7.3 Use the Trace Handler

To obtain information from tracing refer to:

- 7.5 Trace Output
- 7.6 Trace Handler Guidelines and Performance

7.1 Create a Trace Handler

Firstly, you need to create an object in your application to act as the trace handler. This object must implement the iTraceHandler interface supplied with Visual LANSA which has the following methods available:

- Initialize Invoked when the trace handler is installed as the active trace handler for the application
- Terminate Invoked when the trace handler is uninstalled
- TraceMessage Called when either the TraceMessageText or TraceMessageData method is invoked in your application RDMLX code.
- TracingState Called when either the TraceMessageText or TraceMessageData method is invoked in your application RDMLX code. Can be True or False.

This Trace Handler object must be installed as part of your application and will act as the system trace handler at run time.

Refer to 7.2 Example Trace Handler to get started.

7.2 Example Trace Handler

This is an example of a self contained Trace Handler object. It checks for the existence of a text file called TracingOn.txt and if found will install itself at run time and a trace file will be created. Any trace statements coded in to the application will be output to this trace file.

Copy and paste the source in to a new RDMLX reusable part. Save and compile.

Function Options(*DIRECT)

Begin_Com Role(*EXTENDS #PRIM_OBJT *implements #prim_app.iTraceI

Define Field(#FilHandle) Type(*dec) Length(3) Decimals(0)

Define_Com Class(#prim_alph) Name(#Tab)

Evtroutine Handling(#Com_Owner.CreateInstance)

#Tab := (09).asChar

#Com_owner.InstallTracing

Endroutine

Mthroutine Name(InstallTracing) Help('Plug in the trace handler to the Applica

- * This example uses a text file as the trigger to implement user tracing.
- * Alternately you might consider using a registry entry

If (#Com_owner.FileExists((*Sys_dir + "TracingOn.txt")))

* Set this object as the system help handler

#Sys_appln.TraceHandler <= #Com_owner

Endif

Endroutine

Mthroutine Name(Initialize) Options(*redefine) Access(*private)

#Com_owner.OpenTraceFile

Endroutine

Mthroutine Name(Terminate) Options(*redefine) Access(*private)

#Com_owner.CloseTraceFile

Endroutine

Mthroutine Name(TraceMessage) Help('Executed whenever #sys_appln.Tracel #Com_owner.WriteToFile(#ComponentName #Description #LineNumber #M Endroutine

Mthroutine Name(OpenTraceFile) Help('Create a new trace outputfile') Access Use Builtin(Stm_File_Open) With_Args(#Com_owner.GetNextFile Append N Endroutine

Mthroutine Name(WriteToFile) Help('Write an entry in the trace output file') A Define_Map For(*Input) Class(#prim_alph) Name(#ComponentName)

Define_Map For(*Input) Class(#prim_alph) Name(#Description)

Define_Map For(*Input) Class(#prim_nmbr) Name(#LineNumber)

Define_Map For(*Input) Class(#prim_alph) Name(#MessageText)

Define_Com Class(#prim_Dat) Name(#Now)

#MessageText := #Now.now.AsLocalizedDateTime.AsString + #Tab + #Comp Use Builtin(Stm_File_Write) With_Args(#FilHandle #MessageText) To_Get(#

Endroutine

Mthroutine Name(CloseTraceFile) Access(*private)

Use Builtin(Stm_File_Close) With_Args(#FilHandle) To_Get(#io\$sts)

Endroutine

Mthroutine Name(TracingState) Options(*redefine)

#MessageTracingActive := True

Endroutine

Mthroutine Name(GetNextFile) Access(*Private)

Define_Map For(*Result) Class(#Prim_alph) Name(#Result)

Define_Com Class(#prim_nmbr) Name(#Extension)

Begin_Loop Using(#Extension)

#Result := *Sys_dir + "UserTrace" + "." + #Extension.asstring.rightAdjust(3 "

Leave If(*Not #Com_owner.FileExists(#Result))

End_Loop

Endroutine

Mthroutine Name(FileExists) Access(*private)

Define_Map For(*input) Class(#prim_alph) Name(#Path)

Define_Map For(*result) Class(#prim_boln) Name(#Result)

Use Builtin(OV_FILE_SERVICE) With_Args(Check_File #Path) To_Get(#io\$

#Result := (#io\$sts = OK)

Endroutine

End_Com

7.3 Use the Trace Handler

After creating a Trace Handler it can be added to your application. This can be done as part of the initialization with a simple Define_Com statement. For example, if the Trace Handler was created as a component called Tracing, add a statement like:

Define_Com Class(#Tracing) name(#TraceHandler)

If you are using the 7.2 Example Trace Handler this is all you need to do to expose the Trace Handler for use in your application. It is a self contained object that will set itself as the system trace handler if a suitable trigger condition is satisfied. In the example this is the existence of file TracingOn.txt in the system source folder.

You are now ready to start adding tracing commands to your application source at places of interest in the RDML code. There are two methods available to do this TraceMessageText and TraceMessageData

TraceMessageText supports four variant input parameters. When used the TraceMessage method in the Trace Handler will be executed once for each of the variables specified.

#Sys_Appln.TraceMessageText(#Empno #Givename #Surname)

TraceMessageData has a source parameter and nine variant parameters. The variant parameters are substituted into the source parameter as if the Substitute intrinsic were being used.

#Sys_Appln.TraceMessageData("Employee &1 Name &2 &3" #Empno #Give

Refer to 7.4 Example Use of Tracing for a sample form incorporating trace commands.

7.4 Example Use of Tracing

This is a form example to demonstrate how to use the tracing methods in your RDML code. It uses a trace handler defined from the sample supplied in 7.2 Example Trace Handler. To trace ensure output is produced create a file TracingOn.txt in the LANSA system folder.

Function Options(*DIRECT)

Begin_Com Role(*EXTENDS #PRIM_FORM) Caption('Doubleclick an item Define_Com Class(#PRIM_LTVW) Name(#List) Componentversion(2) Displated Define_Com Class(#PRIM_LVCL) Name(#LVCL_3) Displayposition(1) Parent Define_Com Class(#PRIM_LVCL) Name(#LVCL_4) Displayposition(2) Parent Define_Com Class(#PRIM_LVCL) Name(#LVCL_5) Displayposition(3) Parent Define_Com Class(#PRIM_RDBN) Name(#TraceMessageData) Buttonchecked Define_Com Class(#PRIM_RDBN) Name(#TraceMessageText) Caption('Use

Define_Com Class(#Tracing) Name(#TraceHandler)

Evtroutine Handling(#com_owner.CreateInstance)

Set Com(#com_owner) Caption(*component_desc)

Clr_List Named(#List)

Select Fields(#List) From_File(Pslmst)

Add_Entry To_List(#List)

Endselect

Endroutine

Evtroutine Handling(#List.DoubleClick)

If (#TraceMessageData.ButtonChecked)

#sys_appln.TraceMessageData("Employee &1 Name &2 &3" #Empno #Giver Else

#sys_appln.TraceMessageText(#Empno #Givename #Surname)

Endif

Endroutine

End_Com

7.5 Trace Output

Trace files will be created in the LANSA system folder as x_UserTrace.999, where 999 will be the next consecutive number available. A new trace file is created every time tracing is started.

The trace file output is formatted as:

Date Time ComponentNameLineNumber [TraceMessageData Result]

7.6 Trace Handler Guidelines and Performance

The number of trace statements to embed in your RDML code is entirely up to you.

Inclusion of too many trace statements may be counterproductive and will produce enormous trace files which are difficult to analyse. At the other extreme, only including sporadically placed trace commands into your RDML code with no real structure may be of little use in identifying issues.

It is recommended to do some analysis of your application to identify critical code and areas of interest, for example, after a record has been read or before a record is updated, or perhaps when invoking a subroutine or method to manipulate data, it may be beneficial to add tracing commands before and/or after the code is executed.

Also consider standardizing the format of your trace output to include the type of the processing to be performed as well as the data values at the time. This can greatly improve the understanding of the output without necessitating great familiarity with the RDML source itself.

#Sys_Appln.TraceMessageText("UPDATE Employee &1 Name &2 &3" #Em]

The trace feature is designed specifically so the trace commands can be embedded in your application and left there without adverse performance impact. If the trigger mechanism to start tracing is not activated, the trace handler will not be installed. Consequently when a call to the trace method is made there is nothing to do. This is equivalent to calling an empty method and has a negligible performance overhead. Simple tests demonstrate that 100,000 trace statements add about a quarter of a second to the run time when tracing is inactive. If tracing is activated, the trace processing added about 45 seconds to the total execution time.

8. Editor Basics

8.1 The Ribbon	8.2 Quick Access	8.4 Context Menus	8.5
	Toolbar		Commands
8.3 Mini-	8.6 Keyboard Shortcuts	8.7 Product	
Toolbars		Information	

8.1 The Ribbon

The ribbon is a command bar that organizes the LANSA commands into a series of tabs a the top of the LANSA Editor window. Some commands and tabs are always there, others appear when you are working with a particular type of object.



To find out what an individual command does, see 8.5 Commands.

Also see 8.1.1 Navigate the Ribbon with Access Keys.

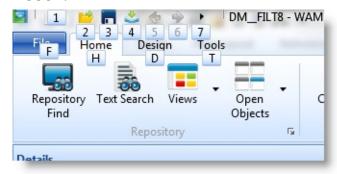
Also See

8.4 Context Menus

↑ 8. Editor Basics

8.1.1 Navigate the Ribbon with Access Keys

When you want to use the keyboard to move around the Ribbon, press the ALT key to enable keyboard control and display the key tips for access keys on the ribbon:



As you can see in the picture above, the access keys for the 8.2 Quick Access Toolbar are numbers. The other access keys are for the different tabs on the ribbon.

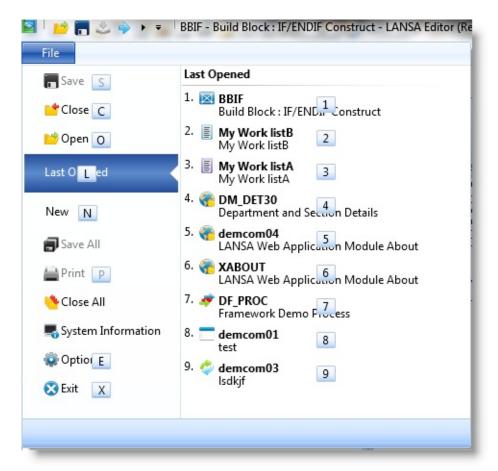
For example to display the access keys for the Home tab press H to display the access keys for the tab:



Now you can use the access key to invoke a command on the tab. For example to compile an object, press M:



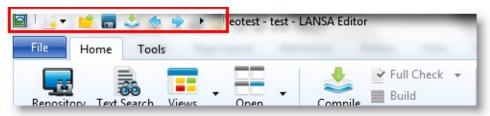
In the File tab you can choose a file from the file list using a number:



You only ever see the key tips for the active tab. To hide the key tips press Esc.

8.2 Quick Access Toolbar

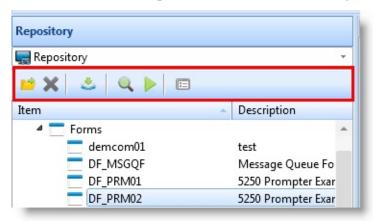
The Quick Access Toolbar provides access to commonly used commands.



Note that the LANSA Version 12 menus are accessible from the Quick Access Toolbar. See Finding Version 12 Commands in Version 13 IDE .

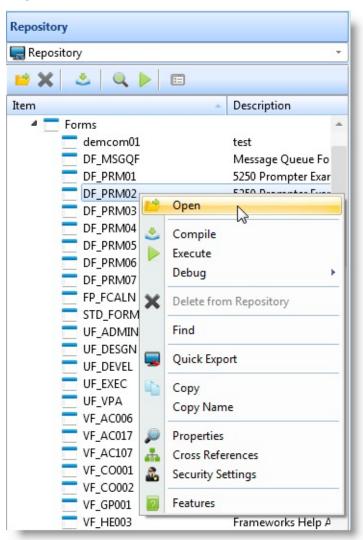
8.3 Mini-Toolbars

Some tabs have a mini-toolbar for quick access to commonly used commands:



8.4 Context Menus

The commands applicable to an object can also be invoked by right-clicking the object and choosing the command from the context menu:



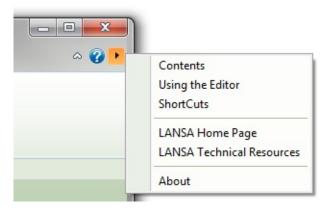
↑ 8. Editor Basics

8.5 Commands

Following the command name, in brackets, is the ribbon on which you will find it.

About LANSA

Displays LANSA Product Information.



ActiveX Attributes

Sets the attributes required to expose this component as an

ActiveX control.

(Design)

Refer to ActiveX Controls in the Developer Guide.

Align...

Aligns a group of selected components in the Design tab.

(Design)

Refer to Align.

Animate (Debug)

In animate mode the debug automatically steps from one command to the next while showing the line currently being

executed.

You can set the delay between commands using the Delay

command.

Refer to Animate Application Execution.

Animate All

(Debug)

This option is otherwise the same as animate, except that it also executes functions or components which have been marked as not debuggable.

You can set the delay between commands using the Delay command.

Refer to Animate Application Execution.

Build

Checks the syntax of your code and generates the code for the

(Home) target platform you have specified.

Clear All Clear all breakpoints set in the open application.

Breakpoints Refer to Set Breakpoints.

(Debug)

Closes Closes current the object.

(File)

Closes all objects.

Comment Shortcut: Ctrl + W

Selection Comment selected lines of code.

Compile Checks the syntax of your code, generates the code and compiles it for the target platform you have specified.

Refer to Compile Tab.

Contents Displays a list of all LANSA documentation so that you can

search on a topic in all the LANSA guides or choose a specific

guide to view.

Dropdown next to Help icon

Continue Continue the execution of the open application.

Execution Refer to Start Debugging.

(F5)

(Debug)

Copy Copies the selection to the Clipboard.

(Home)

Cut Cuts the selection and puts it in the Clipboard.

(Home)

Debug Start the Debug operation. Refer to Debugging Applications for

(Home) more information.

Delay Specify the delay time between the execution of commands.

(Debug) Refer to Animate Application Execution.

Delete Deletes the selection

(Home)

DeployWhen you select this command, the Deployment Tool will start.

(Tools) You can use this tool to create a list of objects, both LANSA

and non-LANSA, to be deployed. Refer to the Deployment Tool

Guide.

Design Shortcut (F10) to open the Design Tab.

(F10)

Design Checks the design of your screen or report.

Check

Design Sets the language of captions in the Design tab.

Language Refer to Development Language in the Administrator Guide.

(Design)

Design ThemeSets the theme for the application being designed. There are other considerations in using themes. To find out more, refer to

(Design) Themes in the Developer's Guide.

Disable All Disable all breakpoints set in the open application.

Breakpoints Refer to Set Breakpoints.

(Debug)

Enable for Enables an object for RDMLX.

RDMLX RDMLX fields and RDMLX files can only be used in functions and components which have been enabled for RDMLX. Also,

language enhancements are only available in RDMLX.

Make sure your object does not contain syntax errors before you

enable it for RDMLX.

Refer to RDML and RDMLX Partition Concepts in the

Administrator's Guide.

End Debug Ends the current debug session.

(Debug) Refer to Start Debugging.

Error Logs Shows the local and web runtime error logs. See Fatal Errors.

(Home)

Execute Runs your code on the platform you specify.

(Home) Refer to Executing Applications.

Exit Closes the editor.

(File)

Find Next Finds the next occurrence of the specified text in the Source tab.

(Home) Refer to Find.

Find Finds the previous occurrence of the specified text in the Source

Previous tab.

(Home) Refer to Find.

Find... Finds the specified text in the Source tab.

(Home) Refer to Find.

Full Check Checks the syntax of your code.

(Home)

Halt Halts the program execution and terminates the debug session.

Execution Refer to Start Debugging.

(Debug)

History Shows a list of components which have been executed. You can

(Home) use this list to quickly execute components.

Import Import translations to the current partition.

Translations

(Tools)

Integrator Launches Integrator Studio. **Studio**

(Tools)

Interactive Verification

Sets interactive verification to On (the Default), Off and Verify Now.

(Settings) When objects are being edited in the LANSA Editor, a

verification process is run to provide feedback to you about the state of the object being edited. For source objects, such as functions and components, verification occurs as the source changes, while for non-source objects such as fields and files, verifications are run every 1.5 seconds. This verification process produces the error/warning triangles and messages seen throughout the Editor.

For large objects such as files with many references to other file, for example, lookup rules, it may be desirable to turn off the verification until the object is saved, rather than continually check that the details entered by the user are correct. This check can be a performance overhead, so turning off interactive verification allows you to make many changes and then either verify as part of the save of the file. You can verify on demand using Ctrl+F9.

LANSA Client

Launches LANSA's end user reporting tool, LANSA Client, if it is installed.

(Tools)

Import (Tools)

Use this option to import data into the repository. To do this you must have first used LANSA export function which creates files that can then be imported into Visual LANSA. They can be directly imported from a PC "local directory" using shared folder support or from a directory on a drive. If a shared folder

is used, PC Support must be started.

Please note that before you start to import data into the repository, your repository must be already configured adequately. Refer to LANSA Import in the *Administrator Guide*.

LANSA Use the dro **Home Page** home page.

Use the dropdown list nex to the help icon to open the LANSA home page

Layout Manager Shows the Layout Manager dialog.

Refer to Layout Managers in the Developer Guide.

Load Other File

From New file drop down list, select File, then Load Other File. Loads file definitions which are maintained outside of LANSA. Load Other Files in the Technical Reference Guide.

Lock Design Toggles the locking of the design.

(Design) Refer to Lock Design.

Logical Modeler

Use the Logical Modeler to design and manipulate a complete logical data model using elements, entities and relationships and then build a physical database from this model. Refer to the

Logical Modeler Guide.

Message

(Tools)

Shows the Message Files dialog.

Files... Refer to Message File Maintenance in the Administrator Guide.

(Tools) Refer to User Message Files for information about using User

Message Files in the Multilingual Application Design Guide.

New

Creates a new object.

(File)

Next Object Shows the next open object.

(Quick access toolbar)

Object Enables or disables Object Locking.

Locking Refer to Object Locking in the *Administrator Guide*.

(Settings) When object locking is on, locks are put on objects so that

multiple users cannot edit them simultaneously.

In single user environments, object locking can be turned off. Setting object locking to Read only sets all objects in the

partition to READ ONLY status.

Open Displays the Open dialog for you to enter the name of an object. (File) This option allows you to directly open an object provided you

know the name of it. Refer to Open an Object for more details.

Open Switch between the objects open in the editor. **Objects**

(Home)

Paste Inserts the Clipboard contents at the insertion point.

(Home)

Previous Shows the previous open object. **Object**

(Quick access toolbar)

Print Prints the source code of the object being edited.

(File) Refer to Printing in Visual LANSA.

Product Displays the LANSA Product Information dialog.

Information Refer to Product Information.

(Help dropdown list)

Publish Publishes all weblets.

(Tools)

Quick Quickly export component definitions.

ExportRefer to Quick Export in the Administrator Guide.

(Tools)

Redo Reinstates the change which was cancelled using Undo.

(Home)

Refresh
Master
Object List
(Home)

Retrieve information about objects in the master repository. The Master Repository State column in the Repository and Favorites tabs indicates whether an object exists in the master repository.

Refer to Refresh Master Object List in the Administrator Guide.

Replace... Replaces selected text with the specified text in the Source tab

(Home) Refer to Replace.

Repository Locates objects in the repository.

Find... Refer to Repository Find.

(Home)

Save Saves the definition of the current object.

(File)

Save All Saves the definition of all open objects.

(File)

Saved Shows the Saved Definitions dialog.

Definitions

Options Opens the LANSA Settings to modify the Editor settings.

(File)

Shortcuts Lists the keyboard shortcuts that can be used in the editor.

(Help Refer to Keyboard Shortcuts.

dropdown list)

Size... Changes the size of a group of selected components in the

(Design) Design tab.

Refer to Size.

Source Moves focus to the **Source Tab**.

Shortcut

(F9)

Source

Provides a code search within source code with the Audit Audit Stamps, if selected in Source Settings in the LANSA Settings.

Find...

(Home)

Start Host Starts the Host Monitor for Repository Synchronization.

Monitor When checking objects in or out of the master repository, it is

(Home) not necessary to manually start the Host Monitor as it is started

automatically in the background.

Refer to Start and Stop the Host Monitor in the Administrator

Guide.

Step Into Executes the current statement and breaks at the next line.

(Debug) You can use Step Into to execute code one statement at a time.

After stepping through each statement, you can see its effects by

looking at the Variable window. Refer to Run Parts of Application.

Step Over Execute the procedure called by the current line and break at the

(Debug) line following the current line.

> Step Over is identical to Step Into except when the current statement contains a call to a procedure, Step Over executes this procedure as a unit and then steps to the next statement in the

current procedure.

Refer to Run Parts of Application.

Styles Select the Visual Style to be used for the current component.

(Design)

System Contains the settings that control the processing of items related **Information** to the master, if running a slave system. For further information,

(File) refer to System Definitions in the Administrator Guide.

Text Search Opens the Text Search dialog to search your Source code as

(Home) described in Text Search.

Translate Translate LANSA object details.

(Tools) For further information, refer to Translate Objects in the

Application Deployment Tool guide.

Uncomment Shortcut: Ctrl + Shift + W

Selection Uncomment selected lines of code

Undo Ctrl + Z

(Home)

Upgrade Cancels previous change.

(Tools)

Use Upgrade repository definitions.

Template
(Design)

The upgrade tools include a facility to convert files to use an identity column to store the relative record number in the

Wizard... database.

Using the Shows a list of templates you can use to create applications. **Editor**

(Help dropdown

list)

Views Displays the topic Editor Tabs in the User Guide.

(Home)

Error Logs Use the Views command to display the various tabs in the

editor.

Web Use the Visual LANSA Framework to quickly create component-based LANSA applications for the Windows environment. Refer to Visual LANSA Framework Guide.

(Home)

Wizards To locate fatal errors in your code, select this command, to open

(Tools) the Web Runtime Error Log Viewer.

↑ 8.4 Context Menus

8.6 Keyboard Shortcuts

For most features, you can use Visual LANSA with or without a mouse. The keyboard can also be used to select items and actions within a window. The Screen and Report Designing facilities in Visual LANSA do need a mouse.

- 8.6.1 Keyboard
- 8.6.3 Move the cursor in the Source tab
- 8.6.4 Position the current line
- 8.6.5 Tag and comment lines
- 8.6.6 Collapse and expand code
- 8.6.7 Format text
- 8.6.8 Deletions
- 8.6.9 Copy, cut and paste
- 8.6.10 Help
- 8.6.11 Selection
- 8.6.12 Undo/Redo
- 8.6.13 Find and replace
- 8.6.14 Command Assistant Keystrokes
- 8.6.15 Menu editor
- 8.6.16 Select with the mouse
- ↑ 8. Editor Basics

8.6.1 Keyboard

Selecting with the Keyboard

To set the focus to a control item (check box, push button, entry field, etc.) within a dialog box, use the Tab key on the keyboard. The Tab key will select the next control item or input field within the dialog box. Within a group of control items, such as check boxes or push buttons, the Arrow keys are used to move from one control item to another. Press the spacebar to select or deselect an item.

The control item in focus will have a small dotted box around it. If the control item is an entry field, an I-beam pointer will appear.

When you press Enter, the selected push button will be accepted.

↑ 8.6 Keyboard Shortcuts

8.6.2 Close Tabs

To close the tab which has the focus use Control + F4.

8.6.3 Move the cursor in the Source tab

Action Key to use

Move to start of text on current line Home

Move to top of window Home Home

Move to start of text in current file Home Home Home

Move to start of text in current file Control + Home

Move to end of text on current line End

Move to bottom of window End End

Move to end of text in current file End End End

Move to end of text in current file Control + End

Move up one page PageUp

Move down one page PageDown

Move to next part of statement Ctrl + RightArrow

Move to previous part of statement Ctrl + LeftArrow

8.6.4 Position the current line

These keys position the current line in the window.

Action	Key to use
Position the current line on the top of the window	Ctrl + T
Position the current line in the middle of the window	Ctrl + M
Position the current line on the bottom of the window	Ctrl + B

8.6.5 Tag and comment lines

Action	Key to use
Toggle a tag on a line	Ctrl + F2
Remove all tags	Alt + F2
Jump to the previous tagged line	Ctrl + Up Arrow
Jump to the next tagged line	Ctrl + Down Arrow
Comment selected text	Ctrl + W
Uncomment selected text	Ctrl + Shift + W

8.6.6 Collapse and expand code

Action	Key to use
Collapse blocks. This command blocks a level at a time. If you keep collapsing the blocks, the entire code will be collapsed, except for lines that are not in blocks.	Ctrl + F9
Expand blocks. This command expands a level at the time	Ctrl +F10
Collapse the block containing the line that has the editor focus	Ctrl + Shift + F9
Expand the collapsed block that has the editor focus	Ctrl + Shift + F10

8.6.7 Format text

Action Key to use

Makes selection/next character upper case Alt + UpArrow

Makes selection/next character lower case Alt + DownArrow

8.6.8 Deletions

Action Key to use

Delete word right Ctrl + Delete

Delete word left Ctrl + BackSpace

Delete from current position to end of line Alt + Shift + L

8.6.9 Copy, cut and paste

Action	Key to use
Cut line and place it on clipboard	Ctrl + L
Copy line and place it on clipboard	Alt + Q
Copy selection or line	Plus key in number pad
Copy selected text to the clipboard	Ctrl + Insert
Paste text from the clipboard	Shift + Insert
Copy selected text to the clipboard	Ctrl + C
Cut selected text to the clipboard	Ctrl + X
Paste text from the clipboard	Ctrl + V

8.6.10 Help

The Help Key is used to display help text.

Action	Key to use
Displays the help topic relating to the context of where you are in the LANSA Editor.	F1

8.6.11 Selection

Select an object directly

Ctrl + O to open this dialog where you can enter the name of the object. (There is no wildcard selection.)



Action	Key to use
Extend/reduce selection to next part of statement	Ctrl + Shift + RightArrow
Extend/reduce selection to start of previous part of statement	Ctrl + Shift + LeftArrow
Block select to end of line	Shift + End
Block select to start of line	Shift + Home
Block select to start of file	Ctrl + Shift + Home
Block select to end of file	Ctrl + Shift + End
Block select to page down	Shift + PageDown
Block select to page up	Shift + PageUp
Block select part of statement	Left Button double- click
Line selection	Shift + UpArrow
Line selection	Shift + DownArrow
Select all text	Ctrl + A

8.6.12 Undo/Redo

Action Key to use

Undo operation Ctrl + Z

Redo operation Ctrl + Y

8.6.13 Find and replace

Action Key to use

Find text Ctrl + F

Replace text Ctrl + H

Find next F3

Find previous Shift + F3

Go to line Ctrl + G

8.6.14 Command Assistant Keystrokes

You can use the mouse or these keystrokes in the Command Assistant which can be opened from the Source tab.

Action	Key to use
Show Command Assistant	F4
Apply any command or value	Enter
Select command, value etc.	Up and down arrow keys
Expand or collapse a node in the Command Assistant	To expand command parameters: Ctrl + Shift + =
	To collapse the parameters: Ctrl + - (minus sign).
	Everywhere else: Left and right arrow keys.
Move inside the Command Assistant	Tab, and Shift + Tab
Select multiple entries in the Command Assistant	Shift + ArrowDown/Up

You can also use the mouse:

Action	Key to use
Select more than one value	Ctrl + mouse click
Select a value from the Command View	mouse click

8.6.15 Menu editor

Action Key to use

Create new menu title or submenu Tab

Add menu item Enter

Delete menu item Ctrl + Delete

Delete submenu Ctrl + Shift + Delete

Move in the menu Arrow keys

Note that a submenu can only be dropped onto an open menu. Duplicated menu items or shortcuts are indicated by small red triangles.

8.6.16 Select with the mouse

Action Key to use

Mouse block select Left Button Down

Mouse column select Alt + Left Button Down

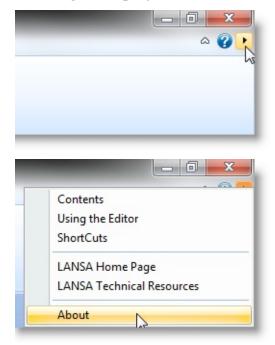
Mouse line select Ctrl + Left Button Down

Or

Left Click in the Line Number Area

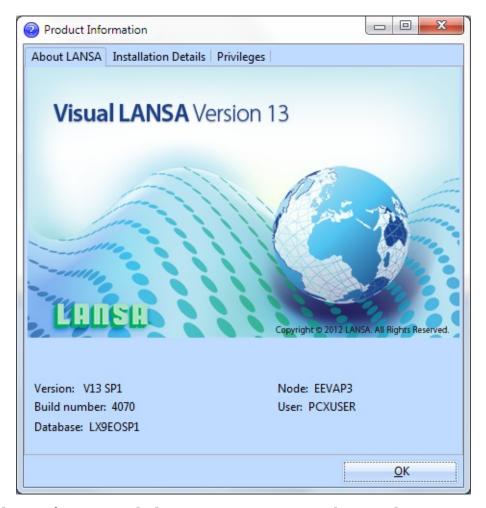
8.7 Product Information

The Product Information dialog is displayed from the Help menu:



This dialog includes the following tabs:

- 8.7.1 About LANSA
- 8.7.2 Installation Details
- 8.7.3 Privileges

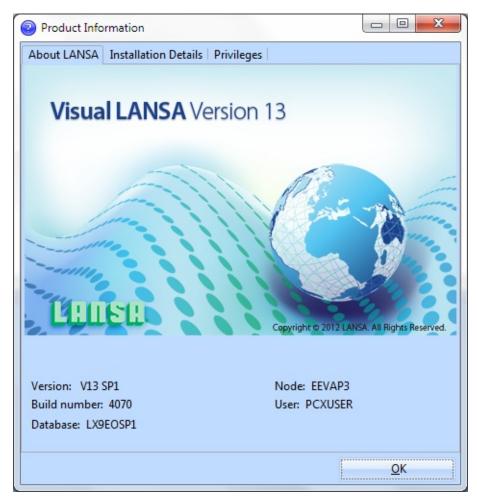


The Product Information dialog is very important when applying program updates and when you are analyzing problems in Visual LANSA. You will require this information when reporting issues to LANSA Support.

↑ 8. Editor Basics

8.7.1 About LANSA

The About LANSA tab presents summary information about the LANSA software version as well as the current user and database information.



The equivalent System i information is referred to as LANSA About.

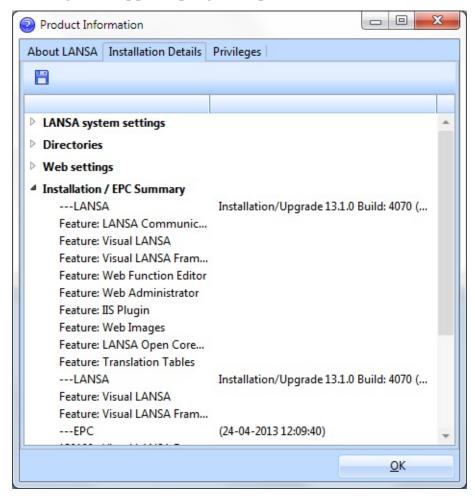
Also See

Logon Parameters in the Administrator Guide

↑ 8.7 Product Information

8.7.2 Installation Details

The Installation Details tab uses a tree view to display detailed information about the installed system. You can review this information to verify your installation settings and applied program updates.



By pressing the Save toolbar button, all installation details will be saved to the LANSA_About.txt file in the LANSA root directory. This information is typically used when contacting LANSA support.

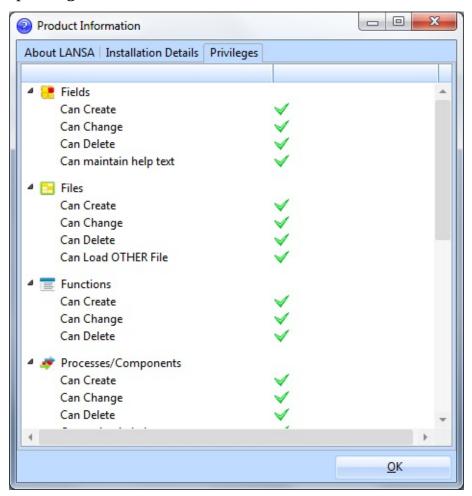
Also See

Applying EPCs in the Administrator Guide.

↑ 8.7 Product Information

8.7.3 Privileges

The Privileges tab shows the privileges of the user currently being logged on. To change the privileges, see Edit User Definitions.



↑ 8.7 Product Information