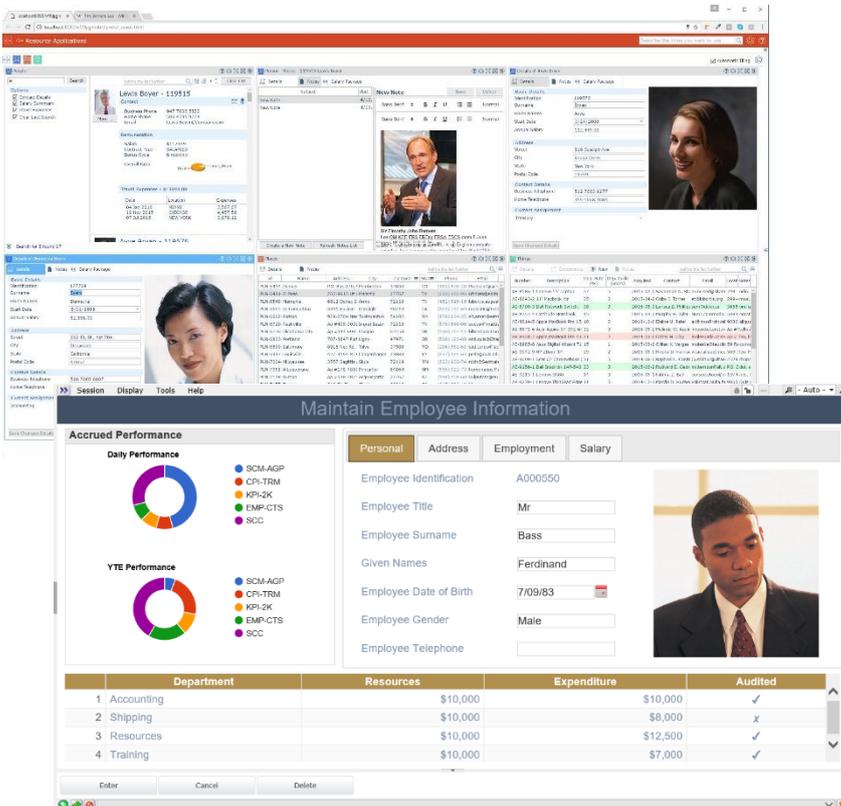




LANSA Composer 6.1, aXes 4.1.0 and new VLF version

Just in case you haven't heard: recently, we released new versions of our LANSAs products Composer, aXes and VLF (Visual LANSAs Framework). This Newsletter provides an insight into the new features and enhancements in these three products which allow you generate amazing web based applications as the images below highlight (VLF and aXes shown below).



INSIDE THIS ISSUE

- LANSA Composer 6.1, aXes 4.1.0 and new VLF version 1
- LANSA Composer 6.1..... 2
- aXes 4.1.0 13
- New VLF version 24
- Coming soon in VLF 32
- QFRCOBJCVN system value can cause V14 SP1 LANSAs for i upgrade failure 36
- Performance slowdown in LANSAs applications caused by Mutex wait issue..... 38
- Latest News 39
- LANSA EPC..... 39
- Web Server 39
- IBM 39



LANSA Composer 6.1

What's New in LANSAs Composer Version 6.1

1. Installation and Upgrade
2. Processing Sequences and Logs
3. Transformation Maps
4. Deployment
5. New and Enhanced Activities
6. Other New and Enhanced Features

1. Installation and Upgrade

Upgrading LANSAs Composer Server

If you are upgrading a prior version of LANSAs Composer server (on IBM i or Windows), this version supports upgrading LANSAs Composer from both versions 5.0 and 6.0. Upgrading directly from versions earlier than 5.0 to version 6.1 is, however, not supported. You must upgrade first to version 5.0 and then to version 6.1.

LANSAs Composer Client

Altova MapForce version 2017SP1 is provided on the media. It is strongly recommended that MapForce is upgraded to this level on all LANSAs Composer clients.

LANSAs Composer Server on IBM i

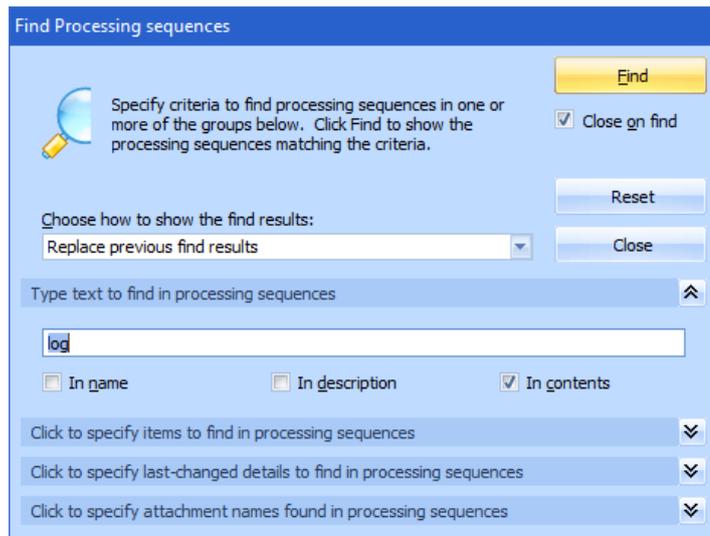
The IBM i installation now completes the imports of supplied activities and other definitions as a part of the install job. Previously it was necessary to install and connect a LANSAs Composer client to run the *Server Initialisation Wizard* in order to complete this part of the installation. The Server Initialisation Wizard is no longer shown in normal circumstances.

2. Processing Sequences and Logs

The following significant enhancements apply to working with Processing Sequences and Processing Sequence Logs:

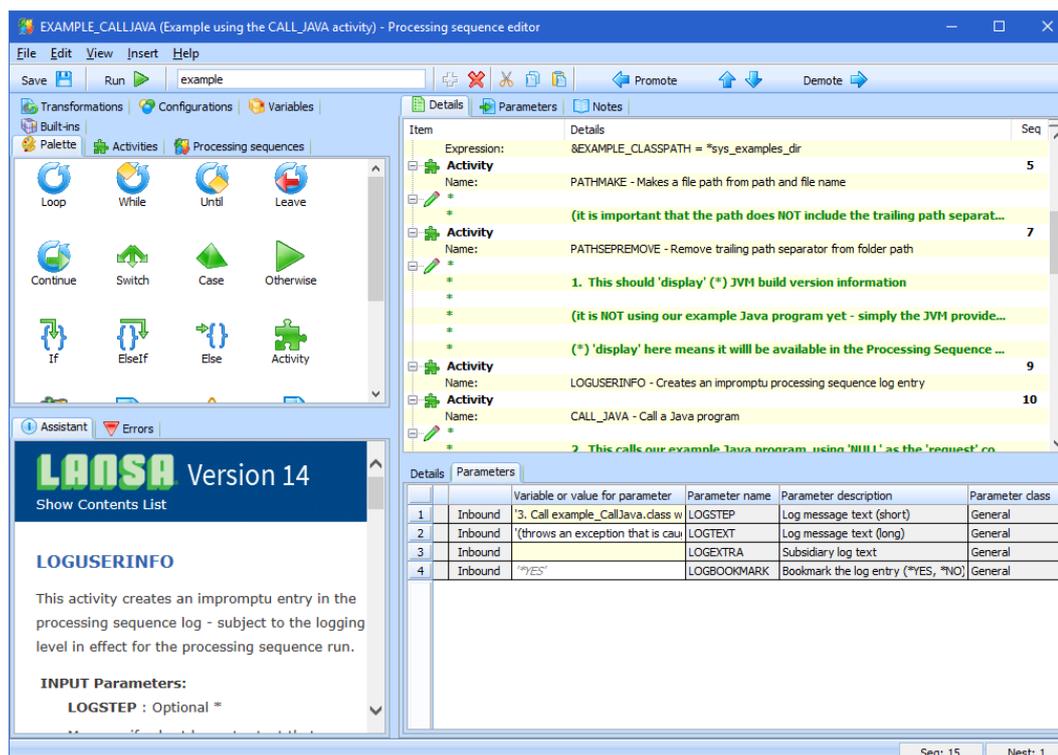
Search Contents of Processing Sequences

You can now easily find all Processing Sequences containing a specified search term. When you use the new *In contents* find option in the *Find Processing Sequences* window, LANSA Composer will search all parts of each qualifying processing sequence definition, including variable names, activity and map parameter names and values, processing sequence parameter names and descriptions, expressions used in ASSIGN, IF, CASE and other directives, as well as literal values and comments.



Find in the Processing Sequence Editor

The *Processing Sequence Editor* now provides a *Find* capability to assist in finding instances of a specified search term in your Processing Sequence as you edit it.



Bookmarking Processing Sequence Log Entries

LANSA Composer's Processing Sequence Logs are enhanced with the capability to store bookmarks against the log entries.

The LANSA Composer clients Processing Sequence Log viewer is enhanced with the ability to show the bookmarks and to search forwards and backwards for them.

Users can record bookmarks for their solutions using the related enhancement to the LOGUSERINFO activity, and/or the new LOGBOOKMARK activity.



Otherwise, and so as not to dilute the efficacy of this new capability, only a very limited number of functions in LANSA Composer will automatically record bookmarks. They are:

- Activity TXDOC_REGISTER
- Activity WAIT_FILESREADY

Other Processing Sequences and Logs Features

Other enhancements applying to working with Processing Sequences and Processing Sequence Logs include:

- In the *Processing Sequence Editor*, when a variable or built-in variable is dragged from the Variables or Built-ins tabs onto the Processing Sequence, LANSA Composer will now add an instance of the LOGVARIABLE activity to log the variable's (or built-in variable's) value.
- Further Processing Sequence run-time performance improvements have been made, especially affecting Processing Sequences that make heavy use of variable lists.
- Apparently "active" processing sequences that have not logged any activity for six hours are now flagged as 'Abended?' (likelyabend) in the Process Monitor, Management Console and Run History displays. The "Processing Sequence Log" display window now offers the option to confirm the abnormal end using a new "Mark as ended abnormally" command on the File menu.
- The Processing Sequence Log Viewer now shows the sequence numbers(s) of the currently selected log entry in the status bar. In addition, it provides a new Choose columns command on the View menu to let you choose and reorder the columns that are shown.
- The version stamp of the Processing Sequence version used for a Processing Sequence run is now included in the "Additional information" attached to the DXS0006 (Processing sequence started) log entry.
- When applicable, the request server log file contents are now retrieved and written to the processing sequence log. This applies on Windows servers only, and only for SYNCHRONOUS requests processed by the request server on the LOCAL system.
- The URL used to display a Processing Sequence Log directly (using the Management Console) is now sensitive to either or both of optional keywords EXPANDALL and/or SHOWDETAILS in the URL used to display the log (directly). Caution should be exercised using this feature in connection with large Processing Sequence Logs. For more information, refer to Direct URL Links to Processing Sequence Logs.

3. Transformation Maps

There are several enhancements to the Transformation Map support in this version of LANSА Composer:

Transformation Map Execution Through Altova MapForce Server (Windows only)

This version of LANSА Composer on Windows server provides new support for preparing maps for and executing Transformation Maps through Altova MapForce Server. (This support is not presently available for LANSА Composer on IBM i servers.)

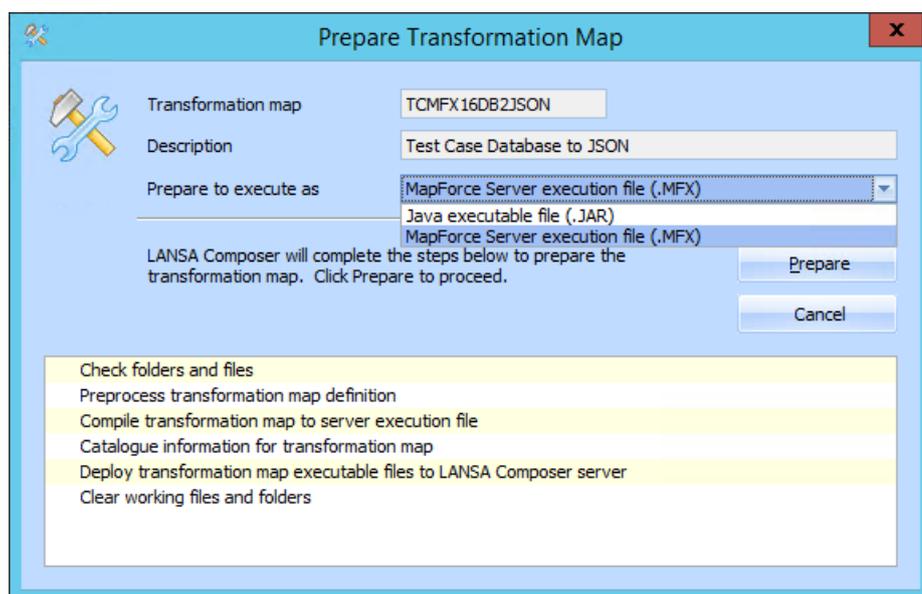
This enhancement permits use of certain MapForce functionality (such as JSON mapping and REST function calls) that could not previously be used in Transformation Maps in LANSА Composer.

This support is subject to an optional licence feature, and, additionally requires a licensed installation of Altova MapForce Server on the same server system as LANSА Composer Windows Server. Enquire with your LANSА representative for more information.

Note that there are some changed design considerations for maps prepared this way.

Most notably:

- Maps designed for use this way should have the "Built-in" engine set as the selected language, to ensure feature compatibility with the intended execution environment. (LANSА Composer will automatically make this selection for new maps where this mode is selected as the default in System Settings.)
- File based map components (such as for XML, EDI and text files) do not automatically result in an input parameter by which a dynamic file path may be provided. Instead, the map designer must specifically identify such components in MapForce as using dynamic file names and attaching an input component to them.
- Database configurations are not used in connection with database components in your maps. Instead we recommend using MapForce's global resources support to make your maps independent of the execution environment (for example, design vs production environments). LANSА Composer provides support for specifying a global resource definition file and configuration name in System Settings that will be used at run-time for this purpose.

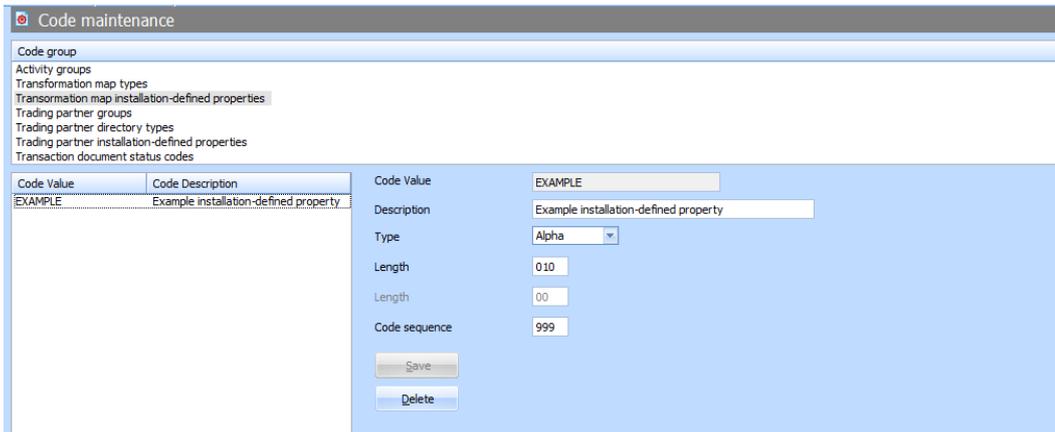


Installation-Defined Properties for Transformation Maps

LANSА Composer has long supported installation-defined properties and property values for Trading Partners. Now similar support is provided for Transformation Maps.

Installation-specific properties for Transformation Maps can be defined in Code Maintenance. If your installation has defined custom properties for Transformation Maps, then you can enter values for those properties for each Transformation Map.

You can access the property values for a transformation map in a processing sequence by using the Transformation Map (*TRANSFORM) Built-in Variable Qualifiers.



The screenshot shows the 'Code maintenance' window. On the left, a tree view lists various code groups, with 'Transformation map installation-defined properties' selected. Below this is a table with two columns: 'Code Value' and 'Code Description'. The table contains one entry: 'EXAMPLE' with the description 'Example installation-defined property'. To the right of the table is a form for editing the selected property. The form fields are: 'Code Value' (EXAMPLE), 'Description' (Example installation-defined property), 'Type' (Alpha), 'Length' (010), 'Length' (00), and 'Code sequence' (999). At the bottom of the form are 'Save' and 'Delete' buttons.

Code Value	Code Description
EXAMPLE	Example installation-defined property

Code Value: EXAMPLE
Description: Example installation-defined property
Type: Alpha
Length: 010
Length: 00
Code sequence: 999
Buttons: Save, Delete

4. Deployment

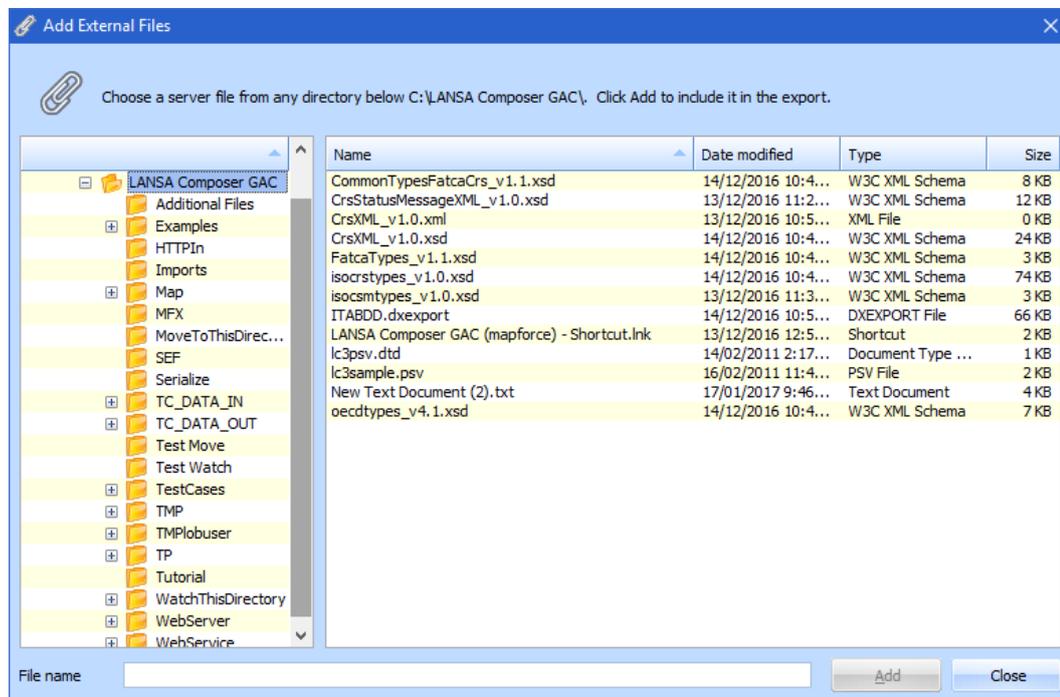
Improvements to deployment support in this version include:

Include External Files

Many LANSa Composer solutions rely on files that are external to LANSa Composer's database – such as XML schema files, body text files for SMTP Mail Details configurations, skeleton text files for use with the TEXT_SUBSTITUTE activity and many other file types.

You can now include such files in your export providing they are held in the *Home path relative to server* specified in System Settings or a "child" directory of that location.

When you include such files, LANSa Composer stores the *relative* path in the export file and, on import, it reconstructs the file path relative to the corresponding system setting in the target system. In addition, the import will, if necessary, create the sub-directory paths on the target system on import.



Unattended Export and Import Operations on the LANSА Composer Server

In prior versions, all export and import operations had to be initiated through the LANSА Composer client software. In this version, you can now execute export and import operations in an unattended mode on your LANSА Composer Server.

To do so:

- On an IBM i server, you use the COMPOSER command, which has been extended with new options and parameters to support this.

```
Type command, press Enter.
==> LC6PGMLIB/COMPOSER REQUEST(EXPORT) EXPORTLIST(MY_EXPORT_LIST)
      EXPORTTO('/LANSA_Composer_lc6pgmlib/lic/MY_EXPORT001.dxexport')
```

```
Type command, press Enter.
==> LC6PGMLIB/COMPOSER request(IMPORT)
      importfrom( '/LANSA_Composer_lic6pgmlib/lic/MY_EXPORT001.dxexport' )
```

- On a Windows server, you use the dxstart.exe program, which has been extended with new options and switches to support this.

```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.
C:\Windows\system32>cd C:\Program Files (x86)\LANSA Composer Server\X_Win95\X_Lansa\Execute
C:\Program Files (x86)\LANSA Composer Server\X_Win95\X_Lansa\Execute>dxstart.exe /request=dxexport /exportlist=MY_EXPORT_LIST /exportto="c:\temp\MY_EXPORT.dxexport"
```

```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.
C:\Windows\system32>cd C:\Program Files (x86)\LANSA Composer Server\X_Win95\X_Lansa\Execute
C:\Program Files (x86)\LANSA Composer Server\X_Win95\X_Lansa\Execute>dxstart.exe /request=dximport /importfrom="c:\temp\MY_EXPORT.dxexport"
```

To perform an unattended export on your LANSА Composer Server you must use an Export List that you have already defined through the LANSА Composer client for the purpose.

5. New and Enhanced Activities

The following supplied activities are new or revised for this version of LANSА Composer:

- The CALL_FUNCTION and COMPOSER_RUN activities provide a new LOGRQSSERVER parameter that enables further diagnostic logging for requests processed through the LANSА Composer request server, on Windows servers only.
- The FIND_TPMAP activity will now report a warning in the Processing Sequence log if an otherwise eligible linked map is not returned because its status is Inactive.
- The FOR_EACH_CSVROW iterator activity is enhanced with new input parameters MAXROWS, PARSEOPTION and SKIPFIRSTROW and a new output parameter, CSVROW. Additionally, *TAB may now be specified for the SEPARATOR parameter. The new features provide much more flexibility to most-appropriately handle a wide-variety of CSV formatted files with the best possible performance.

Because the new parameters and default values might otherwise cause a change of behaviour for existing solutions, the existing FOR_EACH_CSVROW activity has been deprecated (as ZZFOR_EACH_CSVROW). This means existing solutions will continue to use the deprecated version of the activity until/unless they are explicitly changed.

- The FOR_EACH_TOKEN iterator activity will find tokens within a string. Tokens can be individual values using a space as a separator or can be multiple values within quotation marks.
- A new GET_MAP_PARAMETERS activity retrieves details of the parameters for a specified transformation map.
- The LOGVARIABLE activity will now list all the list values if the specified variable is a variable list. Previously you had to use the LOGLIST activity to log variable list values. With this enhancement, you can simply use LOGVARIABLE for all cases.
- A new LOGBOOKMARK activity specifies that an adjacent log entry should be bookmarked, so that it can more easily be located when displaying the Processing Sequence Log.
- A new LOGCOMPLETION activity creates a processing sequence completion log entry. It is very similar to the LOGUSERINFO activity, but offers the additional feature that the log entry created by this activity will be visible in Processing Sequence run history lists displayed in LANSА Composer, including the Process Monitor and the Process Monitor page of the Management Console.

You may use LOGCOMPLETION more than once in your Processing Sequence, and all the messages will be available in the Processing Sequence Log, but only the last logged completion message will be shown in the run history lists.

- A new, optional LOGSTEP parameter for the LOGUSERINFO activity allows the solution designer to specify alternate "short" text for the log entry instead of the "Impromptu log message" that is presently always used. In addition, the activity now has a new LOGBOOKMARK parameter that can be used to specify that the resulting log entry should be bookmarked, so that it can more easily be located when displaying the Processing Sequence Log. The default is *YES, meaning that existing solutions using LOGUSERINFO will generate bookmarks after applying the upgrade.

- Still on the LOGUSERINFO activity, LANSACOMPOSER will now apply variable substitution to the strings provided in the LOGSTEP and LOGTEXT parameters, using the same rules/syntax as apply to the SUBSTITUTE_VAR activity. For example, if you specify the string "&MYCOUNT files processed from &MYFOLDER(&MYINDEX) at *now_local", the variable and built-in variable references &MYCOUNT, &MYFOLDER(&MYINDEX) and *now_local will be replaced by the values of the variables that they represent.
- A new LOGBOOKMARK parameter can be used with the TXDOC_REGISTER activity to specify that the log entry (created when a new transaction document is successfully registered by this activity) should be bookmarked, so that it can more easily be located when displaying the Processing Sequence Log. The default is *YES, meaning that existing solutions using TXDOC_REGISTER will generate bookmarks after applying the upgrade.
- The NEXTNUMBER activity is enhanced with new RESTART and RESTARTVALUE parameters that may be used to conditionally or unconditionally trigger numbering for a specified number series to restart at 1 (one). The defaults for the new parameters mean that the behaviour for existing solutions will not change.
- In version 6.0, processing sequence variables possible size was increased to 1024 (*) from the previous 256 byte limit. However, not all supplied activities had immediately been updated to receive and/or return the longer values. The complete set of SQL database activities have now been updated to take advantage of this. This offers benefits including:
 - it is easier to construct and use complex SQL statements
 - SQL statement parameter values longer than 256 bytes may be used
 - returned database column values longer than 256 bytes may be received

(*) Due to the necessity, when used in certain contexts to surround alphanumeric values with quotes (and to escape embedded quotes), the maximum length in practice will be 1022 or less.
- A new TEXTFILE_SPLIT activity can split a text document file such as fixed-length fields (FLF) or comma-separated values (CSV) into multiple files according to a specified delimiter value.
- A new TEXTFILES_COMBINE activity can combine one or more source text document files into a single aggregate target document file.
- A new DOCTRCOUNT parameter on the TXDOC_REGOUTX12 and TXDOC_REGOUTEDI activities allows you to specify the number of transaction sets (messages) that will be contained in the outbound transaction document. The activity will create as many DXX4MS records as specified, facilitating later outbound mapping of the transaction document. For each DXX4MS record, the activity will allocate the necessary transaction set control number.
- When the WAIT_FILESREADY activity is used with operations *PUTGET or *GET and a file is successfully retrieved from the cache, the new LOGBOOKMARK parameter can be used to specify that the log entry should be bookmarked, so that such entries can more easily be located when displaying the Processing Sequence Log. The default is *YES, meaning that existing solutions using WAIT_FILESREADY will benefit from the enhanced behaviour after installing the upgrade.
- A new WRITE_FILE activity writes text to a file. Variables substitution may be used in the text strings, permitting you to easily mix text with Processing Sequence variables values – or to create a line of comma-separated values to write. Refer to the supplied EXAMPLE_WRITEFILE01 Processing Sequence for an example of using the WRITE_FILE activity.

- By default, the XML_QUERY activity is optimized to avoid reloading the XML document if it is the same as the preceding usage in the same Processing Sequence. If, however, your solution re-uses the same XML document path and name but rewrites the file contents, then you may need to force the XML_QUERY activity to reload the XML document. You may do so by specifying the new *RELOAD option in the XMLOPTIONS parameter on the first use of XML_QUERY after re-writing the XML document content.

6. Other New and Enhanced Features

Amongst other revisions in this version of LANSА Composer are:

- A range of new date built-in variables are available for use in Processing Sequences. They are *NOW_LOCAL_DATE, *NOW_LOCAL_YYYY, *NOW_LOCAL_YYYYMM, *NOW_LOCAL_MM, *NOW_LOCAL_WEEK and *NOW_LOCAL_DOW. The new built-in variables were provided in connection with the new restart numbering enhancements made to the activity, but they may, of course, be used in whatever way your BPI solutions demand.
- When using the Run operation of the provided SOAP web service to run a Processing Sequence, you may now omit the parameter names – in that case, LANSА Composer will match the values provided to the parameter names that are defined for the specified processing sequence according to the sequence in which you provide them.
- Deprecated activities are available to display and select on the *All Activities* tab of the Activities instance list, by means of a new *Deprecated* node shown after the alphabetical groups. This is provided so that customers may use cross-reference information for these activities to see where they are used and possibly replace them.

Note that it is not generally necessary to replace usage of deprecated activities, but some customers prefer to do so as a good housekeeping measure. In some cases, behaviour of replacement activities may have changed (which is often the reason that an original activity may have been deprecated) and you should be prepared to retest your solutions if you replace the deprecated activities.

- When a LANSА System Configuration defines a connection to a local LANSА system on a Windows server, LANSА Composer now provides a folder browser for the LANSА system path entry field. In addition, the input field 'Database name' has been relabelled as 'Datasource name' to more clearly identify its meaning and documentation has been clarified.
- System Settings have been reorganized to achieve greater clarity around the *File locations*. All file locations are now organized into two sections: *Server file locations* and *File locations relative to client*. Some minor reorganization of other system settings was also done in connection with these changes. A Test button is provided that can run a simple test to verify that the *home path* and *home path relative to server* refer to the same location on the LANSА Composer server's file system.
- System Settings relating to Transformation Maps have been reorganized and extended in connection with the new support for Transformation Map Execution Through Altova Mapforce Server (Windows only).

- Additional columns are available in the *Choose Columns* dialogue for the instance lists in the *LANSA Composer Document Manager*, comprising the two available user value fields and the six staging file “key” fields.
- New example processing sequence EXAMPLE_CALLJAVA illustrates use of the CALL_JAVA activity to call a Java program.
- A new Troubleshooting the LANSA Composer Request Server section has been added to the LANSA Composer Guide.
- LANSA Composer help and documentation is now accessed from the LANSA web by default. This helps to ensure that customers always have access to the latest version of the documentation for the software version they are using. The documentation is still distributed and installed locally and customers may choose to use the local copy, if they need or prefer to, using the *LANSA Composer Options* dialogue.



aXes 4.1.0

1. New Features

AutoGUI+

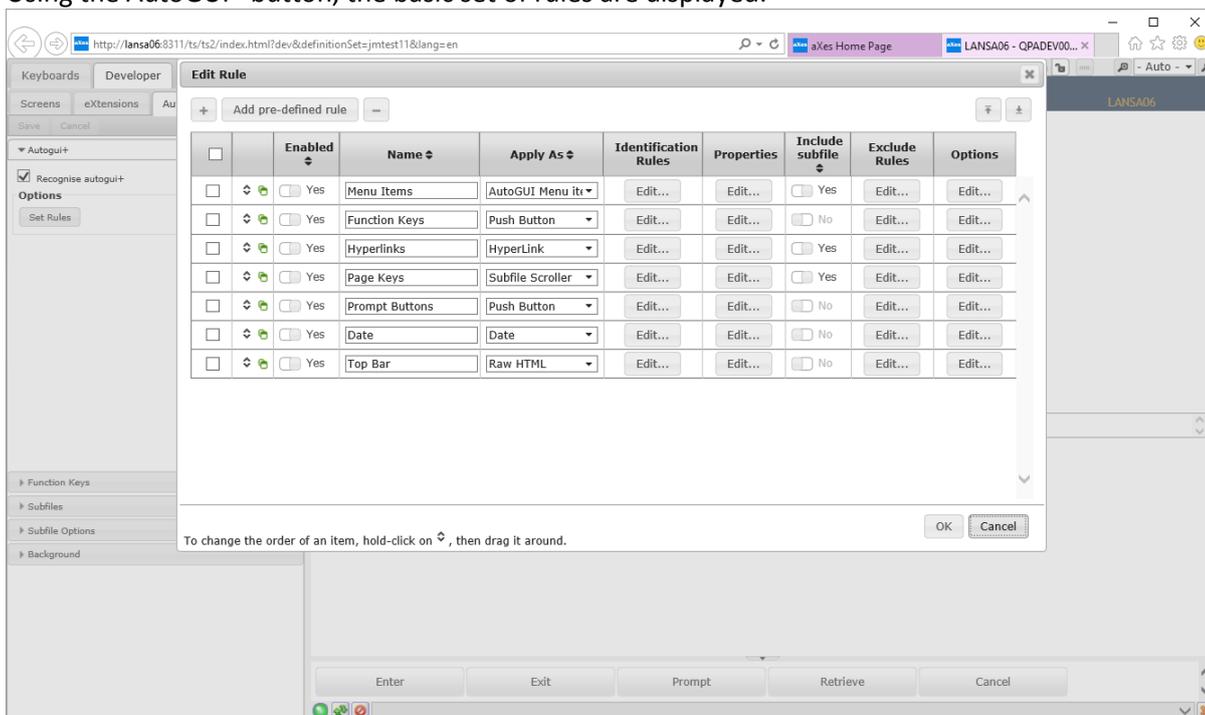
AutoGUI+ is a new feature which allows users to customize their projects by defining a set of rules. aXes will apply the rules automatically. These rules are very powerful in that they apply to each 5250 screen of your application. So with this, you can apply customisations on an application wide basis, rather than screen by screen. The feature is accessible from the AutoGUI tab and clicking on the AutoGUI+ set rules button.

Each rule is made of:

- a set of identification rules,
- one extension to apply to a target field, as defined in the identification rules,
- a set of extension properties to apply to the selected extension,
- a set of exclusion rules to avoid rules being applied to certain screen configurations.

We have also added some predefined rules to make it easier for the user to add common rules.

Using the AutoGUI+ button, the basic set of rules are displayed:



You can turn off as many rules you wish, using the enabled button.

The extensions available for AutoGUI+ assignment are:

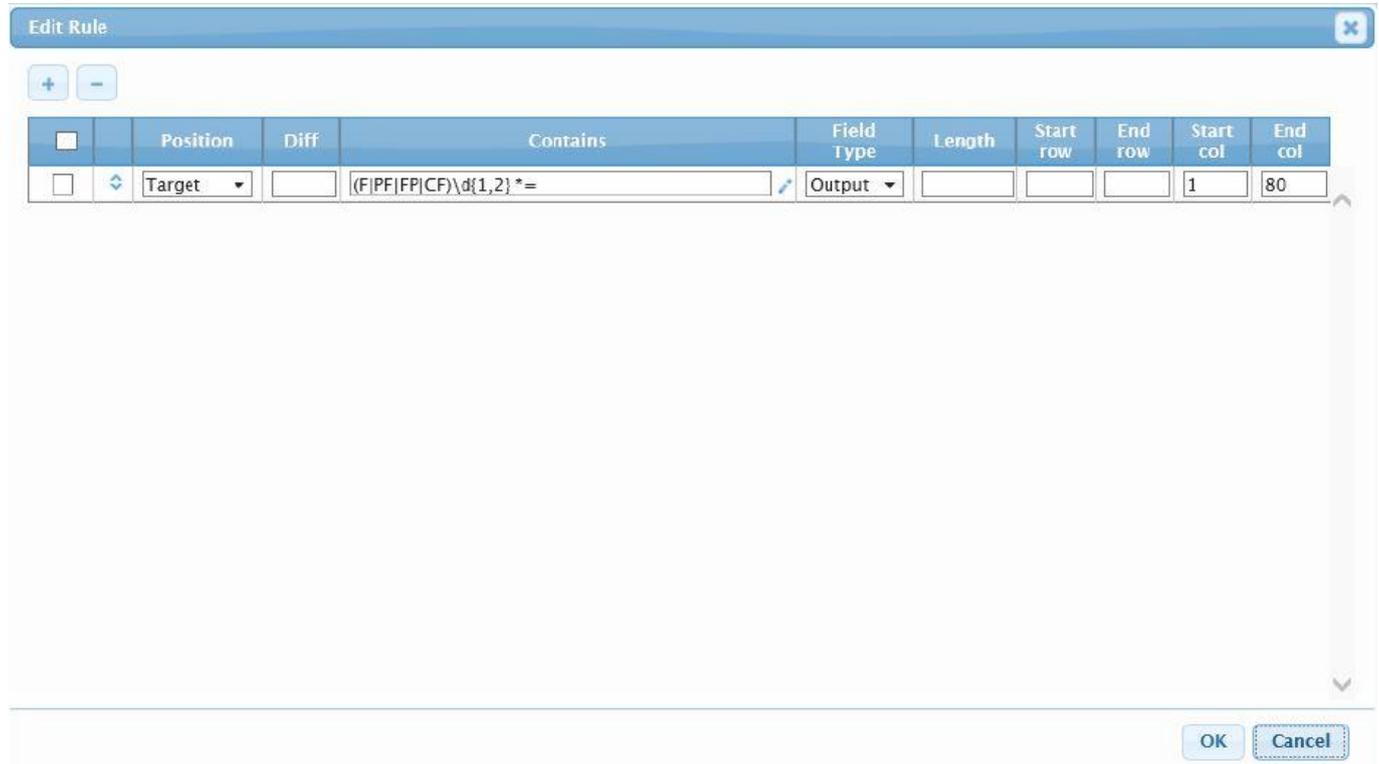
- AutoGUI+ menu Item
- Checkbox
- Date
- Dropdown
- Label
- Hyperlink
- Push Button
- Radio button
- Subfile scroller

A particular line-item is made of the following information:

- Name, for the user to customise.
- What is the extension to apply when this rule is satisfied?
- In which identification conditions is this rule true?
- What are the specific properties for the extension to apply?
- Is this rules to be applied to subfiles as well?
- In which conditions is this rule not applied?
- Finally, some extension specific conditions.

There is also export/import functionality allowing developers to easily copy rules between projects or save the rules for future use.

The identification rules allow the developer to define precisely when the rule is applied. It defines the target field, which is the field to which the selected extension should be applied, and any remarkable fields around it (left, right, top, bottom).



It is also possible to exclude any screens or screen element from the AutoGUI+ rules by defining an exclude rule. It can be via a screen name or by identification of a unit screen element.

Edit Exclusion Rules
✕

+
-

<input type="checkbox"/>		Action	Screen name contains	Element value contains	Field Type	Length	Start row	End row	Start col	End col
<input type="checkbox"/>	⬇	Exclude screen ▾	<input type="text"/>	dep458	Output ▾		1	1	1	15

OK Cancel

A regular expression is a special text string for describing a search pattern. We have provided a simple regular expression generator to help in producing regular expressions from logic blocks, complete with a regular expression tester.

Build Regular Expression
✕

+
-

Expression: dep458abc|object(def|ghi)

Target string: Test

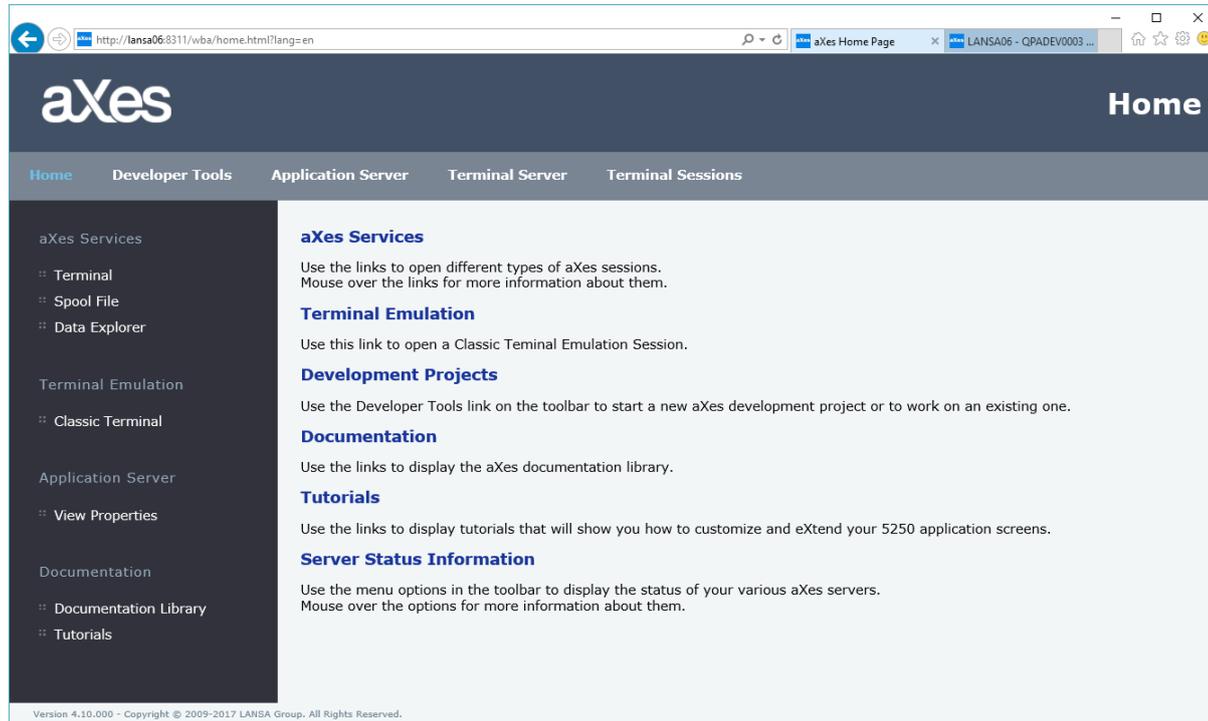
<input type="checkbox"/>		Option	Expression
<input type="checkbox"/>	⬇	Add ▾ ?	abc
<input type="checkbox"/>	⬇	Or ▾ ?	object
<input type="checkbox"/>	⬇	Start Group ▾ ?	
<input type="checkbox"/>	⬇	Add ▾ ?	def
<input type="checkbox"/>	⬇	Or ▾ ?	ghi
<input type="checkbox"/>	⬇	End Group ▾ ?	

OK Cancel

To change the order of an item, hold-click on ⬇, then drag it around.

New homepage

aXes 4.1.0 delivers a new homepage

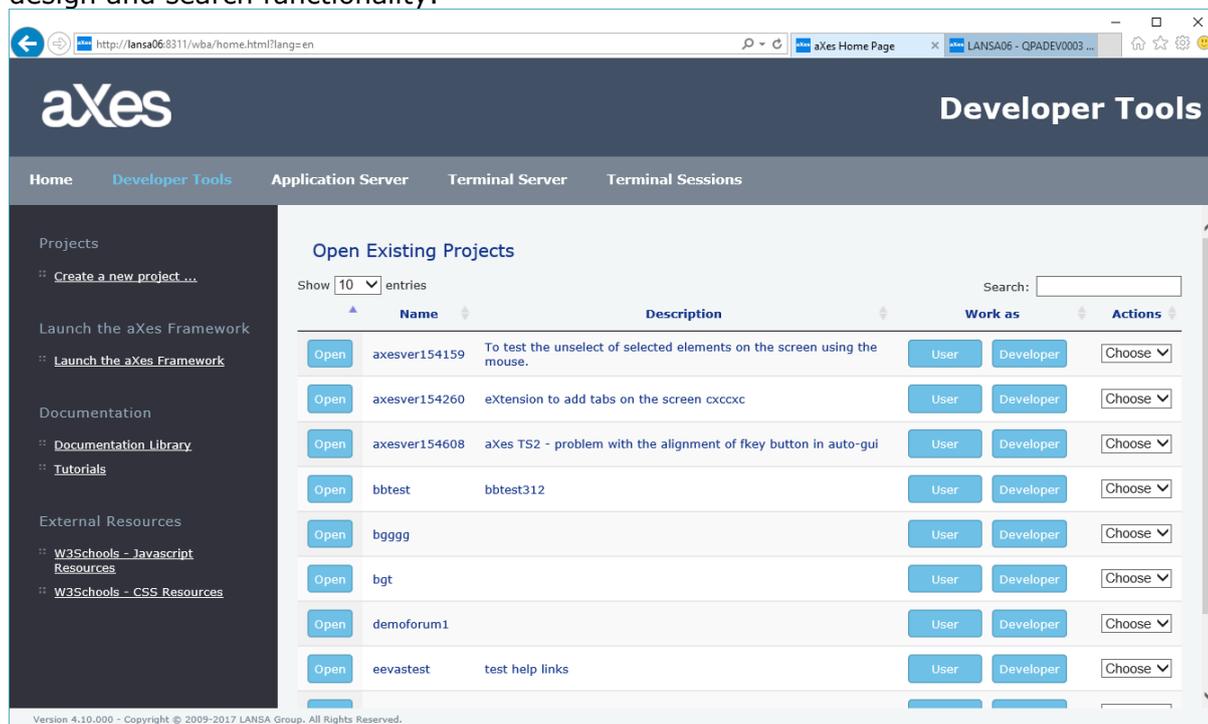


New developer's projects homepage

There is also a new developer's projects homepage.

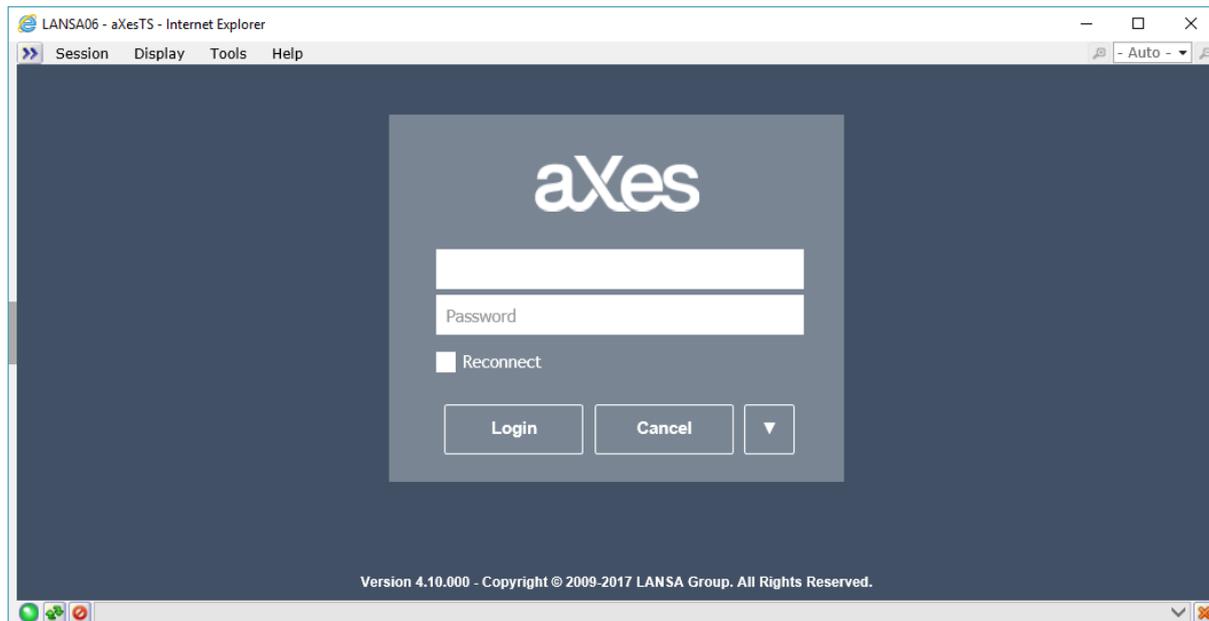
Note:

New quick access buttons for user (TS2) and Developer(TS2) straight off the list of projects, list design and search functionality.



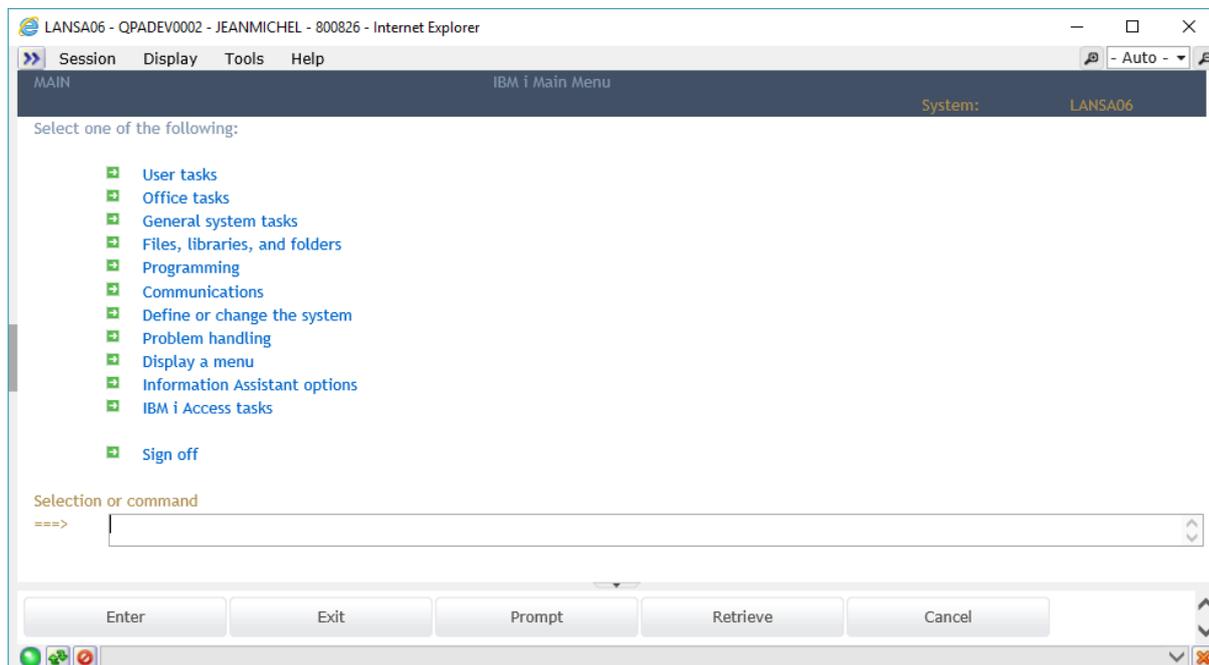
New TS2 logon design

aXes 4.1.0 also delivers a new TS2 logon design.



New Terminal Design out of the box

Note the colours, top banner, new menu icon and function key panel at the bottom.



2. New eXtensions

New IBM i services eXtensions

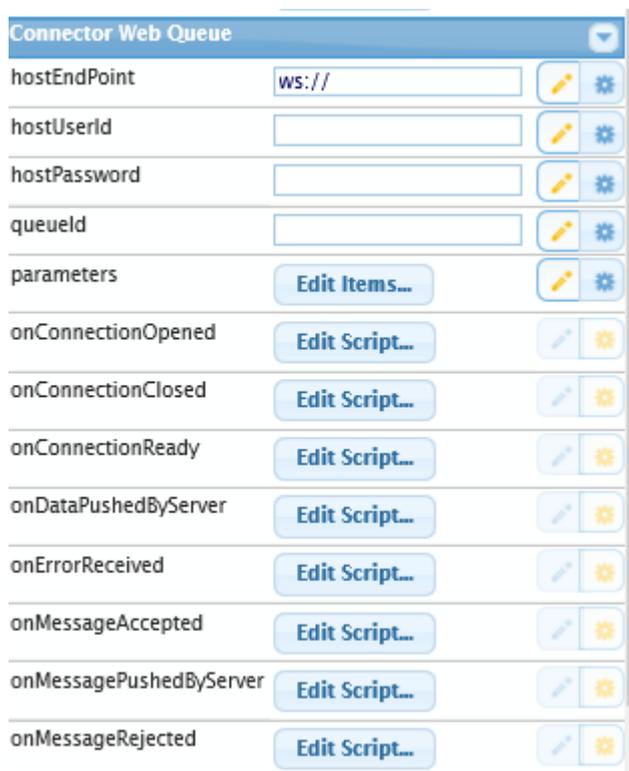
The suite of IBM i services extensions has been designed to work with aXes to provide advanced IBM i functionality to increase business value of your applications.

WebQueue

WebQueue is an API that provides communication between a web application and a server via a WebSocket connection.

Using WebQueue, developers can send/receive messages to/from program data queues using JavaScript via a WebSocket connection. Programs monitoring the program data queues can access resources and services on an IBM i server.

Developers use JavaScript and other development tools when building web applications. At points in the applications where developers require server resources, they can use the WebQueue API to access the resources. WebQueue provides web applications with a full-duplex connection to a server, over which the application and the server can exchange messages.



ConnectorJS

The WebQueue API is service provided by a feature called ConnectorJS that comes with the browser accessible ConnectorJS JavaScript library that interfaces with ConnectorJS Server to access requested server resources. ConnectorJS Server invokes the resources to produce the required results, and returns responses to web applications.

ConnectorJS also equips developers with interfaces they can use from JavaScript to access server resources in web applications. Server resources include programs, commands, files and folders, email, and messaging using program data queues and/or the Java Message Service (JMS) API.

The properties can be set via the developer tools (show below) or programmatically through the aXes API. See below for the properties available to the developer.

ConnectorJS Email

The ConnectorJS email API allows developers to assemble and send an email.

To use the API, developers code the API interface using JavaScript and deploy the ConnectorJS JavaScript library with the web application JavaScript. This API requires data defining server resources (e.g. a command, program, or data queue) and an IBM i server (host).

Email content includes subject, addressee (to), sender (from) and email body.

Developers can assemble all email details required to create generic email messages. Use the generic assembly method for emails with various subjects sent to recipients chosen when preparing an email. Inserting some of the email details in the configuration file allows CJS to use details provided via the API call and configuration details to assemble an email. This assembly method is useful for pre-prepared or specific subject matter emails such as support requests or sales information. Most of the email details can reside in the configuration file and developers can provide the email body content via a property in the API call.

To:	TO Recipients here
Cc:	CC Recipients here
Bcc:	BCC Recipients here
Subject:	Subject here
Message here	
<input type="button" value="Clear"/> <input type="button" value="Send"/>	

ConnectorJS Program Call

The ConnectorJS program call API allows developers to call a program.

The API call can return a single value once the call has been processed. The properties can be set via the developer tools (show below) or programmatically through the AXES API.

See below for the properties available to the developer.

Connector Program Call		
hostEndPoint	<input type="text" value="http://"/>	 
hostUserId	<input type="text"/>	 
hostPassword	<input type="text"/>	 
program	<input type="text"/>	 
parameters	<input type="button" value="Edit Items..."/>	 
asynchronous	<input type="checkbox"/>	 
onErrorReceived	<input type="button" value="Edit Script..."/>	 
onPgmCallCompleted	<input type="button" value="Edit Script..."/>	 

ConnectorJS Print Service

Print a report from a browser using the ConnectorJS Print Service (CJS).

The CJS Print Service enables developers to prepare a report and send it to a printer accessible from a browser. The Print Service collates data formatted as JSON with a report layout and content specification to produce the report. It can accept data from spooled files, data from CJS Query results, data from an aXes terminal screen, or data generated by JavaScript.

What can the print service do? Suppose you want to take invoice details from an IBM i spooled file and print an invoice, adding details that do not appear in the spooled file, e.g. a company logo. Another example is printing a receipt for an online order.

Developers create a report layout by defining page elements including text, images, and graphics, and their page location. Page properties include, title, DPI (page resolution), size (e.g. A4), orientation (portrait and landscape). Each element (e.g. text, bar code, line, rectangle, or image) on the page has a set of properties defining how to present the element. The properties include location on the page (x and y coordinates), font, colour, line properties (e.g. style and width), rotation, and URL representing an image location.

New common eXtensions

New common extensions are available for desktop and mobile, and only for TS2.

Multiline Edit Box

The Multiline Edit Box allows user input of more than one line of text. The text display depends on the size of this extension. The text will go to the next line when it overflows the width while vertical scrollbar becomes visible when the text overflows the height.

3. Developer tools

Developer tools enhancements

TS2 development tool modernization

The TS2 development tool has been enhanced with the following features.

- The editor has been updated to apply the jQuery UI theme throughout the User Interface in a consistent manner, providing the developer with a unified look and feel.
- The theme of the developer tools is customizable through the configuration file.

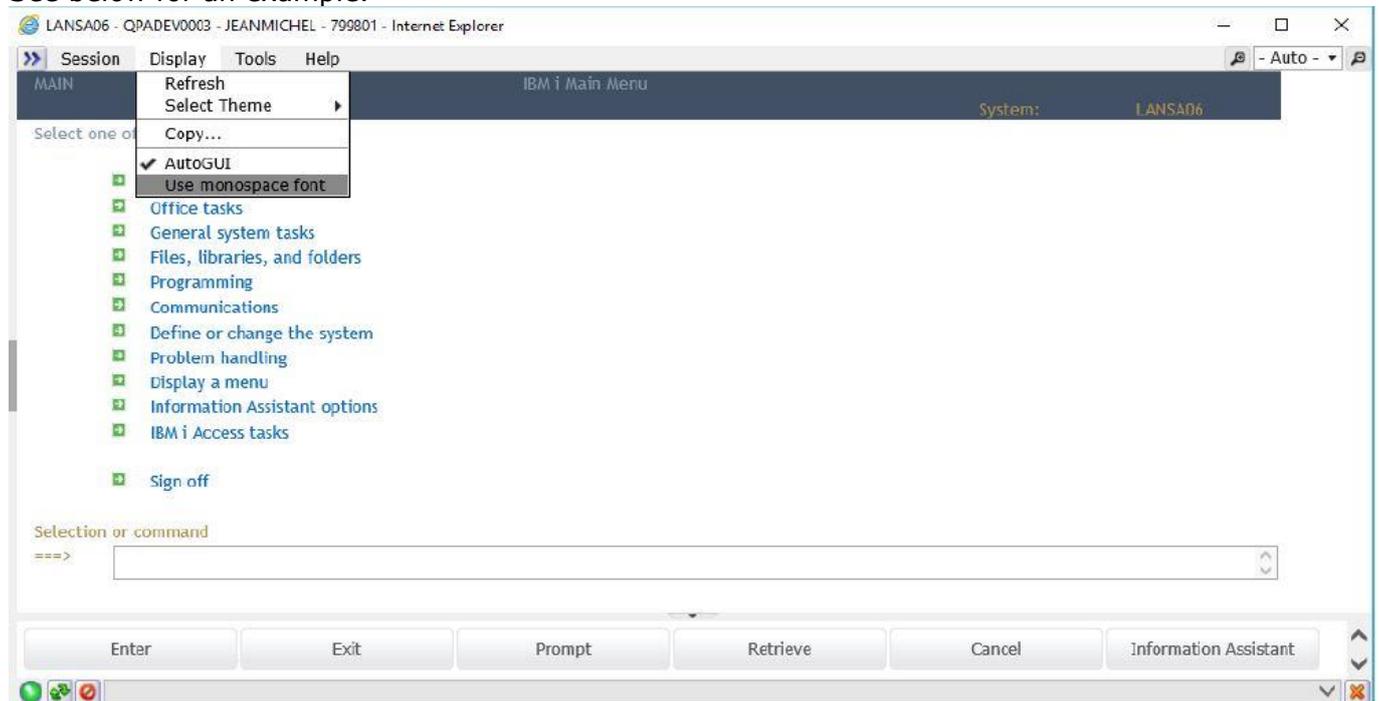
4. Compatibility considerations

With this release, we have introduced many performance and visual improvements. In this respect, the user is advised to read the following point which explains changes which might impact their application.

Proportional fonts vs Monospace fonts

Axis 4.1.0 has introduced proportional fonts thus screens might look somewhat different and misaligned in some circumstances. To help with these changes we have introduced an option on the display menu of the terminal window which allows the user to switch easily from proportional to monospace font.

See below for an example.



Terminal styles

The styles throughout the product utilize the jQuery themes more consistently. Consequently, some screens might have a different visual appearance since the old CSS themes are being retired.

One solution is to use the new AutoGUI+ to apply the terminal styles to your application.

You can add them by adding a predefined rule in AutoGUI+ called Terminal Themes.

The screen below shows the Copy File screen with jQuery Themes only.



The screen below shows the Copy File screen with jQuery Themes and Terminal Styles active.



AutoGUI vs AutoGUI+

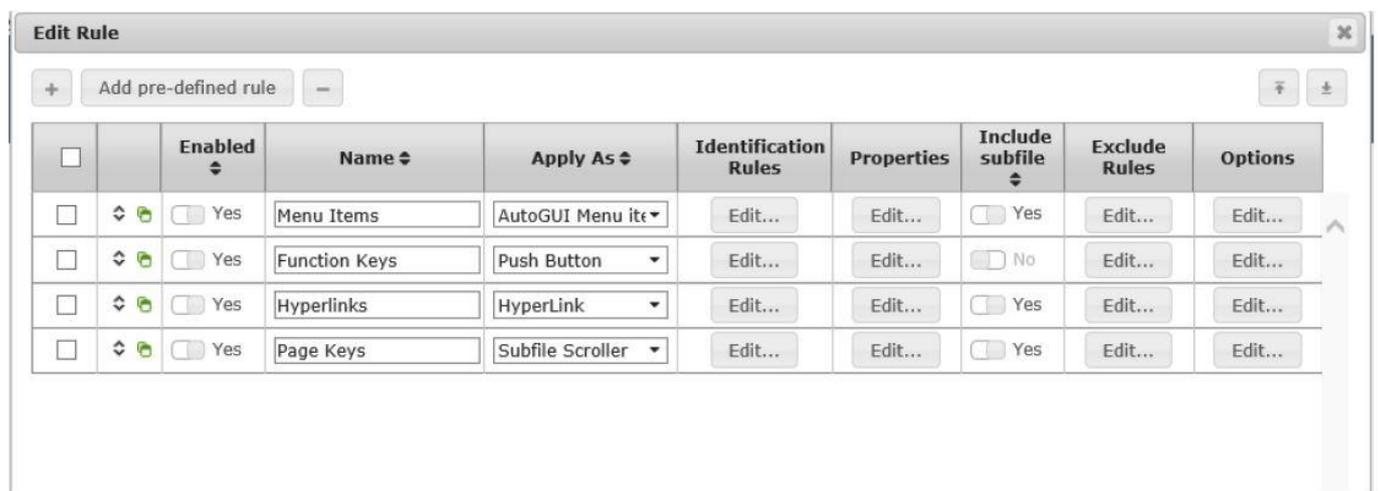
AutoGUI refers to the legacy automatic refacing feature of previous aXes releases. AutoGUI+ is the new feature released with this version.

AutoGUI+ is an upgrade from AutoGUI rules. In some cases, your existing projects might implement upgraded AutoGUI+ rules. This could be the case if you have never changed the default AutoGUI rules in your project, therefore never saved any version of your rules.

To help user who wish to keep the existing AutoGUI behaviour we have added an import of the basic AutoGUI rules.

To revert to the legacy rules, open the AutoGUI+ window, check all the rules displayed and click 

After confirmation, you have an empty list. In the same popup click on import  and select the file /ts/screens/defaultautoguirules.json. After validating the import, the user will have reverted to the legacy set of rules.





New VLF version

The new features in the EPC141017 version of the Framework are:

Add your own layer of object security in VLF-ONE

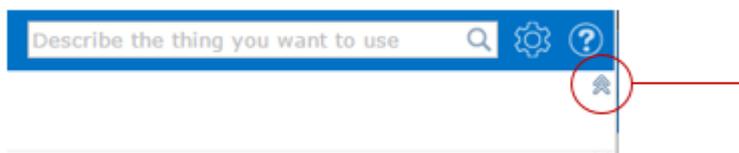
You can now replace the standard VLF-ONE authority model with your own custom version just like in VLF-WIN.

Component snap-in prompter enhancements

The snap-in part prompter for Framework developers is now quicker and easier to use, for example a filter snap-in prompter will only show filters.

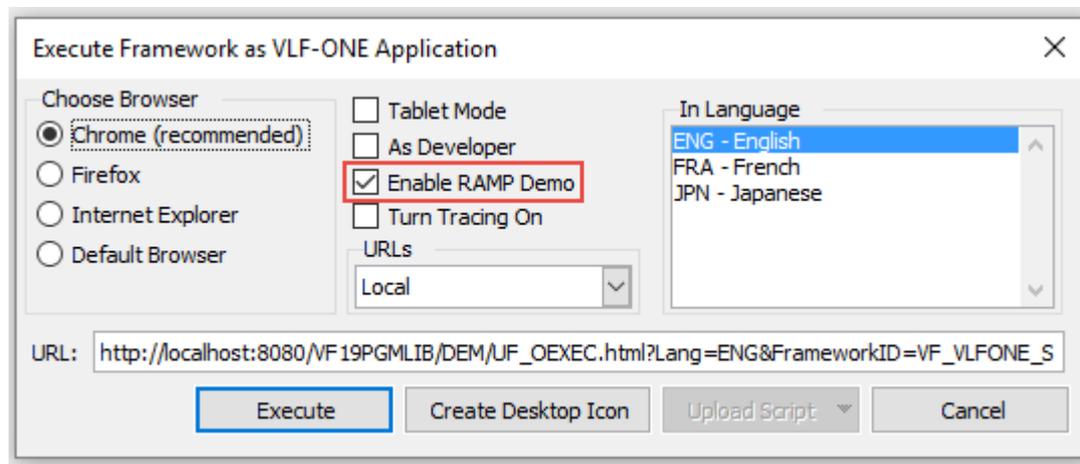
VLF-ONE toolbar collapse/expand remembered

The state of the option to collapse or expand the VLF-ONE tool bar is now remembered from session to session.



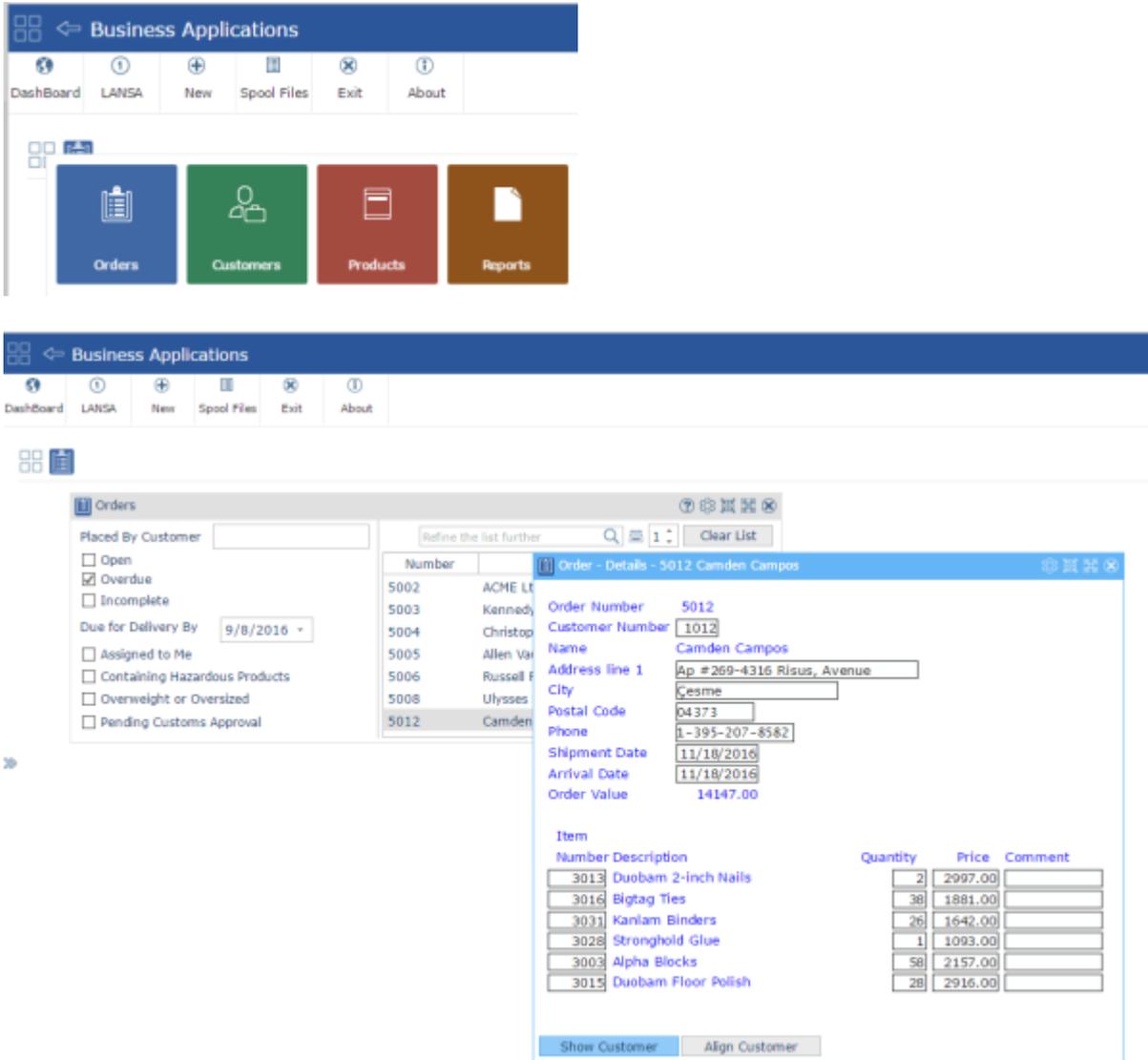
RAMP VLF-ONE demonstration system

The VLF-ONE launch point has a new option to enable shipped RAMP demonstration material within the Framework:



The material is designed to facilitate RAMP demonstrations and training.

When the demo is enabled, the Orders, Customers and Products business applications all start to use RAMP sessions:



The demonstration system is a representational 5250 application, not a real 5250 application. it is designed to be used in demos and support the story line defined in this video: <http://www.lansa.com/resources/videos.htm>.

Signon user profile returned in VLF-ONE

User profile validation at signon in VLF-ONE has been made more flexible.

VLF-ONE JavaScript files combined for better reliability

All VLF-ONE internal JavaScript files can be combined into a single file to improve the reliability of the user's first use of VLF-ONE.

UF_OEXEC has been changed to load the single file instead of loading all the individual VF_*.js files. (See the integration tab for component UF_OEXEC.)

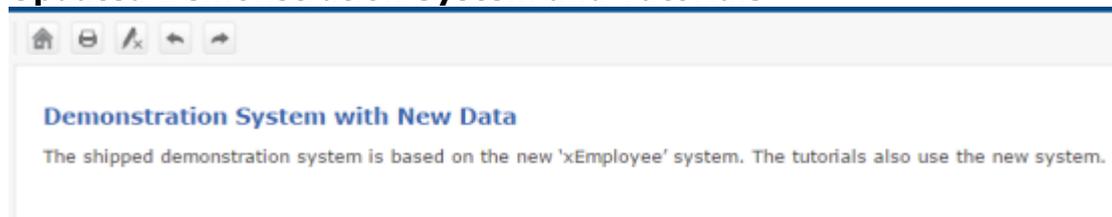
To take advantage of this feature, modify your version of UF_OEXEC, and add a script VLFONE_Engine.js under the Integration tab.

Combining the JavaScript files is optional. Pre-existing versions of UF_OEXEC will continue to work if not changed.

Port is defaulted when executing Framework as a VLF-ONE application

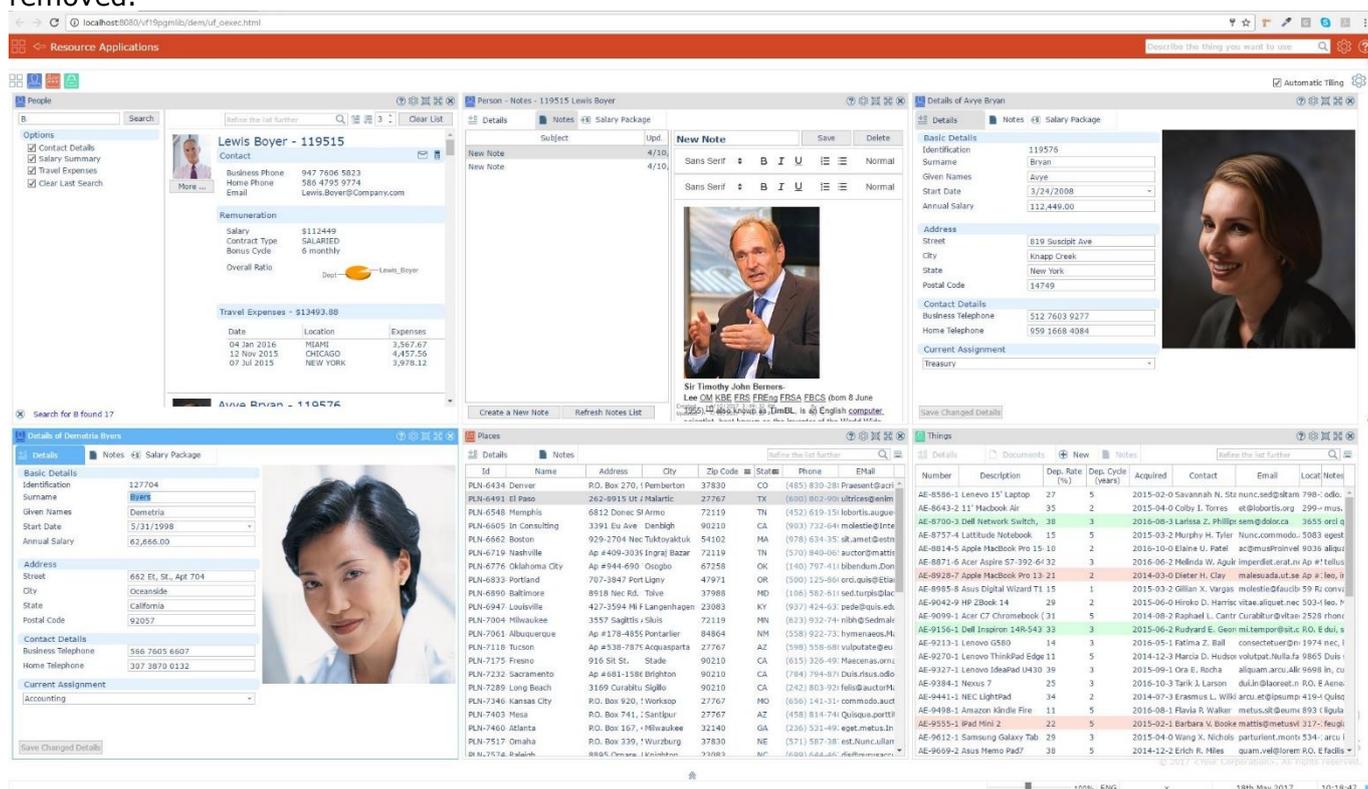
When you execute the Framework as a VLF-ONE application, the default port is automatically set.

Updated Demonstration System and Tutorials



All shipped demonstration material has been changed over to use the newer 'xEmployee' data base tables.

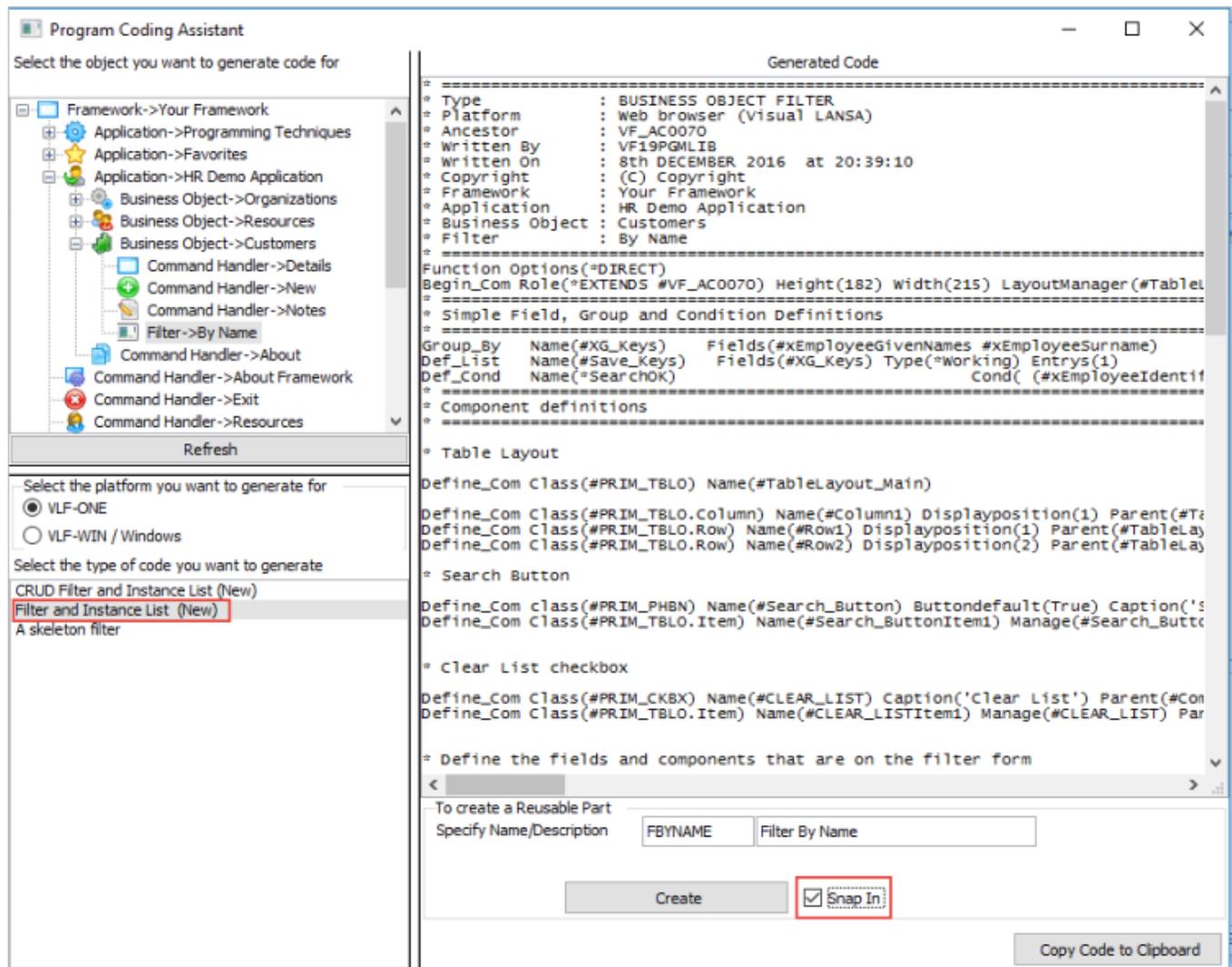
References to the old PSLMST, PSLSKL, DEPTAB, SECTAB and SKLTAB database tables have been removed.



Code Assistant enhancements

The process by which Code Assistants generate components has been streamlined.

Program Coding Assistants now generate, compile and snap in components in one process.

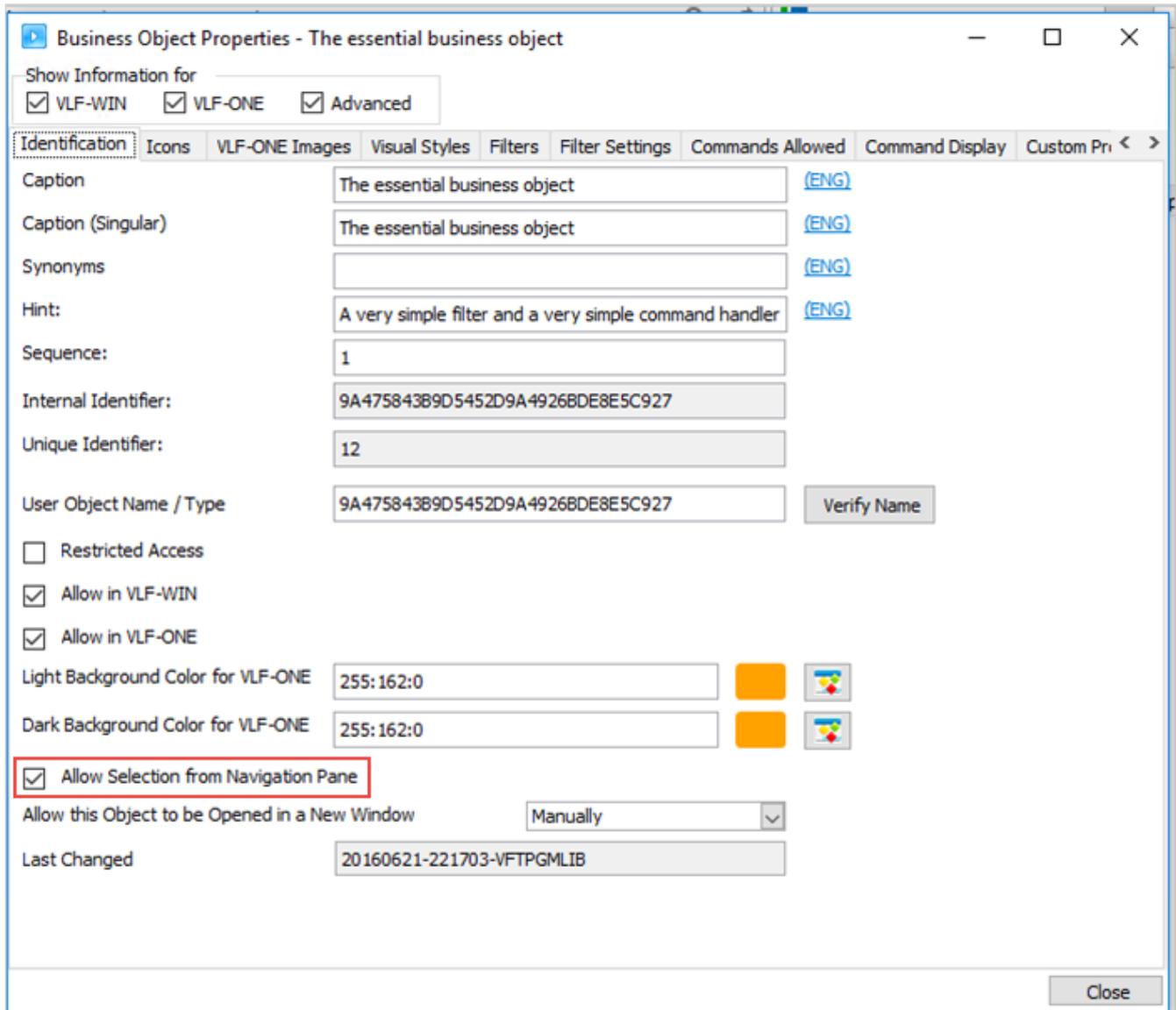


Now when you execute a coding assistant to generate a filter or a command handler, you can choose the instance list columns or fields, generate, compile and snap in the component in one go.

The new filter or command handler will appear in your Framework and start working immediately.

Control the visibility of VLF-ONE business objects in the navigation pane

The business object property Allow Selection from Navigation Pane is now available in VLF-ONE.



Page-at-a-time instance list

A Page-at-a-Time Instance List is now available in VLF-ONE.

A new type of simple instance list browser named VF_UM0440 is now shipped. It uses page-at-a-time display technique that may use less system resources when displaying large instance lists.

When an instance list is simple the default instance list browser is VF_UM0400.

You can specify the name VF_UM0400 on the business object's Instance List / Relationships tab, but generally it will correctly default for a simple instance list.

VLF-ONE Snap in Instance list Browser Id

This is an example of a VF_UM0400 displayed instance list:

Id	Name	Address	City	Zip Code	Phone	EMail
PLN-6434-5	Denver	P.O. Box 270, 93; Pemberton		37830	(485) 830-2885	Praesent@acrisus
PLN-6491-5	El Paso	262-8915 Ut Ave Malartic		27767	(600) 802-9065	ultrices@enimNui
PLN-6548-5	Memphis	6812 Donec Strei Armo		72119	(452) 619-1507	lobortis.augue@ir
PLN-6605-5	In Consulting	3391 Eu Ave Denbigh		23083	(903) 732-6469	molestie@Integer
PLN-6662-5	Boston	929-2704 Nec Av Tuktoyaktuk		54102	(978) 634-3536	sit.amet@estmau
PLN-6719-5	Nashville	Ap #409-3039 Di Ingraj Bazar		43744	(570) 840-0651	auctor@mattis.ca
PLN-6776-5	Oklahoma City	Ap #944-690 Var Osogbo		67258	(140) 797-4101	bibendum.Donec.
PLN-6833-6	Portland	707-3847 Porttiti Ligny		47971	(500) 125-8609	orci.quis@Etiamb
PLN-6890-6	Baltimore	8918 Nec Rd. Tolve		37988	(106) 582-6105	sed.turpis@iacus.
PLN-6947-6	Louisville	427-3594 Mi Rd. Langenhagen		36131	(937) 424-6339	pede@quis.edu
PLN-7004-6	Milwaukee	3557 Sagittis Ave Sluis		29403	(623) 932-7445	nibh@Sedmalesu
PLN-7061-6	Albuquerque	Ap #178-4859 Ei Pontarlier		84864	(558) 922-7321	hymenaeos.Maur
PLN-7118-6	Tucson	Ap #538-7879 Ti Acquasparta		94248	(598) 558-6886	vulputate@eu.edu
PLN-7175-6	Fresno	916 Sit St. Stade		77441	(615) 326-4930	Maecenas.ornare.
PLN-7232-6	Sacramento	Ap #681-1586 Si Brighton		36063	(784) 794-8766	Duis.risus.odio@f.
PLN-7289-6	Long Beach	3169 Curabitur S Sigillo		65823	(242) 803-9269	felis@auctorMaur
PLN-7346-6	Kansas City	P.O. Box 920, 57i Worksop		21007	(656) 141-3146	commodo.auctor.
PLN-7403-6	Mesa	P.O. Box 741, 24; Santipur		64837	(458) 814-7407	Quisque.porttitor.
PLN-7460-6	Atlanta	P.O. Box 167, 49; Milwaukee		32140	(236) 531-4924	eget.metus.In@ti
PLN-7517-6	Omaha	P.O. Box 339, 50i W?rzburg		97210	(571) 587-3879	est.Nunc.ullamco
PLN-7574-6	Raleigh	8895 Ornare, Roe Knighton		79701	(699) 644-4634	dis@purusaccums
PLN-7631-6	Miami	Ap #122-5403 Ei Carstairs		57623	(803) 439-4565	dolor@malesuada
PLN-7688-6	Oakland	7482 Ullamcorper Dornoch		19490	(819) 811-8028	fames.ac.turpis@
PLN-7745-6	Minneapolis	Ap #420-7322 Lt Poviglio		74623	(903) 328-3357	justo.Proin@liber
PLN-7802-7	Tulsa	P.O. Box 451, 11i Sint-Pieters-Kapel		98020	(279) 407-3076	lectus.a@semmol

Page-at-a-Time Instance List VF_UM0440

If you use very large simple instance lists, you might consider using VF_UM0440 instead.

VF_UM0440 is very like VF_UM0400 except that it presents data in a page at a time format. Using page at a time display technique may reduce the amount of browser resources required to display an instance list.

To use VF_UM0440 specify it as the snap in instance list browser:

VLF-ONE Snap in Instance list Browser Id

You can tell that it is working because it displays a pagination control area above the instance list:

Page 1. Showing items 1 (G00000) to 40 (G00039) of 2000 Orders.

First <<Back 1 2 3 4 5 6 7 8 ... 49 50 Next>> Last

VF_UM0440 is designed to work in conjunction with the quick search field above it because this allows users to quickly locate items in very large lists.

The default page size is 40. You can change the page size at any time by using this type of operation:

```
(#AVLISTMANAGER.InstanceListBrowser *As #Vf_um0440).avPageSize := 100
```

Responsive design testing for VLF-ONE applications

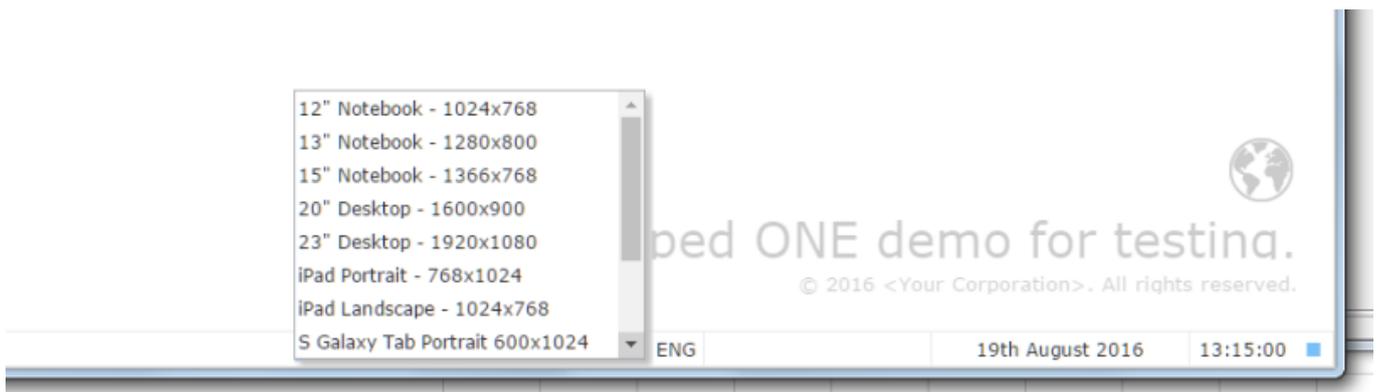
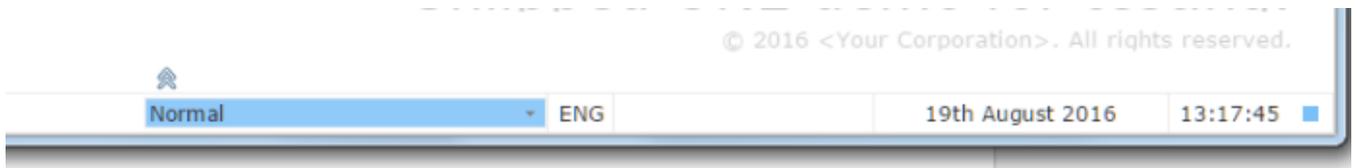
You can now dynamically modify the size of your application's browser window to have an approximate idea of how it will look on different sized screens.

The option to emulate sizes is available in URLs which include Developer=YES.

To see what the emulation looks like start the latest shipped VLF-ONE demo:



And execute the Framework as a VLF-ONE application. Use the drop-down on the status bar to change the screen size:



Note that the emulation is simple, and sometimes popup menus, floating panels, etc. may project outside the emulated monitor. This usually happens from the right hand or bottom edges.

Upload/Download examples

The new Upload/Download Examples show techniques for uploading and downloading data.

aXes-TS2 language code

The aXes-TS2 language code is the language code used in RAMP sessions associated with the Visual LANSAs Framework/LANSAs language. You only need this value if you are using RAMP and the aXes-TS2 execution engine inside a VLF-WIN or VLF-ONE application.

aXes language codes follow the 2-character web browser language codes, so "en" is English, "fr" is French, "ja" is Japanese, etc.

If you do not specify the language code, the aXes-TS2 execution engine defaults to the current language of the user's web browser.

RAD-PADs support HTML5

All RAD-PADs are now HTML5 compliant.

Silent signon example available at the Visual LANSAs Forum

Learn how a user who is logged on to a VLF-WIN Framework can start a VLF-ONE session without logging on again.

See [Starting a VLF-ONE session from VLF-WIN without logging on again](#).

No WAM, WEBEVENT or Fast Part samples

WAM, WEBEVENT and fast part demonstration and examples are no longer shipped.

CSV separator character can be changed

In some countries, the "," (comma) is not used as the CSV file separator.

A new system wide property #uSystem.DefaultCSVSeparator can be set to specify the character to be used.

This impacts trace files and instance lists sent to MS-Excel, and it can be seen being used in the download example DF_T68H40.

Coming soon in VLF

A new VLF version is due a for delivery soon.

Here’s a preview of just a few of the new features:

Column Filtering

End users can now filter instance list items by a column’s content.

For example, here is an employee instance list that has been filtered to show employees who live in two states only:

Identification	Surname	Given Names	Date of Birth	Gender	State	Annual Salary	Start Date
119886	Alston	Kim	10211987	Female	Massachu	9 05242008	
130509	Ferguson	Imelda	04101977	Female	Massachu	9 11131997	
126647	Fowler	Lawrence	02141986	Male	Delaware	0 09172005	
135214	Moses	Carlos	05311955	Male	Massachu	0 01031997	
144832	Patterson	Sade	03231962	Female	Massachu	5 10261997	
129210	Sanford	Karyn	12211973	Female	Delaware	4 07242002	

California (19)
 Delaware (2)
 Illinois (9)
 Massachusetts (4)
 New York (13)
 Ohio (10)
 Pennsylvania (18)
 Texas (15)
 Virginia (10)
 Clear Selection
 Close

And here is a gender column being used to show just Male or Female employees:

Quick Search

Date of Birth	Gender	State	Annual
03231961	Female		
12211973	Female		
07281971	Female		
01071950	Male		
08121989	Female		
04211970	Male	Virginia	
10021965	Male	Pennsylvania	
02111957	Male	Ohio	
04231990	Male	Texas	
09251980	Female	New York	
03261956	Male	Illinois	

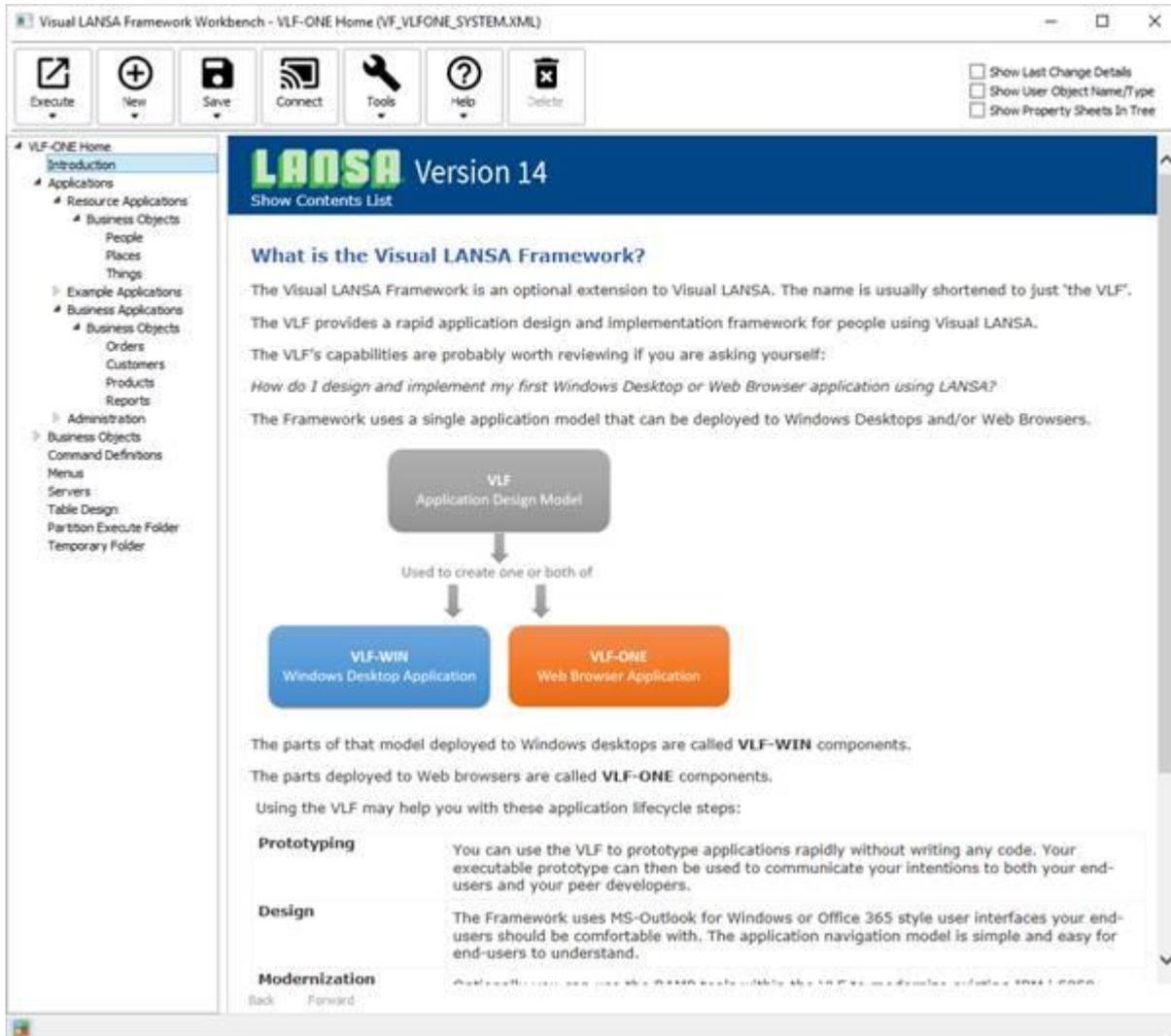
Female (5)
 Male (7)
 Clear Selection
 Close

No coding is required to use this new filtering feature; you just have to enable it in your framework’s design.

New design and role for the VLF designer workbench

The existing VLF design workbench is being given a new simpler user interface.

It will also become the main place that VLF development activities are initiated from:

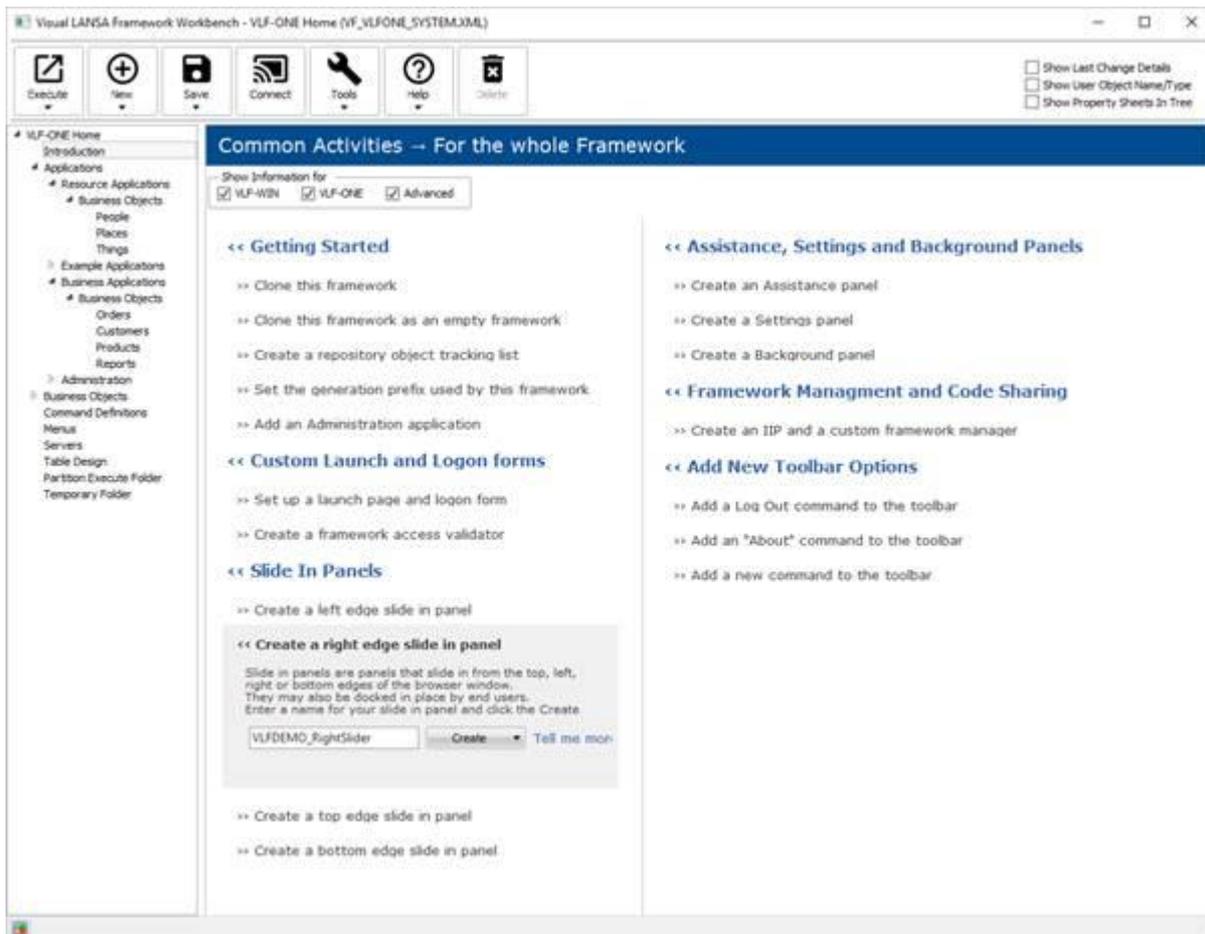


New Common Activities feature

The workbench redesign features has two new Common Activities panels, one that works at the framework level, and one that works at the business object level.

The options on these panels should make many common VLF developer activities simpler to understand and faster to complete.

For example, the expended common activity will generate, compile and snap in a VLF-ONE right edge slider pane in just one click:



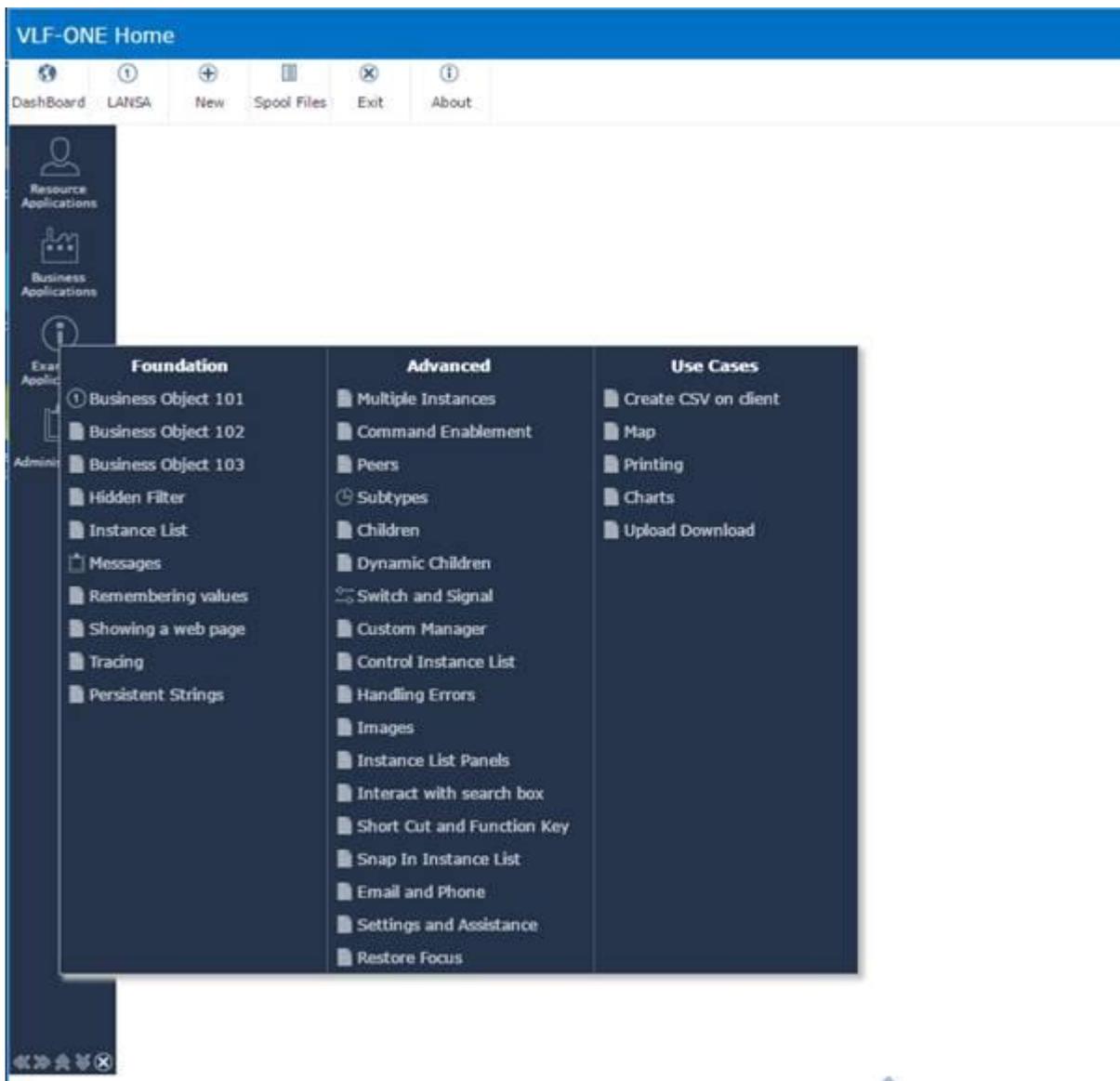
New VLF-ONE Navigation Menu Options

VLF-ONE's navigation menu style is being extended to allow end users to use a more traditional navigation menu style.

By using these new options an end user can cause the navigation menu to dock to the left, right, top or bottom of the browser window:



For example, this is what the menu looks like when docked to the left edge of the browser window:



QFRCOBJCVN system value can cause V14 SP1 LANSA for i upgrade failure

When upgrading to V14 SP1, a few customers have experienced an upgrade failure with the following messages in the joblog.

```
CPF37A7 Escape 30 11/01/17 13:46:09.639393 QSRROCPR QSYS 064B DC@LOADERU LANSAINS01 *STMT
To module . . . . . : DC@LOADERU
To procedure . . . . . : DC@LOADERU
Statement . . . . . : 227500
Message . . . . . : FRCOJCVN value not allowed.
Cause . . . . . : The value you specified for FRCOJCVN is not allowed with
the current value of the QFRCCVNRST system value. Recovery . . . : The
following combinations of FRCOJCVN and QFRCCVNRST are allowed: -- If
FRCOJCVN(*NO) is specified, then the QFRCCVNRST system value must have a
value of either 0 or 1. -- If FRCOJCVN(*YES *RQD) is specified, then the
QFRCCVNRST system value must have a value of either 0, 1, or 2. -- If
FRCOJCVN(*YES *ALL) is specified, then the QFRCCVNRST system value can have
any valid value and FRCOJCVN(*YES *ALL) will override the system value.
CPF37A7 Information 30 11/01/17 13:46:09.640889 DC@LOADERU LANSAINS01 *STMT DC@LOAD25 LANSAI
NS01 *STMT
From module . . . . . : DC@LOADERU
From procedure . . . . . : DC@LOADERU
Statement . . . . . : 231600
Statement . . . . . : 5700
Message . . . . . : INS0160 = Upgrade point of no return was passed. Do not
use the system. Do not attempt to upgrade again..
Cause . . . . . : This message is used by application programs as a general
escape message.
```

A change was included in V14 and V14 SP1 to have FRCOBJCVN(*YES) on all RSTOBJ (& RSTLIB) commands when importing on IBM i. However, a conflict can arise between the FRCOBJCVN(*YES) and the QFRCCVNRST system value, when it is set greater than 2.

Solution

The solution is to:

- End the upgrade
- Restore your LANSA system from backup
- Set the QFRCCVNRST system value to an allowed value i.e. 0
- Rerun the upgrade

Notes

- You must restore from backup as per the message in the upgrade joblog *INS0160 = Upgrade point of no return was passed. Do not use the system. Do not attempt to upgrade again.*
- This issue can affect both LANSA installs and LANSA upgrades.
- The next version of the LANSA installation software will be enhanced to prevent this conflict affecting the install/upgrade.

Performance slowdown in LANSAs applications caused by Mutex wait issue

LANSAs customers who have large systems involving many thousands of LANSAs runtime jobs have sometimes reported a slowing down in performance of the LANSAs jobs. In extreme cases, the performance can become so slow that the LANSAs runtime jobs stop responding.

Symptom

One of the most obvious symptom that you are experiencing this issue is that all jobs will be sitting on a MTXW condition so very little can be processed.

Background

The problem can occur when there is a very large number of LANSAs runtime jobs starting at about the same time – in one particular case it could have been about 6,000 requests (the number will depend on machine/subsystem configuration). The job call stacks showed the jobs were in the job initialisation stage (not yet in LANSAs code) and were waiting on a mutex from IBM program QTCP/QTMSUTL72, all waiting for the same IFS file.

Solution

IBM have acknowledged this Mutex contention issue and have produced a PTF to resolve the issue. The link to the APAR to resolve the issue:

<http://www-01.ibm.com/support/docview.wss?uid=nas2SE66961>

The PTF to apply is **SI64349**

Note: This is an IBM i 7.2 and later issue only as QTMSUTL72 is an IBM service program used for sending email and is new to IBM i 7.2.

Customer testimony

We have applied the PTF and have seen very positive improvements. Prior to applying to production, we tested the PTF by starting thousands of LWEB_JOBS into a held queue and then released that queue. Prior to the PTF's we saw large amounts of Mutex wait contention. After the PTF the Mutex contention all but disappeared and the jobs started almost immediately even when we started 10,000 jobs at once. Analysis since the PTF has been applied shows MTXW contention time reduced to almost nothing on LWEB_Jobs so the issue has been resolved.

Latest News

LANSA EPC

EPC141031 is now released with the latest VL web changes, with scrolling fixes and enhancements a particular focus.

See:

<http://www.lansa.com/support/notes/epc/epc141031.htm>

Web Server

LANSA for the Web will stop working after upgrading the O/S from 7.1 to 7.2 or 7.3.

After an Operating System upgrade from 7.1 to either 7.2 or 7.3, the HTTP server instance for V14 or V14 SP1 will not start and a HTP8016 error will be logged in the http joblog. A patch is available from LANSA that will install a 7.2/7.3 ready version of the Apache Module. A different patch is required depending on whether you are on V14 GA or V14 SP1. Contact your local LANSA Support group and request the patch appropriate to your version of LANSA V14.

V14 SP1: HF141001.zip

or

V14 GA: HF140005.zip

See:

<http://www.lansa.com/support/v14news/>

IBM

IBM have announced a date for withdrawing support for 7.1 (April 30, 2018) and Apache 2.2 (December 31, 2017).

See:

http://www-01.ibm.com/common/ssi/ShowDoc.wss?docURL=/common/ssi/rep_ca/0/897/ENUS917-080/index.html&lang=en&request_locale=en