LANSA NEWSLETTER

LANSA V15

We are very proud to announce LANSA Version 15. It contains lots of new features and enhancements that are detailed it in this Newsletter.

INSIDE THIS ISSUE

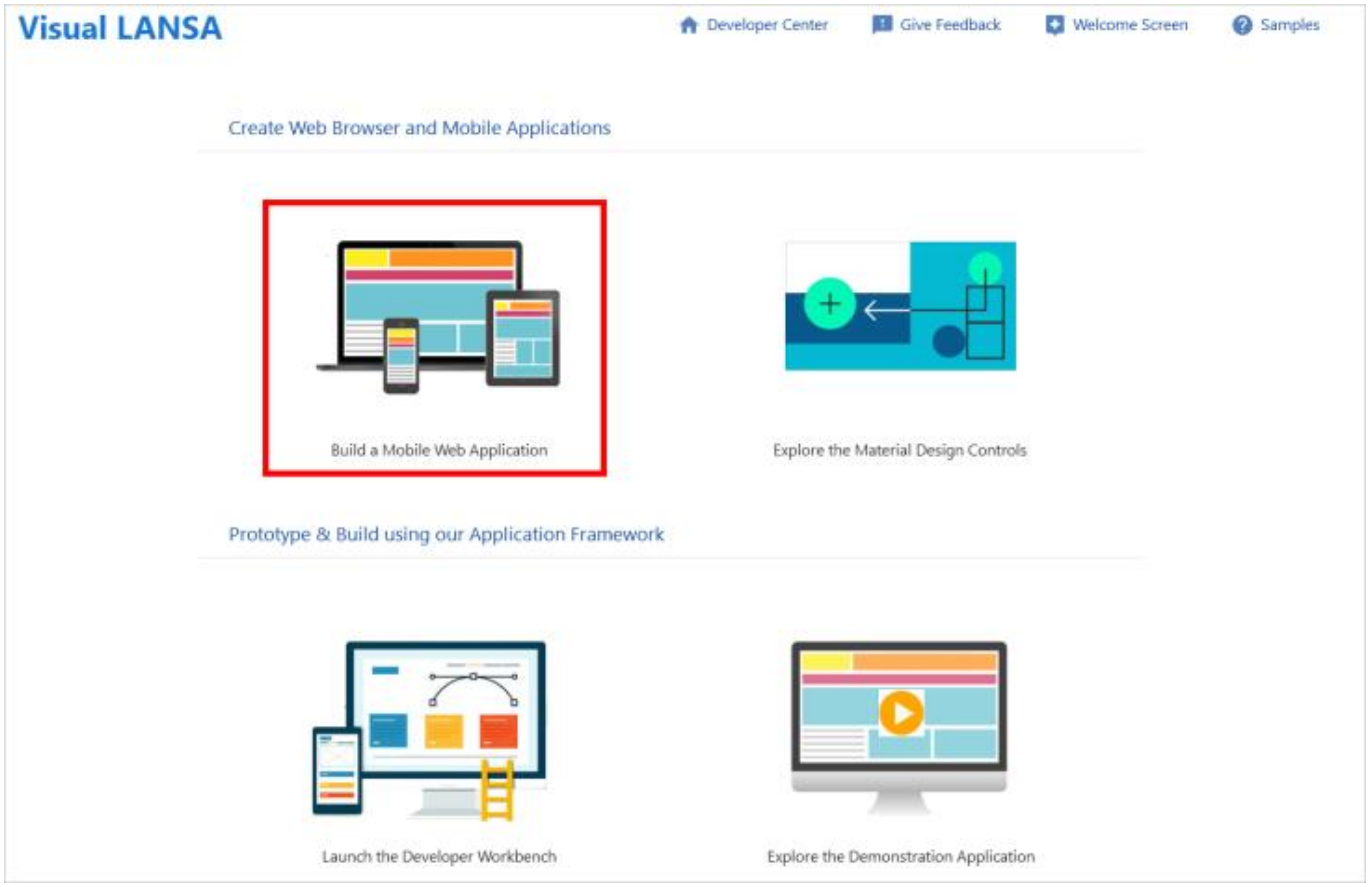
New in LANSA for Web

New Web and Mobile Templates

The easiest way to start creating responsive mobile applications is to use a template. In LANSA Version 15, new templates have been added to show, for example, the use of charts, standard prebuilt contact page, embedded webpages and maps.

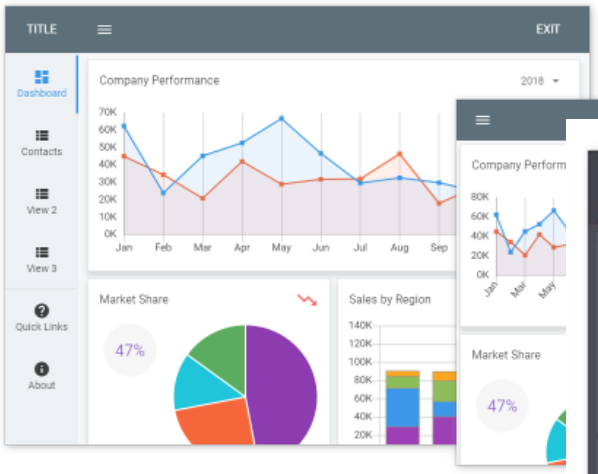
Using Templates

In the IDE, choose the Build a Mobile Web Application on the Home page of the Visual LANSA IDE to use templates:



Template for Dashboard App

The dashboard application template – with light or dark background – creates an application with a dashboard type main web page, as well as a standard Contact Us page, a sample Contacts list, some example charts, and a view with example links.

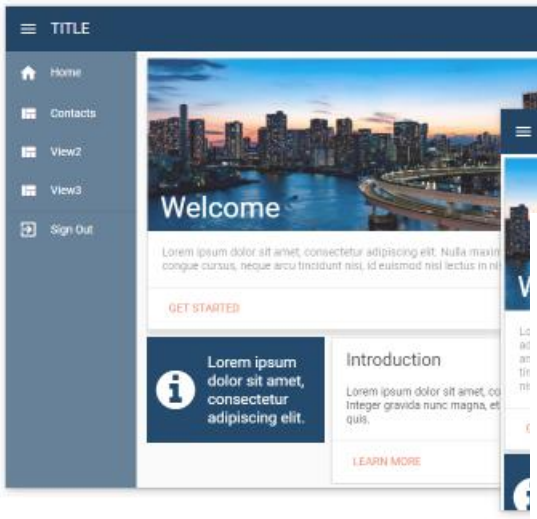


Dashboard - Light

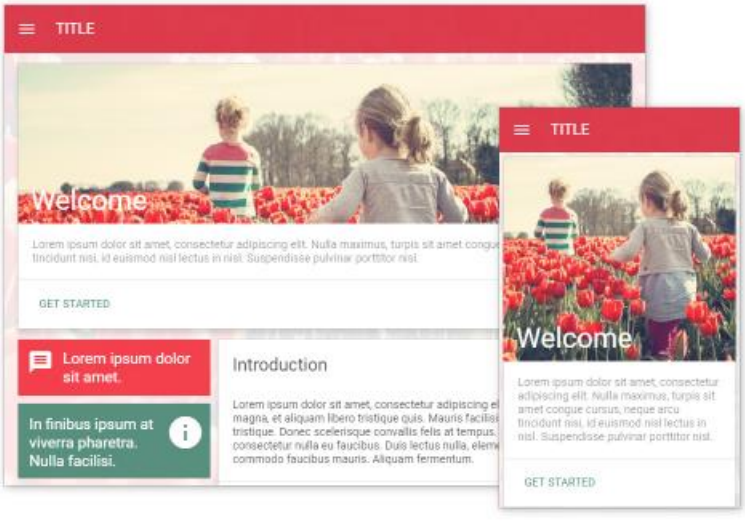
Dashboard - Dark

Template for an App with a Home Screen

The home screen application templates – with two different themes – creates an application with a Home web page and a Contact Us, a contacts list, a chart, and a links view.



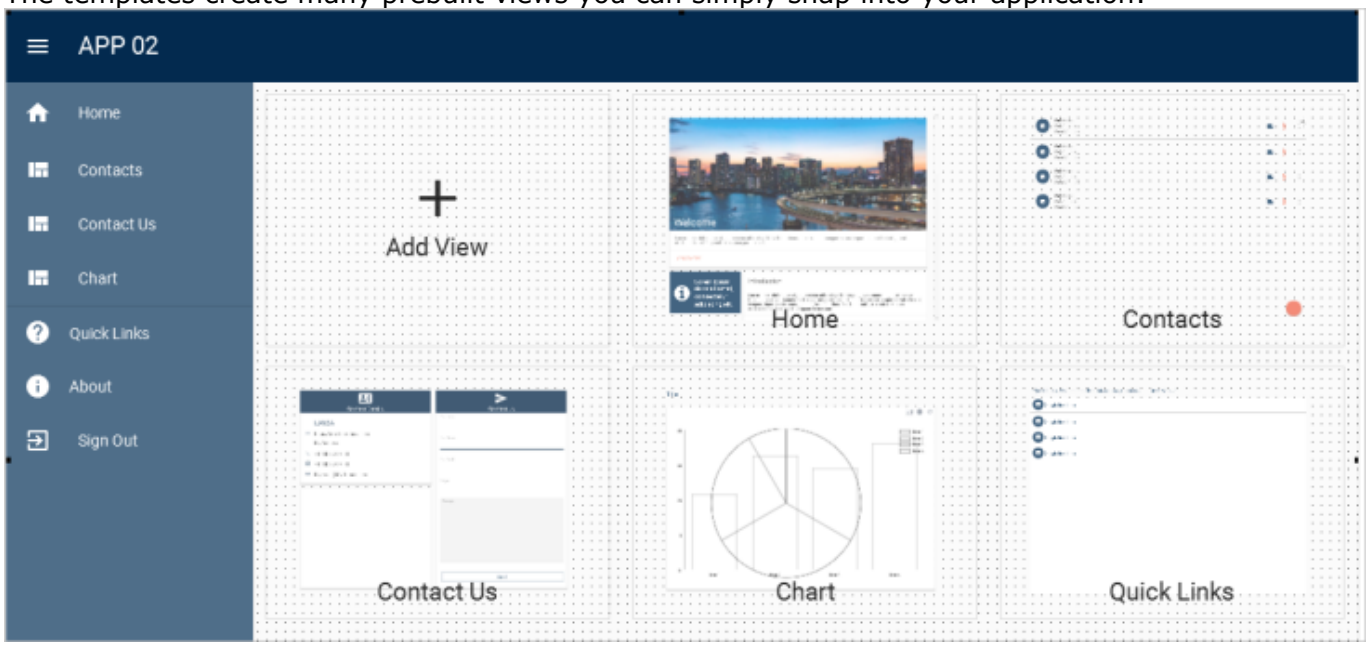
Welcome - Skyline Theme



Welcome - Tulips Theme

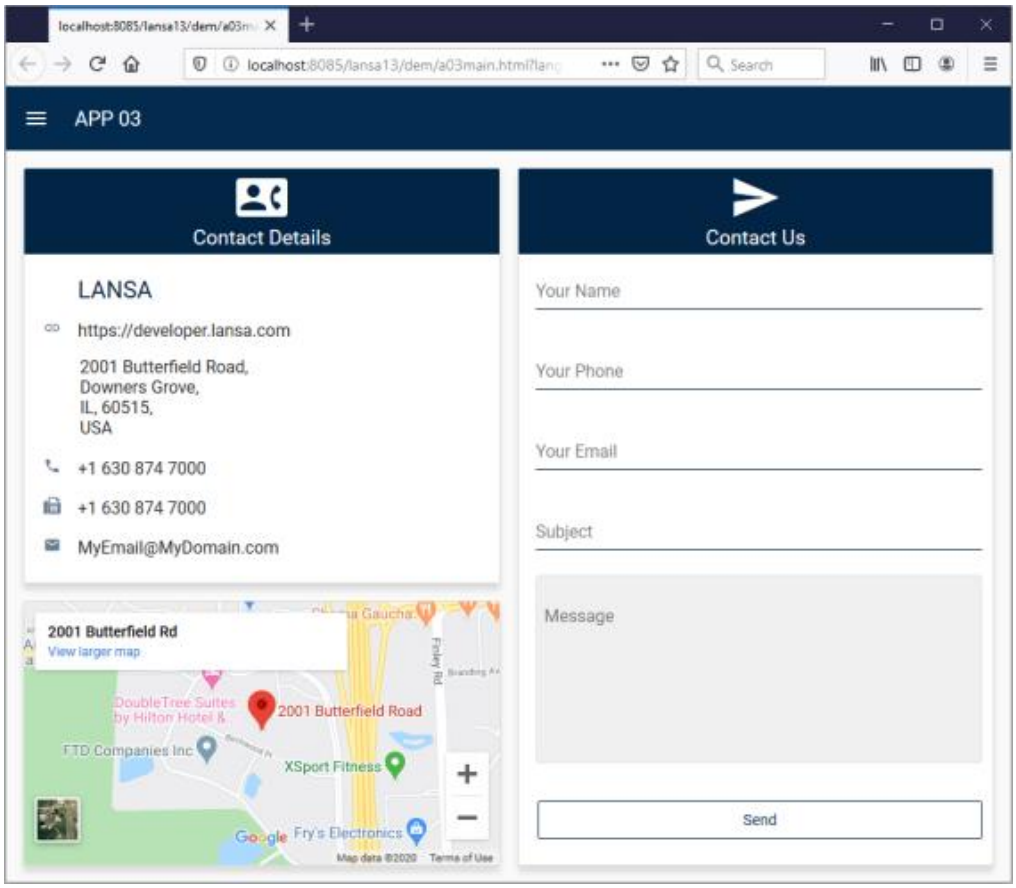
Ready-to-Use Views

The templates create many prebuilt views you can simply snap into your application:



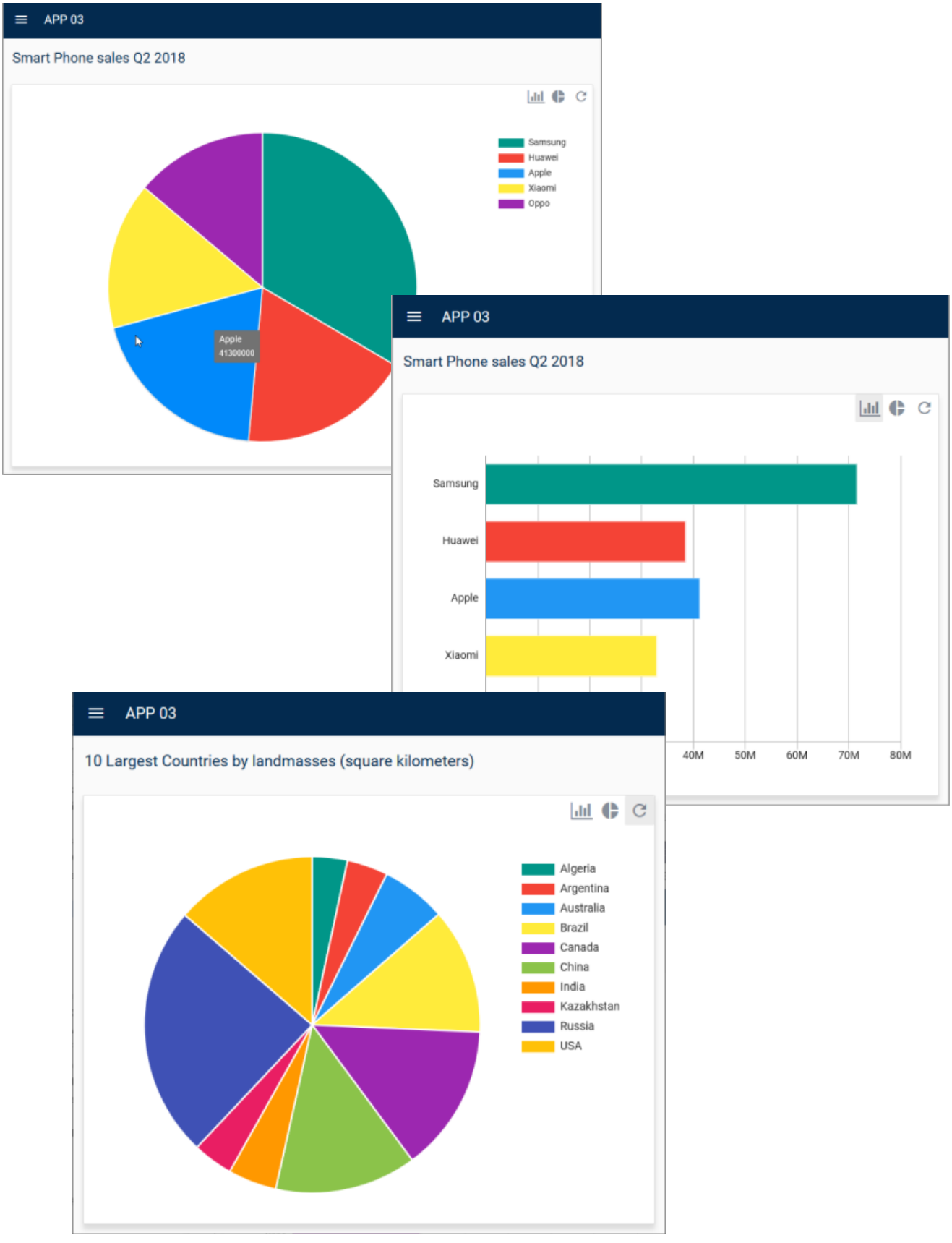
Contact us

The Contact Us view shows contact details and the address of the organization pinned on the map and provides SMS and email capabilities.



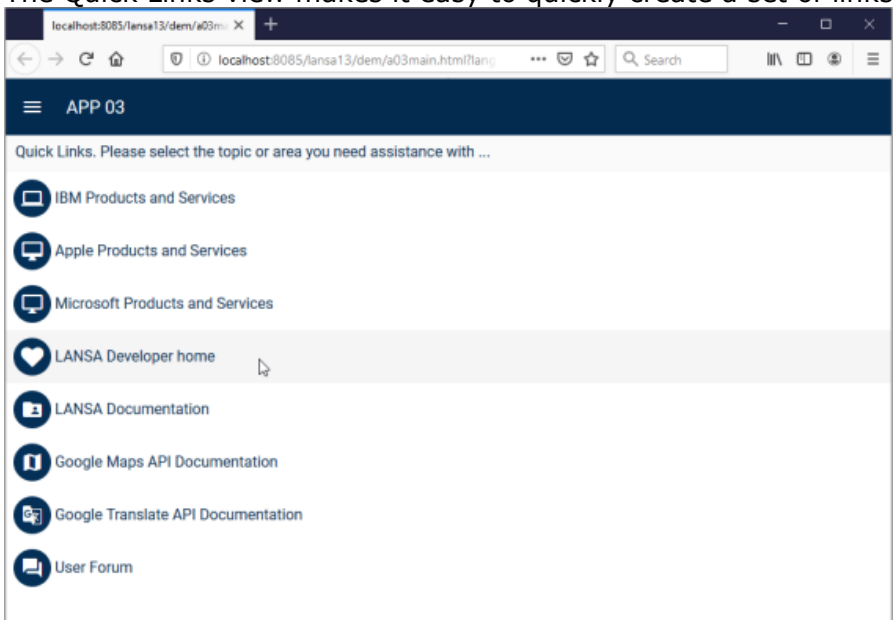
Charts

The Charts view offers a choice of charts which are connected to server-side data.



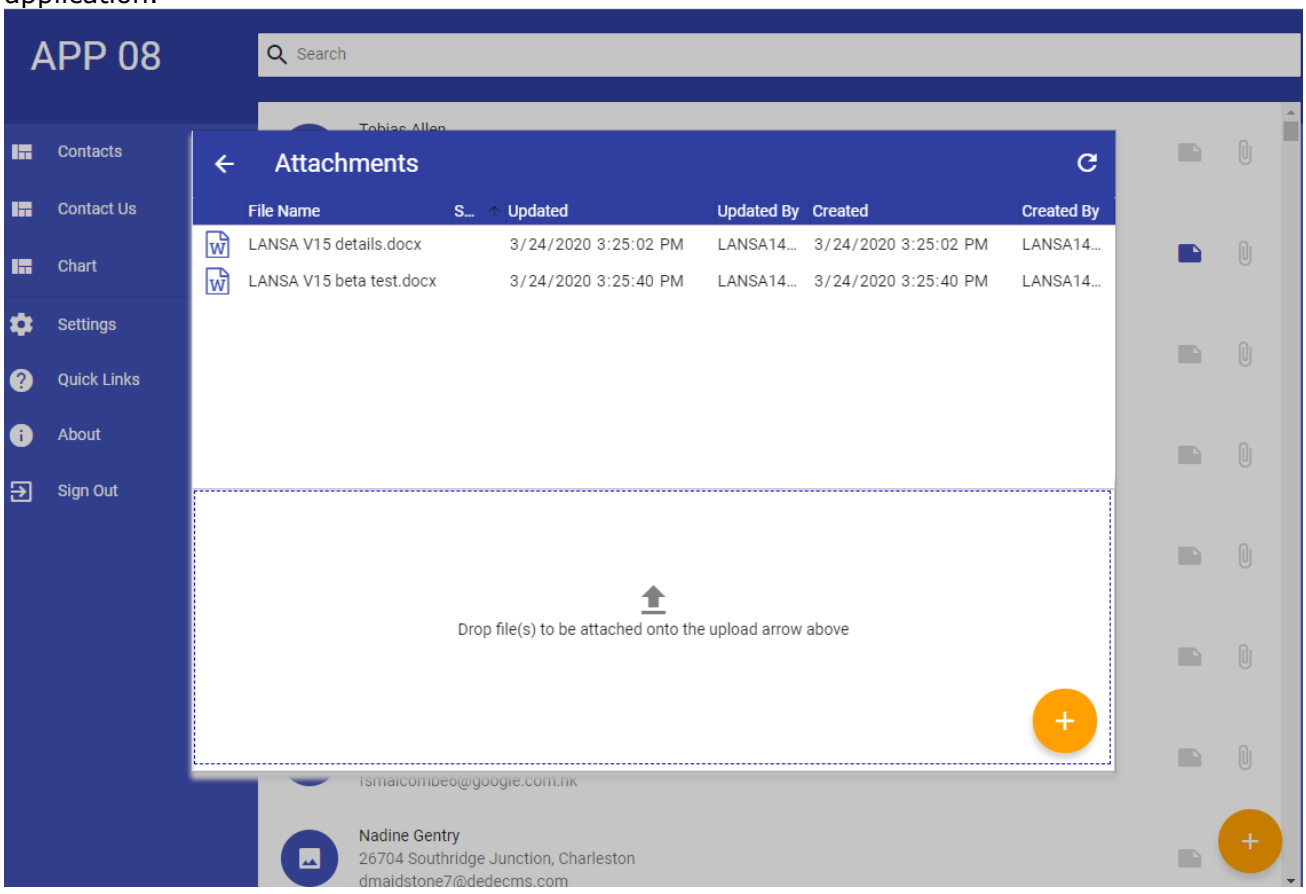
Quick Links

The Quick Links view makes it easy to quickly create a set of links to external resources.



Attachments

The Attachment view is used to easily associate multiple attachments and notes with any application.

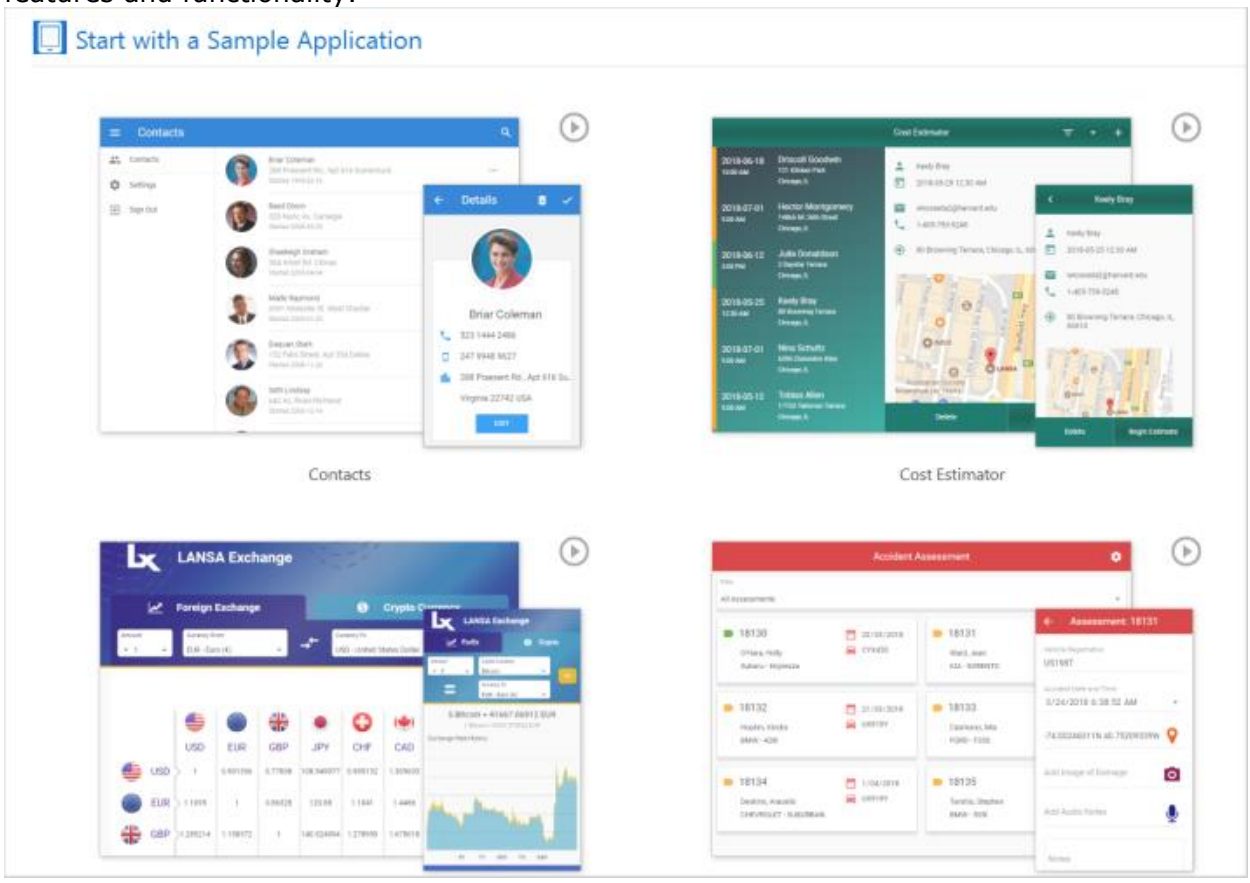


About

An About view shows copyright and details about your application.

Sample Application Enhancements

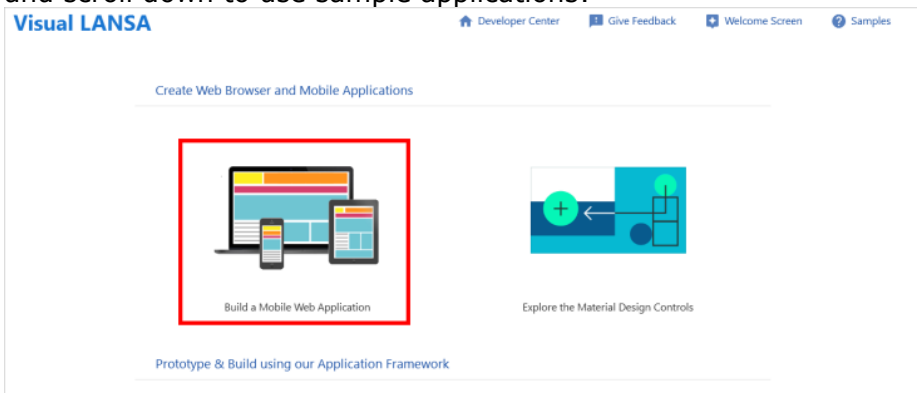
The sample applications shipped with Visual LANSA have been enhanced to showcase the following features and functionality.



Using Sample Apps

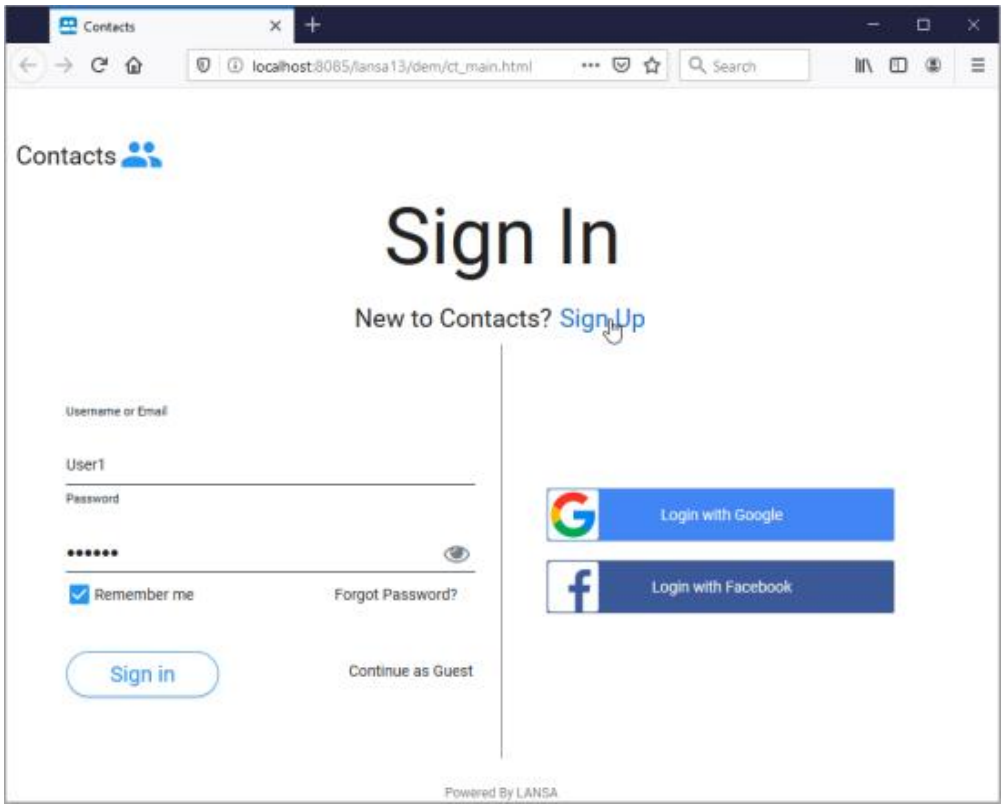
Sample apps are available in the LANSA IDE and some of them are also available online in the LANSA Developer Center.

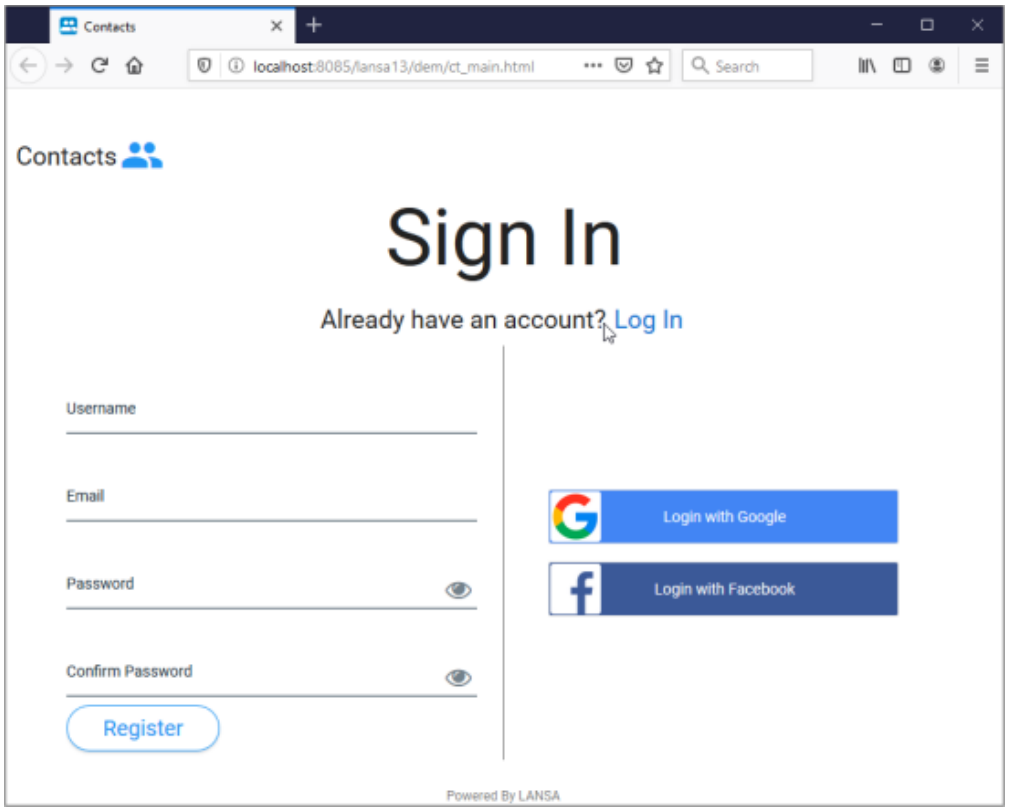
In the IDE, choose the Build a Mobile Web Application on the Home page of the Visual LANSA IDE, and scroll down to use sample applications:



Sign In Dialog

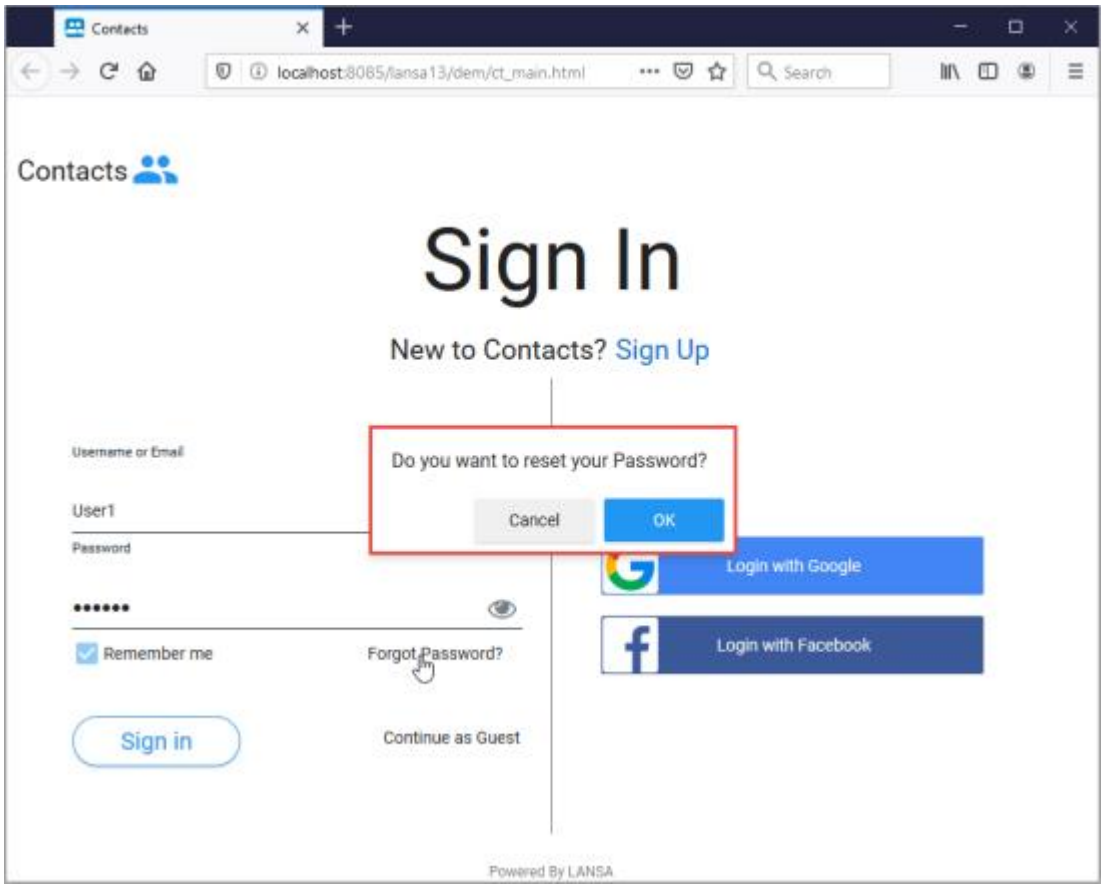
The standard Sign In dialog is used for local user management including registering new users, logging in and maintaining user profiles.





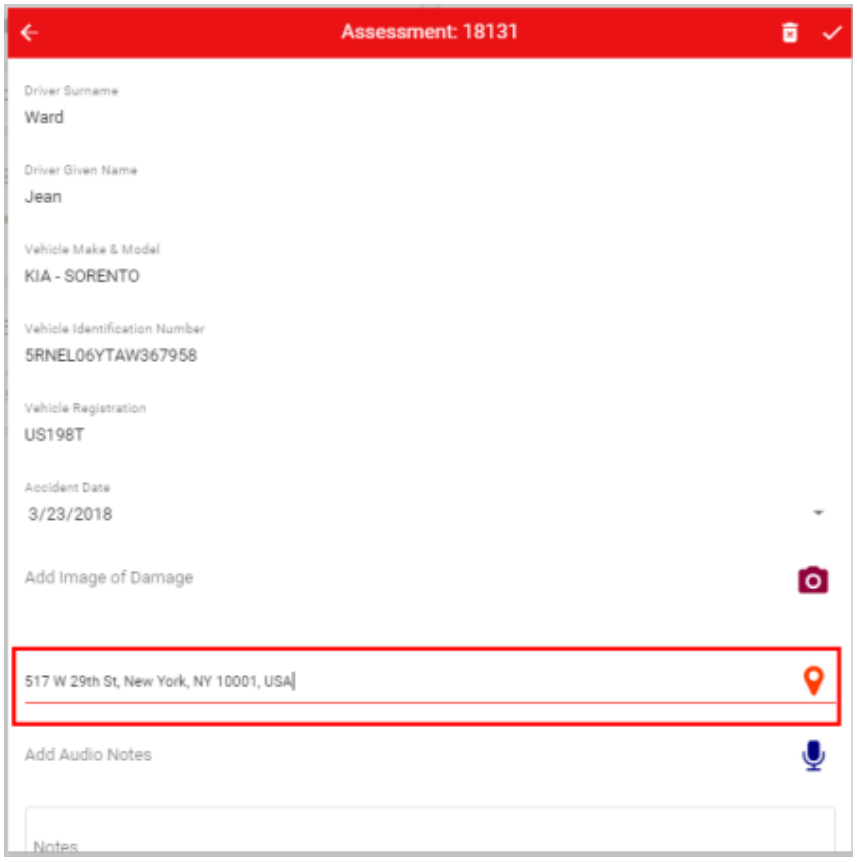
Password Reset

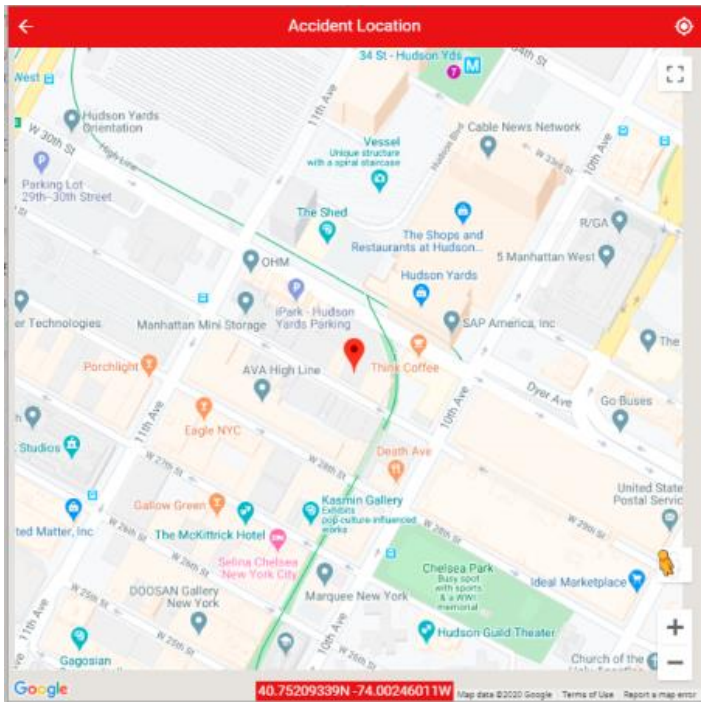
The Sign In dialog uses Sendgrid to reset passwords using email.



Addresses as Google Maps Pins

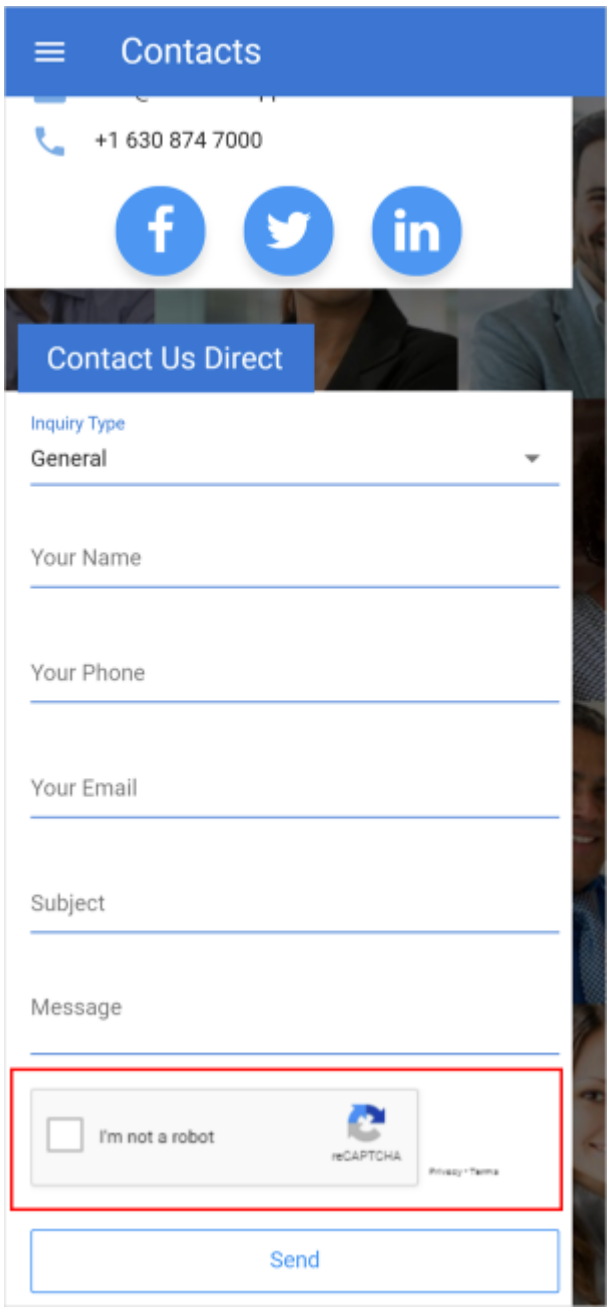
The Accident Assessor sample application shows how to use Google geo-encoding and decoding to obtain geographic coordinates (latitude and longitude) of address and location pin information on a Google map, or, conversely, how to use address details to decode the latitude and longitude information from it.





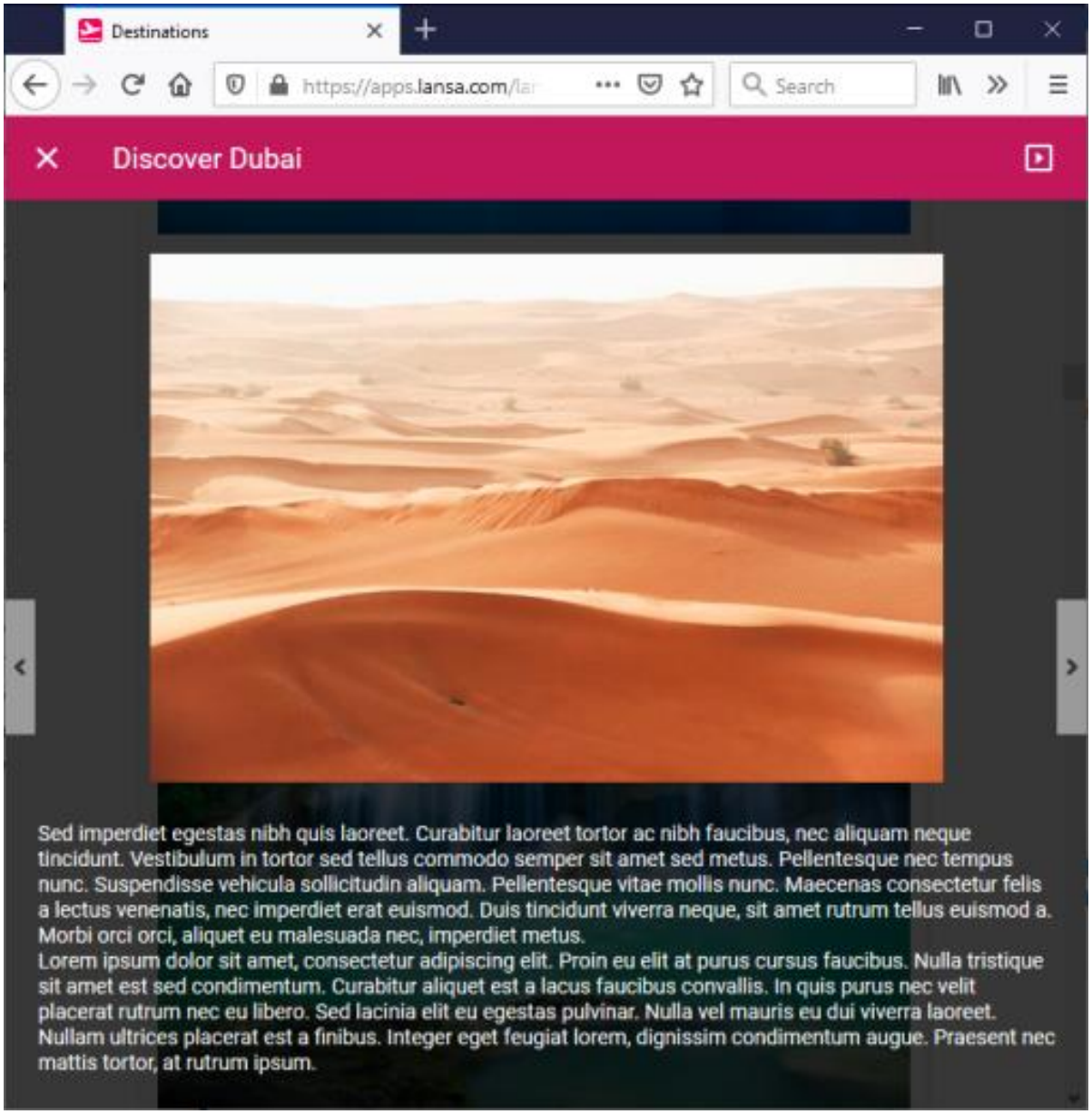
reCAPTCHA to Stop Spamming

The Contacts sample application shows how to use Google's reCAPTCHA service (*I'm not a robot*) to verify and stop apps from being spammed.



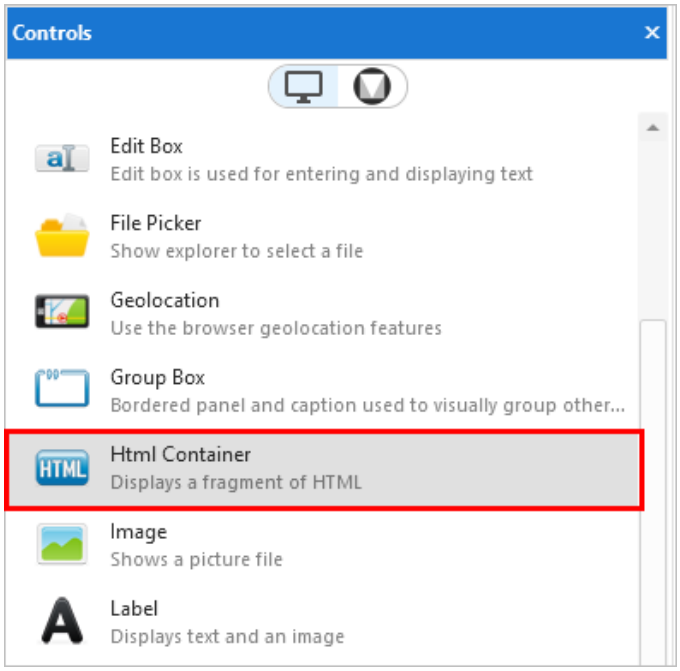
Carousel of Multiple Images with Navigation

The Destinations sample application showcases an image carousel with forward and backward buttons for navigation.



Embedded Webpages

You can use the HTML container to embed any external web content to your web application components or reusable parts.



The HTML container allows raw HTML to be embedded in a page. This is useful you want to use a particular feature of HTML which cannot easily be emulated in LANSA.

Adding an HTML container on a web page or reusable part will result in a DIV element in the generated HTML. Any HTML you provide will be encapsulated in this DIV. The container can be resized and positioned by layout managers as appropriate.

In this simple example an HTML container shows a heading:

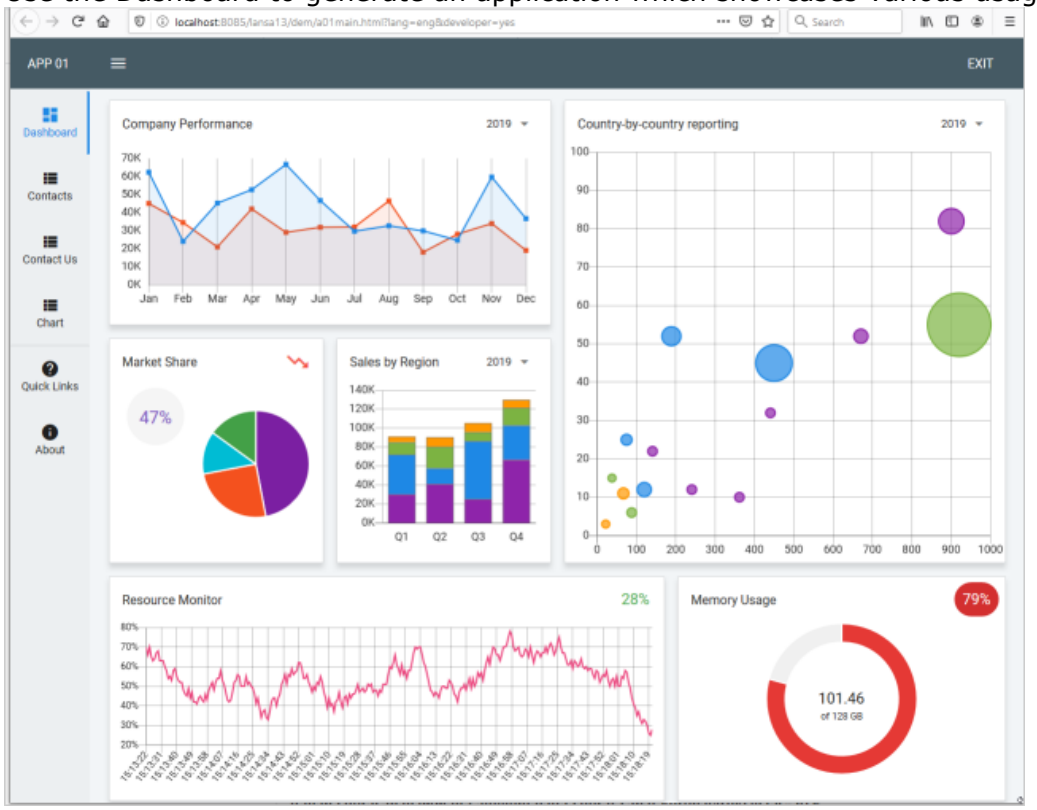
```
Define_Com Class(#PRIM_WEB.HtmlContainer) Name(#Html1) Parent(#COM_OWNER)
Evtroutine Handling(#Com_owner.Initialize)
   #Html1.html := '<h1>My heading</h1>'
Endroutine
```

Note that using raw HTML introduces browser dependencies and browser specific functionality.

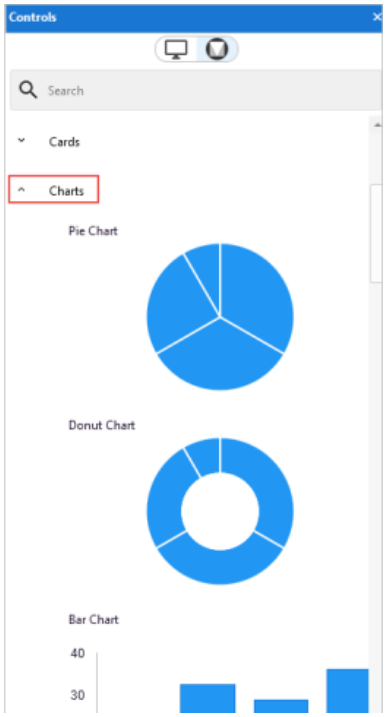
Charts as Primitive Controls

LANSA chart controls are now available for web applications. These powerful native controls replace the Google chart widgets which were available in previous LANSA versions.

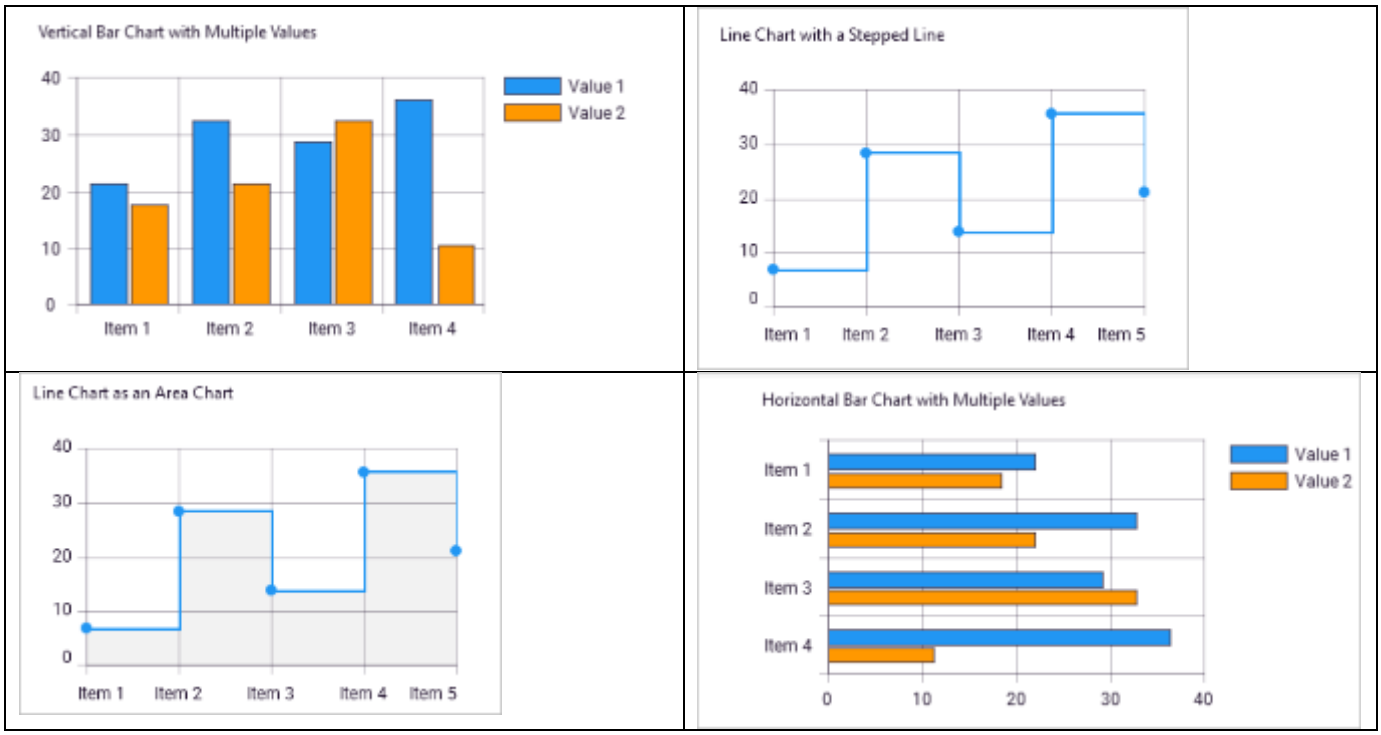
Use the Dashboard to generate an application which showcases various usages for charts:



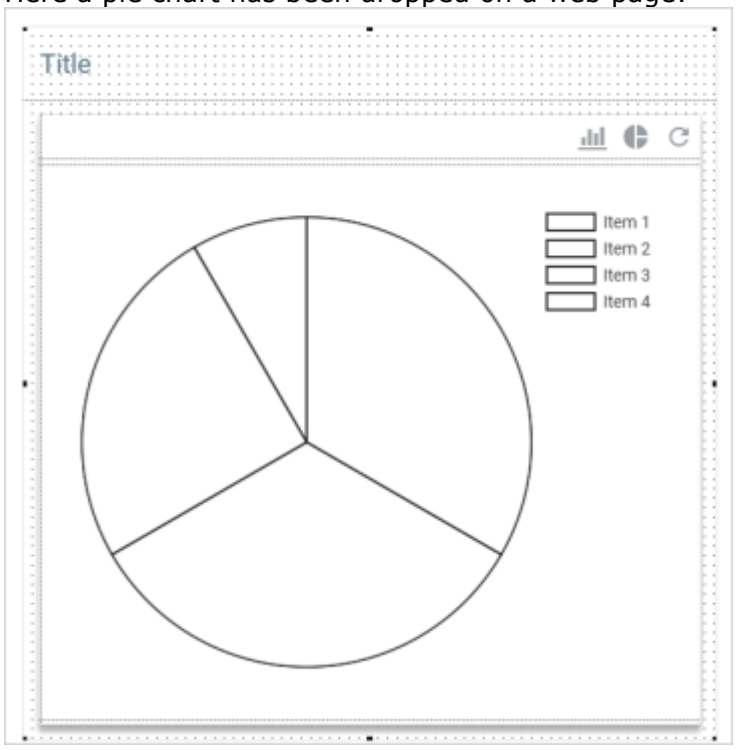
Use the Controls tab to add charts to your component:



These are the available chart styles:



To add a chart to your web page or view, simply drag and drop it from the Controls tab.
Here a pie chart has been dropped on a web page:

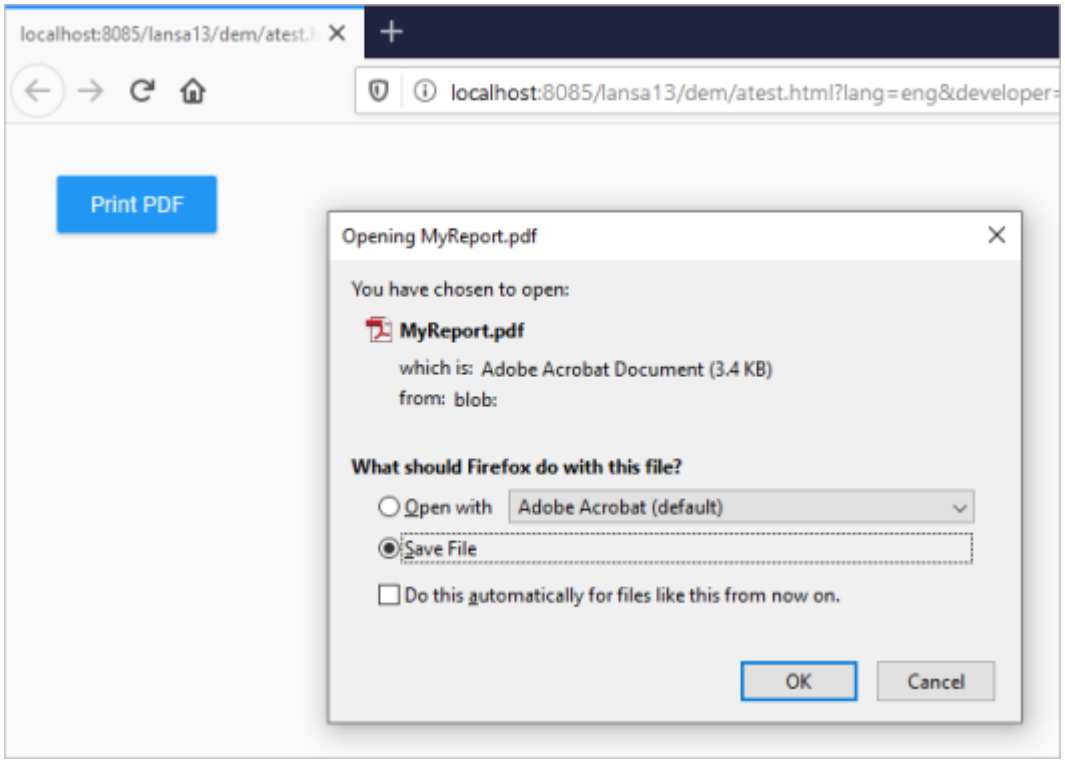


Here is the chart's definition in the Source tab:

```
* Define_Com Class(#PRIM_CHRT.BarChart) Name(#BarChart) Displayposition(3) Gridlines(Horizontal)
Height(398) Left(33) Parent(#Panel) Tabposition(3) Tabstop(False) Top(73) Width(486)
Visible(False)
* Define_Com Class(#PRIM_CHRT.BarChartCaption) Name(#BarChartCaption) Parent(#BarChart)
* Define_Com Class(#PRIM_CHRT.BarChartValue) Name(#BarChartValue) Displayposition(1)
Parent(#BarChart)
```

PDF Reports from Web Pages

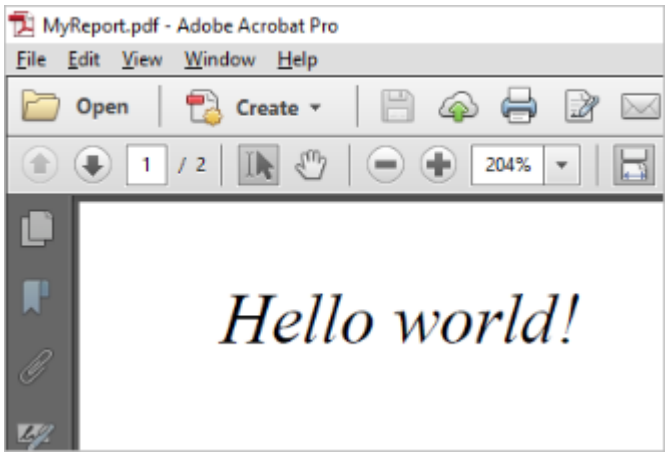
The new PDF component can be used to create PDF files from a web page.



This sample web page code produces a PDF file with the text 'Hello World'':

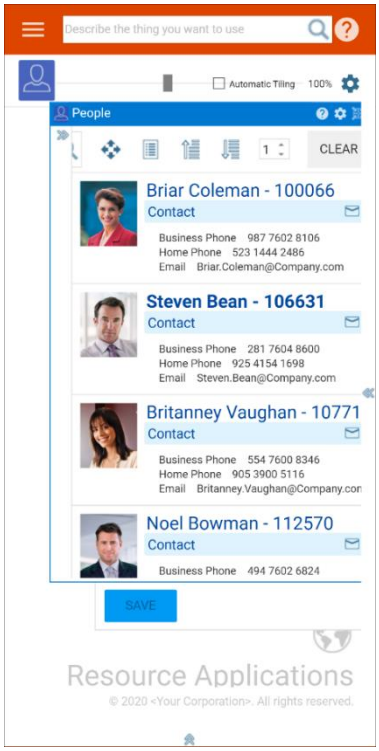
```
Begin_Com Role(*EXTENDS #PRIM_WEB) Theme(#SYS_THEME<MaterialDesignBlue>)

* Simple PDF report
Define_Com Class(#PRIM_PDF) Name(#pdf)
Define_Com Class(#PRIM_MD.RaisedButton) Name(#Button) Caption('Print PDF') Displayposition(1) Left(32)
Parent(#COM_OWNER) Tabposition(1) Themedrawstyle('MediumTitle') Top(32)
Define Field(#CreationDate) Type(*DATETIME)

Evtroutine Handling(#Button.Click)
#pdf.start

#pdf.setDocumentProperties Title('My PDF report') Author('John Smith') Creator('My Company, Inc')
Keywords('Weekly report')
#CreationDate := #CreationDate.Now
#pdf.setCreationDate Datetime(#CreationDate)

#pdf.setFont Font('Times') Style(Italic)
#pdf.setFontSize Size(14)
#pdf.text Text('Hello world!') Left(10) Top(10)
#pdf.addPage
#pdf.text Text('This line is on the second page') Left(10) Top(10)

#pdf.save Filename('MyReport.pdf')

Endroutine
End_Com
```

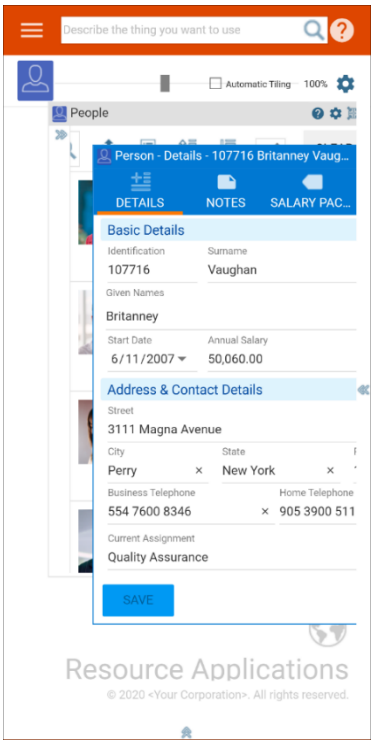
The result PDF:



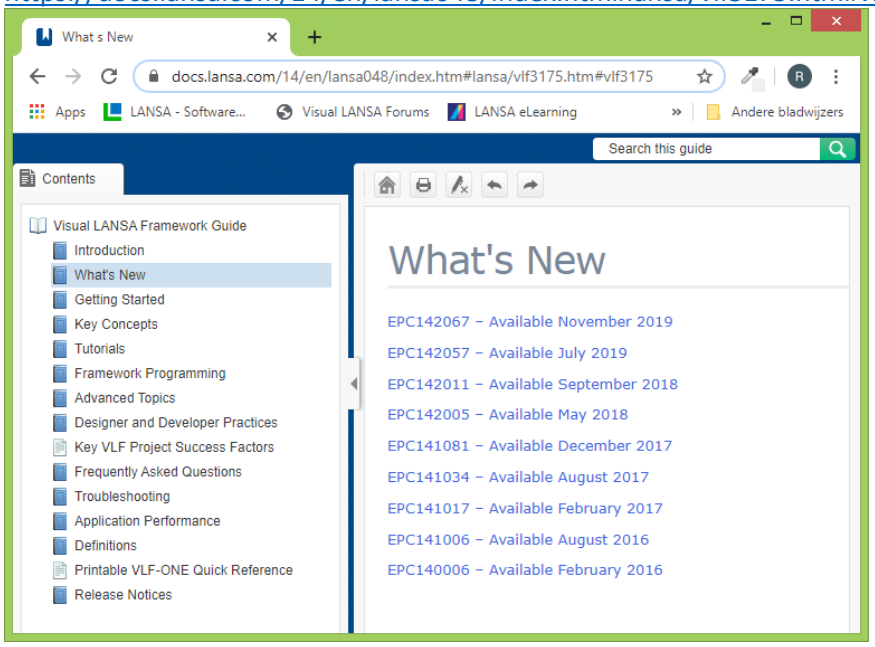
New in Visual LANSA Framework

See What's New in Visual LANSA Framework in EPC142011, EPC142057 and EPC142067:





https://docs.lansa.com/14/en/lansa048/index.htm#lansa/vlf3175.htm#vlf3175



Details can also be found in the July 2019 Technical newsletter:
https://lansa.com/support/newsletter.html

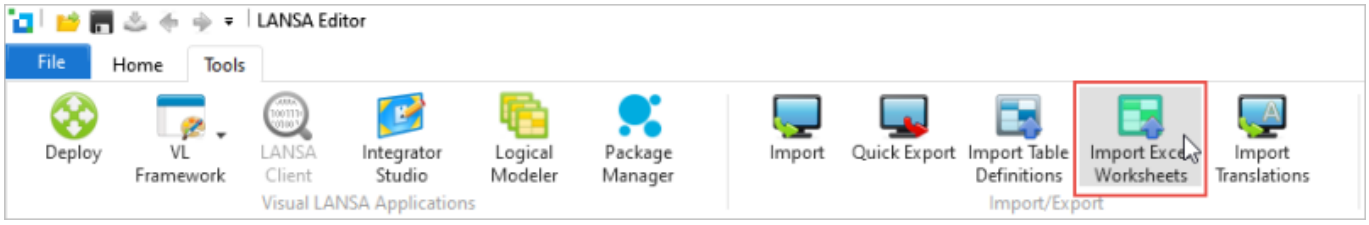
Visual LANSA IDE Enhancements

Import Excel Worksheets

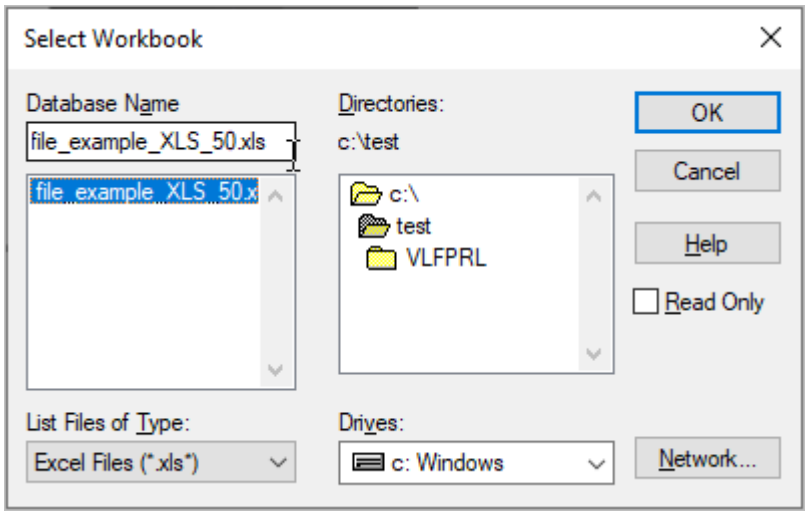
An option has been added to the ribbon to support loading worksheets from an Excel Workbook into LANSA tables.

It is only available on machines that have the appropriate driver to support for Excel 12.0 installed. A message will be issued if you select this option without the appropriate driver installed.

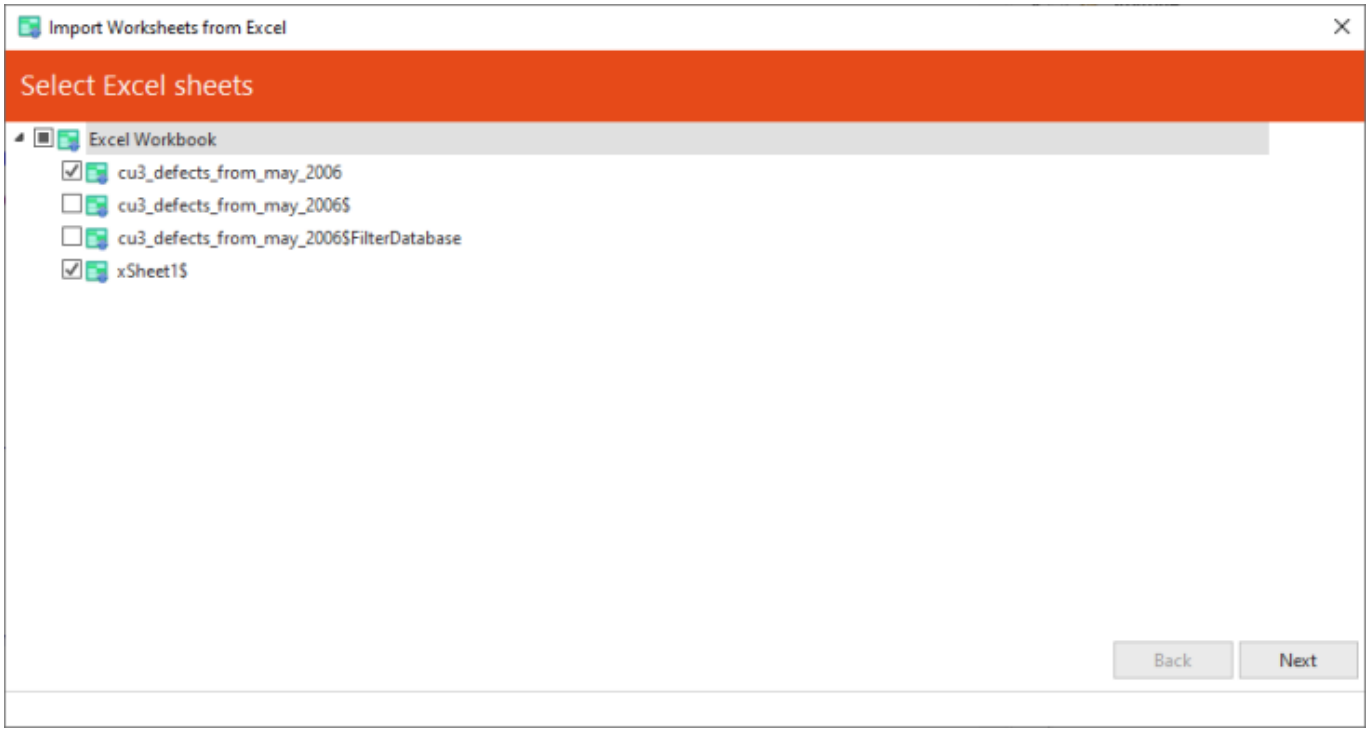
- To import a worksheet, select Import Excel Worksheets from the Tools tab on the ribbon:



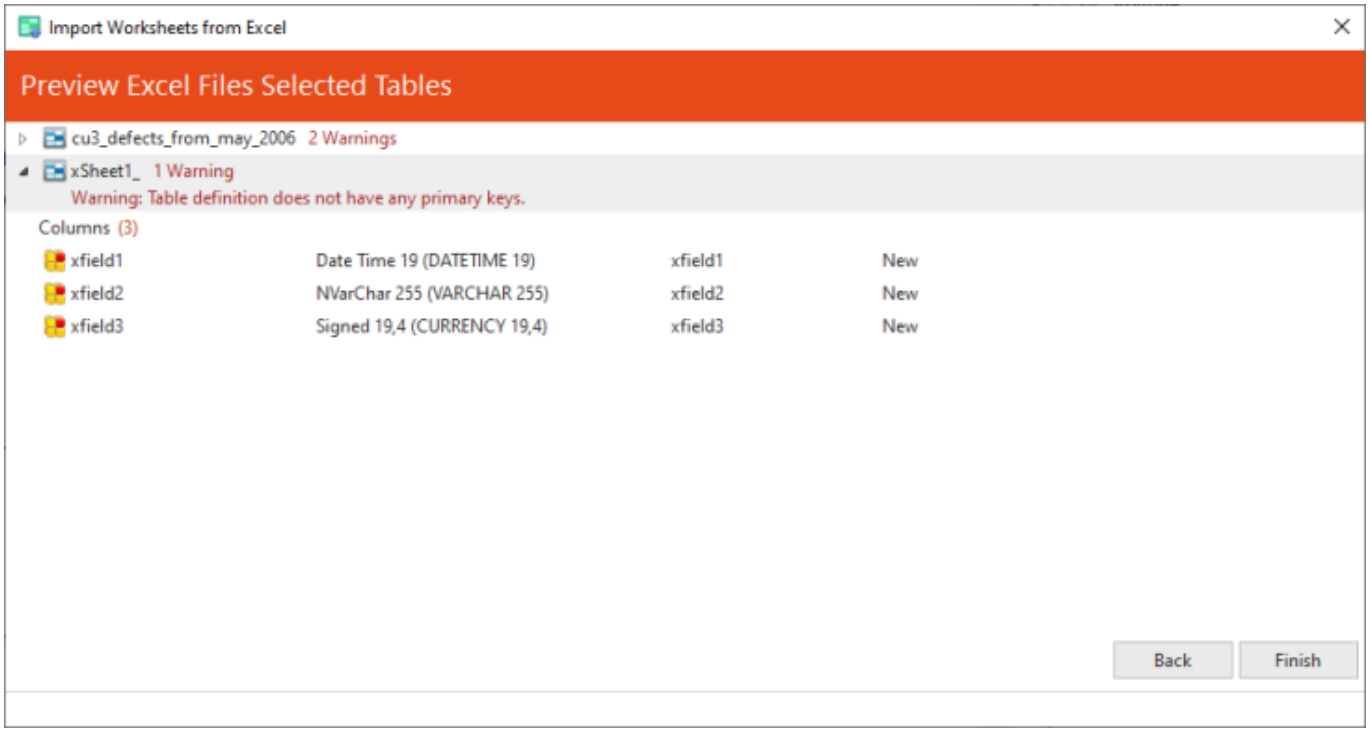
- Choose the file to be imported:



- Select the worksheets to be loaded as LANSA tables:



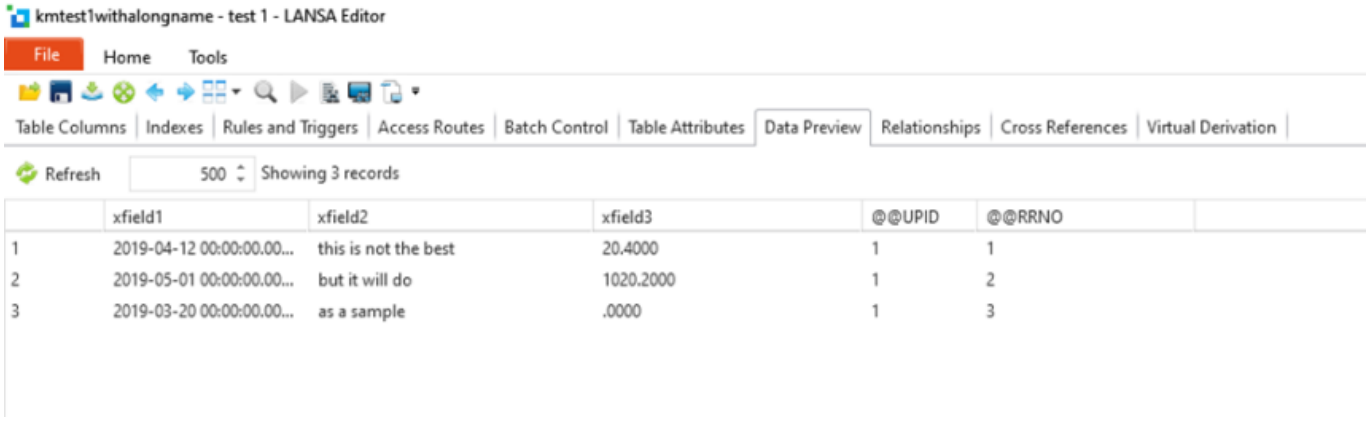
- Verify the selected work sheets and columns which will be generated:



Each work sheet generates a LANSA table, the name/identifier/description of the table needs to be provided. You can select to include all or a subset of the columns from the original worksheet definition.

After the tables are generated and compiled, data will be copied into LANSA tables corresponding with the data from the Excel worksheet.

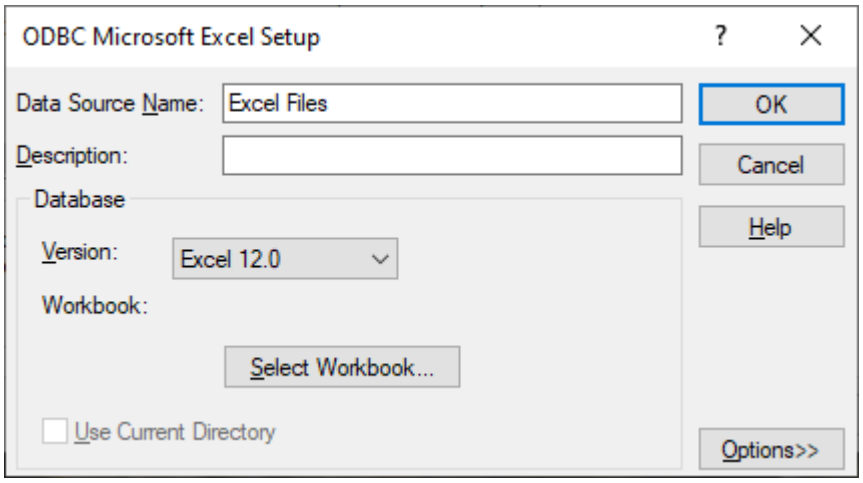
- Verify the table(s) have been created with expected columns and data copied into them:



The temporary tables created during the import are automatically removed upon completion of the processing.

Installing the Driver

This feature is only available if the host machine has the appropriate driver to support for Excel 12.0 installed (by default this driver installs a DSN called "Excel Files").

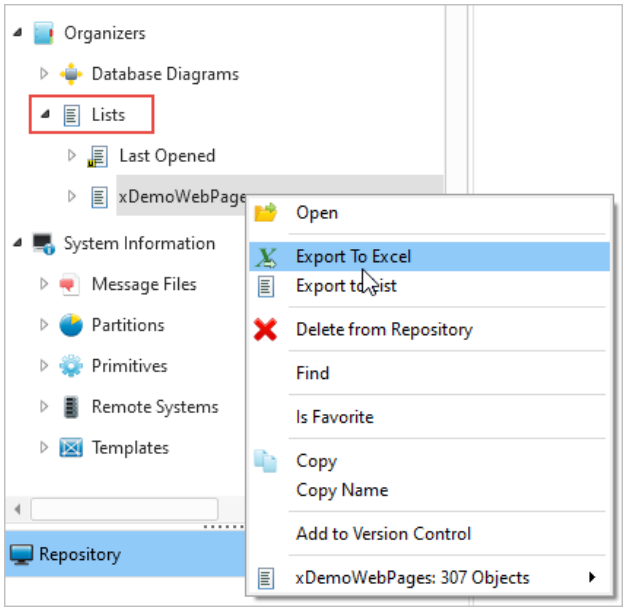


If the driver is not installed, you can download it at: https://www.microsoft.com/en-au/download/details.aspx?id=13255 (right-click the link and choose to open the link in a new tab or window).

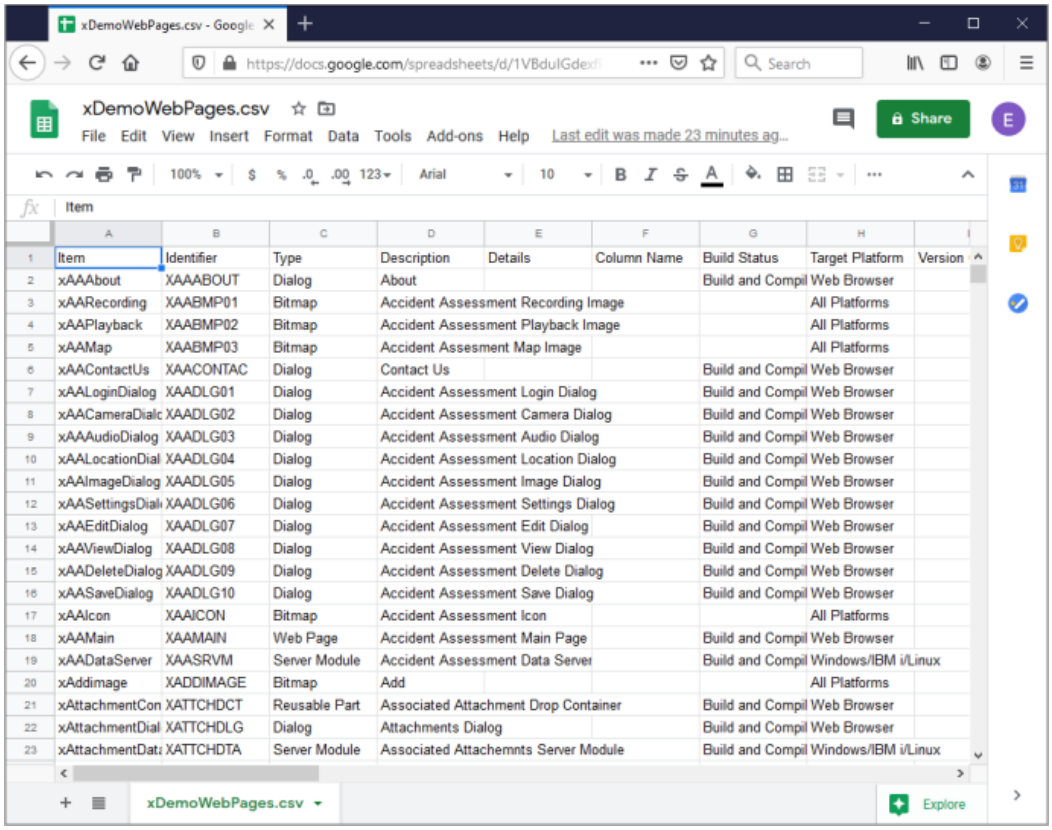
Export a List to Excel

You can export the contents of a list to a .csv file.

- Right-click a list to display the context menu
- Select *Export To Excel*



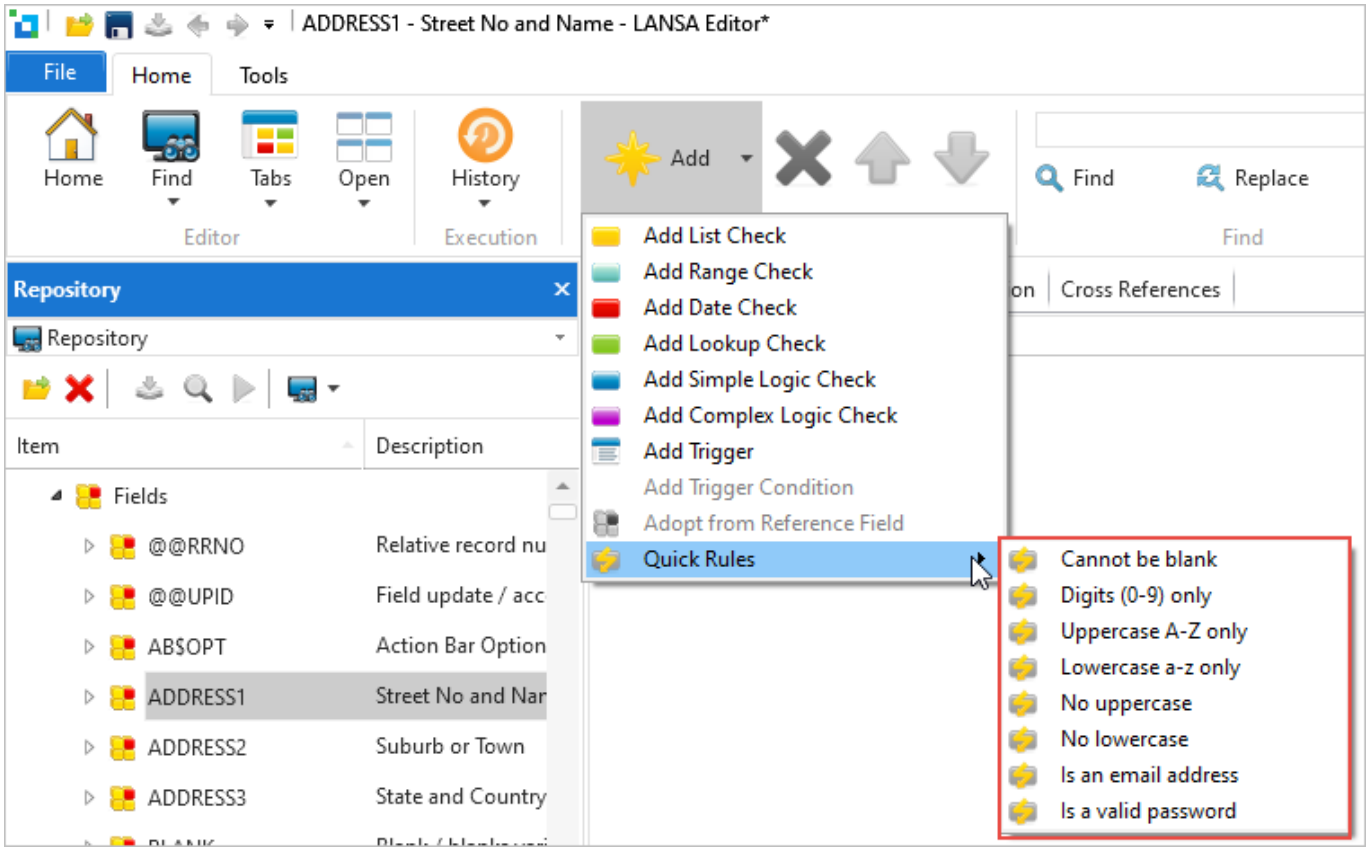
- Specify where you want the file saved and open the .csv file



Item	Identifier	Type	Description	Details	Column Name	Build Status	Target Platform	Version
xAAAbout	XAAABOUT	Dialog	About			Build and Compil	Web Browser	
xAARecording	XAABMP01	Bitmap	Accident Assessment Recording Image				All Platforms	
xAAPlayback	XAABMP02	Bitmap	Accident Assessment Playback Image				All Platforms	
xAAMap	XAABMP03	Bitmap	Accident Assesment Map Image				All Platforms	
xAAContactUs	XAACONTAC	Dialog	Contact Us			Build and Compil	Web Browser	
xAALoginDialog	XAADLG01	Dialog	Accident Assessment Login Dialog			Build and Compil	Web Browser	
xAACameraDialc	XAADLG02	Dialog	Accident Assessment Camera Dialog			Build and Compil	Web Browser	
xAAAudioDialog	XAADLG03	Dialog	Accident Assessment Audio Dialog			Build and Compil	Web Browser	
xAALocationDial	XAADLG04	Dialog	Accident Assessment Location Dialog			Build and Compil	Web Browser	
xAAImageDialog	XAADLG05	Dialog	Accident Assessment Image Dialog			Build and Compil	Web Browser	
xAASettingsDiali	XAADLG06	Dialog	Accident Assessment Settings Dialog			Build and Compil	Web Browser	
xAAEditDialog	XAADLG07	Dialog	Accident Assessment Edit Dialog			Build and Compil	Web Browser	
xAAViewDialog	XAADLG08	Dialog	Accident Assessment View Dialog			Build and Compil	Web Browser	
xAADeleteDialog	XAADLG09	Dialog	Accident Assessment Delete Dialog			Build and Compil	Web Browser	
xAASaveDialog	XAADLG10	Dialog	Accident Assessment Save Dialog			Build and Compil	Web Browser	
xAAIcon	XAAICON	Bitmap	Accident Assessment Icon				All Platforms	
xAAMain	XAAMAIN	Web Page	Accident Assessment Main Page			Build and Compil	Web Browser	
xAADataServer	XAASRVM	Server Module	Accident Assessment Data Server			Build and Compil	Windows/IBM i/Linux	
xAddimage	XADDIMAGE	Bitmap	Add				All Platforms	
xAttachmentCon	XATTCHDCT	Reusable Part	Associated Attachment Drop Container			Build and Compil	Web Browser	
xAttachmentDial	XATTCHDLG	Dialog	Attachments Dialog			Build and Compil	Web Browser	
xAttachmentDat	XATTCHDTA	Server Module	Associated Attachemnts Server Module			Build and Compil	Windows/IBM i/Linux	

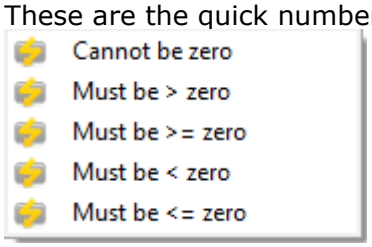
Add Quick Rules to Fields

Predefined Quick Rules which previously were available only for tables are now available for fields. Using quick rule to minimizes the effort in adding common rules.

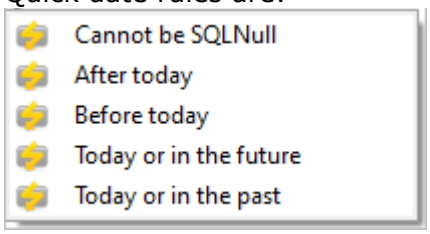


The rules available are based on the type of data. You can see quick rules available for string columns above.

These are the quick number rules:



Quick date rules are:



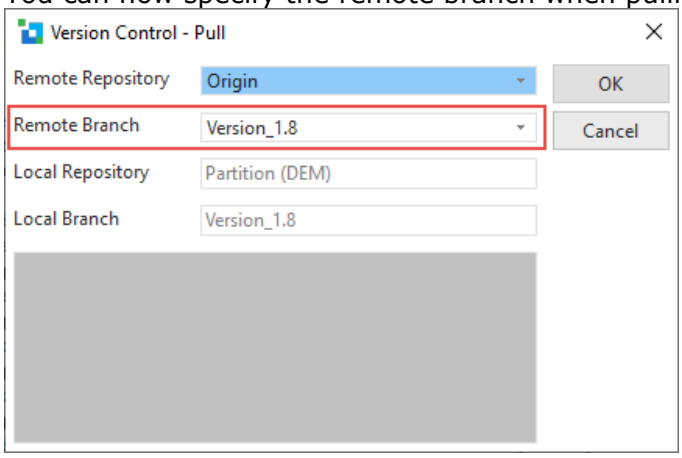
Version Control Git Enhancements

Git functionality in the LANSA IDE has been enhanced and new Git repository actions are now available.

Enhancements

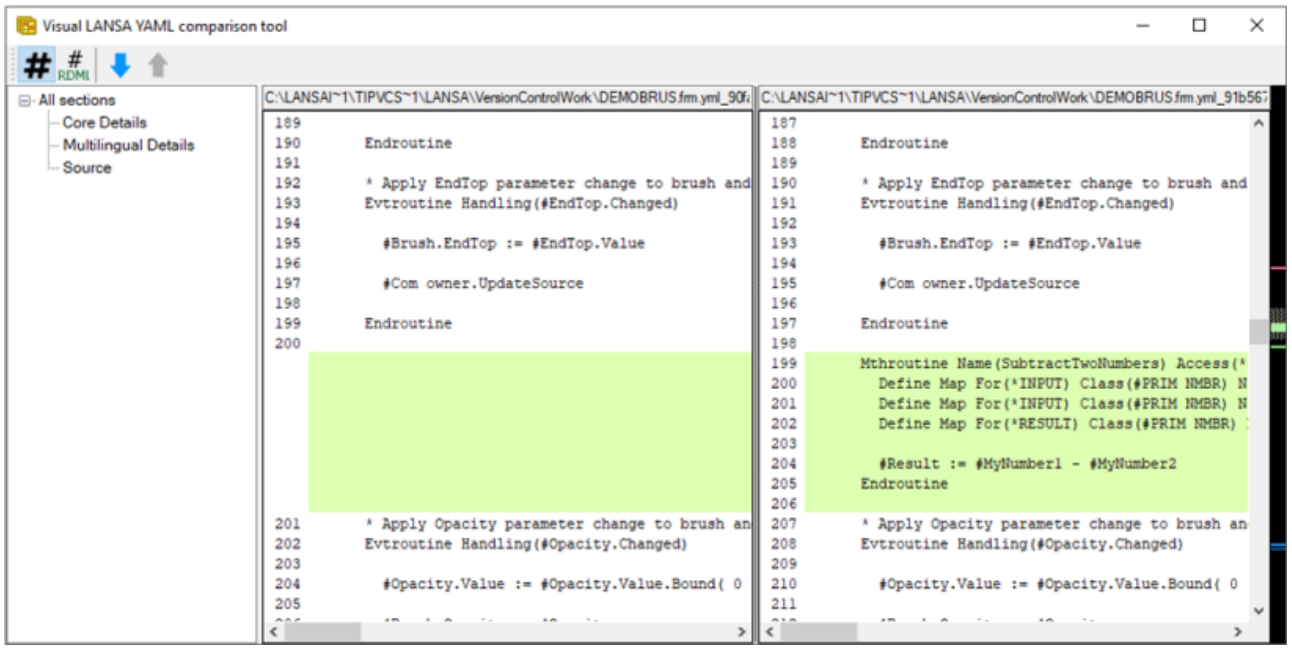
Pull and Push

You can now specify the remote branch when pulling and pushing changes:



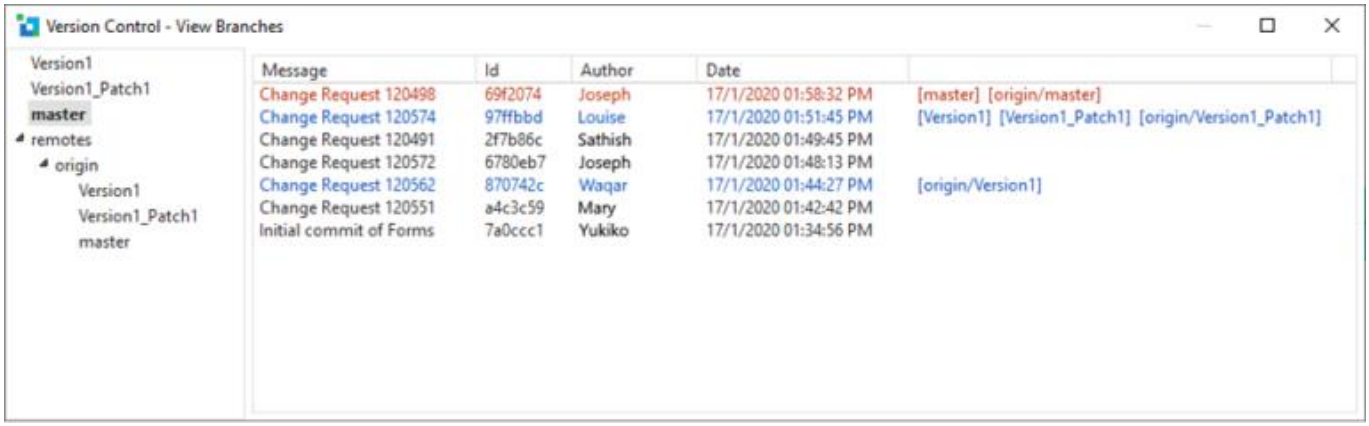
History Capabilities – Individual File

The new Visual LANSA YAML comparison tool shows the differences between two changes:



New Branch Actions

You can use the View Branch action to display all local and remote branches and to view all commits in the current repository.



You can use it to perform the following actions:

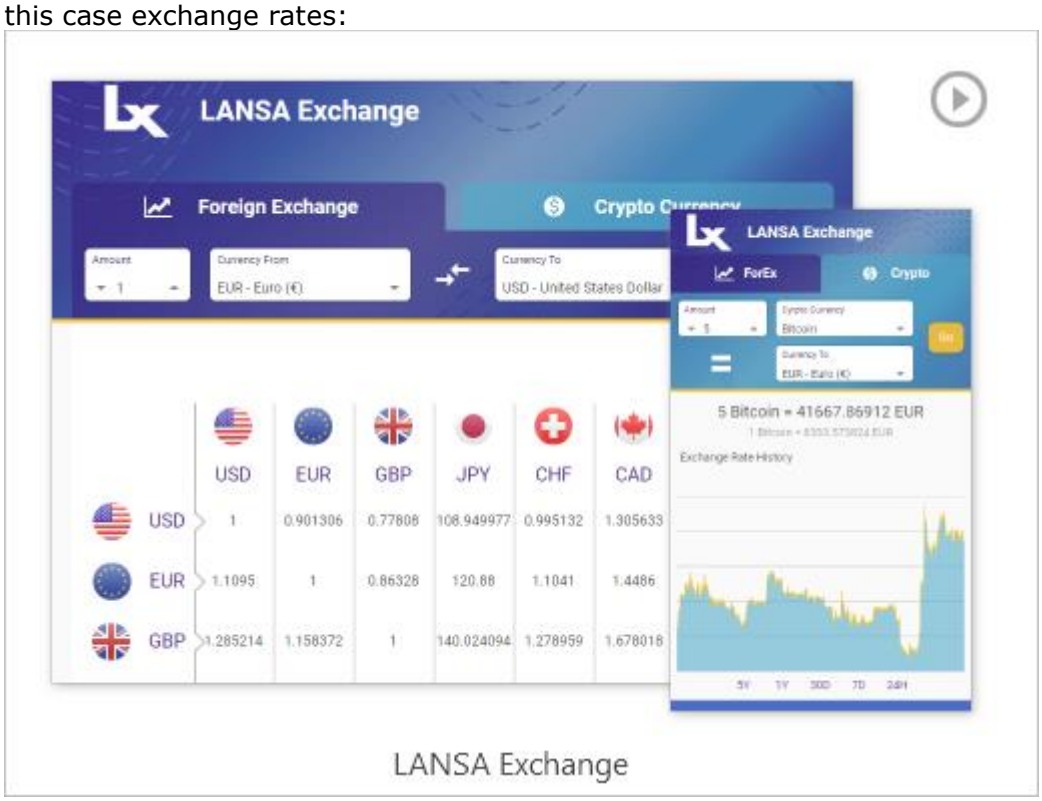
- Create a new branch.
- Checkout a branch.
- Merge branches.
- Rebase branches.

For more information see Merge From and Rebase Onto in the User Guide.

RESTful Web Services

Web Services Sample Application

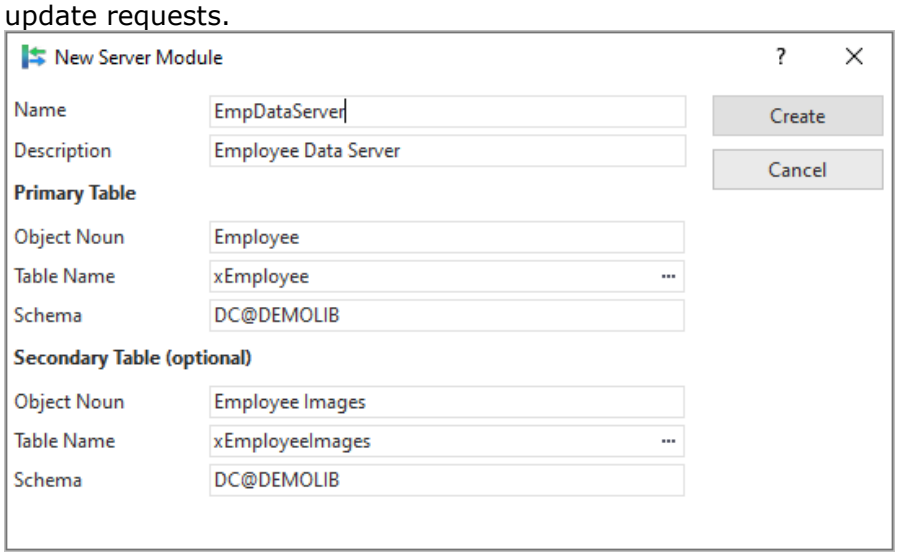
The new LANSA Exchange sample application shows how to use a web service to retrieve data, in this case exchange rates:



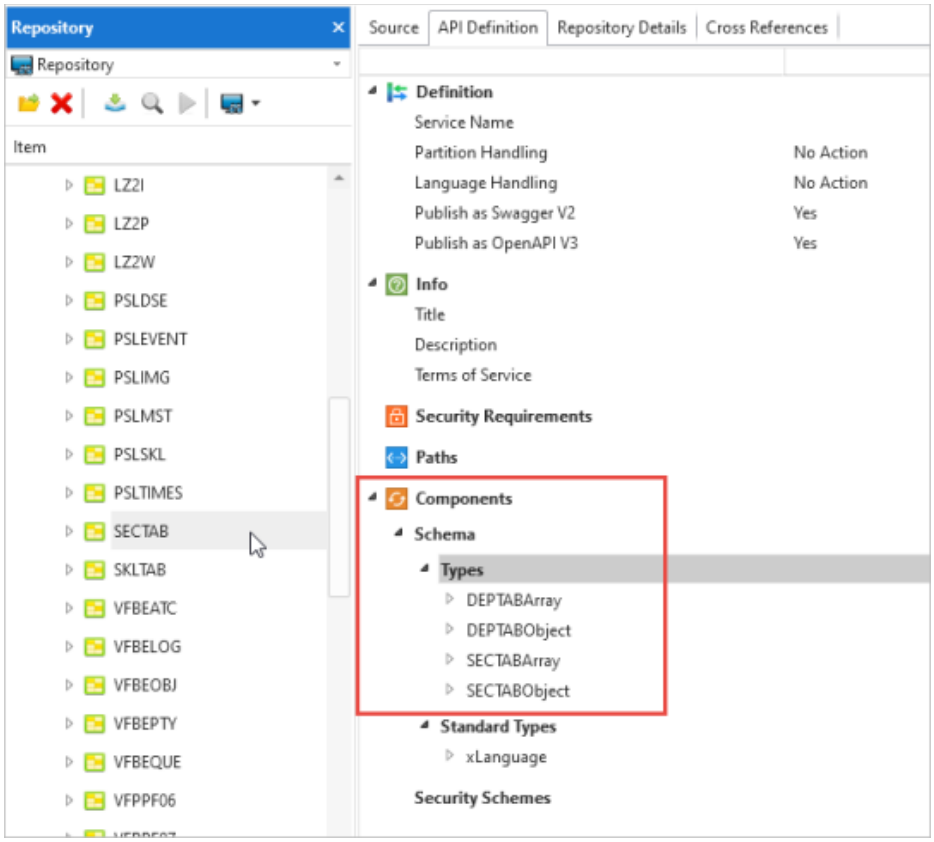
LANSA Exchange

Generate REST APIs

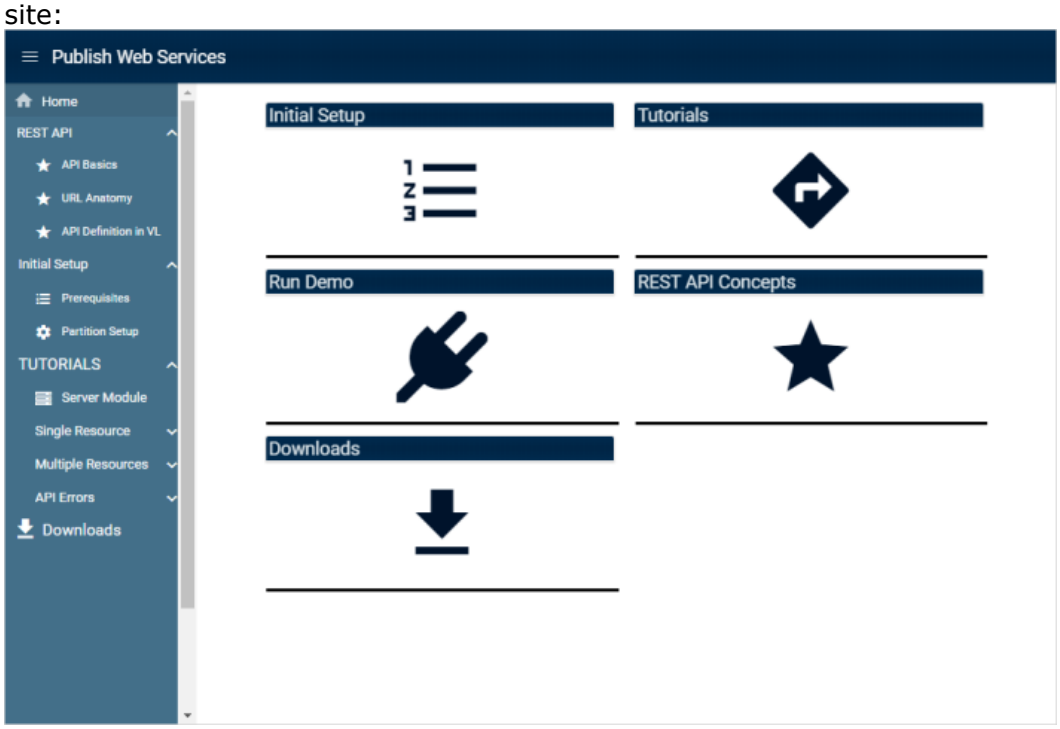
You can now generate server modules which contain RESTful API code based on the tables nominated when creating the server module. APIs are generated to support create, delete, get, and update requests.



You can also automatically generate schema and parameter definitions in a server module by dragging and dropping tables and field definitions to the Schema Types in the API Definition tab of a server module:



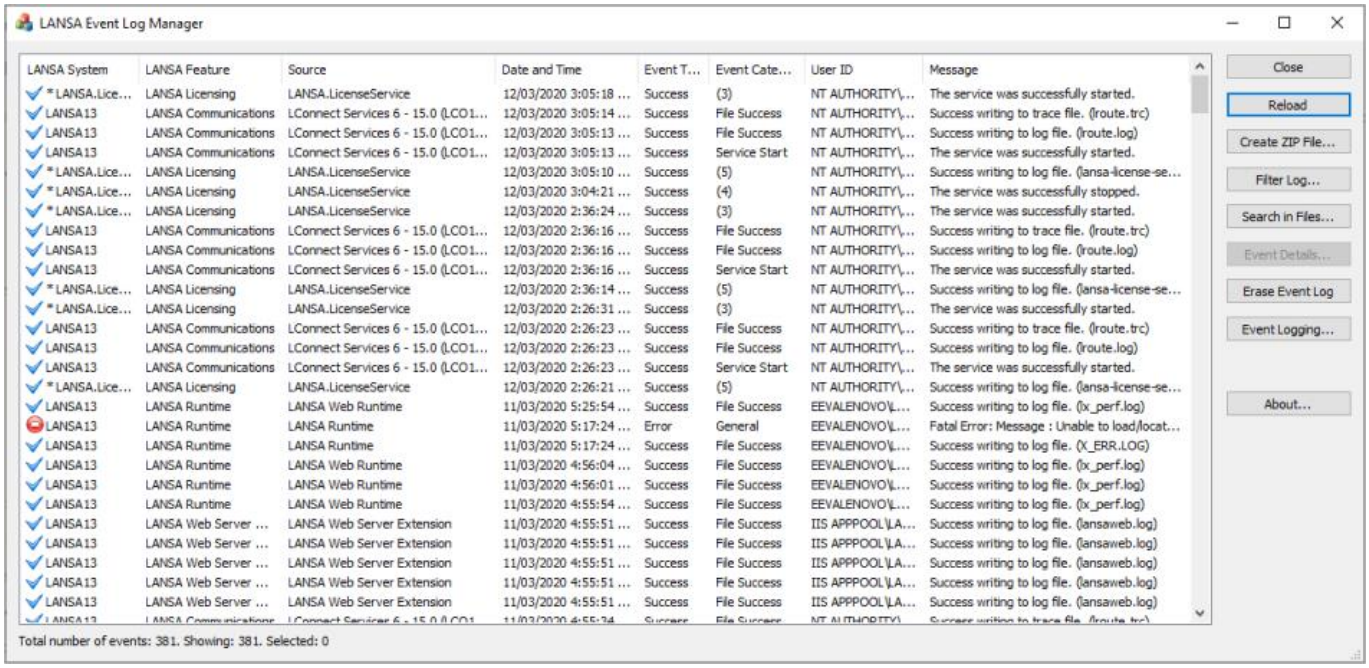
An app showing how to use REST APIs to publish web services is available at the LANSA Developer site:



Other Enhancements

Error Log Manager for LANSA Windows Products

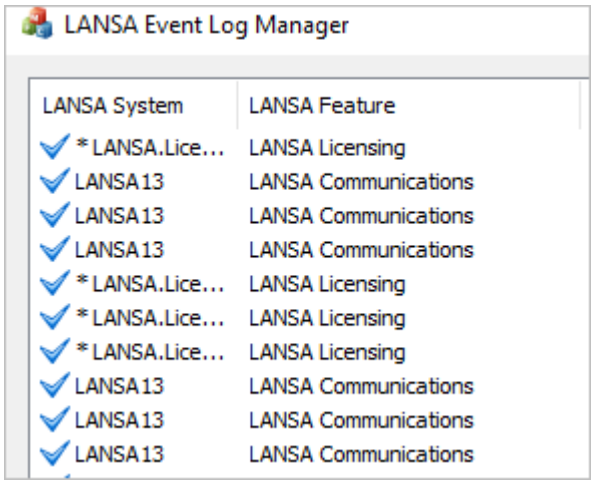
The new LANSA Event Log Manager collects and centralizes detailed event and error information for all Visual LANSA related products, thus eliminating the need to manually search for log and trace files.



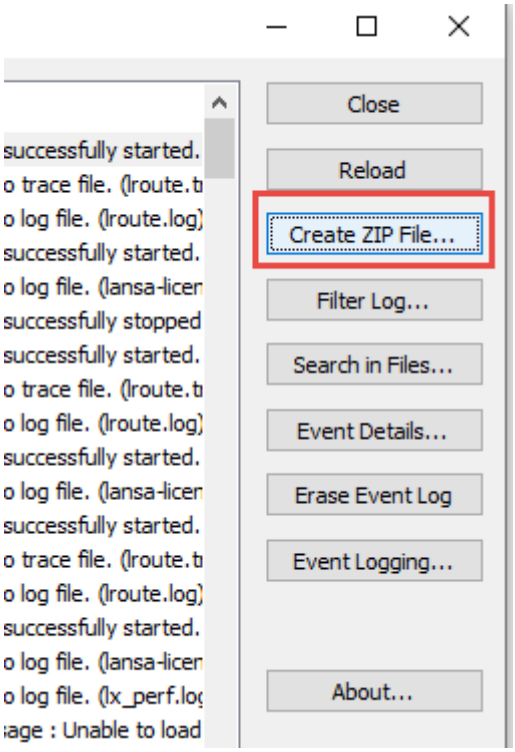
It makes error resolution easy by recording the location of trace and log files and the reason for why a file could not be created.

<u>The centralized event and error log information is available for:</u>

- Main Install
- Deployment
- LANSA Web Server Extension
- LANSA Integrator
- LANSA Communications
- LANSA Open
- LANSA Runtime
- Import
- Web Administrator
- Web Function Editor



If you need to report a problem, you can use *Create Zip File…* to collect information about the PC, the LANSA System involved in logging, the LANSA related registry settings for Windows User IDs involved, and the version number of the LANSA related files in each installation to create a zip file.

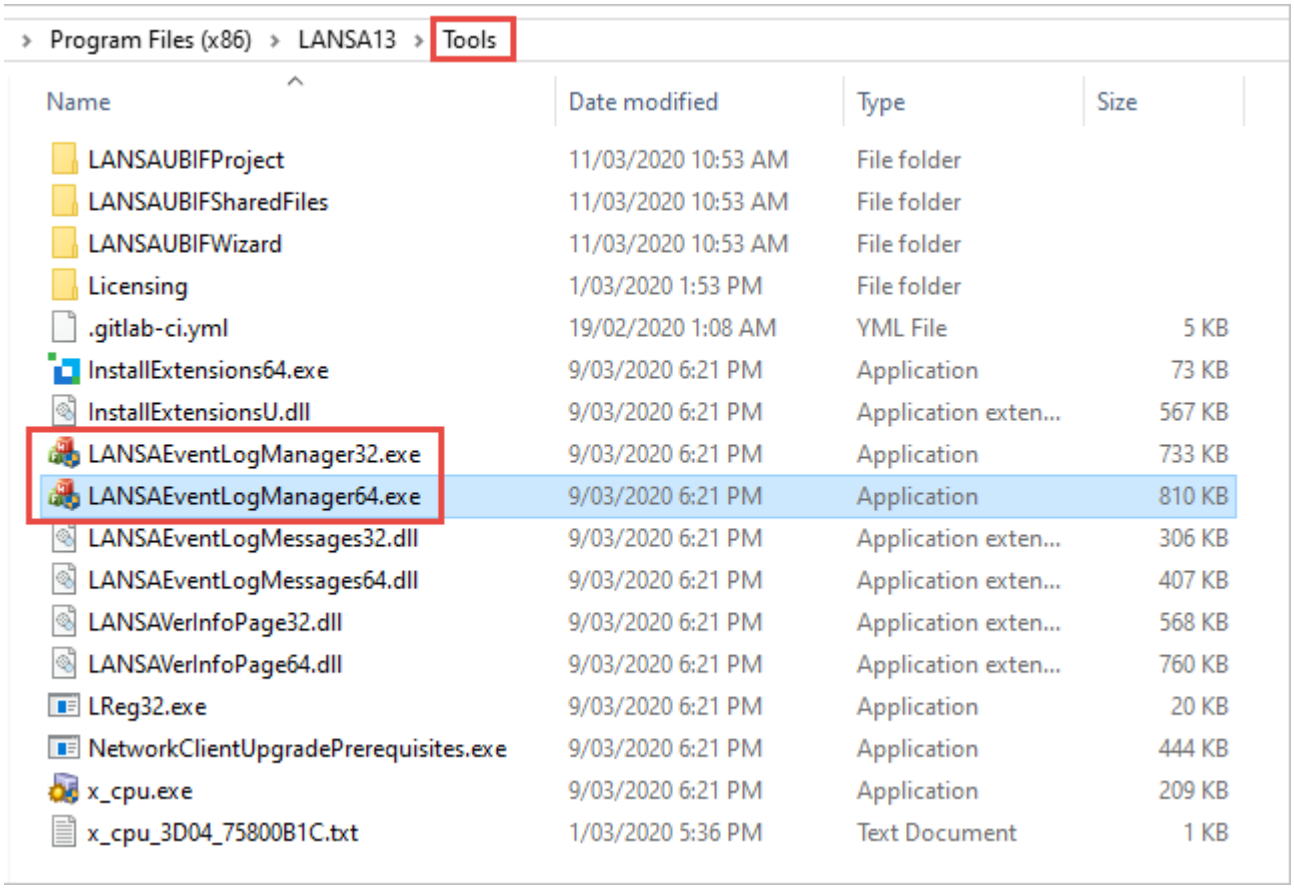


You can also turn event logging off or limit the amount of data logged with each event log record.

Use the Search in Files… option to search for text in trace/log files and configuration files.

Erase the event log when you want to replicate a specific scenario and only see information relevant to it.

To start the Error Log Manager, locate and execute LANSAEventLogManager32 or LANSAEventLogManager64 (for 32 or 64 bit operating systems) in the Tools directory of your LANSA system:



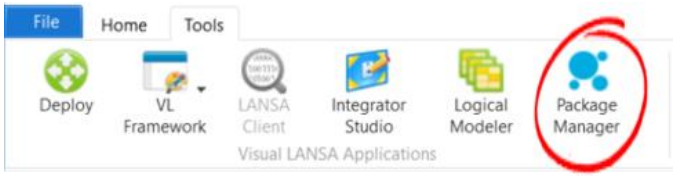
LANSA Package Manager

LANSA Package Manager (LPM) has been introduced to Visual LANSA to manage the download and installation of LANSA packages from the Developer Center that contain common RDML samples and examples to jump-start a developer when building applications.

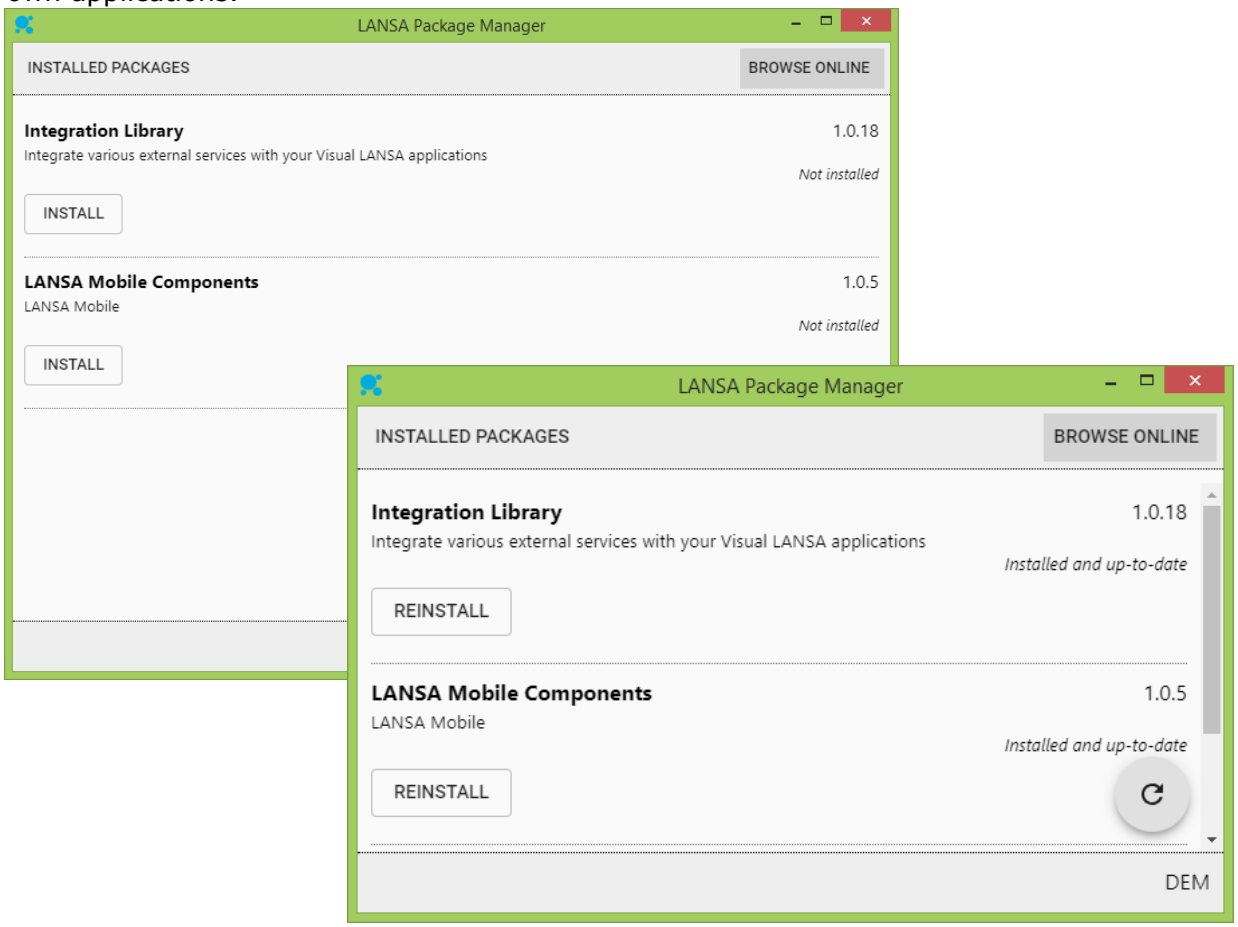
Using LPM, developers can access the latest Integration Library to various external services and Mobile components within their own applications. It also allows LANSA to update these components with new features, functionality, samples, documentation to keep up with the latest suite of services.

How to Open the Package Manager from Visual LANSA IDE

From your Visual LANSA IDE, go to the Tools tab:

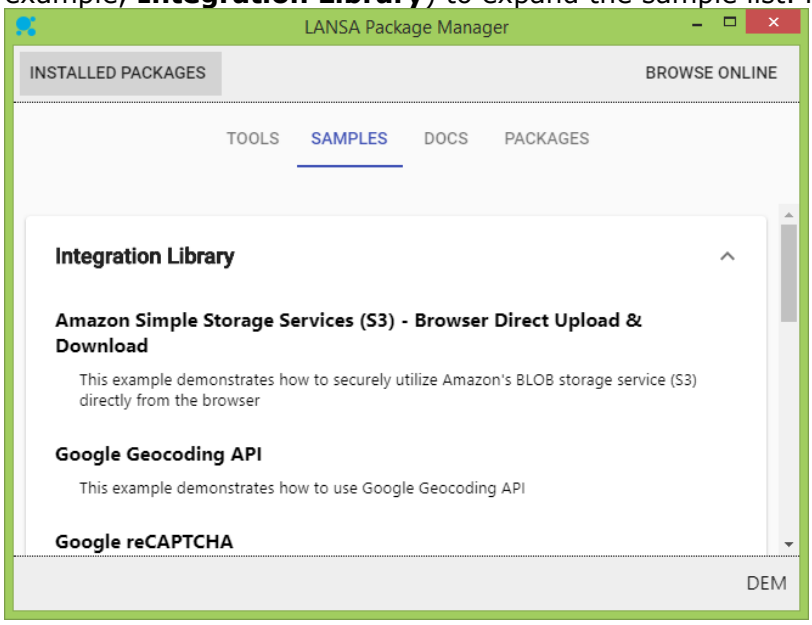


The LANSA Package Manager lets you install LANSA packages on your Visual LANSA development environment. LANSA packages are reusable code that LANSA makes available for you to use in your own applications.

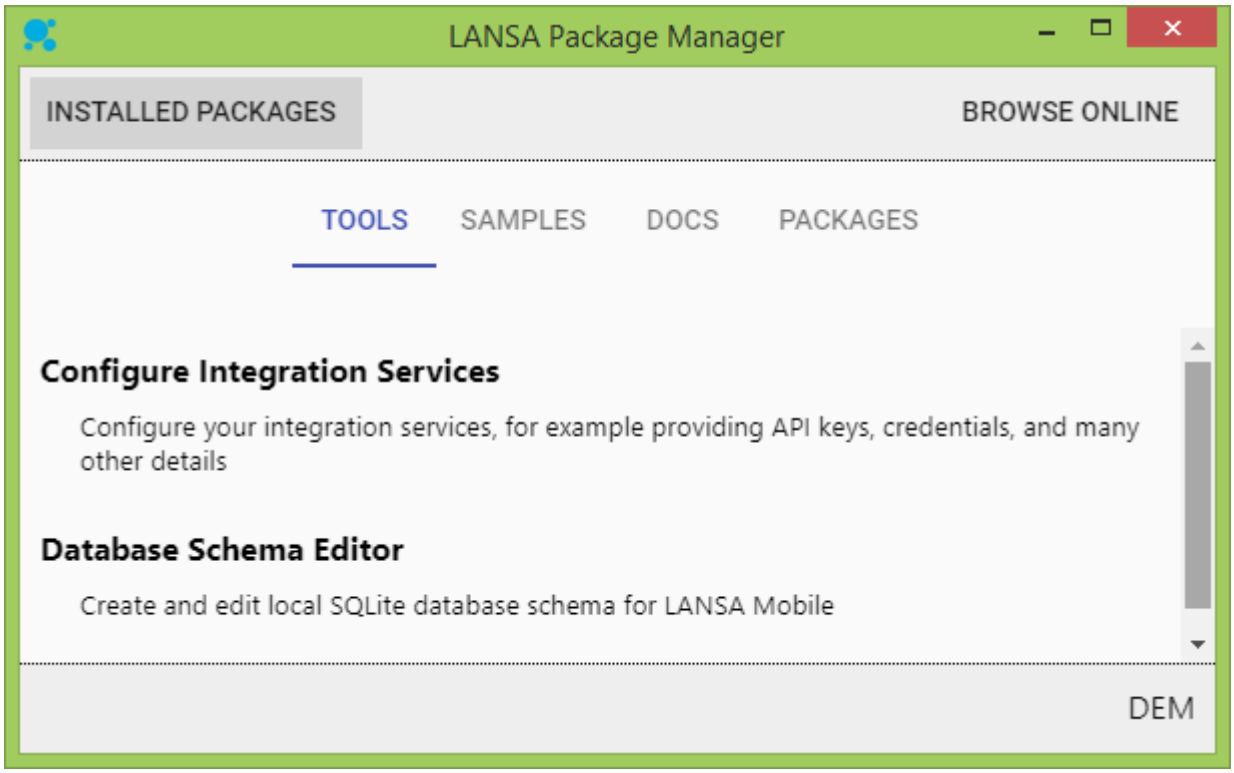


Running samples and tools that come with the package

After installing a package, you can run the samples (and often tools) included in the package. Go to the **Installed Packages** section, then the **Samples** tab. Click on one of the packages (for example, **Integration Library**) to expand the sample list. Pick a sample and click to run.



Some packages also come with some supporting tools. For example, the Integration Library package comes with a configuration tool that let you easily configure the API keys to use, and the **LANSA Mobile** package comes with the **Schema Editor** tool that allows you to edit your SQLite database schema.



Using LANSA Integration Library Packages

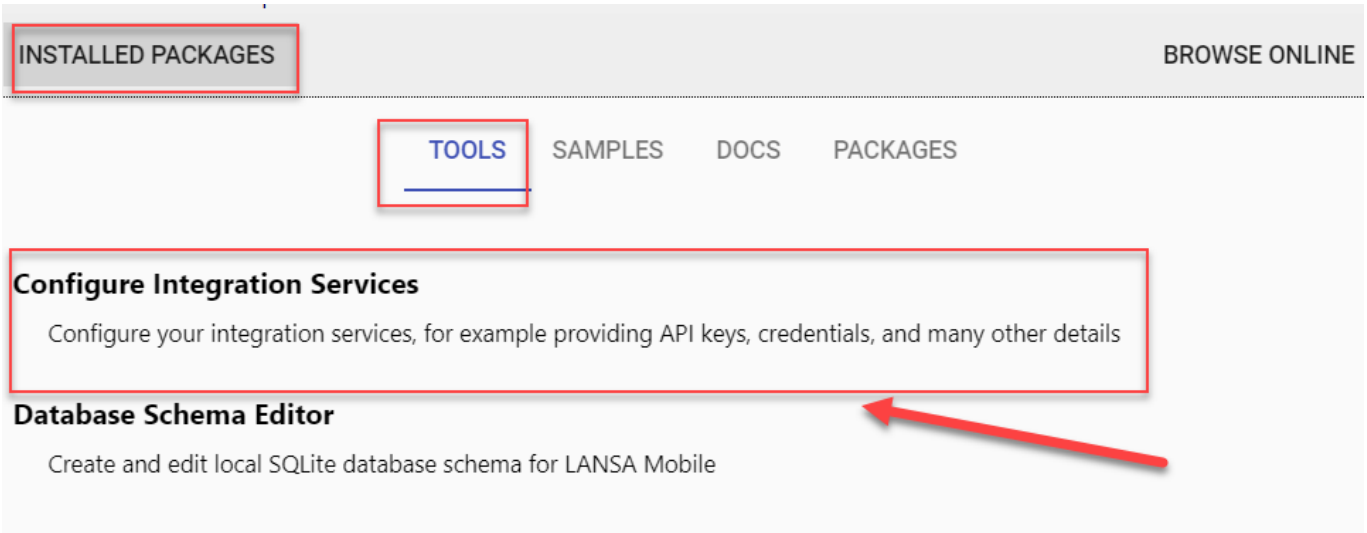
The Integration Library is a LANSA package that you can use in your web application to integrate external services (offered by various providers) in your applications, such as Google/Microsoft sign-on, sending email, SMS, geocoding, and many others.

Configuring the Integration Library

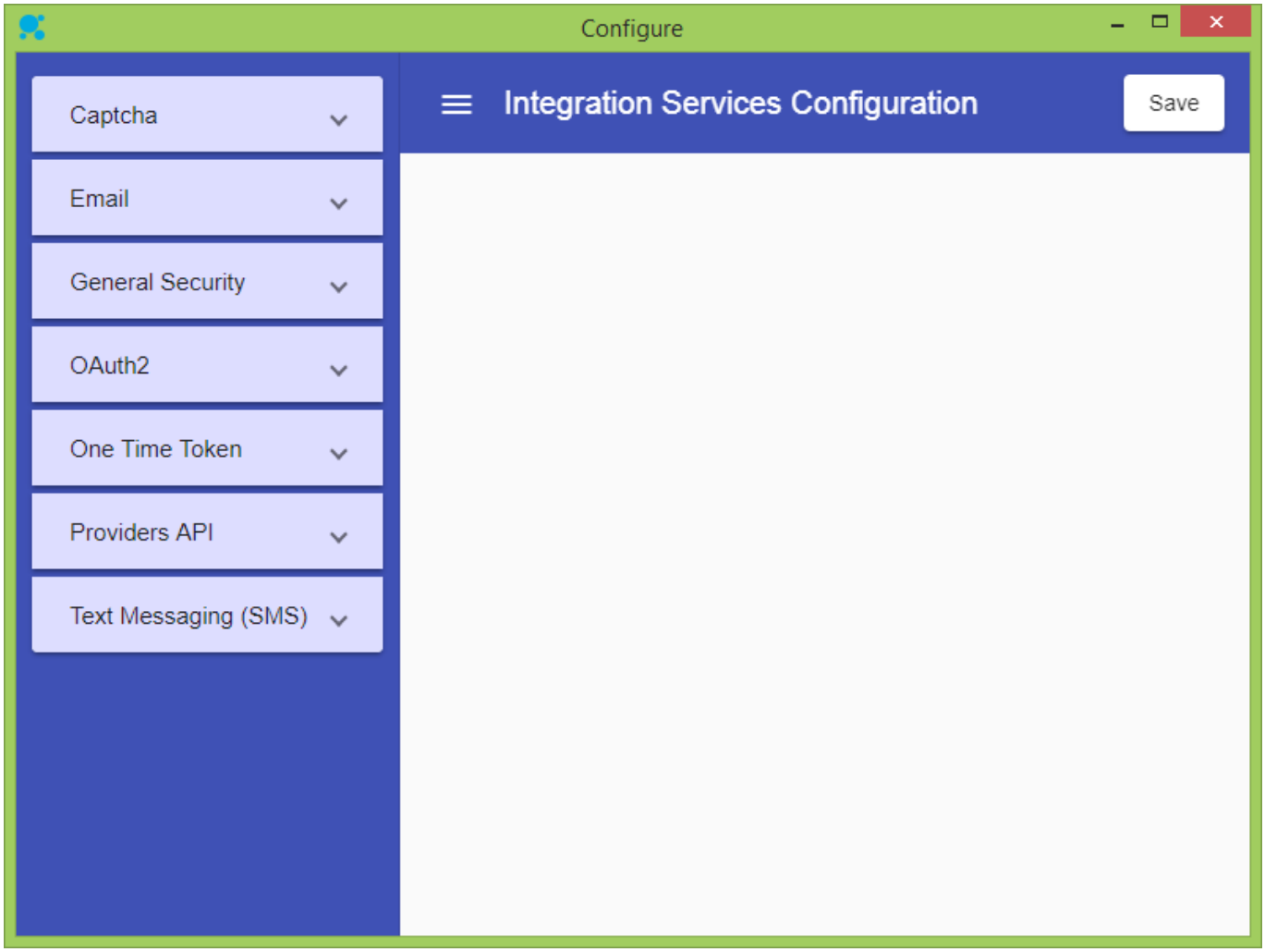
Most of the components in the **Integration Library package** need to be configured before they can be used. These components make use of services provided by a third party, either **for free** or **for a fee**. Either way, they would need to be able to identify you as a user of the service.

Most providers would grant you a unique string, called **API Key**, that can be used to uniquely identify you, as a user of the service. So, the first step before you can use a component in the Integration Library is to login to the service provider's website (for example, Google API Console), and obtain an API key. If it's a paid service, you'd generally need to also register a payment method (generally a credit card).

Once you have your API key, you'd need to tell the Integration Library that you want to use the API key. You can do that easily using the Integration Library Configuration Tool, which can be launched from the Tools tab in the LANSA Package Manager (Installed Packages section).



The tool will launch and present you with a menu of services to configure.



Summary of Prebuilt connectors:

- Amazon Simple Storage Services – Browser Direct Upload & Download
- Google Geocoding API
- Google reCAPTCHA
- Google Translate API
- Login Form with 2-factor authentication
- Single sign-on using Google, Microsoft, and Facebook
- OpenWeatherMap API
- Sending Email using SendGrid API
- Sending SMS Using Twilio

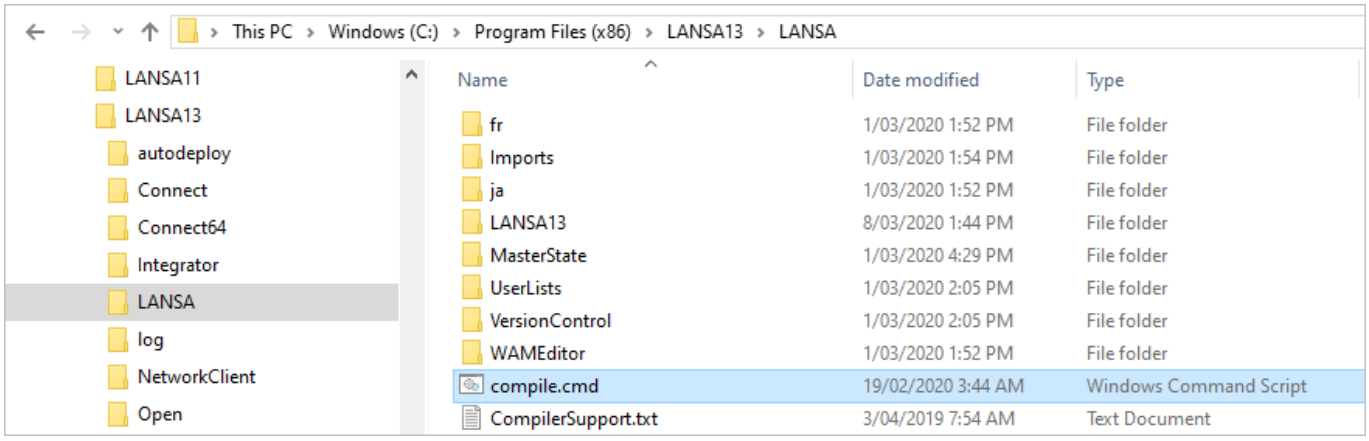
Command Line Build

The Windows command file **compile.cmd** can be used to compile Visual LANSA objects from the command line in a Visual LANSA version controlled (VCS) configuration. You can use it to schedule jobs, utilize CI/CD methodology or simply to run your builds from the command line to utilize the options available.

For example, this command compiles only objects that have changed, are in version control and are part of the project represented by list OL1 in partition DEM:

Compile /PARTITION=DEM /PROJECT=OL1 /OBJECTS=CHG

The compile.cmd file is located in <RootDir>\LANSA where <RootDir> is the root directory where Visual LANSA is installed.



Free Format RPG in LANSA Integrator

Improvements have been made to the Code Generator to produce free format RPG in LANSA Integrator.

Additional MySQL Data Type Support

New data types used in MySQL databases can now be imported to LANSA tables:

- Unsigned BigInt
- Int
- SmallInt
- TinyInt