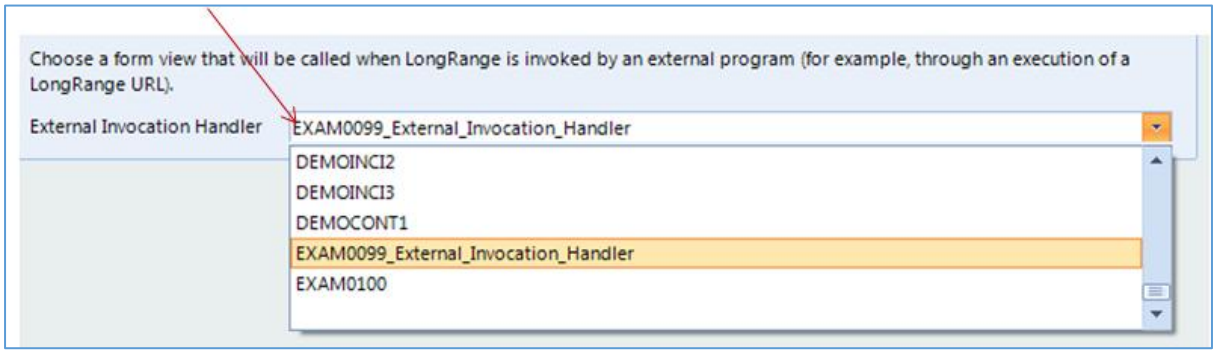
LongRange Runtime Version 14 (RV14)

Activating LongRange from a URL

In LongRange Studio you can now specify a form view as your 'External Invocation Handler':



Doing this means your LongRange application can be automatically 'activated' (ie: started up or awoken) from a URL that:

- Arrives in an email
- Arrives in an SMS
- Is on a web page

Once the External Invocation Handler is activated it can examine parameters on the URL and switch control to whatever form view should handle the request.

This means, for example, that you can send a user an email requesting they approve order 627273. When they touch the hyperlink in the email they are immediately shown the order 627273 and can touch a button to approve it. This type of processing is a foundation of 'alert' or 'push' driven business processes – where users are automatically 'alerted' to anomalies and 'pushed' to complete a business process.

See http://www.longrangemobile.com/docs/LRProgRPG/index.htm#!Documents/activatinglongrangef.htm

Auto-Configuring LongRange from a URL

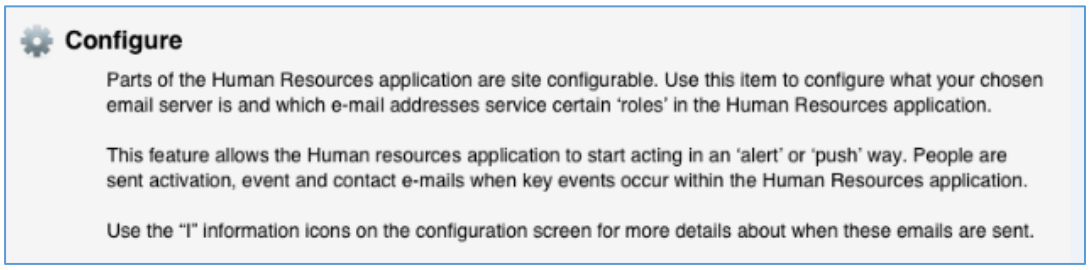
You can now auto-configure LongRange servers from a URL received in an email, and SMS or accessed from a web page.

This reduces the probability of end users making configuration mistakes. See the guide for example of HTML page that an end user can use to install and configure LongRange on their device in 3 or 4 screen touches.

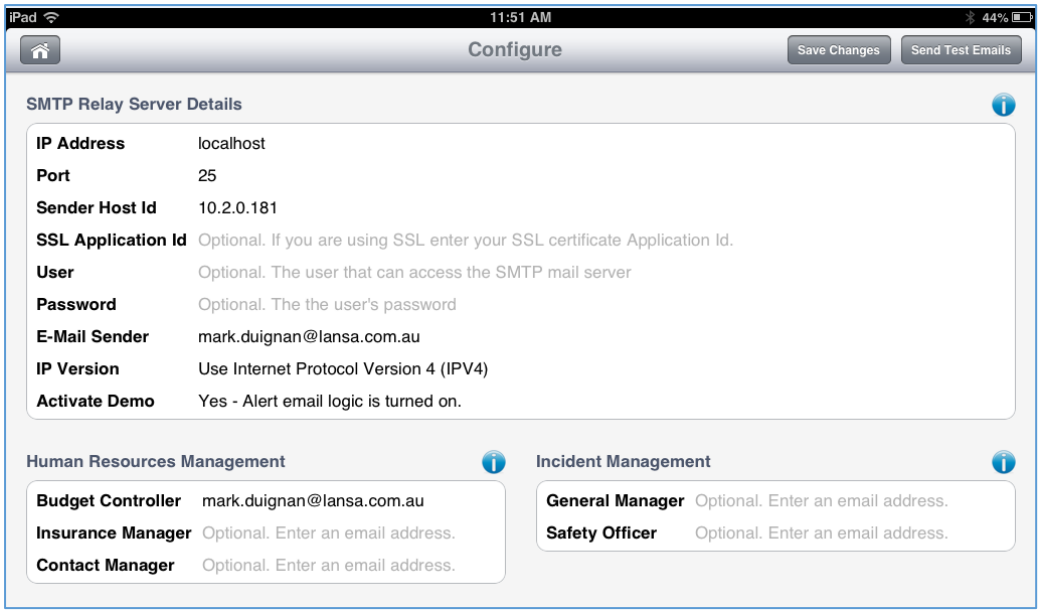
See http://www.longrangemobile.com/docs/LRProgRPG/index.htm#!Documents/autoconfiguringlongr.htm

Demonstration Human Resources application - 'Alert' or 'Push' capabilities added

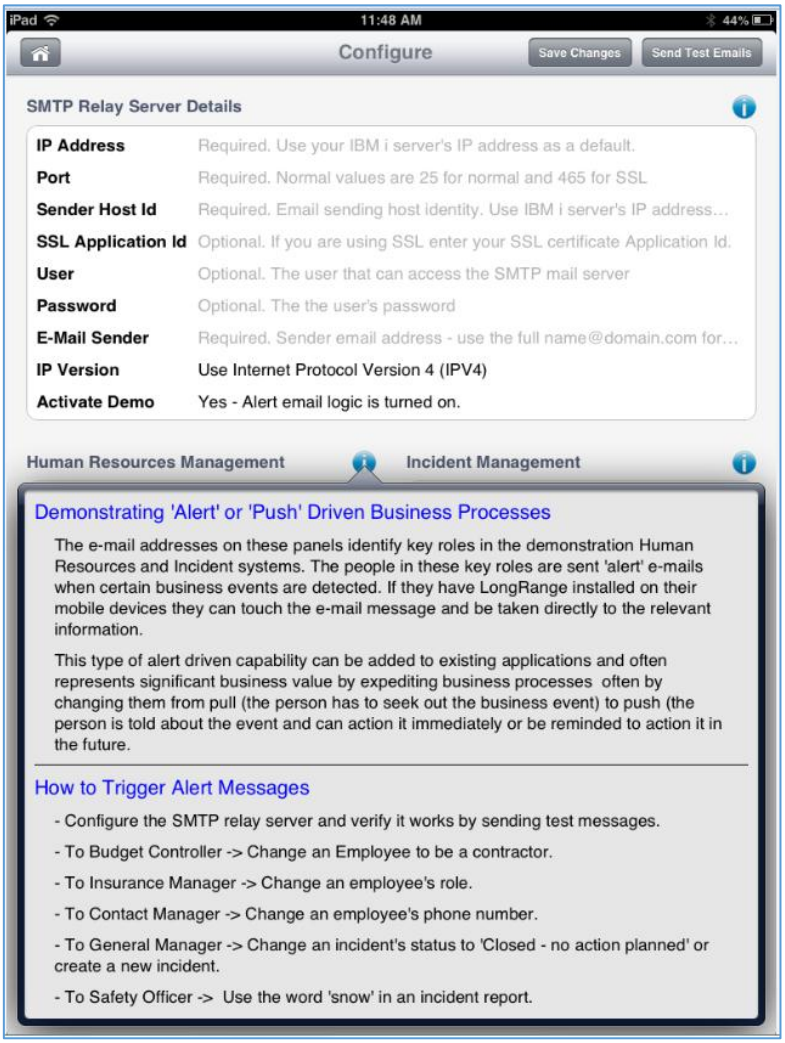
You can use the new Configure option on the Human Resources menu to set up the alert/push demonstration:



This Configure form asks you to define an SMTP mail relay server to use and to specify the e-mail addresses of the people fill certain key 'roles' in the Human Resources and Incident demonstration systems:



Use the information icons to see when and how alert/push notifications are sent to the people in these roles:



From a business role perspective this demo feature activates these business scenarios:

Human Resources - The Budget Controller

The budget controller is sent a message when an employee is changed to be a contractor.

The business scenario is that this type of role change may have budgetary implications for any projects that the employee is currently involved in.

The budget controller would check out the precise details and review the project budgets accordingly.

Human Resources - The Insurance Manager

The insurance manager is sent a message when the role of an employee changes. The business scenario is that when an employee's role changes whether they need to be insured - or what the cost of their insurance should be - may need to be adjusted.

By doing the adjustments in the most timely manner premium costs can be minimized.

Human Resources - The Contact Manager

The contacts manager is sent a message when any phone number associated with an employee changes. The business scenario is that 'non-integrated' external organizations and hardware devices may have employee contact numbers recorded in them.

The contacts manager is also sent a “vcard” an e-mail containing an attachment that contains the complete employee contact details. This may be used to update the contacts on a device or passed on to others.

By notifying 'non-integrated' organizations and devices of employee contact changes the smooth running of the business and the satisfaction levels of customers may be improved.

Incidents - The General Manager

The general manager gets notified every time an incident is updated to have status 'Closed - no action planned'. This is because only the general manager can sign off on an incident being closed with no further action taken.

The general manager is also sent two e-mails whenever a new incident is created. These emails prompt the manager to create reminders for 7 and 14 day incident progress reviews in his or her diary or calendar.

Incidents - The Safety Officer

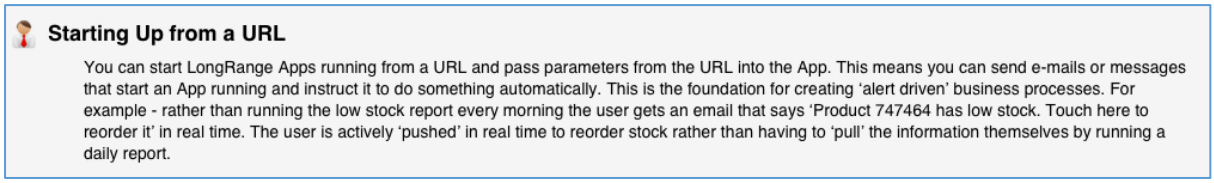
There have been a spate of snow and ice related slips and falls. They are costing the company time and money and cause pain and suffering for those injured. This problem is made worse because the maintenance crews have been slow to remove the slip and fall hazards.

The safety officer is sent an alert whenever an incident containing the word 'snow' appears in the incident system.

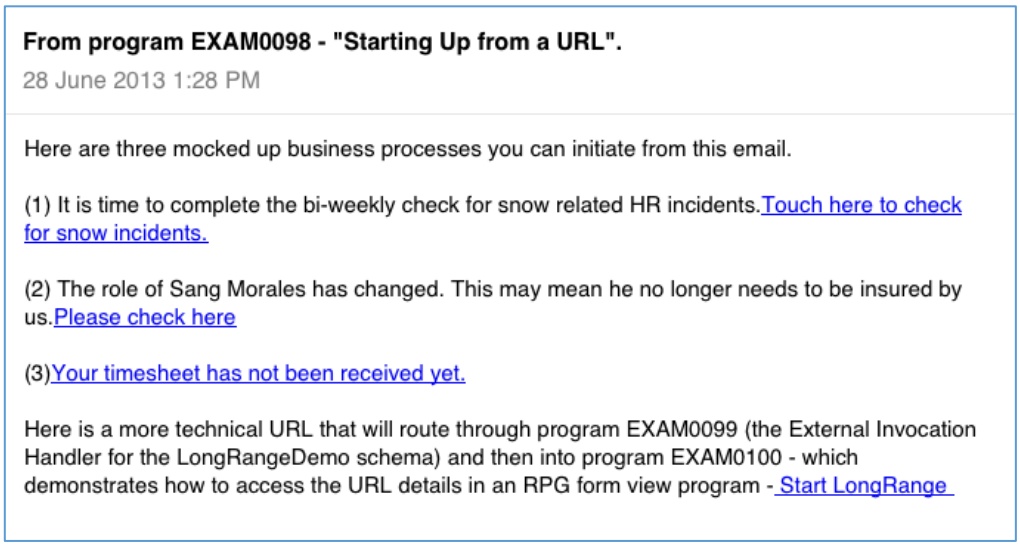
Starting up from a URL

There is a new example to demonstrate sending emails or SMS messages that contain URLs that activate parts of the shipped LongRange demonstration.

On the Programming Examples -> Use Case examples menu:



The email option can send a test mail like this to your mobile device:



The SMS options can send individual test SMS texts containing LongRange URLs to test devices.

By touching the links in the email you can activate LongRange on your device and cause it to search for incidents containing "SNOW" or display the details of an employee.

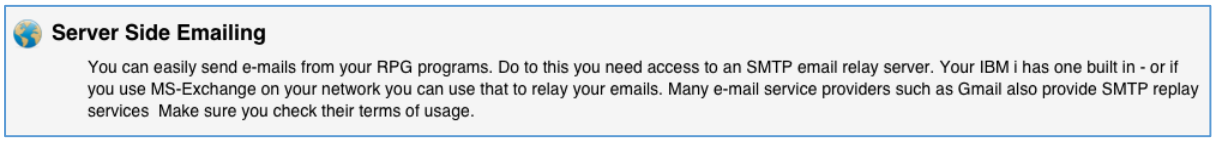
Server Side Emailing (RPG)

A foundation of the alert/push driven business model are emails automatically generated by servers.

IBM I servers have a built-in SMTP mail server- but traditionally it has been hard to access from RPG code.

LongRange-RPG now includes an interface that allows RPG programs to send mails via any accessible SMTP mail relay server – including the native IBM i one, MS-Exchange on a network, Gmail or an ISP's email service.

On the Programming Examples -> Use Case examples menu:

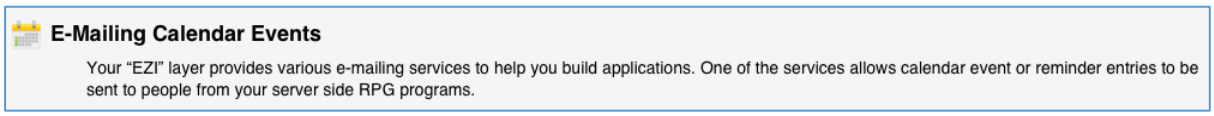


Sending iCalendar Details via Email

Being able to send automatically generated calendar requests (diary entries) and reminders to users, customers, sales staff, etc is an increasingly important part of modern alert or push driven business applications.

The RPG and LANSA version both include examples of sending calendar events by email.

In LongRange-RPG an EZI procedure (EZI_MailVCard) has been introduced to make this a simpler task from RPG code. On the Programming Examples -> Use Case Examples menu:



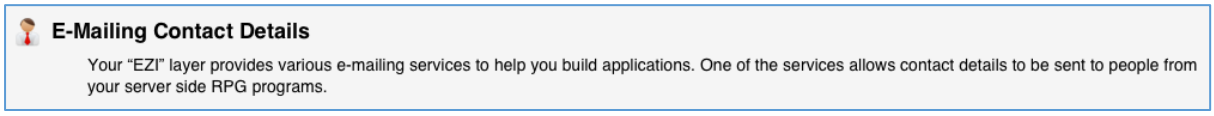
Also see http://en.wikipedia.org/wiki/ICalendar

Sending Contact (Vcard) Details via Email

Being able to exchange automatically generated contact details between users, customers, sales staff, etc is an increasingly important part of modern business applications.

The RPG and LANSA version both includes examples of contact (vcard) exchange by email.

In LongRange-RPG an EZI procedure (EZI_MailEvent) has been introduced to make this a simpler task from RPG code. On the Programming Examples -> Use Case Examples menu:

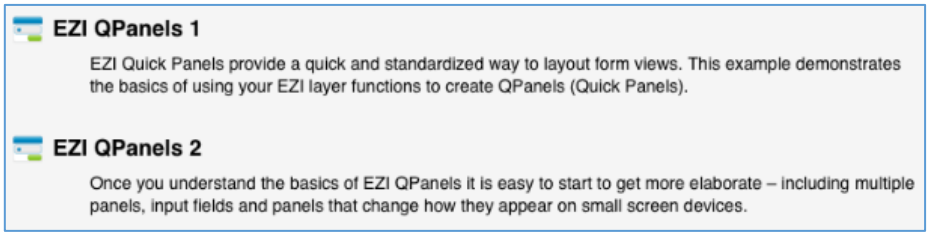


Also see http://en.wikipedia.org/wiki/VCard

"EZI" Quick Panels (RPG only)

The "EZI" layer was first introduced in RV12. It has been expanded to include an EZI Quick Panel that allows basic panels to be quickly formatted in RPG programs.

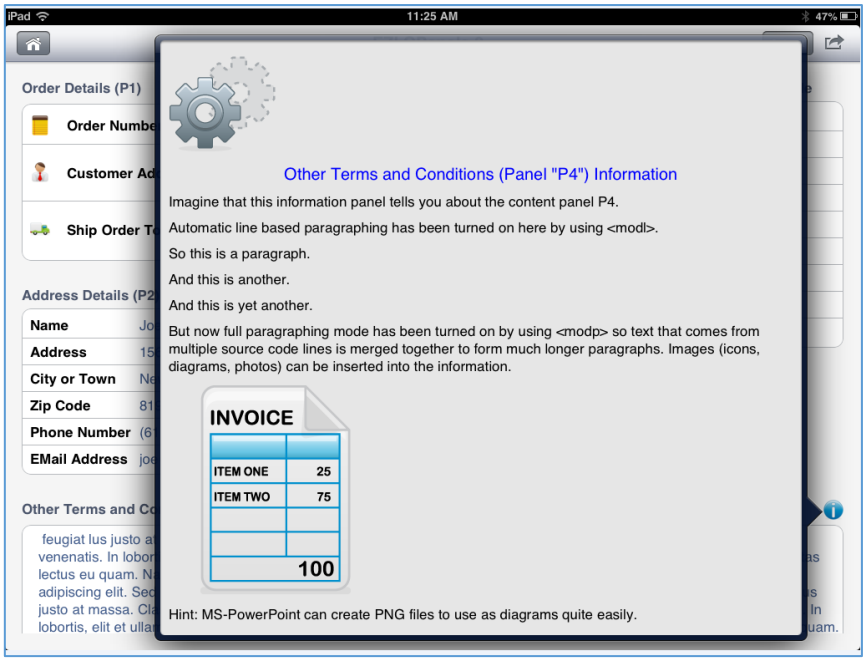
On the Programming Examples -> Introductory Examples menu:



'EZI QPanels 2' demonstrates four EZI Quick panels named P1, P2, P3 and P4 arranged on a form:

The panels flex and change with the device orientation and on small screen devices are arranged under one another instead of side by side. Input fields, drop downs and touchable fields are also demonstrated.

EZI Panels can also automatically present more information (aka: help text) when the user touches the information icon. The text is dynamically read from native IBM I source file members - so it easy to create, test, maintain and deploy using standard IBM facilities…….



"EZI" State Variables (RPG only)

The "EZI" layer was first introduced in RV12. It has been expanded to include a set of procedures that allow RPG programs to permanently remember structured 'state' values (ie: to persistent name-value pairs).

The new procedures are named:

EZI_UsingStateOwner	Set the owner (or container) of the current state data set
EZI_UsingStateObject	Set the object within the current state set
EZI_SaveStateVar	Save the value of a state variable
EZI_SaveNumericStateVar	Save the value of a numeric state variable
EZI_GetStateVar	Get the value of a state variable
EZI_GetNumericStateVar	Get the value of a numeric state variable
EZI_DeleteStateVar s	Delete one or more state variables

If you wanted to associate details with the current user (stored in variable IBMiUserProfile say) and save them you might write code like this:

```
// Imagine variable IBMiUserProfile contains the current user profile
IBMiUserProfile = 'FRED';
// First we give this 'set' of user related information a name ...
EZI_UsingStateOwner('USER_INFORMATION');
// Within the set 'USER_INFORMATION we' are interested in the 'thing'
// named 'USER_XXXXXXX' where XXXXXXX identifies the current user …
EZI_UsingStateObject('USER_'+ IBMiUserProfile);
// Save some details
EZI_SaveStateVar('Name'      : 'Fred');
EZI_SaveStateVar('EMail'     : 'Fred.Bloggs@company.com');
EZI_SaveStateVar('Printers'  : 'PRT1'     : 1); // 'Printers' is a list
EZI_SaveStateVar('Printers'  : 'QPRINTER' : 2);
EZI_SaveStateVar('Printers'  : 'PRT_17'   : 3);
EZI_SaveNumericStateVar('PrinterCount' : 3);
```

Getting these details back later, or into some other program, is done like this ……

```
IBMiUserProfile = 'FRED'; // Say

EZI_UsingStateOwner('USER_INFORMATION');
EZI_UsingStateObject('USER_'+ IBMiUserProfile);


TheUserName     = EZI_GetStateVar('Name');
TheEMailAddress = EZI_GetStateVar('Email');

PCount = EZI_GetNumericStateVar('PrinterCount');

For Instance = 1 to PCount;
    ThePrinterName = EZI_GetStateVar('Printers' : Instance);
EndFor;
```

The get procedures also support a default value – so you do not have to code for the 'not found':

```
TheUserName     = EZI_GetStateVar('Name'  : 1 : 'Name is not available.');
TheEMailAddress = EZI_GetStateVar('Email' : 1 : '*NONE');
```

The delete function handles a variable number of keys …

```
// Delete all state variables
DeleteCount = EZI_DeleteStateVars();

// Delete everything in the 'USER_INFORMATION'set
DeleteCount = EZI_DeleteStateVars('USER_INFORMATION));

// Delete everything in USERINFORMATION related to user 'FRED'
DeleteCount = EZI_DeleteStateVars('USER_INFORMATION ':'USER_FRED'));

// Delete just Fred's email details
DeleteCount = EZI_DeleteStateVars('USER_INFORMATION ':'USER_FRED':'Email'));

// Delete just Fred's second printer
DeleteCount = EZI_DeleteStateVars('USER_INFORMATION ':'USER_FRED':'Printers':2));
```

In using these procedures a grid like this can be useful for design and documentation purposes.

Here are some examples of structures you could invent using state variables:

Set Name	Object Name	State Variable Name
USER_INFORMATION	**USER_<user profile>**	Name
		Email
		Printers (list of 20 max)
		PrinterCount
		CurrentOrder
		CurrentCustomer
DEVICE_INFORMATION	**DEVICE_<device GUID>**	AssignedDeviceName
		AssignedDeviceGUID
		DateTimeFirstyUsed
		RestrictedAccess
		RequiresAuthentification
	REGISTEREDUSER_<user profile>	RegisteredDeviceUserName
		LastUsedDateTime

		PreferredPrinter
		CanAccessAccounts
CORPORATE_INFO	**MAIL_SERVER_INFO**	ServerCount
		IP Addresses (list of 10 max)
		Ports (list of 10 max)
	COMPANY_<Company Number>	GenricSenderName
		GenericReplyAddress
		GeneralManagerEMail

Like all the EZI procedures the DDS and RPG source code is provided for you to own and use as you see fit. It is shipped in the LRNG_PROJ library's QDDSSRC, QCLSRC and QRPGLESRC files.

Note: State values are stored in data base table EZIPERSIST in your copy of the shipped LRNG_PROJ library. You can modify this as you like. You need to back up your state variables.

HTML formatted EMails

The RunOperation SendMail prepares emails to be sent from the client device.

It has a new parameter that allows you to format the email body as a HTML document.

http://www.longrangemobile.com/docs/LRProgRPG/index.htm#!Documents/sendmail.htm

Note: The ability of different e-mail clients to display and process HTML varies.

BarCode Scanning and Magnetic Stripe Reading

The barcode scanner element has a new 'ValueSource' property that indicates the source of the received value (BUILTIN, MANUAL etc) – refer to online documentation.

The Linea Pro interface now supports magnetic card stripe reading.

JavaScript API Enhancements

LongRange can be programmed in RPG, COBOL, CL and RDMLX (LANSA).

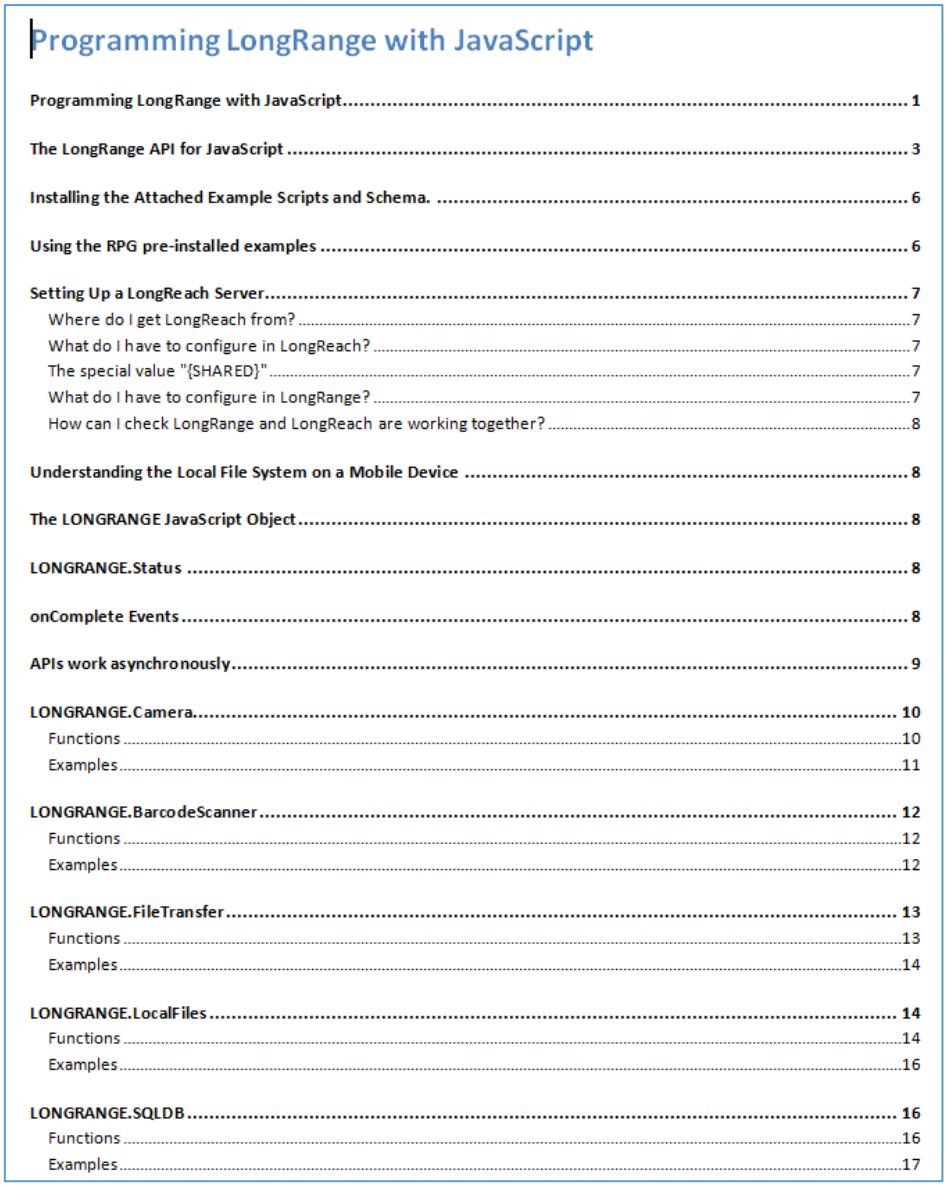
It can also be programmed using JavaScript.

The LongRange JavaScript API (named LONGRANGE) allows any JavaScript from any vendor to programmatically:

- Send files from the local file system to a server system
- Get from files from a server and save them to the local file system.
- Read, write, copy and delete files on the local file system.
- Read bar codes via the camera
- Record photos and videos to the local file system.
- Record audio tracks to the local file system.
- Perform SQL data base operations against a local SQL data base.

The LongRange-API capability has been expanded in RV14.

If you are using the JavaScript API contact your product vendor for the latest copy of this guide:



Programming LongRange with JavaScript

Corrections

The disable property now works correctly in Android.

An issue with the Linea Pro interface honouring the barcode type restrictions specified in the BarcodeTypes property has been corrected.

OpenURL operations involving URLs containing non-ASCII characters now function correctly.

The dynamic use of the ‘Multiline’ TextBox property now function elements correctly in iOS.

Upgrading to RV14

Component to Upgrade	Get the Upgrade Material from Here
Android Client	https://play.google.com/store/apps/details?id=com.lansa.longrange
iPad, iPhone and iPad Client	https://itunes.apple.com/app/longrange/id508286590?mt=8
LongRange Studio	RPG - http://www.longrangemobile.com/upgrade.htm LANSA - http://www.lansa.com/products/longrange/downloadstudio.htm
LongRange Server Package for RPG	http://www.longrangemobile.com/upgrade.htm Use the  option. Open the zip file and read the installation instructions.
LongRange Server Package for LANSA	http://www.lansa.com.au/products/longrange/download.htm Use the  option

If you have any issues use your regional support http://www.longrangemobile.com/support.htm

or ask on the user forum http://longrange.lansa.com.au/