

DataMax Software Group Inc.
1101 Investment Blvd.
El Dorado Hills, CA 95762
USA

RFgen Users Guide

All Editions
RFgen 5.2



Table of Contents

Introduction to RFgen	1	Connection Tab	32
Basic Implementation Steps	2	Connection Tab - SAP	32
Configuration Overview	3	SAP Data Encryption	33
Configurations Shared by Dev Studio and Services Console:	3	To Configure for JDE	34
Connections Shared by Dev Studio and Services Console:	4	Adding A New Web Services Connection ..	39
Configurations Available Only in Dev Studio	4	Configuring the Host Connection	41
Connections Available Only in Dev Stu- dio	4	Configuring User Access Control	44
Configuring the RFgen Application Data- base	4	Adding or Removing Administrators	45
Configuring Application Preferences	7	Dev Studio Configuration Options	46
Configuring Application Services	10	Configuring Menu and Key Settings	47
Configuring Environment Settings	14	Configuring the Scripting Environment ..	49
Configure System Environment	14	Configuring Source Control Options	51
Configuring Transaction Management DB Connection	17	Configuring System Properties	53
Create Application Event Database	18	Download Enterprise Objects	53
Configuring System Queues and Tasks	19	Downloading ERP Business Functions ..	54
Add New Enterprise Connections	21	Downloading JDE Processing Options ..	54
Adding a New DataSource Connection ..	21	Viewing Enterprise Objects	56
Connection, Pooling, Indexing, and Availability Tabs	23	Viewing ERP Business Functions	57
Connection Tab	24	Viewing JDE Processing Options	57
Pooling	27		
Indexing tab	28		
Availability Tab	28		
Add an New ERP System Connection	29		
Configuring RFgen to SAP	30		
		Mobile Unity Platform Services Con- sole	59
		Accessing the RFgen Server Console (Mobile Unity Platform Console) and Services	60
		Management of RFgen Server Services	61
		To Stop or Start Services Using Com- mands	62
		To Enable or Disable Maintenance Mode ..	62
		To Enable Maintenance Mode	63
		Server Authorization	63

Table of Contents

To Activate (Authorize) the Server	63	The Layout Control	83
Authorize by Email	63	The Line Control	84
Phone Authorization Request	64	The ListBox Control	85
Activation by web	64	The Map Control	86
Device Authorizations	65	The Memo Control	87
To authorize Mobile, Offline Clients	66	The Menu Control	87
To authorize Thin clients	66	The Panel Control	88
Enterprise: Reports Menu	66	The Page Control	89
Mobile Development Studio Overview .	68	The PanelList Object	90
Application Tools and Controls	69	The RadioButton Control	91
Applications Menu Bar	70	The Signature Control	91
To Add a New Application (Form)	71	The TabControl Control	92
Form Control Features	71	The TextBox Control	94
To Add, Remove or View Pages	72	The TreeView Control	95
The Page Control	72	Graphical Control Properties	96
Property FAQs	74	How to tell if Themes is being applied ..	97
To Move a Control	75	Control Property Descriptions	97
TIPS for Placing and Sizing Objects on a Form	76	Form Properties Description	109
The Button Control	77	The Page Control	111
The ButtonList Control	77	Using Mobile Themes for a Graphical Control	113
The CheckBox Control	78	Manage Rows Collection	113
The ComboBox Control	78	Manage Displays Collection	113
The DateTime Control	79	Manage Icons Collection	114
The DataGridView Control	80	To Set Text Defaults in a Graphical Control	115
The Frame Control	80	To Validate Entries in a Graphical Control	115
The Image Control	81		
The ImageList Control	82		
The Label Control	82		

Table of Contents

Scripting with VBA	116	Text Resources (Language Translations)	135
Script View Ribbon Menu	116	Text Resources (Language Translations)	136
Scripting Rules of Precedence	116	To Translate Strings into Multiple Locales	137
SideBarMenus	117	Before You Start	137
Configuring SideBar FKeys	118	Step A. Add Your Primary Text Resource	138
Setting System Operations	118	Step B. Add Other Languages and Populate from Reference Language	141
Mobile Themes for SideBar	119	Step C. Translate Text	142
Source Control Options in the Solution Designer	119	To Localize Text in Graphical Displays	142
Source Control Menu and Object States	120	Scripting Modules	144
Shelve Changes	121	To Make a Reference to Other Modules	144
Find Shelve Set	121	Keyboards	144
Testing Mobile AppsOverview	122	To add and customize a keyboard	145
Mobile Apps: Session Property Descriptions	123	To Stylize a Keyboard	148
To Set Your Testing Preferences	124	To Import a Keyboard	149
To Debug Code	125	Devices	150
Debugging options	125	To Add a Device Skin	151
To Test Your App and Bypass the Login App	126	Voice Applications	154
Text Resources (Language Translations)	127	Menus and Roles	155
Database Exports	128	Assign menus to user or user groups	155
Data Transactions	129	Menus Tools	156
To create a Data Transaction macro	130	Menus Simulator	157
Data Transactions Menu	131	To Add Menus	158
To Allow References to Global Scripts	132	User Overview	159
Images	133	To Create New Users	161
Image Menu	134	To Assign Menus and/or Roles to a	162
Themes: Signature	134		

Table of Contents

User	188
To Remove a User	162
To User Access to Applications in a Shared Menu	162
To enable application access to specific users	162
Profiles Overview	166
To Setup a Profile	166
To select applications/items	167
To set the options for a Profile	167
To deploy/install the profile to a client	167
To create a Mobile Profile	167
Profile Options	168
Screen Mapping	172
Themes Overview	173
Themes: Application Properties	174
Application Theme States	176
Theme Element Descriptions	177
Themes: Application Colors	178
To Enable Device Emulation	179
Voice Tasks	183
To Import Voice Tasks	183
Testing Overview	184
Mobile Apps: Session Property Descriptions	185
To Set Your Testing Preferences	186
Testing Mobile AppsOverview	187
Mobile Apps: Session Property Descriptions	188
Service Requests Testing	188
SQL Query Testing	188
Transaction Queue Testing	190
Testing Overview	191
Mobile Apps: Session Property Descriptions	192
Device Management Options	192
Solution Deployment	193
Reports	193
Application Statistics	195
Task List	196
Performance Monitoring	196
To Setup Performance Monitoring	197
Setup Connection to Transaction Database	197
Setup Your Events Database	198
Set Performance Monitoring Thresholds	199
System Event Log Menu	200
Upgrade Report	201
Utilities	201
Validate System	202
To Exporting Solution Files	202
To Import Solution Files	203
Script Validation	203
RFgen Client Software Overview	205
Client Configuration Settings	205
Thin Client Overview	206
Connect and Deploy to RFgen Clients	206
Client Network Control Service	206

Table of Contents

Third-Party Mobile Device Management Tools	207
Installing the RFgen Windows CE Client	207
Installing to the RFgen Server	207
RFgen Client Install Process	207
Activating Mobile (Batch) Clients	211
Activating Mobile (Batch) Clients	212
Mobile Enterprise Dashboard	214
Mobile Enterprise Dashboard Menu	215
Overview of Dashboard Views	216
RFgen Server Connections	216
To Set the Language in Your Dashboard	217
To Change Your Dashboard's Appearance	218
To Configure Your Views	218
To Monitor and Interact with an Active Client	221
To Broadcast a Message	222
To Suspend or Terminate a Session	223
Display Options	223
Transaction Management Dashboard	225
Configuring the Transaction Management Dashboard	225
Run on Server	226
RFqueue Status Message	226
To Create or Select Your Data Source	227
System DataBase is Access	228
System DataBase is Oracle	229
System DataBase is SQL	229
System DataBase is SQLite	230
To Change Your Dashboard's Theme	230
To Set the Locale in Your Dashboard	231
User Management Console	232
User Management Console Menu	232
User Management Console Menu Bar	233
To Add Menus and/or Roles	235
To Find and Replace, Export or Import Menus	237
To Remove or Rearrange Menus	238
To Remove a Menu and its Form	238
Menus Simulator	239
To Create New Users	239
To Assign Menus and/or Roles to a User	240
To Remove a User	240
User Tools: Export or Import Users	241
To Search Users	242
Graphical Control Properties	244
How to tell if Themes is being applied	244
Control Property Descriptions	244
Appendix B: Screen Mapping	257
Theory of Operation	257
Screen Mapping	257
How to Make Screen Mapping Work	258
Programming Philosophy	258
Design Considerations for Screen Mapping	258
Screen Mapping Levels	258

Table of Contents

Logon Security Considerations -	
Screen Mapping	259
System Integrity Considerations	259
Keyboard/Special Key Configurations .	260
Runtime Environment Variables	260
Configuring the Host Connection	260
Availability	264
VT220 Key Mapping	266
How to record a macro for a screen map	267
Building a Start Menu Macro	268
The Send Keys Selection	269
Start Menu Macro – Record System	
Sign on	269
Start Menu Macro – Identify System	
Menu	270
Start Menu Macro – Record System	
Signoff	271
Start Menu Macro – Record Error	
Recovery	271
Start Menu Macro – Test Scripts	271
Building a Host Screen Macro	272
Host Screen Macro – Go to the Application Screen	273
Host Screen Macro – Identify the Application Screen	273
Host Screen Macro – Return to the Main Menu	273
Host Screen Macro – Test Scripts	274
Building a Host Transaction Macro	274
Recording Options	277
Building a Screen Mapping Application ...	278

Introduction to RFgen

RFgen's Mobile Unity Platform™ is the product that enables communication and the exchange of data between devices (Windows desktop, Windows CE, Android and iOS), databases (SQLite, SQL, Oracle, etc) and Enterprise Resource Planning (ERP) systems such as Oracle JD Edwards, Oracle Cloud, SAP, or other legacy systems so that users can perform warehouse tasks such as scanning barcodes and transacting changes to inventory using mobile devices. For example, cycle counts can be performed on or off-site regardless of whether the user is connected to the company's network.

The Mobile Unity Platform (also called the RFgen server) features:

- Tools for customizing or developing new warehouse applications
- Consoles for administrating and maintaining RFgen Services
- Dashboards and consoles for monitoring user sessions, transactions, and device access to the server

By default, the Mobile Unity Platform installer also installs the **RFgen Mobile Development Studio**, which provides the software tools and platform used to design, test, and deploy mobile applications/solutions.

To enable communication between the RFgen server and devices (Windows Desktop, Windows CE, Android, or iOS), install the [RFgen client software](#).

For more details on installing RFgen products, see these other guides (pdfs) which are also available from the RFgen online help:

- [RFgen Install and Upgrade Guide](#)
- [RFgen Mobile Client Install Guide for Android](#)
- [RFgen Mobile Client Install Guide for iOS](#)
- [RFgen Mobile Client Install Guide for Windows CE/Mobile](#)
- [RFgen Mobile Client Install Guide for Windows Desktop Client](#)

TIP

Before you can begin using the Mobile Unity Platform or Mobile Development Studio, you will need to configure the RFgen server, database application, database connections and applicable ERP connections.

Basic Implementation Steps

Before you configure your Mobile Development Studio or the Mobile Unity Platform Services Console, it is necessary to understand the basic concepts of your Data Base Management System (DBMS).

- Knowledge of data structures is of particular importance for database applications since it is necessary to understand the basic concepts of 'tables', 'fields/columns', and 'data types' prior to creating applications.
- Understanding Structured Query Language (SQL) 'syntax' is also helpful with database applications.
- Experience with the Microsoft Visual Basic/VBA programming language is helpful in the development of advanced data collection applications, for use with both SQL databases and legacy host-based applications.
- Once you have installed the Mobile Unity Platform Services (RFgen Server) you will need to configuration your server services and other environment features as well as set up data source connections if they have not be setup already.

For information on configuring the Mobile Unity Platform Services console and/or the Mobile Development Studio, see [Configuration Overview](#).

After you have configured application data source(s), and [added new Enterprise connections](#), the following steps are generally implemented for ODBC or SQL-compliant databases:

1. Transaction tables specific to your application are designed via your main database system (i.e., data to be captured is defined in your database). Typically, an individual 'transaction table' with appropriate field definitions is established for each data entry process.
2. The Database transaction table field definitions are downloaded to RFgen. (See [Enterprise Connections > Download Enterprise Objects](#).)

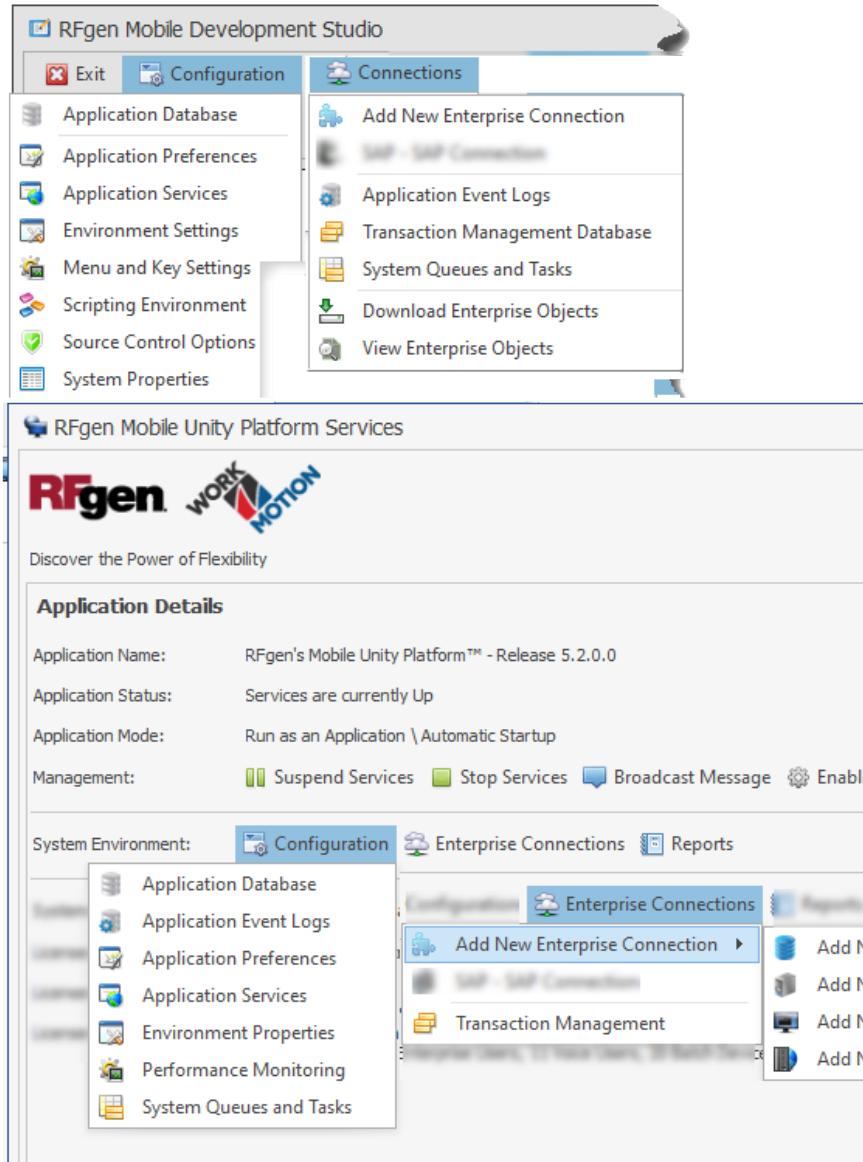
These definitions contain all the necessary information concerning the transaction data to be written to your database. (No data is downloaded from the database.) Table definitions may be partially edited (inside Mobile Development Studio), if required. Similarly, ERP business rules and functions may be accessed and downloaded, if using the RFgen ERP integration suites.

3. Data entry Applications are made for each transaction table. Pre-programmed data properties such as 'defaults', 'validations/edits', 'table validations', etc. are added, as required.
4. (Optional) Once your configuration and resources are setup, you can, if desired, use the [Mobile Development Studio](#) to design, modify, script, test and deploy mobile applications.
5. For information with connecting with your clients, refer to the installation guide for your client's platform.
 - For Windows CE/Mobile device or Windows desktop client, see [Windows Desktop Client Install Guide](#) or [Windows CE/Mobile Install Guide](#).
 - For Android client, see the [Android Client Install Guide](#).
 - For iOS client, see the [iOS Client Install Guide](#).

Configuration Overview

The picture below shows the configuration menus in the Mobile Development Studio (Dev Studio) and the Mobile Unity Platform Services Console (Services Console). The following configuration and connection menus appear on both platforms in the event your RFgen Server is on a **different** system than your Dev Studio.

If your Dev Studio and RFgen Services Console are on the same system, whether you configure from Dev Studio or Services Console, the results will be the same.

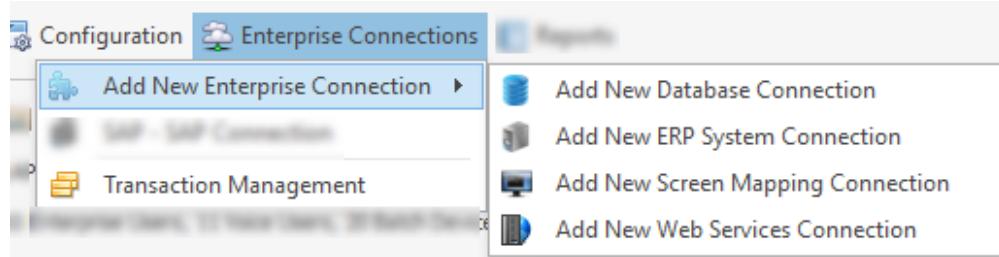


Configurations Shared by Dev Studio and Services Console:

- The [Application Database](#) configures the database storing all the solution objects.
- The [Application Preferences](#) is for Service Console/Dev Studio user interface and language settings, and Mobile Application Designer/Testing and scripting settings.

- The [Application Services](#) is primarily for service communication and security. It includes the server name, port configurations, setup for servers used in load balancing/custers, service run mode, NAT Firewall setup, and Administrative and Encryption configurations.
- The [Environment Settings](#) contains the system options, timeout values, Scanner Options, Google Maps Integration (License Key), device menu strip SideBar Options and Menu Options, and a way to assign actions to device function keys.

Connections Shared by Dev Studio and Services Console:



An unlimited number of data connections can be added to your Enterprise solution. The various types of connections are access from the RFgen Services Console and/or the Mobile Development Studio Connections menu.

- The [Add New Enterprise Connection](#). Use this menu to create a connection with a non-ERP based data source, a commercial ERP system, legacy Screen Mapping hosts, or a web service.
- The [Transaction Management Database](#) - Creates a connection with a transaction management database. For details, see "Configuring Transaction Management Database".

Configurations Available Only in Dev Studio

- The [Scripting Environment](#) settings to allow direct access to Active Directory Objects (ADO) and XML language extension parameters and have them globally loaded into BAS files.
- The [Source Control Options](#) allow developers to use a third-party source control product if its plug-in is supported.
- The [System Properties](#) for adding user-defined system properties.
- The [User Access Control](#) console allows you to authenticate connections between the RFgen Server and the *Mobile Platform Unity Management Console*, and *User Management Console*.

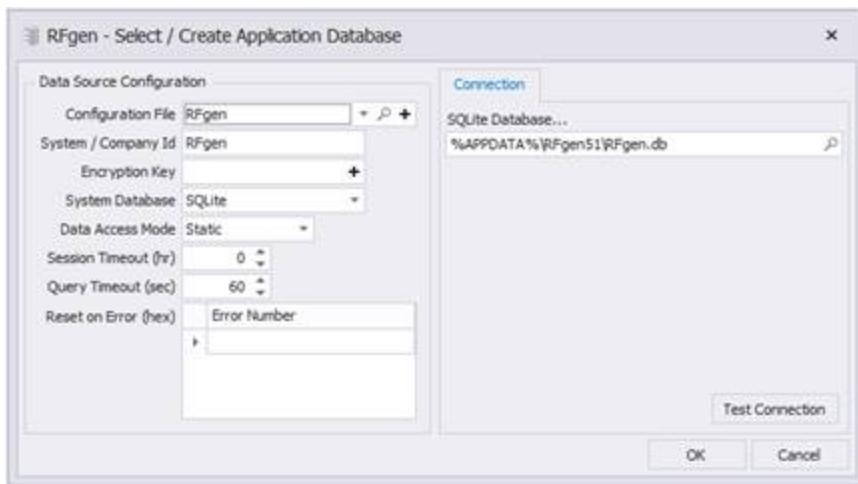
Connections Available Only in Dev Studio

- The [Download Enterprise Objects](#) option enables you to download all or specific types of objects from a connected database.
- The View Enterprise Objects option enables you to select and display previously downloaded objects.
- For details, see *Download Enterprise Objects* or *View Enterprise Objects*.

Configuring the RFgen Application Database

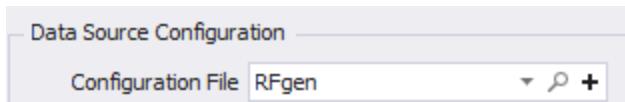
In order to provide a database for storing and maintaining your RFgen Mobile Applications (which help run your Transactions), you need to connect to a database application/server/system to your RFgen server-/system.

From the Mobile Development Studio or Mobile Unity Platform Console: Click on **Configuration > Application Database**.



Data Source Configuration Values

By default, a Configuration File called 'RFgen.rfc', defines the profile of the solution database, as shown below.



If you need to change the rfc file or select a different rfc file, you can use the list, search or plus (+) icons to browse to the %APPDATA%\ProgramData\RFgen51 folder.

The **System / Company Id** field is used to describe the owner of the configuration file. Since there are many configuration files referencing different databases for different customers or copies of the same customer's database, this field acts as the description.

An '**Encryption Key**' entry provides users the ability to encrypt their Application Database. This feature allows the database to be locked so that users may not view or modify Application objects or VBA scripts. When active, a unique key may be entered in the Application Database selection window to lock, encrypt, unlock, or decrypt the database.

- **To encrypt the database:** Enter a key (e.g. 'abcdef') in the **Encryption Key** textbox, and click the + icon.
- **To lock the database:** Display the panel again, remove the key and click 'Save'. The database is now locked. Applications will execute but may not be accessed.
- **To unlock the database:** display the panel and again enter the key, click 'Save'. The database will be unlocked.
- **To decrypt the database:** Enter the key, click the Encrypt button and click 'Save'. The database will be decrypted and unlocked. You must not export encrypted applications to a non-encrypted MDB. The server will prompt for the password and decrypt the exported application.

The **System Database** drop down field selects which type of database is to be used to host the solution objects.

The server supports Access, SQLite, SQL Server and Oracle as database containers. The solution stores the information to connect to these databases in an "rcf" file. You can also select these rfc files when exporting / importing to that database container.

Data Access Mode sets the cursor to either Static or Dynamic when retrieving data from the database. Usually, Static is best because it is fast and safe. However, if you have a database like Pervasive that will actually make a copy of the data from the database system to the RFgen system when using a static cursor, you can change this option to Dynamic, so performance will not suffer. Internally, this sets the cursor option to either adoOpenStatic or adoOpenDynamic.

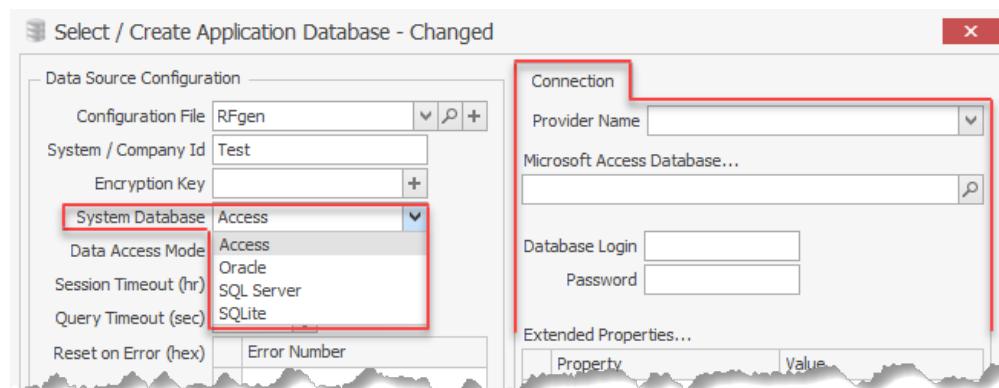
The **Session Timeout** value (in hours) will disconnect and reconnect to the database at the specified interval. This may be required if the database is configured to not allow a connection that never times out.

The **Query Timeout** specifies how long the server should wait before giving up on the ODBC driver to come back with a response.

Reset on Error is a list of hex values that if returned by the ODBC driver will cause a reset of the connection. The process for adding a value is to first get the error number from the error log. Example: the error log shows -21456327. Use the Windows calculator in Programmer mode, select Dec and Dword options, enter the number and if you need the negative sign use the ± button to change its sign. Then click the Hex option. You should get: FEB89A39. Enter this value into the box with a "0x" prefix like: 0xFEB89A39

Connection Tab

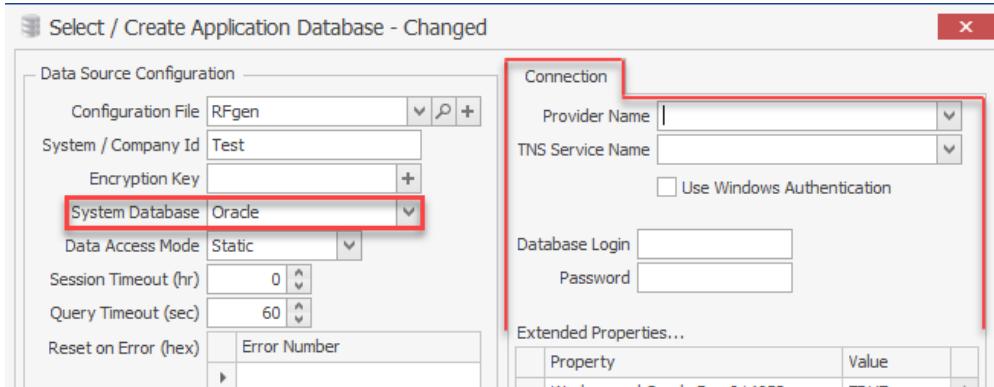
The System Database changes the configuration fields in the Connection Tab.



The **Provider Name** selection will depend on the type of database you want to use. All provider options must already exist on the server to be used .

For an **Access** database, select the appropriate **Provider Name** for the type of system (32 bit or 64 bit). The **Microsoft Access Database** path, **Database Login**, **Password** and **Extended Properties** are then used to make the connection. In the case of Access most of these fields are not necessary.

For an **Oracle** database, an ODBC is not used. The **TNS Server Name** points to the Oracle server. Also specify the Provider Name and review the Extended Properties for accuracy. The Use Windows Authentication option will take advantage of the Active Directory when connecting to the database.



For **SQLite** database connections just specify the DB file itself. There are no other settings. You can specify a location and name that does not exist and clicking the Test Connection button will create the database for you.

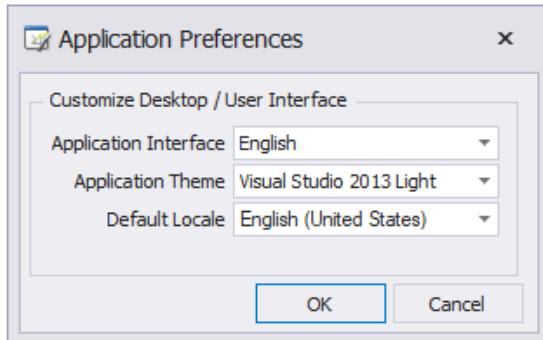
Click **Test Connection** to verify connectivity. If the database has not already been setup to support the solution tables they will be created at this time.

Save will also create what is necessary but won't test the connection. Clicking the **Test Connection** and/or **Save** button will check if the database came from an older release and ask if you want it upgrade.

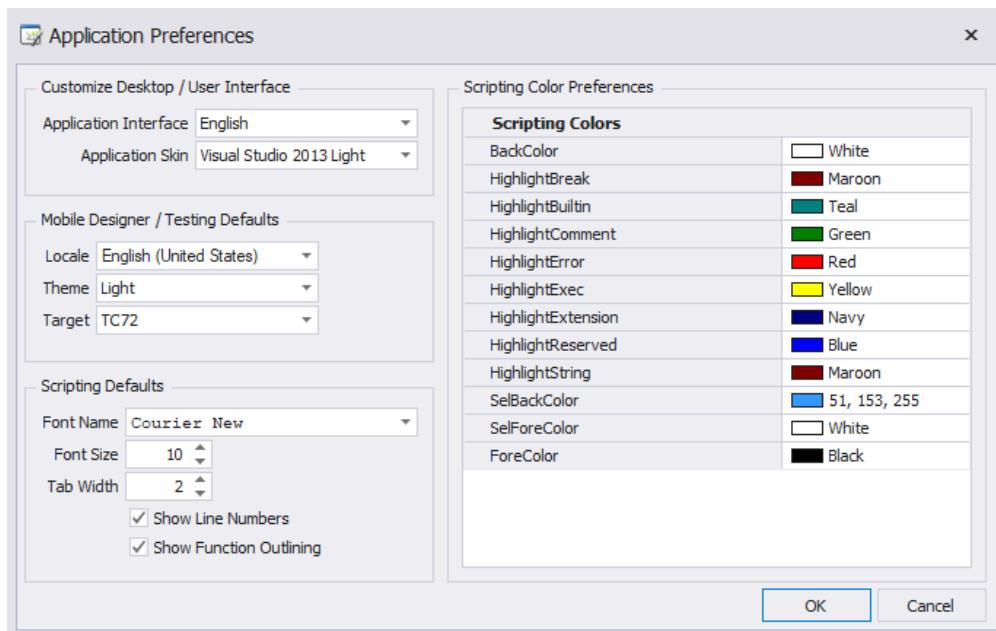
Configuring Application Preferences

Use this screen to set the Windows skin, language defaults, scripting defaults, and scripting colors for your RFgen Mobile Unity Platform console (Service console) and/or your Mobile Development Studio.

Example screen: Mobile Unity Platform Console > Configuration > Application Preferences Screen.



Example Screen: Mobile Development Studio > Configuration > Application Preferences Screen.



Customize Desktop/User Interface

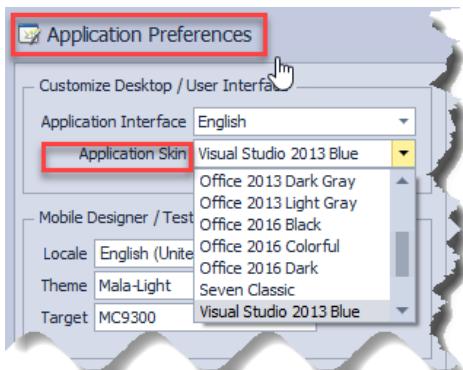
The **Application Interface** changes language the RFgen Mobile Development Studio, RFgen Server and its consoles into the language defined in this field. The options are: Arabic, Chinese, French, Japanese and Spanish.



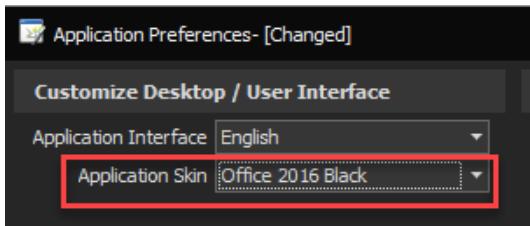
Application Preferences -- Application Skin

The **Application Theme** or **Application Skin** option changes the coloring theme of your Services Console or Mobile Development Studio windows according to selected Microsoft Office theme.

Example of Visual Studio 2013 Blue application skin:



Example, a Office 2016 Black application skin.

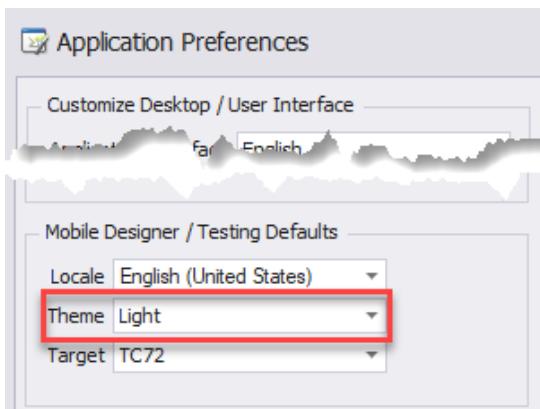


The Application Theme/Skin is available in the Service Console and Development Studio.

In the Services Console > Configuration > Application Preferences screen, the **Default Locale** translates your mobile applications into a specific locale. In order for the mobile application to appear in the specified locale, you must have designed the application with the appropriate text strings. For more details see [Language Translations](#).

Mobile Designer / Testing Defaults is only available in the **Development Studio > Configuration > Application Preferences** screen.

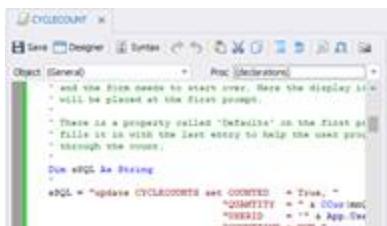
The **Locale** translates your mobile applications into a specific locale. In order for the mobile application to appear in the specified locale, you must have designed the application with the appropriate text strings. For more details see [Language Translations](#).



The **Theme** option sets the default theme that is used for the design and presentation of a Mobile Application.

(The theme you create or modify is under **Solution Explorer > Mobile Themes**.)

Scripting Defaults



The **Scripting Font**, **Scripting Size**, and **Tab Width** values affect the code windows.

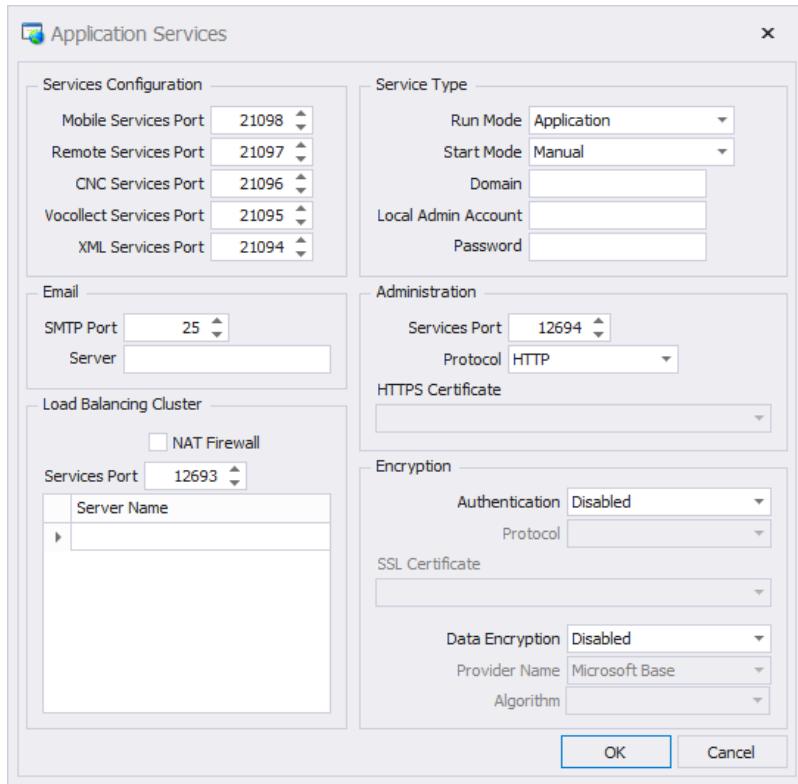
The **Show Line Numbers** and **Show Function Outlining** checkboxes turn on these settings.

Scripting Color Preferences

The **Default Colors** allow you to customize the coloring of various elements/types of scripts. For example, you can color all text strings as "Maroon".



Configuring Application Services



Services Configuration

The **Mobile Services Port** is the port used to transmit the graphical user interface between the development environment and the client. The default setting is port 21098.

The **Remote Services Port** is the port used to transmit the data portion of the session between the development environment and the graphical client. The default setting is port 21097.

The **CNC ServicesPort** enables the server to perform actions on the mobile client when the device is not in its cradle. The Client Network Control service enables the server to listen for client requests and synchronize information between the server and the client. The default setting is port 21096.

The **Vocollect Services Port**is the address that the Vocollect product uses to communicate with the server. Vocollect is a hardware solution that replaces barcode scanners for a speech-processing device that accepts the spoken word as data input. The default setting is port 15008. Note: create a Vocollect profile that uses 15008 for both the LUT and ODR services.

The **XML Services Port** is the address used for interfacing external / 3rd party data with server applications. The default setting is port 21094.

Load Balancing Cluster

Load Balancing is used to balance client connections between servers. Whenever there is a client connection request to the cluster, the load balancing logic determines which server has the lowest number of client connections and forwards the request to that server.

The **Load Balance Services Port** is used by the server to communicate with the other RFgen servers so that connected clients can share the client load. The default setting is port 12693.

Enter the **Server Names** on all the servers participating in the cluster so that they can share and load balance client connections between one another. This allows each load-balanced server to know who the other load balanced server is in the event one of them fails.

If you have your load balanced RFgen servers are behind a firewall, check the **NAT Firewall** box. This enables the resolution of the DNS server name(s) to the IP address of the servers if they are behind a firewall.

RFgen 5.2 can resolve to multiple IP addresses. If you had 300 devices on one server and added a second load balanced server, you would simply update DNS, and get your devices to connect to the new server. This saves the time and effort of having to update the devices individually.

License Usage: Even though each server is licensed for a specific number of licenses, the combined number of licenses is the total allowed even if one of the servers should fail. For a period of seven days the remaining servers will accommodate the total number of licenses before reverting back to the number of licenses it was originally designed to run.

This feature supports load balancing as well as server failover capability. Even if server number one is authorized for 10 users and server number two is authorized for 20 users, if either server goes down, the other will allow 30 concurrent connections for a period of seven days before reverting back to its original number of users. Adding additional servers authorized for zero users in this configuration would essentially add load balancing and hot spare failover capability.

Vocollect Connections: RFgen does not support load balancing of Voice Connections.

Email

There is a language extension called SMTP that can be used to send email under any circumstances required. The **Server** and **Services Port** properties here can be defaulted and automatically used by the SMPT object or left blank and entered as part of the scripting.

Service Type

The recommended configuration for running RFgen as a service is to use a local administrator to the PC where RFgen is installed.

The **Run Mode** lets you choose whether RFgen runs as an Application or a Service. **Application mode** runs a single instance of the RFgen server as a Windows program after a user logs into the RFgen server and the server service is launched (automatically or manually) from the desktop. **Service mode** runs the RFgen server as a background process and does not interact with the desktop. The start/stop of the RFgen server is not dependent on a user logging into Windows and having it launched from a desktop.

If you choose a **Run Mode: Application** and the **Start Mode: Automatic**, RFgen will start RFgen as an application when a user logs in to the Windows desktop.

If you choose **Run Mode: Service** and **Start Mode: Automatic**, RFgen services will be configured in the Windows services list. This means the RFgen server services will start automatically when the Windows server is started.

If you choose **Run Mode: Application or Service** and **Start Mode: Manual** neither will start until you start them manually.

The **Domain** field contains the name of the local PC.

The **Local Admin Account** and **Password** fields which are the credentials for a local administrator account. This typically avoids a problem where a domain admin account is believed to have enough rights but really does not.

Administration

The **Services Port** typically uses 12694. This port is used for communication between the server and the *Mobile Enterprise Dashboard*.

The communication **Protocol** supports HTTP or HTTPS. If using HTTPS, the certificate for HTTPS needs to be installed to the local computer certificate store in order for it to appear under HTTPS Certificate drop down list.

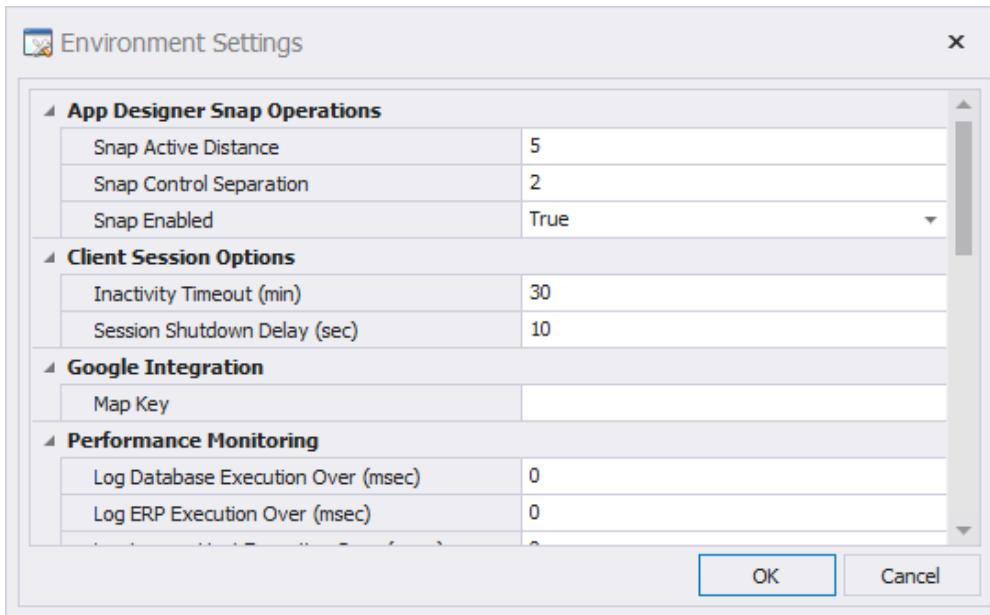
Encryption

The two categories are device authentication and data encryption settings.

Authentication is used to verify the user credentials beyond the RFgen login process. If Authentication is Enabled, set your Protocol as "NTLM, and SSL." When a RFgen client first tries to connect, it will pop up a dialog box to capture user information (user id, password, and domain.) An encrypted package of this information will be sent to the configured protocol. A core Windows service on the RFgen server will attempt to authenticate the login request and accept or reject the connection.

Data Encryption is used to secure the data being transmitted in a wireless environment. For the **Provider Name**, Microsoft provides several cryptographic choices and algorithms that are taken from what the operating system is capable of doing. The client must be configured exactly the same way as the server or it will not connect. For example, when Data Encryption is Enabled, the Provider Name choices are supplied by your operating system. Your selection of the provider, also changes the choices available in the **Algorithm** drop down list.

Configuring Environment Settings



Configure System Environment

Configure System Environment – App Designer Snap Operations

This configures the **Snap-to-Grid** feature which is automatically spaces and places graphical controls so they are aligned with other controls on a form. The grid displays a horizontal and vertical lines to help you align objects left (or right) sides and horizontal alignment.

The **Snap Active Distance** controls the sensitivity of the snap-to-grid and its detection of the object being moved. A high value makes it very sensitive and a low value is less sensitive.

The **Snap Control Separation** controls the distance between the x and y values of the grid. (I.e. The granularity of alignment in pixels.)

The **Snap Enable** turns the snap-to-grid function on/off. True = on; False=off.

Configure System Environment – Client Session Options

A **Client Inactivity Timeout (min)** of 30 minutes is set for network data collection devices (i.e., no activity at the device for 30 minutes will cause the device to be logged off). This setting may be modified as desired.

A **Session Shutdown Delay (sec)** of 120 seconds waits an additional 2 minutes after the mobile device sends the “disconnect” command, before ‘releasing’ the session. Sometimes a mobile device will terminate a session and reboot, but the user’s intention is to reconnect and keep working.

Configure System Environment – Google Integration



If you create an application that uses the Google Map, Route Planning, or Google Geo-Location and Tagging Support functionality, an activation license key from Google is required. To learn more about obtaining an activation license key, go to the following URL and click on "Paid".

<https://developers.google.com/maps/pricing-and-plans/>

and

<https://developers.google.com/maps/documentation/javascript/adding-a-google-map>

You will need to acquire an API key from Google and register your application in the Google API Console.

Note that the Map layout control in the Toolbox can be used to integrate the Google maps/Google GPS/Tagging Support into your application.

Configure System Environment – Performance Monitoring

If specific connectors are taking too long to process a request, these properties can be configured to capture processing requests that take over a certain amount of time. Set the property to zero to disable. Some connections usually take longer than others. For example, the database execution time will usually be significantly faster than a screen mapping connection or a Web Service connection which may be especially slower. Setting all the properties to the same number would not be appropriate.

To use these options, you'll need to configure a transaction management database. The following settings are all in milliseconds except for Log System Usage Statistics. The **Log System Usage Statistics** values are Disabled or can be set to record system statistics in 15, 30, and 60-minute intervals.

- Log Database Execution Over (msec)
- Log ERP Execution Over (msec)
- Log Legacy Host Execution Over (msec)
- Log Script Execution Over (msec)
- Log System Usage Statistics (min)
- Log Web Service Execution Over (msec)

Configure System Environment – Scanner Options

Scan for Pre-Angle and Post-Angle filter entries are character strings that are automatically sent from a scanner. They 'surround' the scanned data. They are optional and neither is required.

Common pre-angles include a location number, or perhaps an operator number. Common post-angles include control characters such as a tab or perhaps a carriage return-line feed. See your scanner documentation for information concerning how to establish these entries, or how to disable them.

Pre-amble and post-amble entries entered here are used by RFgen: (1) to identify scanner input, and/or (2) to automatically strip the pre/post entries from the character sequence received from a scanner. They will also cause a VBA Application 'OnScan' event to trigger.

Valid values are \n for new line, \r for return, \t for tab, \# where the # is any single character, and a group of characters like HELLO. If multiple characters are used then they are looked for as string text.

Configure System Environment – System Auditing and Logging

The following are all methods for archiving the backend system and therefore can be logged if strict compliance to regulatory law is required. There are three modes, *Disabled*, *Basic*, and *Extended*. This simply refers to the level of detail provided in the log. The logging options include:

- Archive SQL Update Requests?
- Archive Stored Procedure Calls?
- Archive Transaction Executions?
- Days to Store Archived Data?

Configure System Environment – System Options

Bypass Vocollect User Login allows a connecting Vocollect device to use the operator's name and menu stored on the Talkman device without RFgen verifying that the user is in the RFgen database or assigning them a menu.

Note: this is done on a space separated word basis so a phrase like "selected items" would not be affected as the word "selected" is not "select".

These are sample words a user might want to filter for: alter, analyze, associate, audit, backup, call, close, commit, connect, create, dbcc, delete, deny, desc, disassociate, drop, exec, execute, explain, grant, insert, intersect, join, kill, lock, minus, open, purge, recover, rename, restore, revoke, rman, rollback, rpc, select, shutdown, startup, truncate, update, union.

Additional Note: unless you are using a public kiosk where user tampering is of concern, this feature is not recommended.

Disable Development Studio Device Connections allows an application to be started on a downed production server and prevent mobile clients from connecting while an issue is being debugged. Otherwise the Application testing window would keep being interrupted by production clients.

Discard Form Data When Chaining sets the current application data to null when another application is called, rather than keeping the original data in memory (the default condition) should the calling application be returned to.

Enable European Currency Support enables support for international currency formats, by using the current system regional and language option settings (e.g. \$1,234.56 becomes \$1.234,56).

Enable Language Translations "True" will globally turn on language translations function in the Mobile Development Studio. allows for the names of the prompts on the screen to be used directly without the legacy RFPrompt wrapper function. So txtPlant.Caption = "Hello" is valid syntax.

Filter User Input for SQL Injection – This option will check the SQL statement for specific key words and remove them before sending the SQL request to the ODBC driver. All semi-colons (;) are removed; any single quote (') marks are doubled; any instances of a double dash (--) are replaced with a single dash; and any words specified in a user-created FilterInput.ini file stored in the RFgen installation directory will be removed. For example, if one line in the ini file had the word "select" then a user input of "select * from inventory" would become " * from inventory" with the select word missing.

Items Collections Are One Based tells RFgen, whether to begin the count of a series of values at one or zero. The default for Listbox and PanelList controls is a zero. Solutions upgraded from version 4.1 or earlier will have this turned on automatically.

Report All Errors During Application Testing – This option will bring immediate attention to a developer for incurred "soft" errors by displaying an error (i.e. message dialog) in the Mobile Development Studio. For example, each screen mapping command "SM.SendText" could timeout and fail, but the script will continue along (unless they are checking the return value for the function.) Turning this on will visually alert them of this type of "soft" error condition.

Require Double Click to Select From Lists for the user preference regarding lists, can be turned on or off if the users want to single click or double click list items. This will include controls like the list box and search lists as well.

Restrict Online Access to Known Devices controls whether the server automatically accepts or rejects online (Thin) client connection requests in the Mobile Development Studio > Devices > Authorized Devices screen, and the Mobile Unity Platform Console > Device Authorizations screen.

- If set to True, clients not previously approved by the RFgen administrator for connection to the server are rejected. An administrator will need to manually approve the connection in the Authorized Devices screen, and then the client will be allowed to connect next time a connection request is made.
- If set to False, this feature is disabled, and all known/unknown online client requests are accepted automatically.

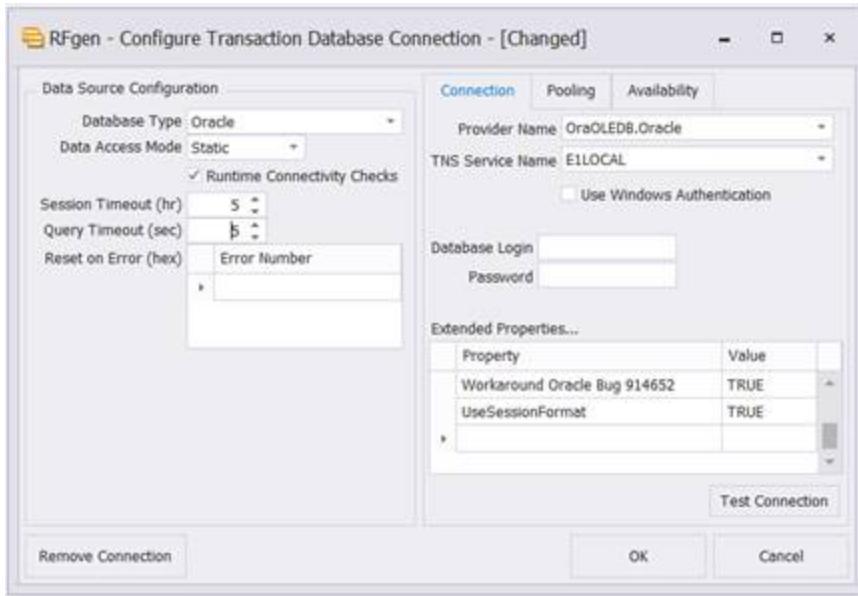
Note: This feature only applies to the automated accept/rejection of ONLINE clients (clients with a Thin profile). Offline clients (clients with Mobile profile) will still require manual approval for connection in the Authorized Devices screen regardless of whether this feature is set to True/False.

Use Legacy Message Box controls the which method the RFgen system uses with the VBA Language Extension App MsgBox function.

If this equals **True**, it will support the older method RFgen versions 5.1 and older used with the App.MsgBox when the customized Message Box text was used. If set to **False**, the newer method will be used in the App.MsgBox function. For more details, refer to the Developers Reference Guide topic for [App.MsgBox](#).

Configuring Transaction Management DB Connection

To configure transaction connections, select **Enterprise Connections > Transaction Management Database**.



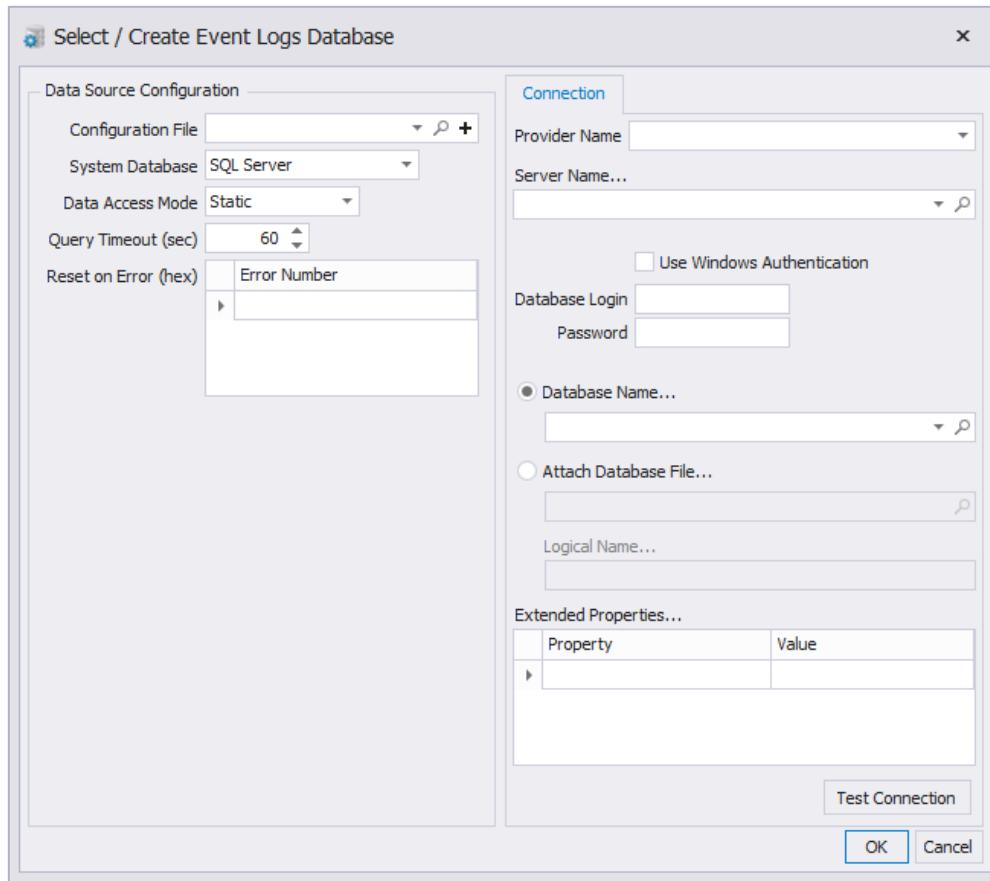
Configuring a Transaction Management database connection is the same as any other database connection. All the same fields apply. This connection is simply dedicated for the queuing process. See the section “Configuring Database Connections” for details.

There is one property that is unique: **Reset on Error (hex)** when turned on will reset the data connection under the following conditions. This should not be used unless deemed necessary.

If you submit a SQL statement to the DB, and an error code is returned, RFgen will look for the code in the *Reset on Error (hex) table*.

- If the error code is found, RFgen will try to reset the data connector. (RFgen won’t resend the submitted query that produced the error code.)
- If RFgen cannot find the error code in the *Reset on Error (hex) table*, RFgen won’t reset the data connector.

Create Application Event Database



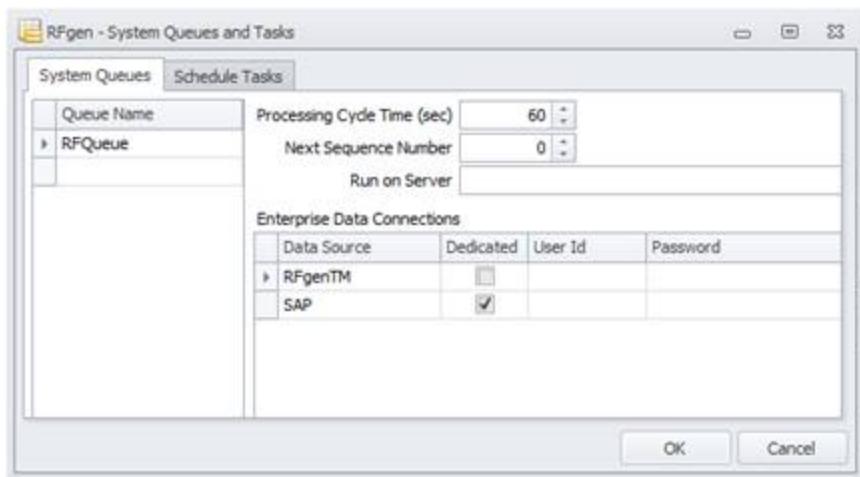
RFgen by default has event logging turned on and viewable from the Mobile Development Studio > Reports > System Event Logs screen or the Mobile Unity Platform Services console > Reports > Application Logs screen.

If you would like to have application events directed a database, [other than the default RFgen provides](#), use the Mobile Development Studio > Connections > Application Event Log screen to setup a connection to your own database.

The **Application Event Log > Data Source Configuration** is setup using the similar method used to connect to other Data Source systems.

The **Connection Tab** setup is also similar to setup of other database connectors. For examples, see the [Connection Tab](#) topic.

Configuring System Queues and Tasks



System Queues Tab

This table of queues allows for several queue processes to take place at the same time. The name RFQueue is the default name for the first thread that will process transactions.

The **Processing Cycle Time** number is the number of seconds that will pass before the system checks this queue for transactions and if one or more are found the entire queue is processed.

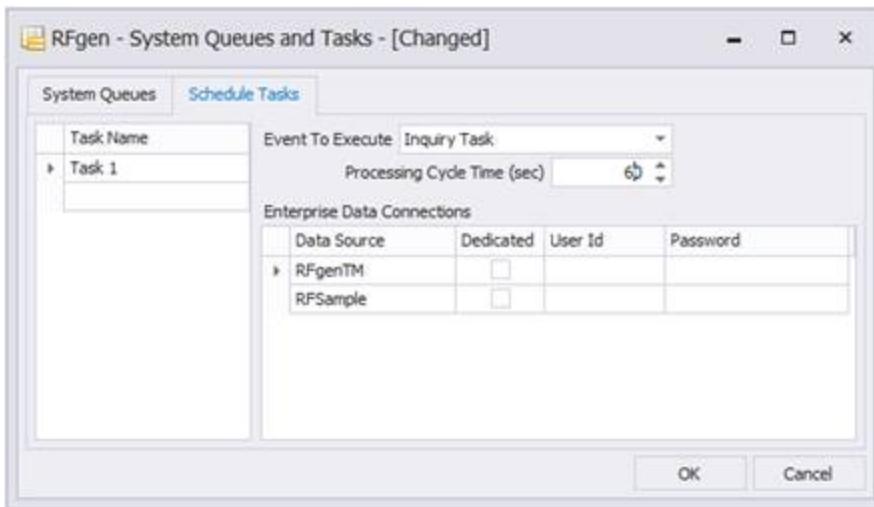
The **Next Sequence Number** option will allow the user to change the sequence number used when queuing or making entries in the logs.

The **Run on Server** is used to specify a server to prevent your queued transactions and events from running on all servers that are actively used for load balancing. If this field is left blank, RFgen will not check for any server changes and run all processes on all server(s) connected to the RFgen database. If the server IP address or "Local Host" is present, then RFgen checks which server to use and runs processes against the assigned server.

Enterprise Data Connectors

Each data source can be configured separately for each queue meaning that each queue can have either a dedicated connection to a specific data connector or it can share a limited pool of handles to that data connection. This is the **Dedicated** check box option. In either case a user and password could be specified for the queue to use when it communicates with that connection.

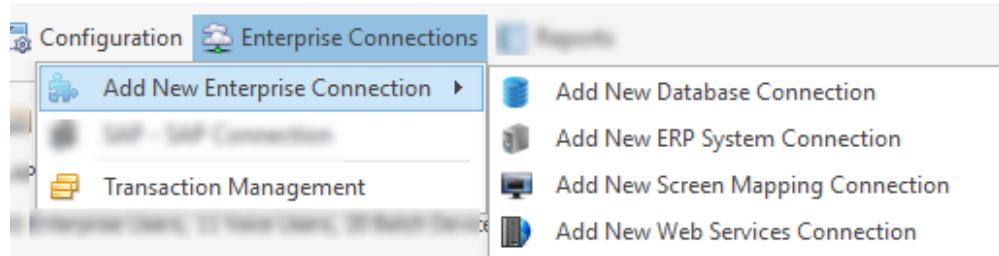
Note: Each queue process uses one client license because it is, in essence, an automated user performing tasks against the server and the backend systems. For example, a 10-client licensed system with three separate queue processes will only allow up to seven concurrent devices.



Schedule Tasks

The **Schedule Tasks** tab allows the user to specify a task name and then assign it to a Timed Event macro chosen from the **Event to Execute** drop down. The **Processing Cycle Time** is in seconds and determines how often the server will run the Timed Event macro. Just as with the queues each data source can be configured separately as well as taking a client license. See the above section for more details.

Add New Enterprise Connections



To add a new database connection to a data source such as SQLite, DB2, ODBC, OleDb, Oracle, SQL Server, or MS Access, see [To Add an New Data Source Connection](#).

To add a new Enterprise Resource Platform (ERP) such as Oracle EBS, see [To Add/Connect to JD Edwards](#), or for SAP see [Configuration for SAP](#).

To add a new Screen Mapping connection from the RFGen server to host systems (for legacy systems), see [Connect to a New Host](#).

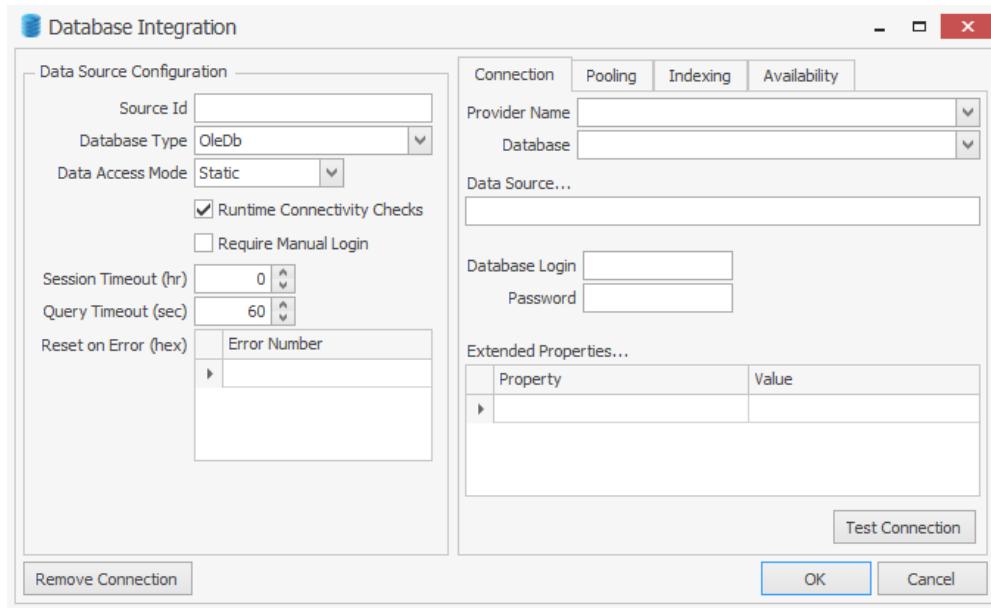
To Add a New Web Services Connection see [Adding to a New Web Service Connection](#).

Adding a New DataSource Connection

To add a connection to a new data source, from the **Mobile Development Studio** or **Mobile Unity Platform Services Console**, click on **Enterprise Connections > Add New Database Connection**.

RFgen supports connection to home-grown or commercial databases that are SQL-compliant.

Use the **Data Integration > Data Source Configuration screen** to configure how RFgen will access the data hosted by a commercial or home-grown database and set the timeouts should a non-optimal event occur.



The **Source Id** is the name of data connection. Spaces and extended characters are not recommended for this field.

The **Database Type** drop down field selects which type of database is to be used to host the solution objects. Changing this value changes the content of the Connection screen, to show database specific configuration fields.

Data Access Mode sets the cursor to either Static or Dynamic when retrieving data from the database. Usually, Static is best because it is fast and safe.

However, if you have a database that will make a copy of the data from the database system to the RFgen system you can change this option from Static to Dynamic, so performance will not suffer.

Internally, this sets the cursor option to either adoOpenStatic or adoOpenDynamic.

The **Runtime Connectivity Checks** and **Require Manual Login** are enabled if checked.

The **Runtime Connectivity Check** is a process that will verify whether the datasource is still connected. If the user doesn't have the permission to access a table, the connection will reset (via a select count process) and go into a loop.

If **Require Manual Login** is checked, the client will not be allowed to connect to the database unless the user on the client is authorized.

For example, with JD Edwards, if **Require Manual Login** is checked, when the user logs in to process a transaction, his/her login user ID is checked against the Named User list.

If the Named User list is not setup correctly in RFgen (or JD Edwards), or, if the user isn't on the Name User list, a database error may result.

If **Require Manual Login** is unchecked, the client will be allowed to connect with the database automatically.

The **Session Timeout** value will disconnect and reconnect to the database at the specified interval. This may be required if the database is configured to not allow a connection that never times out. This value is in hours.

The **Query Timeout** specifies how long the server should wait before giving up on the database driver to come back with a response.

Reset on Error (hex) is a list of hexadecimal values that if returned by the ODBC driver will cause a reset of the connection. The process for adding a value is to first get the error number from the error log.

For example, the error log shows -21456327. Use the Windows calculator in Programmer mode, select Dec and Dword options, enter the number and if you need the negative sign use the ± button to change its sign. Then click the Hex option.

You should get: FEB89A39. Enter this value into the box with a "0x" prefix like: 0xFEB89A39

Connection, Pooling, Indexing, and Availability Tabs

See the [Connections Tab](#) topic for details on creating a connection to the source provider.

See the [Pooling Tab](#) topic for information on setting up users in a pool.

See the [Indexing Tab](#) topic for information on configuring Discovery and Indexing schemes to optimize performance when extracting data.

See the [Availability Tab](#) for information on scheduling downtime so RFgen doesn't attempt to connect if the source is not available.

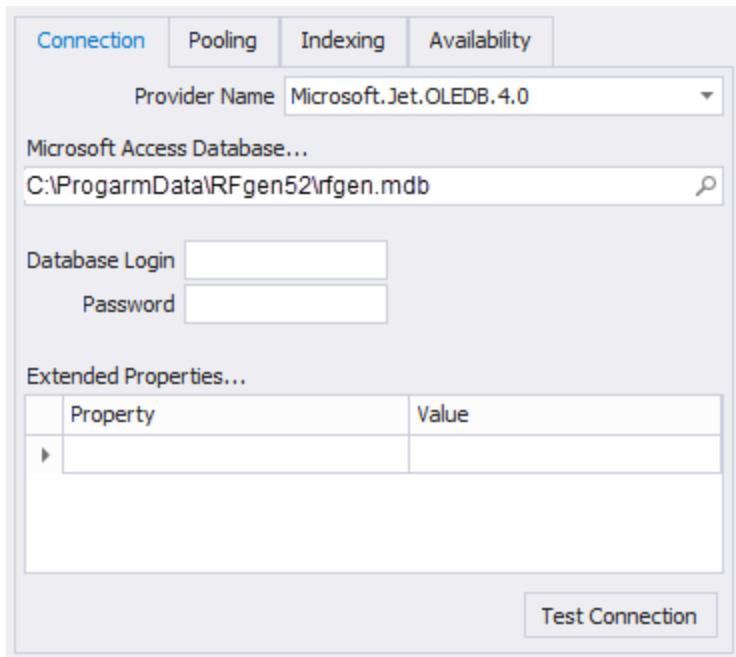
Connection Status Indicator

When the server connects to a database, it will display a connection indicator at the bottom of the Mobile Development Studio window.

If a  red circle appears in the indicator, a valid icon has not been made.

To troubleshoot an invalid database connection, click on the Mobile Development Studio Reports menu bar selection, then check your Event Logs to see if a message has been generated. Most likely, a problem was encountered with your Data Source entry.

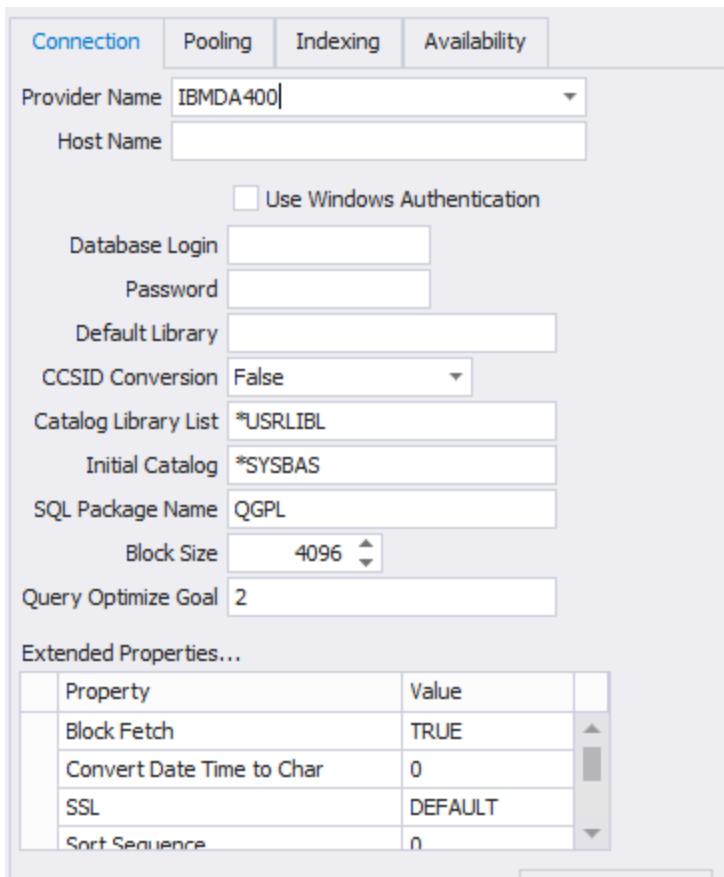
Connection Tab



The **Provider Name** selection will depend on the type of database you want to use. Note that these providers are not necessarily installed. All provider options must already exist on the server to be used.

For an Access database select the appropriate Provider Name for the type of system (32 bit or 64 bit).

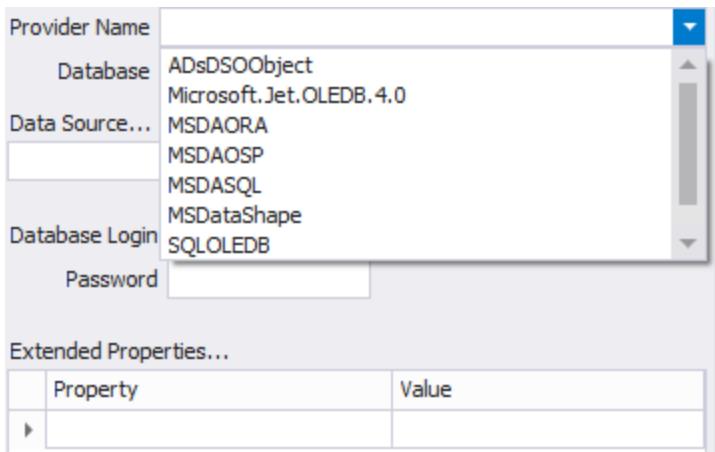
The path, login, password and extended properties are then used to make the connection. In the case of Access most of these fields are not necessary.



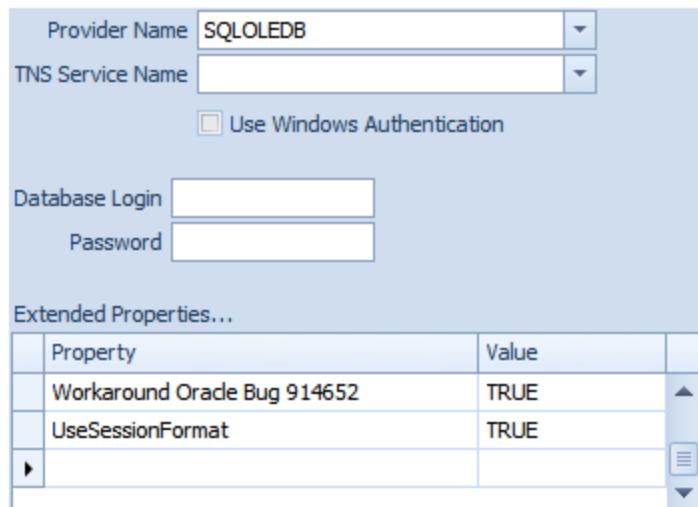
In the case of DB2, you can specify all the same settings that would normally go in the ODBC DSN entry for the iSeries Access driver.

For ODBC DSN entries that come from Control Panel / Administrative Tools, this option assumes that the connection has already been established the server will just reference what is setup in Control Panel. This option should be used if other programs also rely on the same database connection and it is easier to maintain the settings in one place rather than many.

The OleDb option is the most generic method of connecting to a database. The Provider Name shows many options, most of which need to be manually installed (acquired by the manufacturer) before the server can take advantage of them.



In the case of Oracle, ODBC is not used but the TNS Server Name points to the Oracle server. Also specify the Provider Name and review the **Extended Properties** for accuracy. The Use Windows Authentication option will take advantage of the Active Directory when connecting to the database.



For SQL Server specify the Provider Name, Server Name and Database Name. The Use Windows Authentication option will take advantage of the Active Directory when connecting to the database. If you want to connect directly to the MDF file itself, specify the Attach Database File option and locate the database file directly. The Logical Name is typically the filename without a file extension and should not be necessary.

The **Extended Properties** are usually not required.



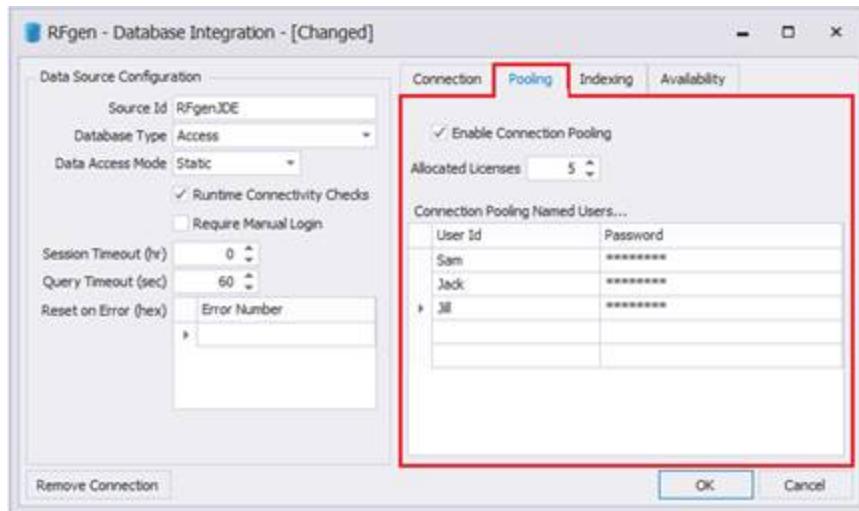
For SQLite database connections just specify the DB file itself. There are no other settings. You can specify a location and name that does not exist and clicking the Test Connection button will create the database for you.

Finally click on the **Test Connection** button to verify connectivity. If the database has not already been setup to support the solution tables they will be created at this time. Clicking the Save Changes button will also

create what is necessary but won't test the connection. Either button will also notice if the database came from an older release and ask if you want it upgraded.

Pooling

Pooling refers to configuring multiple database connectors into one licensed connection for license conservation purposes. Pooled sessions can setup for Named Users if the pooled connection is for Oracle JD Edwards.

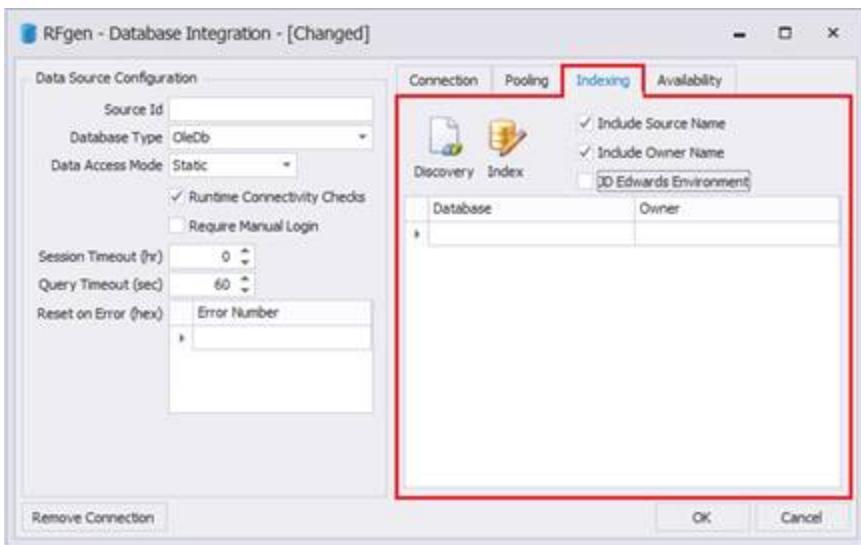


The **Enable Connection Pooling** checkbox turns on Pooling.

As **Allocated Licenses** are incremented, a Pool(n) session appears for each. Enter a User ID and Password for each in the corresponding boxes (if different from the defaults).

The connection pooling User ID and Password fields contained in this window are for allowing users to log in under non-default settings. As each session is taken from the pool (when simultaneous access is required) the next pool's settings will be used. For example, if your system only allowed two connections with a particular User ID, Pool (3)'s User ID and Password could be specified and the first two will be taken from the default information.

Indexing tab



This has advanced features for connecting to the database.

The **Discovery** option will attempt to fill in the grid automatically with all the databases and owners ultimately indexing everything in the database.

The **Index** icon means the server performs the indexing when the user saves the connection. The server will index tables within those additional databases so they can be referenced by name only. For example, F0005 is a control table in the database. Using this indexing it may be accessed simply as F0005 and the server will qualify it with database.owner.F0005 internally before SQL execution.

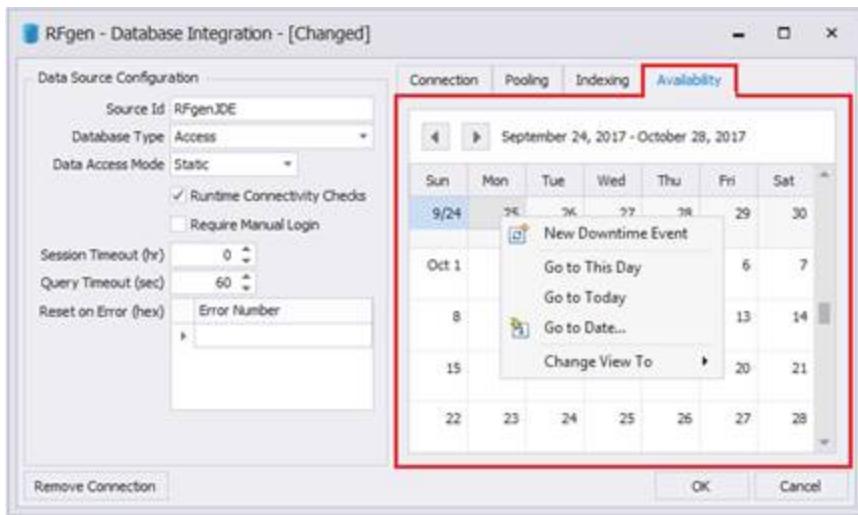
The **Source or Owner Name** information are options to more uniquely qualify the tables or database structure in case other connected databases have tables that are named the same when downloading the tables. For example, your main database may be SQL Server or Oracle, but you have a need to connect to tables contained in other databases or entirely different ERP or legacy systems. Databases may be different in type, as long as they are SQL compliant.

The **Database** and **Owner** grid allows the user to restrict which tables are indexed for a specific data connection. In this grid, specify the list databases and / or a list of owners of the tables that are necessary for the data collection application.

The **JD Edwards Database Indexing** option is a JDE specific option that will query certain JDE tables to determine how the system is actively configured for the selected environment and then index specific JDE tables.

Availability Tab

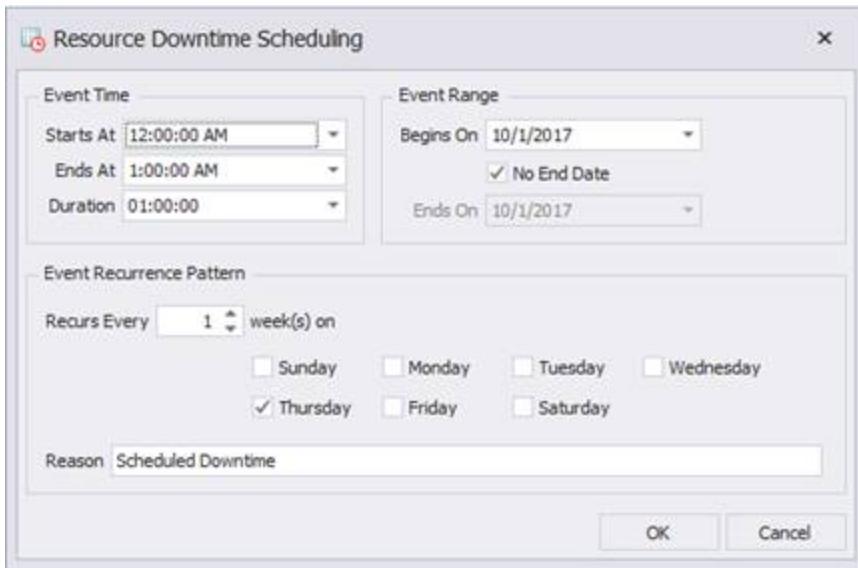
The **Availability** option is used to schedule down time for maintenance purposes.



To schedule downtime, right-click on the date or days in the calendar and select the appropriate item from the menu.

In this example, the **New Downtime Event** was selected.

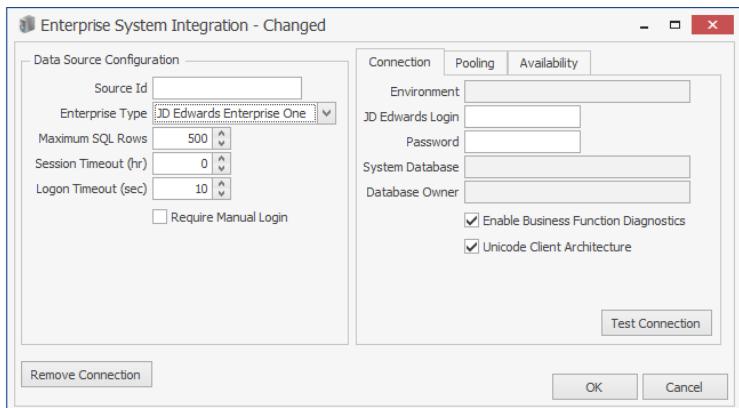
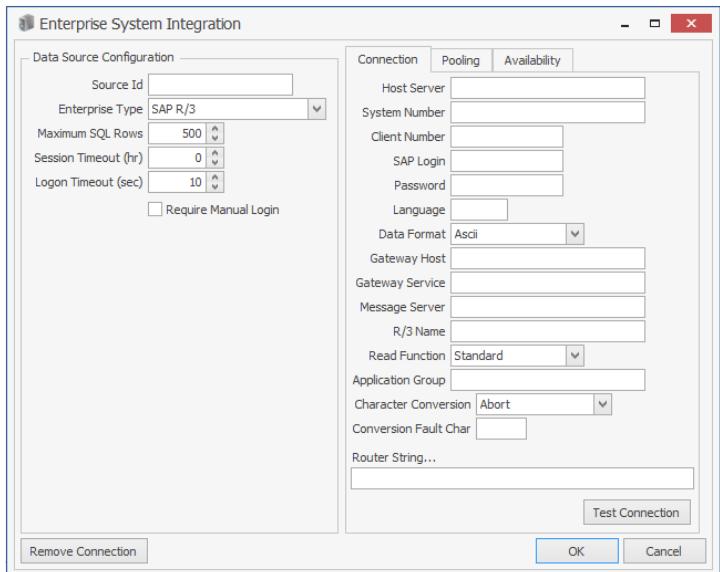
In the Event Time box, the connection will be unavailable for 30 minutes, every Thursday, between 12 AM and 1 AM beginning Oct 1, 2017 and will reoccur until an End Date is supplied.



Add an New ERP System Connection

To add an ERP connection from the Mobile Development Studio or from the RFgen Mobile Unity Platform Services Console select **Connection or Enterprise Connection > Add New Enterprise Connection > Add New ERP System Connection**. The Enterprise System Integration screen displays.

Depending on your Enterprise Type, your screen will look different than the examples shown below.



If you are connecting to Oracle JD Edwards, see [Configuring for Oracle JD Edwards](#).

If you are connecting to SAP, see [Configuring for SAP](#).

Related Topics

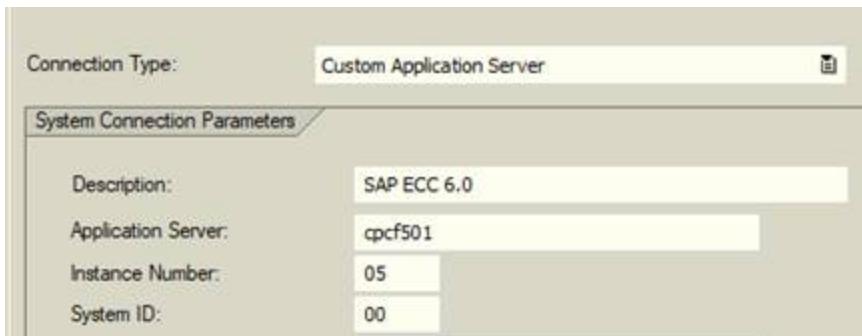
See [Pooling Tab](#) for information for setups that have pooled users as connected clients.

See [Availability Tab](#) for information on the scheduling downtime.

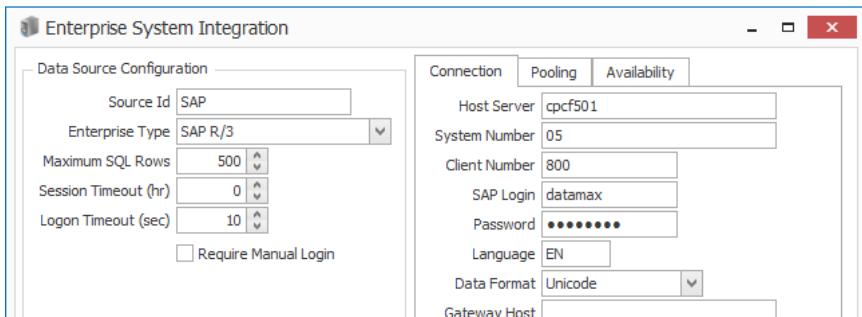
Configuring RFgen to SAP

RFgen is a zero-footprint integration to SAP, meaning there is no custom coding or additional software to install on your SAP server.

RFgen makes the connection to SAP using the SAP Connection Properties found in your SAP Logon screen.



Example of an SAP Connection Parameters Screen



Example of Corresponding RFgen Data Source Screen

From that connection, RFgen can download the schema for any remotely enabled function module/BAPI.

Setup Requirements

RFgen requires a 'User ID' to make the connection to the SAP server. A System ID user is recommended for this purpose. For example, you can enter "RFgenConnect" or even "RFgen".

Requires a non-expiring password

Must be assigned authorization objects S_RFC and S_TABU_DIS

To configure an SAP connection

Select **Enterprise Connections > Add New Enterprise Connection > Add New ERP System Connection**.

First enter a **Source ID** name and then change the **Enterprise Type** to *SAP R/3*.

Next, enter a value for the **Max SQL Rows**. The **Max SQL Rows** prevents a 'lock-up' scenario in case a query was bringing back too many records by accident. ERP systems can typically have millions of records and this could prevent a frozen client. The command `ERP.ReadData` can perform SQL statements against the SAP connector.

The **Session Timeout** value (in hours) will disconnect and reconnect to the ERP at the specified interval. This may be required if the ERP is configured to not allow a connection that never times out.

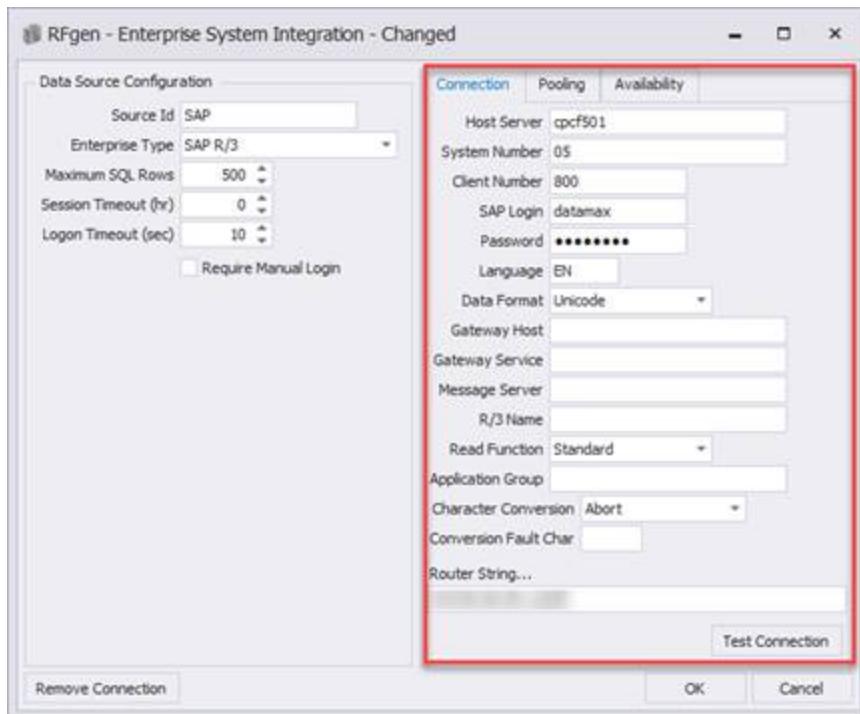
The **Logon Timeout** (seconds) will return with a failure in the error log if a request to log on to SAP never comes back.

You can select a **Manual Login** if desired.

Connection Tab

See [Connection Tab for SAP](#) information on completing the Connection tab fields.

Connection Tab - SAP



For SAP, the **Host Server** is the application server. Enter the **System Number**, **Client Number**, **SAP Login**, **Password**, and **Language** with the same data as stored in the SAP GUI Logon Pad application. The System Number also dictates the port number being used to communicate with SAP. For instance a 01 entry would possibly set the port to 3301 where a 05 entry would set the port to 3305.

The **Data Format** option tells RFgen how to interact with the system, either by using Unicode-formatted communication or non-Unicode (ASCII) formatted data.

The **Gateway Host**, **Gateway Service**, **Message Server**, **R/3 Name**, **Application Group** and **Router String** are optional parameters. If your Logon Pad requires these settings, then add them here for RFgen.

Read Functions has two options: *Standard* and *BBP*. The Standard option executes SQL queries using the RFC_READ_TABLE function module; the BBP option executes SQL queries using the BBP_RFC_READ_TABLE function module. Only used with SAP HANA.

For SAP load balancing, configure the following fields: **System Number**, **Client Number**, **User ID**, **Password**, **Message Server**, **R/3 Name**, and **Application Group**. Leave the **Host Server** blank since the Application Group setting will distribute requests to multiple host servers. In some cases, you must leave the System Number blank as well.

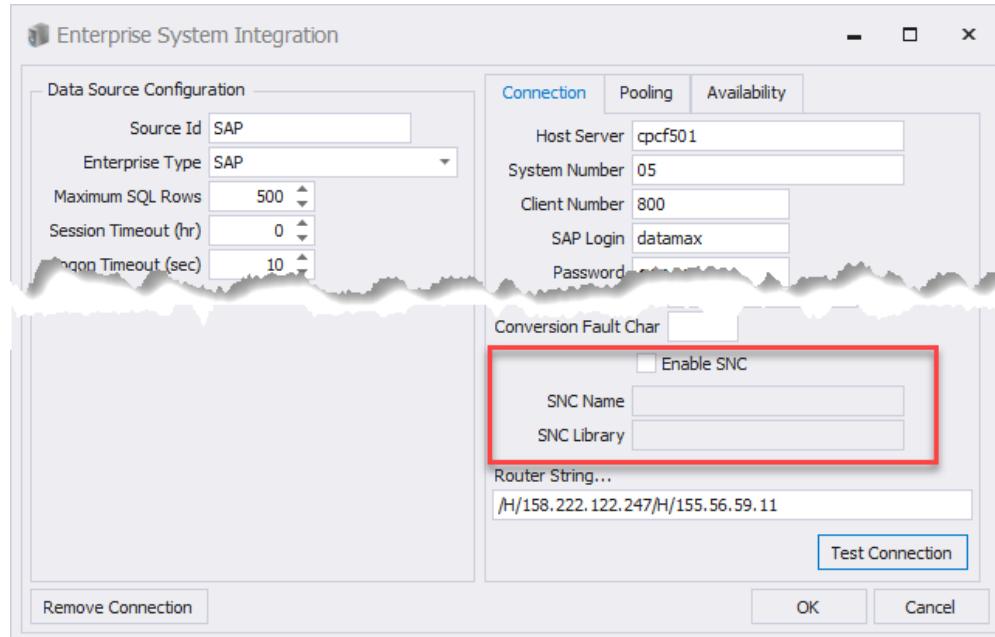
The **Character Conversion** and **Conversion Fault Char** properties are designed to handle problems when SAP sends data in a different code page than what RFgen is configured to display. If the text does not have a translation, RFgen can be configured to abort the conversion, copy the bad character, or replace the character with the character entered in the Conversion Fault Char property.

Related Topics

See [Pooling Tab](#) for information for setups that have pooled users as connected clients.

See [Availability Tab](#) for information on the scheduling downtime.

SAP Data Encryption



If you have users (clients) who collect and process data that is processed at sites outside your firewalls, you may want your data encrypted for security purposes. For example, a salesperson who processes sales orders offsite; A warehouse worker who inventories lumber at an outdoor storage facility or workers who are connected but through the cloud.

RFgen supports the SAP Secured Network Communications protocol which secures the communication paths between various SAP system client server components. With SAP SNC, you receive application-level, end-to-end protection for data communicated between two SNC-protected components.

The SNC function is provided through the SAP Cryptographic Library which delivers encryption functions in SAP systems. For more information on the use of or implementation of the SAP Cryptographic Library, see SAP documentation.

To enable a Secure Network Connection (SNC) for SAP

1. Click on **Connections** > [your SAP Enterprise Connection] from the Mobile Development Studio.
Or click on **Enterprise Connections** > [your SAP Enterprise Connection] from the Mobile Unity Platform Console.

Note: If you are creating an Enterprise connection for the first time, refer to the online help for [Adding a new enterprise connection](#). Then return to this process.

2. The Enterprise System Integration screen displays.
3. In the Connections tab, check the Enable SNC box.
4. Enter the SNC name.
5. Enter the path to the SAP Cryptographic Library (where SNC dll is stored).
6. Click on **Test Connection**.
7. If your test is successful, click **OK**.

To Configure for JDE

A RFgen/ERP solution set consists of the following items:

- The “engine” provided by RFgen Mobile Development Studio or Mobile Unity Platform product
- the RFgen Application Database
- the customer’s datasets.

We used the *Mobile Development Studio* to perform our setups, but the *Mobile Unity Platform Console* can similarly be used as well.

Once RFgen is installed (and tested), setup the ERP-specific source dataset. When setting up a new RFgen source dataset it does not matter what your existing database type is as it can be updated post installation.

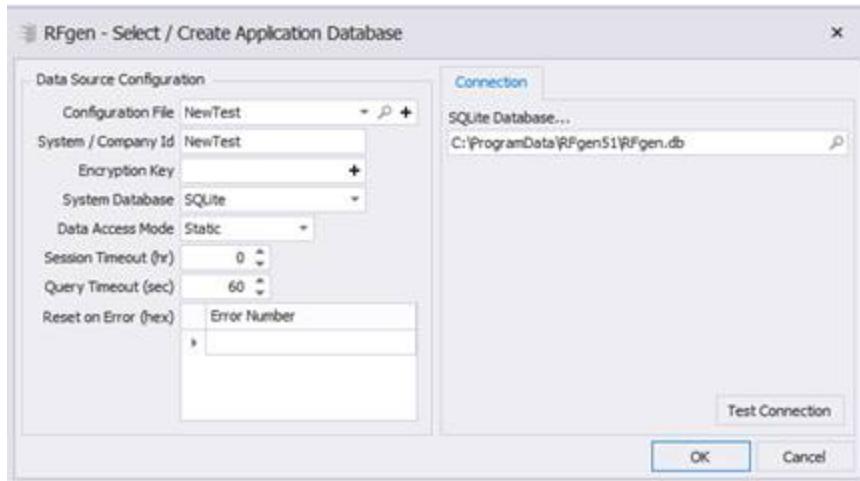
Step 1: Obtain Open Source Database for JDE.

The most current version of the RFgen/JDE Opensource database is always available from the support team at RFgen (support@rfgen.com) for customers and partners currently under a support contract with RFgen.

Load this database to the RFgen system's folder where the RFgen admin will have read/write privileges. For new customers, all of these steps are normally completed by your RFgen consultant.

Step 2: Set the RFgen Application Database.

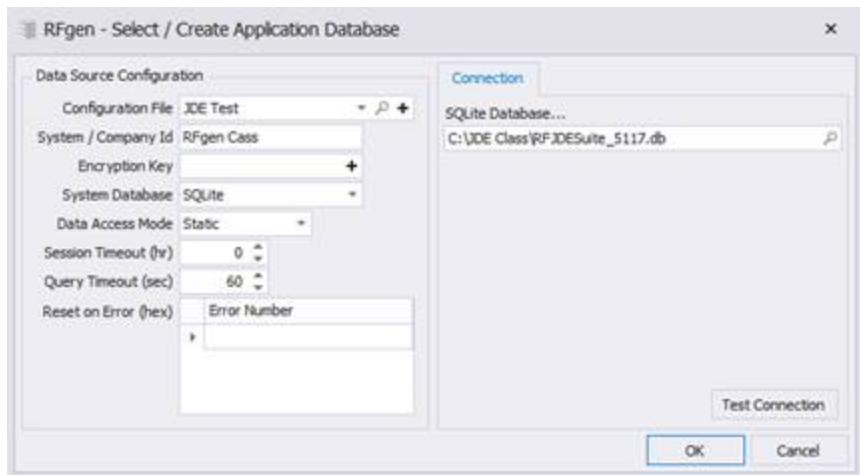
- a. Select **Configuration > Application Database** from the Mobile Development Studio menu. Your display may look different from the image below.



- b. Enter the values as suggested below.

- Configuration File: Provide a new configuration file name by selecting the '+' sign. We recommend a name that has some type of meaning. In this case we will use 'JDE Test'
- System/Company Id: Place an appropriate name into this field. A RFgen Partner might place their customer's ID or name into this field. In this example we will use 'RFgen Cass'.
- Encryption Key: Do not setup an encryption key at this time as your starting environment is not encrypted.
- System Database: Select the database type. Currently we are releasing our starting databases as .db datasets, so select SQLite.
- Data Access Mode: Can be Static or Dynamic. Use the default "Static."
- Session Timeout (hr): This will disconnect and reconnect to the ERP at the specified interval. This may be required if the ERP is configured to disallow a connection that never times out. The default is 0 hours.
- Query Timeout(sec): Keep the default as is.
- Reset on Error (hex): Leave this left blank for now.
- SQLite Database: Select the RFgen Application Database from Step 1.

c. Click **Test Connection**. If the "Connection successful" message displays, click **OK** to exit the message, then click **OK** to save changes.



d. At this point RFgen will attempt to make the following connections:

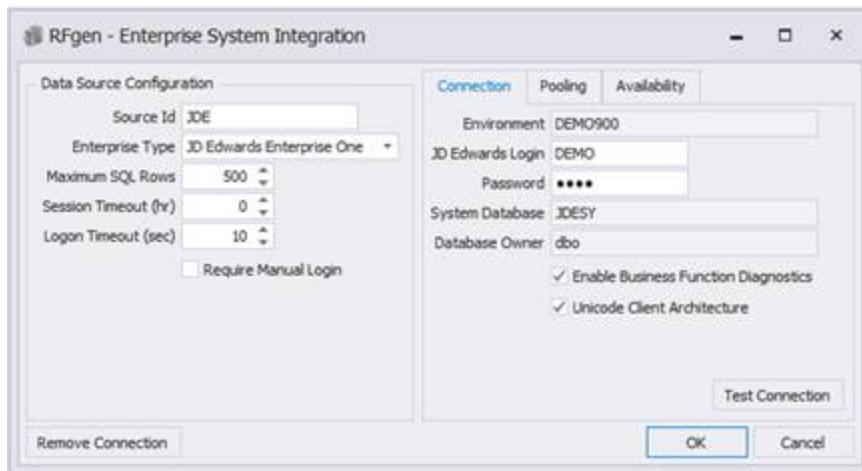
- RFgen (This is the base set.)
- RFgenTM (This is the optional transaction manager.)
- JDE (This is the JDE connector.)
- JDE_DEMO (This is our example, JDE database.)

e. Since our setup is still incomplete, the connection status icon displays a warning.



Step 3: Set the JDE Connector

- a. Select **Enterprise Connections > JDE – JD Edwards Connection** from the menu. Your screen may look different than the one below.



- b. Enter the values as suggested below.

Source Id: For JD Edwards, please leave this value as 'JDE'.

Enterprise Type: For JDE Enterprise environments, leave the value as 'JD Edwards Enterprise One'.

JD Edwards Login and Password: Enter a JDE Administrator during the original install. This may be changed later.

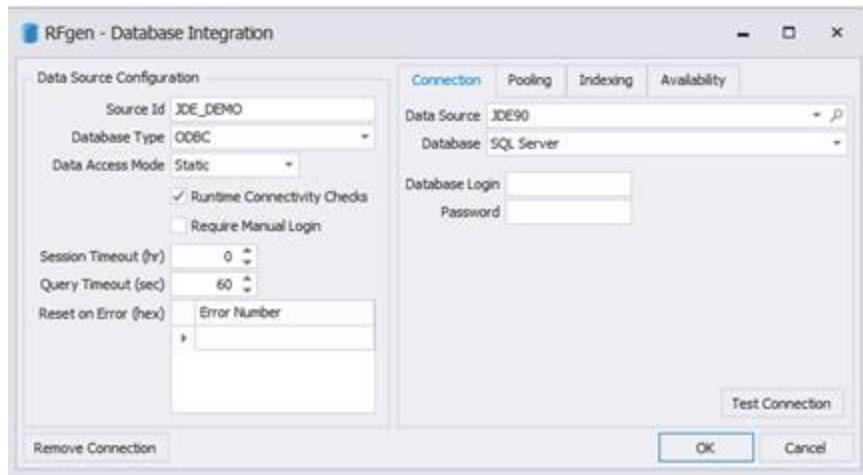
NOTE: The Environment, System Database and Database Owner are pulled from the customers JDE.INI file and cannot be changed.

The other default values should display and match the screen above for original setup. Be sure the Require Manual Login is not checked on.

- c. **Test and Save.** Selecting the 'Test Connection' button will connect to the user's JDE system using the JDE Development Client (aka JDE FAT client) that was installed on the system. If you receive the 'Connection successful' message, click OK to exit the message, then select the OK button to save the change.

Step 4: Set the JDE Data Connector

- a. Select **Enterprise Connections > JDE_DEMO** from the menu.

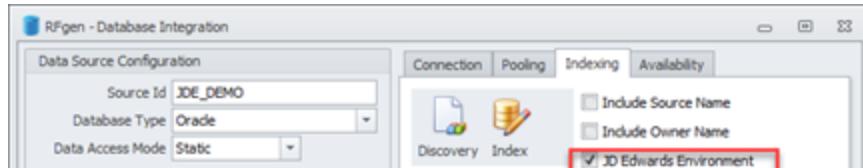


b. The Source Id is the name you wish to use. The Database Type and Source need to be correctly entered based on the customer's JDE Development client database connection information. The specific questions will differ based on the database type selected. Most common for JDE is either an ODBC connection (as noted above) or an Oracle connection. Contact your JDE database administrator and/or your Oracle administrator for this information.

c. To validate your settings, click **Test Connection**.

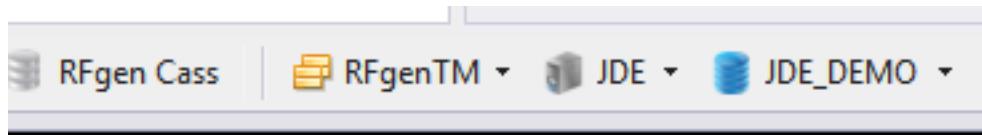
d. After the "Connection successful" message displays, click on the **Indexing tab**.

e. As JDE contains multiple databases, select the **JD Edwards Environment** checkbox first.



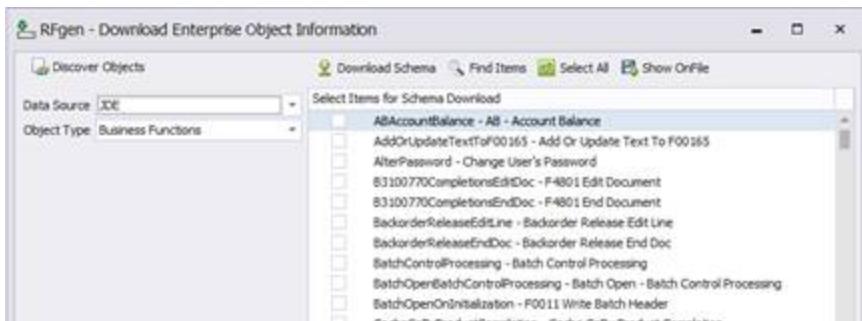
f. Then click the **Discovery** icon. This will list the JDE datasets. After the list is populated, select the Index icon, which will index those datasets to allow RFgen to quickly read the JDE data.

g. Upon completion, click the **OK** button to save changes. The footer should show both connections are valid.



Step 5: Load / Update JDE Business Functions

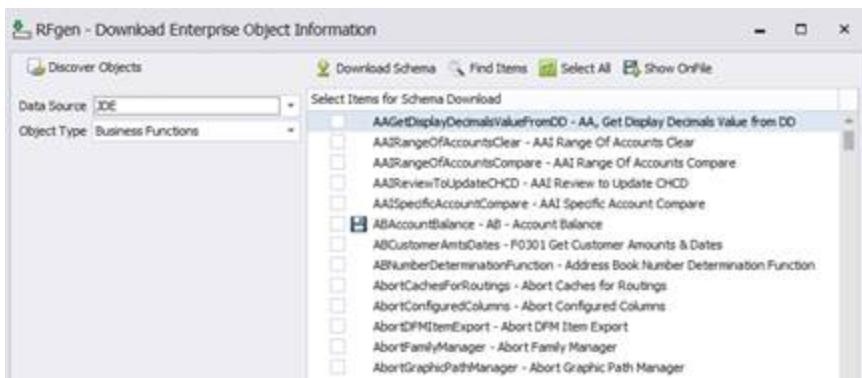
- Select **Enterprise Connections > Download Enterprise Objects** from the menu.
- Select **Data Source: JDE**, then select, **Object Type: Business Functions**.
- Tap the **Discover Objects** icon.



d. This will open JDE and pull all of the JDE BSNFs from the customer's JDE-connected JDE environment. Upon completion, the system will show all BSNFs in alphabetical order.

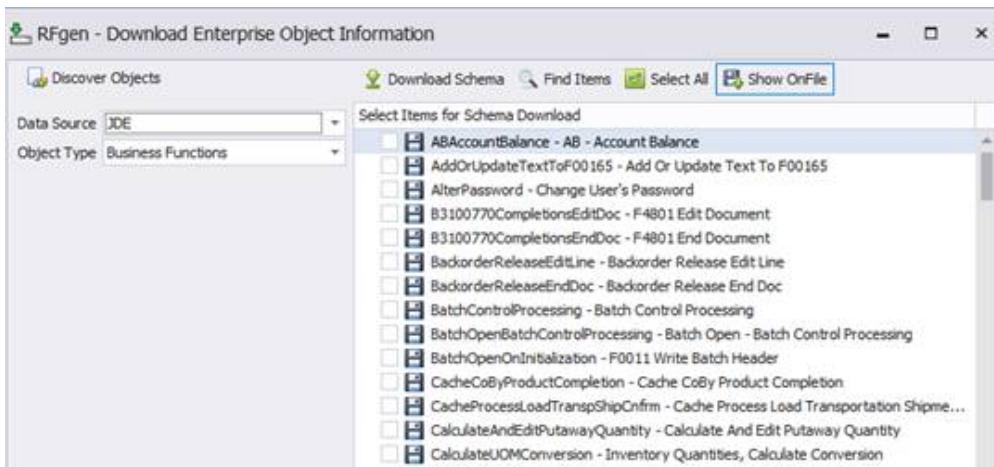
Step 6: Update RFgen to Match Current JDE BSFNs

In the example below, note the ABAccountBalance item contains a small disk icon.



This disk icon shows that this BSNF is the one currently used by your RFgen environment.

a. For the original setup, select the **Show OnFile** icon.



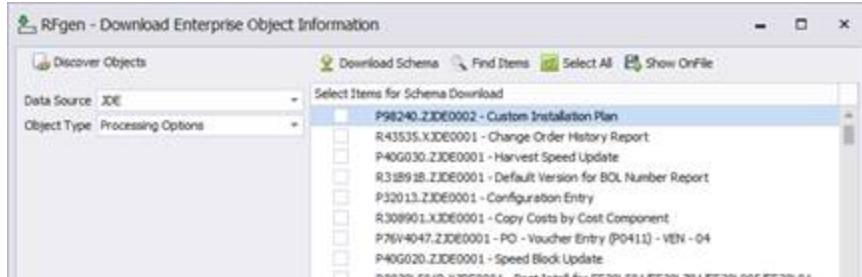
b. With ONLY THOSE ONFILE SHOWING, click the **Select All** icon, which checkmarks all the listed items.

c. Select the **Download Schema** icon. This will update RFgen to match the current JDE BSFNs.

Note: This step should be completed with each JD Edwards new release, or anytime an effected BSFN is changed by JDE.

Step 7: Load JDE Processing Options

- a. Select the menu item Enterprise Connections | Download Enterprise Objects.
- b. Select **Data Source: JDE** and **Object Type: Processing Options**.
- c. Click the **Discovery Object** icon.



This will open JDE and pull all of the Processing Options from the customers JDE connected JDE environment. Upon completion, the system will show these in alphabetical order.

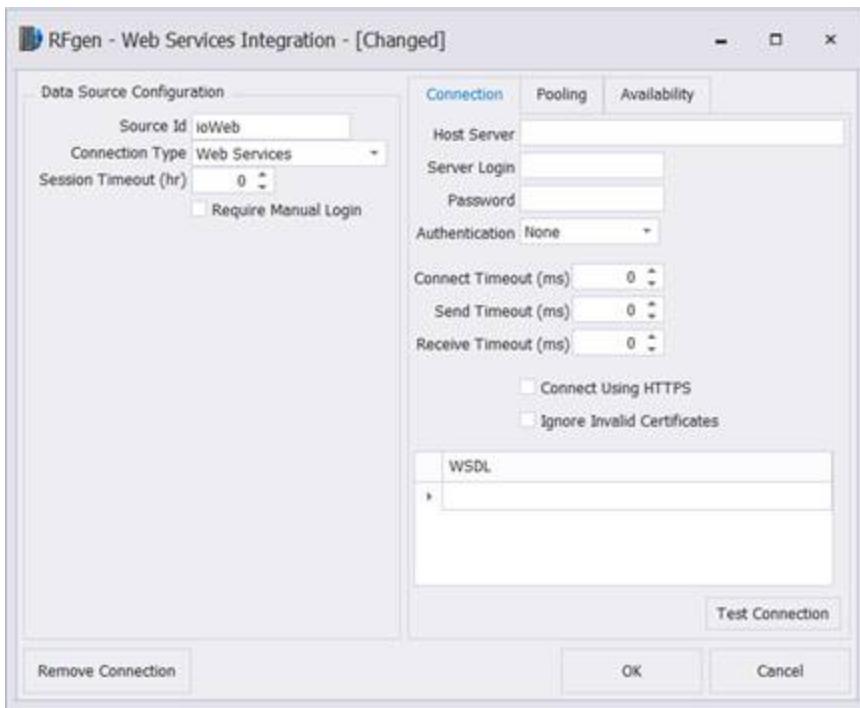
- d. With ONLY THOSE ONFILE SHOWING, click the **Select All** icon, which will checkmark all the listed items.
- e. Next, select the **Download Schema** icon. This will update RFgen to match the current JDE Processing Options.

Note: This step should be completed with each JD Edwards new release, or when an affected processing option is changed by JDE.

You are now done.

Adding A New Web Services Connection

To configure a Web connection, select **Enterprise Connections > Add New Enterprise Connection > Add New Web Services Connection**.



Enter a **Source ID** which will be the name to reference when Web object's DataSource property.

Select the Connection Type from the Web Services list menu.

The **Session Timeout** value (in hours) will disconnect and reconnect to the ERP at the specified interval. This may be required if the ERP is configured to not allow a connection that never times out.

The **Host Server** is the IP address or DNS name of the server being used to process requests.

A **Server Login** and **Password** can be entered if required by the server.

Select the type of Authentication to use when sending the User Name and Password over the Internet: *None*, *Text*, or *Digest*. *Digest* is generally an application of the MD5 cryptographic hash.

The **Connect Timeout** is a number in milliseconds that will terminate a request for connection if this value is exceeded.

The **Send Timeout** is a number in milliseconds that will terminate a request from the client sent to the server if it has not received the HTTP request from the client.

The **Receive Timeout** is a number in milliseconds that will terminate a response from the server to the client if this value is exceeded.

Connect using HTTPS is used when the web server communicates using the encrypted protocol.

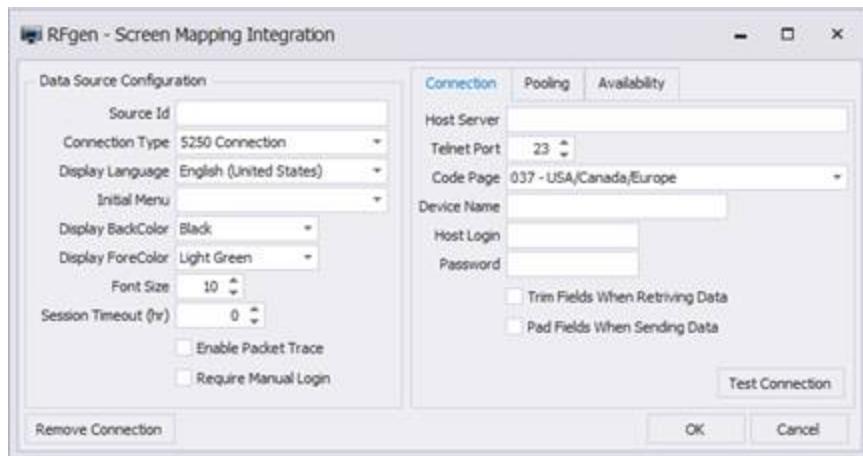
The **Ignore Invalid Certificates** option is a True or False value (only for HTTPS connections) indicating that the data connector will ignore certificate errors from the server. If this is set to False, and there is an error, it will be logged in the RFgen error log and the Web Object's Execute method will return a False value.

The **WSDL** grid is for the XML-based file which describes the business logic, parameters and data structure to be used when processing/exchanging information with the web logic server.

The **Pooling** and **Availability** options work exactly as described in the database connector section.

Configuring the Host Connection

In the Mobile Development Studio, click on **Enterprise Connections > Add New Enterprise Connection > Add New Screen Mapping Connection**. The following window will appear.



The first entry is the **Source Id** used to reference the data connection only. This can have any value but spaces and extended characters are not recommended.

Choose the **Connection Type** (VT220, TN5250 or TN3270); i.e., the protocol used to communicate with your host system. Notice that there is an additional option called Console Application. This type is designed to launch a console application rather than use a telnet server and then pass that display through the server to the device using the HostScreen prompt control. One example would be the SAP console application (SAPCNSL.EXE) running on the server and being displayed and allowing interaction with the user on a mobile device. Simply specify a process or executable name to run and any passing parameters necessary.

The preferred option is UTF-8 but if a legacy system's output is language specific then the **Display Language** field should be changed to make the screen render correctly. The Language field can be left as (Default) if a code page is specified or if UTF-8 is used.

Preferences for the emulation screen include the **Back Color**, **Fore Color** (the color of the font) and **Font Size**. These are only for development since the screens are hidden during production.

The **Session Timeout** value (in hours) will disconnect and reconnect to the legacy server at the specified interval. This may be required if the legacy server is configured to not allow a connection that never times out.

In the case of communication errors the **Enable Packet Trace** option can be set and a trace log of the communication will be captured. This is used by support staff to diagnose issues on behalf of the customer. Please contact support if this switch is necessary.

If the **Require Manual Login** is checked, a connection request is created between the user and the ERP system. If this box is unchecked, the user login uses the ERP connection between RFgen and the ERP system.

Connection Tab

Next, type in the **Host Server** name or IP address. The **Telnet Port** is the port that the server uses to communicate with your host. The default for a telnet server is port 23.

If TN5250 or TN3270 are selected, you may enter a **Code Page** for specifying the language being used in the protocol and an **IBM Device Name** for the host system. Code pages were selected for loading when you loaded the screen mapping software. These fields are hidden in the VT setup.

For VT220 the **Data Stream** field can be set to either Standard or UTF-8 to accommodate the type of packet data coming from the host system.

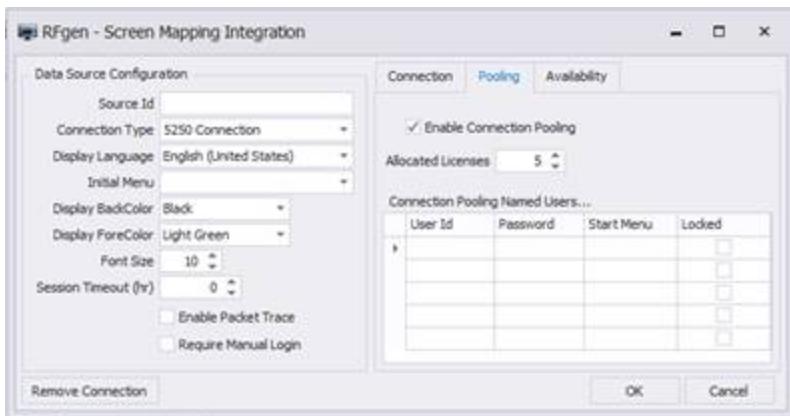
When using the connection type 3270 or 5250, the **Device Name** field is designed to make each connected device appear unique to the host system. Leaving it blank, the host system will not distinguish between the connecting clients. Fill this field in with a name and the server will automatically add a three digit, zero padded number to each client so the host system will see each connecting session as a unique device.

The **Host Login** and **Password** fields are used only if SSH is used when connecting to the host system. Under the VT220 options, if Connect via SSH is checked then the Host Login and Password are required.

Trim Fields When Retrieving Data set to enabled will auto trim spaces from the host output fields. If a variable is defined for a section of the host screen (like where error messages are displayed), this feature will trim the text for easier use in message boxes, for example.

The **Pad Fields When Sending Data** option when enabled will use spaces to pad any input. A variable defined for a region of the host screen where input will take place also has a length property assigned at the time the field was defined. If the data is 3 characters, but is placed in a host screen field designed for 10 maximum characters, the server can pad the input data to fill up the host screen input field.

There are some additional properties for the VT220 mode only. **Echo Characters Locally** means that the server will print the typed characters on the telnet screen because the host is accepting the keystrokes but not showing them to the user. **Wrap Text at End of Line** will force the server to place the additional text on the next available line if it doesn't fit in the current field. Most host system will do this automatically. **Destructive Backspace** means that the server will receive a backspace command and apply it to the screen as a command that removes the last character. Some systems would move the cursor but not remove the character. **Send Whole Key Packets** forces the server to submit keystrokes in one packet instead of two in some cases. Most host systems already support keystrokes coming in as one or more packets. **Send Return + Line Feed** will add a carriage return plus a line feed to the Enter keystroke when communicating with the host. **Connect via SSH** will establish an SSH (secure) connection to the host from the server. If this option is turned on then the SSH **User Name** and **Password** fields will be required.



Pooling Tab

Connection Pooling can be enabled and the maximum connections allowed in the pool can be selected. This selection will determine how the server and its clients will interact with your host system. The options for the **Pooling Status** are:

Disabled – Setting connection pooling to disabled will cause the server to spawn a connection to the host system for each active mobile device. Each connection will be linked to a particular device on a one-to-one basis, and will be shut down when that device disconnects. Note: there is no limitation on the number of connections allowed.

Enabled – Setting connection pooling to Enabled will cause the server to spawn a single connection to your host system. As each device requires access to the host system, they will go to the pool and retrieve one of the available connections. When they are finished, the device will release the connection back to the pool. If no connections are available, the server will start a new connection (up to the specified maximum) and add it to the pool. After 10 minutes of non-use, an opened pooled connection will be terminated releasing resources on the server and potentially licenses on the host system. Keep in mind that unless the SM.BeginTrans and SM.CommitTrans commands are used, it would be possible for one user to position the screen in one place while another user also uses that pooled connection to perform their tasks causing both users to get failures.

The **Connection Pooling Named Users** grid dictates how each host session is started. You may also override the default settings by configuring a specific pooled session separately.

Session - Each of the individual pooled connections are listed separately. This provides for specific settings for each connection.

User Id- If the host system requires that unique names be used or creating multiple logins with the same user is prevented, each pooled connection can have its own user ID. Session, user, and password information can be obtained at runtime with the commands SM.SessionUser, SM.SessionPwd, and SM.SessionID.

Password- This is the corresponding password used for each unique user ID.

Start Menu- Each session can have its own main menu. When a session is requested and no main menu is specifically assigned or the "(Default)" value is used, the next available session will execute the requested main menu based on the scripts and chosen transaction. If a session is requested and the next available session does have a main menu assigned, and it is not the required one, other sessions will be evaluated for a matching main menu. If one is found and available, it will be used.

Locked- The ability to lock a session means that the session can ONLY be used with the specified main menu and will not allow other main menus, even if all other available sessions are in use. For example, there are 10 pooled sessions, five locked on main menu A and five locked on main menu B. If a session with main menu A is requested and all five sessions for main menu A are currently used, the server will look to the sessions assigned to main menu B. If they are not locked, the server will take one of them. Since they are locked into main menu B, in use or otherwise, the server will wait for one of the first five to be released.

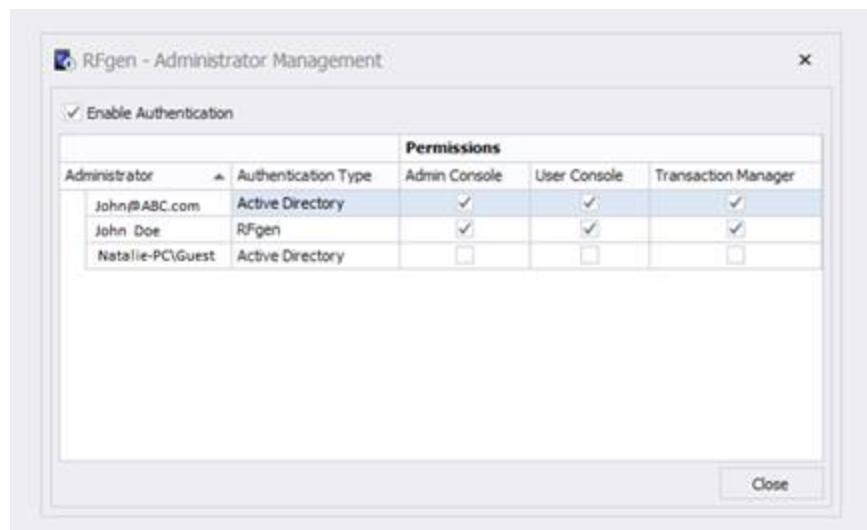
The purpose of locking a set number of sessions to a specific main menu is to ensure that there is always some bandwidth available for certain transactions. Not locking them means that they will be marked with a preference for a type of transaction (the use of a specific main menu), but will switch to another main menu when necessary.

For example, there are 10 pooled sessions available and the first five have one main menu assigned and the last five have a second main menu assigned. When a session with the second main menu is requested, the 6th session handle will be used. This is only significant because of the Locked property.

The **Test Connection** button will verify all settings before saving the connection. This is not required.

The **Save** button will save changes but will not test and verify settings.

Configuring User Access Control



The **User Access Control** feature authenticates users who are accessing the RFgen server from the *Admin Console* (the Mobile Enterprise Dashboard), the User Console (*User Management Console*), or the Transaction Manager (*Transaction Management Dashboard*).

This screen can be accessed from the *Mobile Development Studio*, **Configuration > User Access Control** menu.

When **Enable Authentication** is checked, all Mobile Enterprise Console (Admin Console), User Management Console and/or Transaction Manager Dashboard user sessions are authenticated. When its unchecked none of these console/user sessions are authenticated.

The **Administrator** column lists users who are allowed to access the server via the Admin Console, User Console, or Transaction Console. To add an administrator, the person's user account must exist in Active Directory or exist as a Local User Account. *RFgen Administrators* is a special account that include users who were added from the Local Users Account. *Active Directory Administrators* are users whose accounts are managed in Active Directory.

The **Authentication Type** column lists either *RFgen* or *Active Directory*.

When checked, the user has access to the server from the specified **Admin Console**, **User Console**, and/or **Transaction Manager Console**.

Adding or Removing Administrators

1. From the RFgen Server: Click on Configuration > User Access Control.
2. Right-click on the Administrator column and select **Add RFgen Administrator...** or Add Active Directory Administrator.
3. The screen Enter RFgen credentials displays.
4. Enter the User's Name and Password and click OK. The user's name appears under the Administrator column as well as the Authentication type used. Check which console the user should be able to access the server and click OK to save changes.

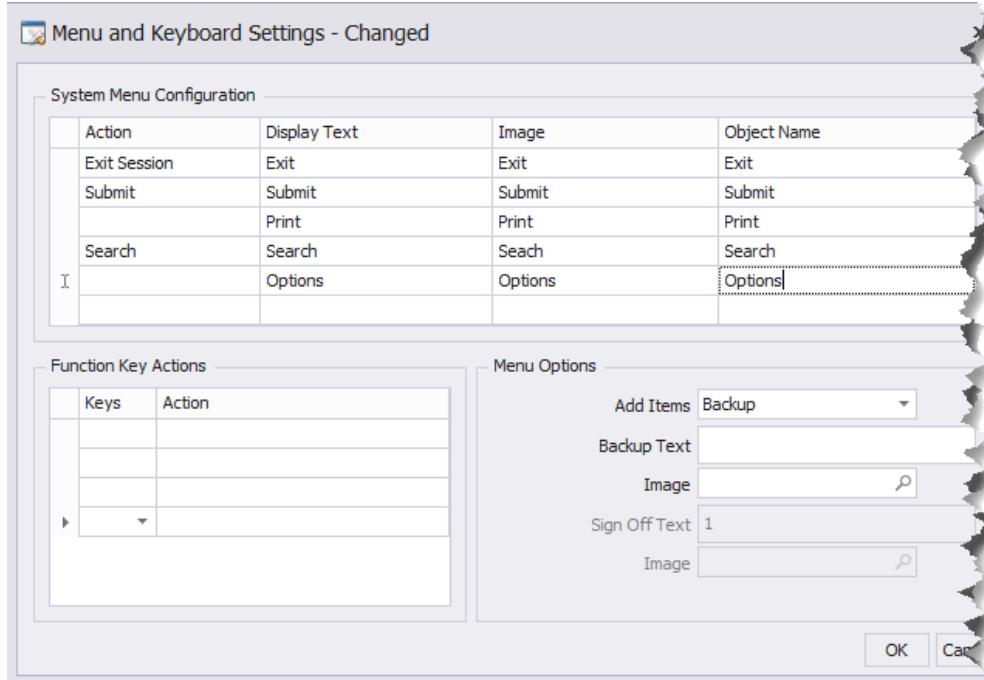
Dev Studio Configuration Options

The following configurations apply only to the Mobile Development Studio. These configuration options are:

- Menu and Keyboard Settings
- Scripting Environment
- Source Control Options
- System Properties (For Creating Data Defaults)
- Download Enterprise Objects
- View Enterprise Objects

Configuring Menu and Key Settings

This screen is only available from the **Mobile Development Studio > Configuration menu**, and is used for providing RFgen client system actions that would be executed from an RFgen-designed, mobile application menu or RFgen SideBar menu (also called the Menu Strip).



System Menu Configuration

Use this table to select factory-provided system operations and associate the icon (image), text, and object to be displayed on the SideBar menu or in an solution menu.

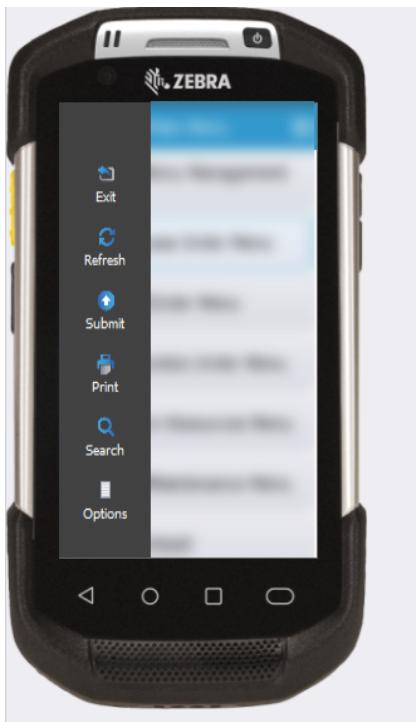
Action column - RFgen provides a list of possible actions, but it is important to verify if that specified system operation is available in the device. The Action values available for selection are: Call Event, Call Form, Call Menu, Cancel Operation, Chain Form, Clear Input, Configure Device, Exit App, Exit Menu, Exit Session, Scan, Search, Sign Out (also known as Sign Off or LogOff), Submit, Toggle Keyboard, Toggle SideBar.

Display Text column - the label of the object or icon.

Image column - See Mobile Development Studio > Images for details on how to upload an image before you select the image (or icon) to be associated and used as a button to launch the action on a menu.

Object Name column - The object is the item which has an action associated with it.

Note: For more details on scripting the button on a SideBar, see "Menu Strip Extensions".



Function Key Actions

This table is used to assign a physical Function Key to a system function.

Key column - Supports Fkeys from 1 to 24.

Action column - Below is a list of possible Actions, but the list also depends on the functions available in the device. The Action values available for selection are: Call Event, Call Form, Call Menu, Cancel Operation, Chain Form, Clear Input, Configure Device, Exit App, Exit Menu, Exit Session, Scan, Search, Sign Out (also known as Sign Off or LogOff), Submit, Toggle Keyboard, Toggle SideBar.

Menu Options

The values in this section are used for the navigation operations assigned to an the Application Menu that is created in the Mobile Development Studio.

Add Items This is used to set the action for navigation (backup) or the log off operation.

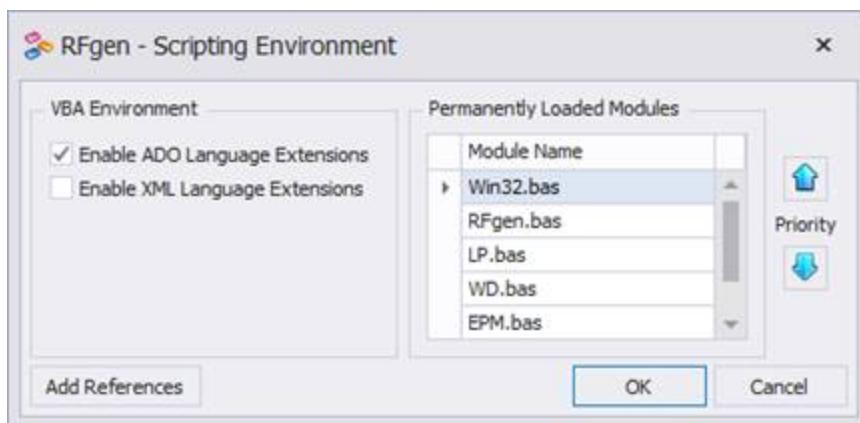
Backup Text This allows you to navigate the user back to a previous screen in a mobile application.

Image See Mobile Development Studio > Images for details on how to upload an image before you select the image (or icon) to be associated and used as a button to launch the action on a menu.

Sign Off Text and Image

This enables you to customize the text and icon image for the Log Off operation which is used in Mobile Development Studio > Applications. For details on how to upload an image before you select the image (or icon) to be associated and used as a button to launch the action on a menu.

Configuring the Scripting Environment



This screen is only available from the **Mobile Development Studio > Configuration** menu, and is used for enabling language extensions into the Studio so you can access the extensions for scripting purposes.

VBA Environment

Enable ADO Language Extensions allow you to access database(s) directly in VB rather than just through the pre-built RFgen programming extensions available for database access. If you are planning to write your own database access code, you will need to check the ADO option. Support for the method will automatically be loaded as required.

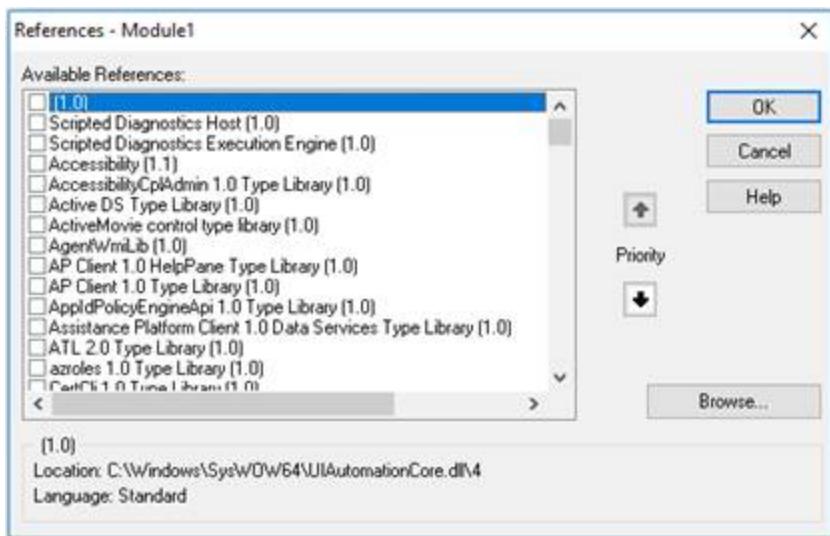
Enable XML Language Extensions provides additional parameters for manually specifying XML communication settings.

Show Line Numbers and **Show Function Outlining** features are meant to enhance the programming environment to provide collapsible functions, where code has changed, line numbers, etc.

Permanently Loaded Modules

The **Win32** and **RFgen BAS** files are always loaded for each transaction. If another BAS file is created and the programmer does not want to place the code into either of these pre-loaded modules, then it may be added to this list. The Win32.bas is typically used to store global variables. The RFgen.bas is typically used to contain functions and procedures that need to be accessible from any transaction. If a BAS file needs to be referenced for only a few or one transaction, the VBA Scripts / References menu option should be used.

The **Add References** button will globally add Global Assembly Cache (GAC) classes to the RFgen solution. This is the window that appears.



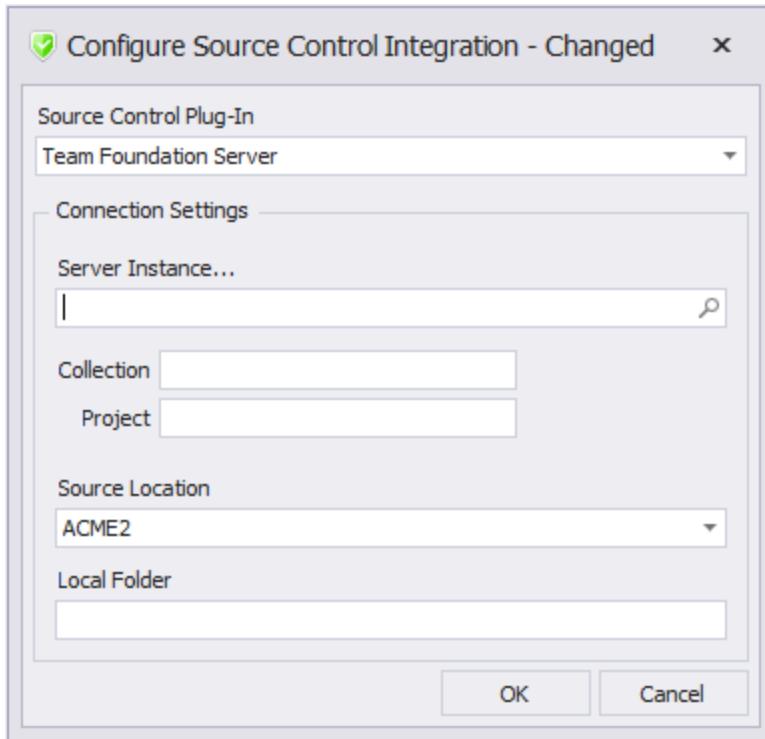
Configuring Source Control Options

The **Source Control Integration** lists source control products that can be used to provide the development process to check-in / check-out developed objects such as users, menus, applications and images. When you install the Mobile Development Studio, or the Mobile Unity Platform, this feature is automatically installed.

The source control product such as the Microsoft Visual Studio should be installed on the local developer's PC before you can setup the connection. Depending on how your network administrator has setup your environment, and security credentials, the user may need to log into the source control server in order to initialize the connection.

To Configure the Source Control Connection

1. In your Mobile Development Studio menu bar, click on **Configuration > Source Control Options** to display the Source Control Integration screen.



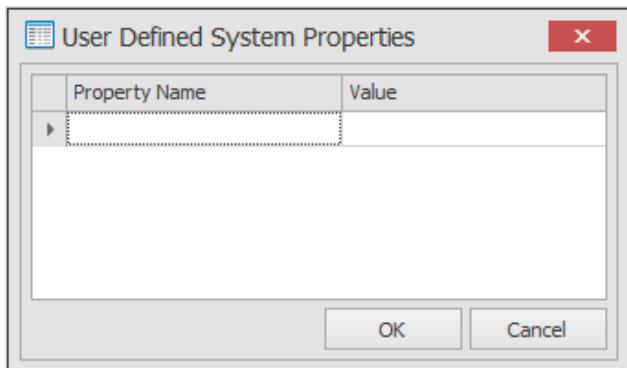
2. **Source Control Plug-in** - Lists the source control servers that are available for connection. Select the server which serves as your source for connection. If no servers are detected, "None" will display. Refer to your product's documentation for setting up user access and projects.
3. **Connection Settings - Server Instance** - The source control server instance and remaining properties are identical settings as those in the source control product itself.
4. Click on the Search icon to select the server which has the source control product you want to connect to. For example, if you had your environment setup for the Microsoft Team Foundation Server, a **Connect to <Name of your server>** screen will display. Depending on your product, this is where you select the specific server, then enter another screen to add the server's name or URL, setup the connection path, port (i.e.

8080), and protocols (HTTP or HTTPS), and team project collections. For example, a "Connect to Team Foundation Server" screen displays for the user who has the Microsoft Team Foundation Server setup.

5. The **Collection** and **Project** fields will display your selection from the previous step.
6. The **Source Location** dropdown menu lists the Data Source Configuration System/Company ID source. (To review your setups for this, see **Configuration > Application Database**.)
7. The **Local Folder** is where you store your files locally.
8. Click **OK** to exit and save your changes.

For additional information, see [Solution Source Control Options](#), [Menus/Object States](#), [Find Shelve Sets](#), and, making [Shelve Changes](#).

Configuring System Properties



This screen is only available from the **Mobile Development Studio > Configuration menu**, and is used for creating data values or formats that can be accessed by a mobile application if its referenced by the application.

Entering a name and value in this box is like creating a global constant with a read-only value. For example, if this installation was designed for multiple warehouses but this particular installation was for a warehouse called 'Main Street' then entering the property name of 'Warehouse' and a value of 'Main' would allow the programmer to identify which warehouse was being used. The command used is `System.EnvironmentProperty`.

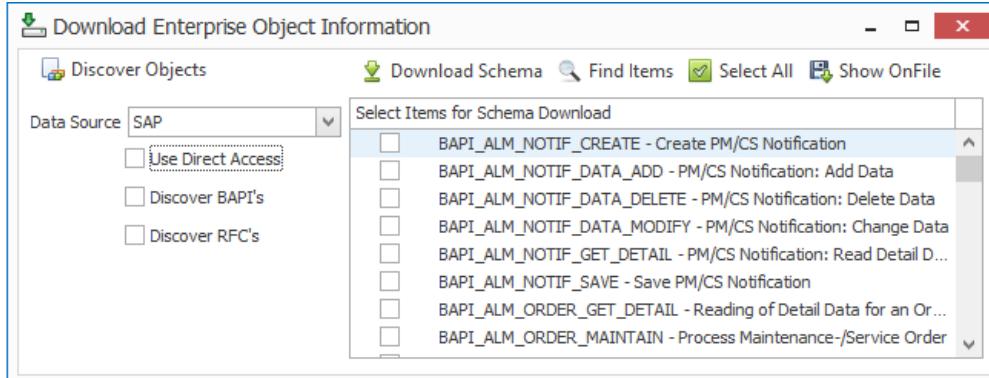
Download Enterprise Objects

This screen is only available from the **Mobile Development Studio > Download Enterprise Objects**, and is used for downloading objects that define the arrangement and retrieval of data coming from an ERP.

If you want to access table fields directly, or to take advantage of backend functions or stored procedures, the server needs to know which Data sources to connect to and which items to download. It also needs the required tables or procedures and the object's structure in order to internally generate the proper calls and perform the reads and writes.

To access the tables, RFgen will prepopulate the display with relevant objects in the file when you select **Enterprise Connections > Download Enterprise Objects**. Once your schemas are displayed, you can select and refresh the schemas without performing a Discovery first.

If however you need to discover objects, this function is also available.



Discover Objects allows for all objects to be discovered or just selected items that may be selected.

Download Schema starts the download process

Find Items provides a text search of items using the words you enter.

Select All selects all entries in the list.

Show On-File limit the list to previously downloaded tables.

You can also check these boxes to filter items before you download: Use Direct Access, Discover BAPIs, Discover RFCs

Since RFgen is SQL compliant, it is important to note that database table and field names should not use any of the reserved words listed in the section describing the VBA commands.

Downloading ERP Business Functions

To work with business functions from an ERP system, users must first transfer (download) the desired business functions from the ERP system into RFgen. To do so:

1. Click on the **Enterprise Connections > Download Enterprise Objects**.
2. Select the ERP connection from the Data Source drop-down and select **Discover Objects** from the menu.
3. Click the rows of functions to be downloaded (or click Select All), and then click **Download Schema**.

Selecting all business functions from an ERP system will save an extremely large set of data in the RFgen application database and could take a very long time. Only download the business functions that are required by the applications.

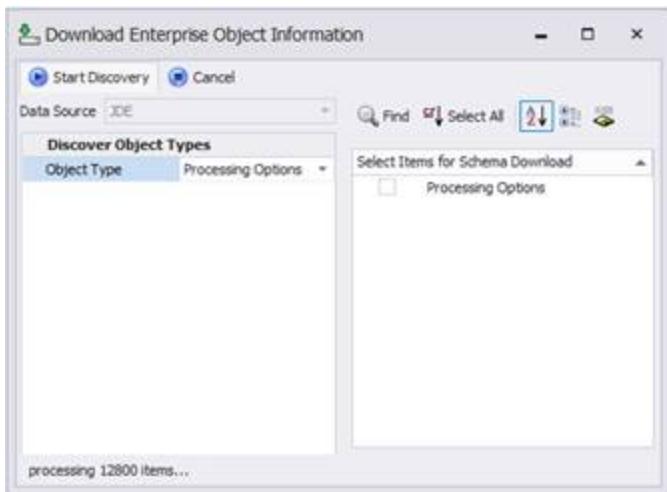
Blue entries have been previously downloaded. The SAP Discovery Filters allow you to select, if just BAPIs are downloaded or if RFCs are as well.

Downloading JDE Processing Options

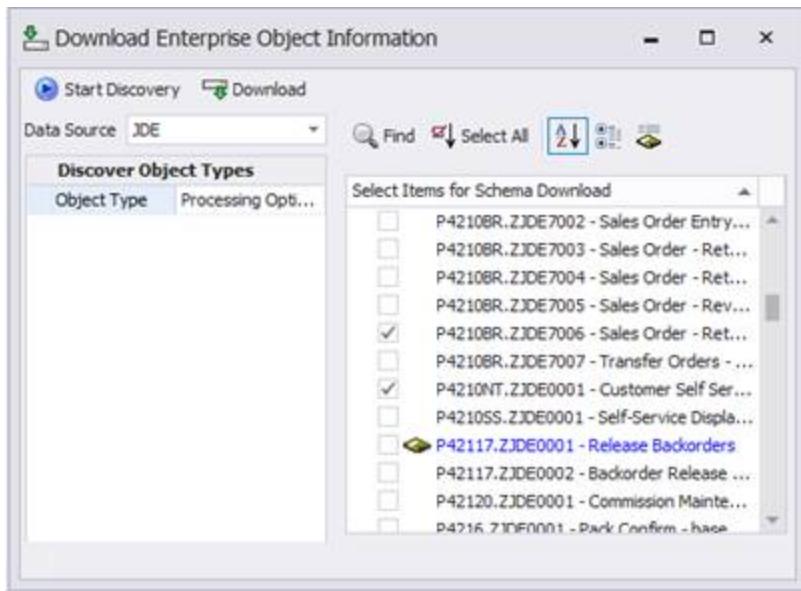
To work with business functions from an JDE system, users must first  the desired business functions from the JDE system into RFgen.

1. Click on the **Enterprise Connections > Download Enterprise Options**.
2. Select the JDE connection source from the **Data Source** menu.
3. Select *Processing Options* from **Discover Object Types >Discovery Mode** drop-down menu.

4. Click **Discover Objects** from the menu.

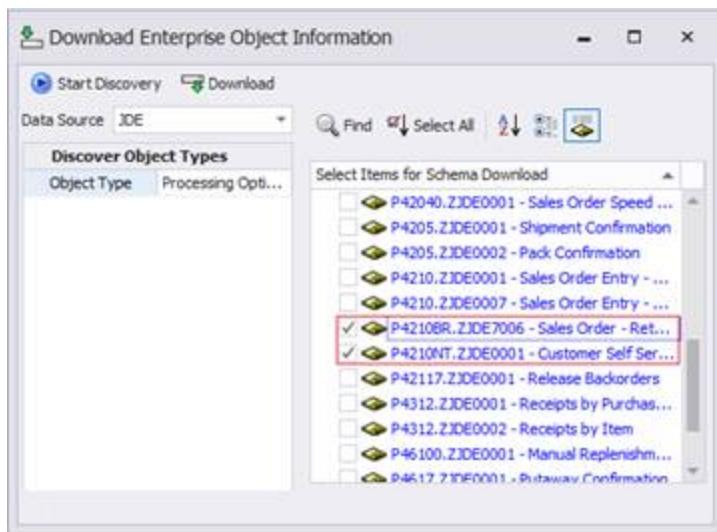


5. The Discovered objects display in the right pane. Select the desired processing option to be downloaded or click *Select All*. You can also use Find to located the specific item(s) to be downloaded.



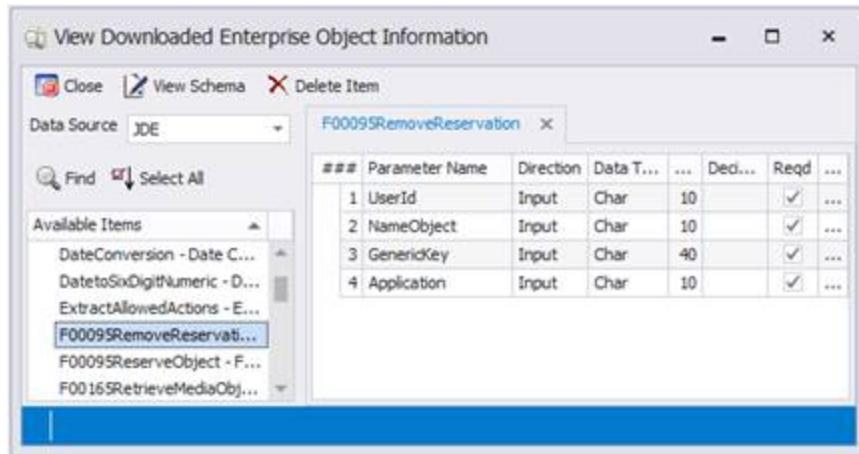
6. Items in blue indicate they have been downloaded before.

7. Click Download.



In this example above, the checked items are the new downloads added to the list of prior downloads.

Viewing Enterprise Objects



This screen is only available from the **Mobile Development Studio > Connections menu**, and is used for viewing Enterprise Objects that have been downloaded to the Mobile Development Studio.

Previously downloaded object schemas may also be viewed by clicking on the Enterprise Connections – View Enterprise Objects selection.

After choosing a data source select a name and click the "View Schema" menu option or simply double-click the table name to view its parameters.

Shown above are the field definition items for a chosen table. Each transaction table must have at least one primary key ('PartNo' as indicated above). RFgen identifies database keys simply by determining which database items are 'indexed'. If more than one item is indexed, the first item encountered will be marked as the primary key.

In general, use of Numeric, Text/String, Date, and Currency 'Data Types' in your database is suggested, as more esoteric data types may cause problems when trying to update your database table(s).

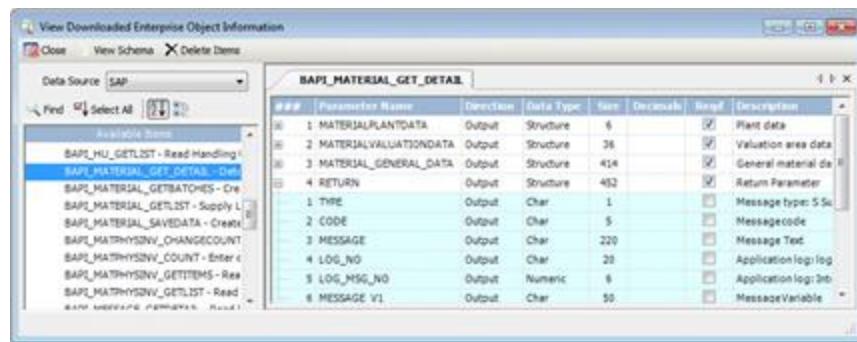
Only fields defined as updateable will be sent to the database when a transaction is completed. This is only true if RFgen is generating the SQL statements internally as opposed to user created SQL statements. Fields not defined as 'null allowed', will send a space or a 0 (zero) if no data is collected for them. Fields marked as 'primary key' are used to access the table data and may be used to retrieve selected data.

When viewing the SAP specific function properties after a download, please note that for "packed" or compressed numeric data elements RFgen displays the byte length and the allowed number of decimals instead of the actual number of characters allowed in the field.

Selecting a table entry and choosing Delete from the menu only removes the stored structure of that table from the RFgen configuration. This delete has no impact on the actual database.

Viewing ERP Business Functions

Business Function definitions may be viewed by clicking on the Enterprise Connections – View Enterprise Objects selection. A view window will appear.

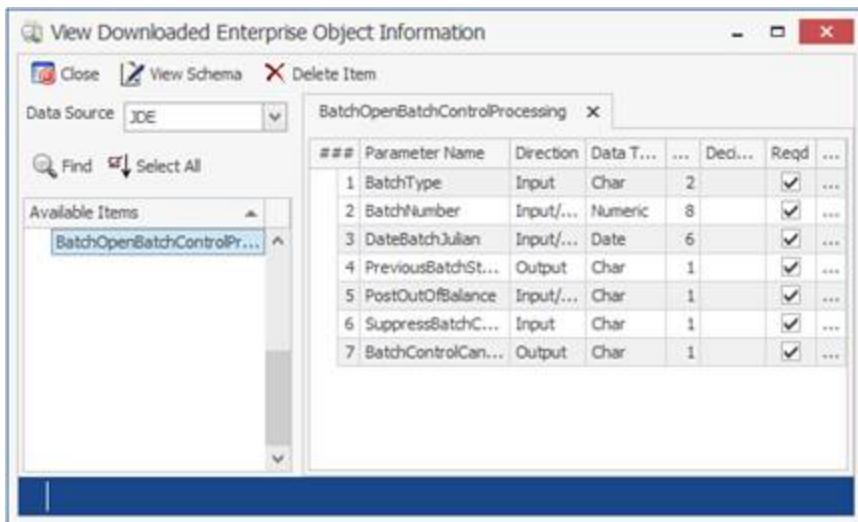


A list of downloaded business functions is displayed. If a description is stored in the database and was retrieved by RFgen, it can be displayed after viewing the parameters. Select a name and click the "View Schema" menu item or simply double-click the business function name to view its parameters.

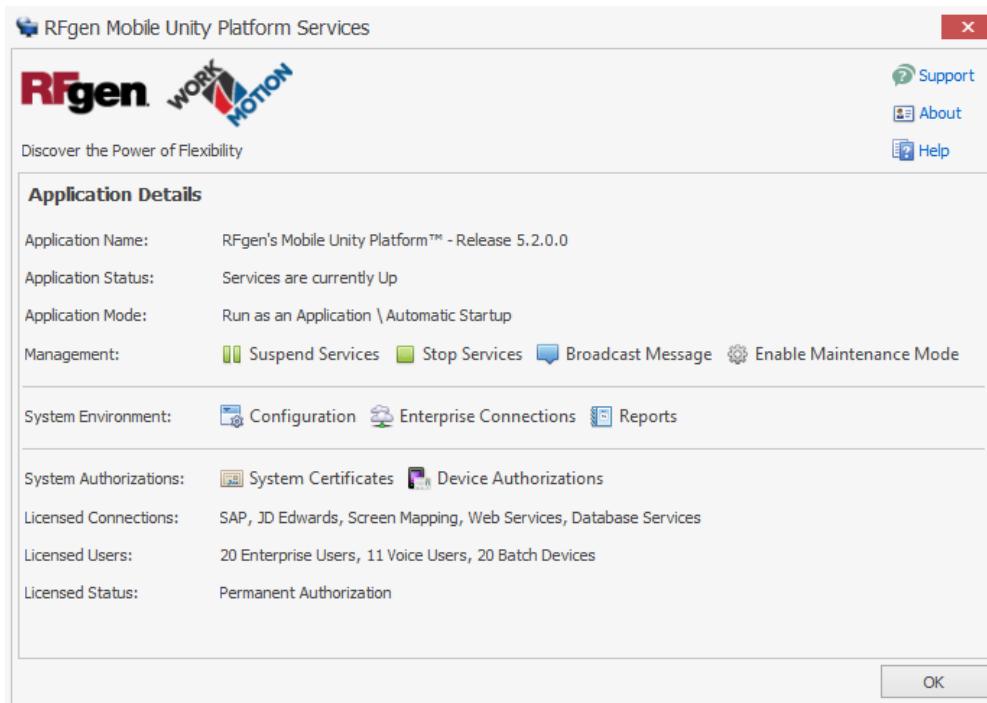
Viewing JDE Processing Options

To view JDE Process Options parameter information, follow these steps:

1. Click on Enterprise Connections - View Enterprise Objects.
2. Select the JDE connection source from Data Source.
3. The Processing Options displays in the left pane of the View Downloaded Enterprise Objects Information screen.
4. Select the Process Option then click View Schema from the menu. The Process Option's parameters and its associated details will display in the right pane. Double-clicking the Processing Option will also display the details in the right pane.



Mobile Unity Platform Services Console



The **Mobile Unity Platform Console** (Services console) is a graphical interface for administration of your server services at a glance. The Services console also allows you to:

- Identify the version of RFgen installed on the server.
- Check the running status of server services. It tells you if your services are up or down, and whether the server is in Maintenance Mode.
- Stop or start the server service and allow or disallow users to connect (Maintenance Mode), and broadcast a message to all connected users.
- Configure the server and setup connections to database systems and enterprise (ERP systems).
- View reports (application event logs) and export them to Excel.
- View server authorization status, number of licensed uses. If your server is not yet activated, it can be activated through here. (See Web Authorization.)
- View which mobile devices are connecting with your server and controls which ones are authorized to connect.

When started, the Server enables multiple communication sessions between your server and your remote devices (up to the number of authorized users).

System Environment > Configuration > Application Services may be used to specify a different port. The service will administer to all types of clients (GUI-based devices, XML, Vocollect, etc.) simultaneously.

For additional information on configuring the server or connecting to your ERP/source of data, see [Configure Your RFgen Server or Mobile Development Studio](#).

For an overview of the Client - Server connection process, see [Basic Implementation Steps](#).

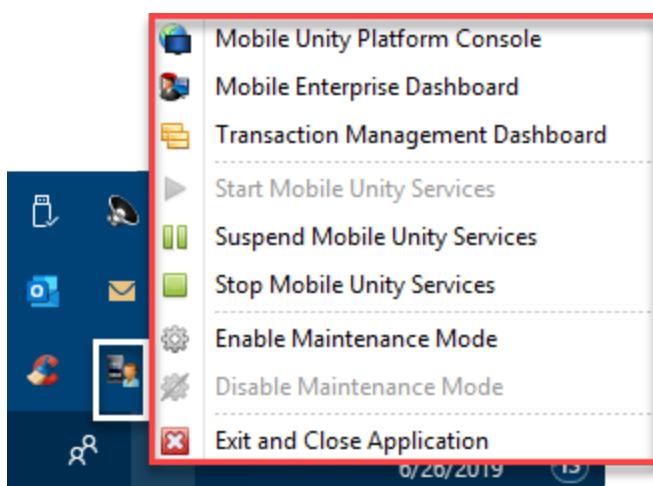
Accessing the RFgen Server Console (Mobile Unity Platform Console) and Services

If you want to display the RFgen server console from your Microsoft Windows System Tray, follow these steps.

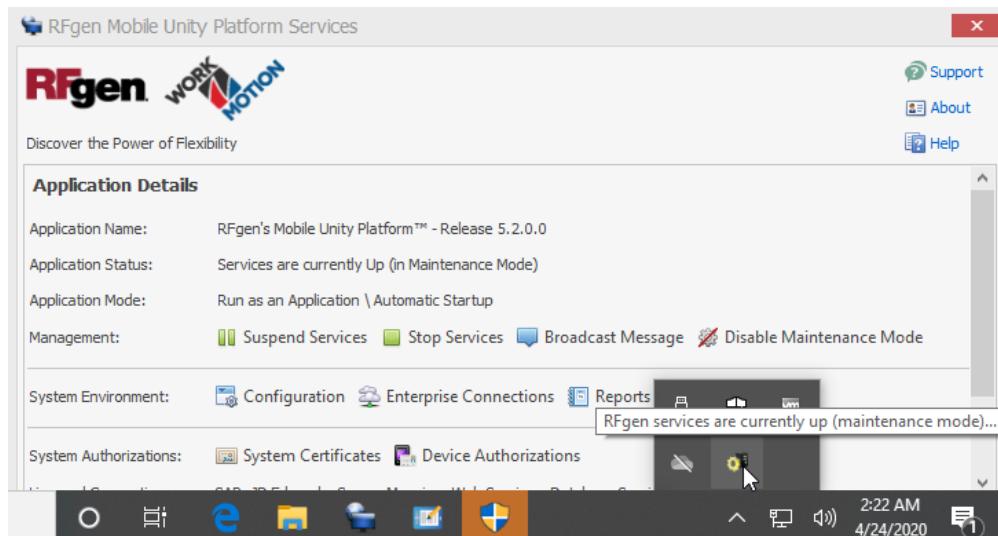
For this procedure, first add your Mobile Unity Platform console icon to the Windows System Tray. For details refer to your Microsoft Windows System Help.



1. Click on the Mobile Services Manager icon  in your Windows system tray.
2. The Mobile Services Manager Menu icon displays.



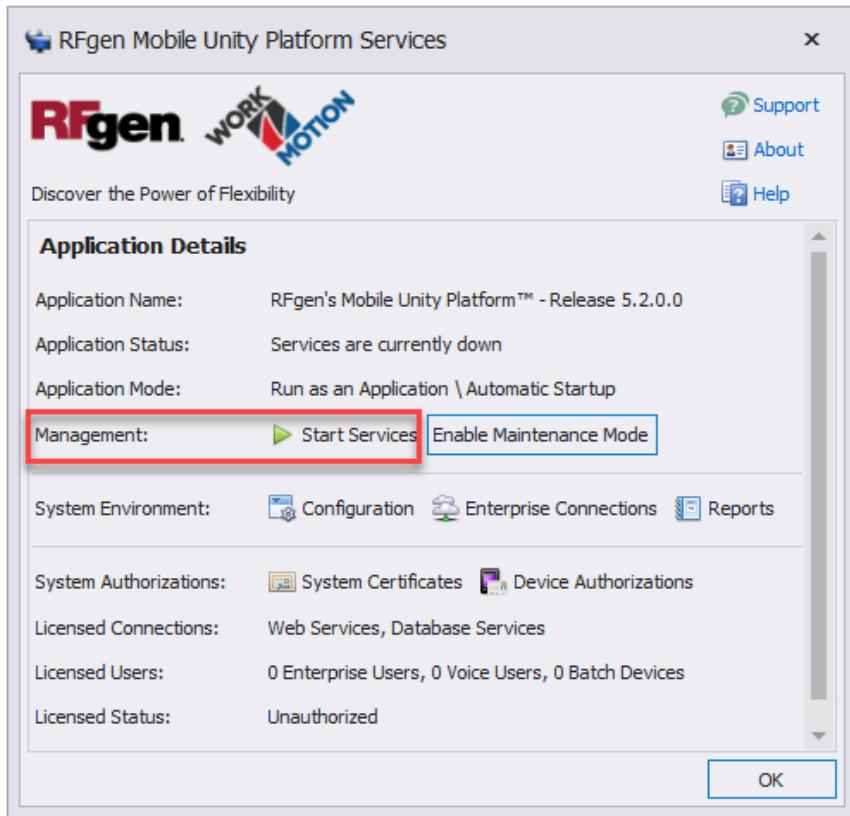
3. Click on the Mobile Unity Platform Console.
4. The RFgen Mobile Unity Platform Services Console displays.



Management of RFgen Server Services

The Mobile Unity Platform Services console provides a summary of your server's status and functions to change its operational status.

To run the application services status is shown as currently down or suspended, it may need to be license and [authorized](#).



Suspend/Stop/Start/Resume Services If you need to stop or suspend your services, simply click on the appropriate button. The difference between a suspension and a stop is that if you Stop Services, your server goes through a longer process to restart the service than if it's simply paused. You turn on your services via the Windows Task Bar menu.

Note: You can also [use commands to start or stop RFgen server services](#).

To Stop or Start Services Using Commands

In order to start and stop the Server from the command line, navigate to the install directory using the DOS 'dir' and 'cd' commands. Then use the following commands:

rfscm510 –startup

rfscm510 –shutdown

If you stop the service, the console will automatically be stopped.

To Enable or Disable Maintenance Mode

Broadcast Message allows you to send out a message to all connected users. The message you enter will appear across their screens of connected mobile device users.

To Enable Maintenance Mode

Enable Maintenance Mode or **Disable Maintenance Mode** is a service that prevents new users from connecting to the server while retaining existing connections. This helps administrators disconnect users gracefully before shutting down the server for maintenance purposes.

The server defaults to Disabled Maintenance Mode unless you enable Maintenance Mode.

If the Mobile Services Manager is in Maintenance Mode it displays this icon: 

To enable Maintenance Mode from the Services Console - Management, click on the **Enable Maintenance Mode** button.

If you don't see the **Enable Maintenance Mode** button, either make your console window wider, or click on the drop down arrow icon next to the Stop Services button.

You can also change the Enable Maintenance Mode / Disable Maintenance Mode from the Windows Task Manager.

Server Authorization

To turn on RFgen server services, a RFgen authorization certificate is required for each server where the RFgen server is installed. This requirement includes RFgen servers used for load balancing, clustering and backup. If your Sales representative has not activated your server, you can also send an email request to RFgen Support or activate it using the link in the Mobile Unity Platform Console.

If the server service is paused (suspended), check if your server needs to be authorized.

To Activate (Authorize) the Server

If your RFgen server is licensed and unauthorized, the services will be suspended (paused). To activate it, you will need a system authorization certificate installed to your server. This certificate can be obtained several ways:

- [Email 'support@rfgen.com'](mailto:support@rfgen.com)
- [Phone request to RFgen Support](#)
- [RFgen Web Authorization process](#)

Authorize by Email

1. Upon purchase of a license, RFgen assigns and emails you a Customer ID and serial number. The Customer ID and Serial Number will look similar to the example below:
Customer ID: 7382

Serial Number: KXI8N-9384

2. Submit your Customer ID, serial number, and your server system's ID/fully qualified domain name to:

Support@RFgen.com

Subject: Technical \ Licensing

Note: If you lost your customer ID or serial number, you can still request an authorization certificate via email. Be sure to include your company's information, server's system ID/fully qualified domain name, and your email address. The server system ID is necessary to help RFgen create the authorization certificate.

3. Support will verify your purchase information and then email you: Customer ID, serial number, authorization certificate (RFgen.cert), and certification installation instructions.

Phone Authorization Request

Before you call RFgen Support (916) 939-2065 Have the following items prepared:

Submit your Cutomer ID and Serial Number (these came with your license) and your server's system id (fully qualified domain name).

Example:

Customer ID: 7382

Serial Number: KXI8N-9384

System ID: CS51-ACME-2016.Prod.ACME.com

Note: If you lost your customer ID or serial number, you can still request an authorization certificate via email. Be sure to include your company's information, server's system ID/fully qualified domain name, and your email address. The server system ID is necessary to help RFgen create the authorization certificate.

RFgen Support will send you an Authorization certificate with installation instructions.

Activation by web

To authorize the service for permanent operation using the web, follow these steps if the server has never been authorized, or was once authorized and the System Id has not changed. If your server has been authorized before or your System ID has changed, then contact RFgen Support for assistance.

From the **Mobile Unity Platform Console > System Authorization**, click on the System Certificates.

The RFgen – Service Authorization window displays.

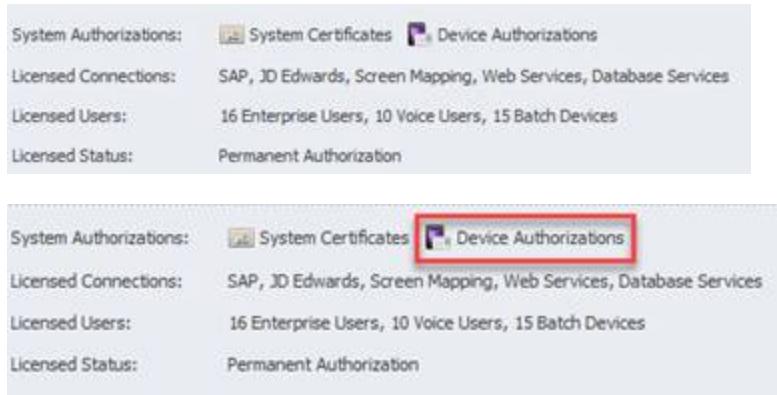


1. Enter your **Customer Id** and **Serial Number** and click the **Web Authorization** button.
2. For example:

Customer ID

7382 ;Serial Number KXI8N-9384

3. Your system ID is filled in automatically by the Mobile Unity Platform Console.
4. Click on the Web Authorization button.
5. The window flashes (nothing else displays.) When done, the License Status changes to "Permanent Authorization."



Device Authorizations

This feature allows you to view which devices requested a connection to the RFgen server and which ones you authorized (checked), depending on your Restrict Access settings in the server. The list of devices can be viewed from your Mobile Unity Platform console (Services console) or Mobile Development Studio. Click on **Services Console > Device Authorization** or **Dev Studio > Devices > Authorized Devices** menu.

To set Device Authorizations on in the server, see [Configure Environment Settings > System Options > Restrict Access to Known Devices](#). This dictates if all online clients (Thin clients) will require manual authorization from the Services Console / Dev Studio. By default, all Offline (batch/mobile) clients will require manual authorization from the Services Console / Dev Studio regardless of whether the Restrict Access feature is enabled or disabled.

For example, if Restrict Access to Known Devices is enabled (checked) in Configuration > Environment Settings, then all Online Clients (Thin Clients) will require you to check the Online Access box of the requesting device in order to allow it to connect to the server. If Restrict Access disabled, then the Offline Authorization column is hidden and manual authorization of Thin clients is not required.

RFgen - Mobile Device Authorizations						
Online Access	Offline Auth...	Device Name	Device Type	Device Id	Device GUID	Last Connected
<input checked="" type="checkbox"/>	<input type="checkbox"/>	WIN-JDE920.rf...	Desktop Client	0225A2D9-1C7C...	0225A2D9-1C7C...	12/13/2018 12:43
<input checked="" type="checkbox"/>	<input type="checkbox"/>	RFGENQA42.RF...	Desktop Client	18BC536F-CD97...	18BC536F-CD97...	8/7/2018 14:20:3

Example (above) of Restrict Access enabled in Configuration > Environment Settings. The clients are view in Dev Studio > Devices > Authorized Devices.

Example (below) of Restrict Access disabled in Configuration > Environment Settings. The clients are view in Dev Studio > Devices > Authorized Devices.

RFgen - Mobile Device Authorizations					
Offline Authorization	Device Name	Device Type	Device Id	Device GUID	Last Connected
<input checked="" type="checkbox"/>	WIN-JDE920.rfgen...	Desktop Client	0225A2D9-1C7C...	0225A2D9-1C7C...	12/13/2018 12:43
<input checked="" type="checkbox"/>	RFGENQA42.RFge...	Desktop Client	188C536F-CD97...	188C536F-CD97...	8/7/2018 14:20:31

Additional Details:

What is the [Device GUID](#)?

To authorize Mobile, Offline Clients

When you deploy a mobile profile to a Mobile, Offline client (also called "Batch" or "Fat" client), the client connection to the server must be manually authorized by the RFgen administrator.

Batch client authorization can be performed from the RFgen Mobile Unity Platform Console or the RFgen Mobile Development Studio.

To authorize the device from the Mobile Unity Platform Console:

1. Display the RFgen Mobile Unity Platform Console Menu.
2. Click on **System Authorization > Device Authorizations**.
3. The Mobile Device Authorization table displays.
4. In the Offline Authorizations column and select the row to the device you want to authorize and click on the box to authorize it.

Note: Offline Clients also require a license to operate. For more details, see Activate Client License.

To authorize Thin clients

1. The server must be licensed.
2. The sever must also be authorized for the volume of Thin Clients that are going to connect to the server.
3. The server's **Environment Properties > Restricted Online Access** setting must be **unchecked**.
If server's **Environment Properties > Restricted Online Access** setting is **checked** this forces a [manual authorization of all clients](#) (Thin and Batch).

When a Thin client attempts a connection, the server will check if it has enough licenses and then authorize and accept the connection automatically.
(missing snippet link)

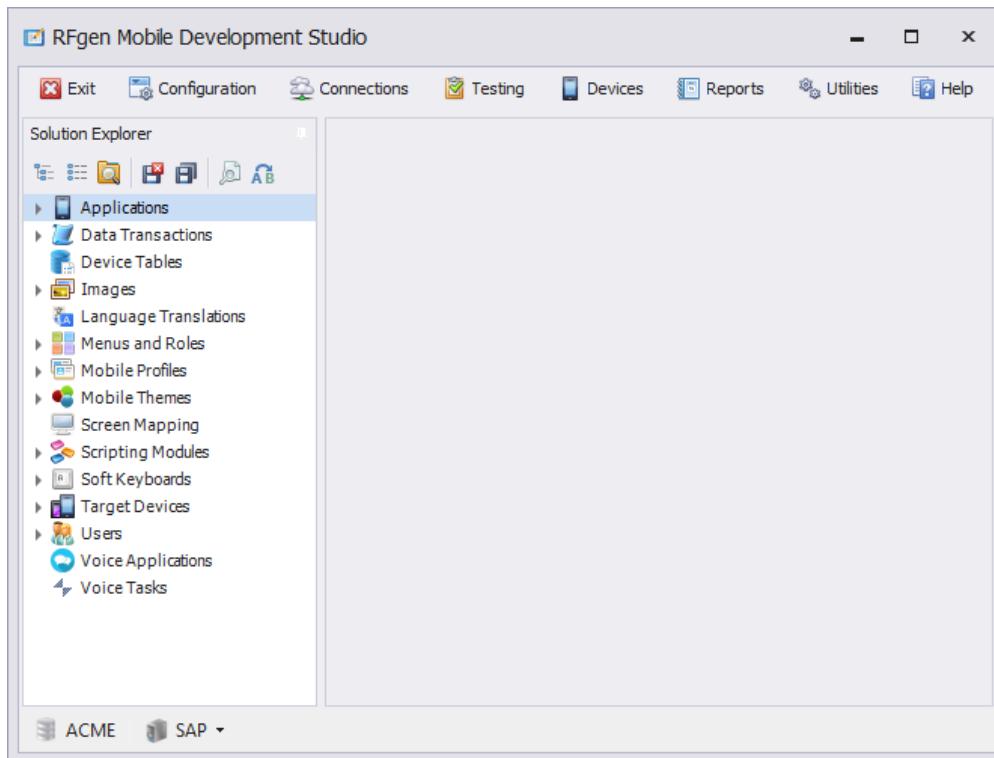
Enterprise: Reports Menu

The **Reports Menu** enables you to monitor and configure event logs for the server.

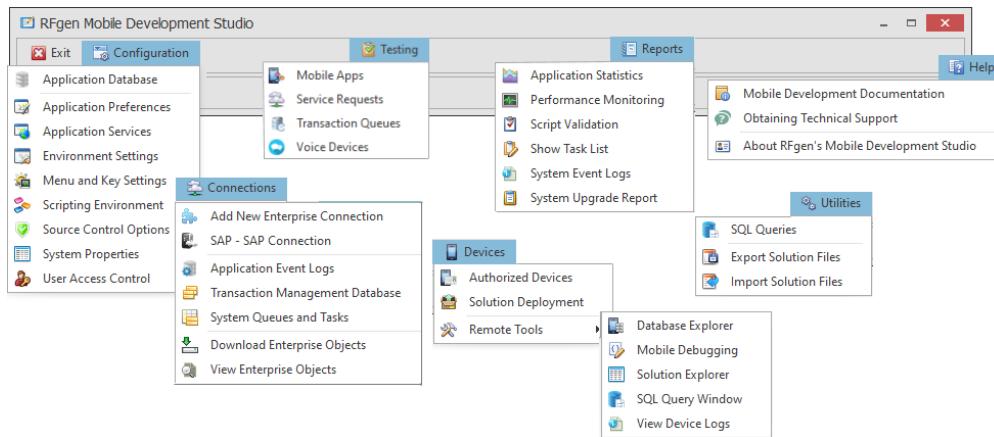
- The **Application Logs (Logfiles)** display system errors. It includes a SQL filter, and Export to Excel tool.

- The **Performance Monitoring** log lists events generated by the execution of scripts that exceeded a threshold value. (i.e. Flagged events that exceed processing time thresholds.) It includes Display Options and an Export to Excel tool.

Mobile Development Studio Overview



The **Mobile Development Studio** provides all the tools you need to create a mobile solution, test your solution, and manage the client after the RFgen client software has been installed. Below is a quick view of the ribbon menu items.



The top menu enable you to: a) Configure the RFgen software; b) Setup Data Source connections; c) Configure as applicable ERP or host or web object connections; d) Configure your Transaction Database connections; e) test your apps and sessions; f) manage devices (i.e. connection sessions to the RFgen server); and g) report on performance.

The **Solution Designer** panel provides the tools to:

- Design and create mobile applications for different methods of data collection.
- Design and view how your screen looks in a target device.
- Script the application.
- Define users and menus to access the application.
- Build installation files for specific client platforms.
- Store and maintain objects for reuse or reference when building applications. (i.e. Globally accessible scripting modules, data transactions, translated terms/phrases, images, and soft keyboards).

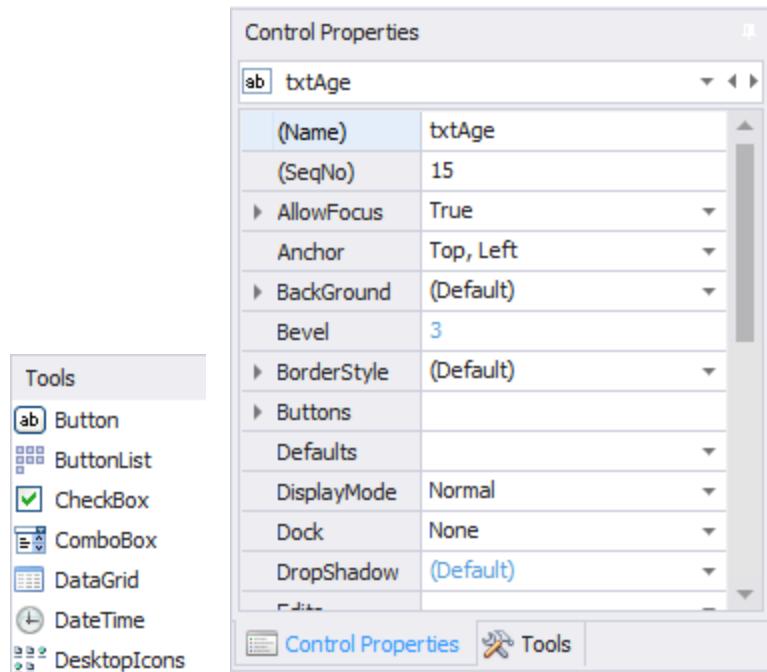
If you have special voice-driven hardware equipment, the Solution Designer can be used to create voice applications that are executed via verbal commands.

Application Tools and Controls

Applications are composed of graphical objects (Forms, Pages, and Graphical Controls), and code (scripts, functions, etc).

When you create a new application in the Solution Explorer, it adds a new **Form** and **Control Properties** for the form. The form provides multiple functions: it allows you to visualize the layout of the objects on your form, provide style elements that are consistent across multiple pages of an application, and contains the various controls which are dragged to the form.

The **Tools** panel contains graphical controls (also called "prompts") which are used to layout the user interface for your application. The Tools panel includes the Button, ButtonList, CheckBox, DataGridView, DateTime, Frame, Image, ImageList, Label, Layout, Line, ListBox, Map, Memo, Menu, Panel, PanelList, RadioButton, Signature, SpinEdit, TabControl, TextBox, RadioButton, Signature, SpinEdit, TabControl, TextBox, and TreeView Control.



The **Control Properties** panel contains the tool/control attributes (background color, size, etc). For more details, see the [Control Properties Tab](#) topic. All controls' property definitions are listed in the [Graphical Control](#)

[Properties](#) help topic as many of their definitions and are common to all controls, but not all property values are shared by all properties. There are some controls that have unique properties not shared by other controls.

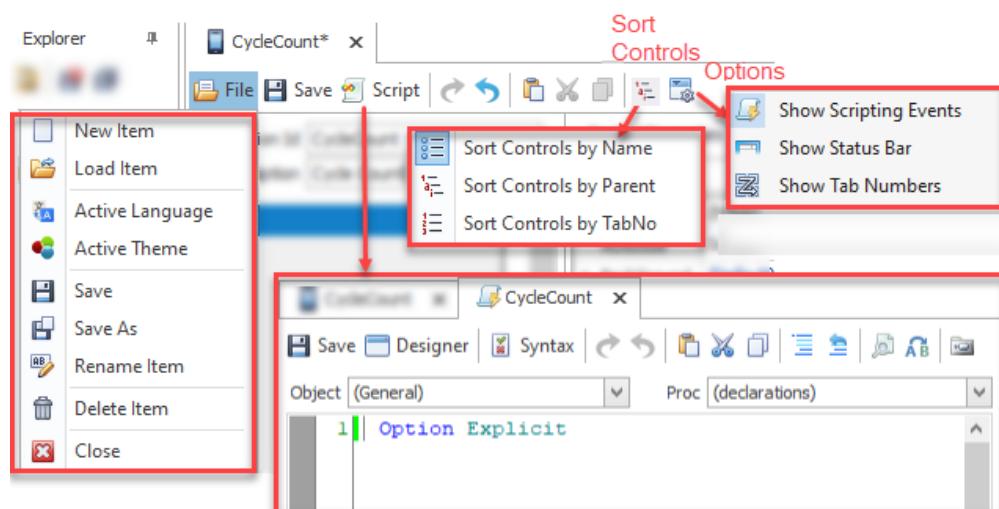
Forms and Pages. When you create a new application, it adds a form. The form's primary function is to add application displays (screen areas), set the general look and feel for the application, create a display for each locale by country if desired, and add pages which can be used to hold multiple controls (labels, textboxes, and other graphical objects). For more details, see [Form Properties](#).

Page and Controls. By default, controls are owned by the page or form they were dragged to.

If you have multiple pages, but only want a control to appear on a specific page, drag the control on to the page and it will only be available on that page. Controls that are on a form and made visible (or invisible) regardless of which page a user is on.

Container Controls. While most controls may be the parent of another control, RFgen also provides graphical Container whose purpose is to maintain the layout order and position of a child control. For example, the Frame, Layout, Panel, and Panellist controls are container controls. Some will display an outline (i.e. Layout control) but won't appear functional until runtime. **Note:** Some controls such as the Layout controls have unique properties when its on a Form instead of a Page.

Applications Menu Bar



File File menu provides the standard functions you use to create a New Item (application form), load it to a folder, rename the existing file, delete, save the file, save the file as a different file, and close the file. The **Active Language** and **Active Theme** display which is local (translated language) and which mobile theme is actively configured from Configuration > Application Preferences.

Script Script opens the VBA scripting tab for the application. This allows you to script your application and switch to the Design view as well.

Undo and **Redo** the last task/action.

Remove Removes the object selected.

 Copies and pastes the last item copied.

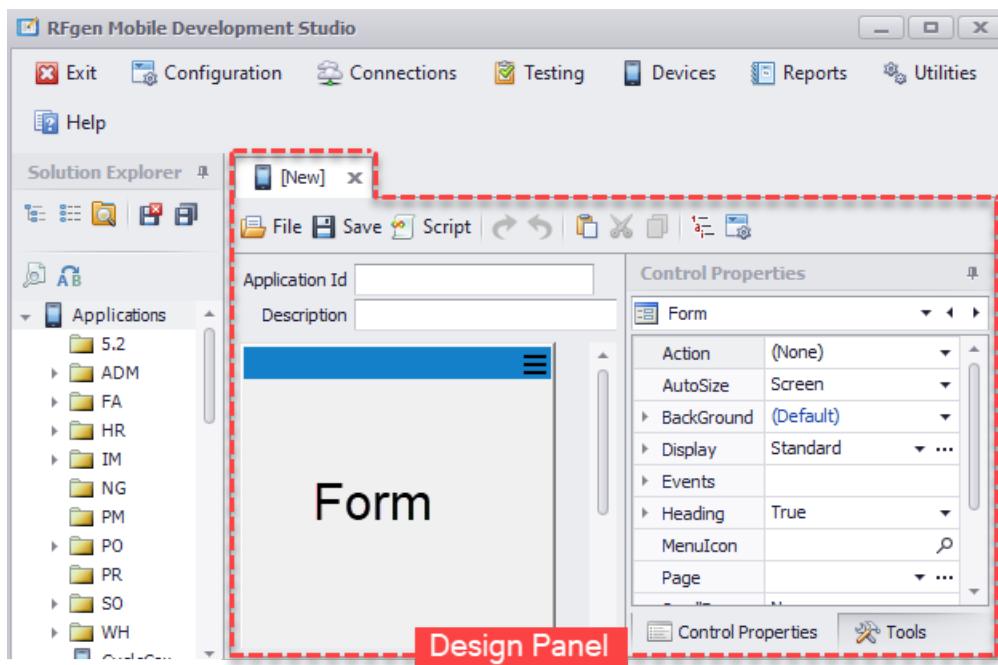
 Sorts Controls by the control's name, its parent (for nested controls), and by the Tab number.

 Options menu enables you to Show Scripting Events adds or removes the Events (i.e. Click, GotFocus, Load etc.) to the control's Control Properties list. Show the Status Bar on your form in the Designer view, and Show Tab Numbers.

To Add a New Application (Form)

Click on **Solutions Explorer** > **Applications** > right-click on **Applications** > select **Add New Applications**.

A blank form we'll call the **Design** panel, and **Control Properties** displays.



Enter your **Application ID** and **Description** and click on the **Save** icon.

Your application is added to the Applications folder and you can begin [adding pages](#) and [controls](#) to your application form, or stylizing it through Mobile Themes.

Form Control Features

The Form acts like a parent template to other application pages, where its properties are both global and unique. Its global in that any graphical object placed on the form is visible to other pages, unless the area occupied by the object is obscured by another graphical object like a solid-colored page.)

For example, if you place the Layout control directly on the Form, and then add pages that towards the bottom of the form and you left the Layout control exposed, the objects inside the Layout control will be visible to all the pages.

The form inherits its look/feel from the **Solution Explorer > Themes: [name of the theme]: Application** settings. The values from themes are used if the child object on the form is exposed. (i.e. If you set a logo in the upper left corner of the form and its not obscured by a page control, then the logo set in the theme will display. For more details, see the [Themes Overview](#) topic.

One of the most important features is the **Display** property where its used to set which device display is used to host your graphical objects. If desired, you can set your displays so that the placement of objects is dynamic or static. For example, if you want RFgen to automate layouts, this means if the the display resolution, size and orientation changes, RFgen will the presentation of the objects relative to the edges of the nested and parent objects and display space. BUT if you want your display to be static, you can setup you displays to be exactly the same as well.

Displays and Locales

You can choose to have an application use multiple types of device displays, and associate a specific locales like Chinese, German, French etc. for each display so an application can be deployed to different devices and locales. (See [Solution Explorer - Devices](#) topic or the [Form Properties: Display property](#) for more details.) Having multiple display types within an app can make it easier to code apps for companies who use a variety of different apps for the same task.

The screen's sizing, automatic translation, number of displays (display types), and look and feel are only controlled by the properties in the form. The form's heading, display of the menu icon= and\or back button < and the menu's appearance and interactive behaviors are controlled in Mobile Themes and by the properties in the Form.

For a description of the properties used on the Form, see the [Form Properties](#) topic.

To Add, Remove or View Pages

To add pages to your form, simply click on the Tools tab in the Solution Explorer, select the Page control and drag it to your form.

For every page you add, a page icon displays on top of the form.

To remove a page, click on the icon on top of the form. Make sure the page in the Controls property is selected, then use the "Cut" icon or "Cut" action from the right-click menu to remove the page.

The Page Control

You set unique properties for each page through the Page Control. The values set here will override values set in the Form or from Mobile Themes. But if the value is set to "(Default)" then its values are inherited from the Form (if the Form's corresponding property value is set) or from Mobile Themes, (if the Form's corresponding property value is set to (Default)).

(Name) - Sets the name of the page. The default name is Page# (i.e. Page1, Page2 etc.)

(PageNo) - Displays the sequence of the page relative to the other pages. For example 1, 2, or 3. To add or remove a page, change the Page property in the Form.

(SeqNo) - A read-only sequence number RFgen assigns to a control. The SeqNo is NOT the same same as PrompNo (prompt number) or TabNo. The VBA Extension "App.PrompNo" returns the SeqNo.

AllowFocus - "True" allows the control to have focus, False does not. If set to True, this adds the DisplayMode property to the Control Properties list. If AllowFocus = True, the **AllowBackup**, **FocusOnClick**, **OnEnter** and **TabNo** properties are added. If AllowFocus = False, then these options are not displayed.

- *AllowBackup* - "True" enables navigation to the prior page, depending on the option selected. For example, the up arrow on a device keyboard, the SHIFT+TAB keys, or a prompt can be set to move the user to the previous page. "False" disables the function.

- *FocusOnClick* - "True" allows a page to have focus at runtime. You can set this to "False" if you want to use the page as a container or "staging" of information or content to be access by other pages but you don't want this page to be viewable at runtime.

- *OnEnter*: *Advance* - The other options you can set when Enter is tapped are: *ExitForm*, *HoldFocus*, *ResetForm*, and *Submit*.

- *TabNo* - Is the order number of the object that can accept focus. This is not the same as the SeqNo (prompt number).

The **Anchor** property sets a child object's position and dimensions relative to where its anchored to the parent. This property is added to the Page Control when the Dock property = None. For more details on Anchor, refer to the [Graphical Contol Property Description](#) topic in the User Guide.

BackGround - Sets the background color and how it is applied (as a solid or gradient) to the page. If set to (Default) the value is inherited from Themes. If the value is NOT "(Default)" then the style and color values were set in the Designer. The percentage of **Transparency** affects how transparent the foreground is while allowing the background to come through.

BorderStyle- For details, see Graphical Control Properties: [BorderStyle](#).

DisplayMode - Sets the DisplayMode option for the page. Unless you are designing the page to contain a specific application state, leave this value as "Normal". The DisplayMode application states are: Attention, Normal, Bold, Disabled, Error, Focus, Information, Link, Normal, Success, Warning. If you want to make sure the properties for a specified DisplayMode are inherited from Themes, make sure the Themes > Application > DisplayMode > Background > Override = True.

Anchor the page and use **Location** coordinates, set Dock = *None*. If you want the page size to be the same as the form's size, use "Fill" which will override the page Size and Location values. If you want the page docked with a limited width, set the doc to left or right. The height will be set automatically.

Events - Allows you to set the event (actions) that can be taken against the page. For information about Event values see the [Event Property](#) topic in the Developer Reference Guide.

Heading - Used to enter the text / page heading. Text entered in "Heading" will displayed only if the **TextId** value is blank (no strings selected). The Heading is populated from the **TextId** if a text string identifier is selected. These are sourced from the Solution Explorer > TextResources > [Name of Object] table. This is also where you can set the heading to use a localized string.

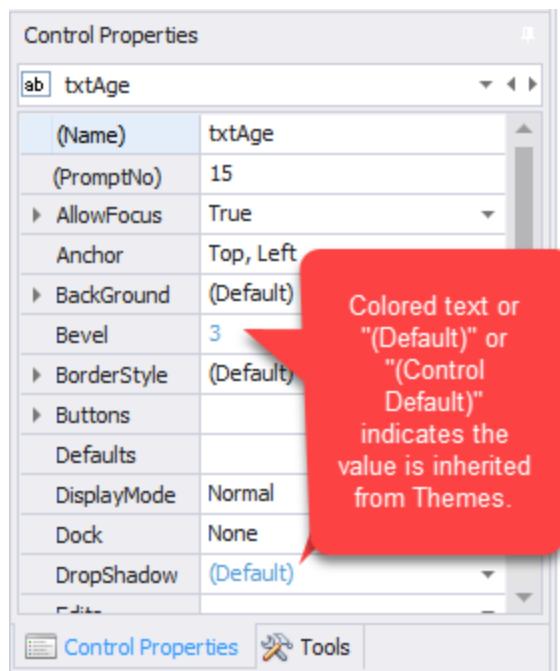
Location- Sets the location of the page (number of pixels to the left of an anchor and number of pixels from the top.)

Size - Sets the page height and width in pixels. If the **Dock** property = Fill, you cannot edit the page height and width.

SystemIcons - Adds or removes icons to the page, and associates specific, system operations with the system icon. The built-in Call Events / operations are Cancel Operation, To add icons / system operation click on the ellipsis (...) which opens the Manage Icon Collections screen. For additional information, refer to the help topic on the [Manage Icons Collection](#). You can also view the icons available through the [SystemIcons List](#) topic. For a list of all the system operations, see the [Systems Operations](#) topic.

Visible - True makes the page visible; False hides it.

Property FAQs



Colored text or
"(Default)" or
"(Control
Default)"
indicates the
value is inherited
from Themes.

Where do I set the property defaults?

If you have an open source database from RFgen, the Control Property values for all controls are preset for you in the Solution Explorer > Themes > [name of the Theme] > [Element] > Property.

If you set the property value in the Solution Designer, that will be the value which is used. BUT if the value is set to "(Default)" then it will use the value which is set to the equivalent property for that control (Element) in Themes. Note that color values use "Control Default".

Why is the text in the property field of the Control Properties panel a different color?

Colored text indicates the value for the property was inherited from Themes. The color of the property value which indicates the value is a default is set by the skin selected in Configuration > Application Preferences > Application Skin (these are provided through Microsoft). Depending on which skin is set, the color may be a color other than blue.

For example, if a Named Color was set in the Application Designer for the Form: Background: Color property, the Color property looks like this:

<input type="button" value="Color"/>	<input checked="" type="checkbox"/> LightSteelBlue
--------------------------------------	--

If a Theme was selected in the Application Designer for the Form: Heading: BackGround: Color (and ColorEx)

<input type="button" value="Color"/>	<input checked="" type="checkbox"/> Theme Color 7
<input type="button" value="ColorEx"/>	<input checked="" type="checkbox"/> Theme Color 7

property, the properties would look like this:

But, if the value is inherited from Themes: [Element Name]: Element Property value, the Application Designer will indicate this with the words "**(Default)**" or "**Control Default**". Here are three examples of properties that are inheriting their values from Themes in the Application Designer:

<input type="button" value="BackGround"/>	<input checked="" type="checkbox"/> (Default)
<input type="button" value="Color"/>	<input type="checkbox"/> Control Default
<input type="button" value="ColorEx"/>	<input type="checkbox"/> Control Default

, where BackGround=(Default), and Color and ColorEx = Control Default.

Is the TabNo the same as the PromptNo?

The (TabNo) property is only available for prompts that can accept focus. It is not the same as the PromptNo property.

The ActiveTheme property is used to specify which Theme the property is to receive its value(s) from should the other properties be set to (Default).

Not all properties appear for all control types. For example Image controls, CheckBox, RadioButton, Page controls and others will have their own unique properties.

Note: To search for information on this page, tap the Ctrl + F keys and enter the name of the property.

Where can I find definitions for the graphical control properties used in the Application Designer?

You can enter the graphical control property name in online help Search or you can also view the list in [Appendix A: Graphical Control Property Definitions](#).

To Move a Control

This topic describes different ways to move a control from a Form to another Page in your application.

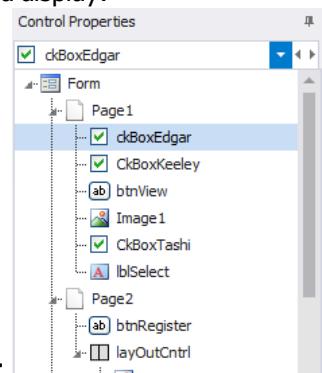
Move by dragging in the Application Designer View

1. Open your application. See [To Add an Application \(Form\)](#) for details.
2. Add the additional pages for your application if you haven't done so already. You should see the page icons display above your application's form.
3. Click on the form or page with the control to be moved.
4. Drag the control to the page icon above the form so that its highlighted. For example, drag your list control from page 1 to page 2. Page 2 is highlighted when its selected.

5. Click on the destination page (i.e. Page 2). An outline of the control's container will display in the header section of the target page. Drag this container down from the header its new location on the page. The control should display once its below the header.
6. Click on Save to save your changes.

Move by dragging in Control Properties Panel View

1. Open your application. See [To Add an Application \(Form\)](#) for details.
2. Add the additional pages for your application if you haven't done so already.
3. From the ribbon menu, click on the  Sort icon and select "Sort Controls by Parent".
- 4.
5. Click on the Control Properties > Drop Down icon. The form, pages and controls nested under each parent should display.



6. Example:
7. Select the item to be moved, then drag it to its parent.
8. Click Save when done.

TIPS for Placing and Sizing Objects on a Form

Before you start, plan which objects you will need in your data collection application, what the workflow prompt order will be, how many pages you'll need. Create a diagram of the various graphics/boxes so you can get an idea of how you want them laid out.

When you are ready, create the app and set the number of pages to be used on the Forms control. Drag-and-drop the desired prompts from Tools into the display area (the page of the form) in the order in which prompts will appear.

You can also double-click on the objects and they will transfer automatically to the display). Note that a data field and its associated Caption (prompt) or label are addressable separately.

The upper left-hand corner of the form/display area is "0,0". The location (Left, Top) of the prompt is relative to the left-hand corner of the form. The next line down is 0,1 (Column 0, Row 1), and so forth. In graphical mode the same applies for pixels.

The easiest way to resizing a control is through the values entered under the Size property. You can also move the control via the icon  and allow the Snap-to-Grid feature align the control with all the other controls on the from.

Once the control is selected as indicated by the Move icon, you can use the arrow keys or your mouse to move the control to a new location. If working with nested controls, grabbing the outside edge of the control will select the parent and child controls.

To resize a control first select it and grab the edge of the control and drag in a direction. Holding down the Shift key and clicking on the arrow keys will move the prompt one pixel at a time.

Holding down <ctrl> while clicking in multiple prompts will select all of them and make moving several prompts easier. Dragging a square around all the prompts to be selected is another method for quickly selecting multiple prompts.

Notice the double arrows located on the Properties tab.  They may be used to select each prompt in the order that they were placed. Click on the left or right arrow to select a different prompt. The combo box provides the same feature in a list format.

To insert a prompt, add the prompt as usual and then change the order on the Properties tab using the Prompt Number property. Another option is to select an existing prompt on the screen that you want the new prompt to follow. Then double-click the new prompt from the Toolbox and it will be inserted in the prompt sequence. To delete a prompt, right-click on the prompt and click on Cut from the popup menu. The prompt will disappear.

To reorder prompts, change the 'Prompt Number' property on the Properties tab for each prompt.

The Button Control



The Button object is used to allow easy access to a function on the application such as Save, Scan, Submit, or Exit. This object can only be seen in the graphical client and supports images on the button itself.

To use a Button, place it on the application and give it the appropriate caption. In text mode, the GotFocus, OnEnter and LostFocus events are executed when the image is next in the tab order. In the graphical mode, the Click event will also execute.

Note: This prompt will never hold the focus. When focus comes to this prompt it immediately processes the event and focus then moves to the next prompt.

For property descriptions, see [Graphical Control Properties](#).

The ButtonList Control



The ButtonList object is a control used to display a list of options in a tiled, vertical or horizontal format. The size of the buttons and the size of the images are independent so they can be sized in any way. The text is also scalable. Horizontal and / or vertical scrollbars can be added but the control supports swiping up and down or left to right. Depending on the size of the control on the form and the size of the overall buttons, columns of buttons will be added or removed based on the available space.

Note: If you want the items to appear as buttons, make sure you set the Style to Buttons, and set the Items: Size property to accommodate the height and width of the Button.

The Size property (not the Item: Size property) sizes the entire ButtonList panel.

The images (icons) are added via script -- for example, they can be added via the List.AddItem extension.

For property descriptions, see [Graphical Control Properties](#).

The CheckBox Control



Final Confirmation?

The CheckBox object is an application prompt that allows the user to select a True or False option based on the object's label.

For property descriptions, see [Graphical Control Properties](#).

For theme properties, see [Mobile Themes CheckBox](#).

The ComboBox Control

Reason Code



The ComboBox object is an application prompt that allows multiple items to be displayed in an area of the application, one of which may be selected. Items may be sorted and/or selected as required by the needs of the application.

To use a ComboBox, VBA script is used to populate and manage items displayed in the box or the ComboBox prompt can contain the values itself by entering them in the List property.

The **ColumnSet** property group is used to add **ColumnSets** (the structure or template to create a column) and set specific properties for each **Column** within a ColumnSet. For example, you can add 3 columns in the

ColumnSet1 group, and in the first column, use CheckBoxes, in the second Column, use Text (to label the checkbox) and the 3rd column, use Memo so the user can enter notes in this column. The **ColumnSet: Columns: [Column#]: (Style) property** enables you to select the format of the item in each column. The values are: (Text), CheckBox, Decimal, Image, and Memo.

To stylize the appearance of a column, see ListOptions.

The **ListOptions property group** provides many features for how the a list of items are presented and selected. Some of the key features are:

- AllowRoll - allows the user to reverse the order to see items that may be at the end of a list.
- AltRow - enables different shading for every other row
- CheckBox - Stylizes the checkbox if this was the "(Style)" chosen for a ColumnSet: Column: [Column#].
- Image - Sets the spacing if this was the "(Style)" chosen for a ColumnSet: Column: [Column#].
- RowStyle - Sets the background color and forecolor text for a row
- SelectRow - Sets the style for a selected row in the list. "(Default)" means it uses the Theme values.
- SortMode - Sets how the items in a list are sorted -- Ascending, Descending or None.

For property descriptions, see [Graphical Control Properties](#).

The DateTime Control



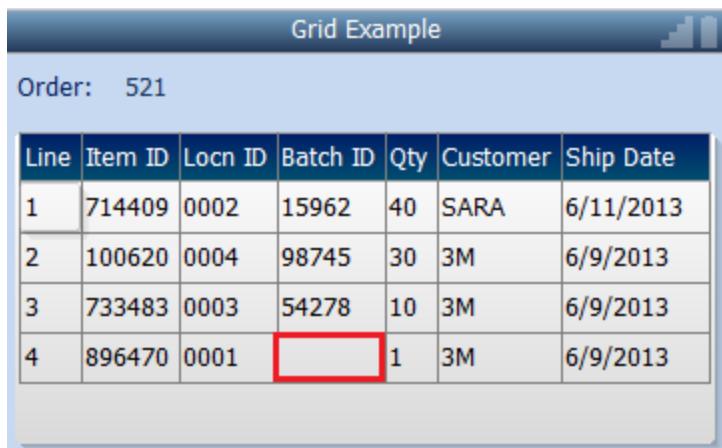
The DateTime control displays the time in 12-hour format. The time is based on the device supplying the application. For example, if the device is in thin mode, the time is supplied from the system that is hosting the application on the device. If the device is running in Batch mode, the time is based on the device itself.

The Caption for the DateTime control defaults to "hh:mm TT".

For details on displaying the date and time in other formats, refer to the Developer's Reference Guide on [VBA Language Extension, Prompt-Specific Extensions - Formats](#).

For property descriptions, see [Graphical Control Properties](#).

The DataGrid Control



The DataGrid object supports binding large sets of data and displaying it in columns and rows so that users can view, select and edit the data.

Data edits (adding, modifying and deleting text in a cell at runtime) are supported in the DataGrid -- scripting for data edits is not required.

To add, populate, and manage the contents of the DataGrid, use VBA scripts.

Refer to *VBA Language Extensions, Prompt-Specific Extensions* such as **List** for details.

For property descriptions, see [Graphical Control Properties](#).

The Frame Control



The Frame object is used to put a box around other prompts to give a grouping effect. The Frame prompt sequence must come before the prompts that will be inside the frame. Otherwise, the frame will cover the prompts on the inside.

The frame can be stretched to be either a single horizontal line or a vertical line and its caption is optional.

Note: This prompt will never hold the focus. When focus comes to this prompt it immediately processes the GotFocus, OnEnter and LostFocus events and focus then moves to the next prompt.

For property descriptions, see [Graphical Control Properties](#).

The Image Control



The Image object is used to display a picture on the application. This object supports a large variety of image formats in a Thin Client environment. They are BMP, DIB, GIF, JPG, WMF, EMF and ICO. This object can only be seen in the graphical version of the client. When using the Windows CE / Mobile environment, this control only supports the BMP format.

Image property - This property is the container for your resource image and its size is controlled by the Size property and its placement is indicated in the Location property.

LockAspect and Stretch properties - These two properties work together to enable you to control how an image is resized relative to the Image property which contains the image resource.

- If LockAspect and Stretch are set to True, then the image will stay proportional and not become distorted when the Image container is resized enlarged or shrunk or when change the dimensions of the Size property. If you enter an odd height and width combination, the image will automatically adjust its size to retain its aspect ratios.
- If LockAspect is set to True, and Stretch is set to False, then the image size will not be stretched or resize when the Image control is resized. Instead, the image will retain its original resource size. If you want space between the image and container borders, use this setting. For example, if the image was 219 x 219 pixels, but the container was sized to a width of 400 and a height of 100, part of the image would be cut off and there would be space between the edges of the image and the image container.
- If LockAspect is set to False, and Stretch is set to True, the image will stretch to the edge of the Image container. So if your image was 219 x 219 pixels, and you entered width=400 and height=100, the image will stretch out but become distorted.
- If LockAspect and Stretch are both set to False, the image will still retain its aspect ratio and not resize.

Other Image Control Property Descriptions

For property descriptions, see [Graphical Control Properties](#).

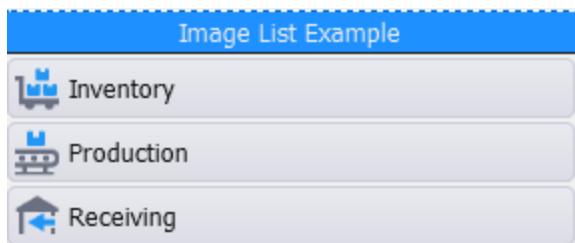
Image Control and Events

The GotFocus, OnEnter and LostFocus events are executed when the image is next in the tab order or when the user clicks on the image. In this case, the Click event will also execute.

At runtime check the Defaults property to get the name of the image resource currently loaded into the prompt. If it is blank the image may have come from the file system using the RFPrompt().ImagePath property.

Note: this prompt will never hold the focus. When focus comes to this prompt it immediately processes the events and focus then moves to the next prompt. If it is simply clicked on, the focus will stay where it was before the click.

The ImageList Control



This object is a control used to display a list of options in an icon list format. The image size dictates the size of the row's height. The text is also scalable. Horizontal and / or vertical scrollbars can be added but the control supports swiping up and down or left to right. If a title row is not required, it can be turned off.

Images (icons) are added using scripts (i.e. List.AddItem).

For property descriptions, see [Graphical Control Properties](#).

The Label Control



Labels are used to add text for informative, breadcrumbing, or decorative purposes on the app.

When focus comes to this object, it immediately processes the GotFocus, OnEnter and LostFocus events and focus then moves to the next prompt. It never holds the focus.

Labels can be formatted so they appear on one line or wrapped (MultiLine).

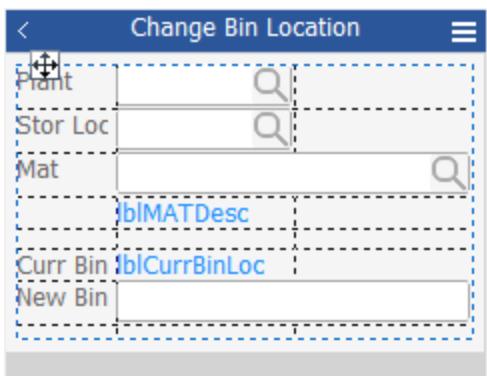
The **Caption** of a label can:

- hold hard-coded text. (TextId is left blank and DataLink = False.)
- be left blank and at runtime, be populated with text sourced from another object (i.e. TextBox). (**DataLink** = True; **LinkTo** = source object name.)
- be left blank and at runtime, be populated with the translated TextId. (**DataLink** = False; **TextId** = TextID in Text Resources.)

The **Layout** property ONLY displays when the label is placed on the form. Its absent when the label is on a page. When present the **Col** (column), **Row**, and **Span** properties are used to specify which columns or rows the label property will span for decorative and content sizing purposes.

The properties used here are similar to the ones used by all the other controls. For property descriptions, see [Graphical Control Properties](#).

The Layout Control



The **Layout** object is a parent control that helps keep child objects (i.e. labels, textboxes etc.) aligned when a prompt changes its size at run time. While the Layout control cannot receive focus, its child objects can.

In a Layout control, RFgen adjusts the widths of columns and heights of rows using the space available inside the control to calculate the alignment of objects. RFgen allocates the available space using these factors:

- The number of columns and rows present
- The contents of each cell
- The property (or methods) assigned to each column or row

When assigning values, note that the Layout control's overall width and height cannot be made larger or smaller by the Column and/or Row's width/height values.

For example, if your application was translated from English into German and the text expanded 25%, the Layout control would automatically adjust the text and its adjacent controls' position so they remain aligned to each other.

This auto-alignment feature saves development time creating separate layouts of an application that will be localized into different languages. It also saves the developer's time positioning individual controls when the application is used on devices with small or large display areas. For example, if your device changes from a small to large screen the Layout control will also scale with the anchor points.

Layout Control Properties

(Name) - Sets the name of the Layout Control.

(SeqNo) - the sequence order number of the object on the app.

Anchor - Sets the coordinates of where a page is docked/anchored for screen sizing purposes. The points where this is anchored is important as this also increases or decreases the width and height of the Layout control and subsequent resizing of the objects inside the Layout control.

AutoSize - Sets whether the objects inside the Layout control will scale/resize.

- "**(None)**" - disables this feature.
 - "**Content**" - expands the row (or cell) when the objects contained by the row expand or contract. For example, if the form is set to be translated, the Local = German, the longer text string in a Label could expand and push the TextBox on the same row further to the right, automatically. But if the Local = English, then the

TextBox would move back closer to the Label so there isn't as large of a gap between the two.

- **"Height"** or **"Width"** - sets the AutoSizes only to the width or only to the height of the object in the Layout control.

BackGround - See the [Graphical Control Properties](#).

Columns - Allows you to add, move, remove and stylize columns in the Manage Column Collections screen. For information on the (SizeMode), Margins, Value, and Visible, see the [Manage Column Collections](#) topic.

Columns: LinkedToPages - Available only if Layout is on the Form. This property enables the child object in the Layout control to be linked to the specified page. At runtime, the object is visible on the page its linked to.

Rows - Allows you to add, move, remove and stylize columns in the Manage Rows Collections screen. For information on the (SizeMode), Margins, Value, and Visible, see the [Manage Column Collections](#) topic.

Rows: LinkedToPages - Available only if Layout is on the Form. This property enables the child object in the Layout control to be linked to the specified page. At runtime, the object is visible on the page its linked to.

DisplayMode, Dock, Events, Location, Size and Visible are defined and used similarly to other controls. For property descriptions, see [Graphical Control Properties](#).

Rows

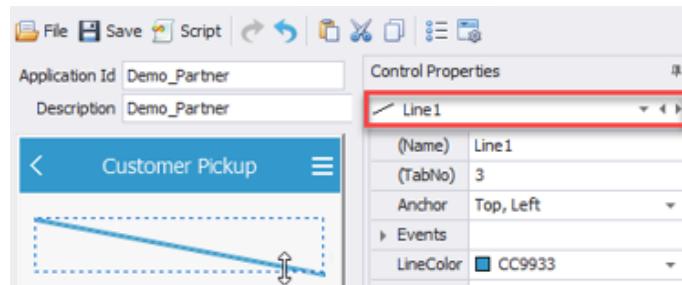
When you add or remove columns or rows, the total count (collection count) of columns and rows are displays respectively in brackets { }.

The total amount equals the number of columns or rows set in the property group. The count is not affected by the settings in Configuration > Environment Properties > Item Collection is one-based. You can not assign names to the column or row.

See "[How to use the Layout control](#)" for more information.

For property descriptions, see [Graphical Control Properties](#).

The Line Control



The **Line** control is drawn via the dual x, y rectangular coordinates of the line.

Line Properties

(Name) - The name you assign to this control. If none is provided, the name defaults to Line1, Line2 etc.

(TabNo) - The assigned sequence of the control. Child controls are renumbered under each parent.

Anchor - For property descriptions, see [Graphical Control Properties..](#)

Events - See [VBA Events](#) topic in the Developers Reference Guide.

LineColor - Theme colors are derived from Mobile Themes. Named Colors derived from the web. Custom colors are the RGB colors you assigned. If the Custom RGB = 0, then the Custom value is using the Control Default which is set in Mobile Themes.

LineSize - The thickness of the lines in points.

Visible - True makes this visible; False makes it invisible.

Point0 - the starting point of the line.

X - The vertical distance between the start of the line and anchor property value (i.e. Top, Left of form).

Y - The horizontal distance between the start of the line and horizontal anchor (i.e. Top, Left of form).

Point1 - The ending point of the line

X - The vertical distance between the end of the line and anchor property value (i.e. Top, Left of form).

Y - The horizontal distance between the end of the line and horizontal anchor (i.e. Top, Left of form).

Visible - "(Default)" uses the value set in Mobile Themes. "True" will display the line; "False" hides it

The ListBox Control

Entre	Prep	Weight
<input type="checkbox"/> Roast Beef	rare	2oz
<input type="checkbox"/> Salmon	smoked	5oz
<input type="checkbox"/> Chicken	BBQ	5oz

The Listbox object is an application prompt that allows items to be displayed in an area of the application. Items may be sorted and/or selected as required by the needs of the application. A Listbox can have multiple columns, selection styles (spinners, checkboxes, or none).

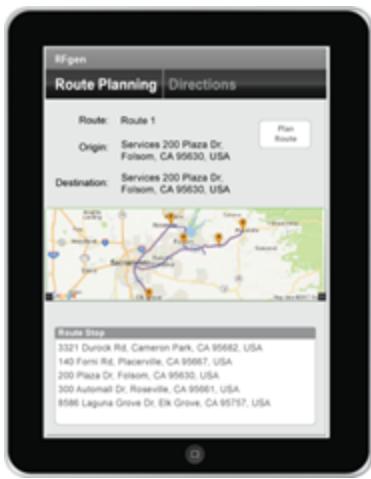
To use a ListBox, select the desired layout and selection styles from the Columns property and use VBA script to populate and manage items displayed in the box. The ListBox prompt can contain the values itself by entering them in the 'ListData' property and separating each column using the pipe character (|). Or, you can enter data using scripts.

A ListBox is different from a SearchList object. Clearing the application prompts and displaying items in a list in full screen mode uses a SearchList.

Note: If you would like to convert the Listbox to a PanelList, you can right-click on the object and select "Convert to PanelList." This action will literally convert your ListBox control into a PanelList Control.

For property descriptions, see [Graphical Control Properties..](#)

The Map Control

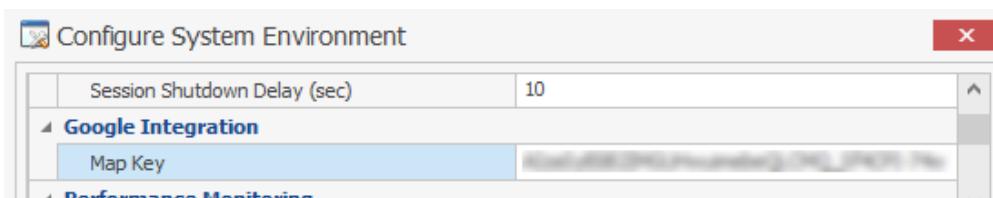


The Map object is a control that provides a subset of the features available from Google maps and global positioning station (GPS) APIs. You can use it to create applications that:

- Calculate the most efficient route from a set of addresses or GPS coordinates.
- Display a map of a route plan.
- Convert GPS coordinates into an address or converts an address to GPS coordinates.
- Zoom in or out via the "+" plus and "-" minus signs. (The "+" and "-" buttons are present in the graphical design view, and are functional at runtime.)
- To use the Map object requires VBA scripting, an API key provided by Google maps, and registration of your app and API with Google API Console. To learn more about obtaining an activation license key, go to the following URL and click on "Paid".
- The Map object cannot hold focus and therefore does not have a TabNo (tab number).

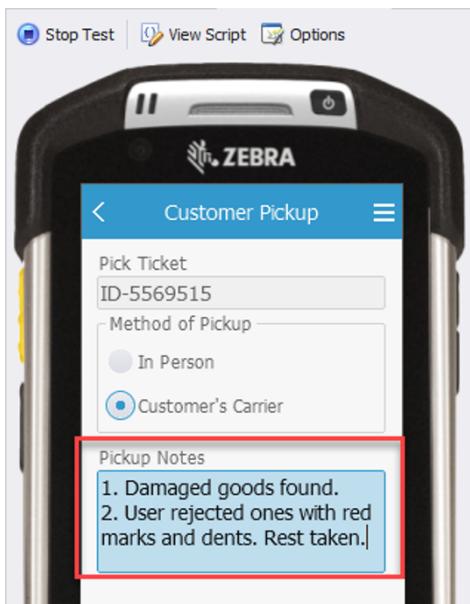
<https://developers.google.com/maps/pricing-and-plans/>

Once you have obtained your activation license, enter it under **Configuration > Environment Properties > Google Integration Map Key**



For property descriptions, see [Graphical Control Properties](#).

The Memo Control



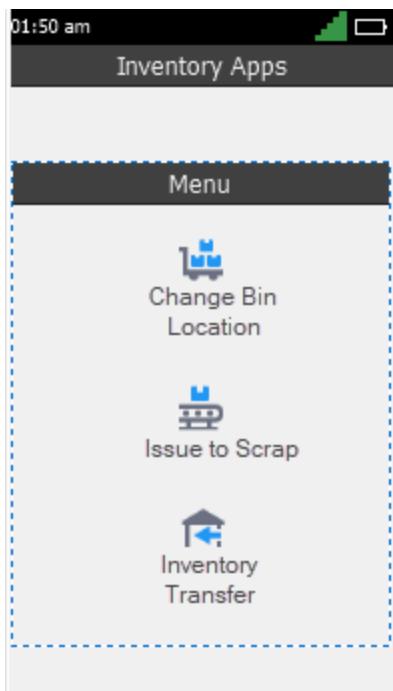
The Memo object is similar to a textbox except that the end user can enter a lot more characters at runtime than a textbox. It can also be used to display the error message entered in the memo ErrorMessage prompt at runtime, or the content via a script.

You can also display the search icon in the textbox to enable searches of data the textbox is linked to. To enable the search icon, in the application's Script tab, select the OnSearch event from Procedures.

For property descriptions, see [Graphical Control Properties](#).

The Menu Control

This control allows you to arrange the look and feel of applications menus and is also designed to work with the RFgen Menu and Roles objects in the Solution Explorer pane. Once you design details such as the menu arrangement style type, you can also visualize how it will look on a device when populated with all the menu items in the Menu Simulation section of Menus and Roles.



The **Style property** is used to select how you want the menu items arranged. The values are: icons, icons + Text, Text only (no icons), as buttons or as panels (icons and text).

The **Image property** is used to set the size and margin values for icons/image. Note that your icons resources are assigned to an application(s) in the Menu and Roles Solution Explorer.

The **Items property** provides a container and the shape of the menu object (i.e. If you selected the "Panel" as your Style, then you set the panel look and feel under Items).

All the other properties have a similar usage as the properties in the other graphical controls.

For property descriptions, see [Graphical Control Properties](#).

The Panel Control



The Panel Object is a parent control that enables you to easily perform group actions such as deleting, hiding

or moving all child controls at once by deleting, hiding or moving the parent Panel control. It also makes it easier to apply common grouping values such as the background.

When placing a child object on the panel, make sure its owner is listed as the **Panel**.

Note: Since the Panel is a container, it does not hold Focus and therefore does not have a **TabNo** (tab number).

For property descriptions, see [Graphical Control Properties](#).

The Page Control

You set unique properties for each page through the Page Control. The values set here will override values set in the Form or from Mobile Themes. But if the value is set to "(Default)" then its values are inherited from the Form (if the Form's corresponding property value is set) or from Mobile Themes, (if the Form's corresponding property value is set to (Default)).

(Name) - Sets the name of the page. The default name is Page# (i.e. Page1, Page2 etc.)

(PageNo) - Displays the sequence of the page relative to the other pages. For example 1, 2, or 3. To add or remove a page, change the Page property in the Form.

(SeqNo) - A read-only sequence number RFgen assigns to a control. The SeqNo is NOT the same same as PrompNo (prompt number) or TabNo. The VBA Extension "App.PrompNo" returns the SeqNo.

AllowFocus - "True" allows the control to have focus, False does not. If set to True, this adds the DisplayMode property to the Control Properties list. If AllowFocus = True, the **AllowBackup**, **FocusOnClick**, **OnEnter** and **TabNo** properties are added. If AllowFocus = False, then these options are not displayed.

- *AllowBackup* - "True" enables navigation to the prior page, depending on the option selected. For example, the up arrow on a device keyboard, the SHIFT+TAB keys, or a prompt can be set to move the user to the previous page. "False" disables the function.

- *FocusOnClick* - "True" allows a page to have focus at runtime. You can set this to "False" if you want to use the page as a container or "staging" of information or content to be access by other pages but you don't want this page to be viewable at runtime.

- *OnEnter*: *Advance* - The other options you can set when Enter is tapped are: *ExitForm*, *HoldFocus*, *ResetForm*, and *Submit*.

- *TabNo* - Is the order number of the object that can accept focus. This is not the same as the SeqNo (prompt number).

The **Anchor** property sets a child object's position and dimensions relative to where its anchored to the parent. This property is added to the Page Control when the **Dock** property = None. For more details on Anchor, refer to the [Graphical Contol Property Description](#) topic in the User Guide.

BackGround - Sets the background color and how it is applied (as a solid or gradient) to the page. If set to (Default) the value is inherited from Themes. If the value is NOT "(Default)" then the style and color values were set in the Designer. The percentage of **Transparency** affects how transparent the foreground is while allowing the background to come through.

BorderStyle- For details, see Graphical Control Properties: [BorderStyle](#).

DisplayMode - Sets the DisplayMode option for the page. Unless you are designing the page to contain a specific application state, leave this value as "Normal". The DisplayMode application states are: Attention,

Normal, Bold, Disabled, Error, Focus, Information, Link, Normal, Success, Warning. If you want to make sure the properties for a specified DisplayMode are inherited from Themes, make sure the Themes > Application > DisplayMode > Background > Override = True.

Anchor the page and use **Location** coordinates, set Dock = *None*. If you want the page size to be the same as the form's size, use "Fill" which will override the page Size and Location values. If you want the page docked with a limited width, set the doc to left or right. The height will be set automatically.

Events - Allows you to set the event (actions) that can be taken against the page. For information about Event values see the [Event Property](#) topic in the Developer Reference Guide.

Heading - Used to enter the text / page heading. Text entered in "Heading" will displayed only if the TextId value is blank (no strings selected). The Heading is populated from the **TextId** if a text string identifier is selected. These are sourced from the Solution Explorer > TextResources > [Name of Object] table. This is also where you can set the heading to use a localized string.

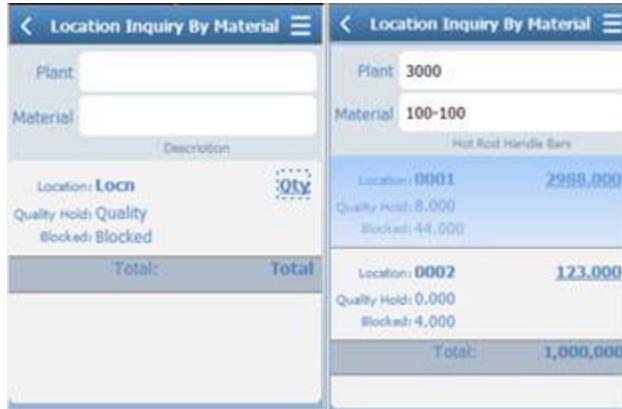
Location- Sets the location of the page (number of pixels to the left of an anchor and number of pixels from the top.)

Size - Sets the page height and width in pixels. If the **Dock** property = Fill, you cannot edit the page height and width.

SystemIcons - Adds or removes icons to the page, and associates specific, system operations with the system icon. The built-in Call Events / operations are Cancel Operation, To add icons / system operation click on the elipsis (...) a which opens the Manage Icon Collections screen. For additional information, refer to the help topic on the [Manage Icons Collection](#). You can also view the icons available through the [SystemIcons List](#) topic. For a list of all the system operations, see the [Systems Operations](#) topic.

Visible - True makes the page visible; False hides it.

The PanelList Object



The PanelList control creates rows of panels much like a ListBox except each panel may contain a variety of child controls arranged in any order on a panel and, if the control is data-centric, it can be bound to a column. Multiple objects bound to columns on a panel form a set of columns (called a ColSet) that can be populated with data.

Using the PanelList Control

Simply set the number of panels you will need in the **Panels** property using the + or - buttons.

Each click of the + button adds a **PanelRow** object which can then be stylized and used to hold other controls.

The PanelList and PanelRows can be treated as templates that will contain the data from a database so that at runtime, your application can have specific areas (i.e. PanelList) dedicated for static information (Plant Name & Location), and PanelRow1 dedicated to the presentation of data from multiple records (Different categories of Items, inventory counts and bin locations), and another PanelRow dedicated to the presentation of total counts for all records totaled in PanelRow2. So at runtime, the user could see the static information, multiple panels (one for each record), and a total count as the final panel.

When designing the app, the **ActivePanel property** is used to display the **PanelRow** you want to work with. For example, if you added 3 panels, the designer will add 3 child PanelRow objects as PanelRow1, PanelRow2, and PanelRow3. If you only wanted to work on the design of PanelRow2, then you would set the **ActivePanel** property to PanelRow2.

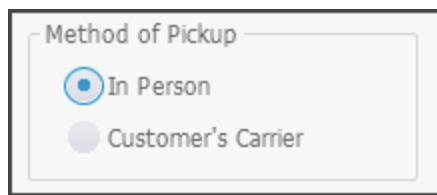
You can also use the Control Properties dropdown menu to select the PanelRow you want to design.

When you add child controls such as a label or textbox to a PanelRow, the **BindToColumn property** in the label or textbox controls displays. Use the BindToColumn property to set the control to specific column in the PanelList or a specific PanelRow. Note that columns are added via VBA Language Extensions in the script view of the app.

The other properties in the PanelList and PanelRows are similar to those used in other controls.

For property descriptions, see [Graphical Control Properties](#).

The RadioButton Control



The Radio Button allows the user to choose an option from a group of related options and where the selected option is mutually exclusive. Use a container control such as a Frame to group the RadioButtons.

The [Graphical Control Properties Definitions](#) section describes properties for all the controls. This includes ActiveTheme, AllowFocus, Anchor, BackGround, BorderStyle, Caption Disabled, Events, Location, Size and Visible.

See [Mobile Themes: RadioButton](#) for a description of the properties unique to this control.

The Signature Control



Example of the Signature Control



Example of the Signature control at runtime with a hand-written signature

The Signature control is used to capture any hand-written text that will then be saved as a two-tone bitmap image.

The image can be stored to a bmp file by using the VBA Language Extension **Image.Bitmap**. The Bitmap property contains a byte stream representing the signature which can be written to a BMP file if desired, or stored in a binary type field in a database.

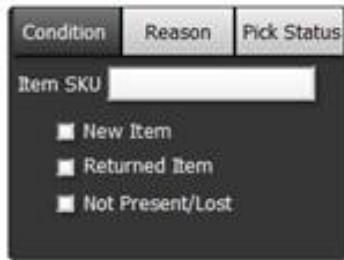
The image can also be placed in the **Text** VBA Language Extension of the Signature prompt so that the image (Signature) can be re-displayed. The Text property of this prompt will contain a text representation of the signature which can easily be stored in a character type field in a database.

Note: There is no built-in pattern matching to compare recorded signatures.

For information on the VBA Language Extensions available for the Signature control, refer to the *RFgen Developers Reference Guide* which can be accessed from the Mobile Development Studio Online Help.

For property descriptions, see [Graphical Control Properties](#).

The TabControl Control

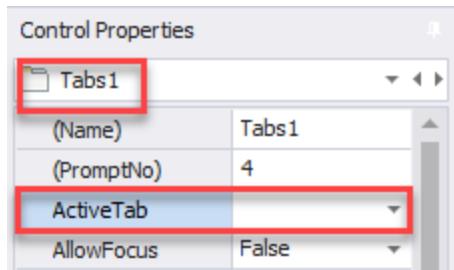


The **TabControl** enables users to toggle between pages or prompt groups at runtime. No coding is required to enable switching between tabs. This control consists of a parent (i.e. Tabs1) and child objects (TabPage1, tabPage2...). The TabControl allows you to stylize the tab pages so they look similar, or customize them so each looks different.

You can also add icons to each tab (tab stub) using the Image property, and translate them from a text resource via the TextID property.

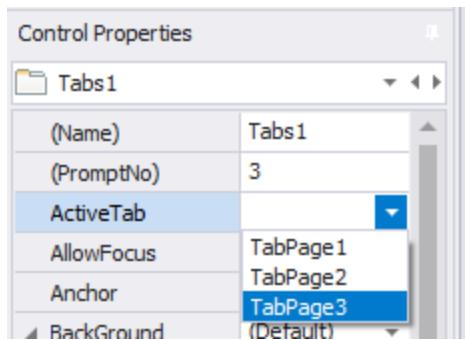
How to work with the Tab control

The Tabs parent properties are used to set the look and feel for the child TabPages when the **Active Tab** property is blank.

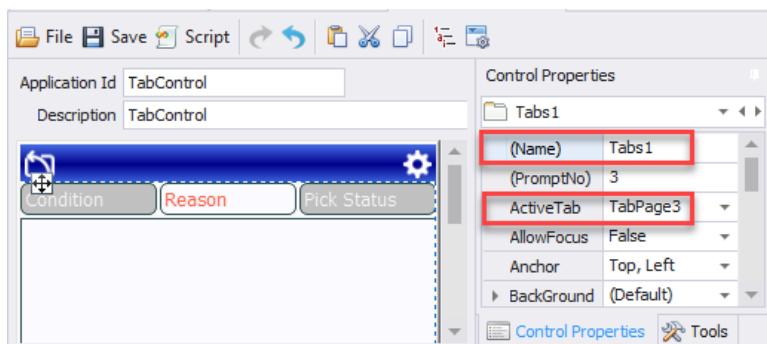


For example, in Tabs1 Control Properties, you can set the BackGround > BackColor to White, and the TabOptions > Active > ForeColor to Orange. Then all the child tabs will have a White background and Orange text inside the tab stub.

But if you need to **set some exceptions for a specific child TabPage**, you can do this too. Use the **Active Tab** property drop down menu to select theTabPage and make your changes.

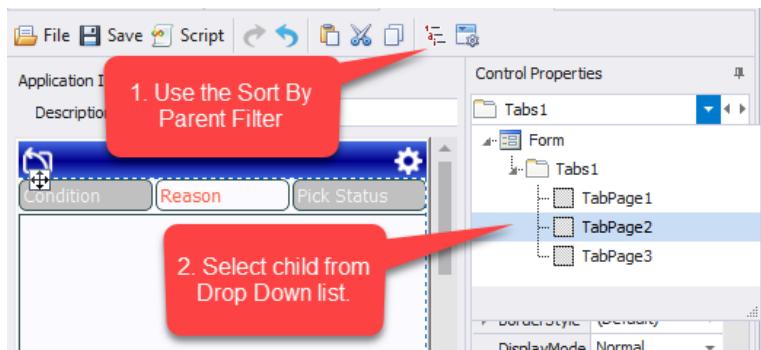


If you selected a child tab under **Tabs1**, you are now able to overwrite the Parent Tab1 property settings for the selected TabPage. For example, if you want to change the Background color ONLY for TabPage3, but allow the other two children to use the styles set in Tab 1, then your Control Properties selection would show (Name): Tabs1, ActiveTab: TabPage3.



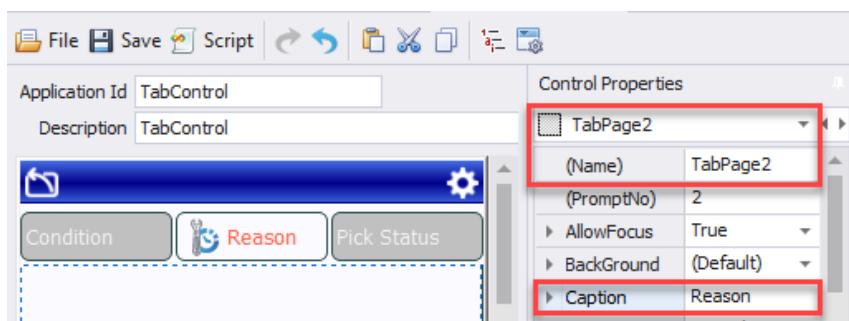
To Set the Caption or other Specific Values

To set specific values for a TabPage, (i.e. the caption), then you need to select that TabPage so its name appears in the Control Panel. There are two ways to do this: a) Use the Sort Control by Parent View to make the child object selection, or b) while viewing the Tabs1 control, double-click the page at the bottom.



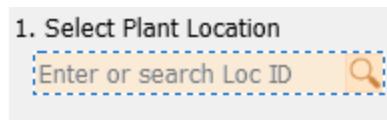
The easiest to use is with the Sort by Parent view.

In this example, **TabPage2** was selected, so the Caption, TextID (for translation), and an image icon can be added to the TabPage.



For property descriptions, see [Graphical Control Properties](#).

The TextBox Control



The TextBox is a container (orange box above) which has many practical uses in an application. Its most commonly used to prompt the user to enter data by:

- clicking the search button inside the textbox and selecting a value from a table
- entering data values from a scanned barcode
- entering data via a soft or physical keyboard

Additional Features

The TextBox also had graphical properties that go beyond using the TextBox as a prompt for data entry purposes. Here are a few examples:

- You can use it to hold text without having focus on the prompt. (Set **Allow Focus** = False.)
- You require data be entered before the textbox loses focus via the **Required** property.
- Assign other icons and actions to an event beside **Search** inside the TextBox via the **Button property group**. On some platforms the Event accesses the device's system operation like scanning/camera.
- Set pre-formatted or customized text via the **Defaults** property so repetitive data values such as the location of a warehouse isn't re-enter for every transaction.
- Link Error Messages to customized text via the **ErrorMessages - TextId** property for easier error handling in your script.
- Launches a virtual keyboard (also called a soft keyboard or soft input panel).
- Link a data a user entered to a Label's caption via the **LinkLabel** property.
- Instead of having a Label control, use the **NullText** inside the textbox to label what to enter.
- Automate the TextBox to automatically accommodate the fontsize when **IntegralHeight** is set to True.

For more details on the properties used here, see the [Graphical Control Properties](#) topic.

To enable Search in the TextBox control

1. Make sure the AllowFocus is set to True and Visible is set to True.
2. Set Buttons > Action to Search.
3. Set Buttons > Image to Search. (This will display a magnifying glass icon.)
4. Set Buttons > Visible to True. (So the image will display.)
5. Set the Alignment location for the Search icon (left or right side of the textbox).
6. Set Events > OnSearch to True. This will open the Script designer and add the OnSearch procedure.
7. Enter your script for the Search text.

To enable Scan in the TextBox control

1. Make sure the AllowFocus is set to True and Visible is set to True.
2. Set Buttons > Action to Event.
3. Set Buttons > Image to Camera. (This will display a Camera icon.)
4. Set Buttons > Visible to True. (So the image will display.)
5. Set the Alignment location for the Camera icon (left or right side of the textbox).
6. Set Events > OnScan to True. This will open the Script designer and add the OnScan procedure.
7. Enter your script for the OnScan procedure.

The TreeView Control



The TreeView object enables developers to present data in an outline format where information is sorted hierarchically. Each group (also called a parent) may contain child items and parent "containers" can be expanded or collapsed for ease of navigation to other parent items.

This object is very similar to the ListBox object except that the Tree supports multi-level, indented lists that can be expanded or collapsed at run time. For example to add data and manage the contents of the Tree, use VBA prompt extensions such as List.AddItem.

For property descriptions, see [Graphical Control Properties](#).

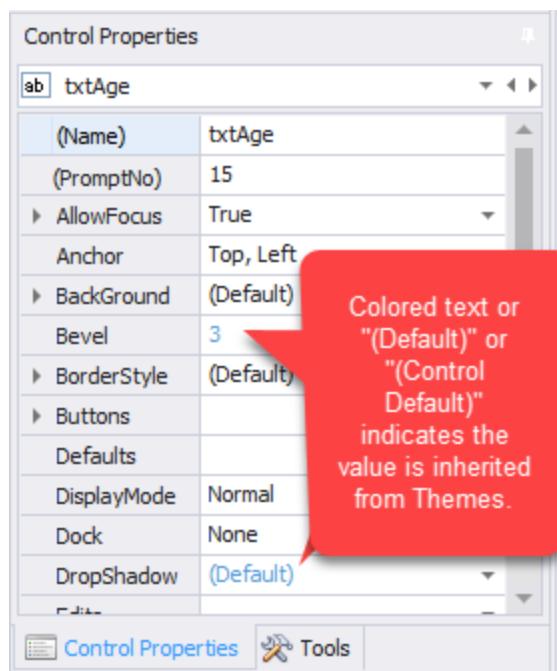
Graphical Control Properties

Descriptions for all the control properties are listed alphabetically on this page. Tap **Ctrl + F** keys to **Search** for a property definition on this page.

The **(Name)** property is a standard property for all controls except the Form.

The **(PromptNo)** property is the Prompt sequence value RFgen assigns to objects placed on a form or a page. If you move the location of a control or page, the PromptNo also updates, or you can edit it directly in the Application Designer Control Panel. If your script uses App.PromptNo VBA extension, it should return the PromptNo value for the specified control.

How to tell if Themes is being applied



If the value for a property appears in colored text or the value shows (Default) or Control Default this means the property value is using the corresponding value set in Themes.

Control Property Descriptions

Not all properties appear for all control types. For example Image controls, CheckBox, RadioButton, Page controls and others will have their own unique properties.

The **Action** property under the **Button** property provides built-in operations for buttons in controls like the TextBox. (For example, Button > Action = Search and Button > Image = Search can be used with the OnSearch event to display a list of items created from script.)

The **Alignment** property places text and images relative to another object within a control. For the CheckBox and Radio Button controls, the images are shifted relative to the labels when you select a position in the Alignment drop down menu. (See example below)

The screenshot displays two separate 'Control Properties' windows side-by-side. The top window is for a 'RadioButton' control (RadioButton1) and the bottom window is for a 'CheckBox' control (CheckBox1). Both windows show the 'Alignment' property highlighted with a red box. The 'Alignment' dropdown menu is open in both windows, showing options like Middle, Left, Middle, Right, etc. The status bar at the bottom of the interface also highlights the 'Alignment' property. The left sidebar lists various control types: Application Id, Description, RadioButton, RadioButton, RadioButton, CheckBox, CheckBox.

For the Button control, Alignment positions the Caption text and the image within the button space. To separate the image from the Caption text, use the Image > Alignment to position the image relative to the Caption text. For the Image control, the image is positioned relative the edges of the image box.

The **AllowFocus** of set to True, enables the function/operation described below. "True" also enables you to select actions for focus. False prevents the control from receiving focus, removes a TabNo assignment or prevents one from being assigned. If set to True, you can enable these options:

- *AllowBackup*- to the previous control;
- *AutoSelectText* - enables selection of text where the prompt lands;
- *EraseOnBackup* - will erase the text if user backs up
- *FocusOnClick* - enables focus via a mouse click in addition to tapping a Tab key;
- *OnEnter - Advance* - dropdown option for: **Exit** the form, **Hold** the focus, **Reset** the form (i.e. Clear entries), or **Submit** when the user taps the Enter key.

The **AllowRoll** property is used on controls that contain long lists of items and helps you "jump" to items faster. For example if you press the up arrow while on the first line in a long list of items, the last item from the list will display. Or, if you are at the bottom of a list, and press a down arrow, you are taken to the top of the list.

The **AlternateItem** property group sets the background color, font color (ForeColor), visibility of the colors for alternate rows in a list (ComboBox, DataGridView, ListBox, PanelList etc.). To display the alternate color, set Visible to True.

The **Anchor** property sets a child object's position and dimensions relative to where its anchored to the parent. For example if a TextBox control was anchored to a Form on all four sides, the TextBox would be proportionally resized if the Form was displayed on a different mobile device. But if the TextBox was Anchor equaled "Bottom", "Right", then its size won't change, and its position will stay relatively the same. If no other anchors except the Bottom is set, then RFgen automatically centers the child object inside the parent. If the parent is resized, the chile object's size is not changed even if the parent's size changes.

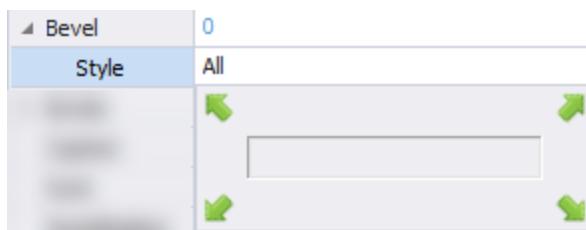
The **AutoSize** property will stretch the object's background to the lowest and right-most portion of the screen, depending on the option selected. This property is available for the Button, DateTime, Frame, Label, and Panel controls. The values are: (None), Content, Height, and Widthl. If AutoSize is set to "(None)" then you can drag-n-resize the control. But if the control is in AutoSize mode, the ability to drag-n-resize the control is disabled and the control will be sized in accordance to its contents. For example, if you have a Panel control that was empty (no controls inside it), and AutoSize was set to "Contents," "Height," or "Width," then the Panel will automatically resize to 0 pixels. If there was an image in the Panel, and AutoSize was set to "Contents" then it would resize the Panel around the image relative using the right-most and bottom portion of the panel as the start location. See the Size property for the size in pixels. See Manage Paged Collection for details on AutoSize Property for Pages.

The **BackColor and BackColorAlt** (previously called **BackColor(1)** and **BackColor(2)**) properties are used to create either solid backgrounds or gradients depending on the option chosen in the Background Fill property. The color can be set from using Custom Color tab or by a 6-character hex value (which gives you 16 million colors to choose from).

The **BackGradient** This property was replaced by the **BackGround** property.

The **BackGround** property has three categories for setting the background color of a control. If set to Solid, RFgen applies the value from Color 1. If a directional values such as Diagonal Right, Diagonal Left, Vertical etc. is selected, RFgen applies the values from **Color 1** and **Color 2** to create a gradient background. If Transparent is selected, the next-closest background color behind the control will be used. For example, if your Mobile Theme Application BackGround = Red, and your Mobile Theme Label BackGround=Transparent, and your Label Caption color = White, then your Label text will appear as white on a red background in Mobile Themes. In the application designer, the Label text will also appear as white text on a red background if the application's Form Active Theme = (Default) and the Label Active Theme = (Default). But, if your Mobile Theme Label BackGround = Solid and the colors 1 and 2 = Pink, then in the solution designer, the label's background will appear as white text on pink because uses the Label's solid background color blocks the background color from the Application Form.

The **Bevel** (graphical mode only) property sets the curvature of a square's and rectangle's corner edge where 0 is no bevel and 100 turns the object into a circle or oval shape. The **Bevel Corners**



The **BindToColumn** is only used with data-centric controls on PanelRows in the PanelList control. Its used to bind a data-centric controls such as a Textbox or Label to a specific column. For example, if your first column is 1, then the **BindToColumn** value should be "1". The list values can be ordered to start with 0 or 1; You can force the ordered list to start with "1" by checking the box under Configuration > Environment Properties > Environment > List Items Collection is One Based.

The **BorderStyle** (graphical mode only) property controls sets the style of an object's boarder. The (Default) is Flat. Options are: None, Flat, Sunken, Raised, Thick, and Underline.

The **BorderStyle** (graphical mode only) property controls the border of the prompt. Options are Standard, Active Border, No Border, Visible with Focus and Transparent.

The **Brush** property is used to select a color that will override a Theme color.

The **Button** property group stylizes the search icon button that displays inside the TextBox when the OnSearch event is set to True.

The **Button Pressed** has been obsoleted in 5.2. This property was used to set the color values for this control when its selected. In 5.2 all press states styles (colors) are now calculated from the colors assigned to the button so there is no need to configure them.

The **ButtonSize** property was removed in version 5.1 of RFgen. See the **Size** property.

The **ButtonStyle** property was obsoleted in 5.2.

The **Button Pressed** property was obsoleted in 5.2. All press states are now automatically calculated to apply a selected appearance using the BackGround coloring of the button.

The **Caption** property is used to hold text that is static or dynamic.

Note that the **TextId** property under Caption is used to populate the contents of the Label or TextBox at runtime, depending on the value that's used. For more details, see the TextId property description.

Caption - LinkTo property - used to mirror the text between the textbox and one or more labels after the user taps Enter. This is accomplished by using the same ID in both controls. To enable/disable the **DataLink** property must be set to True.

The **CellMargins** property sets the distance between the contents of a cell and the cell border in pixels, and is used to make it easier to view multiple lines of text in a control.

The **Checked** property sets the status of a CheckBox prompt.

The **CheckBox** property group is used to stylize the elements in the CheckBox, ComboBox, DataGridView, ListBox, and TreeView controls. To enable the checkbox as a subfeature in controls used for listing text and values in columns or grids, you must select the "CheckBox" value in the control's Column > (Style) property. The CheckBox property group includes the BackGround, BorderStyle, ForeColor, Margins, and Size (for sizing the checkbox).

The **Colorize** property, if set to True, converts simple grey-colored images to the color selected in the **Brush** property. For example, the **Colorized Brush** can change the grey in a grey-colored Chevron icon to  to red . This tool is intended for the system icons that are provided by RFgen in the InlineButton property.

The Colorize Brush is NOT intended for changing the color of images or icons comprised of multiple colors as it will apply the same color to entire image.

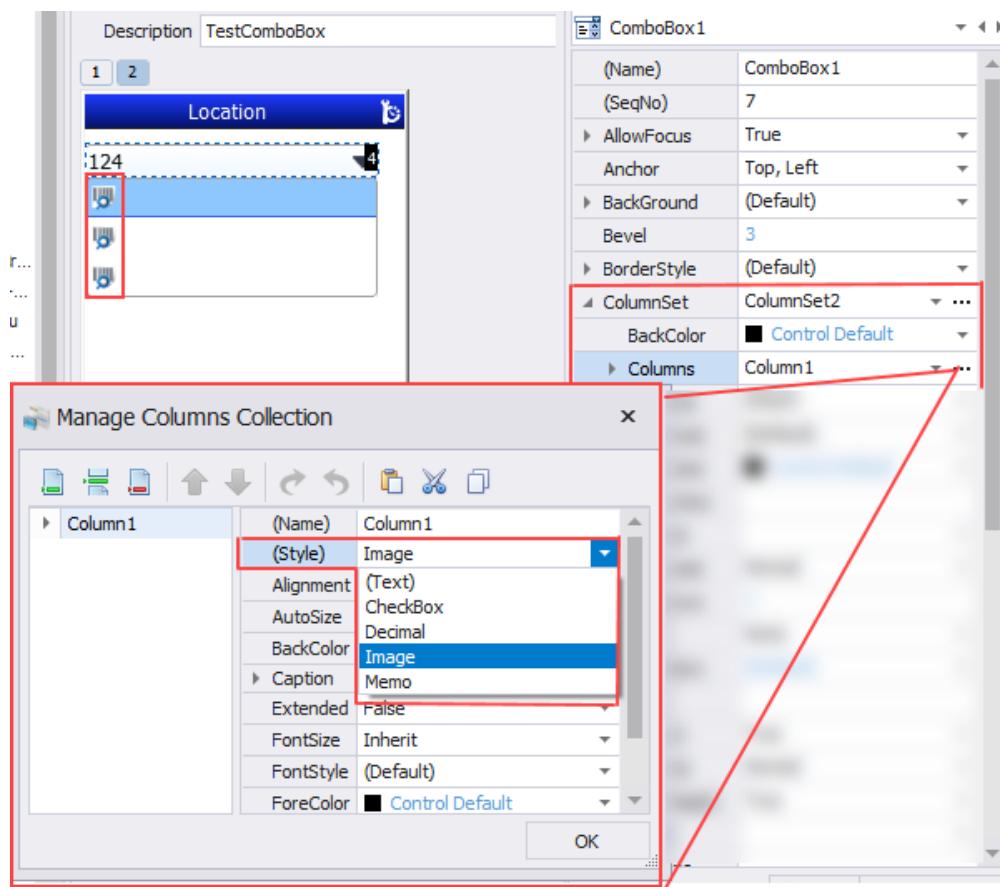
The **ColumnSet** property group (available in the ComboBox) is used to set the (template) of a column and the styling of data each column created within a column set.

- a) If you click on the down arrow, a lists of your column sets displays so you can select the one you want to edit;
- b) If you click on the elipsis (...) it opens the **Manage ColumnSet Collection** window and is used to add, insert, delete column sets and set the colors and fonts for each one.

Once your columns have been added and stylized, the subproperties and values will be listed under columns. For more details, see *Manage Columns Collection*.

The **Columns** subproperty group (under ColumnSet property) is used to customize and control the presentation of the data within a specific column. For example, if you can set column 1 to contain CheckBoxes with white text on a black background, column 2 to contain text only with a grey background and red text, and restrict data via TrimSpaces property.

The **Columns (Name)** property sets the unique that for the column for reference purposes.



The **Column (Style)** lists factory-provided styles: *(Text)* for display of read-only values, *CheckBox* for selection of data items, *Decimal* for display of numeric text, *Image* (icon), *Memo* for data entry by the user, or the *SpinEdit* button to add/decrease a value. **TrimSpaces** will remove leading and trailing spaces from the data so column alignment will be smaller. The **Width** property can either be set to a specific size or -1 to indicate that the column should stretch to the right taking up any available space.

The **Color 1** and **Color 2** properties are used in combination to create gradient colors in properties such as the Background property. These values can be set in the Custom, Named Color, or Theme tab of the color palette tool.

DataLink property enables the mirroring of data from a source object (i.e. TextBox) to the destination label. The source is set in the **Caption > LinkTo** property. False disables the link between the source and destination label.

The **Defaults** property sets any number of built-in values or custom values as the initial value of the prompt. For more details, see [To Set Text Defaults in a Control](#) and for a list of text default options, refer to the topic [Text Default Options](#) in the Developers Reference Guide which can be downloaded from the RFgen Online Help and Documentation Page. The Defaults property appears for property groups such as EditText which are typically in the TextBox and Memo control.

The **DefaultList** property sets any number of built-in values or custom values as the initial value of the prompt. For more details, see [To Set Text Defaults in a Control](#) and for a list of text default options, refer to the

topic [Text Default Options](#) in the Developers Reference Guide which can be downloaded from the RFgen Online Help and Documentation Page. The DefaultList property appears in the ListBox control.

The **Display** property. See the [Form Display property](#).

The **DisplayMode** property provides a list of factory-provided styles (from Themes) that can be used to standardize the appearance of objects.

The **Dock** property is used to automate the placement and scale of sibling objects under a parent object. While the Anchor property is also used to position and set sizing relative to whether its anchored, the docking property forces sibling to respect the space of neighboring objects so you can dock items in a series.

For example, if a Label and TextBox were placed above a Button control and all three were on the same page, and their dock properties = Top and Fill, RFgen would first align the Label and TextBox to the top of the screen and space them out to fill the screen's width. Then RFgen would position the Button control to fill in the remaining area of the screen.

DropShadow places a dark border around the bottom and right sides of the control for a 3D effect.

The **Edits** property sets any number of built-in values as the requirement for the entered data. For example, it can be used to validate the data entered manually or data that was scanned to this field. This property is usually a member of the TextBox control or Memo control EditText property group. For a list of Edit Property Options used to validate text or perform other checks against a source, see [Edit Property Options](#) in the RFgen Developers Reference Guide.

The **EditText** property sets the color of the text (i.e. text inside a memo) that a user can edit at runtime. EditText may contain these subproperties used to stylize text at runtime: Alignment, FontSize, FontStyle, and Margins. It may also include the Edits property for inserting values that perform validations against the data entered in the prompt.

The **EntryRequired** property, set to True, forces users to enter data into the prompt, while setting it to False, allows users to skip the field. If the prompt never gets the focus, this property will not get used.

The **ErrorMessage** property is text displayed as an App.MsgBox when the data entered fails to meet the criteria in the Edits property. You can link this to a text resource if you enter the **TextId** which is stored under Solution Explorer > Text Resources.

The **ExtendedColumn** property specifies which column will be stretched to the right edge of the control. The default is the last column designated by -1 but specifying 1, 2, or 3 as examples would use the remainder of the width by stretching a middle column.

The **Events** section contains a list of Visual Basic for Applications (VBA) scripting events. The default values are set to False. If you select True, the script view for your application displays and the selected event object's script module is added. For a description of an Event, see the [VBA Events](#) topic or enter the Event's name in the Search field above. If Events is hidden, click on the Show Scripting Events in the [Options Menu](#).

The **FocusStyle** property specifies how an object shows it is the active/selected object at run time. The default is Standard. Other options are: Active Border or Visible w/Focus.

The **FontSize** property (graphical mode only) defaults to a baseline value called "Normal" which is set in **Mobile Themes > Application > FontSize**. You increase or decrease the font size (in points) if you want

to override the default value. For example, if your Normal = 11 points but you want your Labels > Caption text = 14 points, you would set your Labels > Caption > FontSize to +3. In controls where the FontSize property is nested into subproperties, you may see "Inherit" used instead of "Normal".

The **FontStyle** property (graphical mode only) sets the prompt's data field display to a particular style. The default is Normal. Other options are combinations of Bold, Italic and Underline.

The **ForeColor** property (graphical mode only) allows the user to select from a color pallet or enter a 6 character hex value (for 16 million colors) to set the fore color of the label caption of the prompt. For the controls such as the Button control, the ForeColor property will also apply to the icon, if the Icon: Colorize value is set to True.

The **Format** property is an extension of the VBA Format command and pre-formats the entered data to the mask entered here. See the VBA Format command for examples. The double quotes are not necessary as they are in the VBA Format command.

The **FrameStyle** property lets the user create rectangles, vertical or horizontal lines for the frame control only.

The **GenerateMember** property was removed in RFgen 5.2. This property helps improve an application's performance by telling RFgen what it should or shouldn't generate. The values are: *None*, *Member Only*, *Event Only* and *All* (which is Member + Event).

None will NOT generate a member variable; *Member Only* generates a member variables with no events; *Events Only* generates object events only, but no member variables; and *All* generates both member variables and object events. The default for prompts like the Textbox and Labels is "Member Only," whereas prompts like Buttons, DataGridView, CheckBoxes, ComboBoxes, Maps, MenuList and Signature will default to "All."

The **Heading** property group consists of additional properties that are used to stylize the header of a control. You can turn on/off the display of the header (Visible = True), change its BackGround, BorderStyle, add text/caption to the Heading field, and set the FontSize, FontStyle, FontColor, margins, and associate the text/caption in the Heading field with a text resource ID for translation purposes.

The **Heading Caption** property group includes: Color - sets the caption text color and color of any characters used in the form header. FontSize and FontStyles set the caption text font style and size. The Margins (left, right, top and bottom) are in pixels and set the distance between the caption edge of the header. For example a margin of 20 pixels would make the header bigger.

Heading Icon Property Group - is used to add icon buttons and stylize the icon in the header. Icons are added for navigation or menu access purposes. The position of the icon(s) is based on factors such as how many icons were added, icon size, margin and padding allocated to the icon, and positioning (alignment setting) of the header text.

- Heading Icon: Margins - TBD
- Heading Icon: Pressed - TBD
- Heading Icon: Size - The width and height of the icon in pixels.
- To add an icon - TBD

The **Image** property group is available in the ButtonList control and Menu control. For the **Image control**, the Image property group is used to select an image resource, select its alignment within its container on a form, page, or inside another control, and set how its proportioned when its resized. For more details, see the [Image Control](#) topic. For all the ButtonList, ImageList, and Menu controls, the Image property group sets the size and margin for the image(s) listed in these controls but does not allow selection of an image resource. For the **Button control**, the Image property group is used to select an image resource, set the alignment, margins, and size within a button.

ImageID has been deprecated in 5.2. In 5.1 it was used to select an image resource.

ImageMode has been deprecated in 5.2. In 5.1 it was used to position the image (i.e. Top-Left, Top-Cener, Top-Right, Disabled, Stretch, or Tile)

ImagePath has been deprecated in 5.1. In 5.0 this property set the file path to an image located on the hard drive instead of the Resources > Images folder.

ImageSize has been deprecated in 5.1. In 5.0, it set the width and height of the graphic itself regardless of the size of the control. Images can be displayed a number of ways and this property sets the image size for graphical lists, button or desktop menu lists.

InLineButton property group - Is present only for the ComboBox, Map, SpinEdit, and TextBox controls. In the TextBox, this property group enables you to add a customized button that is associated with an event and stylized how the icon looks. For the other controls, it simply stylizes the buttons that come with the controls.

InputState puts the prompt in a state where its "ReadOnly" (no data can be inputted to this prompt). "Disabled" also prevents the prompt from receiving inputs. "Normal" allows inputs (i.e. write data) to the prompt.

The **IntegralHeight** property dynamically sets the height relative to the fontsize it contains and prevents manual changes to the height of the control if the IntegralHeight Property is set to True. For example, the TextBox height will change relative to the TextOptions FontSize if you change the FontSize from Normal to +32. You can change the TextBox's location and width it if you want to.

If you IntegralHeight to False, the TextBox size does not size automatically when you change the TextOptions FontSize, but you can manually make the TextBox taller and change its location and width it if you want to.

The **Items** property is used with the ButtonList, ComboBox, Menu, and other controls that list items. It provides a set of subproperties for styling the elements in a group -- some of which may not be used. These subproperties are: BackGround, BoarderStyle, ScaleText ("True" scales with the size of the item; False uses the Text FontSize), Separation (distance in pixels between items), and Size (height and width in pixels). See also "Selected Item."

The **Keyboard** In controls where user entries are accepted (i.e. Memo or TextBox control), this option can be set to bring up a soft keyboard for input when the text box gets the focus. Note: Keyboard characteristics are set in Solution Explorer Keyboards.

The **KeyField** property is for linked textboxes only and designates which prompts will be used as key fields when attempting to perform an internal SQL Update statement for the linked application. This property is automatically filled in when the user downloads a table or view structure and links the application to that structure.

The **Layout** property is added to a child control when the child control becomes a member of the Layout control. The subproperties are: Col, ColSpan, DockingMode, Row, and RowSpan.

The **LineColor** property selects the color of the lines between rows or columns in a control that supports multiple rows or columns.

The **LineSize** property sets the thickness of a line in the LineControl.

The **LineStyle** property is used for list type controls that also use the Columns property. As the name implies, it makes horizontal and/or vertical visible or keeps them hidden.

The **LinkLabel** property links TextBox entries to be the specified label so that the entries are mirrored to the label at runtime.

The **LinkToPages** property is available under the Columns property and Rows property of the Layout control when the Layout control is on a form. If the Layout control is on a page, this property is not present. It creates a link between the objects in the column or row of the Layout control and the pages of a form. For example, if you linked Row 1 to pages 1 through 3, you would see the contents of Row 1 on pages 1 through 3.

The **ListData** property is for list boxes, combo boxes and list views only and contains a collection of values to be assigned to the prompt when the application loads.

The **ListHeading** property allows the code environment to overwrite the caption of the prompt with formatted data from a database lookup using the Prompt.List.SetColumn method.

The **ListHeight** property is for combo boxes only and sets the number of rows the control will use when displaying a list of possible values.

The **ListSorted** property is for list boxes, combo boxes and list views only and keeps the contents of the list sorted.

The **ListStyle** property changes the presentation of the data displayed between a Standard text list, an Image List that uses images next to the text description, Buttons or Desktop style like a Windows desktop. This is the control used on the internal RFMenu form.

The **Location** property sets the position of the control in pixels for graphical applications. The location is relative to the parent container and may have different values for different controls. For example the Layout Control location is identified by the number of pixels from the top-left of a form. But if a button was inside the Layout control, the button's location , whereas a and in rows and columns for fixed-length character applications.

The **Logo Property Group** is used primarily in [Mobile Themes - Applications](#).

The **Manage Collection** property. See Columns Property.

The **Manage Columns Collection** is available for the **Columns** property in the ComboBox, Layout, ListBox, and DataGrid controls. Its used to add, insert, and remove columns. In controls used to list text and values, the Collection helps you design and stylize columns. You can stylize text and the control's background color, set your alignments, set the caption, enable or disable "TrimSpaces," and allow the columns to be extended, formatted, and visible (or hidden). In the ComboBox or Layout controls , these properties are not present.

In the Layout control (used for containing other controls) the Manage Columns Collection is designed to add,

insert, and remove columns and assign column names and SizeModes (values that set how the object in the cell of the Layout control will be sized), and whether the column is hidden or visible. For more details, see *How to use the Layout Control*.

The **Manage Displays Collection** property. See [Manage Displays Collection](#) topic.

The **Manage Icons Collection** property group is used to add or remove icons, and if in a list, set the list order.

The **Manage Rows Collection** property. See [Manage Columns Collection](#).

The **Margins** property is used to pad the spacing between the rows or images of the displayed data.

The **MastInput** property (available only in the TextBox control) is used to mask the input with asterisks if the value is True. The default is set to False.

The **MaxWidth** property is used to set the maximum space allowed for the control. For example, the Radio Button maximum width is the widest space allowed for the button and its text label.

The **MenuIcon** property on the Form is used to provide a background image.

The **Multiline** property is now called "TextOptions". For details, see "TextOptions."

The **(Name)** property is the internal name / identifier of a control/prompt. Tip: As a best practice, follow the Hungarian notation where textboxes are named 'txtPart' and list boxes are named 'lstParts' as examples. This way, when referring to them in the script, there is an inherent understanding of what types of data will be used for the prompt.

The **NormalizeText** property will trim the spaces from both sides of the displayed data or captions of the buttons or desktop icons.

The **NullText property** - Replaces WaterMark property. Its used to help users know what should be entered, but is not retained as a data value for transactions.

The **Overflow** property specifies which way the remaining items will be displayed. If there are more items than will fit on the device's screen this option can be set to horizontal or vertical which means the user can swipe bottom-to-top or right-to-left to access the remaining data.

The **(PageNo)** property lists the order sequence of a page on a form.

The **Password** property, for the data field portion of the prompt, sets the display of the text equal to asterisks (*) instead of clear text.

The **PenColor** property sets the color of a signature in the Signature control. In Mobile Themes Signature element, the pen color inherits the value from the ForeColor property.

The **PromptNo** property is the Prompt sequence value RFgen assigns to objects placed on a form or a page. If you move the location of a control or page, the PromptNo also updates, or you can edited it directly in the Application Designer Control Panel. If your script uses App.PromptNo VBA extension, it should return the PromptNo value for the specified control.

The **Required** property is used in prompts such as the Memo control where input is required before the user can continue to the next page, prompt or task in the application.

The **RowAltColor** and **RowSelector** (TreeView Control only) properties sets every other row to the color selected and enables users to select the row (True) as opposed to just viewing the content in each row.

The **ScaleToFit** property increases or decreases the icon captions to fit inside buttons or desktop icons when the overall size of the button or desktop tile size is changed. This feature is enabled when its True and is disabled if set to False.

The **Scrollbars** can be enabled for horizontal scrolling, vertical scrolling, both, or none. If set to Automatic, RFgen calculates and displays the scrollbars for you. This property is for select controls only. See [Mobile Themes > Scrollbar](#) for more details.

The **Selection** property group is used stylize a selected item from a group of items or list in a parent control such as the ButtonList, CompboBox, DesktopIcons, ImageList, ListBox, or PanelList. The subproperties include BackGround, Border, Text, Transparency and Visible. To compare against the unselected item properties, see Items property topic.

The **ShowBorder** property will hide or show the border of an element. The values are True, False, or (Default). (Default) uses the property value set for the control (element) in Mobile Themes. *True* will display the border; *False* will hide it.

The **ShowInForm** is a child property of the Menu: Heading property. If the Heading property is set to True, then ShowInForm can be set (True/False) to suppress the menu's heading caption. **True** hides (suppresses) the Menu's own heading, and the Form's heading is displayed as the Menu's heading. **False** will enable the menu's heading and remain visible if the Menu control object is not obscuring the Form's header. If the menu object was set to the same dimensions of the form (and overlayed the header), then Form's heading will be used as the Menu's heading even though ShowInForm is set to false.

The **ShowLines** property will hide or show the lines between rows and columns. The options are (Default) which uses the theme properties, None for hiding all the lines, Horizontal for showing only the lines between rows, Vertical for showing only the lines between columns, and Both for showing lines between rows and columns.

The **SelColor** refers to the color of the selection bar shown in controls like the combo box or list box. The highlighted value is what will be chosen when the user presses the enter key.

The **Size** is the background height and width of the control in pixels. If AutoSize is used, it can influence the height and width values. In the ButtonList control, this sizes the buttonlist container-- not the items in the list.

The **Sorted** This was replaced by SortMode in RFgen 5.2.

The **SortMode** property is used in list controls and will sort content in accordance to the value selected. The values are "(Default)", None, Ascending, or Descending.

The **Source** property (the HostScreen control only) selects an executable to be emulated within the Host Screen control.

The **StretchImage** is used to either shape an image to the size of the control or allow the image to be its natural size whether it fits in the control or not.

The **Style** property is used in the ButtonList control and specifies whether the items in the ButtonList are to be presented as a squarish buttons or icons (text and icon but no button as a container).

The **SystemIcons** property group is used to associate RFgen-supplied actions (Call Event) with a customized icon or a RFgen-supplied icon. For example, if you wanted to include a Cancel operation in your application, use the functions/properties in this group to select the Cancel operation and also associate it with an icon. When you are done adding icons, a value in brackets {} shows the total number of icons associated with the control. For specific details on how to link a customized icon to a Action, see [To link SystemIcons with system operations.](#)

The **TabNo** property is cursor/prompt sequence number for controls that can accept the focus. This property is not the same as the SeqNo property. For example, a Label control cannot accept focus but have a SeqNo but will not a TabNo. Since a TextBox can have focus and have a SeqNo and TabNo. You can edit a TabNo so to force where the cursor goes after a specific object loses focus.



To view the list of controls/objects' tab numbers, click on the Sort icon or see [Sort Controls](#).

The **Text** property group is used in the ButtonList Control and specifies the color of the font, size of the font, style of the font, and position via the margin values. Whether the values here are overridden by other property settings in the control will depend on how deep this property is nested.

The **TextID** property links a text (word or statement) resource to the object. It can have dual uses, depending on the value used. If the identifier is the TextId from the table in the Solution Explorer > Text Resources folder, its used to translate the text string. If the value is preceded with a "%", it links user inputs from the source TextBox to a destination Label at runtime.

The **TextOptions** property sets whether the text will be on a Single Line or Multiple Lines.

The **Theme** property changes the border of the title bar area to one of several hardcoded styles.

The [Title] property in Themes > Dialog is used stylize in the title of a Dialog box (pop up messages dialog box).

The **Transparency** property sets the level of transparency of a child control and bases its coloring on the parent (i.e.Form, Page1, Panel). "(Default)" uses the level of transparency set in the corresponding Mobile Theme element. "None" will use the item's BackGround color. A percentage value (5 %, 10% etc), sets the transparency level.

The **UseMenuTheme** property will override the local properties and apply the default theme properties for the menu control.

The **ValidationTable** property presents a list of downloaded tables that can be used to verify that the data entered already exists in this table and the Validation Field. The two properties must be used together.

The **ValidationField** property presents a list of table fields specified by the Validation Table property. This is the reference field to determine if the data entered in the prompt already exists. If it does not, the Error Message property will be used to warn the user.

The **Visible** property, set to True, makes a prompt or element visible, while setting it to False makes it invisible. Even though the prompt may be invisible, the GotFocus, OnEnter and Lost Focus events will still be executed for this prompt if the focus automatically shifts from a prompt before this prompt to one after this prompt.

The **WaterMark** property group is available for the ComboBox, TextBox, Memo and SpinEdit controls and is used to stylize the appearance of text that appears in the textbox as an example entry for users. There are two ways you can add text (a caption) in the WaterMark property: a) Enable Language Translations (which enables Text Resources) and select a text string from the WaterMark property; or b) Disable Language Translation, and type in your text (caption) into the WaterMark property. If you do not want any text, you can also just leave it blank. At runtime, the user will be able to manually edit the WaterMark text or scan a barcode which will overwrite the WaterMark text.

The **ZOrder** was obsoleted in 5.1.

The **Design Mode** (which used to be under Display tab) has to **Configuration > Desktop Preferences** in version 5.1.

Form Properties Description

You can either set your Form properties values in the Designer, set the values in [Mobile Themes: Application](#).

The **Action** property will associate the form with common application tasks such as Login (logging into the ERP system), Exit (exiting an application), Menu (hosting menu applications), and Resets (clearing the content in a form so you can perform the next task). For all other application uses, leave this value to "None."

The **AutoSize** property will stretch the form's background based on the options of None, Height, Screen, and Width. How the screen appears in the device's display will also depend on Form and Menu properties. For example, if AutoSize = (None), and the Form size is taller than an Android's screen, then the application screen could be scrolled (swiped up or down) to see more of the screen. But if the AutoSize = Screen, RFgen shrinks the screen so there is no additional extra screen to scroll (swipe up or down). If its set AutoSize = Height, it will stretch the screen in proportion to the display area's height. Likewise if AutoSize = Width, it will stretch to match the display width.

The **AutoTranslate** property has been removed. For information on translation of displays, see [How to translate strings in graphical displays](#).

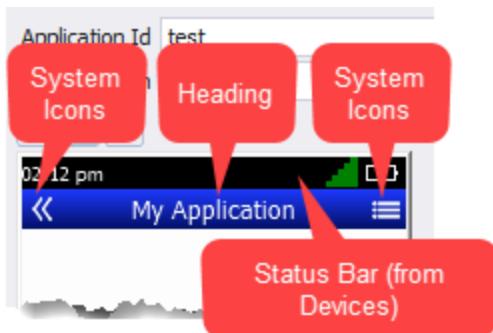
The **BackGround** property will create either solid or gradient backgrounds depending on the option chosen from the drop-down list. "(Default)" means the value is inherited from Mobile Themes. The **Color** and **ColorEx** will use the theme color selected from the drop down list or the value set in Mobile Themes if Color = Control Default (in light grey). You can also override the default with a theme color, a web-named color, or a customized RGB color. If the value is set in the Designer and is not a color that is inherited, the value of the color and the color its set to will display. For example BackGround=Solid; 

Display lists which display is used if more than one was created; adds, modifies, or removes displays, and set the properties for each display. To add or modify a display, click on the ellipsis (...) which opens the Manage Display Collection window. You can have multiple displays associated with a form so that when launching the application, it can be set to display in a locale specific to the user and designed to display for a specific device type (scanner/mobile device). For more details, see [Manage Displays Collection](#).

The **Display** property also list **DPI** is the dots/inch value for a display of a **Target Device**; **Landscape** ("True" sets the orientation as landscape, and "False" disables it); **Locale** (links the application elements for this display to the text resource strings setup in the Solution Explorer > Text Resources table); and, **Target** device (links this display to the specified target device in the Solution Explorer > Devices list).

The **Events** property lists the Visual Basic for Applications (VBA) language extensions used to script additional functions (preset VBA procedures) in your application and for some prompts, set specific, defaulted values. Use this list to globally set for the form the event's value as true or false. For a description of each Event, see the topic [VBA Events](#) in the [RFgen Developers Reference Guide](#).

The **Heading** property contains the text that appears in the heading of a form/page and controls the appearance of the form header.



Whether the text displays depends on these values:

- Heading: Visible =True; and
- If a heading TextId is assigned, it will be used instead of the text entered in the Heading property field.

If the Form Heading: Visible = True, but the individual pages do not have a heading assigned, then the Form Heading's text is used.

If the Form: Heading: Visible = True, and the page has a heading assigned, then the assigned page heading is used. If you have a mix (i.e. Page 1 has a heading but page 2 does not, then the Form's Heading is used if the page doesn't have one assigned.)

If the Form: Heading: Visible = False, then the heading is disabled on the form and all pages, even if the page had a heading assigned.

The heading's look and feel are derived from the theme if set to "[\(Default\)](#)" or "[Control Default](#)". If you set the property, the value will take precedence over one from Themes, and the value will often appear in black text. For values that are inherited these are represented in colored text. For example, if the margins are number-based, and are set in Themes, it may appear as [colored](#) text. For example "Margins: [2,2,2,2](#)." To add icons to the heading see the SystemIcons property.

Note: The height of the black status bar is visible in the form's Display: Target: StatusBar properties. The status bar value is sourced from the Solution Explorer > Devices > [device skin name] property.

The **Heading BackGround** - sets how the gradation of color; Color - sets the form's background color to a theme color, a web-named color, or a customized RGB color if a default is not set.

The **Heading Border** property sets the border color. "(Default)" is set by the Theme: Application: Heading styles. Or, you can set the color.

The **Heading Margins** property sets the space between heading text and the edges of the heading container. The values are in pixels and are for the top, bottom, left and right.

The **Heading TextId** property has a drop down list that links to the text resource identifier from the Solution Explorer > Text Resources > [name of text resource object] > TextId column.

The **Heading Visible** property enables or hides the heading value. See above for details on the rules. False hides it. (Default) will use the values from **Mobile Themes > Form Heading**.

The **MenuIcon** property sets the image for a menu that allows the user to select other applications. If using open source, this can be linked to the menu hosted by the RFgenMenu application . Or, it can be scripted to launch a customized menu of your choice.

The **Pages** property adds or removes the number of pages from the form. If you remove pages, the last pages (highest number) is removed first followed by the next highest.

The **Roles** property is used to limit user access to an application by assigning a role to the application that the user does NOT have assigned to him/her. If Role is assigned a name, (i.e. "Admin, Management, Advanced") in the application, but the user does not have that name assigned to his/her file in the RFgen Development Studio Users screen, then when the application is assigned to a specific menu, and the user is also assigned to that specific menu, the application won't appear in the menu. The absence of the role name in the user's profile prevents the user from accessing the application. If however, if one of the role names (i.e. "Advanced") is present in the user's profile, then the application will show up in the menu its assigned to.

The **ScrollBars** property types are Automatic, Both, Horizontal, Vertical, or None. To customize the colors and line thickness, see Mobile Themes > ScrollBars.

The **Size** property sets the Form's height and width in pixels. One or both values are used if AutoSize is set to (None), Height, or Width.

The **SystemIcons** property associate a system action (i.e. Call Event, Call Form) with the icon that is placed in the Application header. You add and select the subproperties from the Manage Icons Collection screen which displays when you click the ellipsis (...). For more details see [Manage Icons Collection](#).

The **TextOptions** property group sets the look and feel for the text on the form. This group includes the **FontSize**, **FontStyle**, and **ForeColor**. If the FontSize value is set to "Normal" the value is inherited from Themes > Applications > FontSize. If the FontSize is set to a minus or plus value, the font size will be decreased or increased by the value selected relative to the size set in Themes > Applications > FontSize. All values are in points. Refer to the [Graphical Properties Definition](#) topic for the definitions on FontStyle or ForeColor.

The Page Control

You set unique properties for each page through the Page Control. The values set here will override values set in the Form or from Mobile Themes. But if the value is set to "(Default)" then its values are inherited from the Form (if the Form's corresponding property value is set) or from Mobile Themes, (if the Form's corresponding property value is set to (Default)).

(Name) - Sets the name of the page. The default name is Page# (i.e. Page1, Page2 etc.)

(PageNo) - Displays the sequence of the page relative to the other pages. For example 1, 2, or 3. To add or remove a page, change the Page property in the Form.

(SeqNo) - A read-only sequence number RFgen assigns to a control. The SeqNo is NOT the same same as PrompNo (prompt number) or TabNo. The VBA Extension "App.PrompNo" returns the SeqNo.

AllowFocus - "True" allows the control to have focus, False does not. If set to True, this adds the DisplayMode property to the Control Properties list. If AllowFocus = True, the **AllowBackup**, **FocusOnClick**, **OnEnter** and **TabNo** properties are added. If AllowFocus = False, then these options are not displayed.

- **AllowBackup** - "True" enables navigation to the prior page, depending on the option selected. For example, the up arrow on a device keyboard, the SHIFT+TAB keys, or a prompt can be set to move the user to the previous page. "False" disables the function.

- **FocusOnClick** - "True" allows a page to have focus at runtime. You can set this to "False" if you want to use the page as a container or "staging" of information or content to be accessed by other pages but you don't want this page to be viewable at runtime.

- **OnEnter**: *Advance* - The other options you can set when Enter is tapped are: *ExitForm*, *HoldFocus*, *ResetForm*, and *Submit*.

- **TabNo** - Is the order number of the object that can accept focus. This is not the same as the SeqNo (prompt number).

The **Anchor** property sets a child object's position and dimensions relative to where its anchored to the parent. This property is added to the Page Control when the **Dock** property = None. For more details on Anchor, refer to the [Graphical Control Property Description](#) topic in the User Guide.

BackGround - Sets the background color and how it is applied (as a solid or gradient) to the page. If set to (Default) the value is inherited from Themes. If the value is NOT "(Default)" then the style and color values were set in the Designer. The percentage of **Transparency** affects how transparent the foreground is while allowing the background to come through.

BorderStyle- For details, see Graphical Control Properties: [BorderStyle](#).

DisplayMode - Sets the DisplayMode option for the page. Unless you are designing the page to contain a specific application state, leave this value as "Normal". The DisplayMode application states are: Attention, Normal, Bold, Disabled, Error, Focus, Information, Link, Normal, Success, Warning. If you want to make sure the properties for a specified DisplayMode are inherited from Themes, make sure the Themes > Application > DisplayMode > Background > Override = True.

Anchor the page and use **Location** coordinates, set Dock = *None*. If you want the page size to be the same as the form's size, use "Fill" which will override the page Size and Location values. If you want the page docked with a limited width, set the doc to left or right. The height will be set automatically.

Events - Allows you to set the event (actions) that can be taken against the page. For information about Event values see the [Event Property](#) topic in the Developer Reference Guide.

Heading - Used to enter the text / page heading. Text entered in "Heading" will be displayed only if the TextId value is blank (no strings selected). The Heading is populated from the **TextId** if a text string identifier is selected. These are sourced from the Solution Explorer > TextResources > [Name of Object] table. This is also where you can set the heading to use a localized string.

Location- Sets the location of the page (number of pixels to the left of an anchor and number of pixels from the top.)

Size - Sets the page height and width in pixels. If the **Dock** property = Fill, you cannot edit the page height and width.

SystemIcons - Adds or removes icons to the page, and associates specific, system operations with the system icon. The built-in Call Events / operations are Cancel Operation, To add icons / system operation click on the ellipsis (...) a which opens the Manage Icon Collections screen. For additional information, refer to the help topic on the [Manage Icons Collection](#). You can also view the icons available through the [SystemIcons List](#) topic. For a list of all the system operations, see the [Systems Operations](#) topic.

Visible - True makes the page visible; False hides it.

Using Mobile Themes for a Graphical Control

You can setup a theme that will save all UI control property settings under a unique Mobile Theme in the **Solution Explorer > Mobile Themes > Element >** [name of the prompt/graphical control], and make these your default values in the same control under the **Solution Explorer > Application** designer. For example, if you created a theme called "Silver and Blue" and set it as the default under **Configuration > Application Preferences**, when you create a new application, and the graphical controls such as Layout, TextBoxes, Labels, and Buttons, will use the property values from Silver and Blue.

If you want to overwrite the settings from the theme, you can simply change the value on the control's property in the Application Designer. (i.e. Change the Background color from default to Brown.)

Manage Rows Collection

The Manage Rows Collection allows the user to stylize the elements comprising rows used in the Layout Control, ComboBox, DataGrid, Layout, ListBox, and TreeView control. The properties describe here are content specific and are not always used by every control.

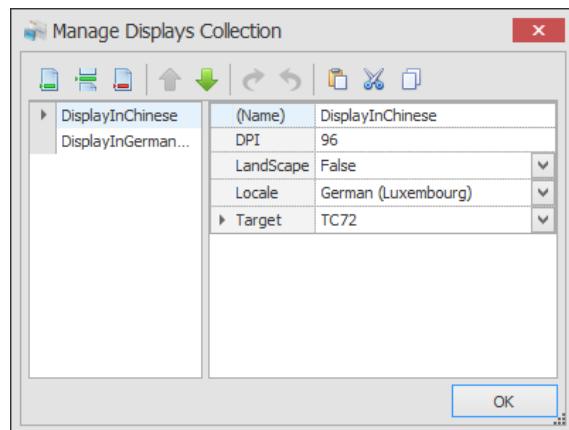
Adding, Removing, or Deleting Rows

Select the appropriate icon and update the (Name) and other property values.

The bottom-green icon adds a column; The red icon, removes a selected row, and the inbetween pages icon insert a row.

The properties here are the same as those described in the Manage Columns Collection. For more detail, see the [Manage Columns Collection](#) topic.

Manage Displays Collection



The **Form > Display > Manage Displays Collection** window allows you to have multiple displays linked to a form so you can host a variety of displays for an application. For example, you can create a Cycle Count application with three displays where each one is localized in a different language and designed for a specific target device. The application will display in the chosen device at runtime (see Testing).

If you select a RFgen factory-provided Device image in the Target field, and set its orientation, the display height and width will be shown in the Form > Size property. For more information about the target Device feature, see [Devices](#). Use the icons on top to add, insert, move or remove a display. On the right column, select the value to be applied. "(Name)" add the name of the display; the DPI is a preset value for scaling the display relative to device selected in Target. The Target drop down contains the list of target devices and their screen details. At runtime, the application will display in the skin of the selected target device. The Locale drop down sets the local of the language to be used with this display. At runtime, the text will appear localized if the Solution Explorer > Language Translation and Configuration > Environment Properties are setup properly.

 To add a new display, click on the Add icon.

 To insert a display between others, click on the Insert icon.

(Name) - The unique name for your display. No spaces or special characters are allowed (for example, don't use @, #, \$, %).

Description - This information automatically displays for RFgen provided Target Devices.

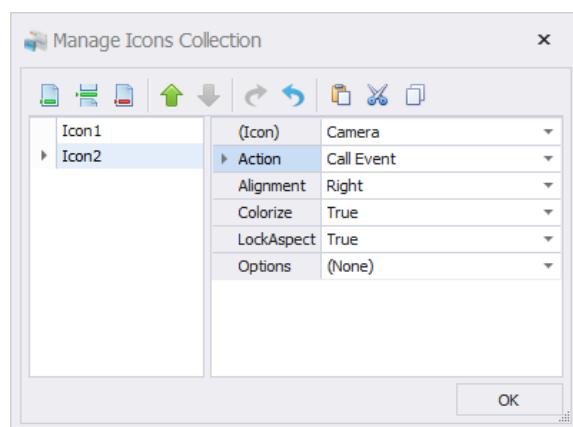
DPI - the resolution value. Windows standardizes to 96 dots/inch. This should match the scaled resolution set for the device display area.

Landscape - True will rotate the display. For example, if device screen is normally vertical, this will rotate to Landscape. False will not rotate the display.

Locale - Select the locale for the display. Additional setups are needed. See [To Localize Text](#).

Target - Enables you to select and display the skin of the target device image as your display. See [To add a target device](#) to upload your own customized device.

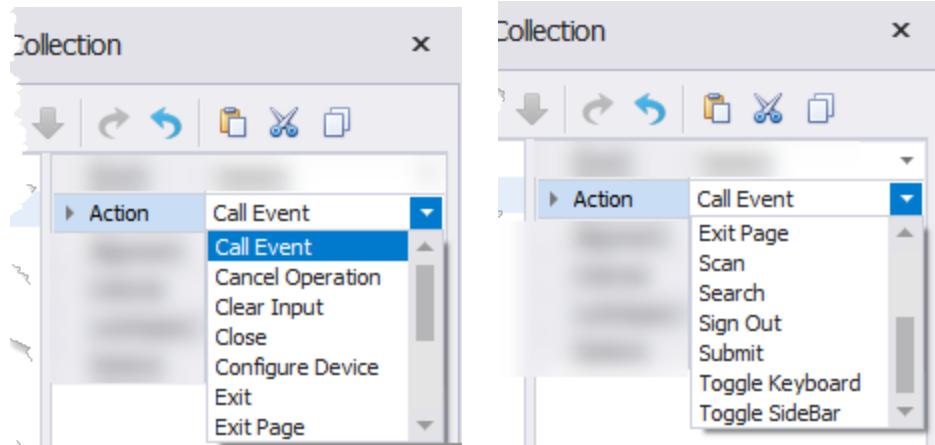
Manage Icons Collection



(Icon) - Allows you to select which image from (Custom) or the list of icons.

"(Custom)" images are selected from the images/icons stored in the Solution Explorer > Images group. The RFgen provided icons are: Arrow Filled, Arrow Outline, Camera, Check, Chevron, Cross, Edit Outline, Edit Filled, Enter, Exit, Gear, Hide, Home, Issue, Lock, Menu Dots, Menu Lines, Message, Next, Next Thick, Pen Outline, Pen Filled, Power Off, Print, Refresh, Reload, Return, Search, Settings, Show, Tools, Unlock, and Warning.

Name - The name of the image resource selected from the drop down list.



Action - is a menu of factory-provided actions you can associate with the icon.

Alignment - positions the icon on the right or left side of the heading bar.

Colorize - True enables you to assign a color (solid, not blended or shaded) to the icon. False uses the color the icon as is.

LockAspect - True locks in the height and width proportion ratios; False will allow ad hoc size changes.

Options - sets the orientation of the icon. (None) is the default. Values are Flip Horizontal, Flip Vertical, Rotate Right or Rotate Left.

To Set Text Defaults in a Graphical Control

A default property allows a data value to be generated automatically. A default property is available for each application entry prompt and defaults are entered in the Properties field. For example, if you had two warehouses, one in Boston and the other in Sacramento, and wanted the user to default to the Boston Warehouse from a TextBox control prompt called "Warehouse", you can default the value by entering specific string values in Defaults property of the EditText property in the graphical TextBox control, or you can enter the values via script.

For a description of the possible default values and syntax, refer to the topic Text Default Values in the Developers Reference Guide which can be downloaded from the RFgen Online Help and Documentation Page.

To Validate Entries in a Graphical Control

For controls that accept user entries (via scanned data or manually entered data), you can use formatted script in the Edits property of a graphical control which accepts user edits in order to verify if the data that was entered is valid. For more details on preformatting the values for performing a validation using a graphical

property, see [Edit Property Options](#) in the Developers Reference Guide which is available from the RFgen Online Help and Documentation Page.

Scripting with VBA

For information on the Visual Basic Application events, functions, commands and other environmental scripts refer to the [RFgen Developers Guide](#).

Script View Ribbon Menu

When the script window is opened, the menu bar at the top of the window provides some standard and custom features/

To **show line numbers** in your scripting window, click on **Configuration > Application Preferences > Scripting Defaults**, and check the "Show Line Numbers" box. This setting can also be used to change the font type, font size and tab width. You can globally change your scripting colors from the Scripting Colors section also in Application Preferences.

Save - option saves the current script to the solution database.

Designer - returns you to the application designer view.

Syntax - performs a syntax check of the script.

Undo, Redo, Cut, Copy and Paste. The **Comment** and **Un-Comment** options allow quick removal or addition of code blocks.

The **Find** and **Replace** icons allow the user to find or replace text within the current application or all applications and macros at the same time.

 **References** icon allows non-globally loaded **Scripting Modules** to be associated to the application. This requires the setup of the specific file under the Scripting Modules Tree and the file's Load Module should be set to "When Referenced."

(See the **Scripting Modules – Load Modules** tree for more details.) Check those modules you wish to include with the current application. Modules designated **Win32.bas** and **RFgen.bas** are automatically included for each application.

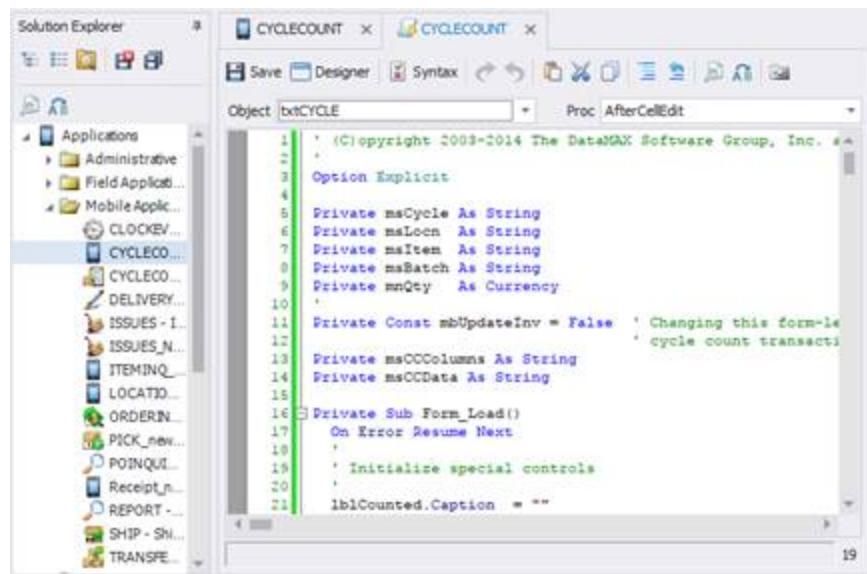
Shortcut Keys

The Scripting windows also provide some keyboard shortcuts that may make moving around in the environment easier. Ctrl + A will highlight all the text in the window. Ctrl + G will request a line number and then move the focus to that location. Ctrl + Spacebar will pop up a search window for language extensions, embedded procedure parameters and other similar uses.

Scripting Rules of Precedence

Click the **Solution Explorer > Applications >**  **Script** icon designer to display the code view and functions for a selected application.

The Scripting view of your application displays.



A screenshot of the Microsoft Visual Studio IDE. The left pane shows the Solution Explorer with various application projects listed. The right pane shows the code editor for a file named CYCLECOUNT. The code is written in Microsoft Visual Basic (VBA) and defines a class named CYCLECOUNT with properties like msCycle, msLocn, msItem, msBatch, and mnQty, and methods like Form_Load(). The code also includes comments and a copyright notice from DataMAX Software Group, Inc.

```

1  ' (C)opyright 2003-2014 The DataMAX Software Group, Inc.
2
3  Option Explicit
4
5  Private msCycle As String
6  Private msLocn As String
7  Private msItem As String
8  Private msBatch As String
9  Private mnQty As Currency
10 '
11 Private Const mbUpdateInv = False ' Changing this form-le
12                                         ' cycle count transacti
13 Private msCCColumns As String
14 Private msCCData As String
15 '
16 Private Sub Form_Load()
17     On Error Resume Next
18     '
19     ' Initialize special controls
20     '
21     lblCounted.Caption = ""

```

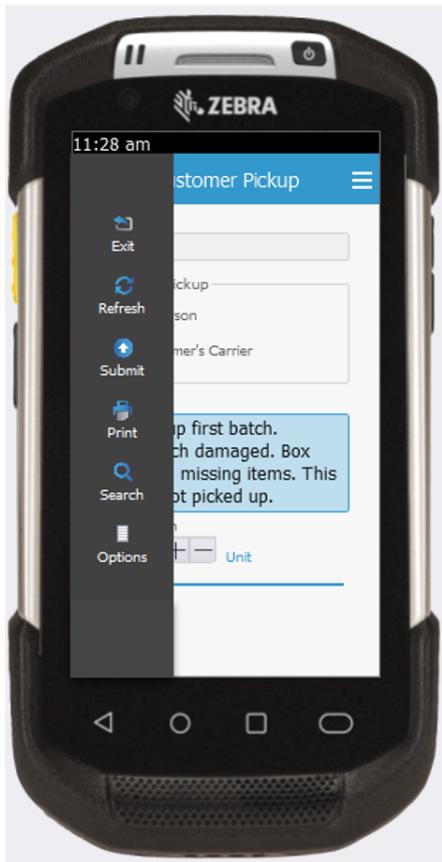
The open source applications were written using Microsoft Visual Basic (VBA).

When using VBA, note that each prompt on the application, and the application itself, may have associated scripting for the numerous associated events. All possible VBA events are described in the section "VBA Events" near the end of this book.

Application events take precedence over prompt events; e.g., events linked to the application will occur prior to the event firing for a prompt.

Note: VBA code is very sensitive to variable typing. Most errors result from using a String variable when an Integer or Long data type is required.

SideBarMenus



The SideBar Menu allows the user to choose an action which is hosted by the device while using the application. For example, while taking count of inventory, a user may need to scan the barcode while using an iOS or Android device, or take a picture of the image without having to exit or log out of the the application (i.e. Cycle Count) they are in. This kind of menu can be created using a variety of setups in the Mobile Development Studio.

Configuring SideBar FKeys

You can also configure function keys to a system function (device hosted operation) from the **Mobile Development Studio > Configuration > Menu and Key Settings** screen. For more details on this feature, see [Configuring Menus and Keys](#).

Setting System Operations

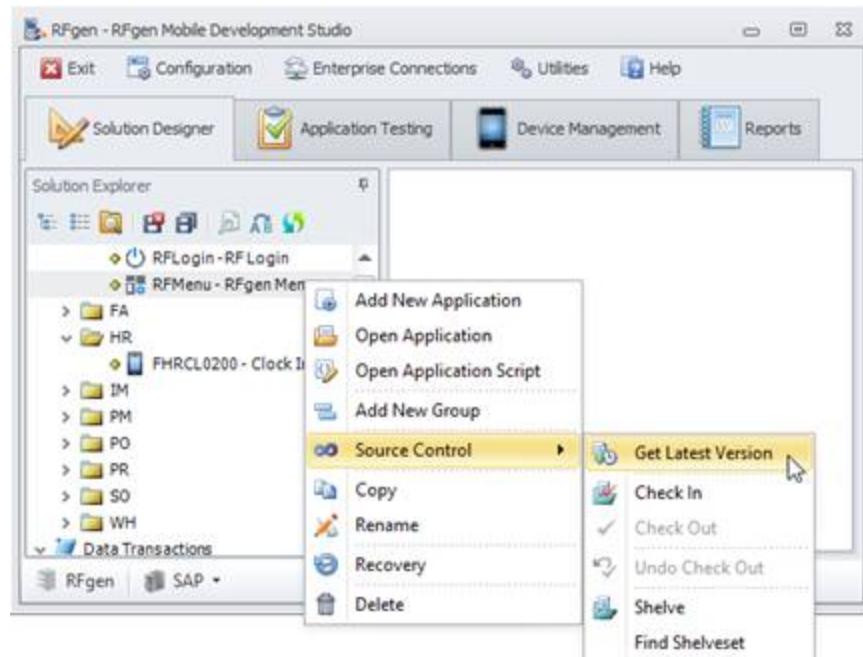
System Menu Configuration				
	Action	Display Text	Image	Object Name
▶	Configure Device	Settings	<input type="text"/>	Settings
	Scan	Scan Barcode	<input type="text"/>	Scan

To link device system operations with an icon, set the values in the System Menu Configuration table located in the **Mobile Development Studio > Configuration > Menu and Key Settings** screen. For details, see [Configuring Menus and Keys](#).

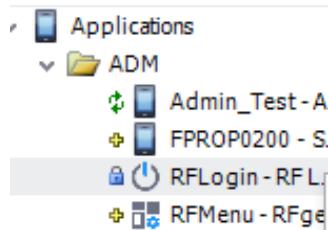
Mobile Themes for SideBar

Mobile Theme SideBar is used to set the look and feel of the SideBar as well as how its accessed and positioned within the client screen. For more details see [Themes SideBar](#).

Source Control Options in the Solution Designer



When a source control system such as Microsoft Team Foundation Server (TFS) is configured to manage objects in RFgen, a secondary icon will display next to each Solution Explorer object icon.



For example, a yellow + sign, green arrows, red checkmark or a blue lock will display to the left of the RFgen Login icon.

The Source Control Options manages Applications, Data Transactions, Device Tables, Images, Mobile Profiles, Mobile Themes, Screen Mapping Scripts, Scripting Modules or Soft Keyboard Codes. It does not manage Menus and Roles or Users.

For additional information, see [Configuring Source Control](#) (between RFgen and a source control product), [Menus/Object States](#), [Find Shelve Sets](#), and, making [Shelve Changes](#).

Source Control Menu and Object States

The menu selection options are:

- ⊕ The yellow plus sign means the object (i.e. application) has not been checked into the source control system. This also indicates RFgen is connected to a source control system.



The Check In menu option places new objects (objects that have not been checked in before) under source control. If the object was previously checked out, the source control system registers the check in as another version of this object. Notations of what has changed is usually required.



The Get Latest Version menu option retrieves the latest version (last modified) object that has been checked in. When you get the latest version, you are getting a copy of it, but leaving it "open" for others to also get a copy. To prevent others from working on the object, you must check the object out.

The Check Out menu option retrieves a copy of the object and locks the file so others cannot modify it. This is known as a "check out."



The red check mark means YOU checked out this object and are the owner. When the object is checked out, no one else can modify the object.



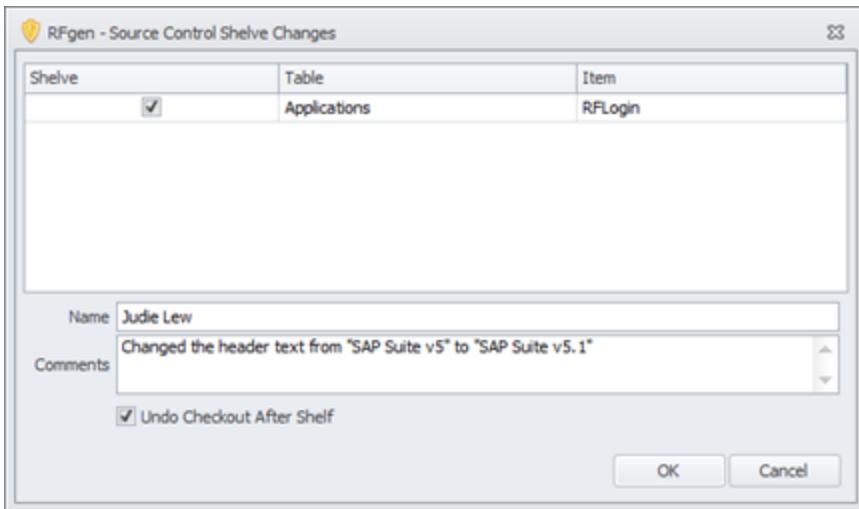
The green cycle icon means the object's version status is unclear. This can occur if a user checked out the object, made code changes on an RFgen system that is not connected/configured to the Source Control system, then reloaded the object to the RFgen connected to the Source Control system. The unclear version status can also occur if a second user loads an object with the same name as another object already under Source Control system, but the object was not from the Source Control system (i.e. The object came from a system that was not under Source Control).



The Undo Check Out menu option reverses the process. The object reverts to the version it was at before it was checked out and unlocks the object so its available to other team members.

The Find Shelveset menu option helps you find shelved objects, view any comments entered, and check out the object.

Shelve Changes



"Shelving" of code is helpful for organizations that use Peer Reviews, Quality Assurance or reject/accept processes prior to checking their source code into their repository or source control system.

Name – This is a free form text field. It does not accept special characters.

Comments – This is a free form text field.

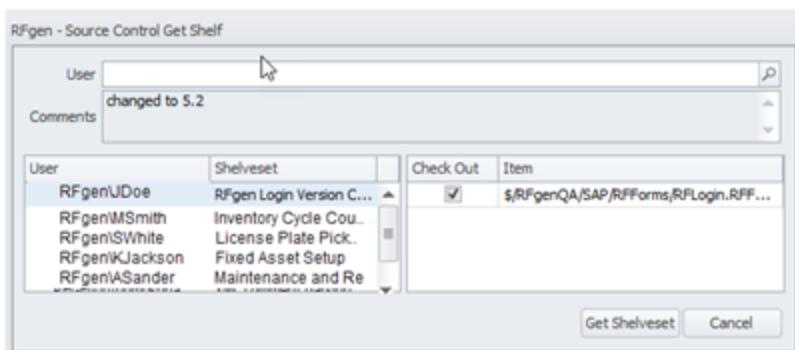
Undo Checkout After Shelf – If you want to continue working on the object, you will need to "Check Out" the object from the right-click menu. Since RFgen does not access to checked-out objects, the Undo Checkout After Shelf helps enable others (like a QA reviewer) to access the object after you had checked it out.

By default, the "Undo Checkout After Shelf" box is checked. If unchecked, others will not have permission to access the shelved object.

For more details on Shelving, see [Find Shelve Set](#).

For information on configuring RFgen to connect to a Source Control system/server, see [Configuring Source Control Integration](#).

Find Shelve Set

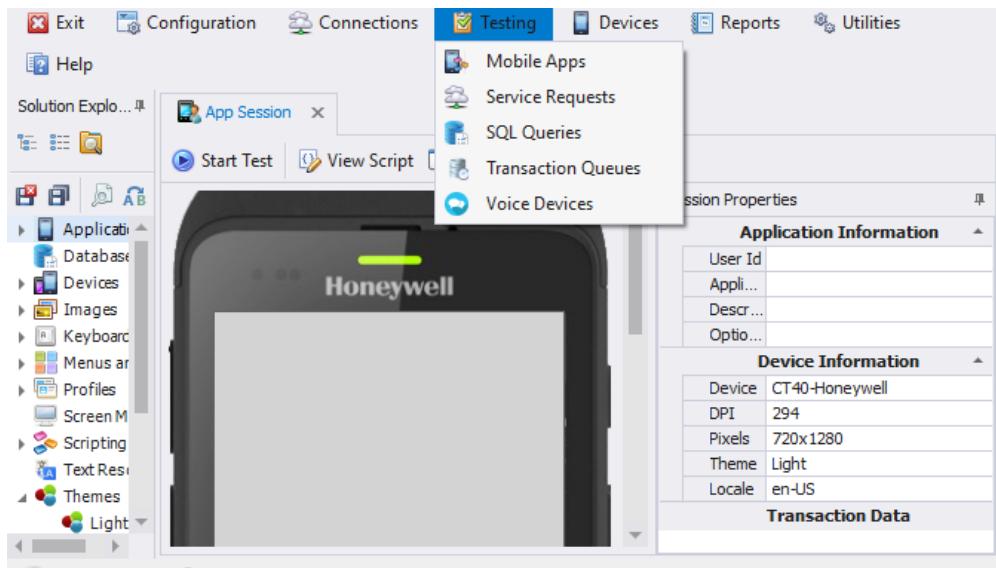


When you select Find Shelve Set from the menu Source Control, the RFgen – Source Control Get Shelf screen displays.

This allows you to check out the shelved version of the object and get a copy of the object and the changes. After you have reviewed it, you can then perform a Check In which will save the changes permanently.

For additional information on Source Control, see [Configuring Source Control](#) (between RFgen and a source control product), [Solution Source Control Options](#), [Menus/Object States](#), and, making [Shelve Changes](#).

Testing Mobile AppsOverview



The **Testing** Menu in the Mobile Development Studio provides:

- runtime sessions for **Mobile Applications** so you can see how the application screen looks in a device at runtime, view and debug code or objects (i.e. applications, screen maps, services).
- a means of testing **Service Requests** for integrations with cloud based solutions.
- runtime sessions for testing **Transaction Queues** or **Transaction Events** so you can check Cycle Time and Iterations.
- data steam portions of other third-party data collection solution that are voice driven and use unique hardware and software for the testing of **Voice Devices**.

Connection Limitation

When you test device connections with the Mobile Development Studio, the Mobile Development Studio limits the number of simultaneous connections between itself and remote devices. For example, you can only have a total of 3 connected sessions.

Stop Test - This allows you to start and stop your test sessions.

View Script - For information on testing your script see [To Debug Code](#).

Options - For information on this feature, see [Testing Options menu](#).

Mobile Apps: Session Property Descriptions

Application Information

User ID - Displays the user who is logged in during the test session. For example if you had a user named Sam setup in Users and logged in as SAM, "Sam" would display here.

Application - Tells you which application you are testing.

Description - The description that is associated with the Application you are testing.

Options -

Device Information

These settings can be changed using the Options menu.

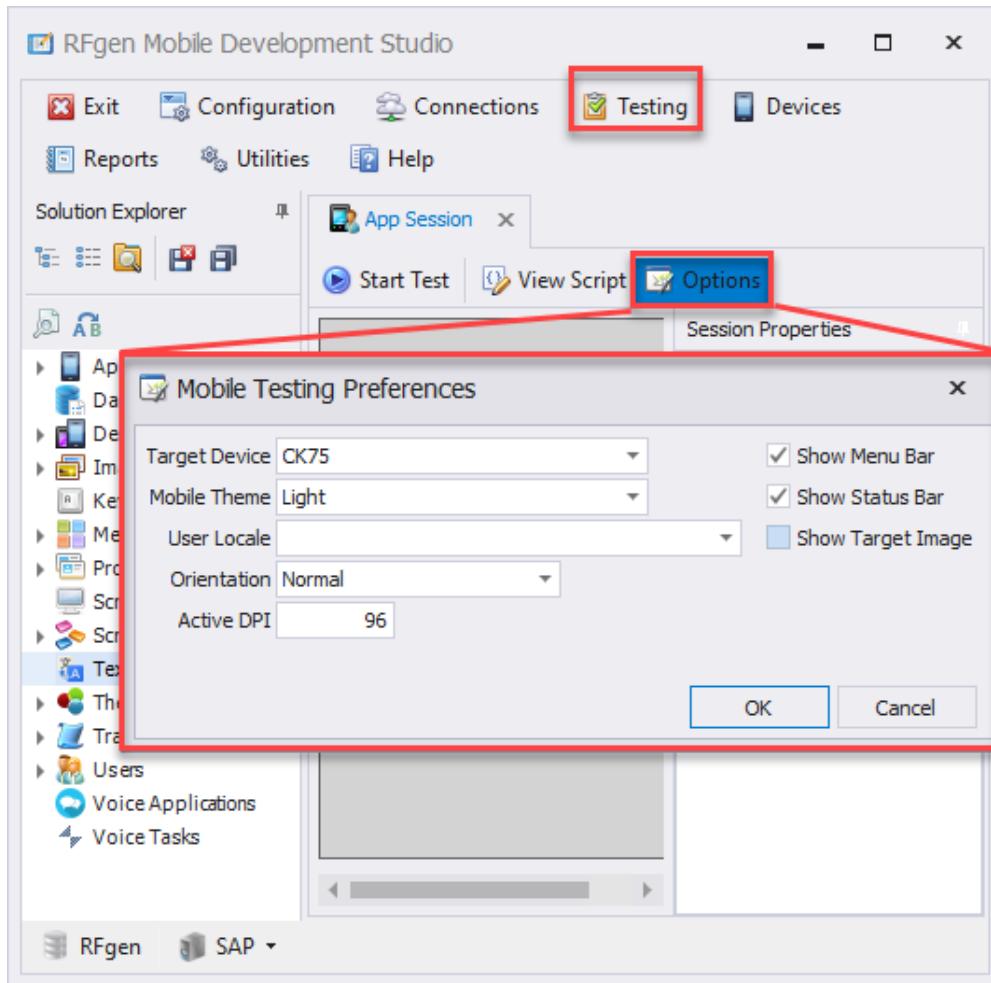
Device - Which target device image is in use.

DPI - Tells you which values are used to scale the device image.

Theme - Tells you which Mobile Theme is used for the application's look/feel.

Locale - Tells you which language is used if you have your application setup for different locales.

To Set Your Testing Preferences



The Mobile Testing Preferences window displays when you click on the **Testing > Mobile App > Options** icon.

When testing, the values set in your Configuration > Environment Settings and Configuration > Application Preferences are used. However, if you want to change the device skin, theme, locale (if application was designed for localization), and orientation, you can change these features through the Options screen.

Use the **Options** screen to change:

- Target Device - sets the device type (device image).
- Mobile Theme - sets which theme is used; otherwise the theme from Configuration > Application Preferences > Theme is used.
- User Local - sets the locale if the application strings were translated for multiple languages. The default is US English.
- Orientation of the device screen - Normal versus Rotated.
- Active DPI - controls the scaling of the image. 96 dpi is a standard for smaller devices. The value depends on the resolution of the target device.

Show Menu Bar - If checked, this displays the menu bar of the application so you can test your function calls from the menu bar.

Show Status Bar - If checked, this displays the status bar of the application (i.e. device battery status, device wireless connection status). Will show how much space is used but statuses will not be testable.

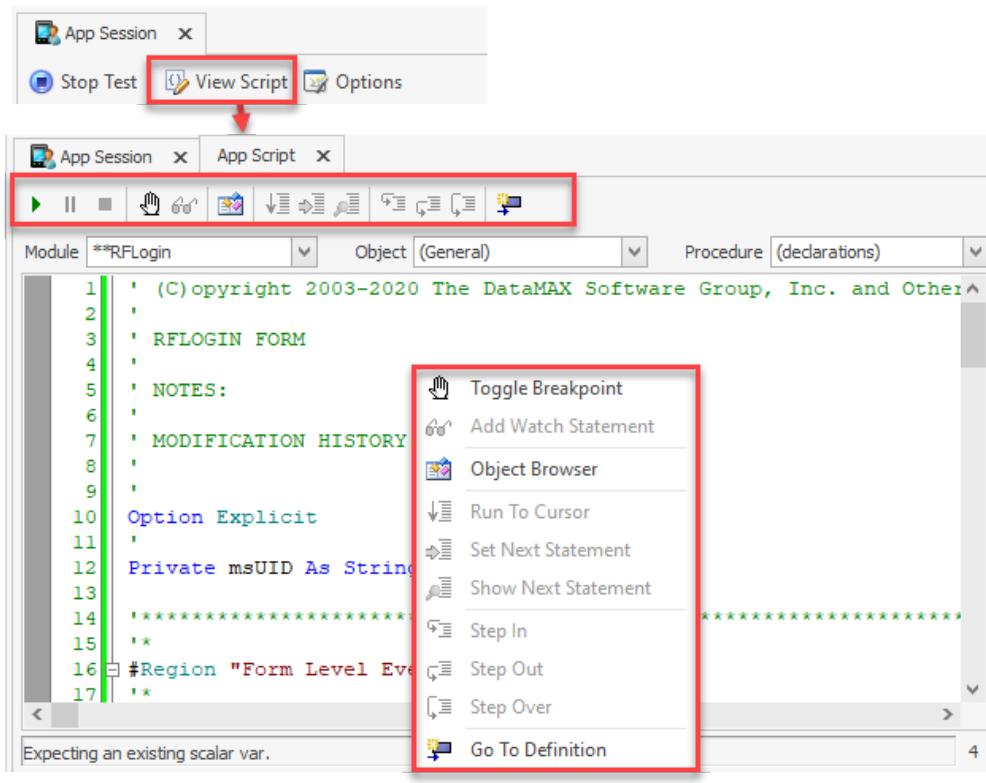
Show Target Image - If checked, application displays in the target device. If unchecked, the application displays in a screen (no device).

To Debug Code

To view your script and start debugging from the Solution Explorer, click on **Testing > Mobile Apps > Start Test button**. The App Session tab will display.

To run your script from the graphical view of your app, click on **Start Test**.

To run your script in code view, click on the **View Script** icon which will display the App Script tab. Click on the green arrow to run your script.

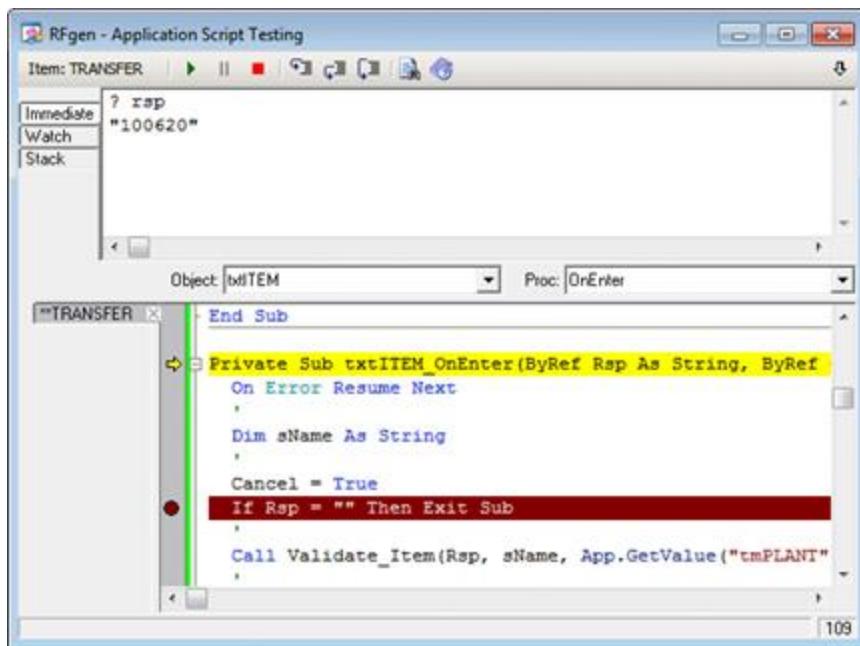


Debugging options

Right-click on the background of your script to bring up the debugging features you can use to step through your code and look for specific lines of code.

1. Click on the blue pause button and then trigger an event. Code execution will be paused on the first line of that event.

- To debug code in the Form Load event, add the line 'Stop' so that execution is halted on that line.



- After you have entered your user ID and password, the menu assigned for the user will display.
- Use your keyboard/keypad arrow keys to select an application and press <Enter>.
- In graphical mode you may use the mouse to select and execute menu items. In test mode, you may enter data exactly as if you were using a remote device.
- The Application Values grid on the Watch tab shows what your data item (record) looks like as data is added.
- Click on 'Stop Test' to stop the testing process.

To Test Your App and Bypass the Login App

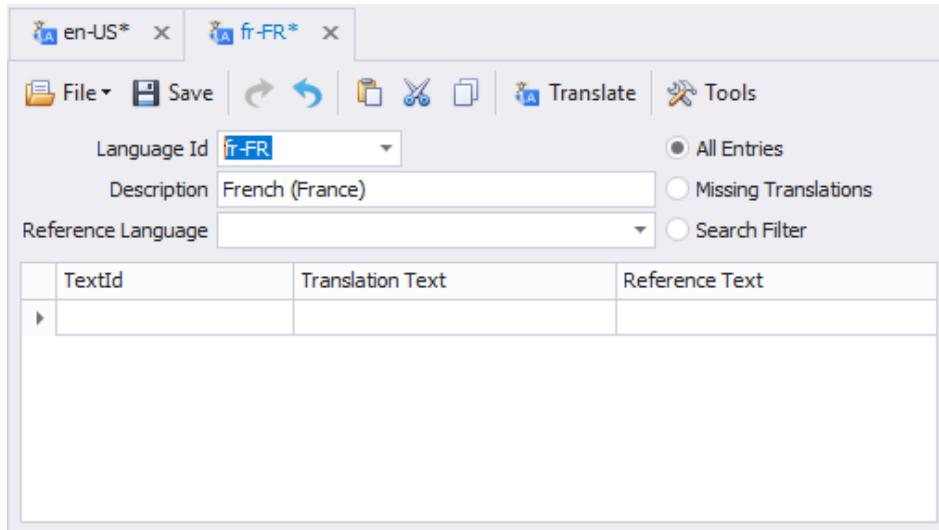
If your RFgen server is connected to a RFgen-provided ERP database of applications, you'll find that when you start testing applications, the test will first bring up the RFgen login app, then go onto the menu. If you have a customized application you want to test without having to first go through the RFgen login screen, you can launch your app directly by using the App.CallForm function in the script view.

- Start with a connection to an open source 5.2 db like SAP.
- In Dev Studio, click on **Testing > Mobile Apps**.
- Do not click on Start Test, instead click the **View Script** icon.
- The script view screen displays.
- Click on the green arrow (>).
- The Immediate, Stack, and Watch tabs display.

7. On the Immediate tab, enter App.CallForm("The Name of Your App")
8. Press **Enter**.

Two taps for your application tabs will display. If you click on the tab that displays the graphical view of your app, you can continue testing as the user. Or, you can use the script view to walk through code.

Text Resources (Language Translations)



The **Solution Explorer >Text Resources** node translates and stores the text strings so you can easily create and distribute mobile applications in the language that is best for the end user. This is especially helpful for companies that have plants in different countries but want to use the same application/transactions in their warehouses/plants. All translations are performed through the Google Translator service.

How this works

You create a master list of text strings in the language that's native to you. For example, if you are an English speaker, then you'll create your master list of strings and text identifiers in English. Once your master list is created, you then create a new Text Resource table in the alternate language but reuse the text ids text strings from your master list to populate the new table and then translate the text strings. Modifications made to the master text resource list will allow you to easily update the other Text Resource tables in other languages.

For details on how to create Text Resources and Translate them, see [To Translate Strings](#).

Translations performed by Google are not context-aware.

As mentioned earlier, RFgen uses the Google translator to perform translations. This means the term Google translated may not be the correct term for how that term is used in context in the country's business culture. For example, in the United States, the term "Plant" is commonly used for warehouse, but in other countries, the literal translation is a green plant that grows in your yard and does not have the contextual meaning of a warehouse or manufacturing facility. **It is recommended that a native speaker review the translations to avoid any translations that are out of context.**

Tools Menu

You can also use the **Tools** menu to:

- Import System Entries (RFgen-provided system operation text ids)
- Export your text strings to Excel
- Import your text string to Excel

Other Actions

Table entries can be grouped, filtered, and searched.

Language columns can also be moved, but only single entries may be deleted at a time. (You cannot remove the entire Language column.)

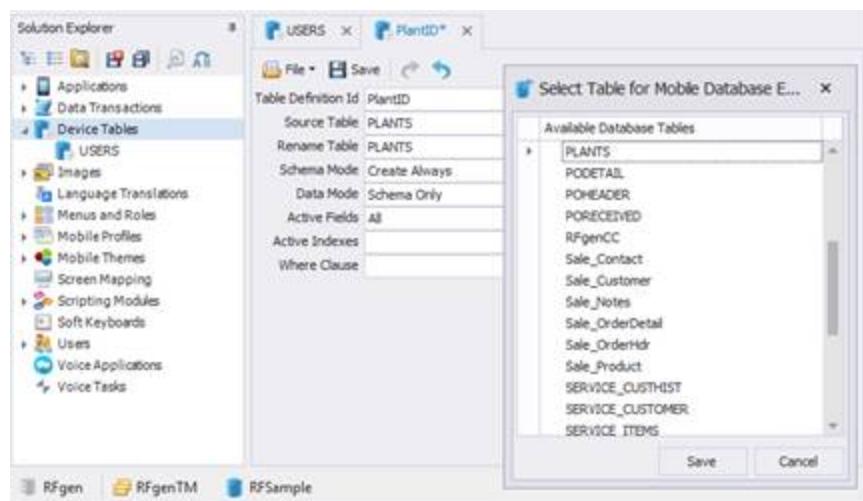
The **File** icon is used to create a new item (text resource), reload a resource, save an existing text resource with a different file name, delete, a text resource, and close a text resource.

The **Reload** and **SaveAs** features will allow you to create a new file from an existing file.

The **Save** button will save the text resource object name using the Language ID that selected from the Language ID menu list.

For example, if you selected "de-DE," "en-US - English," or "fr-FR" the Text Resource objects are named this as well, and the description is also automatically applied for the resource id (for example "fr-FR - French (France)").

Database Exports



The **Solution Explorer > Database Exports** feature allows you to preconfigure tables for processing data on a device when the device is in batch or offline mode.

Once your table(s) are setup, you can add it to a mobile device's profile via **Solution Explorer > Profiles > Select Apps** and include it as one of the selected applications in the profile.

To add a new table or group, use the Right-click menu and fill in the Table form to configure the table and how its modified on the device.

The **Table Definition ID** is the name or identification name of the table.

The **Source Table** is the name of the table in the database referenced through the connector.

The **Rename Table** property allows the user to change the name, as it will exist on the mobile device. Note that SQLite naming follows much stricter rules than other databases, so renaming a table might be necessary for Android or iOS platforms.

Schema Mode sets how the table will be created. There are three options: *Data Only*, *Create Always* (default) and *Create on Change*.

- *Data Only* means that RFgen will not create the table at all. It must exist in the database already. The Data Mode property will determine how the data is filled in.
- *Create Always* means if the table already exists, it will be deleted and recreated on the mobile database.
- *Create on Change* means that the existing table schema on the mobile database will be compared to the equivalent table schema on the server. If there is a difference, the existing table schema on the mobile database will be replaced with the updated table schema from the server.

The **Data Mode** property has 3 options: Clear and Copy (default), Copy Only and Schema Only.

- *Clear and Copy* will delete the contents of the table and re-populate it with the new data from the server.
- *Copy Only* will place the data from the server on top of the existing data. Where it is the same, there is no change. Where it is different and depending on the keys, either it will overwrite data or insert new records.
- *Schema Only* will always send an empty table to the mobile database.

Active Fields are the fields that should be copied from the server database and used in the mobile database. This allows the user to have a smaller table structure, if the server's table contains fields that are not required for the mobile data collection effort. Leaving it blank assumes all the fields are necessary.

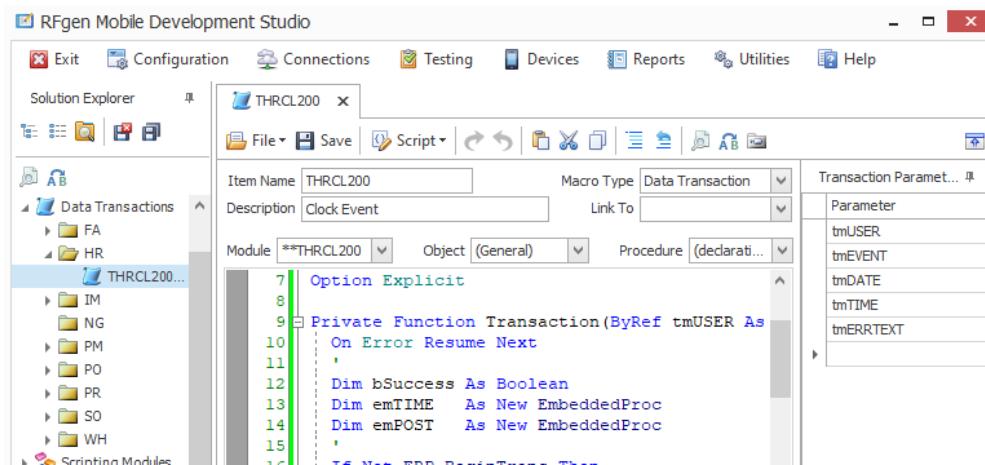
Active Indexes specifies the indexes on the mobile database for more efficient data retrieval.

Trim Mode trims off the characters from a string of characters starting from the left or right side of the string.

The **Where Clause** specifies a subset of the data from the server's table in case the server contains more data than is necessary on the mobile database.

Data Transactions

The transaction design window has three purposes: to create and edit Timed Event macros, Host Transaction and Data Transaction macros.



A Timed Event macro is a macro that runs on a timer configured under the Configuration / Transaction Management / Transaction Management Events dialog. If you want some script to run without a user being present, create a Timed Event macro and enable the Transaction Management capabilities. There are no passing parameters for Timed Event macros.

A Data Transaction macro is a script that can accept parameters passed in and out and can execute in a queued or non-queued manner. You would create a Data Transaction macro: if multiple applications could take advantage of the same process, you need a history of the data being processed kept by the Transaction Manager, applications are run in a Mobile environment and later uploaded to the server for processing or queuing is implemented for all applications.

For more details, see [To Create a Data Transaction Macro](#).

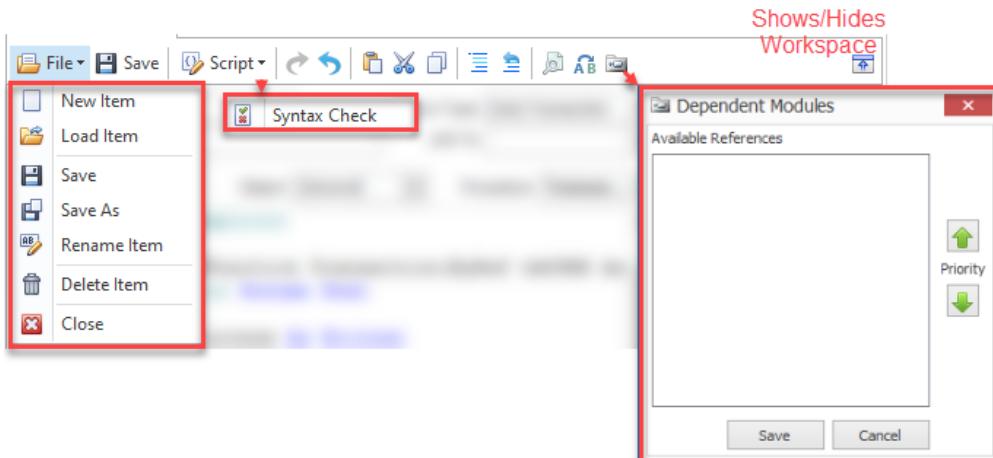
To create a Data Transaction macro

1. Right-click on **Solution Explorer > Data Transactions** and select **Add New Transaction** from the menu. If desired, you can also create a new folder to group your transactions by selecting **Add New Group** from the menu and moving your new transaction into this folder.
2. Enter an **Item Name, Description**. Set the **Macro Type** to Data Transaction.
3. Add parameters for each value being passed in to the macro. The Location and Length columns do not apply to Data Transaction macros. A maximum of 31 can be used due to the integration with the VBA environment. To work around this, you may concatenate multiple values separated by a unique string like " | " and only use 1 parameter.
4. Select the Transaction function from the Procedure drop-down and a shell function will be created for you.

For Menu descriptions, see the [Data Transactions Macro Menu](#).

A Host Transaction macro can be created and linked to a Host Screen macro. This is described in the Screen Mapping section.

Data Transactions Menu



The ribbon menu for Data Transaction displays once a macro is selected.

The **File** menu provides the following functions:

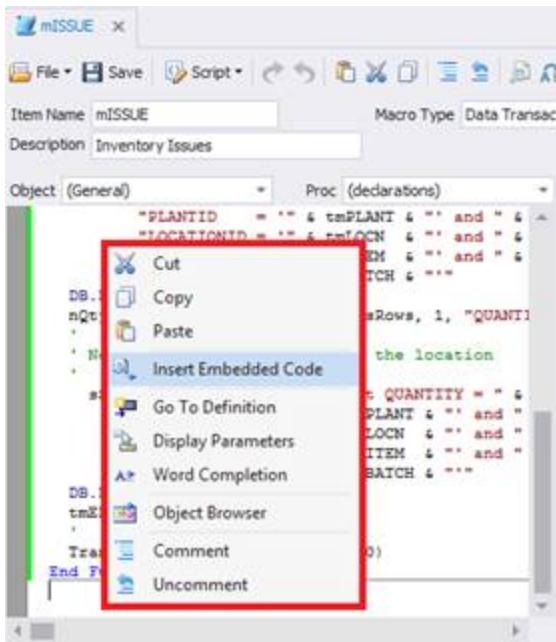
- Clears all fields and provides a blank window for creating a new transaction macro.
- Reloads the macro from the original source. (I.e. You accidentally made a change and can "reload" your content.)
- Saves the current script to the solution database. **File > Save As** and **File > Rename** rename an existing files.
- Allows you to delete an item.
- Closes the file.
- > - Performs a syntax check of the script.

Also provided are **Undo**, **Redo**, **Cut**, **Copy** and **Paste**.

- quick removal or addition of code blocks.

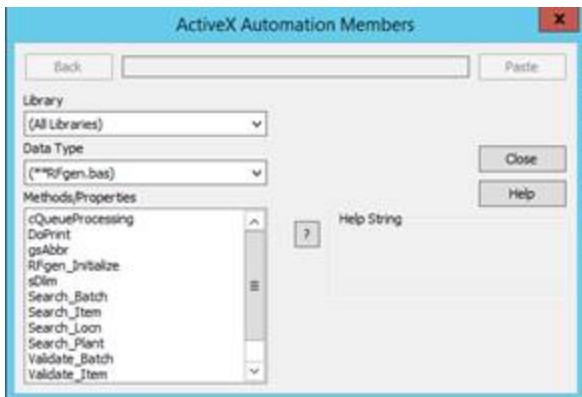
Find and **Replace** icons allows the user to find text or replace text within the current application.

To Allow References to Global Scripts



To allow references to global scripts that are in a VBA module but are not globally available, follow these steps:

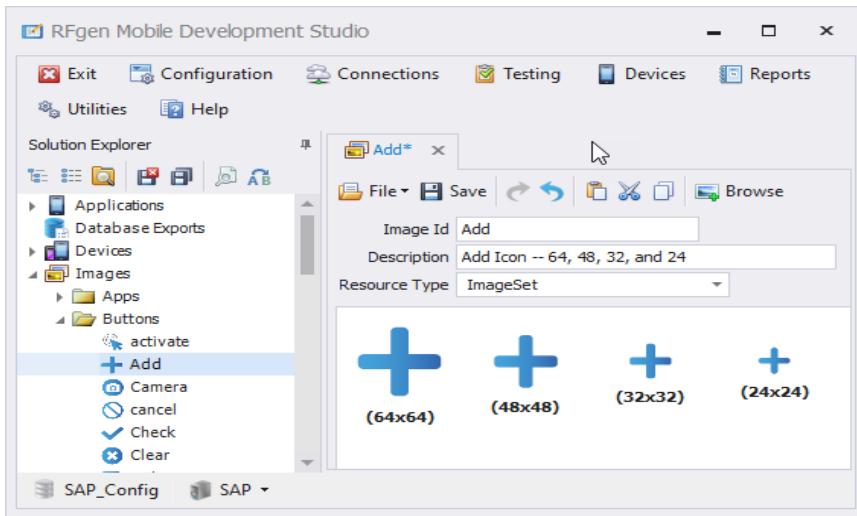
1. Right-click on any space in your scripting form to bring up the Script Edit menu shown above.
2. Select the **Object Browser**.
3. The **Active X Automation Members** window displays.



This lists existing VBA modules from **Solution Explorer >Scripting Modules**.

4. Check those modules you wish to include with the current macro. Modules designated Win32.bas and RFgen.bas are automatically included for each macro.

Images



Use **Images** folder to store the images (icons, background images, company logos). You can use the Browse button to upload any image and size. While RFgen does not set a limit on the image size, its best to upload an image that will be closely scaled to the final viewing size within an application. RFgen will store your original size, and scale the image (i.e. menu icon) automatically or in accordance to the values manually set in the application.

The images can be referenced by Image prompts at design-time or runtime or used as part of the configuration for mobile device backgrounds. This window allows the user to drag and drop an image for quick selection.

The **Image ID** is referenced from the GUI properties or from the code to extract this image from the application database.

The **Description** is an optional field used to describe the image.

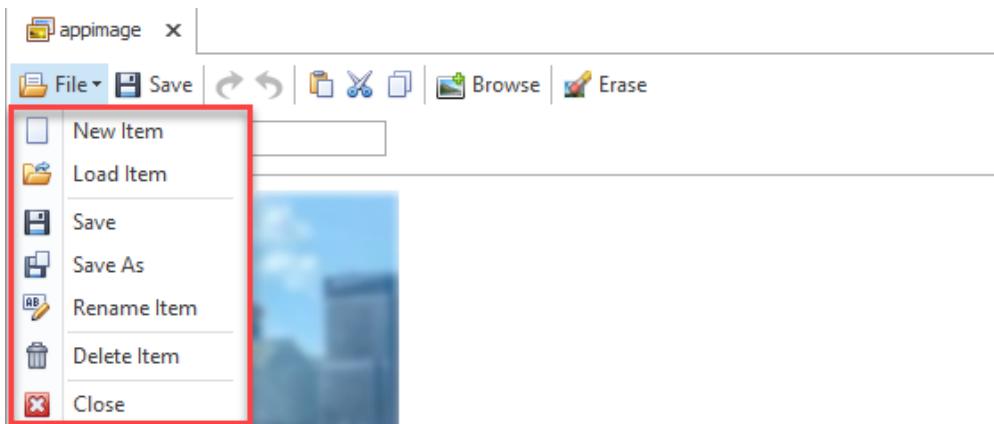
The **Resource Type** categorizes the image as an ImageSet or Animation.

The Resource Type - **ImageSets** option store a group of images and displays their size. The Animation option allows the storage of images setup for animation (movement).

The Resource Type - **Animation** option stores a series of images setup for animation (movement). The Frame Rate is how long you want the image to run in milliseconds. (100 milliseconds is the minimum.) The Frames field is the number of frames in your image. When selecting the image, set the Icon property to (Custom) and enter the Name of the image that was uploaded to your Images folder.

Use the **Browse** button to upload images.

Image Menu



The **File** menu enables you to perform standard file operations.

The **Browse** icon is used to import images from your computer or other locations to the Images tree. Supported formats are jpg, gif, bmp, png and jpeg.

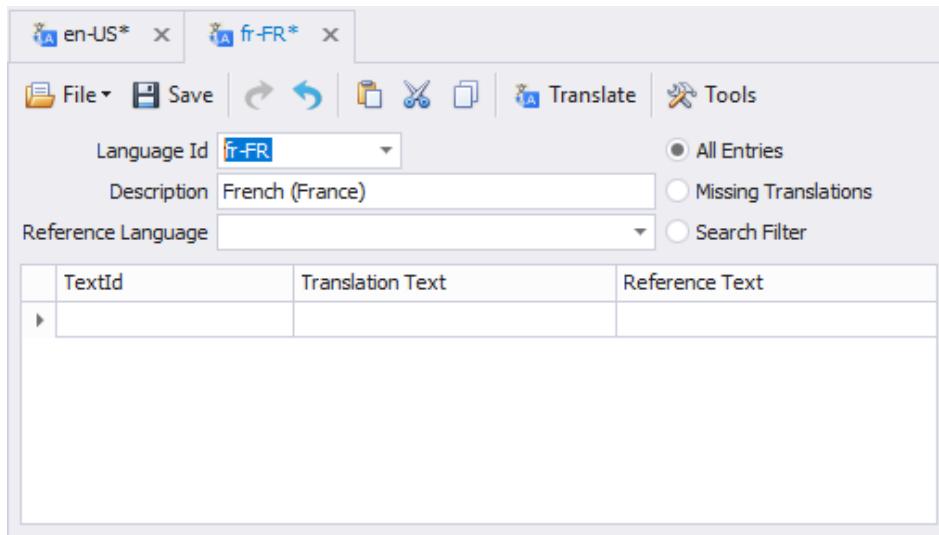
Themes: Signature



This sets the style for the Signature control in the Solution Explorer and is typically used in applications where authorization sign off is required.

For information on property definitions, see [Graphical UI Control Property Definitions](#).

Text Resources (Language Translations)



The **Solution Explorer >Text Resources** node translates and stores the text strings so you can easily create and distribute mobile applications in the language that is best for the end user. This is especially helpful for companies that have plants in different countries but want to use the same application/transactions in their warehouses/plants. All translations are performed through the Google Translator service.

How this works

You create a master list of text strings in the language that's native to you. For example, if you are an English speaker, then you'll create your master list of strings and text identifiers in English. Once your master list is created, you then create a next Text Resource table in the alternate language but reuse the text ids text strings from your master list to populate the new table and then translate the text strings. Modifications made to the master text resource list will allow you to easily update the other Text Resource tables in other languages.

For details on how to create Text Resources and Translate them, see [To Translate Strings](#).

Translations performed by Google are not context-aware.

As mentioned earlier, RFgen uses the Google translator to perform translations. This means the term Google translated may not be the correct term for how that term is used in context in the country's business culture. For example, in the United States, the term "Plant" is commonly used for warehouse, but in other countries, the literal translation is a green plant that grows in your yard and does not have the contextual meaning of a warehouse or manufacturing facility. **It is recommended that a native speaker review the translations to avoid any translations that are out of context.**

Tools Menu

You can also use the **Tools** menu to:

- Import System Entries (RFgen-provided system operation text ids)
- Export your text strings to Excel
- Import your text string to Excel

Other Actions

Table entries can be grouped, filtered, and searched.

Language columns can also be moved, but only single entries may be deleted at a time. (You cannot remove the entire Language column.)

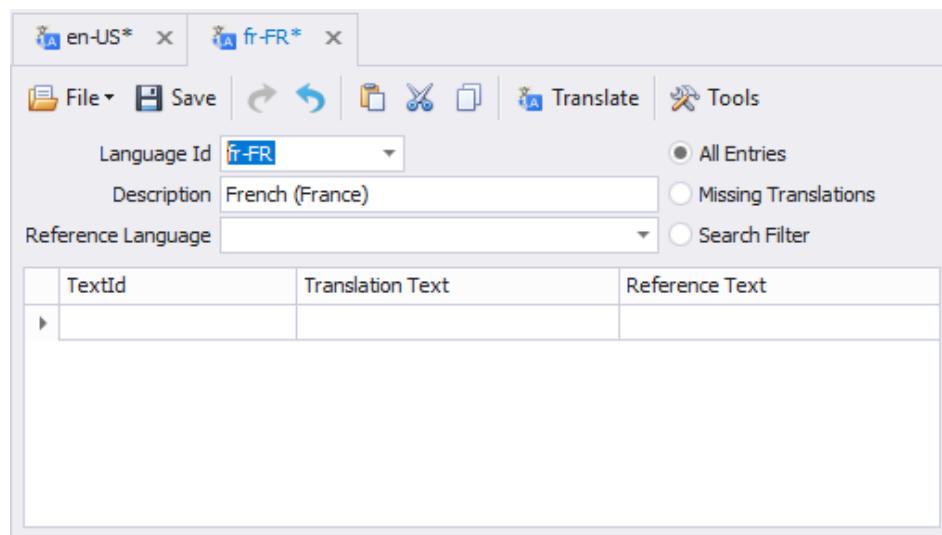
The **File** icon is used to create a new item (text resource), reload a resource, save an existing text resource with a different file name, delete, a text resource, and close a text resource.

The **Reload** and **SaveAs** features will allow you to create a new file from an existing file.

The **Save** button will save the text resource object name using the Language ID that selected from the Language ID menu list.

For example, if you selected "de-DE," "en-US - English," or "fr-FR" the Text Resource objects are named this as well, and the description is also automatically applied for the resource id (for example "fr-FR - French (France)").

Text Resources (Language Translations)



The **Solution Explorer >Text Resources** node translates and stores the text strings so you can easily create and distribute mobile applications in the language that is best for the end user. This is especially helpful for companies that have plants in different countries but want to use the same application/transactions in their warehouses/plants. All translations are performed through the Google Translator service.

How this works

You create a master list of text strings in the language that's native to you. For example, if you are an English speaker, then you'll create your master list of strings and text identifiers in English. Once your master list is created, you then create a next Text Resource table in the alternate language but reuse the text ids text strings from your master list to populate the new table and then translate the text strings. Modifications made to the master text resource list will allow you to easily update the other Text Resource tables in other languages.

For details on how to create Text Resources and Translate them, see [To Translate Strings](#).

Translations performed by Google are not context-aware.

As mentioned earlier, RFgen uses the Google translator to perform translations. This means the term Google translated may not be the correct term for how that term is used in context in the country's business culture. For example, in the United States, the term "Plant" is commonly used for warehouse, but in other countries, the literal translation is a green plant that grows in your yard and does not have the contextual meaning of a warehouse or manufacturing facility. **It is recommended that a native speaker review the translations to avoid any translations that are out of context.**

Tools Menu

You can also use the **Tools** menu to:

- Import System Entries (RFgen-provided system operation text ids)
- Export your text strings to Excel
- Import your text string to Excel

Other Actions

Table entries can be grouped, filtered, and searched.

Language columns can also be moved, but only single entries may be deleted at a time. (You cannot remove the entire Language column.)

The **File** icon is used to create a new item (text resource), reload a resource, save an existing text resource with a different file name, delete, a text resource, and close a text resource.

The **Reload** and **SaveAs** features will allow you to create a new file from an existing file.

The **Save** button will save the text resource object name using the Language ID that selected from the Language ID menu list.

For example, if you selected "de-DE," "en-US - English," or "fr-FR" the Text Resource objects are named this as well, and the description is also automatically applied for the resource id (for example "fr-FR - French (France)").

To Translate Strings into Multiple Locales

This topic explains how to translate strings from a base language into other languages (i.e. Spanish (Spain), Chinese(Simplified) etc).

RFgen uses Google Translator to perform translations. **Translations performed by Google are not context-aware.** This means the term Google translated may not be the correct term for how that term is used in context in the country's business culture. For example, in the United States, the term "Plant" is commonly used for warehouse, but in other countries, the literal translation is a green plant that grows in your yard and does not have the contextual meaning of a warehouse or manufacturing facility.

It is recommended that a native speaker review the translations to avoid any translations that are out of context.

For an overview of this feature, see [Text Resources - How this Works](#).

Before You Start

Make sure you have **Enable Language Translations** set to "True" in **Configuration > Environment Settings > System Options: Enable Language Translations**. Once its is enabled, the text strings will

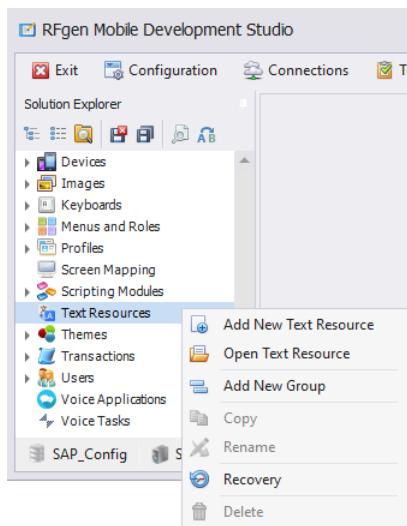
be available to other Solution Explorer objects like Applications, Menus and Roles, Testing etc as TextIds.

If you have a list of text strings you want to translate, see the **Solution Explorer > Text Resources Tools** icon and select **Import From Excel**.

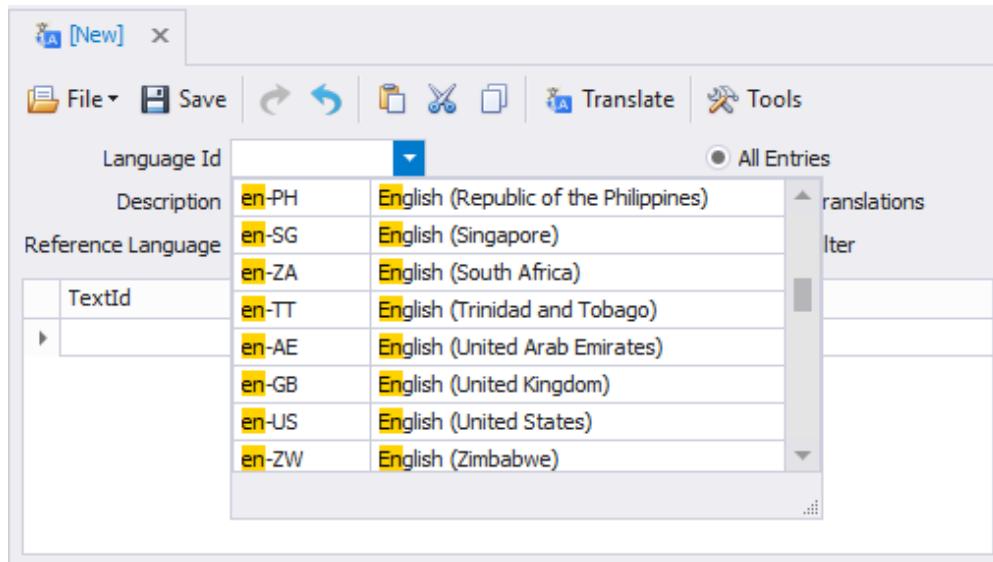
Step A. Add Your Primary Text Resource

1. First create your primary list of text strings (Text Resource table) in your native language. This text resource object will be your primary starting point/base for the strings to be used all the other languages. It will also serve as the source that provides additions and deletions to the other Text Resource objects in the other languages.

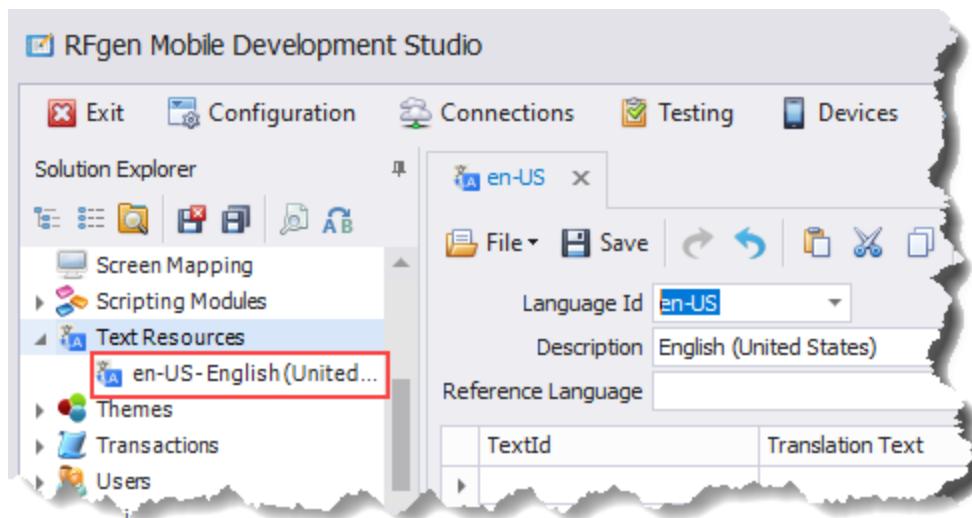
Navigate to **Solution Explorer > Text Resource**, right-click on the **Text Resources** icon and select **Add New Text Resource**.



2. In the popup screen [New], select your language from the **Language Id** menu list. Scroll to or enter the first two letters of the language you want while in the drop down menu. In our case, we entered "en-US."



3. **Save** your selection. This adds a new file under the Text Resource folder using the language it identifies with. In this example, the new resource is en-US-English(United States).



In the tab, leave the **Reference Language** blank.

	TextId	Translation Text	Reference Text
I	Plant_id	Plant	
I	Document_id	Document Number	
I	Location_id		

4. Enter your text string identifiers in the **TextId** column.

5. Enter the meaning of the id in the **Translation Text** column. The text in the **Translation Text** column is the text that will display in your application at runtime when its linked properly to the element in the application.

For example "Plant_id" and "Plant." The Translation Text can accept any alpha-numeric characters and symbol.

If the text string is colored, (**Location_id**) this means the entry is incomplete and a value is missing in one of the fields. In this case, the string "Location" was missing from Translation Text. Once the missing text is entered, the color changes back.

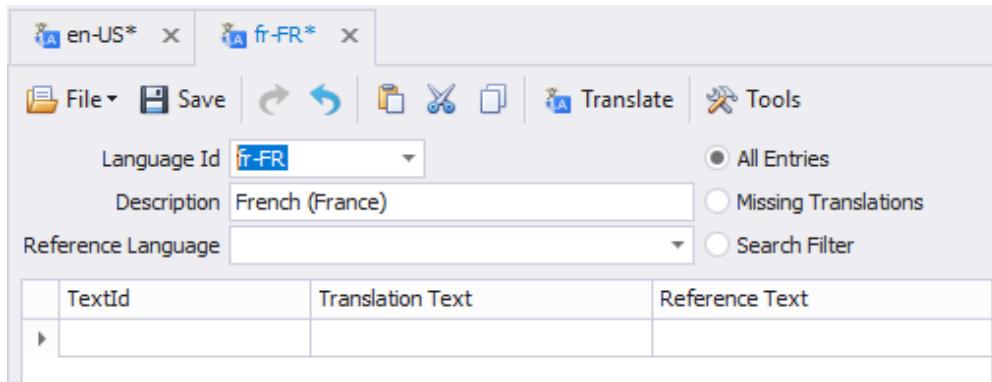
Keep the **All Entries** button selected.

6. Click **Save** to save changes.

Step B. Add Other Languages and Populate from Reference Language

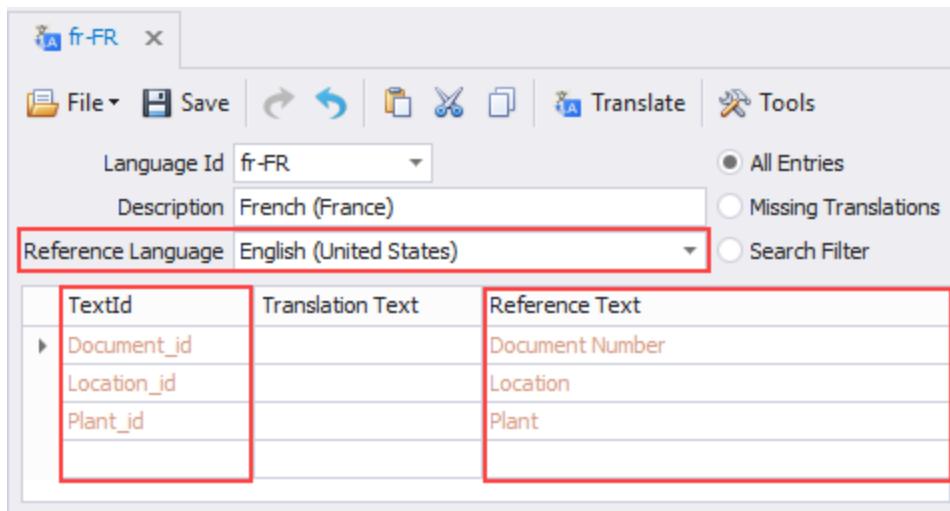
Once your primary Text Resource is created, you can now create the other tables that will store your translated strings in the locale of desired language. In this example, the new table will be used to store strings in French.

1. Right-click on the **Solution Explorer > Text Resources** icon and select **Add New Text Resource**.
2. In the Language Id menu, select the desired language. For example, enter "fr" to create a French Text Resource table.



3. Populate the fr-FR table's **TextId** and **Reference Text** from Step A above by selecting your primary resource language from the the **Reference Language** menu list.

In this case, we'll select "English (United States)".



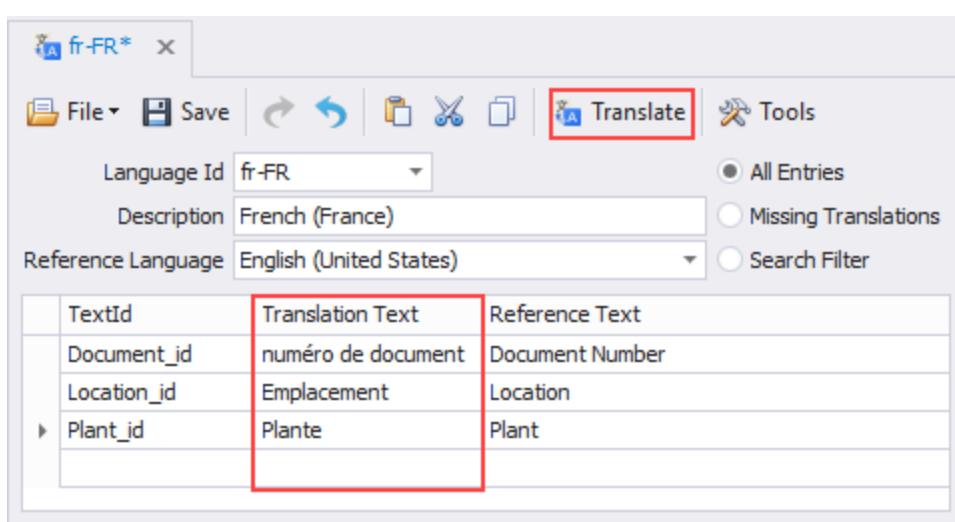
4. The fr-FR TextId and Reference columns are filled in using the TextIds and Translation Text from the primary resource (En-US).

The colored text indicates the Translation Text is missing.

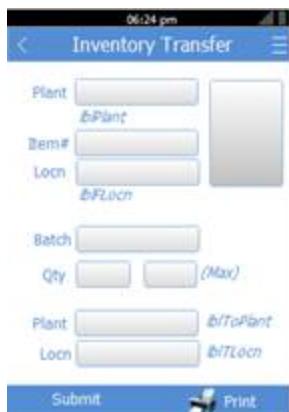
Click **Save**. The fr-FR file is added under the Text Resources folder.

Step C. Translate Text

1. To translate the missing values click on the **Translate** icon.
2. Click **Yes** on the pop-up message:
Translate missing entries using Google Translation Services?
3. The Translated Text is now filled in with the text strings in French.

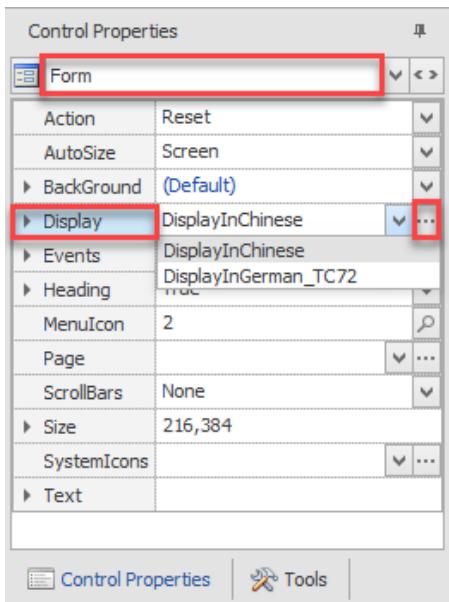


4. Click **Save**. This adds the object under the Text Resources node.
5. Repeat steps B and C to create additional translations.

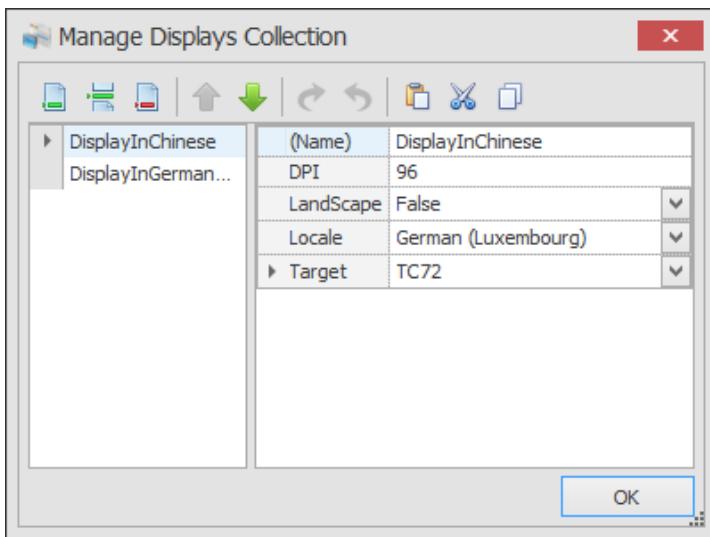
To Localize Text in Graphical Displays

Before you localize strings associated with a control, ensure the string to be localized has been entered in the **Solution Explorer > Language Translations** table.

1. From **Solution Explorer > Applications > [your application]**, open your application and select the **Form**. From the Form select the **Displays** property.



Click on the dots ...



2. Create a new Display GUI by clicking on the add or insert icons in the ribbon menu. Add the your display's name, target device DPI value, the orientation of the device, select a locale, and the target device type.

Note: The Locale value here needs to match the Locale set in Language Translations.

3. To translate a caption, click on the property that contains the caption to be translated. If the drop down menu is blank, you many need to add the text string ID to the **Solution Explore > Language Translation** table.

You can manually (you type in the translated string) in the Solution Explorer > Language Translations

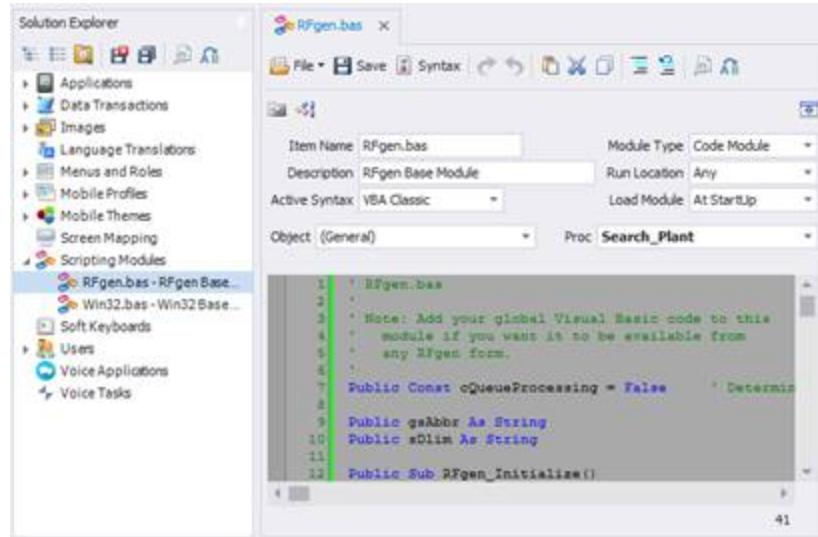
and then have each display pickup the translated string, or you can use the Google Translation Engine to translate your strings for you and then have the values associated with your strings in the display.

4. In the Caption property, expand the drop-down list.

An ID list displays. Select the text ID from this list. Click **Save**.

5. In the Form, change the **Active Display** property to the Display which is set the desire Locale.
6. The translated text should appear in the prompt on the form.

Scripting Modules



These modules are used to manage global scripts that can be used by any application, timed event, transaction macro, etc.

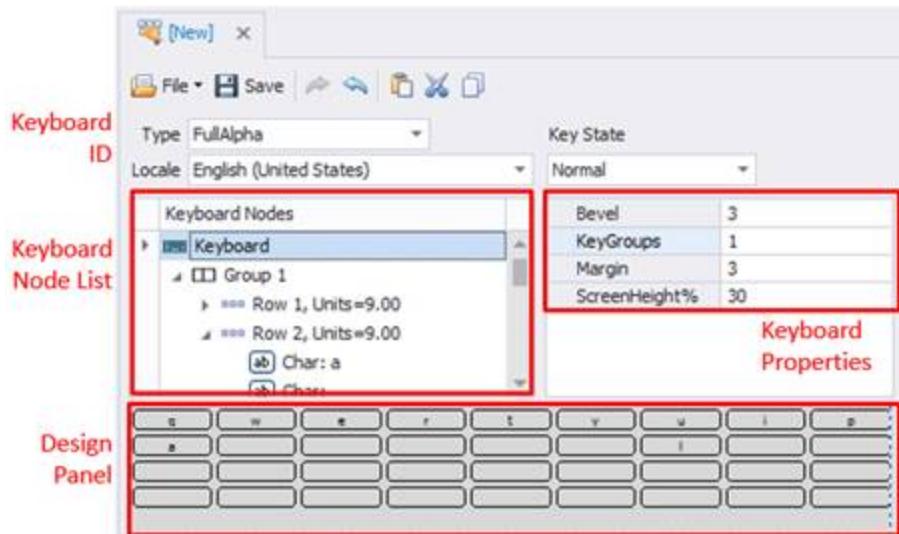
The **RFgen.bas** and **Win32.bas** files known as BASIC (Beginners All-purpose Symbolic Instructional Code) are installed by default in the **Scripting Modules**. Customized applications provided by RFgen are written in VBA (Visual Basic for Applications) and are in Macros (prescribed programs that automate routing tasks like fetching data and populating a specific field in an application.)

- Both .bas files contain global definitions of variables, functions, and procedures that can be referenced by any application or macro script.
- If you want these global definitions, variables, functions, or procedures to be available from any form in your application, add your Visual Basic code one of these base modules and customize it.
- Note: In RFgen version 5.0, "Scripting Modules" was named "VBA Modules".

To Make a Reference to Other Modudules

1. At the top of the Scripting Modules form set the Load Module to "When Referenced".
2. Click **Save**.

Keyboards



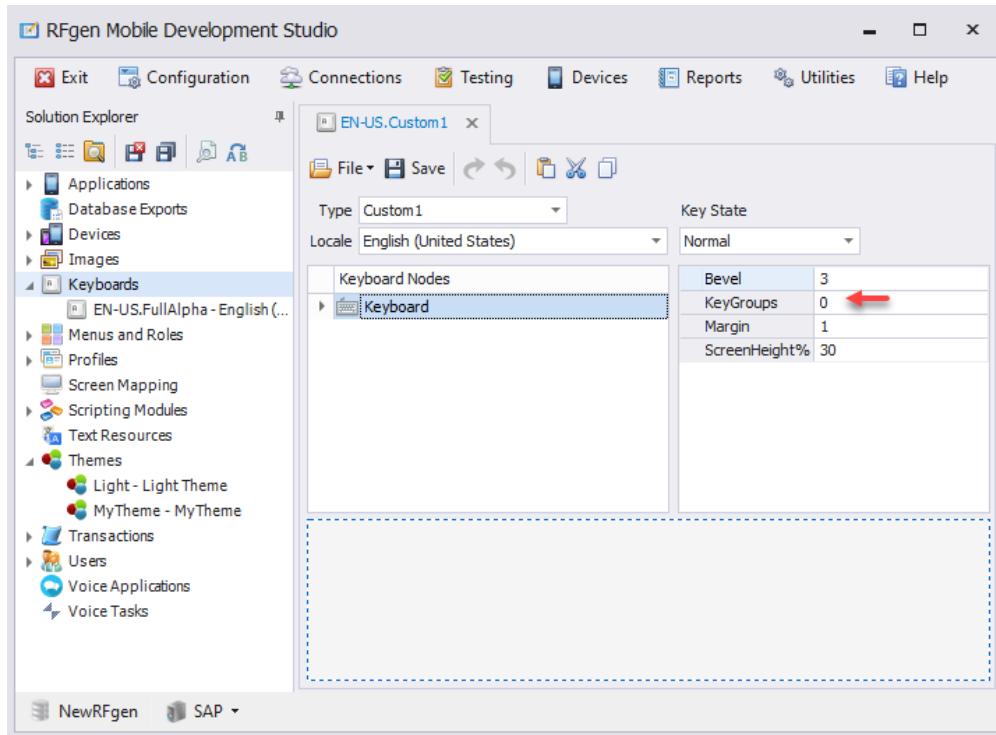
The **Keyboards** feature enables you to customize soft keyboard interface for entries that cannot be scanned and may need to be localized (translated) into another language. For example, if you added a new keyboard node (object), under the type "FullAlpha," this keyboard would be displayed if you selected "FullAlpha" in the Keyboard property of a Memo control or TextBox control.

To add and customize a keyboard

Before you start -- decide what type of keyboard you need, and if you need to customize a keyboard, or use a pre-configured file.

To use a preconfigured file, see [To import a keyboard file](#). To create your own keyboard see [To Add and Customize a Keyboard](#).

1. In the Solution Explorer, navigate to the Keyboards node.
2. Right-click on Keyboards and select **Add New Keyboard** to add a keyboard, or select an existing one under the Keyboards node. Depending on your selection, a screen similar to the one below will display.

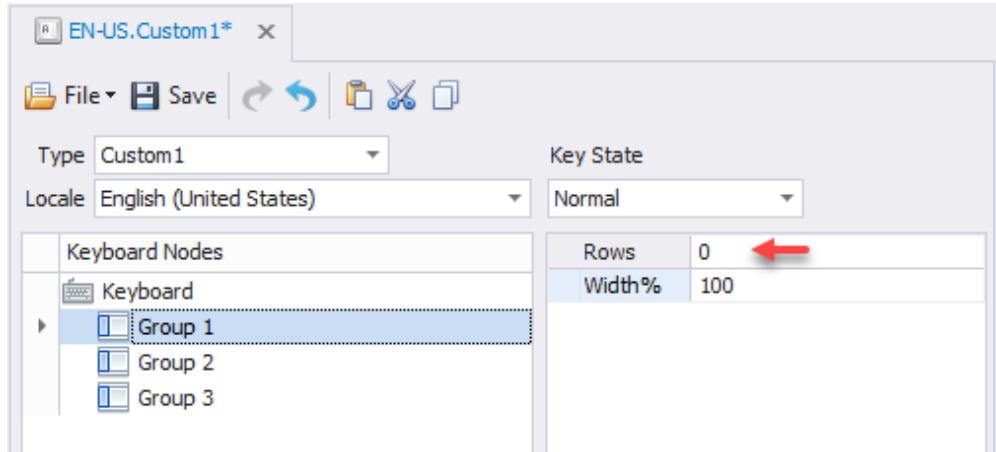


3. Select a Type, Locale, and Key State from the drop down list. Change the default of 30% of your target screen height (ScreenHeight%) if needed.

The Keyboard is organized by KeyGroups-- groups of common character types. For example, one for function keys, the second for alphabetical keys, and the third for numbers.

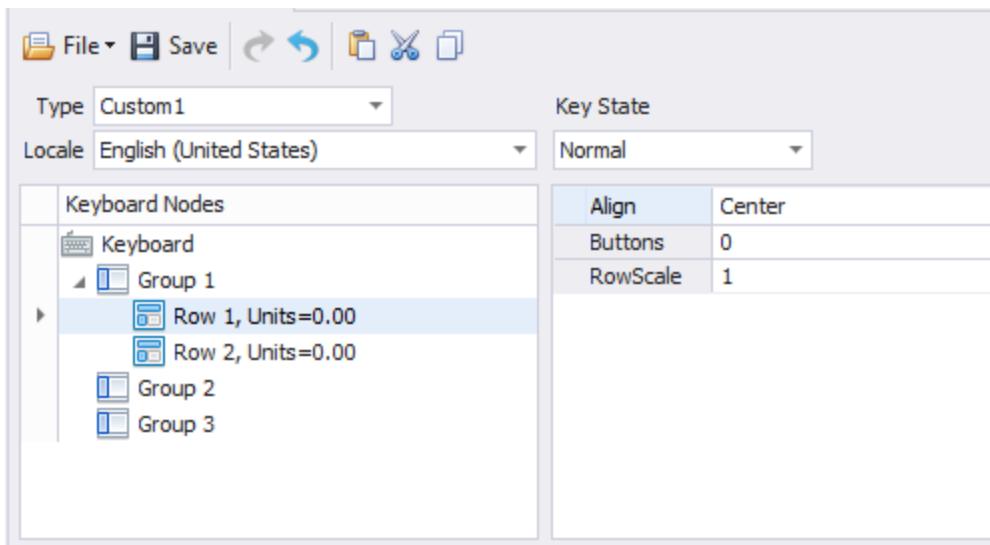
Enter the number of groups you want for KeyGroups.

After you enter a number for keygroups, each group is added under Keyboards.

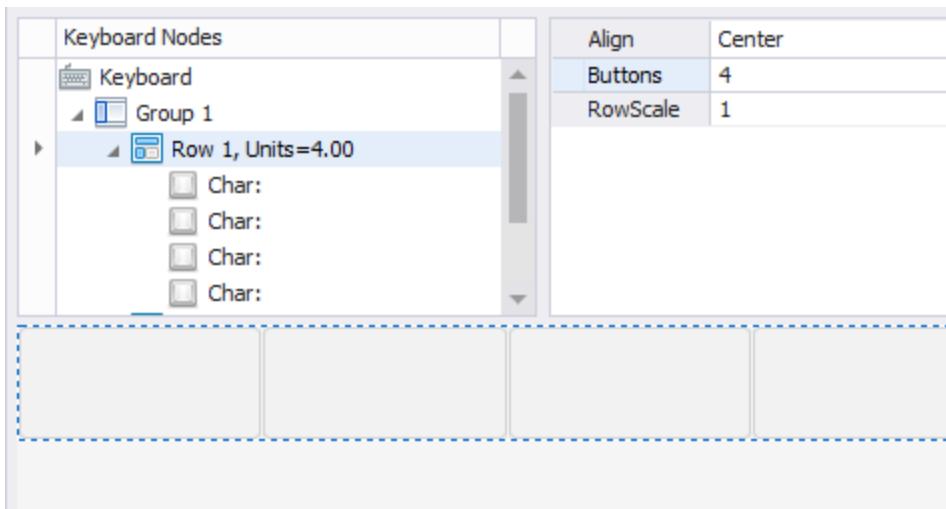


6. Select a Group, then enter the number of rows desired for each group. Unless needed, keep the width as 100 percentage for groups width for each row.

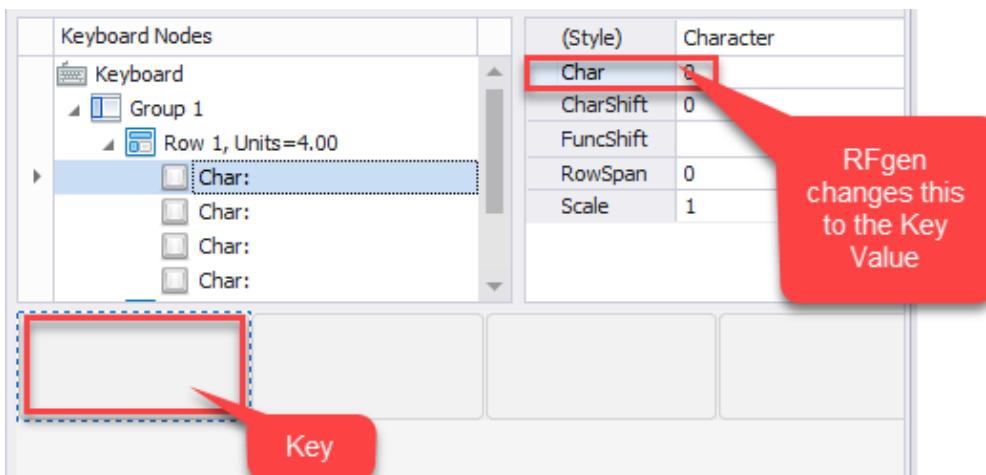
The rows are added under each group. Select a row and enter the number of Buttons. Keep the RowScale as 1 for now.



After you enter the number of buttons/row, the row units will equal the number of buttons. A list of characters are added under the row and in the display area, an layout of the row.



7. Select a character under the row, and then enter the values you want display on the key of the keyboard. For example, if you want the first char in the row (far left) to be the number "1", enter "1" for the Char value (upper right). RFgen will automatically change the Character to the Key Code Value for the local of the keyboard. For example "1" is key code value 49.



Repeat assignment of the values for all characters in each row for each group.

To Stylize a Keyboard

The look and feel of the keyboard is set from Solution Explorer > Themes > [name of the theme] > Element = Keyboard.

Keyboard Designer Features and Options

Bevel – Shapes the outer edges of keyboard.

Caption – Enables you to enter text for buttons that may have a function other than a character or numeric key value. (i.e. Tab, Shift, Delete.) **Caption** displays if **Key State** = **CapsLock** and **(Style) = Character + Text** or **KeyDown + Text**.

KeyGroups. Groups the sections of the keyboard. For example, the left section is alphabetical keys and the right section, numeric keys.

Keyboard ID – Enter a text or numeric identifier.

Key State – Associates character or function of the key based on one of these three states: Normal, CapsLock, and FuncLock

For blank space buttons, set the **(Style)** to *BlankSpace*.

If the **Key State** is *Normal*, you can assign a button two values: the lower-case letter (**Char**) and its upper case (**CharShift**).

If **Key State** is *FuncLock*, you can only assign one value, the Fkey to the button. Leave **Char** and **CharShift** to 0.

Type – Allows presets settings for all alphabetical characters, alpha-numeric characters, or all numeric characters.

Locale – The Local/language that the characters/labels of each key in the keyboard.

Margin – Sets the spacing between buttons.

RowSpan – Sets the button height based on the number of rows there are above and/or below the button.

ScreenHeights% – Is a percentage of the screen size.

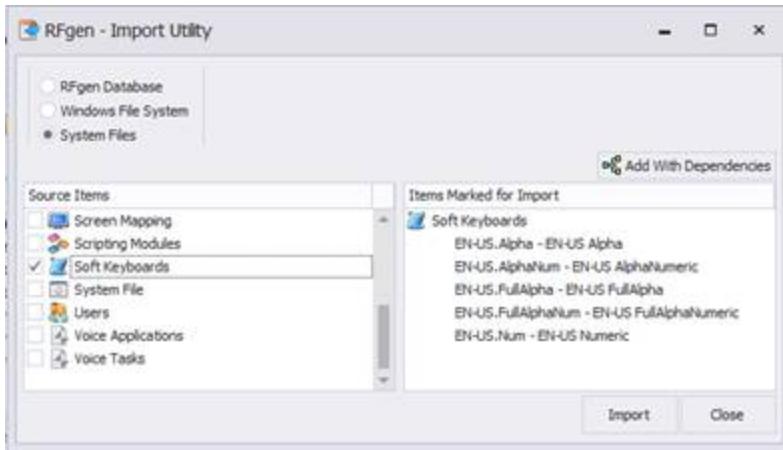
Units – The number of units a row is divided into. (I.e. number of equally sized buttons in a row.) The most granular number of units will be the base set of units for all rows. This way, a button can be allowed to span 3 or more units.

The profile automatically sets the keyboard theme (if a keyboard is used), but wanted to verify if this is the case.

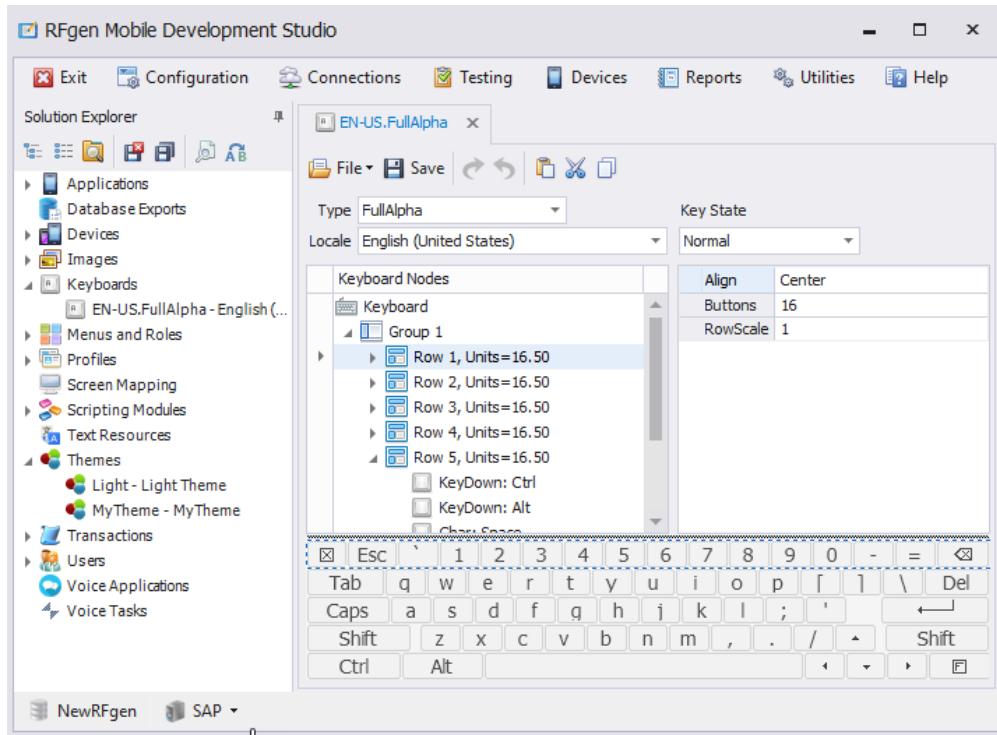
To Import a Keyboard

If you would like a template to help you get started with modifying a softkeyboard, you can import softkeyboards from System Files.

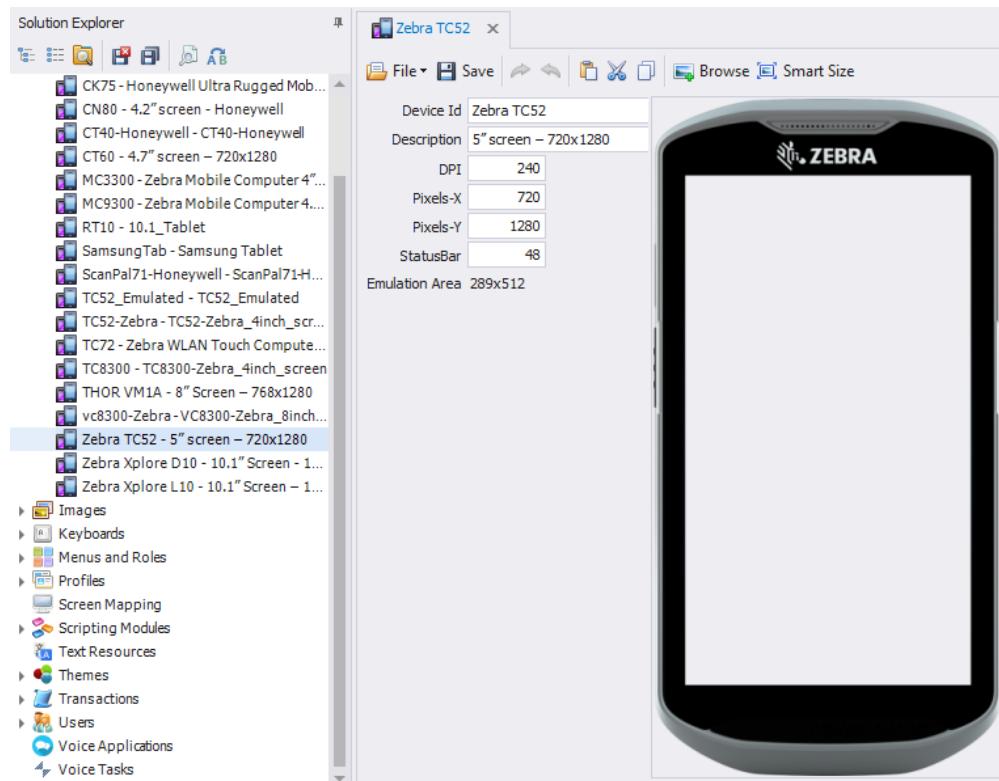
1. Select **Utilities > Import Solution Files**.
2. Select the **System Files** button.
3. In the Source Items panel, expand **Soft Keyboards**, and check the the specific Keyboard(s) to be imported.



4. (Optional) Click on **Add With Dependencies** to include any other files that might be used with Soft Keyboards.
5. Click on the **Import** icon. This will import the templates to the **Solutions Explorer > Keyboards** tree.



Devices

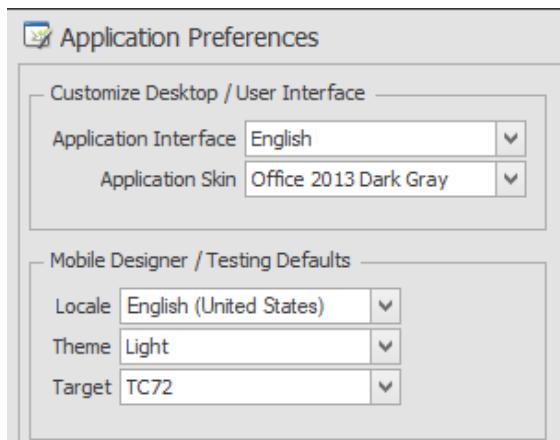


RFgen provides ready-to-use images of popular devices for quick and easier design layout of applications as RFgen.

If you don't see your device listed, click on the **Browse** button to [upload your own image](#), and set the dots/inch and display area resolution for scaling purposes.

The device (device skin) also features auto-scaling of prompts in an application installed on different devices if the devices are of relatively similar resolution. If the device image is larger than the RFgen Mobile Dev Studio screen/Designer, use the **Smart Size** button to scale the large image further in the emulator. This will resize it to fill the display area of what you can see in Dev Studio.

Once your target device is uploaded, or, if you have a favorite skin you want used for all your applications and in testing, you can set a default, target device in **Configuration > Application Preferences**.



To Add a Device Skin

If the device image (skin) for your [target device](#) is not listed under Devices, you can upload your own device skin, set the display's resolution and parameters, save the image, then use it for application design purposes.

Image and Information Requirements

The image must have the screen area cut out so the background is transparent. The width at the top/bottom of the screen to be the same for application layout/design purposes. The transparent area helps RFgen detect the application form/display area, so this should be as accurate as possible.

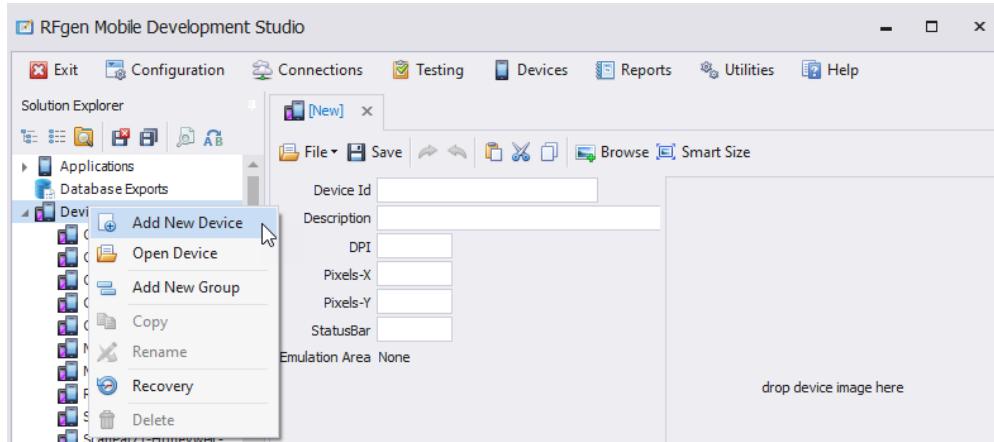
The best file types to use are png, jpeg or bmp. Do not use gif. RFgen does NOT accept files from applications such as PhotoShop or CoralPaint. (If you modify a file in one of these tools, make sure you export or save it as png, jpeg, or bmp. RFgen can accept high resolution files as RFgen has a built-in tool to reduce the size if the image if its larger than the RFgen Development Studio interface.

You will need the screen's dimensions which you can obtain from the Device Product Datasheet or Specification. It should list the **screen width and height in pixels**, the **screen size** (measured diagonally across the screen), and the screen resolution DPI. If the manufacturer datasheet/specifications did not provide the product's screen dpi, **you can calculate the dpi by**:

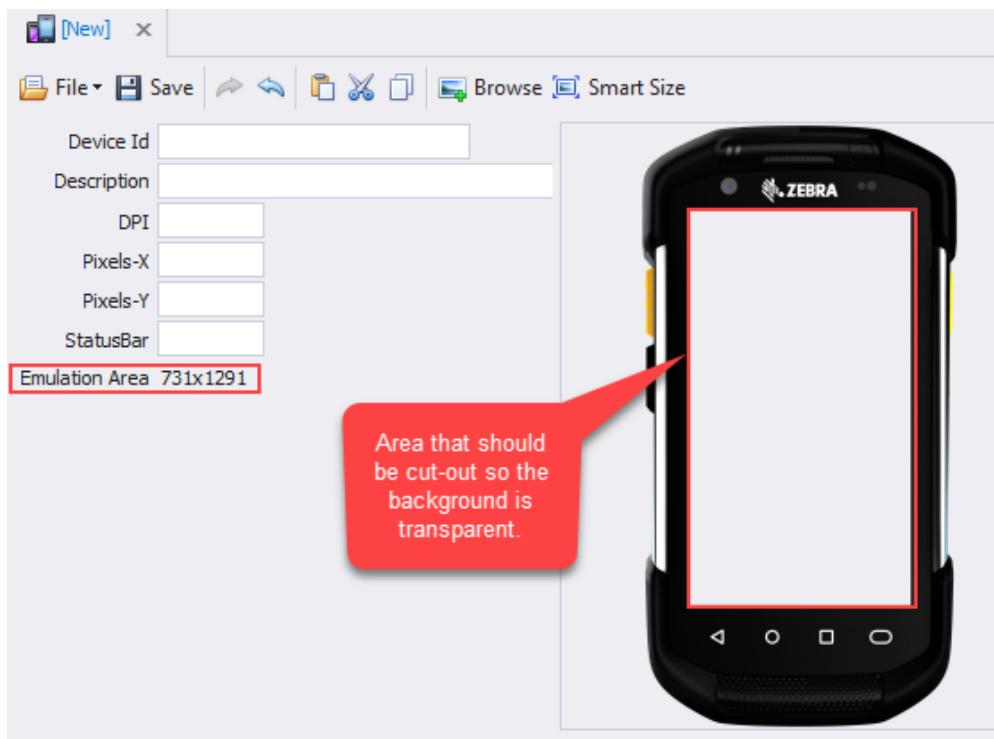
- a. Adding the squares of width (pixels) and height (pixels).
- b. Getting the square root of the sum from step a.

c. Dividing the square root from step b by the screen size (inches). This gives you the dpi.

Steps



1. From the Solution Explorer > Devices node, right-click on Devices and select **Add New Device**. A "[New]" tab displays.
2. Click the **Browse** icon. A Windows browser pop-up displays.
3. Select the image file to be added. Click Open. The image displays as shown below.



4. Enter the **Device Id**, **Description**, **DPI** resolution (dots/inch), screen width (**Pixels-X**) and length (**Pixels=Y**).

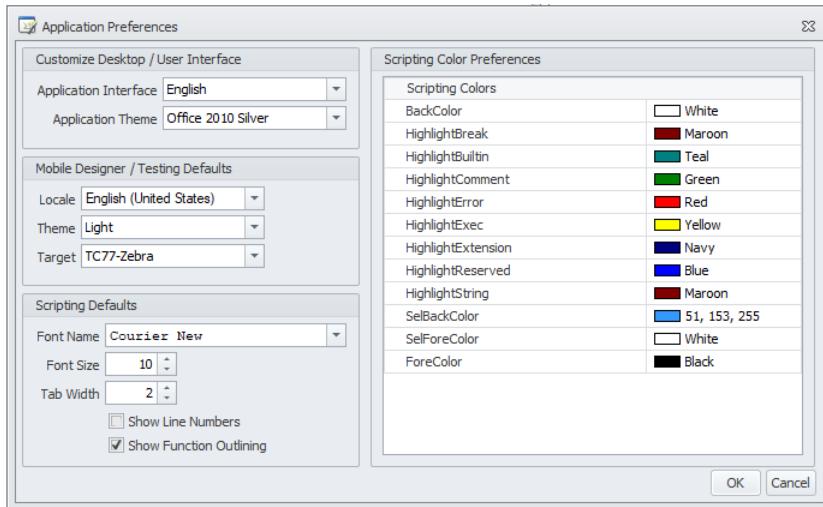
Refer to "Image and Information Requirements" above for details on obtaining the proper information. If RFgen cannot detect the cut-out area for the screen, the words "Emulator Area None" under the status bar.

Note: Commas are not accepted values for DPI, Pixels X or Y.

The **Status Bar** can be left blank or filled in. This value dedicates the screen area consumed by the status bar so RFgen knows how much to use for the form display area.

5. Click **Save**. You are now ready to use the Target Device in an application form.

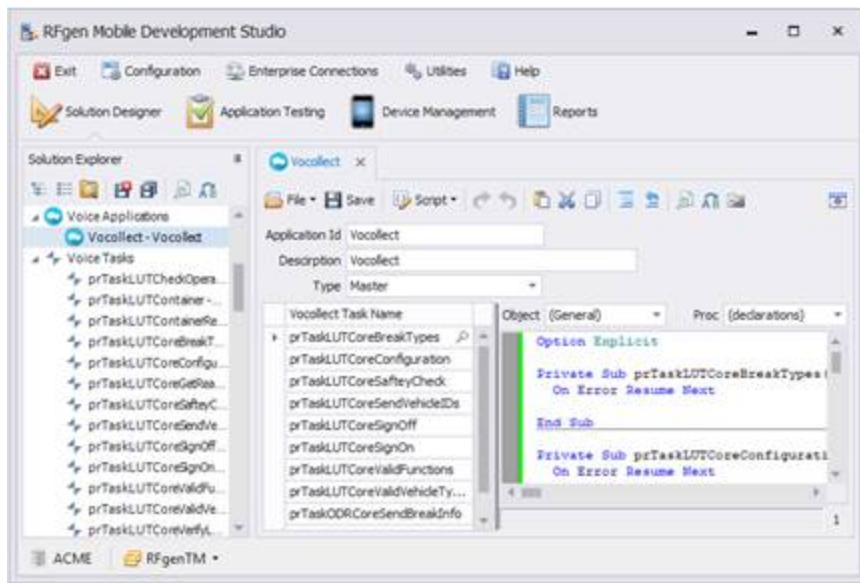
TIP



If you want ALL of your devices to use this particular device, open Configuration > Application Preferences, and set Target Devices to the desired device. This setting will apply to new application forms but it will not overwrite settings in existing applications.

If you need to share this image file with another RFgen server, you can export it using Utilities > Export Solution Files: <select desire Device Files to be exported>. Enter the destination Windows File System location and click Export. A message will display when this is done.

Voice Applications



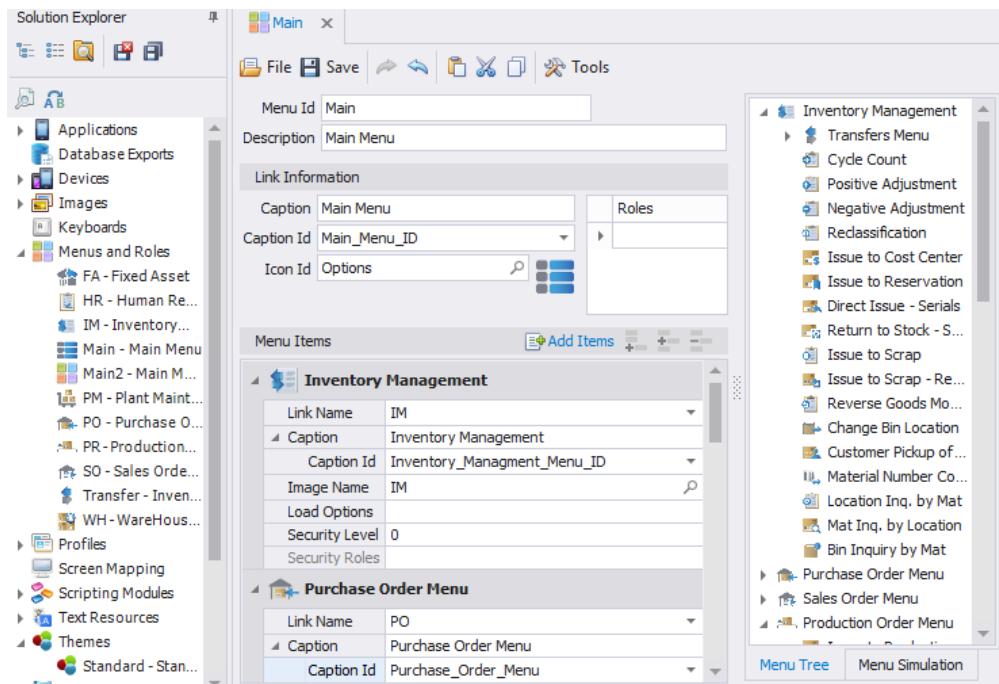
Voice Applications are warehouse tasks you execute through verbal commands saved in an application that is supported through third-party systems such as Vocollect. It requires servers that support voice-driven solutions and devices.

To create a new voice application

Before you start, you must first have the desired voice task. See **Voice Tasks** and **To Import Voice Tasks** for details.

1. Right click on the Voice Application icon to add a new application.
2. Enter your **Application ID** and **Description**.
3. Select the **Type**. Master refers to the master database from which the task (script) is resourced.
4. Under Vocollect Task Name, in a blank row, click on the search icon. The Select Tasks list will display.
5. Click on the desired Task(s) to be added then click OK.
6. The selected tasks should display in the Vocollect Task Name field.
7. In the Proc (Process) listbox, select the task prTask to be added to your application.
8. When done, click Save.

Menus and Roles



Roles and Menus are the application(s) you assign to a menu (i.e. Inventory Menu) or user role. You can control which application(s) a user may access by assigning them a specific menu in the Solution Explorer > Users > [user name]> Menu Tree.

The *Main Menu* shown above is an example of a parent menu that will be used to give the user access to applications under each submenu *Inventory Management Menu*, *Purchase Order Menu*, *Sales Order Menu* etc.

You can also setup a user to access a specific application by creating an menu just for that application.

Use the **Menu Tools** icon in the menu bar to manage menu content. For more details, see [Menu Tools](#).

The **Menu Tree** is a type of view that displays all your menus and items linked to each menu.

The **Menu Simulation** view shows how your menus would look on screen or inside a device. For more details, see [Menu Simulation](#).

Assign menus to user or user groups

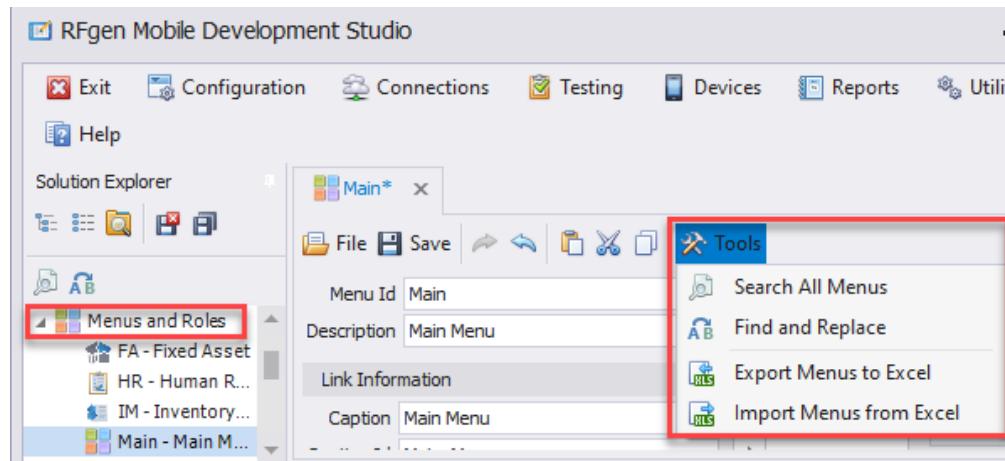
Once the menus, submenus and applications are setup, you can assign the menus to user accounts in the **Solution Explorer > Users** tree. This helps control which Mobile Apps a user will can access on their mobile device or Windows desktop system.

When testing your solution, the process generally begins with the Login app, then the menu and then the application selected in testing.

If you plan on testing using this path, a default user (i.e. Sam), menu and app (i.e. Login), and menu of other apps should be setup before you begin testing.

NOTE: The *User Management Console* also provides features that allows Console users to setup Menus and Users without providing Console users full access to everything on the server. This enables database updates from the server or the console by different users. For more details, refer to the *User Management Console* section in this guide.

Menus Tools



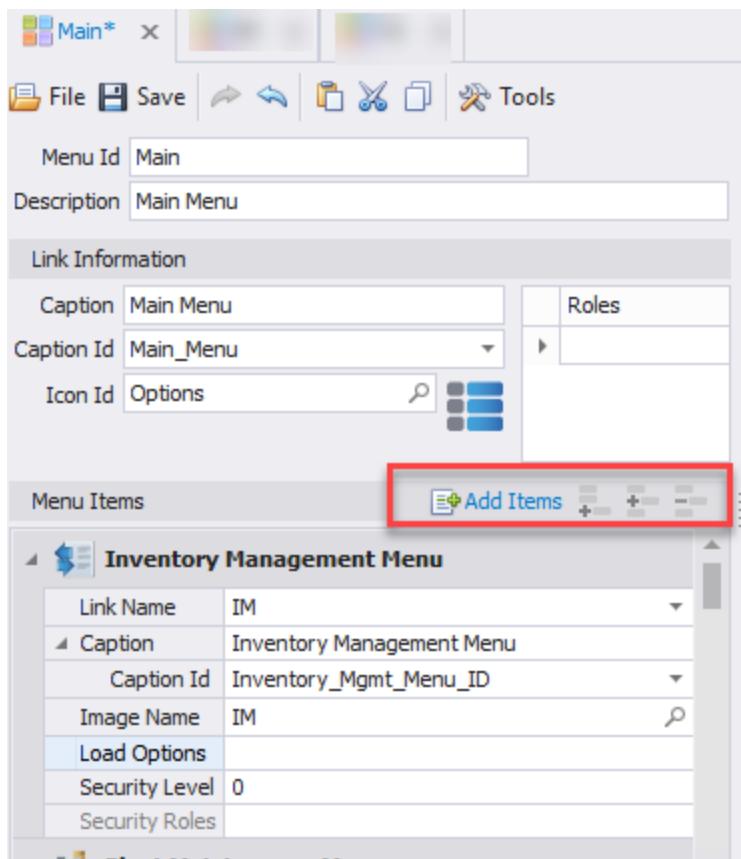
Menu Tools in the Solution Explorer > Menus and Roles is used to search all menus, Find and Replace specific strings (i.e. Menu ID, description etc), or import / export menu or menu items to Excel.

Search All Menus - This filter enables you to search for entire records and/or fields that match a text string or ID in your RFgen Menu and Roles folder (Solution Explorer > Menus and Roles tree). You can optionally export the results to Excel. Available values are: Record Body, Description, Heading, Default Icon, Item Id, Item Text, and Item Security Level. The results are listed in a table with headers *Menu Id*, *Found In*, and *Text*. This tool support matching of upper and lower case letters and whole words.

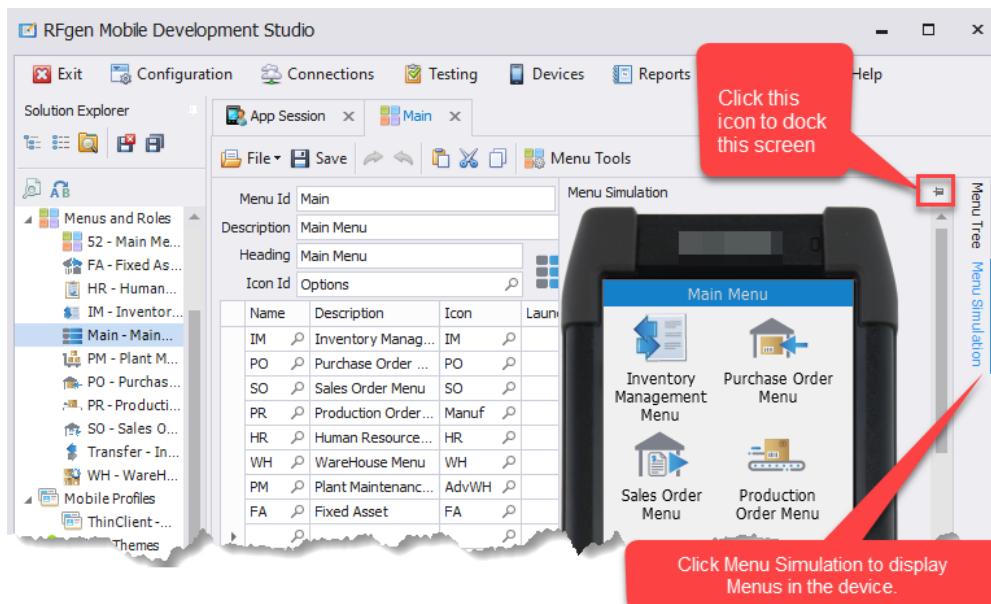
Find and Replace - BEFORE you use this powerful tool to replace text (names of menus etc), generate a list of the item you are replacing in "Search All Menus" and export it to an Excel file. The Results will list the *Module*, *Name*, *Line #* and *Text* where the changes occurred. Once a change is made, it cannot be "undone".

Import or Export Menus - See [To Export or Import Menus](#).

Add Items - Once you complete the menu fields (i.e. Menu Id, Description etc.) you can add your menu items through the **Add Items** too



Menus Simulator



To view your menus as they would appear on screen, click on the **Menu Simulator** tab.

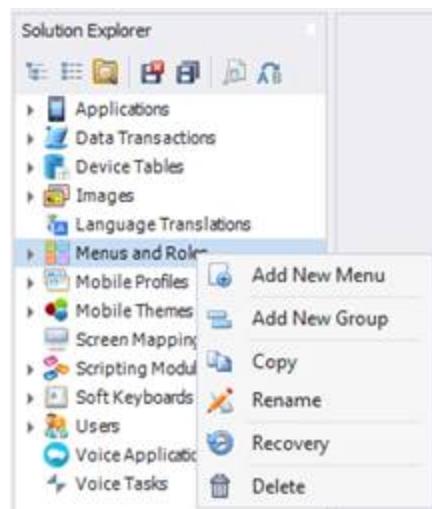
Note that you can have the menus display only in the screen or with the screen inside the device to view the full effect.

To view the menus inside a device, (called the device viewer/emulator) check the "**Show Target Image**" in the **Testing > Mobile Apps > Options** menu.

Note: If you wanted to change to a different device, use the Testing Options menu.

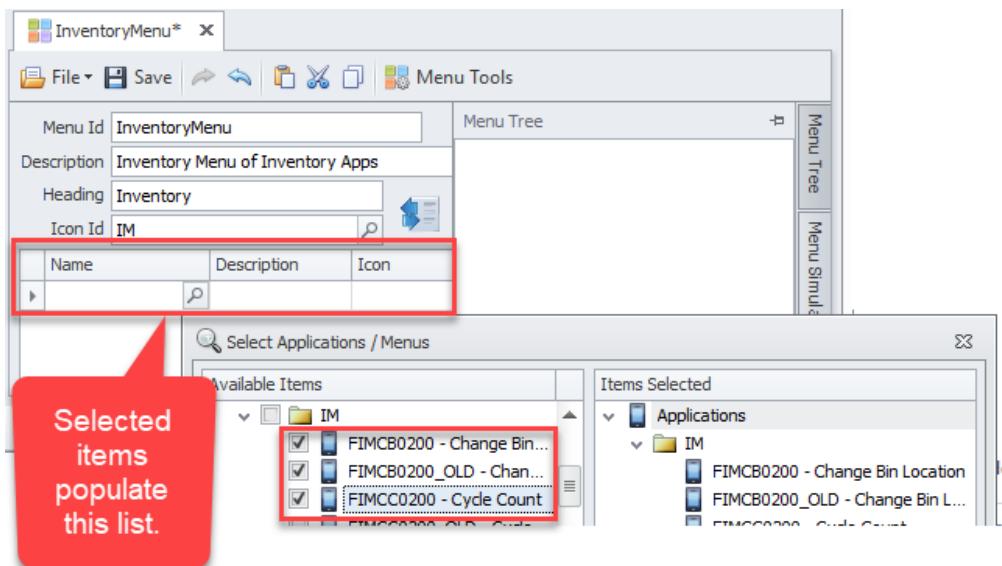
To dock the Simulator view, click on the tack icon in the upper right corner so it points down. If its sideways, the view is hidden.

To Add Menus



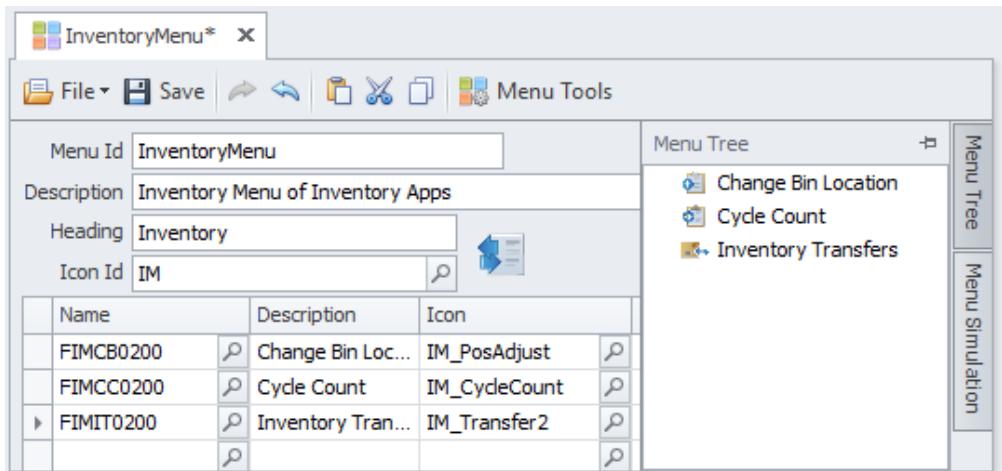
This topic describes how to create multiple categories (i.e. multiple 1st level) and second-level items for your menus.

1. From the **Solution Explorer**, right-click on **Menus and Roles** and select **Add New Menu**.
2. A blank form displays.
3. Complete the **Menu Id**, **Description** and **Heading** as your top-level (parent) menus. The **Icon ID**, icon image and **Heading** are optional. This creates the top-level menu.
4. After you created your top-level menu, add the items (applications). These will be organized as child menus and appear as second-level menu items at runtime.
5. In the **Name** field, click on the **Search** icon to list your applications.



Example Inventory Menu for three applications.

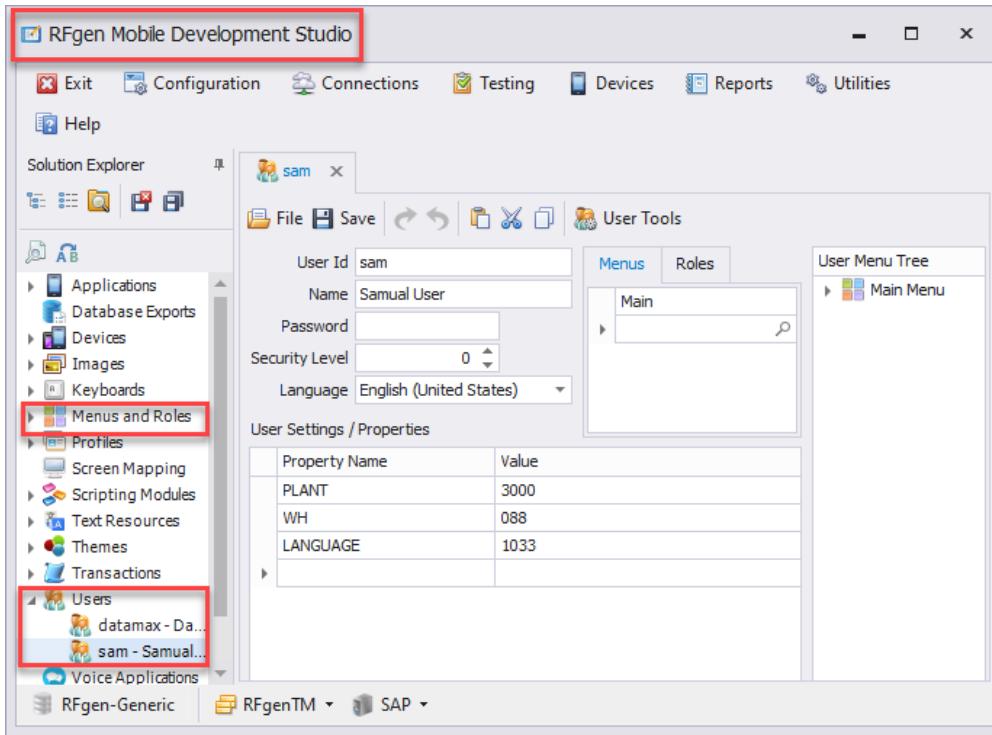
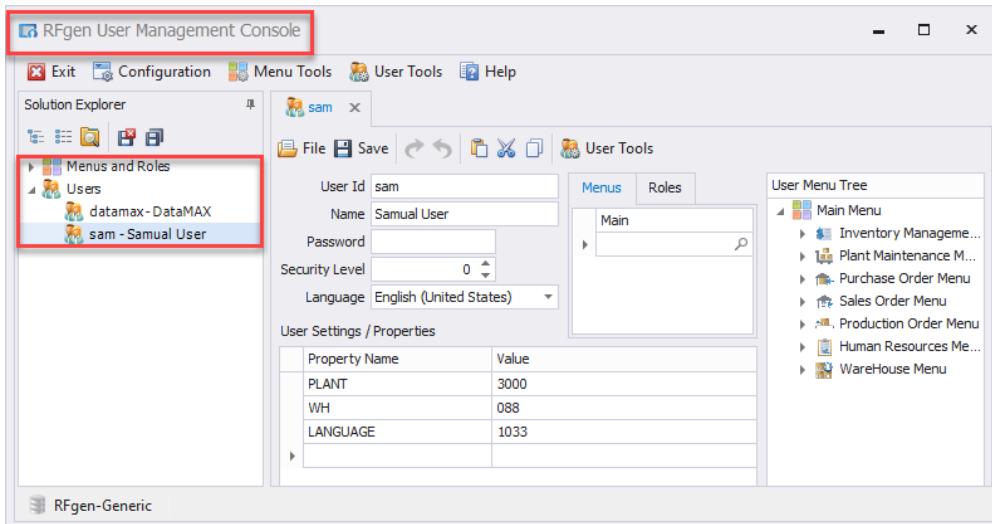
6. Check the ones you want to add. Complete the **Description** field and add the icon you want shown for the application. Your selected items are added to the Menu Tree.
7. (Optional) To add an image, in the Icon field, click the **Search** icon which obtains images from the Images resource folder.



8. Click **Save**.

To view the menu as it would appear in the target device, click on **Menu Simulation**.

User Overview



You can setup who has access to specific applications via the Menus and Roles settings in the **User Management Console** or **RFgen Developers Studio** by: 1) Creating a menu or menu group in the **Menu and Roles** screen; 2) Assigning applications to that menu; 3) In the Users screen, adding individuals; and 4) Assigning the menu to individual users. The default setup allows all users access to the menu and application under a menu unless you "tag" each with a specific Role name.

To exclude all users' access to specific menus/applications, you assign/tag the individual's user profile with a unique role name and assign that same role name to the menu and to the application.

When the user logs in, the user will only see the Menu item and applications with the matching role name that were assigned to the individual.

Role names can also be assigned via script. See the [App.UserRoles](#) in the Developers Reference Guide.

Additional Information

For descriptions of the user fields, or, how to add or remove a user, see [To Add Users](#) topic.

For more details, see [To Limit User Access to Applications](#).

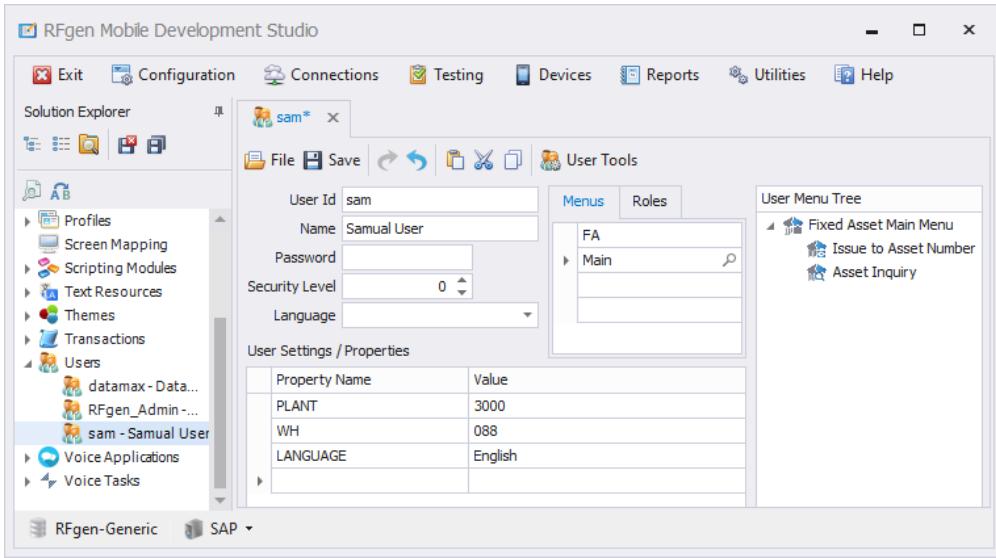
To Create New Users

Create your menus (or roles) in Menus and Roles before you create your users.

If you want to limit access to some users (Warehouse employees) but allow access to other users (i.e. Administrators and Managers) see the topic [To limit access to specific applications in a menu assigned to multiple users](#).

1. Navigate to the **Solution Designer > Users tree**.
2. Right-click on an existing user (or in the blank space) to add a new user, or right-click on the "Users" object and select **Add New User** from the menu.
3. The [New] user tab displays. Enter the user's information.
4. The **User Id** is required, but the **Password** is optional for a user account. SAM's startup menu is 'Main Menu'.
5. The **Security Level** is a numeric value between 0 – 100 that will be compared to the menu's required security level before allowing that user access to the following menus or forms.
6. The **Language** is used to assign a locale to the user's session. This field is optional.
7. The **User Settings / Properties** field is an advanced developer feature that is used to associate data values that are used repeatedly with the individual. For example, if Mary works in a specific warehouse, and you want her login to be associated with that specific warehouse (i.e. Plant ID: 3000), the information entered in the Property Name and Value fields will associate that Mary with plant 3000 so she does not have to enter the id "3000" when interacting with an application that requires a plant ID.
8. Continue with [To Assign Menus to a User](#).

To Assign Menus and/or Roles to a User



1. From the **Users** tree, select the user you want to work with if the user profile isn't displayed already.
2. In the **Roles/Menus** table of the **Users** tab, click on the Find icon and check the menu item to be added.
3. Click **OK** when done. The selection appears in the User Menu Tree (far right panel).
4. Click **Save**.

To Remove a User

You can remove a user by right-clicking on the user and selecting **Delete**.

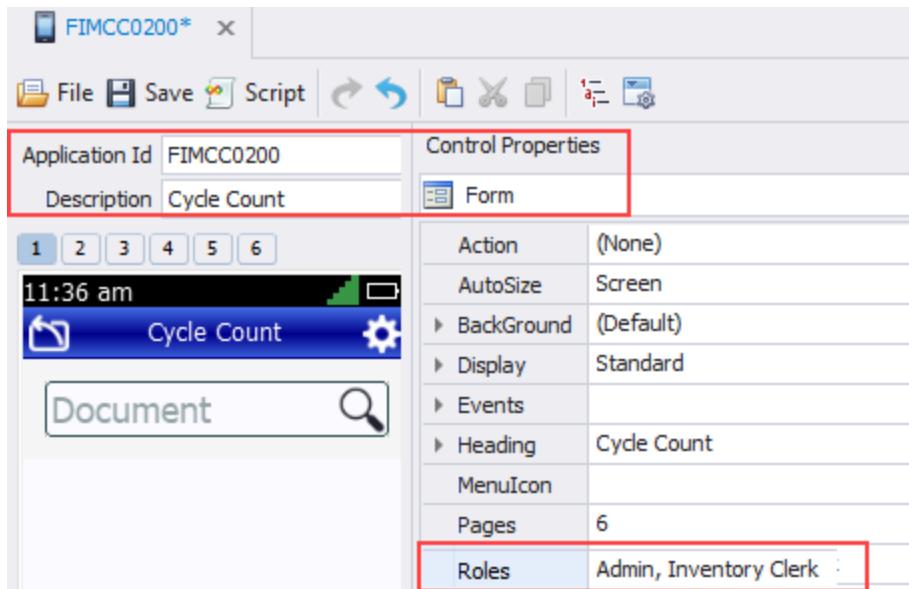
To User Access to Applications in a Shared Menu

If you want to limit some users (Warehouse employees) from having access to certain applications (i.e. administrative applications) while providing access to other applications in a menu assigned to users with different roles, you can do with by entering role names in the application, assigning the roles to the users in the Solution Explorer > Users group, and also setting up which roles are able to access a menu in the Solution Explorer > Menus and Roles profiles.

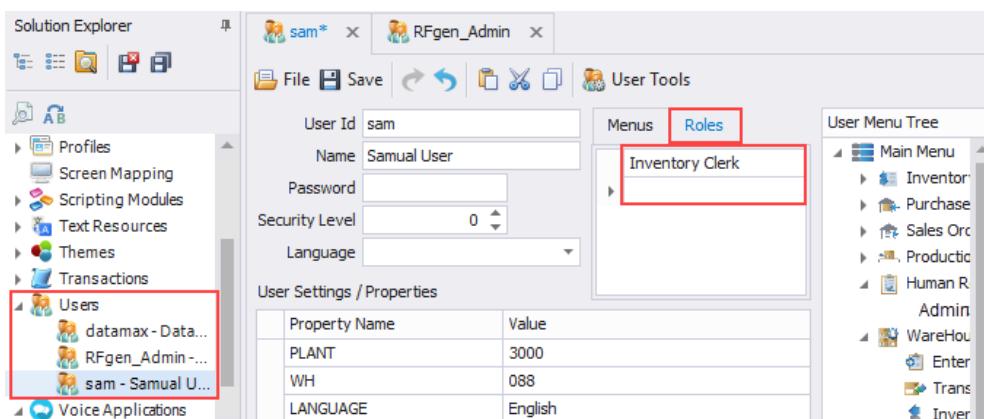
For more details on how this works or to set the role name via script rather than the user interface, see the topic "[Users](#)".

To enable application access to specific users

1. Navigate to the **Solution Designer > Applications > Form**.
2. In the **Form Control Properties > Role Properties**, enter one or more role names. In our example, we have a Cycle Count application that should be accessed by the RFgen Administrator and all warehouse clerks. So we setup two roles: "Admin" and "Inventory Clerk". We also have a Human Resources app that we want ONLY the RFgen Administrator to access, so in that application's Form Control > Role Properties field, we only enter "Admin".

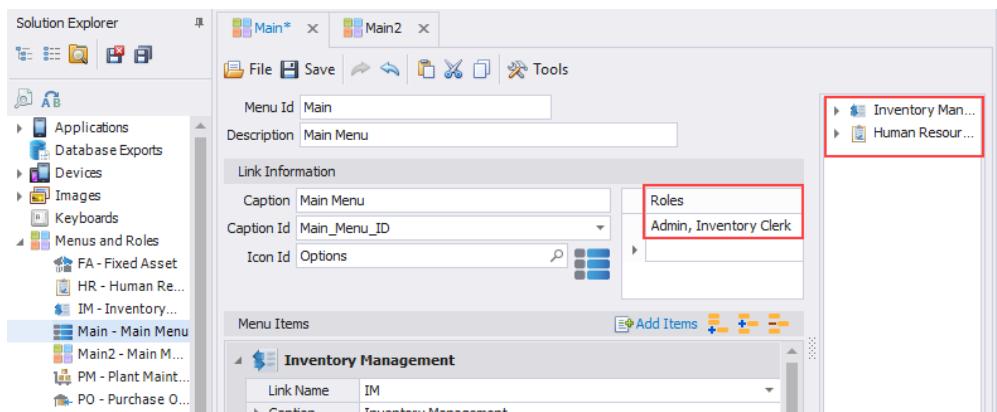


3. **Save** the changes.
4. In the **Solution Designer > Users > [user profile]** screen, add the role name to the Role tab.
For example, if you had a user named Chris who is the RFgen administrator and needs access to all applications, enter "Admin" in Chris' Role tab.
For all other users who process inventory but should not have access administrator type of applications, enter "Inventory Clerk" in their Role tab.



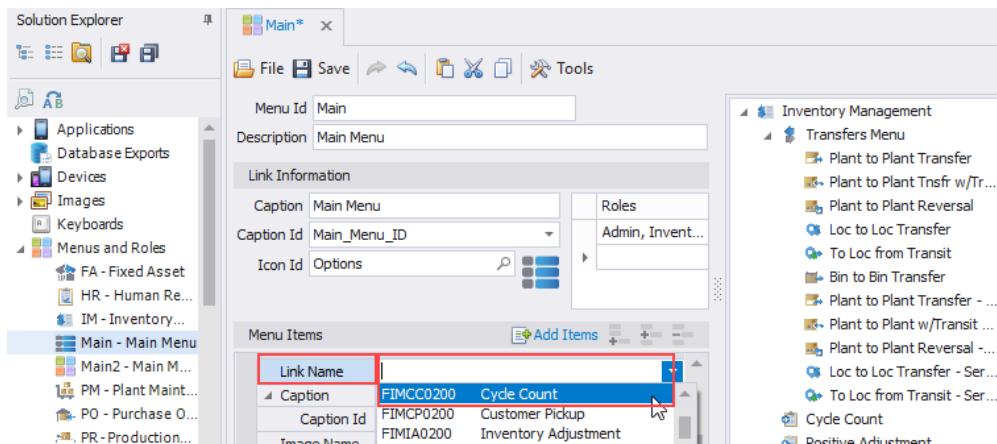
5. In the **Solution Designer > Menus and Roles > [Menu profile]** screen, assign the Application(s) to the Menu tab.

For our example, a menu called "Main Menu" was already created in the Menus and Roles folder. This Main Menu already has an Inventory Submenu and an HR submenu assigned. In this case, we add the "Admin" and "Inventory Clerk" roles added to the Main Menu Roles tab.



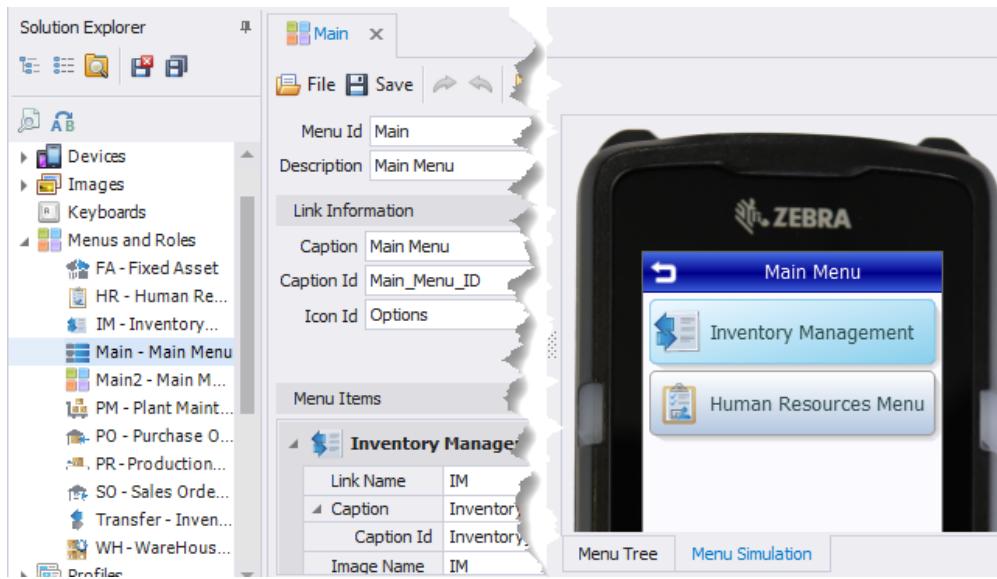
The Cycle Count from step 1 is added under Menu Items so it will be a member of the "Main" menu.

Click Add Items icon. In the empty table select the down arrow under Link Name, and select the Cycle Count app.



This step is repeated for the HR application. Now the Menu "Main" is comprised of an couple of Inventory apps and an HR app.

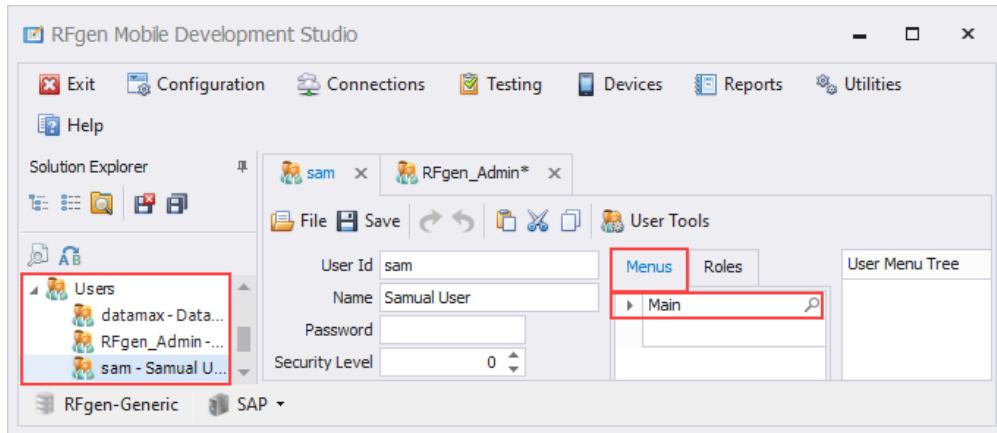
Optional: Click on the Menu Simulator to preview your menu.



6. Return to the **Solution Designer > Users > [user profile]** screen.

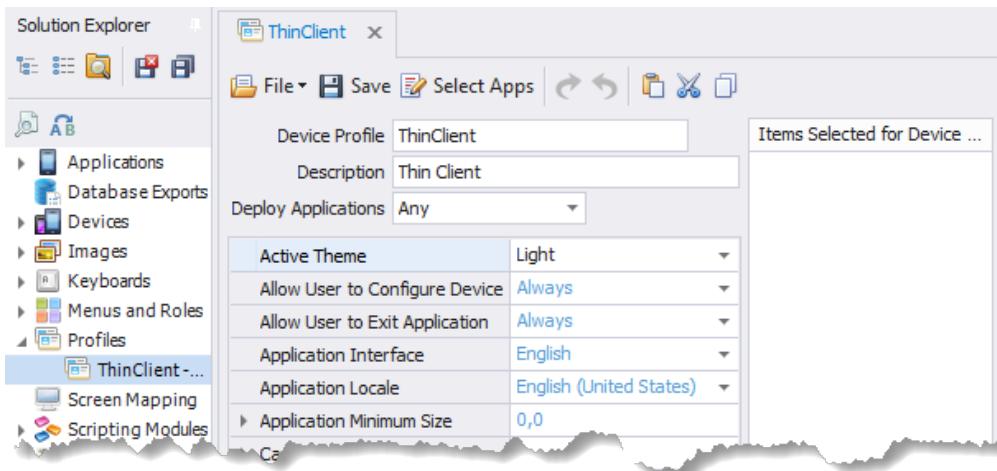
Assign the Menus for the User via the Menus tab. Select the user from the tree, and enter "Main" in the Menu's tab

In our example, the RFgen Admin and Sam (Warehouse employee) are both assigned the "Main" menu in the Menu tab.



7. To test the menu, go to Testing. When Chris logs in, Chris will see the Main menu and have access to the apps that contain the tag "Admin". When other users with the Inventory Clerk role login, he or she will see the Main menu but only those apps which had the role "Inventory Clerk" assigned.

Profiles Overview



A **Profile** is used to package the configuration settings (i.e. Server ID, Applications, Theme, Camera settings etc) of your RFgen client. Once your profile(s) are setup, they are deployed to the device once the RFgen client connects with the server, or by other means (i.e. physical transfer via thumb drive, deployment via RFgen Solution Deployment etc.)

To Setup a Profile

In the Solution Explorer, navigate to the Profiles object. Right-click on Profiles and select "New" from the menu, and fill in the information in the following fields.

For more details, see [To Create a Profile](#).

Device Profile is the name you assign the profile you are creating. A common name to use is "Thin" or "Batch" but you can use any name that suits your needs.

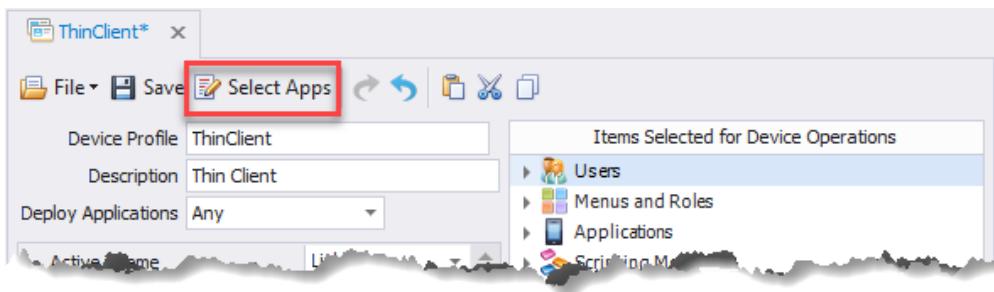
- * **Thin Client** The traditional wireless real-time interface to the server where the mobile device is restricted to the RF environment.
- * **Offline Client** Also called Offline client - allows the user to leave the RF environment and manually or automatically switch to and from a connected state and continue processing data.

Note that the Mobile Settings for Thin and Batch are different. For example, Thin Clients do not require a Local Database setup because the application and data are "projected" from the server and is not saved locally. However, if a Batch Client disconnects from the network, collection may continue as data is saved to the device's local database.

The **Description** explains the purpose of the profile and displays in the tree when saved.

Deploy Applications contains the list of client platforms. *Any* is used to deploy a profile to any client type (Android, iOS, or Windows Desktop). For Windows CE, select Windows CE as special packaging of the profile may be needed via a CAB file.

To select applications/items



1. Click the Select Apps button on the top of the Device Profiles screen.
2. Check the boxes to the items you want added to the profile. The Check Dependencies box at the bottom of the screen can also be used to include icons, images, etc that are used in the application.
3. Click OK.
4. A list of your selected items will display in the right panel "Items Selected for Device Operations".

To set the options for a Profile

For Profile settings, see the [Profile Option Descriptions](#) topic.

To see the profile as it would appear on the client, see the [Client Configuration Settings](#) topic in the Client Installation Guide.

To deploy/install the profile to a client

- If deploying to a Windows CE/Mobile device or Windows desktop client, see [Solution Deployment](#), and [Windows Desktop Client Install Guide](#) or [Windows CE/Mobile Install Guide](#).
- If deploying to an Android client, see the [Android Client Install Guide](#).
- If deploying to an iOS client, see the [iOS Client Install Guide](#).

To create a Mobile Profile

1. Right-click on **Solution Explorer > Profiles** and select **Add New Profiles** from the menu. If desired, you can also create a new folder to group your profiles by selecting **Add New Group** from the menu.
2. Enter an **Device Profile** name and **Description**. You cannot have any spaces in the Device Profile name.
3. Set which platform you want this profile to deploy to from the **Deploy Applications** list.

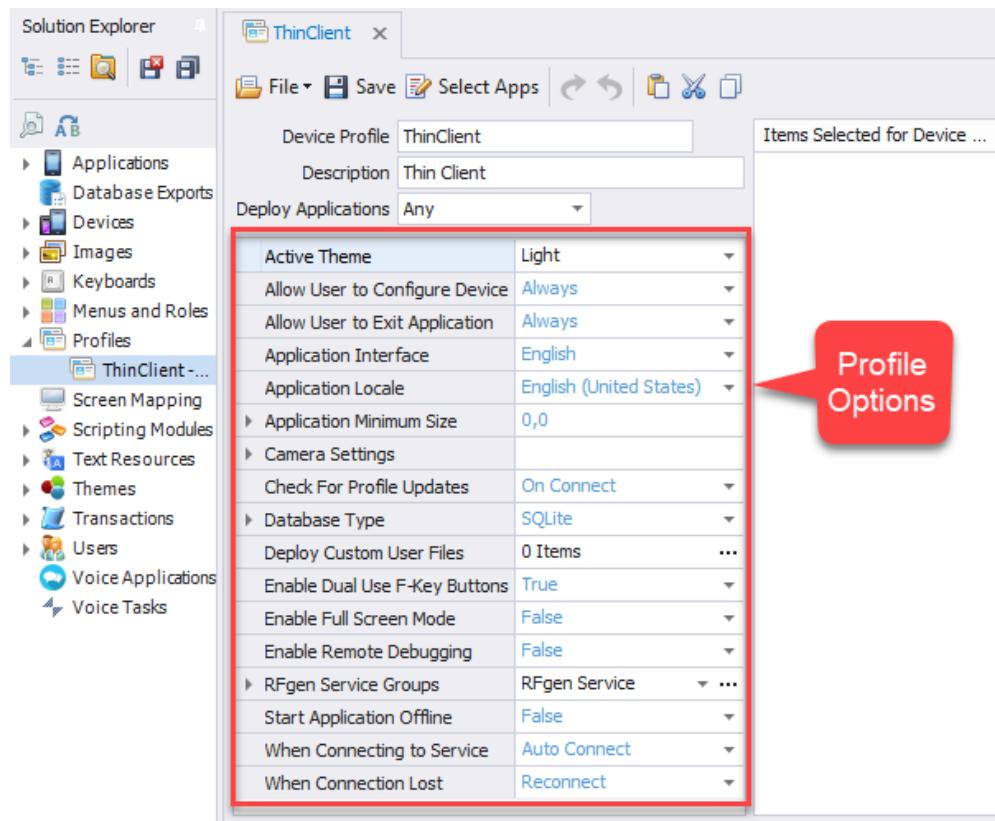
"Any" will create a profile that can apply to devices that have Android, iOS, Windows, or WindowsCE (Window Mobile) operating system (OS). Or, you can select the specific operating system which will include characteristics that are supported only on the OS selected.

4. Select the values in the list. For descriptions, see the [Profile Option Descriptions](#) topic.

**If you want to create a Batch or Fat client profile that enables your client to run applications while off line, make sure "Start Application Offline" is set to True, and set "When Connection is Lost" to "Go Offline." You should also setup a database to be installed on the client so data can be stored on the device while its offline in "Database Type."

Android, iOS, and Windows profiles are deployed to the RFgen client once a connection has been established between the server and client. For Windows CE, refer to [Solution Deployment](#) which is used to build CAB files for transferring a profile to a Windows CE device.

Profile Options



A profile is a collection of settings which dictate how the client will connect to the server, which applications are used, whether the client can process transactions while offline , and other maintenance settings). You select the values for each feature/option below so that when the client connects, it will adopt the settings. Note that the default is to enable the client to process transactions when its connected to the server; If you want your client to function as a "mini RFgen Server" (Batch or Fat Client), you'll need to change the "Start Application Offline" and optionally, setup the mini-database that can be used to hold processed data on the device until it can be uploaded upon reconnection.

At the top of the window are these categories: **Device Profile** (profile object name), **Description** (description of the profile), and **Deploy Application** (menu of selectable platform options). The name of the device profile does not support spaces but does allow alpha and numeric characters. The description allows spaces for a longer description of the object. In Deploy Applications, "All" is the default for all client platforms or you can select a specific platform. Some of the settings below are specific to the platform.

Active Theme - The mobile theme that is applied to the application(s) when the user is connected or offline.

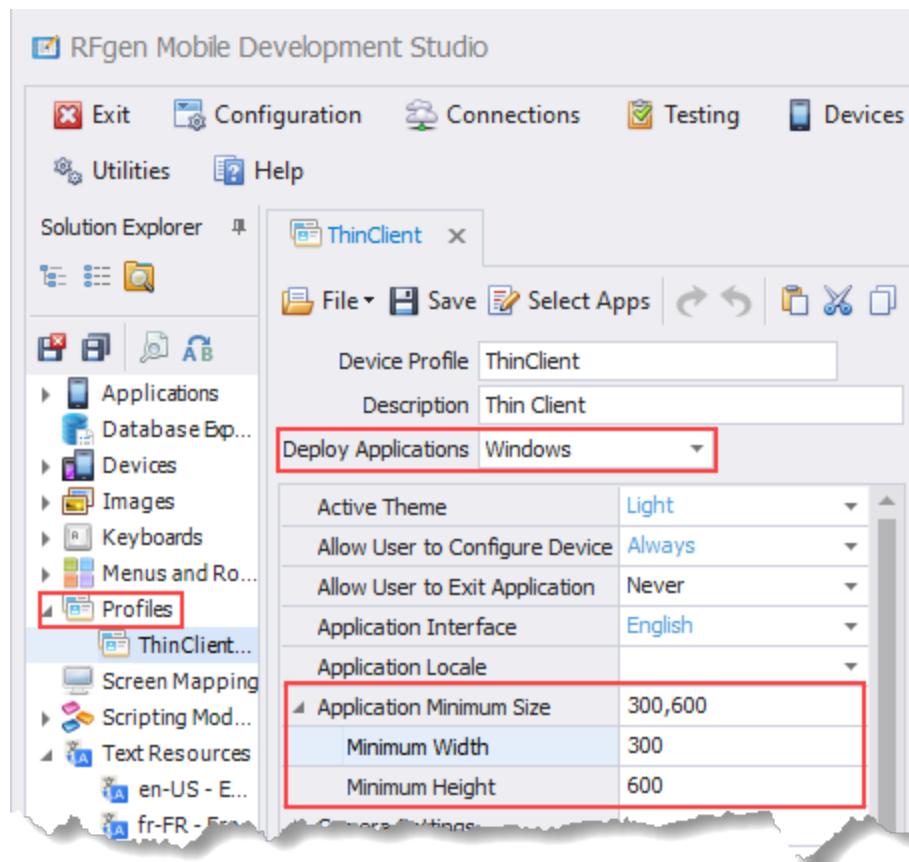
Allow User to Configure Device - Always, Never or With Password. The *Always* and *With Password* values will allow the user to open the Client Configuration once the user enters a password after they launch the RFgen Client Configurator. To set the password, click on the drop-down list in front of the Allow User to Configure Devices. (This may take two clicks to see; first to set the *Always* or *With Password* value, and then you go back up and look for the drop down and click the drop down in front of the option to see Supervisor Password. If "Never" is set, when the user attempts to open the RFgen Client Configurator, the user will see a message explaining he/she is locked out.

Allow User to Exit Application - Always, Never or With Password. The *With Password* for a user is set under Solution Explorer > Users. If this is for users with a password that is set outside of RFgen (such as a list controlled in Active Directory or from an ERP then the "*With Password*" does not apply.

Application Interface - The RFgen application is capable of supporting the following user interfaces in these languages: Arabic, Chinese, English, French, Japanese, or Spanish. If you created text strings that translated some of the application values (i.e Plant, Bin, Asset), use the Application Locale setting.

Application Locale - Select the country locale for the target profile. This assumes you created a list of translated text strings that can be transported with your Profile.

Application Minimum Size - Available for Windows Desktop client only. This setting is not available for All, Android, iOS, or Windows CE.



This allows you to set the minimum width and height of the application screen when its running on a Windows Desktop client. It can be used to restrict the user from resizing the screen so its too small to find on the computer. This value is in pixels; other metrics are not supported.

Camera Settings: Auto Process Barcodes - *True* will append <Enter> (the Enter or Return key) as a post-amble when the user scans a barcode. The post-amble moves the cursor automatically to the next field. *False* will not append a post-amble, and the cursor will remain in the same location until the user taps the Return/ Enter key.

Camera Settings: Camera Modes - The modes sets whether one or two post-ambles are used. If **Auto Process Barcodes** is set to True with **Native Wedge** RFgen will append two Return characters to a scanned transaction. If **Auto Process Barcodes** is set to True with **Standard** RFgen will append one Return character to a scanned transaction. If Auto Process Barcodes is set to False, no values are appended.

Camera Settings: Maximum Image Size - This is the maximum size an image will be for the Device.TakePicture command in pixels. Since newer devices tend to capture high quality images (and larger file sizes) by default, use this option to limit the image size so it transfers faster to the server.

Check for Profile Updates - This sets the method and how often the client will check the server for a Profile and compare if there are differences. If differences exist, the client profile would be updated. This can be set to *Manual* (when the user requests it), *On Connect* (only checks for profile updates when the client connects to the server), or *Daily* (checks are performed even if the client is never disconnected from the server).

Database Type - When the client is set to process data/transactions off-line, the database that is selected from the menu will be included in the Profile. If the client is to process transactions only when its connected (in a session with the RFgen server), then select *None*.

Deploy Custom User Files - If there are special files that you want installed to a client, you can add a file and specify its installation (as a copy from the source) and set installation location.

Database Type: Storage Location, Storage Path, Storage Name - These fields are enabled if a database type is selected and are used to define the location, storage path, and storage name of the database stored on the client.

Enable Dual Use F-Key Buttons - If your solution included F-Keys, set this value to True. If you only want the F-keys on the client/device to be used, then set this value to False.

Enable Full Screen Mode - True will use size the application to fill the display screen space on a device. False will not.

Enable Remote Debugging - True allows the client to be accessed for debug and trouble-shooting purposes. False prevents remote access.

RFgen Service Groups - Is used to set the RFgen Server information. Click the ... to open the **Manage Enterprise Service Groups** screen. **Service Description** is the unique name of the RFgen Server Service. The **Instance Name/Address** is the Windows Server name or IP address. If the RFgen servers are setup for load-balancing, the client will attempt connection with the next live server in the group if the server its initially connects with fails.

Start Application Offline - True enables your client to start up an application when the client is NOT connected (hosting a live session) with the RFgen server. This enables your client to operate as a "Mini-RFgen" server (also called a Batch or Fat Client). False will only enable the client to run an application while its connected to the server (Thin Client).

When Connecting to Service - *AutoConnect* will connect the client to the server that is listed in the RFgen Services group. If however, you have multiple RFgen Servers setup, you can enable the user on the client to choose which server he/she connects with if you select the *Select Service* option. (See details on the RFgen Services Group.)

When Connection is Lost - If the Profile is setup for processing data/transactions when connected to the RFgen server, select *Reconnect*. If the client is to be used off line (as a Mobile, Batch, or Fat Client), then select *Go Offline*.

Related Information

For basic information on setting up and installing a client, refer to the appropriate install guide for your client platform:

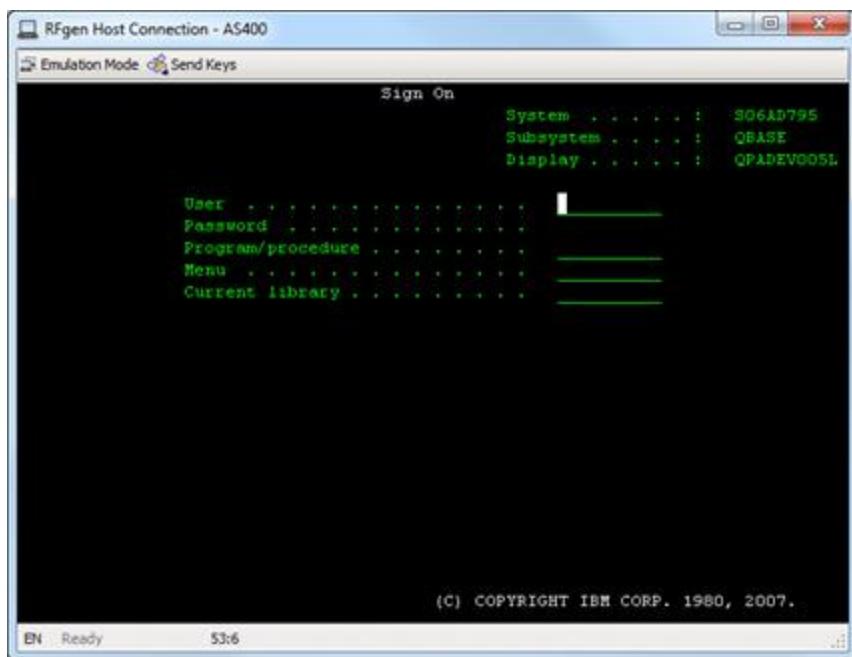
[Android Client Install Guide](#)

[iOS Client Install Guide](#)

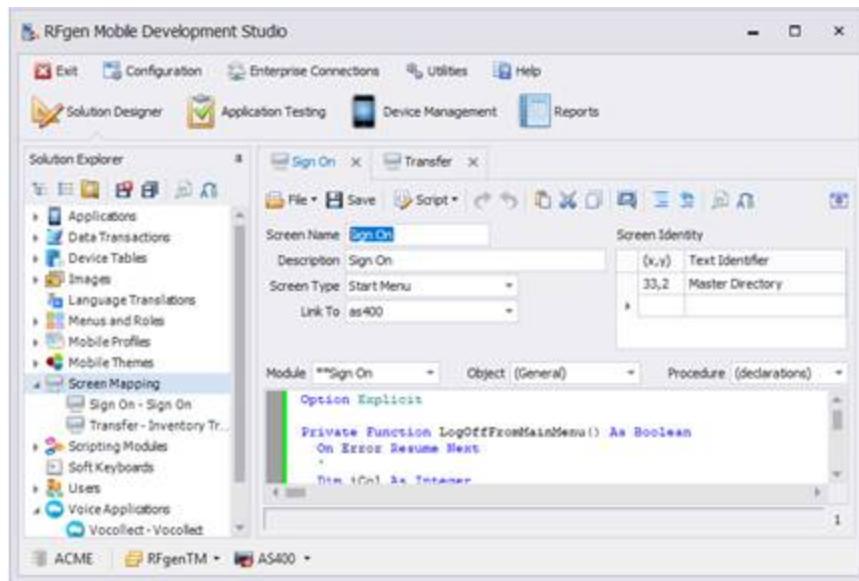
[Windows CE/Mobile Install Guide](#)

[Windows Desktop Client Install Guide](#)

Screen Mapping

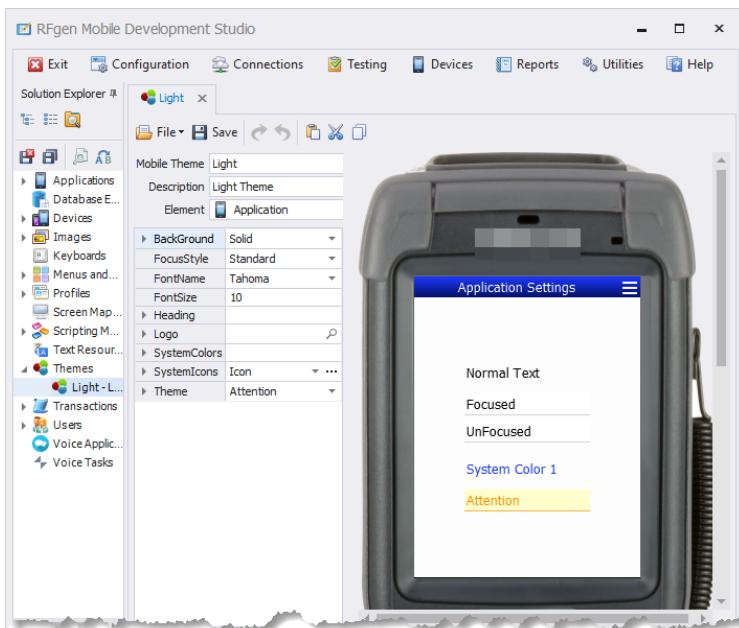
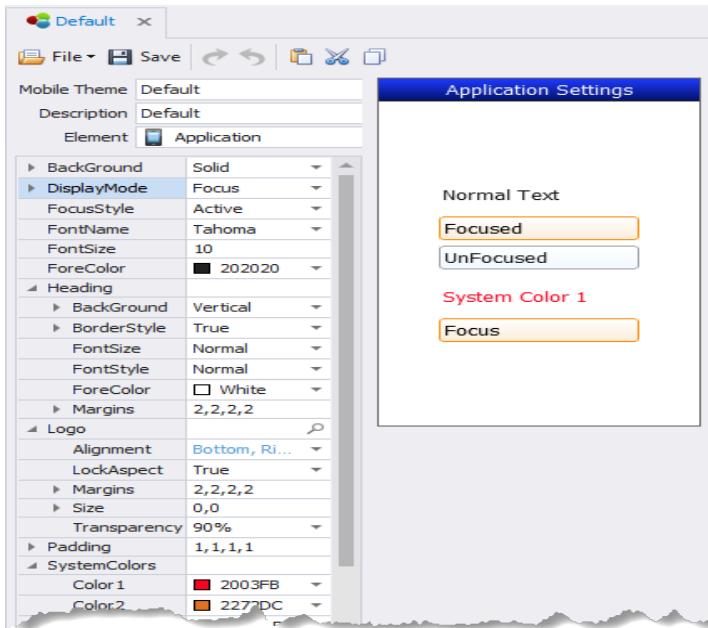


Screen Mapping is a technology used with legacy systems to enable execution of tasks on green-screen consoles. RFgen provides modules that can be used to create the instructions that appear on the console.



For more information on setup and support of screen mapping systems, refer to [Appendix B - Screen Mapping](#).

Themes Overview



The **Theme** (also called "Mobile Themes") is a collection of control property values and event/focus settings that are used to create a common color theme for an application and common look and feel when a user interacts with the application. For example, you can create a theme that uses your company's colors and company's logo, and highlights a selection if bright yellow. Or, you can have another theme that uses colors which work best in low light conditions and also provides a WARNING state color themes.

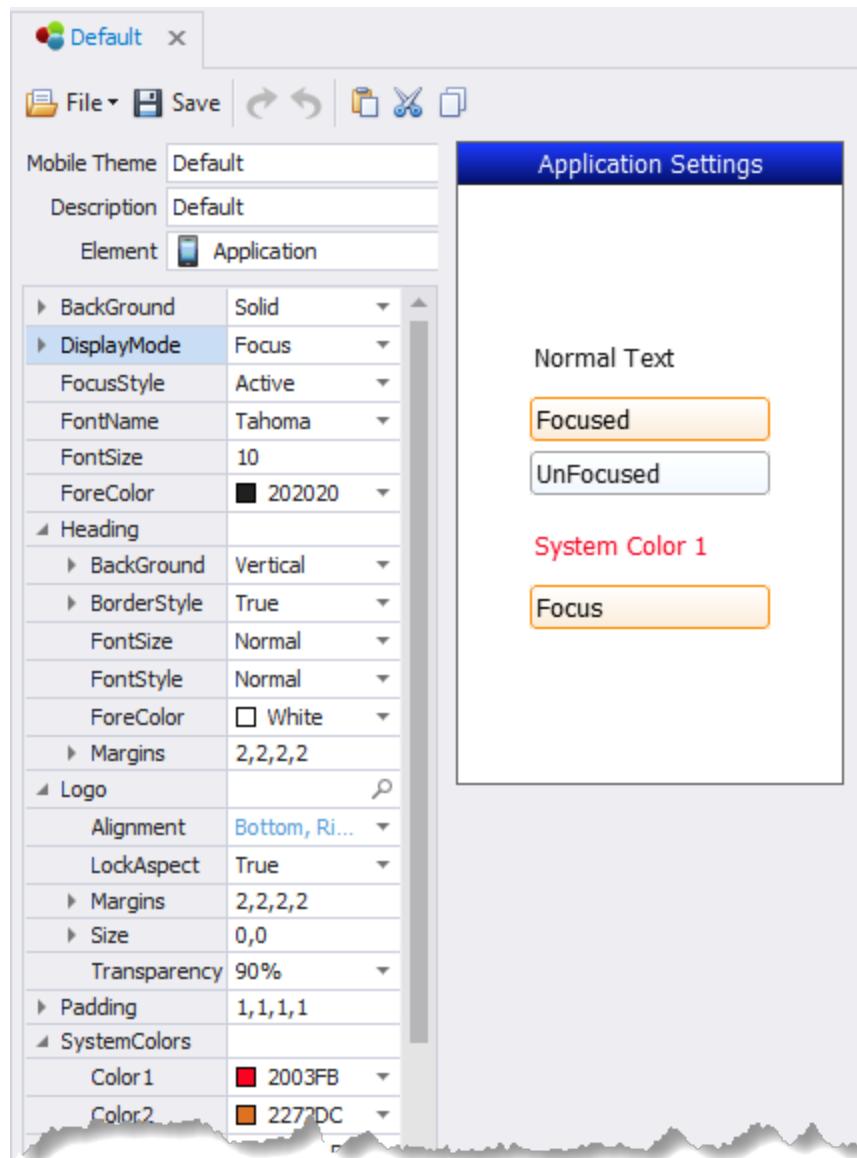
Themes are structured by **Element** (Application, Button Control, Label Control etc) and serves as the parent for the corresponding element in the **Solution Explorer > Applications** designer. For example, if you created a theme called "Light Grey" and set your Button Control's BackGround property to light gray, then all buttons added to a form will automatically have a gray background.

- To add, remove, rename, copy or recover a theme, use the Mobile Themes Right-Click Menu.
- To view a theme element in a device, see [To Enable Device Emulation](#).

Related Topics:

- For Application property definitions see [Theme Applications](#).
- For property definitions on a graphical control, see [Graphical Control Properties](#).
- For element descriptions and details on element properties, see [Theme Elements](#).

Themes: Application Properties



Global Properties Versus Configuration Settings

The Application element contains parent and system-wide configuration settings.

How do changes affect existing applications in the Application Designer?

- Values set in Mobile Themes apply to newly created child items in the Application designer.
- Defaulted values "(Default)" will be updated.
- Locally set values are not changed.

BackGround Property - has four categories for setting the background color: *Solid*, *directional*, *Transparent*, and *None*. **Solid** applies BackColor and ignores BackColorEx. The **Directional** (Diagonal Right, Diagonal Left, Vertical etc.) applies BackColor and BackColorEx in a gradient direction. **Transparent** allows the background color from next closest parent or container behind the control to show through. **None** is similar to transparent as it ignores both BackColor and BackColorEx. Most controls will also have a **(Default)** value that appears in [colored text](#). This value uses the Theme > Application > BackGround settings.

For details on **BackColor** and **BackColorEx**, see [Graphical Properties Definitions](#).

DisplayMode - While all the other properties are used to set the normal look/feel for all applications, the DisplayMode is only used to stylize unique application states. Each property group includes an **Override** property which only applies the values at the current prompt if set to True. For example, if the Focus BackGround = (Default), and the Override = False, then the BackColor values will use the values from the Application > BackGround and not use the BackGround values set for Focus. For more detail, see [Theme States](#).

FocusStyle - sets how all editable prompts, for all applications display at runtime. The FocusStyles are Standard, Active Border, or Visible with Focus. When set to *Active*, this property only applies to edit controls (i.e. ComboBox, SpinEdit, TextBox). *Standard* displays all graphical controls, but only the control with focus has the I bar or prompt. *Active Border* is similar to Standard except it color-highlight the control's border when it has focus. *Visible with Focus* is a progressive display where the control (i.e. its borders) are hidden until it has focus. It retains scanned data/entries, and hides the graphical control as the user clicks though to the next prompt.

FontName - the fontfamily to be used. This is the ONLY location where you select the font type. The selection here is global to all the other controls.

FontSize Property – (for graphical mode only) defaults to a baseline value called "Normal" which is set in **Mobile Themes > Application > FontSize**. You increase or decrease the font size (in points) if you want to override the default value. For example, if your Normal = 11 points but you want your Labels > Caption text = 14 points, you would set your Labels > Caption > FontSize to +3. In controls where the FontSize property is nested into subproperties, you may see "Inherit" used instead of "Normal"

ForeColor Property sets the font color for all controls when its set from Mobile Themes > Applications. For state based coloring of fonts (ie. Error messages) see Themes.

Heading Property Group - sets the look and feel for the form's heading/heading and the positioning of text and icons (icon buttons) in the heading. Note that the form heading is not the same as Status Bar which can also display at the top of a form. Its nested properties include: BackGround, BorderStyle, FontSize, FontStyle, ForeColor, and Margins. For property descriptions, see [Graphical Control Properties](#).

Logo - This image can be added anywhere on the form, form header, or used as a background "watermark" on the form. To add the logo (i.e. company's logo), first presize the logo outside of RFgen, upload it to the Images node, then assign the Logo filename to the Logo ImageName.

Logo Alignment - will position the logo relative to the Form Display area or if the Logo is used .

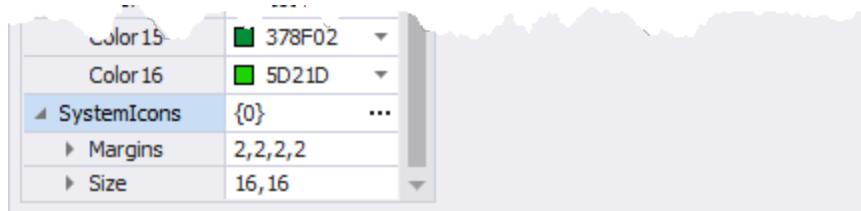
Logo LockAspect - True will

Logo Margins - is the distance in pixels between the Logo image and parent container (i.e. Form). Values are Bottom, Left, Right, and Top.

Logo Size - is the Height and Width of the logo image in pixels.

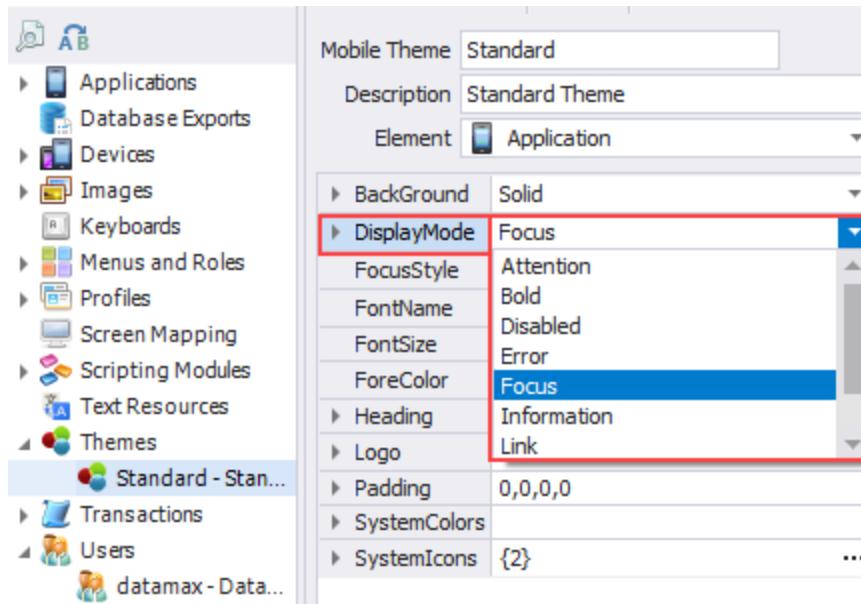
Logo Transparency - sets the percentage of transparency where 100% is entirely transparent.

SystemColors - This is a system-wide function which is used to setup color themes that can be accessed by other properties within Mobile Themes and from the Application Designer. For information on using this feature, see [Mobile Themes System Colors](#).



SystemIcons - are used to associate a system action (i.e. Call Event, Call Form) with the icon, which is placed in the Application header. The alignment and margins for these icons is performed in the Heading > Icons property. For more details see [Manage Icons Collection](#).

Application Theme States

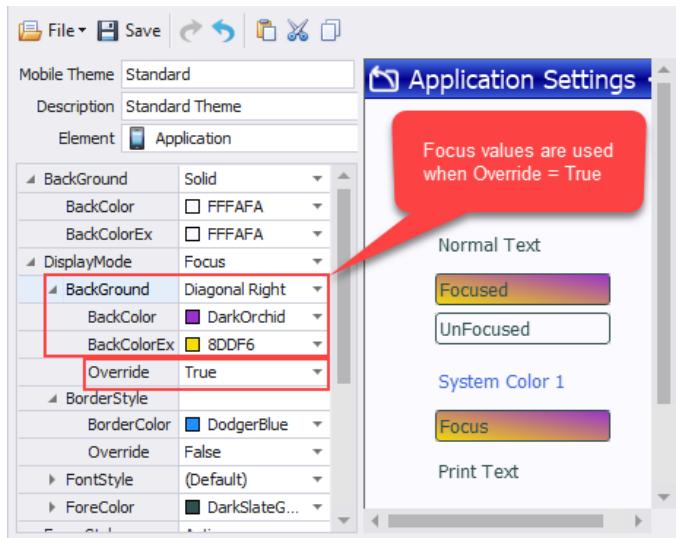


While all the other properties in the Application group are used to set the normal look/feel for all applications, the **DisplayMode** is used to stylize unique application states.

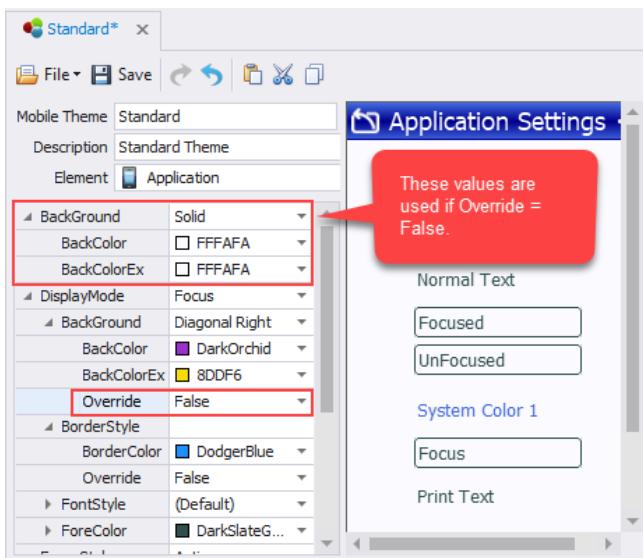
This enables you to have a theme for displays that need the user's attention, or when an error occurs, or you want to convey the severity of a condition or problem. This group also includes the appearance of prompts with/without focus. The built-in application states are: **Attention, Bold, Disabled, Error, Focus, Information Link, Success, or Warning.**

The **DisplayMode BackGround, BorderStyle, FontStyle, and ForeColor** properties include an **Override** property. This flag determines whether these theme values will override the current prompt's setting.

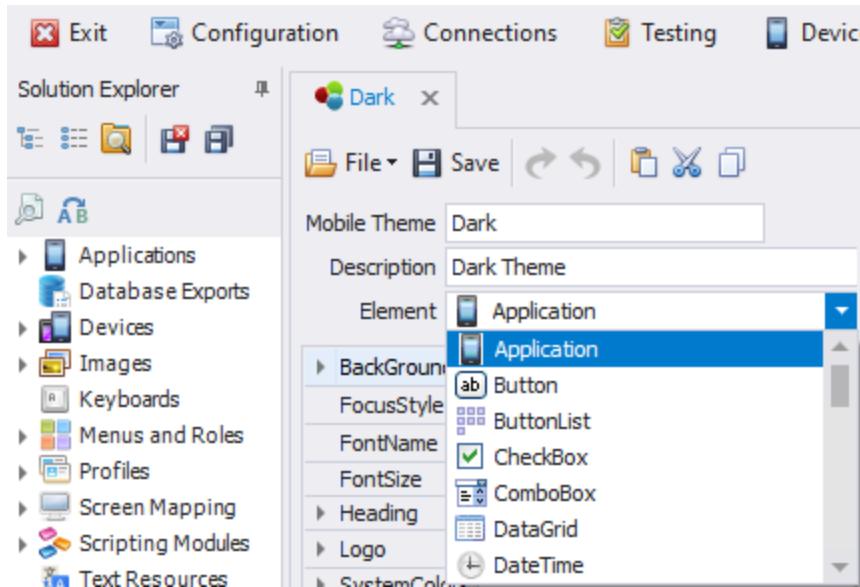
- If **Override = True**, the DisplayMode style is applied to the current prompt, even if the style has a (Default) value.



- If **Override = False**, the DisplayMode style is NOT applied, and the style from Themes > Application are applied.



Theme Element Descriptions



The available elements are: [Application](#), [Button](#), [ButtonList](#), [CheckBox](#), [ComboBox](#), [DataGridView](#), [DateTime](#), [DesktopIcons](#), [Dialog](#), [Frame](#), [Image](#), [ImageList](#), [KeyBoard](#), [Label](#), [ListBox](#), [Map](#), [Memo](#), [Menu](#), [Page](#), [Panel](#), [PanelList](#), [RadioButton](#), [ScrollBar](#), [Search](#), [SideBar](#), [Signature](#), [SpinEdit](#), [TabControl](#), [TextBox](#), and [TreeView](#).

If the control property in the Solution Designer > Application > Control Panel is set to "(Default)", the value is adopted from the styles and values you've set here. You can override the individual properties of the control once the control is added to the Form.

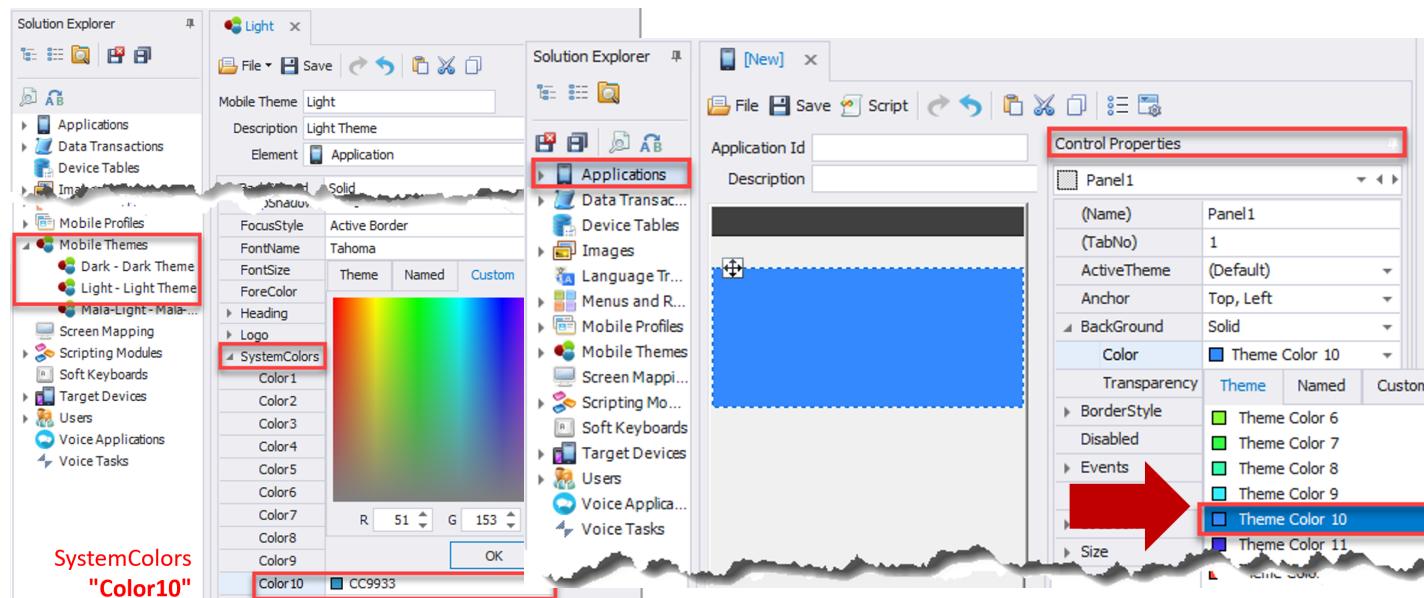
To customize a theme for an element (graphical control), select the desired control and change its properties.

If you have an open source database configured, RFgen provides pre-configured themes that you can modify for each element.

For property descriptions, see [Graphical Control Properties](#).

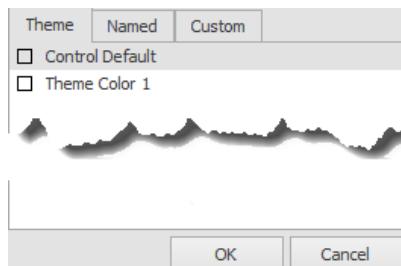
Themes: Application Colors

In Themes, you can customize up to 16 theme colors (called SystemColors) which can then be assigned to other Control Properties in the Application designer.



Example of a customized color, saved as Theme "Color 10" in the Mobile Themes Application element.

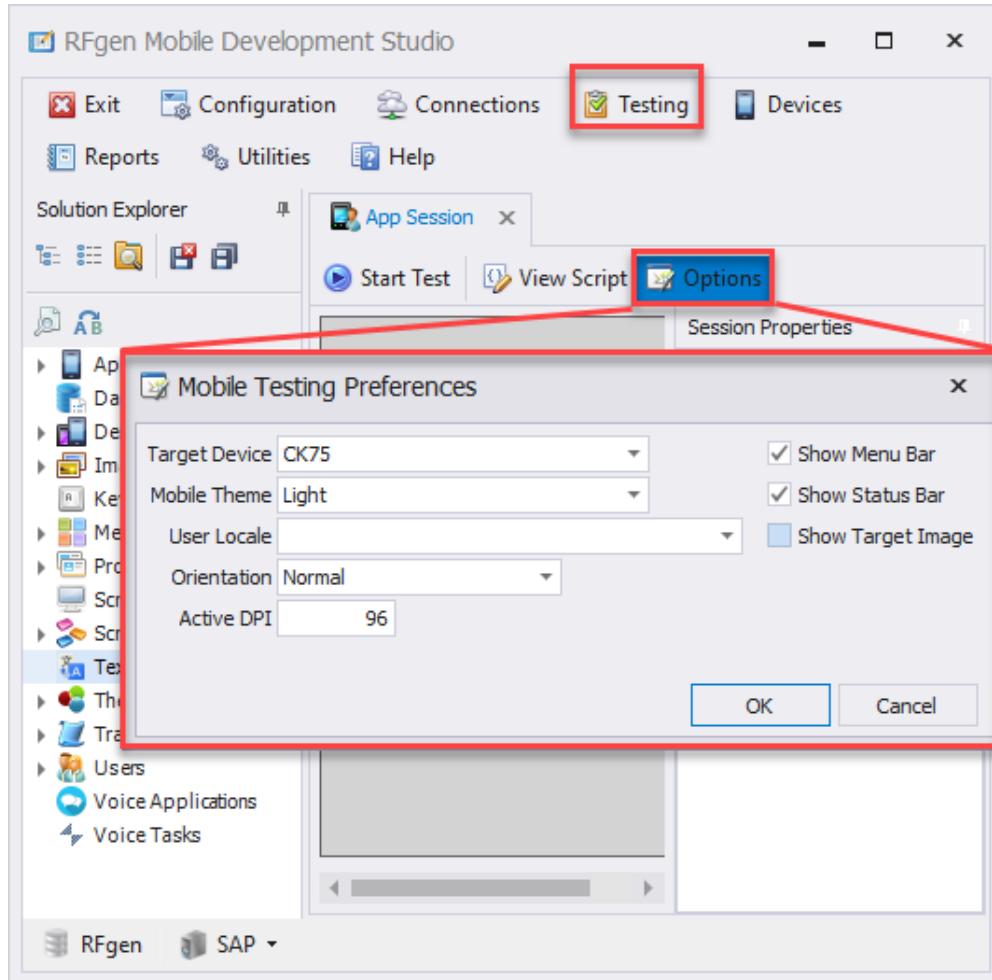
To customize a theme color, select a Color palette from Mobile Themes > Element = Application > System Colors property, then set the RGB values under the Custom tab, or select a color from the Named tab. Use the Theme tab to set whether this property is accessed as a Control Default, or as a unique Theme Color from the Solution Explorer > Applications Designer.



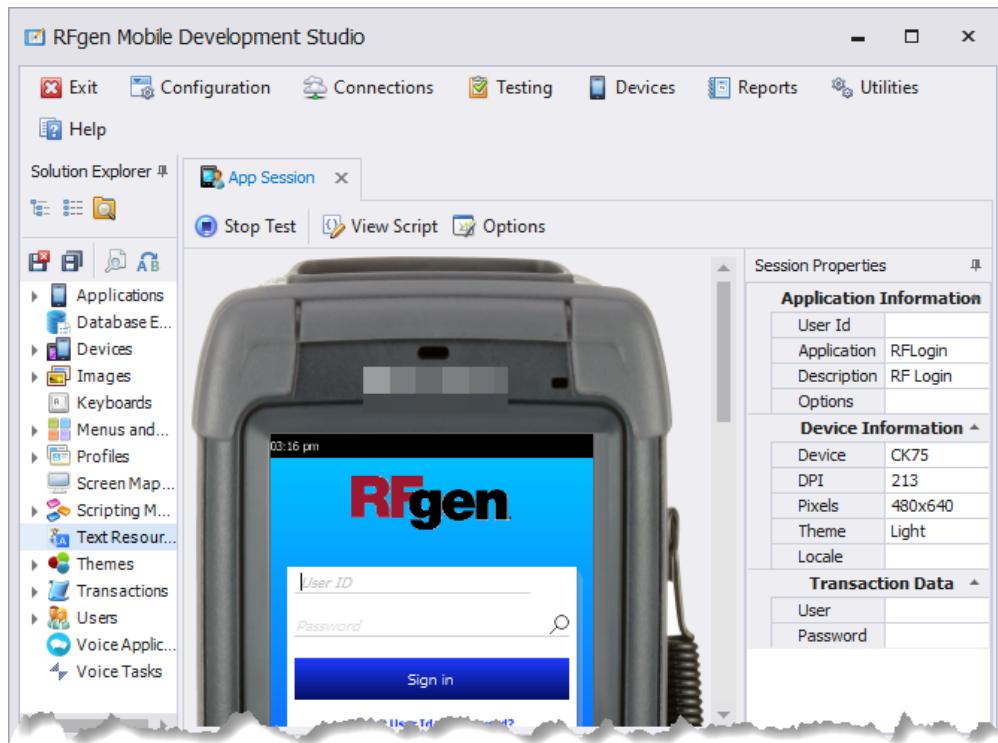
To Enable Device Emulation

To view your theme elements, menu, or application (via the Test screen) in the skin of your target device, go to:

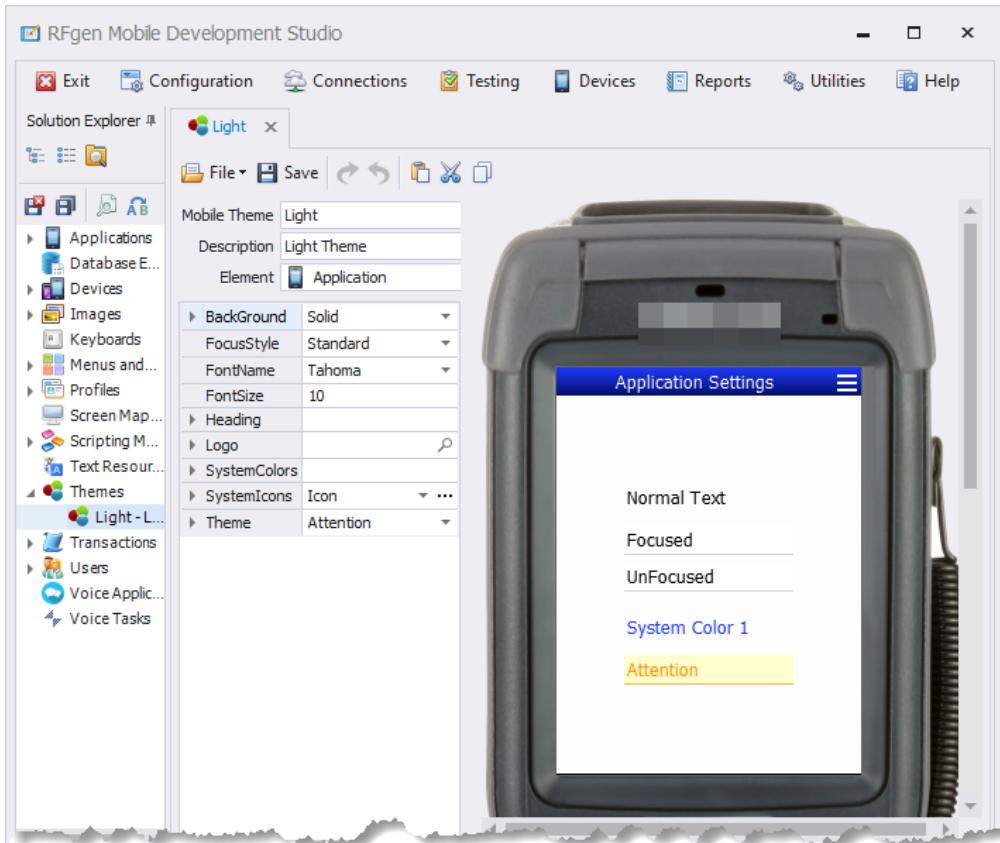
1. **Testing > Mobile Apps > Options Menu** icon.
2. Checkmark the **Show Target Image** box.
3. Click **OK**. This will enable the emulation of the selected Target device in Themes and Menus and Roles and when you run your application in Application Testing.



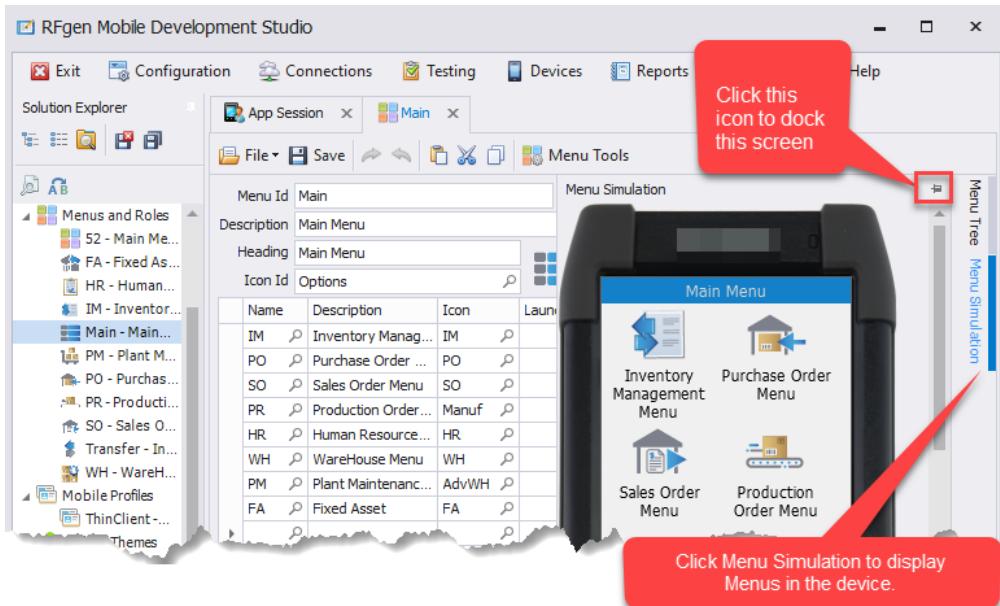
Application in the Device Emulator while Testing example:



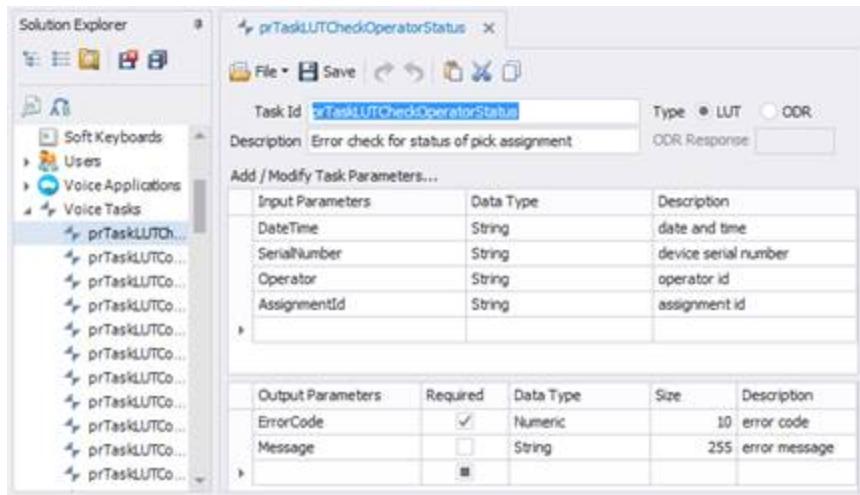
Theme Element in the Device Emulator example:



Menu in the Device Emulator example.



Voice Tasks



Vocollect Tasks are pre-defined scripts that execute on Vocollect supported platforms and interact with RFgen to provide the voice solution together with the backend data connection solution.

RFgen provides pre-entered tasks which can be added to your Vocollect Tasks Tree by importing them from the Utilities / Import Mobile Applications menu. See below for details.

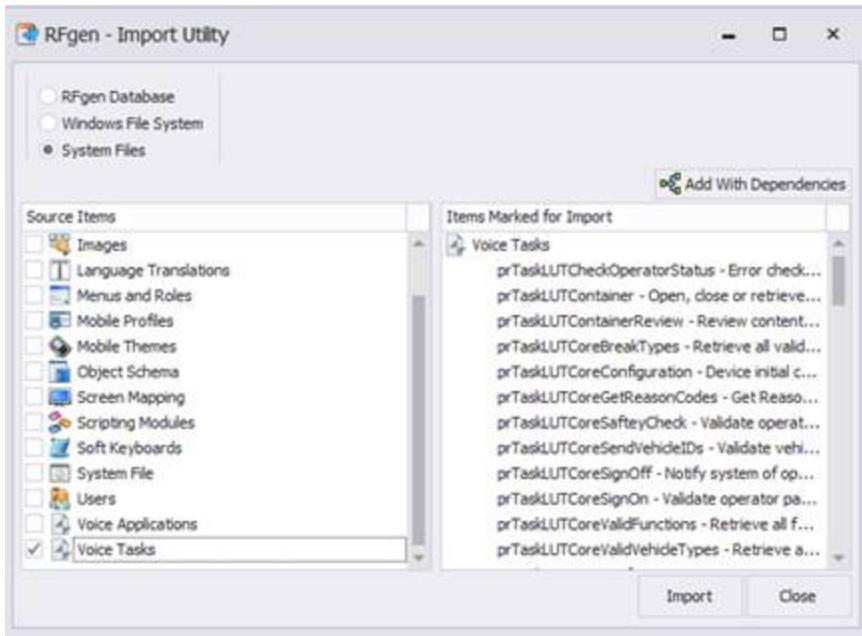
The **Task ID** is RFgen's identifier but in the case of the pre-defined tasks Vocollect provided the names.

The **Description** clarifies the task and displays in the tree.

The **Type** is either LUT or ODR. LUT is two-way communication between the Vocollect device and RFgen and ODR represents one-way communication. For ODR, there are no output parameters. The **ODR Response** field can contain any value and it represents an acknowledgement bit from RFgen to make the Vocollect device stop polling RFgen for a response. Types are input only and are used for tasks such as updating a count or updating a status. The **Project Group** lets the user group like items together in the navigation tree.

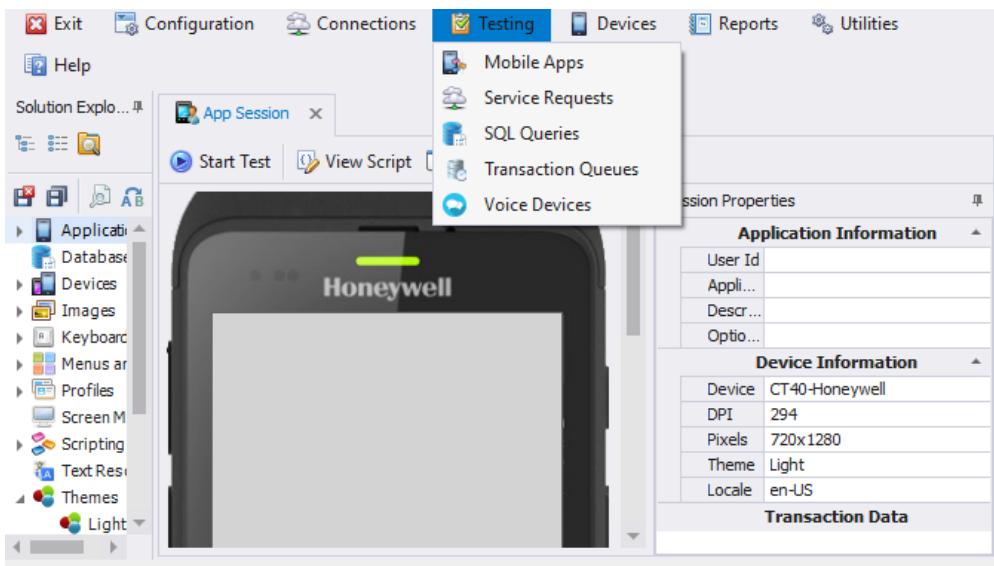
The **Add/Modify Task Parameters** are defined by Vocollect and are configured here.

To Import Voice Tasks



1. Click on **Utilities > Import Mobile Applications**.
2. Click on System Files (at the top).
3. In the Source Items pane, scroll to the bottom and select Voice Tasks.
4. In the Items Marked for Import (right pane) select the tasks to be imported.
5. Click on Import. A message on the number of items imported displays.

Testing Overview



The **Testing** Menu in the Mobile Development Studio provides:

- runtime sessions for **Mobile Applications** so you can see how the application screen looks in a device at runtime, view and debug code or objects (i.e. applications, screen maps, services).
- a means of testing **Service Requests** for integrations with cloud based solutions.
- a interface to execute **SQL Queries**, query results to a file via Excel, and enable or disable your ERP Connections when executing a SQL query.
- runtime sessions for testing **Transaction Queues** or **Transaction Events** so you can check Cycle Time and Iterations.
- data steam portions of other third-party data collection solution that are voice driven and use unique hardware and software for the testing of **Voice Devices**.

Connection Limitation

When you test device connections with the Mobile Development Studio, the Mobile Development Studio limits the number of simultaneous connections between itself and remote devices. For example, you can only have a total of 3 connected sessions.

Stop Test - This allows you to start and stop your test sessions.

View Script - For information on testing your script see [To Debug Code](#).

Options - For information on this feature, see [Testing Options menu](#).

Mobile Apps: Session Property Descriptions

Application Information

User ID - Displays the user who is logged in during the test session. For example if you had a user named Sam setup in Users and logged in as SAM, "Sam" would display here.

Application - Tells you which application you are testing.

Description - The description that is associated with the Application you are testing.

Options -

Device Information

These settings can be changed using the Options menu.

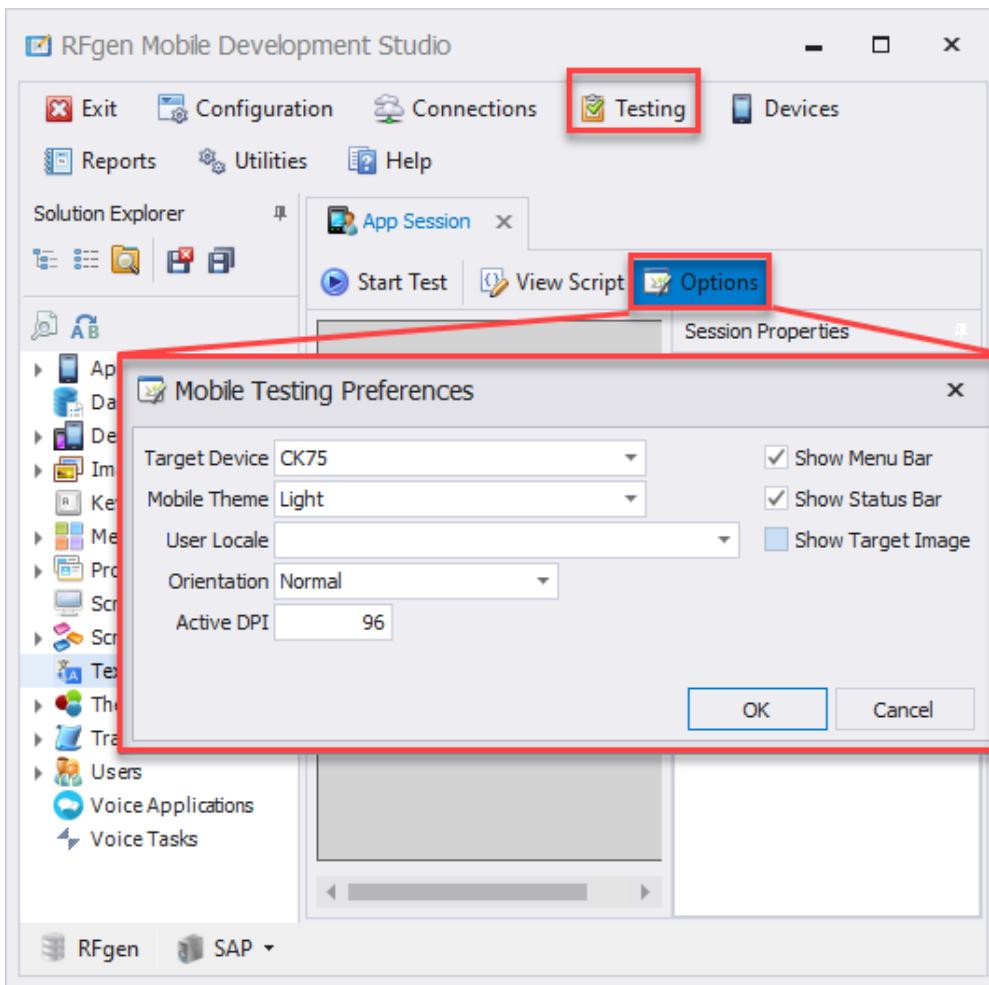
Device - Which target device image is in use.

DPI - Tells you which values are used to scale the device image.

Theme - Tells you which Mobile Theme is used for the application's look/feel.

Locale - Tells you which language is used if you have your application setup for different locales.

To Set Your Testing Preferences



The Mobile Testing Preferences window displays when you click on the **Testing > Mobile App > Options** icon.

When testing, the values set in your Configuration > Environment Settings and Configuration > Application Preferences are used. However, if you want to change the device skin, theme, locale (if application was designed for localization), and orientation, you can change these features through the Options screen.

Use the **Options** screen to change:

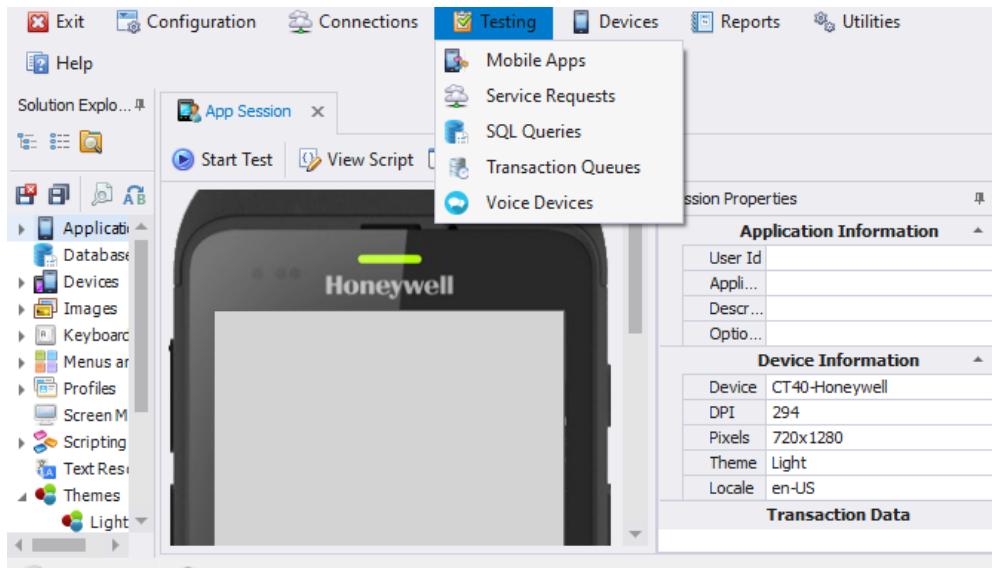
- Target Device - sets the device type (device image).
- Mobile Theme - sets which theme is used; otherwise the theme from Configuration > Application Preferences > Theme is used.
- User Local - sets the locale if the application strings were translated for multiple languages. The default is US English.
- Orientation of the device screen - Normal versus Rotated.
- Active DPI - controls the scaling of the image. 96 dpi is a standard for smaller devices. The value depends on the resolution of the target device.

Show Menu Bar - If checked, this displays the menu bar of the application so you can test your function calls from the menu bar.

Show Status Bar - If checked, this displays the status bar of the application (i.e. device battery status, device wireless connection status). Will show how much space is used but statuses will not be testable.

Show Target Image - If checked, application displays in the target device. If unchecked, the application displays in a screen (no device).

Testing Mobile AppsOverview



The **Testing** Menu in the Mobile Development Studio provides:

- runtime sessions for **Mobile Applications** so you can see how the application screen looks in a device at runtime, view and debug code or objects (i.e. applications, screen maps, services).
- a means of testing **Service Requests** for integrations with cloud based solutions.
- runtime sessions for testing **Transaction Queues** or **Transaction Events** so you can check Cycle Time and Iterations.
- data steam portions of other third-party data collection solution that are voice driven and use unique hardware and software for the testing of **Voice Devices**.

Connection Limitation

When you test device connections with the Mobile Development Studio, the Mobile Development Studio limits the number of simultaneous connections between itself and remote devices. For example, you can only have a total of 3 connected sessions.

Stop Test - This allows you to start and stop your test sessions.

View Script - For information on testing your script see [To Debug Code](#).

Options - For information on this feature, see [Testing Options menu](#).

Mobile Apps: Session Property Descriptions

Application Information

User ID - Displays the user who is logged in during the test session. For example if you had a user named Sam setup in Users and logged in as SAM, "Sam" would display here.

Application - Tells you which application you are testing.

Description - The description that is associated with the Application you are testing.

Options -

Device Information

These settings can be changed using the Options menu.

Device - Which target device image is in use.

DPI - Tells you which values are used to scale the device image.

Theme - Tells you which Mobile Theme is used for the application's look/feel.

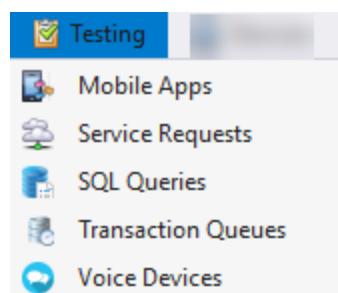
Locale - Tells you which language is used if you have your application setup for different locales.

Service Requests Testing

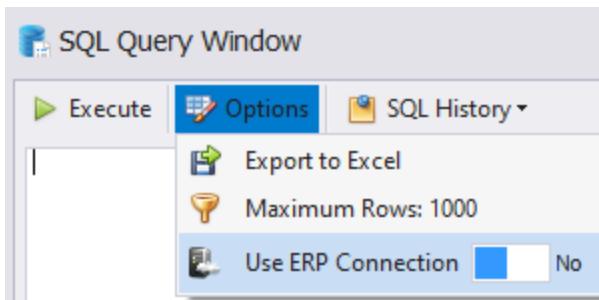
To test RFgen Services between your RFgen server and clients, you'll first need to install and configure an application running in character mode or graphical mode.

Mobile Development Studio, by itself, allows one usable network device to log in, for test purposes, without requiring a software license. Multiple device tests require using the Server that can allow over 1000 concurrent users for a period of two hours until the service is stopped and restarted. An RFgen Software license is required to permanently activate your RFgen network (by means of the RFgen 'Mobile Enterprise Service Management' icon that then appears on your Windows system tray).

SQL Query Testing



To access this function click on **Testing > SQL Queries**. The **SQL Query Window** displays.



The **Execute** option tests SQL statements so you can see results before executing the statement in code. The same screen is also used to preserve and recall the statements from **SQL History**. This utility can also be used to undo updates, check results, delete values or even adding and dropping tables. Any SQL command entered here is submitted to the ODBC driver for execution. There are no limitations by RFgen as to what can be submitted. For more information, see SQL Query Tips below.

The **Options > Export to Excel** utility will prompt for a location to save an Excel file. Microsoft Excel does not need to be installed on the system for this function to work.

The **Options > Maximum Rows: 1000** utility allows you to change the 1000 maximum row default to a different row value.

The **Options > Use ERP Connection** function allows you to enable a connection to a specific ERP which simplifies the process of executing a SQL statement against an ERP connection such as SAP, Oracle JDE, Oracle EBS, Deltek CostPoint, or Microsoft Dynamics. You slide the box over to enable or disable the connection.

To test the query, enable the ERP mode (slide box to the right), enter your statement in the query box and press Execute. The execution will display results in the second box.

SQL Query Tips

The SQL Query Window gives a current snapshot of the data in your database. As transactions are applied against your data, you will need to re-execute your SQL statement.

The screenshot shows the RFgen - SQL Query Window interface. At the top, there are three tabs: Execute, Options, and SQL History. The SQL History tab is currently selected. Below the tabs, a multi-line text box contains the SQL query: "Select * from T001L;". The main area displays a table with the following data:

	MANDT	WERKS	LGORT	LGOBE	SPART	XLONG	XBUFX
▶	800	0001	0001	Lake View			
	800	0001	0002	Main Store Room			
	800	0001	0088	Lager 0088 (WM)			
	800	0002	0001	Lager 0001			
	800	0002	0004	Lager 0004 (WM)			
	800	0005	0001	Kit Kat Store Rm			
	800	0005	0002	Craison Store Rm			

Below the table, a message indicates: "processing time = 2.460 seconds, 500 records returned..."

If the intent is "select * from TableName", then only specifying the table name will default to the "select * from" when executed.

The multi-line text box allows the entry of several SQL statements. In this case, highlight the intended SQL statement and click Execute from the menu.

Multiple SQL statements must be separated by semi-colons ";". For example:

```
Select * from items;
Select * from itemmaster;
```

These will be considered two different SQL statements. If no text is highlighted, then it will look at the current insertion point to determine which SQL statement to execute based upon semi-colon delimiters. Further, if you click on Options menu / Display Query History – then double-click on an item, it will append it to the SQL window instead of replacing the existing contents.

The **SQL History** option allows you to select from any of the previously executed SQL commands and to also limit the output to a maximum number of records.

Transaction Queue Testing

This option will let the user test Timed Event macros and queue processing.

The **Item Type** option switches between queue processing and timed Events. If Transaction Queue is selected, the **Item Name** specifies which queue is to be processed. If Transaction Event is selected, then the Item

Name list is populated with the configured events under the Transaction Management / Processing Events option.

Run on Server option is used to specify which server should be used in the event you have multiple servers for load balancing. If this field is left blank, RFgen will bypass this field and process transaction queues on all systems connected to it. But if you have the server IP address or server name, RFgen will process transactions with this the server.

The **Cycle Time** is an interval in seconds that is how often the queue will be checked for new transactions. For events this is how often RFgen waits between each execution of the event.

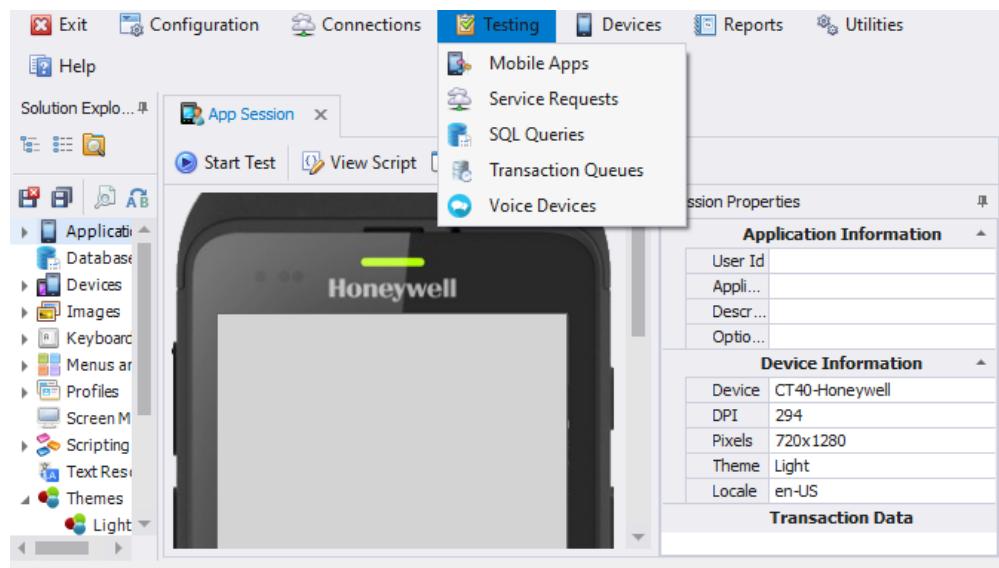
RFgen will continue testing for a total number of times specified in the **Iterations** box.

Select the item to be tested from the drop-down options. In the case of Transaction queues, the user should have already queued what they need tested. The window pane on the right will display the parameters of the item being tested.

As testing is taking place the window will show a status of the processing. If a transaction fails, the error message is displayed. If it is successful, the amount of time required to process that transaction is displayed with a success message.

The Debug Script menu option allows the user to stop, debug and test the scripting of the Transaction or Timed Event macros as they are executed.

Testing Overview



The **Testing** Menu in the Mobile Development Studio provides:

- runtime sessions for **Mobile Applications** so you can see how the application screen looks in a device at runtime, view and debug code or objects (i.e. applications, screen maps, services).
- a means of testing **Service Requests** for integrations with cloud based solutions.
- a interface to execute **SQL Queries**, query results to a file via Excel, and enable or disable your ERP Connections when executing a SQL query.

- runtime sessions for testing **Transaction Queues** or **Transaction Events** so you can check Cycle Time and Iterations.
- data steam portions of other third-party data collection solution that are voice driven and use unique hardware and software for the testing of **Voice Devices**.

Connection Limitation

When you test device connections with the Mobile Development Studio, the Mobile Development Studio limits the number of simultaneous connections between itself and remote devices. For example, you can only have a total of 3 connected sessions.

Stop Test - This allows you to start and stop your test sessions.

View Script - For information on testing your script see [To Debug Code](#).

Options - For information on this feature, see [Testing Options menu](#).

Mobile Apps: Session Property Descriptions

Application Information

User ID - Displays the user who is logged in during the test session. For example if you had a user named Sam setup in Users and logged in as SAM, "Sam" would display here.

Application - Tells you which application you are testing.

Description - The description that is associated with the Application you are testing.

Options -

Device Information

These settings can be changed using the Options menu.

Device - Which target device image is in use.

DPI - Tells you which values are used to scale the device image.

Theme - Tells you which Mobile Theme is used for the application's look/feel.

Locale - Tells you which language is used if you have your application setup for different locales.

Device Management Options

The **Access / Authorized Devices** screen tracks authorized devices and enables administrators to: a) manually authorize mobile clients, and; b) manually authorize Thin Clients (if the server is configured to Restrict Online Access), or b) view automatically authorized Thin Clients. This feature is used for authorizing all types of client connections.

The **Application Deployment** provides various ways to transfer CAB or CNC files to Windows CE\Windows Mobile devices. The server can then update mobile applications, menus, users etc. as part of as needed for data collection. If the server is upgraded, and new client files need to be distributed, this function can help perform this task. Full profiles can also be sent to the device.

The **Application Explorer** will support exporting components to Windows CE\Windows Mobile devices.

The **Database Explorer** shows the data on Windows CE\Windows Mobile device's database.

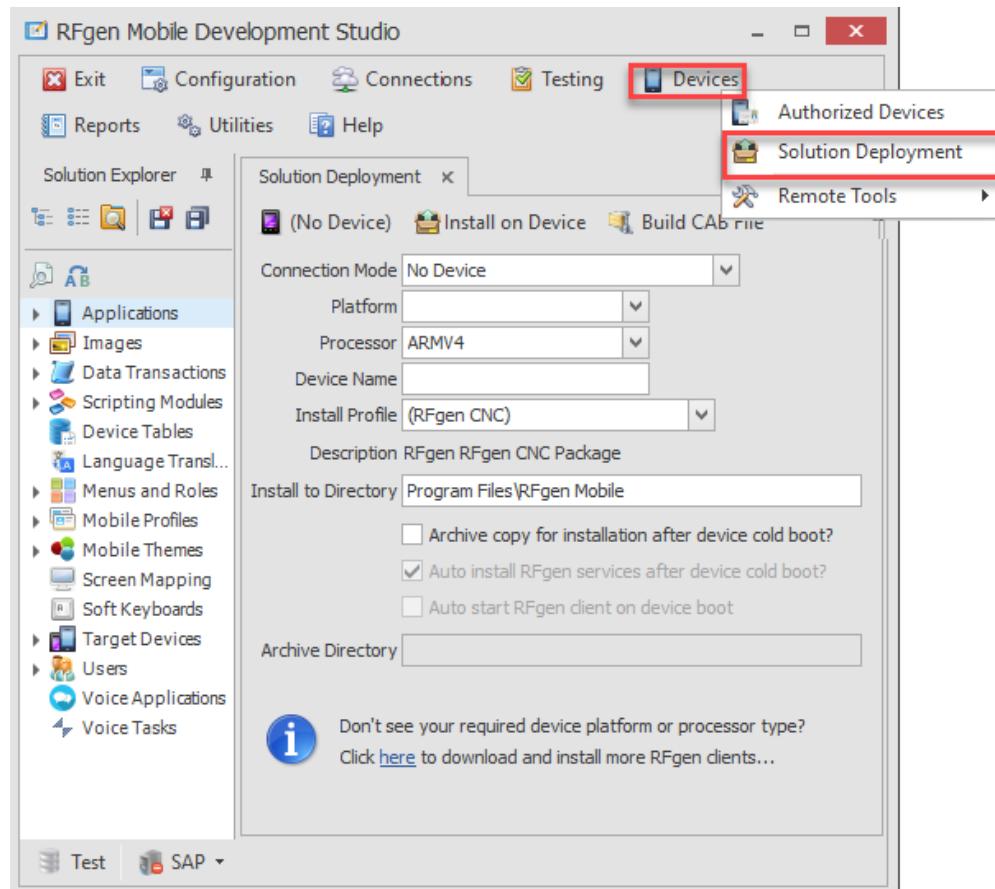
The **Log Viewer** will show the error log file on Windows CE\Windows Mobile devices.

The **SQL Explorer** will let the user read and write directly to the user database on the device

Solution Deployment

The Solution Deployment tab is a collection of client-server connection and deployment methods for installing the Mobile Profile to a Windows CE devices or desktop and other files.

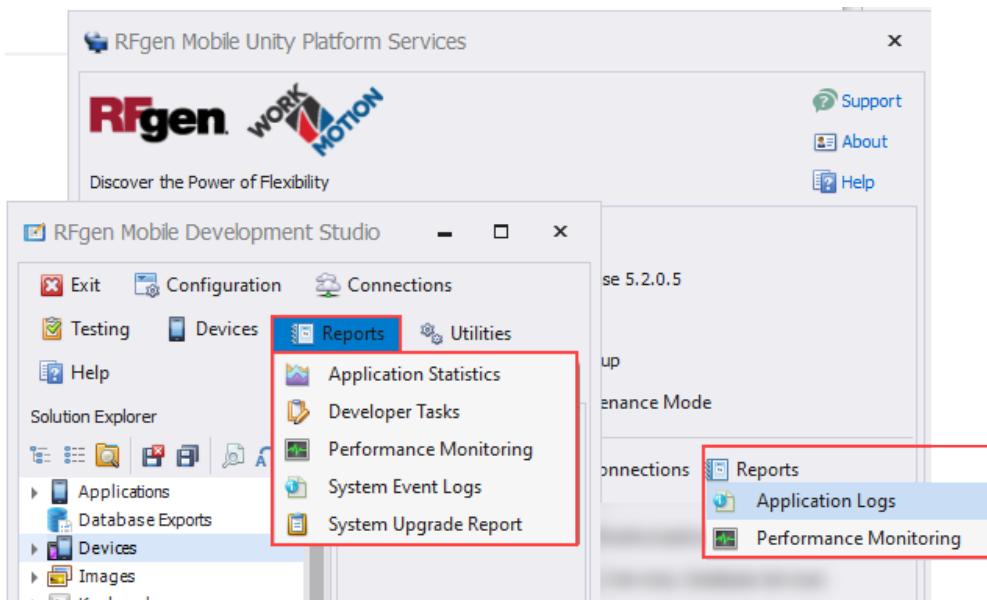
All communication and interaction with the mobile device goes through the CNC, when enables the device to "listen" for connection requests by the server. (For information on obtaining and downloading this application, refer to the *RFgen Installation and Upgrade Guide*.)



Reports

Reports can be accessed from the Mobile Unity Platform Services console (Services Console) or the Mobile Development Studio (Dev Studio).

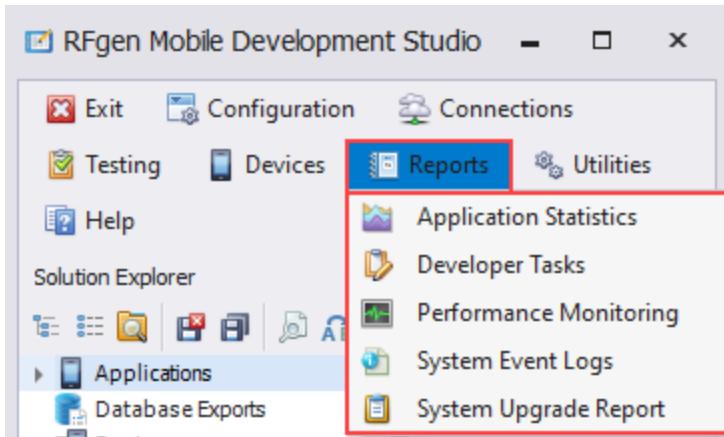
Reports Available in the Services Console and Dev Studio.



The **Application Logs (Event Log Files)** display system errors in the Event Log screen, and includes a SQL Filter and Export to Excel tool. This is available in the Services Console and Dev Studio Reports menu. See [Configuration > Application Event Logs](#) for more information.

The **Performance Monitoring** log lists events generated by the execution of scripts that exceeded a threshold value. (i.e. Flagged events that exceed processing time thresholds.) It includes an SQL Filter, Display Options and an Export to Excel tool. The report is available from both the Services Console and the Dev Studio Reports menu. This requires configuration. See [Environment Settings: Performance Monitoring](#).

Reports Only Available in Dev Studio



The **Application Statistics** report displays statistics regarding database connections and ERP transactions.

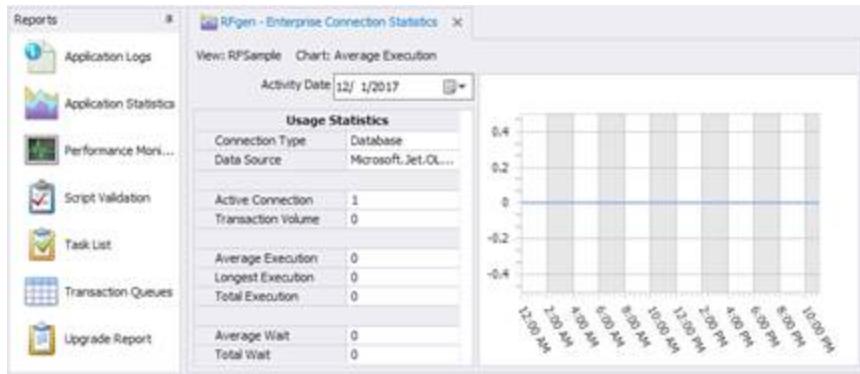
The **Developer Tasks** includes a list of programmer tasks to be completed if the script was flagged with a comment and the words "TODO" after upgrading major versions of RFgen (i.e RFgen version 5.1 to 5.2).

The **System Event Logs** lists the error messages and events that occur between RFgen and other systems it communicates with.

The **System Upgrade Report** lists the results of a Validate System check from the Studio > Utilities menu. This reports on what was left unfinished or needs further investigation when you upgrade RFgen via RFgen's Mobile Unity Platform installer. For example, if you were to update from RFgen 5.1 to 5.2, this report may warn you about which applications may need more attention.

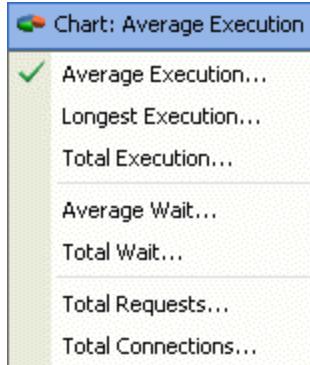
Application Statistics

Clicking on the 'Reports' menu selection, the 'Application Statistics' will display statistics regarding database and ERP transactions.



The View menu option selects between each of the configured data connections and the Transaction Management database.

The Chart menu option will show the performance of a given statistic over the course of the day. The options are:



Average Execution is the typical time it takes to execute one call to the specified data connector. The graph shows this average across the whole day. The lower the number, the better. Typical values should be well under one second.

Longest Execution is the longest time RFgen had to wait for one call to the specified data connector. The lower the number, the better. Typical values should be well under one second.

Total Execution is an accumulated amount of time that RFgen has spent waiting for all executed calls to the specified data connector.

Average Wait refers to how long on average a user must wait for RFgen to provide them a connection to the specified data connector using the Connection Pooling process. Typical values should be less than one second. If the user must wait longer, then the connection pool should be increased.

Total Wait is an accumulated amount of time that users have spent waiting for RFgen to assign a data connection handle from the pool.

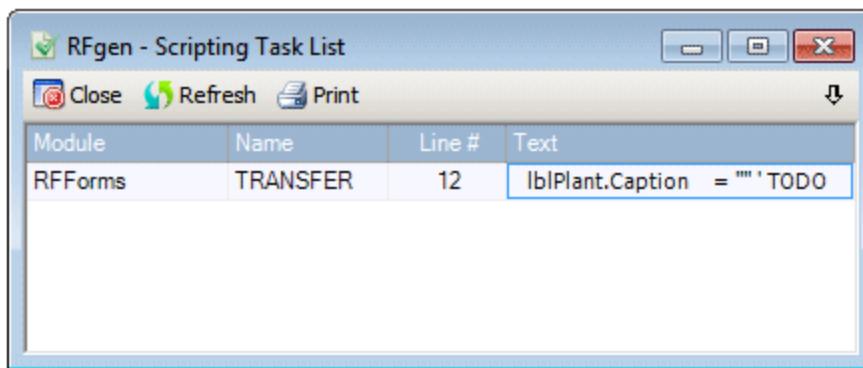
Total requests is the total number of times RFgen access the specified data connector for any reason.

Total Connections is the number of currently open connections to this data connector. Without Connection Pooling, each logged in user will have their own connection. With Connection Pooling enabled, the maximum should be the limit placed on the pool in the configuration and the minimum should be one.

To enable the Statistics, choose the Configuration / Performance Monitoring menu option, select the Record Usage option and change the value from Disabled to some increment for refreshing the data.

Task List

The Task List is a list of all TODO code markers suggesting the programmer left something unfinished. If the code contains a comment mark and the “TODO” word then that line of code will appear in this list. Double-click any entry in the list and that script window and code line will get the focus.



Module	Name	Line #	Text
RFForms	TRANSFER	12	lblPlant.Caption = ""' TODO

Performance Monitoring

The Performance Monitoring report is a list of all flagged execution events such as database, ERP, legacy host, web service and scripting executions that exceeded the millisecond threshold values. The thresholds are setup in the Configuration > Environment Settings > Performance Monitoring table. The time of the event, data source, code module, function, line number and parameters used are all displayed.

For more details, see [To Setup Performance Monitoring](#).

To Setup Performance Monitoring

To view data in the Dev Studio Reports Performance Monitoring screen, the following items will need to be setup:

1. Setup a connection to a transaction database
2. Setup connection to the events database
3. Set threshold values so to trigger data to be captured in the events and database monitors
4. Run test application/data so to verify the monitors are receiving the data.

Before you start

- Will need a database application. RFgen supports Access, DB2, ODBC, OleDb, Oracle, SQL Server and SQLite.
(SQLite is the easiest and most common application to use.)
- You will need the path to your RFgen Program Data file so to connect to the IPC.db and the RFgenTM. These are typically located in the Program Data\RFgen 5.1\IPC.db file or the Program Data\RFgen 5.1 file.
- Your own test data to verify the output in the Performance Monitor. (i.e. Use one of your applications.)

The IPC.db file contains performance data and events.

Setup Connection to Transaction Database

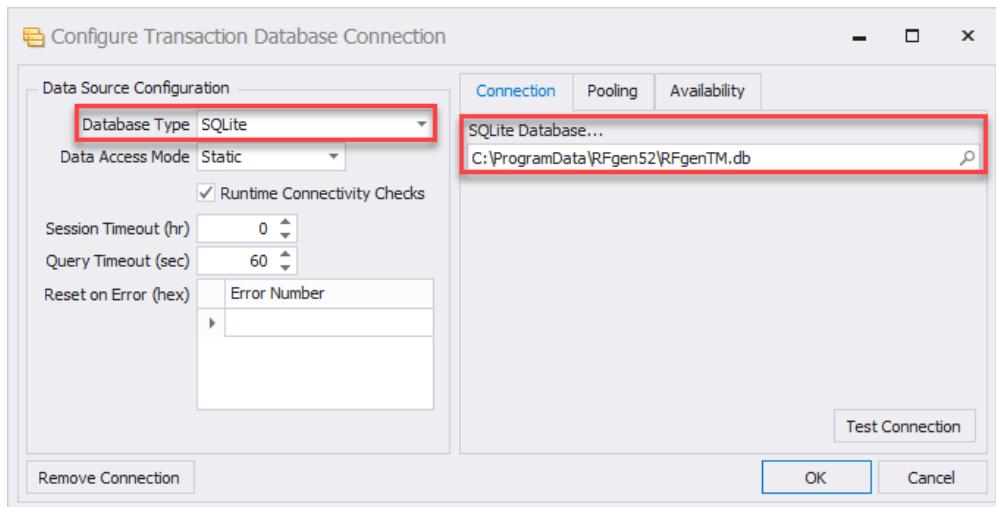
In this example we will setup a connection to a Transaction database using SQLite. This database will be used to track the transactions if the transaction executions trigger one of the thresholds set in Environment Settings > Performance Monitoring.

These steps can be performed in the Mobile Development Studio or in the Mobile Unity Platform Console.

You can use any standard database; for the purposes of this example, we used Microsoft SQLite.

1. Open **Connections > Transaction Management Database**.
2. In the **Configure Transaction Database Connection, Database Type** menu select **SQLite**.

3. In the SQLite Database... search, enter the path or search and select **RFgen TM.db**



4. Leave the other values the same for now. Click on **Test Connection** and verify you have a good connection, then click **OK**.

5. A small RFgenTM icon should display in the lower left corner of your screen.

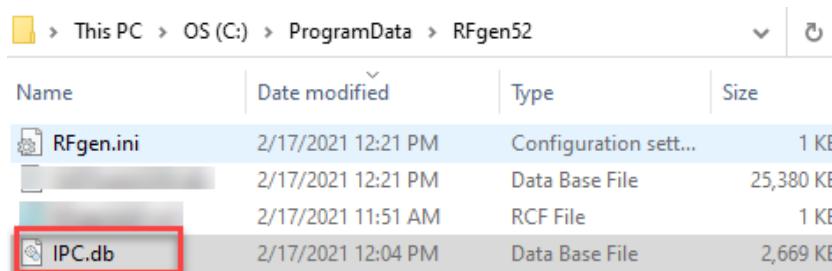
Setup Your Events Database

1. Open **Connections > Application Events Log**.

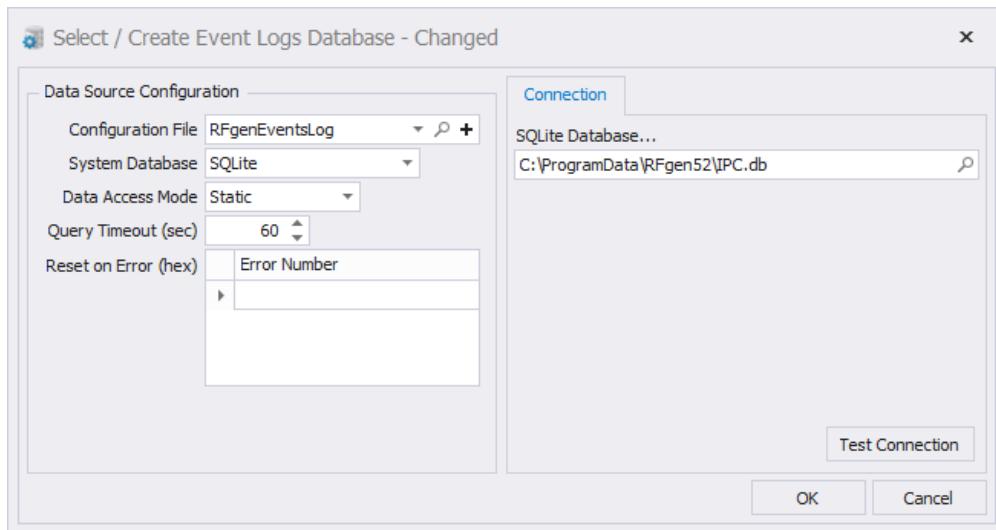
2. In the **Select / Create Event Logs Database, Configure File** field click on the + sign. In the pop-up Window, enter a configuration file name. The configuration file can be any name as long as there are no spaces in the name. For example, "RFgenEvent". RFgen will automatically default the extension to .rcf. Do NOT use an rcf file that is purposed for your rfgn applications as the format of the database would be different than the format used for the rfgn applications database.

3. In the **System Database** drop down list, select **SQLite**.

4. In the SQLite Database field select the Search button and select **IPC.db** then click **Open**.



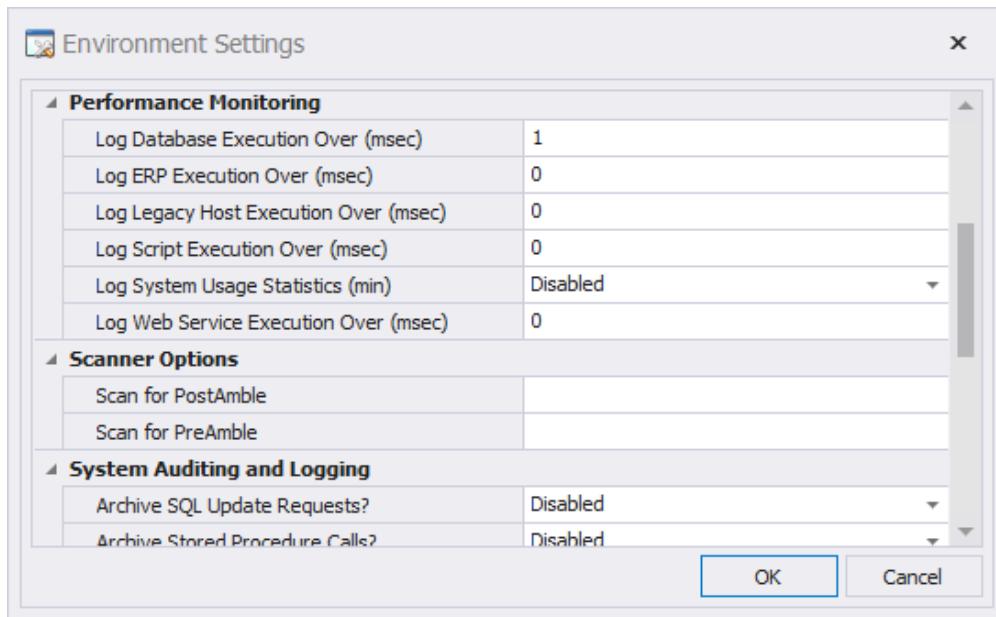
5. The screen will look similar to this screen below.



6. Click on **Test Connection**. A Good Connection or Connection Successful message should display. Click **OK** to close.

Set Performance Monitoring Thresholds

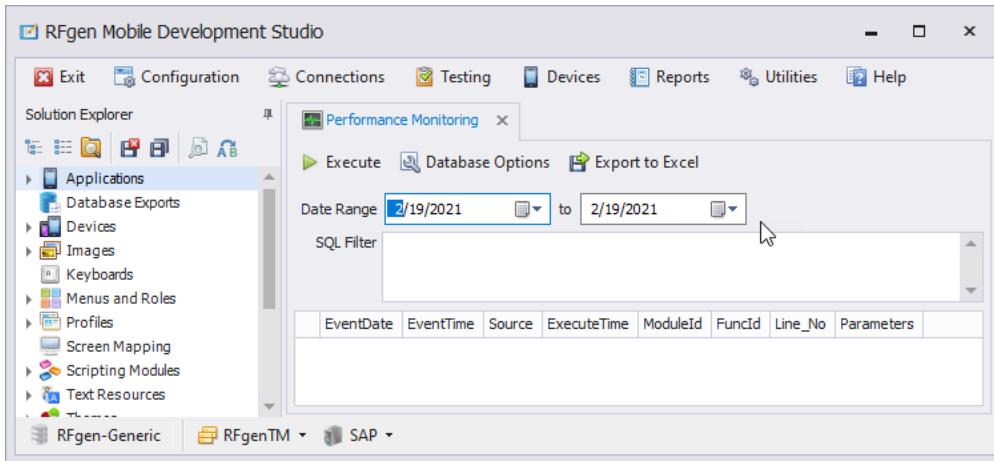
1. In the **Configuration > Environment Settings > Performance Monitoring** table enter a 1 in the Log Database Execution Over (msec) table.



2. Click **OK** to exit.

Remember -- In real environments, these thresholds would be set for a specific purpose and value. Arbitrary values would otherwise cause constant logging and added traffic to your environment.

3. Open your **Reports > Performance Monitoring Screen**.

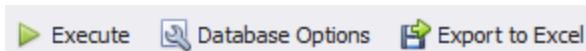


4. If you have been running transactions since you connected to the two databases above, click on the Execute button to view your data.

The screen may look similar to this one below. However, if you do not have a transaction to generate data, see the next process To Setup Test Data.

EventDate	EventTime	Source	ExecuteTime	ModuleId	FuncId	Line_No	Parameters
2021-02-19	12:20:20	FIMCC0200	1437		txtMat.OnSearch()	0	
2021-02-19	12:20:14	FIMCC0200	8484		txtMat.OnSearch()	0	
2021-02-19	12:20:05	FIMCC0200	1438		txtMat.OnSearch()	0	
2021-02-19	12:20:01	FIMCC0200	1047		txtMat.OnSearch()	0	
2021-02-19	12:19:57	FIMCC0200	1578		txtMat.OnSearch()	0	
2021-02-19	12:19:53	FIMCC0200	1750		txtMat.OnSearch()	0	
2021-02-19	12:19:49	FIMCC0200	1672		txtMat.OnSearch()	0	
2021-02-19	12:19:44		328		EmbedProc.Execute()	0	
2021-02-19	12:19:44	FIMCC0200	671		txtDoc.OnEnter()	0	
2021-02-19	12:19:43	FIMCC0200	2485		txtDoc.OnSearch()	0	
2021-02-19	12:19:42		328		EmbedProc.Execute()	0	
2021-02-19	12:19:28		2625		System.Initialize()	0	

System Event Log Menu

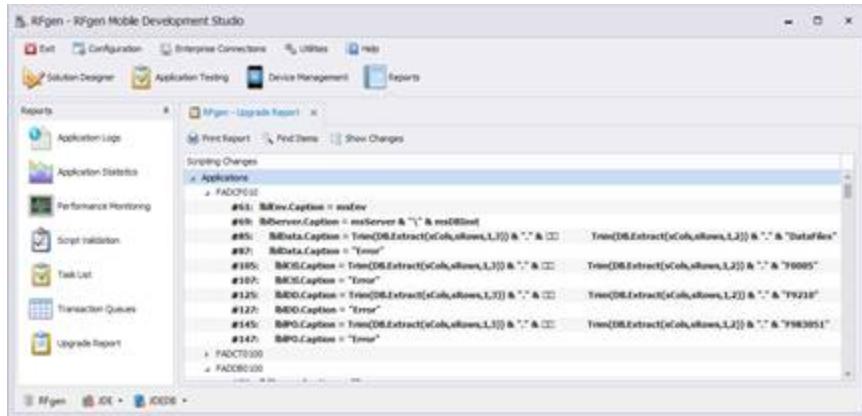


The **Execute** button runs the SQL Filter described below.

The **Database Options** sets the error log's maximum number of rows and allows you to delete items in the table.

The **Export to Excel** option creates an XLS file at the selected path. Excel does not have to be installed on the system.

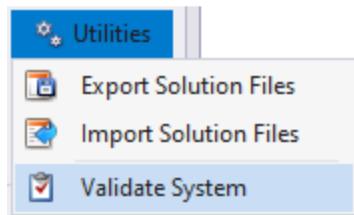
Upgrade Report



When RFgen Mobile Device Studio is upgraded to newer version, if an issue was encountered, the issue will display in this Upgrade Report. If the install was successful, the Upgrade Report will remain blank.

In general, the Upgrade Log will provide "Upgrade Markers" which are indicators that the upgrade process left something unfinished or is simply a warning requesting investigation. Double-click any entry in the list and that script window and code line will get the focus.

Utilities



The Utilities menu option provides tools to execute script validations and import or export objects (solution files) .

Exporting and Importing Solution Files

If your applications have already been developed and you merely wish to use them on another system, you'll need to transfer the Mobile Development Studio objects (Applications, Menus, Users and VBA code and macros) from your current system to the destination system.

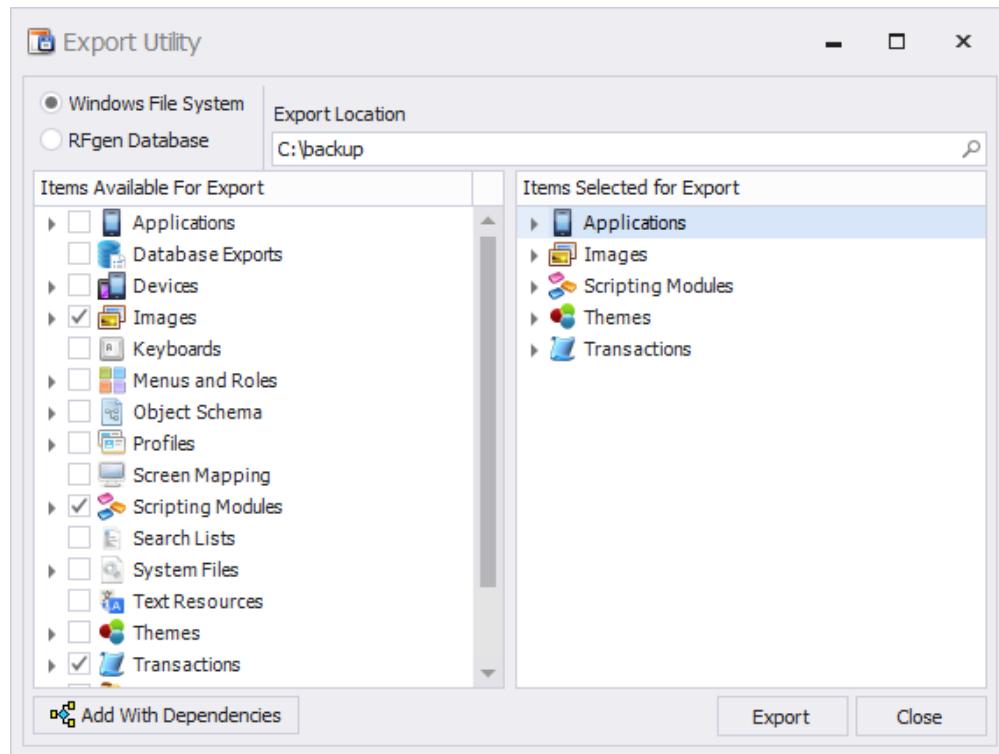
Objects are freely transferable from/to other solution databases, or other external files, for the following purposes: (1) production usage, (2) ongoing development, and (3) backup / retrieval.

To transfer an entire set of (same release) applications, and overwrite the current set, simply copy the solution database from the development system to the production system while the production system is not in use. Be sure to alter any data connectors if necessary.

Validate System

This utility is used to validate script in all the applications in your RFgen Mobile Development Studio. If an issue is found, the output log highlights the application with something that's invalid in red (or a different color) text. This is also referred to as [Script Validation](#).

To Exporting Solution Files

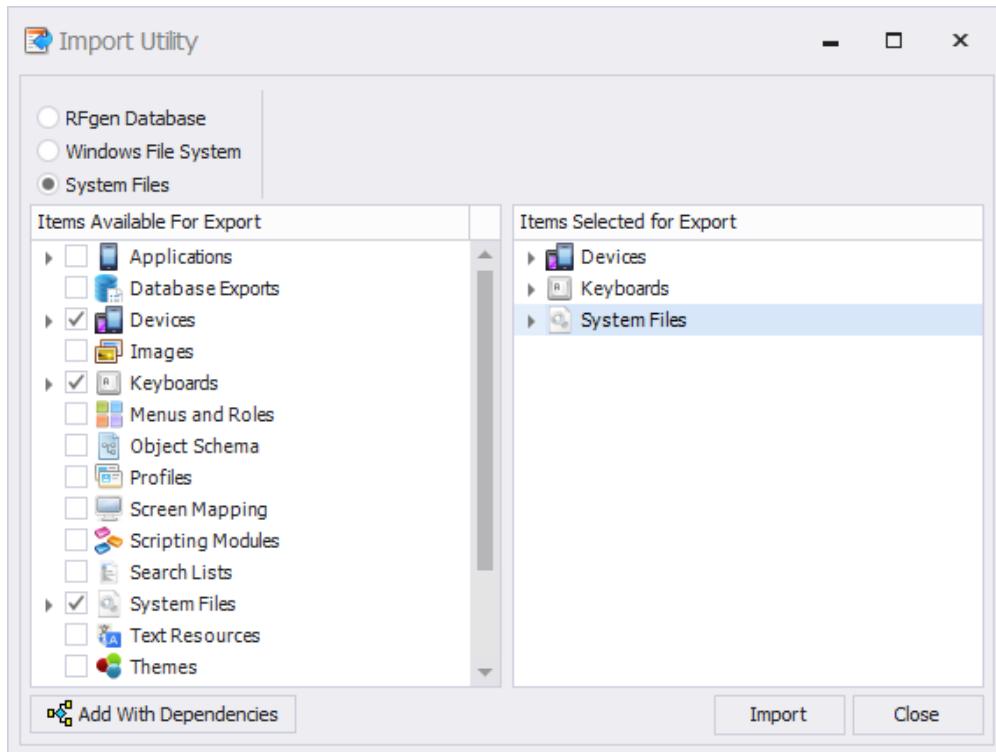


1. Select **Utilities > Export Solution Files** from ribbon menu.
2. Click on the source **Windows File System** or **RFgen Database** and enter the destination, Export Location.
3. The, select the source database to be exported, for example, we show the RFgen Database and its configuration file "RFgen". You can also select the search icon to search for an existing file.
4. Enable or disable the "Add with Dependencies" button.
5. Choose the objects to be exported, then click Export.

Importing objects works similarly, except that you will be overwriting items in your local solution from a remote file.

When importing or exporting items, the release number used to create the items must be the same as the release number for the items being overwritten. You will be stopped if they are not the same.

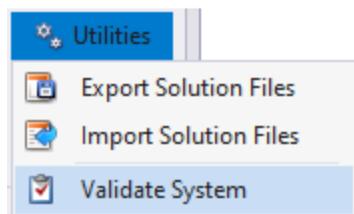
To Import Solution Files



This tool imports database configuration files, files from your Windows system (i.e. images to be added to your RFgen Images folder, text source files for translations, files from another programmer), or RFgen System files (i.e. RFgen provided Device skins, RFgen keyboard files etc.).

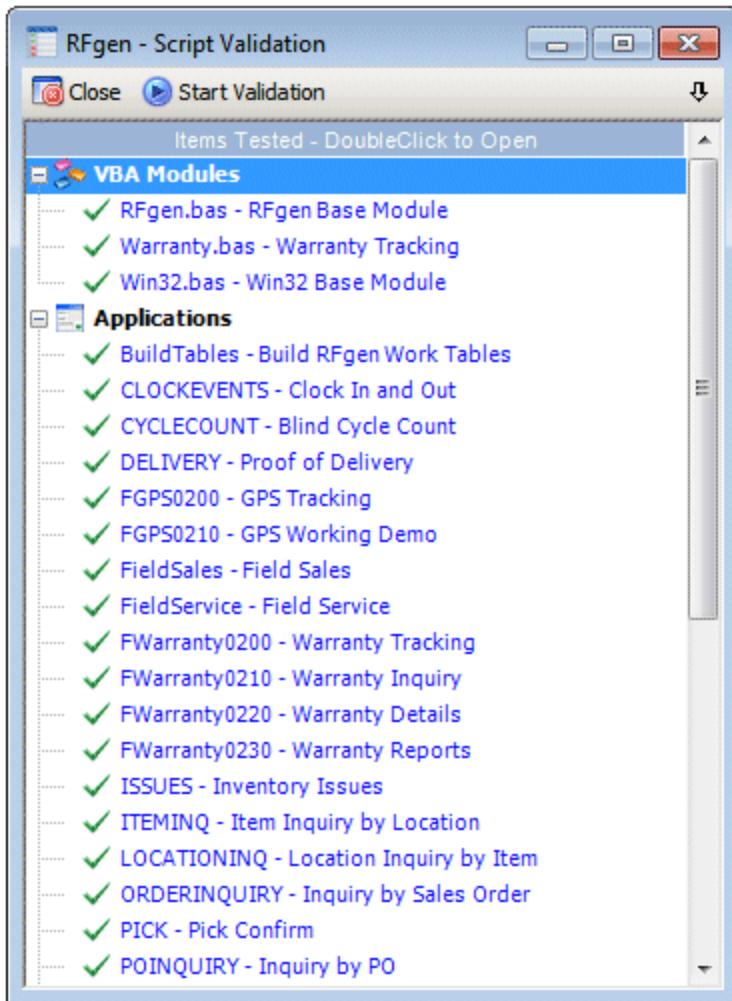
1. Select **Utilities > Export Solution Files** from ribbon menu.
2. Select a source from the list above: RFgen Database, Window File System, or System File.
If importing a Windows File, enter the Windows directory of your source file in the **Import Location** box.
3. In the **Source Items** panel, select the files and folders you want to import.
4. Click the **Import** button.

Script Validation



Use the **Validate System** utility in the Mobile Development Studio to perform a VBA syntax check for all coded objects. Any application or macro, etc. that has a syntactical error will display the yellow triangle-warning icon. Double-clicking on any line will load and display that code page for convenience.

This tool can also be used BEFORE you upgrade your RFgen software in order to track if there are any pre-existing issues within the script of an application.



RFgen Client Software Overview

The **RFgen** Client software enables mobile devices to:

- Communicate with the RFgen server so you can deploy a Profile (a file containing the collection of applications, server settings, access permissions etc) to the device.
- Configure the RFgen clients' profiles through the RFgen Configuration tool (included when you install the RFgen client).
- Communication with 3rd party, mobile device management tools (for deployment of client software on a mass basis).

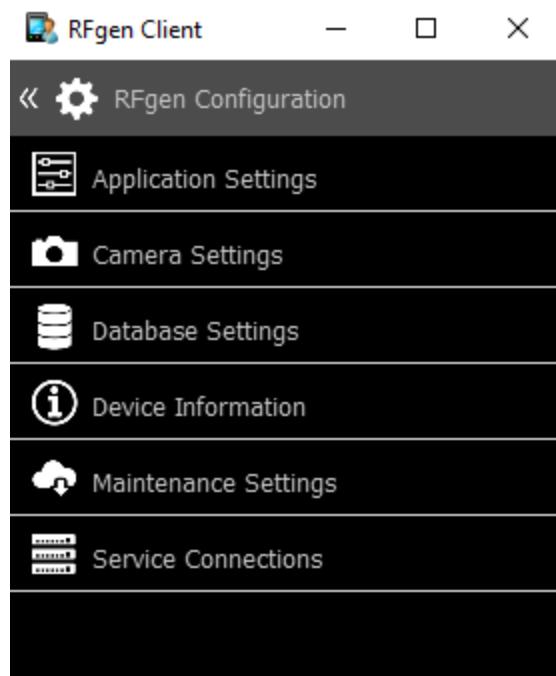
The four basic device platforms are Android, Apple iOS (but not Macintosh or Apple computer platforms), Windows desktop systems, and the compact embedded, Windows CE.

This guide describes covers:

- Where to obtain the RFgen Client software
- Which OS versions are supported
- Instructions for customized installations (i.e. Android)
- How to connect the client to the server after its been installed to the device
- The possible dialog or error messages you might see and what they mean

For details on installing or transferring the RFgen Client software to your physical device, refer to your manufacturer's documentation and the documentation for the version of the operating system of the platform.

Client Configuration Settings



RFgen Configuration is a collection of status and settings that are used to change how a mobile client (Android, iOS, Windows Desktop, or Windows Mobile/CE) starts up, receives updates, and displays your application screens.

For more information, see the specific topics on [Application Settings](#), [Camera Settings](#), [Database Settings](#), [Device Information](#), [Maintenance Settings](#), or [Service Connections](#).

Thin Client Overview

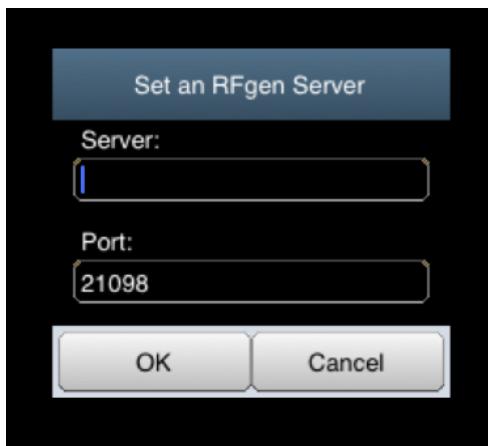
While in thin client mode, the user interacts with a session running on the server. Since all the processing takes place on the server, the mobile device cannot be a point of failure or lose data.

If the wireless device goes out of range of the network, the mobile device screen will appear to stop since the server cannot order the screen to refresh. The client on the mobile device will continue attempts at reconnecting and will then resend the last piece of data entered. RFgen has added a "Guaranteed Packet Delivery" system to the protocol to ensure no loss of data and an always-synced application.

The advantage to the thin client is real time updates to backend systems as well as complete validation data available to ensure the collected data is as accurate as possible. The disadvantage is the need for a wired, wireless, or cellular connection available while collecting data. This client does not require any authorization process.

Connect and Deploy to RFgen Clients

Once you have created your mobile profiles in the Mobile Development Studio, and installed RFgen software to target devices, you enter the RFgen Server hostname or IP address in the RFgen screen. This will request the server to communicate with the client and download the mobile profile of your choosing.



Note: If you do not want to download from the server over a network and your device is Windows Desktop, CE or Android, click [here](#) for alternate methods of installing a mobile profile.

Client Network Control Service

On the mobile device there is an icon  that represents the CNC (Client Network Control) service. Clicking on this icon may give various options depending on the implementation. The purpose of this service is to allow

requests from the server to be performed on the device.

Some core capabilities include the ability to detect that a server upgrade has occurred and to auto-update the client environment. Further, if you've deployed in a mobile (off-line capable) environment, CNC provides support for "Application Synchronization" requests. In this scenario, if you've changed any applications that are in the mobile device's profile, it can automatically detect the application / profile changes, build a custom deployment package and remotely update the device – all without the intervention of IT personnel.

The CNC service is installed when the RFgen Mobile Client is installed.

If needed, the CNC can also be packed and installed separately to a Windows Mobile/Windows CE device. Refer to the Device Management > Mobile Device Installation Utility in the Mobile Development Studio for more details.

Third-Party Mobile Device Management Tools

RFgen supports third-party MDM tools for configuring the RFgen iOS client.

For more details refer to the ***RFgen 5.1 Installation and Upgrade Procedures Guide***, which is available from your Program File\RFgen51\Documentation folder on Windows OS up to Windows 8, or is listed from the Apps on Windows 2012 and 2016 servers.

Installing the RFgen Windows CE Client

The **RFgen Mobile Client** (the **CE Client.exe**) for Windows CE installs software enables communicate between the device and the RFgen server.

In order for a Windows CE to process data, the client needs to be able to communicate with the server, be configured to work as an RFgen client, and have the specific applications for transacting data installed. The RFgen solution deployment feature that packages these configurations/files is called a Profile. Profiles are created in the Mobile Development Studio and transferred to the client inside a CAB file. (In other words the CAB file is the format used to package a Profile created in RFgen.)

Installing to the RFgen Server

The **RFgen Client** software (which you download from the RFgen Portal) installs CE-version specific software that enables the client to communicate with the server, and it also provides files used by the Mobile Development Studio to build CAB files.

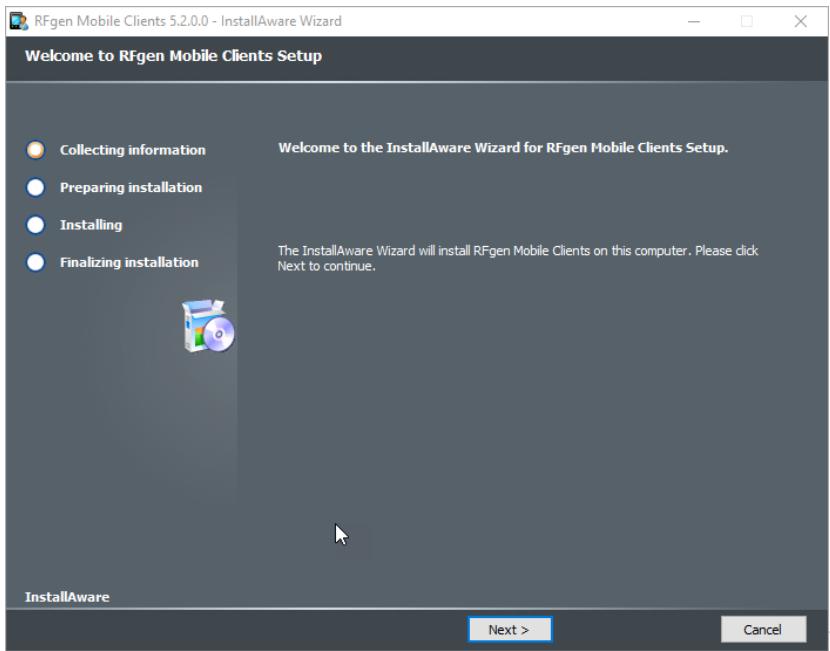
Therefore, before you can deploy any solution to a Windows CE device, you'll need to install the RFgen Client software on the same server where your **Mobile Development Studio** resides.

Note: By default, this package installs to c:\Program Files (x86)\RFgen52 folder. **If you installed the 64-bit version of the RFgen 5.2 Mobile Development Studio or installed the 64-bit version of the RFgen 5.2 Mobile Unity Platform server, make sure the RFgen Client installs to the same folder (i.e C:\Program Files\RFgen52).** By having them in the same RFgen folder location, the Mobile Development Studio will be able to locate the files needed when you are ready to build or deploy files to the client.

RFgen Client Install Process

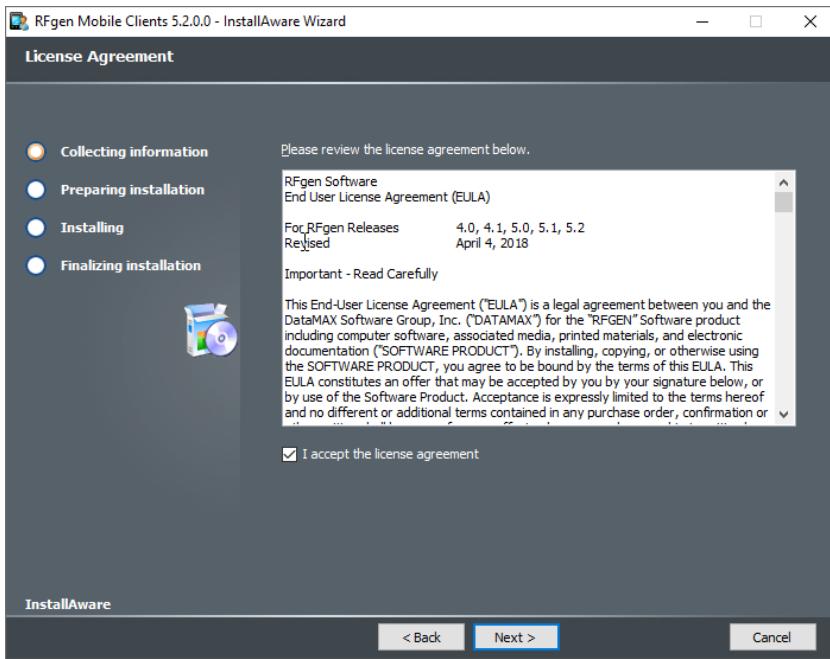
1. After you downloaded the RFgen Client package, launch it as you would any other Windows application.

The **Welcome** screen displays.



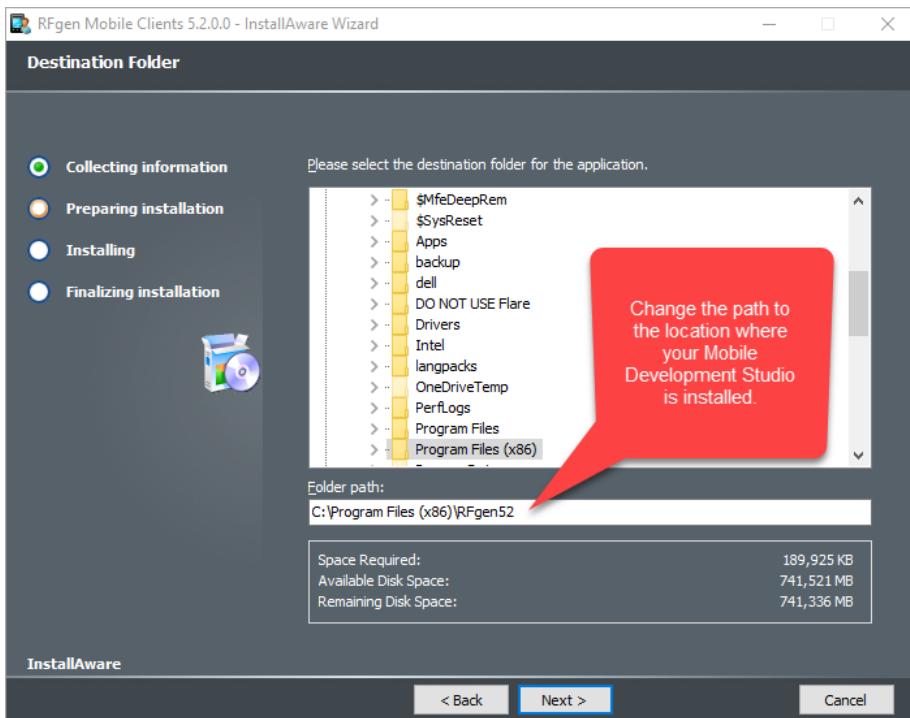
2. If you need to exit the process, click Cancel. Otherwise, click **Next** to continue the install process.

3. The **License Agreement** screen displays.

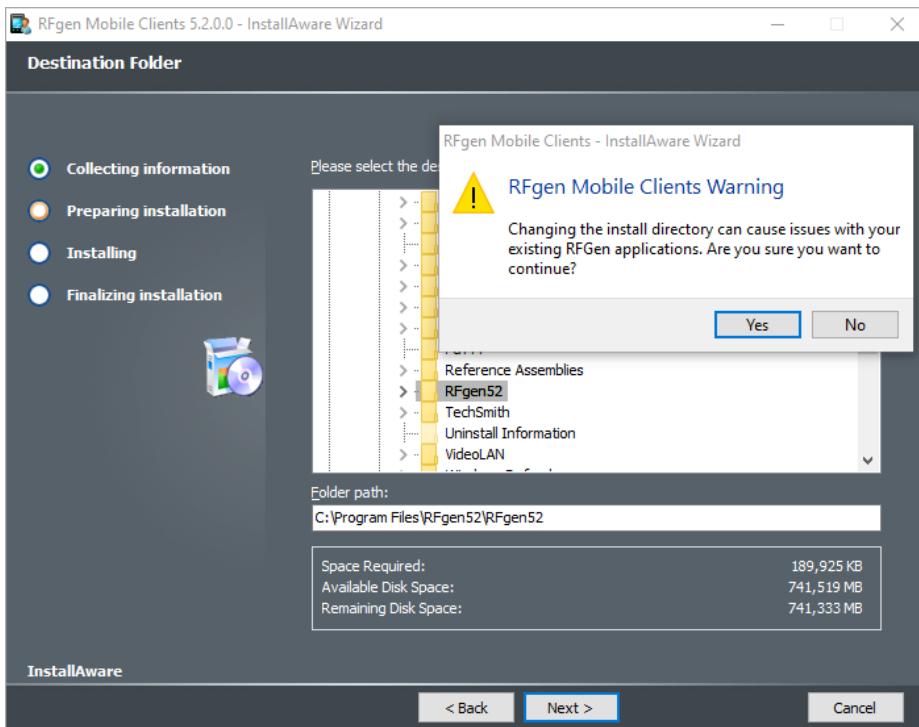


4. Click the checkbox and then click **Next** to continue.

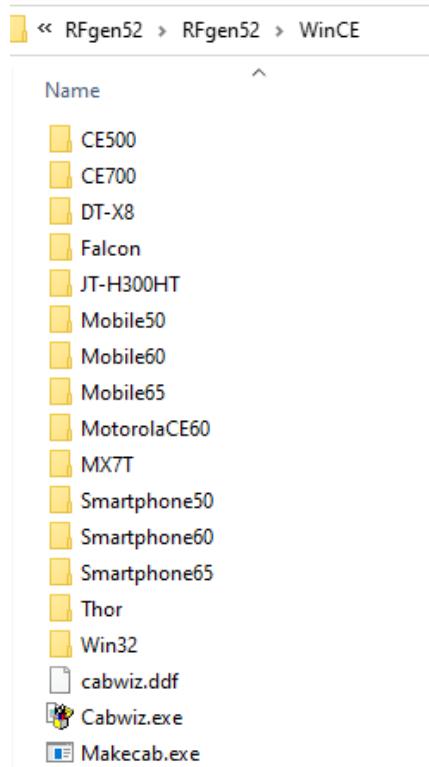
5. The **Destination Folder** screen displays.



6. Change the path to the location where your Mobile Development Studio resides. For example, if you installed the 64-bit version of the Mobile Unity Platform server, its default path is C:\Program Files\RFgen52.
7. The **RFgen Mobile Clients Warning** screen displays. Click Yes to continue; No if you want to go back and change the path again.



8. Click **Next** to continue.
9. The **Ready to Install** screen displays.
10. Click **Next** to continue. When its done, click the **Finish** button.
11. Note that the RFgen 5.2 folder under the parent RFgen 52 folder contains the various Windows CE device files and Cabwiz.exe and Makcab.exe files.



12. Now you are ready to prepare your CAB files for Windows CE. For more details, see [Solution Deployment](#) and [To Create CAB](#) files.

Activating Mobile (Batch) Clients

On Mobile Clients, the RFgen administrator can deploy authorization certificates to RFgen clients on Android or iOS systems through the server **Device Management > Device Authorizations** feature.

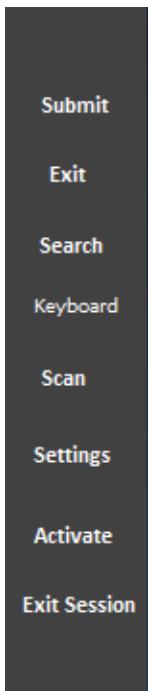
Deployment of the certificate occurs after the device has received a Mobile Profile and the RFgen Administrator has setup the certificate to be deployed to a specific device (from the server).

To activate / authorize the license from the device, launch the device **Menu Strip > Options** and tap the **Activate** button should display.

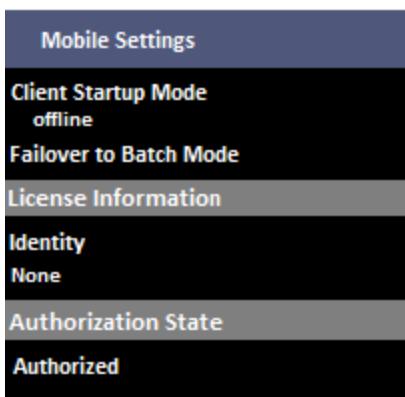
EXAMPLE

This is an example of how you can activate the device. Your own application and menus may be different as these can be customized by the developer.

1. Connect to the RFgen server.
2. When your mobile application Login displays, click on the menu button to display the menu strip.
3. Open the Menu strip, and tap **Activate**. This button is only present when the certificate is present for a download.



You can also verify if the device is authorized by reviewing the device's **RFgen Configuration > Mobile Settings > License** Information: Authorization State.



Activating Mobile (Batch) Clients

On Mobile Clients, the RFgen administrator can deploy authorization certificates to RFgen clients on Android or iOS systems through the server **Device Management > Device Authorizations** feature.

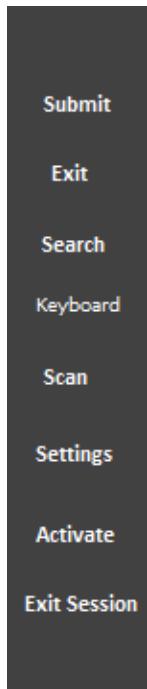
Deployment of the certificate occurs after the device has received a Mobile Profile and the RFgen Administrator has setup the certificate to be deployed to a specific device (from the server).

To activate / authorize the license from the device, launch the device **Menu Strip > Options** and tap the **Activate** button should display.

EXAMPLE

This is an example of how you can activate the device. Your own application and menus may be different as these can be customized by the developer.

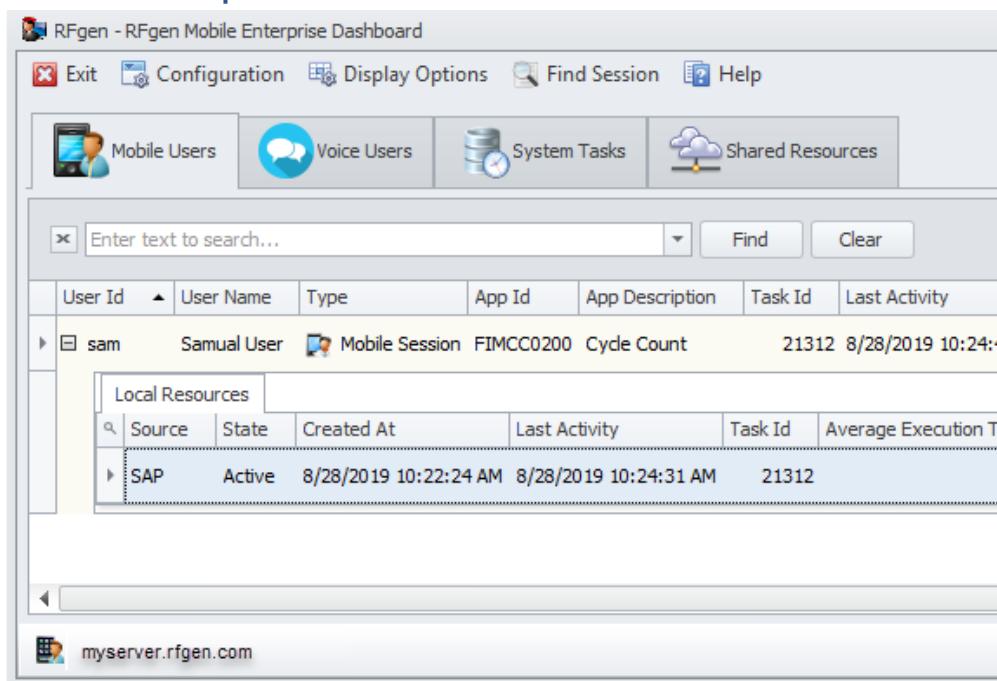
1. Connect to the RFgen server.
2. When your mobile application Login displays, click on the menu button to display the menu strip.
3. Open the Menu strip, and tap **Activate**. This button is only present when the certificate is present for a download.



You can also verify if the device is authorized by reviewing the device's **RFgen Configuration > Mobile Settings > License Information** Information: Authorization State.



Mobile Enterprise Dashboard

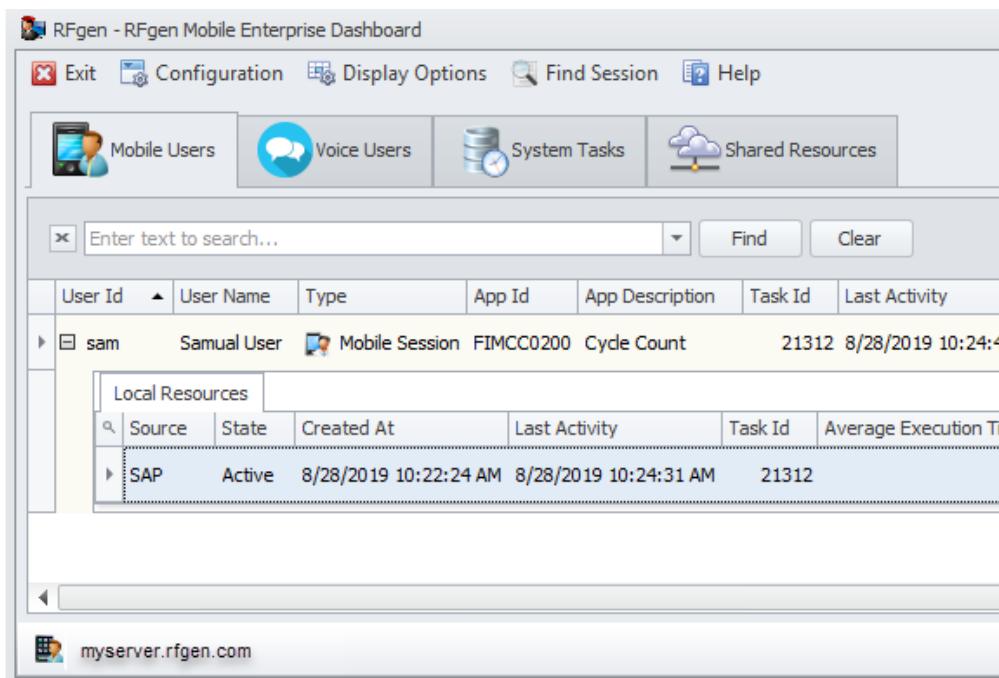


The Mobile Enterprise Dashboard allows you to view and manage the remote sessions running under the server. You can change your views to monitor different types of sessions. For example: Mobile Clients, Voice Users, System Tasks or Shared Resources.

As devices log in, they are displayed in the dashboard.

This capability may be used as a system resource. For example, a hardwired (networked) user who connects to the system will receive the same screen that appears on the screens of remote devices.

Mobile Enterprise Dashboard Menu



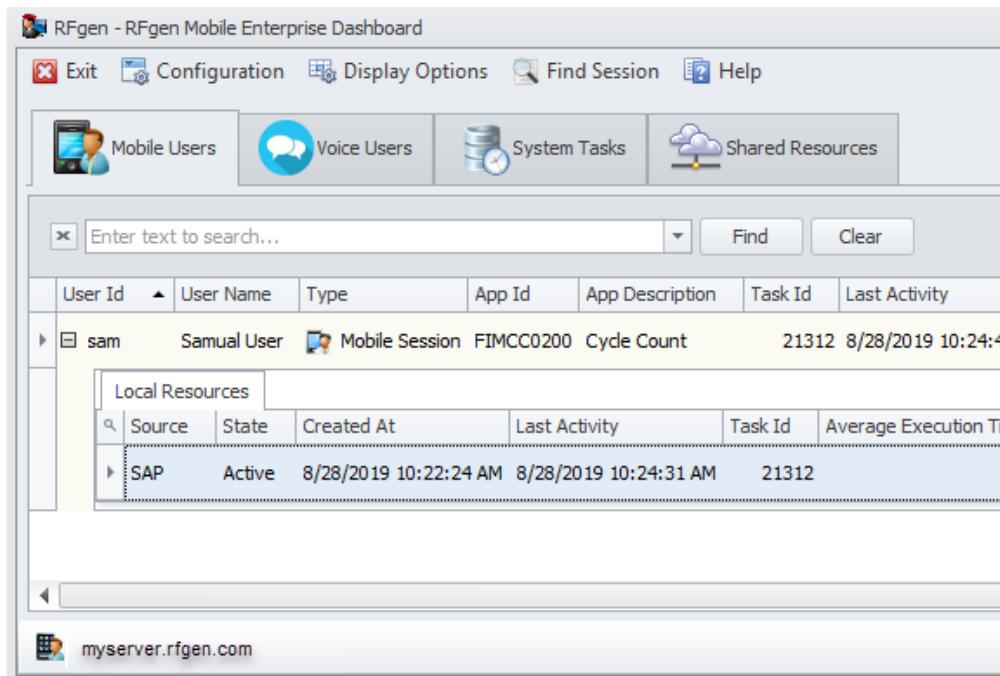
Configuration has two settings: *Application Preferences* and *Select Server Group*. The *Application Preferences* allows you to change the Windows Theme for the dashboard. *Select Server Group* stores a list of discovered RFgen servers. Once a connection is established, the selected server will display at the bottom of the Dashboard. If the connection is invalid, a red (-) will appear.

Display Options lists the column headers used to view user connection information in the Dashboard area.

Find Session allows you to find a specific session when you many sessions going on in the dashboard. For a description of each item in this menu, refer to **To Configure Your Views**.

Help menu allows you to access the topics from the RFgen Manual, obtain information on how to access Support, and view version and platform information about the Dashboard.

Overview of Dashboard Views



The **Mobile Enterprise Dashboard**, enables you to monitor and manage remote sessions running under the server. This includes:

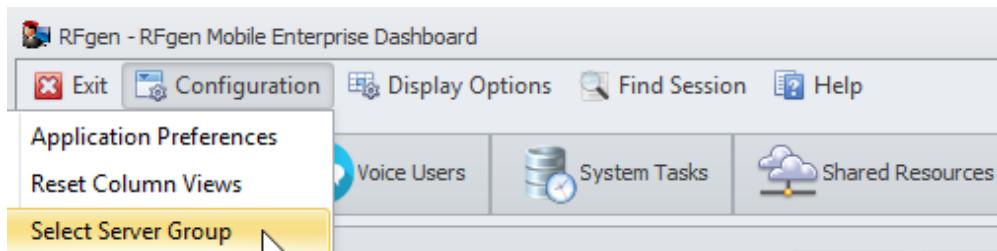
- **Mobile User** sessions - shows the session by user name/ID once the user logs in.
- **Voice User** sessions for users of voice applications.
- **System Tasks** sessions by the system or data source (i.e. SAP, Oracle etc.)
- **Shared Resources** for viewing pooled sessions (where you have license pooling setup for an ERP)

Through this dashboard, the administrator can perform tasks such as joining a session, send messages to a user, and suspending or terminating their session. Specifics about a session can also be collected. For example, if you want to see how long a task is taking to execute, you can look for this information on the System Tasks tab.

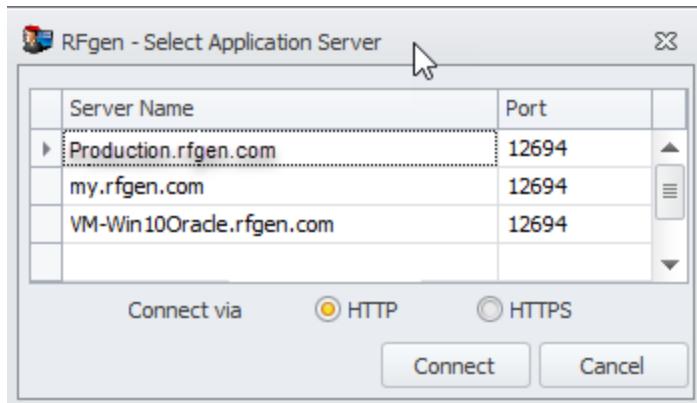
Note: When the session ends the display disappears.

RFgen Server Connections

1. From the RFgen Mobile Enterprise Dashboard, select: **Configuration > Select Server Group** .



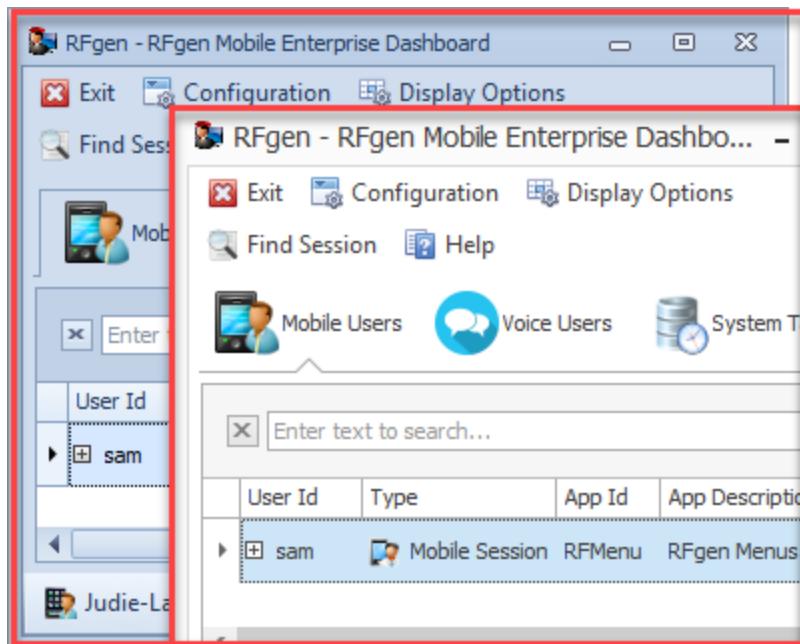
2. The **Select Application Server** screen displays.



Select the server as the source for viewing information in the dashboard.

3. Select the connection type (HTTP versus HTTPS).
4. Enter your credentials if required to access the server.
- Press Connect.**
5. The server/server group you connected to displays in the lower left corner of the dashboard.

To Set the Language in Your Dashboard

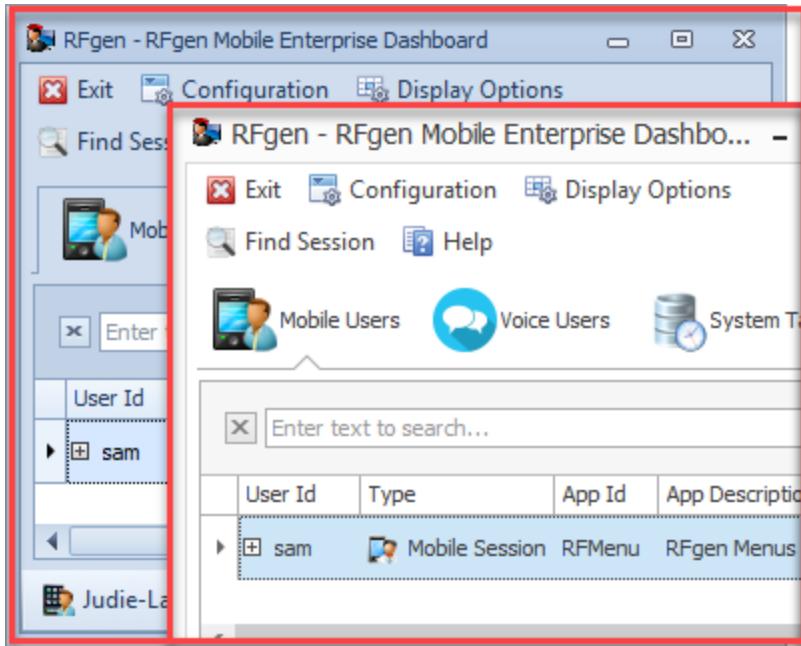


You can change the local\language your RFgen Mobile Enterprise Dashboard from English to any of the languages listed in the Application Interface drop down menu. This will only affect your view of the dashboard and will have no affect on the Mobile Applications viewed by end users.

The languages supported by the RFgen are: English, Arabic, Chinese, French, Japanese, and Spanish.

1. From the RFgen Mobile Enterprise Dashboard, click on Configuration > Application Preferences.
2. Select the language you want to use from the Application Interface drop down menu. For example, English to Arabic. You can also set the Default Locale (i.e. specify the language used in a region such as English (United States)).
3. Click **OK**. The screen will shutdown immediately. (Or, you may need to manually restart to make the changes take place.)

To Change Your Dashboard's Appearance



You can change the general look and feel of your RFgen Mobile Enterprise Dashboard by changing its skin/Application Skin. This will only affect your view of the dashboard and will have no affect on the Mobile Applications viewed by end users.

1. From the RFgen Mobile Enterprise Dashboard, click on Configuration > Application Preferences.
2. Select the Application Theme from the drop down menu.
3. Click **OK**. The screen changes immediately to the chosen theme.

To Configure Your Views

Select the icon for which you want to design a view. For example, select the **Mobile Users** icon.

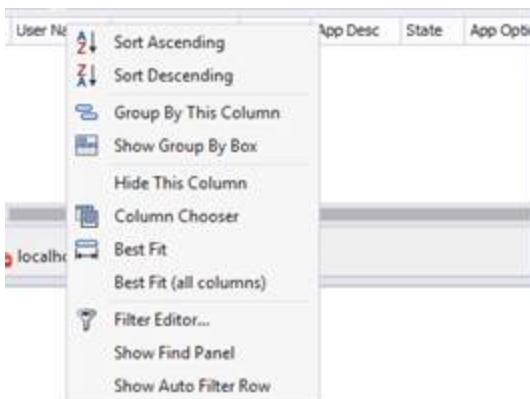
In the display panel area, you can choose to:

- a. Rearrange the order of columns – by selecting then dragging it to its new location
- b. Hide a column – by selecting this from the Right-Click Edit menu
- c. Add a column from the **Display Options** menu

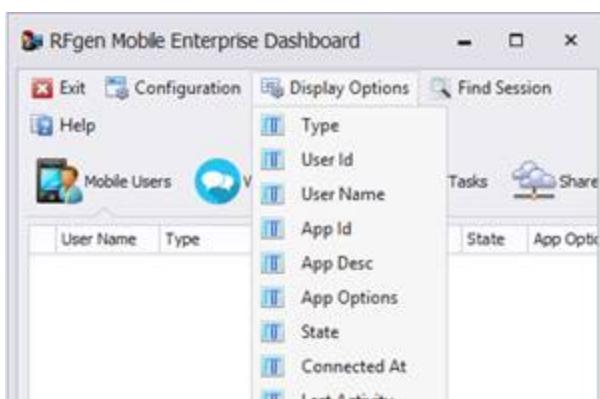
d. Sort and filter columns using the options from the Right-click menu.



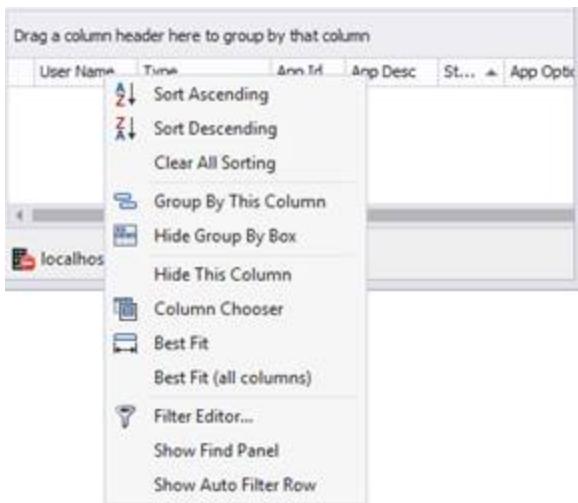
Example a. Rearranging columns



Example b: Hide or Show a Column



Example c: Add a column from Display Options menu



Queue sessions are shown for each queue that is setup. Graphical and Character sessions are displayed for each connected user and represent the type of device they are using.

User Id	Type	User Name	App Id	App Desc	State	Task Id	Co
▶ [+] sam	Mobile Device	Demo	RFMenu	RFgen Menus	Active	3416	11

This view appears when the dashboard is first started. As data entry devices log in, each appears in its own row.

User Id	Type	User Name	App Id	App Desc	State	Task Id
▶ [+] sam	Mobile Device	Demo	RFMenu	RFgen Menus	Active	3416
Local Resources						
▶ [+] JDE_DEMO	Active	11/29/2017 5:20:37 PM	11/29/2017 5:20:37 PM	3416		

The pointer indicates which row is selected.

The "+" icon allows you to expand the details for the selected row (logged-in device). The "-" will hide the details for the selected row.

User Id	Type	User Name	App Id
[-] sam	Mobile Device	Demo	RFMenu

To bring up the filter icon, click on the background of a column header.

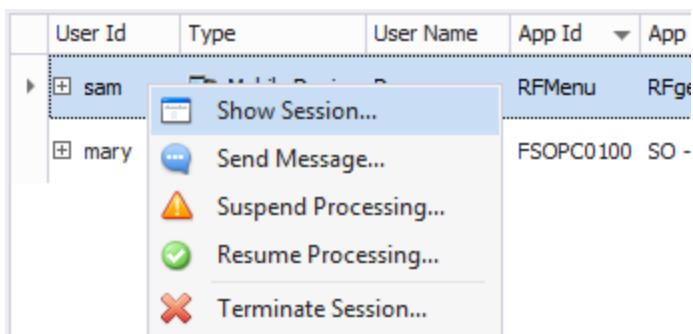
User Id	Type	User Name	App Id	App Desc	State	App
	(Custom)					
+ mary		mary	RFMenu	RFgen Menus	Active	
		sam				

You can also view a summary of the values for a given column by clicking on the filter icon at the top of a column heading.

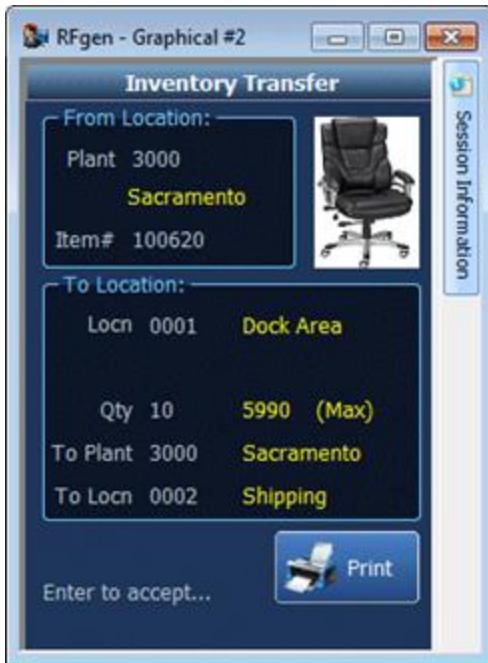
See [Display Options](#) for more details on these headings.

To Monitor and Interact with an Active Client

1. To monitor or interact with an active client session, simply right-click on the row of the device show session you want to monitor. A selection menu will appear.



2. Click on **Show Session...**. The selected client device screen window will appear on your screen.



3. While the session window is open, all activity for the device will appear in the window.

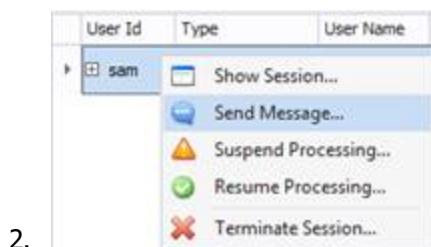
To take control of the session, click inside the screen display area and interact with the prompts. The same screen will appear on your screen. At this point the user on the client will see your actions.

To end your session, click on the "X" in upper right corner to end your remote session. The client session will continue to run unless you used the "Suspend Processing..." or "Terminate Session" commands from the right click menu.

To Broadcast a Message

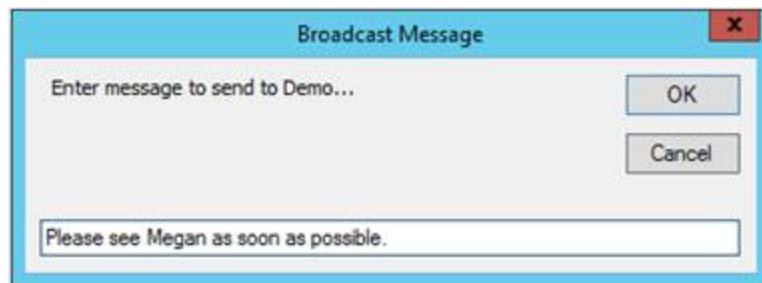
Send Message allows you to send a message to the device user from the Mobile Enterprise Dashboard.

1. To send a message to a specific client, right-click the row of a device show session and select **Send Message...** from the menu.



2.

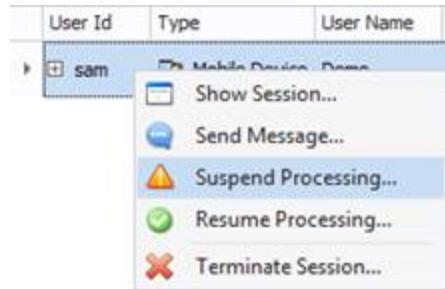
A message box will display. Enter your message and click **OK**.



3.

The client will get a pop-up message on their screen.

To Suspend or Terminate a Session

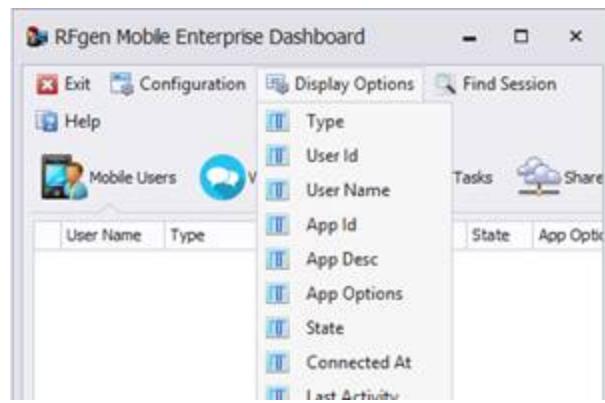


To temporarily stop this session from collecting data, choose **Suspend Processing** from the menu.

To terminate this session, click **Terminate Session**. This action will terminate the communication session for the remote device.

Display Options

The Display Options menu lists the column headings that are used in the Mobile Users, Voice Users, System Tasks, and Shared Resources tabs. To change the column headings, select **Display Options** menu and then right-click on the column heading you want to add or change. The list below describes each header type.



The Display Options details are:

Type – Type of connection. For example, a Windows Desktop connection will show up as a Mobile Device.

User ID – the user ID or operator who logged in.

User Name – The full name of the logged in user

App Id – shows the menu or application screen name currently being viewed by the user.

App Desc – the description of the current form.

App Options – any passed in parameters to the current form from the menu

State – shows either Disabled or Active depending on the suspend status of the client connection.

Connected At – shows when the connection was established.

Last Activity – shows when the very last keystroke was made by the user.

Server Name – Is the name you assign the RFgen server or its IP.

Task Id – is the process identifier of the client session executable that can be located in the processes list of the Task Manager.

IP Address – assigned IP address of the device.

GUID – This will be a GUID identifying graphical devices since in some environments the IP address alone is not enough to uniquely identify a client session.

Platform – Describes the platform of the RFgen Client, which can be: Windows Desktop, Windows CE or Mobile, Android, or iOS client.

Size – the size of the screen display used by the client's application.

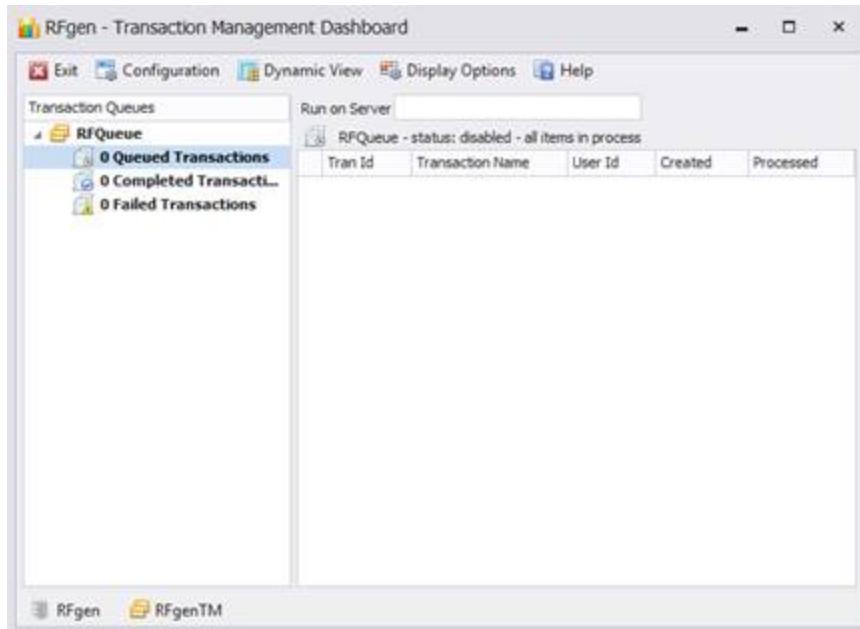
Theme – the mobile theme that is used by the client's application.

Locale – The Microsoft Locale ID value used by the client's application. For example, 1033 is English - United States.

RTL – Right-to-Left (versus Left-to-Right) setting – which is the orientation of the application for the locale of the client. For example, English is read from right to left.

Transaction Management Dashboard

The Transaction Management Dashboard is used to manage queues and queue processing. Each queue can be started or stopped individually and completed or failed transactions can be edited and resubmitted.



Three types of logs are available: "In Process" transactions are data collection entries waiting to be posted to the host application (typically because the host is offline or not available, or the batch client is offline, and cannot connect and sync up with the host server); "Completed" transactions and "Rejected" transactions may also be displayed. Transactions may be edited, reposted, marked as completed or deleted by means of the right-click menu option from the desired record.

Configuring the Transaction Management Dashboard

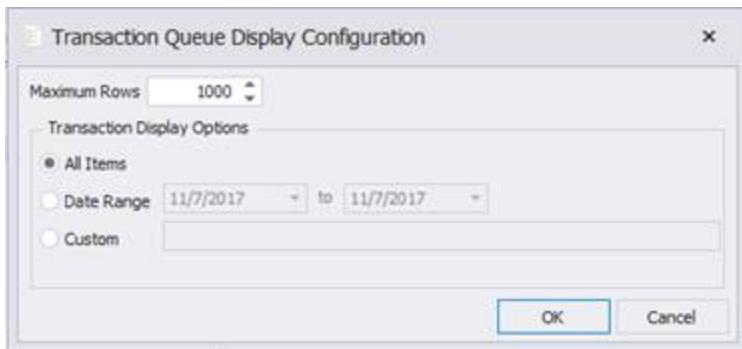


The **Configuration Menu** contains:

- **Application Preferences** - for setting your [user interface themes](#) and [locale](#).
- [**Application Database**](#) which stores the solution objects displayed in the User Management Console.

The **Dynamic View** button toggles between Static and Dynamic views of transactions. The Static View will display transactions that have already occurred whereas the Dynamic View displays transactions as they occur in real time.

The **Display Options** are used to narrow down the records being displayed in this window. Click the toolbar button on the far right to get this configuration screen.



Over time the list of completed transactions can become very large.

Maximum Rows will limit the display to the first configured number of entries. To see the most recent entries, use the data range option and set the Maximum Rows to a high value.

Transaction Display Options – All Items shows an unrestricted list of entries and **Date Range** will limit the entries to a date-based on their created date.

The **Custom** option is an ability to specify your own Where clause for the lookup. The actual names of the fields in the Queue database must be known as well as the type of field. An example would be:

where SeqNo = 1

(See *TM.GetItemsEx* for examples of table fields and types.)

The **Help** will display the online help for the Transaction Management Dashboard.

Run on Server

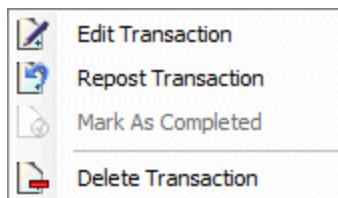


The **Run on Server** field is used to specify which server should own processing of a transaction if you have multiple servers connected to RFgen. If this field is left blank, RFgen will continue to work with all server(s) connected to it. If a server IP or "LocalHost" is entered, RFgen will work only with this server for queued transactions.

RFqueue Status Message



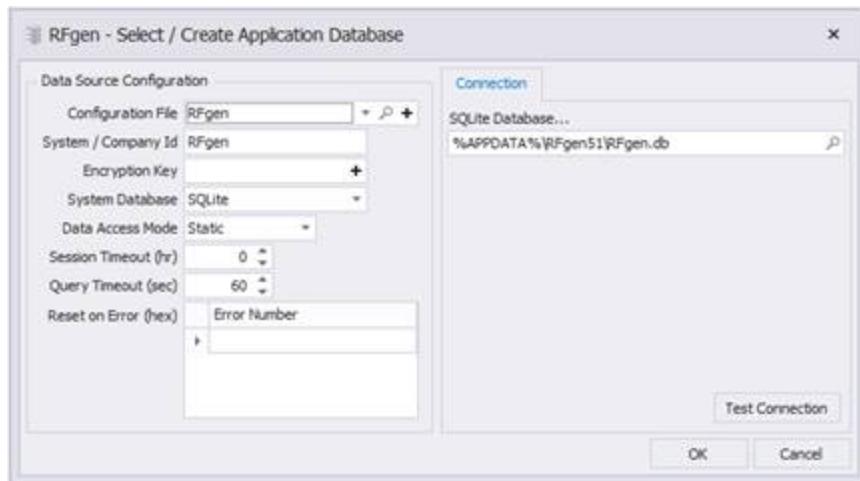
If you receive this message, your RFQueue is disabled because the Processing Cycle Time value is set to "0". To change this, go to the **Mobile Development Studio >Configuration > System Queues and Tasks** to modify the value.



The Transaction Information pane on the right side will show details about the queued transaction as well as the values of the passed in parameters.

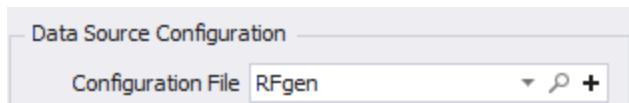
To Create or Select Your Data Source

In order to provide a database for storing and maintaining your RFgen Mobile Applications (which help run your Transactions) you need to connect to a database application/server/system to your RFgen server/system.



To configure/create your datasource, from the Transaction Management Dashboard menu bar, click on **Configuration > Application Database**. The RFgen - Select / Create Application Database Displays.

By default, a Configuration File called 'RFgen.rfc', defines the profile of the solution database, as shown below.



If you need to change the rfc file or select a different rfc file, you can use the list, search or plus (+) icons to browse to the %APPDATA%\ProgramData\RFgen51 folder.

This **Configuration File** was created when the Mobile Unity Platform software (RFgen Server) was installed. It identifies a Microsoft Access file called 'RFgen.mdb' located in the C:\Users\<username>\AppData\Roaming\RFgen51 directory as the database that contains the programming items (Applications,

Menus, Users and VBA code) written with the Mobile Development Studio, including the pre-scripted items. This is only the default place where the sample applications are deployed. It is not necessary to use this location.

The **System / Company Id** field is used to describe the owner of the configuration file. Since there may be many configuration files referencing different databases for different customers or copies of the same customer's database, this field acts as the description.

The **System Database** drop down field selects which type of database is to be used to host the solution objects. Changing this value changes the window to show database specific configuration fields. The server supports Access, SQLite, SQL Server and Oracle as database containers. The solution stores the information to connect to these databases in an "rcf" file. You can also select these rfc files when exporting / importing to that database container.

Data Access Mode sets the cursor to either Static or Dynamic when retrieving data from the database. Usually, Static is best because it is fast and safe. However, if you have a database like Pervasive that will actually make a copy of the data from the database system to the RFgen system when using a static cursor, you can change this option to Dynamic, so performance will not suffer. Internally, this sets the cursor option to either adoOpenStatic or adoOpenDynamic.

The **Session Timeout** value (in hours) will disconnect and reconnect to the database at the specified interval. This may be required if the database is configured to not allow a connection that never times out.

The **Query Timeout** (in seconds) specifies how long the server should wait before giving up on the ODBC driver to come back with a response.

Reset on Error is a list of hex values that if returned by the ODBC driver will cause a reset of the connection. The process for adding a value is to first get the error number from the error log.

Example: the error log shows -21456327. Use the Windows calculator in Programmer mode, select Dec and Dword options, enter the number and if you need the negative sign use the ± button to change its sign. Then click the Hex option. You should get: FEB89A39. Enter this value into the box with a "0x" prefix like: 0xFEB89A39

Connection

If connecting to [Access Database...](#)

If connecting to [Oracle Database...](#)

If connecting to [SQL...](#)

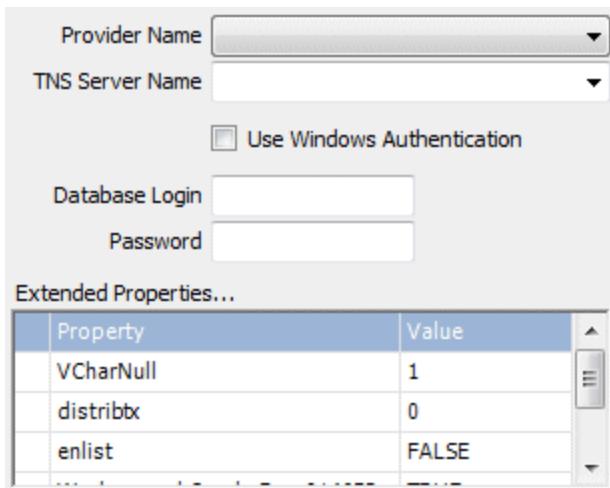
If connecting to [SQLite...](#)

Finally click on the **Test Connection** button to verify connectivity. If the database has not already been setup to support the solution tables they will be created at this time. Clicking the Save Changes button will also create what is necessary but won't test the connection. Either button will also notice if the database came from an older release and ask if you want it upgraded.

System DataBase is Access

For an Access database, select the appropriate Provider Name for the type of system (32 bit or 64 bit).

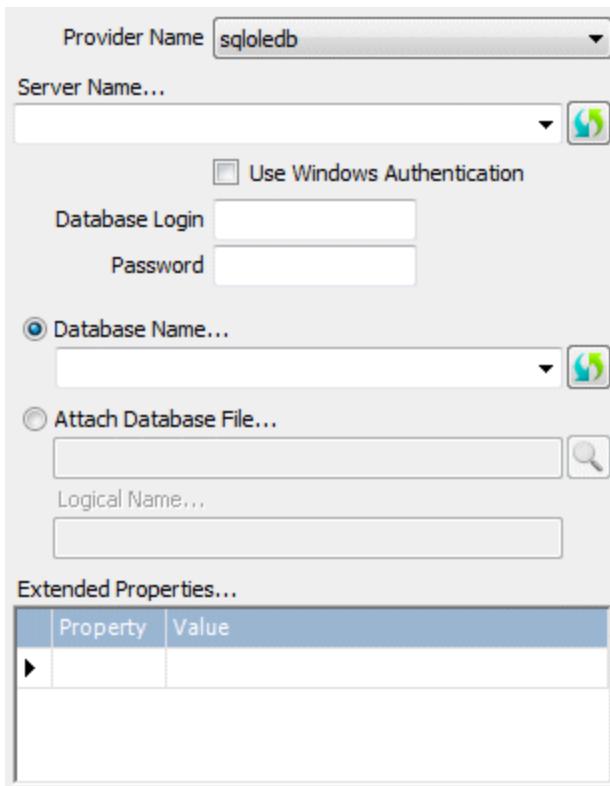
The path, login, password and extended properties are then used to make the connection. In the case of Access most of these fields are not necessary.



System DataBase is Oracle

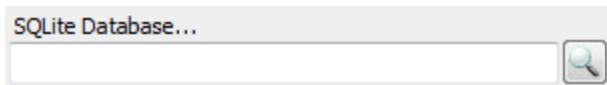
In the case of Oracle, ODBC is not used, but the TNS Server Name points to the Oracle server. Also specify the Provider Name and review the Extended Properties for accuracy. The Use Windows Authentication option will take advantage of the Active Directory when connecting to the database.

System DataBase is SQL



For SQL Server specify the Provider Name, Server Name and Database Name. The Use Windows Authentication option will take advantage of the Active Directory when connecting to the database. If you want to connect directly to the MDF file itself, specify the Attach Database File option and locate the database file directly. The Logical Name is typically the filename without a file extension and should not be necessary. The Extended Properties are usually not required.

System DataBase is SQLite



For SQLite database connections just specify the DB file itself. There are no other settings. You can specify a location and name that does not exist and clicking the Test Connection button will create the database for you.

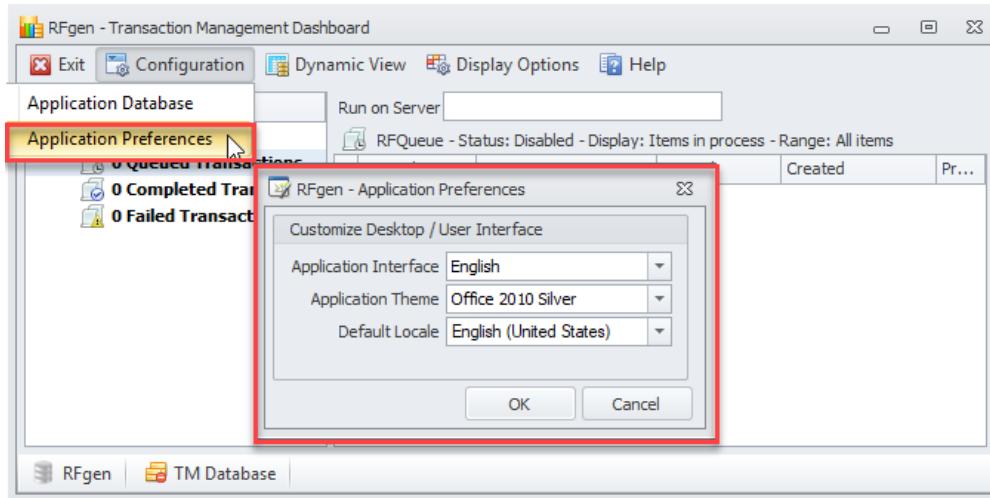
Finally click on the **Test Connection** button to verify connectivity. If the database has not already been setup to support the solution tables they will be created at this time. Clicking the Save Changes button will also create what is necessary but won't test the connection. Either button will also notice if the database came from an older release and ask if you want it upgraded.

To Change Your Dashboard's Theme

You can change the general look and feel of your RFgen Transaction Management Dashboard by changing its skin/Application Skin. This will only affect your view of the dashboard and will have no affect on the Mobile Applications viewed by end users.

1. From the RFgen Transaction Management Dashboard, click on **Configuration > Application Preferences**.
2. Select the Application Theme from the drop down menu.
3. Click **OK**. The screen changes immediately to the chosen theme.

To Set the Locale in Your Dashboard

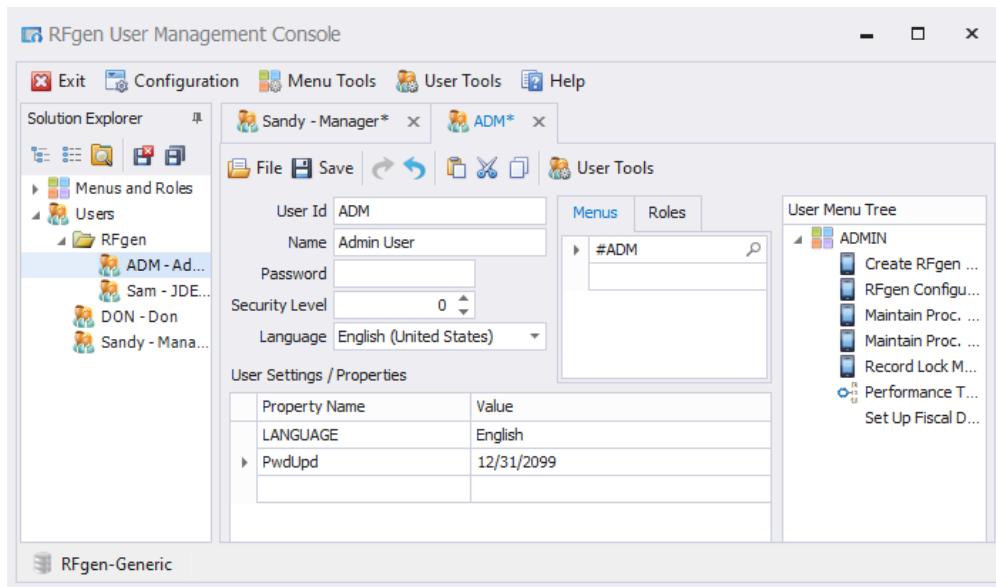


You can change the local\language your dashboard from English to any of the languages listed in the Application Interface drop down menu. This will only affect your view of the dashboard and will have no affect on the Mobile Applications viewed by end users.

The languages supported by the RFgen are: English, Arabic, Chinese, French, Japanese, and Spanish.

1. From the RFgen Transaction Management Dashboard, click on **Configuration > Application Preferences**.
2. Select the language you want to use from the Application Interface drop down menu. For example, English to Arabic. You can also set the Default Locale (i.e. specify the language used in a region such as English (United States)).
3. Click **OK**. The screen will shutdown immediately. (Or, you may need to manually restart to make the changes take place.)

User Management Console

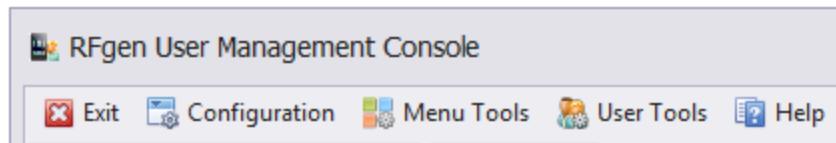


The **User Management Console** (UMC) enables warehouse managers to add, remove and manage Mobile Application users to/from the RFgen Mobile Unity Platform (RFgen Server). This allows the manager to manage users, assign them menus and specific applications to each user without having to ask the RFgen Administrator for help. Changes made in the User Management Console are reflected in the Mobile Development Studio Users tree.

If additional changes are required to applications, code, macros or resources, these changes can be performed by the RFgen Administrator through the **RFgen Mobile Development Studio**.

For an overview of how applications, menus/roles, and users are setup so to allow multiple users to have access to all menus/applications, or just a subset of menus/applications that is distributed to a group of users, see [User Overview](#). To view how the RFgen manager/developer can restrict access to specific apps, see [To Limit User Access](#).

User Management Console Menu



Exit allows you to exit the console.

Configuration Menu displays:

- **Application Preferences** - for setting the language, such as English or other locals for all applications, user interface themes (the coloring scheme of your User Management Console), and the Default Local for your User Management Console.

- **Application Database** - sets the Data Source (the database that stores the mobile applications that you assigned to users).

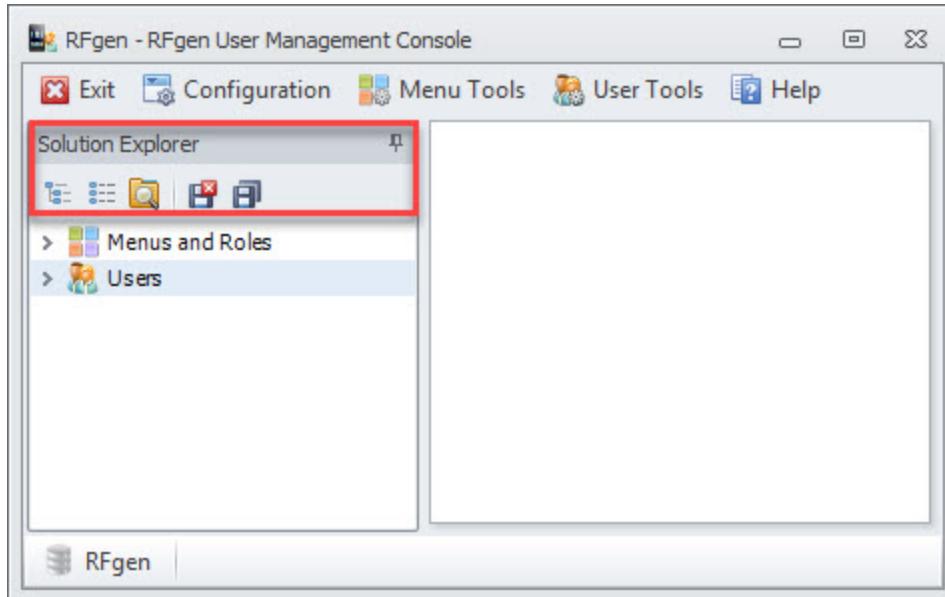
The **Menu** menus provide:

- Search for a Menu(s)
- Import and export Menus to Excel.

User Tools menus:

- Search by user attributes, then export the list.
- Import and export of users to Excel.

User Management Console Menu Bar

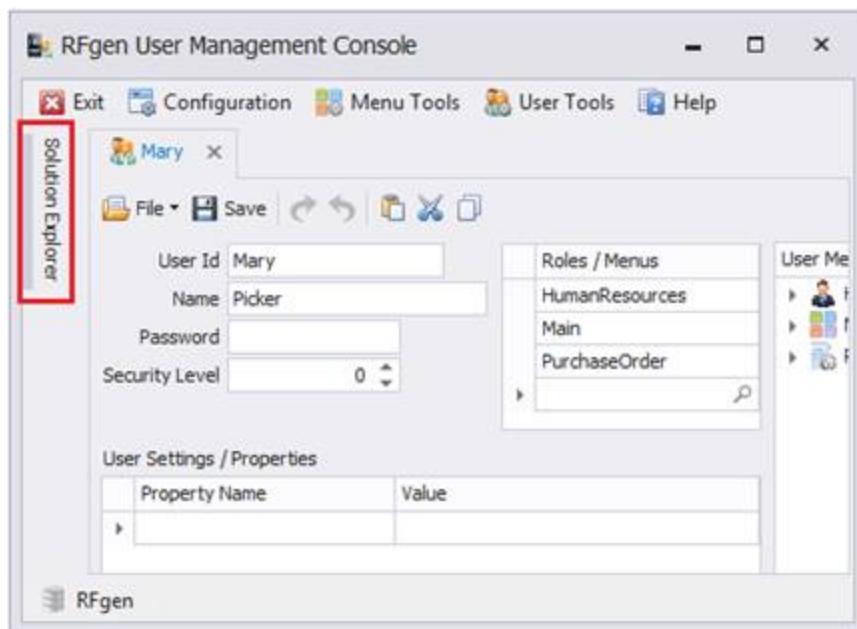


Icon Descriptions

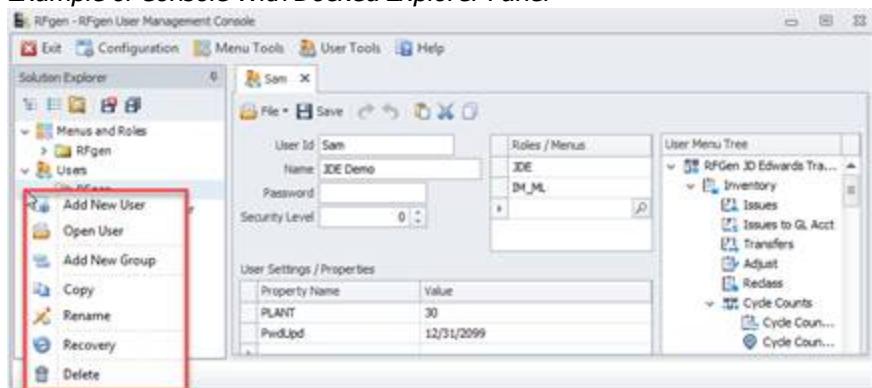
- Expands all nodes in the tree
- Collapses all nodes in the tree
- Closes all open objects
- Saves all unsaved objects
- Searches and replaces content in files
- Allows you to dock the Solution Explorer panel to the left side of your Mobile Development Studio screen and toggle the hide or show bar which then gives you more space in the Studio. When the panel is docked, you

can click on the blue bar and hide the panel. To unhide the panel, click the blue bar. To redock a panel, click the pin again.

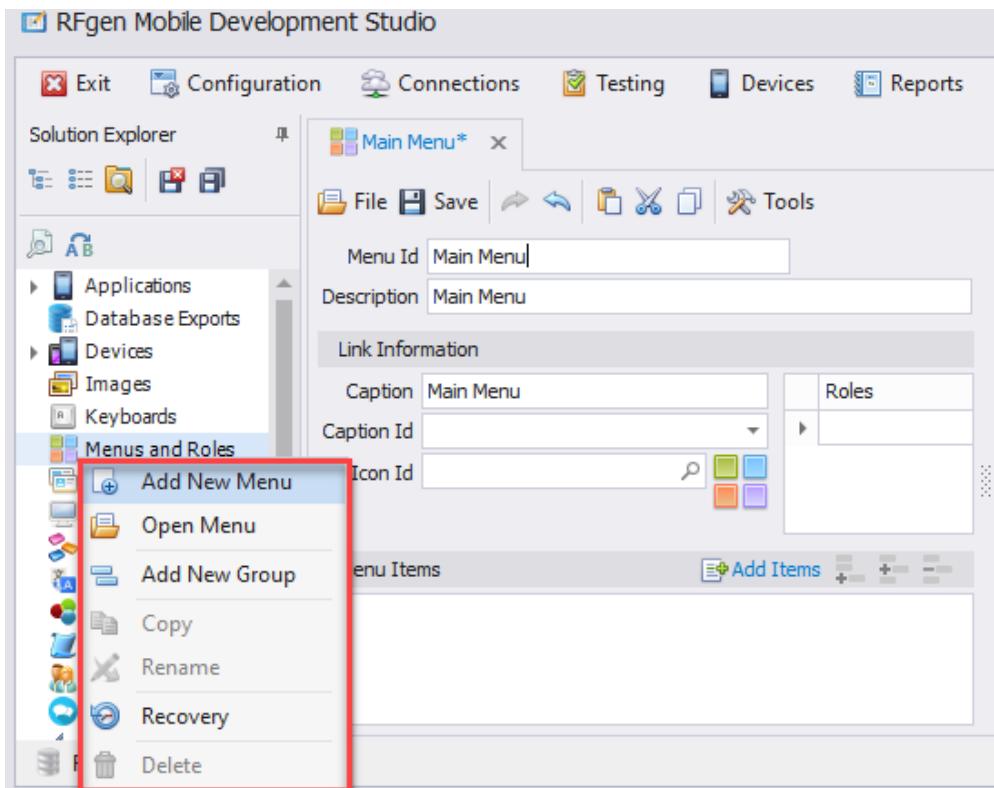
Example of Console With Solution Explorer Panel Is Selected



Example of Console With Docked Explorer Panel



To Add Menus and/or Roles



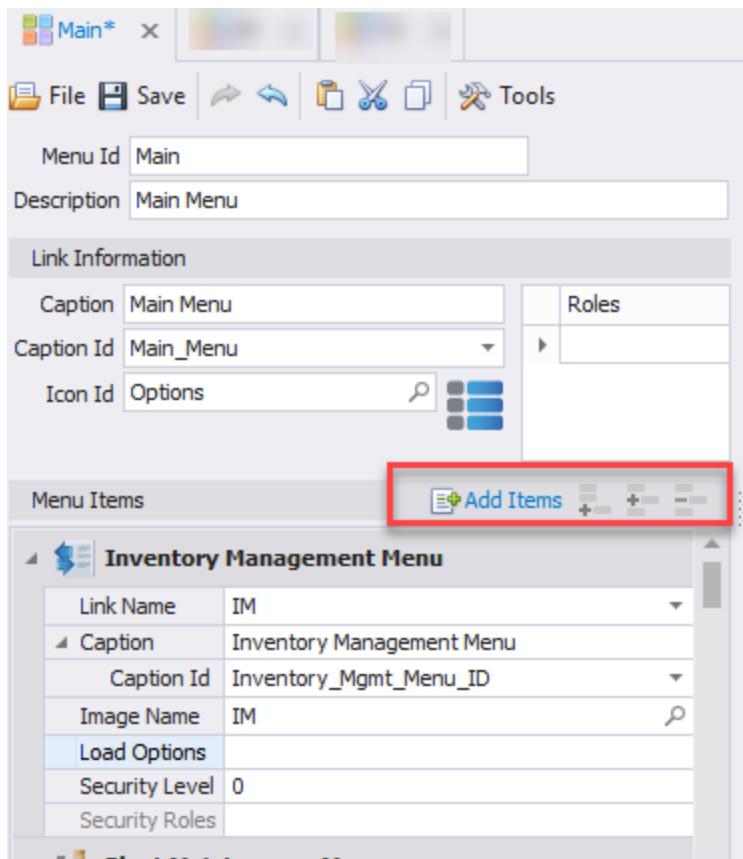
- From the from the or the **Mobile Development Studio**, in the **Solution Explorer**, right-click on **Menus and Roles** and select **Add New Menu**.
From the **User Management Console**, right-click on **Menus and Roles** and select **Add New Menu**. A blank Menu tab displays.
- Complete the Menu Id and Description. Avoid using spaces or unique characters as your Menu ID. The description accepts spaces.
- Link Information**. This section is used to link a source to the Menu Name if you want the menu name to be localized in the language of the user. This section also is used to restrict who can access the menu via the Role name that is setup.

The **Caption** is the menu name. The **Caption ID** is used to link the caption to a text ID. (To localize the string, you must have the string or Text ID setup and translated in [Solution Explorer > Text Resources](#).)

The **Icon ID** links the image from the [Solution Explorer > Images](#) list to this field. Click on the search icon in the **Icon ID** field to find and select an icon image to associate with the Main menu.

The **Roles** field is used to link a specific role (i.e. Administrator, Manager etc) with the menu. For more details on how to use setup Roles so to enable/limit access to menus and the items under a menu, see [Menu and Roles > Roles](#).

4. **Menu Items.** This section uses the **Menu Tool** icon to add the applications or another menu that will be linked to the parent menu. For example, the Inventory Management Menu is linked to the Main Menu. You can add the menu, link it to the resource id (for translation purposes) using the options in the Menu Items tool.



Load Options can be used to define how the sub menu is launched.

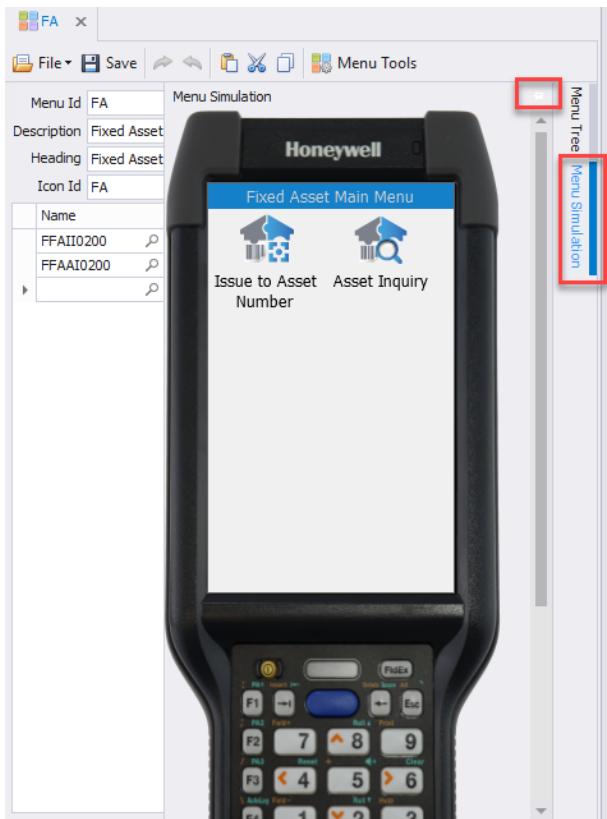
Security Level is a numeric value between 0 – 100 that will be compared to the menu's required security level before allowing that user access to the following menus or forms.

Verify your Work

Click on the Menu Tree tab to view your menu hierarchy and associated applications. If it looks okay, you can review how it may look on a device.

Menu Simulation

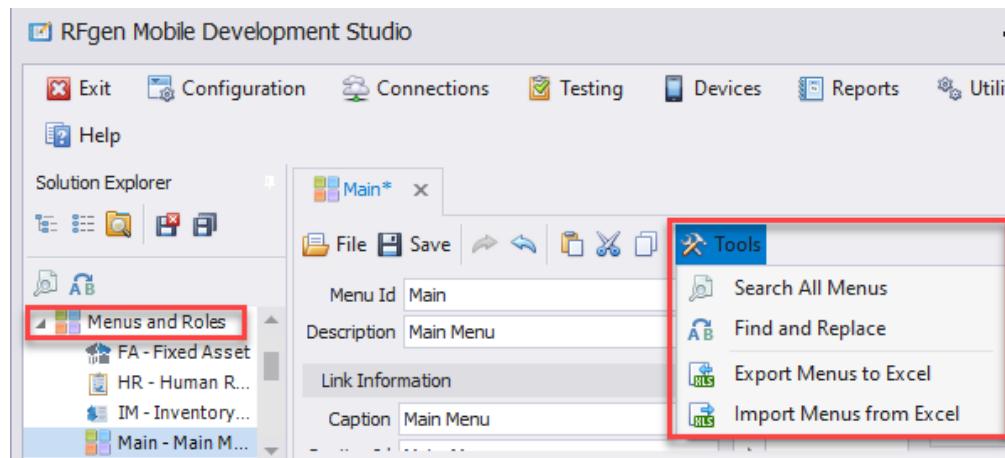
- To view how the menus would look on screen, click on Menu Simulation. For details on viewing it inside a device, see the [Menu Simulation](#) topic.



Related Information

For information on importing, exporting, editing and searching your menus, see [Menu Tools](#).

To Find and Replace, Export or Import Menus



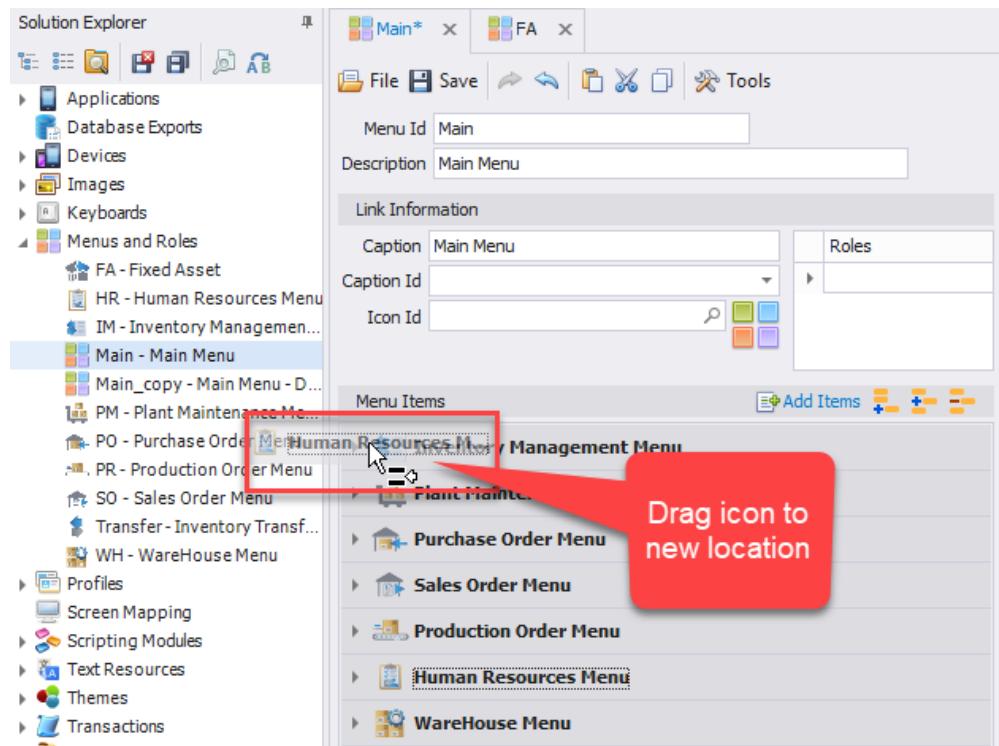
To Search or Replace a Menu

1. Select **User Management Console > Menu Tools** icon.
2. Select **Search All Menus** or **Find and Replace**.
3. Enter the filter criteria to locate the menus associated with applications.
4. If desired, you can copy your results and click **OK**.

To Export or Import to Excel

1. Select **User Management Console > Menu Tools** icon.
2. Enter the destination for the export (or import) in the **Export Location** (or **Import Location**) box.
3. Check the Menus you want to include and click **Export** (or **Import**) button.
4. A confirmation screen displays. Click **OK**. Your list should appear in the location specified.

To Remove or Rearrange Menus

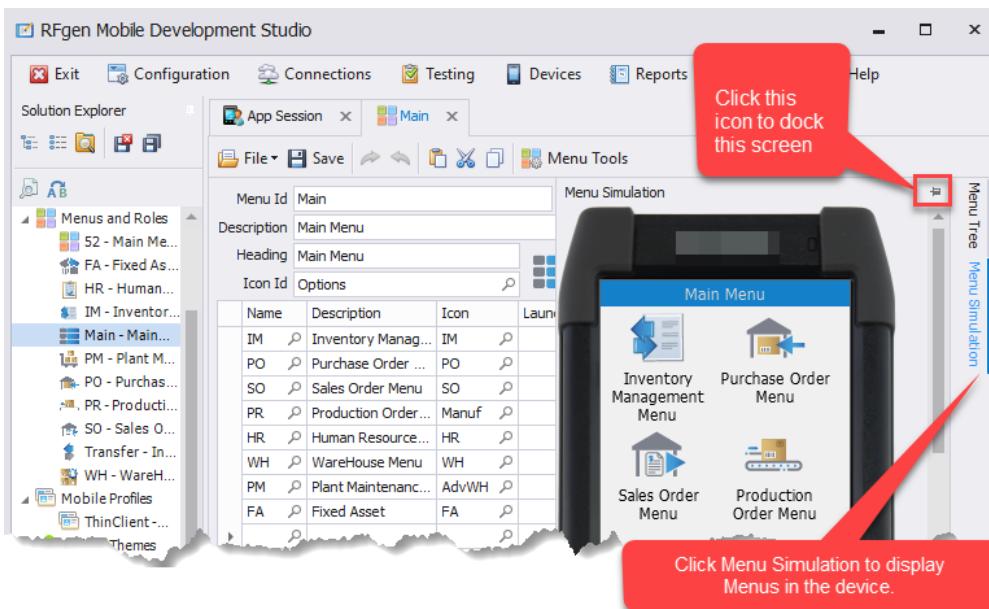


1. In the Menus and Roles screen, click on the icon or the title of the menu item you want to rearrange.
2. Drag it to its new location then release. The example above moves the Human Resources Menu to the top of the list.

To Remove a Menu and its Form

Click on the existing menu icon then click on **Delete** on your keyboard. Or, right-click on the icon and select "Delete Selected Items" from the menu. This menu allows you to add, remove or modify the menu form when you select the menu form's icon.

Menus Simulator



To view your menus as they would appear on screen, click on the **Menu Simulator** tab.

Note that you can have the menus display only in the screen or with the screen inside the device to view the full effect.

To view the menus inside a device, (called the device viewer/emulator) check the "**Show Target Image**" in the **Testing > Mobile Apps > Options** menu.

Note: If you wanted to change to a different device, use the Testing Options menu.

To dock the Simulator view, click on the tack icon in the upper right corner so it points down. If its sideways, the view is hidden.

To Create New Users

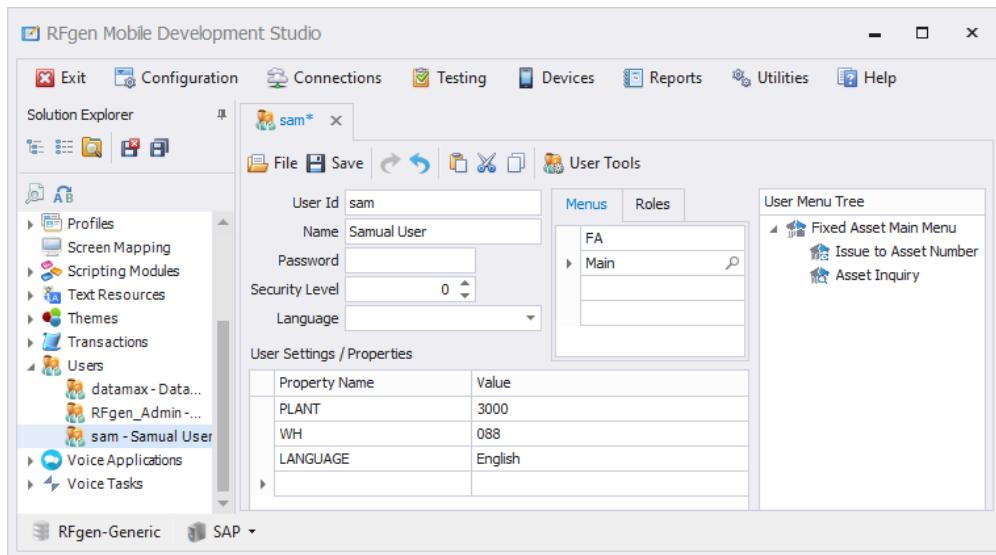
Create your menus (or roles) in Menus and Roles before you create your users.

If you want to limit access to some users (Warehouse employees) but allow access to other users (i.e. Administrators and Managers) see the topic [To limit access to specific applications in a menu assigned to multiple users](#).

1. Navigate to the **Solution Designer > Users tree**.
2. Right-click on an existing user (or in the blank space) to add a new user, or right-click on the "Users" object and select **Add New User** from the menu.
3. The [New] user tab displays. Enter the user's information.
4. The **User Id** is required, but the **Password** is optional for a user account. SAM's startup menu is 'Main Menu'.
5. The **Security Level** is a numeric value between 0 – 100 that will be compared to the menu's required security level before allowing that user access to the following menus or forms.
6. The **Language** is used to assign a locale to the user's session. This field is optional.

7. The **User Settings / Properties** field is an advanced developer feature that is used to associate data values that are used repeatedly with the individual. For example, if Mary works in a specific warehouse, and you want her login to be associated with that specific warehouse (i.e. Plant ID: 3000), the information entered in the Property Name and Value fields will associate that Mary with plant 3000 so she does not have to enter the id "3000" when interacting with an application that requires a plant ID.
8. Continue with **To Assign Menus to a User**.

To Assign Menus and/or Roles to a User

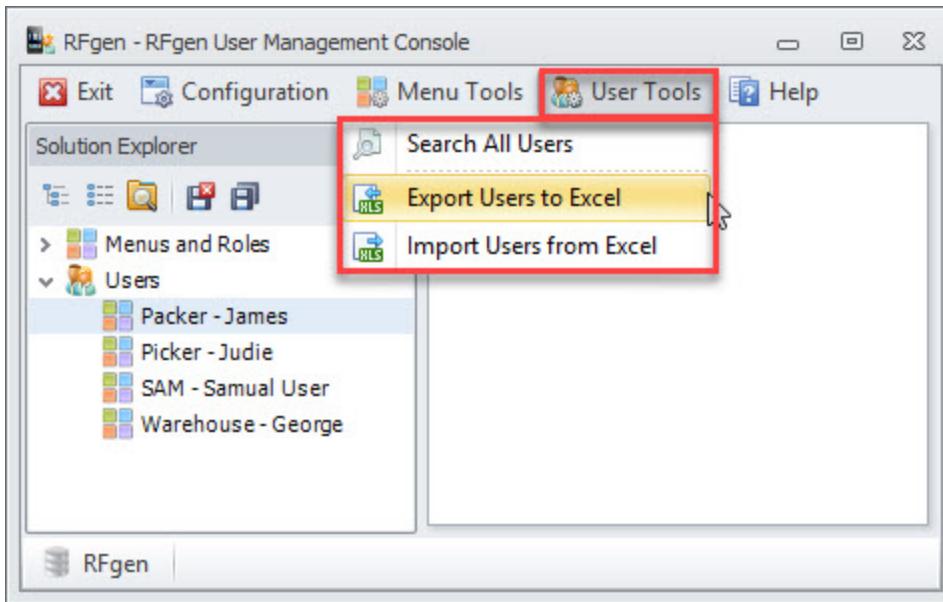


1. From the **Users** tree, select the user you want to work with if the user profile isn't displayed already.
2. In the **Roles/Menus** table of the **Users** tab, click on the Find icon and check the menu item to be added.
3. Click **OK** when done. The selection appears in the User Menu Tree (far right panel).
4. Click **Save**.

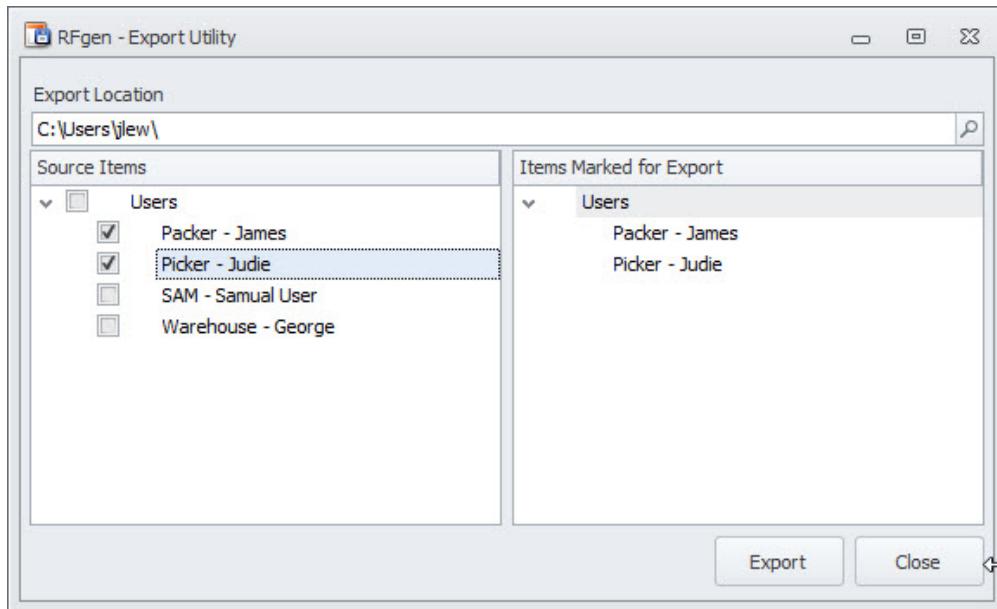
To Remove a User

You can remove a user by right-clicking on the user and selecting **Delete**.

User Tools: Export or Import Users

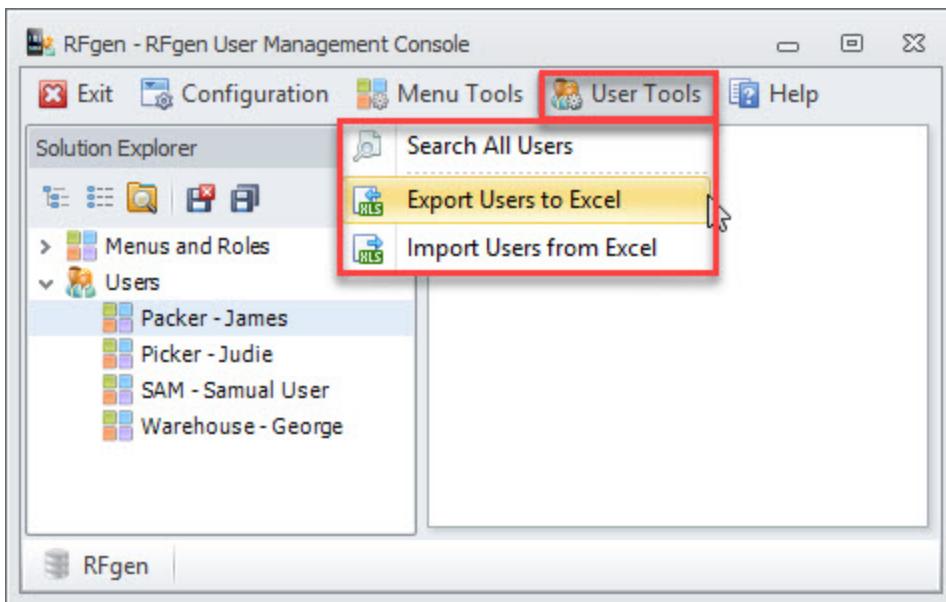


1. To export or import users to a Excel spreadsheet, click on the **User Management Console > User Tool > Export (or Import) to Excel** option. A **Export Utility** screen displays.
2. Enter the destination for the export (or import) in the **Export Location** (or **Import Location**) box.
3. Check the users you want to include and click **Export** (or **Import**) button.



4. A confirmation screen displays. Click **OK**. Your list should appear in the location specified.

To Search Users



1. To search for a user/users, click on the **Users > Tool** button. The Search All Users Utility screen displays.
2. Click in the **Search Mode** and select the desired search type from the drop down menu. This filter allows you to search users if you know their name, or security level etc.
3. Enter your search information in the **Text to Find** box. You can also select the option to match the upper or lower case text and by the whole word. Click **Find**.
4. If the information is found, it displays in the area below. From here, you can then copy the list to Excel.
5. Click the Excel button and enter the destination location. Click **Save** when done.

Appendix A - Graphical Control Property Definitions

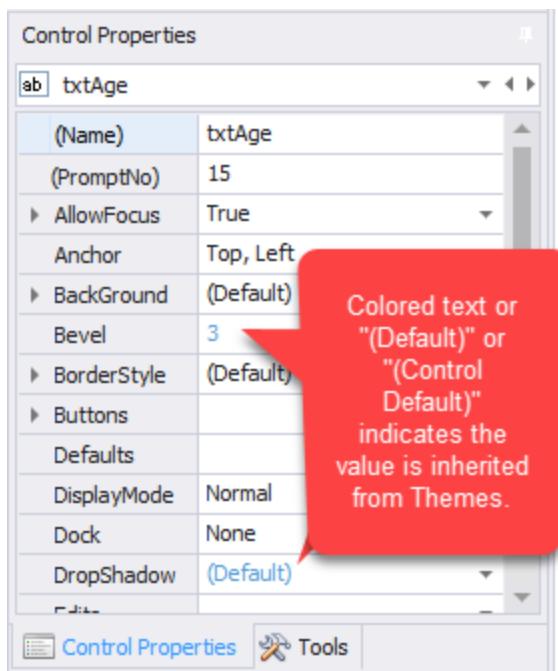
Graphical Control Properties

Descriptions for all the control properties are listed alphabetically on this page. Tap **Ctrl + F** keys to **Search** for a property definition on this page.

The **(Name)** property is a standard property for all controls except the Form.

The **(PromptNo)** property is the Prompt sequence value RFgen assigns to objects placed on a form or a page. If you move the location of a control or page, the PromptNo also updates, or you can edit it directly in the Application Designer Control Panel. If your script uses App.PromptNo VBA extension, it should return the PromptNo value for the specified control.

How to tell if Themes is being applied



Colored text or
"(Default)" or
"(Control
Default)"
indicates the
value is inherited
from Themes.

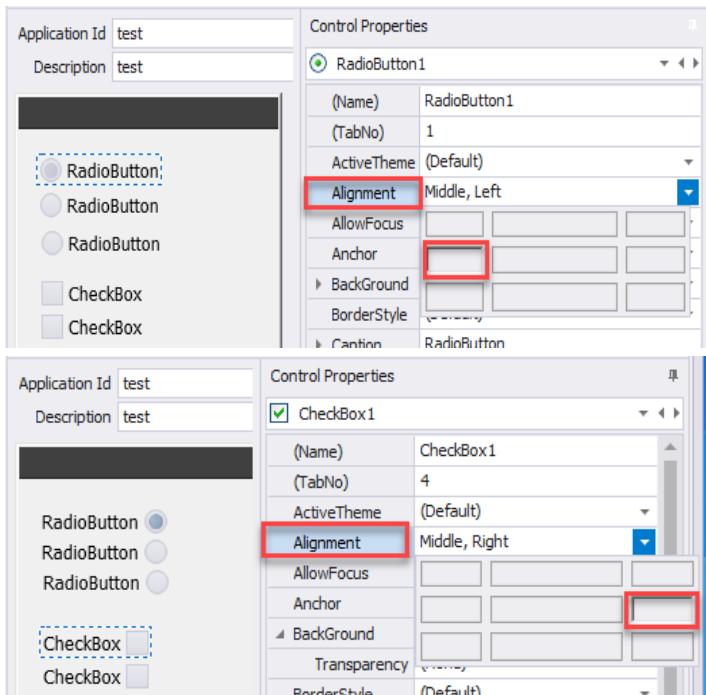
If the value for a property appears in [colored text](#) or the value shows [\(Default\)](#) or [Control Default](#) this means the property value is using the corresponding value set in Themes.

Control Property Descriptions

Not all properties appear for all control types. For example Image controls, CheckBox, RadioButton, Page controls and others will have their own unique properties.

The **Action** property under the **Button** property provides built-in operations for buttons in controls like the TextBox. (For example, Button > Action = Search and Button > Image = Search can be used with the OnSearch event to display a list of items created from script.)

The **Alignment** property places text and images relative to another object within a control. For the CheckBox and Radio Button controls, the images are shifted relative to the labels when you select a position in the Alignment drop down menu. (See example below)



For the Button control, Alignment positions the Caption text and the image within the button space. To separate the image from the Caption text, use the Image > Alignment to position the image relative to the Caption text. For the Image control, the image is positioned relative the edges of the image box.

The **AllowFocus** of set to True, enables the function/operation described below. "True" also enables you to select actions for focus. False prevents the control from receiving focus, removes a TabNo assignment or prevents one from being assigned. If set to True, you can enable these options:

- *AllowBackup*- to the previous control;
- *AutoSelectText*- enables selection of text where the prompt lands;
- *EraseOnBackup* - will erase the text if user backs up
- *FocusOnClick* - enables focus via a mouse click in addition to tapping a Tab key;
- *OnEnter* - *Advance* - dropdown option for: **Exit** the form, **Hold** the focus, **Reset** the form (i.e. Clear entries), or **Submit** when the user taps the Enter key.

The **AllowRoll** property is used on controls that contain long lists of items and helps you "jump" to items faster. For example if you press the up arrow while on the first line in a long list of items, the last item from the list will display. Or, if you are at the bottom of a list, and press a down arrow, you are taken to the top of the list.

The **AlternateItem** property group sets the background color, font color (ForeColor), visibility of the colors for alternate rows in a list (ComboBox, DataGrid, ListBox, PanelList etc.). To display the alternate color, set Visible to True.

The **Anchor** property sets a child object's position and dimensions relative to where its anchored to the parent. For example if a TextBox control was anchored to a Form on all four sides, the TextBox would be proportionally resized if the Form was displayed on a different mobile device. But if the TextBox was Anchor equaled "Bottom", "Right", then its size won't change, and its position will stay relatively the same. If no other anchors except the Bottom is set, then RFgen automatically centers the child object inside the parent. If the parent is resized, the chile object's size is not changed even if the parent's size changes.

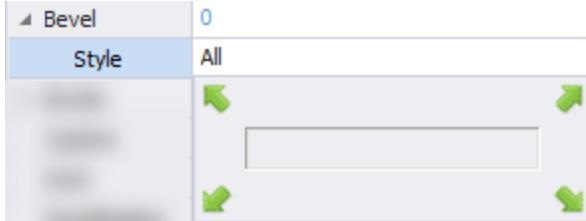
The **AutoSize** property will stretch the object's background to the lowest and right-most portion of the screen, depending on the option selected. This property is available for the Button, DateTime, Frame, Label, and Panel controls. The values are: (None), Content, Height, and Widthl. If AutoSize is set to "(None)" then you can drag-n-resize the control. But if the control is in AutoSize mode, the ability to drag-n-resize the control is disabled and the control will be sized in accordance to its contents. For example, if you have a Panel control that was empty (no controls inside it), and AutoSize was set to "Contents," "Height," or "Width," then the Panel will automatically resize to 0 pixels. If there was an image in the Panel, and AutoSize was set to "Contents" then it would resize the Panel around the image relative using the right-most and bottom portion of the panel as the start location. See the Size property for the size in pixels. See Manage Paged Collection for details on AutoSize Property for Pages.

The **BackColor** and **BackColorAlt** (previously called **BackColor(1)** and **BackColor(2)**) properties are used to create either solid backgrounds or gradients depending on the option chosen in the Background Fill property. The color can be set from using Custom Color tab or by a 6-character hex value (which gives you 16 million colors to choose from).

The **BackGradient** This property was replaced by the **BackGround** property.

The **BackGround** property has three categories for setting the background color of a control. If set to Solid, RFgen applies the value from Color 1. If a directional values such as Diagonal Right, Diagonal Left, Vertical etc. is selected, RFgen applies the values from **Color 1** and **Color 2** to create a gradient background. If Transparent is selected, the next-closest background color behind the control will be used. For example, if your Mobile Theme Application BackGround = Red, and your Mobile Theme Label BackGround=Transparent, and your Label Caption color = White, then your Label text will appear as white on a red background in Mobile Themes. In the application designer, the Label text will also appear as white text on a red background if the application's Form Active Theme = (Default) and the Label Active Theme = (Default). But, if your Mobile Theme Label BackGround = Solid and the colors 1 and 2 = Pink, then in the solution designer, the label's background will appear as white text on pink because uses the Label's solid background color blocks the background color from the Application Form.

The **Bevel** (graphical mode only) property sets the curvature of a square's and rectangle's corner edge where 0 is no bevel and 100 turns the object into a circle or oval shape. The **Bevel Corners**



The **BindToColumn** is only used with data-centric controls on PanelRows in the PanelList control. Its used to bind a data-centric controls such as a Textbox or Label to a specific column. For example, if your first column

is 1, then the **BindToColumn** value should be "1". The list values can be ordered to start with 0 or 1; You can force the ordered list to start with "1" by checking the box under Configuration > Environment Properties > Environment > List Items Collection is One Based.

The **BorderStyle** (graphical mode only) property controls sets the style of an object's border. The (Default) is Flat. Options are: None, Flat, Sunken, Raised, Thick, and Underline.

The **BorderStyle** (graphical mode only) property controls the border of the prompt. Options are Standard, Active Border, No Border, Visible with Focus and Transparent.

The **Brush** property is used to select a color that will override a Theme color.

The **Button** property group stylizes the search icon button that displays inside the TextBox when the OnSearch event is set to True.

The **Button Pressed** has been obsoleted in 5.2. This property was used to set the color values for this control when its selected. In 5.2 all press states styles (colors) are now calculated from the colors assigned to the button so there is no need to configure them.

The **ButtonSize** property was removed in version 5.1 of RFgen. See the **Size** property.

The **ButtonStyle** property was obsoleted in 5.2.

The **Button Pressed** property was obsoleted in 5.2. All press states are now automatically calculated to apply a selected appearance using the BackGround coloring of the button.

The **Caption** property is used to hold text that is static or dynamic.

Note that the **TextId** property under Caption is used to populate the contents of the Label or TextBox at runtime, depending on the value that's used. For more details, see the TextId property description.

Caption - LinkTo property - used to mirror the text between the textbox and one or more labels after the user taps Enter. This is accomplished by using the same ID in both controls. To enable/disable the **DataLink** property must be set to True.

The **CellMargins** property sets the distance between the contents of a cell and the cell border in pixels, and is used to make it easier to view multiple lines of text in a control.

The **Checked** property sets the status of a CheckBox prompt.

The **CheckBox** property group is used to stylize the elements in the CheckBox, ComboBox, DataGrid, ListBox, and TreeView controls. To enable the checkbox as a subfeature in controls used for listing text and values in columns or grids, you must select the "CheckBox" value in the control's Column > (Style) property. The CheckBox property group includes the BackGround, BorderStyle, ForeColor, Margins, and Size (for sizing the checkbox).

The **Colorize** property, if set to True, converts simple grey-colored images to the color selected in the **Brush** property. For example, the **Colorized Brush** can change the grey in a grey-colored Chevron icon to  to red . This tool is intended for the system icons that are provided by RFgen in the InlineButton property.

The Colorize Brush is NOT intended for changing the color of images or icons comprised of multiple colors as it will apply the same color to entire image.

The **ColumnSet** property group (available in the ComboBox) is used to set the (template) of a column and the styling of data each column created within a column set.

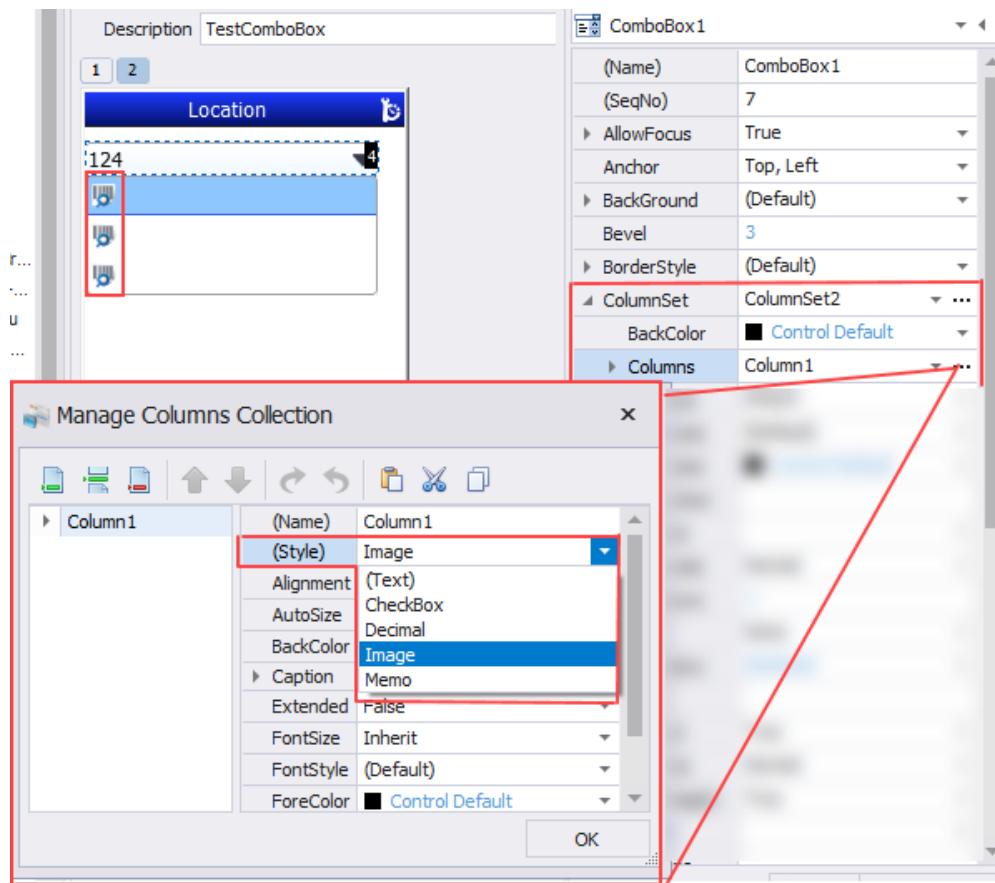
a) If you click on the down arrow, a lists of your column sets displays so you can select the one you want to edit;

b) If you click on the ellipsis (...) it opens the **Manage ColumnSet Collection** window and is used to add, insert, delete column sets and set the colors and fonts for each one.

Once your columns have been added and stylized, the subproperties and values will be listed under columns. For more details, see *Manage Columns Collection*.

The **Columns** subproperty group (under ColumnSet property) is used to customize and control the presentation of the data within a specific column. For example, if you can set column 1 to contain CheckBoxes with white text on a black background, column 2 to contain text only with a grey background and red text, and restrict data via TrimSpaces property.

The **Columns (Name)** property sets the unique that for the column for reference purposes.



The **Column (Style)** lists factory-provided styles: *(Text)* for display of read-only values, *CheckBox* for selection of data items, *Decimal* for display of numeric text, *Image* (icon), *Memo* for data entry by the user, or the *SpinEdit* button to add/decrease a value. **TrimSpaces** will remove leading and trailing spaces from the data so column alignment will be smaller. The **Width** property can either be set to a specific size or -1 to indicate that the column should stretch to the right taking up any available space.

The **Color 1** and **Color 2** properties are used in combination to create gradient colors in properties such as the Background property. These values can be set in the Custom, Named Color, or Theme tab of the color palette tool.

The **DataLink** property enables the mirroring of data from a source object (i.e. TextBox) to the destination label. The source is set in the **Caption > LinkTo** property. False disables the link between the source and destination label.

The **Defaults** property sets any number of built-in values or custom values as the initial value of the prompt. For more details, see [To Set Text Defaults in a Control](#) and for a list of text default options, refer to the topic [Text Default Options](#) in the Developers Reference Guide which can be downloaded from the RFgen Online Help and Documentation Page. The Defaults property appears for property groups such as EditText which are typically in the TextBox and Memo control.

The **DefaultList** property sets any number of built-in values or custom values as the initial value of the prompt. For more details, see [To Set Text Defaults in a Control](#) and for a list of text default options, refer to the topic [Text Default Options](#) in the Developers Reference Guide which can be downloaded from the RFgen Online Help and Documentation Page. The DefaultList property appears in the ListBox control.

The **Display** property. See the [Form Display property](#).

The **DisplayMode** property provides a list of factory-provided styles (from Themes) that can be used to standardize the appearance of objects.

The **Dock** property is used to automate the placement and scale of sibling objects under a parent object. While the Anchor property is also used to position and set sizing relative to whether its anchored, the docking property forces sibling to respect the space of neighboring objects so you can dock items in a series.

For example, if a Label and TextBox were placed above a Button control and all three were on the same page, and their dock properties = Top and Fill, RFgen would first align the Label and TextBox to the top of the screen and space them out to fill the screen's width. Then RFgen would position the Button control to fill in the remaining area of the screen.

The **DropShadow** places a dark border around the bottom and right sides of the control for a 3D effect.

The **Edits** property sets any number of built-in values as the requirement for the entered data. For example, it can be used to validate the data entered manually or data that was scanned to this field. This property is usually a member of the TextBox control or Memo control EditText property group. For a list of Edit Property Options used to validate text or perform other checks against a source, see [Edit Property Options](#) in the RFgen Developers Reference Guide.

The **EditText** property sets the color of the text (i.e. text inside a memo) that a user can edit at runtime. EditText may contain these subproperties used to stylize text at runtime: Alignment, FontSize, FontStyle, and Margins. It may also include the Edits property for inserting values that perform validations against the data entered in the prompt.

The **EntryRequired** property, set to True, forces users to enter data into the prompt, while setting it to False, allows users to skip the field. If the prompt never gets the focus, this property will not get used.

The **ErrorMessage** property is text displayed as an App.MsgBox when the data entered fails to meet the criteria in the Edits property. You can link this to a text resource if you enter the **TextId** which is stored under Solution Explorer > Text Resources.

The **ExtendedColumn** property specifies which column will be stretched to the right edge of the control. The default is the last column designated by -1 but specifying 1, 2, or 3 as examples would use the remainder of the width by stretching a middle column.

The **Events** section contains a list of Visual Basic for Applications (VBA) scripting events. The default values are set to False. If you select True, the script view for your application displays and the selected event object's script module is added. For a description of an Event, see the [VBA Events](#) topic or enter the Event's name in the Search field above. If Events is hidden, click on the Show Scripting Events in the [Options Menu](#).

The **FocusStyle** property specifies how an object shows it is the active/selected object at run time. The default is Standard. Other options are: Active Border or Visible w/Focus.

The **FontSize** property (graphical mode only) defaults to a baseline value called "Normal" which is set in **Mobile Themes > Application > FontSize**. You increase or decrease the font size (in points) if you want to override the default value. For example, if your Normal = 11 points but you want your Labels > Caption text = 14 points, you would set your Labels > Caption > FontSize to +3. In controls where the FontSize property is nested into subproperties, you may see "Inherit" used instead of "Normal".

The **FontStyle** property (graphical mode only) sets the prompt's data field display to a particular style. The default is Normal. Other options are combinations of Bold, Italic and Underline.

The **ForeColor** property (graphical mode only) allows the user to select from a color pallet or enter a 6 character hex value (for 16 million colors) to set the fore color of the label caption of the prompt. For the controls such as the Button control, the ForeColor property will also apply to the icon, if the Icon: Colorize value is set to True.

The **Format** property is an extension of the VBA Format command and pre-formats the entered data to the mask entered here. See the VBA Format command for examples. The double quotes are not necessary as they are in the VBA Format command.

The **FrameStyle** property lets the user create rectangles, vertical or horizontal lines for the frame control only.

The **GenerateMember** property was removed in RFgen 5.2. This property helps improve an application's performance by telling RFgen what it should or shouldn't generate. The values are: *None*, *Member Only*, *Event Only* and *All* (which is Member + Event).

None will NOT generate a member variable; *Member Only* generates a member variables with no events; *Events Only* generates object events only, but no member variables; and *All* generates both member variables and object events. The default for prompts like the Textbox and Labels is "Member Only," whereas prompts like Buttons, DataGrid, CheckBoxes, ComboBoxes, Maps, MenuList and Signature will default to "All."

The **Heading** property group consists of additional properties that are used to stylize the header of a control. You can turn on/off the display of the header (Visible = True), change its BackGround, BorderStyle, add text/caption to the Heading field, and set the FontSize, FontStyle, FontColor, margins, and associate the text/caption in the Heading field with a text resource ID for translation purposes.

The **Heading Caption** property group includes: Color - sets the caption text color and color of any characters used in the form header. FontSize and FontStyles set the caption text font style and size. The Margins (left, right, top and bottom) are in pixels and set the distance between the caption edge of the header. For example a margin of 20 pixels would make the header bigger.

Heading Icon Property Group - is used to add icon buttons and stylize the icon in the header. Icons are added for navigation or menu access purposes. The position of the icon(s) is based on factors such as how many icons were added, icon size, margin and padding allocated to the icon, and positioning (alignment setting) of the header text.

- Heading Icon: Margins - TBD
- Heading Icon: Pressed - TBD
- Heading Icon: Size - The width and height of the icon in pixels.
- To add an icon - TBD

The **Image** property group is available in the ButtonList control and Menu control. For the **Image control**, the Image property group is used to select an image resource, select its alignment within its container on a form, page, or inside another control, and set how its proportioned when its resized. For more details, see the [Image Control](#) topic. For all the ButtonList, ImageList, and Menu controls, the Image property group sets the size and margin for the image(s) listed in these controls but does not allow selection of an image resource. For the **Button control**, the Image property group is used to select an image resource, set the alignment, margins, and size within a button.

ImageID has been deprecated in 5.2. In 5.1 it was used to select an image resource.

ImageMode has been deprecated in 5.2. In 5.1 it was used to position the image (i.e. Top-Left, Top-Cener, Top-Right, Disabled, Stretch, or Tile)

ImagePath has been deprecated in 5.1. In 5.0 this property set the file path to an image located on the hard drive instead of the Resources > Images folder.

ImageSize has been deprecated in 5.1. In 5.0, it set the width and height of the graphic itself regardless of the size of the control. Images can be displayed a number of ways and this property sets the image size for graphical lists, button or desktop menu lists.

InLineButton property group - Is present only for the ComboBox, Map, SpinEdit, and TextBox controls. In the TextBox, this property group enables you to add a customized button that is associated with an event and stylized how the icon looks. For the other controls, it simply stylizes the buttons that come with the controls.

InputState puts the prompt in a state where its "ReadOnly" (no data can be inputted to this prompt). "Disabled" also prevents the prompt from receiving inputs. "Normal" allows inputs (i.e. write data) to the prompt.

The **IntegralHeight** property dynamically sets the height relative to the fontsize it contains and prevents manual changes to the height of the control if the IntegralHeight Property is set to True. For example, the TextBox height will change relative to the TextOptions FontSize if you change the FontSize from Normal to +32. You can change the TextBox's location and width it if you want to.

If you IntegralHeight to False, the TextBox size does not size automatically when you change the TextOptions FontSize, but you can manually make the TextBox taller and change its location and width it if you want to.

The **Items** property is used with the ButtonList, ComboBox, Menu, and other controls that list items. It provides a set of subproperties for styling the elements in a group -- some of which may not be used. These subproperties are: BackGround, BoarderStyle, ScaleText ("True" scales with the size of the item; False uses the Text FontSize), Separation (distance in pixels between items), and Size (height and width in pixels). See also "Selected Item."

The **Keyboard** In controls where user entries are accepted (i.e. Memo or TextBox control), this option can be set to bring up a soft keyboard for input when the text box gets the focus. Note: Keyboard characteristics are set in Solution Explorer Keyboards.

The **KeyField** property is for linked textboxes only and designates which prompts will be used as key fields when attempting to perform an internal SQL Update statement for the linked application. This property is automatically filled in when the user downloads a table or view structure and links the application to that structure.

The **Layout** property is added to a child control when the child control becomes a member of the Layout control. The subproperties are: Col, ColSpan, DockingMode, Row, and RowSpan.

The **LineColor** property selects the color of the lines between rows or columns in a control that supports multiple rows or columns.

The **LineSize** property sets the thickness of a line in the LineControl.

The **LineStyle** property is used for list type controls that also use the Columns property. As the name implies, it makes horizontal and/or vertical visible or keeps them hidden.

The **LinkLabel** property links TextBox entries to be the specified label so that the entries are mirrored to the label at runtime.

The **LinkToPages** property is available under the Columns property and Rows property of the Layout control when the Layout control is on a form. If the Layout control is on a page, this property is not present. It creates a link between the objects in the column or row of the Layout control and the pages of a form. For example, if you linked Row 1 to pages 1 through 3, you would see the contents of Row 1 on pages 1 through 3.

The **ListData** property is for list boxes, combo boxes and list views only and contains a collection of values to be assigned to the prompt when the application loads.

The **ListHeading** property allows the code environment to overwrite the caption of the prompt with formatted data from a database lookup using the Prompt.List.SetColumn method.

The **ListHeight** property is for combo boxes only and sets the number of rows the control will use when displaying a list of possible values.

The **ListSorted** property is for list boxes, combo boxes and list views only and keeps the contents of the list sorted.

The **ListStyle** property changes the presentation of the data displayed between a Standard text list, an Image List that uses images next to the text description, Buttons or Desktop style like a Windows desktop. This is the control used on the internal RFMenu form.

The **Location** property sets the position of the control in pixels for graphical applications. The location is relative to the parent container and may have different values for different controls. For example the Layout Control location is identified by the number of pixels from the top-left of a form. But if a button was inside the Layout control, the button's location , whereas a and in rows and columns for fixed-length character applications.

The **Logo Property Group** is used primarily in [Mobile Themes - Applications](#).

The **Manage Collection** property. See Columns Property.

The **Manage Columns Collection** is available for the **Columns** property in the ComboBox, Layout, ListBox, and DataGrid controls. Its used to add, insert, and remove columns. In controls used to list text and values, the Collection helps you design and stylize columns. You can stylize text and the control's background color, set your alignments, set the caption, enable or disable "TrimSpaces," and allow the columns to be extended, formatted, and visible (or hidden). In the ComboBox or Layout controls , these properties are not present.

In the Layout control (used for containing other controls) the Manage Columns Collection is designed to add, insert, and remove columns and assign column names and SizeModes (values that set how the object in the cell of the Layout control will be sized), and whether the column is hidden or visible. For more details, see *How to use the Layout Control*.

The **Manage Displays Collection** property. See [Manage Displays Collection](#) topic.

The **Manage Icons Collection** property group is used to add or remove icons, and if in a list, set the list order.

The **Manage Rows Collection** property. See Manage Columns Collection.

The **Margins** property is used to pad the spacing between the rows or images of the displayed data.

The **MastInput** property (available only in the TextBox control) is used to mask the input with asterisks if the value is True. The default is set to False.

The **MaxWidth** property is used to set the maximum space allowed for the control. For example, the Radio Button maximum width is the widest space allowed for the button and its text label.

The **MenuIcon** property on the Form is used to provide a background image.

The **Multiline** property is now called "TextOptions". For details, see "TextOptions."

The **(Name)** property is the internal name / identifier of a control/prompt. Tip: As a best practice, follow the Hungarian notation where textboxes are named 'txtPart' and list boxes are named 'lstParts' as examples. This way, when referring to them in the script, there is an inherent understanding of what types of data will be used for the prompt.

The **NormalizeText** property will trim the spaces from both sides of the displayed data or captions of the buttons or desktop icons.

The **NullText property** - Replaces WaterMark property. Its used to help users know what should be entered, but is not retained as a data value for transactions.

The **Overflow** property specifies which way the remaining items will be displayed. If there are more items than will fit on the device's screen this option can be set to horizontal or vertical which means the user can swipe bottom-to-top or right-to-left to access the remaining data.

The (**PageNo**) property lists the order sequence of a page on a form.

The **Password** property, for the data field portion of the prompt, sets the display of the text equal to asterisks (*) instead of clear text.

The **PenColor** property sets the color of a signature in the Signature control. In Mobile Themes Signature element, the pen color inherits the value from the ForeColor property.

The **PromptNo** property is the Prompt sequence value RFgen assigns to objects placed on a form or a page. If you move the location of a control or page, the PromptNo also updates, or you can edit it directly in the Application Designer Control Panel. If your script uses App.PromptNo VBA extension, it should return the PromptNo value for the specified control.

The **Required** property is used in prompts such as the Memo control where input is required before the user can continue to the next page, prompt or task in the application.

The **RowAltColor** and **RowSelector** (TreeView Control only) properties sets every other row to the color selected and enables users to select the row (True) as opposed to just viewing the content in each row.

The **ScaleToFit** property increases or decreases the icon captions to fit inside buttons or desktop icons when the overall size of the button or desktop tile size is changed. This feature is enabled when its True and is disabled if set to False.

The **Scrollbars** can be enabled for horizontal scrolling, vertical scrolling, both, or none. If set to Automatic, RFgen calculates and displays the scrollbars for you. This property is for select controls only. See [Mobile Themes > Scrollbar](#) for more details.

The **Selection** property group is used to style a selected item from a group of items or list in a parent control such as the ButtonList, ComboBox, DesktopIcons, ImageList, ListBox, or PanelList. The subproperties include BackGround, Border, Text, Transparency and Visible. To compare against the unselected item properties, see Items property topic.

The **ShowBorder** property will hide or show the border of an element. The values are True, False, or (Default). (Default) uses the property value set for the control (element) in Mobile Themes. *True* will display the border; *False* will hide it.

The **ShowInForm** is a child property of the Menu: Heading property. If the Heading property is set to True, then ShowInForm can be set (True/False) to suppress the menu's heading caption. **True** hides (suppresses) the Menu's own heading, and the Form's heading is displayed as the Menu's heading. **False** will enable the menu's heading and remain visible if the Menu control object is not obscuring the Form's header. If the menu object was set to the same dimensions of the form (and overlaid the header), then Form's heading will be used as the Menu's heading even though ShowInForm is set to false.

The **ShowLines** property will hide or show the lines between rows and columns. The options are (Default) which uses the theme properties, None for hiding all the lines, Horizontal for showing only the lines between rows, Vertical for showing only the lines between columns, and Both for showing lines between rows and columns.

The **SelColor** refers to the color of the selection bar shown in controls like the combo box or list box. The highlighted value is what will be chosen when the user presses the enter key.

The **Size** is the background height and width of the control in pixels. If AutoSize is used, it can influence the height and width values. In the ButtonList control, this sizes the buttonlist container-- not the items in the list.

The **Sorted** This was replaced by SortMode in RFgen 5.2.

The **SortMode** property is used in list controls and will sort content in accordance to the value selected. The values are "(Default)", None, Ascending, or Descending.

The **Source** property (the HostScreen control only) selects an executable to be emulated within the Host Screen control.

The **StretchImage** is used to either shape an image to the size of the control or allow the image to be its natural size whether it fits in the control or not.

The **Style** property is used in the ButtonList control and specifies whether the items in the ButtonList are to be presented as a squarish buttons or icons (text and icon but no button as a container).

The **SystemIcons** property group is used to associate RFgen-supplied actions (Call Event) with a customized icon or a RFgen-supplied icon. For example, if you wanted to include a Cancel operation in your application, use the functions/properties in this group to select the Cancel operation and also associate it with an icon. When you are done adding icons, a value in brackets {} shows the total number of icons associated with the control. For specific details on how to link a customized icon to a Action, see [To link SystemIcons with system operations](#).

The **TabNo** property is cursor/prompt sequence number for controls that can accept the focus. This property is not the same as the SeqNo property. For example, a Label control cannot accept focus but have a SeqNo but will not a TabNo. Since a TextBox can have focus and have a SeqNo and TabNo. You can edit a TabNo so to force where the cursor goes after a specific object loses focus.

To view the list of controls/objects' tab numbers, click on the  Sort icon or see [Sort Controls](#).

The **Text** property group is used in the ButtonList Control and specifies the color of the font, size of the font, style of the font, and position via the margin values. Whether the values here are overridden by other property settings in the control will depend on how deep this property is nested.

The **TextID** property links a text (word or statement) resource to the object. It can have dual uses, depending on the value used. If the identifier is the TextId from the table in the Solution Explorer > Text Resources folder, its used to translate the text string. If the value is preceeded with a "%", it links user inputs from the source TextBox to a destination Label at runtime.

The **TextOptions** property sets whether the text will be on a Single Line or Multiple Lines.

The **Theme** property changes the border of the title bar area to one of several hardcoded styles.

The [Title] property in Themes > Dialog is used stylize in the title of a Dialog box (pop up messages dialog box).

The **Transparency** property sets the level of transparency of a child control and bases its coloring on the parent (i.e.Form, Page1, Panel). "(Default)" uses the level of transparency set in the corresponding Mobile

Theme element. "None" will use the item's BackGround color. A percentage value (5 %, 10% etc), sets the transparency level.

The **UseMenuTheme** property will override the local properties and apply the default theme properties for the menu control.

The **ValidationTable** property presents a list of downloaded tables that can be used to verify that the data entered already exists in this table and the Validation Field. The two properties must be used together.

The **ValidationField** property presents a list of table fields specified by the Validation Table property. This is the reference field to determine if the data entered in the prompt already exists. If it does not, the Error Message property will be used to warn the user.

The **Visible** property, set to True, makes a prompt or element visible, while setting it to False makes it invisible. Even though the prompt may be invisible, the GotFocus, OnEnter and Lost Focus events will still be executed for this prompt if the focus automatically shifts from a prompt before this prompt to one after this prompt.

The **WaterMark** property group is available for the ComboBox, TextBox, Memo and SpinEdit controls and is used to stylize the appearance of text that appears in the textbox as an example entry for users. There are two ways you can add text (a caption) in the WaterMark property: a) Enable Language Translations (which enables Text Resources) and select a text string from the WaterMark property; or b) Disable Language Translation, and type in your text (caption) into the WaterMark property. If you do not want any text, you can also just leave it blank. At runtime, the user will be able to manually edit the WaterMark text or scan a barcode which will overwrite the WaterMark text.

The **ZOrder** was obsoleted in 5.1.

The **Design Mode** (which used to be under Display tab) has to **Configuration > Desktop Preferences** in version 5.1.

Appendix B: Screen Mapping

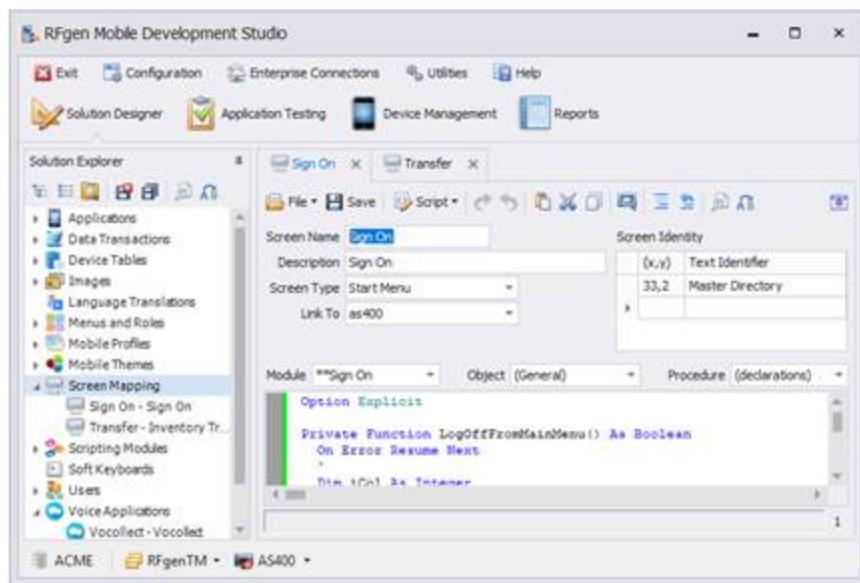
Theory of Operation

Screen Mapping works by identifying a Main Menu in the host system that is used as a base reference point for navigation to and from transaction screens. This is the starting point for transaction processes. The navigation process (i.e., series of keystrokes) required to proceed from a login state to the Main Menu may be automatically recorded by the system. A small visually unique portion of the main menu screen is marked for identification purposes. If required, multiple areas of the main menu may be marked. This allows the server to know that it has arrived at the requested destination.

The next step is the navigation from the main menu to the transaction screen. Transaction screens are the displays that users use to input data into the host system. Again, the navigation process (i.e., required keystrokes) to reach a screen and then return to the main menu may be automatically recorded.

When a transaction screen has been identified, the next step in the process is to identify the fields on the screen where the data will be entered. Screen fields are marked and each is given a field name. These fields may optionally be used in much the same way as ODBC fieldnames are used by the system; i.e., they may be used to create transactions by dragging the host screen's marked fields on to an application screen. Doing this, means that the user can scan and enter all the data on the mobile device and the server already knows how to log on to the host system, how to navigate to the proper screen and where to place the collected data on the host screen, all without having to program the scripts yourself.

Screen Mapping



The Screen Mapping module enables mobile applications to be connected to multiple host systems like the AS/400, IBM Mainframe, UNIX systems and other character-based 'legacy' applications. This kind of screen does not use application Forms, Pages, or graphical prompts/controls.

In practice, screen mapping applications use keystrokes recorded for a host screen's navigation and data entry, along with the collected data and play back of the keystrokes, while replacing the recorded data with

the newly collected data. Accuracy in staging and applying keystrokes is of the utmost importance. The recording capabilities provide this needed level of accuracy. The solution provides three host protocols: TN5250, TN3270, and VT220 in order to interact with legacy hosts.

The screen mapping applications may be created by means of an automatic recording processes, and point and click, drag and drop development methods. The automatic recording processes create Visual Basic for Applications (VBA) macros (i.e., scripts) that utilize pre-built screen mapping extensions for system navigation and data handling. An intuitive set of VBA extensions have been designed to interact with any character-based legacy application. Users may, of course, modify scripts as desired or create new scripts. **Screen mapping supports transaction queuing so that when a host is offline, data collection may continue uninterrupted.** The system thus allows true 24/7 support for critical data collection operations.

For more details on how to record macros, see [How to make Screen Mapping Work](#).

Note: Support of Telnet displays, (character-based applications) was removed from RFgen in 5.1 but has been retained in 5.0.

How to Make Screen Mapping Work

The Screen Mapping module is included in the Mobile Development Studio/RFgen. When loaded and authorized, the system functions (simultaneously) as both an ODBC database server and a legacy host terminal server.

To properly function with a host AS/400, IBM mainframe, UNIX, or other legacy-based system, the server must be part of a communications network, capable of interacting with a host via TCP/IP networking protocols.

Programming Philosophy

Programming Screen Mapping applications differ from typical data collection applications in that problems, if and when encountered, need to be handled automatically by the application program without the involvement of the remote data collection users. For example, the RFgenSM module contains built-in commands, such as 'SM.GetText', that can be used to search for specified text in a host screen (at a specific 'Col, Row' location, or anywhere on the screen). This, plus other diagnostics, allow programmers to positively identify the correctness of 'happenings' within a host application.

Design Considerations for Screen Mapping

Before starting a screen mapping project, users should consider certain project design issues related to the following topics: Screen Mapping Level, Logon Security, Data Integrity, Keyboard/Special Key Configuration, and Runtime Environment/Variables.

Screen Mapping Levels

The Screen Mapping interface is divided into 3 levels of usage: **Low, Medium and High** levels.

Low level use is represented by scripting in the VBA environment all aspects of interaction between the server and the host system using the SM object's methods and procedures. These commands allow the developer complete control over the host session. Examples include sending/receiving text, control keys, cursor positioning, "WaitFor" statements, "Find" statements, etc. It is entirely possible for the user to

write/program complete solutions using only these low-level commands. These commands are documented in the Screen Mapping Extensions section.

Medium level use is typified by the creation Host Screen macros and / or Data Entry macros using the recording capabilities. At any time in the VBA script, one of these macros can be called and the host screen can be made to navigate or transact instantly. This capability simplifies the programming of the navigation requirements within a host system. Calling a transaction macro will place all collected data into the host screen's fields and submit the screen to the host for processing. Transaction macros can have input / output parameters. These parameters are used to send and receive data from the host screen. Because of the solution's unique design, transaction data can be stored while the host is offline, and send to it for processing later when the connection is re-established.

Medium level usage entails the development of a VBA script to call the pre-recorded macros. One Screen Mapping command '*SM.CallMacro*' is oftentimes sufficient to update the host.

High level use of the Screen Mapping capabilities is represented by the automatic recording of the 'Host Screen' and 'Transaction Macros' discussed above. Host transaction fields are then embedded in applications in much the same manner (e.g., drag and drop) as table fields from ODBC databases. Using embedded methods, data automatically posts to the host once all input fields have been entered (note: posting was accomplished manually as the last step in Medium level usage, not automatically as with embedded fields).

A final note: All automatically recorded macros are created using base low-level commands. Thus, users have complete access to all VBA scripts, including modifying them as desired. An example of user modifications might include checking for error conditions (such as bad data) and/or warning, error, or informational messages. Copying and pasting the recorded macro script and placing it in the application directly is a quick way to build a low level solution.

Logon Security Considerations - Screen Mapping

There are many ways a programmer can implement security. One important thing to remember with screen mapping is **that the end-user is never on-line with the host system**. The end-user has no way of interacting with the host system that you haven't provided for. With this in mind, the following are a few examples of login security methods:

The developer can create a "Login" Transaction Macro that is linked to the Login Host Screen. If a new user needs to sign on, this can be accomplished through a simple call to the "Login" macro.

The user ID and password specified when the user logged in could be provided to the script. The login would occur when the user called the first macro.

A generic login could be specified, and the user changed dynamically using a system function such as "sign-on" or "change-to". This command could be executed as part of a single Transaction Macro, or as a separate one called only when the user changes.

System Integrity Considerations

Screen mapping actions that cause a host system to be updated should be acknowledged by the host before another command is sent. The host interface is designed to automatically accommodate for this as much as possible. The script or macro waits for the host to not be busy before reading from or writing to the host screen. In a host-busy condition (input inhibited), The server will wait for the timeout period specified in the

settings for the condition to clear. However, screen mapping VBA extensions should be incorporated to provide ways of acknowledging successful actions. For instance, the following commands may be used to provide programmer control over host sessions:

SM.WaitForText – This function looks for a unique text string on the screen for a specified amount of time. If the text is found, it returns True, otherwise, it times out with a value of False.

SM.WaitForCursor – This function waits until the screen input cursor stops at the desired location for a specified amount of time. If the cursor stops at the desired location, it returns True, otherwise, it times out with a value of False.

SM.WaitForScreen – This function looks for the desired screen identifier for a specified amount of time. If the application is in the correct screen, it returns True, otherwise, it times out with a value of False.

SM.WaitForHost – This function is designed for vt220 connections. As the vt220 protocol does not typically include input inhibit conditions, this function will return True once the host responds to the previous action command, otherwise, it will time out with a value of False.

SM.CallMacro – This function has a “Queue-Offline” argument, which, if enabled, will store the transaction if the host is offline. These “store-and-forward” transactions will be processed automatically once the host connection is re-established.

Note: the ‘SM.WaitFor...’ commands are primarily useful (and perhaps required) for VT hosts.

See Screen Mapping Extensions section for more information.

Keyboard/Special Key Configurations

We understand that not all terminal emulations use the same keyboard layout. Accordingly, a developer is provided with 2 options to send special keys to the host:

The first is a pop-up window that is accessed by clicking on the ‘Hot-Key’ icon at the top of the Host Session window. You can then select the desired special key from the list in the window and transmit it to the host.

A second option is to re-map selected keys on your keyboard to transmit the special keys instead. These remappings are defined by clicking on the Session menu item at the top of the Host Session window. To re-map a key simply press the key(s) you want to re-map and then select the desired special key to send from the drop-down list.

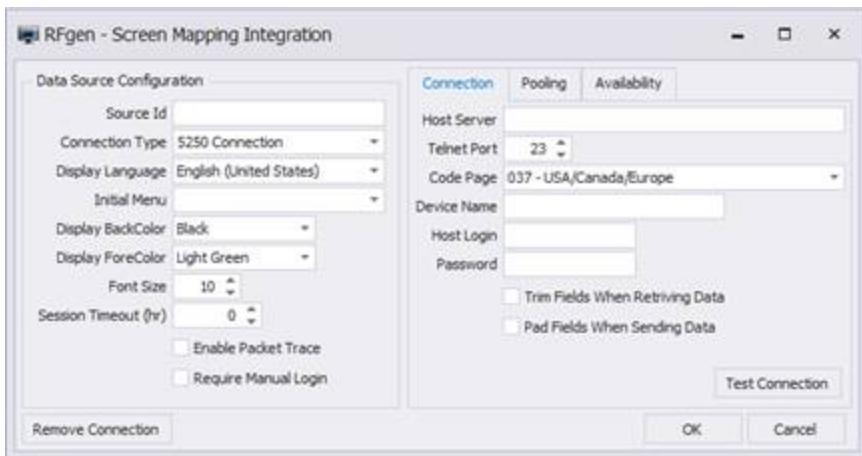
See the VT220 Key Mapping section for more information on Hot-Keys.

Runtime Environment Variables

A programmer can make use of any of the standard language extensions while creating or modifying macros. In addition, any global variables specified in the “Win32.bas” or “RFgen.bas” modules are available for use in any macro. However, the user cannot call another macro from within a macro.

Configuring the Host Connection

In the Mobile Development Studio, click on **Enterprise Connections > Add New Enterprise Connection > Add New Screen Mapping Connection**. The following window will appear.



The first entry is the **Source Id** used to reference the data connection only. This can have any value but spaces and extended characters are not recommended.

Choose the **Connection Type** (VT220, TN5250 or TN3270); i.e., the protocol used to communicate with your host system. Notice that there is an additional option called Console Application. This type is designed to launch a console application rather than use a telnet server and then pass that display through the server to the device using the HostScreen prompt control. One example would be the SAP console application (SAPCNSL.EXE) running on the server and being displayed and allowing interaction with the user on a mobile device. Simply specify a process or executable name to run and any passing parameters necessary.

The preferred option is UTF-8 but if a legacy system's output is language specific then the **Display Language** field should be changed to make the screen render correctly. The Language field can be left as (Default) if a code page is specified or if UTF-8 is used.

Preferences for the emulation screen include the **Back Color**, **Fore Color** (the color of the font) and **Font Size**. These are only for development since the screens are hidden during production.

The **Session Timeout** value (in hours) will disconnect and reconnect to the legacy server at the specified interval. This may be required if the legacy server is configured to not allow a connection that never times out.

In the case of communication errors the **Enable Packet Trace** option can be set and a trace log of the communication will be captured. This is used by support staff to diagnose issues on behalf of the customer. Please contact support if this switch is necessary.

If the **Require Manual Login** is checked, a connection request is created between the user and the ERP system. If this box is unchecked, the user login uses the ERP connection between RFgen and the ERP system.

Connection Tab

Next, type in the **Host Server** name or IP address. The **Telnet Port** is the port that the server uses to communicate with your host. The default for a telnet server is port 23.

If TN5250 or TN3270 are selected, you may enter a **Code Page** for specifying the language being used in the protocol and an IBM **Device Name** for the host system. Code pages were selected for loading when you loaded the screen mapping software. These fields are hidden in the VT setup.

For VT220 the **Data Stream** field can be set to either Standard or UTF-8 to accommodate the type of packet

data coming from the host system.

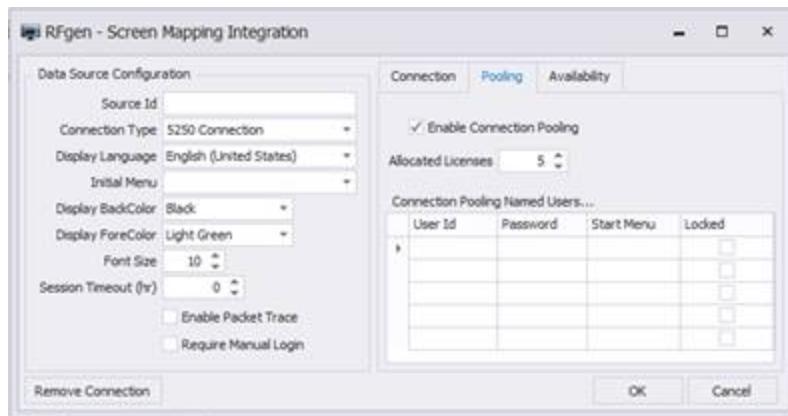
When using the connection type 3270 or 5250, the **Device Name** field is designed to make each connected device appear unique to the host system. Leaving it blank, the host system will not distinguish between the connecting clients. Fill this field in with a name and the server will automatically add a three digit, zero padded number to each client so the host system will see each connecting session as a unique device.

The **Host Login** and **Password** fields are used only if SSH is used when connecting to the host system. Under the VT220 options, if Connect via SSH is checked then the Host Login and Password are required.

Trim Fields When Retrieving Data set to enabled will auto trim spaces from the host output fields. If a variable is defined for a section of the host screen (like where error messages are displayed), this feature will trim the text for easier use in message boxes, for example.

The **Pad Fields When Sending Data** option when enabled will use spaces to pad any input. A variable defined for a region of the host screen where input will take place also has a length property assigned at the time the field was defined. If the data is 3 characters, but is placed in a host screen field designed for 10 maximum characters, the server can pad the input data to fill up the host screen input field.

There are some additional properties for the VT220 mode only. **Echo Characters Locally** means that the server will print the typed characters on the telnet screen because the host is accepting the keystrokes but not showing them to the user. **Wrap Text at End of Line** will force the server to place the additional text on the next available line if it doesn't fit in the current field. Most host system will do this automatically. **Destructive Backspace** means that the server will receive a backspace command and apply it to the screen as a command that removes the last character. Some systems would move the cursor but not remove the character. **Send Whole Key Packets** forces the server to submit keystrokes in one packet instead of two in some cases. Most host systems already support keystrokes coming in as one or more packets. **Send Return + Line Feed** will add a carriage return plus a line feed to the Enter keystroke when communicating with the host. **Connect via SSH** will establish an SSH (secure) connection to the host from the server. If this option is turned on then the SSH **User Name** and **Password** fields will be required.



Pooling Tab

Connection Pooling can be enabled and the maximum connections allowed in the pool can be selected. This selection will determine how the server and its clients will interact with your host system. The options for the **Pooling Status** are:

Disabled – Setting connection pooling to disabled will cause the server to spawn a connection to the host system for each active mobile device. Each connection will be linked to a particular device on a one-to-one basis, and will be shut down when that device disconnects. Note: there is no limitation on the number of connections allowed.

Enabled – Setting connection pooling to Enabled will cause the server to spawn a single connection to your host system. As each device requires access to the host system, they will go to the pool and retrieve one of the available connections. When they are finished, the device will release the connection back to the pool. If no connections are available, the server will start a new connection (up to the specified maximum) and add it to the pool. After 10 minutes of non-use, an opened pooled connection will be terminated releasing resources on the server and potentially licenses on the host system. Keep in mind that unless the SM.BeginTrans and SM.CommitTrans commands are used, it would be possible for one user to position the screen in one place while another user also uses that pooled connection to perform their tasks causing both users to get failures.

The **Connection Pooling Named Users** grid dictates how each host session is started. You may also override the default settings by configuring a specific pooled session separately.

Session - Each of the individual pooled connections are listed separately. This provides for specific settings for each connection.

User Id- If the host system requires that unique names be used or creating multiple logins with the same user is prevented, each pooled connection can have its own user ID. Session, user, and password information can be obtained at runtime with the commands SM.SessionUser, SM.SessionPwd, and SM.SessionID.

Password- This is the corresponding password used for each unique user ID.

Start Menu- Each session can have its own main menu. When a session is requested and no main menu is specifically assigned or the "(Default)" value is used, the next available session will execute the requested main menu based on the scripts and chosen transaction. If a session is requested and the next available session does have a main menu assigned, and it is not the required one, other sessions will be evaluated for a matching main menu. If one is found and available, it will be used.

Locked- The ability to lock a session means that the session can ONLY be used with the specified main menu and will not allow other main menus, even if all other available sessions are in use. For example, there are 10 pooled sessions, five locked on main menu A and five locked on main menu B. If a session with main menu A is requested and all five sessions for main menu A are currently used, the server will look to the sessions assigned to main menu B. If they are not locked, the server will take one of them. Since they are locked into main menu B, in use or otherwise, the server will wait for one of the first five to be released.

The purpose of locking a set number of sessions to a specific main menu is to ensure that there is always some bandwidth available for certain transactions. Not locking them means that they will be marked with a preference for a type of transaction (the use of a specific main menu), but will switch to another main menu when necessary.

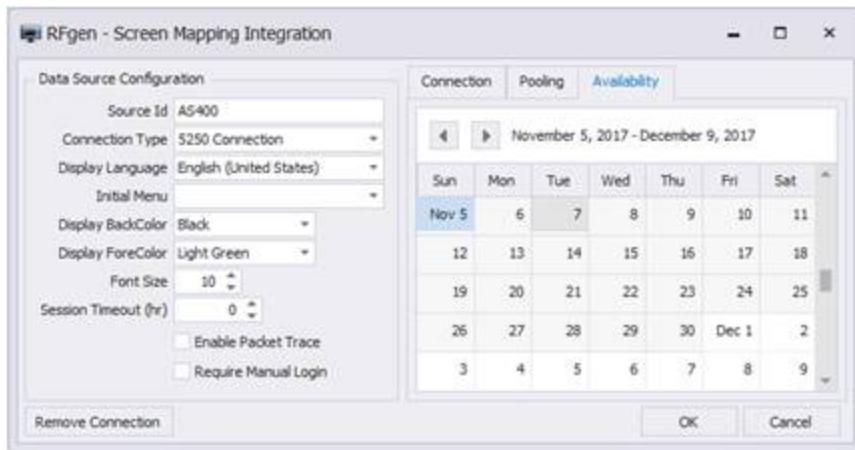
For example, there are 10 pooled sessions available and the first five have one main menu assigned and the last five have a second main menu assigned. When a session with the second main menu is requested, the 6th session handle will be used. This is only significant because of the Locked property.

The **Test Connection** button will verify all settings before saving the connection. This is not required.

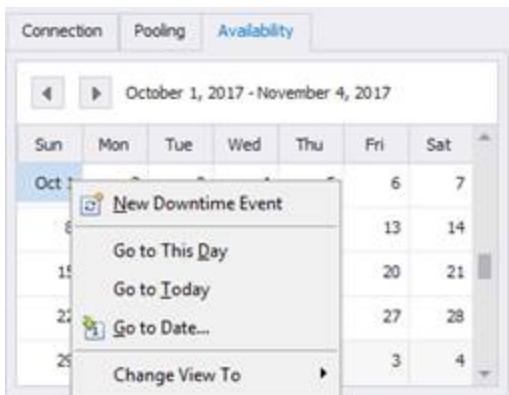
The **Save** button will save changes but will not test and verify settings.

Availability

The Availability tab in the 'Screen Mapping Integration' Window allows users to schedule 'down time' for a host connection; i.e., **the server will disable the connection** during the time that a host is offline. The following panel will appear.

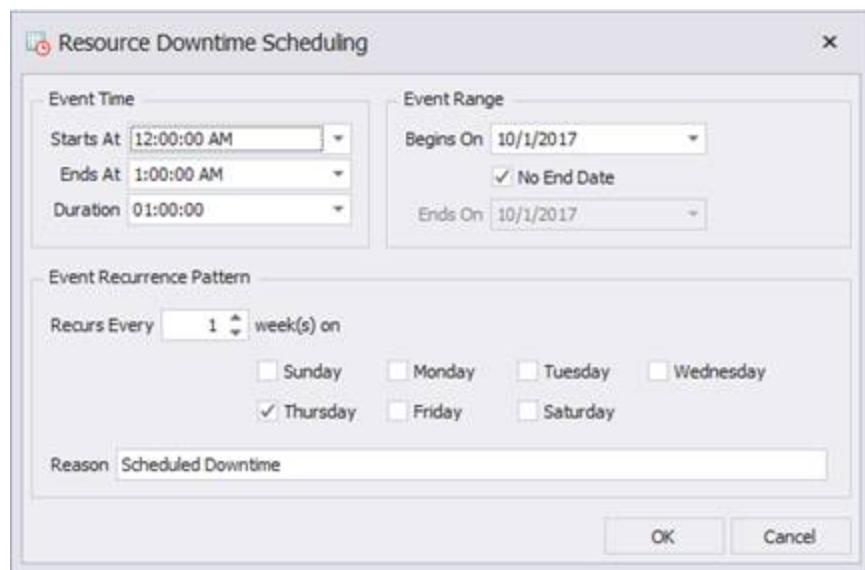


To schedule downtime, right-click on the date or days in the calendar and select the appropriate item from the menu.



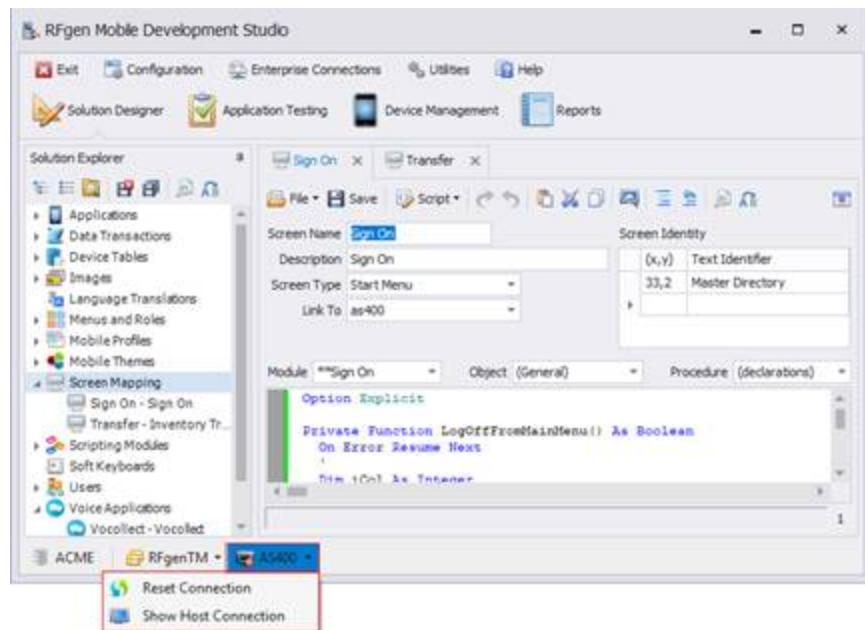
In this example, the New Downtime Event was selected.

In the Event Time box, the connection will be unavailable every Thursday, for 30 minutes between 12 AM and 1 AM beginning Oct 1, 2017 and reoccur until an End Date is supplied.

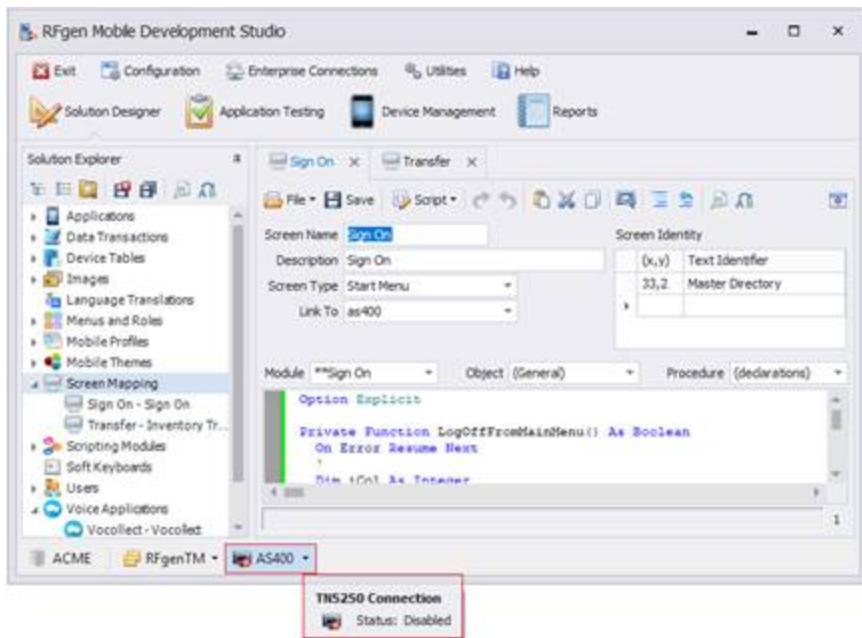


During the down time, transactions can be 'queued' for automatic posting to your host when the connection becomes available.

When you 'Save' the screen mapping configuration entries, a session with your host should be available by right-clicking on the screen mapping connector and selecting Show Host Connection and the host name will appear as a 'connection indicator' at the bottom of the Mobile Development Studio window. A red circle in the connection indicator...



indicates that a connection has not been established with your identified host.



With a host connection established, your screen mapping development project is ready to be started.

VT220 Key Mapping

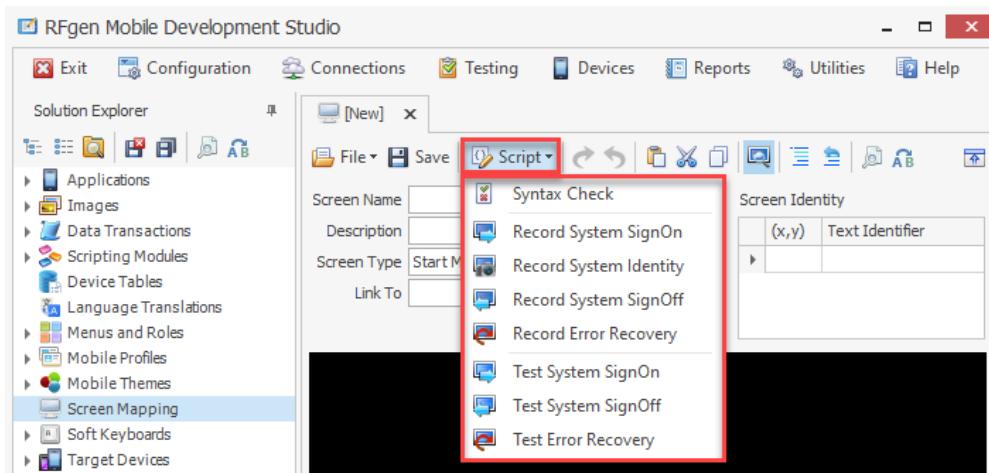
The default key mapping for VT sessions can be edited if certain keys are not working correctly with the host. Use the Import utility from the menu, choose Cached System Files, and import the VT220 Key Map Template. A file called VT220.VKM will be placed in the install directory and can be edited with a text editor.

```

File Edit Format View Help
TERMINAL_ID = 'VT220'
BACKSPACE = 8
BREAK = 3
CURSOR_UP = 27 + '[A'
CURSOR_DOWN = 27 + '[B'
CURSOR_RIGHT = 27 + '[C'
CURSOR_LEFT = 27 + '[D'
DELETE = 127
DO = 27 + '[29~'
ENTER = 8
ESCAPE = 27
F1 = 27 + 'OP'
F2 = 27 + 'OQ'
F3 = 27 + 'OR'
F4 = 27 + 'OS'
F5 = 27 + '[15~'
F6 = 27 + '[17~'
F7 = 27 + '[18~'
F8 = 27 + '[19~'
F9 = 27 + '[20~'
F10 = 27 + '[21~'
F11 = 27 + '[23~'
F12 = 27 + '[24~'
F13 = 27 + '[25~'
F14 = 27 + '[26~'
F15 = 27 + '[28~'
F16 = 27 + '[29~'

```

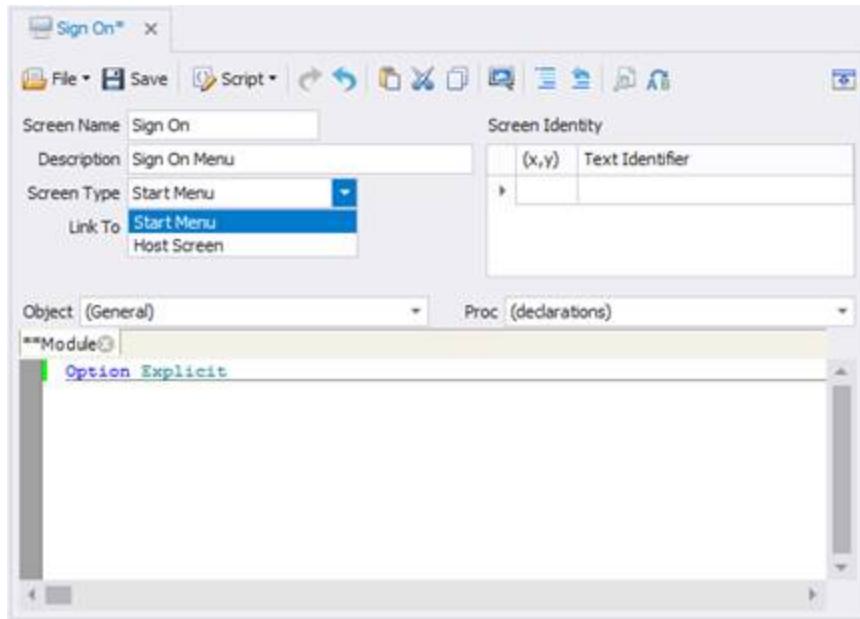
How to record a macro for a screen map



1. Right-click on **Screen Mapping** to add a new host, or create a new screen mapping macro or right-click on the title of an existing macro to make additional changes.
2. Double-clicking on an existing macro opens the Host Screen Editing window.
3. Click on the Script icon.
4. In the drop-down list, select the menu option to record or test the desired item.

There are two types of macros that can be created here. **Start Menu** macros take the data connection as it is when first connected and logs in and navigates to a main menu used as a generic starting point for all screen mapping transactions. **Host Screen** macros are used when a specific transaction is chosen by the mobile user to navigate the host system to the proper screen meant to accept specific data (ex.: Cycle Count screen). Additionally, this macro can be used to play back the keystrokes of a user entering the collected data into the screen itself. This macro stores the x,y coordinates of the fields on this host screen and places the collected data in the proper places before sending additional keystrokes to submit the data (ex.: F8). At that time the host screen processes the data just as if the user entered the data directly to the host screen.

Building a Start Menu Macro



Begin by entering the **Item Name** of the Start Menu macro. This will be the name given to the primary main menu used to log the host system in. Fill in the **Description** field, select Start Menu for the **Screen Type** and **Link To** the name of the screen mapping data connection.

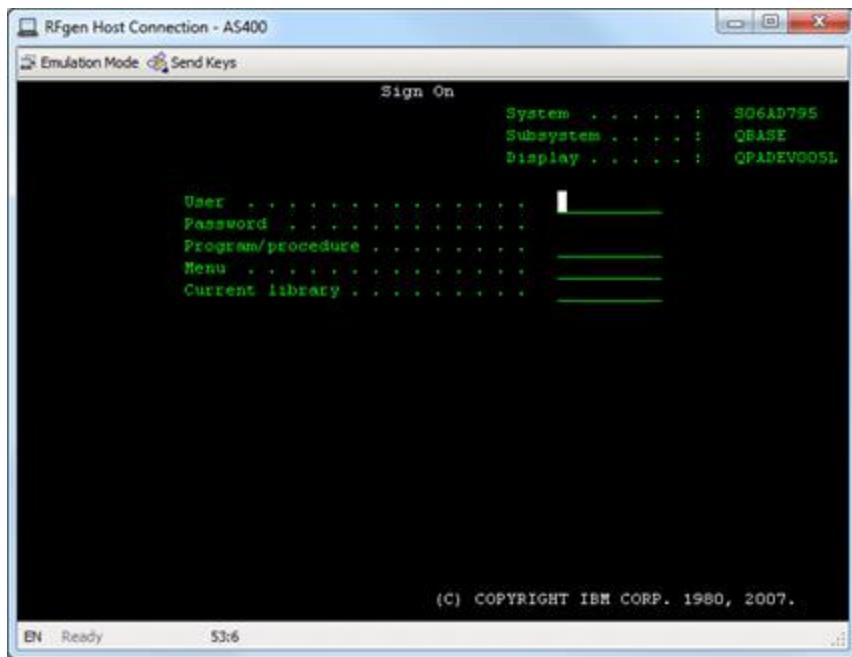
Next select the Recording menu option.



There are four different scripts that this macro can contain. **Record system sign-on** records the keystrokes to successfully log in to the host system and navigate to the main menu. **Identify system menu** records the x,y coordinates of some text on the host screen, so when the system attempts to reach this page it will compare the host screen to what is known to be the proper screen. **Record system sign-off** contains the keystrokes recorded for exiting the main menu and going back to the login screen. **Record error recovery** records the keystrokes to do whatever the user must to get the host back to the main menu. Usually the safest solution is to back out as far as will ever be needed and then log in again.

When recording these macros the Host Session window will appear. In the case of this host system, it is an

IBM AS/400 connected to RFgen using a TN5250 telnet protocol.



The first toolbar icon represents the mode the window is in. Examples are Emulation Mode, Recording Mode and Identifying Mode.

The Send Keys Selection

Selecting a 'Hot Key' from the available list is an alternative to re-mapping your keyboard. The following list of keys will appear.

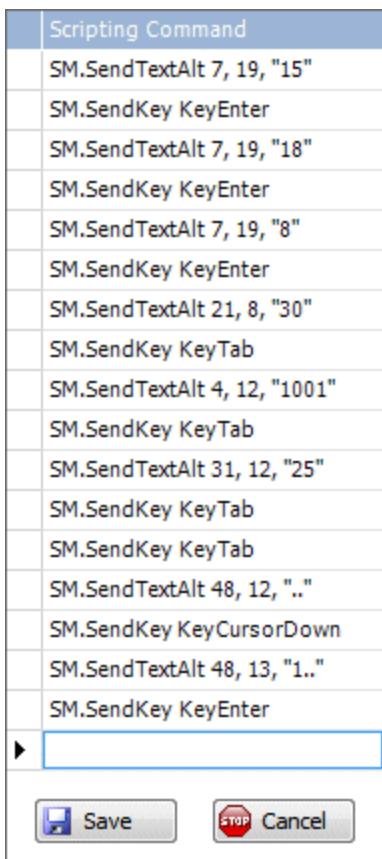
Clicking on a menu item will send the selected key to your host. If you are currently recording a macro, the selected key will become part of that macro.

Start Menu Macro – Record System Sign on

Choose the Recording menu option and select Logon to the Main Menu. The Host Session window will display with a column on the right for recording and editing all keystrokes.

Enter the keystrokes necessary to display the main menu. During the recording phase the user may right-click on the host system to bring up a menu of additional commands that may be inserted into the script. See the Recording Options section below for a complete description of these commands.

The Scripting Commands column on the right can also be edited in case a mistake is made, variables need changing or timeout values need to be adjusted. If a step is forgotten, such as waiting for text to appear on the screen before performing the next keystroke, simply position the insert arrow on the row to be preceded by the missing command and perform that command. For example, highlight a section of text, right-click and select the Add WaitForText Statement option. If a keystroke was pressed accidentally, simply delete it from the list. Be sure to put the insert arrow back at the bottom before continuing.



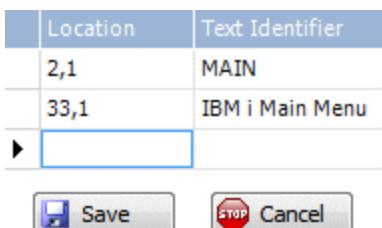
When completed, click Save. Cancel may be clicked at any time to cancel the recording session. An internal macro called **LogOnToMainMenu** is recorded by this step.

Start Menu Macro – Identify System Menu

In order for the server to positively identify the main menu, a portion of the screen needs to be 'marked'. From the recording toolbar button options choose Identify system menu.

Next, select unique text for this screen on the host system by left-clicking and dragging across the text and then select the Mark Field menu option. If necessary, multiple selections can be made.

The marked region and its coordinates are placed in a grid on the right side of the host screen window where the whole list is captured and can be edited.



When complete, click 'Save' to save all marked areas. These identifiers will appear in the Screen Properties

window.

Start Menu Macro – Record System Signoff

From the recording toolbar button options choose record system signoff. The Host Session window will display with a column on the right for recording and editing all keystrokes.

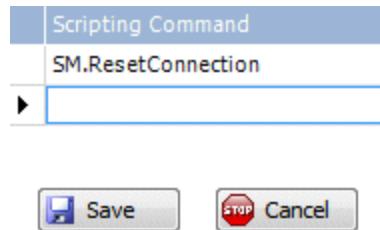
Enter the keystrokes necessary to logoff from the main menu. During the recording phase, the user may right-click on the host system to bring up a menu of additional commands that may be inserted into the script. This was described in the first step.

When completed, click Save. Cancel may be clicked at any time to cancel the recording session. An internal macro called **LogOffFromMainMenu** is recorded by this step.

Start Menu Macro – Record Error Recovery

This step records the keystrokes that will navigate the host system out of any possible screen or menu back to the known main menu. If the host system pops up additional screens like system messages that the scripts do not take into account, then the server would notice that none of the recorded identifiers match were the host is and run this script.

From the recording toolbar button options choose record error recovery. Enter the keystrokes necessary to get back to the main menu. Possibly, an easier solution would be to type in the SM.ResetConnection command as shown.



When the system resets, it automatically re-runs the LogOnToMainMenu macro taking the host to the main menu. In this case, the complete **AbortNavigation** macro would look like this:

```
Private Function AbortNavigation() As Boolean  
On Error Resume Next  
'  
SM.ResetConnection  
'  
AbortNavigation = SM.WaitForScreen("Base", 10)  
End Function
```

Be sure the host system is not adversely impacted by using the SM.ResetConnection command.

Start Menu Macro – Test Scripts

After the first four steps have been completed, the recorded macros should be tested.



The drop-down menu from the 'Test' button allows you to select the macro to be tested. A message box will appear showing the success or failure of the macro.

Important:

To test the **LogOnToMainMenu** macro, your host screen should first be positioned at your login screen.

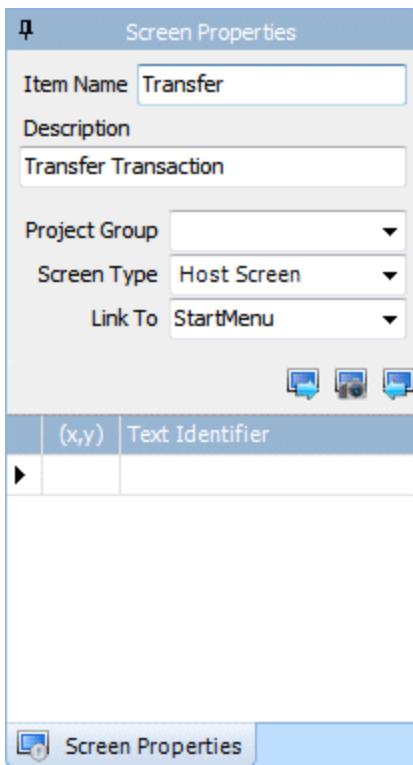
To test the **LogOffFromMainMenu** macro, your host screen should be positioned at your Main menu.

To test the **AbortNavigation** macro, your host screen should be positioned anywhere in the menu structure but not on an application screen.

Be sure to save all the work that has been done before moving on to the Host Screen macro recording steps.

Building a Host Screen Macro

The Host Screen macro contains just the navigation to and from a transaction screen. A Transaction macro is linked to this macro so when a form tries to update the host screen the Transaction macro will use this Host Screen macro to perform the required navigation.



Begin by entering the **Item Name** of the Host Screen macro. Fill in the **Description** field, select a **Project**

Group if desired, select Host Screen for the **Screen Type** and **Link To** the name of the start menu just defined.

Next select the Recording menu option from the toolbar buttons.

There are three different scripts that this macro can contain. **Go to the Application Screen** records the keystrokes to successfully navigate the host system to the particular transaction screen from the already defined main menu. **Identify the Application Screen** records the x,y coordinates of some text on the host screen so when RFgen attempts to reach this page, it will compare the host screen to what is known to be the proper screen. **Return to the Main Menu** is the keystrokes recorded for exiting the transaction screen and going back to the main menu.

Host Screen Macro – Go to the Application Screen

Choose the Recording menu option and select Go to the Application Screen. The Host Session window will display with a column on the right for recording and editing all keystrokes.

Before proceeding, please ensure that you are on the host's Main menu (as previously identified).

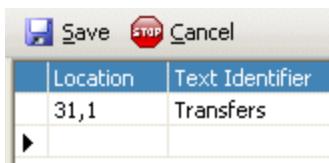
Enter the keystrokes necessary to navigate to the transaction screen. During the recording phase, the user may right-click on the host system to bring up a menu of additional commands that may be inserted into the script. When completed, click 'Save'. 'Cancel' may be clicked at any time to cancel the recording session. An internal macro called **GoToScreen** is recorded for the screen by this step.

Host Screen Macro – Identify the Application Screen

In order for the server to positively identify the application screen a portion of the screen needs to be 'marked'. From the Recording menu option, choose Identify the Application Screen.

Next, select unique text for this screen on the host system by left-clicking and dragging across the text and then select the Mark Field menu option. If necessary, multiple selections can be made.

The marked region and its coordinates are placed in the grid on the right side of the host screen window where the whole list is captured and can be edited.



When complete, click 'Save' to save all marked areas. These identifiers will appear in the Host Screen Editing window.

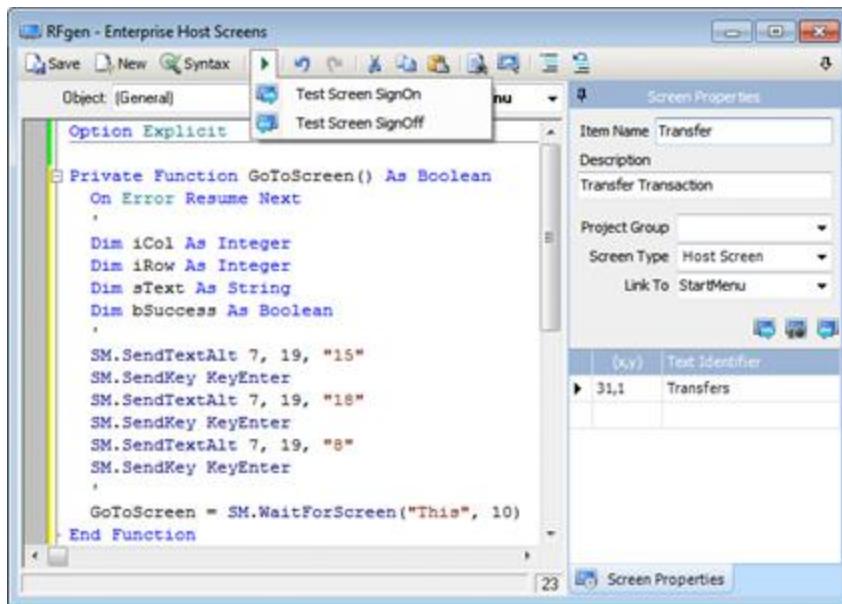
Host Screen Macro – Return to the Main Menu

From the Recording menu option choose Return to the Main Menu. The Host Session window will display with a column on the right for recording and editing all keystrokes.

Enter the keystrokes necessary to return to the Main menu. When completed, click 'Save'. 'Cancel' may be clicked at any time to cancel the recording session. An internal macro called **ReturnToMainMenu** is recorded for the application screen by this step.

Host Screen Macro – Test Scripts

After each step is completed, or at the end of all steps, the recorded macros should be tested. Click on the Scripts menu option and the script window will appear.



The drop-down menu from the 'Run' button allows you to select the macro to be tested. A message box will appear showing the success or failure of the macro.

Important:

To test the **GoToScreen** macro, your host screen should first be positioned at your main menu.

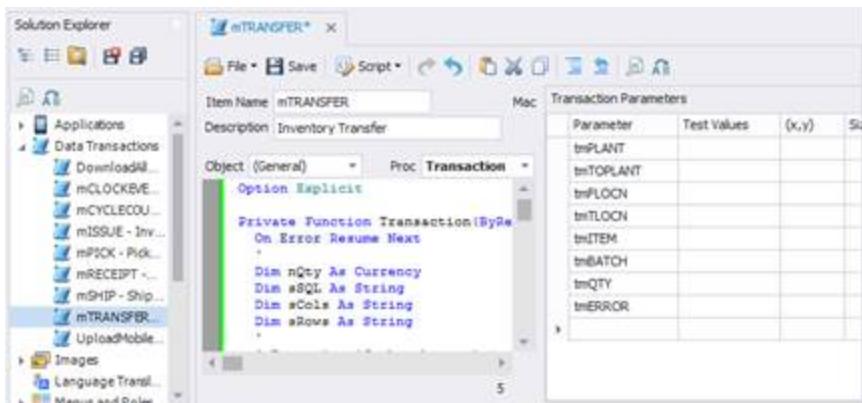
To test the **Transaction Script** macro, your host screen should be positioned at your transaction screen. This macro was created when recording the Enter a Sample Transaction option.

To test the **ReturnToMainMenu** macro, your host screen should be positioned at your transaction screen.

Many keystrokes may have been used to navigate between screens or around the transaction screen itself that may not be necessary. By default, the text written to the screen uses the coordinates to locate the proper input field so any additional tabs, for example, to move between fields are not necessary and can be deleted. The reserved word "This" and "Base" are internally substituted at runtime depending on what names were given to the transaction macro and the main menu macro. Be sure to save all the work before exiting.

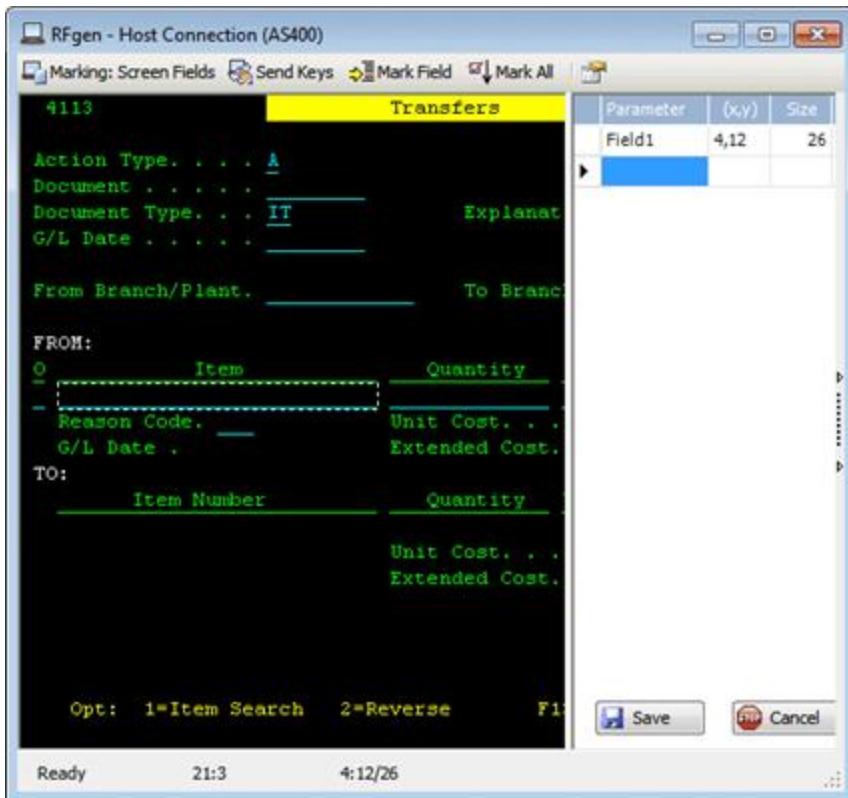
Building a Host Transaction Macro

The Host Transaction macro contains just the recording of the sample transaction. This macro is linked to the Host Screen macro so when called it can determine the navigation steps. Under the Transactions section create a new macro, give it a name and description, select Host Transaction macro type and link it to the Host Screen macro previously recorded.

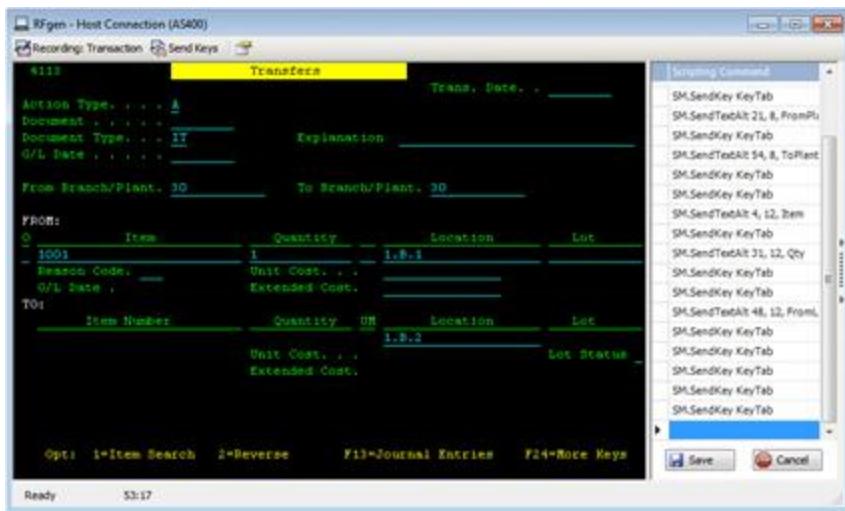


Click the first toolbar button above the parameter grid to navigate the host screen to the correct location.

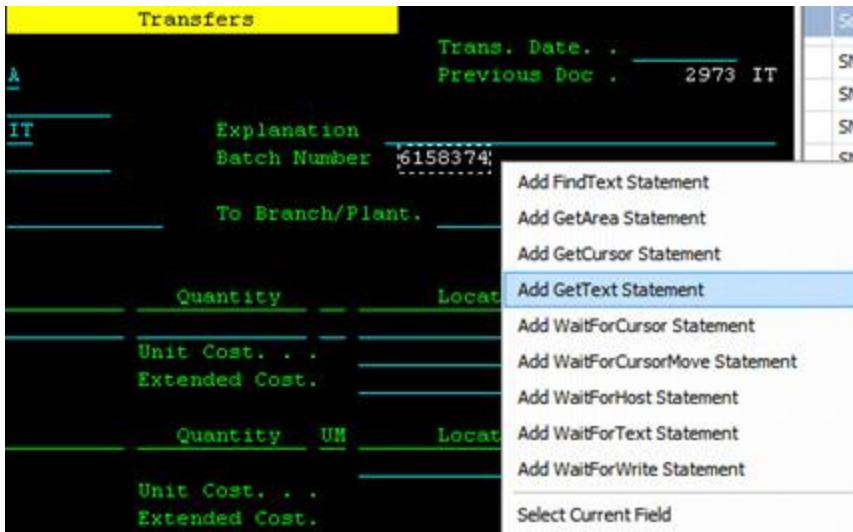
Next click the second toolbar button to mark all the fields on the screen that will be required for data entry.



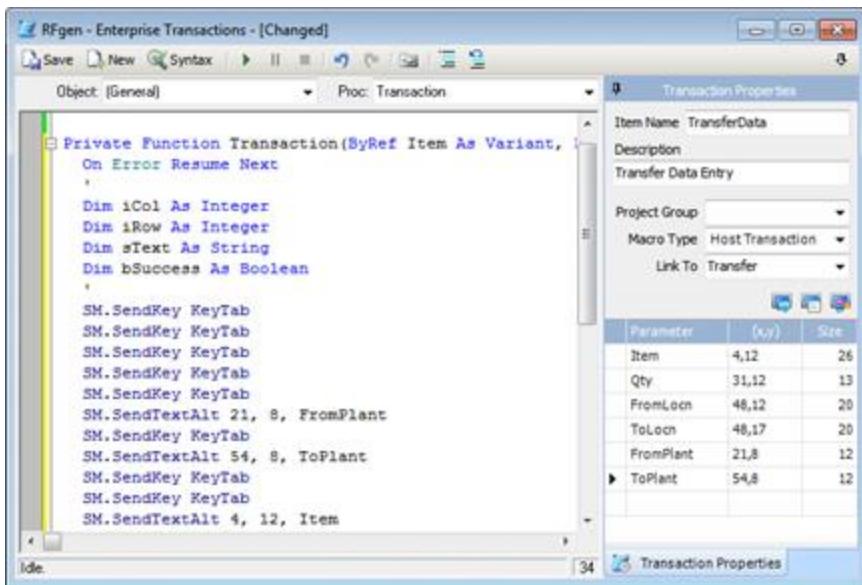
Finally click on the third toolbar button to record a sample transaction.



After recording and submitting the data there may be indicators that it was successful or possibly failed. In this case there is a Batch number generated if the submission was successful so I select the batch number area and right-click to add the GetText command. In the code I will make sure something is retrieved. If nothing is returned then the bottom of the screen will contain an error message and the same GetText command can be used at those coordinates to get the text and display it to the user in a message box.



After the recording there may be a need to clean up the code if extra keystrokes were used and not necessary.



Notice that there are several tab keystrokes but the SendTextAlt command references X,Y coordinates. This makes all the tab characters unnecessary and worth deleting.

Now click on the Play toolbar icon to test the complete data entry portion of the macro assuming you can continue to reuse the same sample data.

Save and exit the Transaction Macro design screen. The last step is to create an application that links to this macro.

Creating a new application and linking it to the data entry macro provides all the marked fields in the toolbox window. Drag them on to the screen, save the application, place the new application on a menu and test the result. Set the host screen itself either to the main menu or the login screen to test all macros working together.

This application does not require any VBA scripting because the macros recorded all the steps. There is the option of taking the code out of the macros and placing them directly in the application itself if more direct control is required. An example might be the login screen on the host requires named user instead of a generic user. In this case the recording can be taken and placed in the RFLogin form. However in doing so the Host Screen macro does not have a Start Menu macro to rely on so the Host Screen and Host Transaction macro code would need to be placed in the application itself and the macros deleted.

Recording Options

While recording a host macro, you can highlight a region or field with the mouse and right-click on the highlighted area. A popup menu appears with the following selections. These options are used to add control statements to the current macro during the recording process.

Add FindText. SM.FindText is used to determine if a specified text string is currently displayed on the host screen. It can be used to look for the text in a specific screen location or to search the entire host screen for the text. See Screen Mapping Extensions for details.

Add GetArea. This command gets the text off the screen in any rectangular area. With the option to trim the

result, selecting a column of data from the screen, parsing it and using it is much easier than getting all the data with several SM.GetText commands.

Add GetCursor. Used to determine where the cursor is currently located on the host screen. See Screen Mapping Extensions for details.

Add GetText. This selection adds SM.GetText which is used to retrieve text from the host screen at the column/row position established by the area highlighted by the mouse. See Screen Mapping Extensions for details.

Add WaitForCursor. This does the same as SM.WaitForText, only with the cursor; i.e., the script waits until the cursor reaches the specified location, before executing the next statement. See Screen Mapping Extensions for details.

Add WaitForCursorMove. This function is also used to time your commands to the host session. With this command, you specify only an amount of time in seconds. If the cursor has changed positions within that time, a True result is returned. Otherwise, it will timeout and return False. See Screen Mapping Extensions for details.

Add WaitForHost. SM.WaitForHost is used to time your commands to a vt220 host session. With it, you can delay sending text or keys to the host session, or retrieving data from the host session until the host has responded to the last command sent. See Screen Mapping Extensions for details.

Add WaitForText. Adds an SM.WaitForText statement to the macro, with the column/row position and the length being established by the area highlighted by the mouse, i.e., the script waits until the text appears at the specified location. See Screen Mapping Extensions for details.

Add WaitForWrite. This function waits for a specified number of seconds for data to be entered at a specific location and returns a True or False. If data was written within the wait time, True is returned. If the number of seconds expires first, False is returned. See Screen Mapping Extensions for details.

Select Current Field. Selects the current input field and records its attributes.

Building a Screen Mapping Application

Once the recorded macros are built, there are two ways they can be implemented.

Create an application with data fields, collect and validate the data and use the Embedded Procedure object to pass that data to the Host Screen macro. (In the code window there is a right-click option to insert embedded code. Select Transaction Macros as the data source and then pick the appropriate Host Screen macro. See the Embedded Procedure section for more details.) The macro passes back either a True or False (based on setting the function name “Transaction” equal to one value), indicating the success or failure of the macro to complete its script. Alter the Host Screen macro’s script to send back an appropriate Boolean value. This method isolates the code responsible for interacting with the host system which is ideal for version control and frequent updates to application code unrelated to the host system.

Take the code generated in the macro and place it in the application itself. This gives the application total control and has another advantage. If the login screen is a host screen that must allow different user credentials, then the solution cannot rely on the automatic logging in and navigation to the main menu. An application can be created that collects the user credentials and screen maps the data to the host login screen. Having already

recorded the macro, that code can be placed in the application directly and the macro may be deleted to avoid confusion. Taking this path requires that all applications be responsible for navigating and populating screens and fields on the host because the host is not automatically taken to a main menu, a generic starting point for the Host Screen macros. Since the Host Screen macros are linked to a Start Menu macro and they have been replaced by an application that performs that task after a custom login, Host Screen macros will not work. The solution is to create a Transaction macro with the same code, since it does not rely on links to previous macros.