



Datafree Max Developer Reference Guide

This document is provided as a reference for Datafree Max Application developers.

If you are not familiar with Datafree Max Application development, please refer to the Datafree Max - Getting Started Guide which explains the development process and provides example code to get you started.

Sample code can be found here: <https://github.com/binuadmin/biNu-sample-1>

Index

1. Platform Architecture	4
2. biNu Component Overview	5
Application configuration file - <app> element	5
Application Example Screens	7
Document configuration file - <doc> element	8
Document Example Screens	11
3. Element Descriptions	12
<app>	12
<drawer>	13
<bottom_nav>	13
<menu>	13
<share>	14
<tabs>	15
<style>	15
<doc>	15
<list>	17
	19
<md>	20
<bookmark>	20
<webview>	22
<item>	24
<filter>	25
<option>	25
<menuitem>	26
<msg>	27
4. Markdown and Style Reference	28
Markdown Coding Reference	29
biNu Default Stylesheet	30
Supported Style Properties	32
Markdown Styles - Example Code	33
Supported Fonts	34
Supported Colors	34
5. Design Considerations	35
Using the App Context	36

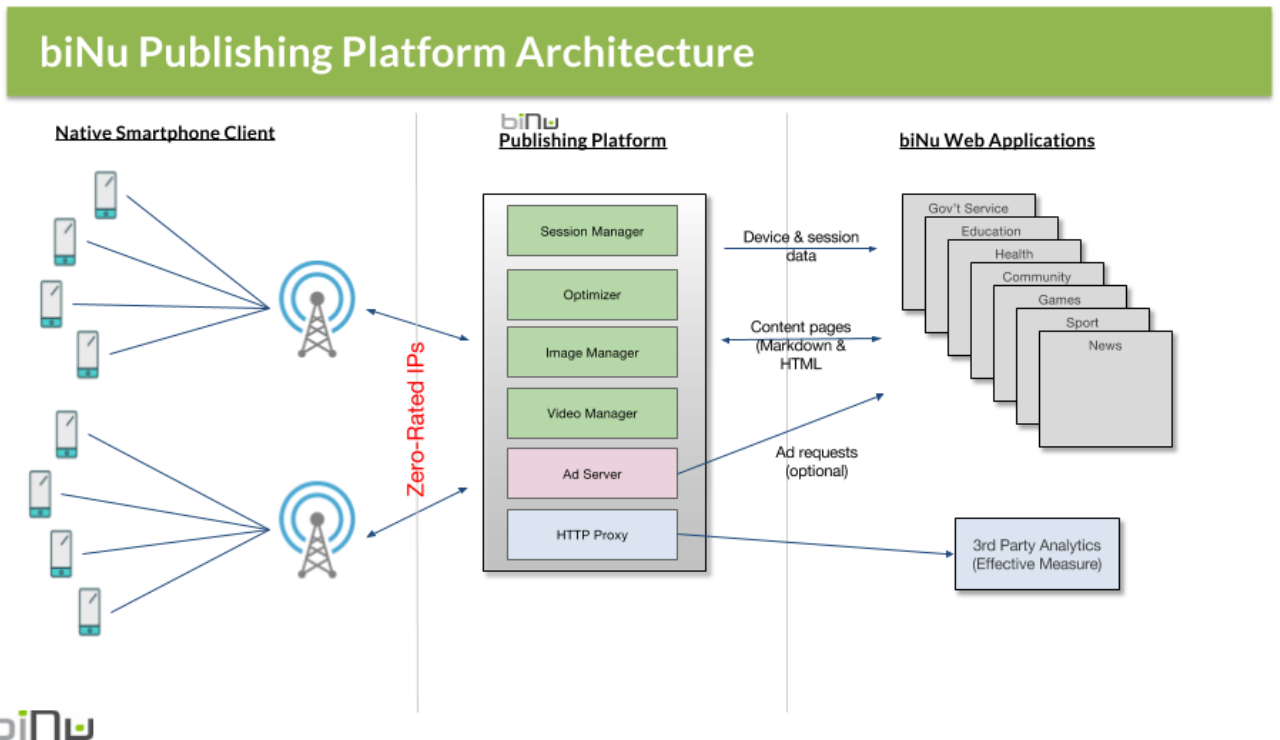


Handling Different Screen Sizes	37
HTML Support	37
Forms	38
Rules when coding forms	38
Caching	39
Cache Busters	39
Free Data	39
Using filters	40
Registering a new biNu Application	43
Onboarding Screens	43

1. Platform Architecture

The Datafree Max Publishing Platform is comprised of 3 major components:

- The biNu client App which is issued to you as a custom branded app to be published by the developer with their own certificate. The App is not modified by the developer directly; all configuration, layout and content information is supplied from the server-side application and rendered in the client App. biNu development is the creation of the server-side application. biNu supports Android 4.1 and above.
- The Datafree Max Optimisation Server provides optimised content to the smartphone App to reduce bandwidth. The server is developed and managed by biNu and is not configured by the developer .
- The server-side biNu Application which consists of XML files to define the configuration and layout of the client App and provide content to be rendered in the App. This is the only component that needs to be developed to create a biNu Application.



2. biNu Component Overview

A Datafree MAX Application is built from components which define the menus, style and content rendered in the biNu client App. Components are built from XML files.

Each component is listed in this section with an example however full details for each entity are available in the [Element Descriptions](#) section of this document.

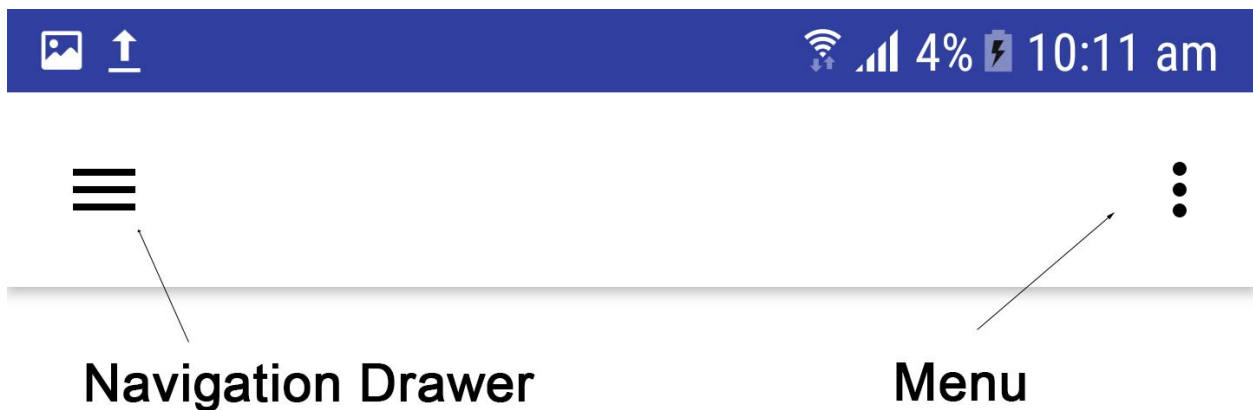
Application configuration file - `<app>` element

This file is the homepage URL specified within the generated client App. There can only be one Application configuration for the biNu Application.

e.g. `http://yourhost/yourbinuapplication/app.xml`

The Application configuration file is loaded when the App launches and is not changed or reloaded until the App is quit and restarted. It specifies the title, style and main navigation components which are:

- The Navigation Drawer at the top left of the screen - defines by the `<drawer>` element
- The Menu at the top right of the screen - defined by the `<menu>` element



Application Configuration Example

This `<app>` code loads 3 menu items into the Navigation Drawer, 3 menu items into the Menu and a stylesheet.

```
<app title="MyApplication"
styleurl="http://link_to_your_stylesheet.css">
<Drawer>
<MenuItem img="image_for_menu_item_1.jpg"
href="document_for_menu_item_1.xml">
Menu Item 1
</MenuItem>
<MenuItem img="image_for_menu_item_2.png"
href="document_for_menu_item_2.xml"
default="true">
Menu Item 2
</MenuItem>
<MenuItem img="image_for_menu_item_3.jpg"
href="document_for_menu_item_3.xml">
Menu Item 3
</MenuItem>
</Drawer>
<Menu>
<MenuItem action="usage">About</MenuItem>
<MenuItem action="share">Share App</MenuItem>
<MenuItem action="rate">Rate App</MenuItem>
</Menu>
<share action="send" title="biNu" text="Shared by biBu, check out this great
product http://www.bi.nu" subject="biNu Developer"/>
</app>
```

Notes:

There can be only one `<drawer>` and one `<menu>` entity.

If the share menu item is present there must be a `<share>` element in the code.

Each Menu Item references a Document configuration file `<doc>` which will be loaded into the page when the item is selected.

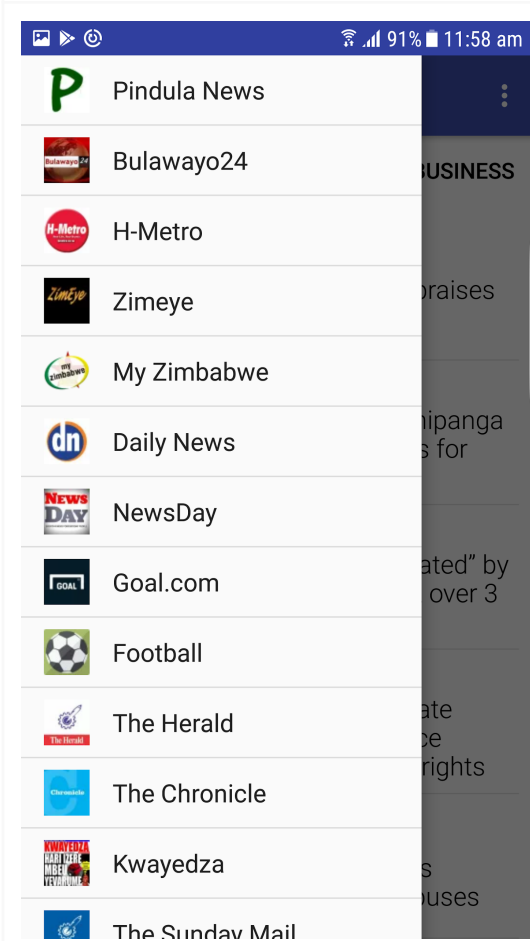
Menu Item 2 will be loaded on launch as it has the attribute `default` set to true.

Application Example Screens

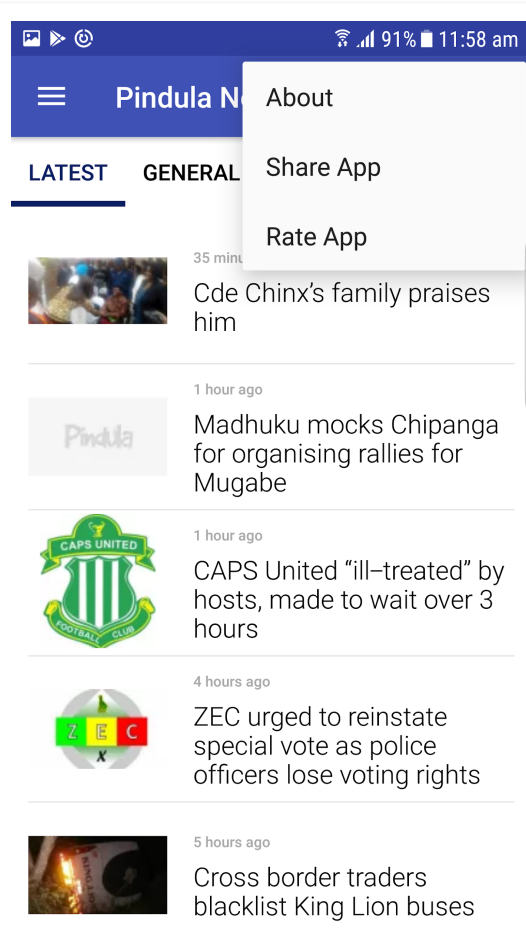
Example Title



Example Drawer



Example Menu



Document configuration file - `<doc>` element

The Document configuration file specifies what is displayed on the page. It can contain either Navigation or Content type elements but not both:

- Navigation** - either a `<list>` element OR a `<tabs>` element
- Content** - a combination of `<webview>`, `<md>`, `` elements
- optionally, one `<share>` element

Each child element is introduced in the following examples, for a full description refer to the **Element Description** section later in this document.

If the `<doc>` file is on the same domain as the `<app>` file it can be referenced using a relative URL, otherwise it must be a fully qualified path. These examples are static but in practice many `<doc>` files will be generated dynamically as they would typically extract data from various data sources and wrap it for biNu use.

Document List Example

This `<list>` example displays a list of 2 items, each item has an image and a line of text specified using Markdown

```
<doc title = "Document List Example">
  <list>
    <item layout="relative"
href="link_to_item_1.xml">
<img url="image_to_display_in_list.jpg"/>
    <md>Text to be displayed for item 1</md>
    </item>
    <item layout="relative"
href="link_to_item_2.xml">
    <img url="image_to_display_in_list.jpg"/>
<md>Text to be displayed for item 2</md>
  </item>
</list>
</doc>
```


Document Tab Example

This `<tabs>` example displays a row of 3 tabs at the top of the page. Each tab is defined by a menuitem which has the same behaviour as a menuitem in the Navigation Drawer.

```
<doc title='Tabbed Document Example'>
<tabs show="true" title="tabs">
<MenuItem href="link_to_tab_1.xml">Tab 1</MenuItem>
<MenuItem href="link_to_tab_2.xml"
default="true">Tab 2</MenuItem>
  </tabs>
</doc>
```

Document Markdown Example

This `<md>` example displays 2 blocks of Markdown text on the page.

```
<doc title="Markdown Example">
<md style = "category">
  <![CDATA[CATEGORY NAME]]>
  </md>
  <md style = "desc">
<![CDATA[
# header level 1
]]>
</md>
</doc>
```

Note: This Markdown contains special characters so it should be enclosed in CDATA tags to avoid issues when the XML is parsed. If you are not familiar with CDATA tags there is a tutorial here: https://www.tutorialspoint.com/xml/xml_cdata_sections.htm

Document Webview Example

This [webview](#) example displays an html form.

```
<webview target="browser" ><![CDATA[
  <h3>Leave a comment:</h3>
  <form name="fm1" action="http://yourprogram.php" method="get">
  Name: <br> <input type="text" name="name" value="John Doe"> <br>
    Rating: <br>
      <input type="radio" name="rating" value="1"> 1
      <input type="radio" name="rating" value="2"> 2
      <input type="radio" name="rating" value="3"> 3
  <br>Comment:
  <textarea name="comment">enter comments here ...</textarea>
  <br><input type="submit" value="Submit Comment">
</form>
]]></webview>
```

Note: CDATA tags are required for the webview content.

Document Image Example

This [webview](#) example displays an image [img](#) and has the description below in Markdown [md](#).

```
<doc title = "Document Title">
<img url="myimage.jpg"/>
<md>Image Description</md>
</doc>
```

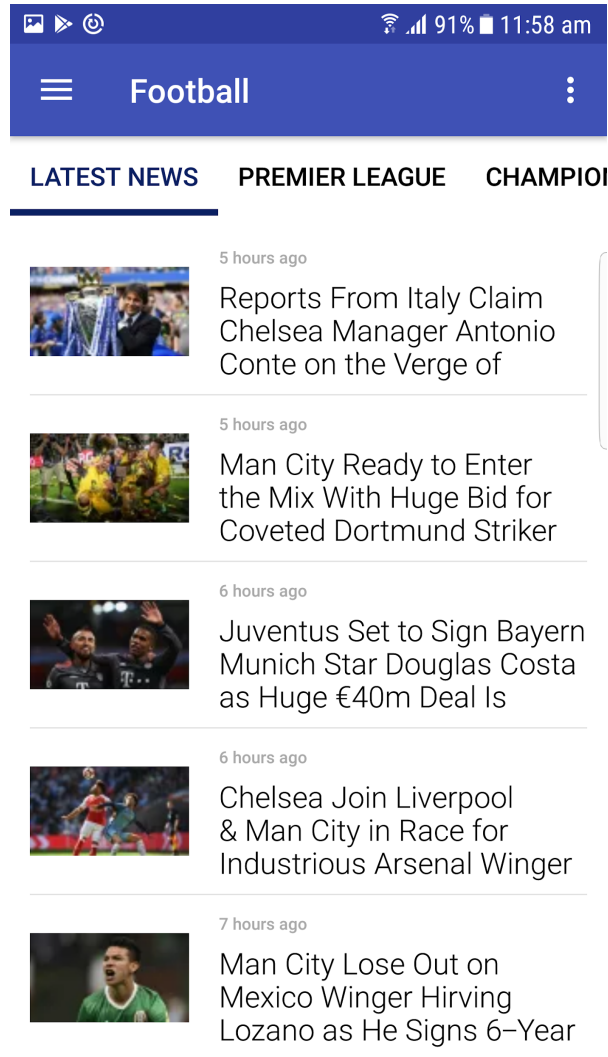
Document Share Example

This [share](#) example specifies the action to be taken when the document is shared

```
<share action="send" title="biNu" text="Shared by biBu, check out this great
product http://www.bi.nu" subject="biNu Developer"/>
```

Document Example Screens

Example Tabs and List Items



Example Image and Markdown



3. Element Descriptions

<app>

Specifies the title, style and main navigation components for the app.

Content: None

Attributes:

Attribute	Description	Notes
styleurl	URL of the stylesheet css file for this App. This will be merged with the built-in default css . Note. ignored if the style element is present.	May be the absolute URL or relative URL
title	App Name - displayed at the top of the screen	The title is not shown if the logo is present
logo	URL of the logo to displayed at the top of the screen	If present, the title is not shown. May be jpg or png file
showfree	The free data header displays the status of the connection eg wifi and also if the carrier is providing free data. More information here	Boolean 'true' or 'false'

Child Elements:

Element	Description	Cardinality
drawer	Defines the contents of the navigation drawer (left hand side)	1
bottom_nav	Defines the Bottom Navigation Bar for the App	0,1
menu	Defines the contents of the Menu (right hand side)	0,1
share	Specifies how the App should be shared.	0,1
style	Contains a custom stylesheet in css format to be merged with the built-in default stylesheet . The css defined in the styleurl attribute will be ignored if present	0,1
msg	Defines a popup message for display when app starts	0,1

<drawer>

Defines the list of menu items to be displayed in the Drawer Menu of the Application

Content: None

Attributes: None

Child Elements:

Element	Description	Cardinality
menuitem	Defines the content of each menu item in the drawer	1,many

<bottom_nav>

Defines the items to be displayed in the Bottom Navigation Bar of the Application
If present then the defaults on these menu items override the ones set in the drawer

Content: None

Attributes: None

Child Elements:

Element	Description	Cardinality
menuitem	Defines the content of each item in the Nav Bar Should only be between 2 and 5 items	1,many

<menu>

Specifies the menu items to be displayed on the (overflow) menu on the right hand side of the screen.
The menu is optional.

Content: None

Attributes:

Attribute	Description	Notes
style	The name of a class in the stylesheet to be applied to the menu	See style section of this document

Child Elements:

Element	Description	Cardinality
menuitem	Defines the content of each menu item in the menu	1,many

<share>

Specifies the action to be taken when the enclosing component (either the <app> or <doc>) is shared. The action enables the device to determine which applications can receive the share.

Content: None

Attributes:

Attribute	Description	Notes
action	Action to be taken for menu item	'usage' or 'share' or 'rate'
email	Sent to target application when shared	These attributes are sent with the share request and used as required by that application. E.g. a share to firefox will give the option to bookmark or open a URL if a URL exists is in the text attribute.
subject	Sent to target application when shared	
text	Sent to target application when shared	
title	Sent to target application when shared	

Child Elements: None

<tabs>

Specifies a list of tabs to be displayed across the top of the page.

Content: None

Attributes:

Attribute	Description	Notes
title	Title of the tab to be displayed	Truncated with ellipses (...) if too long
style	The name of a class in the stylesheet to be applied to the tabs	See style section of this document
showtab	Show or hide the actual tabs. Hiding them will still allow tab functionality - swipe left/right	"true" or "false"

Child Elements:

Element	Description	Cardinality
menuitem	Defines the content of each tab item	1,many

<style>

Provides the stylesheet to be used for this app. The styleurl for the app is ignored if this element is present. This stylesheet is merged with the built-in [default stylesheet](#) in the normal CSS behaviour.

Content: Valid CSS file

Attributes: None

Child Elements: None

<doc>

Specifies what is displayed on the page. Can contain either list style content or page style content but not both:

List Style: either one [list](#) element OR one [tabs](#) element

Page Style: - any combination of [webview](#), [md](#), [img](#) elements
- optionally, one [share](#) element

Content: None

Attributes:

Attribute	Description	Notes
title	The document title displayed. If autotab is on then the title will be derived from the List title if set otherwise it will be inherited from the source list's doc title.	

Child Elements:

Element	Description	Cardinality
list	A list of items to be displayed on the page. If present, must be the only child element	0,1
tabs	A list of menu items to be displayed as tabs across the top of the page If present, must be the only child element	0,1
img	An image to be displayed on the page	0,many
md	A block of Markdown to be displayed on the page	0,many
webview	An in-line block of html or url reference to a webpage to be displayed on the page	0,many
share	Specifies how the document should be shared. The share icon is only shown if this element is present	0,1
bookmark	Specifies if this doc can be bookmarked - the bookmark icon will be displayed in the status bar for this item. Only valid when doc is displayed in its own right i.e. not in a tab screen.	0,1

<list>

Specifies a list of items to be displayed on the page.

Content: None

Attributes:

Attribute	Description	Notes
style	The name of a class in the stylesheet to be applied to this list	See style section of this document
divider-height	Height of the divider between list items	0 = no divider
autotab	Autotab defines the way a list is transformed to a tab page when a list item is clicked.	<p>“hide” - the item is shown within the context of the list and responds to left and right swipes but does not show tab headers</p> <p>“show” - the item is shown within the context of the list and responds to left and right swipes and tab headers are shown</p> <p>“off” - the item is shown without the list context so swiping does not move through the list items</p>
id	The unique identifier for the list, it is used to determine when to send the selected items to the server - it is the name of the parameter to include those items in the call. See Using Filters	This is required if the list has a favourite filter element
action	If present, action can only be “fav” and it will then add favourite behaviour to the list. See Using Filters	“fav” or no action
layout	If action is present then this defines the position of the selector icons - eg stars Only valid option is “fav_r” which places the icons on the right of each list item. See Using Filters	“fav_r” or no layout

Child Elements:

Element	Description	Cardinality
item	Individual list item	0,many
filter	Defines a popup list header which is used to filter the list	0,1

	items. See Using Filters	
--	--	--

An image to be displayed. Can be png, jpg or svg.

Content: None

Attributes:

Attribute	Description	Notes
src	URL to the image	Relative or absolute URL
binuimgopt	Image optimisation parameters See Image Optimisation Guide	Not normally required
href	URL to be requested when this image is clicked. Note the href may optionally have the parameter 'binuvid' added to it to invoke video ads.	Optional
weight	The relative width of this <md> content when it is in a list <item>. The screen width of the <md> is calculated as actual screen width multiplied by this <md> weight value and divided by the sum of the weight attributes of all content components in the enclosing list item.	Only applicable when the is contained in a list <item>
mode	Specifies whether the image is downloaded and rendered automatically or manually. Default is to show the image, <code>mode="onclick"</code> will show a placeholder image and only download and display the image when clicked.	Optional, defaults to show image

Child Elements: None

<md>

A block of Markdown text to be displayed on the page. Supported Markdown tags and best practices are defined in the [Markdown Coding Reference](#) section.

Content: Markdown code (enclosed in CDATA tags)

Attributes:

Attribute	Description	Notes
style	The name of a class in the stylesheet to be applied to the tabs	See style section of this document
weight	The relative width of this <md> content when it is in a list <item>. The screen width of the <md> is calculated as actual screen width multiplied by this <md> weight value and divided by the sum of the weight attributes of all content components in the enclosing list item.	Only applicable when the <md> is contained in a list <item>
linkify	Controls whether any embedded links in the Markdown can be clicked as it would leave biNu and open the link in the browser	“true” or “false”

Child Elements: None

<bookmark>

Indicates that the doc can be bookmarked. The bookmark icon will be displayed in the status bar for this doc. If selected, this item will be listed when the bookmark drawer item is selected. Note that the bookmark drawer item is only present if there is at least on bookmarked document.

Content: None

Attributes:

Attribute	Description	Notes
days	The number of days to retain the bookmark	1-365
thumburl	The image to be displayed in the bookmarked items list - absolute or relative url	required
title	The text to be displayed in the bookmarked items list	required



Child Elements: None

<webview>

An in-line block of html or url reference to a webpage to be displayed on the page. Note that by default webview uses the Chrome rendering engine and appear as a page within the app. However, this behaviour can be modified by the proxymode parameter.

Content: HTML code (enclosed in CDATA tags) [nb: unless href attribute is present]

Attributes:

Attribute	Description	Notes
href	URL to an html file to load into the webview	If present there should not be webview content, it will be ignored
style	The name of a class in the stylesheet to be applied to this webview	See style section of this document
proxymode	<p>Defines the method used to make requests. Data will be either free or paid as follows:</p> <p>FREE DATA</p> <p>urlproxy - not supported opens a webview with all traffic sent to the urlproxy which is an httpclient and able to process https traffic. It can inspect the traffic and optionally: limit by filesize (in pub console) limit by mime type (in pub console) limit by domain (whitelist in pub console) Inspect urls within response and modify them (domainmappings in console)</p> <p>native - opens webview using the Chrome rendering engine and uses the standard proxy feature of the browser to send all traffic via the binu proxy</p> <p>inapp - this is used by the Max product and traffic is processed through the Max interceptors. This is single threaded. Cookies are shared between the webview and the app</p> <p>PAID DATA</p> <p>direct - opens webview and does not use a</p>	

	<p>proxy</p> <p>external - opens the webview using a Chrome custom tab which overlays the app</p>	
showprogress	Show the progress bar as the webview is loading	<p>“true” or “false”</p> <p>Default “true”</p>
baseurl	The base url to use for calls if the html contains relative paths	
disablejs	Disable javascript from running in the webview	<p>“true” or “false”</p> <p>Default “false”</p>
internal	<p>Specifies the behaviour of links within the HTML.</p> <p>If internal is true, the link must point to a valid biNu XML <doc> which will be opened as a new webview in biNu.</p> <p>If internal is false, the link must point to a valid HTML file and it will be opened in the same biNu webview if it is from the same domain as the webview otherwise it will open in the Browser.</p>	<p>“true” or “false”</p>
binuimgopt	Image optimisation parameters	<p>Not normally required.</p> <p>See Image Optimisation Guide</p> <p>To turn off optimisation - set to false</p>

Child Elements: None

<item>

A list item.

Content: None

Attributes:

Attribute	Description	Notes
href	URL to request when the item is tapped or is the default item	Relative or absolute URL
style	The name of a class in the stylesheet to be applied to this webview	See style section of this document
layout	A specific layout to use to display the item Relative layout places the components one above the other in the list Linear layout places the components side by side in the list	“relative”, “linear”, “flow”
id	The unique id of the row. This is required of the list has a favourite filter.	
tags	A comma separated list of tags that are used to determine what is shown when the list has a tag filter. See Using Filters	

Child Elements:

Element	Description	Cardinality
img	Image to display	0,1
md	Block of Markdown to display for the item	0,1
webview	A webview to display	0,1

<filter>

A list filter to enable the selective display of list items. See [Using Filters](#)

Content: None

Attributes:

Attribute	Description	Notes
id	The unique id of the filter.	Not used
show	Defines whether the filter pop up is shown at the top of the list.	
prompt	Text to be displayed in the filter popup box as a prompt	

Child Elements:

Element	Description	Cardinality
option	Individual popup item	1,many

<option>

A list filter option item. See [Using Filters](#)

Content: The text to be displayed for the option

Attributes:

Attribute	Description	Notes
value	The value to be selected in the list for display when the option is selected. For a favourite list there are 2 predefined values: #ALL and #SELECTED. For a tag list the value is a tag name.	
default	Specifies the option to be selected as the default when the page is loaded.	true or false

Child Elements: none

<menuitem>

A menu item.

Content: Menu Item Text

Attributes:

Attribute	Description	Notes
img	URL of the image for this menu item	Relative or absolute URL Optional Only valid for parent < drawer >
href	URL to be requested when this menu item is clicked Note the href may optionally have the parameter 'binuvid' added to it to invoke video ads.	Only valid for parent < drawer >
default	Automatically load this item when this menu is loaded	"true" or "false" Default "false" Only valid for parent < drawer >
style	The name of a class in the stylesheet to be applied to this menu item	See style section of this document Only valid for parent < drawer >
action	Action to take for this menu item	Valid in both drawer and menu
msg	Message to send with the action	Only valid for parent < menu >

Child Elements: None

<msg>

An optional message to be displayed when the app starts.

Content: The message to be displayed as CDATA

Attributes:

Attribute	Description	Notes
id	Unique ID for this message. If the message is to be displayed once it will not be displayed again until the id is changed. We advise the id is incremented to re-invoke a message.	Optional text
title	The title, shown at top of the message box	
action	View is the only option at the moment	"view"
onexit	Defines the action to be taken when the user dismisses the message box	"finish" - quit the app "continue" - continue to the app
mode	When is the message displayed, once, daily or always	"once" - displayed once and then not again unless the message id is changed Default "always"
style	The name of a class in the stylesheet to be applied to the message text	See style section of this document

Child Elements: none

4. Markdown and Style Reference

Markdown is a simple, lightweight markup for formatting text to be displayed on screen. Markdown tags have a similarity to HTML tags but there are far fewer Markdown tags. biNu supports most but not all standard markdown tags.

More information on Markdown can be found [here](#)
And here [The Ultimate Guide to Writing & Publishing with Markdown](#)

For example:

A Header 1 tag in HTML is:

```
<h1>Header Text</h1>
```

A Header 1 tag in Markup is:

```
## Header Text
```

Note 1: there must be a space between the # and the text

Note 2: two spaces at the end of text invoke a new line

Styles can be specified to control how the markdown is displayed. Styles are defined once in the application, in the Application stylesheet defined within the <app> tag. If not specified, biNu default styles are used. Styles in the stylesheet can be applied to Markdown or HTML tags but they are defined in the stylesheet by their HTML tag rather than their Markdown tag.

For example:

The stylesheet entry: `h1 {color:blue;}`
will apply to Header 1 tags in HTML and in Markdown

Note: Markup tags are never used in stylesheets.

There are limitations on the usage of styles in a biNu Application:

1. Nested styles are not supported
2. One style block is allowed per tag
3. The last style definition for the tag will take precedence

Markdown Coding Reference

This table shows the supported Markup tags and their HTML equivalents to be used in the stylesheet.

Style Name	Syntax	Style tag
Header	<code>#, ##, ###, ####, #####, #####</code>	h1, h2, h3, h4, h5, h6
Code block	<code>```\nCodeblock\n```</code>	codeblock
Italics	<code>*italics*</code>	em
Underline	<code>__underline__</code>	u
Link	<code>[Link text](URL)</code>	a
Internal Link	<p><code>[Link text](biNu URL)</code> biNu URL must return biNu XML and use either binu-http or binu-https protocol Relative urls will always be handled as internal and must return xml. Only root relative and document relative urls will be supported (i.e. not protocol relative urls) e.g. /path/app? param=val and path/app?param=val will be supported and must return xml //mydomain.com/path/app?param=val will not be supported at all</p> <p>Absolute urls will be external by default and open in the custom tab browser as they do now. However, they may be forced to be internal by inserting the binu- prefix</p>	a
Unorder list	<ul style="list-style-type: none"> * Unordered list item one * Unordered list item two 	ul
Blockquote	<code>> blockquote</code>	blockquote
Paragraph	<i>Double line gap</i>	p
Bold	<code>**bold**</code>	b

Image	![Alt Text](image url "title" =width%) Width is optional	img
Horizontal Rule	---	hr
Ordered List	1. Ordered list item one 2. Ordered list item two	ol
Strikethrough	~~strike~~	del
Body	This will define style for whole page	body

biNu Default Stylesheet

```
default {
  color: black;
  background-color: white;
  margin-left: 8;
  margin-right: 8;
  binulmgOpt: c_limit,q_auto:eco;
}

cards {
  background-color: lightgray;
  elevation: 4;
  orientation: vertical;
  columns: 1;
  padding: 2;
}
cards item { corner-radius: 8; text-align: left; margin-left: 8; margin-right: 8; font-size: 24; font-style:
italic; font-family: sans-serif; }

list {
  color: black;
  background-color: white;
  divider: black;
  orientation: vertical;
  padding-left: 12;
  padding-right: 12;
  padding-top: 8;
  padding-bottom: 8;
}
list.item { height: 100; text-align: center; padding-top: 0; padding-left: 0; padding-right: 0;
padding-bottom: 5; font-size: 12; font-style: italic; }

item { height: 100; text-align: center; padding-top: 0; padding-left: 0; padding-right: 0;
padding-bottom: 5; font-size: 12; font-style: italic; }

item.firstitem { height: -2; }

md {
  color:black;
  background-color: white;
  padding:24;
  vertical-align:middle;
}
md.h1 { color:black; font-size: 36; font-style: bold; font-family: sans-serif-thin; line-space:36;}
md.h2 { color:black; font-size: 24; font-style: normal; font-family: sans-serif-condensed;
```



```
line-space:48;}
md.h3 { color:black; font-size: 12; font-style: italic; font-family: sans-serif-light; line-space:5;}
md.h4 { color:black; }
md.h5 { color:black; }
md.h6 { color:black; font-style: bold-italic; }
md.p { color:black; font-size: 18; font-style: normal; font-family: sans-serif-light; line-space:28;}
md.code { color:black; font-size: 12; font-style: normal; font-family: monospace; }
md.em { color:black; font-size: 18; font-style: italic; }
md.em2 { color:black; font-size: 18; font-style: bold; }

tabs {
  color: black;
  selected-color: #ff0a2062;
  indicator-color: #ff0a2062;
  indicator-height: 4;
}

webview {
  width: 480;
  height:720;
  disable-js: 0;
  internal-only: 0;
  binulmgOpt: c_limit,q_auto:eco;
}
```


Supported Style Properties

Stylesheets in HTML provide a rich set of properties to align and format text and other elements on the page. biNu supports a subset of standard css properties which are listed in this table.

Note: style names may not the contain hyphen (-) character

Property	Elements	Purpose
color	item, md	Colour of font and horizontal rule. See Supported Colours
height	item	The height of element, in pixels if positive otherwise: -1 indicates constrain height to container height -2 indicates constrain height to contents of element
line-space	md	The spacing between text lines as a proportion of the text size eg 1.2 is 1.2 x normal text height
background-color	item, md	Background color of an element. See Supported Colours
text-align	Item, md	Change the alignment of text. Options: left (default), center, right
margin-left right top bottom	item, md	Change the margins of an element
Border-left right top bottom	Item, md	Change the border of an element
Padding-left right top bottom	Item, md	Change the padding of an element
font-size	Item, md	In Scaled Pixel(sp) , whole numbers only.
font-style	md	The font style. Options: normal, italic, bold, underline, strike-through
font-family	md	Name of the font. See Supported Fonts
max-lines	md	Maximum number of lines to display. Text will be truncated after this number of lines.

Markdown Styles - Example Code

Stylesheet entry

```
<style>
p { color: teal; background-color: silver; }
h1 { font-size:16px; color: red; }
h2 { font-size:10px; text-align: center; }
h3 { font-size:10px; color: purple; text-align:right; }
hr { color: grey; }
b { color: #FF4500; }
em { color: green; }
del { color: magenta; }
</style>
```

Markdown Code

```
<md><![CDATA[
A horizontal line follows this one
---
# This is a level 1 header
## This is a level 2 header

This is a ~~paragraph~~ innovation!
**And very important!**
*Did I mention it was important?*
***No but really... This is IMPORTANT!***

* Unordered List Item 1
* Unordered List Item 2

1 Ordered List Item 1
2 Ordered List Item 2
]]></md>
```

Supported Fonts

All the [fonts supported on android](#) are supported in a biNu Application. Fonts can be used along with text-style to make them bold and italics.

Supported Font	Font-family
Roboto Thin	sans-serif-thin
Roboto Light	sans-serif-light
Roboto Regular	sans-serif
Roboto Medium	sans-serif-medium
Roboto Black	sans-serif-black
Roboto Condensed Light	sans-serif-condensed-light
Roboto Condensed Regular	sans-serif-condensed
Noto Serif	serif
Droid Sans Mono	monospace
Cutive Mono	serif-monospace
Coming Soon	casual
Dancing Script	cursive
Carrois Gothic SC	sans-serif-smallcaps

Supported Colors

The color literal can be used to change color of font, horizontal rule and markown tag element background. Apart from using colour literals, any HEX value is also supported

Literal	Hex Value	Literal	Hex Value
black	#000000	fuchsia or magenta	#FF00FF
dark gray or dark grey	#444444	aqua	#00FFFF
gray or grey	#888888	Lime or green	#00FF00
light gray or light grey	#CCCCCC	maroon	#800000
white	#FFFFFF	navy	#000080
red	#FF0000	olive	#808000
green	#00FF00	purple	#800080
blue	#0000FF	silver	#C0C0C0
yellow	#FFFF00	teal	#008080
cyan	#00FFFF		

5. Design Considerations

From our extensive experience developing for high latency network with low capability phones, we have some tips to ensure your users have a great experience:

- Your Application should have a simple design which will work well on a low-end phone and consume less data.
- Design a page structure with good spacing between content and a simple and intuitive layout
- Use standard color styles, this is recommended by Android and helps to make the page more appealing to use [read more here](#)
- Images look pretty but consume lots of bandwidth. biNu optimises images for delivery to the client but you should try to minimise the size and number of images on a page as part of the design.
- Use markdown pages as much possible. Markdown is very quick to render and consumes less data
- Avoid complex functionality and page layouts which will usually slow down the apps, add to the mobile cost and quickly alienate end users

Using the App Context

The biNu Android App supplies context data in the request headers (X-Binu:) which may be used to customise the behaviour of the Application. The data is provided in a json string as follows:

```
X-Binu:{
  "did":"device id",
  "aid":"",
  "v":3,
  "ip":"10.106.107.74",
  "loc":{
    "lat":0,
    "long":0,
    "country":"au",
    "lang":"en-AU",
    "tz":"Australia/Sydney"
  },
  "device":{
    "make":"Samsung",
    "model":"Galaxy",
    "os":"7.1.1",
    "w":1080,
    "h":1794,
    "density":2.63
  },
  "conn":{
    "network":"Android",
    "simProvider":"Android",
    "type":"MOBILE-LTE",
    "superType":"MOBILE-UNKNOWN",
    "hni":"50502"
  },
  "app":{
    "id":"19",
    "ver":"34",
    "binuVer":"1.0.2.qa",
    "inst1":1499400289,
    "firstInst1":1499400289,
    "downloadSrc":""
  }
}
```

Handling Different Screen Sizes

Android provides a number of tools to help & develop apps to run on multiple screen sizes. It scales fonts, images etc to suit the screen size.

Android recognises 4 basic screen sizes; **Small, Medium, Large, Extra Large**

It is recommended that you design Application for medium screen sizes and test on other screen sizes to verify the design. Occasionally the structure of the page will need to be modified for the different screen sizes which should be done in the Application.

To utilise Android Screen scaling biNu interprets all font size definitions in the stylesheets as [Android's SP](#). SP sizes (font) are automatically scaled to look similar relative to screen size. However, they can also be scaled by the user through accessibility settings in global device settings e.g user with poor eyesight can set fonts as "Large". Developer should design their app to work with various font sizes selected by the user to avoid layout issues.

High quality/resolution images should be used irrespective of the screen sizes. biNu will automatically optimize and scale the images to appropriate size in order to minimize data. You are not needed to provide images in multiple formats nor write complex logic to select different images based on screen size etc. The biNu Image Optimiser Guide provides a full description of the optimisation process.

HTML Support

biNu supports content using HTML. In general HTML will use more data and render slower than markdown content, particularly on older phones.

HTML must be displayed within `<Webview>`. The webview can be in-line as part of a page, as a list item, within a card or appear as a full page. Javascript is supported. Note that all html must be enclosed in CDATA tags e.g. `<![CDATA[html code here]]>`

The HTML can reference its own Stylesheet defined within the HTML.

Forms

Forms in biNu are implemented in HTML and must be used from within webview (in-line or otherwise). All HTML Form elements can be used along with JavaScript.

Example form code:

```
<webview> <![CDATA[
  <h3>Leave a comment:</h3>
  <form name="comment1"
action="http://yourdomain.com/action/comment.php"
method="get">
```

```
    Name: <br>
<input type="text" name="name" value="John Doe"> <br>
    Rating: <br>
    <input type="radio" name="rating" value="1"> 1
    <input type="radio" name="rating" value="2"> 2
    <input type="radio" name="rating" value="3"> 3
<br> Comment: <br>
<textarea name="comment">enter comments here
</textarea> <br>
    <input type="submit" value="Submit Comment">
</form>
]]> </webview>
```

Rules when coding forms

There are some rules that must be obeyed for HTML forms to work with biNu:

1. The form must use the GET method. POST is not supported.
2. The action must always be provided and must use absolute URL, relative URLs are not supported.
3. File Upload is not supported.
4. An HTML form can exist only within one Webview.



Caching

Caching content is important as it helps to reduce bandwidth usage while enabling the user to read content offline. Caching pages can get tricky, you need to ensure content remains relevant by sending new content to user when requested but also try to avoid sending content that user already has.

You can use TTL (Time To Live) in page headers to control how long a page should be cached. Caching is managed by [Picasso](#) and [Retrofit](#). Images consume large amount of data and are therefore set to minimum TTL of 1 hour by the biNu platform to help conserve data.

Cache Busters

Sometimes content should not be cached and default caching may be a problem. To get around this, cache busting should be used. A Cache buster can be implemented by providing unique key at the end of a URL so each request is unique and therefore forces client to fetch the data from biNu. This can be masked in Analytic / Reporting systems so it doesn't distort reporting

Cache busting should be used judiciously as it can significantly increase mobile data usage.

Free Data

Add description here

Using filters

A list may have a filter to enable the user to choose which items are displayed in the list.

The filter has 2 distinct mechanisms for enabling selection and they may be combined in a filter, these are favourites and tags.

1. Favourite Filters

Favourites are enabled if the list action = "fav", each item will automatically display a favourite icon. If the user clicks the icon, the item becomes a selected (favourite) item and the item's id is saved into the application database for the given list id.

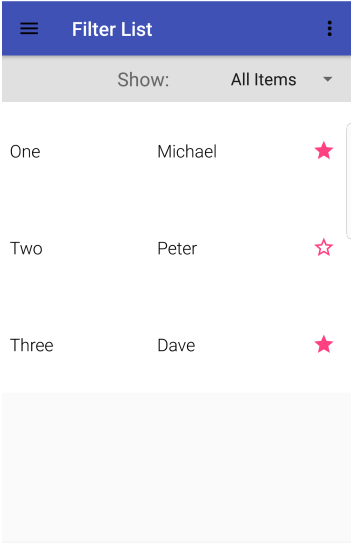
If the filter popup is shown at the top of the list, the user may choose to see all list items or just selected items.

Additionally, if any href in the application contains a list_id parameter, the saved item ids for the parameter's list id are automatically added to the request to enable them to be used in the application logic.

E.g. an href parameter of `list_id=myFavList` will tell the client to get any saved item ids in the list with the id=myFavList and add them to the url as the parameter of `list_val=item1,item2,item3`

Here is an example of a favourite filter in a list element

```
<doc title = "Filter List">
  <list style="mystyle" id="favList1" action="fav" layout="fav_r" >
    <filter id="filter1" show="true" prompt="Show:">
      <option value="#ALL">All Items</option>
      <option value="#SELECTED">Selected Items</option>
    </filter>
    <item id="1">
      <md>One</md>
      <md>Michael</md>
    </item>
    <item id="2">
      <md>Two</md>
      <md>Peter</md>
    </item>
    <item id="3">
      <md>Three</md>
      <md>Dave</md>
    </item>
  </list>
</doc>
```

	<p>Assume the user chooses item 1 and 3 as favourites and the item ids are stored for the list.</p>
	<p>When the list id is encountered in the code - such as:</p>
	<pre>href="http://myhost/myapplication?list_id=favList1"</pre>
	<p>The request will have the saved list ids added as follows:</p>
	<pre>http://myhost/myapplication?list_id=favList1&list_val=1,3</pre>

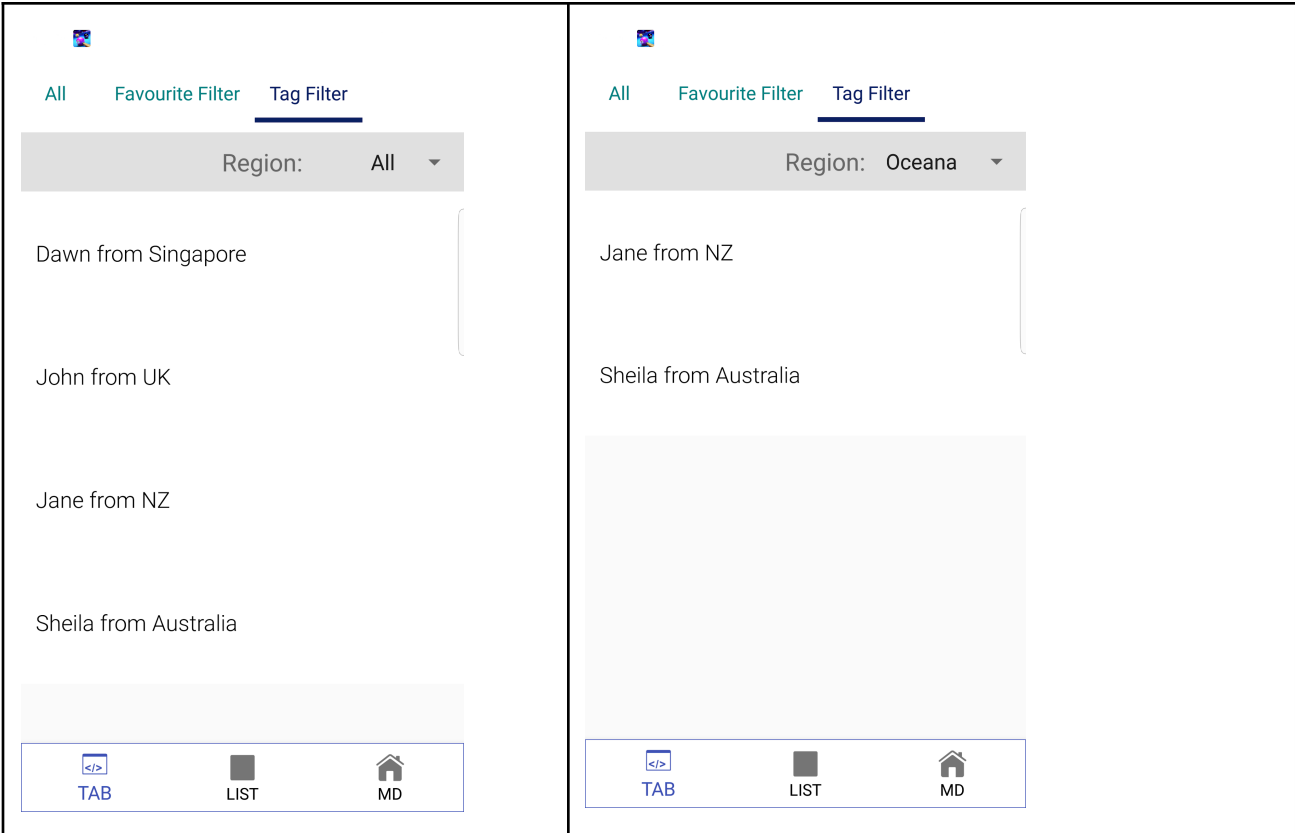
2. Tag Filters

Tag filters determine which items are shown based on selected tags associated with each item. The filter is used to dynamically hide and show items in the list but no data is saved for further use.

Here is an example of a tag filter in a list:

```
<list >
  <filter id="tagfilter" show="true" prompt="Region:">
    <option value="">All</option>
    <option value="oceana">Oceana</option>
    <option value="asia">Asia</option>
  </filter>
  <item tags="asia,world" >
    <md>Dawn from Singapore</md>
  </item>
  <item tags="europe,world" >
    <md>John from UK</md>
  </item>
  <item tags="oceana,world">
    <md>Jane from NZ</md>
  </item>
  <item tags="oceana,world">
    <md>Sheila from Australia</md>
  </item>
</list>
```

And here is how it looks



Registering a new biNu Application

The [Developer Portal](#) is used to manage biNu Applications.

To register a new Application, log on to the Developer Portal and choose 'Register New Application'. You will need to provide the following information which is 'burned' into the Android apk file when it is generated. Note that you can regenerate the App at any time if changes are required.

You will need to provide:

- the Application name and version;
- the Application package name (e.g. com.mycompany.application);
- a homepage URL pointing to a publicly available <app> XML document ;
- An onboarding URL which optionally provides as set of [onboarding screens](#)
- an app icon - a 120x120 pixel PNG file;
- a splash screen image - a 1080x1920 PNG image;

And optionally:

- Style elements for the action bar (at the very top of the page):
 - o Dark action icon colour - set this if the Action bar colour is white or very light
 - o Action bar text colour - the colour of the text in the action bar
 - o Action bar colour - the action bar background colour
- The colour of the status bar

Onboarding Screens

When the user first loads an new or updated apk on their device they can optionally be shown a set of pages that provide information on the use of the application, these are called onboarding screens.

In a biNu application, the onboarding screens are defined as a [tabs document](#) and the client code shows them at app launch and allows the user to swipe through the tabs or dismiss them altogether.

The url to the Onboarding tabs document is specified in the application configuration burnt into the apk when it is generated however the tabs page and associated doc pages are maintained as part of the publisher application and can be modified as such.

Here are 2 examples of onboarding screens.

Note that when they are presented to the user in the app , they are overlaid with page position dots

and a 'skip' link.

Free browsing

The free browsing icon indicates that no data will be used while browsing the app on Vodacom, Cell C and Telkom Mobile.

Low data usage

The low data usage icon indicates that a small amount of data will be used while using the app on all other networks.