

AAMO Framework

Developer Guide

Guide version: 1.0
AAMO version: 1.0
Author: Cleuton Sampaio



<http://www.thecodebakers.org>



CREATIVE COMMONS CORPORATION IS NOT A LAW FIRM AND DOES NOT PROVIDE LEGAL SERVICES. DISTRIBUTION OF THIS LICENSE DOES NOT CREATE AN ATTORNEY-CLIENT RELATIONSHIP. CREATIVE COMMONS PROVIDES THIS INFORMATION ON AN "AS-IS" BASIS. CREATIVE COMMONS MAKES NO WARRANTIES REGARDING THE INFORMATION PROVIDED, AND DISCLAIMS LIABILITY FOR DAMAGES RESULTING FROM ITS USE.

License

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. TO THE EXTENT THIS LICENSE MAY BE CONSIDERED TO BE A CONTRACT, THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1. Definitions

1. **"Adaptation"** means a work based upon the Work, or upon the Work and other pre-existing works, such as a translation, adaptation, derivative work, arrangement of music or other alterations of a literary or artistic work, or phonogram or performance and includes cinematographic adaptations or any other form in which the Work may be recast, transformed, or adapted including in any form recognizably derived from the original, except that a work that constitutes a Collection will not be considered an Adaptation for the purpose of this License. For the avoidance of doubt, where the Work is a musical work, performance or phonogram, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered an Adaptation for the purpose of this License.
2. **"Collection"** means a collection of literary or artistic works, such as encyclopedias and anthologies, or performances, phonograms or broadcasts, or other works or subject matter other than works listed in Section 1(f) below, which, by reason of the selection and arrangement of their contents, constitute intellectual creations, in which the Work is included in its entirety in unmodified form along with one or more other contributions, each constituting separate and independent works in themselves, which together are assembled into a collective whole. A work that constitutes a Collection will not be considered an Adaptation (as defined below) for the purposes of this License.
3. **"Creative Commons Compatible License"** means a license that is listed at <http://creativecommons.org/compatiblelicenses> that has been approved by Creative Commons as being essentially equivalent to this License, including, at a minimum, because that license: (i) contains terms that have the same purpose, meaning and effect as the License Elements of this License; and, (ii) explicitly permits the relicensing of adaptations of works made available under that license under this License or a Creative Commons jurisdiction license with the same License Elements as this License.
4. **"Distribute"** means to make available to the public the original and copies of the Work or Adaptation, as appropriate, through sale or other transfer of ownership.
5. **"License Elements"** means the following high-level license attributes as selected by Licensor and indicated in the title of this License: Attribution, ShareAlike.
6. **"Licensor"** means the individual, individuals, entity or entities that offer(s) the Work under the terms of this License.
7. **"Original Author"** means, in the case of a literary or artistic work, the individual, individuals, entity or entities who created the Work or if no individual or entity can be identified, the publisher; and in addition (i) in the case of a performance the actors, singers, musicians, dancers, and other persons who act, sing, deliver, declaim, play in, interpret or otherwise perform literary or artistic works or expressions of folklore; (ii) in the case of a phonogram the producer being the person or legal entity who first fixes the sounds of a performance or other sounds; and, (iii) in the case of broadcasts, the organization that transmits the broadcast.
8. **"Work"** means the literary and/or artistic work offered under the terms of this License including without limitation any production in the literary, scientific and artistic domain, whatever may be the mode or form of its expression including digital form, such as a book, pamphlet and other writing; a lecture, address, sermon or other work of the same nature; a dramatic or dramatico-



musical work; a choreographic work or entertainment in dumb show; a musical composition with or without words; a cinematographic work to which are assimilated works expressed by a process analogous to cinematography; a work of drawing, painting, architecture, sculpture, engraving or lithography; a photographic work to which are assimilated works expressed by a process analogous to photography; a work of applied art; an illustration, map, plan, sketch or three-dimensional work relative to geography, topography, architecture or science; a performance; a broadcast; a phonogram; a compilation of data to the extent it is protected as a copyrightable work; or a work performed by a variety or circus performer to the extent it is not otherwise considered a literary or artistic work.

9. **"You"** means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.
10. **"Publicly Perform"** means to perform public recitations of the Work and to communicate to the public those public recitations, by any means or process, including by wire or wireless means or public digital performances; to make available to the public Works in such a way that members of the public may access these Works from a place and at a place individually chosen by them; to perform the Work to the public by any means or process and the communication to the public of the performances of the Work, including by public digital performance; to broadcast and rebroadcast the Work by any means including signs, sounds or images.
11. **"Reproduce"** means to make copies of the Work by any means including without limitation by sound or visual recordings and the right of fixation and reproducing fixations of the Work, including storage of a protected performance or phonogram in digital form or other electronic medium.

2. Fair Dealing Rights. Nothing in this License is intended to reduce, limit, or restrict any uses free from copyright or rights arising from limitations or exceptions that are provided for in connection with the copyright protection under copyright law or other applicable laws.

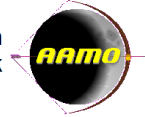
3. License Grant. Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

1. to Reproduce the Work, to incorporate the Work into one or more Collections, and to Reproduce the Work as incorporated in the Collections;
2. to create and Reproduce Adaptations provided that any such Adaptation, including any translation in any medium, takes reasonable steps to clearly label, demarcate or otherwise identify that changes were made to the original Work. For example, a translation could be marked "The original work was translated from English to Spanish," or a modification could indicate "The original work has been modified.";
3. to Distribute and Publicly Perform the Work including as incorporated in Collections; and,
4. to Distribute and Publicly Perform Adaptations.
5. For the avoidance of doubt:
 - a. **Non-waivable Compulsory License Schemes.** In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme cannot be waived, the Licensor reserves the exclusive right to collect such royalties for any exercise by You of the rights granted under this License;
 - b. **Waivable Compulsory License Schemes.** In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme can be waived, the Licensor waives the exclusive right to collect such royalties for any exercise by You of the rights granted under this License; and,
 - c. **Voluntary License Schemes.** The Licensor waives the right to collect royalties, whether individually or, in the event that the Licensor is a member of a collecting society that administers voluntary licensing schemes, via that society, from any exercise by You of the rights granted under this License.

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. Subject to Section 8(f), all rights not expressly granted by Licensor are hereby reserved.

4. Restrictions. The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

1. You may Distribute or Publicly Perform the Work only under the terms of this License. You must include a copy of, or the Uniform Resource Identifier (URI) for, this License with every copy



of the Work You Distribute or Publicly Perform. You may not offer or impose any terms on the Work that restrict the terms of this License or the ability of the recipient of the Work to exercise the rights granted to that recipient under the terms of the License. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties with every copy of the Work You Distribute or Publicly Perform. When You Distribute or Publicly Perform the Work, You may not impose any effective technological measures on the Work that restrict the ability of a recipient of the Work from You to exercise the rights granted to that recipient under the terms of the License. This Section 4(a) applies to the Work as incorporated in a Collection, but this does not require the Collection apart from the Work itself to be made subject to the terms of this License. If You create a Collection, upon notice from any Licensor You must, to the extent practicable, remove from the Collection any credit as required by Section 4(c), as requested. If You create an Adaptation, upon notice from any Licensor You must, to the extent practicable, remove from the Adaptation any credit as required by Section 4(c), as requested.

2. You may Distribute or Publicly Perform an Adaptation only under the terms of: (i) this License; (ii) a later version of this License with the same License Elements as this License; (iii) a Creative Commons jurisdiction license (either this or a later license version) that contains the same License Elements as this License (e.g., Attribution-ShareAlike 3.0 US); (iv) a Creative Commons Compatible License. If you license the Adaptation under one of the licenses mentioned in (iv), you must comply with the terms of that license. If you license the Adaptation under the terms of any of the licenses mentioned in (i), (ii) or (iii) (the "Applicable License"), you must comply with the terms of the Applicable License generally and the following provisions: (I) You must include a copy of, or the URI for, the Applicable License with every copy of each Adaptation You Distribute or Publicly Perform; (II) You may not offer or impose any terms on the Adaptation that restrict the terms of the Applicable License or the ability of the recipient of the Adaptation to exercise the rights granted to that recipient under the terms of the Applicable License; (III) You must keep intact all notices that refer to the Applicable License and to the disclaimer of warranties with every copy of the Work as included in the Adaptation You Distribute or Publicly Perform; (IV) when You Distribute or Publicly Perform the Adaptation, You may not impose any effective technological measures on the Adaptation that restrict the ability of a recipient of the Adaptation from You to exercise the rights granted to that recipient under the terms of the Applicable License. This Section 4(b) applies to the Adaptation as incorporated in a Collection, but this does not require the Collection apart from the Adaptation itself to be made subject to the terms of the Applicable License.
3. If You Distribute, or Publicly Perform the Work or any Adaptations or Collections, You must, unless a request has been made pursuant to Section 4(a), keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or if the Original Author and/or Licensor designate another party or parties (e.g., a sponsor institute, publishing entity, journal) for attribution ("Attribution Parties") in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; (ii) the title of the Work if supplied; (iii) to the extent reasonably practicable, the URI, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and (iv) , consistent with Section 3(b), in the case of an Adaptation, a credit identifying the use of the Work in the Adaptation (e.g., "French translation of the Work by Original Author," or "Screenplay based on original Work by Original Author"). The credit required by this Section 4(c) may be implemented in any reasonable manner; provided, however, that in the case of a Adaptation or Collection, at a minimum such credit will appear, if a credit for all contributing authors of the Adaptation or Collection appears, then as part of these credits and in a manner at least as prominent as the credits for the other contributing authors. For the avoidance of doubt, You may only use the credit required by this Section for the purpose of attribution in the manner set out above and, by exercising Your rights under this License, You may not implicitly or explicitly assert or imply any connection with, sponsorship or endorsement by the Original Author, Licensor and/or Attribution Parties, as appropriate, of You or Your use of the Work, without the separate, express prior written permission of the Original Author, Licensor and/or Attribution Parties.
4. Except as otherwise agreed in writing by the Licensor or as may be otherwise permitted by applicable law, if You Reproduce, Distribute or Publicly Perform the Work either by itself or as part of any Adaptations or Collections, You must not distort, mutilate, modify or take other



derogatory action in relation to the Work which would be prejudicial to the Original Author's honor or reputation. Licensor agrees that in those jurisdictions (e.g. Japan), in which any exercise of the right granted in Section 3(b) of this License (the right to make Adaptations) would be deemed to be a distortion, mutilation, modification or other derogatory action prejudicial to the Original Author's honor and reputation, the Licensor will waive or not assert, as appropriate, this Section, to the fullest extent permitted by the applicable national law, to enable You to reasonably exercise Your right under Section 3(b) of this License (right to make Adaptations) but not otherwise.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTIBILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination

1. This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Adaptations or Collections from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.
2. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

1. Each time You Distribute or Publicly Perform the Work or a Collection, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.
2. Each time You Distribute or Publicly Perform an Adaptation, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.
3. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.
4. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.
5. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.
6. The rights granted under, and the subject matter referenced, in this License were drafted utilizing the terminology of the Berne Convention for the Protection of Literary and Artistic Works (as amended on September 28, 1979), the Rome Convention of 1961, the WIPO Copyright Treaty of 1996, the WIPO Performances and Phonograms Treaty of 1996 and the Universal Copyright Convention (as revised on July 24, 1971). These rights and subject matter take effect in the relevant jurisdiction in which the License terms are sought to be enforced

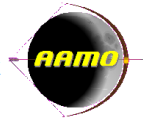


according to the corresponding provisions of the implementation of those treaty provisions in the applicable national law. If the standard suite of rights granted under applicable copyright law includes additional rights not granted under this License, such additional rights are deemed to be included in the License; this License is not intended to restrict the license of any rights under applicable law.

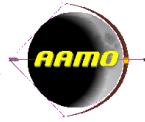
Creative Commons Notice

Creative Commons is not a party to this License, and makes no warranty whatsoever in connection with the Work. Creative Commons will not be liable to You or any party on any legal theory for any damages whatsoever, including without limitation any general, special, incidental or consequential damages arising in connection to this license. Notwithstanding the foregoing two (2) sentences, if Creative Commons has expressly identified itself as the Licensor hereunder, it shall have all rights and obligations of Licensor.

Except for the limited purpose of indicating to the public that the Work is licensed under the CCPL, Creative Commons does not authorize the use by either party of the trademark "Creative Commons" or any related trademark or logo of Creative Commons without the prior written consent of Creative Commons. Any permitted use will be in compliance with Creative Commons' then-current trademark usage guidelines, as may be published on its website or otherwise made available upon request from time to time. For the avoidance of doubt, this trademark restriction does not form part of the License. Creative Commons may be contacted at <http://creativecommons.org/>.



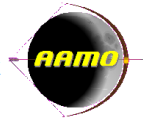
- [License](#)
- [Creative Commons Notice](#)
- [Overview](#)
 - [Basic structure of an AAMO application](#)
- [What is necessary to develop AAMO apps](#)
 - [About the Lua language](#)
- [How to create a new AAMO project](#)
 - [Creating an Android Project](#)
 - [Creating an iOS project](#)
- [Internationalization and Localization](#)
- [Screen definition file](#)
 - [UI](#)
 - [MENU](#)
 - [Events](#)
 - [Functions](#)
 - [LABEL](#)
 - [Events](#)
 - [Functions](#)
 - [TEXTBOX](#)
 - [Events](#)
 - [Functions](#)
 - [BUTTON](#)
 - [Events](#)
 - [Functions](#)
 - [CHECKBOX](#)
 - [Events](#)
 - [Functions](#)
 - [LISTBOX](#)
 - [Events](#)
 - [Functions](#)
 - [WEBBOX](#)
 - [Events](#)
 - [Functions](#)
 - [IMAGEBOX](#)
 - [Events](#)
 - [Functions](#)
- [Application Programming Interface](#)
 - [addListBoxOption](#)
 - [clearListBox](#)
 - [close](#)
 - [closeDatabase](#)
 - [eof](#)
 - [execSQL](#)
 - [exitScreen](#)
 - [getCheckBox](#)
 - [getCurrentScreenId](#)
 - [getError](#)
 - [getGlobalParameter](#)
 - [getLabelText](#)
 - [getLocalizedText](#)
 - [getTextField](#)
 - [loadScreen](#)



[log](#)
[navigateTo](#)
[next](#)
[OpenDatabase](#)
[query](#)
[setCheckBox](#)
[setGlobalParameter](#)
[setLabelText](#)
[setTextField](#)
[showMenu](#)
[showMessage](#)
[showScreen](#)

[Databases](#)

[Database messages](#)



Overview

"Aamo" means "moon" in the language of the Xavante Indians.

AAMO The framework is a platform for creating applications for mobile devices, such as those based on Android (Google) or iOS (Apple).

With AAMO, you create your application in a fast and simple fashion, using non verbose language, and can execute it immediately on many mobile platforms.

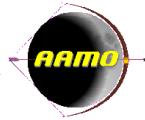
An AAMO application consists of XML files that describe the screens and their controls (UI), and optionally files containing source code in Lua. Lua is a modern programming language, designed to run "embedded" (within other software), small, fast and complete. Moreover, it is used by many frameworks and products worldwide.

Basic structure of an AAMO application

The simplest application is composed only by a single XML file. Let's create a "Hello World" using the AAMO. To begin, create a text file and save it as: "ul.xml." Then type the following contents:

```
<?xml version="1.0" encoding="UTF-8"?>
<ui>
  <version>0.3</version>
  <uuid>1</uuid>
  <title>Teste AAMO</title>
  <element>
    <id>1</id>
    <type>2</type>
    <percentTop>0</percentTop>
    <percentLeft>0</percentLeft>
    <percentWidth>20</percentWidth>
    <percentHeight>10</percentHeight>
    <text>Name</text>
  </element>
  <element>
    <id>2</id>
    <type>1</type>
    <percentTop>0</percentTop>
    <percentLeft>21</percentLeft>
    <percentWidth>35</percentWidth>
    <percentHeight>10</percentHeight>
  </element>
  <element>
    <id>3</id>
    <type>3</type>
    <percentTop>20</percentTop>
    <percentLeft>40</percentLeft>
    <percentWidth>20</percentWidth>
    <percentHeight>10</percentHeight>
    <text>Ok</text>
    <onClickScript>lua::aamo.showMessage("Hello " .. aamo.getTextField(2))</onClickScript>
  </element>
</ui>
```

When executed, this application displays the following screen:



You can run it on any device for which the AAMO framework is available. Currently, only works on Android and iOS (iPad, iPhone and iPod touch).

(see the chapter on installation to run this application on the device of your choice)

Let us understand it's structure. To begin, you must understand the "language" of AAMO, ie the framework's concepts..

Application

An application AAMO is a set of text files, all contained in the "app" folder (Android: "/" assets / app" in iOS is the group "app").

An application can be composed of one or more screens (XML files), zero or more event handling files (files "*. lua"), zero or more schema files database (in XML) and zero or more translation files ("properties").

An application AAMO is completely portable between platforms, simply copy the contents of the "app". It is not necessary to change the application to run it on Android or iOS.

Screen

For AAMO everything the user sees in the "display" device at any given time is a "screen". No more complicated names for simple things ("View", "Activity" etc).

An application can have one or more screens. To be executed by AAMO, an application must have at least one screen.



The screen has an identifier (`<uuid> 1 </ uuid>`) and must be named "ui.xml". Moreover, it is loaded automatically by AAMO.

Other screens can be numbered from 2 "ui_2.xml", "ui_3.xml" etc.. The display number must appear on the tag "`<uuid>`" and the file name, for example, the file "ui_2.xml" must have a tag "`<uuid> 2 </ uuid>`".

Control

A screen can have one or more controls. A control is a visual element for user interaction. Currently there are seven types of controls:

- TEXTBOX (type 1): field for entering text;
- LABEL (type 2): a label;
- BUTTON (type 3): a touchable button;
- CHECKBOX (type 4): a binary option box;
- LISTBOX (type 5): a list of selectable items;
- WebBox (type 6): mini-web browser;
- IMAGEBOX (type 7): to display touchable images;

A control is identified by its "id", which is a sequential number (starting from 1). There can't be two controls with the same "id" on the same screen.

Events and treatment

AAMO is based on events. The code in your application runs as a result of an events happening. To deal with the events there are two ways: 1) you write the handling code within the XML file, or 2) you write event handling file (with the "lua" extension).

Analyzing the structure of the sample application

The small sample application will allow us to understand AAMO structure. To begin with, let's analyze the elements of screen definition tag (`<ui>`):

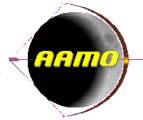
```
<ui>
  <version>0.3</version>
  <uuid>1</uuid>
  <title>AAMO Test</title>
```

We are defining a screen, which is loaded automatically at the beginning of the application (uuid = 1). The tag "VERSION" indicates what is the minimum version of AAMO to run this application, and the tag "`<uuid>`" indicates the number of the screen, which must appear in the file name. For the first screen, this is not true because the name must be: "ui.xml." For other screens, their number must come after a character "_", eg "ui_2.xml."

There are more elements that can be set on the screen, such as event handlers, menu background color, but we'll leave for later.

The controls are declared with the tag "`<element>`", identified by the "id" and type subelements of the control definition. We also need to configure their placement and size. For example:

```
<element>
  <id>1</id>
  <type>2</type>
  <percentTop>0</percentTop>
  <percentLeft>0</percentLeft>
```



```
<percentWidth>20</percentWidth>
<percentHeight>10</percentHeight>
<text>Name</text>
</element>
```

Here we declare a "LABEL" control (type 2), and indicate its content ("`<text> Name </ text>`"). However, the main properties are:

- "PercentTop" position relative to the top of the screen;
- "PercentLeft" position relative to the left of the screen;
- "PercentWidth" width, as a percentage of the screen's width;
- "PercentHeight": height, as a percentage of the screen's height;

Everything is defined as a percentage relative to the size of the screen. This ensures that an AAMO app shows proportionally in any type of device, from a smartphone to a tablet. The tags "`<percentTop>`" and "`<percentLeft>`" indicate the coordinates of the upper left corner of the control, pointing out that, on mobile devices, the value of the y-axis is zero at the top and grows downward. Thus, the upper left corner is positioned based on the percentage of screen height (`percentTop`) and the percentage of screen width (`percentLeft`).

The other two tags indicate the width (`percentWidth`) and height (`percentHeight`) control, all based on percentages of the width and height of the device screen.

Some types of controls such as Label and TextBox, have extra properties, such as "`<text>`", and event handlers:

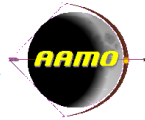
```
<element>
  <id>3</id>
  <type>3</type>
  <percentTop>20</percentTop>
  <percentLeft>40</percentLeft>
  <percentWidth>20</percentWidth>
  <percentHeight>10</percentHeight>
  <text>Ok</text>
  <onClickScript>lua::aamo.showMessage("Hello " .. aamo.getTextField(2))</onClickScript>
</element>
```

This is another control type: a BUTTON. Besides the "text" property, it also allows adding an event handler: "onClickScript." When the user touches the button, AAMO will invoke the handling specified in the "onClickScript" property. In this case, is an immediate command:

```
aamo.showMessage ("Hello" .. aamo.getTextField (2))
```

The event handling can consist of immediate commands, which are stored in the XML file itself, or external routines written in Lua. For example, to change the program to use an external routine, just inform its name:

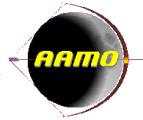
```
<element>
  <id>3</id>
  <type>3</type>
  <percentTop>20</percentTop>
  <percentLeft>40</percentLeft>
  <percentWidth>20</percentWidth>
  <percentHeight>10</percentHeight>
  <text>Ok</text>
  <onClickScript>clicked</onClickScript>
</element>
```



And we would create an external routine called "clicked.lua" with the following contents:

```
aamo.showMessage ("Hello" .. aamo.getTextField (2))
```

What differentiates one from another is the presence of the prefix "lua:." at the beginning of the content of the tag.



What is necessary to develop AAMO apps

You need to develop the application, embed it within a "cradle" project, and distribute under the rules specific for each platform (Android or iOS).

AAMO uses the concept of "cradle" projects, which are native to each platform. You copy the project (we will show how it should be done) and include your application in the "app" folder. No need to change the code. In fact, you even don't need to know the native languages and environments of each platform.

AAMO is open source and free, so you do not need to buy any license for development and distribution. The only requirement is that you distribute the license file (license.txt ...) that come along with each "cradle" project.

About the Lua language

This manual does not teach the Lua programming language. If you want to learn it, there are many books and tutorials on the Internet, including:

- <http://www.lua.org/about.html>;
- <http://www.lua.org/manual/5.1>;



How to create a new AAMO project

AAMO is absolutely impartial about the mobile platform. You can develop using Android and then run it using iOS, no problem, and no need to modify your application.

If you use AAMO Editor, you already will have a Java/Swing “cradle”, which runs on desktop, so there is no need to download a “cradle” project to develop your application. AAMO Editor is not covered in this manual, so if you're interested, we suggest consulting the appropriate documentation at the sites:

- <http://www.aamoframework.org>;
- <http://www.thecodebakers.org>;
- <http://code.google.com/p/aamoframework/>;

Creating an Android Project

Download the Android cradle project (see: <http://code.google.com/p/aamoframework/downloads/list>), open "eclipse" (properly configured with the "ADT" plugin), and select "import / general / Existing projects into Workspace ", pointing to the AAMO zip file.

After doing this, you have a copy of the cradle project, with the name and the original AAMO configuration. If you want to make a quick test, it's ok. But if you want to distribute your project, you better change a few things:

1) Rename the application package:

Select your project right, open the "Android Tools" and select "Rename Application Package"; Enter the new package name. Warning: this is the name of your application on Google Play, pay attention!

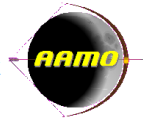
2) Change the version of its application in AndroidManifest.xml:

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="org.thecodebakers.testeaamo"
    android:versionCode="1"
    android:versionName="1.0">
```

In this version (0.3 or 1.0 alpha), AAMO only works with version 2.2 of the Android API. After making the changes, just copy ALL files from your application to the folder "assets / app".

Creating an iOS project

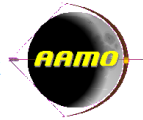
Download AAMO iOS cradle project (see: <http://code.google.com/p/aamoframework/downloads/list>), and unpack it in the folder of your choice. Open Xcode and open the project



on it. In the "Project Navigator" click twice slowly on the project's root folder (the first, in blue), and change its name.

If you want, select the Target and change the package name, in the "Bundle identifier". Also change the fields "Version" and "Build".

Open the group, "app" and delete all files that are there, then paste your application's files.



Internationalization and Localization

Internationalization (i18n) is the process of preparing an application to use multiple languages and preferences in accordance with different cultures. Localization (L10n) is the process of adapting an internationalized application to a specific language and culture.

AAMO supports both techniques in a simple and transparent way. First, create the translation files, specifying the language and country. These files are like property files, and have the syntax:

<text key> = <translation>

Example:

```
test=Translated test
locmsg=Default message
loclista=Beach
first=First
murl=http://pt.wikipedia.org/wiki/Brasil
```

The XML file screen, and the Lua code can refer to localized text. For example, we can create a LABEL with a localized text:

```
<text>l10n::test</text>
```

Any time AAMO recognizes the prefix "l10n" (L + 10 + N) on a property, it will search the text in the active translation file.

The properties that currently accept translation prefix are:

- text: BUTTON, LABEL, TEXTBOX;
- url: WebBox;
- picture: IMAGEBOX;
- option: <menu>;

Within the Lua source code, we can use the "aamo.getLocalizedText ()" function:

text = aamo.getLocalizedText ("<text key>")

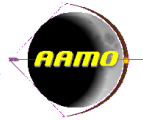
The default translation file is named:

```
aamol10n.properties
```

This file must always be kept with the other application files, inside the folder "app."

We can create files by language, for example:

- aamol10n_en.properties;



- aamol10n_pt.properties;
- aamol10n_fr.properties;

Just add the suffix "_" with the language code based on ISO 639-1 (http://www.w3schools.com/tags/ref_language_codes.asp).

We can also "specialize" the translation, adding the ISO 3166 country code (http://userpage.chemie.fu-berlin.de/diverse/doc/ISO_3166.html).

We may have the files:

- aamol10n_en_GB.properties;
- aamol10n_en.properties;
- aamol10n.properties;

AAMO will always try to find the complete file (language and country). If not possible, AAMO will seek the file with appropriate language suffix, and if not found, AAMO will search for the default file (aamol10n.properties). AAMO only looks for the files when it finds translated text referece ("l10n" prefix or aamo.getLocalizedText() function), otherwise, it is not necessary to create translation files.



Screen definition file

UI

Every UI file must start with the AAMO tag "<ui>." The elements of this tag are:

- "menu": defines the screen's menu;
- "version": specifies the AAMO version for which the screen was written;
- "uuid": the identifier of the screen, which should be compatible with your file name;
- "title": currently just a comment!
- "onLoadScript": screen's load event handling;
- "onEndScript": screen's disposal event handler;
- "onLeaveScript": screen's pause event handler;
- "onBackScript": screen's reactivation event handling;
- "backgroundColor": screen's background color;

You need to understand how AAMO deals with screens. It has a screen "stack" (LIFO). When you load a screen, the previous screen goes down one position on the stack. When a screen is disposed, the previous screen climbs up one position on the stack and gets displayed.

When a screen is loaded, ie did not exist on the stack, its event handling "onLoadScript" is invoked. It will only be invoked again if the screen is disposed from the stack and reloaded again. A disposed screen ceases to exist in memory.

When a screen is disposed, its "onEndScript" event handling is invoked. This occurs for all screens EXCEPT for the first screen (ui.xml). The first screen will NEVER have its "onEndScript" event handling invoked.

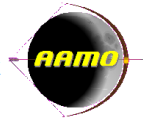
The event "onLeaveScript" is executed whenever a screen invokes another. In this case, the original screen goes down one position in the stack and, therefore, it's considered "on pause". Its "onLeaveScript" handling will be invoked.

The event "onBackScript" occurs whenever a screen returns to the top of the stack, unless it is being loaded at this time (recent loaded screens receive the "onLoadScreen" event). If the screen "A" loads the screen "B", then the "onLeaveScript" event occurs on the "A" screen. By disposing the screen "B", and returning to the screen "A", its "onBackScript" event will be invoked.

Note that event handlers are completely optional.

The "backgroundColor" property lets you specify the background color of the screen in hexadecimal HTML code (http://www.w3schools.com/html/html_colornames.asp). For example:

- #000000: black;
- #FFFFFF: white;
- #FF0000: red;
- #00FF00: green;
- #0000FF: blue;



- #D3D3D3: light gray;

It is the RGB format (Red, Green, Blue), which specify the amount of red, green and blue for each color. Each two characters specify the amount. See website below:

http://www.w3schools.com/html/html_colornames.asp

MENU

The menu is a kind of control. There can be only one menu per screen. We specify menus inside the "<ui>" tag. For example:

```
<ui>
    <menu>
        <option> l10n :: first </ option>
        <option> Option One </ option>
        <option> Option Two </ option>
    <onMenuSelected> selected </ onMenuSelected>
    </ menu>
    ...
</ ui>
```

Each "<option>" tag sets the text that appears in the menu. The "onMenuSelected" event handling will be invoked whenever the user tap a menu item. In this example, the file "selected.lua" will be invoked (could also be immediate command: "lua::").

Note that the menu items can be localized too.

There can be only one menu per screen, and it has to be statically defined in the screen file.

To display the menu use the command "aamo.showMenu()". If you are using an Android device, the hardware "menu" button will trigger the menu. However, for compatibility, it is recommended that you created a button on all screens that use menu:

```
<element>
    <type>3</type>
    <id>NNN</id>
    <percentTop>YYYYYY</percentTop>
    <percentLeft>XXXXXX</percentLeft>
    <percentHeight>20</percentHeight>
    <percentWidth>30</percentWidth>
    <text>l10n::menuLabel</text>
    <onClickScript>lua::aamo.showMenu()</onClickScript>
</element>
```

Events



When the user clicks a menu item, the "onMenuSelected" handling is invoked. AAMO creates two global parameters (see the definition of global parameters ahead):

- "aamo::selectedMenuIndex": contains the selected index (starting from zero);
- "aamo::selectedMenuText": contains the selected text (translated, if applicable);

Functions

- aamo.showMenu(): invoked when configured to display the menu screen.

LABEL

A label control is "read only", or just to display information to the user. Its type is 2 and can be defined as follows:

```
<element>
  <id> 2 </ id>
  <type> 2 </ type>
  <percentTop> 10 </ percentTop>
  <percentLeft> 0 </ percentLeft>
  <percentWidth> 30 </ percentWidth>
  <percentHeight> 10 </ percentHeight>
  <text> l10n::test </ text>
</ element>
```

Note that the "text" property is translated, making the AAMO search the translation of the key "test" in the translation file.

The properties "percentXXX" were explained in the introduction to this manual.

Events

The LABEL control does not respond to events.

Functions

- text = aamo.getLabelText (<label id>): returns the contents of the LABEL;
- aamo.setLabelText (<label id>, "<label text>") changes the contents of the LABEL;

TEXTBOX



A box for entering text. Can be defined as:

```
<element>
  <id> 1 </ id>
  <type> 1 </ type>
  <percentTop> 10 </ percentTop>
  <percentLeft> 40 </ percentLeft>
  <percentWidth> 40 </ percentWidth>
  <percentHeight> 10 </ percentHeight>
</ element>
```

The TEXTBOX is the control of type 1. We define its content with the tag "text", which can be located.

Events

The TEXTBOX control does not respond to events.

Functions

- text = aamo.getTextField (<id of TEXTBOX>): get contents of a TEXTBOX;
- aamo.setTextField (<id of TEXTBOX>, "<Text>") changes the contents of a TEXTBOX;

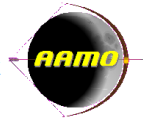
BUTTON

It's a button that can be touched by the user. Can be defined this way:

```
<element>
  <id> 4 </ id>
  <type> 3 </ type>
  <percentTop> 40 </ percentTop>
  <percentLeft> 0 </ percentLeft>
  <percentWidth> 20 </ percentWidth>
  <percentHeight> 10 </ percentHeight>
  <text> Menu </ text>
  <onClickScript> lua::aamo.showMenu () </ onClickScript>
</ element>
```

Its type number is 3.

Events



- "OnClickScript" invoked when the user touches the button;

Functions

- A BUTTON has no manipulation functions.

CHECKBOX

A checkbox is a box of binary choice (on / off). Its type is 4 and can be defined as follows:

```
<element>
  <id> 3 </ id>
  <type> 4 </ type>
  <percentTop> 25 </ percentTop>
  <percentLeft> 5 </ percentLeft>
  <percentWidth> 10 </ percentWidth>
  <percentHeight> 10 </ percentHeight>
  <checked> 1 </ checked>
  <onChangeScript> changed </ onChangeScript>
</ element>
```

The property "checked" is a binary value: 0 (zero) or 1 (one), meaning, respectively, cleared and marked.

Events

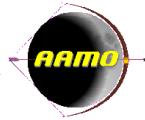
- "OnChangeScript" invoked when the status of the CheckBox changes;

Functions

- `state = aamo.getCheckBox (<CHECKBOX id>):` returns 0 (zero) or 1 (one), depending on the state of the checkbox (checked or unchecked, respectively);
- `aamo.setCheckBox (<CHECKBOX id>, <value>):` select or clear the checkbox, depending on the value passed (1 or 0, respectively);

LISTBOX

A list of elements that allows the user to select one of them. It is like a menu, but more flexible, as it can be changed dynamically (and you may have more than one Listbox per screen). Its type is 5, and can be defined as follows:



```
<element>
  <id> 1 </ id>
  <type> 5 </ type>
  <percentTop> 0 </ percentTop>
  <percentLeft> 0 </ percentLeft>
  <percentWidth> 100 </ percentWidth>
  <percentHeight> 40 </ percentHeight>
  <onElementSelected> processing </ onElementSelected>
</ element>
```

When the user selects an item from the list, "onElementSelected" event handling is invoked two global parameters and are created (see the definition of global parameter below):

- "aamo::selectedIndex": contains the selected index (starting from zero);
- "aamo::SelectedText": contains the selected text;

Events

- "OnElementSelected": occurs when the user touches an item in the list. You should check the two global parameters to determine which element was selected;

Functions

- aamo.addListBoxOption (<LISTBOX id>, "<option text>"): adds the specified text to the listbox;
- aamo.clearListBox (<LISTBOX id>): deletes all elements of the listbox;

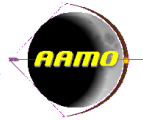
WEBBOX

A mini web browser built into the screen. Its type is 6 and can be defined like this:

```
<element>
  <id> 1 </ id>
  <type> 6 </ type>
  <percentTop> 0 </ percentTop>
  <percentLeft> 0 </ percentLeft>
  <percentWidth> 100 </ percentWidth>
  <percentHeight> 40 </ percentHeight>
  <url> http://www.google.com </ url>
</ element>
```

The parameter "url" indicates that the initial URL to be displayed. Note that it can also be a translation key (l10n::):

```
<url>l10n::urlkey</ url>
```



Events

- Currently WebBox does not respond to events.

Functions

- `aamo.navigateTo (<WEBBOX id>, "<url text>")`: makes the WebBox navigate to the specified URL. The url parameter can also be translated, for example:
`aamo.navigateTo (1, "l10n::Murl");`

IMAGEBOX

A box to display images. Its type is 7 and can be define like this:

```
<element>
  <id> 4 </ id>
  <type> 7 </ type>
  <percentTop> 75 </ percentTop>
  <percentLeft> 0 </ percentLeft>
  <percentWidth> 100 </ percentWidth>
  <percentHeight> 30 </ percentHeight>
  <picture>l10n::image</ picture>
  <stretch> 1 </ stretch>
  <onClickScript> lua::aamo.showMessage ("Clicked Image") </
onClickScript>
</ element>
```

ImageBox has two specific properties:

- `picture`: the image file name or translation keyword (can be localized);
- `stretch`: if 1 (one) the image is stretched to cover IMAGEBOX. If 0 (zero) the image appears in its normal proportion;

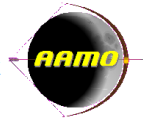
Events

- "OnClickScript": if you touch the image, this treatment is invoked;

Functions

- `aamo.setPicture (<IMAGEBOX id>, "<filename>")`: changes the image being displayed by IMAGEBOX.





Application Programming Interface

AAMO provides several functions for use within the Lua source code.

As already mentioned, the event handling source code can be immediate, within the event-handling tags (prefix "lua::"), or inside an external file, with the "lua" extension.

If there is error in Lua commands processing (immediate or external), error messages are displayed on the console of the native project. With AAMO Editor you can run and debug the Lua code using the IDE.

All API functions begin with the prefix "aamo.". We will present the list in alphabetical order.

addListBoxOption

- `aamo.addListBoxOption (<LISTBOX id>, "<text>")`: adds the specified text to the listbox;

clearListBox

- `aamo.clearListBox (<LISTBOX id>)`: deletes all elements of the listbox;

close

- `aamo.close ("query name")`: Closes the database query;

closeDatabase

- `aamo.closeDatabase ("<database name>")`: Closes a database that was opened (or created). Can be used when the screen is paused, to save resources;

eof

- `aamo.eof state = ("query name")`: returns the Boolean value "True" if there are no more records returned by the specified query. You must provide the query's name;



execSQL

- `aamo.execSQL (<SQL text>, <SQL parameters>)`: executes the specified update query using the provided parameters. example:

```
aamo.execSQL ("insert into contact (id, name, address, email)
values (?, ?, ?, ?)"
NULL, aamo.getTextField (4),
aamo.getTextField (6),
aamo.getTextField (2))
```

exitScreen

- `aamo.exitScreen ()`: disposes the current screen from the stack;

The current screen will be disposed from the stack (and memory). Before that, the screen's "onEndScript" event handler will be invoked. The previous screen (in the stack) will be displayed, and its "onBackScript" event handler is executed.

However, if the current screen is the first screen, the command "exitScreen" will not do nothing.

getCheckBox

- `state = aamo.getCheckBox (<CHECKBOX id>)`: returns 0 (zero) or 1 (one), depending on the state of the checkbox (checked or unchecked, respectively);

getCurrentScreenId

- `id = aamo.getCurrentScreenId ()`: returns the identifier of the current screen;

getError

- `errorcode = aamo.getError ()`: returns the last AAMO error code;

AAMO Currently reports the following error codes:

- 10: Missing Parameter;
- 11: File not found;
- 12: Null value;
- 100: translation File not found;



getGlobalParameter

- `<object> aamo.getGlobalParameter ("<name>")`: returns the value of the parameter or nil if it does not exist;

Global parameters are kept in a dictionary and are available to all screens. You can create Text, Number or Boolean parameters.

getLabelText

- `text = aamo.getLabelText (<LABEL id>)`: returns the contents of the LABEL;

getLocalizedText

- `text = aamo.getLocalizedText("<key>")`: searches for the key in the translation file and returns its translated text. Do not use the prefix "l10n ::" in the key parameter;

getTextField

- `text = aamo.getTextField (<TEXTBOX id>)`: get TEXTBOX contents;

loadScreen

- `aamo.loadScreen (<SCREEN id>)`: loads a screen in the stack (and displays it);

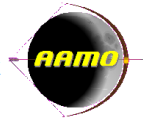
Whether the screen is already on the stack or not, a new screen will be loaded. Its "onLoadScript" event handler will be invoked. In the previous screen the "onLeaveScript" event handler will be invoked. The new screen is considered the current screen and its controls and events available to the user.

log

- `aamo.log ("<text>")` writes a message on the device's log;

navigateTo

- `aamo.navigateTo (<WEBBOX id>, "<url>")` makes the WebBox navigate to the specified url. Can also be a translation key, for example:
`aamo.navigateTo (1, "l10n::Murl");`



next

- `<Table Lua> = aamo.next("query name")`: returns next record of the named query. This command should be used only after an `"aamo.query()"` command. The result is a table of numbers and strings with the fields contents of the record (Strings only). Before using the fields check the `"aamo.eof()"` funcion;

In this version, due to deficiencies in access to the SQLite Android 2.2, the field values are returned in String format.

OpenDatabase

- `aamo.openDatabase ("<database name>")`: Opens the database whose name is in the file `"app/db/bd.xml"` (inside `<name>` tag);

In this version of AAMO only a single database can be opened. In fact, there can be only a single database at all (bd.xml file inside the "app/bd" subfolder). The database is defined according to the XML file. See more database chapter.

query

- `<Table Lua> = aamo.query("query name", "<SQL select> ", "<parameters>")`: creates a new query and returns the fields of the first record as a Lua table. Example:

```
aamo.query cursor = ("query" "select * from contact where id =?", id)
```

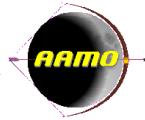
Read more about databases in the database chapter.

setCheckBox

- `aamo.setCheckBox (<CHECKBOX id>, <value>)`: select or clear the checkbox, depending on the value passed (1 or 0, respectively);

setGlobalParameter

- `aamo.setGlobalParameter ("<name>" <content>)`: add a global parameter with the given name and content, which can be: text, number or boolean. If the parameter already exists, then its contents is changed;



A global parameter can be used to store information to be shared across the application. AAMO itself makes use of global parameters to indicate elements of the list or menu selected. If there is already a parameter with the given name, its value will be replaced by new content informed.

setLabelText

- `aamo.setLabelText (<LABEL id>, "text")` changes the contents of the LABEL;

setPicture

- `aamo.setPicture (<IMAGEBOX id> "image file name")` loads the image shown on imagebox. Note that the file must be inside the "app" folder. Example:
`aamo.setPicture (1, "teste.png");`

Image file name can be a translation key. Example:

```
aamo.setPicture (5, "l10n:banner")
```

setTextField

- `aamo.setTextField (<TEXTBOX id>, "<Text>")` changes the contents of a TEXTBOX;

showMenu

- `aamo.showMenu ()`: displays the screen's menu;

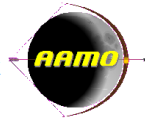
showMessage

- `aamo.showMessage ("<message text>")` displays a message dialog to the user;

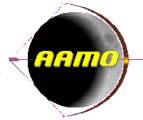
showScreen

- `aamo.showScreen (<Screen id>)`: check if there is already a screen with that number on the stack, then promote it to be the current screen. If not, load a new version of the screen;

This command disposes all the screens between the current screen and the desired screen by invoking the events:



-
- "OnEndScript" discharged for each screen;
 - "OnBackScript": only the desired display;



Databases

AAMO allows you to create and access SQLite databases in a fully transparent and platform independent way. Both platforms served by AAMO (Android and iOS) let you use SQLite databases, that's why we use it.

In this version there are some limitations:

1. Only a single database can be declared;
2. Only one database can be opened;
3. The query will only return the string values of the fields, even numeric ones;

To use a database is necessary to open it with the command: `aamo.openDatabase("<database name>")`. Let's see a simple example:

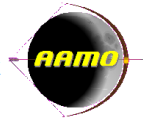
```
aamo.openDatabase("contacts")
cursor = aamo.query("query", "select * from contact")
while not aamo.eof("query") do
    for i=1,#cursor do
        aamo.log(cursor[i])
    end
    cursor = aamo.next("query")
end
aamo.close("query")
aamo.closeDatabase("contacts")
```

In this example show all the contacts table records, and all fields of each record, displaying the values in the log. Note the condition "#cursor" in "for" command, which lets you browse all the elements of a Lua table

The "aamo.openDatabase()" command creates the database if it not exists. AAMO uses the definition stored in "app/db/bd.xml" to create the database. Here is the definition file of the sample database:

```
<?xml version="1.0" encoding="UTF-8"?>
<aamo-bd>
    <version>1</ version>
    <name>contacts</ name>

    <table>
        <name>contact</ name>
        <Columns>
            <column>
                <primarykey/>
                <name>ID</ name>
                <type>integer</ type>
                <notnull/>
            </column>
            <column>
                <name>name</ name>
                <type>text</ type>
            </column>
        </Columns>
    </table>
</aamo-bd>
```



```
<column>
  <name>address</ name>
  <type>text</ type>
</column>
<column>
  <name>email</ name>
  <type>text</ type>
</column>
</columns>
</table>
</aamo-bd>
```

Database messages

There are 3 more error messages:

- 20: error opening database;
- 21: error executing query (returns nil);
- 22: error executing SQL (no query);