

# Minimalist Application Markup Language

## By Air Boigus

The Minimalist Application Markup Language, or MAML, is a very simple markup language used to make presentation-like applications. The look and structure of MAML is very similar to HTML, so anyone familiar with HTML should be able to pick up MAML very quickly.

---

This PDF explains concepts and consistencies of MAML and does **not** contain information on the tags, attributes, and parameters within MAML. Another attached PDF will explain information related to that.

### Terminology

The words used to describe parts of MAML are fairly self explanatory, though if they *do* seem a little complicated or if you would like to know what everything means just in case, below is a list of all the terms used within this document:

- **Tag** - A tag is a piece of text in a MAML document that is surrounded in angled brackets ('<' and '>'). For example "<window>" and "</image>" are tags.
- **Attribute** - Inside tags, there are name/value pairs that are used to customize a selection of properties. The names in these pairs are called attributes. For example "width" and "src" are attributes.
- **Parameters** - In the name/value pairs described above, the values are called parameters. For example, "true" and "#ff0000ff" are parameters.
- **Elements** - Most tags represent graphics that get drawn to a window. These graphics are called elements. For example, the "rectangle" tag draws a rectangle element to the window.
- **Object** - In MAML, an object is, well, just about anything. It's used as a general term to describe things. For example, the Unrecognized

Object Error occurs when a general piece of text can not be identified as a tag, attribute, or parameter.

## Conventions

There are a few ways of formatting a MAML document that are not required, but allow for a better, more uniform look. Below is a list of all recommended conventions:

- Spaces belong after each attribute/parameter pair, but not around the '=' character that assigns the parameter value to the attribute. (Example: "</... x="0" y="32" radius="8">")
- If a tag does not have any inner text, use only one tag and prefix it with a '/' (Example: "</image">")
- If a tag has inner text, separate the opening tag, the inner text, and the closing tag all on different lines. **Example:**

```
<paragraph>
    Hello, World!
</paragraph>
```

- When another tag or inner text is within a tag, press [tab] to provide a visible hierarchy.
- Use either a 2-spaced or a 4-spaced tab. Don't be quirky.

## Automatic Formatting

Markup languages like HTML are known for their syntactical leniency. MAML automatically formats certain things in order to make the displayed document look better and to make the internal parsing easier. Below is the set of rules that MAML follows:

1. All sets of whitespace characters get shortened to a single space.
2. Any whitespace characters before and after a piece of text gets removed
3. All characters inside tags and attributes get converted to lowercase.
4. All parameters, other than hexadecimal color values, and inner text stay as they are and not get converted to lowercase.

## Files

Inevitably, when writing a MAML document, you will need to enter a file path of some type. You could be entering the path of your MAML document into the command line, entering a path of an image source file, and so on. MAML has a set of rules for searching for these files, as shown below:

**\*\*For Unix-based/like (Linux, BSD (I think), Mac OS X)**

1. A file path starting with '/' tells the MAML parser to start looking for the file at the root directory.
2. A file path *not* starting with '/' tells the MAML parser that the file path is relative, and will look for the file relative to your MAML document.

## Colors

When setting the value of a color-related parameter, an RGB or RGBA hexadecimal code must be used. The format should follow either "#RRGGBBAA" or "#RRGGBB" and must be prefixed with '#.'

## Errors

The MAML parser was built to catch as many errors as possible. This was a challenge during development due to the many types of possible errors. Below is a list of all possible errors and when they can occur:

- **File Not Found Error** - This occurs when, well, a file can not be found.
- **File Not Supplied Error** - This error occurs in the command line when a MAML file is not supplied.
- **Too Many Files Error** - This error occurs in the command line when more than one MAML file is supplied to be parsed.
- **Internal Error** - Errors of this type occur when a mistake was made in the source code of the MAML parser program. The error message will print out the line of code at which the error occurred. Please contact me through the MAML Github page or Gmail (airboigus@gmail.com) and report the error's line number when this error occurs.

- **No Attribute Parameter Supplied Error** - This occurs when an attribute name is given, but no parameter value is given.
- **Invalid Attribute Parameter Error** - This error occurs when a parameter's formatting does not match the proper formatting of the paired attribute.
- **Unrecognized Object Error** - This occurs when a piece of text, not including inner text or a parameter, is unidentifiable. You may have just made a small typo, so check your code!
- **Children Not Allowed Error** - *Uhhhh, yeah this sounds kinda weird, huh...* MAML, in its current state, does not support a parent-child hierarchy like HTML. Therefore, when tags are placed within tags, this error occurs. Do not fret, though, for MAML has a built-in algorithm that skips across all children tags to find the matching closing tag.
- **Missing Closing Tag Err** - As the name implies, this error occurs when no closing tag has been supplied. The program *will* continue to run after the error.
- **Too Many Default Scenes Err** - This occurs when more than one scene is set as the default scene. Just remove the "default" attribute from as many scenes as needed, and there you go!

For readability and simplicity, all MAML error messages have a uniform format. Most MAML errors should look similar to "[Error Environment] | [Error Type] | [Ignore Status] | Snippet: [Error Snippet]." Internal Errors in MAML should look like "[Error Environment] | [Error Type] | [Report Information]." More detail on the formatting of MAML error messages is described below:

- **Error Environment** - All errors in MAML occur only in specific environments. For example if the error environment in your error message says "CMD ERR," then that error can only occur in the command line. Below is a list of all possible error environments
  - **ERR** - A general error that can occur in multiple places
  - **CMD ERR** - An error that occurs in the command line

- **DOC ERR** - An error that occurs in the supplied MAML document
- **INTERNAL ERR** - An error that occurs in the source code of the MAML parser program
- **Error Type** - This describes the actual type of error that occurred. All possible error types were explained previously in the documentation, so you should know by now what this entails.
- **Ignore Status** - In MAML, some errors are minor and can be ignored, while some are major and prevent the MAML parser program from running. An example of an error that can be ignored may be an Unrecognized Object Error, while an error that can *not* be ignored may be a File Not Supplied Error.
- **Error Snippet** - When an error occurs in MAML, a small snippet is given to help you debug your code or to help you find what was wrong in your command line statement.
- **Report Information** - When an Internal Error occurs in MAML, a source code line number and contact information will be given. Please contact me using the given contact information and report the line number given. Within a small amount of time, the error should hopefully be resolved and a new version of MAML should be distributed soon.

## Parameter Values

MAML has a variety of parameter types that all follow certain formatting rules. Below is a list of all parameter types:

- **Integer** - When passing an integer parameter value, it may be negative or positive, but, being an integer, it can not be fractional. (eg. "-69")
- **Boolean** - Boolean values in programming are data types that are either true or false. In MAML, a boolean parameter must be all lowercase in order to work properly. (eg. "true")
- **Raw String** - Raw strings in MAML are strings of text that do not get modified, and do not have any formatting. (eg. "Hello, World!")

- **File** - Files in MAML, must follow the format as described in the **Files** section. (eg. `"/home/airboigus/Documents/nut.maml"`)
- **List** - Lists in MAML are sets of strings separated by commas. (eg. `"bold, italic, underlined"`)
- **Color** - Colors in MAML must follow the format as described previously in the **Colors** section. (eg. `"#ff8800"`)