

GM User Documentation

Contents

Introduction	1
Design Assumptions For This Framework.....	2
Number One Rule	2
Major Framework Components.....	3
Campaign Properties.....	3
Basic	3
Hero Lab POR File.....	3
TokenLibCampaignData	3
TokenLibCharacterSheet.....	3
TokenLibCharacterSheetData	3
TokenLibConfiguration.....	3
TokenLibUtilities	3
Library Tokens.....	3
Image Tables	4
Campaign Macros	4
GM Macros.....	4
Windows Used For Macros By This Framework	4
Campaign Window	4
GM Window	4
Selection Window	4
Recommended Optional Additions To This Framework	4
MapTool Help, Add Default Tables	4
lib:frames	5
Upgrading To Newer Version Of My Framework.....	7
Upgrading Campaign Settings.....	7
Upgrading The Library Tokens	7
How The Upgrade Process Works.....	7

Introduction

The goal for this document is to help GMs understand how to use my framework. This is a document that is very much in progress. I will continue updating it as I have time.

Design Assumptions For This Framework

The design assumptions for this framework are as follows. Understanding them will make it easier for you to learn the framework.

1. Only a minimal number of rules would be automated.
2. All decisions would be made and enforced by the GM.
3. An audit log was needed because GMs and Players forget.
4. Campaign management was needed because GMs need help.
5. Many rules depend on context too complex to be automated.
6. GMs and Players may only be interested in using some features but not others.
7. When features are not desired to be used, just like on a character sheet, they are simply not filled in.
8. A detailed breakdown of all the modifiers for attacks is desired to help ensure valid game play.
9. To avoid trying to support all the feats, class abilities, spells, etc., attacks would be copied, named appropriately, and modified, on the fly if desired.
10. Players like to determine what spells they want to prepare in advance, not when the GM says their characters are resting.
11. Campaigns like to use a wide variety of material, rules, and home rules.
12. Make things intuitive to decrease learning time.
13. Flying, levitation, and other things involving elevation would be needed.
14. Hero Lab(R) is needed especially by GMs to import NPCs into this framework, otherwise it would take too long to enter the data.
15. Being able to synchronize changes from Hero Lab(R) to this framework would be needed to support Players, who might update there Hero Lab(R) files between games.
16. Upgrades for PC and NPC tokens should be fully automated, even if the tokens have not been used for a long time and are suddenly thrown onto a map and used.
17. Some chat messages should be private and others public.

Number One Rule

If you do not need it or want it, leave it blank. Blank means no spaces. Blank means empty. The macros are designed to effectively ignore it, if it is blank.

This is effectively what people do with character sheets. If they don't need it, they leave it blank.

Major Framework Components

The following items are critical parts of this framework:

- Campaign Properties
- Library Tokens
- Image Tables
- Campaign Macros
- GM Macros

Campaign Properties

These are used to setup properties for the different token types used by this framework:

- Basic
- Hero Lab POR File
- TokenLibCampaignData
- TokenLibCharacterSheet
- TokenLibCharacterSheetData
- TokenLibConfiguration
- TokenLibUtilities

Basic

This is the type to use for character tokens. Some RPG game systems call these PCs or NPCs.

Hero Lab POR File

I suspect this type is obsolete. I'll have to do some research. My tokens based on Hero Lab® POR files have the type "Basic", just like other character tokens.

TokenLibCampaignData

This is the type for the library token used to hold campaign data.

TokenLibCharacterSheet

This is the type for the library token used to hold macros that interact with a character's character sheet.

TokenLibCharacterSheetData

This is the type for the library token used to hold data global to all character sheets.

TokenLibConfiguration

This is the type for the library token used to hold configuration data for the campaign.

TokenLibUtilities

This is the type for the library token used for macros used by the GM mostly for campaign management.

Library Tokens

This framework uses the following library tokens (in the Example.cmpgn they are located on the "Library Tokens" map):

- Lib:Configuration
- Lib:CampaignData
- Lib:CharacterSheetData
- Lib:Utilities
- Lib:CharacterSheet

Of those, **only two of them are replaced** when **upgrading** to a new version of my framework:

- Lib:Utilities
- Lib:CharacterSheet

The **others contain your data** and **should never be replaced** when **upgrading**!

More on upgrading in a separate section.

[Image Tables](#)

[Campaign Macros](#)

[GM Macros](#)

Windows Used For Macros By This Framework

The following windows are used for macros by this framework:

- Campaign window
- GM window
- Selection window

Campaign Window

This window has macros that will (usually) be applied to the selected tokens. A security check is made to ensure the person running the macro has permission to do so on the selected tokens.

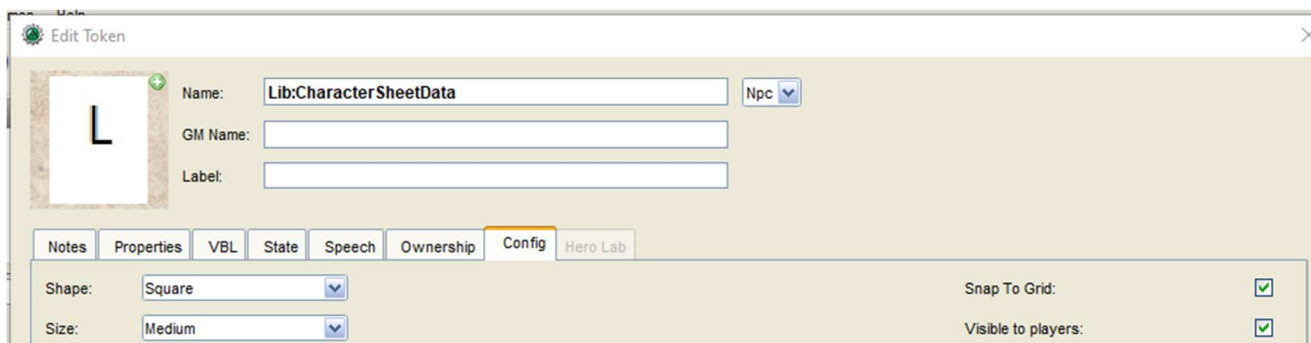
Players and GMs use them to interact with their tokens.

GM Window

This window has the macros that GMs use to run the game and their campaigns. Players cannot see these, but even if they did a security check would prevent them from running these macros.

Selection Window

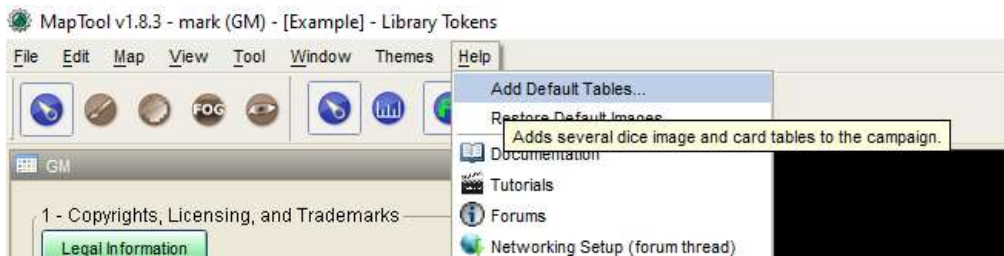
This window is used only used to edit macros for library tokens used by this framework. They should be setup on a map that is **not** visible to players. However, the tokens, themselves, must have the “Visible to players” property set:



Recommended Optional Additions To This Framework

[MapTool Help, Add Default Tables](#)

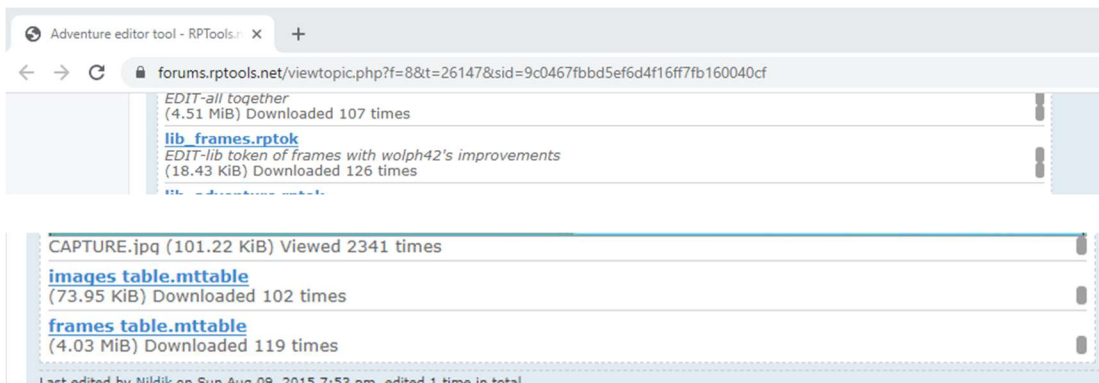
In MapTool using the Help menu, click on "Add Default Tables":



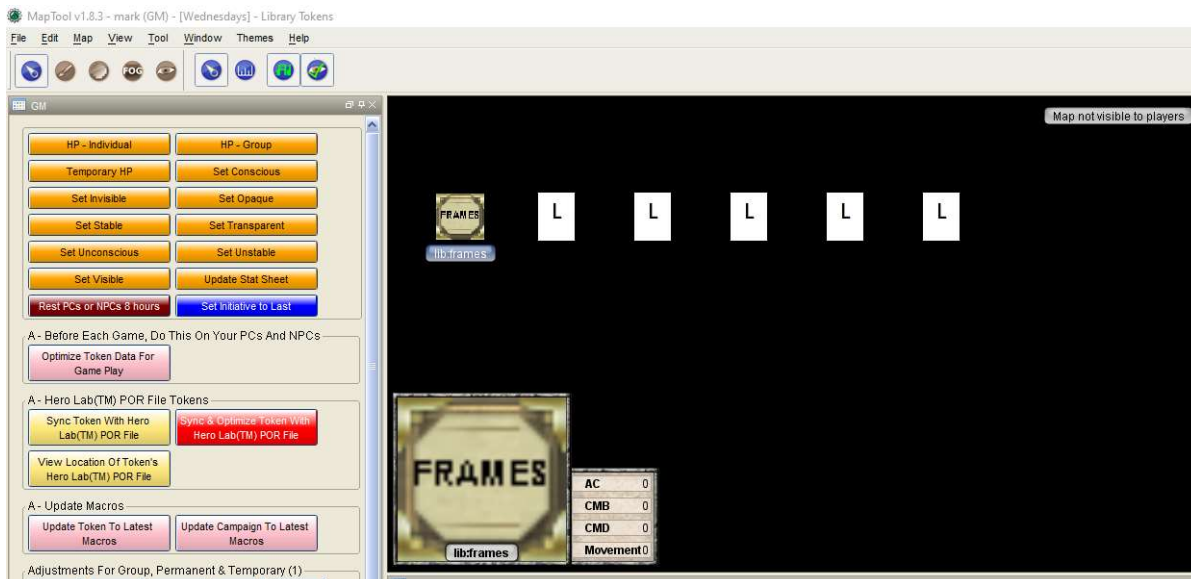
This will add image tables for all the dice, which this framework will then use automatically. If while playing, you try to use a type of die that is not in the image tables, then numbers will be used instead of dice images.

lib:frames

Download [lib_frames.rptok](#) and [frames_table.mtttable](#) [here](#).



If this is installed along with its associated image table, this framework will use it automatically -- but be sure to setup **both** or you will get an error. I recommend putting it on the same map as the library tokens for this framework. When you do it right, it will look like this:







Adding this feature will greatly improve the appearance of the forms created by this framework.

However, forms are often cached for the tokens to save time, so if you add this to your campaign after playing for a while or optimizing your tokens:

Select your tokens and use the **"Clear Cache" macro** on the bottom of the **GM window**. This will **cause the forms used by the tokens to be rebuilt using lib:frames**.

An example of a form used with lib:frames:

An example die roll with the dice images:

Melee Attack - Longspear (20/x3)		
Attack	Type	Damage
$d20 + 2 \text{ (BAB)} + 0 \text{ (Str)} + 0 \text{ (Size Modifier)} + 0 \text{ (To Hit Modifier)} + 0 \text{ (Adjustment)} =$  $+ 2 = 16$	Longspear	$1d8 + 0 \text{ (Str)} + 0 \text{ (Damage Modifier)} + 0 \text{ (Adjustment)} =$  $+ 0 = 5$

Upgrading To Newer Version Of My Framework

When upgrading to a newer version of my framework, you'll need to save the following from the new versions

Example.cmpgn file:

- Campaign Settings
- Lib:Utilities
- Lib:CharacterSheet

Upgrading Campaign Settings

Normally, what you would do is Export the Campaign Settings from the new Example.cmpgn file. Then, Import those Campaign Settings into your current *.cmpgn file (that you're using for your campaign).

It's all really cool how this works because your data is preserved. Effectively, you'll just get the new properties that were defined by the newest Campaign Settings file.

There's an awkward problem, though. When saving Campaign Settings, that also saves the Tables (see the Tables window). When you Import the new Campaign Settings file, that will replace all the Tables you have.

Therefore, you want to use the following steps to upgrade to the latest Campaign Settings:

1. Backup your latest *.cmpgn file – in case something goes wrong.
2. From your latest *.cmpgn file (for your campaign), export all of the tables in the Tables window – and save them somewhere.
3. From the latest Example.cmpgn file, export Campaign Settings to a file.
4. Into your latest *.cmpgn file, import the latest Campaign Settings from the file.
5. Go to the Tables window and delete all the tables – then import all the tables you exported earlier (this is especially important for the Handouts table!).

Upgrading The Library Tokens

1. From the latest Example.cmpgn file, save the tokens Lib:Utilities and Lib:CharacterSheet (they are on the Library Tokens map).
2. Replace the old Lib:Utilities and Lib:CharacterSheet tokens with the new ones.
3. Don't touch the other tokens (including library tokens).

How The Upgrade Process Works

When you execute macros on your character tokens from the Campaign window or from the GM window, those macros will do a check to see if any upgrades to your tokens are needed. If so, an upgrade will be performed automatically for those tokens.