GM User Documentation

Last Updated: 11/13/2021

Contents

Licensing

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 For Using This Framework

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.

Why Use MapTool

MapTool has several advantages over other virtual tabletops:

- MapTool is maintained by a very friendly and helpful development team. I can't say enough about this!

- MapTool has so far been very stable and backward compatible. I've used it heavily since version 1.4.0.5 through 1.7.0, and I've just starting using 1.8.3. Somewhere in there, I started referring to MapTool as a professional grade virtual tabletop (VTT).

- MapTool is developed mostly in Java – not JavaScript! For the 1 ½ years I was using it very actively for my campaigns, unlike other virtual tabletops, nearly all the problems I had with it were my own problems – not a bug in MapTool!

- MapTool has a Hero Lab® import feature that, for me as a GM combined with the macros in this framework, has been just an unbeatable experience.

- MapTool 1.8.3 has a really fast script engine that further improves the gaming experience.

- MapTool has a vision and lighting system based on Lumens that works great not only for light sources – but also darkness such as darkness spells.

- MapTool has a great macro API for creating macros and the Campaign Properties feature was brilliantly designed to make upgrading to new macro versions of this framework easy!

- MapTool has a user interface (UI) where you can rip out frames from the UI and reassemble them on the same or another monitor as a separate window or windows!

Advanced Topic: lib:elevation Will Make You Fly (Strongly Recommended)

To understand this feature, you'll have to read about library tokens. You might want to use MapTool for a while before trying this.

My feeling for a long time was that a major feature needed for MapTool was a revamped or replaced VBL/MBL (vision blocking layer/movement blocking layer) that supports the **concept of elevation** – in other words, flying, levitation, and outdoor terrain features such as trees, forests, valleys, hills, buildings, cliffs, under water features, etc.

Well, guess what. Thanks to **melek#4527** (on Discord), we now have a **lib:elevation** library token. What it does is exactly what we need and is quite powerful. It allows you to create elevation layers. It does this by copying the VBL/MBL from the existing layer, if you want, into a newly created layer.

I start by creating all the VBL/MBL I need for my ground level. Typically, I have a "ground level" layer and, then, I might create a "flying" layer. First, I create the "ground level" layer using what is now just a convenient button click and dialog that pops up with options, which just copies all the VBL/MBL I just created.

Then, when I create a "flying" layer, I follow the same process. You select the elevation layer you want to be on with simple mouse clicks from your list of elevation layers. For that layer, you can change the VBL/MBL all you want – it will not affect the other layers.

This feature is very powerful. Normally, I use this sort of thing on outdoor terrain. Various objects block a token's view at ground level – but if someone flies high enough they can see and move past those objects and, thus, are not inhibited by VBL or MBL.

You could also use this feature to change the view a token has as he/she approaches the edge of a cliff. While the token stands back far away from the edge of the cliff, he/she can't see what's at the base of the cliff due to the angle. As the token approaches the edge of the cliff, more and more of the base of the cliff is revealed until the token can see straight down. You can implement this by setting up multiple elevation layers. Really, an awesome feature!

melek#4527 also has another library token called **lib:simpledoors**, which enables you to put doors on your maps that open and close. I strongly recommend this, too.

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

Image tables are used by this framework for handouts, dice rolls, and to beautify forms.

Campaign Macros

Campaign macros are used on the Campaign window for macros used with the selected tokens.

GM Macros

GM macros are used on the GM window for macros the GM uses to manage the campaign.

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.

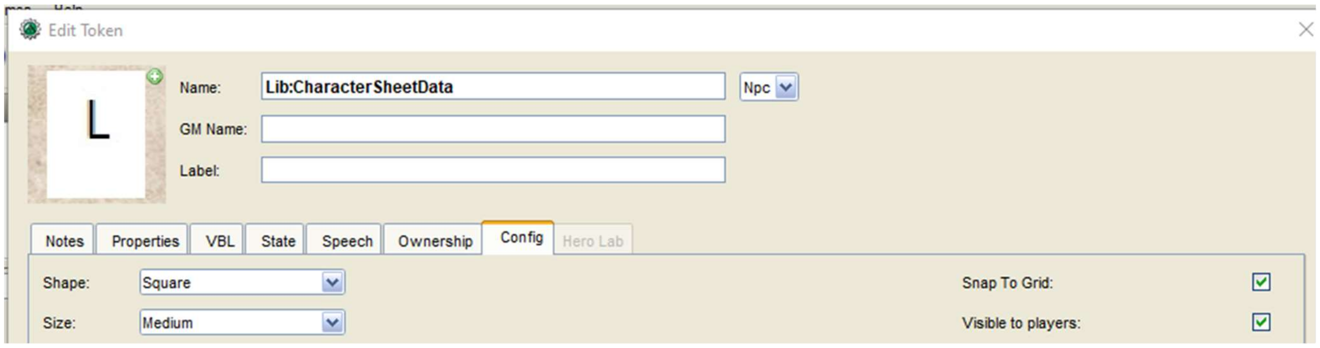
Players and GMs use them to interact with their tokens.

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:

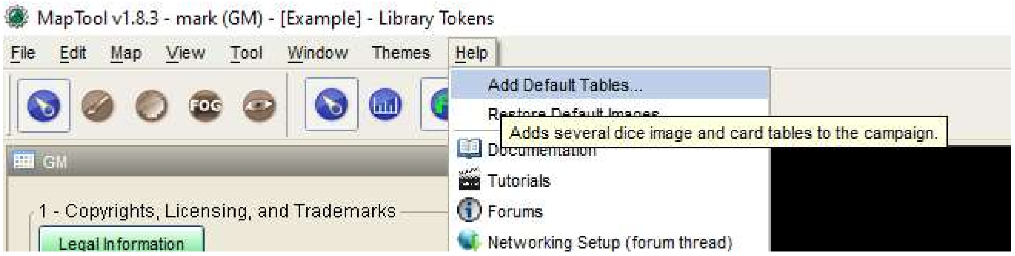


Name: Lib:CharacterSheetData Npc
GM Name:
Label:
Shape: Square
Size: Medium
Snap To Grid: ✓
Visible to players: ✓

Recommended Optional Additions To This Framework

MapTool Help, Add Default Tables

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

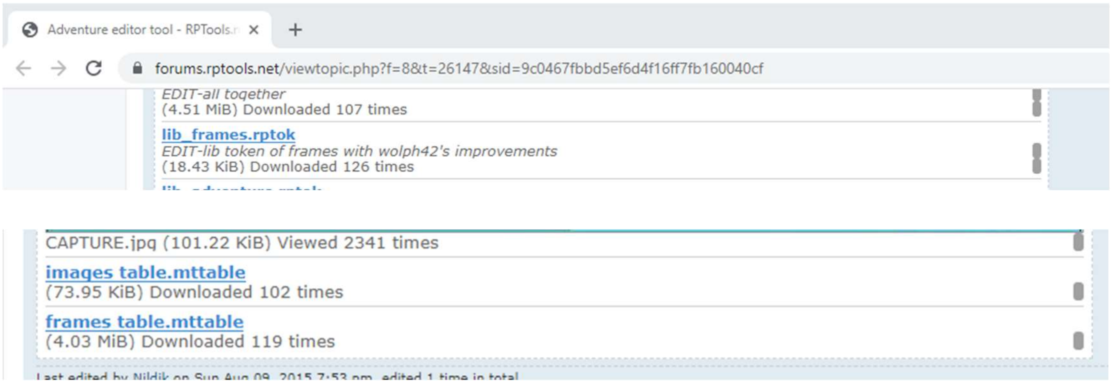

MapTool v1.8.3 - mark (GM) - [Example] - Library Tokens
File
Edit
Map
View
Tool
Window
Themes
Help
Add Default Tables...
Restore Default Images
Documentation
Tutorials
Forums
Networking Setup (forum thread)
GM
1 - Copyrights, Licensing, and Trademarks
Legal Information

Adds several dice image and card tables to the campaign.

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

Down load **lib_frames.rptok** and **frames_table.mttable** here.


Adventure editor tool - RPTools.r
forums.rptools.net/viewtopic.php?f=8&t=26147&sid=9c0467fbbd5ef6d4f16ff7fb160040cf

EDIT-all together
(4.51 MiB) Downloaded 107 times

lib_frames.rptok
EDIT-lib token of frames with wolph42's improvements
(18.43 KiB) Downloaded 126 times

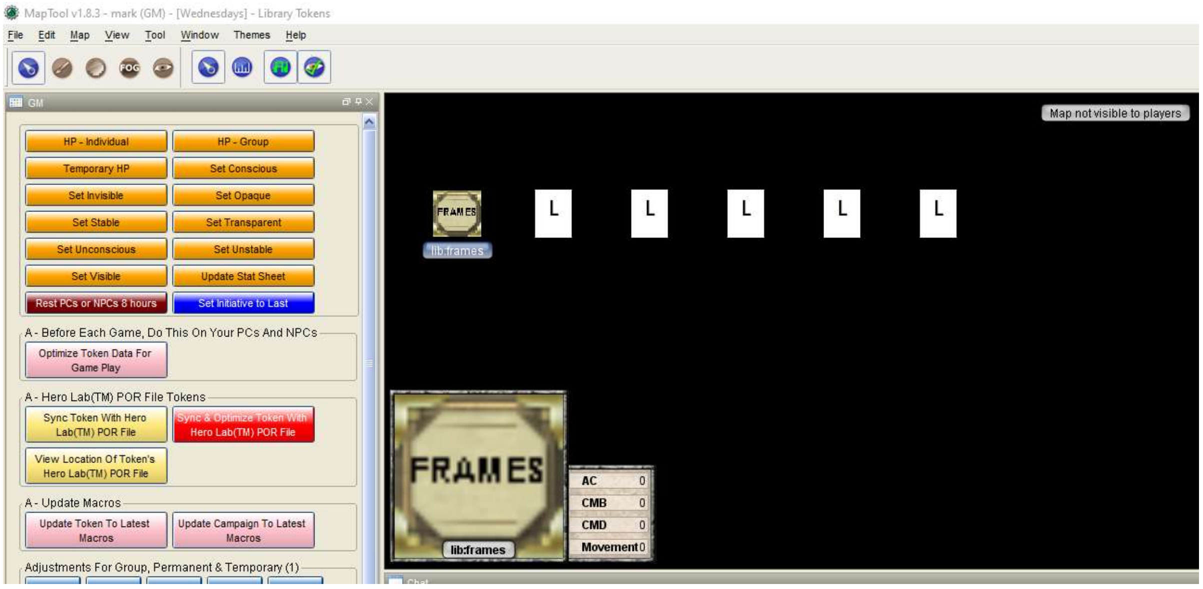
CAPTURE.jpg (101.22 KiB) Viewed 2341 times

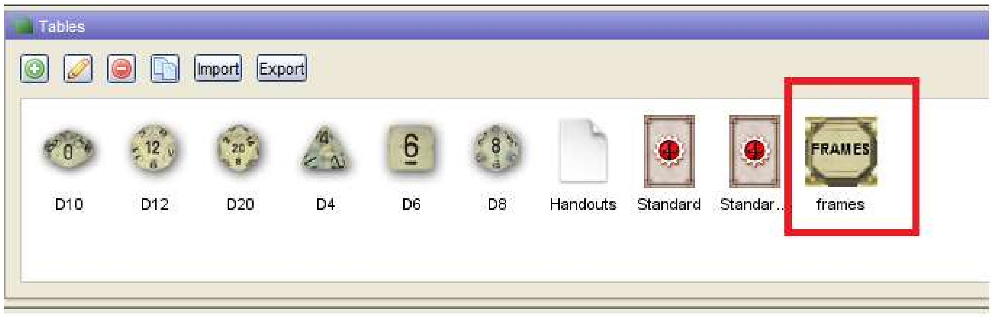
images table.mttable
(73.95 KiB) Downloaded 102 times

frames table.mttable
(4.03 MiB) Downloaded 119 times

Last edited by Nildik on Sun Aug 09, 2015 7:53 pm, edited 1 time in total.

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:


MapTool v1.8.3 - mark (GM) - [Wednesdays] - Library Tokens
File
Edit
Map
View
Tool
Window
Themes
Help
GM
HP - Individual
HP - Group
Temporary HP
Set Conscious
Set Invisible
Set Opaque
Set Stable
Set Transparent
Set Unconscious
Set Unstable
Set Visible
Update Stat Sheet
Rest PCs or NPCs 8 hours
Set Initiative to Last
A - Before Each Game, Do This On Your PCs And NPCs
Optimize Token Data For Game Play
A - Hero Lab(TM) POR File Tokens
Sync Token With Hero Lab(TM) POR File
Sync & Optimize Token With Hero Lab(TM) POR File
View Location Of Token's Hero Lab(TM) POR File
A - Update Macros
Update Token To Latest Macros
Update Campaign To Latest Macros
Adjustments For Group, Permanent & Temporary (1)
Map not visible to players
FRAMES
lib:frames
L
L
L
L
L
FRAMES
lib:frames
AC 0
CMB 0
CMD 0
Movement 0

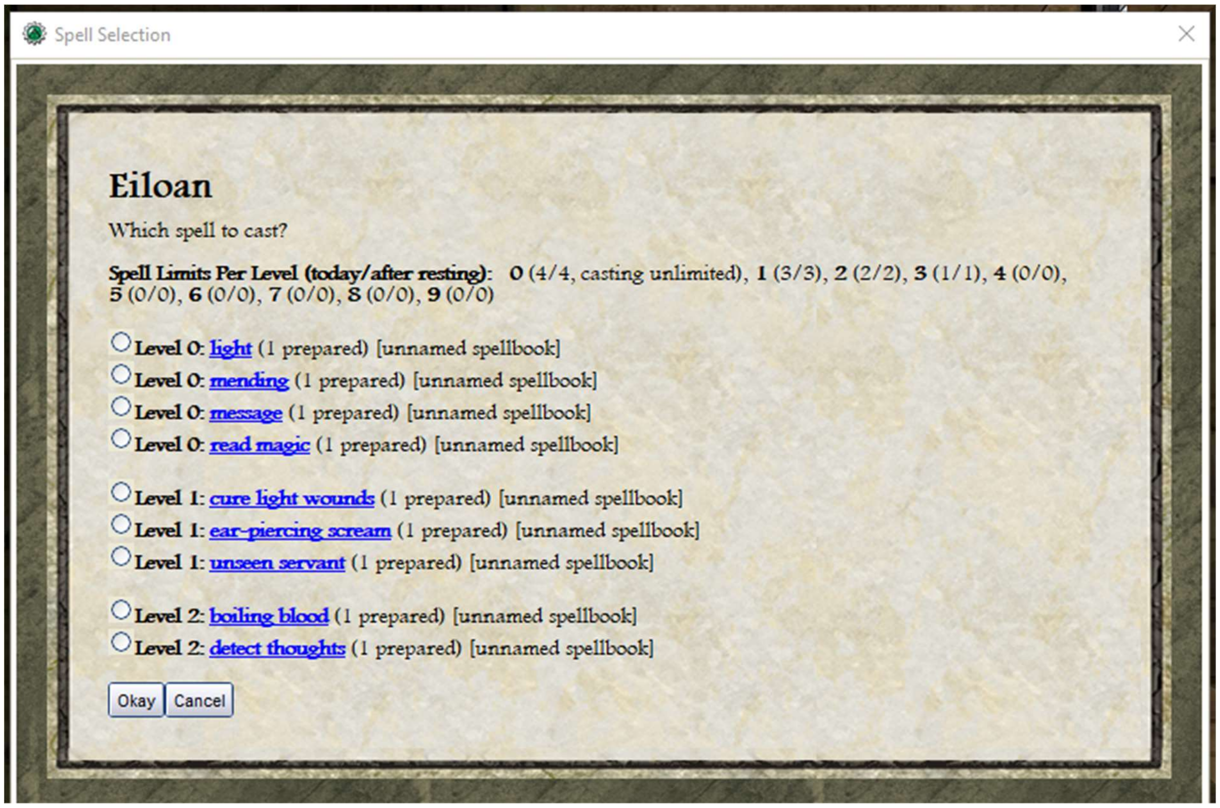


Adding this feature will greatly improve the appearance of the forms created by this framework.

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:



Spell Selection

Eiloan

Which spell to cast?

Spell Limits Per Level (today/after resting): **0** (4/4, casting unlimited), **1** (3/3), **2** (2/2), **3** (1/1), **4** (0/0), **5** (0/0), **6** (0/0), **7** (0/0), **8** (0/0), **9** (0/0)

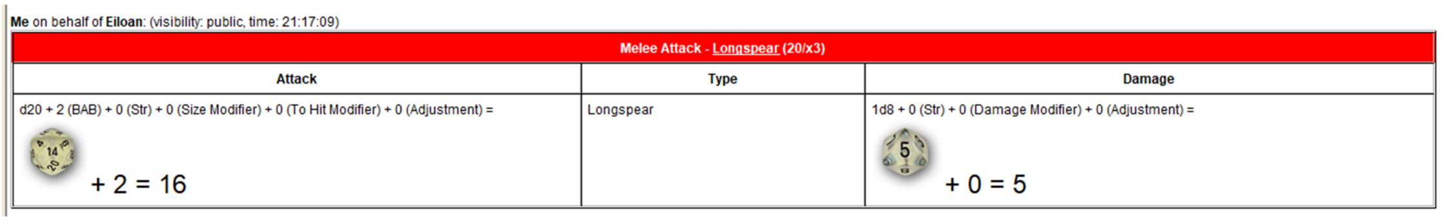
○ **Level 0:** light (1 prepared) [unnamed spellbook]
○ **Level 0:** mending (1 prepared) [unnamed spellbook]
○ **Level 0:** message (1 prepared) [unnamed spellbook]
○ **Level 0:** read magic (1 prepared) [unnamed spellbook]

○ **Level 1:** cure light wounds (1 prepared) [unnamed spellbook]
○ **Level 1:** ear-piercing scream (1 prepared) [unnamed spellbook]
○ **Level 1:** unseen servant (1 prepared) [unnamed spellbook]

○ **Level 2:** boiling blood (1 prepared) [unnamed spellbook]
○ **Level 2:** detect thoughts (1 prepared) [unnamed spellbook]

Okay Cancel

An example die roll with the dice images:



Me on behalf of **Eiloan**: (visibility: public, time: 21:17:09)

Melee Attack - Longspear (20/x3)

Attack	Type	Damage
d20 + 2 (BAB) + 0 (Str) + 0 (Size Modifier) + 0 (To Hit Modifier) + 0 (Adjustment) = 14 + 2 = 16	Longspear	1d8 + 0 (Str) + 0 (Damage Modifier) + 0 (Adjustment) = 5 + 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).

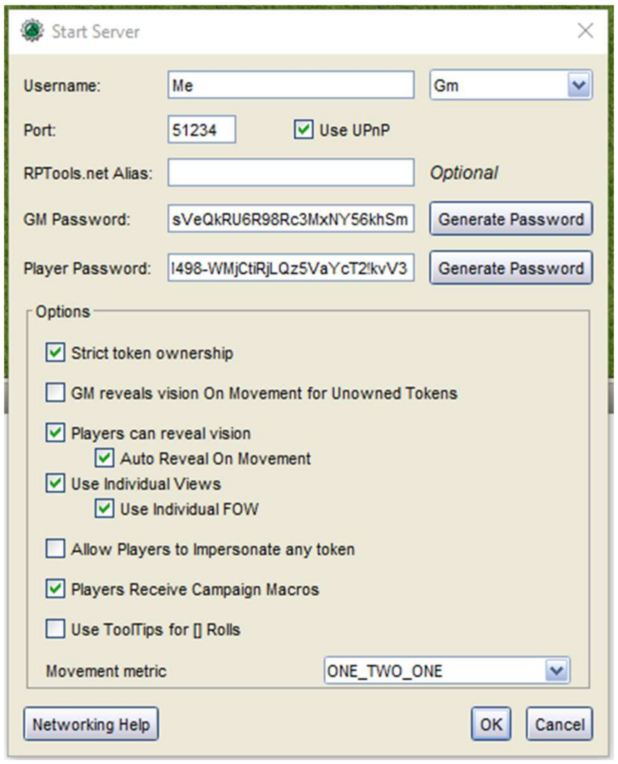
How The Upgrade Process Works

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.

Settings To Use To Start The MapTool Server

These are settings I recommend – and recommend using better passwords that people can type easily.

The "Players Receive Campaign Macros" is required for this framework:


Start Server

Username: Me   Gm

Port: 51234   [✓] Use UPnP

RPTools.net Alias: Optional

GM Password: sVeQkRU6R98Rc3MxNY56khSm   Generate Password

Player Password: l498-WMjCtiRjLQz5VaYcT2!kvV3   Generate Password

Options

- [✓] Strict token ownership
- [ ] GM reveals vision On Movement for Unowned Tokens
- [✓] Players can reveal vision
  - [✓] Auto Reveal On Movement
- [✓] Use Individual Views
  - [✓] Use Individual FOW
- [ ] Allow Players to Impersonate any token
- [✓] Players Receive Campaign Macros
- [ ] Use ToolTips for [] Rolls

Movement metric: ONE_TWO_ONE

Networking Help   OK   Cancel

Hero Lab® Import And Synchronization

This framework enables GMs to import characters from Hero Lab® into this framework – a big time saver!

After the initial import, these characters can be changed in Hero Lab® and re-synchronized with this framework. Basically, what the framework does is remember what the Hero Lab® file looked like last time, preserve the changes you've made to your character within this framework in MapTool, and import changes made to the Hero Lab® file. This feature isn't perfect, but it goes a long way to managing PCs (player characters) for players that like to use Hero Lab® -- or you, the GM, if you like to maintain your NPCs (non-player characters) there, too.

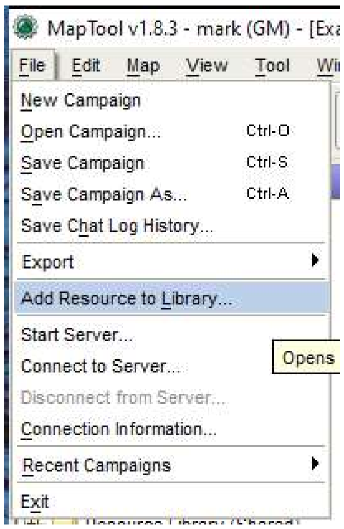
How To Do The Initial Hero Lab® Import

The process goes like this:

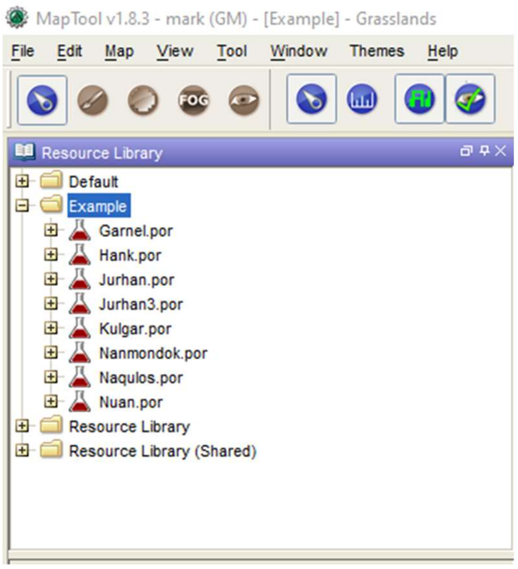
1. Use the "Add Resource to Library" feature of MapTool to add Hero Lab® POR files to your Resource Library.
2. From the Hero Lab® POR file, drag a token onto a map.
3. Use the GM macro "Sync Token With Hero Lab® POR File" to import the data into this framework.

Add Resource to Library

Use this feature of MapTool:

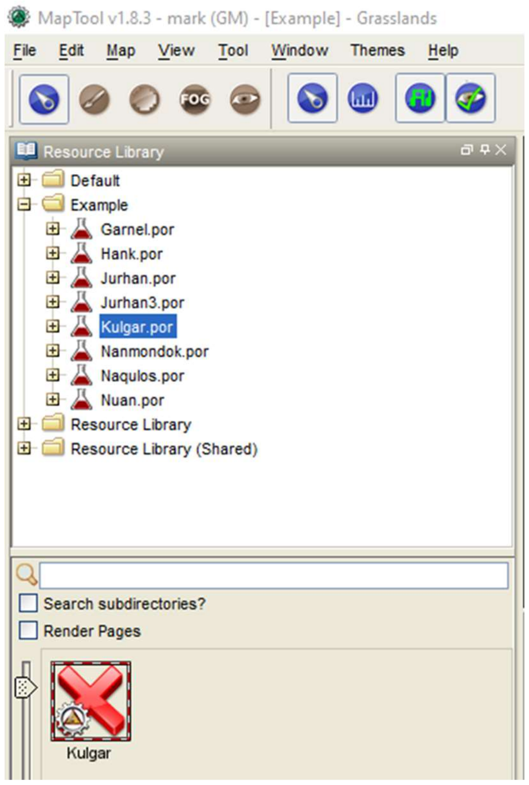


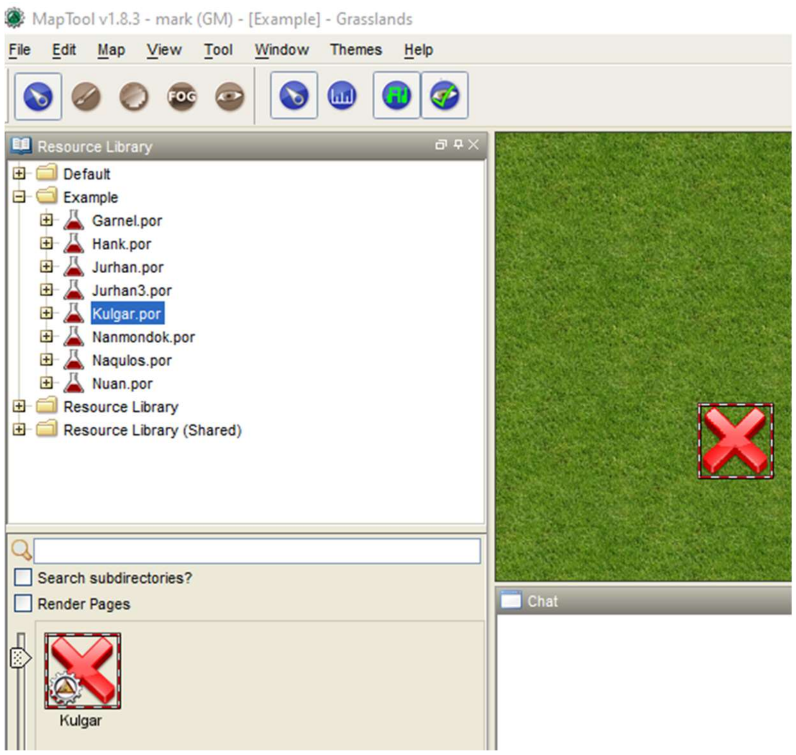
Once successful, your Resource Library will look something like this:



Drag Hero Lab® POR File Token To Map

Using the bottom part of the Resource Library window, drag a token to one of your maps (note: Kulgar does not have an image but if he did, you would see his token image below instead of an “X”):



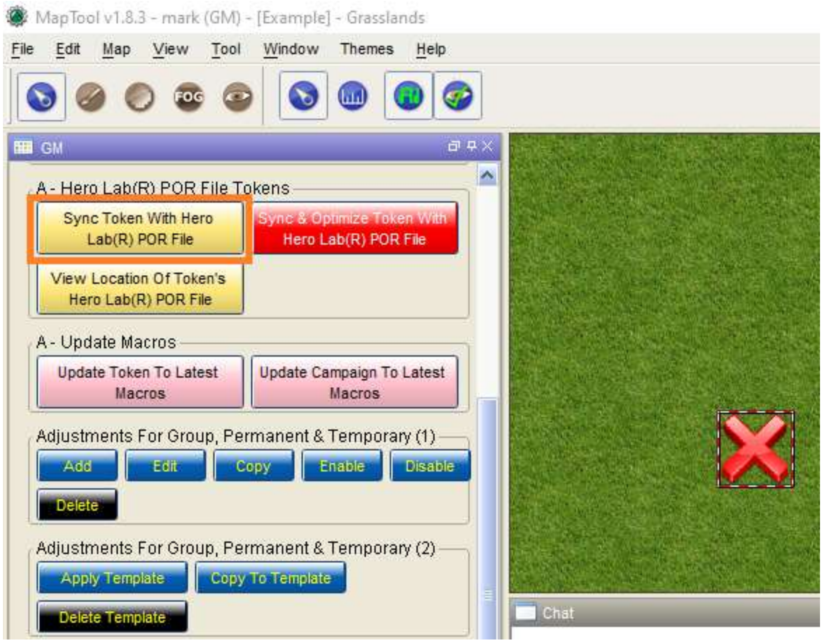


Run GM Macro “Sync Token With Hero Lab® POR File”

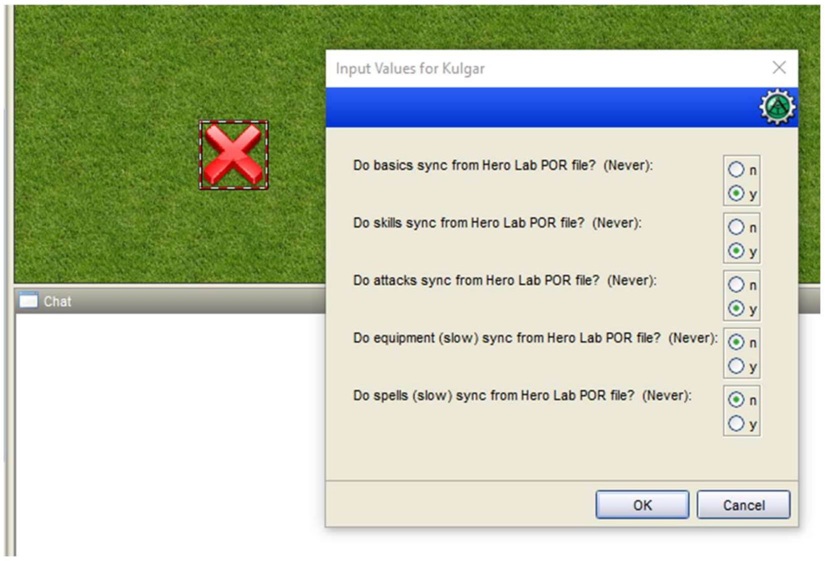
The GM macro **“Sync Token With Hero Lab® POR File”** is the fastest way to import a huge number of tokens all at once if you just want to get it done. I’m not clear exactly how long it takes per token because I no longer do it this way (see ***Consider Doing It All In One Step With “Sync & Optimize Token With Hero Lab® POR File”***).

However, tokens imported this way will not be optimized. Their macros initially will tend to run a little slower. You can drag them onto your maps and use the GM macro **“Optimize Token Data For Game Play”** to optimize only those tokens you plan to use – but more on that later below.

With the token selected, you can run the GM macro **"Sync Token With Hero Lab® POR File"**:


MapTool v1.8.3 - mark (GM) - [Example] - Grasslands
File Edit Map View Tool Window Themes Help
GM
A - Hero Lab(R) POR File Tokens
Sync Token With Hero Lab(R) POR File
Sync & Optimize Token With Hero Lab(R) POR File
View Location Of Token's Hero Lab(R) POR File
A - Update Macros
Update Token To Latest Macros
Update Campaign To Latest Macros
Adjustments For Group, Permanent & Temporary (1)
Add
Edit
Copy
Enable
Disable
Delete
Adjustments For Group, Permanent & Temporary (2)
Apply Template
Copy To Template
Delete Template
Chat

You will, then, see the following dialog:


Input Values for Kulgar

Do basics sync from Hero Lab POR file? (Never): n [✓ y]

Do skills sync from Hero Lab POR file? (Never): n [✓ y]

Do attacks sync from Hero Lab POR file? (Never): n [✓ y]

Do equipment (slow) sync from Hero Lab POR file? (Never): [✓ n] y

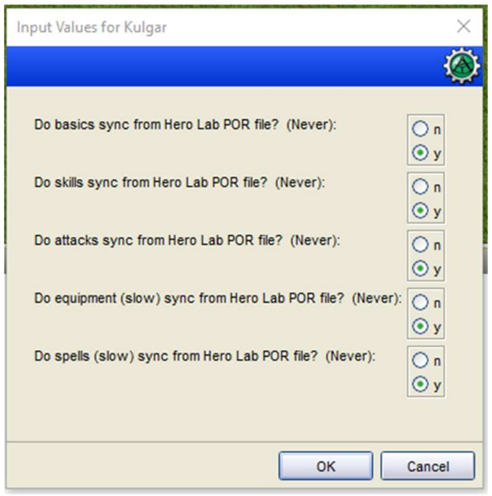
Do spells (slow) sync from Hero Lab POR file? (Never): [✓ n] y

If you are a GM importing NPCs on-the-fly for an encounter, you might just want to import the first three items: basics, skills, and attacks – because importing equipment and spells takes a bit longer but with the performance improvements of MapTool 1.8.3 and higher even these import fairly quickly.

The "(Never)" means you've never imported that item before for this token.

Most of the time, importing an NPC takes a minute or so. This character is a bit complex – it's a PC I used to play in a game and he's 8th level – so doing the whole import for him takes a while.

So, for this example, I choose the following:


Input Values for Kulgar

Do basics sync from Hero Lab POR file? (Never): n [✓ y]

Do skills sync from Hero Lab POR file? (Never): n [✓ y]

Do attacks sync from Hero Lab POR file? (Never): n [✓ y]

Do equipment (slow) sync from Hero Lab POR file? (Never): n [✓ y]

Do spells (slow) sync from Hero Lab POR file? (Never): n [✓ y]

And the Chat window shows:

Me on behalf of **Kulgar**: (visibility: GM, time: 16:02:49)
Starting *synchronizing token 'Kulgar' at 2021-03-11 16:02:49 with Hero Lab POR file:*

E:\RPG\Example\Kulgar.por

Me on behalf of **Kulgar**: (visibility: GM and self, time: 16:02:51)
Updating token to latest macros ...

Me on behalf of **Kulgar**: (visibility: GM and self, time: 16:02:51)
Updating token to macros supporting base64.encode() to protect data from potential user input errors ...

Me on behalf of **Kulgar**: (visibility: GM and self, time: 16:02:52)
Updated token macro version. Due to the use of base64.encode(), the minimum recommended MapTool version for this token is MapTool 1.5.2.

Me on behalf of **Kulgar**: (visibility: GM and self, time: 16:02:52)
Updated token buffs & conditions to now be controlled through Adjustments ...

Me on behalf of **Kulgar**: (visibility: GM and self, time: 16:02:58)
Updated token 'Kulgar' to latest macro version.

Me on behalf of **Kulgar**: (visibility: GM, time: 16:03:05)
Imported From Hero Lab: Basics *for token 'Kulgar' successfully imported at 2021-03-11 16:03:05 from Hero Lab POR file:*

E:\RPG\Example\Kulgar.por

Me on behalf of **Kulgar**: (visibility: GM, time: 16:03:19)
Imported From Hero Lab: Skills *for token 'Kulgar' successfully imported at 2021-03-11 16:03:19 from Hero Lab POR file:*

E:\RPG\Example\Kulgar.por

Me on behalf of **Kulgar**: (visibility: GM, time: 16:03:20)
Imported From Hero Lab: Attacks *for token 'Kulgar' successfully imported at 2021-03-11 16:03:20 from Hero Lab POR file:*

E:\RPG\Example\Kulgar.por

Me on behalf of **Kulgar**: (visibility: GM, time: 16:03:36)
Imported From Hero Lab: Equipment *for token 'Kulgar' successfully imported at 2021-03-11 16:03:36 from Hero Lab POR file:*

E:\RPG\Example\Kulgar.por

Me on behalf of **Kulgar**: (visibility: GM and self, time: 16:04:21)

Spell Status: Kulgar

Level	Topic	Status	Comment or Issue	Maximum Limit or Suggested Action
0	Today's Spells	**Problem**	Spells are using 6 slots	Your limit is 5!
0	Spells After Resting	**Problem**	After resting, your spells would be using 6 slots	Your limit is 5!

Me on behalf of **Kulgar**: (visibility: GM, time: 16:04:21)
Imported From Hero Lab: Spells *for token 'Kulgar' successfully imported at 2021-03-11 16:04:21 from Hero Lab POR file:*

E:\RPG\Example\Kulgar.por

Me on behalf of **Kulgar**: (visibility: GM and self, time: 16:04:22)

Spell Status: Kulgar

Level	Topic	Status	Comment or Issue	Maximum Limit or Suggested Action
0	Today's Spells	**Problem**	Spells are using 6 slots	Your limit is 5!
0	Spells After Resting	**Problem**	After resting, your spells would be using 6 slots	Your limit is 5!

Me on behalf of **Kulgar**: (visibility: GM, time: 16:04:22)
Imported From Hero Lab: Spell Limits *for token 'Kulgar' checked at 2021-03-11 16:04:22 from Hero Lab POR file:*

E:\RPG\Example\Kulgar.por

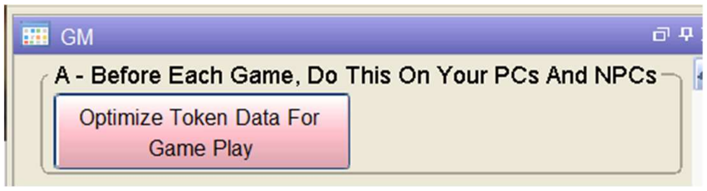
Me on behalf of **Kulgar**: (visibility: GM, time: 16:04:23)
Done *synchronizing token 'Kulgar' at 2021-03-11 16:04:23 with Hero Lab POR file:*

E:\RPG\Example\Kulgar.por

That's it! This token is now ready to play with!

GM Macro "Optimize Token Data For Game Play"

Select your tokens on your map and click this button:


GM
A - Before Each Game, Do This On Your PCs And NPCs
Optimize Token Data For Game Play

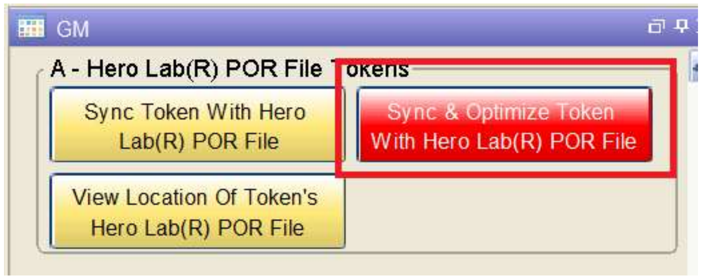
This will make the macros run faster for your tokens. I know what I said above. Normally, I do not run this before each game. You can, if you want. After the performance improvements for MTScript macros in MapTool 1.8.x, I find that even my slowest macros for tokens run with very good performance.

Consider Doing It All In One Step With "Sync & Optimize Token With Hero Lab® POR File"

Instead of using **"Sync Token With Hero Lab® POR File"**, I recommend running **"Sync & Optimize Token With Hero Lab® POR File"**, instead, because your tokens' macros will run faster on game day. What this does in reality is run the **"Sync Token With Hero Lab® POR File"** and **"Optimize Token Data For Game Play"** macros for each token. This is what I normally do.

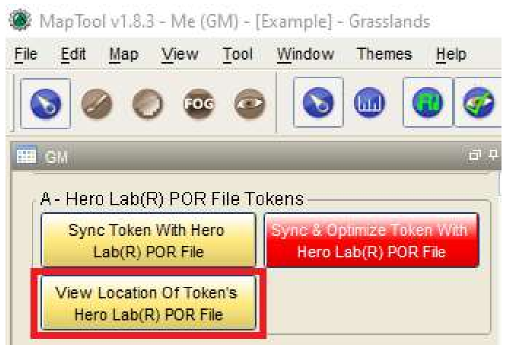
It typically takes about 1 minute per token – so I recommend selecting a group of tokens, clicking the macro below, and walking away. Look at the Chat window for status. When the whole thing is done, it will tell you how it all went and how much time it took. I don't typically have any problems, so normally you do not need to look at these messages in detail.

Click the following to do it all at once:

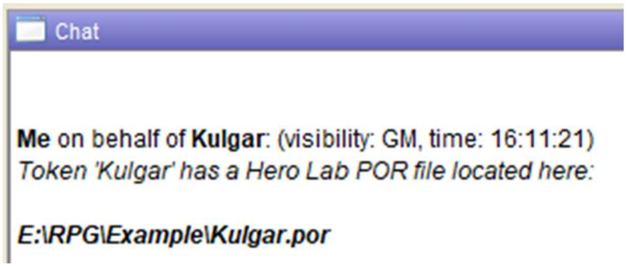

GM
A - Hero Lab(R) POR File Tokens
Sync Token With Hero Lab(R) POR File
Sync & Optimize Token With Hero Lab(R) POR File
View Location Of Token's Hero Lab(R) POR File

Synchronize Token With Hero Lab® POR File

Make sure the edited Hero Lab® POR file is in the same location as when you imported the file. If you've forgotten, click the GM macro "View Location Of Token's Hero Lab® POR File":


MapTool v1.8.3 - Me (GM) - [Example] - Grasslands
File
Edit
Map
View
Tool
Window
Themes
Help
GM
A - Hero Lab(R) POR File Tokens
Sync Token With Hero Lab(R) POR File
Sync & Optimize Token With Hero Lab(R) POR File
View Location Of Token's Hero Lab(R) POR File

You'll see the following:

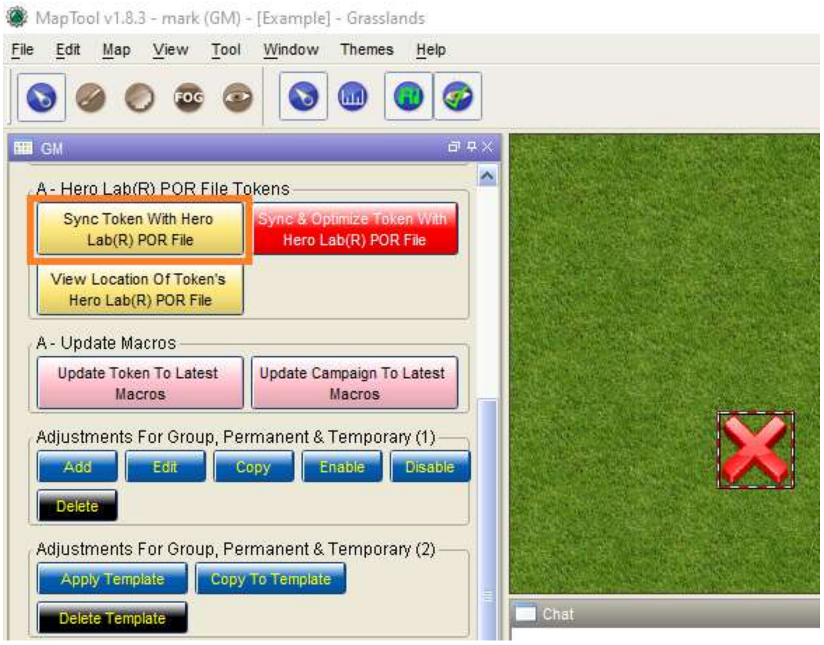

Chat

Me on behalf of **Kulgar**: (visibility: GM, time: 16:11:21)
Token 'Kulgar' has a Hero Lab POR file located here:

E:\RPG\Example\Kulgar.por

Then, there's only one thing you need to do here.

Just click the GM macro "Sync Token With Hero Lab® POR File".


MapTool v1.8.3 - mark (GM) - [Example] - Grasslands
File Edit Map View Tool Window Themes Help
GM
A - Hero Lab(R) POR File Tokens
Sync Token With Hero Lab(R) POR File
Sync & Optimize Token With Hero Lab(R) POR File
View Location Of Token's Hero Lab(R) POR File
A - Update Macros
Update Token To Latest Macros
Update Campaign To Latest Macros
Adjustments For Group, Permanent & Temporary (1)
Add
Edit
Copy
Enable
Disable
Delete
Adjustments For Group, Permanent & Temporary (2)
Apply Template
Copy To Template
Delete Template
Chat

You'll see messages in the Chat window – just like before – and that's it!

Do A Bunch Of Tokens At Once

All you have to do is select multiple tokens that use Hero Lab® and repeat the above steps.

This enables you to walk away and come back from a long import of multiple tokens! It typically takes about 1 minute per token.

What's Not Done Yet

Importing feats, traits, drawbacks, tricks, deeds, special abilities, special attacks, and special qualities from Hero Lab® is not done yet. Because most of the players I play with don't have Hero Lab®, I've been using this feature with NPCs. As a GM, I found that this feature was not urgent – but now I have a player I've been playing with for a long time that has made a major investment in Hero Lab®. You can bet this feature is coming soon!

Overview Of Important Features And Concepts

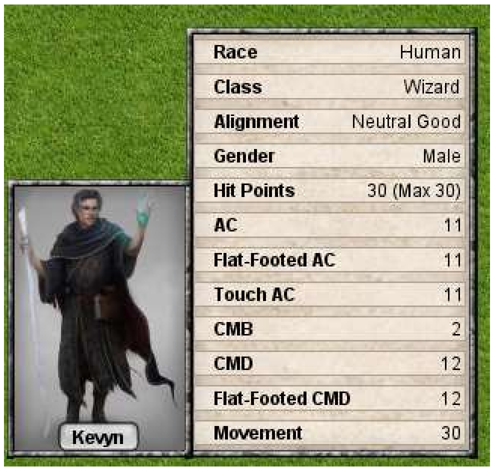
Token Ownership

Generally, players can only affect their own tokens. The only exception is that a player can give the token belonging to another player equipment items – like a torch.

GMs can affect any token.

Stat Sheet

When you mouseover a token, its stat sheet will be displayed as appropriate depending on whether you are the owner, GM, or another player. Here's what I see for a token as the GM:



Race	Human
Class	Wizard
Alignment	Neutral Good
Gender	Male
Hit Points	30 (Max 30)
AC	11
Flat-Footed AC	11
Touch AC	11
CMB	2
CMD	12
Flat-Footed CMD	12
Movement	30

The contents of this stat sheet change automatically as appropriate. New fields appear when needed and others disappear when they are no longer needed.

Campaign Window

This window is used mainly to interact with the selected token's character sheet.

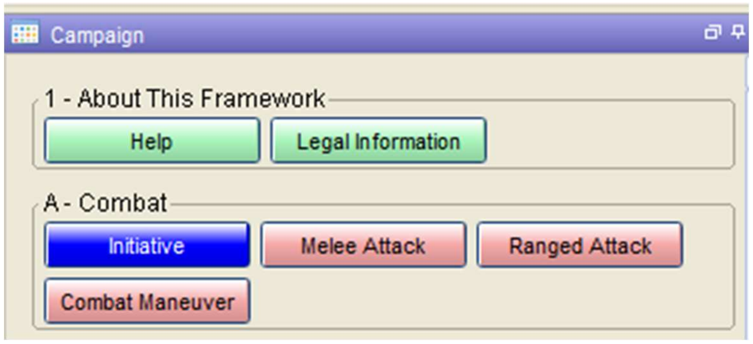
Color Coded Buttons

The buttons for the macros for this window are color coded and have high contrast colors to work better for visually impaired players.

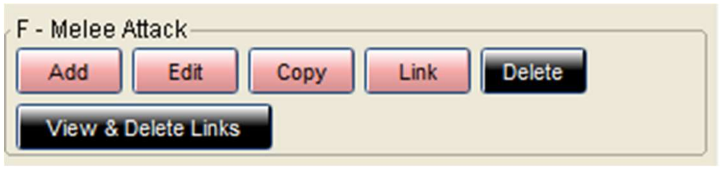
Near the top of the window are macros (buttons) you would use most frequently during game play.

Closer to the bottom of the window are macros (buttons) you would use to edit the character sheet.

Here's an example. To do a melee attack, you'd select the token making the melee attack and press this button near the top of the window:



To add more types of melee attacks, you'd select the token and click the "Add" button below – and notice the color of these buttons match (with "Delete" and "View & Delete Links" an exception):



“Copy” Buttons

“Copy” buttons exist in several places to enable you to copy something and, then, use the “Edit” button to make minor changes.

For melee attacks (see above), it’s common to have different kinds of attacks using, for example, the same weapon.

Say, for example, the character sometimes has a spell cast upon them that increases their size. You might want to have one type of melee attack for when the character is their normal size – and another for when they are enlarged.

“Give Item” Button

This button is used for a selected token to give an item to another selected token. You select both tokens. When you click the “Give Item” button, you will be prompted to indicate which token is giving the item.

“Change Elevation” Button

This button enables you to set the elevation (in feet or whatever unit you want) of the selected token. If the elevation is non-zero, it will be displayed under the token. This helps you keep track of characters that are flying or levitating.

Negative numbers might be used for characters that are below ground or swimming, but I can’t remember if that’s supported.

“Display Special Conditions” Button

Special conditions are things that affect a character in certain circumstances. When you edit different parts of the character sheet, you will often be given the opportunity to describe special conditions.

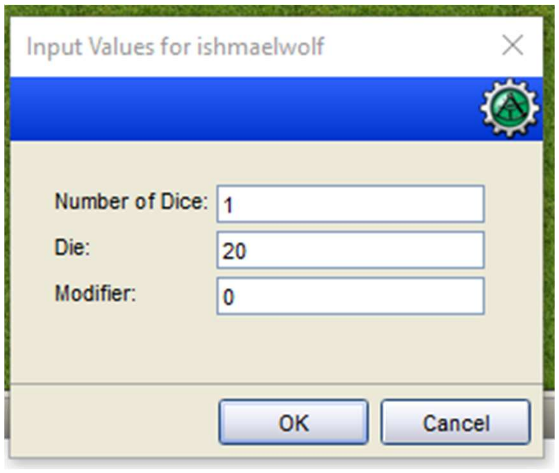
“Only Seen By You” Groups Of Buttons

The idea behind these was to allow players to review their characters without causing an entry to be logged in the Chat window.

When it’s their turn, though, they should use the other buttons so that a Chat window entry will be logged and you have a record of what happened during your game.

“Any” Dice Button

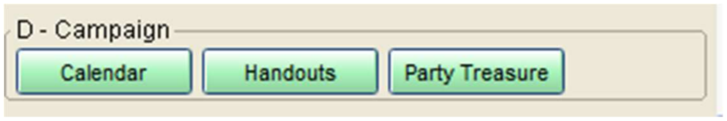
After selecting a token and clicking “Any”, this button brings up a dialog that looks as follows:


Input Values for ishmaelwolf

Number of Dice: 1
Die: 20
Modifier: 0

You can specify any type of die, even something like “3” (for a “d3”).

“Campaign” Buttons



The following buttons display campaign information for players (and GMs). These buttons display information in a player view. No GM information is displayed.

“Handouts” Button

The “Handouts” button requires a token to be selected and displays only those handouts relevant to that token.

“Log” Buttons

These buttons display the relevant information for the selected token. The purpose of these buttons is to display the audit log for the selected token in whatever context is appropriate.

The audit log helps players and GMs keep track of when certain actions or events important to a token occurred – and this feature has been very useful in quickly settling arguments (such as, for example, a player who swears they never casted that spell and should be able to cast it now).

This feature is very useful when a single day in game time lasts for multiple game sessions in real world time.

“Adjustments, Permanent & Temporary” Buttons

Adjustments are those things you do to a character because they’ve picked up a condition, had a spell cast on them, used a light source, had ability damage or drain, get a bonus to a skill, etc.

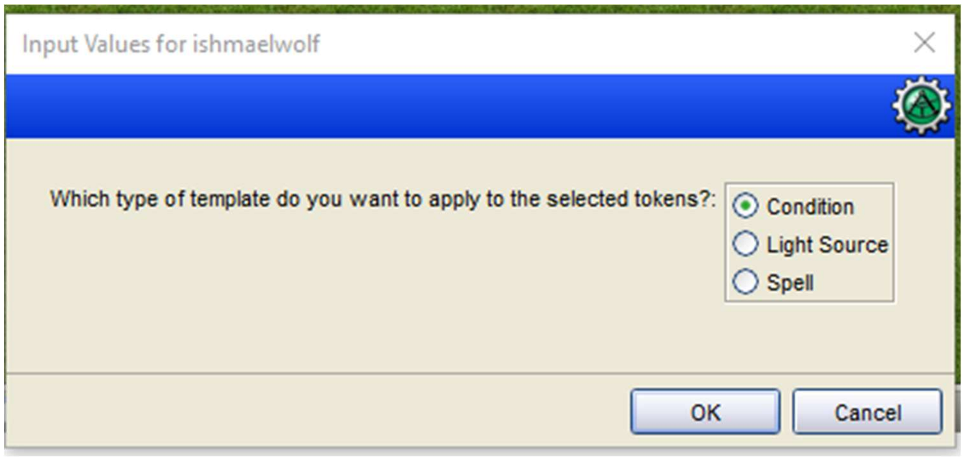
Adjustments are applied automatically to a character when relevant and enabled.

For example, a character that has suffered strength damage will have an adjustment applied automatically to their melee attack rolls – unless some other ability is the ability they use for attacks due to a feat, class feature, or something.

This is a very powerful feature. It can also help you keep track of when adjustments expire. The stat sheet will show these adjustments (if you want).

“Apply Template” Button

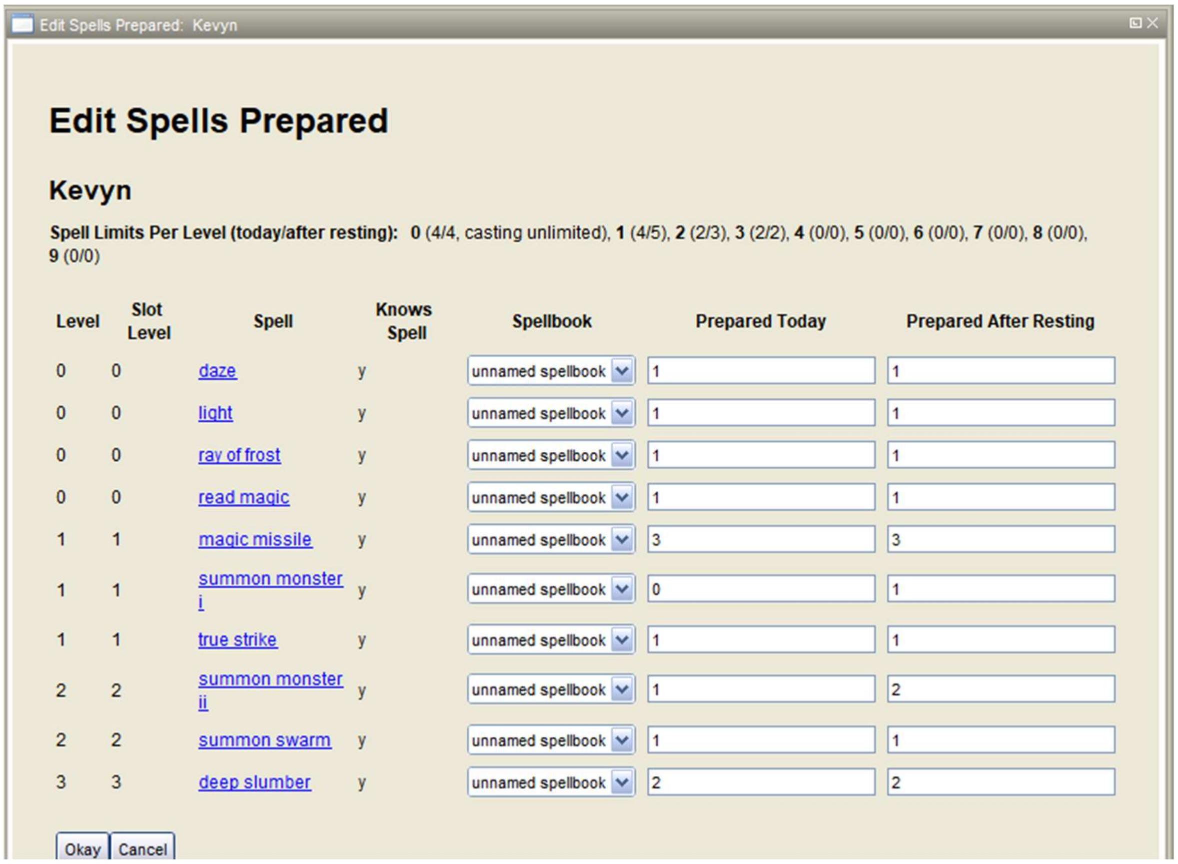
Adjustment “templates” are supported. So, you can setup templates for conditions, light sources, and spells in advance and simply apply them to selected tokens as needed:


Input Values for ishmaelwolf

Which type of template do you want to apply to the selected tokens?: [✓ Condition] Light Source Spell

“Edit Spells Prepared” Button

This feature is for prepared spell casters. It brings up a dialog that looks like this:


Edit Spells Prepared: Kevyn

Edit Spells Prepared

Kevyn

Spell Limits Per Level (today/after resting): **0** (4/4, casting unlimited), **1** (4/5), **2** (2/3), **3** (2/2), **4** (0/0), **5** (0/0), **6** (0/0), **7** (0/0), **8** (0/0), **9** (0/0)

Level	Slot Level	Spell	Knows Spell	Spellbook	Prepared Today	Prepared After Resting
0	0	daze	y	unnamed spellbook	1	1
0	0	light	y	unnamed spellbook	1	1
0	0	ray of frost	y	unnamed spellbook	1	1
0	0	read magic	y	unnamed spellbook	1	1
1	1	magic missile	y	unnamed spellbook	3	3
1	1	summon monster i	y	unnamed spellbook	0	1
1	1	true strike	y	unnamed spellbook	1	1
2	2	summon monster ii	y	unnamed spellbook	1	2
2	2	summon swarm	y	unnamed spellbook	1	1
3	3	deep slumber	y	unnamed spellbook	2	2

Some important notes about this view:

- Spells can be organized into spellbooks (more on this later).
- We can track whether or not a spell is known to the caster (more on this later).
- “Prepared After Resting” allows players to define the spells to be prepared after resting at any time – so there’s no more delay after the GM rests the party. This is a feature I haven’t found in other VTTs.

“Validate Spell Limits” Button

After entering a tokens spells-per-day limits in “Edit Spells Per Day” and, for prepared spell casters, setting up the prepared spells in “Edit Spells Prepared”, this button can be used to produce a report to look for any problems.

Players won’t be prevented from casting spells – but each time they do any problems will be reported.

“Set Item Locations” Button

This feature enables players and GMs to easily move equipment items around and place them in slots on the character or in containers the character carries.

“Rollup” Button

This feature is used to consolidate all the money a character has by the type of coin.

Be careful. This cannot be reversed.

For example, if a character has some gold pieces in a backpack and some in a belt pouch, clicking this button will consolidate all of it into one place.

GM Window

This window is used to manage the campaign, do some in-game things, and do some maintenance activities on the tokens.

(More details coming soon!)