GM User Documentation

Contents

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.

- 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!

What Does MapTool Need

My feeling is that a major feature needed for MapTool is a revamped or replaced VBL (vision 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.

As a GM, this feature is critical for me and my current campaign – though when I was a GM for a campaign with 1st to 6th level characters that was all underground or inside, I did not need those features and had the best time of my life with MapTool. Other VTTs (virtual tabletops) I tried at the time, also did not support elevation.

However, there is now a VTT that does support elevation but I am not as happy with it as MapTool – so I am using all of my influence to try to get the MapTool developers to work on that feature so that I can come back and use MapTool for my more advanced campaigns.

Another way to deal with it, though, if you still want to use MapTool in some of these more difficult scenarios is to grab tokens that should have vision into particular areas blocked off by VBL – and wave them around in that area to at least give them temporary vision.

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:
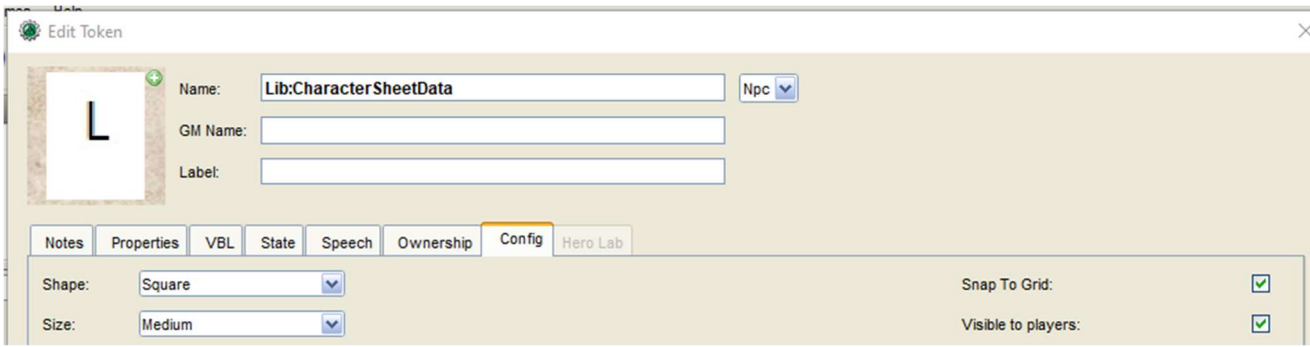
Recommended Optional Additions To This Framework

MapTool Help, Add Default Tables

In MapTool using the Help menu, click on "Add Default Tables":
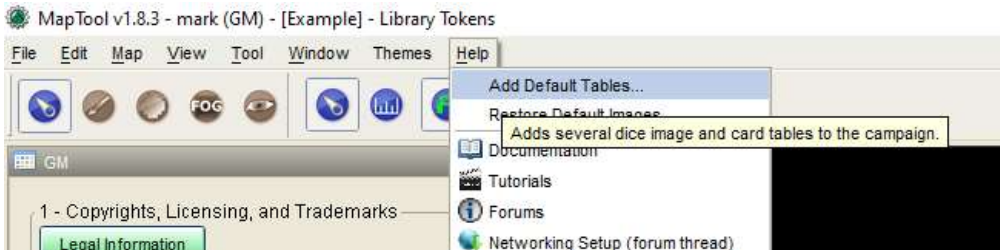
This will add image tables for all the dice, which this framework will then use automatically. If while playing, you try to use a type of die that is not in the image tables, then numbers will be used instead of dice images.

lib:frames

Down load **lib_frames.rptok** and **frames_table.mttable** here.
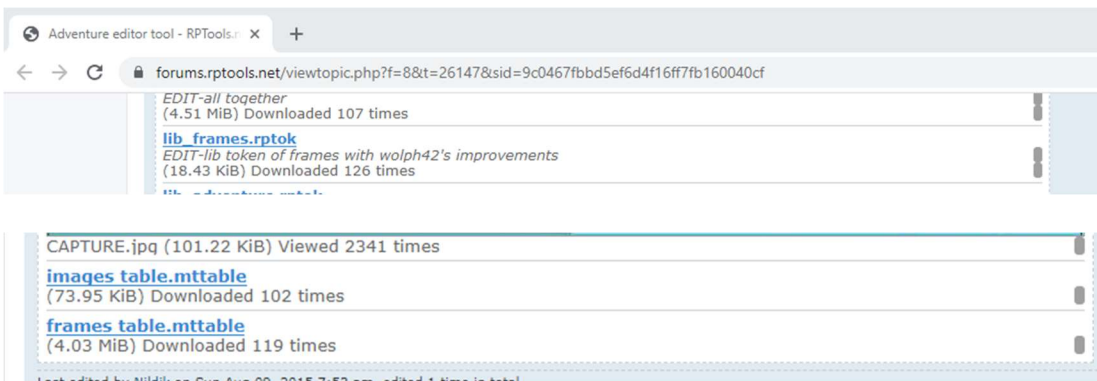
If this is installed along with its associated image table, this framework will use it automatically -- but be sure to setup **both** or you will get an error. I recommend putting it on the same map as the library tokens for this framework. When you do it right, it will look like this:
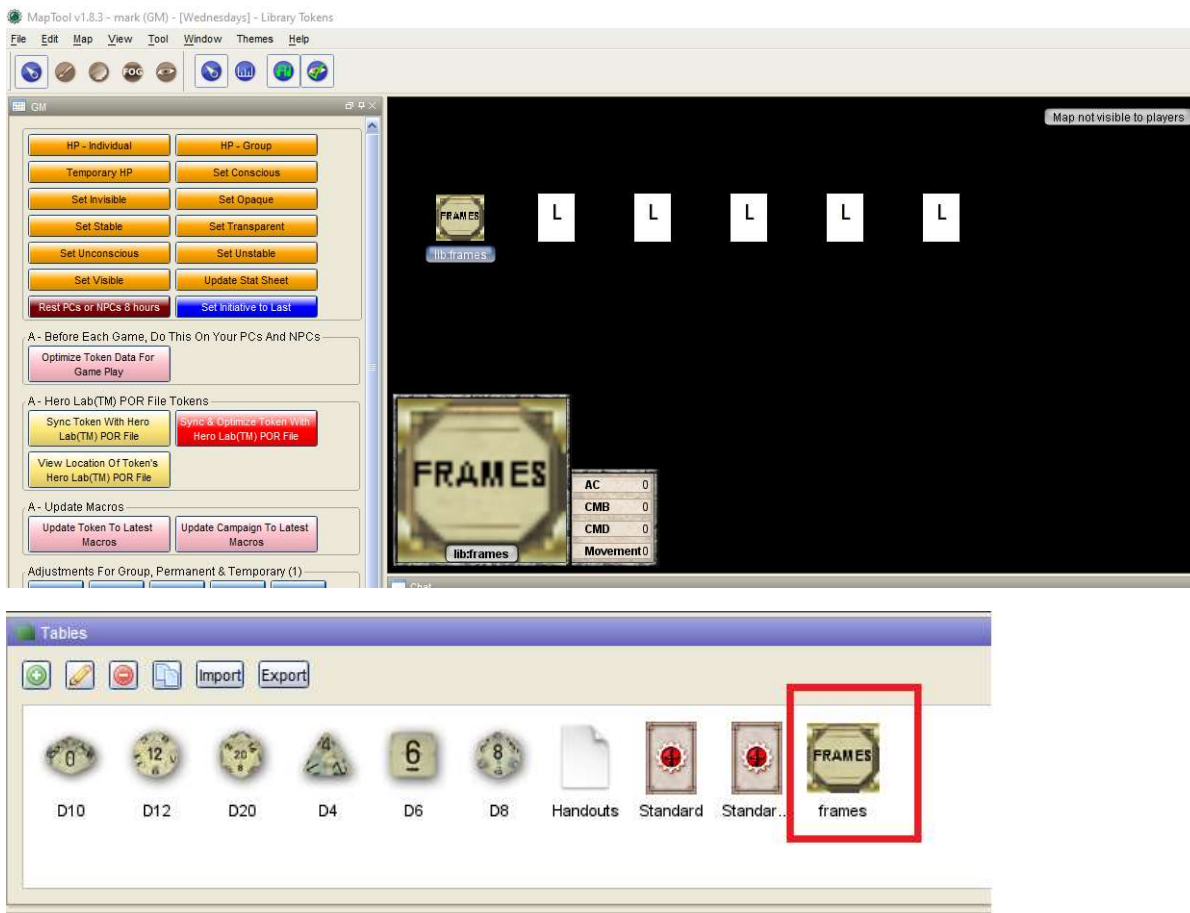
Adding this feature will greatly improve the appearance of the forms created by this framework.

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:

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:

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.
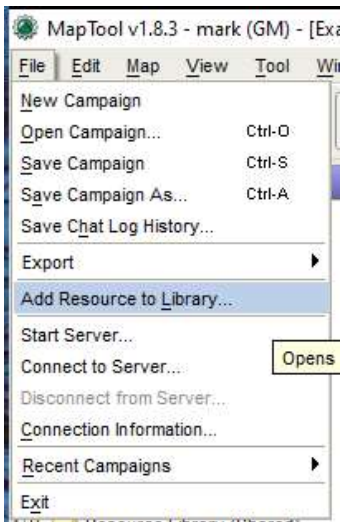
How To Do The Initial Hero Lab® Import

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:
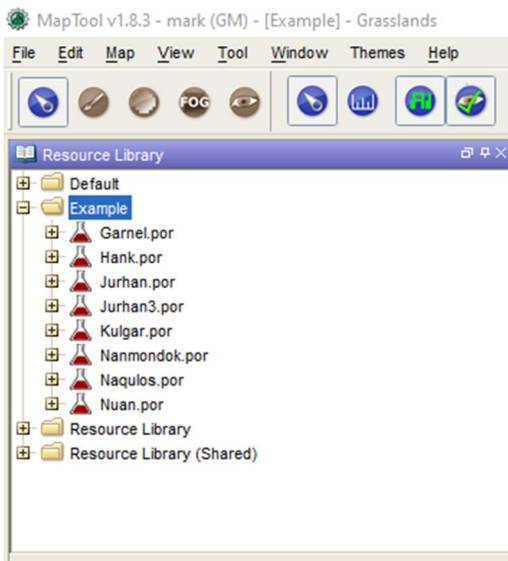
Drag Hero Lab® POR File Token To Map

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"):
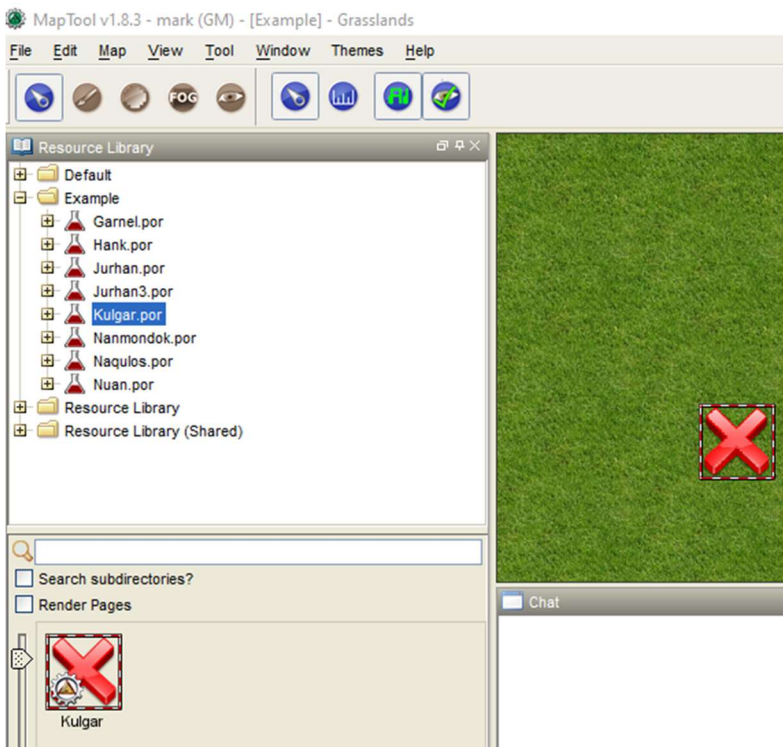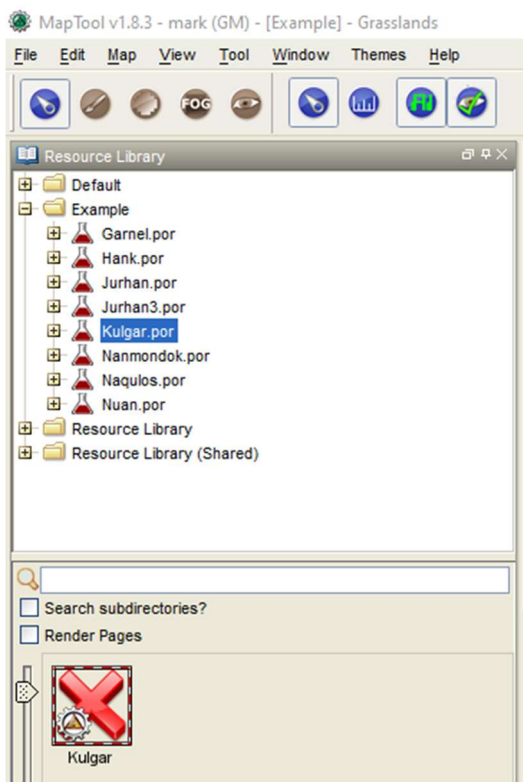
Run GM Macro "Sync Token With Hero Lab® POR File"

With the token selected, run the GM macro "Sync Token With Hero Lab® POR File":
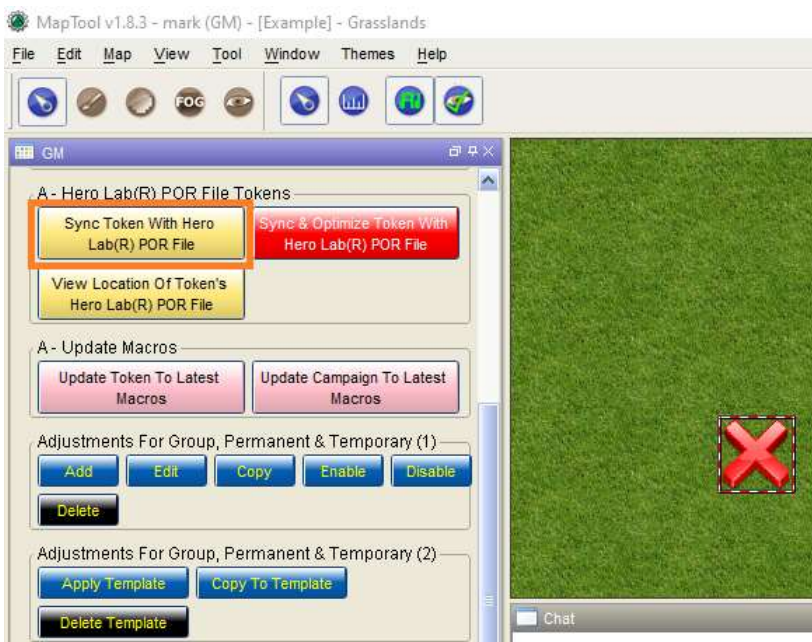
You will, then, see the following dialog:
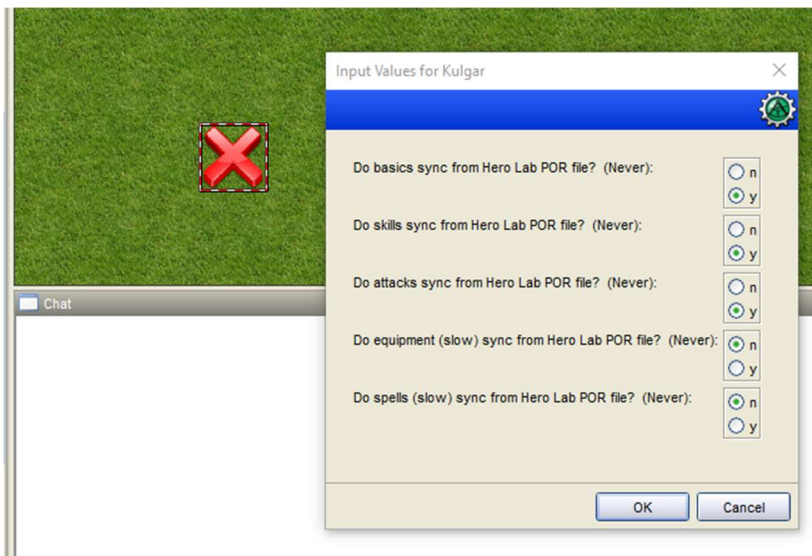
If you are a GM importing NPCs on-the-fly for an encounter, you might just want to import the first three items: basics, skills, and attacks – because importing equipment and spells takes a bit longer.

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

OK   Cancel
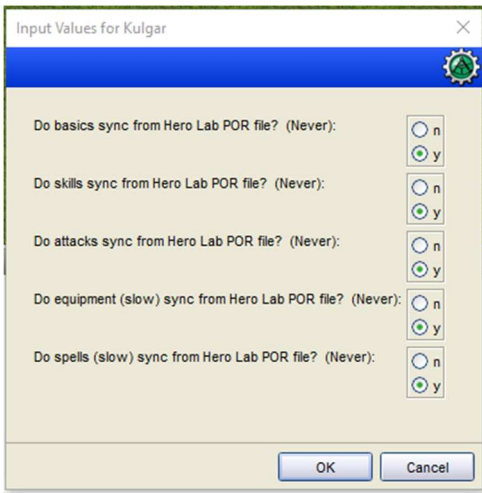
And the Chat window shows:

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!

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":

You'll see the following:

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

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

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!