Getting started

To get started in your own campaign:

1. Go to the Components directory, and import the tokens and tables into your campaign.
2. Import Campaign.mtmacset into your Campaign window.
3. Click the **Initialize Framework** macro on the **Lib:BBFramework Interface** token.
4. Click the **Show DM Tools** macro on the **Lib:BBFramework Interface** token.

Or, use the test campaign if you want to get started right away in a campaign with some PCs and monsters already set up. Just click the **Show DM Tools** macro on the **Lib:BBFramework Interface** token.

Setting up a new monster

1. Export your monster from the D&DI adventure tools as a .monster file.
2. Run the CharacterImporter.jar tool, and import your monster file into a macro set.
3. Create a new token in the same map as the framework tokens, with the image you want to use for your monster. It can have any name.
4. Import the new macro set into the new token.
5. Select the new token and click its **Initialize** macro. The token is automatically renamed and set up with some macros for the monster's powers. You can edit these if you want, to change the text displayed when you fire a monster power.
6. Go to the map in which you want your monster to encounter the PCs.
7. Click the **Spawn** button in the DMtools window. Set how many of each type of monster you want to create, and click **Spawn**. The new tokens will be automatically created and set up.
   If you have a token already selected, the new tokens will be placed immediately to the right of the selected token. If you don't have a token selected when you click **Spawn**, your new tokens will be created at 0,0.
   You can also set up an automatic naming scheme for the new tokens. It is evaluated using the eval() function, so you can include any MapTools functions in the line.
   If at any point you want to reset a monster token to its original state, you can click the **Reset** button in the DMtools window.

Controlling a monster

Unlike framework PCs, each monster token is automatically set up with macros for all of its traits and powers. Open the **Selection** window to see the list of powers available for each monster. The macro buttons have mouseover windows that provide summaries of the powers the way they appear in the Compendium. These macros call their base macros on the library token for the monster, launch attacks if necessary, and track power usage automatically.

When you launch a power with an attack, the power configuration window always opens. You can change things here if you want, but mostly this is just a chance to select the targets for the power. When you have the target tokens selected in the main MapTool window, click **Attack!** and the attack will be rolled and written to chat.

Any framework conditions that you apply to a monster (Dazed, Combat Advantage, etc.) are also represented by macro buttons that get automatically added to the monster token. Pink indicates a macro with a timed expiry (end or start of a turn, or save-ends). Clicking the macro button erases the condition from that monster.

To recharge a used-up power with Recharge, just click its macro button. It should go back to red. Note that if you use the notification system, recharge powers with rolls are recharged automatically at start of turn.

If at any point you want to control a PC that uses the framework, you can. Just like a player, you have to impersonate the PC token, then click its **Refresh All** macro to show and update the frames that list all the PC powers and conditions.

Campaign properties

I heartily recommend stealing the token property list in the campaign properties of the demo campaign. It makes the statsheet show the DM lots of useful information about monsters and PCs automatically when you hover your mouse over the token, including current defenses, speed, etc.

The DMtools window

This window contains some controls that apply to all selected tokens (monsters and PCs).

- Health controls: Enter a number in the box. Click **-** to have the selected tokens take damage, **+** for healing, or **Temp HP** to assign temporary hit points. Use **+surge** to spend a healing surge and add its value to the number in the box (only for PC tokens).
- Condition controls: Select a condition from the list. Click **-** to remove that condition from all selected tokens. Click **+** to add it to all selected tokens. Click **+mod** to modify the condition before adding it (for example, to set when the condition expires). Click **new** to create a new condition from scratch.
- Spawning controls (see previous sections).
- **DCs and Damage**: Opens a handy crib sheet with appropriate DCs for each PC level

(RAW from the Rules Compendium). Also contains the "normal" and "limited" damage by level tables from the DMG; clicking any of the damage expressions rolls the expression and prints the result to chat.

Notifications and condition expiry tracking

The framework can automatically provide notifications of start-of-turn events, end-of-turn events and condition expiry:

- At the start of each creature's turn, that creature's owner(s) will receive a reminder to take ongoing damage, to regenerate, to recharge powers (to the DM only for monsters), and to remove any existing conditions that were set to expire at the start of that creature's turn.
- At the end of each creature's turn, that creature's owner(s) will receive a reminder to save against all save-ends conditions (with links to roll the saves), and to remove any existing conditions that were set to expire at the end of that creature's turn.
- If a creature has a condition that is set to expire at the start or end of a different creature's turn, the affected creature's owner(s) will receive a notification when the expiry of the condition is detected.

In order for this system to operate, you need to use the framework to set up and advance initiative. It's not possible to respond to initiative changes in Maptool. Therefore, in order to provide notifications of events that are tied to initiative changes (like expiring conditions, save-ends conditions, ongoing damage, and regeneration), we have to wrap the commands to change initiative inside a framework function where we also check for conditions.

Running combat with the framework

To set up an encounter:

1. Select all the monsters and PCs that you want included in the encounter.
2. Click the **Add to Initiative (F2)** macro in the Campaign window.
    - All selected tokens are added to an internal encounter list maintained by the framework. This list is printed out for your reference in the Encounter window. You can hide the Encounter window; you don't really need it.
    - One token is added to the MapTool Initiative window for each type of monster. By default, all monsters of the same class use the same starting initiative value, which is rolled automatically.
    - Each PC token is added to the MapTool Initiative window. Initiative for PC and non-framework tokens is set to 0 initially. Players roll their own initiatives and can set them as usual in the Initiative window.
3. When you click the **Update Initiative List (F5)** macro in the Campaign window, or when you advance initiative to the next token in the list (see the next section), the framework

looks for any changes to the initiative of the tokens in the list, and re-sorts the initiative accordingly if necessary.

4. If you want a monster to delay to a different initiative count, you can simply set its new value as usual in the Initiative window or by right-clicking the token. The next time you update or advance initiative, the new value will get adopted and the list will be re-sorted.
    - If you want to set a new initiative for a token that is not currently listed in the initiative window, you can either add the token to the initiative list and then set its value as you normally would, or you can click the number shown for that token in the Encounter window.
    - If you want to set a new initiative value for *all* monsters of a given class, you can click the number shown for that type name in the Encounter window.

During combat:

- Use the **Go to Next Initiative (F3)** macro in the Campaign window to advance initiative to the next token in the encounter list. If the token is an NPC, it is automatically selected for you. The Initiative window will be automatically updated when initiative passes to a new type of monster or to a PC token.
- If you want to switch up the order of initiative for monsters of the same class (i.e. to play their turns in a different order from the default), you can give initiative to a particular token by selecting it and clicking the **Give Initiative (F4)** macro.
- If a creature dies, you can safely delete its token at any time. The next time you advance or update initiative, the monster's absence will be noted and it will be removed from the rotation. If that token was used as the marker for its monster type in the initiative window, the initiative window will automatically be updated to use a different token of the same monster type instead, if any are left.
- If reinforcements arrive, add them to the encounter using the same **Add to Initiative (F2)** macro in the Campaign window that you used at the start of the encounter. If they are a type of monster that already has an initiative count in this encounter, they will automatically be added to the list on the same initiative count.

After combat:

- Click the **Clear Initiative (F10)** macro in the Campaign window to remove all monsters and PCs from initiative in preparation for the next encounter.

DM Preferences

The Preferences macro on the BBFramework Interface token offers three user preferences that relate to notifications:

- **Auto-process NPC notifications**. This setting instructs the framework to automatically process notifications whenever possible. At the start of turn, it will take ongoing damage, apply regeneration, and attempt to recharge powers. At the end of turn, it will automatically **renew** save-ends conditions -- if you enable auto-processing, you must do save-ends conditions yourself. Other timed conditions slated to expire at the start or end

of a character's turn will automatically be removed.
- **Auto-skip PC notifications**. This setting hides notifications for player characters from the DM, but keeps showing notifications for monsters.
- **Auto-skip NPC notifications**. This setting essentially disables the notification system for NPCs.

PC Preferences

The Preferences macro on PC tokens offers two new user preferences:
- **Auto-process notifications**. As the auto-process NPC notifications setting described above, but for that PC token only.
- **Unsubscribe me (*player name*) from notifications**. As the auto-skip settings described above, but only hides the notifications from each player who checks this box. Other owners of the token who do not check this preference will still receive notifications as normal.

Note that the auto-process setting has "priority" -- if a token is set to auto-process, it will not generate notifications anymore (since they are now handled automatically).

Monster Knowledge

The process of handling monster knowledge checks is now delegated to the players. Whenever the players encounter a monster, they can select it and click the **Monster Knowledge (F12)** campaign macro. A window opens, showing the token image and any information known about the monster.

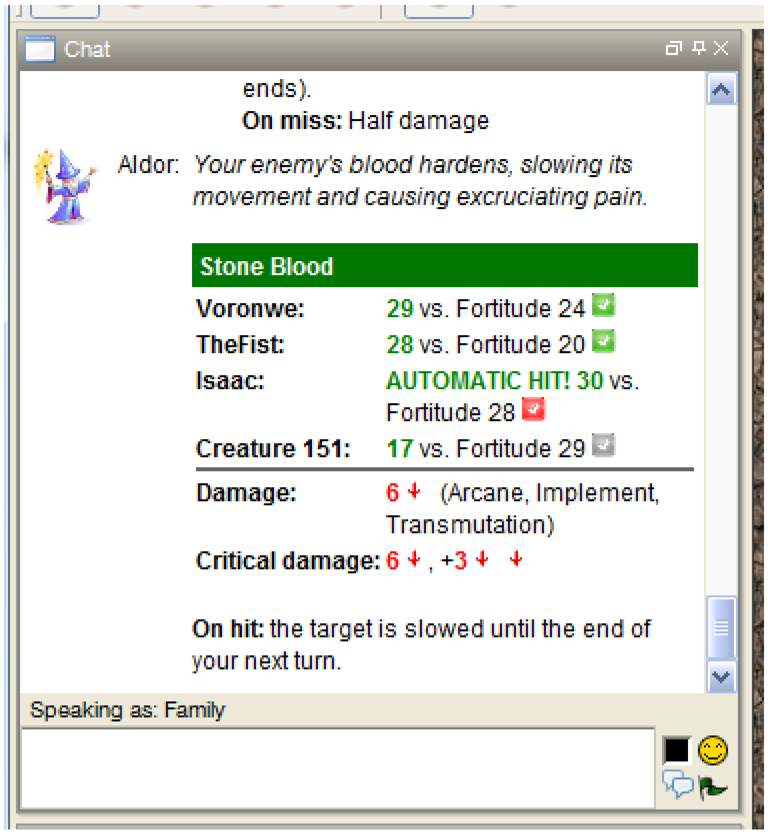
At first, they know nothing about the monster but its size and token image. They need to roll their monster knowledge checks against the appropriate skill (the window attempts to determine the correct skill based on the monster keywords). They enter the best check in the edit field and click **Update**. If their check result exceeds the DC required to know anything about the monster, it will be revealed in the window automatically.

When the DM opens the Monster Knowledge window, he/she can see the full monster details as if looking at its compendium entry. Any info shown with the usual greeny/grey backgrounds is currently visible to players; any info shown with red/pink backgrounds is currently hidden from players.

This goes by RAW from the Rules Compendium: if players beat the moderate DC for the monster's level, they know its name, type and keywords. If they beat the hard DC, they also get to see resistances, vulnerabilities and powers (though I've decided to keep the actual attack bonuses and damage rolls for each power hidden).

Processing attack rolls

Attack rolls made by framework tokens now offer helpful shortcuts for the DM and players to confirm hits and assign damage more rapidly.



ends).
On miss: Half damage

Aldor: *Your enemy's blood hardens, slowing its movement and causing excruciating pain.*

Stone Blood

Voronwe:	29 vs. Fortitude 24 ✓
TheFist:	28 vs. Fortitude 20 ✓
Isaac:	AUTOMATIC HIT! 30 vs. Fortitude 28 ✓
Creature 151:	17 vs. Fortitude 29 ✓
Damage:	6 † (Arcane, Implement, Transmutation)
Critical damage:	6 † , +3 † †

On hit: the target is slowed until the end of your next turn.

DM view of the attack output. The player view shows no checkboxes or defense scores, only the red damage daggers (see last paragraph).

For each target, if the target is a framework token, the current value of the relevant defense is shown after the attack roll. As in the Encounter and Stats windows, blue background indicates a raised value, and purple background indicates a value with a conditional effect that might apply. Mouse over for details. If the attack roll beats the current defense value (for framework monster and PC tokens), a little green checkbox is shown. If the attack roll beats the critical threshold, a red checkbox is shown. If the attack roll misses, the checkbox is grey.

The DM checks the boxes that correspond to the creatures that he decides were successfully hit, which automatically updates which tokens are selected on the map. Then the DM clicks the red dagger that corresponds to the damage that should be dealt to those creatures, which deals the damage to the selected tokens. So, in the example above:

- The DM checks the green box for Voronwe, then the green box for TheFist, then the red dagger next to the value 6 on the damage line to deal those two creatures 6 damage.

- The DM checks the red box for Isaac, which automatically deselects Voronwe and TheFist and changes the selection to Isaac, then clicks the red dagger at the end of the critical damage line to deal Isaac 9 damage.

For critical damage, each component of the damage has its own little red dagger, and an extra dagger at the end of the line reflects the sum of all the components (in this example, 9). The reason for doing things this way is that some critical damage conditions have different possible rolls in different circumstances (e.g. Crusader's Weapon: +1d6 damage per plus, or +1d10 damage per plus against undead creatures). In such a case, you'd have to select the individual conditions that apply, because the framework can't always reliably determine which of those conditions should be included in the damage total.

Note that grey boxes work just the same as green and red, in case the framework makes a mistake, or the player forgets to add some attack bonus that would make the miss into a hit. If the roll is indeed a miss, or if you have a false positive (a green button that shouldn't really have hit), just don't click the checkbox for that roll.

Players have access to the damage daggers only, so they can take damage from monster attacks with a single click. When a player clicks a damage arrow, that damage is dealt to his character (i.e. to the token that player is currently impersonating).

Showing Monster Health

DMs have two user preferences to control how they want to present monster health to players, available by clicking the Preferences macro on the BBFramework Interface token.

- **Show monster health only when bloodied**. This setting hides the monster's health bar until the monster's HP value is less than its bloodied value.
- **Number of health increments**. This setting controls the "resolution" of the health bar -- the number of possible values the health bar can show. For instance, if you set this to 4, the health bar shows full until the NPC's HP value drops below 75%; it then shows ¾ full until the monster becomes bloodied; it then shows ½ full until the monster drops below 25%; it then shows ¼ until the monster dies. The fewer increments you use, the harder it is for players to gauge the actual remaining HP of the monster at any given time.
  You can use a value of 0 to simply make the bar track (as usual) the current hit points divided by max.