

V.3.9.16 BETA (INCLUDES LIBRARY UPDATES)

- Found and fixed an issue preventing players from accessing and editing the saving throw details for owned tokens.
- Added a Tags array to all features, UI pending. This will allow any feature based rule (Features or Spells) to be tagged with things like Spell School, Material Components, or what other details we want to use for a game system.
- Updated feature usage to correctly handle secondary effects on feature usage.
- Overhauled the game log output so feature usage reports correctly for primary and sub-effects.
- Updated the legacy Special Ability log entries so they more closely match the style of the new feature based effect logs.
- Added a two new javascript accessible utility managers and a global java<->javascript helper class which can be addressed internally to D20PRO. In addition, several other GM helper nuggets have been exposed as well. Here's the breakdown:
- (Javascript) => (Java)
- 'abstractApp' => D20PRO application context
- 'maps' => Array of open maps
- 'classBinder' => native collection of CreatureClassTemplates (complex object)
- 'traitBinder' => native collection of FeatureTriggers (Feats and Traits) (complex object)
- 'skillBinder' => native collection of GenericSkills (complex object)
- 'spellBinder' => native collection of GenericSpells (complex object)
- 'statusBinder' => native collection of Status Icons (complex objects)
- 'd20proUtilities' => static helper class with various utilities to move data between javascript and java more easily

D20PROUTILITIES.ACCESSCREATURECLASSTEMPLATE

`CreatureClassTemplate = d20proUtilities.accessCreatureClassTemplate(String className);`

className is a string class name.

returns a CreatureClassTemplate from the binder if it is available, otherwise it creates a new template with default values and the name specified in the call.

D20PROUTILITIES.GENERATEGENERICCREATURECLASS

`GenericCreatureClass = d20proUtilities.generateGenericCreatureClass(String className);`

className is a string class name.

return a GenericCreatureClass generated from the Binder using `d20proUtilities.generateGenericCreatureClass`

D20PROUTILITIES.GENERATEGENERICTRAIT

`GenericTrait = d20proUtilities.generateGenericTrait(String name, String level, String source, String tab, String longDescription)`

name is the trait name

level is the level the trait is awarded/available

source is the book or resource providing the trait

tab is the tab to be sorted onto in the character's trait view

longDescription is the text used to describe the trait (can be HTML)

Returns a GenericTrait object

D20PROUTILITIES.GENERATESPELL

`GenericSpell = d20proUtilities.generateSpell(String name, String level)`

name is the spell name to generate

level is the spell level (not the class level) of the spell

Returns a GenericSpell

V.3.9.16 BETA (INCLUDES LIBRARY UPDATES) CONT.

- In addition to the above, we've also added a PDFManager and a stub for an HTMLManager javascript object set which allows for parsing and consuming content from the associated sources using scripts.

The initial work being done is focused on the 'pdfManager' class and provides tools for reading text and form data from PDF sources then allowing for various actions to be performed to parse this data into useful in-app content.

The PDFManager is based on the Apache PDFBox project while the HTMLManager is based on JSoup.

Neither system is fully functional in this build so I'll refrain from deep diving into the methods of each class for now.

This build will include an example parser and a sample character sheet. Please note the sample uses DNDBeyond as a source however, DNDBeyond's export utility for PDF produces broken PDF forms. In order to repair this form, I've had to convert the source to a PDF/A format using a paid resource on ILovePDF.com. When the parser script is proofed for skills and spells the plan is to take a look at auto detecting the broken form and attempting to repair it in code native to D20PRO.

- Fixed a bug with light visibility for privately owned lights.
- Added a stub feature to enable verbose game log output for skill rolls. This is a proof of concept and will not remain in the state it is in currently. While you can use this feature now, I advise waiting for the full feature before adopting this function. Additionally, the full feature will apply to the range of automated dice roll events, not just skills.
- Lists Library stub created and code work has begun to create a library of List objects to house spell lists, skill lists, gear lists, and much more.
- Library updates!