



CoreRPG Lib User Guide

Contents

Introduction	2
General Comments	3
Notation used in this guide	3
Encounters	4
Equipment	5
Images & Maps	6
Image Pins	7
Magic Items	8
NPCs	9
Reference Manual	10
Story	11
Tables (Rollable)	12
Tokens	13
Treasure Parcels	14
FormattedText	15
#zl;	16

Introduction

PAR5E is a software Parser tool for creating modules and bulk loading data for FantasyGrounds v3.0 or higher rulesets.

The tool (whilst not sharing any source code) has been heavily inspired by the original 4EParser tool by Tenian. It works on a similar basis. You copy and paste text from roleplaying game system rulebooks and modules into text files, mark up the text where necessary and then point PAR5E at them to parse and produce a shiny new module. However, whilst the 4EParser remains forever a valued tool for the D&D 4E FG community, it offers no value to any other FG ruleset user, in addition, it has been some while since it has been actively developed and supported.

In fairness 2013 introduced several other similar 'parser' tools for other rulesets however each addresses the approach differently and again only support specific FG rulesets.

Enter PAR5E.

As data content and FG rulesets are so very tightly coupled, PAR5E has been written modularly in an attempt to enable multiple FG rulesets to be supported.

The PAR5E engine itself does not implement any ruleset specific parsing awareness or logic; instead its ruleset extensibility is implemented through the use of independent ruleset libraries (Java Package and Classes) which are dynamically linked to (at run time) providing the specific ruleset and content parsing functions.

The PAR5E tool itself implements a GUI front-end (Java Swing) to allow the configuration and control of the module output as well as provide the main program control logic for the abstracted parsing process. It provides a library of support functions which can be used by the ruleset libraries for common tasks as well as a main program loop. The main program loop takes care of the sequencing and high level functions such as opening, reading, tokenizing, writing and optimising data files and makes use of registered callback handlers (defined in the ruleset libraries) to perform the actual parsing of ruleset specific content and formatting of output data.

As such, the ruleset libraries define and implement the raw input content parsing methods specific to a role playing game system as well as the output format of the XML specific to a Fantasy Grounds v3 ruleset. When coupled with a suitable ruleset library, PAR5E provides an end to end tool for processing bulk data payloads and producing modules to a high quality standard.

PAR5E is bundled with an example ruleset library for the D&D Next/5E role playing game system which produces modules for the D&D Next/5E Fantasy Grounds ruleset. In addition, PAR5E is bundled with a base ruleset Library for CoreRPG. This base library can be extended to support any CoreRPG derived FG v3 ruleset. The libraries are freely available to the community for re-use and as templates for the development of other ruleset libraries. Further information on developing a ruleset library for PAR5E can be found in the Developing Ruleset Libraries section towards the end of this document.

PAR5E is written in Java and should run on any platform that supports Java and the Swing library. Currently both Microsoft Windows 7 and Apple OS X Mountain Lion have been tested and confirmed as working. Linux and other UNIX based platforms should work albeit have not been specifically tested.

Users who are willing and able to test PAR5E on other platforms including Linux or ruleset developers that wish to exploit PAR5E to enable support for alternative rulesets should contact me for access to the latest development build.

You can contact me via:

- [FantasyGrounds Forum PM](#)
- admin@drzeuss.co.uk

Zeus

Notes

General Comments

This version of the Library supports CoreRPG v3.0.2. As CoreRPG includes only basic reference windowclass definitions, this Library only supports basic campaign class information as well as reference manual text.

Any deviations should be reported on the FG forums in the 5E PAR5E thread.

Users are advised to maintain a copy of the module input files, this should ease effort requirements in re-parsing as and when content formats change.

The Library is released for public testing, whilst the author is confident that the Library performs stably, users are advised to backup any important data prior to parsing with this initial release.

Notation used in this guide

- ◆ Anything in curly braces { } is considered to be variable data.
- ◆ Anything in square braces [] is considered to be a repeatable sequential block.
- ◆ \n designates a carriage return or newline
- ◆ # designates a variable number
- ◆ values separated by a | character represent a choice
- ◆ Lines that begin with -- are ignored. These lines are deemed to be comments.
- ◆ Markup tags are presented as #{n}; and are in **Bold** for easy reference e.g. **#h;**
- ◆ Formatted text fields can contain links to other objects - see #zl;

Encounters

Output to Campaign	Yes
Output to Reference	No
Output to Db	Not supported in this release.
Markup Required	Yes
Dependencies	NPC

Encounters content can be written to both the Campaign/Adventure classes of 5E (under Personalities) and the Reference Library. Encounters content is parsed using a single source file:

◆ encounters.txt

Encounters Data

Input:

- ◆ Empty lines are ignored
- ◆ Encounter content is listed sequentially
- ◆ New categories start with tag **#@;**
- ◆ New encounters start with tag **##;**
- ◆ 1st Line of Encounter is the Name of the Encounter
- ◆ 2nd + Lines list the NPCs for the encounter
- ◆ Personalities can be NPC or Trap/Hazard names.
- ◆ Every Personality line must start with a number.
- ◆ Personalities that do not match will include the invalid npc/trap name but the reference link will be invalid/missing.

```
#@;{Output Category}\n
##;{Encounter Name}\n
[#{NPC_0 Name} Level {#}\n] \n]
```

e.g.

```
#@;B99 The Chaotic Caves
```

```
##;1.Kobold Guard Area
1 Pit Trap Level 1
6 Kobold Level 1
```

```
##;2.Rubbish Heap
5 Rat Level 1
```

Equipment

Output to Campaign	Yes
Output to Reference	No
Output to Db	Not supported in this release.
Markup Required	Tables: Yes Data: No

Equipment can be written to both the Campaign classes of 5E (under Items) and the Reference Library. As Equipment is listed in table form and the descriptions listed separately and alphabetically, Equipment content is parsed using one source file:

◆ equipment.txt

Equipment Table

Input:

- ◆ Empty lines are ignored
- ◆ Equipment Tables must proceed Equipment Data
- ◆ Table data begins with tag **#@;**[ADVENTURING GEAR|ARMOR|WEAPON]
- ◆ 2nd line contains the table field labels, line starts with **#th;**
- ◆ Lines starting with **#st;**{Subtype} key phrase are interpreted as equipment subtypes
- ◆ All table row fields must be separated with a ; (semi-colon) character
- ◆ Incorrect row data will result in warnings and items being skipped

```
[#@;{Table Name}\n
#th;{column header 1; column header 2; ... \n
#st;{subtype}\n
[{column 1 data};{column 2 data}; ... }\n] \n]
```

e.g.

```
[#@;ADVENTURING GEAR
#th;Item;Price;Weight
#st;Standard
Acid (flask);25 gp;1 lb.
Adventurer's kit;9 gp;39 lb.
Alchemist's fire (flask);50 gp;1 lb.
Antitoxin (vial);50 gp;-;
```

Equipment Data

Input:

- ◆ Empty lines are ignored
- ◆ Item names must match names in previous Equipment Table data or else warnings will be highlighted

```
[{name of item}: {description of item}\n]
```

e.g.

```
Blanket, Winter: A heavy blanket perfect for keeping warm or to deaden the sound of
breaking glass.
Block and Tackle: You can use this set of pulleys to lift up to four times the weight
you can normally lift.
```

Images & Maps

Output to Campaign	Yes
Output to Reference	Yes
Output to Db	Not supported in this release.
Markup Required	No

Images can be written to both the Campaign classes of 5E (under Maps & Images) and the Reference Library. Image content is parsed using a single source folder.

◆ Images

Images

Input:

◆ JPG images and maps located in the input images path will be copied to the module.

Image Pins

Output to Campaign	Yes
Output to Reference	Yes
Output to Db	Not supported in this release.
Markup Required	Yes

Dependencies Images

Image pin data can be written to both the Campaign classes of 5E (under Maps & Images) and the Reference Library. Image Pin content is parsed using a single source file:

◆ imagepins.txt

Image Pin Data

Input:

- ◆ Empty lines are ignored
- ◆ Pin content is listed sequentially
- ◆ Each line consists of image/map name, x/y coordinates, class and name of object to link to separated by semi colon ; characters

```
[{name of image/map without extension};{#};{#};{class};{name of object to link to}\n]
```

where # = is x/y coordinate (number)

where {class} = armor|encounter|gear|image|npc|magicitem|map|parcel|referencetext|story|table

e.g.

```
EZ1;100;100;story;B.Orc Lair  
EZ1;100;100;story;C.Orc Lair  
EZ2;100;100;story;D.Goblin Lair Level 1
```


Magic Items

Output to Campaign	Yes
Output to Reference	No
Output to Db	Not supported in this release.
Markup Required	No

Magic Item content can be written to both the Campaign classes of 5E (under Items) and the Reference Library. Magic Item content is parsed using a single source file:

◆ magicitems.txt

Magic Item Data

Input:

- ◆ Empty lines designate the start of a new magic item
- ◆ 1st Line of a new magic item is its name
- ◆ 2nd + Lines of a new magic item are its descriptive text (supports FormattedText tags)

```
[{name of magic item}\n
{Common|Uncommon|Rare|Very rare|Legendary} {magic} {artifact|armor|potion|ring|scroll|
staff|wand|weapon|wondrous items}\n
\n
[{descriptive text line}\n] \n]
```

e.g.

+1 Leather Armor

This magic armor is of fine make, a work as much of art as of utility.

Property: You gain a +1 bonus to AC while you wear this armor.

Javelin of Lightning

This wooden javelin was carved from an ash tree that was struck by lightning. The javelin is tipped with iron, and iron wires with thunderbolt designs trace the javelin's length.

When the weapon is handled, tiny bolts of lightning leap to the wielder's hand from the shaft.

Property: As an action, you can throw the javelin at a point within 100 feet of you. A 5-foot- wide line of lightning travels from you to that point. Each creature in the line must make a DC 13 Dexterity saving throw. The creature takes 4d6 lightning damage on a failed saving throw, and half as much damage on a successful one.

When you use the javelin in this way, it becomes a normal javelin until the next dawn, at which time the magical energy returns.

Wand of Binding

This wand is actually a length of chain held rigid by the magic that infuses it. The chain is crafted from adamantine, and runes of binding and imprisonment adorn each link. Property [Attuned]: You must be a cleric, a druid, or a mage to become attuned to this item.

The wand has 7 charges. As an action, you can expend the specified number of charges to cast one of the following spells from the wand.

#1s;

#1i;hold person (2 charges)

#1i;hold monster (6 charges)

#1e;

If you expend the wand's last charge, roll a d20. On a 1, the wand blackens and crumbles to dust, forever destroyed.

The wand regains 1d6 + 1 expended charges each day at dawn.

In addition, while you hold the wand, you can expend a charge as a reaction whenever you make a saving throw to resist the hold person or the hold monster spell to gain advantage on your saving throw.

NPCs

Output to Campaign	Yes
Output to Reference	No
Output to Db	Not supported in this release.
Markup Required	No

NPC content can be written to both the Campaign/Adventure classes of 5E (under Personalities) and the Reference Library. NPC content is parsed using a single source file:

♦ npcs.txt

NPC Data

Input:

- ♦ Empty lines denote the start of a new NPC
- ♦ Resistances, Vulnerabilities and Immunities can span multiple lines
- ♦ TRAIT/POWER names must start on a new line
- ♦ TRAIT/POWER description text can span multiple lines

```
[{Name of Monster}\n
Space #\n
Reach #\n
Skills {skills}\n
Languages {languages}\n
Equipment {equipment}\n
Notes
[{descriptive text}\n] \n]
```

e.g.

```
Large Snake
Space 1
Reach 1
Skills -
Languages -
Equipment -
Notes
Hit Points 150 (17d10 + 40)
Speed 40 ft., climb 20 ft.
Senses darkvision 60 ft.
Str 18 (+5) Dex 16 (+2) Con 20 (+4)
Int 2 (-4) Wis 10 (+0) Cha 3 (-4)
Alignment unaligned
Languages -
...
...
```

Reference Manual

Output to Campaign	No
Output to Reference	Yes
Output to Db	Not supported in this release.
Markup Required	Yes

Dependencies Only linked objects

Reference Manual content can be written to the Reference Library only. Reference Manual content is parsed using a single source file:

◆ refmanual.txt

Reference Manual Data

Input:

- ◆ Empty lines are ignored
- ◆ Reference Manual content is listed sequentially
- ◆ New chapters start with keyword **#@;**{Chapter Name}
- ◆ New sub chapters start with keyword **#!;**{Subchapter Name}
- ◆ New Reference Pages start with keyword **##;**{Reference Page Name}
- ◆ Reference text starts on the first line after **##;**
- ◆ Reference text is automatically formatted with formattedtext <tags>, there is no need to add tags manually unless further customisation is required. Each text line is treated as a paragraph <p></p>. Text preceding a colon (:) will be auto bolded .
- ◆ The reference text supports FormattedText tags (see FormattedText section)

```
[#@;{Chapter Name}\n
[#!;{Subchapter Name}\n]
[##;{Ref Page Name}\n
[{text line} \n] ]\n ]
```

e.g.

```
#@;Chapter 1
#!;SubChapter title
##;Introduction
The adventures that unfold in the ...
...
...
#h;A Heading
...
....
...
#ls;
#li;1st line
#li;2nd line
#li;3rd line
#le;
...

#!;Basic Rules
##;Another Page Title
...
...
...

##;And Another Page Title
...
...
...
```

Story

Output to Campaign	Yes
Output to Reference	No
Output to Db	Not supported in this release.
Markup Required	Yes

Dependencies Only ZLINK objects

Story content can be written to the Campaign classes of 5E (under Story) only. Story content is parsed using a single source file:

- ◆ Story Data

Story Data

Input:

- ◆ Empty lines are ignored
- ◆ Story content is listed sequentially
- ◆ New categories start with tag **#@;**
- ◆ New Story pages start with tag **##;**
- ◆ Story text is automatically formatted with formattedtext <tags>, there is no need to add tags manually unless further customisation is required. Each text line is treated as a paragraph <p></p>. Text preceding a colon (:) will be bolded .
- ◆ Text supports FormattedText tags (see FormattedText section).

```
[#@;{Output Category}\n
[##;{Page Name}\n
[{text line} \n] ]\n n]
```

e.g.

```
#@;B99 The Chaotic Caves

##;Overview
#h;Dungeon Module B99 The Chaotic Caves
#h;By DrZ
An Adventure for Character Levels 1-3
General Notes: This module ...

##;(Contents)
#h;B99 The Chaotic Caves
#zls;
#z1;story;Overview
#z1;story;What Is This Adventure About?
#z1;story;Make This Your Adventure
#z1;story;Running the Adventure
#z1;story;Introduction
#zle;

##;What Is This Adventure About?
The Chaotic Caves ...
...
```

Tables (Rollable)

Output to Campaign	Yes
Output to Reference	No
Output to Db	Not supported in this release.
Markup Required	Yes

Tables (Rollable) content can be written to the Campaign classes of 5E (under Tables) only. Table content is parsed using a single source file:

◆ tables.txt

Table Data

Input:

- ◆ Empty lines are ignored
- ◆ Table content is listed sequentially
- ◆ New categories start with tag **#@;**
- ◆ New tables are formatted as follows:
 - ◆ **#@;**Category Name
 - ◆ **##;**Table Name
 - ◆ **#!;**1 line short description
 - ◆ 1 or more lines of notes
 - ◆ 1 line of Column labels
 - ◆ 1 or more lines of row data

```
[#@;{Output Category}\n
[##;{Name of Table}\n
{#!;Short description}\n
[{notes}\n]
column;[{column label}; {column2 label};]\n
[row;#;#;[{col1 text}; {col2 text};]\n \n]
```

e.g.

```
#@;B99 The Chaotic Caves
##;Random Encounter
#!;Random Encounter Table
The hungry creatures inhabiting ...
...
column;Encounter
row;1;1;Owlbear
row;2;2;1d6 + 6 rats
...

##;Random Treasure
#!;Random Treasure Table
Many of the ...
....
column;Coin;Gems
row;1;3;1 cp;5 Topaz
row;4;5;1 sp;3 Sapphires
row;6;6;1 ep;2 Diamonds
```

Tokens

Output to Campaign	Yes
Output to Reference	Yes
Output to Db	Not supported in this release.
Markup Required	No

Tokens can be linked to both the Campaign classes of 5E (under Personalities) and the Reference Library (NPC). Token content is parsed using a single source folder.

◆ tokens

Tokens

Input:

- ◆ PNG tokens located in the token path will be copied to the module.
- ◆ Tokens which match the sane name of Traps/NPCs will be linked to the objects. A sane name is a name converted to lowercase and with all spaces and non-alpha characters converted to _ characters. e.g.
Fred Blog = fred_blog.png
- ◆ If a sane name match cannot be made, an attempt to match the first letter of the NPC/Trap against tokens is made (caters for generic letter tokens).

Treasure Parcels

Output to Campaign	Yes
Output to Reference	No
Output to Db	Not supported in this release.
Markup Required	No
Dependencies	Items

Treasure Parcel content can be written to the Campaign classes of 5E (under Items). Treasure Parcel content is parsed using a single source file:

◆ parcels.txt

Treasure Parcel Data

Input:

- ◆ Empty lines are ignored
- ◆ Treasure Parcels content is listed sequentially
- ◆ New Categories start with tag **#@;**
- ◆ New Treasure Parcels start with tag **##;**
- ◆ 1st Line of Treasure Parcel is the Name of the Treasure Parcel
- ◆ 2nd + Lines list the coins and items for the Treasure Parcel seperated by ;

```
[#@;{Output Category}\n
[##;{Name}\n
[{coin|item};#{description} \n] \n]
```

e.g.

```
#@;B99 The Chaotic Caves
##;1.Guard Area Level 1
coin;100;PP
coin;1000;GP
coin;2000;EP
coin;3000;SP
coin;4000;CP
item;1;Cloak of Invisibility
item;1;Ring of Feather Falling
```

```
##;4.Elite Guard Room Level 1
coin;100;PP
coin;1000;GP
coin;2000;EP
coin;3000;SP
coin;4000;CP
item;1;Quarterstaff
item;1;Short sword
item;1;Elven Chain
item;1;Dagger of Venom
```

FormattedText

Where data supports FormattedText field types, PAR5E supports the following tags for easy markup. Note untagged plain text is treated as paragraphs. Paragraphs containing colon characters will make preceding text bolded automatically.

Tag	Description	Markup Used
#h; {text}	Marks the text as header text.	<h>{text}</h>
#p; {text}	Marks the text as paragrah text.	<p>{text}</p>
#b; {text}	Marks the text as bold text.	{text}
#i; {text}	Marks the text as italic text.	<i>{text}</i>
#bp; {text1}:{text2}	Marks the first text as bold text. Second text is plain.	<p>{text1}:{text2}</p>
#ip; {text1}:{text2}	Marks the first text as italic text. Second text is plain.	<p><i>{text1}</i>:{text2}</p>
#ts;	Marks the start of a new table	<table>
#th; {text};{text}	Marks the text as table header text.	<tr decoration="underline"><td>{text} ...
#tr; {text};{text}	Marks the text as a table row.	<tr><td>{text}</td><td>{text}</td></tr>
#te;	Marks the end of a table.	</table>
#ls;	Marks the start of a new list	<list>
#li; {text}	Marks the text as a list row.	{text}
#le;	Marks the end of a list.	</list>
#zfs;	Marks the start of a new Frame.	<frame>
#zfi; {text}	Marks the text as the Frame speaker.	<frameid>{text}</frameid>
#zft; {text}	Marks the text as frame text.	{text}
#zfe;	Marks the end of a Frame	</frame>
#zls;	Marks the start of a new Link List	<listlink>
#zl; {class};{name}	Marks the text a link element	<link class={class} recordname=....
#zle;	Marks the end of a Link List.	</linklist>
#h; {text}	Marks the text as header text.	<h>{text}</h>
#h; {text}	Marks the text as header text.	<h>{text}</h>
#h; {text}	Marks the text as header text.	<h>{text}</h>
#h; {text}	Marks the text as header text.	<h>{text}</h>
#h; {text}	Marks the text as header text.	<h>{text}</h>

#zl;

#zl; is a special tag that enables content to be cross referenced or linked from another object. **#zl;** is supported with formatted text fields only and can only reference content within the same module.

Where supported a **#zl;** list of references to singular objects can be started by using the following syntax:

```
#zls;\n[#zl;{class};{name of object to link to}\n#zle;\n
```

where {class} can be one of:

```
{characterclassname}_ability\nbackground\nclass\nencounter\nfeat\n{characterclassname}_feature\nimage\nitem\nnpc\nreferenceadventuringgear\nreferencearmor\nreferencemagicitem\nreferencenpc\nreferencetext\nreferenceweapon\nskill\nspell\nspeciality\nstory\ntrap\ntreasureparcel
```

where {characterclassname} is a name of a valid character class such as barbarian, cleric, fighter, monk, rogue or wizard

Where supported a **#zl;** list of references to object lists can be started by using the following syntax

```
#zls;\n[#zl;{classlist};{name of list to link to}\n#zle;\n
```

where {classlist} can be one of:

```
abilitylist\nadventuringgearlist\narmorlist\nbackgroundlist\nclasslist\nfeatlist\nfeaturelist\nnpclist\nmagicitemlist\nskilllist\nspecialitylist\nspelllist\ntraplist
```

e.g. From a story page

```
#@;Test
##;Test Page
This is a test of ZLINK
#h;Singular Objects
#zls;
#z1;monk_ability;Flames of the Phoenix
#z1;rogue_ability;Climb Sheer Surfaces
#z1;background;Commoner
#z1;class;Wizard
#z1;ncounter;2.Garbage Heap
#z1;feat;Defensive Ward
#z1;monk_feature;Level 1: Monastic Tradition
#z1;rogue_feature;Level 1: Skill Mastery
#z1;image;EZ1
#z1;item;Crowbar
#z1;item;Dragon Leather
#z1;item;Spellguard Shield
#z1;item;Net
#z1;npc;Kobold
#z1;referenceadventuringgear;Caltrops
#z1;referencearmor;Studded Leather
#z1;referencemagicitem;Dancing Sword
#z1;referencenpc;Orc
#z1;referencetext;Saving Throws
#z1;referenceweapon;Bolas
#z1;skill;Perform
#z1;spell;Magic Missile
#z1;speciality;Metamagician
#z1;story;Make This Your Adventure
#z1;table;32.Gallery of Death Random Treasure
#z1;trap;Green Slime
#z1;treasureparcel;4.Elite Guard Room Level 1
#zle;
#h;Lists
#zls;
#z1;abilitylist;Abilities
#z1;adventuringgearlist;Adventuring Gear Table
#z1;armorlist;Armor Table
#z1;backgroundlist;Backgrounds
#z1;classlist;Classes
#z1;featlist;Feats
#z1;featurelist;Features
#z1;npc;NPCs
#z1;magicitemlist;Magic Items
#z1;skilllist;Skills
#z1;spelllist;Spells
#z1;specialitylist;Specialities
#z1;traplist;Traps
#zle;
```