

Interactive Fiction Document (IFD) Format

Specification and Information

V3

Created by: Dr. Ray

New for Version 4

Version 4 of the Interactive Fiction Document format is an additive extension of Version 3. Everything that was true about Version 3 is also true about Version 4. Generally speaking, what's added to Version 4 is support for armor type items, which increase HP to protect the user from the dangers present in the game, lives, which allow the player to have multiple chances at completing the game, and teleportation, which allows the user to teleport to any area they have previously been in.

Overview of IFD - Purpose

The Interactive Fiction Document (IFD) format is a mark-up style language (think HTML) for specifying interactive fiction games in a simple plain-text file. With such languages, meta-data “tags” are used to represent the meaning of text in the file. Tags have the following format: <*tagname*> is the “open” label of the tag (where “*tagname*” is the name of the tag) and </*tagname*> is the “close” label. Tags always come in label pairs (open and close label pairs): the first without the ‘/’ opens the tag and the second with the ‘/’ before the tagname closes the tag. All text (strings, values, or other nested tags) inside the tag label pairs are part of that tag’s information. When a tag’s open label is nested inside another tag, then its close label must also be inside that same tag, and vice versa.

For IFD Version 2, representable interactive fiction games consist of any number of discrete game “areas” (rooms, zones, etc.) that are interconnected with one another via “links”. This structure represents the areas that a player can move around in, constituting the “map” or “world” of the game. Additionally, in-game interactive “items” are supported. Items have certain characteristics that are detailed by the language (see below), one of which is the area (by number) that they begin the game in.

Specifications for the IFD Format

In the two sections below we list first, the available tags in IFD version 2, and then the required tag structure for a valid IFD document.

Supported Tags

The following is a list of valid tags in this version of the format:

- game: top-level tag, all other tags and information for the game must be nested between the `<game>` and `</game>` labels
- ptype: the ptype tag indicates the creation of a new player; the text inside the tag should list the player type; this is extensible but the two provided types are “basic” and “hpsp”
- area: the area tag indicates the creation of a new area in the game; information about the area is nested within
- desc: this is the “description” tag which contains the textual description (of an area when nested inside an area tag)
- feat: this is the “features” tag that describes an area; only two features are captured: in stadeath and goal (additional information below)
- linkset: this tag represents the in-game connections from one area to four other areas (oriented up, down, left and right)
- links: this is the container tag for all the game’s linksets
- item: the item tag indicates the creation of a new in-game item; any number of item tags can be present; information about the item is nested within; each item must be specified as a specific type (by a single, nested third-level tag, see next tag)
 - basic, use, consume: these are the three types of items
 - name: this is the tag that identifies the “name” of the item in which the name tag is nested
 - desc: this is the “description” tag which contains the textual description (of an item when nested inside an area tag)
 - star: the data in this tag represents the “start area” of the number
 - actmess: the message displayed when an item is “used” or “consumed” • *NEW* actar: the number of the area where the item can be activated (0 means any area)
 - rule: for “use” items, there are any number of nested rule tags. Each tag contains a triples; affected area number, direction, and destination area number. So a rule that contains 1, u, 2 will update the up link in area 1 to point to area 2.
 - effect: for “consume” items, there are any number of effect tags. Each tag contains a comma separated integer couplet. It is up to the specific Player object to interpret the included couplet.
 - *NEW* lives: how many lives the player is given
 - *NEW* armor: tag used inside armor tag to declare item type

- *NEW* life: tag used inside the armor tag, used to show how much HP the armor adds to the player

Required Structure

Tags in the IFD format exist in one of three hierarchical “levels”, listed below as top-level through level 3.

Top-Level Tag: game

All IFD files begin with the `<game>`, and end with the `</game>` label (game is the top level tag). Components inside this tag mean various things, but are all part of the defined game. Specifically, the second-level tags inside the game tag include: `<pype> . . . </pype>`, `<area> . . . </area>`, `<links> . . . </links>`, and `<item> . . . </item>`.

Second-Level Tag: ptype

There should be only one ptype tag. The text inside the tag should coorespond to a specific instance of a Player type. This is extensible: derive a new class to implement the Player class, and update the code to respond to that specific text. The two player types currently are the BasicPlayer class and the HPSPlayer class, which will be created with the text, “basic” and “hpsp”, respectively.

Second-Level Tag: lives

There should be only one lives tag. The text inside the tag should be a single integer.

Second-Level Tag: area

There can be any number of area tags. Internally, areas will be assigned a number from $1 \dots n$ where n is the total number of areas. The first area tag will be assigned area 1, the second 2, etc. The player will always begin the game in area 1.

Each area tag must contain exactly two third level tags (both are required, both may appear only once per area tag): `<desc>` (description) and `<feats>` (features). These are the only tags that can appear nested in an area tag.

area Third-Level Tag: desc

Between the `<desc>` and `</desc>` labels a string of characters representing the area description must appear. This string may take up more than a single line of text but not contain a line break.

area Third-Level Tag: feat

Between the `<feats>` and `</feats>` are two comma-separated boolean values representing the insta-death and goal area features respectively. (1 for true, 0 for false, no spaces may

appear in the comma separated list.

Second-Level Tag: links

Exactly one *links* tag must appear in each IFD file. Only one kind of third-level tag can appear inside the *links* tag, the *linkset* tag. There should be precisely n *linkset* tags where n is the number of total area tags.

IMPORTANT: the order of the *linkset* tags nested in the *link* tag is critical. The first linkset represents the in-game connections emanating from the first area; the second from the second, so forth and so on.

links Third-Level Tag: linkset

Each *linkset* tag contains a single set of 4 comma-separated values. The first linkset represents the links (up, down, left, and right, respectively) for area 1. The second linkset for area 2, etc. as before.

Second-Level Tag: item

There can be any number of item tags. They *do not* have to appear in the file immediately after the area that they are starting in (though this was the style used in “DragonCrawl2.ifd”). It is perfectly okay to choose to put all item tags together at the end of the file, for instance, it’s just that approach is less organized.

NEW item Third-Level Tag: basic, use, consume

Each *item* tag must be immediately followed by a nested type tag; either, *basic*, *use*, or *consume*, representing basic items, items that respond to the use command, and items that respond to the consume command.

Each of these type tags must contain the other tags related to the item. There are three required fourth-level tags (these three are required and may appear only once per item): *<name>*, *<desc>*, and *<star>* (start area). In addition, there may also appear, *<actmess>* (“activate message”), *<actar>* (“activate area”), *<rule>* (representing a rule for modifying the game map), and *<effect>* (representing an effect on the player).

item Fourth-Level Tag: name Between the *<name>* and *</name>* labels a string of characters representing the item name must appear. This string may take up more than a single line of text (again, no line breaks) but likely will only be a few words long.

item Fourth-Level Tag: desc As before, between the *<desc>* and *</desc>* labels a string of characters representing the item description must appear. This string may take up more than a single line of text but contain no line breaks.

item Fourth-Level Tag: star Between the *<star>* and *</star>* labels a single integer must appear. This value represents the “start area” of the item (aka, the number of the area the item begins the game in).

NEW item Fourth-Level Tag: *actmess* Only items of type *use* or *consume* should contain the <*actmess*> tag. Each item of these types of items should have only one such tag. Inside the tag is the text string that will appear when the item is used or consumed.

NEW item Fourth-Level Tag: *actar* Only items of type *use* or *consume* should contain the <*actar*> tag. Each item of these types of items should have only one such tag. Inside the tag is a single integer representing the number of the area where the item may be used or consumed. The value 0 represents that an item may be used or consumed in any area.

NEW item Fourth-Level Tag: *rule* Only items of type *use* may contain the <*rule*> tag. Such items may contain any number of <*rule*> tags. Each <*rule*> tag contains a comma separated triplet representing a single change in a link between area nodes on the game map. For instance, the triplet 3, d, 6 should be understood to mean to change the “down” link for area 3 to now point to area 6. When an item is successfully used (remember, an item may only be used in the correct “activation area”), all of its rules are executed on the map.

NEW item Fourth-Level Tag: *effect* Only items of type *consume* may contain the <*effect*> tag. Such items may contain any number of <*effect*> tags. Each <*effect*> tag contains a comma separated couplet representing a change to the player (often to player stats). The interpretation of the specific values in the couplet is dependant upon the type of player being used for the game.

Additional Notes

One additional note, IFD format also allows for comments. Any line that begins with the # symbol is a comment line. These are particularly useful when trying to keep track of an area’s number (which are automatically assigned in the order that the area tags are listed from top to bottom in the file). Comments can be used for this and, of course, other reasons. Effectively, each comment takes up one line of text; there is no block comment structure supported except for successive comment lines. As expected, comments should be ignored by any game engine that processes IFD formatted files.

Blank lines in the IFD format are ignored.

Example File

One or more sample files will be available to you to see the updated IFD format in action. Generally, these files will be stored with .ifd file extensions, but remember, all IFD formatted files are stored in plain-text documents. The .ifd file extension is for identification purposes only and IFD formatted game files can be created, read and manipulated using any plain-text editor. As such building your own IFD game files is something you are not only able to do, but encouraged to do.