

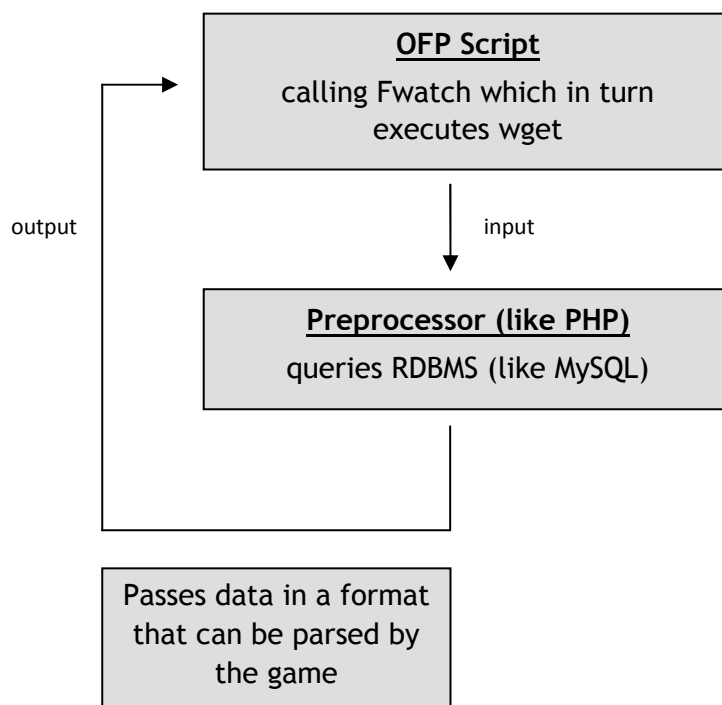
OPERATION FLASHPOINT ADDON DOWNLOAD METADATA STANDARD

0.5

In this document I've described a communication standard between the special OFP script and the web server storing information about OFP accessories.

I've written it for the websites administrators willing to give alternative access to their addon tables by using server-side script processing specific requests and returning information compliant with the standard described here.

Overview



Script "LoadMod . sqs" (version supporting downloads will be supplied with Fwatch v1.13) calls Fwatch to execute program WGET using HTTP protocol to access given URL (with query string attached). Target script on a server retrieves input (GET method), determines request type, connects to the database (if necessary) and returns data in SQF syntax.

Text format for input and output is ANSI.

Types of requests

The OFP Script can send one of the three kinds of requests:

- for addresses to other sites using similar script
- to search for given title in the database
- for a fixed list of files

For any other type of request return error information (see “Output Variables” section).

a) Other Sites

Input: nothing

Output: array with URLs

The point of this request is to spread information about other websites allowing for the same type of data exchange so that the OFP script can build up a list of possible search sites.

In this approach there's no master server and no requirement for the user to paste new URLs on his side. Administrator manually fills up the list with web addresses.

Default source will be a script on the ofp-faguss.com domain.

The OFP script appends question mark (if it already occurs then ampersand) and variables to the URL prior to sending a request.

b) Search

Input: search string and optionally page number

Output: array with file information and optionally number of results and pages

Search request is intended for seeking information about given addon in the database(s). Results and page numbers allow to implement pagination. File information consists of an addon title, (optionally) file size and installation script (or just a link to the file).

c) **Fixed list**

Input: nothing

Output: array with file information

Asks for the list of files selected by the administrator. The goal is to give quick access to the recent or commonly used files. Consider it a “front page”.

Input Variables

Query string may contain a number of variables:

- `othersites` (no value)
- `search` (string)
- `page` (integer)
- `quickaccess` (no value)

Each variable leads to a different type of request (except `page` which supplements `search`).

Variable `search` contains text that is a subject of the database query. `page` is used to determine starting record in the search results.

Examples:

```
domain.com/script/?othersites  
domain.com/script/?search=bas  
domain.com/script/?search=bas&page=2  
domain.com/script/?quickaccess
```

Output Variables

Syntax notes

All returned variables are local (their names start with an underscore). Variables are separated by semi-colon. Last item is value-only. Don't put the separator at the end. Example:

```
_version=0.5;_pages=0;_results=0;_files=[];true
```

Double the amount of quotation marks in any string (`\"` to `\\"\`).

To counter OFP matrix allocation limit I recommend to use addition for arrays with undetermined number of items. Example:

```
_array = ["item1"] + ["item2"];      - follow  
_array = ["item1", "item2"];        - avoid
```

List of all variables:

- `_version` (float)
- `_errorMsg` (string)
- `_url` (array with strings)
- `_pages` (integer)
- `_results` (integer)
- `_files` (array with arrays with strings)
- operation success (bool)

Variables listed by the type of request:

- Other Sites
 - `_version`, `_errorMsg`, `_url`, operation success
- Search
 - `_version`, `_errorMsg`, `_pages`, `_results`, `_files`, op. success
- Quick Access
 - `_version`, `_errorMsg`, `_files`, operation success

1. `_version`

Indicates version number of the data standard. Always pass it regardless of the request type. It will allow for the script to maintain backwards compatibility if the standard changes.

Example:

```
_version=0.5;
```

2. `_errorMsg`

Optional variable storing text that describes the error that has occurred. Message will be displayed in the game. Use `\n` to break long lines or the text will go off the screen.

```
_errorMsg="Invalid input";
```

3. `_url`

List of web addresses to other databases. If a single item is longer than 122 characters then turn it instead into a sub-array with the text split into 122-char long parts.

```
_url=["...", ["...", "..."], "..."];
```

4. `_pages`

Total number of pages in the query result. This variable is optional. Pass it only when input variable `page` was not specified.

```
_pages=6;
```

5. `_results`

Total number of results that a search query can find in the entire database. This variable is optional . Pass it only when input variable `page` was not specified.

```
_results=12;
```

6. `_files`

Contains information about addons possible to download. Sub-array structure is as follows:

- Addon title (string)
- Download size (string)
- Installation script (string)

Title is the text that will be displayed in the listbox in the game.

Size is a string indicating number of B/KB/MB that user will download. It's optional – pass empty string if it's unavailable.

Installation script contains instructions on how to apply given addon to the game. In its absence put URL to file download instead. If the script (or link) is longer than 122 characters then replace it with a sub-array with the text split into 122-char long parts.

```
_files=[["Addon", "2 MB", "http://www.domain.com/addon.rar"]];
```

7. Operation Success

Put boolean value at the end of the output data string. `true` indicates that no issues have occurred and the OFP script will continue to process the request output. In the case of `false` it will cancel current operation and display error message.

```
;true
```

Installation Script

I'll describe it in the near future.