UMCD design document - DRAFT

Mark de Wever Pierre Talbot

January 10, 2018

Contents

Abstract

The campaign server had little love over the years and its code shows its age. Several new features have been wanted over the years: better dependency tracking, better translation integration.

Over the years several initiatives have started, none of these were finished. Early 2012 I restarted this project again this document describes the design and the communication protocol.

Chapter 1

Introduction

This paper describes the design and communication protocol of the umcd, which stands for *U*ser *M*ade *C*ontent *D*aemon. The name was first used by Shadow master.

The server has several new features planned:

Dependency tracking

Translated messages

Wescamp intergration The translations are separate. Automatic updates with Wescamp.

Asio The network handling is switched to Boost asio.

Chapter 2

Communication protocol

2.1 Document Convention

The following tokens will be used to further describe an item.

? The current item is optional (request) or should not appear every time under specified conditions (reply).

***** The current item appears 0, 1 or multiple times.

+ The current item appears 1 or multiple times.

The description of an item can take different formats:

<value> A text between "<>" is replaced with the appropriate value in the current context.

{value} A text between "{}" is optional.

2.2 Server configuration

On the server there are several configuration files:

- The main configuration file, umcd.cfg, this file is stored in the data directory, which is a parameter for the program. This file is further discussed in section 2.2.2.
- In this data directory there is a subdirectory per addon. These directories have the following files:

  umc.cfg The config file for the UMC, this file is based on the client's pbl file. This file is further discussed in section 2.2.3.

  umc.gz The gzipped UMC.

  umc-*lang*.cfg The configuration file for the translation of the UMC. The *lang* is replace with the locale of the translation. This file is further discussed in section 2.2.4.

  umc-*lang*.gz The gzipped mo-file for the translation.

2.2.1 Compressed files

If the compression ratio is bad, the file could not be zipped. But, to facilitate sending data over the network, the files sent will be gather into a tarball (.tar).

2.2.2 umcd.cfg

This file contains the settings of the UMCD.

```
port = PORT
threads = THREADS
dir = DIR
```

The keys have the following meaning:

PORT The TCP/IP port to listen to for incoming requests.

THREADS The number of working threads to start. A value of 0 means start as many threads as there are detected hardware cores or hyperthreading units.

DIR The directory where the add-ons will be stored.

The values are not encapsulated into a global markup because we parse the options with *Boost.Program_options*.

2.2.3 umc.cfg

This file contains the configuration of the umc on the server. The combination of this file together with the various umc-*lang*.cfg files form the master information regarding an umc. All other files that specify the umc or its wire-protocol get their information from these files. Therefore other sections will refer to this section.

The file is specified like:

```
[umc]
  [info]
    id = ID
    name = NAME
    description = DESCRIPTION
    type = TYPE
    version = VERSION
    authors = AUTHORS
    email = EMAIL
    password = PASSWORD
  [/info]
  [language]
    native_language = NATIVE_LANGUAGE
    translatable = TRANSLATABLE
  [/language]
  [dependencies]?
    [dependency]+
      id = ID
```

```
      version? = VERSION_MASK
    [/dependency]
  [/dependencies]
  [state]
    uploader_ip_address = UPLOADER_IP_ADDRESS
    timestamp = TIMESTAMP
    downloads = DOWNLOADS
    uploads = UPLOADS
  [/state]
[/umc]
```

The keys have the following meaning:

[**info**] The main informations provided by the UMC-creator.

ID The unique id of the content. When uploading this value determines whether an UMC is the same UMC.

NAME The name of the content. This string is not unique. This allows two versions of the UMC on the server. This can be used to allow a new beta next to a stable, as long as the ID differs.

DESCRIPTION The description of the UMC.

TYPE The type of the addon. At the moment the following values are allowed:

CAMPAIGN_SP A single player campaign.

CAMPAIGN_MP A multi-player player campaign.

ERA A multi-player era.

MAP A multi-player map.

MAP_PACK A multi-player map pack.

RPG_SP A RPG like single player game. (This also includes builder games in the style of settlers.)

RPG_MP A RPG like multi-player game. (This also includes builder games in the style of settlers.)

MEDIA Miscellaneous resources for UMC authors/users, for example, music packs, packages of general-purpose WML, etc.

OTHER Everything that doesn't fit in one of the above.

VERSION A string determining the version. The recommended string format like: 'x.y.z' or 'x.y.z-q'. The version is parsed as version number and comparisons are done considering all the number and letter $a < z < A < Z < 0 < 9 < Empty$, all other characters are parsed as separator. For example, "aZ=4" is greater than "aZ", and "=" is the separator character.

AUTHORS A colon separated list of authors of the umc. When there is only one author no colon is required.

EMAIL The email address of the primary author or a mailinglist. When there are problems with the umc the Wesnoth team can use this address to contact the authors, so make sure it's valid and doesn't need registration. The address is not used in other ways nor shared with third-parties.

PASSWORD The password required to upload a new version of an UMC or remove an UMC from the server. See also section 2.6.1 and section 2.6.2.

[**language**] The items related to the language files of the UMC.

NATIVE_LANGUAGE The native language id of the UMC, and the language in which are the name and the description.

TRANSLATABLE This flag causes the UMC to be synchronized with Wescamp. It's advisable to only set the flags when the strings are somewhat stable. This flag can only be set if American English is available.

[**dependencies**] Lists the dependencies of the UMC. This allows the client to download all UMC required to use this UMC. The fields mean:

ID The ID of the UMC to depend upon.

VERSION_MASK This mask is used to validate a version used. The format is:

=**VERSION** The version number of the dependency exactly match the VERSION.

>=**VERSION** The version number of the dependency must be greater or equal to VERSION.

>**VERSION** The version number of the dependency must be greater than VERSION.

<**VERSION** The version number of the dependency must be less than VERSION

<=**VERSION** The version number of the dependency must be less than or equal to VERSION

Several values can be used in a mask. They are separated by a space. The masks are and'ed and the user is responsible for setting a sane value; e.g. '<1.0.0 >1.0.0' can't match a version.

If this field is omitted, all versions are allowed.

[**state**] These fields are auto-generated by the server. It's the current state of the UMC.

UPLOADER_IP_ADDRESS The ip address of the last uploader of the umc.

TIMESTAMP The timestamp of the last upload.

DOWNLOADS The number of times the UMC is downloaded.

UPLOADS The number of time a new version of the UMC is uploaded.

2.2.4 umc-*lang*.cfg

.

The configuration file for the translation of the UMC. The *lang* is replace with the locale of the translation.

```
[language]
  id = LANGUAGE
  [translated_umc_strings]
    name = NAME
    description = DESCRIPTION
  [/translated_umc_strings]
  [po_file_details]
    size = SIZE
    timestamp = TIMESTAMP
    translated = TRANSLATED
    fuzzy = FUZZY
    untranslated = UNTRANSLATED
  [/po_file_details]
[/language]
```

ID The id of the language the translation for.

[translated_umc_strings]

The translation of the name and description, in the language specified by "ID".

NAME See section 2.2.3

DESCRIPTION See section 2.2.3

[po_file_details]

SIZE The size of the po-file.

TIMESTAMP The timestamp the po-file was last updated.

TRANSLATED The number of translated strings. Must be >0.

FUZZY The number of strings that are fuzzy.

UNTRANSLATED The number of strings that are not translated.

2.3 UMC configuration

The pbl file contains the information regarding the addon. Its contents are stored on the system of the addon developer and on the server. The file contains the following data:

```
[umc_configuration]
  [info]
    id = ID
    name = NAME
    type = TYPE
    description = DESCRIPTION
    version = VERSION
    authors = AUTHORS
    email = EMAIL
    password = PASSWORD
  [/info]
```

```
[language]
  native_language = NATIVE_LANGUAGE
  translatable = TRANSLATABLE
[/language]
[dependencies]?
  [dependency]+
    id = ID
    version? = VERSION_MASK
  [/dependency]
[/dependencies]
[image]
  icon_name = ICON_NAME
[/image]
[/umc_configuration]
```

The keys have the following meaning:

[**info**] See section 2.2.3.

[**language**] See section 2.2.3.

[**dependencies**] See section 2.2.3.

[**image**]

ICON_NAME The name of the icon that will be displayed in the umc list.

2.4 Wire

2.4.1 Design rational

- The ID is a technical ID and not the couple campaign name + version which is unique too. So we can change the campaign name without breaking dependencies.
- Why do we send the size before each request packet? If we don't, we must know when to stop reading the request. In the Boost.Asio HTTP server sample, the author uses a parser that retains the state of the parsing whenever the parsing ends. If we do the same, we duplicate the current WML parser. We also should write another complete parser with a validator (if we don't want to parse twice). That's why we prefer to read the entire request first, and then parse it when it completes.

2.4.2 Size packet header

Each packet (packet sent from the client to the server) begin with an integer of 4 bytes, the endianess is handle via the pair htonl and ntohl. It contains the size of the metadata (not the binary data).

2.4.3 Special packets

Special packets are sent in reply and can appears instead an expected packet or in addition of this packet.

Error packet

The server could not understand the request, or a value in a field of the request is wrong or missing. In this case an error packet will be sent and it will always take the following form:

```
[error]
  msg = MSG
[/error]
```

MSG Contains a message describing the error.

The common values of MSG are:

1. The value <value> of the field <field_name> is wrong {because <reason>}.
2. The value <value> of the field <field_name> is missing.
3. The field <field_name> is unknown.

Warning packet

The server correctly understand the request but the request is not exactly correct for the reasons describe into *msg*.

```
[warning]
  msg = MSG
[/warning]
```

MSG Contains a message describing the warning.

2.4.4 Request campaign list

Request

```
[request_umc_list]
  favlang = LANGUAGE
[/request_umc_list]
```

The field *favlang* is the favorite language in which the add-ons (title and description for example) must be send. If the content is not fully (or not at all) translated, the UMC description is downloaded in its native language, and the rest is left to the engine *gettext*.

Reply

An error packet can be sent instead of the umc list packet. No special value appears in this packet. See section 2.4.3. If the list is empty, no error packet is sent, the list will just appears with no [umc] tag.

```
[umc_list]
  [umc]*
    [info]
      id = ID
      type = TYPE
      version = VERSION
      size = SIZE
      authors = AUTHORS
    [/info]
    [icon]
      size = SIZE
      extension = EXTENSION
    [/icon]
    [state]
      timestamp = TIMESTAMP
      downloads = DOWNLOADS
      uploads = UPLOADS
    [/state]
    [dependencies]?
      [dependency]+
        id = ID
        version? = VERSION_MASK
      [/dependency]
    [/dependencies]
    [languages]
      [language] +
        id = LANGUAGE
        [translated_umc_strings]
          name = NAME
          description = DESCRIPTION
        [/translated_umc_strings]
        [po_file_details]
          size = SIZE
          timestamp = TIMESTAMP
          translated = TRANSLATED
          fuzzy = FUZZY
          untranslated = UNTRANSLATED
        [/po_file_details]
      [/language]
    [/languages]
  [/umc]
[/umc_list]
```

The fields in [umc] have the following meaning:

[info] See section 2.2.3

SIZE The size of the UMC without translation files (.po).

[icon]

SIZE The size of the icon.

EXTENSION The extension of the file, or in other word, the file format.

[**state**] See section 2.2.3.

[**dependencies**] Lists the dependencies of the UMC, See section 2.2.3.

[**languages**] Contains the description of the add-on in the favorite language requested.

If the translation into the favorite language is not available or incomplete, American English is sent. If American English is not available or complete, the language depends on the native language of this UMC. We let the engine *gettext* merge the different file considering this order of preference: $favorite > AmericanEnglish > native$. So, maximum 3 languages are sent.

[**language**] See section 2.2.4.

The icons are sent in the order of appearance in the [umc_list] packet, the size of the icon is specified in the [icon] tag.

2.4.5 Request umc download

This requests to download an UMC.

Request

```
[request_umc_download]
  id = ID
  language = LANGUAGE
[/request_umc_download]
```

ID The id of the UMC to download.

LANGUAGE The id of the requested language to download.

Reply

An error packet can be sent instead for the common reasons. See section 2.4.3.

```
[warning_packet]?
  msg = MSG
[/warning_packet]
[umc_download]
  id = ID
  size = SIZE
  [languages]
    [language]+
      id = LANGUAGE
      size = SIZE
    [/language]
  [/languages]
[/umc_download]
```

[**warning_packet**] The warning packet can be sent with the following possible values:

1. The language requested is incomplete.
2. The language requested is not available.

See also section 2.4.3.

[**umc_download**]

ID The id of the UMC.

SIZE The size of the UMC without translation files (.po).

[**language**]

ID The id of the language.

SIZE The size of the translation file (.po).

The files send are in the order of appearance in the reply, so first the ID.gz and then the ID-LANGUAGE.gz.

ID.gz The gzipped campaign files. The file is packed in a single config file to be extracted to the local file-system.

ID-LANGUAGE.gz A gzipped mo-file for a language.

2.4.6 Request UMC change password

This request changes the password.

Request

```
[request_change_password]
  id = ID
  current_password = CURRENT_PASSWORD
  new_password = NEW_PASSWORD
[/request_change_password]
```

ID The id of the UMC to change the password.

CURRENT_PASSWORD The current password of the UMC.

NEW_PASSWORD The new password of the UMC.

Reply

```
[umc_change_password]
  id = ID
  [error_packet] ?
    msg = MSG
  [/error_packet]
  [warning_packet] ?
    msg = MSG
  [/warning_packet]
[/umc_change_password]
```

id The id of the UMC to change the password.

[**error_packet**] If this packet appears, then the password could not been changed for the reasons described into this packet. The possible special reasons are:

1. The current password is wrong.
2. The time between two requests is not respected.
3. The new password is too weak.

See also section 2.4.3 for a description.

[**warning_packet**] If this packet appears, the password has been changed but a warning is sent to the user. The warning can take these values:

1. The password is weak, you should use a stronger password.

See also section 2.4.3.

If only the ID appears in the packet, the password has been successfully changed.

2.4.7 Request a new password

Request

Request a new password because the user forgot it. A new auto-generated password is sent by email.

```
[umc_forgot_password]
  id = ID
[/umc_forgot_password]
```

ID The ID of the UMC that the password has been forgotten.

Reply

The reply can be an error packet for the common reasons. See section 2.4.3.

```
[umc_forgot_password]
  id = ID
  mail = MAIL
[/umc_forgot_password]
```

ID The ID of the UMC of the forgotten password.

MAIL The address mail where the password has been sent.

2.4.8 Request umc upload

Request an upload of a new or an updated version of a UMC. A new version have no ID in the .pbl file while an updated version have an ID.

Request

```
[request_umc_upload]
  size = SIZE
  format = FORMAT
  [umc_configuration]
    [info]
      id = ID
      name = NAME
      type = TYPE
      description = DESCRIPTION
      version = VERSION
      authors = AUTHORS
      email = EMAIL
      password = PASSWORD
    [/info]
    [language]
      native_language = NATIVE_LANGUAGE
      translatable = TRANSLATABLE
    [/language]
    [dependencies]?
      [dependency]+
        id = ID
        version? = VERSION_MASK
      [/dependency]
    [/dependencies]
    [image]
      icon_name = ICON_NAME
    [/image]
  [/umc_configuration]
[/request_umc_upload]
```

SIZE The size the complete UMC that is sent next.

FORMAT The format of the UMC, this field can take these values: gz or bz2.

UMC_CONFIGURATION This part is commonly known as the "pbl file". See section 2.3 for details.

The server will decompressed or unpack this file and will look recursively into the repository to check if all the required files are present in respect of the pattern described in their respective section.

The pack must contains, at least:

1. A .pbl file.

But can also contains:

1. An icon file, the name of the icon file is specified in the .pbl file, so the server will look recursively into the archive.

Reply

An error packet can be sent for the common reasons, see section 2.4.3. But also for specific reasons:

1. The translatable flag in the .pbl has been set, but no American English translation can be found.
2. The file <file_type> is malformed: <reason>.
3. A conflict has been detected: <reason>.
4. Wrong password.

If no error packet appears, in case of a new version, the ID field in the .pbl is add and the following response is sent:

```
[request_umc_upload]
  [umc]
    [info]
      id = ID
      type = TYPE
      version = VERSION
      size = SIZE
      authors = AUTHORS
    [/info]
    [icon]
      size = SIZE
      extension = EXTENSION
    [/icon]
    [state]
      timestamp = TIMESTAMP
      downloads = DOWNLOADS
      uploads = UPLOADS
    [/state]
    [dependencies]?
      [dependency]+
        id = ID
        version? = VERSION_MASK
      [/dependency]
    [/dependencies]
    [languages]
      [language] +
        id = LANGUAGE
        [translated_umc_strings]
          name = NAME
          description = DESCRIPTION
        [/translated_umc_strings]
        [po_file_details]
          size = SIZE
          timestamp = TIMESTAMP
          translated = TRANSLATED
```

```
            fuzzy = FUZZY
            untranslated = UNTRANSLATED
          [/po_file_details]
        [/language]
      [/languages]
    [/umc]
[/request_umc_upload]
```

[umc] In case of a new version, the .pbl file is updated with the ID. See section 2.4.4.

2.4.9 Request umc delete

Request

```
[request_umc_delete]
  id = ID
  password = PASSWORD
[/request_umc_delete]
```

ID The ID of the UMC we want to delete.

PASSWORD The password of the UMC.

Reply

An error packet can be sent for the common reasons (see section 2.4.3) but also because:

1. The password is wrong.

Otherwise, a packet with no field is sent.

```
[request_umc_delete]
[/request_umc_delete]
```

2.4.10 Request license

Request

```
[request_license]
  language = LANGUAGE
[/request_license]
```

LANGUAGE The language we want the license.

Reply

The license is sent in the language asked or, if not available, in American English.

```
[request_license]
  [warning_packet] ?
    msg = MSG
  [/warning_packet]
  [translated_text] ?
    [disclaimer]
      ENGLISH_DISCLAIMER
    [/disclaimer]
    [translated_license]
      TRANSLATED_LICENSE
    [/translated_license]
  [/translated_text]
  [english_text]
    ENGLISH_LICENSE
  [/english_text]
[/request_license]
```

Note : The text is not a key value as it can be on multiple lines. We enclosed text in specific markup.

[**warning_packet**] Can be sent if the language of the license is not the language expected.

[**translated_text**] is sent only if there is no warning message before.

ENGLISH_DISCLAIMER A english disclaimer stating that the translated license has no judicial value.

TRANSLATED_LICENSE The translated license in the language requested.

TEXT The text of the license in english.

2.5 Text encoding

The text encoding is important because the server can open the file to verify their validity or to read information in the .pbl file.

The strings could be in something else than American English and for this reason, the encoding must be UTF-8.

2.6 Security

2.6.1 Password storage

The passwords will be stored with a strong hash function such as SHA-512, a salt only knowable by the server is append to this hash.

For example, the library OpenSSL (http://www.openssl.org/docs/crypto/sha.html) provides the algorithm, but an encapsulation in a C++ class would be required.

2.6.2 Password transmission

The password will be transmit via Transport Layer Security (TLS) protocol.

For example, the Boost.Asio library provides tools to transmits data over a ssl layer.