

FreeTV+ Addon Manual

Addon Interface ver 1.01

2011/08/04



Table of Contents

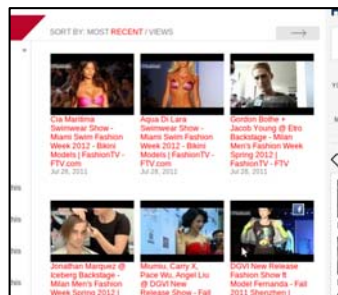
1. FreeTV+
2. Addable Movie Streaming Website
3. Working circumstance for Addon
4. Addon interface functions and examples
5. Configure and Build
6. Appendix
7. External Links

1. FreeTV+

Free TV+ is a unique feature that enables you to play & enjoy FREE web media contents like Youtube and addable other free contents from various internet websites such as documentary and more.

2. Addable Movie Streaming Website

Mostly movie streaming website has a structure like Pic 1 or grouping the watchable movie contents(feed) or available to watch movie contents like Pic 2. In this document it will called feed page such as Pic 1 and play page such as Pic 2.



Pic 1. Feed Page



Pic 2. Play Page

Free TV+ OSD is composed of the streaming URL needed to play contents by analyzing movie streaming webpage or RSS feed based on XML standard, Thumbnail URL needed to UI composition and acquiring other contents information. Below condition is required to produce FreeTV+ service.

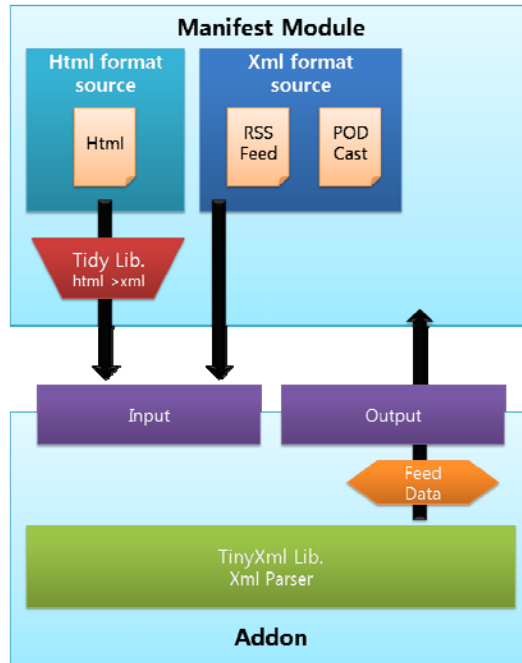
- Possible to have movie streaming URL in web page(HTML) source.
- Possible to have streaming URL by RSS Feed or Podcast
- Possible to use RTMP protocol streaming or can download HTTP protocol
- knowability of contents information(title, play time, thumbnail image..etc)

If the movie streaming service can meet above all kind of conditions, it can be added into the FreeTV+ as addon type.

Please refer to the related site : www.ftv.com/videos (webpage analysis way),
www.earth-touch.com (RSS Feed or Podcast way)

3. Working circumstance for Addon

It is possible to check related point between Manifest module and addon in the Pic 3. Manifest module downloads the webpage requested by Addon and forwards it to Addon. Addon extracts the information from the forwarded webpage and forwards the information to Manifest module. At this time(ei HTML page can not parse as XML parser), Addon can request translating the webpage to XML by using Tidy of Manifest module.



Pic 3. Working of Manifest and Addon

4. Addon interface function and examples

Manifest is required totally 7 interface function to addon and 2 interface function (freetvaddon_get_next_page_url, freetvaddon_get_playinfo)of them is fine with not following service structure.

■ ***int freetvaddon_get_video_info(TiXMLElement *a, ParsingInfoCallback b)***

Use

To parse every type of feed information (titel, thumbnail image URL, video URL, etc..) In the feed page included in video contents list.

Return

To finalize circulation in the recursive function and user can spot specific value

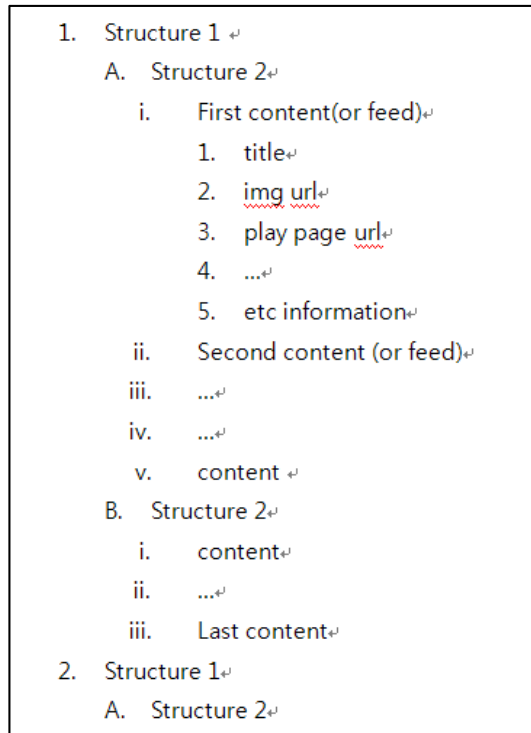
Parameter

TiXMLElement *a: memory of XML's initial element(Node) in feed

ParsingInfoCallback b: callback function pointer to forward information found in XML to Manifest module.

Explanation

This function is searching the page included in feed information and forwarding it to Manifest module. Mostly streaming site has a structure as Pic 4 so the point is that grasp structure of HTML(or XML) first and know how the contents are distributed of.



Pic 4. Structure of Streaming site

To use callback function received from parameter, forwarding the total information of 1 feed to Manifest module. As see the Pic 4, this function has to stop after parsing all of the feed in HTML document not just return only searching one feed information since most streaming site has the contents information paratactically and call the function recursively. Its structure is S_ParsingVideoData and defined as freetvaddon.h

Examples

Pic 6. is the part of the Podcast source[1] supported by www.earth-touch.com and new contents in this site is creating every <item> element. Pic 5. is represent only one feed information in between <item> element and next <item> element. You can find the contents title as value of <title> element in the Podcast XML code and thumbnail image URL as value of <description> element.


```

int freetvaddon_get_video_info(TIXmlElement *rootElem, ParsingInfoCallback parsingCallback)
{
    TIXmlElement *pSubElem;
    char *pszNode = NULL;
    char *pszText = NULL;
    string tempStr;
    S_ParsingVideoData videoData;

    memset(&videoData, 0, sizeof(videoData));
    CHECK_NULL_INT(pSubElem = rootElem->FirstChildElement());
    while (pSubElem)
    {
        CHECK_NULL_INT(pszNode = (char *)pSubElem->Value());
        if (strcmp(pszNode, "item") == 0)
        {
            CHECK_NULL_INT(pszText = (char *)pSubElem->FirstChildElement()->Value());
            if (strcmp(pszText, "title") == 0)
            {
                pSubElem = pSubElem->FirstChildElement("title");
                CHECK_NULL_INT(videoData.title = (char *)pSubElem->GetText());
                CHECK_NULL_INT(pSubElem = pSubElem->NextSiblingElement("description"));

                tempStr = pSubElem->GetText();
                tempStr = tempStr.substr(tempStr.find("http"), tempStr.find(".jpg") - tempStr.find("http") + 3);
                CHECK_NULL_INT(videoData.thumbUrl = (char *)tempStr.c_str());
                CHECK_NULL_INT(pSubElem = pSubElem->NextSiblingElement());
                CHECK_NULL_INT(pSubElem = pSubElem->NextSiblingElement("itunes:summary"));
                CHECK_NULL_INT(videoData.desc = (char *)pSubElem->GetText());
                CHECK_NULL_INT(pSubElem = pSubElem->NextSiblingElement("guid"));
                CHECK_NULL_INT(videoData.pageUrl = (char *)pSubElem->GetText());
                CHECK_NULL_INT(pSubElem = pSubElem->NextSiblingElement("media:content"));
                CHECK_NULL_INT(videoData.playUrl = (char *)pSubElem->Attribute("url"));

                parsingCallback(&videoData);
                CHECK_NULL_INT(pSubElem = pSubElem->Parent()->ToElement());
            }
        }
        } ? end if strcmp(pszNode, "item"... ?
    freetvaddon_get_video_info(pSubElem, parsingCallback);
    pSubElem = pSubElem->NextSiblingElement();
} ? end while pSubElem ?
return 0;
} ? end freetvaddon_get_video_info ?

```

Pic 6. Example Function for earth-touch site

■ void freetvaddon_get_service_info(S_ServiceInitInfo *a)

Use

Forwarding information needed to run addon to Manifest module

Return

Not applicable

Parameter

S_ServiceInitInfo *a: It is the structure including information needed to run Free TV+ addon and defined as freetvaddon.h

Explanation

This function is forwarding the information needed to run Free TV+ addon to Manifest module. Put in the current supported service quality.

Example

Pic 7. is representing example of freetvaddon_get_service_info function.

S_ServiceInitInfo structure is composite of 8 items and refer to the below table 1.

```

void freetvaddon_get_service_Info(S_ServiceInitInfo *serviceInitInfo)
{
    snprintf(serviceInitInfo->serviceName, TITLE_LENGTH, "Metacafe");
    serviceInitInfo->isCookieNeeded = false;
    snprintf(serviceInitInfo->cookiePost, TITLE_LENGTH, "empty");
    serviceInitInfo->isplayUrlNeeded = true;
    serviceInitInfo->istidyNeeded = true;
    snprintf(serviceInitInfo->libFilename, TITLE_LENGTH, "metacafe.%2f.addon", FREETVADDON_VER);
    snprintf(serviceInitInfo->rootUrl, TITLE_LENGTH, "www.metacafe.com");
    serviceInitInfo->isLockNeeded = false;
}

```

Pic 7. freetvaddon_get_service_info Example

Table 1. member of S_ServiceInitInfo

Type	Member	Description
char[]	serviceName	Name of addon service
int	isCookieNeeded	If Cookie is needed to access URL : True Opposite : False
char[]	cookiePost	Post option to get cookie
int	isplayURLNeeded	If play page is needed : True Opposite : False
int	istidyNeeded	HTML page : True , RSS Feed or Podcast : False
char[]	libFilename	Addon file name. Regulation of addon file name : [addonname][version][addon] ex)metacafe.1.00.addon
char[]	rootURL	Unchanged service webpage address ex) www.youtube.com
int	isLockNeeded	parental lock on : True , Opposite : False

■ **int freetvaddon_get_category_count(void)**

Use

Forwarding category numbers having service

Return

Service category number.

Parameter

Notapplicable

Explanation

This function can return the number of service category.

Example

Return value would be fine with invariable number


```
int freetvaddon_get_category_count()
{
    return CATEGORYNUM;
}
```

Pic 8. freetvaddon_get_category_count Example

■ **double freetvaddon_ver(void)**

Use

Forwarding addon interface version.

Return

A real number value in freetvaddon.h file defined as FREETVADDON_VER.

Parameter

Not applicable.

Explanation

This function can return the interface version to Manifest module. So, if Manifest module is lower than wanted addon interface value addon will not loaded. It returns FREETVADDON_VER value defined Freetvaddon.h header.

Example

This function is not necessary to adjust and use the example as is.

```
double freetvaddon_ver(void)
{
    return FREETVADDON_VER;
}
```

Pic 9. freetvaddon_ver example

■ **void freetvaddon_get_category_list(S_CategoryInfo *a)**

Use

Forwarding title of category and URL

Return

Not applicable

Parameter

S_CategoryInfo *a: structure pointer saving category information.

Explanation

Provide category name and category URL

Example

Find the example Pic 10. Save category title will be shown FreeTV+ OSD in the name member and write whole address of available feed page from the category
If the feed page is consists of many pages write the address in the first page and sending remaining pages by using freetvaddon_get_next_page_url.

```

void fretvaddon_get_category_list(S.CategoryInfo *categoryInfo)
{
    categoryInfo->category[0].name = "Wildlife Moments (SD)";
    categoryInfo->category[1].name = "Weekly Highlight (SD)";
    categoryInfo->category[2].name = "Morem Lions Channel (SD)";
    categoryInfo->category[3].name = "Wild Oceans (SD)";
    categoryInfo->category[4].name = "Survival Stars (SD)";
    categoryInfo->category[5].name = "Planet Philharmonic (SD)";
    categoryInfo->category[6].name = "Wildlife Moments (HD)";
    categoryInfo->category[7].name = "Weekly Highlight (HD)";
    categoryInfo->category[8].name = "Morem Lions Channel (HD)";
    categoryInfo->category[9].name = "Wild Oceans (HD)";
    categoryInfo->category[10].name = "Survival Stars (HD)";
    categoryInfo->category[11].name = "Planet Philharmonic (HD)";

    categoryInfo->category[0].url = "http://feeds2.feedburner.com/earth-touch_featured_480p_commentary";
    categoryInfo->category[1].url = "http://feeds2.feedburner.com/earth-touch_podcast_480p";
    categoryInfo->category[2].url = "http://feeds2.feedburner.com/moremi_podcast_ipod";
    categoryInfo->category[3].url = "http://feeds2.feedburner.com/WeeklyMarinePodcast-ipod";
    categoryInfo->category[4].url = "http://feeds2.feedburner.com/kids-ipod";
    categoryInfo->category[5].url = "http://feeds2.feedburner.com/planetphilharmonic-ipod";
    categoryInfo->category[6].url = "http://feeds2.feedburner.com/earth-touch_featured_720p_commentary";
    categoryInfo->category[7].url = "http://feeds2.feedburner.com/earth-touch_podcast_720p";
    categoryInfo->category[8].url = "http://feeds2.feedburner.com/moremi_podcast_720";
    categoryInfo->category[9].url = "http://feeds2.feedburner.com/WeeklyMarinePodcast-hd";
    categoryInfo->category[10].url = "http://feeds2.feedburner.com/kids-hd";
    categoryInfo->category[11].url = "http://feeds2.feedburner.com/planetphilharmonic-hd";
}
} end fretvaddon_get_category_list ?

```

Pic 10. fretvaddon_get_category_list Example

■ ***int fretvaddon_get_next_page_URL(int a, int b, char **c)***

Use

To forward next page address in case the feed page more than one in the same category

Return

Next page is not existed : "0"

Opposite case : "1"

Parameter

int a: current index of category wanting next page

int b: wanting page index. First page has to have starting "1" and if the function has been called parameter will be given "2" (because next page is wanted)

char **c: next feed page URL to be returned

Explanation

In case having many feed pages in a category, this function is using for next feed page address.

Example

This function has current index, wanted page index and memory address to be returned next page URL as a parameter. So, if the URL page is changed in every category it should know the URL will be changed so that it can return the category's page address requested by Manifest. Returning URL is returns every information in the fretvaddon_get_service_info except rootURL. You can see the example in the Pic 11.

```

int freetvaddon_get_next_page_url(int categoryIndex, int curPageNum, char **nextPageUrl)
{
    char restUrl[FREETV_MAX_URL_LENGTH];

    switch (categoryIndex)
    {
        case 0:
            sprintf(restUrl, FREETV_MAX_URL_LENGTH, "?cb,a1364145,1,f,-1/pb,a1364142,%d,f,", curPageNum);
            break;
        case 1:
            sprintf(restUrl, FREETV_MAX_URL_LENGTH, "?cb,a1364145,1,f,-1/pb,a1364143,%d,f,", curPageNum);
            break;
        case 2:
            sprintf(restUrl, FREETV_MAX_URL_LENGTH, "?cb,a1364145,1,f,-1/pb,a1527537,%d,f,", curPageNum);
            break;
        case 3:
            sprintf(restUrl, FREETV_MAX_URL_LENGTH, "?cb,a1364145,1,f,-1/pb,a1364144,%d,f,", curPageNum);
            break;
        case 4:
            sprintf(restUrl, FREETV_MAX_URL_LENGTH, "/c/96251/barn?cb,a1364151,1,f,-1/pb,a1364150,%d,f,-1", curPageNum);
            break;
        case 5:
            sprintf(restUrl, FREETV_MAX_URL_LENGTH, "/c/96257/film_och_drama?cb,a1364153,1,f,-1/pb,a1364152,%d,f,-1", curPageNum);
            break;
        case 6:
            sprintf(restUrl, FREETV_MAX_URL_LENGTH, "/c/96256/kultur_och_noje?cb,a1364155,1,f,-1/pb,a1364154,%d,f,-1", curPageNum);
            break;
        case 7:
            sprintf(restUrl, FREETV_MAX_URL_LENGTH, "/c/96254/samhalle_och_fakta?cb,a1364157,1,f,-1/pb,a1364156,%d,f,-1", curPageNum);
            break;
        case 8:
            sprintf(restUrl, FREETV_MAX_URL_LENGTH, "/c/96253/sport?cb,a1364159,1,f,-1/sb,a1364158,%d,f,-1", curPageNum);
            break;
        default:
            *nextPageUrl = NULL;
            return 0;
    } ? end switch categoryIndex ?

    *nextPageUrl = strdup(restUrl);
    return 1;
} ? end freetvaddon_get_next_page_url ?

```

Pic 11. freetvaddon_get_next_page_URL Example

int freetvaddon_get_playinfo(TiXMLElement *a, const char **b)

Use

It is parsing the contents URL in the play page in such cases the feed page doesn't have the streaming URL to play contents

Return

Necessary URL is found "1"
Necessary URL is not found "0"

Parameter

TiXMLElement *a: initial element(or node) of play page
const char **b: memory address where the contents URL will be returned

Explanation

Like the freetvaddon_get_video_info function, this function can parse the information by searching in the given page. This function is calling itself recursively and searching the page continually but the difference point with freetvaddon_get_video_info function is that it stops the searching if it finds a streaming address (not searching all the page)

Example

Pic 12. is play page of Metcafe site. In this case, it does not have video streaming URL in the play page. So, catch the regulation that is available video streaming URL by extracting video streaming URL for Metcafe play page from the keepvid.com. Pic 13 is the source code of it.

```

<!--[if !IE]>-->
<object type="application/x-shockwave-flash"
  data="http://s.mcstatic.com/Flash/vp/mc/Portal_3.4.0.1.swf"
  width="100%" height="100%" id="fpObj">
  <param name="quality" value="high" />
  <param name="wmode" value="transparent" />
  <param name="allowfullscreen" value="true" />
  <param name="allowscriptaccess" value="always" />
  <param id="flashVars" name="flashvars"
    value="itemID=6842779&duration=76&LEID=44&staticURL=http%3A%2F%2Fs.mcstatic.com&isInd
  />
</object>
<!--![endif]-->

```

Pic 121. Play Page HTML Example

```

int freetvaddon_get_playinfo(TiXmlElement *rootElem, const char **streamUrl)
{
#define VIDEO_STREAM_URL "http://v.mccont.com/ItemFiles/%5BFrom%20www.metacafe.com%5D%20s?_gda_=%s"
#define FILENAME_LEN 64
#define KEY_LEN 64

  TiXmlElement *pSubElem;
  char *pszText;
  char filename[FILENAME_LEN];
  char key[KEY_LEN];

  pSubElem = rootElem->FirstChildElement();
  if (!pSubElem)
    return 0;

  while (pSubElem)
  {
    pszText = (char *)pSubElem->Value();
    if (pszText && strcmp(pszText, "param") == 0)
    {
      pszText = (char *)pSubElem->Attribute("id");
      if (pszText && strcmp(pszText, "flashVars") == 0)
      {
        pszText = (char *)pSubElem->Attribute("value");
        if (pszText
          && get_string(pszText, filename, FILENAME_LEN, "www.metacafe.com%255D%2520", "%22%2C") == 0
          && get_string(pszText, key, KEY_LEN, "key%22%3A%22", "%22%7D") == 0)
        {
          *streamUrl = (char *)malloc(strlen(VIDEO_STREAM_URL) + strlen(filename) + strlen(key));
          sprintf((char *)*streamUrl, VIDEO_STREAM_URL, filename, key);
        }
        else
          *streamUrl = NULL;

        return 1;
      }
    }

    if (freetvaddon_get_playinfo(pSubElem, streamUrl) == 1)
      return 1;

    pSubElem = pSubElem->NextSiblingElement();
  } ? end while pSubElem ?
  return 0;
} ? end freetvaddon_get_playinfo ?

```

Pic 23. freetvaddon_get_playinfo Example

5. Configure and Build

1. Copy freetvaddons_devkit.tar.gz to any directory in linux.
2. Decompress freetvaddons_devkit.tar.gz
`# tar xzvf freetvaddons_devkit.tar.gz`
3. Go to freetvaddons_devkit directory
4. Set environment variables
`# source set_env.sh`
5. Implement your new Free TV+ addon source code
6. Compile addon
`# make`
7. Copy addon and png file to USB memory stick and insert USB memory stick to set-top box when Free TV+ is running. png file is optional and png filename should be the same as addon filename except version and extension, for instance, *addon_test.1.01.addon* and *addon_test.png*

6. Appendix

A. Analysis Streaming URL

In the case of not able to get the video streaming URL, You can check the video download web site [2][3] such as www.keepvid.com if it can be downloaded or not. If download is available and if download URL consists of the combination of obtainable value from the movie site, it can be made by using the source code (refer to the Metacafe addon)

B. Youtube Video ID

Regarding the website playing the movie through the YouTube, it is workable to forward the YouTube video ID only instead of video streaming URL (Refer to the FashionTV addon)

7. External Links

[1]: http://feeds2.feedburner.com/earth-touch_featured_480p_commentary

[2]: <http://www.keepvid.com>

[3]: <http://www.savevideo.me>

<Important Notice>

This document contains confidential and privileged information only to intended recipient.

It should be treated with a full care and used only for Fortis FreeTV+.

Uploading/exposing this document to websites (company, forum, and any type of website) and distributing to unrelated parties is strictly prohibited.