ReviewGen scripting and in-line macros

Scripting

Within the *reviews.mdb* database (*MS Access 97*), topics, and the reviews themselves, a form of scripting has been provided to assist in the creation of nice looking output.

The final document is built up as follows:

* A standard HTML header is written.
* The embedded styles are written.
* The JavaScript code is written. Note that there is (currently) no facility for "automating" the language code, so the language code and those languages defined in the "*Languages*" database must match.
* Topics are scanned to see if any have a *negative* ID number. If so, these are written out of the *include* flag is "yes".
  Topics are written in ascending order, which would be -3, -2, -1…
* Topics are scanned to see if any *positive non-end* topics have a topic link. If so, topic links are now written
  This is not currently implemented, it will be added soon.
* An index of all of the reviewed films is now written.
* If any films were detected with non-Latin localised titles (i.e. Japanese), then these titles are now written - sorted according to the sort order of the Latinised titles. This is not always correct as the Latinised transliteration may begin with a different letter of the alphabet.
  This will be looked at in a later release.
* Link code for hiding things during "prepare for printing" is written if topic #0 exists; followed by topic #0.
  If "prepare for printing" is to be supported, there *must* be a topic zero which is the explanation/display/option markup.
  This is not currently implemented, it will be added soon.
* All *positive non-end* topics which are able to be hidden are now written out.
* If there was a topic #0, finalisation code for topics-to-hide is written.
  This is not currently implemented, it will be added soon.
* All *positive non-end* topics which are not desired to be hidden are now written out.
* One by one, all of the reviews are built and written.
* All *positive end* topics are written. This means those topics with a positive ID and the topic title beginning "@END@" (which is clipped off).
* HTML end code is written.

At all times, numerous macros are available and all data read from the database is processed before being written out to the output HTML file.

Data processing

In the data to be output, several acts of processing are performed:

Newlines
If there are *two* newlines in a row, the sequence "\n<p>\n\n" is output.
If there is only *one* newline, and it isn't the end of the data, the sequence "
\n" is output.
The *exception* is if the first non-whitespace character of a line is "<", the newline will simply be output as-is. The reason for this is to prevent HTML code (i.e. for embedded tables and/or lists) from being mucked about with.

Accented characters
It is *recommended* that high-bit-set ASCII be included in the review texts - for Amélie is more readable than Amélie.
High-bit ASCII will be converted to entities.

The ampersand
The ampersand is *not* converted to an entity *unless* it is followed by a space - so "*Rosemary **&** Thyme*" would be converted, while "*Am**&**eacute;lie*" would not be; however any '&' characters in JavaScript are *never* converted.

Disable entity conversion
It may be required to have a character such as "é" output as-is. If this is the case, escape it.

Escaped values
Because of the necessity to include HTML (such as italics), the angle bracket characters are passed through as-is.
If any are required to be converted to entities < or > then they will need to be escaped. This is back to front to normal - the justification is that you are more likely to want HTML angle brackets than visible angle brackets (the FilmFour reviews doesn't have any), and I did not wish to pollute the HTML with the necessity of escaping the angle brackets.

Escapes are as follows:

\<	<
\>	>
\{	{ - **this is *required* within JavaScript code and styles.**
\}	}
\\	\
\ < highASCII >	Output high-ASCII character as-is, like \é outputs é
\n	Output a newline (CRLF)
\p	Output a double-newline "paragraph" (CRLF, CRLF).
\d###	Output character referenced by decimal value "###"
\x##	Output character referenced by hex value "##"

Characters output with \d or \x must be followed by exactly three digits and two digits respectively, pad shorter values by prefixing zeros - such as \d065 for an upper case 'A'.

Macros
Macros are contained within curly brackets. These are described below.

Macros

Macros are control words enclosed in curly brackets. They may either be a single word command, such as:

```
{DATE}
```

or a command with a parameter, such as:

```
{LINKTO "Amélie"}
```

Note that *any* macro may be used at *any* point of the output, however those which are not valid for what is happening (i.e. Picture1's caption while outputting a topic) will return *undefined* results.

The available macros are as follows:

Macros which may be used anywhere:

{DATE}	Return the *current* date as a simple date in long format. Example: "16 December 1973"
{DATEISO}	Return *current* date in ISO-ish format. Example: "1973/12/16"
{DATEFANCY}	Return *current* date in fancy marked up format. Example: "16th December 1973"
{LINKTO "<x>"}	Returns JavaScript-enhanced <a href> code to link to the film referenced by "x". This should match one of the *titles*. Normally the link would be displayed as the title of the movie (i.e. "*Nausicaä Of The Valley Of The Winds*"), however the parameter CAPTION may be used to provide an alternative name, so the link above can be changed to read simply "*Nausicaä*". Normal: {LINKTO "Amélie"} Captioned: {LINKTO "Amélie" CAPTION "Amie"}
{TIME}	Return *current* time in 24 hour format. Example: "22:08"

Use-anywhere macros intended for topics:
These macros may be used anywhere, but are intended to be used within topics.

{FAVECOUNT}	Returns the number of reviews with a score that is eight or more.
{PICCOUNT}	Returns the number of screenshot pictures provided within the reviews. If pictures are inserted directly into the review text, it is possible to provide a fake picture reference in order to permit the picture count to be correct.
{REVIEWCOUNT}	Returns the number of reviews.

Less common use-anywhere macros:
These macros are not normally to be used, but may be useful in specific situations - such as:

```
Copyright &copy; {YEAR} {AUTHOR}
```

at the end of the document.

{AUTHOR}	Return name of programmer. Example: "Rick Murray"
{MONTH}	Return the *current* month as a number, with January as 1. Example: "10".
{MONTHNAME}	Return the *current* month as a name in the language of the computer's locale. Example: "October"
{OUTPUTPOS}	Returns a value giving the current byte offset into the output file. Note that the value is corrected so that the *first* byte is zero (C / OS style) and *not* one (VB style). Example: "127062"
{SOFTWARE}	Return the software title. Example: "ReviewGen"
{VERDATE}	Return software build date in ISOish format. Example: "2007/07/07"
{VERFULL}	Return the *full* version of the software. This is not commonly used as the revision indicates sub-builds. Example: "1.02r0"
{VERSION}	Return the *normal* version number of the software. Example: "1.02".
{YEAR}	Return the *current* year as a four-digit number. Example: "2007".

Macros which may be used in review text:
The following macros may be used in the review text and also in the extended commentary text. When a review or extended commentary is read, the macros are defined according to the contents of the current review.
These macros may also be used in the movie review *links*, where they are equally available for each movie in turn.

{LINKID}	Return review title as a *#link* title, all lower-case. Anything *not* alphanumeric is converted to an underscore. Example: "jack___sarah" (for "*Jack & Sarah*").
{FLAGS}	Return the flags for an object. This is intended for the movie list link. Example: "F_JAPA+F_ANIM+F_SUBT+F_FAVE"
{LANGINFO}	Return a synopsis of the language/subtitle/dub information, or "" if not applicable. Example: "Japanese animé, subtitled"
{SCORE}	Return the movie's score in HTML markup. Example: "8½"
{SCOREFRAC}	Return the movie's score as a fractional value. Example: "8.5"
{SHORTTITLE}	Returns a three-character title in lowercase. This is intended for extended comment button references. The three characters are the first three characters of the title. This uses the same rules as for the {LINKID} macro. If there are less than three characters in the movie title, "x"s will be suffixed. **NOTE** that this (currently) fails with two movies that begin with the first three letters (i.e. "***Bat****tle Royale*" and "***Bat****tle Royale II*"); this will be addressed in a future version.
{TITLE}	Returns the title, converted to HTML. Example: "Amélie"
{TITLEFLAT}	Returns the title exactly as-is. Example: "Amélie"
{TITLEJS}	Returns the title with JavaScript munging, such as no single or double quotes in the string. Example: "Rick\x27s world"
{TITLELOCAL}	Return the object title in local language, if available. Note - ***not*** in the localised character set, use {LOCTItlEENT} for that. For Japanese this would be the *Romanji* title.
{TITLESP}	Return the title in HTML markup, with the addition of spaces being converted to " ". Example: "Cruel Intentions"

Other review macros:

{LOCTITLEENT}	Returns the entity (localised alphabet) part of the localised title. See below for information on localised titles. Example: "あずみ" (for "*Azumi*").
{LOCTITLEPRO}	Returns the pronunciation part of the localised title. See below for information on localised titles. Example: "(a-zu-mi)"

Macros for picture tables:
The following macros are only valid when picture <table> code is being referenced (in the *Reviews.HTMLLayout* database table. This is for the entries where the *LayoutEntity* field is either "*PicTableOne*" or "*PicTableTwo*".
The software will read the appropriate entry depending on whether there are one or two pictures to insert.
If only one picture, then only *PIC1xxx* is valid and references to *PIC2xxx* will return the same information as *PIC1xxx*. If a two-picture insert then both are valid and both return different information.

The picture width and height are determined on-the-fly by loading the picture and then requesting its dimensions.

{PIC1WIDTH}	Returns the width (in pixels) of picture 1. This is usually *320*.
{PIC1HEIGHT}	Returns the height (in pixels) of picture 1. This is usually *254* (4:3 aspect) or around *212* (letterboxed).
{PIC1NAME}	Returns the filename of picture 1. Example: "azumi1.jpeg"
(PIC1CAPTION}	Returns the caption for picture 1, in correct HTML markup.
{PIC2WIDTH}	Returns the width (in pixels) of picture 2. This is usually *320*.
{PIC2HEIGHT}	Returns the height (in pixels) of picture 2. This is usually *254* (4:3 aspect) or around *212* (letterboxed).
{PIC2NAME}	Returns the filename of picture 2. Example: "azumi2.jpeg"
(PIC2CAPTION}	Returns the caption for picture 2, in correct HTML markup.

The databases

The "*reviews.mdb*" file contains ***five*** database tables:

HTMLLayout
This table holds the layout code for things such as "review title as favourite".
It contains two fields - *LayoutEntity* and *LayoutCode*. This table is sorted according to the entity name, however this is merely a formality. Sort order is irrelevant.
Available layout entities are:

ExtCommentHead	Code to begin an extended comment.
ExtCommentID	ID name to use to refer to extended comment blocks.
ExtCommentTail	Code to finish an entended comment.
FilmInsertLink	Code to use for LINKTO references.
FilmListLink	Code to link back to main films list.
JavaScript	The JavaScript code to insert at the beginning, as a complete block. It is not currently possible to "automate" the code with respect to `show all` / `hide all` extended commentaries; these small modifications will need to be performed manually. This will be addressed in a future release.
MovieLink	Code to use in movies listing if *not* a favourite.
MovieLinkFave	Code to use in movies listing if a *favourite*.
LocaLink	Code to use in movies listing for the localised list, if *not* a favourite.
LocaLinkFave	Code to use in movies listing for the localised list, if a *favourite*.
PicTableOne	Picture table code if only *one* picture defined.
PicTableTwo	Picture table code if *two* tables are defined.
ReferralLink	Code to use for re-reference entry.
ReferralLinkFave	Code to use for re-reference entry if a *favourite*.
TitleEnd	Code to end a title if no language details to include.
TitleEndLangInfo	Code to end a title if there are language details.
TitleStart	Code to begin a review title.
TitleStartFave	Code to begin a *favourite* review title.
TitleStartVFave	Code to begin a ***very*** *favourite* (10+) review title.

Entity names should ***not*** be changed.

HTMLStyles
This table holds the embedded stylesheet definitions. It contains three fields - *StyleID*, *StyleName*, and *StyleCode*. When outputting the style information, the *StyleName* is written, followed by the *StyleCode*, ordered according to the ascending order of *StyleID*. The ID is used purely to dictate output order (it sometimes matters, with styles and CSS), it is not actually written out itself.

Therefore:

< StyleName >	< StyleCode >

could become:

BODY	{ text-align: justify }

The contents of *StyleName* and *StyleCode* are not important to *ReviewGen*.. Each is output in turn. What styles are required is largely dictated by other things, such as style references in the *HTMLLayout* table, and desired alterations from the norm (such as justification of body text).

Languages

This table holds the language references. It ***must*** match the flags/references given in *HTMLLayout.JavaScript* (as the JavaScript code is 'fixed', it is not built-on-the-fly).
This table is so we know what flags relate to what language. It contains three fields - *Language*, *LangID*, and *LangNum*. Entries are sorted according to *LangNum*, though the actual order is not of specific relevance.
How it works - in the review will be specified a language, say, "Japanese". This is looked up in the *Language* field. Once we have it, we can then read the associated *LangID* field, In this case "F_JAPA".
The "F_EBIS" flag is for movies that are mainly in English but have some parts in subtitled XXX. In this situation, the *Secondary Language* of the review is examined to see what this other language is.
The currently defined languages are: Cantonese, English, English-with…, French, German, Japanese, Korean, Mandarin, Russian, and Swedish.

The flags "F_ANIM", "F_DUBB", and "F_SUBT" are hardwired into *ReviewGen* as they are status flags and not languages.

Review

Here it is, the review table… It contains nineteen fields, sorted according to the first:

Title	The title of the film, as given in the EPG.
Score	The film's score, points out of ten. For fractionals: x.14 and x.25 are a quarter. x.12 and x.5 are a half. x.34 and x.75 are three quarters. This is so fractions can be given to 'look' like the fraction (i.e. 7.34 for 7¾) or to be mathematically correct (i.e. 7.75).
LocalisedTitle	If the film has a different title in its own language, that title is given here. This field may be expanded to cater for the title in the local non-Latin alphabet (i.e. Katakana, Cyrillic…), in which case you will need to provide the glyph codes as *ReviewGen* cannot directly edit/display, for example, Asian characters.
Language	The film's language.
SecondaryLanguage	If in another language part-subtitled, this specifies that second language.
Subtitled	Yes/No: is the film subtitled?
Dubbed	Yes/No: has the film been dubbed into English?
Anime	Yes/No: is the film an animé? (specific: Japanese animation)
Date Added	Date review added; not available for the original 65 reviews.
PictureBefore1	Filename of first included picture - i.e. "*azumi1.jpeg*". This picture, if specified, will appear *before* the review text.
PictureBefore1Caption	Caption of first (before) picture.
PictureBefore2	Filename of second included picture, which can be included *before* the review text.
PictureBefore2Caption	Caption of the second (before) picture.
Review	The review text.
PictureAfter1	Filename of the third (first after) included picture, which can be included *after* the review text (and before the extended review, if any).
PictureAfter1Caption	Caption of the third (first after) picture.
PictureAfter2	Filename of the fourth included picture, to appear *after* the review.
PictureAfter2Caption	Caption of the fourth picture.
ExtendedCommentary	The extended commentary/review text, leave blank if there is no extended commentary. Extended commentaries are normally hidden - this is because these commentaries may contain spoilers.

The layout of a review is controlled by several *HTMLLayout* table entries, however the basic format is:.

```
<title> (<score>/10) <langinfo>
                  <pic before 1> <pic before 2>
<review text>
                   <pic after 1> <pic after 2>
<extended commentary>
```

The parts that do not apply in any specific instance (i.e. no language info, no or fewer pictures, no extended commentary, etc) are simply *not* output.

Topics

This table provides additional topics which can be inserted into the output file.
There are six fields, and the table is sorted according to *TopicID*.

TopicID	A number giving the identity of the topic, which in turn determines its placement. IDs do *not* have to run contiguously.
TopicTitle	The title of the topic.
TopicInclude	Yes/No: Should this topic be included in the output file?
TopicIsLinked	Yes/No: If *TopicID* is positive and *non-end*, should it have a topic link before the movies list?
TopicCanBeHidden	Yes/No: If *TopicID* is positive and *non-end*, should it be within the part that can be hidden when the document is "prepared for printing".
TopicText	The content of the topic, which is written out.

Where topics are placed in the output file

Topics are output according to their *TopicID*. There are *four* types of topic:

Negative *TopicID*

Topics with a *negative* ID are written at the very start of the file, after the styles and JavaScript (neither of which are 'visible' objects).
By way of example, the topic with the most negative number is the "Top Of Document", which provides the title and the picture of the girl flying through the window. The next topic provides the "last updated" information. Then comes the NSPCC advert. Finally, the review count. These topics can be added to, additional topics inserted, and for any build of the document they can be selectively included or discluded from output.
For these topics, the *TopicIsLinked* and *TopicCanBeHidden* choices have no relevance and are thus ignored.

TopicID is zero

This topic contains the user-interface part of the "prepare for printing" mechanism, which will probably be a brief explanation plus a "Prepare for printing" button.
If this topic *does not exist* or is set to *not be included*, then the code part of the "prepare for printing" mechanism will not be output.
The *TopicIsLinked* and *TopicCanBeHidden* options are ignored.

Positive non-end *TopicID*

These topics, which have a *positive* ID, will be placed *after* the movie list. Those flagged as *TopicCanBeHidden* will be output first, then the hide code (if there is a topic zero), then any flagged as ***not*** *TopicCanBeHidden*. This behaviour is retained even if there is no "prepare for printing" mechanism.
If the topic has the *TopicIsLinked* flag set, then there will be a link to the topic inserted just before the movie list.
By way of example, such a topic is "Rick recommends" (linked).

Positive end *TopicID*

These are topics which have a *positive* ID and the topic title begins with "@END@" (which is removed). These topics are output, in order, *after* the final review. By way of example, such topics are the "add to favourites" button, the end credits, the return link, and the copyright bit at the very bottom.
The *TopicIsLinked* and *TopicCanBeHidden* options are ignored.

Interlinking reviews

It is strongly recommended that links are provided for references to other FilmFour movies within a review - such as to say "Stylish it is, but Azumi it ain't", the word Azumi should be a link to the *"Azumi*" review.
The {LINKTO "xxx"} macro may be used to insert links, and the optional CAPTION parameter may be used to control how the link looks, if the actual movie title is too long or not desired.

Ensuring the picture count is correct

If, for stylistic reasons, you choose to insert a picture into the review manually, then in order to keep the picture count correct, provide it as one of the pictures to be included, but prefix it with an '@'. No caption should be given. The '@' will instruct the HTML generator to ignore the picture, while the picture counter will include it.

Localised titles

If the review has a localised title, for example if a film is French and the title in English differs from the title in French, then you would write the French (localised) title into the localised title field.
If the review has a localised title in a different alphabet, then this field is extended as follows:

```
<localised title in Latin>@<localised title in its own
alphabet (as Unicode glyphs)>@(<how to pronounce it>)
```

For example:

```
Azumi@&#12354;&#12378;&#12415;@(a-zu-mi)
```

If the localised title is specified in its own alphabet, the pronunciation part *must* be given. This pronunciation should be provided within brackets.
Unicode glyphs *must* be given in *&#xxxx;* form, as *ReviewGen* cannot read/interpret double-byte characters.

Path specifiers

This software is closely tied to my system. As it has a very specific purpose, this is *unlikely* to change.

Database	reviews.mdb within *ReviewGen*'s directory.
Review	C:\RickMisc\wwwsite\ricksworld\digibox\film4review.html
Preview	C:\RickMisc\wwwsite\ricksworld\digibox\film4preview.html
Images	C:\RickMisc\wwwsite\images\film4\

Project files and source code

This is *not* a project intended for public release. I am willing to share source (VisualBasic 5) for those wishing to see how something was implemented, however there are no plans for any official release of the source, executable, or review database.

Disclaimer and contact info

This documentation describes a *work in progress*, therefore please be aware that the available database tables, fields, meanings, and the macros may be subject to change.

The *FilmFour reviews* may be found at:

```
http://www.heyrick.co.uk/ricksworld/digibox/film4review.html
```

The latest version of this document is:

```
http://www.heyrick.co.uk/software/reviewgen/spec.pdf
```

My email address is:

heyrick *-at-* merseymail *-dot-* com
(please note that I only have Internet access for half an hour a week at the local library!)