Workshop: WWP Standard Edition und FrameMaker 6 in der Praxis

Dipl.-Ing. (Univ.) Michael Müller-Hillebrand, cap studio, Erlangen, 4.12.02001
<http://cap-studio.de> <mmh@cap-studio.de>

Installation

- FrameMaker[+SGML] 6.0p405
- WebWorks Publisher Standard Edition 6.0.6 (Mac: 6.0.2):
  ftp://ftp.webworks.com/publisher/standard/wpubse.6.06.exe
  ftp://ftp.webworks.com/publisher/standard/WebWorks.6.0.2.img.hqx
- Mit dem aktuellen AdobePS-Druckertreiber für Ihr Betriebssystem und <wwp-installationsverzeichnis>\lib\wwrast\webworks.ppd einen Drucker »Web-Works Rasterizer« erstellen

FrameMaker-Buch vorbereiten

- Paralleles HTML-Buch erstellen, irrelevante Dokumente entfernen
- Inhaltsverzeichnis an den Anfang, Stichwortverzeichnis ans Ende (mit Hypertext-Verbidungen!)
- Inhaltsverzeichnisebenen festlegen

Erste Konvertierung

- New Project Wizard: Dynamic HTML
- Erstes Mapping (Listen, Überschriften, Beginn neuer HTML-Seiten, Inhaltsverzeichnis- und Index-Formate)

FrameMaker	WebWorks Publisher
Bulleted	SmartList1
Ebene1IX	Level1IX
GruppentitelIX	GroupTitlesIX
Heading1TOC	TOC2
Heading2	NewHTMLPage
Heading2TOC	TOC3
Numbered	SmartList1
Numbered1	SmartList1
SectionTitle	Title
SectionTitleTOC	TOC1
TableTitleFirst	TableTitle

HTML-Layout anpassen

- Zielseite und -grafiken in Support-Verzeichnis kopieren
- Zielseite in Normal.asp umbenennen und mit alter Normal.asp vergleichen:
  - alle Nutzdaten entfernen und durch **$DATA; $NOTES;** ersetzen
  - WWP-Makros einsetzen:
    Im Kopfteil:
    Titel des Dokuments: **$GET_ATTR($PAGE;, title);**
    Aktuelles Datum: **$DATE(c);**
    Gewählte Textkodierung: **$CHARSET;**
    In der Navigationsspalte:
    Inhalt: **$PAGEFIRST(html, name, $CHARSET;);**
    Index: **$PAGELAST(html, name, $CHARSET;);**
    Im Fußteil:
    Aktuelle Seite: **$PAGE(html, basename, $CHARSET;);**
    Vorherige Seite: **$PAGEPREV(html, name, $CHARSET;);**
    Nächste Seite: **$PAGENEXT(html, name, $CHARSET;);**
- CSS-Stylesheets vergleichen und ggf. anpassen
- Erneut generieren

HTML-Layout optimieren

- TOC.asp und Index.asp aus Normal.asp entwickeln
  - In TOC.asp Link auf Inhalt entfernen und Navigationspfeil nach links deaktivieren
  - In Index.asp Link auf Index entfernen und Navigationspfeil nach rechts deaktivieren, Liste der Zielbuchstaben aus bisheriger Index.asp übernehmen
- Stylesheet optimieren
  - Nachsehen, welches class-Attribut verwendet wird und dann die entsprechende Spezifikation im Stylesheet nachtragen bzw. anpassen.
- Inhaltsverzeichnisdarstellung: Eintrag unter **FrameMaker And Generated Files > Properties > Macros > UMTOCFilter** kürzen zu:
  $GET_GLOBAL(tocpara)["[ ][\-0-9xviXVI]+$",""];

Alle Daten zum Nachvollziehen

<http://cap-studio.de/files/wwp-training.html>

Nützliche WWP-Makros

In den *.asp-Dateien können Sie auch folgende WWP-Makros verwenden:

$BASETEMPLATENAME;

The $BASETEMPLATENAME macro returns the name of the original WebWorks Publisher template upon which the current project template is based.

$CHARSET; $CHARSETNAME;

The $CHARSET macro is required to support international character set output in WebWorks Publisher. $CHARSET returns the name of the current character set.

$CHARSET ($CHARSETNAME) returns one of the following values:

- ISO-8859-1 (Latin1)
- UTF-8 (Unicode)
- EUC-KR (Korean)
- GB2312 (Simplified Chinese)
- Big5 (Traditional Chinese)
- SHIFT_JIS (Japanese)

@COPY;

The **@COPY(mode, pathType, filenameSource, filenameDestination);** macro copies the contents of *filenameSource* to *filenameDestination*.

The *mode* argument is optional, and can be one of three possible values:

overwrite	If the file does not exist, @COPY creates the file and copies the contents of *filenameSource* into the new file. If a file does exist, it is overwritten by the contents of *filenameSource*.
if-newer	If the file does not exist, @COPY creates the file and copies the contents of *filenameSource* into the new file. If a file does exist, @COPY checks the last saved date of the *filenameSource* file and compares it to the last saved date of the *filenameDestination* file: the file contents are not copied unless the last saved date of *filenameSource* is newer than that of *filenameDestination*.
none-exists	@COPY checks to see if a file by the name of *filenameDestination* already exists in the target directory. Only if the file has not already been created will the contents of *filenameSource* be copied to the target directory.

The *pathType* argument values are described in section »$DOC; $DOCFIRST; $DOCLAST; $DOCNEXT; $DOCPREV;«.

$DATE;

The $DATE macro returns the date and time when the project was generated. This is based upon the time reported by the user's computer clock. It returns the date in the following format:

January 31, 2000

You can change the output format by using an optional one-character parameter:

Parameter	Date information returned	Example
a	Abbreviated weekday name	Sun
A	Full weekday name	Sunday
b	Abbreviated month name	Jan
B	Full month name	January
c	Short date and time	Jan 31 14:35:03 2000
C	Long date and time	Sunday, January 31 14:35:03 2000
d	2-digit day of month, ranging from 01–31	31
H	2-digit hour, in 24-hour time ranging from 00–23	14
I	2-digit hour, in 12-hour time ranging from 01–12	02
j	3-digit day of year, ranging from 001–366	31
m	2-digit month number, ranging from 01–12	01
M	2-digit minute number	35
p	AM or PM	PM
S	2-digit seconds, ranging from 00–61; allows for leap seconds	03
U	2-digit week number of the year, ranging from 00–53, where Sunday is the first day of week 1	05
w	2-digit weekday number, 0–6, where Sunday begins the week at 0	0
W	2-digit week number of the year, ranging from 00–53, where Monday is the first day of week 1	04
x	Date, usually mm/dd/yy	01/31/00
X	Time, usually hh:mm:ss	14:35:03
y	2-digit year	00
Y	4-digit year	2000
Z	Time zone	Central Standard Time

@DELETE;

The **@DELETE(pathType, filename);** macro deletes the file specified by *filename*.

The *pathType* argument values are described in section »$DOC; $DOCFIRST; $DOCLAST; $DOCNEXT; $DOCPREV;«.

$DOC; $DOCFIRST; $DOCLAST; $DOCNEXT; $DOCPREV;

The $DOC macros returns the filename of the current (first, last, next, previous) FrameMaker document being processed. The format is:

$DOC(pathType,format, optional_encoding);

The first argument tells WebWorks Publisher what format the filename should use.

> *Note:* Do not be confused by the differences between $DOC, $PAGE, and $FILE. $DOC returns the filename of the *current FrameMaker document* file (.fm) being processed. $PAGE returns the name of the *current output page* file being generated. $FILE returns the filename of the *current output file* being processed, much like $PAGE, except that the result can be an output page file or an output graphic file.

The tables list the arguments.

pathType argument	Description
host	Produces a filename and a path using the conventions of the computer running WebWorks Publisher.
html	Produces a path and/or filename in the proper format for use in HTML output documents.
dos	Produces a filename and a path that use DOS conventions. Normally not used in regular output, but may be useful in scripts or complex macros that need to write or read files outside the WebWorks Publisher project.
mac	Produces a filename and a path that uses Macintosh conventions.
unix	Produces a filename and a path that use UNIX conventions. Normally not used in regular output, but may be useful in scripts or complex macros that need to write or read files outside the WebWorks Publisher project.

The second argument tells WebWorks Publisher how much of the filename should be returned.

format argument	Description
basename	The format of the name as it appears in FrameMaker minus any path information, or the WebWorks Publisher–generated filename.
fullname	The complete filename, including the path.
fullpath	Includes the complete path to the file as it appears in FrameMaker, or the WebWorks Publisher–generated filename.
name	The name of the file as it appears in FrameMaker, or the WebWorks Publisher–generated filename. It is important to note that differing amounts of path information may be provided in the FrameMaker source document, especially in links or cross-references: the macro argument can only return as much path/filename data as is available from the FrameMaker source document.
path	The path to the output directory that contains the file, in relationship to the project directory. Rarely used, and then mostly in graphics macros. For example, if the graphics output directory is specified to be images, a file expansion macro referencing a graphics file and using path as its second argument would return images.
relative	Translates the filename relative to the original document.

The third argument is optional and can be any value that is defined for the $CHARSET; macro. You would only use this argument if you were going to emit the name of a file into your generated output pages. You'll notice this argument used in many of the ASP files in your project's Support directory. If this parameter is not provided, the output encoding is the same as the host machine's encoding.

$DOCNUMBER; $DOCTOTAL;

The $DOCNUMBER macro is a numeric macro starting at 1 that returns the current sequential position of the FrameMaker document in the project.

The $DOCTOTAL macro is a numeric macro that returns the total number of FrameMaker documents in the project.

These macros might be used to create a part of the header or footer of a generated output document to indicate the relative position of the FrameMaker source document within the project as a whole.

@EXECUTE;

The **@EXECUTE(script_name);** macro sets the working directory to the project's output directory, and then handles the program or script specified by *script_name*.

> *Note:* The script or program execution must complete before WebWorks Publisher can continue.

$IF_EQUAL; $IF_EQUAL_IGNORECASE;

The **$IF_EQUAL(FirstString, SecondString, true);** macro compares the value of *FirstString* with the value of *SecondString*. If the two values are equivalent, the macro returns the text in the *true* argument.

The **$IF_EQUAL(FirstString, SecondString, true, false)**; macro compares the value of *FirstString* with the value of *SecondString*. If the two values are equivalent, the macro returns the text in the *true* argument. If the two values are not the same, the macro instead returns the text in the *false* argument.

$IF_NOTEQUAL; $IF_NOTEQUAL_IGNORECASE;

$IF_NOTEQUAL works exactly like the $IF_EQUAL macro but in reverse. That is, if the two values in the **$IF_NOTEQUAL(FirstString, SecondString, true);** macro are not equivalent, the macro returns the text in the *true* argument.

In the macro **$IF_NOTEQUAL(FirstString, SecondString, true, false);** if the two values are not equivalent, the macro returns the text in the *true* argument. If the two values are the same, the macro returns the text in the *false* argument.

$IMPORT;

The **$IMPORT(filename);** macro allows the user to import the complete contents of the file specified in *filename* and returns the imported text. If *filename* is not an absolute path, it is assumed to be relative to the project's output directory ($OUTPUTDIR;), usually $PROJECTDIR;/Output.

$IMPORTANDEXPAND;

The **$IMPORTANDEXPAND(filename);** macro pulls in predefined page styles containing macros. The contents of *filename* are imported, then expanded, just as if it were text in the page style.

In addition, the $IMPORTANDEXPAND macro imports the contents of the file specified without regard to line endings (that is, Mac, DOS, or UNIX), so page templates can be developed on any machine and used with WebWorks Publisher.

$IMPORTANDEXPAND can also help WebWorks Publisher to integrate with other editors. Because the various page styles are now available as external files, users of HTML editors such as Adobe GoLive or Macromedia Dreamweaver can take advantage of the WYSIWYG editing of their software to set up page appearance as desired. If you prefer, you can still edit external files with standard text editors such as Windows Notepad, allowing maximum flexibility.

@MOVE;

The **@MOVE(mode, pathType, filenameSource, filenameDestination);** macro moves the contents of *filenameSource* to *filenameDestination*.

The *mode* argument is optional, and can be one of three possible values:

overwrite	If the file does not exist, @MOVE creates the file and copies the contents of *filenameSource* into the new file. If a file does exist, it is overwritten by the contents of *filenameSource*.
if-newer	If the file does not exist, @MOVE creates the file and copies the contents of *filenameSource* into the new file. If a file does exist, @MOVE checks the last saved date of the *filenameDestination* file. The file contents are not copied unless the last saved date of *filenameSource* is newer than that of *filenameDestination*.
none-exists	@MOVE checks to see if a file by the name of *filenameDestination* already exists in the target directory. If the file has not already been created, the contents of *filenameSource* are moved to the target directory.

The *pathType* argument values are described in section »$DOC; $DOCFIRST; $DOCLAST; $DOCNEXT; $DOCPREV;«.

$OUTPUTDIR;

The **$OUTPUTDIR(pathType, format, optional_encoding);** macro returns the path to the directory containing the project's output files.

The *pathType*, *format*, and *optional_encoding* argument values are described in section »$DOC; $DOCFIRST; $DOCLAST; $DOCNEXT; $DOCPREV;«.

$PAGE; $PAGEABSLAST; $PAGEFIRST; $PAGELAST; $PAGENEXT; $PAGEPREV;

The **$PAGE(pathType, format, optional_encoding);** macro returns the filename of the current (very last, first, first page of the last document, next, previous) output page being processed.

> *Note:* Do not be confused by the differences among $DOC, $PAGE, and $FILE. $DOC returns the filename of the *current FrameMaker document* file (FM) being processed. $PAGE returns the name of the *current output page* file being generated. $FILE returns the filename of the *current output file* being processed, much like $PAGE, except that the result can be an output page file or an output graphic file.

The *optional_encoding* parameter can be one of any value defined for $CHARSET. If this parameter isn't provided, the output encoding is the same as the host machine's encoding. In most cases, the parameter is the character encoding for the entire project. You select the default charset in the project wizard when you create a new project. The *optional_encoding* parameter is also used in international filename handling and in all the ASP files for hyperlinking navigation bars.

The *pathType*, *format*, and *optional_encoding* argument values are described in section »$DOC; $DOCFIRST; $DOCLAST; $DOCNEXT; $DOCPREV;«.

$PAGENUMBER; $PAGETOTAL;

The $PAGENUMBER macro is a numeric macro starting at 1 that returns the current sequential position of the current output page within the project.

The $PAGETOTAL macro is a numeric macro that returns the total number of output pages in the project.

$PRODUCTDIR;

The **$PRODUCTDIR(pathType, format, optional_encoding);** macro returns the path to the WebWorks Publisher Professional Edition installation directory.

The *pathType*, *format*, and *optional_encoding* argument values are described in section »$DOC; $DOCFIRST; $DOCLAST; $DOCNEXT; $DOCPREV;«.

$PROJECTDIR;

The **$PROJECTDIR(pathType, format, optional_encoding);** macro returns the path to the current project directory. By default, the $PROJECTDIR directory contains the Output, Temp, and Support directories, although this structure can be changed. It may also contain the project WDT file.

The *pathType*, *format*, and *optional_encoding* argument values are described in section »$DOC; $DOCFIRST; $DOCLAST; $DOCNEXT; $DOCPREV;«.

$PROJECTNAME;

The $PROJECTNAME macro returns the name of the project without the .wdt extension. It is useful to note that the $PROJECTNAME may be different from the name of the Project Directory.

$SEP;

The $SEP macro returns a path separator based on the platform on which the project is generated. On DOS platforms, a backslash (\) is returned; on UNIX platforms, a forward slash (/) is returned; on Macintosh platforms, a colon (:) is returned.

$SPLITFIRST; $SPLITLAST;

The **$SPLITFIRST(pathType, format, optional_encoding);** macro returns the filename of the first (last) output page of the current FrameMaker document being processed. This macro is used when the output from a single FrameMaker source document is split into multiple output page files.

The *pathType*, *format*, and *optional_encoding* argument values are described in section »$DOC; $DOCFIRST; $DOCLAST; $DOCNEXT; $DOCPREV;«.

$SPLITNUMBER; $SPLITTOTAL;

The $SPLITNUMBER macro is a numeric macro starting at 1 that returns the current sequential position of the current output page of the current FrameMaker document being processed.

The $SPLITTOTAL macro is a numeric macro that returns the total number of output pages in the current FrameMaker document being processed.

$SUPPORTDIR;

The **$SUPPORTDIR(pathType, format, optional_encoding);** macro returns the path to the Support directory of the current project. The $SUPPORTDIR directory contains files needed by the project template, including the template readme.txt file and revision history, ASP files, various help files, and so on.

The *pathType*, *format*, and *optional_encoding* argument values are described in section »$DOC; $DOCFIRST; $DOCLAST; $DOCNEXT; $DOCPREV;«.

$TEMPDIR;

The **$TEMPDIR(pathType, format, optional_encoding);** macro returns the path to the directory containing the project's temporary files.

The *pathType*, *format*, and *optional_encoding* argument values are described in section »$DOC; $DOCFIRST; $DOCLAST; $DOCNEXT; $DOCPREV;«.

$TEMPLATENAME;

The $TEMPLATENAME macro returns the name of the template used to create the current project.

$TIME;

The $TIME macro inserts the time of the current generation into a document. The format is in 12-hour time, hh:mmAM/PM Time Zone.

If a different time format is needed, use one or more instances of the $DATE(n); macro instead with the appropriate arguments to create the date format desired. See »$DATE;« for details.

$USERNAME;

The $USERNAME macro inserts the WebWorks Publisher 6.0 end user's name into the output document. For Windows users, the name retrieved will be the name of the registered user. In UNIX, the name retrieved will be the name of the current login.

This macro is typically used to record the user information for the person who made the last changes to a document. $USERNAME is often used along with the $DATE macro to create a listing in the output page footer.

$VERSION;

The $VERSION macro inserts information about the version of WebWorks Publisher 6.0 used to generate the current output document.

@WRITE;

The **@WRITE(mode, pathType, filename, contentsToWrite);** macro writes the argument *contentsToWrite* to the filename specified with *pathType* and *filename*.

The *mode* argument is optional and has two allowed values:

overwrite	If the file does not exist, @WRITE creates the file and copies the contents of *contentsToWrite* into the new file. If a file does exist, it is overwritten by the contents of *contentsToWrite*.
append	If the file does not exist, @WRITE creates the file and copies the contents of *contentsToWrite* into the new file. If the file does exist, the contents of *contentsToWrite* are appended into the existing file.

The *pathType* argument values are described in section »$DOC; $DOCFIRST; $DOCLAST; $DOCNEXT; $DOCPREV;«.