

ISIS Sample Application

Processing HTML files with DocEXEC

User's Guide

Document number htmlue/600/1

November 4th, 2021

ISIS Information Systems Ges.m.b.H.
Alter Wienerweg 12
A-2344 Maria Enzersdorf/Vienna
Tel: +43 (0) 2236 27551 0*
Fax: +43 (0) 2236 21081
Product Support: +43 (0) 2236 27551 111
support@isis-papyrus.com
General Information: info@isis-papyrus.com
Website: www.isis-papyrus.com

Copyright 2021 ISIS Papyrus Software AG

All rights worldwide are the property of
ISIS Papyrus Software AG.
Changes to the Software, Help Texts and Manual may
be made without prior announcement.

Reproduction in whole or in part, of this manual,
or of the information on the accompanying media,
is only permitted with the express permission of
ISIS Information Systems GmbH.
Excepted from this rule are legally and contractually
bound licencees of **ISIS** Papyrus Software AG.

This document is available also in PDF format converted
with the ISIS Papyrus PDF-Server.

Table of Contents

1.0 Introduction	1
1.1 The concept behind HTM2AFP.DFA	1
2.0 Supported tags	2
2.1 Headings	3
2.1.1 Special tag attributes	3
2.2 Font styles	3
2.3 Line breaks	3
2.4 Page breaks	3
2.4.1 Example	4
2.5 Indents	5
2.5.1 Example	5
2.6 Images	5
2.6.1 Example	5
2.7 Ordered and Unordered Lists	5
2.7.1 Example	6
2.7.2 Limitations	6
2.8 Definition lists	6
2.8.1 Example	7
2.8.2 Limitations	7
2.9 Tables	7
2.9.1 Example	7
2.9.2 Limitations	7
3.0 Print formatting with HTML	8
3.1 Page breaks	8
3.2 Images	8
3.3 Links	8
3.3.1 Examples	9
3.4 Indices	9
3.4.1 Example	9
4.0 Definition and usage of variables	10
Index	11

1.0 Introduction

The DocDef **HTM2AFP.DFA** uses HTML files as input data for DocEXEC and shows how HTML tags can be processed (HTML 3.2, no CSS support). This DocDef is neither performance optimized (approximately 0.5 pages per second on a Pentium III 333 MHz with 128 MB RAM) nor does it support every each tag.

1.1 *The concept behind HTM2AFP.DFA*

The DocFormat `DF_FIND_NEXT_TAG` checks the record for tags defined in `DF_BEFORE_FIRST_DOC`, if it does not find any of them it adds the next record via `DF_ADD_RECORD` and searches again. `DF_FIND_NEXT_TAG` then determines which of the found tags comes first in the search string and cuts the string right before the first tag. The part before the first tag is then processed by `DF_FIND_FORMATTING`, which converts any combination of font style tags (` ... bold`, `<i> ... italic`, `<u> ... underline`) to DocEXEC readable strings, and output by `DF_OUTPUT_TEXT`. In the following loop the string starts with the first found tag, thus the main DocFormat will call the appropriate DocFormat for further processing (e.g. `DF_FIND_IMAGE`). Tags which should not trigger a line break (e.g. anchors ... ` `) or occur within other tags (e.g. an image within a table cell) are ignored by `DF_FIND_NEXT_TAG` but processed within the DocFormats starting with "DF_FIND" by calling the required DocFormat (e.g. `DF_INCLUDE_IMAGE`).

2.0 Supported tags

The following tags are supported:

Feature	related tags
Heading	<H1>, <H2>, <H3>, <H4>
Linebreaks	 , <P>, </P>
Font styles	, , <I>, </I>, <TT>, </TT>, <U>, </U> (, , ,)
Ordered lists	, , ,
Unordered lists	, , ,
Definition lists	<DL>, <DL compact>, </DL>, <DT>, </DT>, <DD>, </DD>
Tables	<TABLE border="..." width="...">, </TABLE>, <CAPTION>, </CAPTION>, <TR>, </TR>, <TD width="...">, </TD>
Images	
Indents	<BLOCKQUOTE>, </BLOCKQUOTE>
Pagebreaks	<HR>
Indices	
Preformatted text	<PRE>, </PRE>
Variables	<SCRIPT>, </SCRIPT>

The following tags are removed:

Feature	related tags
General tags	<HTML>, </HTML>, <TITLE>, </TITLE>, <HEAD>, </HEAD>, <BODY>, </BODY>, <!DOCTYPE>, <META>
Alignment	<CENTER>, </CENTER>
Sections	<DIV>, </DIV>
Comments	<!--, -->
Fonts	,

2.1 Headings

Headings of level 1 to 4 can be defined, on the printed document they are output with a leading number. Headings of level 1 are output 36 mm below the top margin.

2.1.1 Special tag attributes

Heading level 1 may contain the ALT attribute to specify the following:

<code><h1 alt="Glossary"> Glossary </h1></code>	outputs the title for a glossary (in combination with a succeeding <code><dl compact></code>)
<code><h1 alt="NoNumbering"> Preface </h1></code>	outputs a chapter which should not be numbered (e.g. the preface should not be numbered like the index)

2.2 Font styles

Any combination of the following font styles may be used:

Font style	Example
bold	<code>bold</code>
<i>italic</i>	<code><i>italic</i></code>
<u>underline</u>	<code><u>underline</u></code>
fixed width	<code><tt>fixed width</tt></code>

2.3 Line breaks

The following tags are supported (no style sheet support for the `<p>` tag):

<code>
</code>	... outputs a single line break
<code><p></code>	... starts a new paragraph (two line breaks are output)
<code></p></code>	... ends a paragraph

2.4 Page breaks

The tag for the horizontal rule is used to specify a page break

<code><hr></code>	... outputs a page break
-------------------------	--------------------------

2.4.1 Example

The page breaks here:

2.5 *Indents*

The tag `<blockquote>` is used to indent each element of the document, nested indents are possible.

`<blockquote>` ... indent starts
`</blockquote>` ... indent ends

2.5.1 Example

Indents

Indents are used to structure the document.

2.6 *Images*

Images of any type from any place can be used, i.e. the path and the extension is ignored by the DocDef. In addition it is possible to add a figure caption, that is a numbered description at the bottom of an image.

2.6.1 Example

The following image contains an ALT attribute which is used by the DocDef to generate a figure caption.

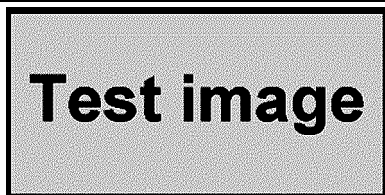


Figure 1: Figure caption

2.7 Ordered and Unordered Lists

Ordered and unordered lists may be nested, any combination is possible. List items may contain images (w/o image captions) and any kind of font style as described above. List item end tags may be left out.

2.7.1 Example

HTML source:

```
<ul>
  <li>Unordered list item 1 - level 1</li>
  <li>Unordered list item 2 - level 1</li>
  <ol>
    <li>Ordered list item 1 - <b>level 2</b> </li>
    <li>Ordered list item 2 - <b><i>level 2</i></b> </i> </li>
  </ol>
  <li>Unordered list item 3 - level 1</li>
</ul>
```

Print-out:

- Unordered list item 1 - level 1
- Unordered list item 2 - level 1
- 1. Ordered list item 1 - **level 2**
- 2. Ordered list item 2 - ***level 2***
- Unordered list item 3 - level 1

2.7.2 Limitations

- List items may not contain tables or definition lists
- Lists may not be placed within table cells or definition lists

2.8 Definition lists

Definition lists are used to output terms on the left side and their definitions on the right side in a table like structure, i.e. a two columns table without a border is generated, whereas the descriptions are left aligned automatically.

<code><d1></code>	starts a definition list
<code><d1 compact></code>	starts a definition list output as a glossary (in combination with a preceding <code><h1 alt="Glossary"></code>)
<code></d1></code>	ends both types of definition list

2.8.1 Example

HTML code:

```
<dl>
  <dt><b>Defintion list term</b></dt>
  <dd>Defintion list description</dd>
</dl>
```

Print-out:

Definition list term	Definition list description
-----------------------------	-----------------------------

2.8.2 Limitations

- Nested definition lists are not allowed!
- Definition lists may not be placed within lists or table cells
- Definition list items may not contain tables or lists

2.9 Tables

Tables may contain images (w/o image captions) and any kind of font style as described above. A table caption may be defined as well.

2.9.1 Example


Row 1 / Col 1 (width = 25%)	<i>Row 1 / Col 2</i> (width = 35%)	Row 1 / Col 3 (width = 40%)
Row 2 / Col 1	Row 2 / Col 2 	Row 2 / Col 3
Row 3 / Col 1	Row 3 / Col 2	Row 3 / Col 3

Table 1: Table width = 80%

2.9.2 Limitations

- End tags may **not** be left out.
- Table cells may only contain images (w/o image captions) and any kind of font style but no other tags
- Table cells may not be merged (ColSpan / RowSpan).
- Only the first row may be used for specifying column widths

3.0 Print formatting with HTML

Since HTML is designed to view documents on the screen, originally there are no tags related to printing. Instead of using new style sheet features which might not be supported by the HTML editor used, general tags which serve a different purpose on the screen are used (e.g. `<HR>` is a horizontal rule on the screen while the DOCDEF takes it as a page break.)

3.1 *Page breaks*

To accomplish page breaks the tag for horizontal rules `<HR>` is interpreted by the DOCDEF as a page break.

3.2 *Images*

To add a sequentially numbered caption below the image the ALT attribute is used (see chapter "Supported tags").

3.3 *Links*

As link texts should not always appear in the printed document the **ALT** attribute is used within the hyperlink tag to toggle between showing the link text (no ALT attribute defined), showing an alternative text (ALT attribute assigned with string), and hiding the link text (empty ALT attribute).

```
<a href="..." alt=""> Link Text </a>
```

hides *Link Text* in the printed document

```
<a href="..." alt="Alternative text"> Link Text </a>
```

shows the **alternative text** defined in the ALT attribute instead of the *Link Text* in the printed document

```
<a href="..."> Link Text </a>
```

shows the *Link Text* in the printed document

3.3.1 Examples

HTML source: ... this completes the installation.
 < a href="details.htm" alt="" >Next chapter
Print-out: ... this completes the installation.

HTML source: For further details < a href="XMLReceiver.htm"
 alt="see chapter "XML Receiver"" >click here
Print-out: For further details see chapter "XML Receiver".

HTML source: ... will be moved into the < a href="XMLReceiver.htm" >
 Completed Queue
Print-out: ... will be moved into the Completed Queue

3.4 Indices

As the definition of index entries is not supported by HTML, the anchor tag is used instead. The contents of the attribute NAME is output in the chapter "Index" along with the page number. If two or more anchors are defined with identical strings in the attribute NAME, the index entry.

3.4.1 Example

HTML code:

Anchor	defined on page
< a name="Agent" > 	7
< a name="Adapter" > 	14
< a name="Agent" > 	14
< a name="Adapter" > 	32
< a name="Agent" > 	48

Print-out:

Index

Adapter	14, 32
Agent	7,14,48
...	...

4.0 Definition and usage of variables

The Javascript feature to define variables and use them in the HTML document is also supported by the DocDef HTM2AFP.DFA. If for instance the same HTML file should be used in different documents where the difference lies only in a few terms (e.g. chapter "Installation of Papyrus *WebControl*" and "Installation of Payprus *WebRepository*") or if the version number for instance appears at several places in the document only a variable at the beginning of the document needs to be changed.

The following example shows how variables are initialized and used in the HTML document:

- Declare and initialize the required variables in the main HTML document (e.g. right before the BODY tag):

```
<SCRIPT LANGUAGE="JavaScript">  
  Product="WebControl";  
  Version="6.00";  
</SCRIPT>
```

- In the HTML source (edited in Notepad or UltraEdit for instance) place the variable (bold text) within SCRIPT tags like in the example below:

Papyrus <SCRIPT>javascript:document.write(**Product**);</SCRIPT> is based on Papyrus Objects ...

- The variable will be resolved by the DocDef HTM2AFP.DFA (bold word):

Papyrus **WebControl** is based on Papyrus Objects ...

Index

<

<A HREF"...">	8
<A NAME"...">	9
	3
 	3
<CAPTION>	7
<DD>	6
<DL>	6
<DT>	6
<H1>	3
<H2>	3
<H3>	3
<H4>	3
<HR>	8
<I>	3
	5, 8
	5
	5
<P>	3
<SCRIPT>	10
<TABLE>	7
<TD>	7
<TR>	7
<TT>	3
<U>	3
	5

A

Anchor	9
--------	---

B

Blockquotes	5
-------------	---

D

Definition lists	6
------------------	---

F

Font styles	3
-------------	---

H

Headings	3
----------	---

I

Images	5, 8
Indents	5
Indices	9
Introduction	1

L

Line breaks	3
Links	8
Lists	5

O

Ordered lists	5
---------------	---

P

Page break	8
Page breaks	3
Print formatting	8

S

Supported tags	2
----------------	---

T

Tables	7
--------	---

U

Unordered lists	5
-----------------	---

V

Variables

10