



## **DTD TOOLS**

*System:* **DTD-TOOLS**

-dtdxpand.exe

-dtd2html.exe

*Version:* **3.4**

*Dated:* **10 november 2003**



|      |                                                 |    |
|------|-------------------------------------------------|----|
| 1    | Installing the program: .....                   | 3  |
| 2    | Upgrading from version 1.X.....                 | 3  |
| 3    | dtd2html: gui version.....                      | 3  |
| 3.1  | Read DTD.....                                   | 4  |
| 3.2  | Output directory .....                          | 4  |
| 3.3  | Title HTML doc .....                            | 4  |
| 3.4  | Catalog file .....                              | 4  |
| 3.5  | Exclude file.....                               | 4  |
| 3.6  | Output options .....                            | 4  |
| 3.7  | Start .....                                     | 5  |
| 3.8  | Result.....                                     | 7  |
| 4    | dtd2html options.....                           | 7  |
| 4.1  | Create Doc with images:.....                    | 7  |
| 4.2  | Remove orphans: .....                           | 8  |
| 4.3  | Use DTD comments .....                          | 9  |
| 4.4  | Use Exclude list.....                           | 9  |
| 5    | dtdxpan: gui version. ....                      | 9  |
| 5.1  | Read DTD.....                                   | 10 |
| 5.2  | Output directory .....                          | 10 |
| 5.3  | Name EXPANDED DTD.....                          | 10 |
| 5.4  | Catalog file .....                              | 10 |
| 5.5  | PI Pattern file.....                            | 10 |
| 5.6  | Start .....                                     | 11 |
| 6    | dtdxpan options .....                           | 11 |
| 6.1  | Create Mini Catalog .....                       | 12 |
| 6.2  | Expand ISO entities:.....                       | 12 |
| 6.3  | Remove DOCTYPE: .....                           | 12 |
| 6.4  | Remove orphans: .....                           | 12 |
| 6.5  | Keep PI's .....                                 | 12 |
| 6.6  | Keep Comments .....                             | 12 |
| 7    | dtd-tools.ini file .....                        | 13 |
| 7.1  | Input dir .....                                 | 13 |
| 7.2  | Filename dtd .....                              | 13 |
| 7.3  | Output dir expand.....                          | 13 |
| 7.4  | Output dir doc.....                             | 13 |
| 7.5  | Title dtd .....                                 | 13 |
| 7.6  | Catalog file .....                              | 13 |
| 7.7  | Logos .....                                     | 14 |
| 7.8  | Accumulate .....                                | 14 |
| 7.9  | Delete orphans .....                            | 14 |
| 7.10 | Expand iso-entities .....                       | 14 |
| 7.11 | Remove DOCTYPE .....                            | 14 |
| 7.12 | Exclude file.....                               | 14 |
| 7.13 | Minicatalog file.....                           | 14 |
| 7.14 | PI Pattern file.....                            | 14 |
| 7.15 | Keep Comments .....                             | 14 |
| 8    | Adding DTD comments to your Documentation ..... | 15 |
| 9    | Starting the programs in batch mode.....        | 16 |
| 10   | Extra Features.....                             | 17 |
| 11   | Oddities .....                                  | 17 |
| 12   | History.....                                    | 18 |
| 13   | Remarks.....                                    | 18 |

## 1 Installing the program:

Installing is just a matter of unzipping the file called **dtd-tools.zip**. Extract all files to a folder of your liking (e.g. c:\dtd-tools)

Extract will unzip the following files in the chosen folder:

- dtdxpan.exe
- dtd2html.exe
- dtd-tools.ini
- company.gif
- collapse.js

After installation there will also be a subdir 'images'. This subdir contains some sample gif files that can be used in conjunction with the dtd2html tool. You don't need this subdir in order to run the program. (See: 5. Options on dtd2html)

## 2 Upgrading from version 1.X

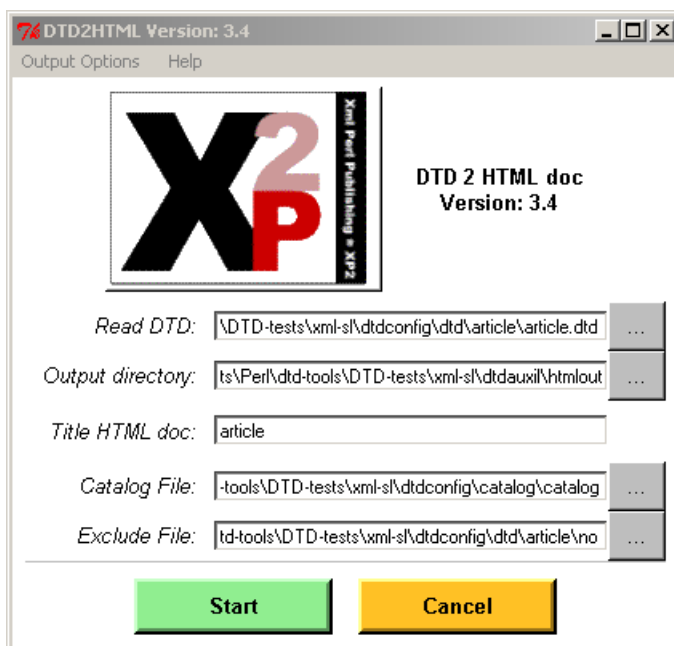
Copy the new executables **dtd2html.exe** and **dtdxpan.exe** over the old ones.

Please do not forget to install the new version of '**collapse.js**' in the directory where you install the program. As the javascripts are no longer inserted in the resulting html files but called in by a link, it is important that the collapse.js that you use has no <script> tags around the script. (as it used to have in the 1.X version)

## 3 dtd2html: gui version.

Before running the program the first time you might want to adapt the default settings in the **dtd-tools.ini** file (see 6. dtd-tools.ini file)

If you want to make a hyperlinked html-documentation of your DTD, start the **dtd2html.exe** program by double clicking it or by typing dtd2html from the command prompt. You will be presented with a window like:





Before pressing start you may want to complete the following choices:

### 3.1 Read DTD

Selects the DTD to document. Clicking the gray button next to the field allows you to select a file. You cannot edit the field that displays your choice.

### 3.2 Output directory

Selects the directory in which the resulting html files will be placed. This is done by clicking the gray button and selecting the directory. You cannot edit the field that displays your choice.

### 3.3 Title HTML doc

This allows you to enter the title of the resulting HTML documentation. This will also be the basename for the resulting HTML files.

### 3.4 Catalog file

If your DTD is modular or uses public entities of a sort, you will have to supply the path to your catalog file in order to resolve the public entities. If public entities are used in your DTD and the catalog file is not given, the tool will not create the requested documentation, but give you an error message.

Click the gray button next to the field and select a catalog file. You cannot edit the field that displays your choice.

Since version 2.1 your catalog file can contain dos-, unix- or unc-style path names.

### 3.5 Exclude file

If you want to exclude some elements from appearing in the final documentation, you can make a file containing the element names of all the elements that you do not want to appear as hyperlinked entries in the HTML documentation.

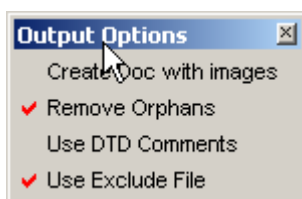
This field allows you to pick the file you created.

The file is a simple text file that one could make using notepad or a similar text editor. Just type in all the element names and make sure to separate them by either white space or a new line. Only the elements that are listed in the file will be excluded (and not their descendants).

You will also have to activate the *Exclude EXCLUDE elements from documentation* option (see 4.4)

### 3.6 Output options

Before starting, check the output options.



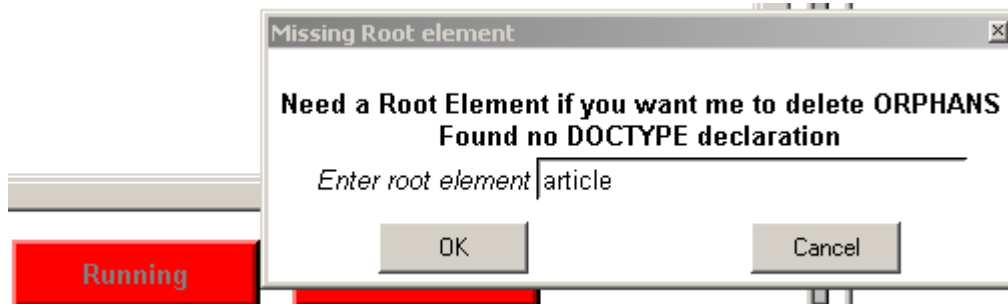
By clicking on these options, you activate them (red checkmark visible) or deactivate them (no checkmark). For more detailed information, see 4.

The existing checkmarks reflect the options of the ini-file (see 7).

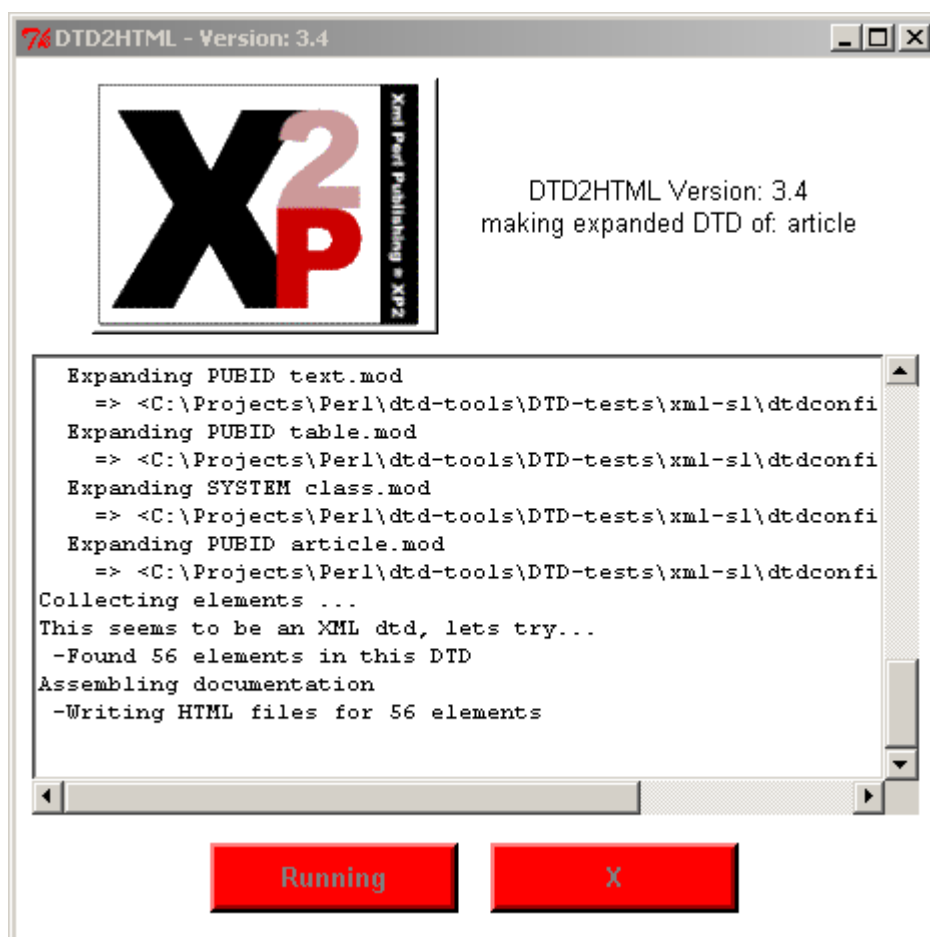
### 3.7 Start

When you completed everything pressing the **Start** button will create your documentation.

When defining the root-element is required (no doctype and remove option activated), the program will assume that the name of the dtd can be used as root element. You do have the option to change this.

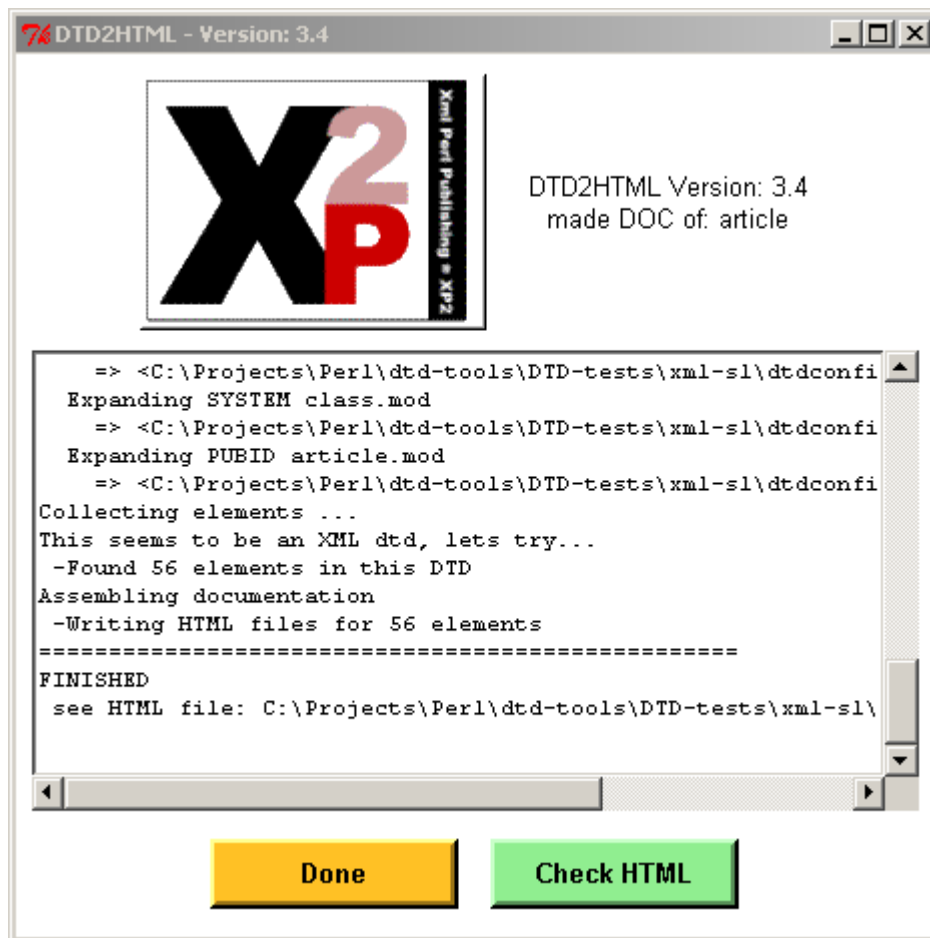


Next, you will be presented a screen similar to:



This screen shows the progress of the program.

When finished, this screen appears:



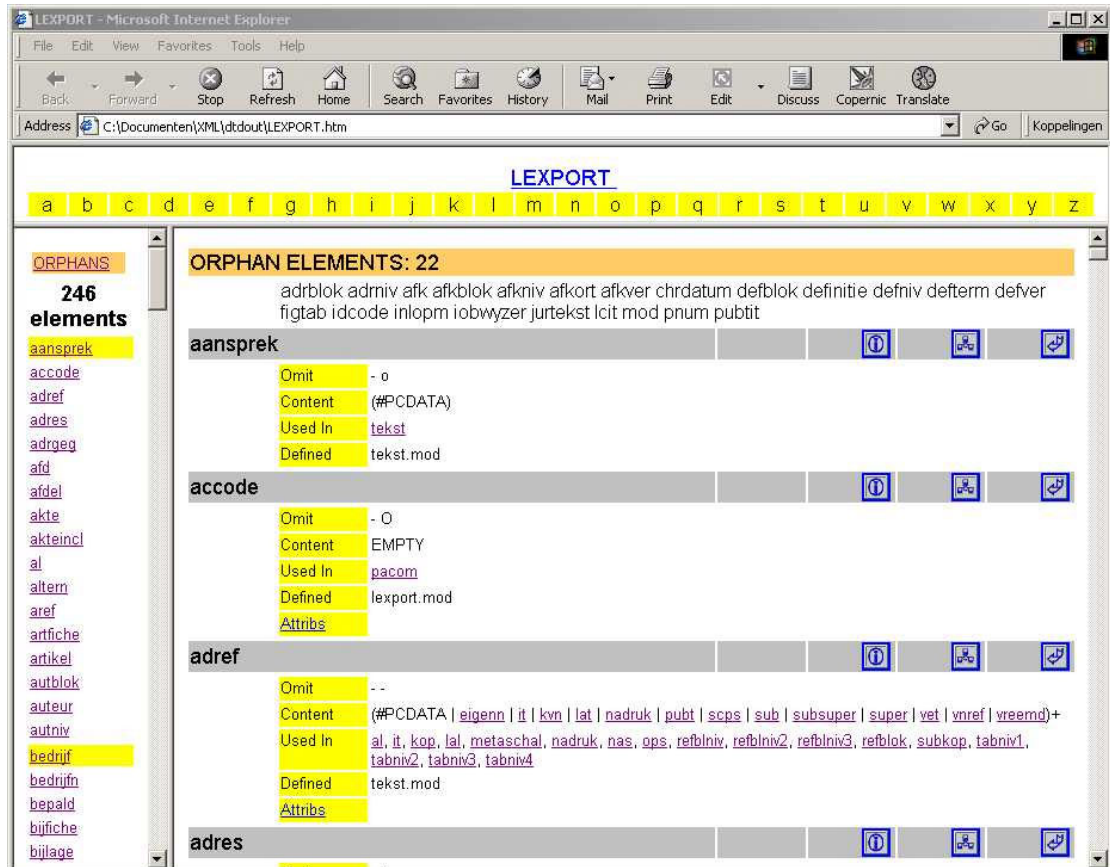
Pressing **Done** will take you back to the first screen where you can make either another selection in order to make another documentation or end the dtd2html program by pressing cancel.

Pressing **Check HTML** will show the resulting HTML documentation you just made in either an existing Internet Explorer window or it will fire up a new IE window.

### 3.8 Result

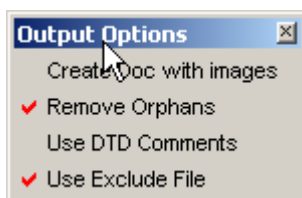
The result will be a hyperlinked HTML document similar to the one show below. It will be created at the *output directory* you have selected and have the name '*title.htm*'. (where *title* stands for the title you have indicated).

In our example it will be: "c:\documenten\xml\dtdout\LEXPORT.htm"



## 4 dtd2html options

When selecting the 'output options' in the menu bar further options can be toggled on or off. This menu can be teared off by clicking on the seperator above the first menu entry.



### 4.1 Create Doc with images:

Selecting this option will create html files that use images as buttons. (see point 2.6). Deselecting it will still create only the 'back' link, but this time as text.

Selecting this option will not copy the appropriate file at the target directory. You will have to do this yourself. The image files have to live at the same directory level as the resulting html files.

Files used:

- back.gif
- blank.gif
- conv.gif
- doc.gif
- info.gif

As you can see there from the previous screenshot of the produced documentation, every element description has 3 buttons associated to it.

The  button is just a shortcut for the 'back' function.

The  button brings you to an 'info' file on each element. If the 'Use Doc Comments' option is 'on' then these files will be created and maintained for you.

The  button will take you to an additional link. This link gives you more info on the conversion to final output of the element. These extra html files, one for each element, (for element <article> called 'articlec.htm') will have to be created manually. The dtd2html tool only creates the links to these files (whether they exist or not)

A sample of these icons is supplied with the program. See subdir: images. You are of course free to use your own.

If the directory where the 'dtd2html' program is started contains a subdir images with the master icons in it, then the program will copy these gif files to the output directory for you.

Documentation produced with the delivered gifs will look like as shown in screen shot of point 3.8.

Documentation produced without the images looks like:

| article                                                                                                                                        |   |   |   |   |   |   |   |   |   |   |   |
|------------------------------------------------------------------------------------------------------------------------------------------------|---|---|---|---|---|---|---|---|---|---|---|
| a                                                                                                                                              | b | c | d | e | f | g | h | i | j | k | l |
| 54 elements                                                                                                                                    |   |   |   |   |   |   |   |   |   |   |   |
| abstract                                                                                                                                       |   |   |   |   |   |   |   |   |   |   |   |
| Content (p)+                                                                                                                                   |   |   |   |   |   |   |   |   |   |   |   |
| Used In article                                                                                                                                |   |   |   |   |   |   |   |   |   |   |   |
| Defined text.mod                                                                                                                               |   |   |   |   |   |   |   |   |   |   |   |
| Attribs                                                                                                                                        |   |   |   |   |   |   |   |   |   |   |   |
| al                                                                                                                                             |   |   |   |   |   |   |   |   |   |   |   |
| Content (##PCDATA   anchor   emphasis   foreign   inline-fig   name   note   noteref   pubtitle   quote   ref   sub   subsuper   super   uri)* |   |   |   |   |   |   |   |   |   |   |   |
| Used In citation, entry, item, p, t-item                                                                                                       |   |   |   |   |   |   |   |   |   |   |   |
| Defined text.mod                                                                                                                               |   |   |   |   |   |   |   |   |   |   |   |

## 4.2 Remove orphans:

This option will remove all elements that have no parent elements from the resulting documentation (although it will enumerate them at the top of the doc - see screen shot in point 2.6).

The program has to know the top element of your DTD, so it will need a valid Doctype declaration to determine this. If you select this option and the selected DTD does not have a doctype declaration at the top, an error will be generated

### 4.3 Use DTD comments

When this option is on, the dtd2html program will maintain the documentation info files for you. These info files are composed of 3 parts:

- the various definitions for this element throughout the various modules and dtd's in which the element is used.
- The various comments that these dtd's contained (see point 8: DTD comments)
- Some local comments that can be manually added to the info files.

### 4.4 Use Exclude list

This option will activate the exclude file (see 2.5). This points to a text file you have to create containing ALL of the elements you want to exclude from becoming hyperlinked entries in your documentation.

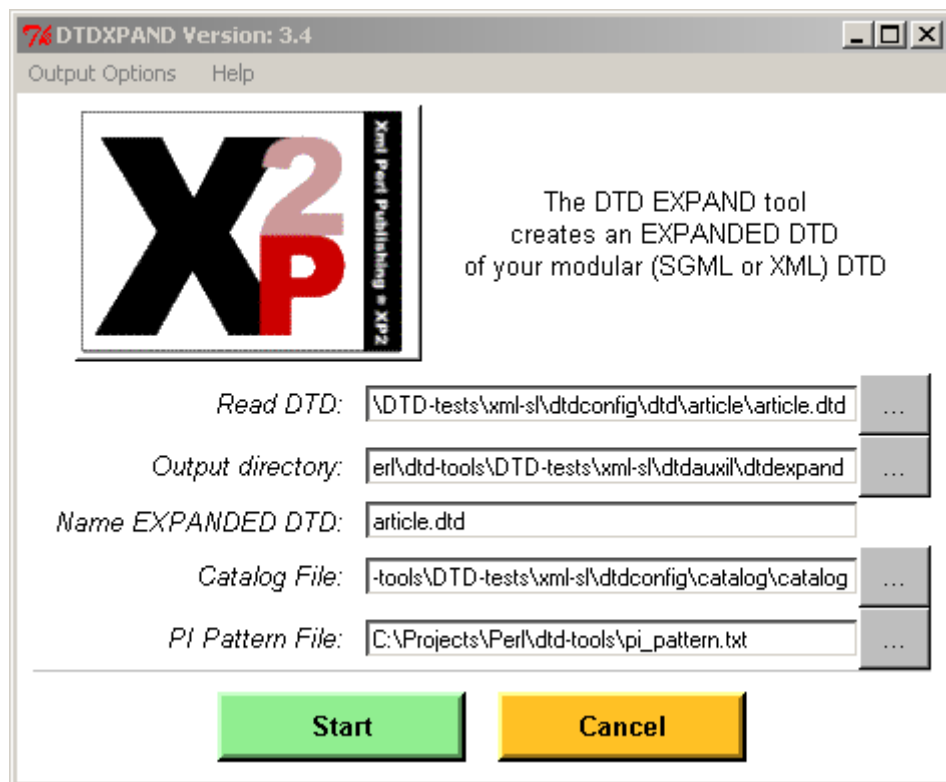
This allows you to skip well-known standard definitions of certain models like: MathML or CALS tables and restrict your documentation to the elements, which are proper to your DTD.

## 5 dtdxpan: gui version.

Before running the program the first time you might want to adapt the default settings in the **dtd-tools.ini** file (see 6. dtd-tools.ini)

If you want to make an expanded version of your modular DTD start the **dtdexpand.exe** program by double clicking on it from explorer or by typing dtdexpand at the command prompt.

You will be presented with a window like:





### 5.1 Read DTD

Selects the DTD for which you want to make the expanded DTD. This is done by clicking the gray button next to the field and selecting a file. You cannot edit the field that displays your choice.

### 5.2 Output directory

Selects the directory in which the resulting expanded DTD file will be placed. This is done by clicking the gray button and selecting the directory. You cannot edit the field that displays your choice.

### 5.3 Name EXPANDED DTD

This allows you to enter the filename of the resulting DTD.

### 5.4 Catalog file

If your DTD is modular or uses public entities of a sort, you will have to supply the tool with your catalog file. Like this it can resolve the public entities. If public entities are called in your DTD and the catalog file is not given then the tool will not create the requested documentation, but give you an error message.

Click the gray button next to the field and select a catalog file. You cannot edit the field that displays your choice.

Since version 2.1 your catalog file can contain dos-, unix- or unc-style path names.

### 5.5 PI Pattern file

If you want to pick up some PI instructions and keep them in your expanded DTD together with the elements in question, you will have to create a PI pattern file, telling the program what kind of PI instructions to look for.

This field allows you to pick the file you created.

The file is a simple text file that one could make using notepad or a similar text editor. Just type in all the patterns and make sure to separate them by either white space or a new line.

If you have defined a pattern like ATTMLINK, the program will look for PI instruction in the format `<?ATTMLINK element_name blablabla?>` and it will put this PI instruction after the element definition and attribute definition of `element_name`.

You will also have to activate the *Keep PI and comments* option in order to activate the PI Pattern file you have chosen.(see 6.5)

## 5.6 Start

When you completed everything pressing the **Start** button will create an expanded DTD. The result will be a single file DTD, which makes its much easier to distribute your DTD.

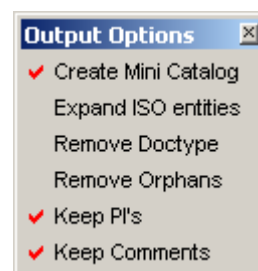
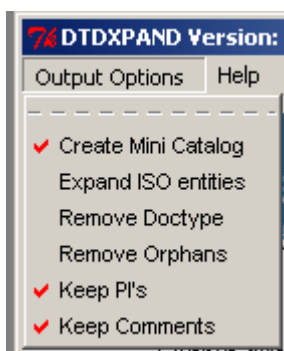
You will be presented a second screen similar to:



Pressing **Done** will take you back to the first screen where you can make either another selection in order to make another documentation or press **Cancel** in order to end the dtd2html program.

## 6 dtdxpand options

When selecting the 'output options' in the menu bar further options can be toggled on or off. This menu can be teared off by clicking on the seperator (dashes) above the first menu entry (left image). After separating it, this becomes a separate window (right image)



### 6.1 Create Mini Catalog

This option will create an catalog file containing the following definitions:

- the SGMLdeclaration (if one can be found in the original catalog)
- the definition for the pubid of the DTD you are creating
- all the definitions for the ISO entities which have not been expanded into the DTD

This mini catalog file is nice to distribute along with the expanded DTD. Just the paths contained in it will probably need some adjusting.

### 6.2 Expand ISO entities:

This option will allow you to include the character entity definitions in the resulting DTD. The definitions included will be the one of the system on which the program is running. Whether using this option is a good idea largely depends on the use of the resulting DTD.

### 6.3 Remove DOCTYPE:

This option will remove the doctype declaration from the resulting DTD.

### 6.4 Remove orphans:

This option will remove all elements that have no parent elements from the resulting DTD (although it will enumerate them as comments at the top of the DTD).

Of course the program has to know the top element of your DTD. It will need a Doctype declaration to determine this. If no doctype is used, the program will ask you for the name of the root-element (it will suggest the name of the DTD, but this can be changed).

If the given root-element does not exist, it will ask again for the root-element.

The resulting DTD will only contain the content models of the actually used elements.

### 6.5 Keep PI's

This option will instruct the program to keep all PI instructions into the expanded DTD.

The PI instructions will be seperated into local and global PI's .

Local PI's are PI instructions that are kept together with the element they refer to. The pattern of these PI instructions are defined in the *PI Pattern File* (see 5.4).

A typical example are the PI's needed for the Sigmalink system which have the form:

```
<?ATTLINK element_name id "SLT" ?>
```

These local PI's can be kept by putting the 'ATTLINK' pattern into the *PI Pattern File*

All other PI instructions are considered to be global and will be bundled and output at the front of the expanded DTD.

### 6.6 Keep Comments

This option will instruct the program to keep local comments into the expanded DTD.

Here again we make the distinction between local and global comments.

Local comments are comments that have the following format:

```
<!-- =element_name bla bla bla -->
```

The difference with the local PI's is that in this case the format is fixed and can not be defined in an external file.

All global comments are ignored and are kept in the expanded DTD.

The local comments will appear in front of the element definition in the expanded DTD.

(element = element\_name).

The comments can be multiline.

One can define multiple local elements per element.



## 7 dtd-tools.ini file

When you become tired of selecting or deselecting certain options or paths over and over, it becomes time to adapt the dtd-tools.ini file to your preferences.

Here is a sample dtd-tools.ini file:

```
input dir:          c:\groep\xml\dtdconfig\dtd      #input dir for dtd
filename dtd:       article.dtd                  #filename of dtd
output dir expand:  \\Ntwk14483\data-tine\bart\     #output dir for expanded DTD's
output dir doc: c:\documenten\xml\dtdout\         #output dir for html doc
title dtd:          article                      #title of dtd
catalog file:       \\Ntwk14483\data-tine\xml\dtdconfig\catalog\catalog.unc
                  #filepath to catalog file (if you need one)
logos:             yes                          #produce html files with links as logo's
accumulate:        yes                          #produce and accumulate documentation from DTD comments
delete orphans: yes #delete orphan elements from html files
expand iso-entities: no #expand iso-entities in final dtd
remove DOCTYPE: no #remove doctype if in source DTD
exclude file:      c:\projects\perl\dtd-tools\dtd-exclude.txt
minicatalog file:  minicatalog
pi pattern file:   C:\Projects\Perl\dtd-tools\pi_pattern.txt
keep comments:     yes
```

As you can see this file can contain a mix of traditional dos path names and unc pathnames. Where one has to specify the path to a file, one can either enter an absolute path or a relative path.

### 7.1 Input dir

Possible values: *Path*

Sets the input directory for both the **dtdexpand** and **dtd2html** tool

### 7.2 Filename dtd

Possible values: *Filename*

Sets the filename of the input dtd for both the **dtdexpand** and **dtd2html** tool.

The combination of input dir + filename dtd will be the start value of the 'Read DTD' field (see 2.1 and 4.1)

### 7.3 Output dir expand

Possible values: *Path*

Sets the initial value of 'output directory' field for the **dtdexpand** tool (see 4.2)

### 7.4 Output dir doc

Possible values: *Path*

Sets the initial value of the 'output directory' for the **dtd2html** tool (see 2.2)

### 7.5 Title dtd

Possible values: *Text*

Sets title of the created HTML file for the **dtd2html** tool or the filename of the resulting DTD of the **dtdexpand** tool

### 7.6 Catalog file

Possible values: *Path to your catalog file*

If a relative path is given, then this will be relative to the **Input dir** you specified.

Since version 2.1 your catalog file can contain dos-, unix- or unc-style path names.



## 7.7 Logos

Possible values: *yes* or *no*

Sets the '*produce HTML with links as graphics*' option on the **dtd2html** tool

## 7.8 Accumulate

Possible values: *yes* or *no*

Sets the '*Use DTD Comments*' option on the **dtd2html** tool

## 7.9 Delete orphans

Possible values: *yes* or *no*

Sets the '*delete orphan elements*' option on both the **dtd2html** and the **dtdexpand** tool

## 7.10 Expand iso-entities

Possible values: *yes* or *no*

Sets the '*expand iso-entities*' option on the **dtdexpand** tool. (see 5.3)

## 7.11 Remove DOCTYPE

Possible values: *yes* or *no*

Sets the '*remove doctype*' option on the **dtdexpand** tool. (see 5.1)

## 7.12 Exclude file

Possible values: *no* or *path to your exclude file*

If a relative path is given, then this will be relative to the **Input dir** you specified.

If you input **no** as the file name, then the exclude file will not be used and the **Use Exclude file** option will be **off**.

## 7.13 Minicatalog file

Possible values: *no* or *path to your minicatalog file*

If a relative path is given, then this will be relative to the **Output dir** you specified.

If you input **no** as the file name, then the minicatalog file will not be created and the **Create Minicatalog** option will be **off**.

## 7.14 PI Pattern file

Possible values: *no* or *path to your PI pattern file*

If a relative path is given, then this will be relative to the **Input dir** you specified.

If you input **no** as the file name, then the PI's and comments will not be kept in the expanded DTD and the **keep PI's** option will be **off**.

## 7.15 Keep Comments

Possible values: *yes* or *no*

If a relative path is given, then this will be relative to the **Input dir** you specified.

If you input **no** as the file name, then the PI's and comments will not be kept in the expanded DTD and the **keep PI's** option will be **off**.

## 8 Adding DTD comments to your Documentation

In order to maintain the info files for the elements automatically one can add 'comments' to the DTD and its modules.

See 4.3 on how to run *dtd2html* in 'Use DTD comments' mode.

These comments have to be structured so the program can recognise them.

A section of the DTD might look like:

```
<!-- =begin html
<h1>subtitle</h1>
<p>The subtitle is used to add an extra explanation to the title. It is always
optional.</p>
<h2>Attributes</h2>
&id-att;
<h3>short</h3>
<p>Very often, the content of the title has an abbreviated version. This may be used for
the table of content, for headers or footers, for labeling references in electronic
versions, etc.</p>
&text-att;
=end html -->

<!ELEMENT subtitle (#PCDATA | anchor | br | emphasis | foreign | inline-fig | name | note
| noteref | pubtitle | quote | ref | sub | supersub | super | url )*>
<!--
  short CDATA #IMPLIED
  %id-att;
  %text-att;
-->
```

Everything that lies between `=begin html` and `=end html` will be looked at by the program and considered to be valid comments if the first line starts with an `h1` tag containing the name of the label of the comment.

This label does not have to be the name of an existing element, but if it is then the comment will be added to the info file of that element.

In most cases, the label will be an existing element.

One can however create labels that don't get attached to elements, but can be used as entities to be used in other comments.

This feature allows you to save a lot of typing work in the case you want to use the same piece of comment over and over again. These boilerplate comments can be called by using an entity construction (eg `&id-att;` in our example)

Important is to know that the info files are 'accumulative'. This means that they will contain info and comments coming from several DTD that have been documented using the *dtd2html* tool.

See below for an example of such a file:

[al in text-lim.mod](#)

**Content** (#PCDATA | [emphasis](#) | [foreign](#) | [inline-fig](#) | [ref](#) | [super](#) | [url](#))\*

**Defined** text-lim.mod

**Attribs**

**Used in:**

article-lim [item](#), [p](#)

[al in text.mod](#)

**Content** (#PCDATA | [anchor](#) | [emphasis](#) | [foreign](#) | [inline-fig](#) | [name](#) | [note](#) | [noteref](#) | [pubtitle](#) | [quote](#) | [ref](#) | [sub](#) | [subsuper](#) | [super](#) | [url](#))\*

**Defined** text.mod

**Attribs**

**Used in:**

article [citation](#), [entry](#), [item](#), [p](#), [t-item](#)

product [citation](#), [entry](#), [item](#), [p](#), [t-item](#)

---

## al in text.mod

The al-element (alineaa) is a wrapper around sentences. It always results in starting at a new line. Comparable with the result of a break-element.

### Attributes

#### ID

The ID attribute gives a unique identifier for this element. This attribute is generated by the system, not by the user.

In the above example one can see that the *al* element has been used in 3 DTD's (article-lim, article and product). There are 2 different content models used: the one as defined in the 'text-lim.mod' module and the one in 'text.mod' module.

The links in the content models and parentlists are redefined so one can jump from info file to info file.

Clicking on the *al in text.mod* link above the content model would bring you to the comment section found for *al* in the module *text.mod*.

The text under the *al in text.mod* heading is coming from comments that were put in the various DTD modules. The text below attributes is coming from an 'entity', in fact the &id-att; used above as an example.

## 9 Starting the programs in batch mode

This mode allows both programs to be run from a bat file.

Like this you can automate recurring batches of DTD documentation or DTD expansion.

And run the bat file every time a change is made to one the modules.

The non GUI mode is run from the command prompt.

You will have to give at least the -b option to the program to make it run in non-gui mode. If the program is started without the -b option, then it will always start in interactive mode

The arguments can be given in any order. And all (except the -b) are optional.

If an argument is not explicitly mentioned then its value will be taken from the dtd-tool.ini file.



```
>dtd2html -b [-i INDIR] [-o OUTDIR] [-d DTD] [-t DTDTitle] [-n]
>dtdxpan -b [-i INDIR] [-o OUTDIR] [-d DTD-orig] [-t DTD-result]
```

For more info on the -i , -o, -d and -t options look at the explanation on running the dtd2html or dtdxpan program.

The -n option allows you to force the program in running with the 'Use DTD Comments' switched off.

### Overview of switches

binary switches

- -b = batch option
- -n = no element documentation extracted from dtd (and/or modules)  
only valid for the dtd2html !

switches with argument

- -i = input directory
- -o = output directory
- -d = name of the original dtd
- -t = name of the expanded dtd/name and title of the html-documentation
- -r = root element

## 10 Extra Features

Program supports the following:

- SGML or XML dtd's (XML schema's are not supported)
- Use of namespace in XML for elements or attributes.
- Use of 'FIXED' in attribute definition will result in having the default value being displayed in the second colon of the attribute definition table.

e.g.: definition like:

```
xlink CDATA #FIXED 'simple'
```

will result in a attribute definition like:

```
xlink CDATA = 'simple' #FIXED
```

## 11 Oddities

The executables 'dtdxpan.exe' and 'dtd2html.exe' are in fact exactly the same.

The program looks at its own name to either start in the documentation mode or the dtd expand mode. As long as the name contains the string 'expand' the program will run in the dtd expand mode. If the name contains 'html' or 'doc' the program will run in the html documentation mode.

Source program was written in Perl5.6 and freestanding executable was made with the PerlApp application that comes with the ActiveState PDK.

You will notice that the first time the program is fired up on your PC it takes longer then normal to start. This is because it first will write a dll file to your temp directory for every module used in the source program. (location of temp dir depends on NT4 or Win2000)

The next time you fire up the program you should see that it start quicker.

## 12 History

- # versie 0.1 - bt 03 nov 00 - small changes - starts from version 5.1 of dtd2html
- # versie 0.2 - bt 03 dec 00 - exclusion/inclusion problem + isochar
- # versie 0.3 - bt 03 dec 00 - optional DOCTYPE
- # versie 0.9 - bt 07 dec 00 - dtd-expand and dtd2html in one prog
- # versie 0.9.1 - bt 9/12/2000 - init of lists
- # versie 0.9.2 - bt 11/12/2000 - readjob of ini file, what if dir does not exist?
- # versie 0.9.3 - bt 29/12/2000 - select dir with drive selection
- # versie 1.0 - bt 17/01/20001 - first production version
- # versie 1.1 - bt 19/01/2001 - small problems
- # versie 1.2 - bt 28/01/2001 - support for namespace
- # versie 1.2.1 - bt 28/01/2001 - problem with system identifiers
- # versie 1.2.2 - bt 25/02/2001 - BadExit problem + relative system identifier
- # versie 1.2.2 - bt 14/03/2001 - more problems with system identifier + orphans
- # versie 1.3 - bt - 14/04/2001 - XML public identifier
- # versie 1.4 - bt - 25/06/2001 - relative paths in catalog
- # versie 1.5 - bt - 2/09/2001 - syntax for fixed attrib
- # versie 1.6 - bt - 2/09/2001 - syntax for attrib definition and quotes in patterns
- # versie 1.7 - bt - 9/09/2001 - changed param-entity pattern
- # versie 1.8 - bt - 21/10/2001 - adaption for WKB Germany
- # versie 1.9 - bt - 20/01/2002 - problem with attlist being skipped
- # versie 2.0 - bt - 23/04/2002 - WISP + Quick Win Q0177 - dtd documentation
- # versie 2.1 - bt - 02/05/2002 - Quick Win 0177 deel 2 - orphans and root element
- # versie 2.2 - bt - 14/05/2002 - inclusion problem in combination with orphans
- # versie 2.3 - bt - 17/05/2002 - system entity problem
- # versie 2.4 - bt - 03/06/2002 - minicatalog en exclude file probleem in dtdxpan
- # versie 2.5n - bt - 19/01/2003 - element doc problem if element defined in main - notation problem
- # versie 3.0 - bt - 25/01/2003 - paths in catalog: relative
- # versie 3.1 - bt - 23/06/2003 - lt and amp entities in comments
- # versie 3.2 - bt - 13/10/2003 - mixed pub/sysid problem in combination with relative paths in catalog + html doc problem
- # versie 3.3 - bt - 02/11/2003 - added sigmalink PI + general comments
- # versie 3.4 - bt - 10/11/2003 - splitted PI and comment options

## 13 Remarks

For any remark or addtional info feel free to contact me at:

[bart@xp2.be](mailto:bart@xp2.be)