



Ditalog - Ideen für die Dokumentation und Online-Hilfe

| | |
|---|-----------|
| Ideen für Dokumentationen..... | 2 |
| E-Mail-Link für Feedback..... | 2 |
| Schriftart für Zahlen..... | 3 |
| Zugang zum Bildschirm voranstellen..... | 5 |
| Zugangsweg im PDF-Output darstellen..... | 5 |
| Zugangsweg in Online-Hilfe darstellen..... | 7 |
| Ideen für Online-Hilfen..... | 9 |
| Kontextsensitive Hilfe per Ditamap..... | 9 |
| Plugin erstellen..... | 10 |
| Seiteninhaltsverzeichnis am rechten Rand..... | 11 |
| Icon Teil 1: Grundform des Icons in Inkscape erstellen..... | 13 |
| Icon Teil 2: ICO-Datei erstellen und einbinden..... | 15 |
| Impressum..... | 17 |

Ideen für Dokumentationen

Ideen und Umsetzungsvorschläge für die Software-Dokumentation, egal ob für den Druck oder Bildschirm.

E-Mail-Link für Feedback

Einen E-Mail-Link für jede Seite als Rückmeldungselement setzen .

Feedback des Nutzers

Durch einen E-Mail-Link auf jeder Seite der Dokumentation wird der Nutzer der Hilfe animiert, eher ein Feedback zur Dokumentation zu geben bzw. speziell Hilfe zur Seite anzufordern.



Abb. 1: Feedback-Element zur Seite

Ein Klick auf den Link öffnet das E-Mailprogramm, in dem der Dateiname der Doku-Seite in der Betreffseite bereits vorgetragen ist.

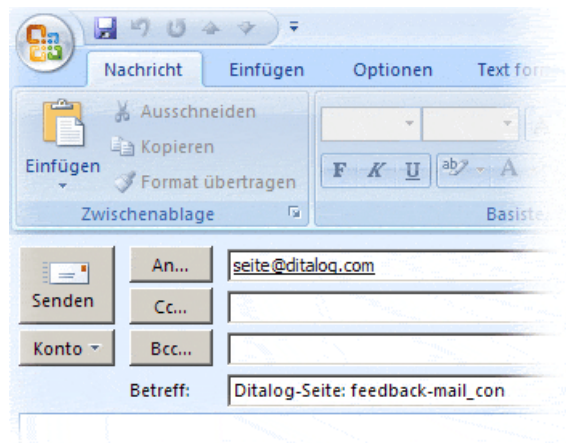


Abb. 2: Adresse und Betreff schon voreingetragen



Abb. 3: Link funktioniert auch im PDF-Output

Attribute des xref-Elements

Die Attribute des xref-Elements sind besonders wichtig: `format="txt"` und `scope="external"` müssen gesetzt sein.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE concept PUBLIC "-//OASIS//DTD DITA Concept//EN"
"http://docs.oasis-open.org/dita/v1.1/OS/dtd/concept.dtd">
<concept id="feedback-mail_con" xml:lang="de-de">
  <title>E-Mail-Link für Feedback</title>

  <abstract><shortdesc>Konzept: Einen E-Mail-Link für jede
Seite als Rückmeldungselement setzen .</shortdesc>
  <p outputclass="quelldatei">Letzte Änderung: 10.09.2009 - E-Mail zur Seite:
  <xref format="txt" href="mailto:seite@ditalog.com?subject=Ditalog-Seite:%20feedback-mail_con"
  scope="external">feedback-mail_con</xref></p></abstract>
```

Abb. 4: Abstract-Element

Beschränkung auf Content-Seiten

Dieser Mail-Link sollte sich möglichst auf Content-Seiten beschränken. Auf Seiten, die nur der Navigation dienen, sorgt dies für Unübersichtlichkeit und ist somit überflüssig. Besonders prekär zeigt sich dies im PDF-Output.

HTML-Format

Probleme und Lösungen zum Output *HTML*.

Letzte Änderung: 02.09.2009 - E-Mail zur Seite: [html_start_con](#)

Plugin tocjs

So installieren Sie das Plugin tocjs und binden es in Ihr HTML-Projekt ein.

Letzte Änderung: 02.09.2009 - E-Mail zur Seite: [tocjs_start_con](#)

Über das Plugin tocjs

Das Plugin erzeugt eine aus- und einklappbare Navigationsleiste in der DITA-OT eingebunden wird. Die Navigation wird durch den Ver...

Abb. 5: Metaangaben der Navigationsdatei HTML-Format sind überflüssig

Spam

Sollte die Hilfe tatsächlich *online* - also im Internet veröffentlicht werden, ist die Veröffentlichung der E-Mailadresse natürlich eine Überlegung wert.

Schriftart für Zahlen

Mit Hilfe der Schriftart CombiNumerals können Sie schnell Nummerierungen durchführen. Die Zahlen werden auf einem Kreis abgebildet (invers) bzw. von einem Kreis umgeben.

Wozu diese Schriftart?

Die Schriftart hilft, schnell kreisrunde Nummerierungen auf Screenshots anzulegen. Wählt man wie im Beispiel die Art **Solid**, werden die Zahlen invers dargestellt. Die Zahl 2 wird erzeugt durch den Buchstaben c. So läßt sich vom Screenshot schnell auf einen ebenfalls durchnummerierten Schritt der Handlungsanleitung referenzieren.

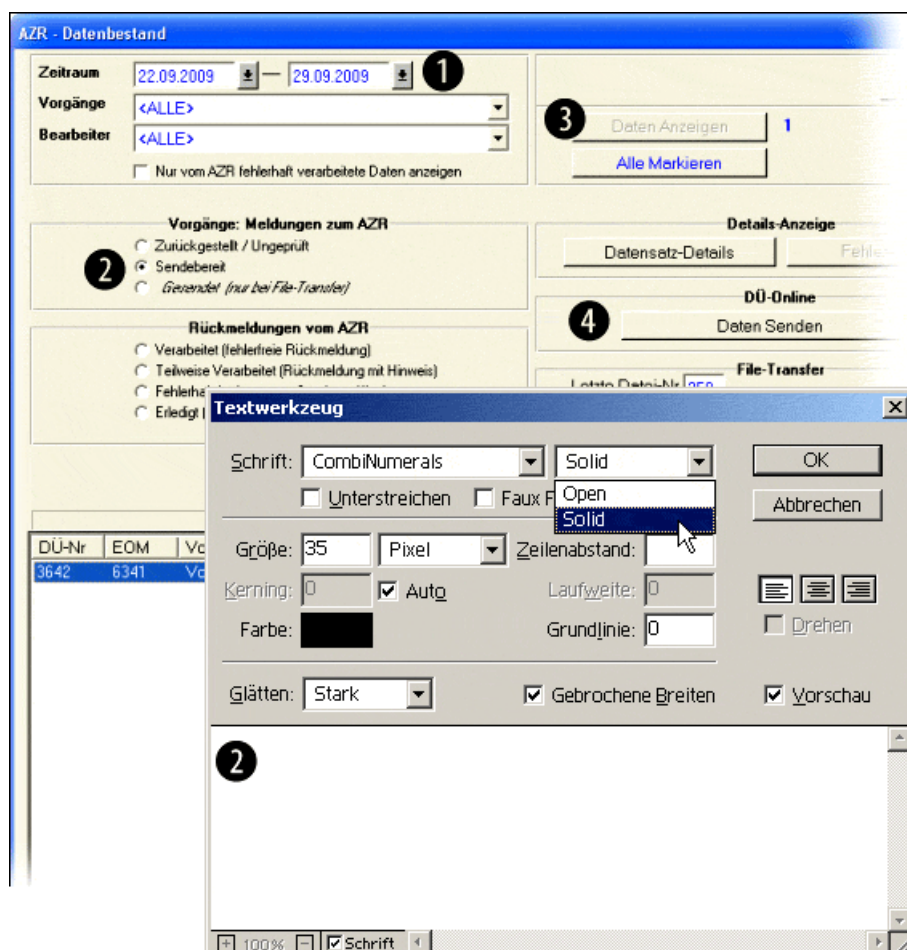


Abb. 6: CombiNumerals in Photoshop

Hinweise vom Autor zum Gebrauch

Den beiden TTF-Dateien ist eine Readme-Datei zum Gebrauch sowie der Installation beigelegt. Hier die Hinweis zum Gebrauch der Schrift:

CombiNumerals(tm) is a typeface for creating circled numbers, popular for use in Web graphics, documentation and instructional material, maps, signs and guide books, for example. With CombiNumerals, you can create any number between 0 and 99. Here's how:

SINGLE-DIGIT NUMBERS - The numbers 0 through 9 are created by pressing the A, B, C, D, E, F, G, H, I, and J keys respectively (you can type either lower- or uppercase).

DOUBLE-DIGIT NUMBERS - The numbers 00 through 99 are created by first pressing one of the number keys (0-9), followed by Shift-number. For example, to create the number 67, press the 6 key followed by Shift-7. You can also create two-digit circled numbers with a leading zero, 07 (0, Sh-7), for example.

MULTIPLE WEIGHTS - CombiNumerals is available in two weights -- Open and Solid. The Open weight displays numbers inside hollow circles. The Solid weight displays white numbers inside filled circles. The Solid weight is the Bold version of the Open weight.

ADDITIONAL CHARACTERS - Several other characters besides numerals are available in this font. For instance, arrows, hands, pointers, and the Mac OS and Windows logos (including circled M and W characters) can be created. These characters are commonly used by people writing documentation, but are also used for things like maps, signs, posters, guide books, etc.

Der Autor

CombiNumerals was designed and created by Sean Cavanaugh, the author of Digital Type Design Guide, The Page Designer's Guide to Working with Type, published by Hayden Books (ISBN 1-56830-190-1). You can contact the author via e-mail <seanc@compuserve.com>.

Links zum Artikel

1. [Download der kostenlosen Version von CombiNumerals von www.maxfonts.com](#)
2. [Buch des Autors mit Schrifterweiterung](#)

Zugang zum Bildschirm voranstellen

Es soll der Zugang zum jeweiligen Bildschirm der eigentlichen Handlungsanweisung vorangestellt und somit extra aufgeführt werden.

Handbuch vs. kontextsensitiver Hilfe

Es gibt zweierlei Arten des Lesens einer Online-Hilfe: linear als Buch oder per Verknüpfung aus der Anwendung. Und immer dann, wenn letzteres geschieht, also eine Handlungsanleitung gezielt durch die kontextsensitive Hilfe geöffnet wird, bedarf es keiner Beschreibung mehr, wie man zum jeweiligen Bildschirm gelangt ist.

Dies ist nur bei Online-Hilfen relevant, jedoch kann man dies getrost auch in Handbüchern übernehmen. Der Aufwand, beim PDF-Output den Zugang zum Bildschirm in die normale Handlungsanleitung als Step zu integrieren, wäre zu hoch. Denn meistens wird man diese immerwiederkehrende Arbeitsschritte per conref-Mechanismus zentral pflegen und einfügen.

Vorschlag einer Zugangsauszeichnung

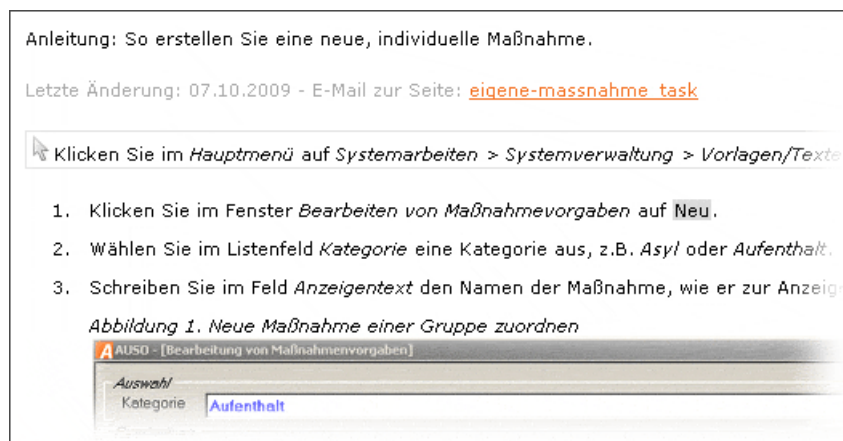


Abb. 7: Zugangsbeschreibung zum beschriebenen Bildschirm auskoppeln

Nutzer, die durch die kontextsensitive Hilfe zum Hilfetopic geleitet wurden, können gleich mit Schritt 1 anfangen.

Zugangsweg im PDF-Output darstellen


Anleitung: So koppeln Sie den Zugangspfad zu einem Bildschirm aus den Handlungsschritten einer Handlungsanleitung aus und ändern entsprechend das XSL-Stylesheet.

1. Der Hinweis auf den Zugangsweg zum Bildschirm soll mit einem Cursor-Icon und dem Wort *Zugang*: gekennzeichnet werden.

BZR-Dateisyntax festlegen

Anleitung: So ändern Sie den Dateinamen der BZR-Datei.

Letzte Änderung: 25.02.2010 - E-Mail zur Seite: [bzs-datei](#)

 **Zugang:** Klicken Sie im *Hauptmenü* auf *System*

1. Die Dateisyntax kann sowohl für die Ausländer- als auch für die Syntax in der Staatsangehörigkeitsbehörde geben. Eintrag *Einbürgerungsbehörde*.
2. Aktivieren Sie das Kontrollkästchen *Dateiname*

Abb. 8: Cursor-Icon vorm Zugangspfad

2. Kopieren Sie das Bild, mit welchem Sie den Zugangspfad "flaggen" möchten, in den Idiom-FO-Plugin-Ordner `C:\DITA-OT1.4.2.1\demo\fo\Customization\common\artwork`
3. Fügen Sie in der Datei `custom.xsl` im Ordner `\demo\fo\Customization\fo\attr\custom.xsl` folgendes Attributset hinzu:

```
<xsl:attribute-set name="p.zugang">
<xsl:attribute name="font-weight">bold</xsl:attribute>
</xsl:attribute-set>
```

4. Fügen Sie in der Datei `custom.xsl` im Ordner `\demo\fo\Customization\fo\xsl` folgenden Eintrag für ein Template hinzu:

```
<xsl:template match="*[contains(@class, ' topic/p ')]">
<xsl:choose>
<xsl:when test="@otherprops='zugang'">
<fo:block space-before="0.5em">
<fo:inline><fo:external-graphic src="url({concat($artworkPrefix,
'/Customization/OpenTopic/common/artwork/icon_cursor.gif')})"/></fo:inline>
<fo:inline xsl:use-attribute-sets="p.zugang"><xsl:text> Zugang:
</xsl:text></fo:inline>
<xsl:apply-templates/>
</fo:block>
</xsl:when>
<xsl:otherwise>
<fo:block xsl:use-attribute-sets="p" id="{@id}">
<xsl:apply-templates/>
</fo:block>
</xsl:otherwise>
</xsl:choose>
</xsl:template>
```

Neu ist hier die `when`-Anweisung. Die `otherwise`-Anweisung ist das normale Template für einen Absatz.

5. Versehen Sie den Absatz, der den Zugangsweg beinhalten soll, mit dem Attribut `otherprops="zugang"`. Der obige Screenshot wurde über einen Absatz im `prereq`-Tag realisiert:

```
<prereq>
<p id="konfigbzs" otherprops="zugang" outputclass="zugang">Klicken Sie
im <wintitle>Hauptmenü</wintitle> auf
<menucascade>
<uicontrol>Systemarbeiten</uicontrol>
<uicontrol>Konfiguration Module</uicontrol>
<uicontrol>BZR.</uicontrol>
</menucascade></p>
</prereq>
```

6. Optional: Weitere Überlegung sollte sein, diesen Zugangsweg nur einmal zu editieren und dann per `conref`-Mechanismus in die jeweilige Handlungsanleitung zu ziehen. Falls sich doch mal ein Bildschirm oder Menü ändert, brauchen Sie nur in der Originaldatei den Pfad oder ähnliches ändern und durch Content Referenzierung wird die Änderung in sämtlichen Handlungsanleitungen übernommen.

- Optional: Falls Sie wie im Beispiel um den Zugangsweg einen Rahmen ziehen möchten, vergeben Sie neben dem `otherprops`-Attribut noch ein `outputclass`-Attribut und erstellen in der CSS-Datei eine Deklaration gleichen Names.

Zugangsweg in Online-Hilfe darstellen

Anleitung: So koppeln Sie den Zugangspfad zu einem Fenster aus den Handlungsschritten aus und kennzeichnen den Zugang per `ditaval`-Datei.

- Der Hinweis auf den Zugangsweg zum Fenster soll durch Flagging in der `ditaval`-Datei mit einem Cursor-Icon gekennzeichnet werden.

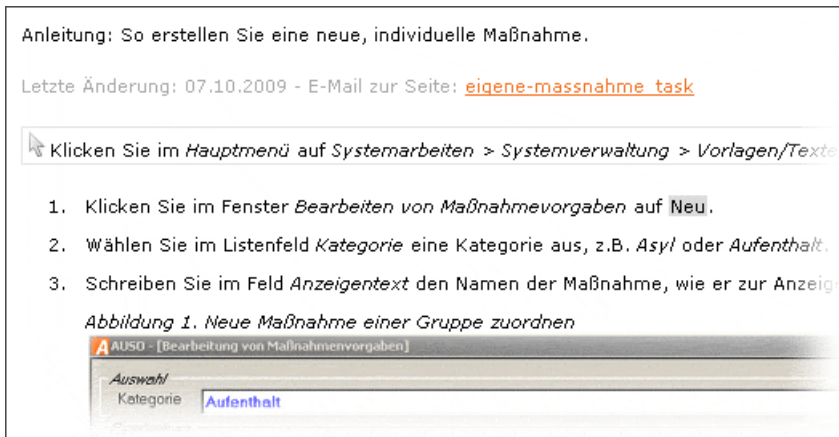


Abb. 9: Cursor-Icon vorm Zugangspfad

- Schreiben Sie dazu folgenden Eintrag in die `ditaval`-Datei:

```
<prop att="otherprops" val="zugang" action="flag">
  <startflag imageref="images/icon_cursor.gif"><alt-text>So kommen Sie
  hin.</alt-text></startflag>
</prop>
```

Bitte achten Sie auf die Pfadangabe bei `imageref`. Sie muss genau übernommen werden. Im Beispiel liegt die `ditaval`-Datei und das Bild in eigenen Verzeichnissen unterhalb des Output-Hauptverzeichnisses:

```
\filters\zugang.ditaval
\images\icon_cursor.gif
```

- Erstellen Sie eine Topic-Datei, in der Sie das Bild `icon_cursor.gif` einbinden.
- Binden Sie diese Topic-Datei (kann auch ein Concept u.a. sein) in die `ditamap`-Datei ein und stellen das Attribut `toc=no` ein. Dieses Einbinden garantiert, dass das `icon_cursor.gif` rechtzeitig! in den Output-Bilderordner kopiert wird.
- Versehen Sie den Absatz, der den Zugangsweg beinhalten soll, mit dem Attribut `otherprops="zugang"`. Der obige Screenshot wurde über einen Absatz im `prereq`-Tag realisiert:

```
<prereq>
<p id="konfigbZR" otherprops="zugang" outputclass="zugang">Klicken Sie
im <wintitle>Hauptmenü</wintitle> auf
<menucascade>
<uicontrol>Systemarbeiten</uicontrol>
<uicontrol>Konfiguration Module</uicontrol>
<uicontrol>BZR.</uicontrol>
</menucascade></p>
</prereq>
```

- Optional: Weitere Überlegung sollte sein, diesen Zugangsweg nur einmal zu editieren und dann per `conref`-Mechanismus in die jeweilige Handlungsanleitung zu ziehen. Falls sich doch mal ein Bildschirm oder Menü ändert, brauchen Sie nur in der Originaldatei den Pfad oder ähnliches ändern und durch Content Referenzierung wird die Änderung in sämtlichen Handlungsanleitungen übernommen.

7. Optional: Falls Sie wie im Beispiel um den Zugangsweg einen Rahmen ziehen möchten, vergeben Sie neben dem `otherprops`-Attribut noch ein `outputclass`-Attribut und erstellen in der CSS-Datei eine Deklaration gleichen Names.

Ideen für Online-Hilfen

Ideen und Umsetzungsvorschläge für Online-Hilfen.

Kontextsensitive Hilfe per Ditamap

Eine Ditamap hilft, Hilfeseiten für die kontextsensitive Hilfe relativ einfach zu erstellen.

Vorteile des Map-Processings nutzen

Die Elemente `<title>` und `<shortdesc>` von eingebetteten Seiten werden immer dann als Vorankündigung angezeigt, wenn ein `<topicref>` andere `<topicrefs>` umschließt. Dieses normale ditamap processing kann sehr gut für die einfache und schnelle Erstellung von kontextsensitiven Hilfeseiten genutzt werden.

Beispiel einer Hilfeseite

Diese Seite wird extra zum Aufruf aus dem Programm über F1 bzw. der Hilfe-Schaltfläche geschrieben. Es ist eine leere Seite. Nur Titel und shortdesc sowie die gebräuchlichsten Links zu den Neuigkeiten in den `related links` wurden gefüllt. Die 4 Links in der Mitte entstehen automatisch durch das map processing.

Einbürgerung Startbildschirm

Kontextsensitive Hilfe - eine Zusammenstellung passender Anleitungen, Konz

Letzte Änderung: 08.09.2009 - E-Mail zur Seite: [csh_ebgmain_con](#)

[Neue Akte anlegen](#)
Anleitung: So legen Sie eine neue Akte an. Sie legen immer dann eine Programm haben. Dies ist sehr häufig in Landkreisen der Fall.

[Akte aus dem Ausländerregister importieren](#)
Anleitung: So legen Sie eine neue Akte an, in dem Sie die Daten des

[Akte suchen](#)
Anleitung: So suchen Sie eine bereits angelegte Einbürgerungsakte.

[Bedienelemente des Startbildschirms](#)
Referenz: Sämtliche Felder und Schaltflächen im Startfenster *Einbürge*

Übergeordnetes Thema: [Kontextsensitive Hilfe](#)

Zugehörige Konzepte
[Was ist neu im Programm?](#)
[Was ist neu in der Dokupedia?](#)

Abb. 10: Kontextsensitive Hilfe per ditamap

Ditamap für die Links

Hier das Beispiel der ditamap für die Link-Generierung. Hier werden die Seiten im Navigationsbaum (TOC) mit angezeigt. Ändern Sie `toc="no"`, kann man nur aus dem Programm auf diese Seite gelangen. Obiges Beispiel ist hier die Datei `csh_ebgmain_con.xml`. Sie umschließt in der Ditamap die 4 Dateien, die als Link oben zur Anzeige kommen. Auf die `csh_ebgmain_con.xml` wird von der Software per F1-Druck verlinkt. Die Datei `csh_start_con.xml` dient lediglich als Anzeige im TOC.

```

<?xml version="1.0" encoding="UTF-8"?>
<DOCTYPE map PUBLIC "-//OASIS//DTD DITA Map//EN"
"http://docs.oasis-open.org/dita/v1.1/OS/dtd/map.dtd">
<map base="" id="csh" scope="local" title="Kontextsensitive Hilfe" toc="yes"
  xml:lang="de-de">
  <topicref href="../../topics/civiso/csh/csh_start_con.xml" scope="local"
    type="concept" xml:lang="de-de">
  <topicref href="../../topics/civiso/csh/csh_ebgmain_con.xml" scope="local"
    type="concept" xml:lang="de-de">
  <topicref href="../../topics/civiso/stammdaten/akte-anlegen_task.xml"
    scope="local" toc="no" type="task" xml:lang="de-de"/>
  <topicref href="../../topics/civiso/stammdaten/akte-importieren_task.xml"
    scope="local" toc="no" type="task" xml:lang="de-de"/>
  <topicref href="../../topics/civiso/stammdaten/akte-suchen_task.xml"
    scope="local" toc="no" type="task" xml:lang="de-de"/>
  <topicref href="../../topics/civiso/ui-ebgmain_ref.xml" scope="local"
    toc="no" type="reference" xml:lang="de-de"/>
</topicref>
</topicref>
</map>

```

Abb. 11: Sichtbarmachen in der Navigation?

ditamap einbinden in Haupt-Map

Da eine einzelne Map für ein Projekt doch sehr unübersichtlich werden würde, wurde sie im obigen Beispiel in eine Extra-ditamap geschrieben. Diese ditamap muss nun in die Haupt-Map eingebunden werden. Hier ist ebenfalls zu entscheiden, ob die Map für die kontextsensitive Hilfe in der Navigation erscheinen soll.

```

<topicref format="ditamap" href="maps/civiso/auswertungen-civiso.ditamap"
  toc="yes" xml:lang="de-de"/>
<topicref format="ditamap" href="maps/civiso/schnittstellen.ditamap"
  toc="yes" xml:lang="de-de"/>
<topicref format="ditamap" href="maps/civiso/csh_civiso.ditamap"
  toc="yes" xml:lang="de-de"/>
<topicref format="ditamap" href="maps/civiso/anhang-civiso.ditamap"
  toc="yes" xml:lang="de-de"/>

```

Abb. 12: Einbinden in die Haupt-ditamap

Plugin erstellen

Konzept: Mit Hilfe eines Plugin individuelle Einstellungen auslagern.

Als Plugin sichern und weitergeben

Irgendwann wollen Sie Ihre individuellen Einstellungen in den Toolkit-XSL-Stylesheets vor einem Update des DITA-OT sichern oder diese an andere weitergeben. Dann ist die Auslagerung Ihrer Templates in ein Plugin angebracht.

Override-Stylesheet-Datei erstellen

Erstellen Sie eine neue XSL-Datei. In den Kopf und Fuß der neuen Datei, die die Templates des Toolkits außer Kraft setzen soll, müssen die Standardeinträge:

```

<?xml version="1.0" encoding="UTF-8" ?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<!-- Hier kommt Ihr angepasstes Stylesheet hinein.-->

</xsl:stylesheet>

```

Zu änderndes Stylesheet in Override-Datei kopieren

Kopieren Sie aus der Datei `DITA-OT\xsl\xslhtml\dita2htmlImpl.xml` dasjenige Template in die soeben neu erstellte Override-Stylesheet-Datei, welches Sie ändern möchten. Suchen Sie nach dem Template immer nach seinem class-Attribut:

```
<xsl:template match="*[contains(@class, 'topic/beispielelement')] ]
```

Wenn Sie das Stylesheet kopiert und geändert haben, speichern Sie die Override-Stylesheet-Datei (`beispiel.xml`) ab. Stellen Sie nun den Originalzustand der Datei `dita2htmlImpl.xml` wieder her, falls Sie darin Ihre Anpassungen vorgenommen hatten.

Plugin-Verzeichnis erstellen

1. Erstellen Sie einen neuen Ordner im Verzeichnis `DITA-OT\demo\beispielname`.
2. Erstellen Sie im neu erstellten Ordner einen neuen Ordner `\xsl`.
3. Kopieren Sie die neue Override-Stylesheet-Datei (`beispiel.xml`) in den Ordner `\xsl`.
4. Erstellen Sie eine neue XML-Datei namens `plugin.xml`:

```
<?xml version="1.0" encoding="UTF-8"?>
<plugin id="org.dita.beispielname">
<feature extension="dita.xml.xhtml" value="xsl/beispiel.xml" type="file"/>
</plugin>
```

und kopieren diese in den Ordner `\demo\beispielname`.

Plugin integrieren

Doppelklicken Sie die `startcmd.bat`, um ein DITA-Kommandofenster zu öffnen und führen Sie folgenden Befehl aus:

```
ant -f integrator.xml
```

Im Ergebnis sollte in der Datei `DITA-OT\xsl\dita2xhtml.xml` die `beispiel.xml` über ein `<xsl:import>` referenziert werden.

Seiteninhaltsverzeichnis am rechten Rand

Bei Concepts sollen die Überschriften jeder section am rechten Rand als kleines Inhaltsverzeichnis dieser Konzept-Seite erscheinen.

Auf der Suche nach FAQ-Template

Bisher hatte ich für jede Frage der FAQ-Liste eine eigene Concept-Datei erstellt. Dies war jedoch zu viel Arbeit. Es sollte eine Ein-Datei-Lösung her. Oben die Fragen als Link, weiter unten dann die Frage/Antwort-Einträge. Was lag näher, als die Demo-FAQ-Spezialisierung im `DITA-OT\demo`-Ordner anzuschauen? Jedoch nach einem Tag gab ich es auf. Die Pfade in der `build_demo.xml` und der `faq2html_shell.xml` zum DTD-Verzeichnis stimmten nicht, da die Verzeichnisse im DITA OT 1.5 neu geordnet wurden. Auch glaube ich, dass mit jeder neuen FAQ-Datei, die man erstellen möchten, man diese im build-Script erwähnen muss.

Nun weiß ich, warum ich keine Beispiele zu dieser FAQ-Spezialisierung gefunden habe: sie ist nicht dokumentiert und in der Handhabung etwas umständlich.

XSL-Stylesheet in der Dokumentation

Natürlich habe ich versucht, selbst XSL-Stylesheet-Anpassungen zu kreieren. Das Ergebnis: keine Fehlermeldung und kein Inhaltsverzeichnis. Bis ich in der Datei `DITA-OT\xsl\xslhtml\dita2htmlImpl.xml` auf den Eintrag

```
<xsl:template name="gen-user-sidetoc">
```

stieß. Schnell danach gegoogelt und hier die Rettung in der neuen Toolkit-Dokumentation und einem XSL-Tutorial gefunden. Hier das Beispielskript vom DITA-OT:

```
<!-- override for main stub -->
<xsl:template name="gen-user-sidetoc">
<!-- if there are nested topics... -->
<xsl:if test="descendant::*[contains(@class,' topic/topic ')]">
<p><table width="150" align="right" border="1" frame="box" rules="none">
<tr><td height="5" bgcolor="#0033CC" align="center">
<b><font color="#FFFFFF">Contents:</font></b></td></tr>
<xsl:for-each select="descendant::*[contains(@class,' topic/topic ')]">
<xsl:variable name="ttext"><xsl:value-of select="*[contains(@class,'
topic/title ')]"/>
</xsl:variable> <tr><td class="toc">
- <a href="#{generate-id()}"><xsl:value-of select="$ttext"/></a>
<!--recursive call for subtopics here/--> </td></tr>
</xsl:for-each></table></p></xsl:if>
</xsl:template>
```

XSL-Stylesheet individuell anpassen

Durch *trial and error* habe ich das Template derart angepaßt, dass immer nur in Concepts rechts oben ein Seiteninhaltsverzeichnis erscheint, welches die *section titles* - sprich die Überschriften zu jeder Sektion - auflistet.

| Seitenübersicht |
|---|
| Auf der Suche nach |
| FAQ-Template |
| XSL-Stylesheet in der Dokumentation |
| XSL-Stylesheet individuell anpassen |

Abb. 13: Die Sektionsüberschriften am rechten Rand des Concepts

```
<xsl:template name="gen-user-sidetoc">
<xsl:if test="descendant::*[contains(@class,' topic/section
')]/*[contains(@class,' topic/title ')]">
<p><table class="table" width="200" align="right" border="0">
<tr><td class="tablehead" align="center">Seitenübersicht</td></tr>
<xsl:for-each select="descendant::*[contains(@class,' topic/section
')]/*[contains(@class,' topic/title ')]">
<xsl:value-of select="*[contains(@class,' topic/title ')]"/>
<tr><td><a href="#{generate-id()}"><xsl:value-of
select="text()"/></a></td></tr>
</xsl:for-each>
</table></p>
</xsl:if>
</xsl:template>
```

Anker generieren

Es mußte noch ein Template her, um Namensanker zu erstellen.

```
<xsl:template name="id">
<a name="{generate-id()}" />
</xsl:template>
```

Dieses Anker-Template muss nun innerhalb des Template für die *section-titles* aufgerufen werden, damit in jeder *section* ein Namensanker generiert wird:

```
<xsl:call-template name="id" />
```

Stylesheet auslagern in ein Plugin

Damit die geänderte Datei `dita2htmlImpl.xsl` nicht beim nächsten DITA-OT-Update verloren geht, ist es ratsam, das `gen-user-sidetoc`-Template in ein Plugin auszulagern. Sie können das Plugin `tocofsections` auf der Startseite www.ditalog.com downloaden.

Sections und ePub

Bei der Generation von ePUB-Dateien wirkt jedoch die Section-Übersicht am rechten Rand als störend. Dann ist es besser, einfach den Plugin-Ordner herauszunehmen bzw. umzubenennen.

Icon Teil 1: Grundform des Icons in Inkscape erstellen

Anleitung: So erstellen Sie die Grundform des Icons für die Online-Hilfe in verschiedenen Größen.

Wenn man nicht gerade Webseiten und Online-Hilfen am Fließband erstellt, kommt man als Technischer Redakteur nicht so oft in die Verlegenheit, ein Favicon erstellen zu müssen. Als ich zum zweiten Mal vor der Aufgabe stand, hatte ich alles bisherige zur Iconerstellung wieder vergessen. Für mich und für manch weiteren Redakteur nun die Anleitung fixiert.

1. Erstellen Sie im Open Source Graphikprogramm *Inkscape* eine Grundform Ihres Icons. Bestimmt hilft Ihnen wie mir ein Graphiker weiter.
2. Es gilt, Icons für die Größen 16, 32, 48 und 128 px zu erstellen. Erstellen Sie ein Quadrat mit 32px Breite und Höhe.

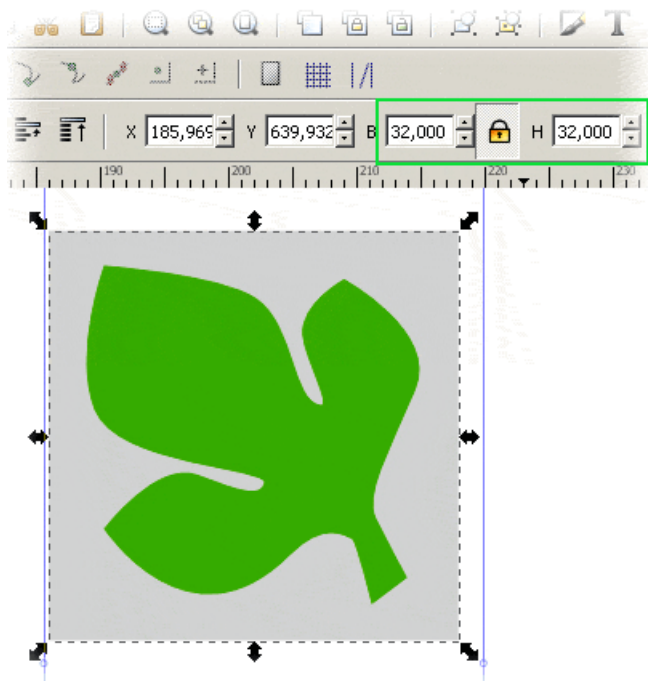


Abb. 14: Breite und Höhe des Quadrats einstellen

3. Bestimmen Sie Farbe der Füllung und der Kontur des Quadrats. Soll das Icon transparent sein, wählen Sie links die Kreuzfarbfläche für "keine Farbe".
4. Optional: Soll das Icon und somit diese Hintergrundquadrat abgerundete Ecken haben, markieren Sie das Rechteck. Anschließend klicken Sie auf das **Rechteck-/Quadrat-Tool** in der Werkzeugleiste und ziehen den Anfasser in Kreisform in der rechten oberen Ecke in die diagonale, linke untere Ecke.

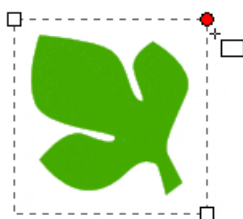


Abb. 15: Mit Rechtecktool die Ecken abrunden

5. Falls Ihnen das Icon nicht plastisch genug erscheint, fügen Sie der Icon-Grundform einen **Farbverlauf** zu.

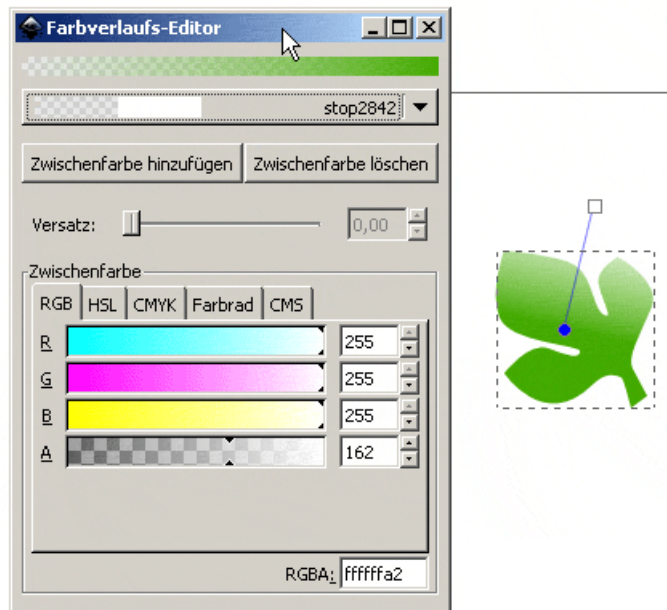


Abb. 16: Mit dem Farbverlaufs-Tool arbeiten

6. Positionieren Sie das zuerst gezeichnete Objekt der Grundform in die Mitte des Hintergrundquadrats.
 7. Markieren Sie beide Objekte und klicken Sie auf **Objekt > Gruppieren**.
 8. Klicken Sie auf **Datei > Bitmap exportieren**.



Abb. 17: Bitmap mit exakter Breite exportieren

9. Klicken Sie im Fenster **Bitmap exportieren** auf **Auswahl**.
 10. Kontrollieren Sie, ob in den Feldern **Breite** und **Höhe** die korrekten Pixelwerte hinterlegt sind.
 11. Optional: Hinterlegen Sie im Feld **Dateiname** einen Ausgabepfad.
 12. Klicken Sie auf **Exportieren**. Es wird im ausgewählten Pfad eine PNG-Datei erzeugt.
 13. Wiederholen Sie die Schritte für die Größen 16, 48, 128 und 256 px.

Links zum Artikel

1. [Inkscape von heise.de downloaden](#)

Icon Teil 2: ICO-Datei erstellen und einbinden

Anleitung: So erstellen Sie mit Hilfe des Freewareprogramms Icon Sushi aus mehreren Graphiken eine ICO-Datei und binden Sie in die HTML-Seiten ein.

Sobald Sie Icon-Graphiken in den Größen 16, 32, 48 und 128 px haben, können Sie daraus eine Icondatei erstellen. Je nach Erfordernis erscheint das jeweilige Icon in der richtigen Größe. So z.B. auf dem Desktop als Verknüpfungssymbol oder im Internetbrowser an diversen Stellen wie z.B. in der URL-Adresszeile. Das Open-Source-Programm *Icon Sushi* hilft bei der ICO-Dateierstellung.



Abb. 18: Icon im Browser Firefox

1. Klicken Sie im Open-Source-Programm *Icon Sushi* auf die Symbolschaltfläche **Öffnen** und laden Sie sämtliche Icon-Graphiken ins Programm. Sie können mehrere markieren und im Block ins Programm laden. Das Icon der Größe 256 px kann nicht ins Programm importiert werden.

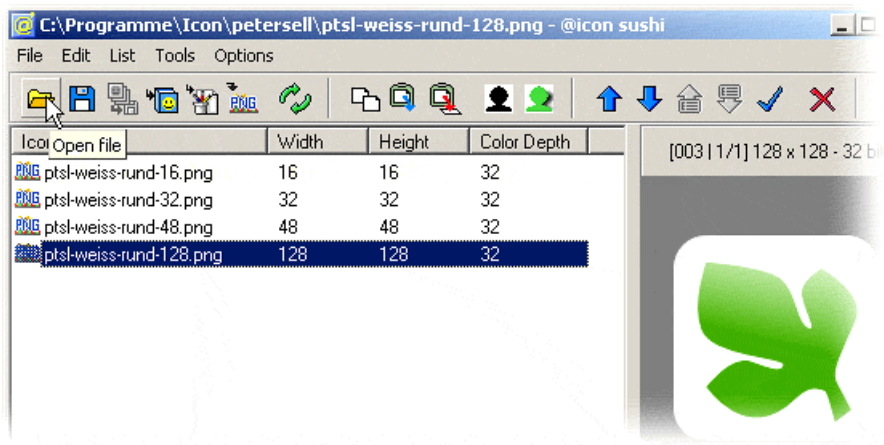


Abb. 19: Graphiken sämtlicher Größen laden

2. Markieren Sie sämtliche Bilddateien und klicken Sie auf die Symbolschaltfläche **Save as Multiple Icon**.

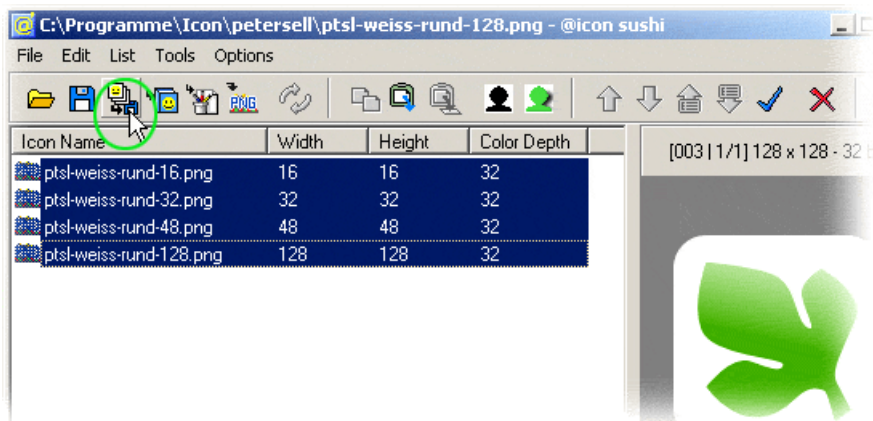


Abb. 20: Als Mehrfach-Icon abspeichern

3. Wählen Sie im Fenster **Save as Multiple Icon** einen Speicher-Ordner und einen Namen für die ICO-Datei. Klicken Sie abschließend auf **Speichern**.
4. Kopieren Sie die ICO-Datei in das Hauptverzeichnis Ihrer Online-Hilfe. Bei einer Wordpress-Installation kopieren Sie die Icondatei in das Hauptverzeichnis bzw. in das Hauptverzeichnis Ihres Templates.
5. Bei Wordpress: binden Sie die Icondatei (dateiname.ico) in die header.php innerhalb des head-Tags ein. Haben Sie keine WP-Installation, tragen Sie die URL ohne php-Tags ein.

```
<link rel="shortcut icon" href="<?php bloginfo('url'); ?>/dateiname.ico" type="image/x-icon" />
```

Links zum Artikel

1. [Icon Sushi von pcmasters.de downloaden](#)

Impressum

| | |
|-------------------|---|
| Autor: | Andreas Petersell petersell@ditalog.com www.ditalog.com |
| Ausgabe: | Ditalog - Ideen für die Dokumentation und Online-Hilfe |
| Redaktionsschluß: | 10.03.2011 |

Index

A

Anker 12

B

build_demo.xml 11

C

CombiNumerals 4

D

dita2htmlImpl.xsl 11
gen-user-sidetoc 11

F

FAQ-Demo-Spezialisation 11
faq2html_shell.xsl 11
Flagging im PDF-Output 5
Flagging in Ditaval-Datei 7
Font CombiNumerals 4

G

gen-user-side-toc 11

I

ico-Datei einbinden 15
Icon Sushi 15

Icondatei ico erstellen 15
Image in Ditaval-Datei 7

K

Kontextsensitive Hilfe 9
per ditamap 9

N

Namensanker 12
Nummerierung in Screenshots 4

O

Override 11

P

Plugin 10
erstellen 10

S

scriptorium.com 12
Seiteninhaltsverzeichnis 12
shortdesc 9
für kontextsensitive Hilfe 9

Z

Zugang zum Bildschirm auskoppeln 5