hyperCMS ContentManagementServer

Version 5.8

Programers Guide

2015-12-22

Inhalt

1 Einleitung

Die folgenden Kapitel behandeln die Funktionsbibliotheken des hyper Content & Digital Asset Management Servers und stellen somit die Dokumentation des API (Application Programming Interface) dar.
Alle Bibliotheken befinden sich innerhalb der hyperCMS Installation im Ordner "function" und können in die jeweiligen Scripts bzw. Templates eingebunden und genutzt werden. Damit lassen sich z.B. auch dynamische Seiten (Applikationen) unter Einsatz des XML-Content-Repository programmieren.
Sollten Sie Ihre Applikation auf einen physisch getrennten Server betreiben, so ist es wichtig, dass die Funktionsbibliotheken auch auf dem Publikationsserver zur Verfügung stehen. In diesem Fall ist es wichtig, dass die entsprechenden Dateien auch am Publikationsserver zur Verfügung stehen.

2 hyperCMS XML-Content-Repository

Das XML-Content-Repository beinhaltet alle XML-Content-Container und stellt somit alle Inhalte native XML zur Verfügung. Die Struktur (Schema) innerhalb eines XML-Content-Containers wird auf Basis des verwendeten Templates dynamisch erzeugt und besitzt folgendes Aussehen:

```
<?xml version="1.0" encoding="UTF-8" ?>
<container>
  <hyperCMS>
    <contentcontainer>0000023.xml</contentcontainer>
    <contentxmlschema>object/page</contentxmlschema>
    <contentorigin>%page%/Publication/testpage.php</contentorigin>
    <contentobjects>%page%/Publication/testpage.php|%page%/ Publication/linkedcopy_of_testpage.php
|</contentobjects>
    <contentuser>demouser</contentuser>
    <contentdate>2002-11-26</contentdate>
    <contentpublished>2002-11-26</contentpublished>
   <contentstatus>active</contentstatus>
  </hyperCMS>
  <head>
    <pagetitle>test</pagetitle>
    <pageauthor>Mr. Content</pageauthor>
    <pagedescription>just a small demonstration</pagedescription>
    <pagekeywords>demo of XML</pagekeywords>
    <pagecontenttype>text/html; charset=UTF-8</pagecontenttype>
    <pagelanguage>de</pagelanguage>
    <pagerevisit></pagerevisit>
  </head>
  <textcollection>
    <text>
      <text_id>headline</text_id>
      <textuser>demouser</textuser>
      <textcontent>fgfdgfdg</textcontent>
    </text>
    <text>
      <text_id>summary</text_id>
      <textuser>demouser</textuser>
      <textcontent><![CDATA[This is a
      <STRONG><EM>summary</EM></STRONG>]]></textcontent>
    </text>
  </textcollection>
  <mediacollection>
    <media>
      <media_id>logo</media_id>
      <mediauser>otheruser</mediauser>
      <mediafile>Publication/demo_hcms0000033.jpg</mediafile>
      <mediaobject>%page%/Publication/Multimedia/demo.jpg</mediaobject>
      <mediaalttext>demoimage</mediaalttext>
      <mediaalign></mediaalign>
```

```
    <mediawidth>200</mediawidth>
    <mediaheight>100</mediaheight>
   </media>
  </mediacollection>
  <linkcollection>
   <link>
    <link_id>verweis</link_id>
    <linkuser>demouser</linkuser>
    <linkhref>http://localhost/index.php</linkhref>
    <linktarget>_blank</linktarget>
    <linktext>click me</linktext>
   </link>
  </linkcollection>
  <componentcollection>
   <component>
    <component_id>teasers</component_id>
    <componentuser>otheruser</componentuser>
    <componentcond>$customer == "private"</componentcond>
    <componentfiles>%comp%/Publication/teaser_1.php|%comp%/Publication/teaser_2.php|</componentfiles>
   </component>
   <component>
    <component_id>banner</component_id>
    <componentuser>demouser</componentuser>
    <componentcond></componentcond>
    <componentfiles>%comp%/banner.php</componentfiles>
   </component>
  </componentcollection>
  <articlecollection>
   <article>
    <article_id>news</article_id>
    <articletitle>Top News</articletitle>
    <articledatefrom>2002-10-01</articledatefrom>
    <articledateto>2002-11-01</articledateto>
    <articlestatus>active</articlestatus>
    <articleuser>demouser</articleuser>
    <articletextcollection>
     <text>
      <text_id>news:headline</text_id>
      <textuser>demouser</textuser>
      <textcontent>News from Scene</textcontent>
     </text>
    </articletextcollection>
    <articlemediacollection>
    </articlemediacollection>
    <articlelinkcollection>
    </articlelinkcollection>
    <articlecomponentcollection>
    </articlecomponentcollection>
   </article>
   <article>
    <article_id>special</article_id>
    <articletitle>Special Info</articletitle>
    <articledatefrom>2002-01-01</articledatefrom>
    <articledateto>2002-01-01</articledateto>
    <articlestatus>inactive</articlestatus>
    <articleuser>otheruser</articleuser>
    <articletextcollection>
     <text>
      <text_id>special:informations</text_id>
      <textuser>otheruser</textuser>
      <textcontent><![CDATA[<STRONG><FONT color=#cc0033>What is really going on behind the
Scene</FONT></STRONG>... find it out]]></textcontent>
     </text>
    </articletextcollection>
    <articlemediacollection>
    </articlemediacollection>
    <articlelinkcollection>
    </articlelinkcollection>
    <articlecomponentcollection>
    </articlecomponentcollection>
   </article>
  </articlecollection>
</container>
```

Nach Durchsicht des Content Containers ist eine Struktur zu erkennen, die sich aus den folgenden wesentlichen Grundelementen für die Content-Ablage zusammensetzt:

- hyperCMS spezifische Informationen
- Meta-Informationen
- Text
- Medien (Bilder oder andere Multimedia-Dateien)
- Links
- Komponenten
- Artikel

Der gesamte Inhalt setzt sich aus diesem Grundbausteinen zusammen, deren Informationen wiederum innerhalb von XML-Tags abgelegt werden.
Artikel nehmen so wiederum die Elemente Text, Medien und Links in sich auf. Der gesamte Inhalt einer Seite oder Komponente lässt sich über den zugehörigen Content-Container beziehen.

2.1 hyperCMS spezifische Informationen

Die in diesem XML-Knoten erfassten Daten stellen primär für das Management des Containers relevante Informationen dar.

```
<hyperCMS>
  <contentcontainer>0000023.xml</contentcontainer>
  <contentxmlschema>object/page</contentxmlschema>
  <contentorigin>%page%/testpage.php</contentorigin>
  <contentobjects>%page%/testpage.php|%page%/linkedcopy_of_testpage.php |</contentobjects>
  <contentuser>demouser</contentuser>
  <contentdate>2002-11-26</contentdate>
  <contentpublished>2002-11-26</contentpublished>
 <contentstatus>active</contentstatus>
</hyperCMS>
```

Erklärung:

contentcontainer	Name des Content Containers (einmalig über alle Publikationen)
contentxmlschema	Schema des Objektes: Seite = page oder Komponente = comp
contentorigin	Objekt (Seite oder Komponente) die zur Generierung des Content Containers führte
contentobjects	Alle Objekte die diesen Content Container benutzen
contentuser	Objekteigentümer
contentdate	Datum der letzten Änderung des Containers
contentpublished	Datum der letzten Publizierung eines Objektes basierend auf den Content Container
contentstatus	Der Status ist "active" solange ein Objekt das auf den Container basiert existiert. Wurden alle Objekte die auf den Container basieren entfernt wird der Status "deleted" gesetzt. Der Container beinhaltet damit den letzten Informationsstand, kann jedoch nicht mehr genutzt werden.

2.2 Meta-Informationen

Die Standard Meta-Informationen einer HTML-Seite werden in diesem XML-Knoten beschrieben.

```
<head>
  <pagetitle>test</pagetitle>
  <pageauthor>Mr. Content</pageauthor>
  <pagedescription>just a small demonstration</pagedescription>
  <pagekeywords>demo of XML</pagekeywords>
  <pagecontenttype>text/html; charset=UTF-8</pagecontenttype>
  <pagelanguage>de</pagelanguage>
  <pagerevisit></pagerevisit>
</head>
```

Erklärung:

pagetitle	Seitentitel
pageauthor	Seitenautor
pagedescription	Beschreibung der Inhalte der Seite
pagekeywords	Liste der Schlüsselwörter der Seite
pagecontenttype	Content-Type (Zeichensatz) der Seite oder Komponente
pagelanguage	Sprachkürzel der Seite
pagerevisit	Wiederbesuch der Seite durch Suchmaschinen

2.3 Text

Diese XML-Knoten speichern den Text.

```
<text>
  <text_id>headline</text_id>
  <textuser>demouser</textuser>
  <textcontent>fgfdgfdg</textcontent>
</text>
```

Erklärung:

text_id	Textidentifikation
textuser	Texteigentümer (letzte Änderung des Textes durch einen Benutzer)
textcontent	Inhalt des Textes

2.4 Medien

Dieser XML-Knoten beschreibt eingebunden Medien.

```
<media>
  <media_id>logo</media_id>
  <mediauser>otheruser</mediauser>
  <mediafile>Publication/demo_hcms0000033.jpg</mediafile>
  <mediaobject>%page%/Publication/Multimedia/demo.jpg</mediaobject>
  <mediaalttext>demoimage</mediaalttext>
  <mediaalign></mediaalign>
  <mediawidth>200</mediawidth>
  <mediaheight>100</mediaheight>
</media>
```

Erklärung:

media_id	Medienidentifikation
mediauser	Medieneigentümer (letzte Änderung des Mediums durch einen Benutzer)
mediafile	eingebunden Mediendatei mit Angabe der Publikation
mediaobject	Pfadangabe zur Multimediakomponente
mediaalttext	Alternativtext des Mediums
mediaalign	Ausrichtung des Mediums
medawidth	Dargestellte Breite des Mediums
mediaheight	Dargestellte Höhe des Mediums

2.5 Links

Dieser XML-Knoten beschreibt die Verlinkung zu Seiten.

```
<link>
  <link_id>verweis</link_id>
  <linkuser>demouser</linkuser>
  <linkhref>http://localhost/index.php</linkhref>
  <linktarget>_blank</linktarget>
  <linktext>click me</linktext>
</link>
```

Erklärung:

link_id	Linkidentifikation
linkuser	Linkeigentümer (letzte Änderung des Links durch einen Benutzer)
linkhref	Referenz (Link) zu einer Seite oder Datei
linktarget	Ziel des Referenzierung (Name des Frames)
linktext	Text der den Link beschreibt/darstellt

2.6 Komponenten

Dieser XML-Knoten beschreibt die Verlinkung zu Komponenten.

```
<component>
  <component_id>teasers</component_id>
  <componentuser>otheruser</componentuser>
  <componentcond>$customer == "private"</componentcond>
  <componentfiles>%comp%/teaser_1.php|%comp%/teaser_2.php|</componentfiles>
</component>
```

Erklärung:

component_id	Komponentenidentifikation
componentuser	Komponenteneigentümer (letzte Änderung der Komponentenreferzierung durch einen Benutzer)
componentcond	Zugeordnetes Kundenprofil zu der Komponente
componentfiles	Referenz (Komponenten-Link) zu einer oder mehreren Komponenten

2.7 Artikel

Dieser XML-Knoten beschreibt die Artikelinformation.

```
<article>
  <article_id>news</article_id>
  <articletitle>Top News</articletitle>
  <articledatefrom>2002-10-01</articledatefrom>
  <articledateto>2002-11-01</articledateto>
  <articlestatus>active</articlestatus>
  <articleuser>demouser</articleuser>
  <articletextcollection>
  </articletextcollection>
</article>
```

Erklärung:

article_id	Artikelnidentifikation
articletitle	Titel des Artikels
articeldatefrom	Beginn der Veröffentlichung des Artikels
articeldateto	Ende der Veröffentlichung des Artikels
articlestatus	Bestimmung der Veröffentlichung des Artikels: active = immer veröffentlicht inactive = nicht veröffenlicht timeswitched = zeitgesteuerte Veröffentlichung
articleuser	Artikeleigentümer (letzte Änderung des Artikels durch einen Benutzer)
articlecollection	Umfasst alle dem Artikel zugeordneten Inhalte

3 Funktionsbibliotheken

3.1 Einbindung einer Bibliothek

Das Einbinden einer Konfiguratzion oder Bibliothek setzt voraus, dass man den absoluten oder relativen Pfad zur Bibliothek kennt. Durch Verwendung der Funktion "require" oder "require_once" und der Angabe des Pfades inklusive der einzubinden Datei werden die enthaltenen Funktionen der Bilbliothek eingebunden. Sobald die Bibliothek eingebunden ist, können deren Funktionen im Script genutzt werden.
Um die hyperCMS-Funktionen nutzen zu können, bedarf es der Einbindung der Datei "hypercms_api.inc.php". Diese Datei beinhaltet alle für die Programmierung benötigten Funktionen.

```
// absolute Angabe unter MS Windows
require_once ("C:/inetpub/wwwroot/hypercms/function/hypercms_api.inc.php");

// relative Angabe unter MS Windows oder auch UNIX-Derivaten
require_once ("function/hypercms_api.inc.php");
```

3.2 Laden der Konfiguration

3.2.1 Content Management Server

Um die Konfiguration von hyperCMS nutzen zu können muss die entsprechende Datei geladen warden. Diese beinhaltet alle wesentlichen Einstellungen des zu behandelnden Mandanten (Site).
Mit Hilfe der Identifikation einer Publikation, z.B. mit der Variable $site kann die Konfiguration einer Publikation geladen werden. Die hyperCMS Hauptkonfigurationsatei befindet sich im Verzeichnis „hypercms/config“ und trägt den Namen "config.inc.php". Die publikationsspezifischen Konfigurationsateien befinden sich im Verzeichnis "data/config". Deren Dateiname setzt sich aus dem Namen der Publikation sowie der Endung ".inc.php" zusammen, Bsp: site.inc.php.

```
// Einbinden der Hauptkonfigurationsatei (auf Pfadangabe ist zu achten):
require_once ("C:/inetpub/wwwroot/hypercms/config.inc.php");

// Einbinden Konfiguration einer Publikation
// Achtung: Bitte verwenden Sie valid_publicationname, um den Namen zu verifizieren, bevor
// Sie die Datei einbinden
if (valid_publicationname ($site))
{
  require_once ($mgmt_config['abs_path_data']."config/".$site.".conf.php");
}
```

Die Config-Dateien können geöffnet und gelesen werden. Jeder Parameter wird darin beschrieben und steht für die Nutzung in Programmen zur Verfügung. Bitte werfen Sie daher einen Blick in die Konfiguration, um mehr über die Parameter und deren Namen zu erfahren.

Es ist auch notwendig eine Sprache zu wählen. Hierfür dient die Variabel $lang. $lang beinhaltet das Sprachkürzel, welche in der Konfiguration "hypercms/config/config.inc.php" eingesehen werden können.

```
// Setzen der Spracheinstellung für Nachrichten von Funktionen, Deutsch (de)
$lang = "de";
```

Da Sie die Funktionen des hyper API benutzen möchten, müssen Sie auch noch dieses einbinden.

```
// Einbinden der Funktionsbibliothek:
require_once ($mgmt_config['abs_path_cms']."/function/hypercms_api.inc.php");
```

Nun können Sie die Funktionen des APIO nutzen, um z.B. einen Content Container einer bestimmtes Objektes über unterschiedliche Methoden zu laden:

```
// Laden der Seite
$pagedata = loadfile ("%page%/MyPublication/home/", "index.php");

// Content Container Name auslesen
$contentcontainer  = filepointer ($pagedata, "content");

// Laden des veröffentlichten Content Container aus dem Content Repository
$containerdata = loadcontainer ($contentcontainer, "published", $user);

// Oder noch einfacher direkt über den Objektpfad
$containerdata = getobjectcontainer ("MyPublication", "%page%/MyPublication/home/",
"index.php", $user);
```

Funktionen laden die Konfiguration einer Publikatzion, sollte diese nicht verfügbar sein.
Da viele Funktionen die Einstellungen einer Publikation benötigen, ist es ratsam die Konfiguration immer einzubinden.

3.2.2 Publication Server

Beachten Sie, dass die Konfiguration des Publication Servers (Publikationsziel) davon getrennt in einer INI-Datei abgelegt ist. Benötigen Sie die Publikationziel-Einstellungen, so müssen Sie die INI-Datei laden und parsen. Danach stehen Ihnen die Variablen in einem Array zur Verfügung.
Die INI-Datei des Publikationszieles befindet sich im externen Repository im Verzeichnis "repository/config". Der Name der Datei entspricht dem Namen der Publikation mit der Dateierweiterung ".ini".

```
// laden und parsen der INI-Datei mit hilfe von PHP
$publ_config = parse_ini_file ("C:/inetpub/wwwroot/repository/config/Mandant_1.ini");

// Zugreifen auf die Variablen des Publikationszieles
echo "Das ist der Document Root der Seiten der Publikation:".$publ_config[abs_publ_page];
```

3.3 Globale Variablen

Viele Funktionen nutzen globale Variablen die in der Konfiguration gespeichert sind und den Funktionen zur Verfügung stehen. Sie sollten daher bei der Wahl der Variablennamen in Ihren eigenen Scripts acht geben, dass Sie nicht die von hyperCMS genutzen globalen Variablen verwenden.
Die folgende Liste zeigt alle globalen Variablen von hyperCMS, die nicht in eigenen Scripts manipuliert/verändert werden dürfen:

```
$mgmt_config
$lang
$lang_name
$lang_shortcut
$lang_codepage
$lang_shortcut_default
```

Viele globale Variablen von hyperCMS sind für die Verwendung in hyperCMS-Scripts und PHP-Scripts nützlich, diese stehen nur dann zur Verfügung, wenn die entsprechende Konfiguration zuvor geladen wurde, oder eine hyperCMS-Script (wird nur während des Publikationsprozesses ausgeführt) in Verwendung ist. Da dies bei der Voransicht als auch beim Publizieren von Seiten und Komponenten passiert, können diese Variablen in hyperCMS-Scripts genutzt werden. Bei dynamischen Applikationen, die bei jedem Aufruf der Seite oder Komponente durch einen Besucher ausgeführt werden, muss die Konfiguration direkt im Template eingebunden werden, sofern Variablen von hyperCMS benötigt werden.

Content Management Server:

$lang	Sprachkürzel lt. config.inc.php
$mgmt_config['url_path_cms'**]**	URL des hyperCMS Root Verzeichnis lt. config.inc.php
mgmt_config['abs_path_cms'**]**	absoluter Pfad zum hyperCMS Root Verzeichnis lt. config.inc.php
$mgmt_config['url_path_page'**]**	URL des Doc Roots der Publikation im Managementsystem
$mgmt_config['abs_path_page'**]**	absoluter Pfad zum Doc Roots der Publikation im Managementsystem
$mgmt_config['url_path_comp'**]**	URL des Komponenten Root der Publikation im Managementsystem
mgmt_config['abs_path_comp'**]**	absoluter Pfad zum Komponenten Roots der Publikation im Managementsystem

Publication Server:
hyperCMS-Scripts können die Variablen ohne weiteres zutun nutzen. Die Werte werden im Array $publ_config gespeichert, sind aber auch optional auch ohne Array nutzbar. Wird das Script/Anwendung bei jedem publikationsseitigen Aufruf ausgeführt, so ist die Konfigurationsdatei gesondert zu laden.

$publ_config['url_publ_page'**]**	URL des Doc Roots der Publikation im Publikationssystem
$publ_config['abs_publ_page'**]**	absoluter Pfad zum Doc Roots der Publikation im Publikationssystem
$publ_config['url_publ_comp'**]**	URL des Komponenten Roots der Publikation im Publikationssystem
$publ_config['abs_publ_comp'**]**	absoluter Pfad zum Komponenten Roots der Publikation im Publikationssystem

Optional (veraltet):

$url_publ_page	URL des Doc Roots der Publikation im Publikationssystem
$abs_publ_page	absoluter Pfad zum Doc Roots der Publikation im Publikationssystem
$url_publ_comp	URL des Komponenten Roots der Publikation im Publikationssystem
$abs_publ_comp	absoluter Pfad zum Komponenten Roots der Publikation im Publikationssystem

Vorlagenvariablen

Es gibt auch die Möglichkeit mit hyperCMS-eigenen Vorlagenvariablen zu arbeiten. Diese Variablen stellen eine Besonderheit dar, da sie nicht mit hyperCMS-Script in Verbindung stehen müssen. Sie sind vielmehr Platzhalter für den Wert einer Variable und können in Vorlagen beliebig eingesetzt werden.
Diese neutrale Form der Variablen sollte primär in Templates Verwendung finden, da damit ein technologieneutraler Einsatz stattfinden kann.
Achten Sie bitte auf die Kleinschreibung aller Variablen!

%container% steht für den Namen des Content Containers eines Objektes.
%template% steht für den Dateinamen der verwendeten Vorlage des Objektes.
%object% steht für den Namen des Objektes.
%date% beschreibt das aktuelle Datum im Format JJJJ-MM-TT.

Für die Einbindung von Mediendateien wird eine Pfadvariable benutzt. Diese Pfadvariable wird beim Publizieren der Seite oder Komponente durch die URL (Adresse) der Konfiguration des Publikationszieles ersetzt:

%media% steht für die Pfadangabe (URL) des Content Medien Repository.
%tplmedia% steht für die Pfadangabe (URL) des Vorlagen Medien Repository.

Auch die Wurzelverzeischnisse der Seiten und Komponenten (publikationsseitig!) lassen sich abrufen:

%url_page% steht für die Pfadangabe (URL) des Seiten-Wurzelverzeichnisses.
%abs_page% steht für die absolute Pfadangabe des Seiten-Wurzelverzeichnisses.
%url_comp% steht für die Pfadangabe (URL) des Komponenten-Wurzelverzeichnisses.
%abs_comp% steht für die absolute Pfadangabe des Komponenten -Wurzelverzeichnisses.

In Zusammenhang mit der Nutzung des hyperCMS APIs ist es oft ratsam, bei Pfadangaben die Platzhalter %page% und %comp% zu nutzen. Diese Pfadvariablen lassen sich nur managementseitig nutzen, sie stehen für die Pfade zu den Wurzelverzeischnissen von Seiten und Komponenten.
Zu beachten ist, dass die Variable immer gepaart mit dem Publikationsnamen das Wurzelverzeichnis bildet, z.B:

%page%/besttrade/ …. Wurzelverzeichnis der Seiten der Publikation "besttrade"

%page%/Publikationsname/ steht für die absolute Pfadangabe des Seiten-Wurzelverzeichnisses.
%comp%/Publikationsname/ steht für die absolute Pfadangabe des Komponenten -Wurzelverzeichnisses.

3.4 Bibliothek Object Operation

Diese Bibliothek beinhaltet alle Funktionen für die Manipulation von Objekten (Seiten, Komponenten oder Dateien). Bitte benutzen Sie ausschließlich diese Funktionen für den Zugriff auf Objekte, die das System verwaltet.

3.4.1 createfolder

Syntax:
createfolder ($site, $location, $foldernew, $user)

Beschreibung:
Erzeugt einen neuen Ordner.

Bsp:
$result = createfolder ("besttrade", "%page%/besttrade/", "company", "brown");

Input-Parameter:

$site	Name der Publikation
$location	absoluter Pfad (Position des neuen Ordners)
$foldernew	Name des neuen Ordners
$user	Benutzername

globale Input-Parameter:
Die folgenden globalen Input Parameter sind ebenfalls der Funktion zu übergeben:

$lang	Spracheinstellung bzw. Sprachkürzel, z.B. "en", "de"

Output:

Array	Array $result das folgende Informationen beinhaltet:
$result[result]	True/False (Konnte der neue Ordner angelegt werden)
$result[add_onload]	JavaScript Code für das onLoad Event
$result[message]	Nachricht über den Erfolg der Aktion bzw. Fehlermeldung
$result[folder]	Name des Ordners

3.4.2 deletefolder

Syntax:
deletefolder ($site, $location, $folder, $user)

Beschreibung:
Entfernt einen bestehenden Ordner. Der Ordner wird nur dann entfernt, wenn er keine Objekte mehr beinhaltet. Alle Objekte müssen daher zuvor mit deleteobject entfernt werden.

Bsp:
$result = deletefolder ("besttrade", "%page%/besttrade/", "company", "brown");

Input-Parameter:

$site	Name der Publikation
$location	absoluter Pfad (Position des neuen Ordners)
$folder	Name des zu entfernenden Ordners
$user	Benutzername

globale Input-Parameter:
Die folgenden globalen Input Parameter sind ebenfalls der Funktion zu übergeben:

$lang	Spracheinstellung bzw. Sprachkürzel, z.B. "en", "de"

Output:

Array	Array $result das folgende Informationen beinhaltet:
$result[result]	True/False (Konnte der Ordner entfernt werden)
$result[add_onload]	JavaScript Code für das onLoad Event
$result[message]	Nachricht über den Erfolg der Aktion bzw. Fehlermeldung
$result[folder]	Name des bestehenden Ordners bei Misserfolg, ansonst leer

3.4.3 renamefolder

Syntax:
renamefolder ($site, $location, $folder, $foldernew, $user)

Beschreibung:
Benennt einen bestehenden Ordner um.

Bsp:
$result = renamefolder ("besttrade", "%page%/besttrade/", "company", "news", "brown");

Input-Parameter:

$site	Name der Publikation
$location	absoluter Pfad (Position des Ordners)
$folder	Alter Name des Ordners
$foldernew	Neuer Name des Ordners
$user	Benutzername

globale Input-Parameter:
Die folgenden globalen Input Parameter sind ebenfalls der Funktion zu übergeben:

$lang	Spracheinstellung bzw. Sprachkürzel, z.B. "en", "de"

Output:

Array	Array $result das folgende Informationen beinhaltet:
$result[result]	True/False (Konnte der Ordner umbenannt werden)
$result[add_onload]	JavaScript Code für das onLoad Event
$result[message]	Nachricht über den Erfolg der Aktion bzw. Fehlermeldung
$result[folder]	Name des Ordners

3.4.4 createobject

Syntax:
createobject ($site, $location, $object, $template, $user)

Beschreibung:
Erzeugt eine neue Seite oder Komponente auf Basis einer Vorlage. Bitte beachten Sie das die Position ($location) auch die Kategorie des Objektes (Seite/Komponente) bestimmt. Dies bedingt weiters, dass es sich beim Wert des Parameters $template um eine gültige Seiten- bzw. Komponentenvorlage handeln muss.

Bsp:
$result = createobject ("besttrade", "%page%/besttrade/", "index", "page_main", "Miller");

Input-Parameter:

$site	Name der Publikation
$location	absoluter Pfad (Position des Objektes)
$object	Name des neuen Objektes (Seite oder Komponente)
$template	Name der zu verwendenden Seiten- oder Komponentenvorlage (Name der Vorlage oder Dateiname)
$user	Benutzername

globale Input-Parameter:
Die folgenden globalen Input Parameter sind ebenfalls der Funktion zu übergeben:

$lang	Spracheinstellung bzw. Sprachkürzel, z.B. "en", "de"

Output:

Array	Array $result das folgende Informationen beinhaltet:
$result[result]	True/False (Erfolg der Aktion)
$result[add_onload]	JavaScript Code für das onLoad Event
$result[message]	Nachricht über den Erfolg der Aktion bzw. Fehlermeldung
$result[object]	Dateiname der Seite oder Komponente
$result[objectname]	Name der Seite oder Komponente
$result[objecttype]	Filetype bzw. File Extension der Datei

3.4.5 deleteobject

Syntax:
deleteobject ($site, $location, $object, $user)

Beschreibung:
Entfernt eine bestehende Seite, Datei oder Komponente.

Bsp:
$result = deleteobject ("besttrade", "%page%/besttrade/", "sales.doc", "Miller");

Input-Parameter:

$site	Name der Publikation
$location	absoluter Pfad (Position des Objektes)
$object	Name des Objektes
$user	Benutzername

globale Input-Parameter:
Die folgenden globalen Input Parameter sind ebenfalls der Funktion zu übergeben:

$lang	Spracheinstellung bzw. Sprachkürzel, z.B. "en", "de"

Output:

Array	Array $result das folgende Informationen beinhaltet:
$result[result]	True/False (Erfolg der Aktion)
$result[add_onload]	JavaScript Code für das onLoad Event
$result[message]	Nachricht über den Erfolg der Aktion bzw. Fehlermeldung
$result[publication]	Name der Publikation bzw. Mandant in dem das Objekt existiert
$result[location]	absoluter Pfad im Filesystem (Position des Objektes)
$result[object]	Dateiname der Seite, Datei oder Komponente
$result[objectname]	Name der Seite, Datei oder Komponente
$result[objecttype]	Filetype bzw. File Extension der Datei

3.4.6 renameobject

Syntax:
renameobject ($site, $location, $object, $objectnew, $user)

Beschreibung:
Umbenennen eine bestehender Seite, Datei oder Komponente.

Bsp:
$result = renameobject ("besttrade", "%page%/besttrade/", "sales.doc", "best.doc", "Miller");

Input-Parameter:

$site	Name der Publikation
$location	absoluter Pfad (Position des Objektes)
$object	Alter Name des Objektes
$objectnew	Neuer Name des Objektes (ohne Dateiendung)
$user	Benutzername

globale Input-Parameter:
Die folgenden globalen Input Parameter sind ebenfalls der Funktion zu übergeben:

$lang	Spracheinstellung bzw. Sprachkürzel, z.B. "en", "de"

Output:

Array	Array $result das folgende Informationen beinhaltet:
$result[result]	True/False (Erfolg der Aktion)
$result[add_onload]	JavaScript Code für das onLoad Event
$result[message]	Nachricht über den Erfolg der Aktion bzw. Fehlermeldung
$result[publication]	Name der Publikation bzw. Mandant in dem das Objekt existiert
$result[location]	absoluter Pfad im Filesystem (Position des Objektes)
$result[object]	Dateiname der Seite, Datei oder Komponente
$result[objectname]	Name der Seite, Datei oder Komponente
$result[objecttype]	Filetype bzw. File Extension der Datei

3.4.7 cutobject

Syntax:
cutobject ($site, $location, $object, $user)

Beschreibung:
Ausschneiden eine bestehender Seite, Datei oder Komponente.

Bsp:
$result = cutobject ("besttrade", "%page%/besttrade/", "index.php", "Miller");

Input-Parameter:

$site	Name der Publikation
$location	absoluter Pfad (Position des Objektes)
$object	Alter Name des Objektes
$user	Benutzername

globale Input-Parameter:
Die folgenden globalen Input Parameter sind ebenfalls der Funktion zu übergeben:

$lang	Spracheinstellung bzw. Sprachkürzel, z.B. "en", "de"

Output:

Array	Array $result das folgende Informationen beinhaltet:
$result[result]	True/False (Erfolg der Aktion)
$result[add_onload]	JavaScript Code für das onLoad Event
$result[message]	Nachricht über den Erfolg der Aktion bzw. Fehlermeldung
$result[object]	Dateiname der Seite, Datei oder Komponente
$result[objectname]	Name der Seite, Datei oder Komponente
$result[objecttype]	Filetype bzw. File Extension der Datei
$result[clipboard]	temporärer Eintrag im Clipboard (kann als globale Variable $clipboard der Funktion pasteobject übergeben werden, diese muss somit keinen Lesezugriff auf die temporäre Datei ausführen)

3.4.8 copyobject

Syntax:
copyobject ($site, $location, $object, $user)

Beschreibung:
Kopieren eine bestehender Seite, Datei oder Komponente.

Bsp:
$result = copyobject ("besttrade", "%page%/besttrade/", "index.php", "Miller");

Input-Parameter:

$site	Name der Publikation
$location	absoluter Pfad (Position des Objektes)
$object	Alter Name des Objektes
$user	Benutzername

globale Input-Parameter:
Die folgenden globalen Input Parameter sind ebenfalls der Funktion zu übergeben:

$lang	Spracheinstellung bzw. Sprachkürzel, z.B. "en", "de"

Output:

Array	Array $result das folgende Informationen beinhaltet:
$result[result]	True/False (Erfolg der Aktion)
$result[add_onload]	JavaScript Code für das onLoad Event
$result[message]	Nachricht über den Erfolg der Aktion bzw. Fehlermeldung
$result[object]	Dateiname der Seite, Datei oder Komponente
$result[objectname]	Name der Seite, Datei oder Komponente
$result[objecttype]	Filetype bzw. File Extension der Datei
$result[clipboard]	temporärer Eintrag im Clipboard (kann als globale Variable $clipboard der Funktion pasteobject übergeben werden, diese muss somit keinen Lesezugriff auf die temporäre Datei ausführen)

3.4.9 copyconnectedobject

Syntax:
copyconnectedobject ($site, $location, $object, $user)

Beschreibung:
Kopieren eine bestehender Seite, Datei oder Komponente auf Basis des gleichen Content Containers.

Bsp:
$result = copyconnectedobject ("besttrade", "%page%/besttrade/", "index.php", "Miller");

Input-Parameter:

$site	Name der Publikation
$location	absoluter Pfad (Position des Objektes)
$object	Alter Name des Objektes
$user	Benutzername

globale Input-Parameter:
Die folgenden globalen Input Parameter sind ebenfalls der Funktion zu übergeben:

$lang	Spracheinstellung bzw. Sprachkürzel, z.B. "en", "de"

Output:

Array	Array $result das folgende Informationen beinhaltet:
$result[result]	True/False (Erfolg der Aktion)
$result[add_onload]	JavaScript Code für das onLoad Event
$result[message]	Nachricht über den Erfolg der Aktion bzw. Fehlermeldung
$result[object]	Dateiname der Seite, Datei oder Komponente
$result[objectname]	Name der Seite, Datei oder Komponente
$result[objecttype]	Filetype bzw. File Extension der Datei
$result[clipboard]	temporärer Eintrag im Clipboard (kann als globale Variable $clipboard der Funktion pasteobject übergeben werden, diese muss somit keinen Lesezugriff auf die temporäre Datei ausführen)

3.4.10 pasteobject

Syntax:
pasteobject ($site, $location, $user)

Beschreibung:
Einfügen einer bestehender Seite, Datei oder Komponente.

Bsp:
$result = pasteobject ("besttrade", "%page%/besttrade/", "Miller");

Input-Parameter:

$site	Name der Publikation
$location	absoluter Pfad (Position des Objektes)
$user	Benutzername
$clipboard	globale Varibale mit dem temporären Eintrag im Clipboard (Damit ist ein Lesezugriff auf die temporäre Datei nicht notwendig.)

globale Input-Parameter:
Die folgenden globalen Input Parameter sind ebenfalls der Funktion zu übergeben:

$lang	Spracheinstellung bzw. Sprachkürzel, z.B. "en", "de"

Output:

Array	Array $result das folgende Informationen beinhaltet:
$result[result]	True/False (Erfolg der Aktion)
$result[add_onload]	JavaScript Code für das onLoad Event
$result[message]	Nachricht über den Erfolg der Aktion bzw. Fehlermeldung
$result[publication]	Name der Publikation bzw. Mandant in dem das Objekt existiert
$result[location]	absoluter Pfad im Filesystem (Position des Objektes)
$result[object]	Dateiname der Seite, Datei oder Komponente
$result[objectname]	Name der Seite, Datei oder Komponente
$result[objecttype]	Filetype bzw. File Extension der Datei

3.4.11 lockobject

Syntax:
lockobject ($site, $location, $object, $user)

Beschreibung:
Sperren einer oder mehrerer bestehender Seiten oder Komponenten die auf den gleichen Content Container beruhen für die exklusive Nutzung eines Benutzers.

Bsp:
$result = lockobject ("besttrade", "%page%/besttrade/", "index.php","Miller");

Input-Parameter:

$site	Name der Publikation
$location	absoluter Pfad (Position des Objektes)
$object	Name des Objektes
$user	Benutzername

globale Input-Parameter:
Die folgenden globalen Input Parameter sind ebenfalls der Funktion zu übergeben:

$lang	Spracheinstellung bzw. Sprachkürzel, z.B. "en", "de"

Output:

Array	Array $result das folgende Informationen beinhaltet:
$result[result]	True/False (Erfolg der Aktion)
$result[add_onload]	JavaScript Code für das onLoad Event
$result[message]	Nachricht über den Erfolg der Aktion bzw. Fehlermeldung
$result[object]	Dateiname der Seite, Datei oder Komponente
$result[objectname]	Name der Seite, Datei oder Komponente
$result[objecttype]	Filetype bzw. File Extension der Datei

3.4.12 unlockobject

Syntax:
unlockobject ($site, $location, $object, $user)

Beschreibung:
Entsperren einer oder mehrerer bestehender Seiten oder Komponenten die auf den gleichen Content Container beruhen für die gemeinsame Nutzung durch alle Benutzer.

Bsp:
$result = unlockobject ("besttrade", "%page%/besttrade/", "index.php", "Miller");

Input-Parameter:

$site	Name der Publikation
$location	absoluter Pfad (Position des Objektes)
$object	Name des Objektes
$user	Benutzername

globale Input-Parameter:
Die folgenden globalen Input Parameter sind ebenfalls der Funktion zu übergeben:

$lang	Spracheinstellung bzw. Sprachkürzel, z.B. "en", "de"

Output:

Array	Array $result das folgende Informationen beinhaltet:
$result[result]	True/False (Erfolg der Aktion)
$result[add_onload]	JavaScript Code für das onLoad Event
$result[message]	Nachricht über den Erfolg der Aktion bzw. Fehlermeldung
$result[object]	Dateiname der Seite, Datei oder Komponente
$result[objectname]	Name der Seite, Datei oder Komponente
$result[objecttype]	Filetype bzw. File Extension der Datei

3.4.13 publishobject

Syntax:
publishobject ($site, $location, $object, $user)

Beschreibung:
Publizieren einer Seite oder Komponente. Alle gebundenen Kopien des Objektes bzw. dessen Content Containers werden ebenfalls publiziert. Gestattet die Berechtigung eines im Einsatz befindlichen Workflows die Publizierung nicht, so wird das Objekt auch nicht publiziert.

Bsp:
$result = publishobject ("besttrade", "%page%/besttrade/", "index.php", "Miller");

Input-Parameter:

$site	Name der Publikation
$location	absoluter Pfad (Position des Objektes)
$object	Name des Objektes
$user	Benutzername

globale Input-Parameter:
Die folgenden globalen Input Parameter sind ebenfalls der Funktion zu übergeben:

$lang	Spracheinstellung bzw. Sprachkürzel, z.B. "en", "de"

Output:

Array	Array $result das folgende Informationen beinhaltet:
$result[result]	True/False (Erfolg der Aktion)
$result[add_onload]	JavaScript Code für das onLoad Event
$result[message]	Nachricht über den Erfolg der Aktion bzw. Fehlermeldung

3.4.14 unpublishobject

Syntax:
unpublishobject ($site, $location, $object, $user)

Beschreibung:
Entpublizieren einer Seite oder Komponente. Link und Task Management werden automatisch ausgeführt. Alle gebundenen Kopien des Objektes bzw. dessen Content Containers werden ebenfalls entpubliziert.

Bsp:
$result = unpublishobject ("besttrade", "%page%/besttrade/", "index.php", "Miller");

Input-Parameter:

$site	Name der Publikation
$location	absoluter Pfad (Position des Objektes)
$object	Name des Objektes
$user	Benutzername

globale Input-Parameter:
Die folgenden globalen Input Parameter sind ebenfalls der Funktion zu übergeben:

$lang	Spracheinstellung bzw. Sprachkürzel, z.B. "en", "de"

Output:

Array	Array $result das folgende Informationen beinhaltet:
$result[result]	True/False (Erfolg der Aktion)
$result[add_onload]	JavaScript Code für das onLoad Event
$result[message]	Nachricht über den Erfolg der Aktion bzw. Fehlermeldung

3.4.15 getlinkedobject

Syntax:
getlinkedobject ($site, $location, $object, $cat)

Beschreibung:
Diese Funktion extrahiert alle Objekte, die auf das gegebene Objekt zeigen. Dies können Seiten-Links oder auch Komponenten-Links sein. Ist das übergebene Objekt eine Seite, so werden alle Objekte ermittelt die einen Seiten-Link auf das Objekt besitzen. Ist das übergebene Objekt eine Komponente, so werden alle Objekte gefunden die einen Komponenten-Link zu dem Objekt besitzen.

Bsp:
$result = getlinkedobject ("besttrade", "%page%/besttrade/", "index.php", "page");

Input-Parameter:

$site	Name der Publikation
$location	absoluter Pfad (Position des Objektes)
$object	Name des Objektes
$cat	optional: Objekt Kategorie [page, comp]

Output:

Array	Array $result das folgende Informationen beinhaltet:
$result	False (Aktion fehlgeschlagen)
$result[publication]	Name der Publikation bzw. Mandant in dem das Objekt existiert
$result[location]	absoluter Pfad im Filesystem (Position des Objektes)
$result[object]	Name des Objektes
$result[category]	Kategorie des Objektes [page, comp]

3.4.16 getconnectedobject

Syntax:
getconnectedobject ($site, $container)

Beschreibung:
Diese Funktion ermittelt alle Objekte, die auf dem gleichen Content Container basieren. Der Name des Content Containers eines Objektes kann mittels der Funktion "getfilename" ermittelt werden.

Bsp:
$result = getconnectedobject ("besttrade", "0000127.xml");

Input-Parameter:

$site	Name der Publikation
$container	Name des Content Containers

Output:

Array	Array $result das folgende Informationen beinhaltet:
$result	False (Aktion fehlgeschlagen)
$result[publication]	Name der Publikation bzw. Mandant in dem das Objekt existiert
$result[location]	absoluter Pfad im Filesystem (Position des Objektes)
$result[object]	Name des Objektes
$result[category]	Kategorie des Objektes [page, comp]

3.4.17 getobjectcontainer

Syntax:
getobjectcontainer ($site, $location, $object, $user)

Beschreibung:
Diese Funktion ladet den Content Container (XML-String) eines bestimmten Objektes. Das Objekt kann eine Seite, Datei, Komponente oder ein Ordner sein.
Die gewünschten Daten können mittels der Funktionen „getcontent“ oder „selectcontent“ aus dem XML-String ermittelt werden.

Bsp:
```
$xmldata = getobjectcontainer ("besttrade", "%page%/Home/", "index.php", "demouser");
```

Input-Parameter:

$site	Name der Publikation
$location	Pfad im Filesystem (Position des Objektes)
$object	Name des Objektes
$user	Benutzername

Output:

XML-String	Rückgabe des geladenen Content Containers
False	Fehler aufgetreten

3.4.18 loadcontainer

Syntax:
loadcontainer ($container)

Beschreibung:
Diese Funktion ladet den Content Container (XML-String) anhand seines Namens oder anhand seiner ID (hier wird der aktuelle in Arbeit befindliche Container als Standard angenommen und geladen).
Die gewünschten Daten können mittels der Funktionen getcontent oder selectcontent aus dem XML-String ermittelt werden.

Bsp:
```
// publizierter Zustand
$xmldata1 = loadcontainer ("00012345.xml");
// Zustand "in Arbeit"
$xmldata2 = loadcontainer ("00012345");
```

Input-Parameter:

$container	Name oder ID (mit vorangestellten Nullen) des Containers

Output:

XML-String	Rückgabe des geladenen Content Containers
False	Fehler aufgetreten

3.4.19 savecontainer

Syntax:
savecontainer ($container, $xmldata)

Beschreibung:
Diese Funktion speichert den Content Container (XML-String) anhand seines Namens oder anhand seiner ID (hier wird der aktuelle in Arbeit befindliche Container als Standard angenommen und geladen).

Bsp:
```
// publizierter Zustand
$result = savecontainer ("00012345.xml", $xmldata);
// Zustand "in Arbeit"
$result = savecontainer ("00012345", $xmldata);
```

Input-Parameter:

$container	Name oder ID (mit vorangestellten Nullen) des Containers
$xmldata	Daten des Content Containers als XML-String

Output:

True	Funktion wurde fehlerfrei ausgeführt
False	Fehler aufgetreten

3.5 Bibliothek File Pointer

Diese Funktionsbibliothek ermöglicht Ihnen den XML-Content-Container aus einem Objekt zu bestimmen bzw. diesen auch neu zu setzen. Sie müssen hierfür das Objekt (Seite oder Komponente) zuvor geladen haben, z.B. via http oder über das Filesystem mit Hilfe von loadfile.

3.5.1 getfilename

Syntax:
getfilename ($filedata, $tagname)

Beschreibung:
Der Funktion wird der Inhalt einer Seite, eines Datei-Objektes oder Komponente und der gewünschte Tag-Name – entweder content, template oder media - übergeben. Daraufhin wird der Dateiname des zugeordneten Content Container, Templates oder der Multimedia Datei zurückgegeben.

Bsp:
```
// lade eine Seite
$filedata = loadfile ("%page%/myPublication/home/", "index.php");

// Content Container auslesen
$contentcontainer  = filepointer ($filedata, "content");
```

Input-Parameter:

$filedata	Content
$tagname	Tagname [content, template, media]

Output:

Filename	Funktion wurde fehlerfrei ausgeführt und liefert Dateiname des Content Containers oder Templates
False	Fehler aufgetreten oder das Objekt wird nicht vom System verwaltet

3.5.2 setfilename

Syntax:
setfilename ($filedata, $tagname, $value)

Beschreibung:
Der Funktion wird der Inhalt einer Seite, eines Datei-Objektes oder Komponente, der Tagname und ein Wert für den File-Parameter des Tags übergeben. Nach Abarbeitung der Funktion und Überprüfung wird True oder False zurückgegeben.

Bsp:
$result = setfilename ($filedata, "template", "fullpage.page.tpl");

Input-Parameter:

$filedata	Seiten-, Datei-Objekt- oder Komponenteninhalt
$tagname	Tagname [content, template, media]
$value	(Datei)name des Content Containers, der Multimedia-Datei oder Templates

Output:

True	Funktion wurde fehlerfrei ausgeführt
False	Fehler aufgetreten

3.6 Bibliothek File Operation

Die folgenden Funktionen für File-Operationen sollten keinesfalls benutzt werden, um Objekte (Seiten, Komponenten oder Dateien) zu laden oder zu speichern.
Sie können diese Funktionen jedoch zum Laden und Speichern von XML-Content-Container verwenden, sollten Sie dies für die Entwicklung von Erweiterungen oder Anwendungen benötigen.

3.6.1 loadfile

Syntax:
loadfile ($abs_path, $filename)

Beschreibung:
Mit hilde dieser Funktion können Dateien geladen werden. Es müssen der Absolutpfad, als auch der Dateiname selbst als Parameter übergeben werden. Die Funktion wartet üblicherweise bis zu 3 Sekunden lang beim Laden von gesperrter Dateien. Wird der User-Parameter $user gesetzt, so kann die Funktion auch gesperrte Dateien des gegeben Benutzers lesen.

Bsp:
$data = loadfile ("%page%/myPublication/home/", "index.php");

Input-Parameter:

$abs_path	absoluter Pfad zur gewünschten Datei, %page% und %comp% können in der Pfadangabe verwendet werden
$filename	Dateiname

Output:

Dateinhalt	Funktion wurde fehlerfrei ausgeführt und liefert den Inhalt der Datei
False	Fehler aufgetreten

3.6.2 savefile

Syntax:
savefile ($abs_path, $filename, $filedata)

Beschreibung:
Mit savefile werden Dateien gespeichert. Hierfür müssen der Absolutpfad, der gewünschte Dateiname, als auch der Inhalt, die in die Datei geschrieben werden soll als Parameter übergeben werden. Ist die Datei gesperrt, so wird nicht gespeichert und False retourniert.

Bsp:
$result = savefile ("%page%/myPublication/home/", "index.php", "file content");

Input-Parameter:

$abs_path	absoluter Pfad zur gewünschten Datei, %page% und %comp% können in der Pfadangabe verwendet werden
$filename	Dateiname
$filedata	Inhalt, der in die Datei geschrieben werden soll

Output:

True	Funktion wurde fehlerfrei ausgeführt
False	Fehler aufgetreten

3.6.3 loadlockfile

Syntax:
loadlockfile ($user, $abs_path, $filename)

Beschreibung:
Damit können Dateien wie mit loadfile geladen werden, es wird aber zusätzlich ein Sperr-Mechanismus ausgelöst.
Diese Funktion sollte nur dann genutzt werden, wenn Daten manipuliert und wieder gespeichert werden sollen. Damit wird sicher gestellt, dass keine anderen Schreibzugriffe eines anderen Users erfolgen können. Beim Speichern muss die Funktion „savelockfile“ benutzt werden, um den Inhalt wieder freizugeben.
Es müssen der Benutzer, der absolute Pfad als auch der Dateiname selbst als Parameter übergeben werden.

Bsp:
$data = loadlockfile ("Miller", "%page%/myPublication/home/", "index.php");

Input-Parameter:

$user	Benutzer der die Datei sperrt
$abs_path	absoluter Pfad zur gewünschten Datei, %page% und %comp% können in der Pfadangabe verwendet werden
$filename	Dateiname

Output:

Dateinhalt	Funktion wurde fehlerfrei ausgeführt und liefert den Inhalt der Datei
False	Fehler aufgetreten

3.6.4 savelockfile

Syntax:
savelockfile ($user, $abs_path, $filename, $filedata)

Beschreibung:
Mit savelockfile werden Dateien gespeichert und entsperrt, die vorher mit loadlockfile geöffnet wurden. Hierfür müssen der Benutzer, der Absolutpfad, der gewünschte Dateiname, als auch der Inhalt, die in die Datei geschrieben werden soll als Parameter übergeben werden.

Bsp:
savelockfile ("Miller", "%page%/myPublication/home/", "index.php", "file content");

Input-Parameter:

$user	Benutzer der die Datei sperrt
$abs_path	absoluter Pfad zur gewünschten Datei, %page% und %comp% können in der Pfadangabe verwendet werden
$filename	Dateiname
$filedata	Inhalt, der in die Datei geschrieben werden soll

Output:

True	Funktion wurde fehlerfrei ausgeführt
False	Fehler aufgetreten

3.6.5 lockfile

Syntax:
lockfile ($user, $abs_path, $filename)

Beschreibung:
Mit lockfile werden Dateien von einem bestimmten Benutzer gesperrt und stehen für dessen exklusive Nutzung zur Verfügung. Hierfür müssen der Benutzer, der Absolutpfad, der gewünschte Dateiname als Parameter übergeben werden.

Bsp:
lockfile ("Miller", "%page%/myPublication/home/", "index.php");

Input-Parameter:

$user	Benutzer der die Datei sperrt
$abs_path	absoluter Pfad zur gewünschten Datei, %page% und %comp% können in der Pfadangabe verwendet werden
$filename	Dateiname

Output:

True	Funktion wurde fehlerfrei ausgeführt
False	Fehler aufgetreten

3.6.6 unlockfile

Syntax:
unlockfile ($user, $abs_path, $filename)

Beschreibung:
Mit unlockfile werden Dateien entsperrt, die vorher mit lockfile gesperrt oder mit loadlockfile geöffnet wurden. Hierfür müssen der Benutzer, der Absolutpfad, der gewünschte Dateiname als Parameter übergeben werden.

Bsp:
unlockfile ("Miller", "%page%/myPublication/home/", "index.php");

Input-Parameter:

$user	Benutzer der die Datei sperrt
$abs_path	absoluter Pfad zur gewünschten Datei, %page% und %comp% können in der Pfadangabe verwendet werden
$filename	Dateiname

Output:

True	Funktion wurde fehlerfrei ausgeführt
False	Fehler aufgetreten

3.6.7 deletefile

Syntax:
deletefile ($location, $file, $recursive)

Beschreibung:
Mit deletefile können Dateien und (leere) Ordner gelöscht werden. Es wird der Pfad der gewünschten Datei übergeben, der Dateiname, und ein Parameter "Rekursiv", der entweder (0) oder (1) beträgt. Wenn recursive 1 gesetzt wurde, wird der gesamte Inhalt des Ordners behandelt, also auch Unterverzeichnisse und deren Dateien, bei 0 werden nur die Dateien des angesprochenen Ordners (falls leer) entfernt.

Bsp:
deletefile ("%page%/myPublication/home/", "index.php", 0);

Input-Parameter:

$abs_path	absoluter Pfad zur gewünschten Datei, %page% und %comp% können in der Pfadangabe verwendet werden
$file	Dateiname
$recursive	0 oder 1, je nachdem ob sich der Vorgang auch auf Unterverzeichnisse auswirken soll

Output:

True	Funktion wurde fehlerfrei ausgeführt
False	Fehler aufgetreten

3.6.8 appendfile

Syntax:
append ($abs_path, $filename, $filedata)

Beschreibung:
Mit appendfile können Inhalte an Dateien angefügt werden. Die Funktion arbeitet wie savefile, der Unterschied besteht allerdings darin, dass bereits vorhandene Daten nicht überschrieben, sondern ergänzt werden. Hierfür müssen der Absolutpfad, der gewünschte Dateiname, als auch der Inhalt, die in die Datei geschrieben werden soll als Parameter übergeben werden.

Bsp:
appendfile ("%page%/myPublication/home/", "index.php", "© 2003 …");

Input-Parameter:

$abs_path	absoluter Pfad zur gewünschten Datei, %page% und %comp% können in der Pfadangabe verwendet werden
$filename	Dateiname
$filedata	Inhalt, der an die Datei angefügt werden soll

Output:

True	Funktion wurde fehlerfrei ausgeführt
False	Fehler aufgetreten

3.7 Bibliothek Edit Content

Die folgenden Funktionen bieten Ihnen die Möglichkeit Inhalte aus XML-Content-Container zu lesen und zu schreiben. Sie können optional auch mit anderen Technologien, die mit XML umgehen können, die Inhalte der Container abfragen. Die Bibliothek Edit Content bietet Ihnen jedoch eine sehr einfache als auch performante Methode hierfür.

3.7.1 setxmlparameter

Syntax:
setxmlparameter ($xmldata, $parameter, $value)

Beschreibung:
Setzt den Wert eines bestimmten Parameters innerhalb der XML-Deklaration (1.Zeile).

Bsp:
$xmldata = setxmlparameter ($xmldata, "encoding", "UTF-8");

Input-Parameter:

$xmldata	XML-String der übergeben und manipuliert werden soll
$parameter	Name des Parameter dessen Wert geändert werden soll
$value	Wert des Paramaters

Output:

XML-String	Rückgabe des manipulierten XML-Strings
False	Fehler aufgetreten

3.7.2 getcontent

Syntax:
getcontent ($xmldata, $tag)

Beschreibung:
Holt den XML-Content aus dem Content Container, der sich innerhalb der Tags $tag befindet. Ein Array mit allen gefundenen Inhalten bzw. Childs wird zurückgegeben und kann in einer Array-Variable gespeichert und weiterverwendet werden.

Bsp:
```
// hole alle text-childs aus Content Container
$text_array = getcontent ($xmldata, "<text>");

// ausgeben aller Text-Childs
foreach ($text_array as $text) echo $text;
```

Input-Parameter:

$xmldata	XML-String der übergeben und durchsucht werden soll
$tag	XML-Tag der die Information bzw. Childs umschliesst

Output:

Array	Array mit allen gefunden Inhalten, der erste Wert/Inhalt kann mit Array[0] angesprochen werden
False	Fehler aufgetreten

3.7.3 getxmlcontent

Syntax:
getxmlcontent ($xmldata, $tag)

Beschreibung:
Holt den XML-Content aus dem Content Container, der sich innerhalb der Tags $tag befindet und belässt im Unterschied zu getcontent die Tags im Rückgabewert (Array). Ein gesamter Node (well-formed) wird daher zurückgeliefert.
Ein Array mit allen gefundenen Inhalten bzw. Childs wird zurückgegeben und kann in einer Variable vom Typ Array gespeichert und weiterverwendet werden.

Bsp:
```
$text_array = getxmlcontent ($xmldata, "<text>");
foreach ($text_array as $text) echo $text;
```

Input-Parameter:

$xmldata	XML-String der übergeben und durchsucht werden soll
$tag	XML-Tag der die Information bzw. Childs umschliesst

Output:

Array	Array mit allen gefunden Inhalten, der erste Wert/Inhalt kann mit Array[0] angesprochen werden
False	Fehler aufgetreten

3.7.4 selectcontent

Syntax:
selectcontent ($xmldata, $parenttag, $childtag, $childvalue)

Beschreibung:
Holt jenen XML-Content bestimmt durch $parenttag aus dem Content-Container, der innerhalb des Childtags $childtag einen bestimmten Wert $childvalue aufweist.
Ein Array mit allen gefundenen Inhalten wird zurückgegeben und kann in einer Array-Variable gespeichert und weiterverwendet werden.

Bsp:
Auszug aus dem Content Container:

```
…..
<text>
  <text_id>summary</text_id>
  <textuser>editor1</textuser>
  <textcontent>This is my summary!</textcontent>
</text>
…..

// hole alle Text-Childs mit der id=summary
$text_array = selectcontent ($xmldata, "<text>", "<text_id>", "summary");

// extrahiere das Summary aus dem gefundenen Inhalt
foreach ($text_array as $text)
{
  $summary = getcontent ($text, "<textcontent>");
}
```

Input-Parameter:

$xmldata	XML-String der übergeben und durchsucht werden soll
$parenttag	XML-Tag der die Information bzw. das Child beinhaltet
$childtag	optional: XML-Tag der die Information umschliesst, die einen gewissen Wert besitzen muss
$childvalue	optional: Wert der Bedingung, das WildCard Zeichen * kann am Anfang und/oder am Ende des Ausdruckes verwendet werden und ist Platzhalter für beliebige weitere Zeichen.

Output:

Array	Array mit allen gefunden Inhalten, der erste Wert/Inhalt kann mit Array[0] angesprochen werden
False	Fehler aufgetreten

3.7.5 selectxmlcontent

Syntax:
selectxmlcontent ($xmldata, $parenttag, $childtag, $childvalue)

Beschreibung:
Holt jenen XML-Content definiert durch $parenttag aus dem Content-Container, der innerhalb eines Childtags $childtag einen bestimmten Wert $childvalue aufweist. Im Unterschied zu getcontent werden die Parent-Tags im Rückgabewert (Array) belassen.
Ein Array mit allen gefundenen Inhalten wird zurückgegeben und kann in einer Array-Variable gespeichert und weiterverwendet werden.

Bsp:
Auszug aus dem Content Container:

```
…..
<text>
  <text_id>summary</text_id>
  <textuser>editor1</textuser>
  <textcontent>This is my summary!</textcontent>
</text>
…..

// hole alle Text-Childs mit der id=summary
$text_array = selectxmlcontent ($xmldata, "<text>", "<text_id>", "summary");

// extrahiere das Summary aus dem gefundenen Inhalt
foreach ($text_array as $text)
{
  $summary = getcontent ($text, "<textcontent>");
}
```

Input-Parameter:

$xmldata	XML-String der übergeben und durchsucht werden soll
$parenttag	XML-Tag der die Information bzw. das Child beinhaltet
$childtag	optional: XML-Tag der die Information umschliesst, die einen gewissen Wert besitzen muss
$childvalue	optional: Wert der Bedingung, das WildCard Zeichen * kann am Anfang und/oder am Ende des Ausdruckes verwendet werden und ist Platzhalter für beliebige weitere Zeichen.

Output:

Array	Array mit allen gefunden Inhalten, der erste Wert/Inhalt kann mit Array[0] angesprochen werden
False	Fehler aufgetreten

3.7.6 deletecontent

Syntax:
deletecontent ($xmldata, $tagname, $condtag, $condvalue)

Beschreibung:
Löscht den gesamten XML-Content definiert durch den Tag $tagname. Als Kriterium für die Auswahl der zu löschenden Childs wird das entsprechende XML-Childtag $condtag und die umschlossene Information $condvalue als Bedingung mitgeschickt.

Bsp:
Auszug aus dem Content Container:

```
…..
<text>
  <text_id>bedingung</text_id>
  <textuser>editor1</textuser>
  <textcontent>This is my summary!</textcontent>
</text>
…..

$xmldata = deletecontent ($xmldata, "<text>", "<text_id>", "bedingung");
```

Input-Parameter:

$xmldata	XML-String der übergeben wird
$parenttag	XML-Tag der die Information bzw. Childs umschliesst, die aus dem Content Container entfernt werden sollen
$condtag	optional: Name des Parameters (XML-Child) das der Bedingung unterliegt
$condvalue	optional: Wert der Bedingung, die erfüllt werden muss

Output:

XML-String	Rückgabe des manipulierten XML-Strings
False	Fehler aufgetreten

3.7.7 setcontent

Syntax:
setcontent ($xmldata, $parenttagname, $tagname, $contentnew, $condtag, $condvalue)

Beschreibung:
Ein XML-String wird übergeben und innerhalb eines bestimmten Parent Nodes ($parenttagname) wird überprüft, ob ein bestimmter Parameter ($condtag) existiert und einen bestimmter Wert ($condvalue) aufweist. Ist die Bedingung erfüllt, wird der Wert des Parameters $tagname durch einen neuen Wert $contentnew ersetzt.

Bsp:
Auszug aus dem Content Container:

```
…..
<text>
  <text_id>bedingung</text_id>
  <textuser>editor1</textuser>
  <textcontent>This is should set!<textcontent>
</text>
…..
```

```
$xmldata = setcontent ($xmldata, "<text>", "<textcontent>", "This is my new value!", "<text_id>", "bedingung");
```

Input-Parameter:

$xmldata	XML-String der übergeben und modifiziert werden soll
$parenttagname	optional: XML-Parenttag
$tagname	optional: XML-Childtag, dessen Wert ersetzt werden soll (wenn Bedingung erfüllt)
$contentnew	Neuer Wert für den XML-Childtag $tagmame
$condtag	optional: Name des Parameters der die Bedingung erfüllen muss
$condvalue	optional: Wert des Parameters für die Bedingung

Output:

XML-String	Rückgabe des manipulierten XML-Strings
False	Fehler aufgetreten

3.7.8 updatecontent

Syntax:
updatecontent ($xmldata, $xmlnode, $xmlnodenew)

Beschreibung:
Alle XML-String $xmlnode wird durch einen neuen String $xmlnodenew in $xmldata ersetzt. Diese Methode ist schneller als setcontent, wenn der aktualisierende XML Node bereits aus dem Container extrahiert wurde.

Bsp:
Auszug aus dem Content Container:

```
…..
<text>
  <text_id>bedingung</text_id>
  <textuser>editor1</textuser>
  <textcontent>This is old content!<textcontent>
</text>
…..
```

$xmldata = updatecontent ($xmldata, "<textcontent>This is old content!<textcontent> ", "<textcontent>This is my new content!<textcontent>");

Input-Parameter:

$xmldata	XML-String der übergeben und modifiziert werden soll
$xmlnode	zu ersetzender XML-String (Node bzw. Substring von $xmldata)
$xmlnodenew	optional: neuer XML-String, wenn leer, so wird der bestehende XML-String entfernt.

Output:

XML-String	Rückgabe des manipulierten XML-Strings
False	Fehler aufgetreten

3.7.9 insertcontent

Syntax:
insertcontent ($xmldata, $insertxmldata, $tagname)

Beschreibung:
Fügt einen XML-String (Child Node) vor dem Ende des übergebenen XML-Parenttags ein. Der modifizierte XML-String wird zurückgegeben.

Bsp:
Auszug aus dem Content Container:

```
…..
  <articletextlist>
    <text>
      <text_id>art1:summary</text_id>
      <textuser>editor1</textuser>
      <textcontent>This is my summary!</textcontent>
    </text>
---------- hier wird ein Child Node eingefügt ------------
    <text>
      <text_id>art1:longtext</text_id>
      <textuser>editor1</textuser>
      <textcontent>This is my summary!</textcontent>
    </text>
-------------------------------------------------------------
  </articletextlist>
…..
```

```
$xmldata = insertcontent ($xmldata, $insertxmldata, "<articletextlist>");
```

Input-Parameter:

$xmldata	XML-String der übergeben und modifiziert werden soll
$insertxmldata	XML-String der eingesetzt wird
$tagname	optional: XML-Parenttag an dessen Ende eingesetzt werden soll

Output:

XML-String	Rückgabe des manipulierten XML-Strings
False	Fehler aufgetreten

3.7.10 addcontent

Syntax:
addcontent ($xmldata, $sub_xmldata, $grandtagname, $condtag, $condvalue, $parenttagname, $tagname, $contentnew)

Beschreibung:
Innerhalb eines Parent Nodes wird ein Child Node hinzugefügt, sofern ein Wert im darüberliegenden Grandparent Node die Bedingung erfüllt. Im Child Node kann auf Wunsch gleichzeitig ein Wert gesetzt werden. Der modifizierte XML-String wird zurückgegeben.

Bsp:
Auszug aus dem Content Container:

```
…..
<article>
  <article_id>art1</article_id>
  <articletitle></articletitle>
  <articledatefrom></articledatefrom>
  <articledateto></articledateto>
  <articlestatus>active</articlestatus>
  <articleuser></articleuser>
  <articletextlist>
    <text>
      <text_id>art1:summary</text_id>
      <textuser>editor1</textuser>
      <textcontent>This is my summary!</textcontent>
    </text>
---------- hier wird ein Child Node eingefügt ------------
    <text>
      <text_id>art1:longtext</text_id>
      <textuser>editor1</textuser>
      <textcontent>This is my summary!</textcontent>
    </text>
-------------------------------------------------------------
  </articletextlist>
</article>
…..
```

```
$xmldata = addcontent ($xmldata, $sub_xmldata, "<article>", "<article_id>", "art1",
"<articletextlist>", "<text_id>", "art1:longtext");
```

Input-Parameter:

$xmldata	XML-String der übergeben und modifiziert werden soll
$sub_xmldata	XML-String der eingebettet werden soll
$grandtagname	Enthält den XML-Childtag, in dem $sub_xmldata eingebettet werden soll
$condtag	optional: Name des Parameters der überprüft werden soll
$condvalue	optional: Wert des Parameters der überprüft werden soll
$parenttagname	optional: XML-Childtag, in dem $sub_xmldata eingebettet werden soll
$tagname	optional: Childtag des eingebetteten XML-String
$contentnew	optional: Content für den Tag $tagname

Output:

XML-String	Rückgabe des manipulierten XML-Strings
False	Fehler aufgetreten

3.8 Bibliothek Meta Data Generator

Diese Funktionsbibliothek ermöglicht Ihnen Keyword-Listen, die Description aus einem Inhalt zu erzeugen. Dies kann zur automatischen Erzeugung bzw. Befüllung von Metadaten verwendet werden.
Es können Meta Daten aus Multimedia-Dateien ausgelesen und im Container eines Objektes gespeichert werden.

3.8.1 getkeywords

Syntax:
getkeywords ($text, $language, $charset)

Beschreibung:
Der Funktion wird der Inhalt übergeben. Damit werden alle Keywords aus dem Text ermittelt und als Keyword-Liste zurückgegeben.

Bsp:
$keywords = getkeywords ("This is just a short text.", "en", "UTF-8");

Input-Parameter:

$text	Content als String
$language	optional: Sprache [en, de], Standard ist "en"
$charset	optional: Character Set, Standard ist "UTF-8"

Output:

Keywords	Komma-getrennte Liste aller Keywords
False	Fehler aufgetreten

3.8.2 getdescription

Syntax:
getdescription ($text, $charset)

Beschreibung:
Dieser Funktion wird der Inhalt übergeben. Daraufhin wird eine Kurzbeschreibung aus dem Text ermittelt und zurückgegeben.

Bsp:
$keywords = getdescription ("This is just a short text.", "UTF-8");

Input-Parameter:

$text	Content als String
$charset	optional: Character Set, Standard ist "UTF-8"

Output:

Keywords	Kurzbeschreibung des Inhaltes
False	Fehler aufgetreten

3.8.3 injectmetadata

Syntax:
injectmetadata ($site, $location, $object, $mediafile, $mapping, $user)

Beschreibung:
Diese Funktion benötigt den Pfad zu einem Multimedia Objekt, den Dateiname des Objektes oder den Dateinamen der Multimedia-Datei sowie ein Mapping für die Speicherung der Daten. Damit wird der Text aus den Meta Daten anhand des Mappings ausgelesen und in der entsprechenden Text-ID des Containers geschrieben.
ACHTUNG: Bestehende Daten des Containers werden damit überschrieben!

Bsp:
```
// Mapping Definition (Meta Data Name -> Text-ID)
// Dublin Core
$mapping['dc:title'] = "Title";
$mapping['dc:subject'] = "Keywords";
$mapping['dc:description'] = "Description";
$mapping['dc:creator'] = "Creator";
$mapping['dc:rights'] = "Copyright";
// Adobe PhotoShop
$mapping['photoshop:SupplementalCategories'] = "Categories";
// Image Resolution defines Quality [Print, Web]
$mapping['hcms:quality'] = "Quality";

$result = injectmetadata ("Publication", "%comp%/test/", "image.jpg", "", $mapping, "Miller");
```

Input-Parameter:

$site	Name der Publikation
$location	Pfad im Filesystem (Position des Objektes)
$object	optional: Name des Objektes (Komponente)
$mediafile	oder optional: Name der Multimedia Datei
$mapping	Mapping Array (Meta Data Name -> Text-ID)
$user	Benutzername

Output:

True	Meta Daten wurden erfolgreich gespeichert
False	Fehler aufgetreten

3.9 Bibliothek Notifications

Diese Funktionsbibliothek versendet automatisierte Nachrichten an einen Benutzer anhand von Grenzwerten eines bestimmten Feldes.
Der Benutzer erhält eine vorformatierte Nachricht mit Information (Links) zu allen Objekten, die in den Suchbereich (Datumsober und -untergrenze) fallen.

3.9.1 licensenotification

Syntax:
licensenotification ($site, $cat, $folderpath, $text_id, $date_begin, $date_end, $user)

Beschreibung:
Der Funktion ermittelt alle Objekte aufgrund des vorgegebenen Suchbereiches (Lokation und Datumsgrenzwerte) und versendet eine E-Mail an einen bestimmten Benutzer mit den Links zu allen betroffenen Objekten.

Bsp:
```
// set language for mail message
$lang = "en";

// send mail to Miller
$result = licensenotification ("Demo-DAM", "%comp%/images/", "comp", "valid_date",
"2012-09-01", "2012-09-30", "Miller");
```

Input-Parameter:

$site	Name der Publikation
$cat	Objekt Kategorie [page, comp]
$folderpath	Pfad für die Defintion des Suchbereiches
$text_id	Text ID des Feldes auf das die Suche angewendet werden soll
$date_begin	Startdatum für die Suche (YYYY-MM-DD)
$date_end	Endedatum für die Suche (YYYY-MM-DD)
$user	Benutzername

Output:

True	Mail wurde erfolgreich gesendet
False	Fehler aufgetreten

4 Komponenten und Applikationen

Wenn Anwendungen in Komponenten integriert werden und Variablen aus einer Seite an eine Komponente übergeben werden müssen, so ist auf folgendes zu achten:
Die Komponenten müssen über das Dateisystem eingebunden werden (nicht via HTTP).
Alle Variablen die an die Komponente zu übergeben sind, sind in der Komponente als global zu definieren.

Bsp:
Eine Seite übergibt eine Variable an eine Komponente.

Hier der Code der Seite:

```
<html>
<head>
<title>page</title>
<head>
<body>
<?php $test="This is just a test!"; ?>
[hyperCMS:components id='component']
</body>
</html>
```

Der Code der Komponente muss wie folgt aussehen:

```
<p>
<?php
global $test;
echo $test;
?>
</p>
```

Im Beispiel wird die Variable $test bzw. dessen Wert "This is just a test!" von der Komponente übernommen und in der Präsentation angezeigt.

5 Database Connectivity

Die Database Connectivity des hyper Content & Digital Asset Management Servers erlaubt die Anbindung von diversen Datenbanken zur Speicherung und Entnahme von Inhalten. Damit können z.B. relationale Datenbanken als externes Content Repository genutzt werden.
Zu diesem Zweck ist je Template der entsprechende hyperCMS-Tag für die Database Connectivity einzufügen, der auf ein DB-Connect File verweißt.
In diesem File werden Funktionen hinterlegt, die hyperCMS aufruft, sofern das Template auf die Funktionsdatei zeigt.
Die Inhalte werden aus der Datenbank gelesen und dem Redakteur angezeigt. Verändert der Redakteur die Inhalte, so können diese auch wieder in die Datenbank geschrieben werden. Es können für Lese- und Schreibzugriffe auch verschiedene Datenbanken aufgerufen werden. Die Funktionen im DB-Connect File bieten nur die Hülle bzw. standardisierte Schnittstelle zu hyperCMS, die durch den Programmierer befüllt werden muss.

Das Thema der Datenbankintegration ist komplex und individuell zu behandeln, da auch bereits bestehende Datenbanken und deren Informationen integriert werden können. hyperCMS gibt kein ER-Modell vor bzw. legt sich auf keine speziellen Datenbank-Produkte fest. Generell kann gesagt werden, dass alle Möglichkeiten von PHP ausgeschöpft werden können, um sich zu diversen Datenquellen zu verbinden.

Neben den notwendigen Parameter für Queries auf relationale Datenbanken wird auch der gesamte Content Conatiner als XML-String übergeben. Damit könnten Dokumente bzw. Inhalte aus den Content Repository auch als Node in XML-Datenbanken abgelegt werden.

Sie selbst bestimmen, wohin Sie Ihre Daten speichern bzw. woher Sie diese holen. Mit PHP besitzen eine mächtige Sprache, die Ihnen Zugriff auf alle gängigen Datenbanken bietet.

Mehr Information zu den Funktionen von PHP finden Sie unter: http://www.php.net

5.1 Erstellen einer Database Connectivity

Möchten Sie eine Database Connectivity erstellen, so erstellen Sie eine Kopie des Files db_connect_default.php, dieses finden Sie in dem gewählten Root-Verzeichnis für die Ablage der Management Daten unter dem folgenden weiterführenden Pfad: /data/db_connect/
Die Kopie des Files nennen Sie z.B. nach der Datenbank, die Sie anbinden möchten.

Danach öffnen Sie die Datei und erhalten Einsicht in die Funktionen. Im Source Code finden Sie auch eine Beschreibung der Funktionen und der übergebenen Parameter als auch des Outputs.

Exemplarisch soll hier ein Lesezugriff auf eine MySQL Datenbank für einen Text-Inhalt dargestellt werden. Wir gehen davon aus, dass in einem Table "TextContent" die Inhalte mit dem Primary Key "container_id" und "text_id", dem Text-Inhalt "Text" sowie dem Text-Typ "Type" vorliegen. Der User sowie die Artikel ID wird nicht gesondert gespeichert, dies ist für die Eindeutigkeit des Inhalts auch nicht notwendig, denn die ID des Content Containers als auch die ID des Elements reichen als Primärschlüssel aus.

```php
// ============================ db connect  ==============================
// this file allows you to access a database using the full PHP functionality.
// you can read or write data from or into a database:

// ========================= read from database ==========================
// the following parameter values are passed to each function for
// retrieving data from the database:
// name of the site: $site [string]
// name of the content container: $container_id [string] (is unique
// inside hyperCMS over all sites)
// content container: $container_content [XML-string]
// identification name: $id [string]

// ------------------------------- text -----------------------------------
// if content is text
function db_read_text ($site, $content_id, $container_content, $id, $art_id, $user)
{
  //-------------------------------------------------------------------------
  // input variables: $id [string], optional: $artid [string], $user [string
  // return value: $text [array]
  //             the array must exactly look like this:
  //             $text[text], optional: $text[type]
  //             constraints/accepted values for article type, see note below
  // note: special characters in $text are escaped into
  //       their html/xml equivalents.
  //       you can decide between unformatted, formatted and
  //       optional text using $type:
  //       unformatted text: $text[type] = textu
  //       formatted text: $text[type] = textf
  //       text option: $text[type] = textl
  //-------------------------------------------------------------------------

  $user = "username";
  $password  = "password";
  $database = "database";

  // connect to database
  mysql_connect ("localhost", $user, $password);
  @mysql_select_db ($database) or die ("Unable to select database");

  // fire SQL-query
  $result = mysql_query ("SELECT Text, Type FROM TextContent WHERE
          container_id=$container_id AND text_id=$id);

  // count returned rows, must be 1 if unique
  $num_of_rows = mysql_num_rows ($result);

  // get the result into an array namend $row
  if ($num_of_rows == 1)
  {
    $row = mysql_fetch_row ($result);

    // set values
    $text[text] = $row[0];
    $text[type] = $row[1];
  }
  else $text = false;

  // close connection
  mysql_close ();

  // return result
  return $text;
}
```

6 Event System

Der hyper Content & Digital Asset Management Server beinhaltet ein Event System, das eine automatisierte Ausführung von Aktionen passierend auf Ereignissen im System ermöglicht. Damit lassen sich z.B. manuelle Vorgänge automatisieren.
Events werden meist durch den Benutzer durch Wahl einer Aktion gestartet, z.B. das Publizieren einer Seite. Ist der entsprechende Event aktiviert, so wird nach erfolgreicher Ausführung des Publikationsprozesses der Seite das Event "onpublishobject" aufgerufen. Die darin definierten Funktionen werden sodann ausgeführt.
Die Events des Event Systems können in der Datei "hypercms_eventsys.inc.php" definiert werden. Diese befindet sich im internen Repository im Ordner "eventsystem". In dieser Datei befinden sich auch weitere wichtige Hinweise, die bei der Ausführung von Events zu beachten sind.
Das Event System ist innerhalb des gesamten Management Systems über alle Publikationen gültig. Das System ist Bestandteil des hyperCMS APIs und wird somit bei jedem Aufruf einer Funktion des APIs ausgeführt.
Events lassen sich in der Datei "hypercms_eventsys.inc.php" aktivieren als auch deaktivieren, sodass der Einsatz der darin definierten Events leicht gesteuert werden kann.
Bei allen Events wird zwischen PRE- und POST-Events unterschieden. Das PRE-Event wird vor der eigentlichen Ausführung der aufgerufenen Aktion gestartet, während das POST-Event nach der erfolgreichen Ausführung der Aktion aufgerufen wird.

Bsp:
Beim Publizieren eines Objektes soll automatisch auch die Seite "index.php" die sich an der gleichen Position befindet publiziert werden, da diese z.B. ein über ein hyperCMS Script generiertes Verzeichnis aller Objekte des gleichen Ordners beinhaltet.

```
// ------------------------------------ on publish object POST event ------------------------------------
function onpublishobject_post ($site, $cat, $location, $object, $user)
{
  // load configuration
  include_once ("config.inc.php");

  // hide the event used in your action (1) otherwise execute event (0)
  $eventsystem[hide] = 1;

  // insert your program code here
  $result = publishobject ($site, $location, "index.php", $user);

  // return true if successful
  if ($result[result) == true) return true;
  else return false;
}
```

7 Liste der hyperCMS API Funktionen

Die Dokumentation aller API Funktionen sind auch auf unserer Website hypercms.com in der aktuellen Version verfügbar. Sie können die Dokumentation ihrer installierten Version als Hilfe und im Browser anzeigen. Nutzen Sie hierzu das ?-Icon im Template Editor um die Referenz aller hyperCMS Tags und API Funcktionen einzusehen.

8 hyperCMS API Function Reference

8.1 Main API Functions

8.1.1 setxmlparameter

Syntax:
setxmlparameter ($xmldata, $parameter, $value)

Input parameters:
$xmldata ... XML content container
$parameter ... paramater name
$value ... paramater value

Output:
XML content container / false on error

Description:
set parameter values in XML declaration (e.g. encoding):
encoding="UTF-8"

8.1.2 getcontent

Syntax:
getcontent ($xmldata, $starttagname)

Input parameters:
$xmldata ... XML content container
$starttagname ... tag name

Output:
result array with the content of the requested XML node (tag) / false on error

Description:
<tagname>content</tagname>
extracts the content between the given $starttagname xml-tags.
only this function will decode special characters (&, <, >) in the content and removes CDATA.
getcontent will only decode values if they are non-xml and non_html. so content inside child nodes
including tags won't be decoded.
wild card character "*" can be used at the end of $starttagname.

8.1.3 geticontent

Syntax:
geticontent ($xmldata, $starttagname)

Input parameters:

$xmldata ... XML content container
$starttagname ... tag name

Output:
result array with the content of the requested XML node (tag) / false on error

Description:
CASE-Insensitive version (XML parser are however always case-sensitive!)
<tagname>content</tagname>
extracts the content between the given $starttagname xml-tags.
only this function will decode special characters (&, <, >) in the content and removes CDATA.
getcontent will only decode values if they are non-xml and non_html. so content inside child nodes
including tags won't be decoded.
wild card character "*" can be used at the end of $starttagname

8.1.4 getxmlcontent

Syntax:
getxmlcontent ($xmldata, $starttagname)

Input parameters:
$xmldata ... XML content container
$starttagname ... tag name

Output:
result array with the content of the requested XML node (tag) / false on error

Description:
<tagname>content</tagname>
extracts the content together with the $starttagname xml tags
this function will NOT decode special characters like function getcontent!
wild card character "*" can be used at the end of $starttagname

8.1.5 getxmlicontent

Syntax:
getxmlicontent ($xmldata, $starttagname)

Input parameters:
$xmldata ... XML content container
$starttagname ... tag name

Output:
result array with the content of the requested XML node (tag) / false on error

Description:
CASE-Insensitive version (XML parser are always case-sensitive!)
<tagname>content</tagname>
extracts the content together with the $starttagname xml tags
this function will NOT decode special characters like function getcontent!
wild card character "*" can be used at the end of $starttagname

8.1.6 selectcontent

Syntax:
selectcontent ($xmldata, $starttagname, $startcondtag, $condvalue)

Input parameters:
$xmldata ... XML content container
$starttagname ... tag name of requested XML node
$startcondtag ... tag holding the conditional value inside the given starttagname
$condvalue ... conditional value

Output:
result array with the content of the requested XML node (tag) / false on error

Description:
<tagname>
.....
<condtag>condvalue</condtag>
.....
</tagname>
extracts the content between the given $starttagname xml tags where the child xml tag $startcondtag
value is equal with the target value $condvalue
wild card character "*" can be used at the end of $starttagname
wild card character "*" can be used at begin and end of $condvalue
Be Aware: $startcondtag must be a child of $starttagname !!!

8.1.7 selecticontent

Syntax:
selecticontent ($xmldata, $starttagname, $startcondtag, $condvalue)

Input parameters:
$xmldata ... XML content container
$starttagname ... tag name of requested XML node
$startcondtag ... tag holding the conditional value inside the given starttagname
$condvalue ... conditional value

Output:
result array with the content of the requested XML node (tag) / false on error

Description:
CASE-Insensitive version (XML parser are always case-sensitive!)
<tagname>
.....
<condtag>condvalue</condtag>
.....
</tagname>
extracts the content between the given $starttagname xml tags where the child xml tag $startcondtag
value is equal with the target value $condvalue
wild card character "*" can be used at the end of $starttagname
wild card character "*" can be used at begin and end of $condvalue

8.1.8 selectxmlcontent

Syntax:
selectxmlcontent ($xmldata, $starttagname, $startcondtag, $condvalue)

Input parameters:
$xmldata ... XML content container
$starttagname ... tag name of requested XML node
$startcondtag ... tag holding the conditional value inside the given starttagname

$condvalue ... conditional value

Output:
result array with the content of the requested XML node (tag) / false on error

Description:
<tagname>
.....
<condtag>condvalue</condtag>
.....
</tagname>
extracts the content between the given $starttagname xml tags where the child xml tag $startcondtag
value is equal with the target value $condvalue
wild card character "*" can be used at begin and end of $condvalue
Be Aware: $startcondtag must be a child of $starttagname !!!

8.1.9 selectxmlicontent

Syntax:
selectxmlicontent ($xmldata, $starttagname, $startcondtag, $condvalue)

Input parameters:
$xmldata ... XML content container
$starttagname ... tag name of requested XML node
$startcondtag ... tag holding the conditional value inside the given starttagname
$condvalue ... conditional value

Output:
result array with the content of the requested XML node (tag) / false on error

Description:
CASE-Insensitive version (XML parser are always case-sensitive!)
<tagname>
.....
<condtag>condvalue</condtag>
.....
</tagname>
extracts the content between the given $starttagname xml tags where the child xml tag $startcondtag
value is equal with the target value $condvalue
wild card character "*" can be used at begin and end of $condvalue

8.1.10 deletecontent

Syntax:
deletecontent ($xmldata, $starttagname, $startcondtag, $condvalue)

Input parameters:
$xmldata ... XML content container
$starttagname ... tag name of requested XML node
$startcondtag ... tag holding the conditional value inside the given starttagname
$condvalue ... conditional value

Output:
XML content container / false on error

Description:

<tagname>
<condtag>condvalue</condtag>
</tagname>
deletes the whole xml content including <tagname>
wild card character "*" can be used at begin and end of $condvalue

8.1.11 deleteicontent

Syntax:
deleteicontent ($xmldata, $starttagname, $startcondtag, $condvalue)

Input parameters:
$xmldata ... XML content container
$starttagname ... tag name of requested XML node
$startcondtag ... tag holding the conditional value inside the given starttagname
$condvalue ... conditional value

Output:
XML content container / false on error

Description:
CASE-Insensitive version (XML parser are always case-sensitive!)
<tagname>
<condtag>condvalue</condtag>
</tagname>
deletes the whole xml content including <tagname>
wild card character "*" can be used at begin and end of $condvalue

8.1.12 setcontent

Syntax:
setcontent ($xmldata, $startparenttagname, $starttagname, $contentnew, $startcondtag="", $condvalue="")

Input parameters:
$xmldata ... XML content container
$startparenttagname ... parent tag name
$starttagname ... tag name of XML node for the new content
$contentnew ... new XML node to be inserted
$startcondtag ... tag holding the conditional value inside the given starttagname
$condvalue ... conditional value

Output:
XML content container / false on error

Description:
<parenttagname>
<condtag>condvalue</condtag>
<tagname>contentnew</tagname>
</parenttagname>
$xmldata = data string to be parsed
$startparenttagname = name of the tag that is a parent node of starttagname (necessary if condition has been set!)
$starttagname = name of the tag (child node)
$contentnew = the content that will be inserted between the child tags $starttagname
$startcondtag = child xml tag where condition will be set
$condvalue = value of the condition
wild card character "*" can be used at begin and end of $condvalue

8.1.13 seticontent

Syntax:
seticontent ($xmldata, $startparenttagname, $starttagname, $contentnew, $startcondtag, $condvalue)

Input parameters:
$xmldata ... XML content container
$startparenttagname ... parent tag name
$starttagname ... tag name of XML node for the new content
$contentnew ... new XML node to be inserted
$startcondtag ... tag holding the conditional value inside the given starttagname
$condvalue ... conditional value

Output:
XML content container / false on error

Description:
CASE-Insensitive version (XML parser are always case-sensitive!)
<parenttagname>
<condtag>condvalue</condtag>
<tagname>contentnew</tagname>
</parenttagname>
$xmldata = data string to be parsed
$startparenttagname = name of the tag that is the parent node of starttagname (necessary if condition has been set!)
$starttagname = name of the tag (child node)
$contentnew = the content that will be inserted between the child tags $starttagname
$startcondtag = child xml tag where condition will be set
$condvalue = value of the condition
wild card character "*" can be used at begin and end of $condvalue

8.1.14 setcontent_fast

Syntax:
setcontent_fast ($xmldata, $startparenttagname, $starttagname, $contentnew, $startcondtag="", $condvalue="")

Input parameters:
$xmldata ... XML content container
$startparenttagname ... parent tag name
$starttagname ... tag name of XML node for the new content
$contentnew ... new XML node to be inserted
$startcondtag ... tag holding the conditional value inside the given starttagname
$condvalue ... conditional value

Output:
XML content container / false on error

Description:
function designed for link management, extremely fast but with limitations (only CASE-Sensitive!)
<parenttagname>
<condtag>condvalue</condtag>
<tagname>contentnew</tagname>
</parenttagname>
$xmldata = data string to be parsed
$startparenttagname = name of the tag that is the parent node of starttagname (necessary if

condition has been set!)
$starttagname = name of the tag (child node)
$contentnew = the content that will be inserted between the child tags $starttagname
$startcondtag = child xml tag where condition will be set
$condvalue = value of the condition
wild card character "*" can be used at begin and end of $condvalue

8.1.15 updatecontent

Syntax:
updatecontent ($xmldata, $xmlnode, $xmlnodenew)

Input parameters:
$xmldata ... XML content container
$xmlnode ... XML node to be replaced
$xmlnodenew ... new XML node

Output:
XML content container / false on error

Description:
updates a given xml string $xmlnode in $xmldata with the content $xmlnodenew.
this method provides a faster way to update xml nodes when the node was selected before.

8.1.16 insertcontent

Syntax:
insertcontent ($xmldata, $insertxmldata, $starttagname)

Input parameters:
$xmldata ... XML content container
$insertxmldata ... XML node to be inserted in starttagname
$starttagname ... tag name of the parent XML node

Output:
XML content container / false on error

Description:
.....
.....
<tagname> <- list start
.....
.....
insertxmldata <- insertxmldata
</tagname> <- list end
.....
insert $insertxmldata string at the end of all child between the parent $tagname

8.1.17 inserticontent

Syntax:
inserticontent ($xmldata, $insertxmldata, $starttagname)

Input parameters:
$xmldata ... XML content container
$insertxmldata ... XML node to be inserted in starttagname
$starttagname ... tag name of the parent XML node

Output:
XML content container / false on error

Description:
CASE-Insensitive version (XML parser are always case-sensitive!)
.....
.....
<tagname> <- list start
.....
.....
insertxmldata <- insertxmldata
</tagname> <- list end
.....
insert $insertxmldata string at the end of all child between the parent $tagname

8.1.18 addcontent

Syntax:
addcontent ($xmldata, $sub_xmldata, $startgrandtagname, $startcondtag, $condvalue, $startparenttagname, $starttagname, $contentnew)

Input parameters:
$xmldata ... XML content container
$sub_xmldata ... xml node to be inserted
$startgrandtagname ... grandparent tag name
$startcondtag ... tag holding the conditional value inside the given starttagname
$condvalue ... conditional value
$startparenttagname ... parent tag name
$starttagname ... tag name of XML node for the new content
$contentnew ... new XML node to be inserted

Output:
XML content container / false on error

Description:
<grandtagname>
<condtag>condvalue</condtag>
<parenttagname> <- list start
.....
.....
..... }
<tagname>contentnew</tagname> } <- sub_xmldata
..... }
</parenttagname> <- list end
</grandtagname>
$xmldata = data string to be parsed
$sub_xmldata = xml node to be inserted
$startgrandtagname (optional) = name of the grand xml tag of parent xml tag where (article)
$startcondtag (optional) = xml tag inside the parent xml tags where condition will be set
$condvalue (optional) = value of the condition
$startparenttagname (optional) = name of the parent xml tag where the xml subschema should be added (list)
$starttagname (optional) = name of the tag (child)
$contentnew (optional) = the content that will be inserted between the child tags

8.1.19 addicontent

Syntax:

addicontent ($xmldata, $sub_xmldata, $startgrandtagname, $startcondtag, $condvalue, $startparenttagname, $starttagname, $contentnew)

Input parameters:
$xmldata ... XML content container
$sub_xmldata ... xml node to be inserted
$startgrandtagname ... grandparent tag name
$startcondtag ... tag holding the conditional value inside the given starttagname
$condvalue ... conditional value
$startparenttagname ... parent tag name
$starttagname ... tag name of XML node for the new content
$contentnew ... new XML node to be inserted

Output:
XML content container / false on error

Description:
CASE-Insensitive version (XML parser are always case-sensitive!)
<grandtagname>
<condtag>condvalue</condtag>
<parenttagname> <- list start
.....
.....
..... }
<tagname>contentnew</tagname> } <- sub_xmldata
..... }
</parenttagname> <- list end
</grandtagname>
$xmldata = data string to be parsed
$sub_xmldata = xml subschema to be inserted
$startgrandtagname (optional) = name of the grand xml tag of parent xml tag where (article)
$startcondtag (optional) = xml tag inside the parent xml tags where condition will be set
$condvalue (optional) = value of the condition
$startparenttagname (optional) = name of the parent xml tag where the xml subschema should be added (list)
$starttagname (optional) = name of the tag (child)
$contentnew (optional) = the content that will be inserted between the child tags

8.2 Get API Functions

8.2.1 getserverload

Syntax:
getserverload ()

Input parameters:

Output:
Returns the average system load (the number of processes in the system run queue) over the last minute and the number of CPU cores

8.2.2 getsession

Syntax:
getsession ($variable, $default="")

Input parameters:
$variable ... variable name

$default ... default value (optional)

Output:
value

8.2.3 getrequest

Syntax:
getrequest ($variable, $force_type=false, $default="")

Input parameters:
$variable ... variable name
$force_type ... must be of certain type
[numeric,array,publicationname,locationname,objectname,url,bool] (optional)
$default ... default value (optional)

Output:
value

Description:
return a value from POST, GET or COOKIE, or a default value if none set

8.2.4 getrequest_esc

Syntax:
getrequest_esc ($variable, $force_type=false, $default="", $js_protection=false)

Input parameters:
$variable ... variable name
$force_type ... must be of certain type
[numeric,array,publicationname,locationname,objectname] (optional)
$default ... default value (optional)
$js_protection ... remove characters to avoid JS injection [true,false] (optional)

Output:
value

Description:
return a escaped value tp prevent XSS from POST, GET or COOKIE, or a default value if none set

8.2.5 getuserip

Syntax:
getuserip ()

Input parameters:

Output:
IP address of client / false on error

Description:
retrieves IP address of the client/user.

8.2.6 getlanguageoptions

Syntax:

getlanguageoptions ()

Input parameters:

global input parameters:
$mgmt_config

Output:
array with 2-gigit language code as key and language name in English as value / false on error

8.2.7 getlanguagefile

Syntax:
getlanguagefile ($lang="en")

Input parameters:
$lang ... language code (optional)

global input parameters:
$mgmt_config

Output:
language file name

8.2.8 getcodepage

Syntax:
getcodepage ($lang="en")

Input parameters:
$lang ... language code (optional)

global input parameters:
$mgmt_config
$hcms_lang_codepage

Output:
code page (character set)

8.2.9 getcalendarlang

Syntax:
getcalendarlang ($lang="en")

Input parameters:
$lang ... language code (optional)

global input parameters:
$mgmt_config

Output:
supported language code for calendar

8.2.10 getescapedtext

Syntax:

getescapedtext ($text, $charset="", $lang="")

Input parameters:
$text ... text as string
$charset ... character set of text
$lang ... 2-digit language code

global input parameters:
$mgmt_config
$hcms_lang_codepage
$hcms_lang

Output:
HTML escaped text

Description:
if the destination character set is not supported by language set the text need to be HTML escaped.

8.2.11 getobjectcontainer

Syntax:
getobjectcontainer ($site, $location, $object, $user)

Input parameters:
$site ... publication [string]
$location ... location [string]
$object ... object [string]
$user ... user [string]

global input parameters:
$mgmt_config

Output:
Content Container [XML]/false

Description:
loads the content container of a given object (page, component, folder)

8.2.12 getcontainer

Syntax:
getcontainer ($containerid, $type)

Input parameters:
$containerid ... container name or container ID
$type ... container type [published

global input parameters:
$mgmt_config

Output:
Contant Container [XML]/false

Description:
obsolete function used as shell for loadcontainer function without loading locked containers

8.2.13 getcontainername

Syntax:
getcontainername ($container)

Input parameters:
$container ... container name (e.g. 0000112.xml.wrk) or container ID

global input parameters:
$mgmt_config

Output:
Array with file name of the working content container (locked or unlocked!) and username if locked

8.2.14 getlocationname

Syntax:
getlocationname ($site, $location, $cat, $source="path")

Input parameters:
$site ... publication name
$location ... location path (as absolute path or converted path)
$cat ... category [page,comp]
$source ... source for name [path,name]

global input parameters:
$mgmt_config
$lang
$hcms_lang_codepage

Output:
location with readable names instead of file names / false on error

8.2.15 getthemelocation

Syntax:
getthemelocation ($theme="")

Input parameters:
$theme ... theme name (optional)

global input parameters:
$mgmt_config

Output:
path to theme / false

Description:
returns the absolute path (URL) to the theme (css and images).

8.2.16 getcategory

Syntax:
getcategory ($site="", $location)

Input parameters:

$site ... publication name (optional)
$location ... location path

global input parameters:
$mgmt_config
$publ_config

Output:
category ['page
comp'] / false on error

Description:
evaluates the category ['page, comp'] of a location.

8.2.17 getpublication

Syntax:
getpublication ($path)

Input parameters:
$path ... converted location path

Output:
publication name

Description:
extract the publication name of a location path.

8.2.18 getlocation

Syntax:
getlocation ($path)

Input parameters:
$path ... location path

Output:
location (without object or folder)

Description:
extract the location excluding object or folder of a location path.

8.2.19 getobject

Syntax:
getobject ($path)

Input parameters:
$path ... location path

Output:
object or folder name

Description:
extract the object or folder of a location path.

8.2.20 getmediacontainername

Syntax:
getmediacontainername ($file)

Input parameters:
$file ... file name

Output:
container name / false on error

Description:
extract the container name from a multimedia file name by using the hcm-ID

8.2.21 getmediafileversion

Syntax:
getmediafileversion ($container)

Input parameters:
$container ... container name or container ID

global input parameters:
$mgmt_config
$user

Output:
media file name / false on error

Description:
extracts the name from the multimedia file by container name or ID in order to get the media file of older content versions.
if the result is false, there is no older media file version.

8.2.22 getobjectid

Syntax:
getobjectid ($objectlink)

Input parameters:
$objectlink ... converted object path or pathes separated by |

Output:
object ID

Description:
converts object path to object ID

8.2.23 getobjectlink

Syntax:
getobjectlink ($objectid)

Input parameters:
$objectid ... converted object ID or IDs separated by |

Output:

converted object link

Description:
converts object ID to object path

8.2.24 getcontainerversions

Syntax:
getcontainerversions ($container)

Input parameters:
$container ... container ID or container name

global input parameters:
$mgmt_config

Output:
array of all versions (array[version-extension] = file-name) / false

8.2.25 gettemplateversions

Syntax:
gettemplateversions ($site, $template)

Input parameters:
$site ... publication name
$template ... template name

global input parameters:
$mgmt_config

Output:
array of all versions (array['YYYY-MM-DD HH:MM:SS'] = file-name) / false

8.2.26 getfileinfo

Syntax:
getfileinfo ($site, $file, $cat="comp")

Input parameters:
$site ... publication name (optional)
$file ... file name incl. extension
$cat ... category [page,comp] (optional)

global input parameters:
$mgmt_config

Output:
array/false

Description:
defines file properties based on the file extension and returns file info in an array:
$result['file']: file name without hypercms management extension
$result['name']: readable file name without hypercms management extension
$result['filename']: file name without file extensions
$result['icon']: file name of the file icon
$result['icon_large']: file name of the large file icon
$result['type']: file type

$result['ext']: file extension incl. dot in lower case
$result['published']: if file published = true else = false

8.2.27 getobjectinfo

Syntax:
getobjectinfo ($site, $location, $object, $user="sys", $container_version="")

Input parameters:
$site ... publication name
$location ... location
$object ... object name
$user ... user name (optional)
$container_version ... container version (optional)

global input parameters:
$mgmt_config

Output:
result array / false on error

Description:
get's all file pointers (container, media, template) and object name from object file and collects info from container version if provided

8.2.28 getfilesize

Syntax:
getfilesize ($file)

Input parameters:
$file ... converted path to file or directory

global input parameters:
$mgmt_config

Output:
result array with file size in kB and file count / false on error

8.2.29 getmimetype

Syntax:
getmimetype ($file)

Input parameters:
$file ... file name incl. extension

global input parameters:
$mgmt_config

Output:
mime_type

Description:
gets the mime-type of the file of its extension.
if file has a version file extension the next file extension will be used.

8.2.30 getfiletype

Syntax:
getfiletype ($file_ext)

Input parameters:
$file_ext ... file extension or file name

global input parameters:
$mgmt_config
$hcms_ext

Output:
file type to be saved in database based on file extension

8.2.31 getvideoinfo

Syntax:
getvideoinfo ($mediafile)

Input parameters:
$mediafile ... path to video file

global input parameters:
$mgmt_config
$mgmt_mediapreview

Output:
video file information as result array / false on error

8.2.32 getbrowserinfo

Syntax:
getbrowserinfo ()

Input parameters:

Output:
client browser + version as array

8.2.33 getcontentlocation

Syntax:
getcontentlocation ($container_id, $type="abs_path_content")

Input parameters:
$container_id ... container id
$type ... type [url_path_content

global input parameters:
$mgmt_config

Output:
location of the container file / false on error

Description:
gets the content location based on the given container id

a split up of folders is necessary since the number of directories is limited by the filesystem, e.g. Linux ext3 is limited to 32000.

8.2.34 getmedialocation

Syntax:
getmedialocation ($site, $file, $type)

Input parameters:
$site ... publication name
$file ... multimedia file name (including hcm-ID)
$type ... type [url_path_media

global input parameters:
$mgmt_config
$publ_config

Output:
location of the multimedia file / false on error

Description:
gets the media repsitory location from $mgtm_config Array.
the function supports up to 10 media repositories.
any other rules for splitting the media files on multiple devices could be implemented as well.

8.2.35 getlockedfileinfo

Syntax:
getlockedfileinfo ($location, $file)

Input parameters:
$location ... location to file
$file ... file name

global input parameters:
$mgmt_config

Output:
Array holding file name incl. lock extension and user name / false on error

Description:
finds the locked file and returns the name and user as array

8.2.36 getusersonline

Syntax:
getusersonline ()

Input parameters:

global input parameters:
$mgmt_config

Output:
Array of online user names / false

8.2.37 getchatstate

Syntax:
getchatstate ($register=true)

Input parameters:
$register ... register stat in session [true/false] (optional)

global input parameters:
$mgmt_config

Output:
state of chat / false on error

8.2.38 getimagelib

Syntax:
getimagelib ()

Input parameters:

global input parameters:
$mgmt_imagepreview

Output:
name of image library used [GD
ImageMagick] / false on error

8.2.39 getfilename

Syntax:
getfilename ($filedata, $tagname)

Input parameters:
$filedata ... file content
$tagname ... hyperCMS tag name in page or component

Output:
file name

Description:
extracts the file name of the hyperCMS content and template pointer tags

8.2.40 gethypertag

Syntax:
gethypertag ($filedata, $tagname, $offset=0)

Input parameters:
$filedata ... file content [string]
$tagname ... full/partly hyperCMS tag name (with or without hyperCMS:) [string]
$offset ... offset position [integer]

Output:
full hyperCMS tag array [array]/false on error

Description:

finds the hyperCMS tag start and end position
and returns an array of the whole tags including all information.
offset value must be integer value and is used to skip search
for hyperCMS tag till offset position of filedata.

8.2.41 gethypertagname

Syntax:
gethypertagname ($tagdata)

Input parameters:
$tagdata ... full hyperCMS tag

Output:
full hyperCMS tag name/false on error

Description:
reads the name of the hyperCMS tag.

8.2.42 gethtmltag

Syntax:
gethtmltag ($filedata, $tag)

Input parameters:
$filedata ... file content
$tag ... full hyperCMS tag (or other identifier)

Output:
full html tag/false on error

Description:
finds the first html tag start and end position of a nested hyperCMS tag
and returns the whole tag including all information.
works also if other script tags are nested in the HTML-tag.
this function is not case sensitive!

8.2.43 gethtmltags

Syntax:
gethtmltags ($filedata, $tag)

Input parameters:
$filedata ... file content
$tag ... full hyperCMS tag or other identifier in html tag

Output:
string from html tag start to end tag/false on error

Description:
finds the nearest html tag start and end position of a nested hyperCMS tag
and returns the whole tag including all information.
this functions works also for html-tag pairs like <a href>, and so on.

8.2.44 getattribute

Syntax:

getattribute ($string, $attribute, $secure=true)

Input parameters:
$string ... string including attributes
$attribute ... attribute name
$secure ... secure attribute value reg. XSS (optional)

Output:
attribute value/false on error

Description:
get the value of a certain attribute out of a string (...attributname=value....)

8.2.45 getoption

Syntax:
getoption ($string, $option)

Input parameters:
$string ... string including options
$option ... option name

Output:
option value/false on error

Description:
get the value of a certain option out of a string (-c:v value -ar 44100)

8.2.46 getcharset

Syntax:
getcharset ($site, $data)

Input parameters:
$site ... publication
$data ... data from template or content container [string]

global input parameters:
$mgmt_config

Output:
array with content-type and charset / false on error

Description:
extract the content-type definition and the character set from the template (1st priority), content container (2nd priority) or publication settings (3rd priority)

8.2.47 getartid

Syntax:
getartid ($id)

Input parameters:
$id ... string including id

Output:
article id/false on error

Description:
extract article id out of the id.

8.2.48 getelementid

Syntax:
getelementid ($id)

Input parameters:
$id ... string including id

Output:
element id/false on error

Description:
extract element id out of the id

8.2.49 getfirstkey

Syntax:
getfirstkey ($array)

Input parameters:
$array ... array

Output:
array key of first element in array if $value is not empty / false on error

8.3 Set API Functions

8.3.1 setsession

Syntax:
setsession ($variable, $content="", $write=false)

Input parameters:
$variable ... temporary hyperCMS variable name or array
$content ... value as string or array (optional)
$write ... write session data for load balancer [true,false] (optional)

Output:
true / false on error

8.3.2 setarticle

Syntax:
setarticle ($site, $contentdata, $contentfile, $arttitle, $artstatus, $artdatefrom, $artdateto, $artuser, $user)

Input parameters:
$site ... publication name
$contentdata ... container (XML)
$contentfile ... container name
$arttitle ... article title array
$artstatus ... article status array
$artdatefrom ... article beginn date array
$artdateto ... article end date array

$artuser ... user array or string
$user ... user name

global input parameters:
$mgmt_config

Output:
updated content container (XML)
false on error

8.3.3 settext

Syntax:
settext ($site, $contentdata, $contentfile, $text, $type, $art, $textuser, $user, $charset="", $addmicrotime=false)

Input parameters:
$site ... publication name
$contentdata ... container (XML)
$contentfile ... container name
$text ... text array
$type ... type array or string of text [u,f,l,c,d]
$art ... article array or string [yes,no]
$textuser ... text user array or string
$user ... user name
$charset ... character set of text content
$addmicrotime ... add microtime to ID [true,false] used for comments

global input parameters:
$mgmt_config
$publ_config

Output:
updated content container (XML)
false on error

8.3.4 setmedia

Syntax:
setmedia ($site, $contentdata, $contentfile, $mediafile, $mediaobject_curr, $mediaobject, $mediaalttext, $mediaalign, $mediawidth, $mediaheight, $art, $mediauser, $user, $charset="")

Input parameters:
$site ... publication name
$contentdata ... container (XML)
$contentfile ... container name
$mediafile ... media arrays (some are optional)
$mediaobject_curr ... article array or string [yes,no]
$mediaobject ... content user array or string
$mediaalttext ... user name
$mediaalign ... chracter set of text content
$mediawidth
$mediaheight
$art
$mediauser
$user
$charset

global input parameters:
$mgmt_config

Output:
updated content container (XML)
false on error

8.3.5 setpagelink

Syntax:
setpagelink ($site, $contentdata, $contentfile, $linkhref_curr, $linkhref, $linktarget, $linktext, $art, $linkuser, $user, $charset="")

Input parameters:
$site ... publication name
$contentdata ... container (XML)
$contentfile ... container name
$linkhref_curr ... current link array
$linkhref ... new link array
$linktarget ... link target array
$linktext ... link text array
$art ... article array or string [yes,no]
$linkuser ... content user array or string
$user ... user name
$charset ... chracter set of text content

global input parameters:
$mgmt_config

Output:
updated content container (XML)
false on error

8.3.6 setcomplink

Syntax:
setcomplink ($site, $contentdata, $contentfile, $component_curr, $component, $condition, $art, $compuser, $user)

Input parameters:
$site ... publication name
$contentdata ... container (XML)
$contentfile ... container name
$component_curr ... component arrays (some are optional)
$component ... article array or string [yes,no]
$condition ... content user array or string
$art ... user name
$compuser
$user

global input parameters:
$mgmt_config

Output:
updated content container (XML)
false on error

8.3.7 sethead

Syntax:
sethead ($site, $contentdata, $contentfile, $headcontent, $user, $charset="")

Input parameters:
$site ... publication name
$contentdata ... container (XML)
$contentfile ... container name
$headcontent ... content array
$user ... user name
$charset ... chracter set of text content

global input parameters:
$mgmt_config

Output:
updated content container (XML)
false on error

Description:
if content is general meta information

8.3.8 setfilename

Syntax:
setfilename ($filedata, $tagname, $value)

Input parameters:
$filedata ... file content
$tagname ... hyperCMS tag name in page or component [content
$value ... template

Output:
filedata/false on error

Description:
sets or creates the file name of the hyperCMS content file, template file, media file or file name pointer

8.4 Connect API Functions

8.4.1 ftp_userlogon

Syntax:
ftp_userlogon ($server, $user, $passwd, $ssl=false)

Input parameters:
$server ... FTP servername or IP
$user ... user name
$passwd ... password
$ssl ... SSL [true,false] (optional)

global input parameters:
$mgmt_config

Output:

true / false on error

Description:
this function connects and performs logon to an FTP server

8.4.2 ftp_userlogout

Syntax:
ftp_userlogout ($conn_id)

Input parameters:
$conn_id ... FTP connection

global input parameters:
$mgmt_config

Output:
true / false on error

Description:
this function disconnects from an FTP server

8.4.3 ftp_getfile

Syntax:
ftp_getfile ($conn_id, $remote_file, $local_file, $passive=true)

Input parameters:
$conn_id ... FTP connection
$remote_file ... path to file on FTP server
$local_file ... passive mode [true,false] (optional)
$passive

global input parameters:
$mgmt_config

Output:
true / false on error

Description:
this function gets a file from the FTP server

8.4.4 ftp_putfile

Syntax:
ftp_putfile ($conn_id, $local_file, $remote_file, $passive=true)

Input parameters:
$conn_id ... FTP connection
$local_file ... path to local file
$remote_file ... path to file on FTP server
$passive ... passive mode [true,false] (optional)

global input parameters:
$mgmt_config

Output:
true / false on error

Description:
this function puts a file to the FTP server

8.4.5 ftp_filelist

Syntax:
ftp_filelist ($conn_id, $path=".", $passive=true)

Input parameters:
$conn_id ... FTP connection
$path ... path to remote directory (optional)
$passive ... passive mode [true,false] (optional)

global input parameters:
$mgmt_config

Output:
result array / false on error

Description:
this function gets a file/directory listing of the FTP server

8.4.6 createsharelink_facebook

Syntax:
createsharelink_facebook ($site, $url)

Input parameters:
$site ... URL to share
$url

global input parameters:
$mgmt_config

Output:
Share URL / false on error

8.4.7 createsharelink_twitter

Syntax:
createsharelink_twitter ($site, $url, $text)

Input parameters:
$site ... URL to share
$url ... message to share
$text

global input parameters:
$mgmt_config

Output:
Share URL / false on error

8.4.8 createsharelink_googleplus

Syntax:

createsharelink_googleplus ($site, $url)

Input parameters:
$site ... URL to share
$url

global input parameters:
$mgmt_config

Output:
Share URL / false on error

8.4.9 createsharelink_linkedin

Syntax:
createsharelink_linkedin ($site, $url, $title, $summary, $source)

Input parameters:
$site ... URL to share
$url ... title
$title ... summary (optional)
$summary ... source (optional)
$source

global input parameters:
$mgmt_config

Output:
Share URL / false on error

8.4.10 createsharelink_pinterest

Syntax:
createsharelink_pinterest ($site, $image_url, $title, $description)

Input parameters:
$site ... image URL to share
$image_url ... title
$title ... description (optional)
$description

global input parameters:
$mgmt_config

Output:
Share URL / false on error

8.5 Security API Functions

8.5.1 rootpermission

Syntax:
rootpermission ($site_name, $site_admin, $permission_str)

Input parameters:
$site_name ... publication name
$site_admin ... publication admin

$permission_str ... permission string from group

global input parameters:
$rootpermission
$mgmt_config

Output:
global permission array/false

Description:
deseralizes the permission string and and returns the root permission array.

8.5.2 globalpermission

Syntax:
globalpermission ($site_name, $permission_str)

Input parameters:
$site_name ... publication
$permission_str ... permission string from group

Output:
global permission array/false

Description:
deseralizes the permission string and returns the global permission array.

8.5.3 localpermission

Syntax:
localpermission ($site_name, $permission_str)

Input parameters:
$site_name ... publication
$permission_str ... permission string from group

Output:
local permission array/false

Description:
deseralizes the permission string and returns the local permission array.

8.5.4 accessgeneral

Syntax:
accessgeneral ($site, $location, $cat)

Input parameters:
$site ... publication
$location ... location (path to folder)
$cat ... object category ['page

global input parameters:
$mgmt_config
$hiddenfolder
$siteaccess

Output:

true/false

Description:
checks general access to certain system folders, publications and returns true if access is granted

8.5.5 accesspermission

Syntax:
accesspermission ($site, $location, $cat)

Input parameters:
$site ... location (path to folder)
$location ... object category ['page
$cat ... comp']

global input parameters:
$pageaccess
$compaccess
$hiddenfolder
$hcms_linking
$mgmt_config

Output:
group with access permissions as array / false on error

Description:
evaluates page and component access permissions and returns group(s).

8.5.6 setlocalpermission

Syntax:
setlocalpermission ($site, $group_array, $cat)

Input parameters:
$site ... publication
$group_array ... group name array
$cat ... object category [page,comp]

global input parameters:
$localpermission

Output:
local permission array / false on error

Description:
sets local permissions of a user group for a specific publication

8.5.7 checkpublicationpermission

Syntax:
checkpublicationpermission ($site, $strict=true)

Input parameters:
$site ... publication name
$strict ... strictly limited to siteaccess only without inheritance [true/false] (optional)

global input parameters:

$mgmt_config
$siteaccess

Output:
"direct" for direct access via group permission / "inherited" for access through inheritance / false

Description:
checks access to a publication based on the site access and inheritance settings

8.5.8 checkadminpermission

Syntax:
checkadminpermission ()

Input parameters:

global input parameters:
$adminpermission

Output:
true/false

Description:
checks super admin permission

8.5.9 checkrootpermission

Syntax:
checkrootpermission ($name)

Input parameters:
$name ... permission name

global input parameters:
$rootpermission

Output:
true/false

Description:
checks root permission

8.5.10 checkglobalpermission

Syntax:
checkglobalpermission ($site, $name)

Input parameters:
$site ... publication name
$name ... permission name

global input parameters:
$globalpermission

Output:
true/false

Description:
checks global permission for a publication

8.5.11 checklocalpermission

Syntax:
checklocalpermission ($site, $group, $name)

Input parameters:
$site ... publication name
$group ... user group name
$name ... permission name

global input parameters:
$$localpermission

Output:
true/false

Description:
checks local permissions of a user group for a specific publication

8.5.12 userlogin

Syntax:
userlogin ($user, $passwd, $hash="", $objref="", $objcode="", $ignore_password=false, $locking=true)

Input parameters:
$user ... username
$passwd ... password
$hash ... hash code of user
$objref ... object reference for hcms linking (object ID)
$objcode ... object code for hcms linking (crypted object ID)
$ignore_password ... ignore passwordcheck needed for WebDAV or access link [true/false]
$locking ... lock IP after 10 failed attempts to login [true/false]

global input parameters:
$mgmt_config
$eventsystem
$hcms_lang_codepage
$hcms_lang
$lang

Output:
result array

Description:
login of user by sending user and password using the variables: $sentuser, $sentpasswd
this procedure will register the user in the hypercms session and in the php session.
the procedure will return true or false using the variable $result.

8.5.13 registerinstance

Syntax:
registerinstance ($instance, $load_config=true)

Input parameters:

$instance ... instance name
$load_config ... load main config of instance [true/false] (optional)

global input parameters:
$mgmt_config

Output:
true/false

8.5.14 createchecksum

Syntax:
createchecksum ($permissions="")

Input parameters:
$permissions ... array or empty

Output:
MD5 checksum

8.5.15 writesession

Syntax:
writesession ($user, $passwd, $checksum)

Input parameters:
$user ... user name
$passwd ... password
$checksum ... checksum

global input parameters:
$mgmt_config

Output:
true / false on error

Description:
writes hyperCMS specific session data of a user

8.5.16 writesessiondata

Syntax:
writesessiondata ()

Input parameters:

global input parameters:
$mgmt_config

Output:
true / false on error

Description:
serializes and writes all session data of a user

8.5.17 createsession

Syntax:
createsession ()

Input parameters:

global input parameters:
$mgmt_config

Output:
true

Description:
Checks if session data of a user is available. This function does access session variables directly!

8.5.18 killsession

Syntax:
killsession ($user="", $destroy_php=true)

Input parameters:
$user ... user name for hyperCMS session (optional)
$destroy_php ... destroy php session [true,false] (optional)

global input parameters:
$mgmt_config

Output:
true

Description:
destroys session data of user

8.5.19 checkdiskkey

Syntax:
checkdiskkey ($users="", $site="")

Input parameters:
$users ... user count (optional)
$site ... publication names (use | as seperator) (optional)

global input parameters:
$mgmt_config

Output:
true/false

Description:
checks the disc key of the installation

8.5.20 checkpassword

Syntax:
checkpassword ($password)

Input parameters:
$password ... password a string

global input parameters:
$mgmt_config
$lang

Output:
true if passed / error message as string

Description:
this function checks the strength of a password and return the error messages or true.

8.5.21 loguserip

Syntax:
loguserip ($client_ip, $user="sys")

Input parameters:
$client_ip ... client IP address
$user ... user logon name (optional)

global input parameters:
$mgmt_config

Output:
true / false on error

8.5.22 checkuserip

Syntax:
checkuserip ($client_ip, $user="", $timeout="")

Input parameters:
$client_ip ... client IP address
$user ... user logon name (optional)
$timeout ... timeout in minutes (optional)

global input parameters:
$mgmt_config

Output:
true if IP is not locked / false if IP is locked or on error

8.5.23 checkuserrequests

Syntax:
checkuserrequests ($user="sys")

Input parameters:
$user ... user name (optional)

global input parameters:
$mgmt_config

Output:
true / false if a certain amount of reguests per minute is exceeded

Description:
provides security for Cross-Site Request Forgery

8.5.24 checkusersession

Syntax:
checkusersession ($user="sys", $CSRF_detection=true)

Input parameters:
$user ... user name (optional)
$CSRF_detection ... include CSRF detection [true,false]

global input parameters:
$mgmt_config

Output:
true / html-output followed by termination

Description:
checks if session data of user is correct. This function does access session variables directly!
requires config.inc.php

8.5.25 allowuserip

Syntax:
allowuserip ($site)

Input parameters:
$site ... publication name

global input parameters:
$mgmt_config

Output:
true / false

Description:
checks if the client IP is in the range of valid IPs.
requires config.inc.php

8.5.26 valid_objectname

Syntax:
valid_objectname ($variable)

Input parameters:
$variable ... variable (string or array)

Output:
variable / false on error

Description:
test if an expression includes forbidden characters (true) or doesnt (false) to prevent directory browsing

8.5.27 valid_locationname

Syntax:
valid_locationname ($variable)

Input parameters:
$variable ... variable (string or array)

Output:
variable / false on error

Description:
test if an expression includes forbidden characters (true) or doesnt (false) to prevent directory browsing

8.5.28 valid_publicationname

Syntax:
valid_publicationname ($variable)

Input parameters:
$variable ... variable (string or array)

Output:
variable / false on error

Description:
test if an expression includes forbidden characters (true) or doesnt (false) to prevent directory browsing

8.5.29 html_encode

Syntax:
html_encode ($expression, $encoding="", $js_protection=false)

Input parameters:
$expression ... variable as string or array
$encoding ... conversion of all special characters based on given character set or to ASCII (optional)
$js_protection ... remove characters to avoid JS injection [true,false] (optional)

Output:
html encoded value as array or string / false on error

Description:
converts a string into the html equivalents (also used for XSS protection).
supports multibyte character sets like UTF-8 as well based on the ASCII value of the character.

8.5.30 html_decode

Syntax:
html_decode ($expression, $encoding="")

Input parameters:
$expression ... variable as string or array
$encoding ... conversion of all special characters based on given character set (optional)

Output:
html decoded value as array or string / false on error

Description:
this function decodes all characters which have been converted by html_encode.

8.5.31 scriptcode_encode

Syntax:
scriptcode_encode ($content)

Input parameters:
$content ... content as string

global input parameters:
$mgmt_config

Output:
escaped content as string / false on error

Description:
this function escapes all script tags.
this function must be used to clean all user input in the CMS by removing all server side scripts tags.

8.5.32 scriptcode_extract

Syntax:
scriptcode_extract ($content, $identifier_start="<?", $identifier_end="?>")

Input parameters:
$content ... content as string
$identifier_start ... identifier of script begin
$identifier_end ... and end

Output:
script code as array / false on error or if noting was found

Description:
this function extracts the script code of a given content.

8.5.33 scriptcode_clean_functions

Syntax:
scriptcode_clean_functions ($content, $type=3, $application="PHP")

Input parameters:
$content ... content as string
$type ... cleaning level type from none = 0 to strong = 3 (no cleaning = 0
$application ... basic set of disabled functions = 1

global input parameters:
$mgmt_config

Output:
result array / false on error

Description:
this function removes all dangerous PHP functions.

8.5.34 url_encode

Syntax:
url_encode ($variable)

Input parameters:
$variable ... variable as string or array

global input parameters:
$mgmt_config

Output:
urlencoded value as array or string / false on error

Description:
this function encodes all characters.

8.5.35 url_decode

Syntax:
url_decode ($variable)

Input parameters:
$variable ... variable as string or array

global input parameters:
$mgmt_config

Output:
urldecoded value as array or string / false on error

Description:
this function decodes all characters which have been converted by url_encode or urlencode (PHP).

8.5.36 shellcmd_encode

Syntax:
shellcmd_encode ($variable)

Input parameters:
$variable ... variable as string or array

Output:
encoded value as array or string / false on error

Description:
this function encodes/escapes characters to secure the shell comand.

8.5.37 hcms_crypt

Syntax:
hcms_crypt ($string, $start=0, $length=0)

Input parameters:
$string ... string to encode
$start ... start position
$length ... length for string extraction

global input parameters:
$mgmt_config

Output:
encoded string / false on error

Description:
unidirectional encryption using crypt, MD5 and urlencode

8.5.38 hcms_encrypt

Syntax:
hcms_encrypt ($string, $key="", $crypt_level="", $encoding="url")

Input parameters:
$string ... string to encode
$key ... key of length 16 or 24 or 32 (optional)
$crypt_level ... crypt strength level [weak,standard,strong] (optional)
$encoding ... encoding [base64,url,none] (optional)

global input parameters:
$mgmt_config

Output:
encoded string / false on error

Description:
encryption of a string. only strong encryption is binary-safe!

8.5.39 hcms_decrypt

Syntax:
hcms_decrypt ($string, $key="", $crypt_level="", $encoding="url")

Input parameters:
$string ... hash-string to decode
$key ... key of length 16 or 24 or 32 (optional)
$crypt_level ... crypt strength level [weak,standard,strong] (optional)
$encoding ... encoding [base64,url,none] (optional)

global input parameters:
$mgmt_config

Output:
decoded string / false on error

Description:
decryption of a string. only strong encryption is binary-safe!

8.5.40 createtimetoken

Syntax:
createtimetoken ($lifetime=0, $secret=4)

Input parameters:
$lifetime ... token lifetime in seconds (optional)
$secret ... secret value (optional)

global input parameters:
$mgmt_config

Output:
token / false on error

8.5.41 checktimetoken

Syntax:
checktimetoken ($token, $secret=4)

Input parameters:
$token ... token
$secret ... secret value (optional)

global input parameters:
$mgmt_config

Output:
true / false

8.5.42 createtoken

Syntax:
createtoken ($user="sys", $lifetime=0, $secret=4)

Input parameters:
$user ... user name (optional)
$lifetime ... token lifetime in seconds (optional)
$secret ... secret value (optional)

global input parameters:
$mgmt_config

Output:
token / false on error

8.5.43 checktoken

Syntax:
checktoken ($token, $user="sys", $secret=4)

Input parameters:
$token ... token
$user ... user name (optional)
$secret ... secret value (optional)

global input parameters:
$mgmt_config

Output:
true / false

8.5.44 createuniquetoken

Syntax:
createuniquetoken ($length=16)

Input parameters:
$length ... token length (optional)

global input parameters:
$mgmt_config

Output:
token as string / false

8.5.45 rand_secure

Syntax:
rand_secure ($min=1000, $max=999999999999)

Input parameters:
$min ... min and max value as integer (optional)
$max

Output:
secure random number / false

8.6 Media API Functions

8.6.1 createthumbnail_indesign

Syntax:
createthumbnail_indesign ($site, $location_source, $location_dest, $file)

Input parameters:
$site ... publication
$location_source ... path to source dir
$location_dest ... path to destination dir
$file ... file name

global input parameters:
$mgmt_config

Output:
new file name / false on error (saves only thumbnail media file in destination location
only jpeg format is supported as output)

Description:
creates a thumbnail by extracting the thumbnail from an indesign file and transferes the generated image via remoteclient.
note for good results, InDesign Preferences must be set to save preview image and at extra large size.

8.6.2 createthumbnail_video

Syntax:
createthumbnail_video ($site, $location_source, $location_dest, $file, $frame)

Input parameters:
$site ... publication
$location_source ... path to source dir
$location_dest ... path to destination dir
$file ... file name
$frame ... frame of video in the seconds or hh:mm:ss[.xxx]

global input parameters:
$mgmt_config
$mgmt_mediapreview

Output:
new file name / false on error (saves only thumbnail media file in destination location only jpeg format is supported as output)

Description:
creates a thumbnail picture of a video frame

8.6.3 createmedia

Syntax:
createmedia ($site, $location_source, $location_dest, $file, $format="", $type="thumbnail", $force_no_encrypt=false)

Input parameters:
$site ... publication
$location_source ... path to source dir
$location_dest ... path to destination dir
$file ... file name
$format ... format (file extension w/o dot) (optional)
$type ... type of image/video/audio file [thumbnail(for thumbnails of images),origthumb(thumbnail made from original video/audio),original(to overwrite original video/audio file),any other string present in $mgmt_imageoptions/$mgmt_mediaoptions] (optional)
$force_no_encrypt ... force the file to be not encrypted even if the content of the publication must be encrypted [true,false] (optional)

global input parameters:
$mgmt_config
$mgmt_imagepreview
$mgmt_mediapreview
$mgmt_mediaoptions
$mgmt_imageoptions
$mgmt_maxsizepreview
$mgmt_mediametadata
$hcms_ext

Output:
new file name / false on error (saves original or thumbnail media file in destination location for thumbnail only jpeg format is supported as output)

Description:
creates an new image from original or creates a thumbnail and transferes the generated image via remoteclient

8.6.4 convertmedia

Syntax:

convertmedia ($site, $location_source, $location_dest, $mediafile, $format, $media_config="", $force_no_encrypt=false)

Input parameters:
$site ... publication name
$location_source ... path to source dir
$location_dest ... path to destination dir
$mediafile ... file name
$format ... target format (file extension w/o dot) of destination file
$media_config ... media configuration to be used (optional)
$force_no_encrypt ... force the file to be not encrypted even if the content of the publication must be encrypted [true,false] (optional)

global input parameters:
$mgmt_config
$mgmt_imagepreview
$mgmt_mediapreview
$mgmt_mediaoptions
$mgmt_imageoptions
$mgmt_maxsizepreview
$mgmt_mediametadata
$hcms_ext

Output:
new file name / false on error

Description:
converts and creates a new image/video/audio or document from original. this is a wrapper function for createmedia and createdocument.

8.6.5 convertimage

Syntax:
convertimage ($site, $file_source, $location_dest, $format="jpg", $colorspace="RGB", $iccprofile="", $width="", $height="", $slug=0, $units="px", $dpi=72, $quality="")

Input parameters:
$site ... publication name
$file_source ... path to source image file
$location_dest ... path to destination dir
$format ... format (file extension w/o dot) of destination file (optional)
$colorspace ... colorspace of new image [CMY,CMYK,Gray,HCL,HCLp,HSB,HSI,HSL,HSV,HWB,Lab,LCHab,LCHuv,LMS,Log,Luv,OHTA,Rec601YCbCr,Rec709YCbCr,RGB,scRGB,sRGB,Transparent,XYZ,YCbCr,YCC,YDbDr,YIQ,YPbPr,YUV] (optional)
$iccprofile ... width in pixel/mm/inch (optional)
$width ... height in pixel/mm/inch (optional)
$height ... slug in pixel/mm/inch (optional)
$slug ... units for width
$units ... height and slug [px,mm,inch] (optional)
$dpi ... dpi (optional)
$quality ... image quality (1 to 100)

global input parameters:
$mgmt_config
$mgmt_imagepreview
$mgmt_mediapreview
$mgmt_mediaoptions

$mgmt_imageoptions
$mgmt_maxsizepreview
$mgmt_mediametadata
$hcms_ext

Output:
new file name / false on error

Description:
converts and creates a new image from original. the new image keeps will be resized and cropped to fit width and height.
this is a wrapper function for createmedia.

8.6.6 rotateimage

Syntax:
rotateimage ($site, $filepath, $angle, $imageformat)

Input parameters:
$site ... publication
$filepath ... path to source media file
$angle ... rotation angle
$imageformat ... destination image format [jpg,png,gif]

global input parameters:
$mgmt_config

Output:
new image file name / false on error

Description:
rotates an image (must be jpg, png or gif) using GD library. not used if ImageMagick is available.

8.6.7 getimagecolors

Syntax:
getimagecolors ($site, $file)

Input parameters:
$site ... publication
$file ... media file name

global input parameters:
$mgmt_config

Output:
result array / false on error

Description:
uses the thumbnail image to calculate the mean color (red, green, blue), defines the colorkey (5 most commonly used colors) and the image type (landscape, portrait, square)

8.6.8 getimagecolorkey

Syntax:
getimagecolorkey ($image)

Input parameters:
$image ... image resource

global input parameters:
$mgmt_config

Output:
color key of image / false on error

Description:
extracts the color key for an image that represents the 5 mostly used colors.
K...black
W...white
E...grey
R...red
G...green
B...blue
C...cyan
M...magenta
Y...yellow
O...orange
P...pink
N...brown

8.6.9 hex2rgb

Syntax:
hex2rgb ($hex)

Input parameters:
$hex ... image color as hex-code

Output:
RGB-color as array / false on error

8.6.10 rgb2hex

Syntax:
rgb2hex ($red, $green, $blue)

Input parameters:
$red ... image color in RGB
$green
$blue

Output:
hex-color as string / false on error

8.6.11 createdocument

Syntax:
createdocument ($site, $location_source, $location_dest, $file, $format="", $force_no_encrypt=false)

Input parameters:
$site ... publication
$location_source ... path to source location
$location_dest ... path to destination location

$file ... file name
$format ... destination file format (extension w/o dot)
$force_no_encrypt ... force the file to be not encrypted even if the content of the publication must be encrypted [true,false] (optional)

global input parameters:
$mgmt_config
$mgmt_docpreview
$mgmt_docoptions
$mgmt_docconvert
$mgmt_maxsizepreview
$hcms_ext
$hcms_lang
$lang

Output:
new file name / false on error

Description:
creates a new multimedia file of given format at source destination using UNOCONV and saves it as thumbnail file at the desitnation location

8.6.12 unzipfile

Syntax:
unzipfile ($site, $zipfilepath, $location, $filename, $user)

Input parameters:
$site ... publication
$zipfilepath ... path to source zip file
$location ... path to destination location
$filename ... name of file for extraction
$user ... user

global input parameters:
$mgmt_config
$mgmt_uncompress
$mgmt_imagepreview
$mgmt_mediapreview
$mgmt_mediaoptions

Output:
true/false

Description:
unpackes ZIP file and creates media files in destination location

8.6.13 zipfiles

Syntax:
zipfiles ($site, $multiobject_array, $destination="", $zipfilename, $user, $activity="")

Input parameters:
$site ... publication
$multiobject_array ... array with path to source zip files
$destination ... destination location (if this is null then the $location where the zip-file resists will be used)
$zipfilename ... name of ZIP-file

$user ... user name
$activity ... activity that need to be set for daily stats [download] (optional)

global input parameters:
$mgmt_config
$mgmt_compress
$pageaccess
$compaccess
$hiddenfolder
$hcms_linking
$globalpermission
$setlocalpermission

Output:
true/false

8.6.14 px2mm

Syntax:
px2mm ($pixel, $dpi=72)

Input parameters:
$pixel ... pixel
$dpi ... dpi (optional)

Output:
pixel / false

Description:
convert mm to pixel

8.6.15 px2inch

Syntax:
px2inch ($pixel, $dpi=72)

Input parameters:
$pixel ... pixel
$dpi ... dpi (optional)

Output:
inch / false

Description:
convert pixel to inches

8.6.16 inch2px

Syntax:
inch2px ($inch, $dpi=72)

Input parameters:
$inch ... pixel
$dpi ... dpi (optional)

Output:
pixel / false

Description:
convert inches to pixel

8.6.17 vtt2array

Syntax:
vtt2array ($vtt)

Input parameters:
$vtt ... VTT string

Output:
array / false

Description:
converts VTT string to array

8.7 Metadata API Functions

8.7.1 getkeywords

Syntax:
getkeywords ($text, $language="en", $charset="UTF-8")

Input parameters:
$text ... text as string
$language ... supported language [de,en]
$charset

global input parameters:
$mgmt_config

Output:
keywords sperated by
/false on error

Description:
generates a keyword list for meta information. supports german and english stop words lists.

8.7.2 getdescription

Syntax:
getdescription ($text, $charset="UTF-8")

Input parameters:
$text ... text as string
$charset

Output:
cleanded description of provided text /false on error

Description:
generates a keyword list for meta information. supports german and english stop words lists.

8.7.3 getgooglesitemap

Syntax:

getgooglesitemap ($site, $dir, $url, $getpara=array(), $permalink=array(), $chfreq="weekly", $prio="", $ignore=array(), $filetypes=array('cfm','htm','html','xhtml','asp','aspx','jsp','php','pdf'), $show_freq=true, $show_prio=true)

Input parameters:
$site ... publication anme
$dir ... directory path
$url ... URL to directory
$getpara ... GET parameters to use for new versions of the URL as array (optional)
$permalink ... permanent links text-ID to use for location as array (optional)
$chfreq ... frequency of google scrawler [never,weekly,daily] (optional)
$prio ... priority [1 or less] (optional)
$ignore ... ignore file names as array (optional)
$filetypes ... allowed file types as array (optional)
$show_freq ... include frequenzy tag [true,false] (optional)
$show_prio ... include priority tag [true,false] (optional)

global input parameters:
$mgmt_config
$publ_config

Output:
xml sitemap / false on error

Description:
generates a google sitemap xml-output.

8.7.4 getmetadata

Syntax:
getmetadata ($location, $object, $container="", $seperator="\n", $template="")

Input parameters:
$location ... location
$object ... object (both optional if container is given)
$container ... container name or container content (optional)
$seperator ... seperator of meta data fields [any string,array] (optional)
$template ... publication name/template name to extract label names (optional)

global input parameters:
$mgmt_config

Output:
string with all meta data from given object based on container

8.7.5 copymetadata

Syntax:
copymetadata ($file_source, $file_dest)

Input parameters:
$file_source ... path to source file
$file_dest ... path to destination file

global input parameters:
$user
$mgmt_config

$mgmt_mediametadata

Output:
true / false

Description:
copies all meta data from source to destination file using EXIFTOOL

8.7.6 extractmetadata

Syntax:
extractmetadata ($file)

Input parameters:
$file ... path to image file

global input parameters:
$user
$mgmt_config
$mgmt_mediametadata

Output:
result array / false on error

Description:
extracts all meta data from a file using EXIFTOOL

8.7.7 xmlobject2array

Syntax:
xmlobject2array ($obj, $namespace="")

Input parameters:
$obj ... XML as object
$namespace ... namespace as array (optional)

Output:
result array / false

Description:
function to convert an xmlobject to an array, provided by xaviered at gmail dot com

8.7.8 id3_getdata

Syntax:
id3_getdata ($file)

Input parameters:
$file ... path to audio file

global input parameters:
$mgmt_config
$hcms_ext

Output:
result array / false on error

Description:
requires getID3 library since EXIFTOOL cannot write ID3 tags so far

8.7.9 id3_writefile

Syntax:
id3_writefile ($file, $id3, $keep_data=true, $movetempfile=true)

Input parameters:
$file ... abs. path to audio file
$id3 ... ID3 tag array
$keep_data ... keep existing ID3 data of file [true,false] (optional)
$movetempfile ... move tempoarary file from unecrypted to encrypted [true,false] (optional)

global input parameters:
$user
$mgmt_config
$mgmt_mediametadata
$hcms_ext

Output:
true / false on error

Description:
writes ID3 tags into audio file for supported file types and keeps the existing ID3 tags

8.7.10 id3_create

Syntax:
id3_create ($site, $text)

Input parameters:
$site ... publication name
$text ... text array (from content container)

global input parameters:
$mgmt_config

Output:
ID3 tag array / false on error

Description:
defines ID3 tag array based on the media mapping of a publication.

8.7.11 xmp_getdata

Syntax:
xmp_getdata ($file)

Input parameters:
$file ... path to image file

global input parameters:
$user
$mgmt_config
$hcms_ext

Output:
result array / false on error

8.7.12 xmp_writefile

Syntax:
xmp_writefile ($file, $xmp, $keep_data=true, $movetempfile=true)

Input parameters:
$file ... abs. path to image file
$xmp ... XMP tag array
$keep_data ... keep existing XMP data of file [true,false] (optional)
$movetempfile ... move tempoarary file from unecrypted to encrypted [true,false] (optional)

global input parameters:
$user
$mgmt_config
$mgmt_mediametadata
$hcms_ext

Output:
true / false on error

Description:
writes XMP tags into image file for supported file types and keeps the existing XMP tags

8.7.13 xmp_create

Syntax:
xmp_create ($site, $text)

Input parameters:
$site ... publication name
$text ... text array (from content container)

global input parameters:
$mgmt_config

Output:
XMP tag array / false on error

Description:
defines XMP tag array based on the media mapping of a publication.

8.7.14 geo2decimal

Syntax:
geo2decimal ($deg, $min, $sec, $hemi)

Input parameters:
$deg ... geo location in degree
$min ... minutes
$sec ... seconds
$hemi ... hemisphere [N,O,S,W]

Output:
decimal result / false

8.7.15 exif_getdata

Syntax:
exif_getdata ($file)

Input parameters:
$file ... path to image file

global input parameters:
$user
$mgmt_config
$hcms_ext

Output:
result array / false

8.7.16 iptc_getdata

Syntax:
iptc_getdata ($file)

Input parameters:
$file ... path to image file

global input parameters:
$user
$mgmt_config
$hcms_ext

Output:
result array / false

8.7.17 iptc_getcharset

Syntax:
iptc_getcharset ($tag)

Input parameters:
$tag ... iptc tag that holds character set information

Output:
charset as string / false on error

Description:
Copied from MediaWiki!
Warning, this function does not (and is not intended to) detect all iso 2022 escape codes.
In practise, the code for utf-8 is the only code that seems to have wide use. It does detect that code.
According to iim standard, charset is defined by the tag 1:90.
in which there are iso 2022 escape sequences to specify the character set.
the iim standard seems to encourage that all necessary escape sequences are
in the 1:90 tag, but says it doesn't have to be.
This is in need of more testing probably. This is definitely not complete.
however reading the docs of some other iptc software, it appears that most iptc software only recognizes utf-8. If 1:90 tag is not present content is
usually ascii or iso-8859-1 (and sometimes utf-8), but no guarantee.
This also won't work if there are more than one escape sequence in the 1:90 tag

or if something is put in the G2, or G3 charsets, etc. It will only reliably recognize utf-8. This is just going through the charsets mentioned in appendix C of the iim standard.

8.7.18 iptc_maketag

Syntax:
iptc_maketag ($record=2, $tag, $value)

Input parameters:
$record ... type of tag (e.g. 2)
$tag ... code of tag (e.g. 025)
$value ... value of tag

Output:
binary IPTC tag / false on error

Description:
convert the IPTC tag into binary code.

8.7.19 iptc_writefile

Syntax:
iptc_writefile ($file, $iptc, $keep_data=true, $movetempfile=true)

Input parameters:
$file ... abs. path to image file
$iptc ... IPTC tag array
$keep_data ... keep existing IPTC data of file [true,false] (optional)
$movetempfile ... move tempoarary file from unecrypted to encrypted [true,false] (optional)

global input parameters:
$user
$mgmt_config
$mgmt_mediametadata

Output:
true / false on error

Description:
writes IPTC tags into image file for supported file types and keeps the existing IPTC tags

8.7.20 iptc_create

Syntax:
iptc_create ($site, $text)

Input parameters:
$site ... publication name
$text ... text array (from content container)

global input parameters:
$mgmt_config

Output:
IPTC tag array / false on error

Description:
defines IPTC tag array based on the medai mapping of a publication.

8.7.21 createmapping

Syntax:
createmapping ($site, $mapping)

Input parameters:
$site ... publication name
$mapping ... mapping definition

global input parameters:
$mgmt_config

Output:
true / false on error

Description:
prepares the PHP mapping array from the provided mapping definition and saves media mapping file

8.7.22 getmapping

Syntax:
getmapping ($site)

Input parameters:
$site ... publication name

global input parameters:
$mgmt_config

Output:
mapping code for display / false

Description:
loads the mapping file of the provided publication.

8.7.23 setmetadata

Syntax:
setmetadata ($site, $location="", $object="", $mediafile="", $mapping="", $user)

Input parameters:
$site ... publication name
$location ... location path (optional)
$object ... object name (optional)
$mediafile ... media file name (optional)
$mapping ... mapping array (meta data tag name -> text-id
$user ... optional)

global input parameters:
$eventsystem
$mgmt_config
$hcms_ext

Output:
true/false

Description:

saves meta data of a multimedia file using a provided mapping in the proper fields of the content container.
if no mapping is given a default mapping will be used.

8.8 Link API Functions

8.8.1 link_db_restore

Syntax:
link_db_restore ($site="")

Input parameters:
$site ... publication name (optinal)

global input parameters:
$mgmt_config

Output:
true / false on error

Description:
this function restores a given or all link management index files

8.8.2 link_db_load

Syntax:
link_db_load ($site, $user)

Input parameters:
$site ... site
$user ... user

global input parameters:
$mgmt_config

Output:
link database [2 dim. array] or true / false on error

Description:
this function loads and locks the link management database
each record of the link management database has the following design:
xml-content container :| absolute path to 1-n objects :| 1-m links used by 1-n objects
important: the link management database has to saved or closed after loading it.

8.8.3 link_db_read

Syntax:
link_db_read ($site)

Input parameters:
$site ... site

global input parameters:
$mgmt_config

Output:
link database [2 dim. array] or true / false on error

Description:
this function loads the link management database for reading without locking

8.8.4 link_db_close

Syntax:
link_db_close ($site, $user)

Input parameters:
$site ... site
$user ... user

global input parameters:
$mgmt_config

Output:
true/false

Description:
closes and unlocks the link management database.

8.8.5 link_db_save

Syntax:
link_db_save ($site, $link_db, $user)

Input parameters:
$site ... link database array
$link_db ... site
$user ... user

global input parameters:
$mgmt_config

Output:
true/false on error

Description:
this function saves und unlocks the link management database

8.8.6 link_db_update

Syntax:
link_db_update ($site, $link_db, $attribute, $contentfile, $cat, $link_curr, $link_new, $option)

Input parameters:
$site ... publication name
$link_db ... link database [2 dim. array]
$attribute ... attribute ['object'
$contentfile ... 'link']
$cat ... content container [optional] [string]
$link_curr ... link category [optional] ['comp'
$link_new ... 'page']
$option ... current link [optional

global input parameters:

$mgmt_config

Output:
link database [2 dim. array] or true / false on error

Description:
this function inserts, updates and removes objects and their links from the link management database (add or update a link)
depending on which link is left empty:
link_curr = "": add new link (just one link matching given category!)
link_new = "": delete current link in use (just one linkm matching given category!)
link_curr & link_new are not empty: update current link with the new one

8.8.7 link_db_insert

Syntax:
link_db_insert ($site, $link_db, $contentfile, $cat, $object)

Input parameters:
$site ... publication name
$link_db ... link database [2 dim. array]
$contentfile ... content container
$cat ... link category ['comp
$object ... page']

global input parameters:
$mgmt_config

Output:
link database [2 dim. array] or true / false

Description:
this function inserts a new record in the link management database
optionally the created object can be also inserted

8.8.8 link_db_delete

Syntax:
link_db_delete ($site, $link_db, $contentfile)

Input parameters:
$site ... link database [2 dim. array]
$link_db ... content container
$contentfile

global input parameters:
$mgmt_config

Output:
link database [2 dim. array] or true / false on error

Description:
this function deletes a record in the link management database

8.8.9 link_db_getobject

Syntax:
link_db_getobject ($multiobject)

Input parameters:
$multiobject ... link database attribut (references to objects seperated by |)

global input parameters:
$mgmt_config

Output:
objects [array] / false on error

Description:
this function splits the object string into an array of objects.

8.8.10 link_update

Syntax:
link_update ($site, $container, $link_old, $link_new)

Input parameters:
$site ... publication name
$container ... container name
$link_old ... old link (converted)
$link_new ... new link (converted)

global input parameters:
$user
$mgmt_config

Output:
true/false

Description:
this function updates the link of the published and working content container and link file

8.8.11 getlinkedobject

Syntax:
getlinkedobject ($site, $location, $page, $cat)

Input parameters:
$site ... publication
$location ... location
$page ... object (name and extension)
$cat ... category [page

global input parameters:
$mgmt_config

Output:
objects which link to the given object [array] or true / false

Description:
this function gets all objects which link to the given object.
works with pages (page links) and components (component links) if link management is enabled.

8.8.12 getconnectedobject

Syntax:
getconnectedobject ($container, $type="work")

Input parameters:
$container ... container name
$type ... container type [work,published,version] (optional)

global input parameters:
$mgmt_config
$user

Output:
connected objects[array]

Description:
this function gets all objects which use the same content container and are therefore connected.

8.8.13 extractlinks

Syntax:
extractlinks ($textcontent, $identifier)

Input parameters:
$textcontent ... text content as string
$identifier ... link identifiert ("href" for hyperreferences

global input parameters:
$mgmt_config

Output:
object links [array] / false on error

Description:
this function extracts all links based on it's identifier from a text and returns an array of all links

8.9 Plugin API Functions

8.9.1 plugin_getdefaultconf

Syntax:
plugin_getdefaultconf ()

Input parameters:

Output:
default value as array

8.9.2 plugin_readmenu

Syntax:
plugin_readmenu ($xml, $pluginFolder)

Input parameters:

$xml ... plugin xml as string
$pluginFolder ... plugin directory

global input parameters:
$mgmt_config

Output:
menu point array used by navigator

Description:
Reads Menupoints and menugroups from the xml data
be carefull with nesting, getcontent is used here and you can't nest groups inside of groups as a subpoint!
pluginFolder contains the folder this plugin is located in, that is needed for the icons and the links
returns an Array containing every group and menupoint with their configuration

8.9.3 plugin_parse

Syntax:
plugin_parse ($oldData=array())

Input parameters:
$oldData ... mgmt_plugin as array (optional)

global input parameters:
$mgmt_config

Output:
mgmt_plugin as array

Description:
Reads the plugin configurations from the file system.
Checks the folder defined in mgmt_config and searched for plugins and their configurations files.
It either takes needed values from the configuration, from the $oldData or defaultConfiguration.

8.9.4 plugin_generatedefinition

Syntax:
plugin_generatedefinition ($arrayName, $array)

Input parameters:
$arrayName ... name of array holding the plugin definitions
$array ... configuration array

global input parameters:
$mgmt_config

Output:
plugin array / false on error

Description:
Generates the Array definition used in php for $array with the name of $arrayName
Run recursively through the array and supports boolean, numeric and string types for the key and value
$arrayName -> Name of the Array, $array the array containing the values and keys

8.9.5 plugin_saveconfig

Syntax:
plugin_saveconfig ($configuration)

Input parameters:
$configuration ... configuration as array

global input parameters:
$mgmt_config

Output:
true / false on error

Description:
Saves the plugin configuration $configuration into the configuration file
The configuration file is located in the data/config directory and is named plugin.conf.php

8.9.6 plugin_generatelink

Syntax:
plugin_generatelink ($plugin, $page, $control=false, $additionalGetParameters=false)

Input parameters:
$plugin ... plugin name
$page ... plugin page (relative reference to the plugins main page)
$control ... control (relative reference to the plugins control page)
$additionalGetParameters ... additional GET parameters

global input parameters:
$mgmt_config

Output:
plugin link

Description:
Generates a link to be used when linking to other pages inside of a plugin.

8.10 User Interface API Functions

8.10.1 toggleview

Syntax:
toggleview ($view)

Input parameters:
$view ... view [detail,small,medium,large]

global input parameters:
$mgmt_config

Output:
true / false

Description:
sets explorer objectlist view.

8.10.2 togglesidebar

Syntax:
togglesidebar ($view)

Input parameters:
$view ... view [true,false]

global input parameters:
$mgmt_config

Output:
true / false

Description:
enables or disables sidebar

8.10.3 setfilter

Syntax:
setfilter ($filter_set)

Input parameters:
$filter_set ... set of filtera as array with keys [comp,image,document,video,audio] and value [0,1]

global input parameters:
$mgmt_config

Output:
true / false

Description:
set filter settings for object view in session.

8.10.4 objectfilter

Syntax:
objectfilter ($file)

Input parameters:
$file ... file name

global input parameters:
$mgmt_config
$hcms_ext

Output:
true / false

Description:
if an object / file name is passing the filter-test.

8.10.5 showtopbar

Syntax:
showtopbar ($show, $lang="en", $close_link="", $close_target="", $individual_button="",

$id="bar")

Input parameters:
$show ... message
$lang ... language code (optional)
$close_link ... close button link (optional)
$close_target ... link target (optional)
$individual_button ... individual button (optional)
$id ... ID of div-layer (optional)

global input parameters:
$mgmt_config
$hcms_charset
$hcms_lang

Output:
top bar box / false on error

Description:
shows the standard top bar with or without close button

8.10.6 showtopmenubar

Syntax:
showtopmenubar ($show, $menu_array, $lang="en", $close_link="", $close_target="", $id="bar")

Input parameters:
$show ... message
$menu_array ... menu as array [key=name
$lang ... value=properties/events]
$close_link ... language code (optional)
$close_target ... close button link (optional)
$id ... link target (optional)

global input parameters:
$mgmt_config
$hcms_charset
$hcms_lang

Output:
top bar box / false on error

Description:
shows the menu top bar with or without close button

8.10.7 showmessage

Syntax:
showmessage ($show, $width=580, $height=70, $lang="en", $style="", $id="hcms_messageLayer")

Input parameters:
$show ... message
$width ... width in pixel (optional)
$height ... height in pixel (optional)
$lang ... language code (optional)
$style ... additional style definitions of div-layer (optional)

$id ... ID of div-layer (optional)

global input parameters:
$mgmt_config
$hcms_charset
$hcms_lang

Output:
message box / false on error

Description:
shows the standard message box with close button

8.10.8 showinfopage

Syntax:
showinfopage ($show, $lang="en")

Input parameters:
$show ... message
$lang ... language code (optional)

global input parameters:
$mgmt_config
$hcms_charset
$hcms_lang_codepage
$hcms_lang

Output:
message on html info page / false on error

Description:
shows a full html info page

8.10.9 showinfobox

Syntax:
showinfobox ($show, $lang="en", $sec=4, $style="", $id="hcms_infoLayer")

Input parameters:
$show ... message
$lang ... language code (optional)
$sec ... display for seconds (optional)
$style ... additional style definitions of div-layer (optional)
$id ... ID of div-layer (optional)

global input parameters:
$mgmt_config
$hcms_charset
$hcms_lang_codepage
$hcms_lang

Output:
message in div layer / false on error

Description:
shows infobox for a few seconds

8.10.10 showsharelinks

Syntax:
showsharelinks ($link, $lang="en", $style="", $id="hcms_shareLayer")

Input parameters:
$link ... link to share
$lang ... language code (optional)
$style ... additional style definitions of div-layer (optional)
$id ... ID of div-layer (optional)

global input parameters:
$mgmt_config
$hcms_charset
$hcms_lang_codepage
$hcms_lang

Output:
message in div layer / false on error

Description:
shows share links

8.10.11 showmetadata

Syntax:
showmetadata ($data, $lang="en", $class_headline="hcmsRowData2")

Input parameters:
$data ... meta data as array
$lang ... hierarchy level
$class_headline ... CSS-class with background-color for headlines (optional)

global input parameters:
$mgmt_config
$hcms_charset
$hcms_lang_codepage
$hcms_lang

Output:
result as HTML unordered list / false on error

8.10.12 showobject

Syntax:
showobject ($site, $location, $page, $cat="", $name="")

Input parameters:
$site ... publication name
$location ... location
$page ... object name
$cat ... category [page,comp] (optional)
$name ... object name (optional)

global input parameters:
$mgmt_config
$hcms_charset
$hcms_lang

$lang

Output:
html presentation / false

8.10.13 showmedia

Syntax:
showmedia ($mediafile, $medianame, $viewtype, $id="", $width="", $height="", $class="hcmsImageItem")

Input parameters:
$mediafile ... mediafile (publication/filename)
$medianame ... name of mediafile for display
$viewtype ... view type [template
$id ... preview
$width ... preview_download
$height ... preview_no_rendering]
$class ... ID of the media tag

global input parameters:
$mgmt_config
$mgmt_mediapreview
$mgmt_mediaoptions
$mgmt_imagepreview
$mgmt_docconvert
$hcms_charset
$hcms_lang_codepage
$hcms_lang
$lang
$pdfjs_path = $mgmt_config['url_path_cms']."javascript/pdfpreview/web/viewer.html?file="

Output:
html presentation / false

Description:
this function requires site, location and cat to be set as global variable in order to validate the access permission of the user

8.10.14 showcompexplorer

Syntax:
showcompexplorer ($site, $dir, $location_esc="", $page="", $compcat="multi", $search_expression="", $search_format="", $mediatype="", $lang="en", $callback="", $scalingfactor="1")

Input parameters:
$site ... publication name
$dir ... current explorer location
$location_esc ... object location (optional)
$page ... object name (optional)
$compcat ... component category [single,multi,media] (optional)
$search_expression ... search expression (optional)
$search_format ... search format [object,document,image,video,audio] (optional)
$mediatype ... media-type [audio,video,text,flash,image,compressed,binary] (optional)
$lang ... callback of CKEditor (optional)
$callback ... saclingfactor for images (optional)
$scalingfactor

global input parameters:
$user
$mgmt_config
$siteaccess
$pageaccess
$compaccess
$rootpermission
$globalpermission
$localpermission
$hiddenfolder
$html5file
$temp_complocation
$hcms_charset
$hcms_lang

Output:
explorer with search / false on error

Description:
creates component explorer incl. search form

8.10.15 showeditor

Syntax:
showeditor ($site, $hypertagname, $id, $contentbot="", $sizewidth=600, $sizeheight=300, $toolbar="Default", $lang="en", $dpi=72)

Input parameters:
$site ... publication name
$hypertagname ... hypertag name
$id ... hypertag id
$contentbot ... content
$sizewidth ... width
$sizeheight ... height of the editor
$toolbar ... toolbar set
$lang ... language
$dpi ... dpi for scaling images

global input parameters:
$mgmt_config
$publ_config

Output:
rich text editor code / false on error

Description:
shows the rich text editor

8.10.16 showinlineeditor_head

Syntax:
showinlineeditor_head ($lang)

Input parameters:
$lang ... language

global input parameters:

$mgmt_config
$hcms_charset
$hcms_lang

Output:
rich text editor code for html head section / false on error

Description:
shows the rich text editor code (JS, CSS) for include into the html head section

8.10.17 showinlinedatepicker_head

Syntax:
showinlinedatepicker_head ()

Input parameters:

global input parameters:
$mgmt_config

Output:
date picker code for html head section / false on error

Description:
shows the date picker code (JS, CSS) for include into the html head section

8.10.18 showinlineeditor

Syntax:
showinlineeditor ($site, $hypertag, $id, $contentbot="", $sizewidth=600, $sizeheight=300, $toolbar="Default", $lang="en", $contenttype="", $cat="", $location_esc="", $page="", $contentfile="", $db_connect=0, $token="")

Input parameters:
$site ... publication name
$hypertag ... hypertag
$id ... hypertag id
$contentbot ... content
$sizewidth ... width
$sizeheight ... height of the editor
$toolbar ... toolbar set
$lang ... language
$contenttype ... content-type
$cat ... category[page,comp]
$location_esc ... converted location
$page ... object name
$contentfile ... container name
$db_connect ... DB-connect file name
$token ... security token

global input parameters:
$mgmt_config
$publ_config
$hcms_charset
$hcms_lang

Output:
message box/false on error

Description:
shows the rich text inline editor

8.10.19 showvideoplayer

Syntax:
showvideoplayer ($site, $video_array, $width=320, $height=240, $logo_url="", $id="", $title="", $autoplay=true, $fullscreen=true, $loop=false, $muted=false, $controls=true, $iframe=false, $force_reload=false)

Input parameters:
$site ... videoArray (Array) containing the different html sources
$video_array ... width (Integer) Width of the video in pixel
$width ... height (Integer) Height of the video in pixel
$height ... logo_url (String) Link to the logo which is displayed before you click on play (If the value is null the default logo will be used)
$logo_url ... id (String) The ID of the video (will be generated when empty)
$id ... title (String) The title for this video
$title ... autoplay (Boolean) Should the video be played on load (true)
$autoplay ... default is false
$fullscreen ... enableFullScreen (Boolean) Is it possible to view the video in fullScreen (true)
$loop ... play loop (optional) [true,false]
$muted ... muted/no sound (optional) [true,false]
$controls ... player controls (optional) [true,false]
$iframe ... use video in iframe (optional) [true,false]
$force_reload ... reload video sources to prevent the browser cache to show the same video even if it has been changed [true,false] (optional)

global input parameters:
$mgmt_config

Output:
HTML code of the video player / false on error

Description:
generates a html segment for the video code we use.

8.10.20 showvideoplayer_head

Syntax:
showvideoplayer_head ($secureHref=true, $fullscreen=true)

Input parameters:
$secureHref ... secure hyperreferences by adding 'hypercms_'
$fullscreen ... is it possible to view the video in fullScreen [true,false]

global input parameters:
$mgmt_config

Output:
head for video player / false on error

8.10.21 showaudioplayer

Syntax:
showaudioplayer ($site, $audioArray, $width=320, $height=320, $logo_url="", $id="", $autoplay=false, $loop=false, $controls=true, $force_reload=false)

Input parameters:
$site ... publication name
$audioArray ... audio files as array (Array)
$width ... ID of the tag (optional)
$height ... autoplay (optional) [true,false]
$logo_url ... play loop (optional) [true,false]
$id ... player controls (optional) [true,false]
$autoplay
$loop
$controls
$force_reload

global input parameters:
$mgmt_config

Output:
String Code for the HTML

Description:
Generates a html segment for the video code we use. False on error.

8.10.22 showaudioplayer_head

Syntax:
showaudioplayer_head ($secureHref=true)

Input parameters:
$secureHref ... secure hyperreferences by adding 'hypercms_'

global input parameters:
$mgmt_config

Output:
head for video player

8.10.23 debug_getbacktracestring

Syntax:
debug_getbacktracestring ($valueSeparator, $rowSeparator, $ignoreFunctions=array())

Input parameters:
$valueSeparator ... separator for arguments
$rowSeparator ... separator for a Row on screen/file
$ignoreFunctions ... functionnames to be ignored

Output:
debug message

Description:
Returns the current backtrace as a good readable string
ignores debug and debug_getbacktracestring

8.10.24 showAPIdocs

Syntax:
showAPIdocs ($file, $return="html")

Input parameters:
$file ... path to API file
$return ... return result as HTML or array [html,array] (optional)

global input parameters:
= array()

Output:
HTML output of documentation / false on error

Description:
generates the documentation of an API file

8.11 Template Engine API Functions

8.11.1 checklanguage

Syntax:
checklanguage ($language_array, $language_value)

Input parameters:
$language_array ... language array with all valid values
$language_value ... language value of attribute in hyperCMS tag

Output:
true if language array holds the given language value / false if not found

8.11.2 checkgroupaccess

Syntax:
checkgroupaccess ($groupaccess, $ownergroup)

Input parameters:
$groupaccess ... group access string from hyperCMS group-tag attribute
$ownergroup ... owner groups as array

Output:
true if current ownergroup has access or invalid input / false if not

8.11.3 transformlink

Syntax:
transformlink ($viewstore)

Input parameters:
$viewstore ... view of object

global input parameters:
$site
$location_esc
$page
$ctrlreload
$mgmt_config

Output:
view with transformed links inside publication for easyedit mode

8.11.4 followlink

Syntax:
followlink ($site, $follow)

Input parameters:
$site ... publication name
$follow ... link to follow

global input parameters:
$mgmt_config

Output:
prepared input (location plus page) for easyedit mode (buildview) / false on error

8.11.5 errorhandler

Syntax:
errorhandler ($source_code, $return_code, $error_identifier)

Input parameters:
$source_code ... source code
$return_code ... return code
$error_identifier ... error identifier

Output:
error message and view of the code with line identifiers

8.11.6 viewinclusions

Syntax:
viewinclusions ($site, $viewstore, $hypertag, $view, $application, $charset="UTF-8")

Input parameters:
$site ... view of object
$viewstore ... hypertag to create view of inlcuded objects
$hypertag ... view parameter
$view ... application
$application ... character set used (optional) view-parameter explanation: $view = "template or any other word" -> the standard text (in table) will be included for the view $view = "preview" -> preview of the content of the included file $view = "publish" -> view the content of the included file as ist is (for publishing)
$charset

global input parameters:
$user
$mgmt_config
$location
$hcms_lang
$lang

Output:
view on the content including the content of included objects

8.11.7 buildview

Syntax:

buildview ($site, $location, $page, $user, $buildview="template", $ctrlreload="no", $template="", $container="", $force_cat="", $execute_code=true)

Input parameters:
$site ... publication name
$location ... location
$page ... object
$user ... user
$buildview ... view parameter (optional)
$ctrlreload ... reload workplace control frame and add html & body tags if missing [yes,no] (optional)
$template ... template name (optional)
$container ... container name (optional)
$force_cat ... force category to use different location path [page,comp] (optional)
$execute_code ... execute_code [true/false] (optional)

global input parameters:
$container_collection
$eventsystem
$db_connect
$mgmt_config
$siteaccess
$adminpermission
$setlocalpermission
$token
$mgmt_lang_shortcut_default
$hcms_charset
$hcms_lang_name
$hcms_lang_shortcut
$hcms_lang_codepage
$hcms_lang_date
$hcms_lang
$lang

Output:
result array with view of the content / false on error

Description:
buildview parameter may have the following values:
$buildview = "formedit": use form for content editing
$buildview = "formmeta": use form for content viewing only for meta informations (tag-type must be meta)
$buildview = "formlock": use form for content viewing
$buildview = "cmsview": view of page based on template, includes hyperCMS specific code (buttons)
$buildview = "inlineview": view of page based on template, includes hyperCMS specific code (buttons) and inline text editing
$buildview = "publish": view of page for publishing based on template without CMS specific code (buttons)
$buildview = "preview": view of page based on template for preview (inactive hyperlinks) without CMS specific code (buttons)
$buildview = "template": view of template based on template for preview (inactive hyperlinks) without CMS specific code (buttons)

8.11.8 buildsearchform

Syntax:
buildsearchform ($site, $template, $ownergroup="")

Input parameters:
$site ... publication name
$template ... template name
$ownergroup ... group access as array

global input parameters:
$user
$mgmt_config
$mgmt_lang_shortcut_default
$hcms_charset
$hcms_lang_name
$hcms_lang_shortcut
$hcms_lang_codepage
$hcms_lang_date
$hcms_lang
$lang

Output:
form view

8.11.9 buildbarchart

Syntax:
buildbarchart ($paper_name, $paper_width=600, $paper_height=300, $paper_top=10, $paper_left=40, $x_axis, $y1_axis, $y2_axis="", $y3_axis="", $paper_style="", $bar1_style="", $bar2_style="", $bar3_style="", $show_value=false)

Input parameters:
$paper_name ... name/id of paper
$paper_width ... width of paper in pixel
$paper_height ... height of paper in pixel
$paper_top ... top space in pixel
$paper_left ... left space in pixel
$x_axis ... x-axis values as array
$y1_axis ... y1-axis values as array
$y2_axis ... y2-axis values as array (optional)
$y3_axis ... y3-axis values as array (optional)
$paper_style ... paper CSS style
$bar1_style ... 1st bar chart CSS style
$bar2_style ... 2nd bar chart CSS style
$bar3_style ... 3rd bar chart CSS style
$show_value ... show y-value in bar [true,false]

global input parameters:
$lang
$mgmt_config

Output:
bar chart view / false on error

8.12 XML API Functions

8.12.1 setxmlparameter

Syntax:
setxmlparameter ($xmldata, $parameter, $value)

Input parameters:
$xmldata ... XML content container
$parameter ... paramater name
$value ... paramater value

Output:
XML content container / false on error

Description:
set parameter values in XML declaration (e.g. encoding):
encoding="UTF-8"

8.12.2 getcontent

Syntax:
getcontent ($xmldata, $starttagname)

Input parameters:
$xmldata ... XML content container
$starttagname ... tag name

Output:
result array with the content of the requested XML node (tag) / false on error

Description:
<tagname>content</tagname>
extracts the content between the given $starttagname xml-tags.
only this function will decode special characters (&, <, >) in the content and removes CDATA.
getcontent will only decode values if they are non-xml and non_html. so content inside child nodes
including tags won't be decoded.
wild card character "*" can be used at the end of $starttagname.

8.12.3 geticontent

Syntax:
geticontent ($xmldata, $starttagname)

Input parameters:
$xmldata ... XML content container
$starttagname ... tag name

Output:
result array with the content of the requested XML node (tag) / false on error

Description:
CASE-Insensitive version (XML parser are however always case-sensitive!)
<tagname>content</tagname>
extracts the content between the given $starttagname xml-tags.
only this function will decode special characters (&, <, >) in the content and removes CDATA.
getcontent will only decode values if they are non-xml and non_html. so content inside child nodes
including tags won't be decoded.
wild card character "*" can be used at the end of $starttagname

8.12.4 getxmlcontent

Syntax:

getxmlcontent ($xmldata, $starttagname)

Input parameters:
$xmldata ... XML content container
$starttagname ... tag name

Output:
result array with the content of the requested XML node (tag) / false on error

Description:
<tagname>content</tagname>
extracts the content together with the $starttagname xml tags
this function will NOT decode special characters like function getcontent!
wild card character "*" can be used at the end of $starttagname

8.12.5 getxmlicontent

Syntax:
getxmlicontent ($xmldata, $starttagname)

Input parameters:
$xmldata ... XML content container
$starttagname ... tag name

Output:
result array with the content of the requested XML node (tag) / false on error

Description:
CASE-Insensitive version (XML parser are always case-sensitive!)
<tagname>content</tagname>
extracts the content together with the $starttagname xml tags
this function will NOT decode special characters like function getcontent!
wild card character "*" can be used at the end of $starttagname

8.12.6 selectcontent

Syntax:
selectcontent ($xmldata, $starttagname, $startcondtag, $condvalue)

Input parameters:
$xmldata ... XML content container
$starttagname ... tag name of requested XML node
$startcondtag ... tag holding the conditional value inside the given starttagname
$condvalue ... conditional value

Output:
result array with the content of the requested XML node (tag) / false on error

Description:
<tagname>
.....
<condtag>condvalue</condtag>
.....
</tagname>
extracts the content between the given $starttagname xml tags where the child xml tag $startcondtag
value is equal with the target value $condvalue
wild card character "*" can be used at the end of $starttagname

wild card character "*" can be used at begin and end of $condvalue
Be Aware: $startcondtag must be a child of $starttagname !!!

8.12.7 selecticontent

Syntax:
selecticontent ($xmldata, $starttagname, $startcondtag, $condvalue)

Input parameters:
$xmldata ... XML content container
$starttagname ... tag name of requested XML node
$startcondtag ... tag holding the conditional value inside the given starttagname
$condvalue ... conditional value

Output:
result array with the content of the requested XML node (tag) / false on error

Description:
CASE-Insensitive version (XML parser are always case-sensitive!)
<tagname>
.....
<condtag>condvalue</condtag>
.....
</tagname>
extracts the content between the given $starttagname xml tags where the child xml tag $startcondtag
value is equal with the target value $condvalue
wild card character "*" can be used at the end of $starttagname
wild card character "*" can be used at begin and end of $condvalue

8.12.8 selectxmlcontent

Syntax:
selectxmlcontent ($xmldata, $starttagname, $startcondtag, $condvalue)

Input parameters:
$xmldata ... XML content container
$starttagname ... tag name of requested XML node
$startcondtag ... tag holding the conditional value inside the given starttagname
$condvalue ... conditional value

Output:
result array with the content of the requested XML node (tag) / false on error

Description:
<tagname>
.....
<condtag>condvalue</condtag>
.....
</tagname>
extracts the content between the given $starttagname xml tags where the child xml tag $startcondtag
value is equal with the target value $condvalue
wild card character "*" can be used at begin and end of $condvalue
Be Aware: $startcondtag must be a child of $starttagname !!!

8.12.9 selectxmlicontent

Syntax:
selectxmlicontent ($xmldata, $starttagname, $startcondtag, $condvalue)

Input parameters:
$xmldata ... XML content container
$starttagname ... tag name of requested XML node
$startcondtag ... tag holding the conditional value inside the given starttagname
$condvalue ... conditional value

Output:
result array with the content of the requested XML node (tag) / false on error

Description:
CASE-Insensitive version (XML parser are always case-sensitive!)
<tagname>
.....
<condtag>condvalue</condtag>
.....
</tagname>
extracts the content between the given $starttagname xml tags where the child xml tag $startcondtag
value is equal with the target value $condvalue
wild card character "*" can be used at begin and end of $condvalue

8.12.10 deletecontent

Syntax:
deletecontent ($xmldata, $starttagname, $startcondtag, $condvalue)

Input parameters:
$xmldata ... XML content container
$starttagname ... tag name of requested XML node
$startcondtag ... tag holding the conditional value inside the given starttagname
$condvalue ... conditional value

Output:
XML content container / false on error

Description:
<tagname>
<condtag>condvalue</condtag>
</tagname>
deletes the whole xml content including <tagname>
wild card character "*" can be used at begin and end of $condvalue

8.12.11 deleteicontent

Syntax:
deleteicontent ($xmldata, $starttagname, $startcondtag, $condvalue)

Input parameters:
$xmldata ... XML content container
$starttagname ... tag name of requested XML node
$startcondtag ... tag holding the conditional value inside the given starttagname
$condvalue ... conditional value

Output:
XML content container / false on error

Description:
CASE-Insensitive version (XML parser are always case-sensitive!)
<tagname>
<condtag>condvalue</condtag>
</tagname>
deletes the whole xml content including <tagname>
wild card character "*" can be used at begin and end of $condvalue

8.12.12 setcontent

Syntax:
setcontent ($xmldata, $startparenttagname, $starttagname, $contentnew, $startcondtag="", $condvalue="")

Input parameters:
$xmldata ... XML content container
$startparenttagname ... parent tag name
$starttagname ... tag name of XML node for the new content
$contentnew ... new XML node to be inserted
$startcondtag ... tag holding the conditional value inside the given starttagname
$condvalue ... conditional value

Output:
XML content container / false on error

Description:
<parenttagname>
<condtag>condvalue</condtag>
<tagname>contentnew</tagname>
</parenttagname>
$xmldata = data string to be parsed
$startparenttagname = name of the tag that is a parent node of starttagname (necessary if condition has been set!)
$starttagname = name of the tag (child node)
$contentnew = the content that will be inserted between the child tags $starttagname
$startcondtag = child xml tag where condition will be set
$condvalue = value of the condition
wild card character "*" can be used at begin and end of $condvalue

8.12.13 seticontent

Syntax:
seticontent ($xmldata, $startparenttagname, $starttagname, $contentnew, $startcondtag, $condvalue)

Input parameters:
$xmldata ... XML content container
$startparenttagname ... parent tag name
$starttagname ... tag name of XML node for the new content
$contentnew ... new XML node to be inserted
$startcondtag ... tag holding the conditional value inside the given starttagname
$condvalue ... conditional value

Output:
XML content container / false on error

Description:
CASE-Insensitive version (XML parser are always case-sensitive!)
<parenttagname>
<condtag>condvalue</condtag>
<tagname>contentnew</tagname>
</parenttagname>
$xmldata = data string to be parsed
$startparenttagname = name of the tag that is the parent node of starttagname (necessary if condition has been set!)
$starttagname = name of the tag (child node)
$contentnew = the content that will be inserted between the child tags $starttagname
$startcondtag = child xml tag where condition will be set
$condvalue = value of the condition
wild card character "*" can be used at begin and end of $condvalue

8.12.14 setcontent_fast

Syntax:
setcontent_fast ($xmldata, $startparenttagname, $starttagname, $contentnew, $startcondtag="", $condvalue="")

Input parameters:
$xmldata ... XML content container
$startparenttagname ... parent tag name
$starttagname ... tag name of XML node for the new content
$contentnew ... new XML node to be inserted
$startcondtag ... tag holding the conditional value inside the given starttagname
$condvalue ... conditional value

Output:
XML content container / false on error

Description:
function designed for link management, extremely fast but with limitations (only CASE-Sensitive!)
<parenttagname>
<condtag>condvalue</condtag>
<tagname>contentnew</tagname>
</parenttagname>
$xmldata = data string to be parsed
$startparenttagname = name of the tag that is the parent node of starttagname (necessary if condition has been set!)
$starttagname = name of the tag (child node)
$contentnew = the content that will be inserted between the child tags $starttagname
$startcondtag = child xml tag where condition will be set
$condvalue = value of the condition
wild card character "*" can be used at begin and end of $condvalue

8.12.15 updatecontent

Syntax:
updatecontent ($xmldata, $xmlnode, $xmlnodenew)

Input parameters:
$xmldata ... XML content container
$xmlnode ... XML node to be replaced
$xmlnodenew ... new XML node

Output:
XML content container / false on error

Description:
updates a given xml string $xmlnode in $xmldata with the content $xmlnodenew.
this method provides a faster way to update xml nodes when the node was selected before.

8.12.16 insertcontent

Syntax:
insertcontent ($xmldata, $insertxmldata, $starttagname)

Input parameters:
$xmldata ... XML content container
$insertxmldata ... XML node to be inserted in starttagname
$starttagname ... tag name of the parent XML node

Output:
XML content container / false on error

Description:
.....
.....
<tagname> <- list start
.....
.....
insertxmldata <- insertxmldata
</tagname> <- list end
.....
insert $insertxmldata string at the end of all child between the parent $tagname

8.12.17 inserticontent

Syntax:
inserticontent ($xmldata, $insertxmldata, $starttagname)

Input parameters:
$xmldata ... XML content container
$insertxmldata ... XML node to be inserted in starttagname
$starttagname ... tag name of the parent XML node

Output:
XML content container / false on error

Description:
CASE-Insensitive version (XML parser are always case-sensitive!)
.....
.....
<tagname> <- list start
.....
.....
insertxmldata <- insertxmldata
</tagname> <- list end
.....
insert $insertxmldata string at the end of all child between the parent $tagname

8.12.18 addcontent

Syntax:
addcontent ($xmldata, $sub_xmldata, $startgrandtagname, $startcondtag, $condvalue, $startparenttagname, $starttagname, $contentnew)

Input parameters:
$xmldata ... XML content container
$sub_xmldata ... xml node to be inserted
$startgrandtagname ... grandparent tag name
$startcondtag ... tag holding the conditional value inside the given starttagname
$condvalue ... conditional value
$startparenttagname ... parent tag name
$starttagname ... tag name of XML node for the new content
$contentnew ... new XML node to be inserted

Output:
XML content container / false on error

Description:
<grandtagname>
<condtag>condvalue</condtag>
<parenttagname> <- list start
.....
.....
..... }
<tagname>contentnew</tagname> } <- sub_xmldata
..... }
</parenttagname> <- list end
</grandtagname>
$xmldata = data string to be parsed
$sub_xmldata = xml node to be inserted
$startgrandtagname (optional) = name of the grand xml tag of parent xml tag where (article)
$startcondtag (optional) = xml tag inside the parent xml tags where condition will be set
$condvalue (optional) = value of the condition
$startparenttagname (optional) = name of the parent xml tag where the xml subschema should be added (list)
$starttagname (optional) = name of the tag (child)
$contentnew (optional) = the content that will be inserted between the child tags

8.12.19 addicontent

Syntax:
addicontent ($xmldata, $sub_xmldata, $startgrandtagname, $startcondtag, $condvalue, $startparenttagname, $starttagname, $contentnew)

Input parameters:
$xmldata ... XML content container
$sub_xmldata ... xml node to be inserted
$startgrandtagname ... grandparent tag name
$startcondtag ... tag holding the conditional value inside the given starttagname
$condvalue ... conditional value
$startparenttagname ... parent tag name
$starttagname ... tag name of XML node for the new content
$contentnew ... new XML node to be inserted

Output:
XML content container / false on error

Description:
CASE-Insensitive version (XML parser are always case-sensitive!)
<grandtagname>
<condtag>condvalue</condtag>
<parenttagname> <- list start
.....
.....
..... }
<tagname>contentnew</tagname> } <- sub_xmldata
..... }
</parenttagname> <- list end
</grandtagname>
$xmldata = data string to be parsed
$sub_xmldata = xml subschema to be inserted
$startgrandtagname (optional) = name of the grand xml tag of parent xml tag where (article)
$startcondtag (optional) = xml tag inside the parent xml tags where condition will be set
$condvalue (optional) = value of the condition
$startparenttagname (optional) = name of the parent xml tag where the xml subschema should be added (list)
$starttagname (optional) = name of the tag (child)
$contentnew (optional) = the content that will be inserted between the child tags

9 Rechtliche Hinweise / Impressum

9.1 Fragen und Anregungen

Sollten Sie weitergehende Fragen oder Anregungen zum Produkt haben, so wenden Sie sich bitte an den Support. Wir stehen Ihnen auch gerne für Fragen bezüglich unseres Reseller-Programms und Partner-Programms zur Verfügung. Zugriff auf die erweitere Online-Demo des hyper Content Management Servers können sie ebenfalls über den Support beantragen.

hyperCMS Support:
support@hypercms.com
http://www.hypercms.com

9.2 Impressum

Verantwortlich für den Inhalt:

hyperCMS
Content Management Solutions GmbH
Rembrandtstr. 35/6
A-1020 Wien – Austria

office@hypercms.com
http://www.hypercms.com

9.3 Rechtliche Hinweise

Vorliegendes Benutzerhandbuch basiert auf der zum Zeitpunkt der Verfassung des Dokumentes verfügbaren Programmversion.

Der Hersteller behält sich Programmänderungen und –Verbesserungen vor.

Fehler und Irrtümer vorbehalten.