Content type

A **content type** in CONTENIDO is a place where authors can enter or select data to be displayed in the frontend.

Content types have to be inserted into a module in order to be used. A module can use more than one content type though.

List of content types

Below you find a list of content types that are provided by the CONTENIDO core. Content types are stored in database in the table "type". Content that is stored using a content type can be found in the table "content".

Name	Class	Description
CMS_HTMLHEAD	cContentTypeHtmlhead	Provides a textfield in which HTML text can be entered (with WYSIWYG editor). This type is principally the same as CMS_HTML.
CMS_HTML	cContentTypeHtml	Provides a textfield in which HTML text can be entered (with WYSIWYG editor).
CMS_TEXT	cContentTypeText	Provides a plain text field in which raw text can be entered.
CMS_IMG	cContentTypeImg	Returns the path to an image configured by CMS_IMGEDITOR.
CMS_IMGDESCR	cContentTypeImgdescr	Returns the description of an image. This type i configured by CMS_IMGEDITOR.
CMS_LINK	cContentTypeLink	Returns the raw link configured by CMS_LINKEDITOR.
CMS_LINKTARGET	cContentTypeLinktarget	Returns the target of a link. This type is configured by CMS_LINKEDITOR.
CMS_LINKDESCR	cContentTypeLinkdescr	Returns the description of a link. This type is configured by CMS_LINKEDITOR.
CMS_HEAD	cContentTypeHead	Provides a plain text field in which raw text can be entered. This type is principally the same as CMS_TEXT.
CMS_DATE	cContentTypeDate	Provides a datepicker to select a date from a calendar and a date format. The selected date is then shown in the selected format.
CMS_TEASER	cContentTypeTeaser	Provides a configuration dialog to manage dynamic article lists (known as "teaser") which are displayed in the frontend.
CMS_FILELIST	cContentTypeFilelist	Provides a configuration dialog to manage dynamic file lists which are displayed in the frontend.
CMS_IMGEDITOR	cContentTypeImgeditor	Provides a configuration dialog to select an image for display in frontend.
CMS_LINKEDITOR	cContentTypeLinkeditor	Provides a configuration dialog to manage a special link in frontend.
CMS_RAW	cContentTypeRaw	Configurable only by and for administrators, not indexable. Available since CONTENIDO version 4.9.5

Using content types

CONTENIDO comes with several content types for different purposes. Content types are defined and set in the specific module and have two display modes: the output mode for the frontend and the edit mode for the backend.

Every content type has its unique content ID. That gives you the possibility to use multiple same content types within the same article but with different content.

The module code parser detects content types by it's short notation. This is like accessing an array entry: CONTENT_TYPE[ID]. If you define the content ID twice or more, that content type will manage the same content. When you are using content types which only return values (e.g. for links and images) you MUST use the content type ID, which was used to define the editor.

Simplified code block for module output

```
<?php
echo "CMS_HTML[1]";

echo "Managing Link: CMS_LINKEDITOR[2]";
echo "CMS_LINK[2] with target CMS_LINKTARGET[2]";
?>
```

Developer notes

If you want to develop your own content type you should have a look at the "type" table and an arbitrary content type class (located in contentido/classes /content_types).

When adding a new content type to the database you have to choose an ID greater or equal to 10.000 cause on a system update all content types with an ID less than 10.000 are deleted and created anew.

getContentType

This function allows to create text-, image-, link- or other elements dynamically. Thus it is possible to give these elements dynamic IDs without having to use the short notation where you have to know the content types ID in advance.

```
/**
 * four for business AG : [http://4fb.de/ http://4fb.de/]
 * OiverL : [http://www.team4media.net/ http://www.team4media.net/]
 * KrissKrass aka ChriZZoW : [http://www.publicstyles.de/ http://www.publicstyles.de/]
 */
function getContentType($container_type, $container_id) {
    global $a_content, $idartlang, $idart, $idcat, $lang, $db, $edit, $sess, $client, $cfg, $cfgClient;
    $sql = "SELECT * FROM ".$cfg["tab"]["type"]." WHERE type = '$container_type'";
    $db->query($sql);
    $db->next_record();
    $cms_code = $db->f("code");
    $cms_idtype = $db->f("idtype");
    if( !$edit ) {
        $db2 = new DB_Contenido;
        $db2->query("
            SELECT
                *
            FROM
                ".$cfg["tab"]["content"]." AS A
                , ".$cfg["tab"]["art_lang"]." AS B, ".$cfg["tab"]["type"]." AS C
            WHERE
                A.idtype = C.idtype
                AND A.idartlang = B.idartlang
                AND B.idart = '".Contenido_Security::toInteger($idart)."'
                AND B.idlang = '".Contenido_Security::escapeDB($lang, $db)."'
                AND A.idtype = '".$cms_idtype."'
                AND A.typeid = '".$container_id."'");
        $db2->next_record();
        $a_content[$db2->f("type")][$db2->f("typeid")] = $db2->f("value");
    }
    $val = $container_id;
    eval($cms_code);
    $tmp_output = str_replace('\\\"','"',$tmp);
    $tmp_output = stripslashes($tmp_output);
    return $tmp_output;
}
```

Please have a look at the forum for questions and help with this code.

cApiArticleLanguage::getContentObject

This function will return an object of the specified content type. You can use this object to get more information about the content.

Example

```
$artLang = new cApiArticleLanguage(1);
$cmsLinkEditor = $artLang->getContentObject("CMS_LINKEDITOR", 1);
echo($cmsLinkEditor->getTarget()); // will echo the link target (e.g. "_blank")
```

Don't forget to take a look at the complete API documentation over here.