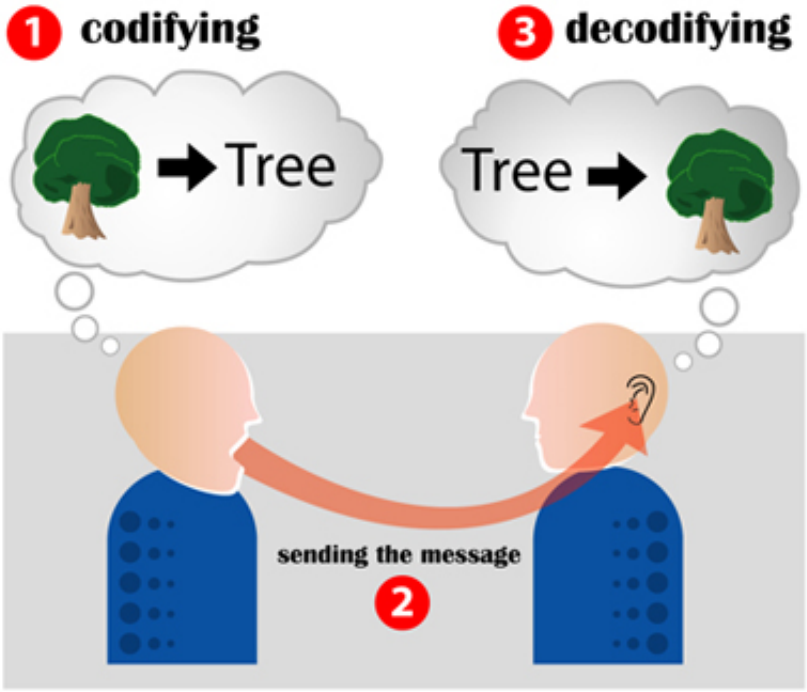
# How to convert file content encoded in windows-cp1251 charset to UTF-8 (with iconv) to be delivered properly encoded to browsing end clients

**Author :** admin




I have a *bunch of old html files all encoded in the historically obsolete Windows-cp1251*. Windows-CP1251 used to be common used 7 years ago and therefore still *big portions of the web content in Bulgarian / Russian Cyrillic* is still transferred to the end users in this encoding.

This was just before the "UTF-8 revolution", where massively people started using UTF-8,
Well it was clear the specific national country text encoding standards will quickly be moved by to **UTF-8 - Universal Encoding format** which abbreviation stands for (*Unicode Transformation Format).*

Though UTF-8 was clear to be "the future", many web developers mostly because of their incompetency or using an old sources of learning how to writen in HTML continued to use *windows-cp1251* in HTMLs. I'm even convinced, there are still developers out there who are writting websites for Bulgarian / Russian / Macedonian customers using obsolete encodings ...

The smarter developers of those accustomed to **windows-cp1251**, **KOI-8R** etc. etc., were using the meta tag to specify the type of charset of the web page content with:

or

Anyhow, still many devs even didn't placed the windows-cp1251 in the head of the HTML ...

The result for the system administrator is always a mess - a lot of webpages that are showing like unreadable signs and tons of unhappy customers.
As always the system administrator is considered responsible, for the programmer mistakes :). So instead of programmers fix their bad cooking, the admin has to fix it all!

One quick work around me as admin has applied to failing to display pages in Cyrillic using the *Windows-cp1251* character encoding was to force *windows-cp1251* as a default encoding for the whole virtualhost or Apache directory with Apache directives like:

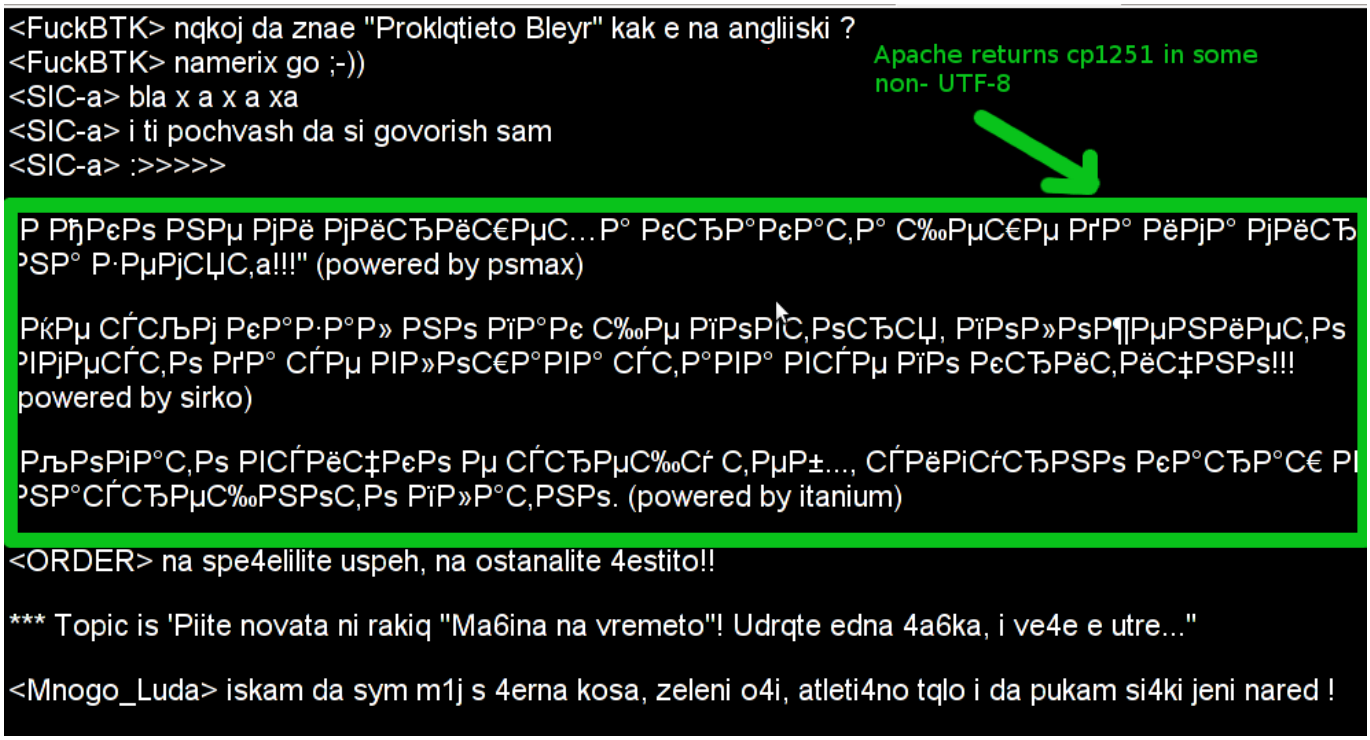
ServerAdmin some_user@some_host.com
DocumentRoot /var/www/html
AddDefaultCharset windows-cp1251
ServerName the_host_name.com
ServerAlias www.the_host_name.com
....
....

AddDefaultCharset windows-cp1251
>/Directory>

Though this mostly would, work there are some occasions, where only *a particular html files* from all the content served by Apache is encoded in **windows-cp1251**, if most of the content is already written in UTF-8, this could be a big issues as you cannot just change the UTF-8 globally to windows-cp1251, just because few pages are written in archaic encoding....
Since most of the content is displayed to the client by Apache (as prior explained) just fine, only particular htmls lets's ay *single.html, single2.html* etc. etc. are displayed with some question marks or some *non-human readable "hieroglyphs"*.

Below is a screenshot from two pages returned to my browser in wrongly set htmls charset:

**Improper Windows CP1251 delivered page in UTF-8 browser view**




**Improperly served encoding CP1251 delivered by Apache in non-utf-8 encoding**

When this kind of issues occur, the only solution is to simply login to the server and use **iconv** command to convert all files returning unreadable content from whatever the non UTF-8 encoding is lets say in my case *Bulgarian typeset of **cp1251** to **UTF-8***

Here is how the *iconv command to convert between windows-cp1251 to utf-8* the two sample files named *single1.html and single2.html*

```
server:/web# /usr/bin/iconv -f WINDOWS-1251 -t UTF-8 single1.html > single1.html.utf8
server:/web# mv single1.html single1.html.bak;
server:/web# mv single1.html.utf8 single1.html
server:/web# /usr/bin/iconv -f WINDOWS-1251 -t UTF-8 single2.html > single2.html.utf8
server:/web# mv single2.html single2.html.bak;
server:/web# mv single2.html.utf8 single2.html
```

I always, make copies of the original cp1251 encoded files (as you see *mv single1.html single1.html.bak*), because if something goes wrong with convertion I can easily revert back.

If there are 10 files with consequential numbers naming they can be converted using a short for loop, like so:

```
server:/web# for i $(seq 1 10); do
/usr/bin/iconv -f WINDOWS-1251 -t UTF-8 single$i.html > single$i.html.utf8;mv single$i.html
single$i.html.bak
mv single$i.html.utf8 single$i.html
done
```

Just as earlier mentioned if single1.html, single2.html ... has in the html :

You should open, each of the files in question and wipe out the line either by hand or use **sed** to wipe it in one loop if it has to be done for lets say 10 files named (single{1..10})

```
server:/web# for i in $(seq 1 10); do
sed '//d' single$i.txt > single$i.txt.new;
mv single$i.txt single$i.txt.bak;
mv single$i.txt.new single$i.txt
```

Well now,