

SEP #12 : Scipad / Add support for different encodings

Title: Scipad / Add support for different encodings

Version: 1.0

Author: Francois Vogel – fvogelnew1@free.fr

Review:

Commented:

State: Draft

Scilab-Version: Trunk

Vote: No

Created: 15/12/08

Abstract

This SEP proposes to enhance Scipad by adding support for other encodings than the system encoding. The user will then be able to edit files encoded in UTF-8 or any other available encoding.

Rationale

Scilab development is currently heading toward more internalization. This trend brings the need for editing files containing text encoded in possibly different encoding than the system encoding.

This is especially true when dealing with Scilab xml help files, which are a frequent use case for developers.

The proposed enhancement aims at providing users the ability to load, edit and save text files in the encoding of their choice (that is: encodings the underlying Tcl installation knows about), all from within Scipad.

Implementation

Implementation is based on the Tcl command *encoding*, see its documentation here:

<http://www.tcl.tk/man/tcl8.5/TclCmd/encoding.htm>

The default system encoding is the platform default encoding that will be detected at Scipad launch.

A new menu will be added to the existing Options menu of Scipad, and will contain all the encodings available (i.e. [*encoding names*]) in a series of radiobuttons.

The encoding name attached to a file will be a new property of each Scipad buffer: *\$listoffile("\$textarea",encoding)*. It is therefore not a global setting but is local and may be different for each buffer.

When the user selects an item from the Encoding menu, it sets *\$listoffile("\$textarea",encoding)* to the selected encoding name and also switches the encoding used by Tcl to this selected encoding (*encoding system \$newencoding*)

When a file is read from disk or saved, *\$listoffile("\$textarea",encoding)* is used to configure the channel (*fconfigure \$ch -encoding ...*), ensuring proper encoding of the file is exported to the OS (when saving) or used for file content interpretation (when loading).

Whenever the encoding currently used is different from the platform default encoding, this will be

shown to the user in the Scipad title bar (and tile title, if applicable) by appending a tag such as "(utf-8)". This is much the same principle as for the already existing "(modified)" or "(readonly)" tag. When the file encoding matches the default encoding, no tag will be appended, thus reducing unnecessary clutter.

No additional tag will be added to the file names stored in the Windows menu of Scipad.

The whole implementation of this SEP will only impact the scipad module of Scilab.

Typical use case in Scipad

1. Select the encoding with which the next file should be read.
2. Open the file.
3. Fiddle with the content, add text, possibly switch buffers, select other encodings, etc.
4. Before saving, select the encoding that Scipad shall use for saving the file.
5. Save the file.

Alternatives

None that I can see at the moment.

Compatibility

This feature will be added in Scilab master GIT branch. There is no specific limitation foreseen for the newly provided features covered by this SEP.

Limitations

1. Encoding conversion

Encoding conversion is not part of this SEP.

This feature would take the current buffer as input. This content is supposed to be encoded in the currently selected encoding (encoding A). Scipad would then convert this content into a different encoding B, which would as a side effect become the current encoding for the file. This would be done in place in the current buffer.

This encoding conversion function would certainly be handy, for instance when a file has been opened with the wrong encoding. Encoding conversion could then be used to check visually that the new encoding better fits the file content.

This feature would be based on Tcl commands *encoding convertfrom* and *encoding convertto*.

The present SEP could be augmented to include an encoding conversion feature if the upcoming discussion turns out to prove there is a need for it.

2. Auto-detection of encoding

Although it would be convenient, this SEP does not propose auto-detection of encodings upon file reading. This is not a trivial topic, apart maybe for xml files. A few references are:

<http://chardet.feedparser.org/docs/faq.html>

<http://feedparser.org/docs/character-encoding.html>

[http://msdn.microsoft.com/en-us/library/aa741220\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/aa741220(VS.85).aspx)

The specific case of xml files is the only one in the typical Scilab workflow for which it would make sense to perform auto-detection of encoding. It is also the only case with a reasonably easy implementation. The proposed approach would be to parse the xml prolog for the "encoding" attribute. This prolog usually (at least in Scilab) looks like the following example:

```
<?xml version="1.0" encoding="UTF-8"?>
```

In case such parsing fails, or if the encoding name found does not match any known encoding, Scipad would not attempt to change the encoding currently selected by the user.

The present SEP could be augmented to include auto-detection of encoding for xml files if the upcoming discussion turns out to prove this is a valuable feature.

Detailed reference implementation

The diffs provided below are against the master git branch dated 14 dec. 2008 (Scipad 7.11)

```
--- tcl/msg_files/fr_fr.msg Sat Dec 13 15:13:14 2008
+++ tcl/msg_files/fr_fr.msg Sun Dec 14 21:10:16 2008
@@ -25,7 +25,7 @@
#

# Scipad texts in French by Francois Vogel
-# in sync at least with v7.11
+# in sync at least with v7.12

# commons
@@ -640,6 +640,7 @@
::msgcat::mcset fr_fr "Execution &errors" "E&rreurs d'exécution"
::msgcat::mcset fr_fr "In Scilab &shell only" "Dans la fenêtre &Scilab seulement"
::msgcat::mcset fr_fr "Copied in a &message box" "Recopiées dans une &boîte de dialogue"
+ ::msgcat::mcset fr_fr "Enc&oding" "Encod&age"
::msgcat::mcset fr_fr "E&xit options" "Option&s de fermeture"
::msgcat::mcset fr_fr "Show closure &X" "Afficher le bouton &X de fermeture"
::msgcat::mcset fr_fr "Exit on &last file close" "Quitte si &dernier fichier fermé"
--- tcl/buffernavigation.tcl Thu Nov 13 21:23:19 2008
+++ tcl/buffernavigation.tcl Sun Dec 14 21:36:56 2008
@@ -358,6 +358,7 @@
# Set all the settings such that $textarea becomes the current one
global pad Scheme ColorizeIt listoffile textareaid
global buffermodifiedsinclastsearch
+ global currentencoding

# clear the selection when leaving a buffer - check first that the
# textarea still exists because it might have been destroyed when
@@ -376,6 +377,7 @@
keyposn $textarea
set Scheme $listoffile("$textarea",language)
set ColorizeIt $listoffile("$textarea",colorize)
+ set currentencoding $listoffile("$textarea",encoding)
schememenus $textarea
highlighttextarea $textarea
TextStyles $textarea
@@ -546,6 +548,7 @@
# it creates a new empty textarea
global winopened listoffile
global listoftextarea pad
+ global defaultencoding

# ensure that the cursor is changed to the default cursor
event generate [gettextareacur] <Leave>
@@ -564,6 +567,7 @@
set listoffile("$pad.new$winopened",undostackdepth) 0
set listoffile("$pad.new$winopened",redostackdepth) 0
set listoffile("$pad.new$winopened",progressbar_id) ""
+ set listoffile("$pad.new$winopened",encoding) $defaultencoding
lappend listoftextarea $pad.new$winopened

addwindowmenuentry $winopened $listoffile("$pad.new$winopened",displayedname)
@@ -617,6 +621,7 @@
set listoffile("$newta",undostackdepth) $listoffile("$ta",undostackdepth)
set listoffile("$newta",redostackdepth) $listoffile("$ta",redostackdepth)
```

```

    set listoffile("$newta",progressbar_id) $listoffile("$sta",progressbar_id)
+   set listoffile("$newta",encoding) $listoffile("$sta",encoding)
    lappend listoftextarea $newta

    addwindowsmenuentry $winopened $listoffile("$pad.new$winopened",displayedname)
--- tcl/defaults.tcl   Sat Dec 13 15:13:14 2008
+++ tcl/defaults.tcl   Sun Dec 14 21:57:31 2008
@@ -296,6 +296,10 @@
    # frame pathname in which $textarea is packed, or "none" if it is not packed
    array unset pwframe

+# default encoding is the system native encoding
+set defaultencoding [encoding system]
+set currentencoding $defaultencoding
+
+#####
+ # source the user preferences file if any
+ # this must happen after the locale selection from Scilab's getlanguage() above
--- tcl/filecommands.tcl   Sat Dec 13 15:13:14 2008
+++ tcl/filecommands.tcl   Sun Dec 14 23:04:27 2008
@@ -111,6 +111,10 @@
    # true: file gets colored
    # false: no colorization for this file
    #
+#   listoffile("$sta",encoding)
+#   name of the encoding in which the file is stored,
+#   e.g. utf-8 or euc-jp or cpl252 or a lot of other possibilities
+#
+ # The windows menu entries are radionbuttons, with the following
+ # properties:
+ # -value is $winopened
@@ -135,6 +139,7 @@
proc filesetasnew {} {
    global winopened listoffile
    global listoftextarea pad
+   global defaultencoding

    # ensure that the cursor is changed to the default cursor
    event generate [gettextareacur] <Leave>
@@ -151,6 +156,7 @@
    set listoffile("$pad.new$winopened",undostackdepth) 0
    set listoffile("$pad.new$winopened",redostackdepth) 0
    set listoffile("$pad.new$winopened",progressbar_id) ""
+   set listoffile("$pad.new$winopened",encoding) $defaultencoding
    lappend listoftextarea $pad.new$winopened

    addwindowsmenuentry $winopened $listoffile("$pad.new$winopened",displayedname)
@@ -341,6 +347,7 @@
    unset listoffile("$textarea",undostackdepth)
    unset listoffile("$textarea",redostackdepth)
    unset listoffile("$textarea",progressbar_id)
+   unset listoffile("$textarea",encoding)

    # the rest of this proc is similar to proc hidetext,
    # but not identical
@@ -838,6 +845,8 @@
+ # $file is not opened - this sets the $listoffile area values for that file
+ # and adds an entry in the windows menu
    global winopened pad listoffile
+   global currentencoding
+
    incr winopened
    dupWidgetOption [gettextareacur] $pad.new$winopened
    set listoffile("$pad.new$winopened",fullname) [file normalize $file]
@@ -854,6 +863,8 @@
    set listoffile("$pad.new$winopened",undostackdepth) 0
    set listoffile("$pad.new$winopened",redostackdepth) 0
    set listoffile("$pad.new$winopened",progressbar_id) ""
+   # there is no automatic detection of encoding (so far) <TODO>: do it at least for xml files
+   set listoffile("$pad.new$winopened",encoding) $currentencoding

    addwindowsmenuentry $winopened $listoffile("$pad.new$winopened",displayedname)
}
@@ -899,6 +910,7 @@
+ # really open/read a file from disk
+ # all readability tests have normally been done before
    global listoftextarea pad closeinitialbufferallowed
+   global listoffile
    set msgWait [mc "Wait seconds while loading and colorizing file"]
    showinfo $msgWait

```

```

        lappend listoftextarea $textarea
@@ -907,6 +919,7 @@
        closefile $pad.new1
    }
    set newnamefile [open $thefile r]
+   fconfigure $newnamefile -encoding $listoffile("$textarea",encoding)
    while {[eof $newnamefile]} {
        $textarea insert end [read -nonewline $newnamefile ]
    }
@@ -1137,7 +1150,7 @@
    # really write the file onto the disk
    # all writability tests have normally been done before
    global filebackupdepth tcl_platform
-   global pad
+   global pad listoffile

    if {$nobackupskip} {
        backupfile $nametosave $filebackupdepth
@@ -1186,6 +1199,7 @@
    }

    set FileNameToSave [open $nametosave w]
+   fconfigure $FileNameToSave -encoding $listoffile("$textarea",encoding)
    puts -nonewline $FileNameToSave [$pad.temptextwidget get 1.0 end]
    close $FileNameToSave

@@ -1329,6 +1343,24 @@
    foreach ta [filteroutpeers $listoftextarea] {
        checkiffilechangedondisk $ta
    }
+}
+
+#####
+# file encoding procs
+#####
+proc setencoding {} {
+   # set the encoding property of the current buffer to be the encoding
+   # currently selected in the encoding options menu
+   # this proc is called when selecting any option of this menu
+   global currentencoding listoffile
+   set textarea [gettextareacur]
+   foreach ta [getfullpeerset $textarea] {
+       set listoffile("$ta",encoding) $currentencoding
+       modifiedtitle $ta
+   }
+   # this is mandatory because any system calls get mangled otherwise
+   # such as passing a filename to the system
+   encoding system $currentencoding
+}

#####
--- tcl/infomessages.tcl      Sat Dec 13 15:13:14 2008
+++ tcl/infomessages.tcl      Sun Dec 14 22:56:31 2008
@@ -173,24 +173,30 @@

proc modifiedtitle {textarea {panesonly "false"}} {
    # Set the Scipad window title to the name of the file displayed in $textarea
-# and add tags (modified, readonly)
+# and add tags (modified, readonly, encoding (if different from the system
+# encoding at Scipad startup))
    # Do the same for the pane title if it exists (i.e. if not maximized)
    # Update also the visual indications of the modified state of the buffer.
    # This includes title bar, colorization of the windows menu entry and
    # colorization of an area in the status bar
    global pad winTitle ScipadVersion listoffile
    global MenuEntryId
-   set fname $listoffile("$textarea",displayedname)
-   set ind [extractindexfromlabel $pad.filemenu.wind $fname]
-   set mod1 ""; set mod2 ""
+   global defaultencoding
+
+   if {$listoffile("$textarea",readonly) == 1} {
+       set mod1 [mc " \[ReadOnly\]"]
+   } else {
+       set mod1 ""
+   }
+
+   if {[isanymodified]} {
+       $pad.statusind configure -background PeachPuff
+   } else {

```

```

        $pad.statusind configure -background [$pad.filemenu cget -background]
    }
+
+   set fname $listoffile("$textarea",displayedname)
+   set ind [extractindexfromlabel $pad.filemenu.wind $fname]
+   if {[ismodified $textarea]} {
+       set mod2 [mc " (modified)"]
+       if {$ind !=-1} {
@@ -198,25 +204,35 @@
+           -activebackground LightSalmon
+       }
+       $pad.statusind configure -background Salmon
-   } else {
+   } else {
+       set mod2 ""
+       if {$ind !=-1} {
+           $pad.filemenu.wind entryconfigure $ind -background "" \
+               -activebackground ""
+       }
+   }
+
+   if {$listoffile("$textarea",encoding) != $defaultencoding} {
+       set mod3 " ($listoffile("$textarea",encoding))"
+   } else {
+       set mod3 ""
+   }
+
+   if {$panesonly == "false"} {
+       # caught because scan will fail when launched from wish
+       if {[catch {
-           scan $ScipadVersion "%s - %s" ScipadVersionNumber ScipadVersionString
+           wm title $pad "$winTitle $ScipadVersionNumber - $fname$mod1$mod2"
+           wm title $pad "$winTitle $ScipadVersionNumber - $fname$mod1$mod2$mod3"
+       }]} {
-           wm title $pad "$winTitle - $fname$mod1$mod2"
+           wm title $pad "$winTitle - $fname$mod1$mod2$mod3"
+       }
+   }
+
+   if {[isdisplayed $textarea]} {
-       [getpaneframename $textarea].panetitle configure \
+       -text "$fname$mod1$mod2"
+       -text "$fname$mod1$mod2$mod3"
+   }
+
+   if {[ismodified $textarea] && \
+       $listoffile("$textarea",thetime) !=0} {
+       $pad.filemenu.files entryconfigure \
--- tcl/mainwindow.tcl Thu Nov 13 21:23:19 2008
+++ tcl/mainwindow.tcl Sun Dec 14 21:34:40 2008
@@ -42,6 +42,7 @@
+   set listoffile("$pad.new$winopened",undostackdepth) 0; # used to enable/disable the undo menu entry
+   set listoffile("$pad.new$winopened",redostackdepth) 0; # used to enable/disable the redo menu entry
+   set listoffile("$pad.new$winopened",progressbar_id) ""; # colorization progressbar identifier
+set listoffile("$pad.new$winopened",encoding) $defaultencoding

+   set chset() {}
+   set words() {}
--- tcl/menues.tcl Thu Nov 13 21:23:19 2008
+++ tcl/menues.tcl Sun Dec 14 22:04:04 2008
@@ -33,6 +33,7 @@
+   global Shift_Tab
+   foreach c1 "$bgcolors $fgcolors" {global $c1}
+   global Tk85
+   global currentencoding

+   #destroy old menus (used when changing language)
+   foreach w [wininfo children $pad.filemenu] {
@@ -380,7 +381,7 @@
+       -variable lang -value $l -command relocate"
+   }
+ }
-# feature temporary disabled as not yet 100% ok (see bindings/issues.txt)
+# feature enabled yet not 100% ok (see bindings/issues.txt) - teasing!
+   menu $pad.filemenu.options.bindings -tearoff 0
+   eval "$pad.filemenu.options add cascade [me "&Bindings style"] \
+       -menu $pad.filemenu.options.bindings "
@@ -405,6 +406,13 @@
+   eval "$pad.filemenu.options.messageboxes add radiobutton \
+       [me "Copied in a &message box"] \

```

```

        -value true -variable ScilabErrorMessageBox"
+   eval "$pad.filemenu.options add cascade [me "Enc&oding"] \
+   -menu $pad.filemenu.options.encodings"
+   menu $pad.filemenu.options.encodings -tearoff 1
+   foreach en [lsort -dictionary [encoding names]] {
+   $pad.filemenu.options.encodings add radiobutton -label $en \
+   -command {setencoding} -value $en -variable currentencoding
+   }
+   menu $pad.filemenu.options.exitopts -tearoff 0
+   eval "$pad.filemenu.options add cascade [me "E&xit options"] \
+   -menu $pad.filemenu.options.exitopts "
--- changelog.txt      Sat Dec 13 15:13:14 2008
+++ changelog.txt      Mon Dec 15 22:43:07 2008
@@ -1,3 +1,9 @@
+Francois VOGEL, 14/12/08
+ * Selection of the file encoding is now possible through a new option menu
+ (this is a request from Yung-Jang Lee) - This is the implementation
+ corresponding to SEP#12
+ * version --> 7.11.SEP12
+
+Francois VOGEL, 13/12/08
+ * Fixed bug 3806 (drag and drop fails because package tkdnd is not found)
+ * File/Revert did not update the readonly flag. Reverting to the saved state
@@ -40,7 +46,8 @@
+   the last peer in the list (instead of the focused peer) did win the title
+   bar label
+ * Added new command "Create help from head comments" - Initial code and
-   function help_from_sci courtesy of Torbjørn Pettersen
+   function help_from_sci courtesy of Torbjørn Pettersen - This is the
+   implementation corresponding to SEP#9
+ * version --> 7.7

+Francois VOGEL, 03/10/08
--- version.xml        Sat Dec 13 15:13:14 2008
+++ version.xml        Mon Dec 15 22:43:17 2008
@@ -15,5 +15,5 @@
-->
<!-- ===== -->

-<VERSION major="7" minor="11" maintenance="0" revision="0" string="SciPad unstable-svn" />
+<VERSION major="7" minor="11" maintenance="SEP12" revision="0" string="SciPad unstable-svn" />
</MODULE_VERSION>

```

Still <TODO>

- Discuss and accept the feature in the current coverage of this SEP.
- Decide whether this SEP should be augmented to cover encoding conversion and/or auto-detection of encodings as well.
- Check implications on bug 2520

Changelog

1.0 – F. Vogel 15/12/08, Initial version.

Copyright

This document has been placed under the NOL.