Localization of GUI Text Prompts

This page is wildly out of date. Localization is now RPM based.

In release 3.10, the sipXecs Configuration Server component uses 225 properties files to define all text prompts that can be localized. During the evolution of the system, properties files are relatively frequently modified (text prompts changed, new files added, existing files removed). Because of that, a set of properties files always matches a specific release/revision of the sipXecs software. This churn causes two problems:

- A localization package is tied to a specific version of sipXecs.
- If we decided to provide a default set of properties files, the set would be outdated after the first change following the creation of the set.

Due to that, be aware that you must prepare an update of your localization package(s) whenever you intend to upgrade to a newer version of sipXecs. Also, instead of providing a default set of properties files ready for localization, a better approach is to provide a Linux script that extracts these files from a checked out revision of sipXecs. That said, providing a TAR file with properties file for each maintenance branch (that is not likely to change anymore) could be also a good idea.

There are two groups of properties files with GUI text prompts - files in the WEB-INF directory structure and files in the etc/sipxpbx directory structure.

1. The first group (the majority of properties files) is located within subdirectories of the WEB-INF directory. During the build of sipXecs, all files within the WEB_INF directory structure are stored in the sipxconfig.war file. To add localized files that correspond to files in this WAR file, a JAR file with the same directory structure (that contains the localized files) is used. This JAR file is used as a supplementary message store that provides additional files not present in the war file. JAR files with properties files are stored into the etc/sipxpbx directory. In order for the Configuration Server component to find newly installed JAR files (installed from localization packages), a restart of Configuration Server is necessary.
2. The second group of properties files is located within subdirectories of the etc/sipxpbx directory. During the build of sipXecs, several system level properties files and all properties files for plugins are placed into this directory structure. To add localized files that correspond to files within etc /sipxpbx, the localized files must be placed into the same location where English files are stored.

Properties Files' Extraction Script

To collect all the properties files from the sipXecs build environment, you'd have to get all the files and manually discard those that are not required. A convenient script is provided to handle this task. To extract properties files from a checked out revision of sipXecs, invoke the following script from the base directory. The script creates a TAR file in your home directory with the file name Properties_<revision>.tar with all the properties files that require translation.

```
#!/bin/bash
# This script extracts files *.properties that contain default (US English)
# prompts used in the sipXecs GUI
# Get the current revision number from subversion
REVISION=$(svn info | grep Revision)
rc=$?
if [$rc -ne 0 ]; then
    echo Unable to determine revision number - aborting!
    echo Please invoke the script from the main sipXecs directory.
    exit $rc
fi
REVISION=$(echo $REVISION | cut -c 11-15)
# Go to the sipXconfig directory
cd sipXconfig
rc=$?
if [$rc -ne 0 ]; then
    echo Unable to enter the sipXconfig directory - aborting!
    echo Please invoke the script from the main sipXecs directory.
    exit $rc
fi
# Create a file name using the revision number
FILENAME=Properties_$REVISION.tar
echo The output file $FILENAME is being created in your home directory...
# Store files *.properties from all subdirectories that contain default prompts
# in a tar file. Exlcude files that do not contain text prompts.
find . | grep properties | grep -v .svn | grep -v _de | grep -v _pl | grep -v _it | grep -v properties.in | \
    grep -v lib.properties | grep -v .\meta | grep -v log4j | grep -v skin.properties | \
    grep -v DurationFormat | grep -v spy.properties | grep -v admin/mail.properties | \
    grep -v properties.map | cut -c 3- | xargs tar cvf ~/$FILENAME
rc=$?
if [$rc -ne 0 ]; then
    echo Unable to create the tar file
else
    echo The tar file $FILENAME was successfully created in your home directory
fi
cd ..
exit $rc
```

Properties Files' Renaming Script

Another step in the localization of GUI text prompts is to rename all files (to add a language ID as the suffix to each file name). Again, to handle this task could be relatively tedious, so a convenient script (rename_properties.sh) is provided below. To rename properties files that were stored in the TAR file by the script above, invoke the script with the TAR file name as the first parameter and your language ID as the second parameter. For properties files, the language ID suffix added to the names of properties files must be either:

- a lower case language tag
- a lower case language tag and an upper case region tag separated by an underscore (e.g., es_MX for Mexican Spanish)
  If your language ID uses both the language tag and the region tag, please make sure to use the correct separator and capitalization.

```
#!/bin/bash
# This script extracts files *.properties from the specified tar file, appends
# the specified language extension to all files and then creates a new tar file
# with the name of the original tar file plus the language extension.
#
# Check the file name to make sure the user specified a tar file with properties files
# Check the language identifier.
if [$# -eq 2 ]; then
    echo $1 | grep .tar >> /dev/null
    rc=$?
    if [$rc -eq 0 ]; then
        echo $1 | grep Properties >> /dev/null
        rc=$?
        if [oldxx: _rc -eq 0 ]; then
            LANGUAGE_ID=$2
            rc=1
            if [$-(#LANGUAGE_ID\} -eq 2 ]; then
                rc=0
            elif [$-(#LANGUAGE_ID\} -eq 5 ]; then
                echo $LANGUAGE_ID | cut -c 3 | grep '-' >> /dev/null
                rc=$?
            fi
        fi
    fi
else
    rc=1
fi
if [$rc -ne 0 ]; then
    echo Usage: rename_properties.sh Properties_XXXXX.tar LanguageID
    echo E.g. : rename_properties.sh Properties_12200.tar cs
    exit $?
fi
# Create a temprorary directory for the files
FILENAME_TAR="$1"
TMP=Temp_Properties
echo Extracting files from $FILENAME_TAR
mkdir $TMP
cd $TMP
rc=$?
if [$rc -ne 0 ]; then
    echo Unable to create the directory $TMP - aborting
    rc=$?
fi
# Make sure the directory is empty
rm -fR *
# Extract the files
echo $FILENAME_TAR | cut -c 0-1 | grep /
if [$? -eq 0 ]; then
    tar xf $FILENAME_TAR
else
    tar xf ../$FILENAME_TAR
fi
# Rename each file
echo Renaming files...
find . | grep .properties | while read line
do
    NEWFILENAME="$(echo $line | cut -c 3- | cut -f 1 --delimiter=.)_$LANGUAGE_ID.properties"
    mv $line $NEWFILENAME
done
# Store the renamed files into a tar file
NEWFILENAME_TAR="$(echo $FILENAME_TAR | cut -f 1 --delimiter=.)_$LANGUAGE_ID.tar"
echo Storing files in $NEWFILENAME_TAR
tar cf ../$NEWFILENAME_TAR *
# Remove the temporary directory
cd ..
rm -fR $TMP
```