Perl Sample Scripts

The following scripts provide examples of using custom scripting capabilities in a typical Agiloft installation.

Note: The Agiloft distribution includes Perl and will automatically use it to execute scripts with a .pl subscript. If you want to use your own version of perl, you may do so by specifying it on the first line of the script, eg #!/usr/local/perl.

```
 ######################################################################
# Synopsis:
#
# Redirect the user after he/she submits the ticket.
#
# There is no need to specify the Perl interpreter if the script is named *.pl
# because the system recognizes .pl scripts as being Perl based and will
# use the Perl that is included in the distribution.
#
# If you use this script with your own copy of Perl, please note that
# it requires SimpleXML module to be installed
#
######################################################################
use strict;
use warnings;
use EWget;
use EWset;
my ($input_fname, $output_fname) = @ARGV;
EWget::load($input_fname);
######################################################################
#
# Only log the user out if he/she is not a member of the Staff
# group. In other words, we want all members of staff to be able
# to log tickets on behalf of users without being logged out.
######################################################################
my $groups = EWget::getGlobalVariable('my_groups');
my $groupAllowedToClose = "Staff";
if ($groups !~ m/(^|,)$groupAllowedToClose(,|$)/){
        EWset::setRedirect("http://www.example.com");
    EWset::exit_ew("AcceptChangesAndExit", $output_fname);
} else {
    EWset::exit_ew("AcceptChanges", $output_fname);
}
```

```
######################################################################
# Synopsis:
#
# Redirect the user after he/she submits the ticket.
#
# There is no need to specify the Perl interpreter if the script is named *.pl
# because the system recognizes .pl scripts as being Perl based and will
# use the Perl that is included in the distribution.
#
# If you use this script with your own copy of Perl, please note that
# it requires SimpleXML module to be installed
#
######################################################################
use strict;
use warnings;
use EWget;
use EWset;
my ($input_fname, $output_fname) = @ARGV;
EWget::load($input_fname);
######################################################################
#
# Only log the user out if he/she is not a member of the Staff
# group. In other words, we want all members of staff to be able
# to log tickets on behalf of users without being logged out.
######################################################################
my $groups = EWget::getGlobalVariable('my_groups');
my $groupAllowedToClose = "Staff";
if ($groups !~ m/(^|,)$groupAllowedToClose(,|$)/){
        EWset::setRedirect("http://www.example.com");
    EWset::exit_ew("AcceptChangesAndExit", $output_fname);
} else {
    EWset::exit_ew("AcceptChanges", $output_fname);
}
```

```
############################################################################
# Synopsis:
# Check if the user belongs to QA group and block closure of the ticket if
#  he/she doesn't.
#
# There is no need to specify the Perl interpreter if the script
# is named *.pl because the system recognizes .pl
# scripts as being Perl based and will
# use the Perl that is included in the distribution.
#
# If you use this script with your own copy of Perl, please note that
# it requires SimpleXML module to be installed
############################################################################
use strict;
use warnings;
use EWget;
use EWset;
my ($input_fname, $output_fname) = @ARGV;
EWget::load($input_fname);
my $groups = EWget::getGlobalVariable('my_groups');
my $groupAllowedToClose = "QA";
if ($groups !~ m/(^|,)$groupAllowedToClose(,|$)/
    && EWget::getValue('current_state', 'wfstate') eq "Closed"
    && EWget::getValue('old_state', 'wfstate') ne "Closed"){
        EWset::setMessage("You don't belong to group, that can close tickets");
       # Set a field value. In this case, we set the status field
        # to its old value
    EWset::setRecordField('wfstate', EWget::getValue('old_state', 'wfstate'));
        EWset::save($output_fname);
}
EWset::exit_ew("AcceptChanges", $output_fname);
```

```
######################################################################
# Synopsis:
# This script performs some common tasks typical of a script.
# You could run this script against the input file example:
# filename.pl input.xml output.xml
#
# There is no need to specify the Perl interpreter if the script
# is named *.pl because the system recognizes .pl
# scripts as being Perl based and will
# use the Perl that is included in the distribution.
#
# If you use this script with your own copy of Perl, please note that
# it requires SimpleXML module to be installed
#
######################################################################
use strict;
use warnings;
```

```
# interface modules to interact with input and output xml files
use EWget;
use EWset;
# Retreive files names from arguments...
my ($input_fname, $output_fname) = @ARGV;
# and load input into internal structures. This is required step before
# accessing input by API calls.
EWget::load($input_fname);
# we will use this to accumulate messages to user
my $message = "";
# Let's retreive user name to provide personalized message.
# That would be a value of Full Name from
# Contacts table entry for user, editing the ticket.
my $full_name = EWget::getGlobalVariable('my_full_name');
# Let's check if user is a creator of record and prohibit
# the editing if he is not, with respective message.
# The same could be done with permissions, but access
# control through scripts can be more sophisticated.
# For example, you could use EWget::getGlobalVariable('my_start_work')
# and EWget::getGlobalVariable('my_stop_work')
# (if there are such fields in Contacts table) to allow
# modifications only in working time.
open (output_file, ">/tmp/aaaa");
print output_file EWget::getGlobalVariable('my_full_name') . "\n";
print output_file EWget::getValue('current_state', 'created_by') . "\n";
close (output_file);
if(EWget::getGlobalVariable('my_login')
    ne EWget::getValue('current_state', 'login')){
# We compare here login from global vars that is login of user, who
# edits the record with
# record creator login. Now let's show the message to describe
# the denial reason.
       EWset::setMessage("Sorry, ".$full_name.
    ", you are not the creator of this record, you cannot modify it.");
        EWset::setExitAction("RejectChanges");
# and set exit code to reject the changes
} else {
       # Now let's try to detect some changes in the record.
    # For example, let's make deleteable flag
        # non-editable without blocking other changes.
       if(EWget::getValue('current_state', 'deleteable')
        ne EWget::getValue('old_state', 'deleteable')){
               # We will notify user, that change is not allowed.
               $message="Sorry, ".$full_name.
        ", you are cannot modify deleteable flag".
        ", your change will be ignored. ";
               # Let's return the old value back.
               EWset::setRecordField('deleteable',
            EWget::getValue('old_state', 'deleteable'));
        }
       # Now some more complex techniques - linked
    # records access.
        # Let's find out, for example, the name and
    # phone of record owner
        # if case is "open" ("owned by" is a Linked
    # Field relationship with the contacts table)
    if("Open" eq EWget::getValue('current_state', 'wfstate')){
               # Retreive phone and name from first (index=0) record
        # linked to owned_by field.
               my $owner_phone = EWget::getLinkedValue("current_state",
                "end_user_name", 0, "direct_phone");
               my $owner_name = EWget::getLinkedValue("current_state",
                "end_user_name", 0, "full_name");
               $message = $message."You can contact ".$owner_name.
                ", owner of this support case by number ".$owner_phone." ";
        }
       # Now check if this case assigned to user or team - example of
    # logic over multiple tables Linked Field
    my @linkedRecordNames = EWget::getLinkedRecordsNamesList("current_state",
        "assigned_team");
       # In fact, there is always 0 or 1 element in this array,
        # since multiple table Linked Fields are not exported
       if(EWget::getTableNameForRecord($linkedRecordNames[0]) eq "teams"){
               $message = $message."This issue assigned to ".
         EWget::getLinkedValue("current_state", "assigned_team", 0, "_name").
            " team.";
        }
       # store full message
        EWset::setMessage($message);
       # accept changes to ticket
        EWset::setExitAction("AcceptChanges");
}
# And don't forget to save the output file.
EWset::save($output_fname);
```