PhUSE US Connect 2019

Paper CT04

# Quick Start Your Trials – A Shell Script for Preparing Your Trials with a Standard Set-up

Edita Karosiene, Novo Nordisk A/S, Søborg, Denmark
Johan Ejstrud, Novo Nordisk A/S, Aalborg, Denmark

## ABSTRACT

In this paper we present a shell script which can set-up your trial in a few seconds. The script is using a template directory where the files needed to start a trial are placed. It can automatically find all the files in the template directory and copy them to the corresponding trial directory. If the trial ID is used anywhere in the file or directory name or within the file content, it will be replaced with an actual trial ID. In addition to this, the script automatically recognizes the updates made in the template directory so files can be easily moved, added or removed within the template with no need to update the script. The shell script can run on UNIX® or Linux® servers and can easily be adjusted for any other tasks as well.

## INTRODUCTION

It is often the case that clinical trials or projects will have a similar technical set-up within the same company. That includes having some dedicated locations where all trial directories are placed and having a naming convention to name them. Moreover, there should be an overlap between directory structure and files that are used for many trials. This suggests an idea to have a trial template directory ready which is utilized as an example for every new trial.

Having such template is already a very good idea, however it is very important to use it efficiently. It might happen that, after a new trial is created, a responsible person would manually copy files and directories that are required for the new trial from the template. That might include renaming some files and directories to be specific for the trial. Such a manual step is wasting time and resources and is prone to human errors.

Another way is to have a script which copies files and directories automatically. In this paper we present such a script called *prepare_trial.sh*. Bellow are the advantages of this script:

- The script is independent of any specific directory structure and naming standards
- The script is independent of the location of your trials in regards to your root directory. For example, it will work the same in *a* and *b* cases bellow:
    a. /root/home/trial_dir
    b. /root/home/very/long/path/to/trial_dir
- The script automatically finds what files and directories are within trial template and copies them to corresponding locations in the chosen trial
- The script automatically recognizes updates made in the template directory. There is no need to update the script if you change something in your template
- If trial ID has to be part of the name of some file or directory, the script can automatically include the correct trial ID for each trial
- If trial ID has to be included in the content of a text file, the script will update the file with the correct trial ID
- Running the script takes few seconds and guarantees that every trial is set-up in the same way with correct files and directories.

## SCRIPT OVERVIEW

- Name: *prepare_trial.sh*
- Type: shell script
- Environment: prepared using HP-UX® server, Korn shell® and Perl v5.8.8
- Runs on: UNIX or Linux servers.

The following sections present the set-up, requirements and usage of the script.

**PREREQUISITES**

- Trial directories should be differentiate by trial IDs
- Names of files and directories should not contain white spaces and other special characters for UNIX-like operating systems
- Trial directory, which files and directories should be copied to, must exist
- A *[trial-id]* tag must be used everywhere in the template directory where actual trial ID should occur in the trial. This is valid for directory names, file names and text inside plain text files.

**SET-UP**

In order to use the script, you should modify two lines in it (Appendix 1, line 13 and 15):

- Provide full path to the directory where all your trials are located. Do not leave any white spaces before or after '=' sign.

```
path_to_trials=/full/path/to/your/trials
```

- Provide a full path to your template directory. Do not leave any white spaces before or after '=' sign.

```
path_to_template_dir=/full/path/to/your/template_dir
```

**USAGE**

The script can run from anywhere on your server as long as you give a full path to it. Make sure that the script is saved in a UNIX format and has correct permissions to be executed. You can use the following command:

```
chmod 755 prepate_trial.sh
```

This will give you read, write and execute permissions and only read and execute permissions to everyone else.

Run the script by giving trial ID as a parameter on the command line. To run the script from your current directory:

```
./prepare_trial.sh t0001
```

## FUNCTIONALITY

The functionality of the *prepare_trial.sh* script is illustrated in Figure 1. The figure shows an example of a template directory which contains four directories: *data*, *documents*, *programs* and *utilities*. In addition to this, *data* directory has *adam* and *sdtm* directories, *programs* directory contains *adam_prog* and *sdtm_prog*. The *adam_prog* directory has some template programs that should be copied to each trial: *mk_adsl.sas* and *mk_adae.sas*. One can also see that within *documents* there is an ADRG template file which should be copied to each trial and renamed to have trial ID in the name. The same is with a SAS® program to run Pinnacle 21 checks within the *utilities*. In these cases, the files will be renamed to have actual trial ID instead of the *[trial-id]* tag.

The *prepare_trial.sh* script should be run after the top level trial directory has been created. For example, a trial with an ID *t0001* has just been initiated, trial directory *t0001* has been created and is empty. As shown in Figure 1, running *prepare_trial.sh* will populate *t0001* directory with all the files and directories from the template. Note that the file names have been updated by replacing *[trial-id]* to *t0001*, e.g. *adrg_[trial-id].docx* is called *adrg_t0001.docx* within the trial.

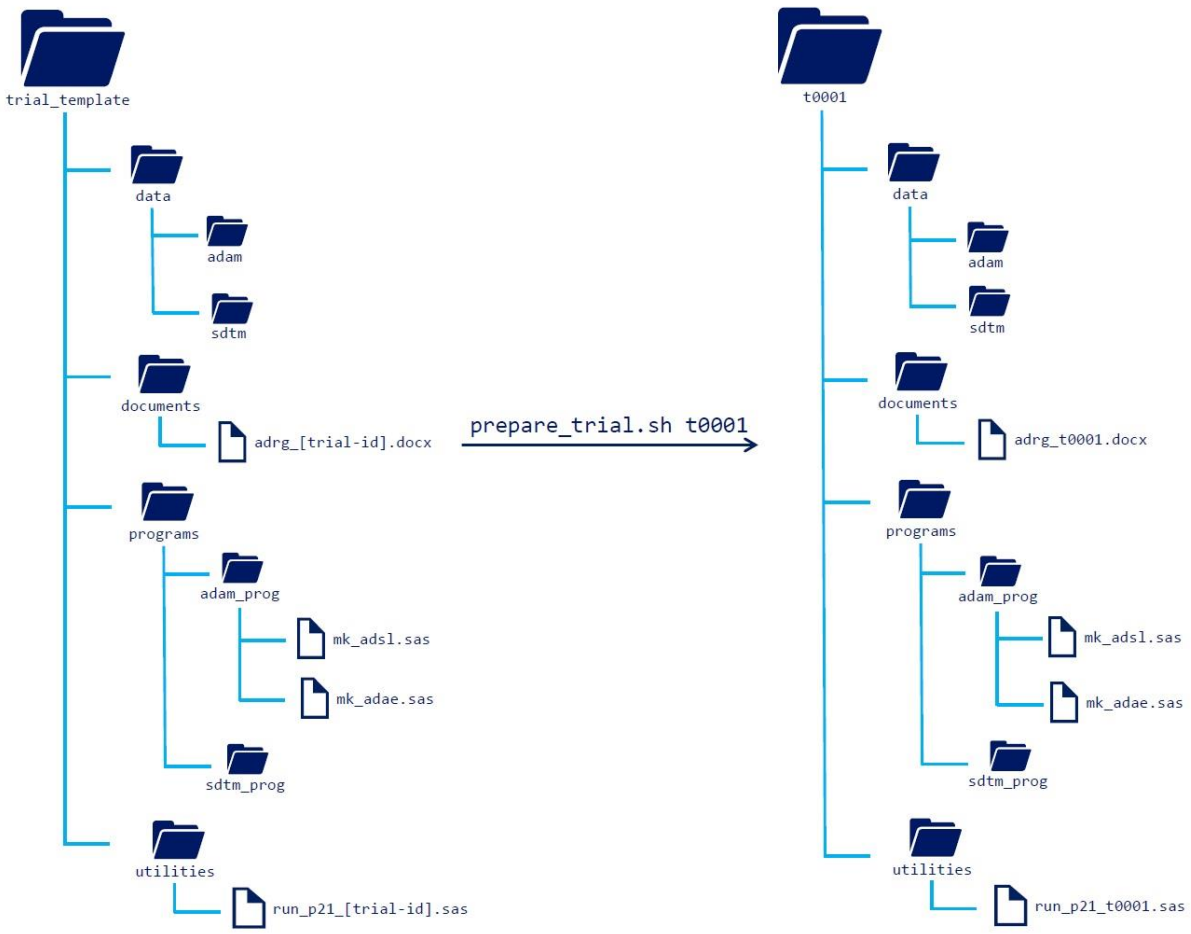


**Figure 1. Schematic functionality of the *prepare_trial.sh* script**

**UPDATING TEMPLATE DIRECTORY**

The script is flexible to changes in the trial template directory. If after some time it has been decided that from now the structure is changing for each trial, one only has to make changes in the template directory. The updates will be reflected for the new trials. For instance, it has been decided to make changes in the *trial_template* directory shown in Figure 1. One more ADaM template program has been created (*mk_adlb.sas*) and should be copied to every trial. Also, the *utilities* directory is not needed anymore. One only has to add *mk_adlb.sas* into *trial_template/programs/adam_prog* and remove *utilities* directory from *trial_template*. Once the template directory is updated, the changes will be reflected in the next trial after the script is executed. This is shown in Figure 2 using a new trial *t0002*.

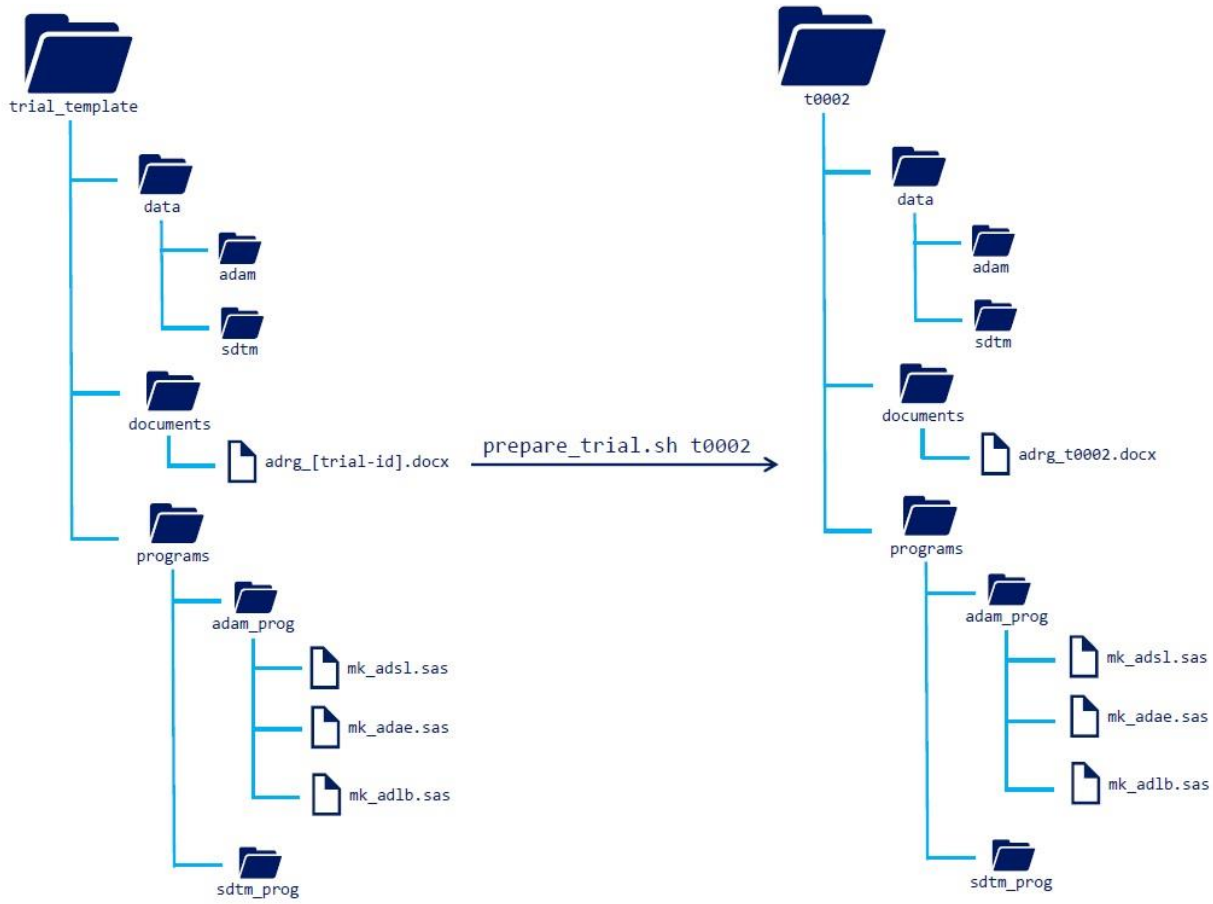


**Figure 2. Schematic representation of how changes in *trial_template* directory are automatically captured by *prepare_trial.sh* script**

**UPDATING CONTENT OF FILES**

An important advantage of *prepare_trial.sh* script is that it can also automatically update the content of the files when copying them from *trial_template* directory to the chosen trial. The script is set-up to replace *[trial-id]* tag to the correct trial ID in the plain text files, such as .txt, .sas, .csv, .sh, .pl, .xml etc. Note, that SAS datasets, Excel® sheets, Word® documents, .pdf files and figures are not plain text files and the replacement will not be done in those. The usage example of this functionally is demonstrated in Figure 3. If you have some template programs that have the trial ID which should be updated in the header or the libname setup, the script will do that automatically.

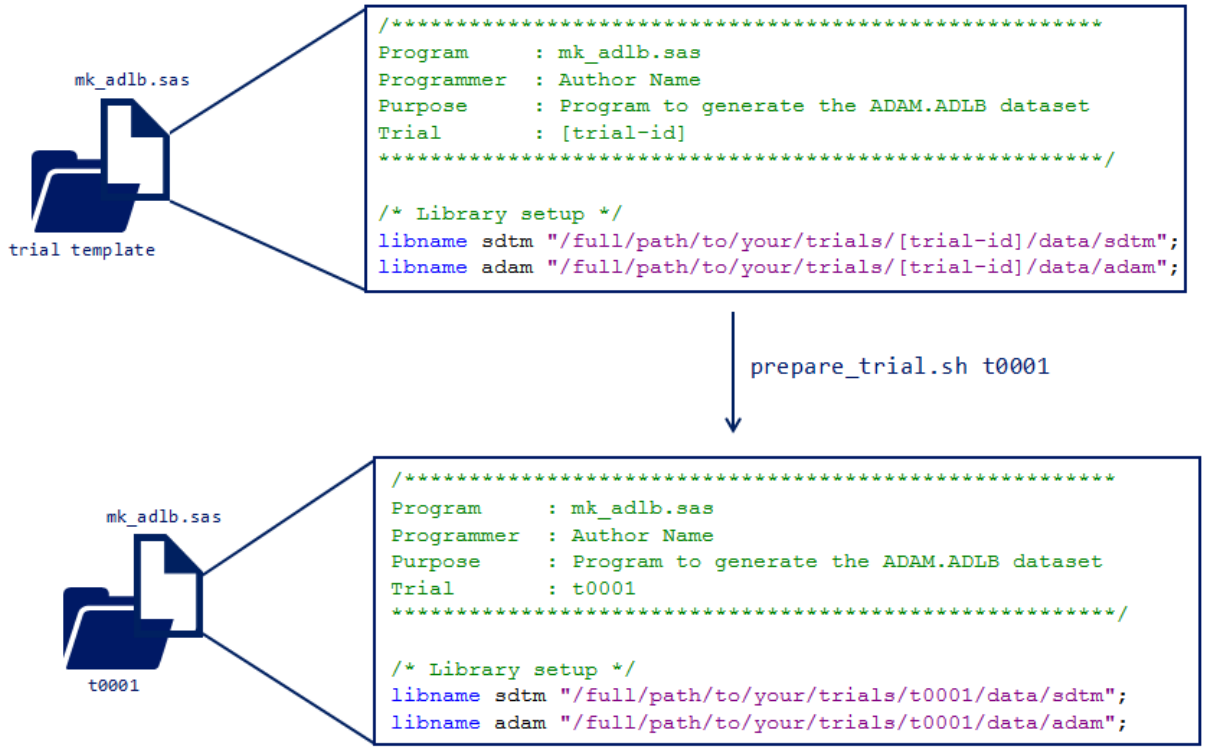


**Figure 3. An example of file content update by *prepare_trial.sh* script**

## A LOOK INTO THE CODE

Writing scripts that should copy files or directories from one location to the other would usually consist of multiple *cp* commands such as

```
cp source_file target_file
```

However, such scripts are very dependent of the full path and the name of each file or directory, making them very inflexible. The *prepare_trial.sh* script solves this issue and in this section we would like to shortly discuss how it is done. Note that the full script is available in Appendix 1.

The following part of the script (Appendix 1, lines 43-50) is finding all the directories that are within *trial_template* and creating directories with the same names inside the chosen trial. Utilizing *find* command, makes the script flexible. *find* command will give a list of existing directories within the template at the time of running the script. Due to this, if some directories have been removed, moved or renamed, it will be reflected in the result.

After getting the list of all directories (*$dirs_in_tmpl_dir*), we loop through that list and construct the name for the new directory (*$new_dir*) which should be called the same as in the template directory but should be inside the chosen trial. To achieve this, *$path_to_template_dir* is replaced by *$trial_dir*. In addition to this, if the directory contains *[trial-id]* tag, it is replaced by the actual trial ID. Finally, the new directory is created if it does not exist yet.

```
dirs_in_tmpl_dir=`find $path_to_template_dir -type d -name "*"`
for dir in $dirs_in_tmpl_dir; do
    new_dir=`echo "$dir" | perl -pe "s|\Q$path_to_template_dir\E|$trial_dir|g" \
            | perl -pe "s|\[trial-id\]|$trial_id|g"`
    if [ ! -d $new_dir ]; then
        mkdir -p $new_dir
    fi
done
```

Similar logic is applied to the files (Appendix 1, lines 54-71). There is an extra step, which is using Perl one-liner to replace *[trial-id]* tag to trial ID inside the text files.

```
perl -i -pe "s|\[trial-id\]|$trial_id|g" $new_file
```

## CONCLUSION

It is a big advantage to have a template trial directory when working on multiple trials which is usually the case for companies and CROs within the pharmaceutical industry. In this paper we presented a script which can help to utilize such a template directory in the most efficient way helping to save time, resources and avoid introducing human errors. The script is developed using UNIX shell programming and runs on UNIX or Linux servers. The script will automatically set-up your trial based on the template directory in a few seconds. It will also include a chosen trial ID where needed including file names, directory names or text file content. The biggest advantages of the script is that it is independent from location of your trials or naming standards. In addition to this, it is flexible to changes in template directory and does not require to be updated after the template directory has been changed. The script is easy to set-up and even easier to run. Take it, use it and enjoy the time you save by not setting-up your trials manually!

## CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors at:


Author Name: Edita Karosiene
Company: Novo Nordisk A/S
Address: Vandtarnsvej 114
2860 Soeborg
Denmark
Work Phone: +45 3079 9516
Email: edka@novonordisk.com
Web: www.novonordisk.com

Author Name: Johan Ejstrud
Company: Novo Nordisk A/S
Address: Alfred Nobels Vej 27
9220 Aalborg
Denmark
Work Phone: +45 3079 8218
Email: jejt@novonordisk.com
Web: www.novonordisk.com

## APPENDIX 1

```
#! /usr/bin/ksh
# Author      : Edita Karosiene, Novo Nordisk A/S, edka@novonordisk.com
# Date        : 2019-01-02
# Description : The script copies files and directories from a template directory to
#               a chosen trial directory. Directory name should match trial-id
# Environment : The script was prepared using HP-UX server, Korn shell and Perl v5.8.8
# Parameters  : trial-id
# Usage       : prepare trial.sh trial-id, e.g. prepare trial.sh t0001
################################################################################################

# Input
# Enter full path to directory where your trials are
path_to_trials=/full/path/to/your/trials
# Enter full path to your template directory
path_to_template_dir=/full/path/to/your/template_dir
# End of input

# Check the number of parameters given
if [ $# -ne 1 ]; then
    echo "Exactly one argument (trial-id) is needed. For example:"
    echo "prepare_trial.sh t0001"
    exit 1
fi
trial_id=$1

# Get full path to trial directory
trial_dir=$path_to_trials/$trial_id

# Check if the template directory exists
if [ ! -d $path_to_template_dir ]; then
    echo "Template directory $path_to_template_dir does not exist."
    exit 1
fi

# Check if the trial directory exists
if [ ! -d $trial_dir ]; then
    echo "Trial directory $trial_dir does not exist."
    exit 1
fi

# Find all the directories inside the template directory and create them in the
# given trial directory if they don't exist. Change [trial-id] to a given trial-id
dirs_in_tmpl_dir=`find $path_to_template_dir -type d -name "*"`
for dir in $dirs_in_tmpl_dir; do
    new_dir=`echo "$dir" | perl -pe "s|\Q$path_to_template_dir\E|$trial_dir|g" \
            | perl -pe "s|\[trial-id\]|$trial_id|g"`
    if [ ! -d $new_dir ]; then
        mkdir -p $new_dir
    fi
done

# Find all the files in the template directory and copy to the given trial
# directory by changing [trial-id] to a given trial-id
print_copy_statement=true
files_in_templ_dir=`find $path_to_template_dir -type f -name "*"`
for file in $files_in_templ_dir; do
    new_file=`echo "$file" | perl -pe "s|\Q$path_to_template_dir\E|$trial_dir|g" \
             | perl -pe "s|\[trial-id\]|$trial_id|g"`
    # If a file does not exist in target location, copy it.
    if [ ! -f $new_file ]; then
        cp $file $new_file
        # Replace "[trial-id]" in file contents with actual trial-id
        perl -i -pe "s|\[trial-id\]|$trial_id|g" $new_file
    else
        if $print_copy_statement; then
            echo "Files that already exist in target location (not copied):"
            print_copy_statement=false
        fi
        echo "\t$file"
    fi
done
```