# `disttree`: A Small Language for Creating Distribution Trees

Ulrich Hoffmann

May 17, 2002

**Abstract**

This document describes the syntax and semantics of the small specification language `disttree` which can be used to conviniently specify the mapping from build trees to distribution trees for software deployment. The reader will learn about the language elements and, a processor for `disttree`, and a small example.

## 1 Introduction

When large software systems are created, the task to finally deploy the software on a target system is of significant work. Typically a build process generates files in a directory structure (the *build tree*), which is structured according to software development needs. The target system often requires a rather different directory structure (the *distribution tree*) to successfully run the software. It is necessary to have a mechanism to map from the build tree to the distribution tree. The small language `disttree` is such a mechanism, that allows to specify how to construct a complete distribution tree from a build tree. The use of a macro substitution mechanism in the specification text allows for easy adaptation to changes in both trees. Based on the specification a `disttree` processor actually copies files fromt the build tree to their appriopriate places in the distribution tree. The processor allows for conventient generation of log/manifest files and is easily integrated in a fully automated build process.

## 2 `disttree` Language Elements

`disttree` is designed to be as simple as possible in order to be easily learned. It is as sophisticated as necessary to express the various needs for convenient distribution tree generation. `disttree` has a line oriented syntax. Each language element has to be written on a line of its own. The line lenght is not limited.

## 2.1 Comments

Comments in `disttree` are important to explain the rationale and the structure of the specification. A `disttree` comment is a lines which first non whitespace character is a number sign `#`. All remaining characters until the end of line are ignored.

## 2.2 Variable Assignments

A `disttree` variable identifier is a lower or uppercase letter, oder _ character possibly followed by additional lower-, uppercase letters or _s or digits. Thus `temp3`, `ROOT`, `CONFIG_DATA` are all variable identifiers, whereas `SRC-DIR` is not. Case is significant.

`disttree` assignments have the form variable=<rhs>

rhs is a string, which possibly contains variable references of the form `$`variable. The value of variable is expanded in order to create a variable free string. This string is assigned to be the value of the variable on the left hand side of the assignement.

## 2.3 Target Directory Specifications

A target directory `[`path`]`

## 2.4 Source File Specifications

path

files and directories

## 2.5 Message Output

`%error` message

`%print`

`%print` message

## 2.6 Conditional Processing

`%ifdef` variable

`%ifndef` variable

`%else`

`%endif`

### 2.7 Looped Processing

`%for` variable1, variable2, ... `in` sequence
`%endfor`

### 2.8 External Command Execution

`@` command

## 3 A Processor for `disttree`

```
Distribution Tree Creator
Create a distribution tree according to disttreespec

Usage: python disttree.py [options] disttreespec
       options:
          -v             verbose
          -q             on copy error, ask retry question
          -n             don't actually perform copy
          -w             just warn if a file does not exist
          -f             force (overwrite existing r/o files)
          -a             reset read-only attributes
          -c             don't create empty directories
          -m             generate md5sums in logfile
          -l logfile     write log file
          -D name=value  preset variable

Example: python disttree.py -v -D DST=X:\ distribution.spec
```

## 4 A sample distribution tree specification

```
# A sample disttree script

# Notify user that generation has started
%print start

# SRC and DST are typically set when invoking
# the disttree processor
%ifndef DST
DST=
%endif

%ifndef SRC
```

```
%error SRC must be set using -D when invoking disttree
%endif

# Setup a target directory ROOT
ROOT=$DST\ROOT

# Define Source locations
LOCVOB=$SRC\localization
COMP1VOB=$SRC\comp1
C1BIN=$COMP1\bin
COMP2VOB=$SRC\comp2
C2BIN=$COMP2\bin
COMP3VOB=$SRC\comp3
C3BIN=$COMP3\bin

# Setup localization information
GERMAN=German,de
FRENCH=French,fr
ENGLISH=English,en

LANGUAGES=$GERMAN,$FRENCH,$ENGLISH

# iterate localization

%for LANG,L in $LANGUAGES
[$ROOT\locale\$LANG]
$LOCVOB\localizations\localization_$L.jar
%endfor

# Some executables go to the bin dir
[$ROOT\bin]
$C1BIN\component1.exe
# or similar
$COMP2VOB\bin\component2.exe
$COMP3VOB\bin\component3.exe

[$ROOT\lib]
# Mentining a directory recursively enumerates files
$COMP1VOB\lib

# Notify user that generation has ended
%print end
```