

---

# **Kallithea Deployment**

*Release 0.0*

**Wolfgang Scherer**

**Aug 24, 2022**

## Quickstart

### Local:

- Deployment documentation: [Kallithea deployment documentation](#)
- Deployment repository: [Kallithea deployment repository](#)
- Installed instance: [Kallithea installed instance](#)

### Official:

- Web-Site: [Kallithea](#)
- Documentation: [Kallithea Documentation](#)
- Repository: <https://kallithea-scm.org/repos/kallithea>
- Bitbucket: [conservancy / kallithea - Bitbucket](#) (as long as it still works)

## **Abstract**

# CONTENTS

<b>List of Figures</b>	<b>ii</b>
<b>List of Tables</b>	<b>iii</b>
<b>List of Code Blocks</b>	<b>iv</b>
<b>1 Installation</b>	<b>1</b>
1.1 Get deployment package . . . . .	1
1.2 System packages . . . . .	1
1.3 Installation / update . . . . .	1
1.4 Initial setup . . . . .	2
1.5 Initialize repository defaults . . . . .	3
1.6 Setup repos/incoming . . . . .	3
1.7 Create clone script . . . . .	4
1.8 Full text search . . . . .	4
1.9 Install a language file for <i>en</i> . . . . .	4
<b>2 Update</b>	<b>5</b>
2.1 Database update . . . . .	5
<b>3 Serve with gearbox</b>	<b>6</b>
<b>4 npm on Debian</b>	<b>7</b>
<b>5 Apache configuration</b>	<b>8</b>
5.1 Apache WSGI . . . . .	8
5.2 Apache as subdirectory . . . . .	9
<b>6 Kallithea deployment</b>	<b>10</b>
6.1 Repositories . . . . .	10
6.2 Documentation . . . . .	10
<b>7 LDAP</b>	<b>11</b>
<b>8 Bug: Logging re-initialized in <code>make_app()</code></b>	<b>12</b>
8.1 Paste Deploy WSGI app configuration . . . . .	12
8.2 Paste Script logging . . . . .	13
8.3 Gearbox . . . . .	15
8.4 Pyramid . . . . .	17
8.5 kallithea-cli . . . . .	19
8.6 Kallithea details . . . . .	19
<b>9 Accept-Language bug</b>	<b>21</b>
9.1 Solution with <code>i18n.notrans</code> patch . . . . .	21
<b>10 Patch needed for Python 2.7.3</b>	<b>24</b>

## LIST OF FIGURES

8.1 loadapp call chain . . . . .	13
----------------------------------	----

## LIST OF TABLES

## LIST OF CODE BLOCKS

## INSTALLATION

See [Installation on Unix/Linux](#)

On Debian, NPM must be installed (see [section 4, npm on Debian](#)).

### 1.1 Get deployment package

```
cd "${HOME}/project"
mkdir -p kallithea
cd kallithea

isy -n
```

Edit `.sync.defs` to reflect correct info:

```
@RUSER sw
@RHOST scherer.wiedenmann.intern
@SCP_REMOTE /home/sw/project/kallithea
```

Get package:

```
./sync.sh --restore
```

### 1.2 System packages

```
sudo apt-get install --yes build-essential git python-pip python-virtualenv libffi-dev python-
↳dev nodejs

test -r /usr/bin/node || sudo ln -s nodejs /usr/bin/node; node --version
test -r /usr/bin/npm || sudo apt-get install --yes npm # Ubuntu 18.04
```

Up to date packages for mercurial and tortoisehg are on [sw-amt.ws](#).

```
apt-get install --yes tortoisehg
```

### 1.3 Installation / update

```
cd "${HOME}/project"
mkdir -p kallithea
cd kallithea

test -d kallithea || \
  hg clone https://kallithea-scm.org/repos/kallithea -u stable

sudo apt install python3-virtualenv
```

(continues on next page)



(continued from previous page)

```
cd "${HOME}/project/kallithea/kallithea"; \
test -d ../kallithea-venv/bin || \
( \
  virtualenv --python=python3 ../kallithea-venv; \
  ../kallithea-venv/bin/pip install --upgrade pip setuptools; \
)
```

Activate environment:

```
cd "${HOME}/project/kallithea/kallithea";
. ../kallithea-venv/bin/activate

sudo apt-get install --yes libsasl2-dev
sudo apt-get install --yes libldap2-dev
pip install pip install python-ldap

hg pull && hg up -C

pip install --upgrade -e .

alembic -c config.ini upgrade head
alembic -c development.ini upgrade head
## or
mkdir -p data
kallithea-cli db-create -c config.ini --user=admin --password=ktBE216 --email=edv@ws-gruppe.
↳de --repos="${HOME}/project/kallithea/repos

python setup.py compile_catalog # for translation of the UI

# `node.js - How can I update NodeJS and NPM to their latest versions? - Stack Overflow
↳<https://stackoverflow.com/questions/6237295/how-can-i-update-nodejs-and-npm-to-their-
↳latest-versions>`

# clean ~/.bashrc afterwards
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.1/install.sh | bash
. ~/.profile.d/nvm
nvm install --lts

kallithea-cli front-end-build

"${HOME}/project/kallithea/kallithea-venv/bin/kallithea-cli index-create -c "${HOME}"/
↳project/kallithea/kallithea/config.ini --full
```

## 1.4 Initial setup

```
cd "${HOME}/project/kallithea/kallithea"

mkdir -p ../log
```

Move existing repositories away until the repository defaults (activated statistics and download) are set:

```
( cd "${HOME}/project/kallithea" || exit 1; \
test ! -d repos || ( mv repos repos-000 ); \
mkdir -p repos )
```

Create new configuration and update it from an existing one:

```
kallithea-cli config-create cfg-HOSTNAME.ini host=0.0.0.0 port=5020
# update configuration with e.g.: diff -u cfg-sw-amt.ini cfg-HOSTNAME.ini (beware the_
↳absolute path names in log handler sections)
# (ediff-files "~/project/kallithea/kallithea/cfg-sw-amt.ini" "~/project/kallithea/kallithea/
↳cfg-HOSTNAME.ini")
```

Generate development and wsgi configurations:

```
../config_admin.sh link HOSTNAME
```

This installs links to created configurations:

Link	Configuration
config.ini	cfg-HOSTNAME.ini
config-dev.ini	cfg-HOSTNAME-dev.ini
config-proxy.ini	cfg-HOSTNAME-proxy.ini
config-wsgi.ini	cfg-HOSTNAME-wsgi.ini

Create new database:

```
kallithea-cli db-create -c config.ini --user=admin --password=ktBE216 --email=edv@ws-gruppe.de --repos="${HOME}/project/kallithea/repos
```

Build frontend:

```
kallithea-cli front-end-build
```

## 1.5 Initialize repository defaults

Either serve with gearbox (see [section 3, Serve with gearbox](#)) or complete Apache configuration (see [section 5, Apache configuration](#)), then login as *admin* and activate downloads and statistics for new repositories und *Admin - Repositorystandards*.

Restore existing repository tree:

```
( cd "${HOME}/project/kallithea || exit 1; \
  test ! -d repos-000 || ( rmdir repos && mv repos-000 repos ) )
```

Clone repositories:

```
cd "${HOME}/project/kallithea
cp ./clone_repos_sw_scherer.sh ../clone_repo.sh
chmod +x ../clone_repo.sh
cd ..
./clone_repo.sh
```

Rescan repositories under *Admin - Settings - Remap and Rescan*, or run:

```
cd "${HOME}/project/kallithea/kallithea && \
kallithea-cli repo-scan --config_file config.ini --remove-missing
```

## 1.6 Setup repos/incoming

Create this as repository group without group access!

```
cd "${HOME}/project/kallithea
mkdir -p repos/incoming
cd repos/incoming
ln -s ../../link_repos.sh .
```



See [Upgrading Kallithea - Kallithea 0.5.0 documentation](#)

## 2.1 Database update

```
. "${HOME}"/project/kallithea/kallithea-venv/bin/activate
cd kallithea
alembic -c development.ini upgrade head
```

## SERVE WITH GEARBOX

```
. "${HOME}"/project/kallithea/kallithea-venv/bin/activate  
gearbox serve -c config-dev.ini
```

## NPM ON DEBIAN

Needs npm on Debian (<https://nodejs.org/fa/download/package-manager/>):

```
wget -qO- https://deb.nodesource.com/setup_6.x | sudo -E bash -  
apt-get install --yes nodejs
```

## APACHE CONFIGURATION

### 5.1 Apache WSGI

```
apt-get install --yes apache2 libapache2-mod-wsgi
```

Enable system locale in `/etc/apache2/envvars` (remove `#` from `./etc/default/locale`).

`/etc/apache2/conf-available/wsgi-kallithea.conf` (replace `@user@` with correct user):

```
# -*- mode: conf; tab-width: 4; -*-
# WSGI script
WSGIDaemonProcess kallithea user=@user@ group=@user@ processes=5 threads=1 \
    python-path=/home/@user@/project/kallithea/kallithea-venv/lib/python2.7/site-packages:/
↪home/@user@/project/kallithea/kallithea-venv/lib/python2.7
#    lang=de_DE.UTF-8

WSGIScriptAlias /kallithea /home/@user@/project/kallithea/kallithea-dispatch.wsgi

# |:sec:| kallithea
<Directory /home/@user@/project/kallithea>
    RewriteEngine on
    RewriteCond %{REQUEST_URI} /kallithea$ [NC]
    RewriteRule ^(\.[^/])$ %{REQUEST_URI}/ [R=301,NC,L]

    # Use only 1 Python sub-interpreter. Multiple sub-interpreters
    # play badly with C extensions.
    WSGIApplicationGroup %{GLOBAL}
    WSGIPassAuthorization On
    WSGIProcessGroup kallithea

    AllowOverride All
    <IfVersion < 2.3 >
    Order allow,deny
    Allow from all
    </IfVersion>
    <IfVersion !< 2.3>
    Require all granted
    </IfVersion>

    # # <IfModule mod_authn_sasl.c>
    # AuthType Basic
    # AuthName "kallithea"
    # AuthBasicProvider sasl
    # # AuthBasicProvider file sasl
    # AuthBasicAuthoritative On
    # AuthSaslPwcheckMethod saslauthd
    # # AuthUserFile /home/@user@/project/kallithea/data/.htpasswd
    # Require valid-user
    # # </IfModule>
</Directory>

# :ide: CMD: restart apache
# . (let* ((fp (buffer-file-name)) (fn (file-name-nondirectory fp))) (shell-command (concat "/"
↪etc/init.d/apache2 restart" ) nil nil))
```

(continues on next page)

(continued from previous page)

```
# :ide: CMD: reload apache
# . (let* ((fp (buffer-file-name)) (fn (file-name-nondirectory fp))) (shell-command (concat "/
↳etc/init.d/apache2 reload" ) nil nil))

# :ide: GOTO: apache2 log
# . (find-file-other-window "/var/log/apache2/")

# :ide: CMD: dired /etc/apache2/conf.d/
# . (let* ((fp (buffer-file-name)) (fn (file-name-nondirectory fp))) (dired-other-window_
↳(concat "/etc/apache2/conf.d/")))
```

```
a2enmod rewrite
a2enconf wsgi-kallithea
service apache2 restart
```

## 5.2 Apache as subdirectory

Apache subdirectory part:

```
SSLProxyEngine on

<Location /kallithea>
  ProxyPass http://localhost:5020/kallithea
  ProxyPassReverse http://localhost:5020/kallithea
  ProxyPassReverseCookieDomain localhost ws24.no-ip.org
</Location>
```

Besides the regular apache setup you will need to add the following line into [app:main] section of your .ini file:

```
filter-with = proxy-prefix
```

Add the following at the end of the .ini file:

```
[filter:proxy-prefix]
use = egg:PasteDeploy#prefix
prefix = /PREFIX
```

then change PREFIX into your chosen prefix



## KALLITHEA DEPLOYMENT

### 6.1 Repositories

Repository of local kallithea deployment instance:

```
cd ~/project/kallithea
mkdir -p repos/public/kallithea-deploy
rm -f repos/public/kallithea-deploy/.hg
ln -s ../../../../../../kallithea/.hg repos/public/kallithea-deploy/
```

Repository of local kallithea instance:

```
cd ~/project/kallithea
mkdir -p repos/public/kallithea
rm -f repos/public/kallithea/.hg
ln -s ../../../../../../kallithea/kallithea/.hg repos/public/kallithea/
```

### 6.2 Documentation

/etc/apache2/conf-available/kallithea-deploy.conf (replace @user@ with correct user):

```
Alias /kallithea/_mnt/kallithea-deploy /home/@user@/project/kallithea/doc/_build

<Directory /home/@user@/project/kallithea/doc/_build>
    Options MultiViews Indexes FollowSymLinks IncludesNoExec
    AllowOverride All
    <IfVersion < 2.3 >
        Order allow,deny
        Allow from all
    </IfVersion>
    <IfVersion !< 2.3>
        Require all granted
    </IfVersion>
</Directory>
```

```
a2enconf kallithea-deploy
service apache2 restart
```

## LDAP

```
sudo apt-get install --yes libsasl2-dev
sudo apt-get install --yes libldap2-dev
pip install python-ldap
```

On page *Admin* -> *Authentication* enable LDAP plugin with the following settings:

LDAP Host	ldap.ws-gruppe.de
Custom LDAP Port	636
Account	uid=sw,ou=Mitarbeiter,dc=ws-gruppe,dc=de
Password	
Connection Security	LDAPS
Certificate Checks	NEVER
Custom CA Certificates	
Base DN	ou=Mitarbeiter,dc=ws-gruppe,dc=de
LDAP Search Filter	
LDAP Search Scope	SUBTREE
Login Attribute	uid
First Name Attribute	givenName
Last Name Attribute	sn
Email Attribute	mail

## BUG: LOGGING RE-INITIALIZED IN MAKE\_APP ()

Paste Deploy provides sufficient support for user-defined run-time configuration values during initialization of a WSGI application. However, logging initialization provided by Paste Script appears to be a mere afterthought, which lacks the necessary support for user-defined configuration values.

The initialization schema is generally

1. Initialize logging (each toolkit rolls their own, but defaults `__file__` and `here` are always set)
2. Load WSGI application (`paste.deploy.loadapp()`)

For Kallithea this can be fixed with `middleware-logging.patch`.

```
hg revert kallithea/bin/kallithea_cli_base.py
hg revert kallithea/config/middleware.py
patch -p1 <../patch/middleware-logging.patch
```

### 8.1 Paste Deploy WSGI app configuration

The design of a web server application based on a configuration file as used in Pylons/Pyramid, TurboGears 2 - and by extensions in Kallithea - originates from Paste, Paste Deploy and Paste Script.

The standard function to create a WSGI application instance is by calling the function `paste.deploy.loadapp()`.

The need for additional run-time configuration values (most prominently the directory of the configuration file as parameter `here`) is recognized and a mechanism to pass on user-defined configuration values is provided as optional argument `global_conf` for `paste.deploy.loadapp()`, although it seems rather strange that the user supplied defaults do not overwrite existing defaults. (I cannot think of any reason, why the programmer's choices should be limited in such a manner).

The (de facto immutable) standard default values are `__file__` and `here`. They are prepared in `paste.deploy.loadwsgi.ConfigLoader`.

```
class ConfigLoader(_Loader):

    def __init__(self, filename):
        self.filename = filename = filename.strip()
        defaults = {
            'here': os.path.dirname(os.path.abspath(filename)),
            '__file__': os.path.abspath(filename)
        }
        self.parser = NicerConfigParser(filename, defaults=defaults)
        self.parser.optionxform = str # Don't lower-case keys
        with open(filename) as f:
            self.parser.read_file(f)

    def update_defaults(self, new_defaults, overwrite=True):
        for key, value in iteritems(new_defaults):
            if not overwrite and key in self.parser._defaults:
                continue
            self.parser._defaults[key] = value
```

User-defined runtime configuration values from `global_conf` are applied in `paste.deploy.loadwsgi._loadconfig()` using `paste.deploy.loadwsgi.ConfigLoader.update_defaults()`.

```
def _loadconfig(object_type, uri, path, name, relative_to, global_conf):
    # ...
    loader = ConfigLoader(path)
    if global_conf:
        loader.update_defaults(global_conf, overwrite=False)
```

The relevant call chain is shown in figure 8.1.

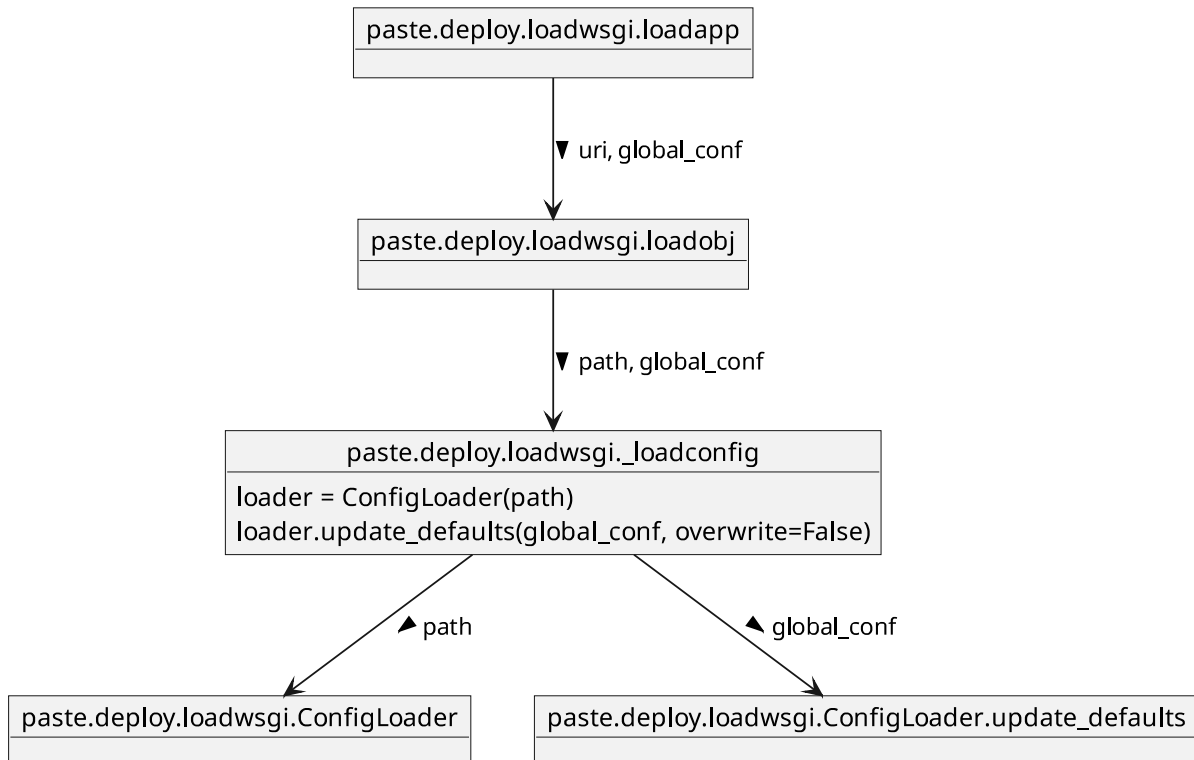


figure 8.1: loadapp call chain

`paste.deploy` does not initialize logging at all which may be the reason for the poor shape of logging initialization.

## 8.2 Paste Script logging

The syntax for running a server with `Paste Script` provides support for specifying run-time configuration defaults:

```
paster serve [options] CONFIG_FILE [start|stop|restart|status] [var=value]
```

`paste.script` initializes logging by calling `logging.config.fileConfig()` with the fixed set of defaults `__file__` and here from method `paste.script.command.Command.logging_file_config()`. The additional run-time configuration defaults are simply ignored.

```
def logging_file_config(self, config_file):
    """
    Setup logging via the logging module's fileConfig function with the
    specified ``config_file``, if applicable.

    ConfigParser defaults are specified for the special ``__file__``
    and ``here`` variables, similar to PasteDeploy config loading.
    """
```

(continues on next page)

(continued from previous page)

```

parser = ConfigParser.ConfigParser()
parser.read([config_file])
if parser.has_section('loggers'):
    config_file = os.path.abspath(config_file)
    fileConfig(config_file, dict(__file__=config_file,
                               here=os.path.dirname(config_file)))

```

```

paste.script.command.Command.logging_file_config (let ((p ("/usr/local/pyramid/lib/python2.
↪7/site-packages/PasteScript-1.7.5-py2.7.egg/paste/script/command.py" 27860 "command.py"
↪51))) (find-file-other-window (car p)) (goto-char (cadr p)))

```

This original implementation is just copied again and again in other frameworks. While missing *loggers* definitions in the configuration file are taken care of, the run-time defaults are never considered.

For Kallithea the example call:

```
paster serve --reload config.ini pid=55 here=not_changed some='thing else'
```

results in duplicate invocations of `logging.config.fileConfig()`.

First call to `logging.config.fileConfig()`:

```

Traceback (most recent call last):
  File "/home/ws/project/kallithea/kallithea-venv/bin/paster", line 8, in <module>
    sys.exit(run())
  File "/home/ws/project/kallithea/kallithea-venv/local/lib/python2.7/site-packages/paste/
↪script/command.py", line 102, in run
    invoke(command, command_name, options, args[1:])
  File "/home/ws/project/kallithea/kallithea-venv/local/lib/python2.7/site-packages/paste/
↪script/command.py", line 141, in invoke
    exit_code = runner.run(args)
  File "/home/ws/project/kallithea/kallithea-venv/local/lib/python2.7/site-packages/paste/
↪script/command.py", line 236, in run
    result = self.command()
  File "/home/ws/project/kallithea/kallithea-venv/local/lib/python2.7/site-packages/paste/
↪script/serve.py", line 278, in command
    self.logging_file_config(log_fn)
  File "/home/ws/project/kallithea/kallithea-venv/local/lib/python2.7/site-packages/paste/
↪script/command.py", line 786, in logging_file_config
    here=os.path.dirname(config_file))
  File "/usr/lib/python2.7/logging/config.py", line 72, in fileConfig
    traceback.print_stack()

fileConfig: fname=/home/ws/project/kallithea/kallithea/config.ini,
           disable_existing_loggers=True,
           defaults={'__file__': '/home/ws/project/kallithea/kallithea/config.ini',
                    'here': '/home/ws/project/kallithea/kallithea'}

```

Second call to `logging.config.fileConfig()`:

```

Traceback (most recent call last):
  File "/home/ws/project/kallithea/kallithea-venv/bin/paster", line 8, in <module>
    sys.exit(run())
  File "/home/ws/project/kallithea/kallithea-venv/local/lib/python2.7/site-packages/paste/
↪script/command.py", line 102, in run
    invoke(command, command_name, options, args[1:])
  File "/home/ws/project/kallithea/kallithea-venv/local/lib/python2.7/site-packages/paste/
↪script/command.py", line 141, in invoke
    exit_code = runner.run(args)
  File "/home/ws/project/kallithea/kallithea-venv/local/lib/python2.7/site-packages/paste/
↪script/command.py", line 236, in run
    result = self.command()
  File "/home/ws/project/kallithea/kallithea-venv/local/lib/python2.7/site-packages/paste/
↪script/serve.py", line 284, in command
    relative_to=base, global_conf=vars)
  File "/home/ws/project/kallithea/kallithea-venv/local/lib/python2.7/site-packages/paste/
↪script/serve.py", line 329, in loadapp
    **kw)
  File "/home/ws/project/kallithea/kallithea-venv/local/lib/python2.7/site-packages/paste/
↪deploy/loadwsgi.py", line 253, in loadapp

```

(continues on next page)

(continued from previous page)

```

    return loadobj(APP, uri, name=name, **kw)
File "/home/ws/project/kallithea/kallithea-venv/local/lib/python2.7/site-packages/paste/
↳deploy/loadwsgi.py", line 278, in loadobj
    return context.create()
File "/home/ws/project/kallithea/kallithea-venv/local/lib/python2.7/site-packages/paste/
↳deploy/loadwsgi.py", line 715, in create
    return self.object_type.invoke(self)
File "/home/ws/project/kallithea/kallithea-venv/local/lib/python2.7/site-packages/paste/
↳deploy/loadwsgi.py", line 152, in invoke
    return fix_call(context.object, context.global_conf, **context.local_conf)
File "/home/ws/project/kallithea/kallithea-venv/local/lib/python2.7/site-packages/paste/
↳deploy/util.py", line 55, in fix_call
    val = callable(*args, **kw)
File "/home/ws/project/kallithea/kallithea/config/middleware.py", line 59, in_
↳make_app
    logging.config.fileConfig(global_conf['__file__'])
File "/usr/lib/python2.7/logging/config.py", line 72, in fileConfig
    traceback.print_stack()

fileConfig: fname=/home/ws/project/kallithea/kallithea/config.ini,
           disable_existing_loggers=True,
           defaults=None

```

## 8.3 Gearbox

Gearbox also uses `paste.deploy.loadapp()`, passing on option definitions from the command line as runtime configuration defaults:

```
gearbox serve [OPTIONS] [args [args ...]]
```

Logging is initialized with function `gearbox.utils.log.setup_logging()`. There are no provisions for merging a global configuration, but fixed defaults for `__file__` and here are prepared the same way as method `paste.script.command.Command.logging_file_config()` does.

```

def setup_logging(config_uri, fileConfig=fileConfig,
                 configparser=configparser):
    """
    Set up logging via the logging module's fileConfig function with the
    filename specified via ``config_uri`` (a string in the form
    ``filename#sectionname``).

    ConfigParser defaults are specified for the special ``__file__``
    and ``here`` variables, similar to PasteDeploy config loading.
    """
    path, _ = _getpathsec(config_uri, None)
    parser = configparser.ConfigParser()
    parser.read([path])
    if parser.has_section('loggers'):
        config_file = os.path.abspath(path)
        config_options = dict(
            __file__=config_file,
            here=os.path.dirname(config_file)
        )

        fileConfig(config_file, config_options,
                   disable_existing_loggers=False)

```

```

gearbox.utils.log.setup_logging (let ((p ("/home/ws/project/kallithea/kallithea-venv/lib/
↳python2.7/site-packages/gearbox/utils/log.py" 180))) (find-file-other-window (car p)) (goto-
↳char (cadr p)))

```

For Kallithea the example call

```
gearbox serve -c config.ini pid=55 here=not_changed some='thing else'
```

results in duplicate invocations of `logging.config.fileConfig()`.

First call to `logging.config.fileConfig()`:

```
Traceback (most recent call last):
  File "/home/ws/project/kallithea/kallithea-venv/bin/gearbox", line 8, in <module>
    sys.exit(main())
  File "/home/ws/project/kallithea/kallithea-venv/local/lib/python2.7/site-packages/gearbox/
↪main.py", line 229, in main
    return gearbox.run(args)
  File "/home/ws/project/kallithea/kallithea-venv/local/lib/python2.7/site-packages/gearbox/
↪main.py", line 150, in run
    return self._run_subcommand(remainder)
  File "/home/ws/project/kallithea/kallithea-venv/local/lib/python2.7/site-packages/gearbox/
↪main.py", line 172, in _run_subcommand
    return cmd.run(parsed_args)
  File "/home/ws/project/kallithea/kallithea-venv/local/lib/python2.7/site-packages/gearbox/
↪command.py", line 31, in run
    self.take_action(parsed_args)
  File "/home/ws/project/kallithea/kallithea-venv/local/lib/python2.7/site-packages/gearbox/
↪commands/serve.py", line 276, in take_action
    setup_logging(log_fn)
  File "/home/ws/project/kallithea/kallithea-venv/local/lib/python2.7/site-packages/gearbox/
↪utils/log.py", line 32, in setup_logging
    disable_existing_loggers=False)
  File "/usr/lib/python2.7/logging/config.py", line 72, in fileConfig
    traceback.print_stack()

fileConfig: fname=/home/ws/project/kallithea/kallithea/config-dev.ini,
           disable_existing_loggers=False,
           defaults={'__file__': '/home/ws/project/kallithea/kallithea/config-dev.ini',
                    'here': '/home/ws/project/kallithea/kallithea'}
```

Second call to `logging.config.fileConfig()`:

```
Traceback (most recent call last):
  File "/home/ws/project/kallithea/kallithea-venv/bin/gearbox", line 8, in <module>
    sys.exit(main())
  File "/home/ws/project/kallithea/kallithea-venv/local/lib/python2.7/site-packages/gearbox/
↪main.py", line 229, in main
    return gearbox.run(args)
  File "/home/ws/project/kallithea/kallithea-venv/local/lib/python2.7/site-packages/gearbox/
↪main.py", line 150, in run
    return self._run_subcommand(remainder)
  File "/home/ws/project/kallithea/kallithea-venv/local/lib/python2.7/site-packages/gearbox/
↪main.py", line 172, in _run_subcommand
    return cmd.run(parsed_args)
  File "/home/ws/project/kallithea/kallithea-venv/local/lib/python2.7/site-packages/gearbox/
↪command.py", line 31, in run
    self.take_action(parsed_args)
  File "/home/ws/project/kallithea/kallithea-venv/local/lib/python2.7/site-packages/gearbox/
↪commands/serve.py", line 280, in take_action
    relative_to=base, global_conf=parsed_vars)
  File "/home/ws/project/kallithea/kallithea-venv/local/lib/python2.7/site-packages/gearbox/
↪commands/serve.py", line 311, in loadapp
    return loadapp(app_spec, name=name, relative_to=relative_to, **kw)
  File "/home/ws/project/kallithea/kallithea-venv/local/lib/python2.7/site-packages/paste/
↪deploy/loadwsgi.py", line 253, in loadapp
    return loadobj(APP, uri, name=name, **kw)
  File "/home/ws/project/kallithea/kallithea-venv/local/lib/python2.7/site-packages/paste/
↪deploy/loadwsgi.py", line 278, in loadobj
    return context.create()
  File "/home/ws/project/kallithea/kallithea-venv/local/lib/python2.7/site-packages/paste/
↪deploy/loadwsgi.py", line 715, in create
    return self.object_type.invoke(self)
  File "/home/ws/project/kallithea/kallithea-venv/local/lib/python2.7/site-packages/paste/
↪deploy/loadwsgi.py", line 152, in invoke
    return fix_call(context.object, context.global_conf, **context.local_conf)
  File "/home/ws/project/kallithea/kallithea-venv/local/lib/python2.7/site-packages/paste/
↪deploy/util.py", line 55, in fix_call
    val = callable(*args, **kw)
  File "/home/ws/project/kallithea/kallithea/kallithea/config/middleware.py", line 59, in_
↪make_app
```

(continues on next page)

(continued from previous page)

```

logging.config.fileConfig(global_conf['__file__'])
File "/usr/lib/python2.7/logging/config.py", line 72, in fileConfig
    traceback.print_stack()

fileConfig: fname=/home/ws/project/kallithea/kallithea/config-dev.ini,
           disable_existing_loggers=True,
           defaults=None

```

```

gearbox.utils.setup_logging (let ((p '("/home/ws/project/kallithea/kallithea-venv/lib/
↳python2.7/site-packages/gearbox/utils/log.py" 913))) (find-file-other-window (car p)) (goto-
↳char (cadr p)))
kallithea.config.middleware.make_app (let ((p '("/home/ws/project/kallithea/kallithea/
↳kallithea/config/middleware.py" 2264))) (find-file-other-window (car p)) (goto-char (cadr_
↳p)))

```

## 8.4 Pyramid

Pyramid provides command `pserve` to serve a WSGI application loaded with `Paste Deploy`. `pserve` also provides support for specifying run-time configuration defaults:

```
pserve [options] [config_uri] [config_vars [config_vars ...]]
```

Logging is initialized with `pyramid.paster.setup_logging()`. There are no provisions for merging a global configuration, but fixed defaults for `__file__` and here are prepared the same way as method `paste.script.command.Command.logging_file_config()` does.

```

def setup_logging(config_uri, fileConfig=fileConfig,
                  configparser=configparser):
    """
    Set up logging via the logging module's fileConfig function with the
    filename specified via ``config_uri`` (a string in the form
    ``filename#sectionname``).

    ConfigParser defaults are specified for the special ``__file__``
    and ``here`` variables, similar to PasteDeploy config loading.
    """
    path, _ = _getpathsec(config_uri, None)
    parser = configparser.ConfigParser()
    parser.read([path])
    if parser.has_section('loggers'):
        config_file = os.path.abspath(path)
        return fileConfig(
            config_file,
            dict(__file__=config_file, here=os.path.dirname(config_file))
        )

```

```

pyramid.paster.setup_logging (let ((p '("/usr/local/pyramid/lib/python2.
↳7/site-packages/pyramid-1.5-py2.7.egg/pyramid/paster.py" 1858 "paster.py" 53))) (find-file-
↳other-window (car p)) (goto-char (cadr p)))

```

For Kallithea the example call

```
pserve config.ini pid=55 here=not_changed some='thing else'
```

results in duplicate invocations of `logging.config.fileConfig()`.

First call to `logging.config.fileConfig()`:

```

Traceback (most recent call last):
  File "<string>", line 1, in <module>
  File "/home/ws/project/kallithea/kallithea-venv/local/lib/python2.7/site-packages/hupper/
↳ipc.py", line 320, in spawn_main
    func(**kwargs)

```

(continues on next page)



(continued from previous page)

```

File "/home/ws/project/kallithea/kallithea-venv/local/lib/python2.7/site-packages/hupper/
↪worker.py", line 265, in worker_main
    func(*spec_args, **spec_kwargs)
File "/home/ws/project/kallithea/kallithea-venv/local/lib/python2.7/site-packages/pyramid/
↪scripts/pserve.py", line 34, in main
    return command.run()
File "/home/ws/project/kallithea/kallithea-venv/local/lib/python2.7/site-packages/pyramid/
↪scripts/pserve.py", line 198, in run
    loader.setup_logging(config_vars)
File "/home/ws/project/kallithea/kallithea-venv/local/lib/python2.7/site-packages/plaster_
↪pastedeploy/__init__.py", line 223, in setup_logging
    fileConfig(self.uri.path, defaults, disable_existing_loggers=False)
File "/usr/lib/python2.7/logging/config.py", line 72, in fileConfig
    traceback.print_stack()

fileConfig: fname=config.ini,
            disable_existing_loggers=False,
            defaults={'pid': '55',
                      '__file__': '/home/ws/project/kallithea/kallithea/config.ini',
                      'some': 'thing else',
                      'here': 'not_changed'}

```

Second call to logging.config.fileConfig():

```

Traceback (most recent call last):
File "/home/ws/project/kallithea/kallithea-venv/local/lib/python2.7/site-packages/hupper/
↪ipc.py", line 320, in spawn_main
    func(**kwargs)
File "/home/ws/project/kallithea/kallithea-venv/local/lib/python2.7/site-packages/hupper/
↪worker.py", line 265, in worker_main
    func(*spec_args, **spec_kwargs)
File "/home/ws/project/kallithea/kallithea-venv/local/lib/python2.7/site-packages/pyramid/
↪scripts/pserve.py", line 34, in main
    return command.run()
File "/home/ws/project/kallithea/kallithea-venv/local/lib/python2.7/site-packages/pyramid/
↪scripts/pserve.py", line 275, in run
    app = loader.get_wsgi_app(app_name, config_vars)
File "/home/ws/project/kallithea/kallithea-venv/local/lib/python2.7/site-packages/plaster_
↪pastedeploy/__init__.py", line 129, in get_wsgi_app
    global_conf=defaults,
File "/home/ws/project/kallithea/kallithea-venv/local/lib/python2.7/site-packages/paste/
↪deploy/loadwsgi.py", line 253, in loadapp
    return loadobj(APP, uri, name=name, **kw)
File "/home/ws/project/kallithea/kallithea-venv/local/lib/python2.7/site-packages/paste/
↪deploy/loadwsgi.py", line 278, in loadobj
    return context.create()
File "/home/ws/project/kallithea/kallithea-venv/local/lib/python2.7/site-packages/paste/
↪deploy/loadwsgi.py", line 715, in create
    return self.object_type.invoke(self)
File "/home/ws/project/kallithea/kallithea-venv/local/lib/python2.7/site-packages/paste/
↪deploy/loadwsgi.py", line 152, in invoke
    return fix_call(context.object, context.global_conf, **context.local_conf)
File "/home/ws/project/kallithea/kallithea-venv/local/lib/python2.7/site-packages/paste/
↪deploy/util.py", line 55, in fix_call
    val = callable(*args, **kw)
File "/home/ws/project/kallithea/kallithea/kallithea/config/middleware.py", line 59, in_
↪make_app
    logging.config.fileConfig(global_conf['__file__'])
File "/usr/lib/python2.7/logging/config.py", line 72, in fileConfig
    traceback.print_stack()

fileConfig: fname=/home/ws/project/kallithea/kallithea/config.ini,
            disable_existing_loggers=True,
            defaults=None

```

Support for WSGI applications is provided with functions `pyramid.paster.setup_logging()` and `pyramid.paster.get_app()`.

Function `func:pyramid.paster.get_app` has a parameter `options`, which is passed on to `paste.deploy.loadapp()` as `global_conf-`

```
get_app(options=dict()) -> loadapp(global_conf=options)
```

```
pyramid.paster.get_app                                (let ((p '("/usr/local/pyramid/lib/python2.
↳7/site-packages/pyramid-1.5-py2.7.egg/pyramid/paster.py" 191 "paster.py" 51))) (find-file-
↳other-window (car p)) (goto-char (cadr p)))
paste.deploy.loadwsgi.loadapp                         (let ((p '("/usr/local/pyramid/lib/python2.
↳7/site-packages/PasteDeploy-1.5.2-py2.7.egg/paste/deploy/loadwsgi.py" 7673))) (find-file-
↳other-window (car p)) (goto-char (cadr p)))
```

## 8.5 kallithea-cli

**kallithea-cli** does not seem to have provisions for specifying run-time configuration defaults.

Logging is initialized in `kallithea.bin.kallithea_cli_base.runtime_wrapper()`, but the standard defaults `__file__` and here are missing. It seems, that using `gearbox.utils.log.setup_logging()` would be the better choice, but there is some incompatible section mangling magic going on, so providing explicit defaults once again once more is the obvious solution.

For **Kallithea** an example call shows, that there is only one invocation of `logging.config.fileConfig()`, because the application is not loaded with `paste.deploy.loadapp()` but instantiated directly with `kallithea.config.middleware.make_app_without_logging()`.

```
Traceback (most recent call last):
  File "/home/ws/project/kallithea/kallithea-venv/bin/kallithea-cli", line 11, in <module>
    load_entry_point('Kallithea', 'console_scripts', 'kallithea-cli')()
  File "/home/ws/project/kallithea/kallithea-venv/local/lib/python2.7/site-packages/click/
↳core.py", line 764, in __call__
    return self.main(*args, **kwargs)
  File "/home/ws/project/kallithea/kallithea-venv/local/lib/python2.7/site-packages/click/
↳core.py", line 717, in main
    rv = self.invoke(ctx)
  File "/home/ws/project/kallithea/kallithea-venv/local/lib/python2.7/site-packages/click/
↳core.py", line 1137, in invoke
    return _process_result(sub_ctx.command.invoke(sub_ctx))
  File "/home/ws/project/kallithea/kallithea-venv/local/lib/python2.7/site-packages/click/
↳core.py", line 956, in invoke
    return ctx.invoke(self.callback, **ctx.params)
  File "/home/ws/project/kallithea/kallithea-venv/local/lib/python2.7/site-packages/click/
↳core.py", line 555, in invoke
    return callback(*args, **kwargs)
  File "/home/ws/project/kallithea/kallithea/bin/kallithea_cli_base.py", line 75,
↳in runtime_wrapper
    logging.config.fileConfig(cStringIO.StringIO(config_bytes))
  File "/usr/lib/python2.7/logging/config.py", line 72, in fileConfig
    traceback.print_stack()

fileConfig: fname=<cStringIO.StringIO object at 0x7f98dfb44c68>,
           disable_existing_loggers=True,
           defaults=None
```

```
kallithea.bin.kallithea_cli_base.register_command (let ((p '("/home/ws/project/kallithea/
↳kallithea/bin/kallithea_cli_base.py" 3188))) (find-file-other-window (car p))
↳(goto-char (cadr p)))
```

## 8.6 Kallithea details

The **Kallithea** documentation shows WSGI dispatch scripts for Apache with `mod_wsgi`. In these examples, logging is explicitly initialized in the dispatch scripts (although the defaults for `__file__` and here are missing).

However, `make_app()` in `kallithea/config/middleware.py` unconditionally initializes logging again (also without any defaults, so not even the substitution `%(here)s` is defined).

In a multi-process WSGI environment (as recommended in the documentation) specifying a single log file leads to processes arbitrarily overwriting messages from other processes. Besides several more complex solutions (see [logging - How should I log while using multiprocessing in Python? - Stack Overflow](#), [Logging Cookbook - Python 3.8.1 documentation](#)) an obvious simple solution is to use a separate log file for each process, differentiated by the process ID, e.g.:

```
[handler_session_log]
class = FileHandler
args = (r'%(here)s/../../log/kallithea-session-%(pid)s.log', 'w')
```

This could be achieved by initializing logging in the WSGI dispatch script:

```
from logging.config import fileConfig
fileConfig(
    INIFILE,
    dict(__file__=INIFILE, here=os.path.dirname(INIFILE), pid=os.getpid())
)
```

However, it fails, when logging is re-initialized without proper defaults by `make_app()`.

The duplicate initialization can be fixed by removing it from `make_app()` entirely, since it is redundant in all cases as the previous analysis shows.

### 8.6.1 Tracebacks for duplicate logging initialization

First call to `logging.config.fileConfig()`:

```
Traceback (most recent call last):
  File "/home/ws/project/kallithea/kallithea-dispatch.wsgi", line 356, in <module>
    dict(__file__=INIFILE, here=os.path.dirname(INIFILE), pid=os.getpid())
  File "/usr/lib/python2.7/logging/config.py", line 72, in fileConfig
    traceback.print_stack()

fileConfig: fname=/home/ws/project/kallithea/kallithea/config-wsgi.ini,
           disable_existing_loggers=True,
           defaults={'__file__': '/home/ws/project/kallithea/kallithea/config-wsgi.ini',
                    'pid': 22263,
                    'here': '/home/ws/project/kallithea/kallithea'}
```

Second call to `logging.config.fileConfig()`:

```
Traceback (most recent call last):
  File "/home/ws/project/kallithea/kallithea-dispatch.wsgi", line 363, in <module>
    application = loadapp('config:' + INIFILE)
  File "/home/ws/project/kallithea/kallithea-venv/lib/python2.7/site-packages/paste/deploy/
↪loadwsgi.py", line 253, in loadapp
    ↪return loadobj(APP, uri, name=name, **kw)
  File "/home/ws/project/kallithea/kallithea-venv/lib/python2.7/site-packages/paste/deploy/
↪loadwsgi.py", line 278, in loadobj
    ↪return context.create()
  File "/home/ws/project/kallithea/kallithea-venv/lib/python2.7/site-packages/paste/deploy/
↪loadwsgi.py", line 715, in create
    ↪return self.object_type.invoke(self)
  File "/home/ws/project/kallithea/kallithea-venv/lib/python2.7/site-packages/paste/deploy/
↪loadwsgi.py", line 152, in invoke
    ↪return fix_call(context.object, context.global_conf, **context.local_conf)
  File "/home/ws/project/kallithea/kallithea-venv/lib/python2.7/site-packages/paste/deploy/
↪util.py", line 55, in fix_call
    val = callable(*args, **kw)
  File "/home/ws/project/kallithea/kallithea/config/middleware.py", line 59, in_
↪make_app
    logging.config.fileConfig(global_conf['__file__'])
  File "/usr/lib/python2.7/logging/config.py", line 72, in fileConfig
    traceback.print_stack()

fileConfig: fname=/home/ws/project/kallithea/kallithea/config-wsgi.ini,
           disable_existing_loggers=True,
           defaults=None
```

## ACCEPT-LANGUAGE BUG

If there is no language file *en* installed, TurboGears2 does not handle Accept-Language correctly, when *en* is the prioritized language, e.g.:

```
Accept-Language: en-US, en;q=0.8, de-DE;q=0.5, de;q=0.3
```

The *C* locale is not supported correctly.

The preferred solution is described in [section 1.9, Install a language file for \*en\*](#).

### 9.1 Solution with *i18n.notrans* patch

This solution is no longer recommended.

The setting *i18n.notrans* can be activated with `tg2-i18n-notrans.patch`:

```
cd "${HOME}"/project/kallithea/kallithea-venv/lib/python*/site-packages/tg/
patch -p2 <"${HOME}"/project/kallithea/patch/tg2-i18n-notrans.patch

cd "${HOME}"/project/kallithea/kallithea-venv/lib/python*/site-packages/tg/
patch -R -p2 <"${HOME}"/project/kallithea/patch/tg2-i18n-notrans.patch
```

#### 9.1.1 Environment

INI settings:

```
## Internationalization (see setup documentation for details)
## By default, the language requested by the browser is used if available.
#i18n.enabled = false
## Fallback language, empty for English (valid values are the names of subdirectories in
↳kallithea/i18n):
i18n.lang =
```

#### 9.1.2 Request with Prio English, German

If there is no language file for *en* installed, the language file *de* is used, which is an **error**.

##### Message file *en* not installed

Since language file *en* cannot be found, the message file *de* is used, which has higher priority than the fallback. **This is an error.**

```
Accept-Language: en-US, en;q=0.8, de-DE;q=0.5, de;q=0.3
```

```
[tg.request_local] Request.languages_best_match: en-US, en;q=0.8, de-DE;q=0.5, de;q=0.3
[tg.request_local] Request.languages_best_match: ['en-US', 'en', 'de-DE', 'de']
[tg.i18n] _parse_locale: lang, territory, script, variant: ('en', 'US', None, None)
[tg.i18n] _parse_locale: lang, territory, script, variant: ('en', None, None, None)
[tg.i18n] _parse_locale: lang, territory, script, variant: ('de', 'DE', None, None)
[tg.i18n] _parse_locale: lang, territory, script, variant: ('de', None, None, None)
[tg.request_local] Request.languages_best_match: en-US, en;q=0.8, de-DE;q=0.5, de;q=0.3
[tg.request_local] Request.languages_best_match: ['en-US', 'en', 'de-DE', 'de']
[tg.i18n] ugettext: tg.translator.info(): content-transfer-encoding: 8bit
[tg.i18n] ugettext: tg.translator.info(): content-type: text/plain; charset=utf-8
[tg.i18n] ugettext: tg.translator.info(): generated-by: Babel 2.7.0
[tg.i18n] ugettext: tg.translator.info(): language: de
[tg.i18n] ugettext: tg.translator.info(): language-team: de <LL@li.org>
[tg.i18n] ugettext: tg.translator.info(): last-translator: FULL NAME <EMAIL@ADDRESS>
[tg.i18n] ugettext: tg.translator.info(): mime-version: 1.0
[tg.i18n] ugettext: tg.translator.info(): plural-forms: nplurals=2; plural=n != 1
[tg.i18n] ugettext: tg.translator.info(): po-revision-date: YEAR-MO-DA HO:MI+ZONE
[tg.i18n] ugettext: tg.translator.info(): pot-creation-date: 2019-11-30 22:58+0100
[tg.i18n] ugettext: tg.translator.info(): project-id-version: PROJECT VERSION
[tg.i18n] ugettext: tg.translator.info(): report-msgid-bugs-to: translations@kallithea-
↳scm.org
```

## Message file *en* installed

Since language file *en* is found, it is used, which is the correct behavior.

```
Accept-Language: en-US, en;q=0.8, de-DE;q=0.5, de;q=0.3
```

```
[tg.request_local] Request.languages_best_match: en-US, en;q=0.8, de-DE;q=0.5, de;q=0.3
[tg.request_local] Request.languages_best_match: ['en-US', 'en', 'de-DE', 'de']
[tg.i18n] _parse_locale: lang, territory, script, variant: ('en', 'US', None, None)
[tg.i18n] _parse_locale: lang, territory, script, variant: ('en', None, None, None)
[tg.i18n] _parse_locale: lang, territory, script, variant: ('de', 'DE', None, None)
[tg.i18n] _parse_locale: lang, territory, script, variant: ('de', None, None, None)
[tg.request_local] Request.languages_best_match: en-US, en;q=0.8, de-DE;q=0.5, de;q=0.3
[tg.request_local] Request.languages_best_match: ['en-US', 'en', 'de-DE', 'de']
[tg.i18n] ugettext: tg.translator.info(): content-transfer-encoding: 8bit
[tg.i18n] ugettext: tg.translator.info(): content-type: text/plain; charset=utf-8
[tg.i18n] ugettext: tg.translator.info(): generated-by: Babel 1.3
[tg.i18n] ugettext: tg.translator.info(): language-team: en <LL@li.org>
[tg.i18n] ugettext: tg.translator.info(): last-translator: Wolfgang Scherer
↳<wolfgang.scherer@gmx.de>
[tg.i18n] ugettext: tg.translator.info(): mime-version: 1.0
[tg.i18n] ugettext: tg.translator.info(): plural-forms: nplurals=2; plural=(n !=_
↳1)
[tg.i18n] ugettext: tg.translator.info(): po-revision-date: 2019-12-01 16:57+0100
[tg.i18n] ugettext: tg.translator.info(): pot-creation-date: 2019-12-01 16:44+0100
[tg.i18n] ugettext: tg.translator.info(): project-id-version: Kallithea 0.5.0
[tg.i18n] ugettext: tg.translator.info(): report-msgid-bugs-to: translations@kallithea-
↳scm.org
```

### 9.1.3 Request with Prio German, English

There is no difference, whether language file *en* is installed or not.

Language file *en* is **not** installed, language file *de* is found.

```
Accept-Language: de-DE, de;q=0.8, en-US;q=0.5, en;q=0.3
```

```
[tg.request_local] Request.languages_best_match: de-DE, de;q=0.8, en-US;q=0.5, en;q=0.3
[tg.request_local] Request.languages_best_match: ['de-DE', 'de', 'en-US', 'en']
[tg.i18n] _parse_locale: lang, territory, script, variant: ('de', 'DE', None, None)
[tg.i18n] _parse_locale: lang, territory, script, variant: ('de', None, None, None)
[tg.i18n] _parse_locale: lang, territory, script, variant: ('en', 'US', None, None)
[tg.i18n] _parse_locale: lang, territory, script, variant: ('en', None, None, None)
```

(continues on next page)

(continued from previous page)

```

[tg.request_local] Request.languages_best_match: de-DE, de;q=0.8, en-US;q=0.5, en;q=0.3
[tg.request_local] Request.languages_best_match: ['de-DE', 'de', 'en-US', 'en']
[tg.i18n] ugettext: tg.translator.info(): content-transfer-encoding: 8bit
[tg.i18n] ugettext: tg.translator.info(): content-type: text/plain; charset=utf-8
[tg.i18n] ugettext: tg.translator.info(): generated-by: Babel 2.7.0
[tg.i18n] ugettext: tg.translator.info(): language: de
[tg.i18n] ugettext: tg.translator.info(): language-team: de <LL@li.org>
[tg.i18n] ugettext: tg.translator.info(): last-translator: FULL NAME <EMAIL@ADDRESS>
[tg.i18n] ugettext: tg.translator.info(): mime-version: 1.0
[tg.i18n] ugettext: tg.translator.info(): plural-forms: nplurals=2; plural=n != 1
[tg.i18n] ugettext: tg.translator.info(): po-revision-date: YEAR-MO-DA HO:MI+ZONE
[tg.i18n] ugettext: tg.translator.info(): pot-creation-date: 2019-11-30 22:58+0100
[tg.i18n] ugettext: tg.translator.info(): project-id-version: PROJECT VERSION
[tg.i18n] ugettext: tg.translator.info(): report-msgid-bugs-to: translations@kallithea-
↳scm.org

```

Language file *en* is installed, language file *de* is found.

```
Accept-Language: de-DE, de;q=0.8, en-US;q=0.5, en;q=0.3
```

```

[tg.request_local] Request.languages_best_match: de-DE, de;q=0.8, en-US;q=0.5, en;q=0.3
[tg.request_local] Request.languages_best_match: ['de-DE', 'de', 'en-US', 'en']
[tg.i18n] _parse_locale: lang, territory, script, variant: ('de', 'DE', None, None)
[tg.i18n] _parse_locale: lang, territory, script, variant: ('de', None, None, None)
[tg.i18n] _parse_locale: lang, territory, script, variant: ('en', 'US', None, None)
[tg.i18n] _parse_locale: lang, territory, script, variant: ('en', None, None, None)
[tg.request_local] Request.languages_best_match: de-DE, de;q=0.8, en-US;q=0.5, en;q=0.3
[tg.request_local] Request.languages_best_match: ['de-DE', 'de', 'en-US', 'en']
[tg.i18n] ugettext: tg.translator.info(): content-transfer-encoding: 8bit
[tg.i18n] ugettext: tg.translator.info(): content-type: text/plain; charset=utf-8
[tg.i18n] ugettext: tg.translator.info(): generated-by: Babel 2.7.0
[tg.i18n] ugettext: tg.translator.info(): language: de
[tg.i18n] ugettext: tg.translator.info(): language-team: de <LL@li.org>
[tg.i18n] ugettext: tg.translator.info(): last-translator: FULL NAME <EMAIL@ADDRESS>
[tg.i18n] ugettext: tg.translator.info(): mime-version: 1.0
[tg.i18n] ugettext: tg.translator.info(): plural-forms: nplurals=2; plural=n != 1
[tg.i18n] ugettext: tg.translator.info(): po-revision-date: YEAR-MO-DA HO:MI+ZONE
[tg.i18n] ugettext: tg.translator.info(): pot-creation-date: 2019-11-30 22:58+0100
[tg.i18n] ugettext: tg.translator.info(): project-id-version: PROJECT VERSION
[tg.i18n] ugettext: tg.translator.info(): report-msgid-bugs-to: translations@kallithea-
↳scm.org

```

**PATCH NEEDED FOR PYTHON 2.7.3**

mercurial-4.9.1

```
diff -ua kallithea-venv/lib/python2.7/site-packages/mercurial/revlog.py-000 kallithea-venv/
↳lib/python2.7/site-packages/mercurial/revlog.py
--- kallithea-venv/lib/python2.7/site-packages/mercurial/revlog.py-000      2019-05-11_
↳06:15:07.113000000 +0200
+++ kallithea-venv/lib/python2.7/site-packages/mercurial/revlog.py  2019-05-11 06:14:31.
↳033000000 +0200
@@ -1682,6 +1682,7 @@
     if rawtext is None:
         rawtext = bytes(bins[0])
         bins = bins[1:]
+    bins = [bytes(_b) for _b in bins]

    rawtext = mdiff.patches(rawtext, bins)
    self._revisioncache = (node, rev, rawtext)
```