ENFORCING FORMATTING RULES ON COMMIT

Philipp Salvisberg
28th October 2021

trivadis
Part of Accenture

PHILIPP SALVISBERG

SENIOR PRINCIPAL CONSULTANT

- Database centric development
- Model Driven Software Development
- Author of free SQL Developer Extensions PL/SQL Unwrapper, db* CODECOP, utPLSQL, plscope-utils, oddgen and Bitemp Remodeler

FORMATTER

4 SQL DEVELOPER

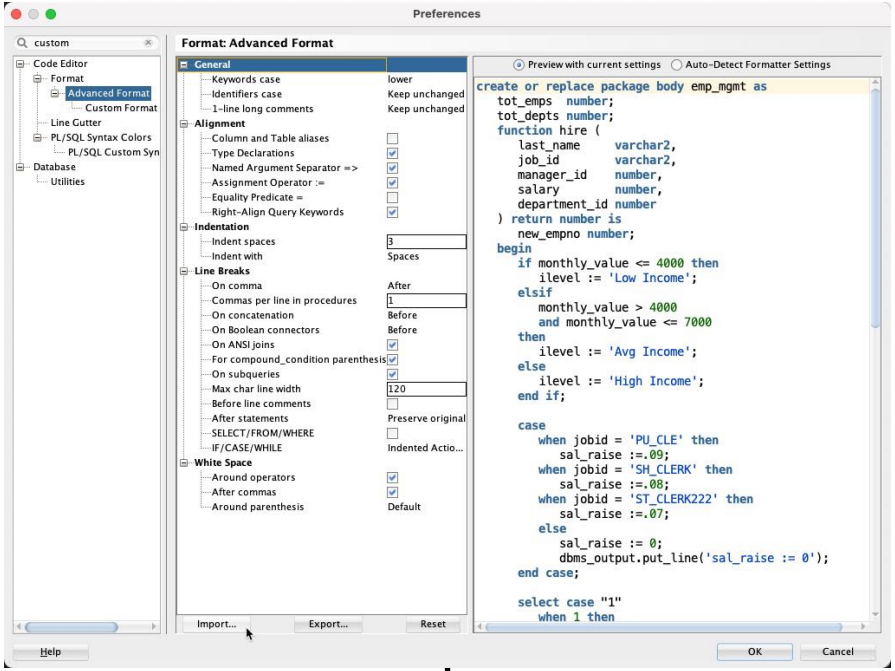
```
 1 select
 2 ename
 3 ,
 4 sal
 5 from
 6 emp
 7 where
 8 deptno
 9 =
10 20
11 ;
```

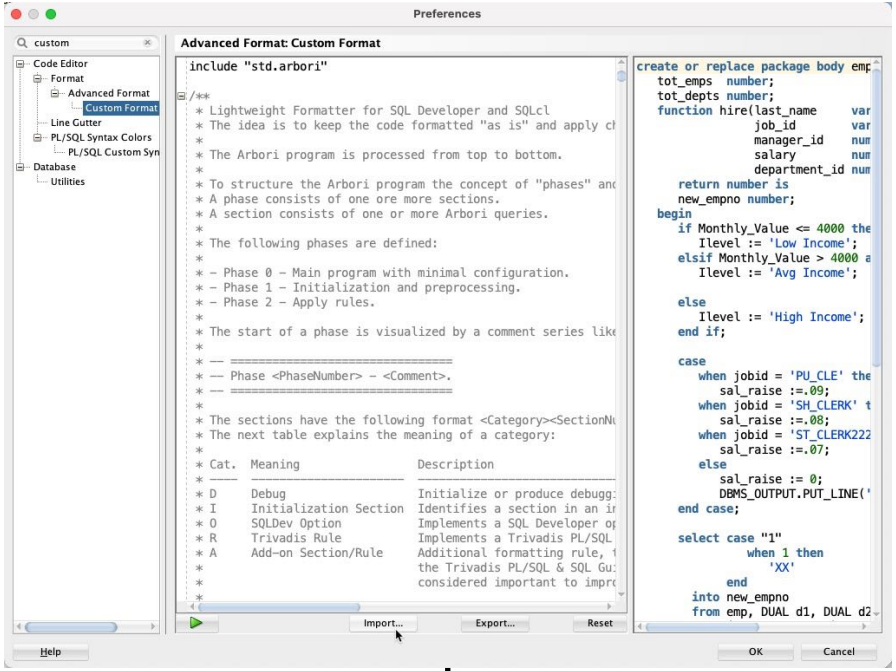

```
1 SELECT
2     ename,
3     sal
4 FROM
5     emp
6 WHERE
7     deptno = 20;
```

```
1 select ename,
2        sal
3   from emp
4  where deptno = 20;
```

```
1 select ename
2       ,sal
3   from emp
4  where deptno = 20;
```

5 FORMATTER SETTINGS





.xml

.arbori

Source: https://github.com/Trivadis/plsql-formatter-settings

trivadis
Part of Accenture

6 SQLcl – BUILT-IN COMMAND

```
FORMAT
---------

FORMAT BUFFER - formats the script in the SQLcl Buffer
FORMAT RULES <filename> - Loads formatter preferences file from SQL Developer export.
FORMAT FILE <input_file> <output_file>
```

7 SQLcl – CUSTOM SCRIPT

```
usage: script format.js <rootPath> [options]

mandatory argument: (one of the following)
  <rootPath>       file or path to directory containing files to format (content will be replaced!)
  <config.json>    configuration file in JSON format (must end with .json)
  *                use * to format the SQLcl buffer

options:
  --register, -r   register SQLcl command tvdformat, without processing, no <rootPath> required
  ext=<ext>        comma separated list of file extensions to process, e.g. ext=sql,pks,pkb
  mext=<ext>       comma separated list of markdown file extensions to process, e.g. ext=md,mdown
  xml=<file>       path to the file containing the xml file for advanced format settings
                   xml=default uses default advanced settings included in sqlcl
                   xml=embedded uses advanced settings defined in format.js
  arbori=<file>    path to the file containing the Arbori program for custom format settings
                   arbori=default uses default Arbori program included in sqlcl
```

More: https://github.com/Trivadis/plsql-formatter-settings/tree/main/sqlcl
https://github.com/Trivadis/plsql-formatter-settings/tree/main/standalone

CAN A FORMATTER BREAK MY CODE?

9 CASE-SENSITIVE CODE

```
SELECT j.c.accountNumber
FROM t j
WHERE j.c.accountType = 'A';
```

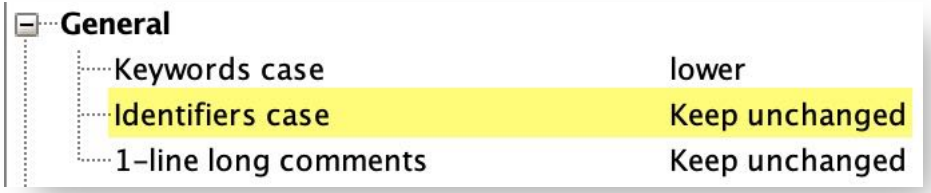
Ctrl + F7

```
select j.c.accountnumber
  from t j
 where j.c.accounttype = 'A';
```

Do not change case of identifiers!

Example of JSON Column C

```
{
    accountNumber:123,
    accountName:"Name",
    accountType:"A"
}
```



10 WRONG WHITESPACE HANDLING

```
set define on
@myscript
set define off
```


```
set define on
@myscript
set
define off
```

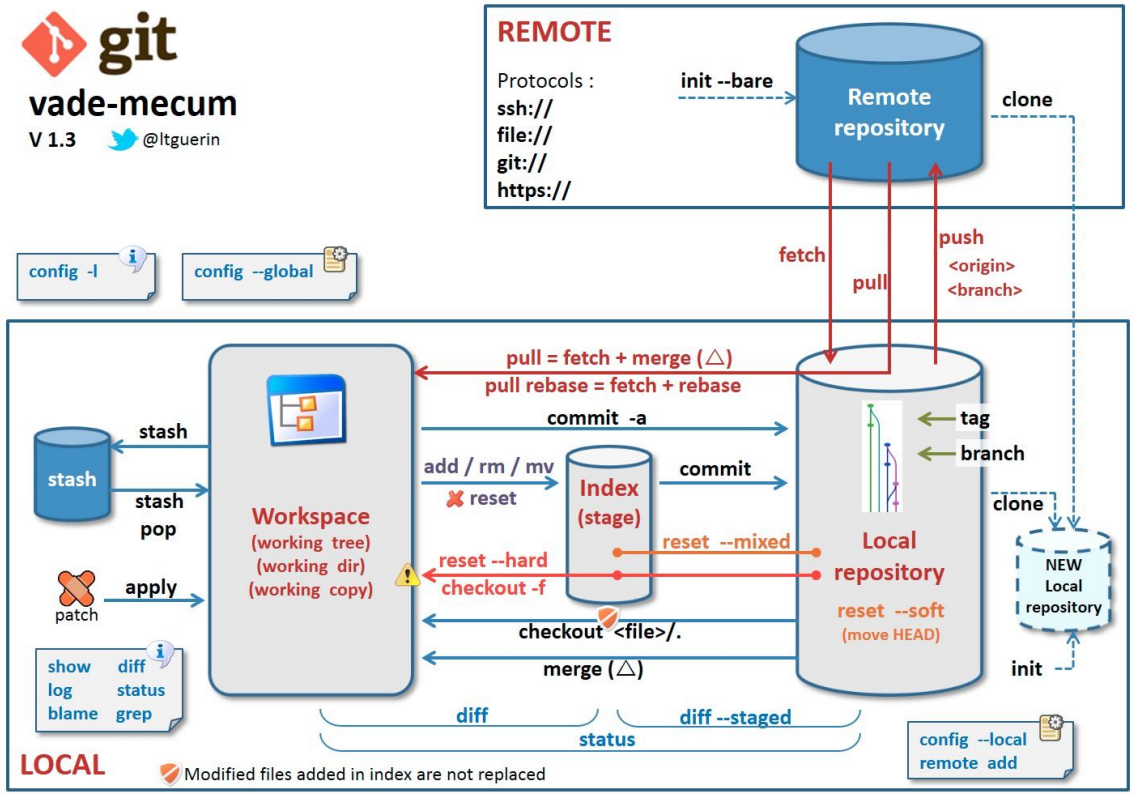
11 NON-WHITESPACE REPLACEMENTS

```
do_something1();
assert_auth(p1, p2);
do_something2();
```


```
do_something1();
undefinedassert_auth(p1, p2);
do_something2();
```

GIT HOOKS

13 LOCAL & REMOTE REPOSITORIES



Remote Hooks

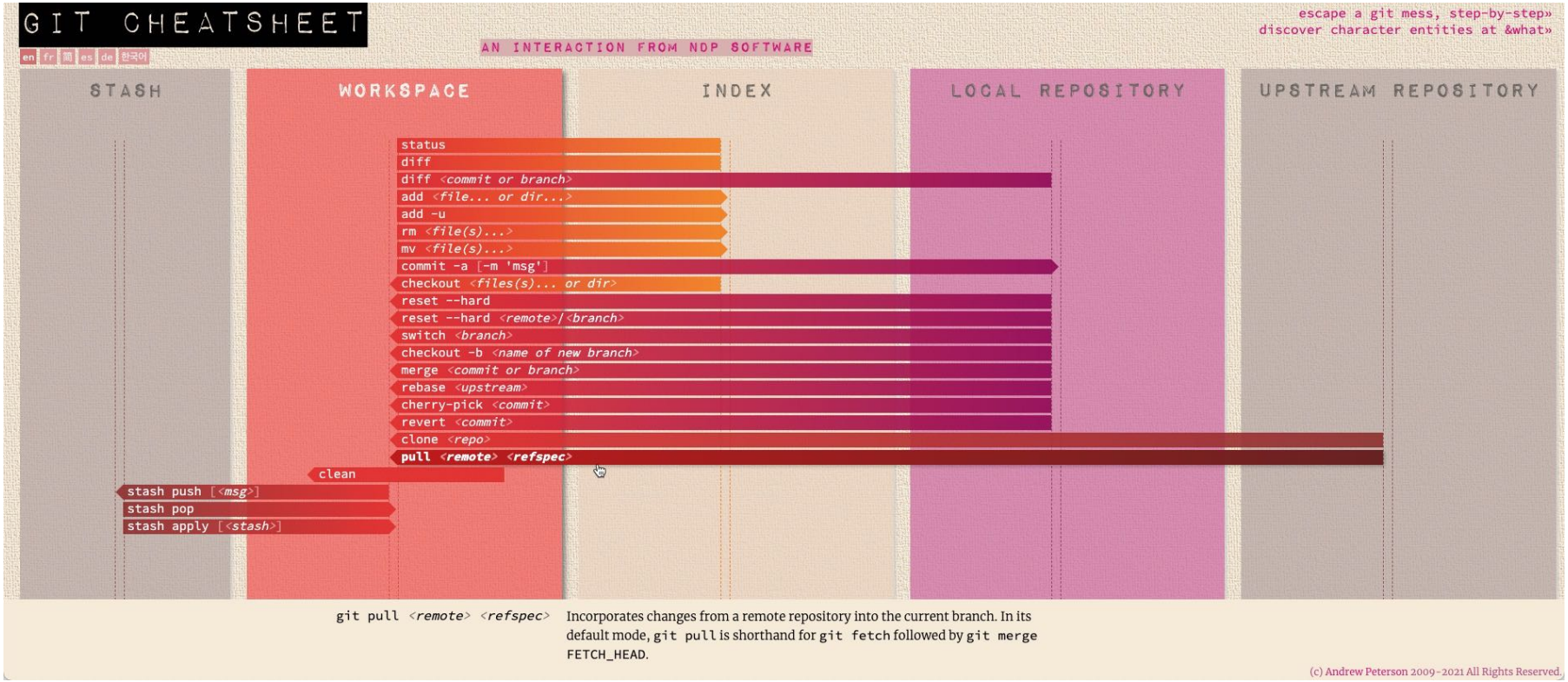
- pre-receive
- update
- post-receive

Local Hooks

- pre-commit
- prepare-commit-msg
- commit-msg
- post-commit
- post-checkout
- pre-rebase
- applypatch-msg
- pre-applypatch
- ...

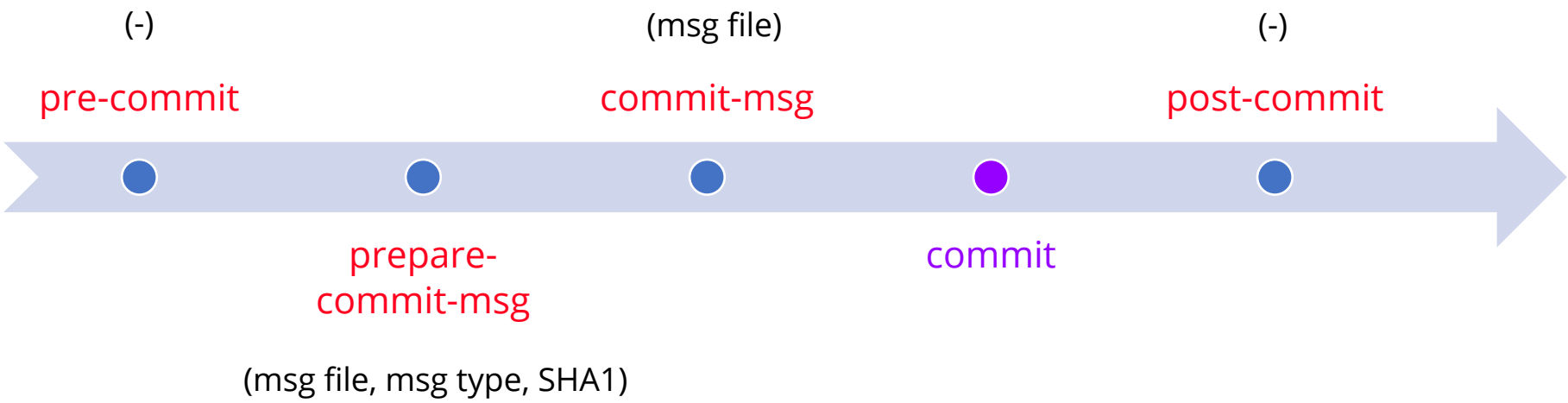
Source: https://labs.sogeti.com/git-overview-in-a-single-image/git-vademecum/

14 GIT CHEAT SHEET



Source: https://ndpsoftware.com/git-cheatsheet.html - loc=workspace;

15 COMMIT WORKFLOW



16 CAN THE EXECUTION OF PRE-COMMIT BE ENFORCED?

- No
- You cannot populate .git/hooks via "git clone" or "git pull"
- You must copy it to .git/hooks and make it executable to activate it
- You can always bypass the hook via "git commit --no-verify"

COMPATIBLITY

18 OPERATING SYSTEMS

uses Git Bash

Downalod: https://git-scm.com/downloads

19 PRE-COMMIT HOOK

- Git Command Line 2.33.1
- SourceTree 4.1.3 (via CLI)
- Visual Studio Code 1.60.2 (via CLI)
- IntelliJ IDEA 2021.2.2 (via CLI)
- Eclipse IDE 2021-03 (via EGit and JGit; no console output)
- SQL Developer 21.2.1
  - Based on JDeveloper which uses JGit 3.6.2
  - JGit 3.7 or later required for pre-commit hook (25-02-2015)
  - Current JGit Version is 5.13 (15-09-2021)

NAÏVE PRE-COMMIT HOOK

21 DOWNLOAD FORMATTER SETTINGS

```
mkdir formatter
cd formatter
SETTINGS_REPO=https://raw.githubusercontent.com/Trivadis/plsql-formatter-settings/main
curl $SETTINGS_REPO/sqlcl/format.js -O
curl $SETTINGS_REPO/settings/sql_developer/trivadis_advanced_format.xml -O
curl $SETTINGS_REPO/settings/sql_developer/trivadis_custom_format.arbori -O
cd ..
git add formatter
git commit -m "formatter settings from Trivadis/plsql-formatter-settings"
```

22 CREATE FILE .git/hooks/.pre-commit

```
#!/bin/sh

function get_staged_files() { :; }

function format_staged_files_in_workspace() { :; }

function update_staging_area() { :; }

get_staged_files
format_staged_files_in_workspace
update_staging_area
exit 0
```

23 GET ADDED OR MODIFIED FILES IN STAGING AREA

```
function get_staged_files() {
    STAGED_FILES=`git diff --cached --name-only --diff-filter=AM`
}
```

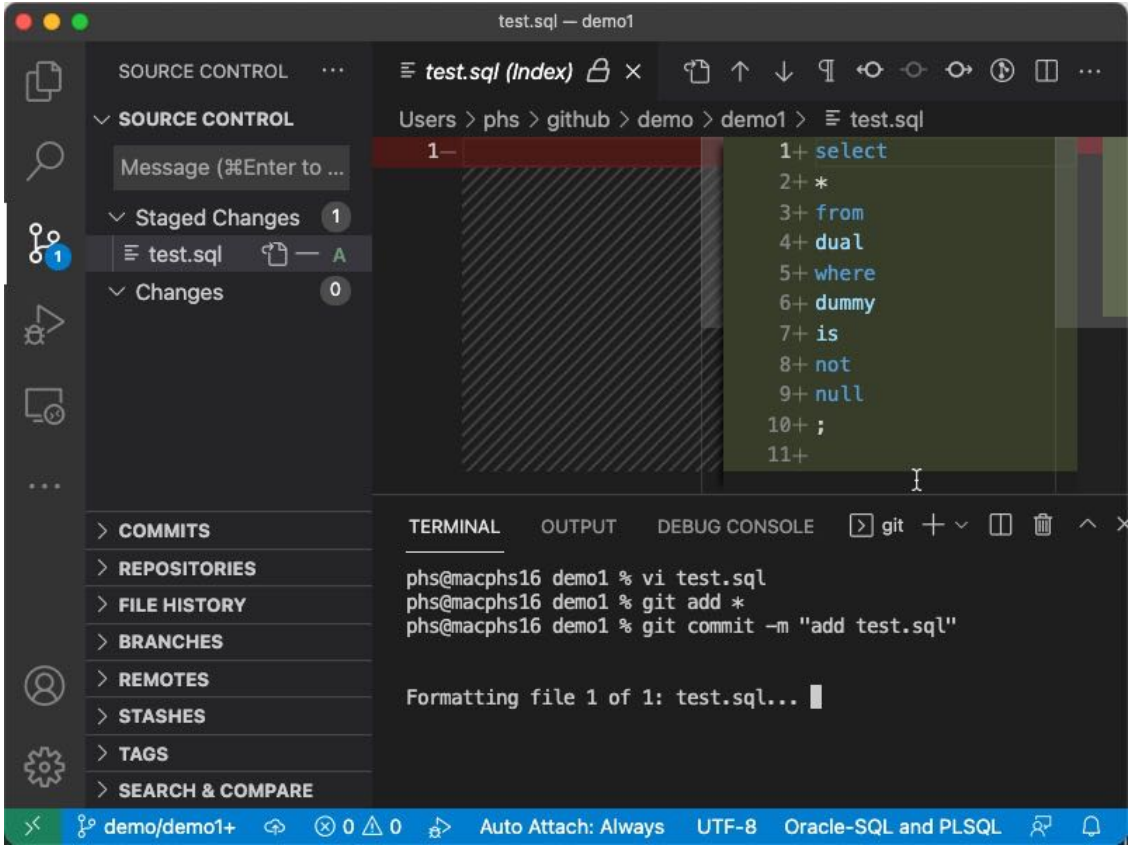
24 FORMAT STAGED FILES

```
function format_staged_files_in_workspace() {
    for file in $STAGED_FILES
    do
        sql -nolog <<EOF
script formatter/format.js "$file" \
xml=formatter/trivadis_advanced_format.xml \
arbori=formatter/trivadis_custom_format.arbori
EOF
    done
}
```

25 UPDATE STAGING AREA

```
function update_staging_area() {
    for file in $STAGED_FILES
    do
        git add $file
    done
}
```

26 DEMO – NAÏVE PRE-COMMIT HOOK



- Add new file
- Add file copies
- Add partial change (1)

trivadis
Part of Accenture

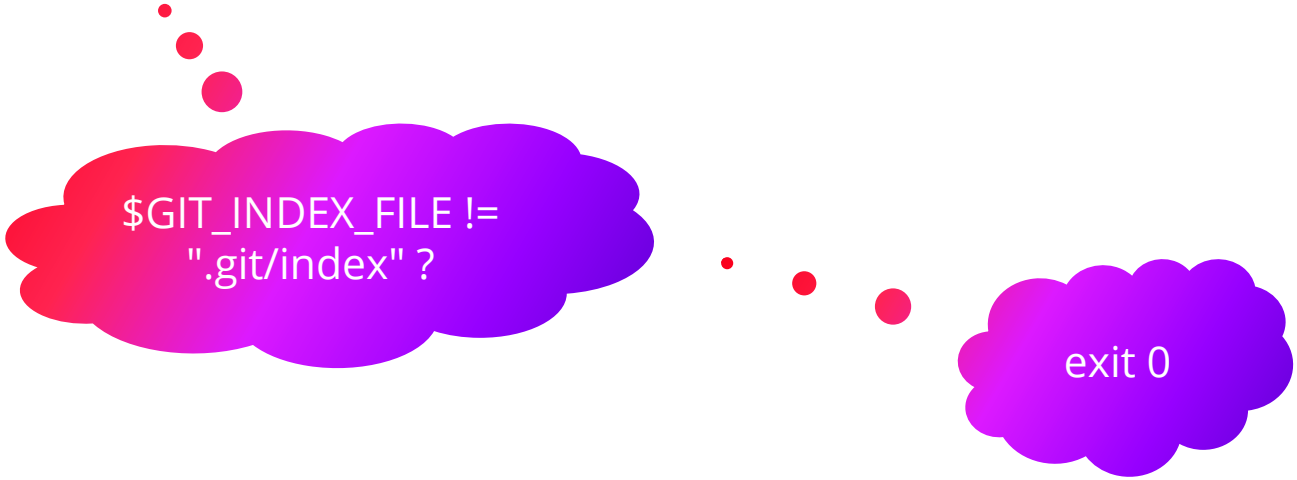
FLAWS OF NAÏVE PRE-COMMIT HOOK

28 NO CHECKS FOR PREREQUISITES

- SQLcl's executable found in path?
- format.js found?

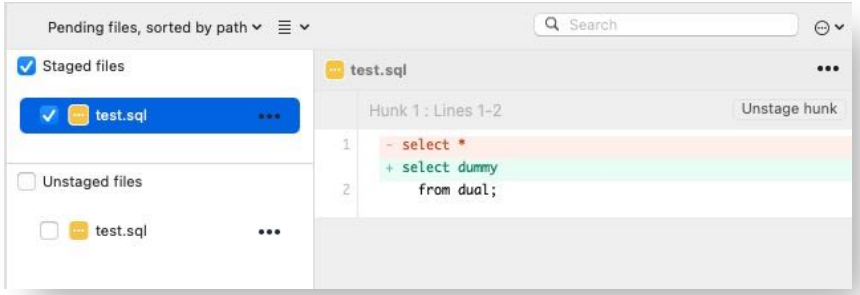
29 NO SUPPORT FOR NON-DEFAULT STAGING AREA

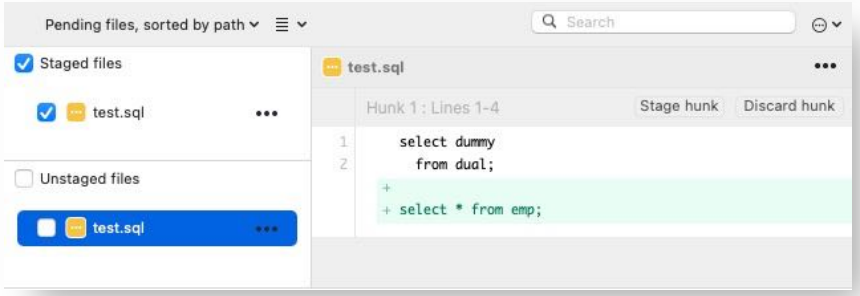
- git commit --all
- git commit --include <file> ...
- git commit --only <file> ...



30 NO SUPPORT FOR PARTIAL COMMITS

- git add --patch <file>





Differences between staging area and workspace for staged files?

exit 0

trivadis
Part of Accenture

31 UNNECESSARY COMMITS

HEAD

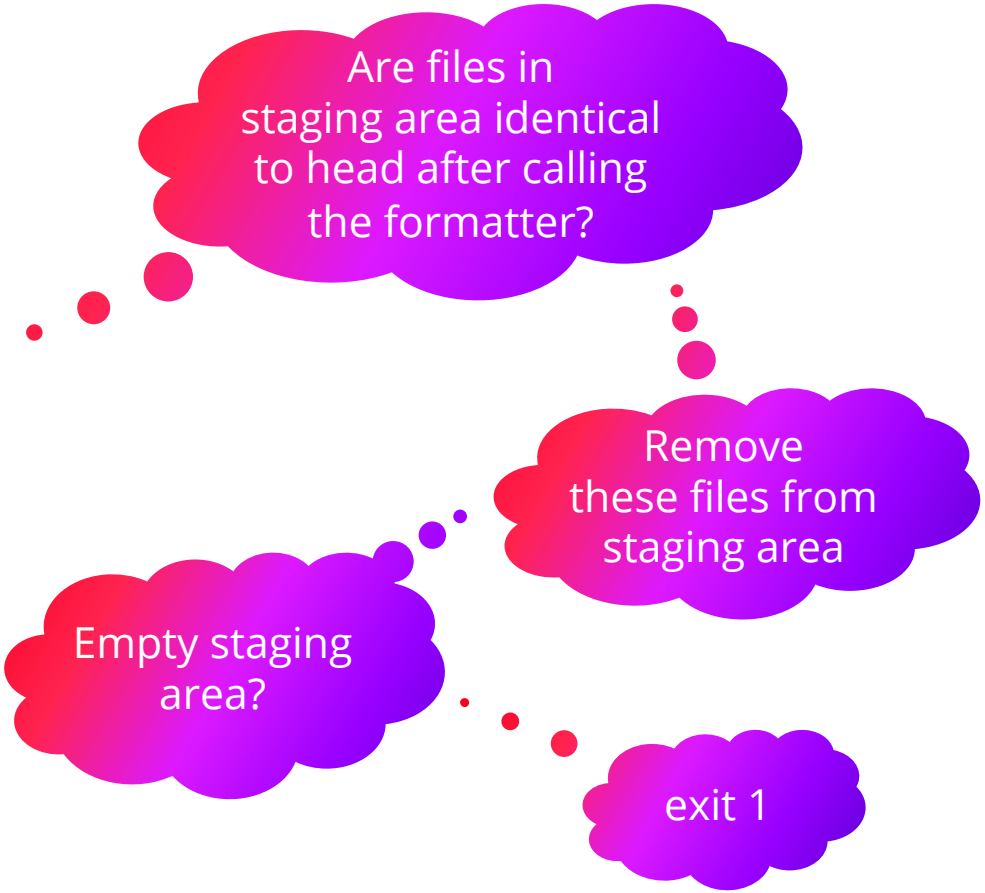
```
select *
  from dual
 where dummy is not null;
```

Staging Area before Formatter Call

```
select *
  from dual
 where dummy is not null
;
```

Staging Area after Formatter Call

```
select *
  from dual
 where dummy is not null;
```



32 OTHER FLAWS OF NAÏVE PRE-COMMIT HOOK

- SQLcl called per staged file
- SQLcl called also for irrelevant files
- Executed formatter statement not shown
- Not suited to initially format all SQL files
- Set up process for other team members could be simplified

BETTER PRE-COMMIT HOOK

34 CONFIGURABLE PRE-COMMIT HOOK

README.md

Minimal SQLcl & `format.js` Variant

1. Clone this repository.
2. Copy the `pre-commit` script into the `.git/hooks` directory of your target Git workspace.
3. Open `.git/hooks/pre-commit` in an editor and change the environment variable `$FORMATTER_GITHUB_DIR` to point to the root of the Git repository you cloned in step 1.

The minimal installation variant has the advantage that you do not have to change anything in your target directory. The downside is that you are dependent on another locally installed Git repository. And you have not documented the formatter settings used for your target repository.

Optimal SQLcl & `format.js` Variant

1. Create a subdirectory `formatter` in the workspace of your target repository
2. Copy the following files into this subdirectory:
   - `pre-commit`
   - `settings/sql_developer/trivadis_advanced_format.xml`
   - `settings/sql_developer/trivadis_custom_format.arbori`
   - `sqlcl/format.js`
3. Open the `pre-commit` script in your workspace in an editor and change the following environment variables:

Enivronment Variable	New Value
`FORMATTER_JS`	`"formatter/format.js"`
`FORMATTER_SQLDEV_SETTINGS_DIR`	`"formatter"`

4. Create a file named `install-pre-commit-hook.sh` in the `formatter` directory and save it with the following content:

```
#!/bin/sh

FORMATTER_DIR="$(dirname $0)"
GIT_HOOK_DIR="$FORMATTER_DIR/../.git/hooks"
cp $FORMATTER_DIR/pre-commit $GIT_HOOK_DIR/pre-commit
chmod +x $GIT_HOOK_DIR/pre-commit
echo "pre-commit hook installed in $GIT_HOOK_DIR/pre-commit."
```

5. Open a terminal window (on Windows use `"C:\Program Files\Git\bin\bash.exe" --cd-to-home`), change to the root directory of your workspace and run the following commands:
   - `chmod +x formatter/install-pre-commit-hook.sh`
   - `formatter/install-pre-commit-hook.sh`
6. Commit the files in the `formatter` subdirectory and push the changes so that others can install the hook with the same formatter configuration.

This installation variant is independent of the `Trivadis/plsql-formatter-settings` Git repository and offers all team members to automatically format their code on `git commit` using the same settings.

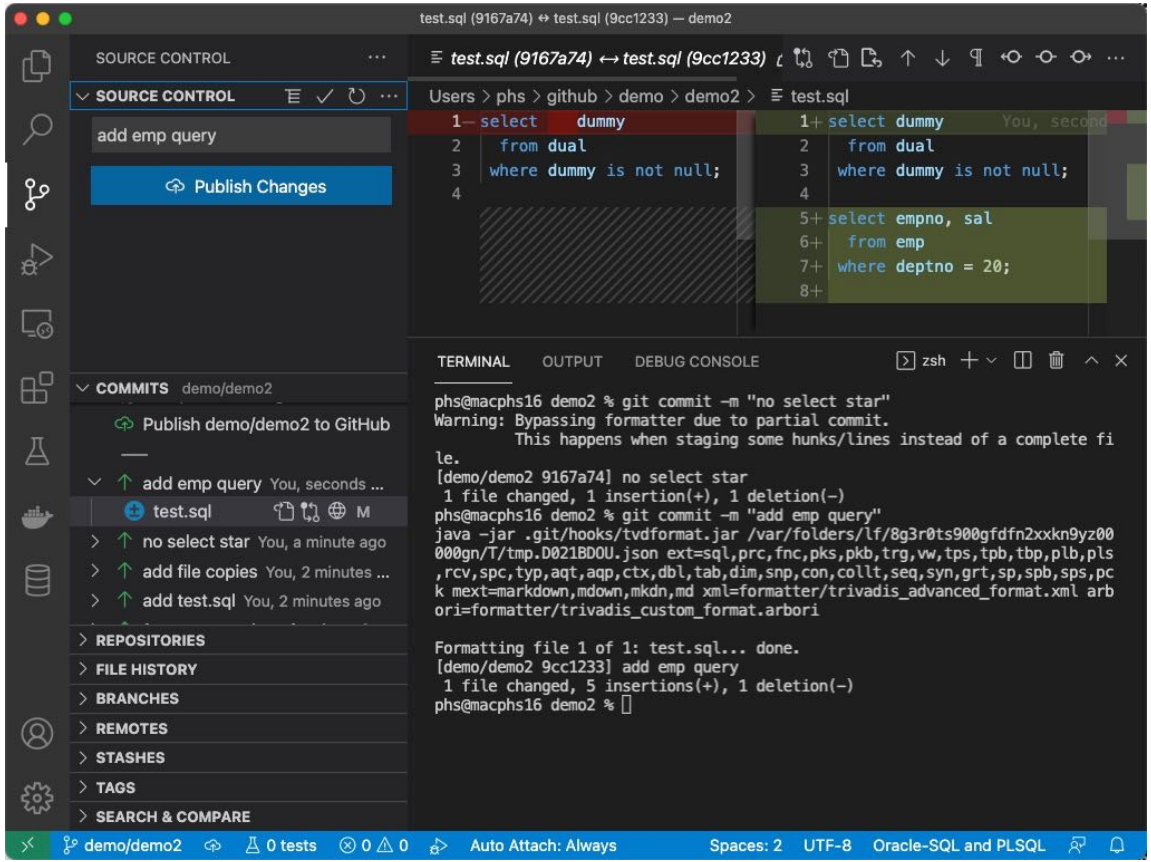
Executable File | 275 lines (241 sloc) | 9.54 KB — Raw | Blame

```
#!/bin/sh

#
# Copyright 2021 Philipp Salvisberg <philipp.salvisberg@trivadis.com>
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#     http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
#

# See https://github.com/Trivadis/plsql-formatter-settings/hook for usage instructions.

# ------------------------------------------------------------------------------
# Configuration section - Change values of variables to fit your needs.
# ------------------------------------------------------------------------------

# Git pre-commit formatter variant:
#    1) tvdformat command via SQLcl (command must be registered via $SQLPATH/login.sql)
#    2) format.js script via SQLcl (the default)
#    3) tvdformat.jar standalone executable JAR (includes `format.js`, no SQLcl required)
FORMATTER_VARIANT="2"

# Location of the local clone of the plsql-formatter-settings Git repository.
# Used as root directory of known subdirectories and files in the Git respository.
FORMATTER_GITHUB_DIR="$HOME/github/trivadis/plsql-formatter-settings"

# Location of the `format.js` JavaScript file. Required only for $FORMATTER_VARIANT 2.
FORMATTER_JS="$FORMATTER_GITHUB_DIR/sqlcl/format.js"

# Location of the `tvdformat.jar` JAR. Required only for $FORMATTER_VARIANT 3.
FORMATTER_JAR="$FORMATTER_GITHUB_DIR/standalone/target/tvdformat.jar"

# Location of the SQL Developer's settings
FORMATTER_SQLDEV_SETTINGS_DIR="$FORMATTER_GITHUB_DIR/settings/sql_developer"

# Formatter options beside $FORMATTER_SCOPE
FORMATTER_EXT="sql,prc,fnc,pks,pkb,trg,vw,tps,tpb,tbp,plb,pls,rcv,spc,typ,aqt,aqp,ctx,dbl,tab,dim,snp,con,collt,seq,syn,grt,sp,spb,sps,pck"
FORMATTER_MEXT="markdown,mdown,mkdn,md"
FORMATTER_XML="$FORMATTER_SQLDEV_SETTINGS_DIR/trivadis_advanced_format.xml"
FORMATTER_ARBORI="$FORMATTER_SQLDEV_SETTINGS_DIR/trivadis_custom_format.arbori"
FORMATTER_OPTS="ext=$FORMATTER_EXT mext=$FORMATTER_MEXT xml=$FORMATTER_XML arbori=$FORMATTER_ARBORI"

# SQLcl options.
# `-nolog` is mandatory and the use of `-` instead of `/` makes it compatible across all platforms.
SQLCL_OPTS="-nolog -noupdates -S"

# Show formatter command before execution? true/false.
FORMATTER_SHOW_COMMAND=true
```

Source: https://github.com/Trivadis/plsql-formatter-settings/tree/main/hook

trivadis
Part of Accenture

35 DEMO – BETTER PRE-COMMIT HOOK



- Add partial change (2)
- Change formatter settings
- Format directory
- Whitespace changes

trivadis
Part of Accenture

CORE MESSAGES

37 FORMATTING CODE VIA GIT PRE-COMMIT HOOK

- See https://github.com/Trivadis/plsql-formatter-settings/tree/main/hook
- You cannot enforce formatting rules
- But you can automate and therefore simplify the formatting task
- Each file is eventually formatted

TOGETHER WE ARE #1 PARTNER FOR BUSINESSES TO HARNESS THE POWER OF DATA FOR A SMARTER LIFE

trivadis
Part of Accenture