

Configuração de Ferramentas de Merge

“ O **SourceTree** não oferece ferramenta própria para *merge* e *diff*, porém permite a configuração de aplicações externas para realizar essas operações. O **P4Merge** atende o uso comum, como uma aplicação de *3-way merge tool*, *freeware* e que é suportada pelo **SourceTree**. E há casos mais específicos, no caso de sistemas desenvolvidos em **Curio**, para os quais é necessária uma ferramenta para lidar com eventuais conflitos nos arquivos de modelo (.uml). O **Oso XML Merge** cumpre esse papel, onde ferramentas como o *P4Merge*, *WinMerge*, ou *Meld* falham ao interpretar diferenças em arquivos baseados em *xml*.

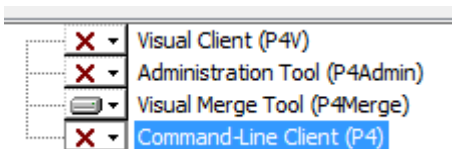
Vamos aprender a configurar as duas ferramentas para uso simultâneo pelo **SourceTree**, ora o **P4Merge**, para arquivos comuns, ora o **Oso XML Merge**, para modelos UML.

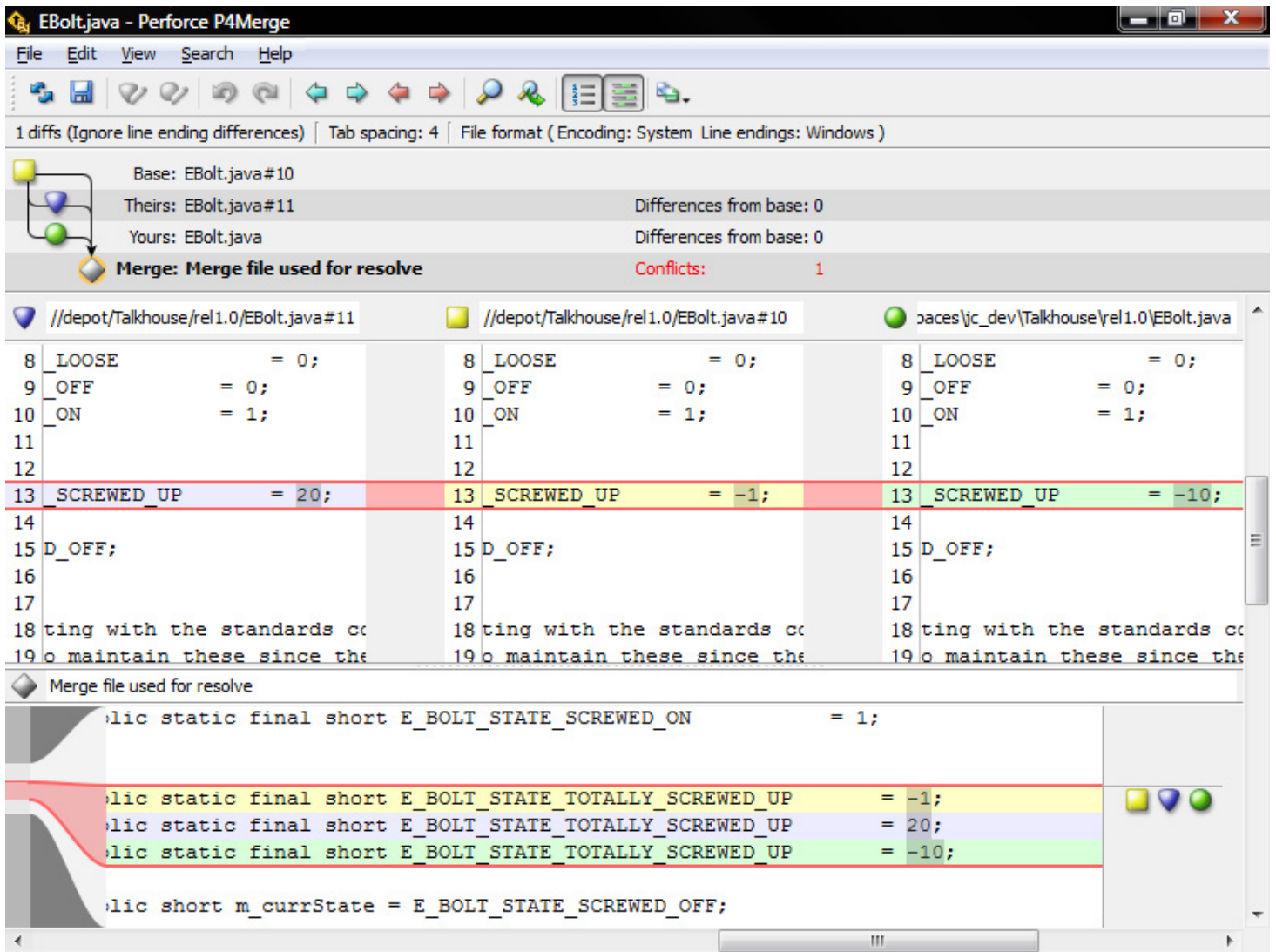
Instalação das Ferramentas

Helix Merge Tool (P4Merge)

Você pode fazer o download da última versão do HELIX P4V: Visual Client, O **P4Merge** faz parte do pacote. Se algum tipo de registro for sugerido você pode ignorá-lo e baixar mesmo assim ().

Durante a instalação, é recomendado **remover** as outras *features*, desse modo:

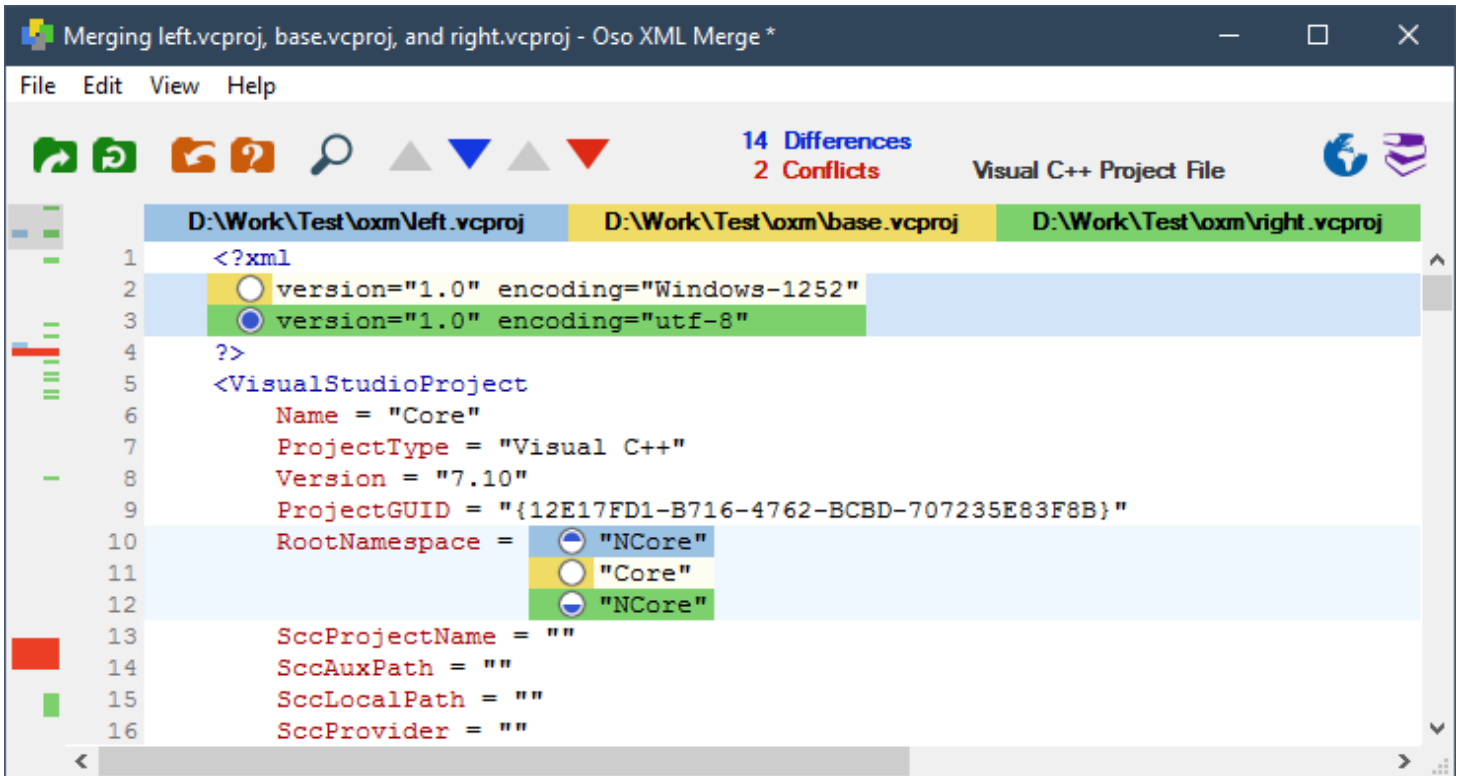




Oso XML Merge

A Evolúgica mantém uma licença para a versão 2.0 do Oso XML Merge. Você pode instalá-lo e registrá-lo com a *license key* `ZADFUI-DDK5FH` . e *username* `isaacpereira` .

OBS: Instalar a versão: **v2.0.1205** para funcionar corretamente.



Para habilitar a interpretação de arquivos `.uml` é necessário fornecer uma especificação de formato xml. Já desenvolvemos uma especificação para os arquivos UML do **StarUML** que pode ser obtida aqui: `StarUML.pmfmt`. Este arquivo deve ser mantido em `C:\ProgramData\Oso\XMLMerge64\2\Formats`.

Configuração de Merge/Diff Tool por extensão de arquivo

Com as duas ferramentas instaladas e configuradas, podemos preparar o git e *SourceTree* para ativar cada uma delas baseada na extensão do arquivo. O propósito é invocar o **P4Merge** para qualquer tipo de arquivo, exceto `.uml`, momento no qual o **Oso XML Merge** deve ser usado.

1. **Configurar *SourceTree* para usar parâmetros `System Default` para ferramentas de Merge e Diff.**

No *SourceTree* vá em `Tools → Options → Diff` e verifique se as configurações são as mesmas da figura abaixo. Isso deixará a cargo do Git instalado no sistema a definição de quais

ferramentas serão usadas para cada ação.

External Diff / Merge

External Diff Tool:

Diff Command: Arguments:

Merge Tool:

Diff Command: Arguments:

Available command arguments for 'Custom': \$LOCAL (= mine), \$REMOTE (= theirs), \$BASE (= common parent), \$MERGED (= merge output)

2. Editar `.gitconfig` do usuário

O trecho abaixo é um exemplo de configuração do `git` com declarações de ferramentas de `merge` e `diff` disponíveis. Por padrão, usaremos *shell scripts* para direcionar a chamada entre as **merge/diff tools** `default` e `uml`. Você pode incluir esse trecho ao final do seu `.gitconfig`, (ele geralmente fica na raiz do seu usuário) ou substituir por este: `.gitconfig`.

```
[merge]
  tool = mergewrapper
[mergetool "mergewrapper"]
  cmd = ~/merge-wrapper.sh \"$BASE\" \"$LOCAL\" \"$REMOTE\" \"$MERGED\"
[mergetool "default"]
  cmd = 'C:/Program Files/Perforce/p4merge.exe' \"$BASE\" \"$LOCAL\" \"$REMOTE\" \"$MERGED\"
[mergetool "uml"]
  cmd = 'C:/Program Files/Oso/XMLMerge/2/OsoXMLMerge.exe' -merge -base \"$BASE\" -left \"$LOCAL\" -right \"$REMOTE\" -result \"$MERGED\"

[diff]
  tool = diffwrapper
[difftool "diffwrapper"]
  cmd = ~/diff-wrapper.sh \"$LOCAL\" \"$REMOTE\"
[difftool "default"]
  cmd = 'C:/Program Files/Perforce/p4merge.exe' \"$LOCAL\" \"$REMOTE\"
[difftool "uml"]
  cmd = 'C:/Program Files/Oso/XMLMerge/2/OsoXMLMerge.exe' -compare -left \"$LOCAL\" -right \"$REMOTE\"
```

3. Criar scripts de *wrapper* de `merge` (`merge-wrapper.sh`) e `diff` (`diff-wrapper.sh`) para invocar condicionalmente cada ferramenta.

Note que o `.gitconfig` do passo anterior invoca por padrão `~/merge-wrapper.sh` e `~/diff-wrapper.sh`, são esses *shell scripts* que implementam o chaveamento entre as execuções das aplicações de merge, você obtê-los aqui (`merge-wrapper.sh`), e aqui (`diff-wrapper.sh`), respectivamente, e pode mantê-los na raiz do seu usuário.

merge-wrapper.sh

```

BASE=$1
LOCAL=$2
REMOTE=$3
MERGED=$4

WDIFF="\.uml"

if echo "$BASE" | grep -q "$WDIFF"; then
echo "Using merge tool for uml file";
  CMD=$(git config mergetool.uml.cmd)
eval $CMD
else
#checking for additional special merge tool cases
#Finally, the default case:
  echo "Using default merge tool";
  CMD=$(git config mergetool.default.cmd)
  eval $CMD
fi

```

diff-wrapper.sh

```

LOCAL=$1
REMOTE=$2

WDIFF="\.uml"

if echo "$LOCAL" | grep -q "$WDIFF"; then
echo "Using diff tool for UML file";
  CMD=$(git config difftool.uml.cmd)
eval $CMD
else
#checking for additional special merge tool cases
#Finally, the default case:
  echo "Using default diff tool";
  CMD=$(git config difftool.default.cmd)
  eval $CMD
fi

```

Revision #3

Created Wed, Oct 11, 2017 2:13 PM by Isaac Pereira

Updated Tue, Nov 3, 2020 8:27 PM by brunoagrizzi