

Technical Memo

Date: 11/19/2002

To: DU GPF

CC: PZ RSA SMU

From: FR

Subject: mall

Name:

Table of content

1. PROBLEMS	2
2. RULES	2
3. EXAMPLE	3



PRELIMINARY documents contain information on a product under development and is issued for evaluation purposes only.

Features characteristic data and other information are subject to change.

TI PROPRIETARY INFORMATION

UNDER NON DISCLOSURE AGREEMENT

STRICTLY PRIVATE

PAGE: 1/4

1. Problems

- Most GPF components can be build in more than one configuration. The number of configurations and the command to build them depends on the component. But for a software release all configurations have to be build and from a release managers point of view this build should be easy and independent from the component.
- Builds on a ClearCase view should use clearmake as build tool (because date/time comparison is not sufficient under ClearCase), otherwise gnumake shall be used. On the other hand not all operations needed for building can be performed by clearmake, namely ClearCase operations like checkout/checkin are not possible (This is by design.). Therefore a combination of clearmake and gnumake has to be used.
- During a release build all GPF components shall be stamped with a “label”. The label is a string that contains the version number of the component (normally a ClearCase label). From a release managers point of view applying such a label should be easy and independent from the component.

2. Rules

In order to ease the release process regarding the above problems each component has to support a `mall` command (normally a batch file).

- Each component has it's own 'home directory'. `mall` will be executed in this home directory (i.e. it is not possible that two components share the same home directory).
- `mall` has (at minimum) to support the following commands:

`mall`

This command builds all configurations of the component. This includes
> local build (home directory) using clearmake on a ClearCase View and gnumake otherwise,
> export of built deliverables (Libraries, Binaries) to the appropriate directories with checkout before (gnumake).

All output produced by the build tools (compiler etc) has to be redirected to a file named `xxx_err.txt` located in the `gpf` directory.

`mall` shall return 0 if the build was successful and 1 in case of an error.

`mall -clean`

This command clears all locally built files (Libraries, Binaries, Objects). No ClearCase operations, no influence on the exported deliverables.

`mall` shall return 0 in this case.

`mall -l <Label>`

This command performs the same actions as `mall` and additionally stamps the deliverables with the label given. Remark: The label should not be kept permanently, i.e. a subsequent make shall build a version with the default label (`DEV_BUILD`).

For the return code see above (`mall`).



PRELIMINARY documents contain information on a product under development and is issued for evaluation purposes only.

Features characteristic data and other information are subject to change.

TI PROPRIETARY INFORMATION

UNDER NON DISCLOSURE AGREEMENT

STRICTLY PRIVATE

PAGE: 2/4

```
mall -test
```

This command executes all test scripts a developer has defined for his/her component. All output produced by the scripts has to be redirected to a file named xxx_err.txt located in the gpf directory.

mall shall return 0 if all tests were successful and 1 in case of an error.

3. Example

Here is a simple example for mall.bat:

```
@echo off
REM build NT UTILS
setlocal

REM special cases -clean and -l <LABEL>
if "%1" == "-clean" goto clean
if "%1" == "-test" goto test

REM default Label:
set LABEL=DEV_BUILD
if not "%1" == "-l" goto build
set LABEL=%2
shift
shift

:build
REM use clearmake if under ClearCase
REM default is gnumake:
set M=gnumake
REM is clearcase installed ?
if "%CLEARCASE_PRIMARY_GROUP%" == "" goto doit
REM are we using a clearcase view ?
cleartool lsview -cview >nul 2>>&1
if errorlevel 1 goto doit
REM on a clearcase view: use clearmake in gnu mode:
set M=clearmake -V -C gnu
:doit

echo *** making NT UTILS ***

REM path to VisualStudio
if "%MSVCDIR%" == "" CALL vcvars32.bat

REM phase 1: the build
echo using %M% >..\..\ntutils.err
%M% DEBUG=0 LABEL=%LABEL% %1 %2 %3 >>..\..\ntutils.err 2>>&1
if %? != 0 goto err
%M% DEBUG=1 LABEL=%LABEL% %1 %2 %3 >>..\..\ntutils.err 2>>&1
if %? != 0 goto err

REM phase 2: checkout if necessary
REM (only if clearcase is used - but cannot use clearmake)
if "%M%" == "gnumake" goto type
gnumake DEBUG=0 %1 export >>..\..\ntutils.err 2>>&1
if %? != 0 goto err
gnumake DEBUG=1 %1 export >>..\..\ntutils.err 2>>&1
if %? != 0 goto err

:type
REM to see the errors
type ....\ntutils.err
goto end

REM simple clean:
:clean
```



TI PROPRIETARY INFORMATION

PAGE: 3/4

PRELIMINARY documents contain information on a product under development and is issued for evaluation purposes only.

UNDER NON DISCLOSURE AGREEMENT

Features characteristic data and other information are subject to change.

STRICTLY PRIVATE

```
del /s /q /y bin\*
del /s /q /y obj\*
goto end

REM simple test:
:test
echo I think this component has no bugs ☺
goto end

:end
endlocal
quit 0

:err
type ..\..\ntutils.err
echo ***** ERROR - see ..\..\ntutils.err
endlocal
quit 1
```



PRELIMINARY documents contain information on a product under development and is issued for evaluation purposes only.

Features characteristic data and other information are subject to change.

TI PROPRIETARY INFORMATION

UNDER NON DISCLOSURE AGREEMENT

STRICTLY PRIVATE

PAGE: 4/4