



GI -NTF

Content

1	INTRODUCTION	1
2	DELIVERABLE FILES.....	1
3	LIBRARIES PRODUCED BUT NOT DELIVERED	2
4	LIBRARIES USED.....	2
5	MAKE INFORMATION	3
5.1	BUILD PATCH	3
5.2	ENVIRONMENT VARIABLES	4
5.3	RESERVED COMPONENT SETTINGS.....	4
5.4	COMPONENT SCRIPTS	5
6	REFERENCES	5
7	TERMINOLOGY	5

History

Version	Date	Adjustments
A	2006-12-20	New patch build

1 Introduction

This document describes how to build an NTF release and a patch for the NTF release.

2 Deliverable Files

Name	Description	Environment
ntf_<rev>_crh109127.<arch>.sh	The base version	



	delivery file. <rev> is a filename-friendly encoding of the R-state, e.g. : r6a <arch> is the computer architecture, e.g: solaris8	
<patch ID>.<build nr>.<crh109127.<arch>.zip	The patch delivery file. <patch ID> is the patch ID, e.g. 1_NTF_R11B. <build nr> is the build number, e.g. 001. <arch> is the computer architecture, e.g: solaris8	

3 Libraries produced but not delivered

Name	Description	Environment
<Library name>		

4 Libraries used

Name	Description	Environment



5 Make information

5.1 Build Patch

The patch is build upon the SUN patch packages. This patch build extends the foundation interface for patch building of components, se [1].

1. Change directory to `/vobs/ipms/ntf/tools`
2. Edit the `mkpatch.cfg` file with the environment variables in (Table 1, chapter 5.2) correct values.
NOTE: Check in the file `mkpatch.cfg` in the vob after successful build.
3. Add the files that shall be included in patch delivery. List the files in file `/vobs/ipms/ntf/patches/BuildPatchFiles/pkg/prototype.template` with type of file, target location, source location, file permission, owner and group.

```
<file type> none <target>=<source> <file permission> <owner> <group>
```

Example:

```
f none bin/ntf.jar=/vobs/ipms/ntf/bin/ntf.jar 0644 root other  
f none bin/ntfagent=/vobs/ipms/ntf/bin/ntfagent 0755 root other
```

Verify that you have all files listed in the `prototype.template` file that shall be delivered with the patch.

NOTE: Check in the file `prototype.template` in the vob after successful build.

4. Run the script `mkpatch.sh`
% `./mkpatch.sh`
5. Answers to the questions if you want to use the environment variables in Table 1 in the patch build.
6. Answers to the question if you want to lable all files in the NTF vob with the patch ID.
7. Wait until the script has finished execution with the result COMPLETED.
8. A delivery file and a README file is generated in the `/vobs/ipms/ntf/delivery` directory
9. Verify the information in the README file.
NOTE: If this patch obsoletes other patches, it's recommended to use the latest version of the obsolete patches README file and edit that file with new information of this patch. This saves a lot of time since you need the information about obsolete patches in the README file.
10. Verify the delivery file.
11. Make VOB elements of the delivery file and the README file in directory `/vobs/ipms/ntf/patches`.



5.2 Environment Variables

Table 1

Variable Name	Description	Value
VERSION	The NTF base vesrsion	R11B
PATCHID	The ID of the patch. NOTE: The build number must be incremented for each build.	1_NTF_R11B.001
PKG	The abbreviation package name	LMENtf
PKGNAME	The full package name and short description	NTF Notification Server
PLATFORM	The target platform where the patch can be installed.	SOL_8
OBSOLETES	Optional parameter. ID of patches that are obsoleted by this patch.	1_NTF_R11B.001
REQUIRES	Optional parameter. ID of patches required by this patch. NOTE: The patch installation terminates if the required patches is not installed.	2_NTF_R11B.002
INCOMPAT	Optional parameter. ID of patches that are NOT compatible with this patch. NOTE: The patch installation terminates if incompatible patches are installed.	1_NTF_R11B.001

5.3 Reserved Component Settings

The reserved component settings below defined in [1] are used by NTF in the comp_* scripts.

- noselfdiag (comp_preinstall, comp_postinstall)
- noshutdownbefore (comp_preinstall)



- nostartafter (comp_postinstall)
- dostartafter (comp_postinstall)
- nounlockafter (comp_postinstall)
- downlockafter (comp_postinstall)

5.4 Component Scripts

The component scripts are located in /vobs/ipms/ntf/patches/BuildPatchFiles/pkg directory. The scripts are copied to the /vobs/ipms/ntf/build directory at build time. The following component scripts exists:

- comp_checkinstall - This script checks that the environment is OK.
- comp_preinstall - This script runs the health check, stores the administrative state, the operational state, shutdown NTF and stop NTF.
- comp_postinstall - This script start NTF, restores the administrative state and runs the health check.
- comp_preremove - This script stop NTF before uninstallation.
- comp_postremove - This script start NTF after uninstallation.

6 References

- [1] GI - Foundation
1/GI-CRH 109 581/1 Uen

7 Terminology

Term	Explanation
------	-------------