POL00397162

Document Title:	SYSMAN Support Tasks for HNG-X
Document Reference:	DEV/INF/LLD/0079
Document Type:	Low Level Design (LLD)
Release:	INT1+
Abstract:	Low Level Design Document detailing the Support Tasks implemented under the SYSMAN2 and SYSMAN3 environments for HNG-X.
Document Status:	DRAFT
Author & Dept:	Graham Comer, SMG; John Bradley, SMG; Contributors: Brian Gallacher, Steve Perkins
External Distribution:	None
Security Risk Assessment Confirmed	YES

Approval Authorities:

Name	Role	Signature	Date
M.Conneely (Author of parent HLD)	Solution Design / Infrastructure Design		

Note: See Royal Mail Group Account HNG-X Reviewers/Approvers Role Matrix (PGM/DCM/ION/0001) for guidance.

POL00397162

0 Document Control

0.1 Table of Contents

POL00397162

POL00397162

0.2 Document History

Version No.	Date	Summary of Changes and Reason for Issue	Associated Change - CP/PEAK/PPRR Reference
0.1	30/03/09	First Draft	
0.2	27/05/09 11/06/09	CP0332 (DXC) and Tripwire additions Non technical and cosmetic changes	CP0332
0.3	30/06/09 07/09/09	Tripwire Task Library Documented Insertion of missing narrative in a number of sections / subsections to complete document content Non technical (e.g. Updating of section 0.3 Review Details contents to reflect up to date list of Reviewers) and cosmetic (to improve legibility) changes	CP0174

0.3 Review Details

Review Comments by :	17th September 2009
Review Comments to :	Graham Comer & John Bradley & Richard Stevens
Mandatory Review	
Role	Name
Solution Design / Infrastructure Design	Mike Conneely (Author of parent HLD)
System Test	John Rogers
SSC	Mik Peach
Optional Review	
Role	Name
Security and Risk Team	CSPOA.security GRO
System Qualities Architect	Dave Chapman
Architect	Jason Clark
Business Continuity	Adam Parker
Service Support	Kirsty Gallacher
HNG-X Service Transition	Graham Welsh
Service Network	Ian Mills
Data Centre Migration	Geoff Butts
Data Centre Migration	Peter Okely
SV&I Manager	Sheila Bamber
Tester	Hamish Munro
RV Manager	James Brett (POL, JTT)
VI & TE Manager	Mark Ascott
Integrity Testing	Alan Child

POL00397162

Integrity Testing	Michael Welch
Development	Graham Allen
Networks Architect (Data Centre)	Mark Jarosz
Issued for Information – Please restrict this distribution list to a minimum	
Position/Role	Name
Lead Architect Systems and Estate Management	Ian Bowen

(*) = Reviewers that returned comments

POL00397162

0.4 Associated Documents (Internal & External)

Reference	Version	Date	Title	Source
PGM/DCM/TEM/0001 (DO NOT REMOVE)	4.0	21-Nov-08	RMGA HNG-X Generic Document Template	Dimensions
PGM/DCM/ION/0001	49.0	20-Aug-2009	HNG-X Document Reviewers / Approvers Role Matrix	Dimensions
DES/APP/HLD/0010			Generic Internal Web Services High Level Design	Dimensions
DES/SYM/HLD/0044			SYSMAN Support Tasks for HNG-X	Dimensions
DEV/INF/LLD/0036			Proactive Monitoring Windows Supporting Agents	Dimensions
DEV/INF/LLD/0025	1.1		Build for Tivoli Provisioning Manager	Dimensions
DEV/INF/LLD/0039			Proactive Monitoring Unix Supporting Agents	Dimensions
HNG-X CP0174 (PVCS Ref CP4645)			Intro of File Integrity Monitoring for HNG-X to Support PCI Compliance (Updated)	RMGA Ch Mgt
HNG-X CP0332 (PVCS Ref CP4863)			SYSMAN Software Delivery to Branch Estate	RMGA Ch Mgt

Unless a specific version is referred to above, reference should be made to the current approved versions of the documents.

0.5 Abbreviations

Abbreviation	Definition
API	Application Programming Interface
APOP	Automated Payments – Out Pay, service identifier
BBND	Broadband checker, service identifier
BKAC	Bank Account Number checker, service identifier
CP	Change Proposal
DVLA	Driver & Vehicle Licensing Agency, service identifier
EES	Enterprise Event Server
EPM	Enterprise Provisioning Server
HDSK	Help Desk, service identifier
HLD	High Level Design
HNG-X	Horizon Next Generation – Plan X
MGRM	MoneyGram, service identifier
OCP	Operational Change Proposal
OLCT	Online Counter Training, service identifier
OS	Operating System

POL00397162

PAF	Postal Address File, service identifier
PGDD	Postal Guaranteed Delivery Date, service identifier
POLFS	Post Office Limited Financial Services
TCOM	The Web Service that implements the BBND, BKAC and PGDD Web Service Agents. Originally called the Telecoms Service and now referred to as the Service Hub Service
TPM	Tivoli Provisioning Manager

0.6 Glossary

Term	Definition
SYSMAN2	The Horizon Systems Management Infrastructure
SYSMAN3	The HNG-X Systems Management Infrastructure

0.7 Changes Expected

Changes
Agreed changes in response to reviewer comments / issues following formal review

0.8 Accuracy

Fujitsu Services endeavours to ensure that the information contained in this document is correct but, whilst every effort is made to ensure the accuracy of such information, it accepts no liability for any loss (however caused) sustained as a result of any error or omission in the same.

0.9 Security Risk Assessment

Security risks have been assessed and it is considered that there are no security risks relating specifically to this document.

POL00397162

1 Introduction

1.1 Purpose

This document describes the Low Level Design of the systems management task infrastructure and details the support tasks that have be Specified with DES/SYM/HLD/0044

1.2 Scope

The associated High Level Design document defines the requirements, design objectives and overall scope for this design.

This document provides the detailed low level design and installation of the deliverables identified within the aforementioned HLD.

1.3 Assumptions

None

1.4 Risks

There are no known risks associated with the design.

POL00397162

2 Requirements

This LLD should address the basic requirements outlined in HLD DES/SYM/HLD/0044.

This LLD should also address the Web Service requirements of Stats collection and MoneyGram Web Service daily restart as outlined in DES/APP/HLD/0010.

The LLD must describe the installation and configuration of all the deployment components.

POL00397162

3 Design Overview

3.1 Installation Methodology

3.1.1 SYSMAN2

SYSMAN2 Deliverables will be manual installs delivered via PVCS

3.1.2 SYSMAN3

SYSMAN3 Tasks will be delivered via Dimensions

3.1.2.1 Framework Tasks

Framework 4.2.2 Tasks will be delivered such that they can be installed via TPM

3.1.2.2 TPM Workflows

TPM Workflows will be delivered as tcdrivers that can be imported into TPM

3.1.2.3 APDE: .tcdriver file creation

tcdriver file creation is described in DEV/INF/LLD/0025

POL00397162

4 Implementation

4.1 MTMR_TECAD_TASKS

This product TEC_NT_Adapter_Tasks Stops or start the Tivoli event adapter. Optionally deletes the local cache.

This has been enhanced to detect the Netcool probe and perform the same operations for that. The task will work against Horizon or HNG-X counters

4.1.1 Installation

4.1.1.1 Installation Files

The following files are delivered for the installation:

File	Description
install_tecad_tasks.sh	Install Script
tec_adapter.tll	Task Library Definition File
check_tec_nt_adapter.sh	Executes the underlying task NT_Check_TEC_NT_Adapter
check_tec_win_adapter.sh	Executes the underlying task NT_Check_TEC_WIN_Adapter
control_tec_nt_adapter.sh	Executes the underlying task NT_Control_TEC_NT_Adapter
control_tec_win_adapter.sh	Executes the underlying task NT_Control_TEC_WIN_Adapter
install_tec_nt_adapter.sh	Executes the underlying task NT_Install_TEC_NT_Adapter
install_tec_win_adapter.sh	Executes the underlying task NT_Install_TEC_WIN_Adapter
nt_check_tec_nt_adapter.sh	Checks the installation of the TEC NT Adapter
nt_check_tec_win_adapter.sh	Checks the installation of the Windows TEC adapter
nt_control_tec_nt_adapter.sh	Stops and starts either the NT TEC Adapter or NT Probe
nt_control_tec_win_adapter.sh	Stops and starts either the Windows TEC Adapter or Windows probe
nt_install_tec_nt_adapter.sh	Installs the TEC NT Adapter
nt_install_tec_win_adapter.sh	Installs the TEC Windows Adapter

4.1.1.2 Installation Process

The product is installed by running install_tecad_tasks.sh

When the installation script is run, it will perform the following actions –

```
##############################################################################
#       GET ARGUMENTS
##############################################################################
PolicyRegion=$1
```

POL00397162


```
#############################################################################
#       INITIALISE
#############################################################################
ALI_OID=`wlookup ServerManagedNode`
eval SERVER=`idlcall $ALI_OID _get_label`

#############################################################################
#       CHECKS
#############################################################################
if [ -z "$PolicyRegion" ]; then
        usage
fi

#############################################################################
#       Check Region
#############################################################################
wsetpr TaskLibrary @PolicyRegion:"$PolicyRegion"
if [ $? != 0 ]; then
        echo "Error setting TaskLibrary resource for PolicyRegion $PolicyRegion"
        exit 1;
fi


#############################################################################
#       Delete existing jobs and tasks
#############################################################################
echo "Deleting Redundant Jobs and Tasks..."

wdeljob Control_Tivoli_NT_Adapter Counter_Tasks >/dev/null 2>&1
wdeljob Control_Tivoli_NT_Adapter NWB_Additional_Tasks >/dev/null 2>&1
wdeljob Adapter_Control_Counter NWB_Additional_Tasks >/dev/null 2>&1
wdeljob Adapter_Control_Server NWB_Additional_Tasks >/dev/null 2>&1

wdeltask Control_Tivoli_NT_Adapter Counter_Tasks >/dev/null 2>&1
wdeltask Control_Tivoli_NT_Adapter NWB_Additional_Tasks >/dev/null 2>&1
```

POL00397162


```
wdeltask Adapter_Control_Counter NWB_Additional_Tasks >/dev/null 2>&1
wdeltask Adapter_Control_Server NWB_Additional_Tasks >/dev/null 2>&1

###############################################################################
#       INSTALL TLL
###############################################################################
echo "Installing TaskLibrary..."
wtll -r -P /usr/lib/cpp -p "$PolicyRegion" tec_adapter.tll


###############################################################################
#       Creating Jobs and Linking into Appropriate Collections
###############################################################################

# TEC Adapter Check & Control
echo "Creating Jobs..."
wcrtjob -D -j Check_TEC_NT_Adapter_Counter -l TEC_NT_Adapter_Tasks -t
Check_TEC_NT_Adapter_Counter -M parallel -m 600 -o 17 -h mastertmr
wcrtjob -D -j Install_TEC_NT_Adapter_Counter -l TEC_NT_Adapter_Tasks -t
Install_TEC_NT_Adapter_Counter -M parallel -m 600 -o 17 -h mastertmr
wcrtjob -D -j Check_TEC_NT_Adapter_Server -l TEC_NT_Adapter_Tasks -t
Check_TEC_NT_Adapter_Server -M parallel -m 600 -o 17 -h mastertmr
wcrtjob -D -j Install_TEC_NT_Adapter_Server -l TEC_NT_Adapter_Tasks -t
Install_TEC_NT_Adapter_Server -M parallel -m 600 -o 17 -h mastertmr
wcrtjob -D -j Control_TEC_NT_Adapter_Counter -l TEC_NT_Adapter_Tasks -t
Control_TEC_NT_Adapter_Counter -M parallel -m 600 -o 17 -h mastertmr
wcrtjob -D -j Control_TEC_NT_Adapter_Server -l TEC_NT_Adapter_Tasks -t
Control_TEC_NT_Adapter_Server -M parallel -m 600 -o 17 -h mastertmr

wcrtjob -D -j Control_TEC_WIN_Adapter_Server -l TEC_NT_Adapter_Tasks -t
Control_TEC_WIN_Adapter_Server -M parallel -m 600 -o 17 -h mastertmr
wcrtjob -D -j Check_TEC_WIN_Adapter_Server -l TEC_NT_Adapter_Tasks -t
Check_TEC_WIN_Adapter_Server -M parallel -m 600 -o 17 -h mastertmr
wcrtjob -D -j Install_TEC_WIN_Adapter_Server -l TEC_NT_Adapter_Tasks -t
Install_TEC_WIN_Adapter_Server -M parallel -m 600 -o 17 -h mastertmr

wln /Regions/${PolicyRegion}/TEC_NT_Adapter_Tasks/Check_TEC_NT_Adapter_Counter
/Administrators/Root_mastertmr-region/SMC_PO_Jobs
wln /Regions/${PolicyRegion}/TEC_NT_Adapter_Tasks/Check_TEC_NT_Adapter_Server
/Administrators/Root_mastertmr-region/SMC_Campus_Jobs
```

POL00397162

wln /Regions/${PolicyRegion}/TEC_NT_Adapter_Tasks/Control_TEC_NT_Adapter_Counter /Administrators/Root_mastertmr-region/SMC_PO_Jobs

wln /Regions/${PolicyRegion}/TEC_NT_Adapter_Tasks/Control_TEC_NT_Adapter_Server /Administrators/Root_mastertmr-region/SMC_Campus_Jobs

wln /Regions/${PolicyRegion}/TEC_NT_Adapter_Tasks/Check_TEC_WIN_Adapter_Server /Administrators/Root_mastertmr-region/SMC_Campus_Jobs

wln /Regions/${PolicyRegion}/TEC_NT_Adapter_Tasks/Control_TEC_WIN_Adapter_Server /Administrators/Root_mastertmr-region/SMC_Campus_Jobs

4.1.1.3 Configuration

No configuration is required

4.1.1.4 Logging

The executed tasks display output to stdout

4.1.1.5 Post Installation

At the end of the installation the Installation files may be removed from the temp directory

POL00397162

4.2 MTMR_HNGX_LOGLEVEL

A task that can set the logging level on a counter/branch or group of branches. Implementation of this is reliant upon an API provided by the counter application. Inventory should be updated when the logging level is changed so that it may be used for reporting..

4.2.1 Installation

4.2.1.1 Installation Files

The following files are delivered for the installation:

File	Description
Call_counter_log_config.sh	Executes the Set_Counter_Log_Config task
Call_counter_log_levels.sh	Executes the Set_Counter_Log_Level task
Counter_log_houskeeping.cfg	Log housekeeping configuration file
Counter_log_levels.cfg	Log Level configuration file
Counter_log_levels.sh	Script to adjust the Log Levels of the Post Office Counter Application
Counter_logging.tll	Tivoli Task Library Language file
Install_counter_tasks.sh	Installation Script
Set_counter_log_config.sh	Script to update the purge counter logfiles config script
Set_counter_log_levels.sh	Script to adjust the Log Levels of the Post Office Counter Application

4.2.1.2 Installation Process

The product is installed by running Install_counter_tasks.sh

When the installation script is run, it will perform the following actions –

- Install the TaskLibrary
- Create the job Set_Counter_LogLevel
- Create the job Set_Counter_LogRetention
- Link the jobs to the SMC_PO_Jobs Regions

4.2.1.3 Configuration

The Counter_log_levels.cfg file is used to set the number of days that counter logs are stored

i.e. c:/counters/counterapp/log=7

POL00397162

4.2.1.4 Logging

The log level is sent as an inventory record by using orainvupdate events as shown below

Log Inventory update to delete old level indicator

${LOGEVENT} -s E -r SYSMGT -e 101 "ORAINVUPDATE: D Counter_Application_Log_Level ${CURRENT_LEVEL}" >/dev/null

Log Inventory update to insert new level indicator

${LOGEVENT} -s E -r SYSMGT -e 101 "ORAINVUPDATE: I Counter_Application_Log_Level ${TARGET_LEVEL}" >/dev/null

4.2.1.5 Post Installation

At the end of the installation the Installation files may be removed from the temp directory

POL00397162

4.3 MANSENTRYCFG_CNTR_LOGS

Purge of counter log files that are older than a defined number of days. This is not strictly a *task* in Tivoli Framework terms but a local process that will be invoked from Tivoli's Sentry scheduler on a daily basis. Note that processes such as this are terminated after running for one minute so this limitation must be considered during the development.

A local configuration file will be used to define the number of days to retain logs and a set of wildcard strings that identifies each subdirectory and set of files that is to be purged. This file will be distributed as a SYSMAN2 software package..

4.3.1 Installation

4.3.1.1 Installation Files

The following files are delivered for the installation:

File	Description
Purge_counter_logfiles.sh	Purge Counter Log Files script

4.3.1.2 Installation Process

- The Sentry is installed by the delivery Sentry process on the master tmr

4.3.1.3 Configuration

See Loglevel Task

4.3.1.4 Logging

The script logs to C:/sysmgt/Counter_Log_Housekeeping/ Counter_Log_Housekeeping_${DATE}.log

POL00397162

4.4 SYSMAN3 Framework Tasks

4.4.1 SMG_Tripwire_Update

4.4.1.1 Function

A manual Tivoli Framework task that will be executed after a tripwire platform has received a software update on Solaris, Linux and Windows platforms.

When a known change is to take place (such as those conducted under CP or OCP) then the final action performed by support on conclusion of the change, should be to run a Tripwire database update using an approved report i.e. one that has been analyzed by Operations Security

4.4.1.2 Parameters

The parameters will be passed interactively on task execution

Parameter Name	IN/OUT	Mandatory	Description
Target	IN	YES	Server Node Name. Supplied as task target
Report	IN	YES	Name of the report to perform update on. Input into task GUI on task execution
Tripwire Passphrase	IN	YES	Passphrase required by tripwire. Input into task GUI on task execution
Administrator	IN	YES	Name of the Tivoli Administrator. Automatically retrieved by the task GUL arg. layout

4.4.1.3 Calls

The task must be executed via the Tivoli Desktop GUI.

Navigate to the SMG_Tripwire_Tasks task library, in the Tasks Policy Region, and execute the SMG_Tripwire_Update task.

Select the Tivoli endpoint that is the target of the Tripwire update to be the task target. Check the 'Output to Desktop' check box and execute the task.

A 'Configure Task Arguments' dialog is displayed (illustrated over leaf).

Enter the name of the report to be used to perform the update in the Report field. Only the report name is required, the task assumes the report directory is:

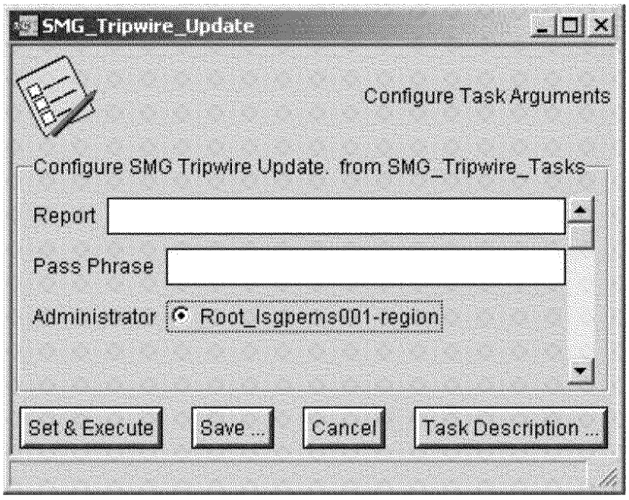
`C:\Program Files\Tripwire\TFS\report` on windows servers and

`/opt/tripwire/tfs/report` on UNIX systems

Enter the local Pass Phrase for the target server in the Pass Phrase field.

POL00397162

Set and Execute the task



Configure Task Argument Dialog

4.4.1.4 Task Pseudo Code

.Reads in passed parameters by the task

- Report name — Report to be used to perform the Tripwire update against
- Pass Phrase — Local pass phrase to authorise the Tripwire update
- Administrator name — Name of the administrator executing the Tripwire update task

Sets log file, dependent on OS:

Operating System	Log File
Windows	`C:/sysmgt/logs/SMG_Tripwire_Update.log`
UNIX	`/opt/sysmgt/logs/SMG_Tripwire_Update.log`

Maps UNIX/Windows event severities:

UNIX Syslog Severity	Windows Event Log Severity
CRITICAL	ERROR

POL00397162

WARNING	WARNING
INFORMATION	INFORMATION

Sets the report directory, dependent on OS:

Operating System	Tripwire Report Directory
Windows	`C:\Program Files\Tripwire\TFS\report`
UNIX	`/opt/tripwire/tfs/report`

Sets the Tripwire update command, dependent on OS:

Operating System	Tripwire Update Command
Windows	`tripwire --update --report-file "$reportDir\$report" --accept-all --local-passphrase $passPhrase`
UNIX	`/opt/tripwire/tfs/bin/tripwire --update --report-file $reportDir/$report --accept-all --local-passphrase $passPhrase`

Opens logfile

Raises a WARNING severity event (UNIX syslog, Windows Event Log), ID 111, indicating that a Tripwire update is being requested, passing back the name of the administrator performing the Tripwire update, and the name of the report being used to perform the update against.

Executes Tripwire update

An event is raised (UNIX syslog, Windows Event Log) indicating the result of update; severity INFORMATION on success and CRITICAL/ERROR on failure.

4.4.1.5 Logging

The Tripwire update task appends to the log.

Operating System	Logfile
Windows	`C:/sysmgt/logs/SMG_Tripwire_Update.log`
UNIX	`/opt/sysmgt/logs/SMG_Tripwire_Update.log`

4.4.1.6 Audit

The result of the Tripwire Update action, including the invoking user, is logged for audit as syslog/Event Log event

POL00397162

Source	ID	Severity	Message
TRIPWIRE	111	WARNING	SMG_Tripwire_Update - $administrator requested Tripwire update on $ENV{ENDPOINT} using report $report.
TRIPWIRE	112	CRITICAL	SMG_Tripwire_Update - $administrator requested Tripwire update on $ENV{ENDPOINT} using report $report FAILED: $!
TRIPWIRE	113	INFORMATION	SMG_Tripwire_Update - $administrator requested Tripwire update on $ENV{ENDPOINT} using report $report COMPLETED.

4.4.2 SMG_Execute_Tripwire

4.4.2.1 Shares

The Execute Tripwire task utilises two shares:

DIAGS$	/opt/ibm/Tivoli/tpm/repository/sysman	on the EPM server
$AUDIT	c:\Audit	on the EDS server

4.4.2.2 Function

A scheduled Tivoli Framework task that will execute sub tasks (SMG_Execute_Tripwire_Client) to effect the generation of a Tripwire integrity report on configured targets. The task will then transfer all generated reports to the EDS $AUDIT share. When the integrity check detects a problem; a copy of the report will also be transferred to DIAG$ share on the EPM server.

All transferred reports are converted to HTML and to a format appropriate to the OS the report is being transferred to.

4.4.2.3 Parameters

The IN parameters are read from PSPID Tags, with the exception of $file and $DIAG which are returned by the client task.

POL00397162

Parameter Name	IN/OUT	Mandatory	Description
$TARGET001 $TARGET002 $TARGET00n	IN	YES	List of targets on which to execute the tripwire client. Read in using pspid tags.
$AUDIT_SERVER	IN	YES	Sever hosting $AUDIT share. Read in using pspid tags.
$AUDIT_DIRECTORY	IN	YES	Target for distribution of client Report file on the EDS Audit Share. Read in using pspid tags.
$DIAG_SERVER	IN	YES	Server hosting DIAGS$ share. Read in using pspid tags.
$DIAG_DIRECTORY	IN	YES	Target for distribution of client report on the EDS DIAG Share. Read in using pspid tags.
$HOLDING_DIRECTORY	IN	YES	Temporary storage location on EMS server for reports being transferred to AUDIT and DIAG servers. Read in using pspid tags.
$file	IN	YES	Returned from client task: ReportName
$DIAG	IN	YES	Returned from client task: DIAG status.

4.4.2.4 Calls

This workflow can be initiated via the following calls:

Framework Call:

A Tivoli Job has been created, in the SMG_Tripwire_Tasks task library, using the command:

```
wcrtjob -D -j SMG_Execute_Tripwire_Job -l SMG_Tripwire_Tasks -t SMG_Execute_Tripwire -M parallel -m 600 -o 17 -h $EMS_Endpoint
```

The job is currently scheduled to run every Monday at 0600.

4.4.2.5 Pseudo Code

Opens logfile for append.

```
/opt/sysmgt/logs/SMG_Execute_Tripwire.log
```

Runs `wgetadmin -n` to get the name of the administrator running the task. If this fails the task raises a CRITICAL syslog event, ID 109, informing of the failure and the task script exits.

Raises a WARNING severity syslog event, ID 110, indicating that a Tripwire integrity check is being requested, passing back the name of the administrator performing the Tripwire integrity check.

POL00397162

Runs smgreadxml using and reads the resulting pidvars.sh file to obtain the parameters listed in Section 4.4.2.3.

Call wruntask on the SMG_Execute_Tripwire_Client subtask against the target list and captures the task output for analysis.

Raises a CRITICAL severity syslog event, ID 104, for each endpoint the task fails to execute on.

If the returned reportname in the task output is a valid filename (Not Error), copies the report from the target to $HOLDING_DIRECTORY on the EMS platform. Converts the file to Windows OS format, if the target is a UNIX platform, and copies the file to the EDS platform AUDIT share. The copy of the file in $HOLDING_DIRECTORY is then deleted.

If status = “DIAG”, copies the report from the target to $HOLDING_DIRECTORY on the EMS platform. Converts the file to UNIX OS format, if the target is a Windows platform, and copies the file to the EPM platform DIAGS share. The copy of the file in $HOLDING_DIRECTORY is then deleted.

4.4.2.6 Logging

The task appends to the log.

```
/opt/sysmgt/logs/SMG_Execute_Tripwire.log
```

4.4.2.7 Audit

The following syslog events are generated by the task

Source	ID	Severity	Message
TRIPWIRE	104	CRITICAL	SMG_Execute_Tripwire - Tripwire integrity check failed to execute on node:$_
TRIPWIRE	105	WARNING	SMG_Execute_Tripwire - $targetServer report possibly in wrong OS format:$endpoint:$fileName
TRIPWIRE	106	WARNING	SMG_Execute_Tripwire - $targetServer report in wrong OS format:$endpoint:$fileName.
TRIPWIRE	107	CRITICAL	SMG_Execute_Tripwire - wadminep pull failed:$endpoint:$targetFile
TRIPWIRE	108	CRITICAL	SMG_Execute_Tripwire - wadminep push failed:$endpoint:$targetFile
TRIPWIRE	109	CRITICAL	SMG_Execute_Tripwire - Failed to get administrator ID.
TRIPWIRE	110	WARNING	SMG_Execute_Tripwire - $administrator requested Tripwire integrity check on the Tripwire estate.
SYSMGT	901	CRITICAL	Monitor failed to open a file
SYSMGT	902	CRITICAL	Monitor system call failed

POL00397162

4.4.3 SMG_Execute_Tripwire_Client

4.4.3.1 Function

A sub task executed by SMG_Execute_Tripwire to execute the tripwire integrity check on Solaris, Linux and Windows platforms

4.4.3.2 Parameters

Parameter Name	IN/OUT	Mandatory	Description
Report	OUT	YES	Name of Report to be retrieved or a "No Report Generated" message if Tripwire integrity check fails
Status	OUT	YES	STATUS (for DIAG share) or "ERROR" if Tripwire integrity check fails

4.4.3.3 Calls

This workflow can be initiated via the following calls:

Framework Call:

```
wruntask -t SMG_Execute_Tripwire_Client -l SMG_Tripwire_Tasks -o 13 -h $taskTarget
```

4.4.3.4 Pseudo Code

Sets log file, dependent on OS:

Operating System	Log File
Windows	C:/sysmgt/logs/SMG_Execute_Tripwire_Client.log
UNIX	/opt/sysmgt/logs/SMG_ Execute_Tripwire_Client.log

Maps UNIX/Windows event severities:

UNIX Syslog Severity	Windows Event Log Severity
CRITICAL	ERROR
MINOR	WARNING

POL00397162

WARNING	INFORMATION

Sets the Tripwire integrity check command, dependent on OS:

Operating System	Tripwire Report Directory
Windows	`tripwire --check --text-report-level 0`
UNIX	`/opt/tripwire/tfs/bin/tripwire --check --text-report-level 0`

Sets the Tripwire convert to HTML command, dependent on OS:

Operating System	Tripwire Update Command
Windows	`twprint --print-report --report-file "$report" --report-format HTML --output-file "$HTMLreport"`
UNIX	`/opt/tripwire/tfs/bin/twprint --print-report --report-file $report --report-format HTML --output-file $HTMLreport`

Opens logfile

Executes Tripwire integrity check

If the Tripwire integrity check fails; an event is raised (UNIX syslog/Windows Event Log), an ERROR and “No Report Generated” message are output from the task. The task then exits. The event raised is of severity CRITICAL/ERROR, ID 102.

The output from the Tripwire integrity check command is processed and the generated report file name is extracted along with the value of V.

If the value of V is greater than 1 DIAG is set to be true.

The generated report is checked to see if exists and if it does it is converted to HTML format. If this conversion fails a WARNING/INFORMATION severity event is raised, but the script continues.

The DIAG status is output along with the name of the HTML report file. If the HTML conversion failed the name of the original report is output instead of the HTML report.

4.4.3.4.1 Unix

```
# /opt/tripwire/tfs/bin/tripwire --check --text-report-level 0


Parsing policy file: /opt/tripwire/tfs/policy/tw.pol
*** Processing Unix File System ***
Performing integrity check...
```

POL00397162


```
The object: "/sys" is on a different file system. Ignoring.
Wrote report file: /opt/tripwire/tfs/report/nas-20090527-123509.twr

TWReport nas 20090527123509 V:320 S:100 A:216 R:8 C:96 L:0 M:13 H:307
Integrity check complete.
```

4.4.3.4.2 Windows

```
# tripwire -check -text-report-level 0

 Parsing policy file: C:\Program Files\Tripwire\TFS\policy\tw.pol
*** Processing Windows File System ***
Performing integrity check...
*** Processing Windows Registry ***
Performing integrity check...
Wrote report file: C:\Program Files\Tripwire\TFS\report\LSBPVSH006-20090513-
1410
05.twr
TWReport LSBPVSH006 20090513141005 V:226 S:1000 A:140 R:2 C:84 L:0 M:4 H:222
Integrity check complete.
```

4.4.3.5 Logging

Appends to logfile

Operating System	Log File
Windows	C:/sysmgt/logs/SMG_Execute_Tripwire_Client.log
UNIX	/opt/sysmgt/logs/SMG_ Execute_Tripwire_Client.log

4.4.3.6 Audit

The following syslog events are generated by the task

POL00397162

Source	ID	Severity	Message
TRIPWIRE	101	WARNING	SMG_Execute_Tripwire_Client - Failed to convert report $report to HTML: $!
TRIPWIRE	102	CRITICAL	SMG_Execute_Tripwire_Client - Tripwire integrity check command failed: $!
TRIPWIRE	103	CRITICAL	SMG_Execute_Tripwire_Client - Tripwire report $report not found.

4.4.3.6.1 Unix

```
logger -t \"SEVERITY TRIPWIRE nnn\" -p user.err \"messgae"
```

Where the Severity is CRITICAL, MINOR or WARNING

Nnn is 100 for CRITICAL, 101 for MINOR and 102 for WARNING

Message is the report name and twreport data from the output from the tripwire command

4.4.3.6.2 Windows

```
c:/sysmgt/tools/LogEvent -r TRIPWIRE -s Severity "message" -e nnn
```

Where the Severity is ERROR, MINOR or WARNING

Nnn is 100 for ERROR, 101 for MINOR and 102 for WARNING

Message is the report name and twreport data from the output from the tripwire command

POL00397162

4.5 RHL_TPM_UTILS: Generic TPM Workflows

This product comprises a set of low level workflows that facilitate the construction of further workflows (such as user invoked tasks) in a modular manner.

It is not proposed that these workflows should be invoked directly by users.

4.5.1 Installation

TPM Workflows are delivered within a .tcdriver file (in essence a zip file)

.tcdriver files must be generated within the Eclipse Framework (APDE) within a TPM Environment.

The resultant compiled and zipped Workflows can be installed within any TPM environment using the TPM supplied `${TIODIR}/tools/tc-driver-manager.sh` command.

The .tcdriver and installation script should be copied to /tmp/RHL_TPM_UTILS prior to execution

No PSPID vars are required for this installation.

The TPM Server must be active prior to installation.

4.5.1.1 Installation Files

The following files are delivered for the installation:

File	Description
rhl_tpm_utils.install	TPM Utility Workflows installation script
rhl_tpm_utils.tcdriver	TPM Utility Workflows

4.5.1.2 Installation Process

The product is installed by running the rhl_tpm_utils.install script.

When the installation script is run, it will perform the following actions –

- Test for installation logfile and directory & create if either found not to exist
- Test required .tcdriver file exists in temp install directory
- Test required TPM Target directory exists – create if not
- Copy .tcdriver file to TPM Target directory
- Attempt uninstallation of existing version of .tcdriver file (this warns to log if no previous version found – but we're ok to ignore if so as this must be a fresh install)
- Run install of supplied .tcdriver file

POL00397162

4.5.1.3 Logging

All installation steps and output from `tc-driver-manager.sh` command are logged to the standard logging directory `/opt/sysmgt/logs` within file `RHL_TPM_UTILS.log`

4.5.1.4 Post Installation

At the end of the installation the Installation files may be removed from the temp directory /tmp/RHL_TPM_UTILS

4.5.2 Audit of User Action

The Task will record the running of the task as a SYSMAN3 Event

4.5.3 SMG_ControlWindowsService

4.5.3.1 Function

A workflow that will stop or start a named Windows Service on a managed computer.

The action, including the invoking user, is logged for audit.

4.5.3.2 Parameters

Parameter Name	IN/OUT	Mandatory	Description
Action	IN	YES	Either "stop" or "start" (to either stop or start the service)
ServiceName	IN	YES	Display Name or Shortname of the Windows Service to be stopped or started
TargetServer	IN	YES	TPM device ID of the Server upon which the Service is to be stopped or started

4.5.3.3 Calls

This workflow can be initiated via the following calls:

Workflow Call:

```
SMG_ControlWindowsService(Action, ServiceName, TargetServer)
```

TPM SOAP Call:

```
$TIO_HOME/soapclient/tpmlteSoap/soapcli.sh <valid_userid>
<valid_password>
"http://<tpmservername>:8777/ws/pid/TpmLiteSoapService?wsdl"
executeDeploymentRequest SMG_ControlWindowsService <Action>
<ServiceName> <TargetServer>
```

POL00397162

4.5.3.4 Pseudo Code

The Task executes a device.execute logival operation to perform a net start/stop on the requested service

4.5.3.5 Logging

Logging is via TPM

4.5.3.6 Audit

Code	Message	Description / Cause
800		Workflow Initiated
801		Workflow Completed Successfully

4.5.4 SMG_FileTransfer

4.5.4.1 Function

A workflow that will transfer a file from one managed computer to another.

The action, including the invoking user, is logged for audit.

4.5.4.2 Parameters

Parameter Name	IN/OUT	Mandatory	Description
SourceServerID	IN	YES	TPM device ID of the Server upon which the required file exists
SourceServerName	IN	NO	Optional hostname of the Source Server
SourcePath	IN	YES	Fully qualified path to the required source file
SourceFile	IN	YES	Filename of the Source file
TargetServerID	IN	YES	TPM device ID of the Server upon which the file should be copied to
TargetServerName	IN	NO	Optional hostname of the Target Server
TargetPath	IN	YES	Fully qualified path for the required target file

POL00397162

TargetFile	IN	YES	Filename of the Target file
DOS2UNIX	IN	NO	Optional switch to perform DOS to UNIX conversion on the file - "Y" will execute this – default (i.e. none specified) is "N"

4.5.4.3 Calls

This workflow can be initiated via the following calls:

Workflow Call:

```
SMG_FileTransfer(SourceServerID, <null>, SourcePath, SourceFile,
TargetServerID, <null>, TargetPath, TargetFile, <null>)
```

TPM SOAP Call:

```
$TIO_HOME/soapclient/tpmlteSoap/soapcli.sh <valid_userid>
<valid_password>
"http://<tpmservername>:8777/ws/pid/TpmLiteSoapService?wsdl"
executeDeploymentRequest SMG_FileTransfer <SourceServerID>,
<SourceServerName> <SourcePath> <SourceFile> <TargetServerID>
<TargetServerName> <TargetPath> <TargetFile> <DOS2UNIX>
```

4.5.4.4 Pseudo Code

The workflow executes a TPM FileRepository.GetFile/PutFile logical operation

4.5.4.5 Logging

Logging is via TPM

4.5.4.6 Audit

Code	Message	Description / Cause
800		Workflow Initiated
801		Workflow Completed Successfully

4.5.5 SMG_UpdateInventory

4.5.5.1 Function

A workflow that can create, update or delete and Inventory record associated with a computer. Utilised where actions are to be recorded against a computer for reporting purposes.

POL00397162

4.5.5.2 Parameters

Parameter Name	IN/OUT	Mandatory	Description
ServerID	IN	YES	TPM ID of Server whose inventory is to be updated
Inv_Component_Name	IN	YES	Name of Component to be Inserted / Updated
Inv_Component_Value	IN	YES	Value to be associated to Component Insert / Update

4.5.5.3 Calls

This workflow can be initiated via the following calls:

Workflow Call:

```
SMG_UpdateInventory(ServerID, Inv_Component_Name, Inv_Component_Value)
```

TPM SOAP Call:

```
$TIO_HOME/soapclient/tpmlteSoap/soapcli.sh <valid_userid> <valid_password>
"http://<tpmservername>:8777/ws/pid/TpmLiteSoapService?wsdl"
executeDeploymentRequest SMG_UpdateInventory <ServerID> <Inv_Component_Name>
<Inv_Component_Value>
```

4.5.5.4 Pseudo Code

The Task performs DCMQuery operations

4.5.5.5 Logging

Logging is via TPM

4.5.5.6 Audit

Code	Message	Description / Cause
800		Workflow Initiated
801		Workflow Completed Successfully

POL00397162

4.6 RHL_TPM_SUPPORTTASKS: User-Invoked TPM Workflows

4.6.1 Installation

TPM Workflows are delivered within a .tcdriver file (in essence a zip file)

.tcdriver files must be generated within the Eclipse Framework (APDE) within a TPM Environment.

The resultant compiled and zipped Workflows can be installed within any TPM environment using the TPM supplied `${TIODIR}/tools/tc-driver-manager.sh` command.

The .tcdriver and installation script should be copied to /tmp/RHL_TPM_SUPPORTTASKS prior to execution

No PSPID vars are required for this installation.

The TPM Server must be active prior to installation.

4.6.1.1 Pre-Requisites

The following products must be installed prior to this product;

Product	Description
WIN_SMG_INSTSUPPLIB	Latest version of the Systems Management Support Library – this contains unloadevent.exe used by the HNGX_Unload_Event_Log workflow
RHL_TPM_UTILS	TPM Utility Workflows

4.6.1.2 Installation Files

The following files are delivered for the installation:

File	Description
rhl_tpm_supporttasks.install	TPM Utility Workflows installation script
rhl_tpm_supporttasks.tcdriver	TPM Utility Workflows

4.6.1.3 Installation Process

The product is installed by running the rhl_tpm_supporttasks.install script.

When the installation script is run, it will perform the following actions –

POL00397162

- Test for installation logfile and directory & create if either found not to exist
- Test required .tcdriver file exists in temp install directory
- Test required TPM Target directory exists – create if not
- Copy .tcdriver file to TPM Target directory
- Attempt uninstallation of existing version of .tcdriver file (this warns to log if no previous version found – but we're ok to ignore if so as this must be a fresh install)
- Run install of supplied .tcdriver file

4.6.1.4 Logging

All installation steps and output from `tc-driver-manager.sh` command are logged to the standard logging directory `/opt/sysmgt/logs` within file `RHL_TPM_SUPPORTTASKS.log`

4.6.1.5 Post Installation

At the end of the installation the Installation files may be removed from the temp directory /tmp/RHL_TPM_SUPPORTTASKS

4.6.2 HNGX_Audited_File_Transfer

4.6.2.1 Function

This workflow will transfer a named file from one managed computer to the TPM DIAGS repository.

A compression option is available to reduce file size.

Files over 100Mb cannot be transferred with this workflow.

DOS formatted files can be converted to UNIX format during a transfer.

4.6.2.2 Parameters

Parameter Name	IN/OUT	Mandatory	Description
SourceServer	IN	YES	Hostname of the server housing the required file (Identified within Favourite Tasks / **Select Target Computers** page when running task)
SourcePath	IN	YES	Fully qualified path to the required file
SourceFile	IN	YES	Filename of the file to be transferred
FileCompress	IN	NO	Option to compress the file prior to transfer with gzip (i.e. creates .gz file) "Y" or "N" (defaults to "N")

POL00397162

DOS2UNIX	IN	NO	Option to convert file from DOS format to UNIX. "Y" or "N" (defaults to "N")

4.6.2.3 Calls

This workflow is to be initiated via the Favourite Tasks page.

Note – This workflow must be defined as per instructions in Section 5 Creating and Sharing Favourite Tasks

4.6.2.4 Pseudo Code

This task invokes the SMG_FileTransfer Workflow

4.6.2.5 Logging

Logging is via TPM native logging facilities

4.6.2.6 Audit

Code	Message	Description / Cause
800		Workflow Initiated
801		Workflow Completed Successfully

4.6.3 HNGX_Control_Windows_Service

4.6.3.1 Function

This workflow will **start** or **stop** a named Windows Service on a managed computer.

This workflow can only be executed against computer definitions recorded as being a Windows platform

4.6.3.2 Parameters

POL00397162

Parameter Name	IN/OUT	Mandatory	Description
TargetServer	IN	YES	TPM device ID of the Server upon which the Service is to be stopped or started (Identified within Favourite Tasks / **Select Target Computers** page when running task)
Action	IN	YES	Either "stop" or "start" (to either stop or start the service)
ServiceName	IN	YES	Display Name or Shortname of the Windows Service to stopped or started

4.6.3.3 Calls

This workflow is to be initiated via the Favourite Tasks page.

Note – This workflow must be defined as per instructions in Section 5 Creating and Sharing Favourite Tasks.

4.6.3.4 Pseudo Code

This workflow invokes the SMG_Control_Windows_Service Workflow

4.6.3.5 Logging

Logging is via TPM native logging facilities

4.6.3.6 Audit

Code	Message	Description / Cause
800		Workflow Initiated
801		Workflow Completed Successfully

4.6.4 HNGX_Diags_Run_df

4.6.4.1 Function

This workflow runs a **df** command on a specified managed computer.

POL00397162

4.6.4.2 Parameters

Parameter Name	IN/OUT	Mandatory	Description
TargetServer	IN	YES	TPM Device ID of managed computer to run command on (Identified within Favourite Tasks / **Select Target Computers** page when running task)
CommandArgs	IN	NO	Any command arguments. e.g. -k (block size in K) -h (human readable output) NOTE – the following chars are not allowed ; & | < >
DiagnosticOutput	IN	NO	Option to save results to a file saved in the TPM Repository DIAGS share. "Y" or "N" (defaults to "N")
CommandOutput	OUT	n/a	Output of the executed command. Displayed within workflow logs and saved to DIAGS file if required.

4.6.4.3 Calls

This workflow is to be initiated via the Favourite Tasks page.

Note – This workflow must be defined as per instructions in Section 5 Creating and Sharing Favourite Tasks.

4.6.4.4 Pseudo Code

Uses the device.execute logical operation to execute a df command

4.6.4.5 Logging

Logging is via TPM native logging facilities

4.6.4.6 Audit

POL00397162

Code	Message	Description / Cause
800		Workflow Initiated
801		Workflow Completed Successfully

4.6.5 HNGX_Diags_Run_dir

4.6.5.1 Function

This workflow runs a **dir** command on a specified managed computer.

This workflow is primarily designed to run the Windows **dir** command on targets with computer definitions recorded as being a Windows platform. However, execution on a UNIX platform will result in similar out to an **ls** command.

4.6.5.2 Parameters

Parameter Name	IN/OUT	Mandatory	Description
TargetServer	IN	YES	TPM Device ID of managed computer to run command on (Identified within Favourite Tasks / **Select Target Computers** page when running task)
CommandArgs	IN	NO	Any command arguments. e.g. *<directory_name>* /S (display subdirectories too) /Q (display owner) NOTE – the following chars are not allowed `; & | < >`
DiagnosticOutput	IN	NO	Option to save results to a file saved in the TPM Repository DIAGS share. "Y" or "N" (defaults to "N")
CommandOutput	OUT	n/a	Output of the executed command. Displayed within workflow logs and saved to DIAGS file if required.

POL00397162

4.6.5.3 Calls

This workflow is to be initiated via the Favourite Tasks page.

Note – This workflow must be defined as per instructions in Section 5 Creating and Sharing Favourite Tasks.

4.6.5.4 Pseudo Code

Uses the device.execute logical operation to execute a dir command

4.6.5.5 Logging

Logging is via TPM native logging facilities

4.6.5.6 Audit

Code	Message	Description / Cause
800		Workflow Initiated
801		Workflow Completed Successfully

4.6.6 HNGX_Diags_Run_ls

4.6.6.1 Function

This workflow runs an **ls** command on a specified managed computer.

This workflow is primarily designed to run the UNIX **ls** command on targets with computer definitions recorded as being a UNIX platform. However, execution on a Windows platform will result in similar out to, using the cygwin provided **ls** command.

4.6.6.2 Parameters

Parameter Name	IN/OUT	Mandatory	Description
TargetServer	IN	YES	TPM Device ID of managed computer to run command on (Identified within Favourite Tasks / **Select Target Computers** page when running task)
CommandArgs	IN	NO	Any command arguments. e.g. -ltc <*directory_name*>

POL00397162

			-la <*file_name*> NOTE – the following chars are not allowed `; & | < >`
DiagnosticOutput	IN	NO	Option to save results to a file saved in the TPM Repository DIAGS share. "Y" or "N" (defaults to "N")
CommandOutput	OUT	n/a	Output of the executed command. Displayed within workflow logs and saved to DIAGS file if required.

4.6.6.3 Calls

This workflow is to be initiated via the Favourite Tasks page.

Note – This workflow must be defined as per instructions in Section 5 Creating and Sharing Favourite Tasks.

4.6.6.4 Pseudo Code

Uses the device.execute logical operation to execute an ls command

4.6.6.5 Logging

Logging is via TPM native logging facilities

4.6.6.6 Audit

Code	Message	Description / Cause
800		Workflow Initiated
801		Workflow Completed Successfully

4.6.7 HNGX_Diags_Run_ps

4.6.7.1 Function

This workflow runs a **ps** command on a specified managed computer.

This workflow is primarily designed to run the UNIX **ps** command on targets with computer definitions recorded as being a UNIX platform. However, execution on a Windows platform will result in similar out to, using the cygwin provided **ps** command.

POL00397162

4.6.7.2 Parameters

Parameter Name	IN/OUT	Mandatory	Description
TargetServer	IN	YES	TPM Device ID of managed computer to run command on (Identified within Favourite Tasks / **Select Target Computers** page when running task)
CommandArgs	IN	NO	Any command arguments. e.g. -ef NOTE – the following chars are not allowed `; & | < >`
DiagnosticOutput	IN	NO	Option to save results to a file saved in the TPM Repository DIAGS share. "Y" or "N" (defaults to "N")
CommandOutput	OUT	n/a	Output of the executed command. Displayed within workflow logs and saved to DIAGS file if required.

4.6.7.3 Calls

This workflow is to be initiated via the Favourite Tasks page.

Note – This workflow must be defined as per instructions in Section 5 Creating and Sharing Favourite Tasks.

4.6.7.4 Pseudo Code

Uses the device.execute logical operation to execute a ps command

4.6.7.5 Logging

Logging is via TPM native logging facilities

4.6.7.6 Audit

POL00397162

Code	Message	Description / Cause
800		Workflow Initiated
801		Workflow Completed Successfully

4.6.8 HNGX_Purge_Aged_Diagnostics

4.6.8.1 Function

This workflow purges old files from the TPM Repository Diagnostics area for the specified Server.

4.6.8.2 Parameters

Parameter Name	IN/OUT	Mandatory	Description
TargetServer	IN	YES	TPM Device ID of managed computer to run command on (Identified within Favourite Tasks / **Select Target Computers** page when running task)
AgeDays	IN	NO	The age in days after which files within the *<TPMrepository>*/sysman/*<TargetServer>*/diags directory should be deleted (Null entry defaults to "31" days)

4.6.8.3 Calls

This workflow is to be initiated via the Favourite Tasks page.

Note – This workflow must be defined as per instructions in Section 5 Creating and Sharing Favourite Tasks.

4.6.8.4 Pseudo Code

Uses a local sriptlet to execute

- find $FileRepositoryPath$FilePath -type f -mtime +$AgeDays -ls -exec rm -v {} \;
  Logging

Logging is via TPM native logging facilities

4.6.8.5 Audit

POL00397162

Code	Message	Description / Cause
800		Workflow Initiated
801		Workflow Completed Successfully

4.6.9 HNGX_Purge_Aged_Logfiles

4.6.9.1 Function

This workflow purges old files from the TPM Repository Logfile area for the specified Server.

4.6.9.2 Parameters

Parameter Name	IN/OUT	Mandatory	Description
TargetServer	IN	YES	TPM Device ID of managed computer to run command on (Identified within Favourite Tasks / **Select Target Computers** page when running task)
AgeDays	IN	NO	The age in days after which files within the *<TPMrepository>*/sysman/*<TargetServer>*/logs directory should be deleted (Null entry defaults to “31” days)

4.6.9.3 Calls

This workflow is to be initiated via the Favourite Tasks page.

Note – This workflow must be defined as per instructions in Section 5 Creating and Sharing Favourite Tasks.

4.6.9.4 Pseudo Code

Uses a local sriptlet to execute

- find $FileRepositoryPath$FilePath -type f -mtime +$AgeDays -ls -exec rm -v {} \;
  Logging

Logging is via TPM native logging facilities

POL00397162

4.6.9.5 Audit

Code	Message	Description / Cause
800		Workflow Initiated
801		Workflow Completed Successfully

4.6.10 HNGX_Retrieve_Windows_Service_Status

4.6.10.1 Function

This workflow runs a **net start** command on a specified windows managed computer

This workflow can only be executed against computer definitions recorded as being a Windows platform

4.6.10.2 Parameters

Parameter Name	IN/OUT	Mandatory	Description
TargetServer	IN	YES	TPM Device ID of managed computer to run command on (Identified within Favourite Tasks / **Select Target Computers** page when running task)
DiagnosticOutput	IN	NO	Option to save results to a file saved in the TPM Repository DIAGS share. "Y" or "N" (defaults to "N")
CommandOutput	OUT	n/a	Output of the executed command. Displayed within workflow logs and saved to DIAGS file if required.

4.6.10.3 Calls

This workflow is to be initiated via the Favourite Tasks page.

Note – This workflow must be defined as per instructions in Section 5 Creating and Sharing Favourite Tasks.

4.6.10.4 Pseudo Code

Uses the device.execute logical operation to execute a net start command

POL00397162

4.6.10.5 Logging

Logging is via TPM native logging facilities

4.6.10.6 Audit

Code	Message	Description / Cause
800		Workflow Initiated
801		Workflow Completed Successfully

4.6.11 HNGX_Unload_Event_Log

4.6.11.1 Function

This workflow runs a custom **unloadevent**.exe command on a specified windows managed computer to retrieve Event Log information.

unloadevent.exe is delivered within common windows platform product WIN_SMG_INSTSUPPLIB.

This workflow can only be executed against computer definitions recorded as being a Windows platform

4.6.11.2 Parameters

Parameter Name	IN/OUT	Mandatory	Description
TargetServer	IN	YES	TPM Device ID of managed computer to run command on (Identified within Favourite Tasks / **Select Target Computers** page when running task)
TargetLog	IN	YES	Event Log to be searched (e.g. Application, System, Security)
EventSource	IN	NO	Message Source – Filters only those Events with the specified source (e.g. “Service Control Manager” or All Events (Default)

POL00397162

MaxDays	IN	NO	Maximum number of days to filter from (e.g. 2 – retrieves only Events within last 2 days) or No Max (Default)
DiagnosticOutput	IN	NO	Option to save results to a file saved in the TPM Repository DIAGS share. "Y" or "N" (defaults to "N")
CommandOutput	OUT	n/a	Output of the executed command. Displayed within workflow logs and saved to DIAGS file if required.

4.6.11.3 Calls

This workflow is to be initiated via the Favourite Tasks page.

Note – This workflow must be defined as per instructions in Section 5 Creating and Sharing Favourite Tasks.

4.6.11.4 Pseudo Code

Uses the device.execute logical operation to execute a `/sysmgt/tools/unloadevent.exe` command

4.6.11.5 Logging

Logging is via TPM native logging facilities

4.6.11.6 Audit

Code	Message	Description / Cause
800		Workflow Initiated
801		Workflow Completed Successfully

4.6.12 HNGX_Unload_Registry_Info

4.6.12.1 Function

This workflow runs a **reg query** command on a specified windows managed computer.

POL00397162

(note – displays all values under key)

This workflow can only be executed against computer definitions recorded as being a Windows platform

4.6.12.2 Parameters

Parameter Name	IN/OUT	Mandatory	Description
SourceServer	IN	YES	Hostname of the server housing the required file (Identified within Favourite Tasks / **Select Target Computers** page when running task)
RegKey	IN	YES	Registry Key name to be queried e.g. HKLM\SYSTEM\CurrentControlSet\Control\TimeZoneInformation)
DiagnosticOutput	IN	NO	Option to save results to a file saved in the TPM Repository DIAGS share. "Y" or "N" (defaults to "N")
CommandOutput	OUT	n/a	Output of the executed command. Displayed within workflow logs and saved to DIAGS file if required.

4.6.12.3 Calls

This workflow is to be initiated via the Favourite Tasks page.

Note – This workflow must be defined as per instructions in Section 5 Creating and Sharing Favourite Tasks.

4.6.12.4 Pseudo Code

Uses the device.execute logical operation to execute a `reg query` command

4.6.12.5 Logging

Logging is via TPM native logging facilities

4.6.12.6 Audit

POL00397162

Code	Message	Description / Cause
800		Workflow Initiated
801		Workflow Completed Successfully

POL00397162

4.7 UNX_ITM_LOGHOUSEKEEP: UNIX Campus Server Log File Housekeeping Process

This product is documented in the ITM LLD DEV/INF/LLD/0039

4.8 WIN_ITM_LOGHOUSEKEEP: WINDOWS Campus Server Log File Housekeeping Process

This product is documented in the ITM LLD DEV/INF/LLD/0036

POL00397162

4.9 RHL_TMF_TASKSITM

This product delivers the systems management requirements specified in POA high-level design DES/APP/HLD/0010 for web service stats collection and a daily restart of the moneygram web service

4.9.1 Background

Web Services were introduced in stages for Horizon as detailed below.

4.9.1.1 S60

The two web services implemented at S60 are known as PAF and DVLA. PAF provides a lookup service for address and postcode queries using data stored locally on the servers hosting the web service. The DVLA web service provides an interface to the DVLA for online authorisation of vehicle re-licensing.

4.9.1.2 S90

The web service implemented at S90 is known as APOP; it is a Web Service that provides the interface for on-line transactions between the Counter and Authorisation Services hosted on Oracle databases. Initially there will be one Authorisation Service for postal orders but in the future it is envisaged that further Authorisation Services will exist to handle different types of vouchers. The APOP Web Service provides routing to these different Authorisation Services through a common interface.

4.9.1.3 T40

The web service implemented at T40 is known as MoneyGram; it is a Web Service that provides the interface for on-line transactions between the Counter and the MoneyGram (MGRM) Host service located in Minneapolis, USA.

4.9.1.4 T60

The web service implemented at T60 is known as Telecoms (TCOM); it consists of two web service components; BBND Service Application and BKAC Service Application that interact with the ADSL Checker Service and BACS checker service respectively.

4.9.1.5 T82

An additional web service was implemented at T82 on the Telecoms server known as Kahala GDD Calculator Service Web Service (PGDD); it provides access from the smartpost and AP-ADC Counter to the Neopost Internet Service in order to provide customers with Guaranteed Delivery Dates (GDD) for international parcels being sent using the ParcelForce World Wide guaranteed delivery services.

4.9.1.6 HNG-X – INT14

Web Services Help Desk, Service Identifier (HDSK) and Online Counter Training, Service Identifier (OLCT) are new for HNG-X.

Fujitsu's Interstage Application Server is used to host web services. In the live environment TCOM is hosted on separate physical servers to APOP, PAF DVLA and MGRM though, by design, the implementation is not restricted in this way and it is possible to host multiple web services on the same server.

POL00397162

The implementation for S90 onwards specifies some service subscripting of folder structures to ensure such multi hosting is supported by this development.

The implementation for TCOM service hosts multiple web services, namely BBND, BKAC and PGDD.

4.9.2 Requirement

4.9.2.1 Collection of Statistics

All the Web Service Agents other than the Online Counter Training Service are required to collect statistics in a fixed format for input to the Data Warehouse.

Systems management requirements for the web services differ only in the configuration data so the implementation will be generic and sufficiently flexible to cater for the introduction of new web services with minimal reconfiguration.

Web services will generate statistical information on a regular basis throughout the day. At midnight, the logs will be moved to an archive directory so that they will be available for collection by Tivoli at 00:15.

On a daily basis, Tivoli must perform the following actions :

- Retrieve logs from all servers hosting the configured web services.
- Translate the retrieved files from DOS to Unix format.
- Transfer the files to the active data warehouse.
- Move files successfully transferred to a PROCESSED directory on the web server they originated from.
- Remove files from the PROCESSED directory that are older than N days; N should be configurable and set to 7.

Servers that are not accessible when Tivoli attempts to collect the data should be ignored and the data collected the following night.

The destination directory to be used on the data warehouse is $REPOSITORY/webstats/YYMMDD/trans, where YYMMDD is the date of the day before the day on which the retrieval is being run. Once all the files have been transferred, the destination directory should be renamed from "trans" to "load".

For HNG-X, the web service specification allows for multiple Web Services to be run on all servers. To enable this, the folder structure containing the web statistics includes the service name.

i.e. /opt/HngxAgents/WS/WSStats/*Svcid*/Stats/Archive. (where Svcid is the specific Authorisation Web Service)

4.9.2.2 MoneyGram Daily Restart

There is a requirement to run a schedule task on the MoneyGram server, which will run a script to stop/restart the Moneygram web service on a daily basis.

The restart should not be performed if a file named STOPPED exists in the relevant administrative folder for the web service.

POL00397162

The service is scheduled to be stopped / restarted at 07:00 each day if the service has not already been restarted in the last two hours. The service should be restarted using the commands WSstop.cmd followed by WSstart.cmd. If required, the time of the stop/start can been delayed from 07:00 via the configuration file.

4.9.3 Implementation

4.9.3.1 Statistics Collection

A configuration file will drive Tivoli tasks for the collection of web services statistics. This will be stored as /opt/sysmgt/cfg/websvc.conf on the TMR server. Configuration data in the file will include:

- A list of web services systems. The list relates to standard definitions within configuration file websvc.conf that specify the servers for a particular platform type. These are specified using the format <PLATFORM>SERVERS="<server01>,<server02>,<servernn>" where <PLATFORM> defines the platform, MGRM, APOP, DVLA, PAF, TCOM and HDSK, and <server01> <server02> <servernn> is a comma-delimited list of Tivoli endpoint names for that platform.
- A list of web services currently active on the each server. The list relates to web services currently active on each server. At HNG-X this is as follows:

Platform (System)	Web Service
DVLA (DVLA)	DVLA
PAF (PAF)	PAF
APOP (APOP)	APOP
MoneyGram (MGRM)	MGRM
Telecom (TCOM)	BBND, BKAC, PGDD
HDSK	HDSK

- The Tivoli endpoint name of the active data warehouse server.
- The value used for $REPOSITORY on the data warehouse. This is used to specify the initial part of the directory path where statistics will be transferred. The default for this, as used on live for other file transfers, will be /pw/stagonl/repository.
- The directory used to store required command files on web servers. The default for this, using Unix format for directory delimiters, will be /WS.
- The directory used to hold the archived web service statistics on multiple Web Service Hosting Servers. The default for this, using Unix format for directory delimiters, will be /opt/HngxAgents/WS/WSStats/*Svcid*/Stats/Archive.
- Various Endpoint test parameters
- The number of days after which statistical files in the PROCESSED directory will be removed; variable WEBSVC_DAYS_OLD_PARAM set to “7”.

The retrieval of files will be implemented using code based on that used in ad-hoc file retrieval. This uses a set of timing values to cancel transfer requests that are not responding. Similar values will be

POL00397162

included with websvc.conf with an additional value specifying the number of retries for failed file transfers.

There are differences in the file transfer method that can be used for transferring files from web service servers compared with transferring files to the data warehouse. Tivoli endpoints can only be the destination of a Tivoli software package and not the source. File transfer from the Web Server servers must therefore use functions provided in Tivoli's wadminep command to transfer individual files back to a Tivoli managed node. In this case, a Tivoli task will be used to list the archived statistics directory on the servers and copy the files to a temporary holding area on the TMR one by one. Once the files have been retrieved, a Tivoli software package can be used to transfer all retrieved files to the data warehouse in a single operation.

The associated Tivoli job will have the following process flow:

Step	Process Description
1	Validate that the "load" directory for the previous day does not already exist on the data warehouse. The presence of the directory indicates that processing has already successfully completed.
2	For each system listed in websvc.conf and each server name for the relevant system listed in websvc.conf, perform the following steps :
2.1	Retrieve a directory listing for the archived statistics directories on the current server.
2.2	Housekeep the processed directory for each web service on the server also create the processed dir if not exists
2.3	For each file in the retrieved directory listing, perform the following steps :
2.3.1	Transfer the file to a temporary holding directory on the TMR.
2.3.2	Translate the file to Unix format using the DOS2UNIX command.
2.3.3	If operations completed ok, record the server and file name so that the source file can be moved once transferred to the data warehouse.
3	If no files were transferred, report a failure status and exit.
4	Transfer the software package to the data warehouse.
5	If the file transfer has completed ok, perform the following steps :
5.1	Rename the data warehouse tmp directory so that destination directory includes yesterday's date/trans
5.2	Rename the data warehouse directory suffix from "trans" to "load".
5.3	Move source files that were successfully transferred to a processed sub-directory of the directory they exist in on each web server.
6	Erase files from the temporary holding directory on the TMR.
7	Report on overall status.

A log file for the job will be written to directory **/opt/sysmgt/websvc/logs**. This will use a rolling set of seven files named **stats_*Day*.log** where *Day* is the relevant day of the week.

POL00397162

The Tivoli job will be scheduled to run at 00:15 daily.

4.9.3.2 Scheduled Daily Restart (MoneyGram only)

The daily restart is a Tivoli task/job triggered via the Tivoli scheduler.

The scheduled job on the TMR is **MGRM_Websvc_Restart** and is scheduled for 07:00 daily. This job is created from the task **mgrm_daily_restart** under the task library **web_service_tasks** in policy region campus. The task calls a shell script **websvc_mgrm_restart.sh** that retrieves the MGRM server names and triggers the subtask **mgrm_daily_restart_ep** on each MGRM server.

The subtask running locally on the server hosting MGRM web services will consist of a script, **websvc_mgrm_daily_restart.sh**, and a configuration file, **websvc_mgrm_daily_restart.cfg** by default, installed in directory **/opt/sysmgt/tools.** These are delivered in product RHL_ITM_MGRM_CFG.

The configuration file for this monitor will contain:

- A space-delimited list of web services hosted by the server.
- The number of minutes to wait since the last restart of the web services.
- The base directory for the web services. This is the directory from which the commands **WSStop.cmd** and **WSStart.cmd** can be invoked.
- The number of minutes after the Tivoli scheduled 07:00 to stop/restart the service. This can be used to vary the stop/restart time on individual servers (optional)
- A subdirectory named **websvc_logs** will be used to store any log files generated by the script.

The log file, **websvc_mgrm_daily_restart.log**, is overwritten each time the script is invoked and includes the date and time at which the current run started so that correct operation of the task can be easily confirmed.

4.9.4 Installation

4.9.4.1 PSPID parameters

Required PSPID parameters are as follows:

<parameter prodver="RHL_TMF_TASKSITM" name="PAFSERVERS" value="lprppws001,lprppws002"/>

<parameter prodver="RHL_TMF_TASKSITM" name="DVLASERVERS" value="lprpdws001,lprpdws002"/>

<parameter prodver="RHL_TMF_TASKSITM" name="APOPSERVERS" value="lprpaws001,lprpaws002"/>

<parameter prodver="RHL_TMF_TASKSITM" name="MGRMSERVERS" value="lprpmws001,lprpmws002"/>

<parameter prodver="RHL_TMF_TASKSITM" name="TCOMSERVERS" value="lprptws001"/>

<parameter prodver="RHL_TMF_TASKSITM" name="HDSKSERVERS" value="lprphws001,lprphws002"/>

<parameter prodver="RHL_TMF_TASKSITM" name="WS_ACTIVE_DWH" value="lsgpems001"/

Note: WS_ACTIVE_DWH is a dummy value and should be replaced with the Tivoli Endpoint name of the Active Data Warehouse server.

POL00397162

All other values are the expected live values and should be replaced as required for development/test rigs.

The xml file containing these values should be:

<*ems server name*>_ parameters.xml and it should reside in the directory /opt/sysmgt/cfg prior to running the installation script.

4.9.4.2 Installation Process

- Check that all files have been copied to the directory /tmp/RHL_TMF_TASKSITM
- All files in this directory should be in UNIX format (if required use dos2unix *)
- Check permissions are correct as listed below.
- Install the tasks as follows (remaining in the /tmp/ RHL_TMF_TASKSITM directory) :
  - **sh install_websvc_tasks.sh** (this may take a few minutes)
- Check the logfile /opt/sysmgt/logs/RHL_TMF_TASKSITM.log.
  If successful, schedule the jobs as follows:
  - **sh install_websvc_scheduling.sh**

4.9.4.3 Logging

The installation logfile produced is /opt/sysmgt/logs/RHL_TMF_TASKSITM.log.

POL00397162

4.10 RHL_TPM_SUPPORTTASKS_BMC: Workflows for NCO Patrol Probe Configuration

4.10.1 Installation

TPM Workflows are delivered within a .tcdriver file (in essence a zip file)

.tcdriver files must be generated within the Eclipse Framework (APDE) within a TPM Environment.

The resultant compiled and zipped Workflows can be installed within any TPM environment using the TPM supplied **`${TIODIR}/tools/tc-driver-manager.sh`** command.

The .tcdriver and installation script should be copied to **`/tmp/RHL_TPM_SUPPORTTASKS`** prior to execution

No PSPID vars are required for this installation.

The TPM Server must be active prior to installation.

4.10.1.1 Pre-Requisites

The following products must be installed prior to this product;

Product	Description
RHL_NCO_PROBEBMC_CFG	Latest version of the Systems Management BMC Patrol Probe Config (Installed on EES Platforms) that contain the executable **`smg_nco_patrol_write.exp`**
RHL_TPM_WORKFLOWS_LIB	TPM System Logging Workflows

4.10.1.2 Installation Files

The following files are delivered for the installation:

File	Description
rhl_tpm_supporttasks_bmc.install	TPM Utility Workflows installation script
rhl_tpm_supporttasks_bmc.tcdriver	TPM Utility Workflows

POL00397162

4.10.1.3 Installation Process

The product is installed by running the **`rhl_tpm_supporttasks_bmc.install`** script.

When the installation script is run, it will perform the following actions –

- Test for installation logfile and directory & create if either found not to exist
- Test required .tcdriver file exists in temp install directory
- Test required TPM Target directory exists – create if not
- Copy .tcdriver file to TPM Target directory
- Attempt uninstallation of existing version of .tcdriver file (this warns to log if no previous version found – but we're ok to ignore if so as this must be a fresh install)
- Run install of supplied .tcdriver file

4.10.1.4 Logging

All installation steps and output from **`tc-driver-manager.sh`** command are logged to the standard logging directory **`/opt/sysmgt/logs`** within file **`RHL_TPM_SUPPORTTASKS_BMC.log`**

4.10.1.5 Post Installation

At the end of the installation the Installation files may be removed from the temp directory **`/tmp/RHL_TPM_SUPPORTTASKS_BMC`**

4.10.2 HNGX_Patrol_Config_Display

4.10.2.1 Function

This workflow displays, within the TPM GUI, the current patrol.def file housed on a specified EES platform server.

A diagnostic log file can be saved to the TPM Repository.

4.10.2.2 Parameters

Parameter Name	IN/OUT	Mandatory	Description
TargetServer	IN	YES	TPM Device ID of managed EES computer to run command on (Identified by hostname within Favourite Tasks / **Select Target Computers** page when running task)

POL00397162

DiagnosticOutput	IN	NO	Y or N to save returned Output to file within the diagnostic share. (default is N)
CommandOutput	OUT	N/A	Variable populated with command output. (Truncated to 3900 bytes in display – full output is available in log if saved)

4.10.2.3 Calls

This workflow is to be initiated via the Favourite Tasks page.

Note – This workflow must be defined as per instructions in Section 5 for Creating and Sharing Favourite Tasks.

4.10.2.4 Pseudo Code

Execute following command on remote server and store results in remote log file;

```
cat /opt/netcool/omnibus/probes/linux2x86/patrol.def
```

Copy remote log file to TPM Server repository

Read copied logfile and display via TPM GUI as CommandOutput

Remove copied log file if DiagnosticOutput = N

4.10.2.5 Logging

Remote command output is stored within **`/opt/sysmgt/logs/HNGX_Patrol_Config_Display.log`** overwritten at every execution.

TPM Diags file is stored is stored within the TPM Repository (accessible via the **DIAGS** share) subdirectory **`/sysman/`***`hostname`***`/diags`** with a filename **`HNGX_Patrol_Config_Display _`** ***`yyyymmddHHMMSS`***`.log`**

4.10.2.6 Audit

Code	Message	Description / Cause
800	HNGX_Patrol_Config_Display: *username*: REQUESTED: *ServerName* cat /opt/netcool/omnibus/probes/linux2x86/patrol.def	Workflow Initiated
801	HNGX_Patrol_Config_Display: *username*: COMPLETED: *ServerName* cat /opt/netcool/omnibus/probes/linux2x86/patrol.def	Workflow Completed Successfully

POL00397162

4.10.3 HNGX_Patrol_Config_Remove

4.10.3.1 Function

This workflow runs an **nco_patrol_remove** command on a specified EES platform.

Removes the definition of the BMC Patrol Agent on the host and port from the patrol.def file for the BMC Patrol Probe.

A diagnostic log file can be saved to the TPM Repository.

4.10.3.2 Parameters

Parameter Name	IN/OUT	Mandatory	Description
TargetServer	IN	YES	TPM Device ID of managed EES computer to run command on (Identified by hostname within Favourite Tasks / **Select Target Computers** page when running task)
HostName	IN	YES	Host to be removed from the definition file
Port	IN	YES	Port matching host to be removed from the definition file
DiagnosticOutput	IN	NO	Y or N to save returned Output to file within the diagnostic share. (default is N)
CommandOutput	OUT	N/A	Variable populated with command output. (Truncated to 3900 bytes in display – full output is available in log if saved)

4.10.3.3 Calls

This workflow is to be initiated via the Favourite Tasks page.

Note – This workflow must be defined as per instructions in Section 5 for Creating and Sharing Favourite Tasks.

4.10.3.4 Pseudo Code

Execute following command on remote server and store results in remote log file;

```
nco_patrol_remove hostname port
```

Copy remote log file to TPM Server repository

Read copied logfile and display via TPM GUI as CommandOutput

POL00397162

Remove copied log file if DiagnosticOutput = N

4.10.3.5 Logging

Remote command output is stored within **`/opt/sysmgt/logs/HNGX_Patrol_Config_Remove.log`** overwritten at every execution.

TPM Diags file is stored is stored within the TPM Repository (accessible via the DIAGS share) subdirectory **`/sysman/`*`hostname`*`/diags`** with a filename **`HNGX_Patrol_Config_Remove_`***`yyyymmddHHMMSS`***`.log`**

4.10.3.6 Audit

Code	Message	Description / Cause
800	HNGX_Patrol_Config_Remove: *username*: REQUESTED: *ServerName* nco_patrol_remove *hostname port*	Workflow Initiated
801	HNGX_Patrol_Config_Remove: *username*: COMPLETED: *ServerName* nco_patrol_remove *hostname port*	Workflow Completed Successfully

4.10.4 HNGX_Patrol_Config_Restore

4.10.4.1 Function

This workflow copies an existing **patrol.def.backup** file onto **patrol.def** on a specified EES platform.

A diagnostic log file can be saved to the TPM Repository.

4.10.4.2 Parameters

Parameter Name	IN/OUT	Mandatory	Description
TargetServer	IN	YES	TPM Device ID of managed EES computer to run command on (Identified by hostname within Favourite Tasks / **Select Target Computers** page when running task)
DiagnosticOutput	IN	NO	Y or N to save returned Output to file within the diagnostic share. (default is N)

POL00397162

CommandOutput	OUT	N/A	Variable populated with command output. (Truncated to 3900 bytes in display – full output is available in log if saved)

4.10.4.3 Calls

This workflow is to be initiated via the Favourite Tasks page.

Note – This workflow must be defined as per instructions in Section 5 for Creating and Sharing Favourite Tasks.

4.10.4.4 Pseudo Code

Execute following command on remote server and store results in remote log file;

```
file /opt/netcool/omnibus/probes/linux2x86/patrol.def.backup
```

If found execute following command on remote server and store results in remote log file;

```
cp -v /opt/netcool/omnibus/probes/linux2x86/patrol.def.backup
/opt/netcool/omnibus/probes/linux2x86/patrol.def
```

Copy remote log file to TPM Server repository

Read copied logfile and display via TPM GUI as CommandOutput

Remove copied log file if DiagnosticOutput = N

4.10.4.5 Logging

Remote command output is stored within **`/opt/sysmgt/logs/HNGX_Patrol_Config_Restore.log`** overwritten at every execution.

TPM Diags file is stored is stored within the TPM Repository (accessible via the DIAGS share) subdirectory **`/sysman/`***`hostname`***`/diags`** with a filename **`HNGX_Patrol_Config_Restore_`***`yyyymmddHHMMSS`***`.log`**

4.10.4.6 Audit

Code	Message	Description / Cause
800	HNGX_Patrol_Config_Restore: *username*: REQUESTED: *ServerName* cp -v /opt/netcool/omnibus/probes/linux2x86/patrol.def.backup /opt/netcool/omnibus/probes/linux2x86/patrol.def	Workflow Initiated
801	HNGX_Patrol_Config_Restore: *username*: COMPLETED: *ServerName* cp -v /opt/netcool/omnibus/probes/linux2x86/patrol.def.backup /opt/netcool/omnibus/probes/linux2x86/patrol.def	Workflow Completed Successfully

POL00397162

4.10.5 HNGX_Patrol_Config_Write

4.10.5.1 Function

This workflow runs an **nco_patrol_write** command on a specified EES platform.

Adds the definition of the BMC Patrol Agent on the host and port to the patrol.def file for the BMC Patrol Probe.

Password supplied with the Agent userid is encrypted with nco_patrol_encrypt.

Copies existing patrol.def file to backup version before changes made..

A diagnostic log file can be saved to the TPM Repository.

4.10.5.2 Parameters

Parameter Name	IN/OUT	Mandatory	Description
TargetServer	IN	YES	TPM Device ID of managed EES computer to run command on (Identified by hostname within Favourite Tasks / **Select Target Computers** page when running task)
HostName	IN	YES	Host to be added to the definition file
Port	IN	YES	Port matching host to be added to the definition file
UserID	IN	YES	UserID used to connect to the BMC Patrol Agent running on the added host
Password	IN	YES	Password for the UserID used to connect to the BMC Patrol Agent running on the added host (Password will be encrypted)
DiagnosticOutput	IN	NO	Y or N to save returned Output to file within the diagnostic share. (default is N)
CommandOutput	OUT	N/A	Variable populated with command output. (Truncated to 3900 bytes in display – full output is available in log if saved)

4.10.5.3 Calls

POL00397162

This workflow is to be initiated via the Favourite Tasks page.

Note – This workflow must be defined as per instructions in Section 5 for Creating and Sharing Favourite Tasks.

4.10.5.4 Pseudo Code

NOTE: SMG Utility smg_nco_patrol_write.exp expect script has been developed to allow the execution of nco_patrol_write and automatically responds to the prompts presented by this executable as passed as parameters to the calling script (expect was used as the UserID and Password prompts could not be redirected back to the executing shell for response via a shell script EOF block)

Execute following command on remote server to backup existing patrol.def file;

```
cp -v /opt/netcool/omnibus/probes/linux2x86/patrol.def
/opt/netcool/omnibus/probes/linux2x86/patrol.def.backup
```

Execute following command on remote server and store results in remote log file;

/opt/sysmgt/tools/smg_nco_patrol_write.exp *hostname port userid password*

Copy remote log file to TPM Server repository

Read copied logfile and display via TPM GUI as CommandOutput

Remove copied log file if DiagnosticOutput = N

4.10.5.5 Logging

Remote command output is stored within **/opt/sysmgt/logs/HNGX_Patrol_Config_Write.log** overwritten at every execution.

TPM Diags file is stored is stored within the TPM Repository (accessible via the DIAGS share) subdirectory **/sysman/*hostname*/diags** with a filename **HNGX_Patrol_Config_Write_*yyyymmddHHMMSS*.log**

4.10.5.6 Audit

Code	Message	Description / Cause
800	HNGX_Patrol_Config_Write: *username*: REQUESTED: *ServerName* /opt/sysmgt/tools/smg_nco_patrol_write.exp *hostname port userid password*	Workflow Initiated
801	HNGX_Patrol_Config_Write: *username*: COMPLETED: *ServerName* /opt/sysmgt/tools/smg_nco_patrol_write.exp *hostname port userid password*	Workflow Completed Successfully

POL00397162

4.11 RHL_TPM_DXC_REP_LOAD: User-Invoked TPM Workflows

4.11.1 Installation

TPM Workflows are delivered within a .tcdriver file (in essence a zip file)

.tcdriver files must be generated within the Eclipse Framework (APDE) within a TPM Environment.

The resultant compiled and zipped Workflows can be installed within any TPM environment using the TPM supplied `${TIODIR}/tools/tc-driver-manager.sh` command.

The .tcdriver and installation script should be copied to /tmp/RHL_TPM_DXCREPLOAD prior to execution

No PSPID vars are required for this installation.

The TPM Server must be active prior to installation.

4.11.1.1 Pre-Requisites

The following products must be installed prior to this product;

Product	Description
DXC_CRYPTO_CLIENT_INT14	NON SMG delivered DXC Client Software
RHL_TPM_UTILS	TPM Utility Workflows

4.11.1.2 Installation Files

The following files are delivered for the installation:

File	Description
rhl_tpm_dxc_rep_load.install	TPM Utility Workflows installation script
rhl_tpm_ dxc_rep_load.tcdriver	TPM Utility Workflows

4.11.1.3 Installation Process

The product is installed by running the rhl_tpm_ dxc_rep_load.install script.

When the installation script is run, it will perform the following actions –

- Test for installation logfile and directory & create if either found not to exist
- Test required .tcdriver file exists in temp install directory

POL00397162

- Test required TPM Target directory exists – create if not
- Copy .tcdriver file to TPM Target directory
- Attempt uninstallation of existing version of .tcdriver file (this warns to log if no previous version found – but we're ok to ignore if so as this must be a fresh install)
- Run install of supplied .tcdriver file

4.11.1.4 Logging

All installation steps and output from `tc-driver-manager.sh` command are logged to the standard logging directory `/opt/sysmgt/logs` within file `RHL_TPM_` DXCREPLOAD`.log`

4.11.1.5 Post Installation

At the end of the installation the Installation files may be removed from the temp directory /tmp/RHL_TPM_ DXCREPLOAD

4.11.2 HNGX_DXC_REP_LOAD

4.11.2.1 Function

This workflow will transfer a CM delivered code from the DXC to the TPM repository via the DXC client.

4.11.2.2 Parameters

None

4.11.2.3 Calls

This workflow is to be initiated via the Favourite Tasks page.

Note – This workflow must be defined as per instructions in Section 5 for Creating and Sharing Favourite Tasks.

4.11.2.4 Pseudo Code

Uses the device.execute logical operation to execute:

```
executedxc.sh -principal DELServer -client EPM -plan CMtoDelServer -direction
download -pass DXCPassword`
```

4.11.2.5 Logging

Logging is via TPM native logging facilities

4.11.2.6 Audit

POL00397162

Code	Message	Description / Cause
800		Workflow Initiated
801		Workflow Completed Successfully

POL00397162

5 Creating and Sharing Favourite Tasks

Please refer to DEV/APP/SPG/0006