



[TITLE * MERGEFORMAT]
[SUBJECT * MERGEFORMAT]



Document Title: [TITLE * MERGEFORMAT]

Document Reference: [DOCPROPERTY "Reference Number" * MERGEFORMAT]

Document Type: [DOCPROPERTY "Document Type" * MERGEFORMAT] (LLD)

Release: [DOCPROPERTY Release * MERGEFORMAT]

Abstract: [COMMENTS * MERGEFORMAT]

Document Status: [DOCPROPERTY Status * MERGEFORMAT]

Author & Dept: [AUTHOR * MERGEFORMAT]

External Distribution: None

Approval Authorities:

Name	Role	Signature	Date
Gill Jackson	Development		

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



[TITLE * MERGEFORMAT]
[SUBJECT * MERGEFORMAT]



0 Document Control

0.1 Table of Contents

[TOC \o "1-3" \h \z \t "POA Appendix Heading 1,1,POA Appendix Heading 2,2"]



[TITLE * MERGEFORMAT]
[SUBJECT * MERGEFORMAT]



0.2 Document History

Version No.	Date	Summary of Changes and Reason for Issue	Associated Change - CP/PEAK/PPRR Reference
0.1	20/06/2007	For review	

0.3 Review Details

Review Comments by :	Monday, 02 nd July 2007
Review Comments to :	[HYPERLINK "mailto:GRO" GRO] & [HYPERLINK "mailto:GRO" GRO]
Mandatory Review	
Role	Name
Solution Design/Infrastructure Design	Roger Barnes
System Test	Harjinder Hothi
SSC	Mik Peach
Optional Review	
Role	Name
Security	Bill Mambery
Business Continuity	Tony Wicks
Service Support	Peter Thompson
HNG-X Service Transition	Steve Godson
Data Centre Migration	Andy Tait & Brett Martin
SV&I Manager	Sheila Bamber
Tester	Hamish Munro
RV Manager	James Brett (POL)
VI Manager	Peter Rickson
Development Host Team Manager	David Harrison
Development Host Team Leader	David Pooley
Development Host Team Member	Anona Stevens
Development Host Team Member	Vishnuvardhan Ramachandran
Development Host Team Member	Mia Brittain
Development Host Team Member	Duncan Brown
Development Host Team Member	Steve Goddard
Development Host Team Member	Tatter Bandna
Development Host Team Member	Wing Pang
Reference Data Team	David Wilcox

©Copyright Fujitsu Services Ltd 2007

[SUBJECT * MERGEFORMAT]

Ref: DEV/APP/LLD/0092

Version: V0.1

Date: 22-Jun-07

Page No: 3 of 16

UNCONTROLLED IF PRINTED

[KEYWORDS * MERGEFORMAT]



[TITLE * MERGEFORMAT]
[SUBJECT * MERGEFORMAT]



Reference Data Team	Kevin McKeown
Issued for Information – Please restrict this distribution list to a minimum	
Position/Role	Name

(*) = Reviewers that returned comments

0.4 Associated Documents (Internal & External)

Reference	Version	Date	Title	Source
PGM/DCM/TEM/0001 (DO NOT REMOVE)	2.0	16/04/2007	Fujitsu Services Post Office Account HNG-X Document Template	Dimensions
DES/APP/HLD/0027	0.3	21/05/2007	TPS High Level Design	Dimensions
DES/APP/HLD/0020	0.3	08/06/2007	Branch DATABASE High Level Design	Dimensions

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
BRDB	Branch Database
HLD	High Level Design
HNG-X	Horizon Next Generation – Plan-X
PAN	Primary Account Number
PCI	Payment Card Industry
POL	Post Office Limited
POL FS	Post Office Limited Financial System
TPS	Transaction Processing Service
XML	eXtensible Mark-up Language

0.6 Glossary

Term	Definition
Hashed PAN	An obfuscation of a PAN based on a one-way hash algorithm. The first six and last four digits of a PAN are not changed by this obfuscation. All hashed PAN values are 19 characters in length , use base-64 characters and have a non-numeric character in



[TITLE * MERGEFORMAT]
[SUBJECT * MERGEFORMAT]



Term	Definition
	the 7 th character position.
Hydra	Phase covering the dual running of Horizon and HNG-X
PCI Compliance	The data held within the system conforms to the requirements of the Payment Card Industry.

0.7 Changes Expected

Changes

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 Copyright

© Copyright Fujitsu Services Limited 2007. All rights reserved. No part of this document may be reproduced, stored or transmitted in any form without the prior written permission of Fujitsu Services.



[TITLE * MERGEFORMAT]
[SUBJECT * MERGEFORMAT]



1 Introduction

1.1 Summary

The Complexity of Migration of a major application such as TPS necessitates an up to date and accurate detail of all tasks required in terms of order, precedence and completeness.

Database object changes will cause database dependencies to become invalid such as runtime views, stored and packaged procedures etc.

Code and data changes will necessarily mean Low level designs will need to be updated and test plans created.

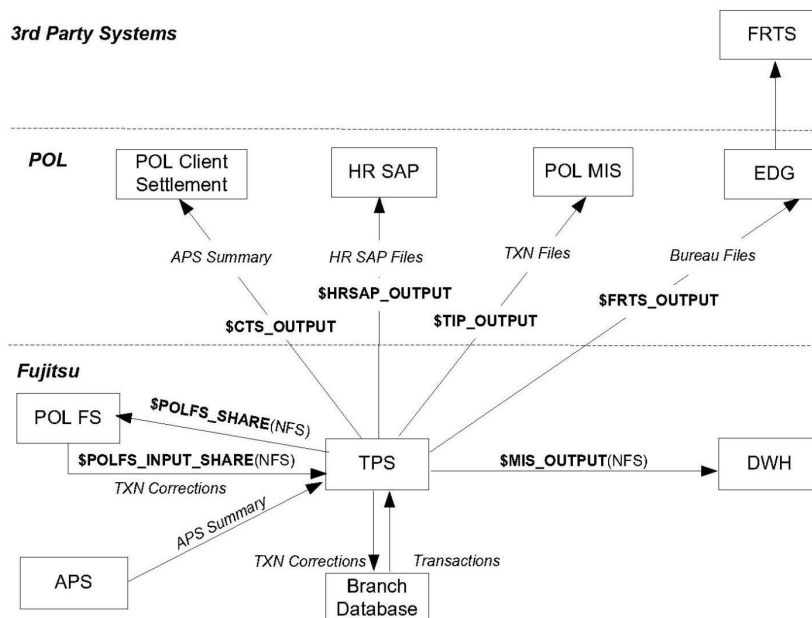
The aim of this document is to act as an establishment of scope and an attempt to minimise the risk inherent in migration and help guide the production of test plans and more importantly provide a sense of control over the period leading up to and indeed during Migration.

Five work areas can be readily identified from the TPS High Level Design and tasks listed beneath each area of work in order of chronological sequence required to complete.

1.2 Scope

The purpose of this document is to implement the application changes specified by Transaction Processing System High Level Design for HNG-X; reference DES/APP/HLD/0027.

The diagram below shows the scope of the Transaction Processing Service consisting of the TPS Host System, the Branch Database and a FTMS File Transfer System.





[TITLE * MERGEFORMAT]
[SUBJECT * MERGEFORMAT]



There are five main areas of work required:

- Riposte Name Removal
- PCI Support
- Application Components
- Migration
- Rollback Segments

1.3 Assumptions

It is assumed that all environment variables for the UNIX user tps are set as for Horizon with the exception of PATH and LD_LIBRARY_PATH which should reference Oracle10g libraries / directories. It is assumed that the TPS database has been migrated to Oracle10g on Solaris10.

2 Riposte Name Removal

To comply with copyright requirements for HNG-X the name RIPOSTE_NUM is replaced with JOURNAL_SEQ_NUMBER.

2.1 Tables

The following are the schema changes to replace any occurrence of RIPOSTE_NUM or RIPOSTE_MESSAGE_NUMBER with JOURNAL_SEQ_NUMBER, RIPOSTE_MESSAGE_PART_1_OF_2 and RIPOSTE_MESSAGE_PART_2_OF_2 with TXN_MESSAGE_PART_1_OF_2 and TXN_MESSAGE_PART_2_OF_2 respectively

Task #	Task	Action
1	Replace the use of the word Riposte in database field names HLD 4.1.11	<p>The name RIPOSTE_NUM is replaced with JOURNAL_SEQ_NUM in the following tables:</p> <p>TMS_RX_APS_TRANSACTIONS_pX TMS_RX_BDC_TRANSACTIONS_pX TMS_RX_CUT_OFF_SUMMARIES_pX TMS_RX_EFT_TRANSACTIONS_pX TMS_RX_EPOSS_EVENTS_pX TMS_RX_EPOSS_TRANSACTIONS_pX TMS_RX_NWB_TRANSACTIONS_pX TMS_RX_CASH_ACCOUNTS_pX¹ TMS_RX_OBCS_TRANSACTIONS_pX TMS_RX_OBCS_STATUSES_pX TMS_RX_STOCK_HOLDINGS_pX¹</p> <p>The name RIPOSTE_MESSAGE_NUMBER is replaced with JOURNAL_SEQ_NUM in the following tables:</p> <p>TMS_RX_COFA_TRANSACTIONS_pX¹ TMS_RX_COUNTER_TRAN_ERRORS TMS_RX_COUNTER_TRAN_ERRORS_ARC</p> <p>The name RIPOSTE_MESSAGE_PART_1_OF_2 and</p>



[TITLE * MERGEFORMAT]
[SUBJECT * MERGEFORMAT]



		<p>RIPOSTE_MESSAGE_PART_2_OF_2 is replaced with TXN_MESSAGE_PART_1_OF_2 and TXN_MESSAGE_PART_2_OF_2 in the following tables:</p> <p>TMS_HARVESTER_EXCEPTIONS_p¹ TMS_HARVESTER_EXCEPTIONS_ARC</p> <p>Where</p> <p>p is either 1 to 65, 65RC</p> <p>or¹ 1 to 64</p> <p>X is 'A' to 'B' (in all cases where p = 1 to 64).</p> <p><u>run tps_riposte_app.sh</u></p> <p>This script removes the occurrence of the label Riposte changes to HNGX</p> <p>Example:</p> <pre>alter table tms_rx_aps_transactions_1A rename column riposte_num to journal_seq_num;</pre>
2	<p>Create Special Agent 'H' views and Synonyms for all tables where RIPOSTE_NUM is replaced with JOURNAL_SEQ_NUM to be accessed by Horizon Agents</p> <p>HLD 4.1.11.1</p>	<p>CREATE OR REPLACE the following scripts:</p> <p><u>run tps_riposte_h_views.sh</u></p> <p>This script creates special agent 'H' views and synonyms</p> <p>Example:</p> <pre>Create or replace view h_tms_rx_aps_transactions_1A select journal_seq_num alias riposte_num, ... from tms_rx_aps_transactions_1A; Create or replace public synonym tms_rx_aps_transactions_1A for ops\$tps.h_tms_rx_aps_transactions_1A;</pre>
3	<p>Truncate redundant Tables</p> <p>HLD 4.1.4</p>	<p>TRUNCATE the following Tables:</p> <p>TMS_RX_CASH_ACCOUNTS_pX¹ TMS_RX_OBCS_TRANSACTIONS_pX TMS_RX_OBCS_STATUSES_pX TMS_RX_STOCK_HOLDINGS_pX¹ TMS_RX_COFA_TRANSACTIONS_pX¹ TMS_RX_COFA_SUMMARIES_pX¹</p> <p>Where</p> <p>p is either 1 to 65, 65RC</p> <p>or¹ 1 to 64</p> <p>X is 'A' to 'B' (in all cases where p = 1 to 64).</p> <p><u>run tps_truncate_tables.sh</u></p> <p>This script truncates selected tablenamees.</p> <p>Example:</p> <pre>truncate table tms_rx_cofa_transactions_1A;</pre>
4	<p>Recreate database views and dependencies</p>	<p>CREATE OR REPLACE the following views:</p>



[TITLE * MERGEFORMAT]
[SUBJECT * MERGEFORMAT]



		<p> XXX_RX_APS_TRANSACTIONS_p XXX_RX_APS_TRANSACTIONS XXX_RX_BDC_TRANSACTIONS_p XXX_RX_BDC_TRANSACTIONS XXX_RX_CUT_OFF_SUMMARIES_p XXX_RX_CUT_OFF_SUMMARIES XXX_RX_EFT_TRANSACTIONS_p XXX_RX_EFT_TRANSACTIONS XXX_RX_EPOSS_EVENTS_p XXX_RX_EPOSS_EVENTS XXX_RX_EPOSS_TRANSACTIONS_p XXX_RX_EPOSS_TRANSACTIONS XXX_RX_NWB_TRANSACTIONS_p XXX_RX_NWB_TRANSACTIONS XXX_RX_CASH_ACCOUNTS_p XXX_RX_CASH_ACCOUNTS XXX_RX_OBCS_TRANSACTIONS_p XXX_RX_OBCS_TRANSACTIONS XXX_RX_OBCS_STATUSES_p XXX_RX_OBCS_STATUSES XXX_RX_STOCK_HOLDINGS_p XXX_RX_STOCK_HOLDINGS XXX_RX_COFA_TRANSACTIONS_p XXX_RX_COFA_TRANSACTIONS XXX_RX_TRANSACTIONS TMS_HARVESTER_EXCEPTIONS </p> <p>Where</p> <p>XXX is TMS, SAV</p> <p>p is 1 to 64</p> <p><u>run tps_riposte_p_views.sh</u></p> <p>This script re-creates views and synonyms for all p (partitioned) tables that have been redefined by RIPOSTE replacement</p> <p><u>run tps_riposte_o_views.sql</u></p> <p>This script recreates the o(verall) views and synonyms for which their underlying table definitions have changed.</p> <p>Therefore total views:</p> <p>$2 \times 64 + 2 = 130$ views per table * 12 + 2 + 1 = 1564</p> <p>Example (pseudo):</p> <pre> SELECT table_name_p FROM user_tables WHERE table_name_p LIKE 'TMS_RX_%' ... CREATE OR REPLACE VIEW SAV_table_name_p AS SELECT * from table_name_pA or B where tps_system_parameters.table_set is A or B CREATE OR REPLACE VIEW TMS_table_name_p AS SELECT * from table_name_pA or B where tps_system_parameters.table_set is B or A </pre>
5	Recompile PL/SQL Modules and dependencies	<p>cd \$ORACLE_HOME/rdbms/admin</p> <p>sqlplus "/ as sysdba"</p> <p>@ utlrp.sql</p>



[TITLE * MERGEFORMAT]
[SUBJECT * MERGEFORMAT]



		Note: There should be no other DDL on the database while running the script. Not following this recommendation may lead to deadlocks.
6	Replace the use of the word Riposte in Pro*C Modules	<p>The name RIPOSTE_MESSAGE_PART_1_OF_2 and RIPOSTE_MESSAGE_PART_2_OF_2 is replaced with TXN_MESSAGE_PART_1_OF_2 and TXN_MESSAGE_PART_2_OF_2 in the following source files:</p> <p>TPSC209.h</p> <p>Create Special Agent 'H' views and Synonyms for all tables where RIPOSTE_NUM is replaced with JOURNAL_SEQ_NUM to be accessed by Horizon Agents in the following source files:</p> <p>TPSC209.pc</p> <p>The name RIPOSTE_NUM is replaced with JOURNAL_SEQ_NUM in the following source files:</p> <p>TPSC253.h TPSC254.h TPSC253.pc TPSC254.pc</p>
7	Recompile Pro*C Modules	<p>Compile the following modules:</p> <p>TPSC209 TPSC253 TPSC254</p>

2.2 Shell Script – tps_riposte_app.sh

The shell script, tps_riposte_app.sh removes the occurrence of the label Riposte by performing the following operation:

1. scan all the tables with RIPOSTE_NUM, RIPOSTE_MESSAGE_NUMBER, RIPOSTE_MESSAGE_PART_1_OF_2 and RIPOSTE_MESSAGE_PART_2_OF_2.
2. rename RIPOSTE_NUM, RIPOSTE_MESSAGE_NUMBER to JOURNAL_SEQ_NUMBER, RIPOSTE_MESSAGE_PART_1_OF_2 and RIPOSTE_MESSAGE_PART_2_OF_2 with TXN_MESSAGE_PART_1_OF_2 and TXN_MESSAGE_PART_2_OF_2 respectively.

2.3 Shell Script – tps_riposte_h_views.sh

The shell script, tps_riposte_h_views.sh creates special agent 'H' views and synonyms for all tables where RIPOSTE_NUM has been replaced with JOURNAL_SEQ_NUM (to be accessed by Horizon Agents). It performs the following:

1. scan all the tables with JOURNAL_SEQ_NUM
2. for each table found create an 'H' view and synonym

Note: run tps_riposte_app.sh first



[TITLE * MERGEFORMAT]
[SUBJECT * MERGEFORMAT]



2.4 Shell Script – tps_truncate_tables.sh

The shell script, tps_truncate_tables.sh truncates selected tablenames. It performs the following:

1. scan all the tables for selected patterns of table names
2. for each table found - truncate it

2.5 Shell Script – tps_riposte_p_views.sh

The shell script, tps_riposte_p_views.sh re-creates views and synonyms for all p (partitioned) tables that have been redefined by RIPOSTE replacement. It performs the following:

1. read the current Table set A or B
2. scan all the tables where RIPOSTE has been changed
3. for each table found recreate views and synonyms using current table set

Note: run tps_riposte_app.sh first

2.6 SQL Script – tps_riposte_o_views.sql

The SQL script, tps_riposte_o_views.sql recreates the overall views for which their underlying table definitions have changed

Note: run tps_riposte_p_views.sh first

3 PCI Support

The current TPS will be updated to meet the requirements for the Payment Card industry (PCI). The PAN field is replaced with a obfuscated value which is 19 alphanumeric characters in length.

3.1 Tables

The following are the schema changes to replace any occurrence of PAN stored as NUMBER with a data type of VARCHAR2 (19).

Task #	Task	Action
1	Change the datatype of the PAN database field HLD 4.1.9	The numeric datatype of PAN is changed to alphanumeric VARCHAR2 in the following tables: TMS_RX_EFT_TRANSACTIONS_pX TMS_RX_NWB_TRANSACTIONS_pX Where p is 1 to 65, 65RC X is 'A' to 'B' (in cases where p = 1 to 64). <u>run tps_pci_app.sh</u>



[TITLE * MERGEFORMAT]
[SUBJECT * MERGEFORMAT]



		<p>This script prepares PCI application changes to HNGX</p> <p>Example:</p> <pre>alter table tms_rx_eft_transactions_1A add(new_pan varchar2(19)); update tms_rx_eft_transactions_1A set new_pan = to_char(pan); commit; alter table tms_rx_eft_transactions_1A set unused column pan; alter table tms_rx_eft_transactions_1A drop unused columns; alter table tms_rx_eft_transactions_1A rename column new_pan to pan;</pre>
2	Recreate database views and dependencies	<p>CREATE OR REPLACE the following scripts:</p> <pre>XXX_RX_EFT_TRANSACTIONS_p XXX_RX_EFT_TRANSACTIONS XXX_RX_NWB_TRANSACTIONS_p XXX_RX_NWB_TRANSACTIONS XXX_RX_TRANSACTIONS</pre> <p>Where XXX is TMS, SAV p is 1 to 64</p> <p>Therefore total views: $2 \times 64 + 2 = 130$ views per table * 2 + 2 = 262</p> <p><u>run tps_riposte_p_views.sh</u> <u>run tps_riposte_o_views.sql</u></p>
3	Recompile PL/SQL Modules and dependencies	<pre>cd \$ORACLE_HOME/rdbms/admin sqlplus "/ as sysdba" @ utlrp.sql</pre> <p>Note: There should be no other DDL on the database while running the script. Not following this recommendation may lead to deadlocks.</p>
4	Change the datatype of PAN in Pro*C Modules	<p>The numeric datatype of PAN is changed to alphanumeric char pan[20] in the following source files:</p> <pre>TPSC270.h TPSC270.pc tip.pc</pre>
5	Recompile Pro*C Modules	<p>Compile the following modules:</p> <pre>TPSC270 TPSC287</pre>

3.2 Shell Script – tps_pci_app.sh

The shell script, tps_pci_app.sh performs the following operation:

1. Get the column_name, table_name from the user_tab_columns for the column name = 'PAN' and the data_type='NUMBER' AND data_precision=19.



[TITLE * MERGEFORMAT]
[SUBJECT * MERGEFORMAT]



2. Add a new column 'pci_pan' with the data type of varchar2(19) to the table as found in (1).
3. assign the column value of column_name as in (1) to the new column 'pci_pan'
4. Drop the column_name as in (1).
5. rename the pci_pan column to the column_name as in (1)

4 Application Components

To Conform to PCI Compliance the PAN value must NOT be held in clear form in any of the database.

Task #	Task	Action
1	PAN - zeroise it to obfuscate in Pro*C Modules	The pan number is zeroised by adding function ZeroPan, it zeroises all digits after the first 6, it should retain the last 4 in the following source files: tip.h tip.pc
2	Recompile Pro*C Modules	Compile the following modules: TPSC287

5 Migration

During the migration from Horizon to HNG-X, Hydra, the Horizon Agents provide data to the Branch Database via tables held in TPS. Transaction data from Horizon counters is required in the Branch Database so that when the counter migrates to HNG-X the historical data is available for reports and summaries.

In order to distinguish between transaction data harvested from the Branch Database and that from the Agent Harvesters a new field is added to each of the relevant tables. This field defaults to 'N' so that any records created by the Agent Harvesters are distinguishable from records created by the Branch database which will set it to 'Y'. The Branch database can then copy all records which are set to the default value of 'N'. The data is transferred to the counter either by Agent processes or via the Branch Database..A new field ACTIONED_IND_HNGX is required for the Branch Database to set once the record has been transferred.

The TPS Agent Harvester populates tables TMS_RX_HNGX_MIGRATION_DAY, TMS_RX_HNGX_MIGRATION_PREP with the Current Stock position of the Branch so that the Branch Database has the correct values at the point of migration to HNG-X.

Task #	Task	Action
1	Add the two new fields, transaction corrections Indicator and transaction Indicator Flag to database fields (to be used by interfaces from BRDB) HLD 4.1.10.1, 4.1.10.2	Add the alphanumeric ACTIONED_IND_HNGX column and the alphabetic ('Y' or 'N') TRANSACTION_IND_FLAG column to the following tables: TMS_RX_APS_TRANSACTIONS_pX TMS_RX_BDC_TRANSACTIONS_pX TMS_RX_EFT_TRANSACTIONS_pX TMS_RX_EPOSS_EVENTS_pX TMS_RX_EPOSS_TRANSACTIONS_pX TMS_RX_NWB_TRANSACTIONS_pX Where



[TITLE * MERGEFORMAT]
[SUBJECT * MERGEFORMAT]



		<p>p is 1 to 65, 65RC</p> <p>X is 'A' to 'B' (in cases where p = 1 to 64).</p> <p><u>run tps_mig_trans.sh</u></p> <p>This script modifies selected tablenames</p> <p>Example:</p> <pre>alter table tms_rx_eft_transactions_1A add(actioned_ind_hngx varchar2(1),transaction_ind_flag varchar2(1) default 'N');</pre>
2	Recreate database views and dependencies	<p>CREATE OR REPLACE the following scripts:</p> <p>XXX_RX_APS_TRANSACTIONS_p XXX_RX_APS_TRANSACTIONS XXX_RX_BDC_TRANSACTIONS_p XXX_RX_BDC_TRANSACTIONS XXX_RX_EFT_TRANSACTIONS_p XXX_RX_EFT_TRANSACTIONS XXX_RX_EPOSS_EVENTS_p XXX_RX_EPOSS_EVENTS XXX_RX_EPOSS_TRANSACTIONS_p XXX_RX_EPOSS_TRANSACTIONS XXX_RX_NWB_TRANSACTIONS_p XXX_RX_NWB_TRANSACTIONS XXX_RX_TRANSACTIONS</p> <p>Where</p> <p>XXX is TMS, SAV</p> <p>p is 1 to 64</p> <p>Therefore total views:</p> <p>$2 \times 64 + 2 = 130$ views per table * 6 + 2 = 782</p> <p><u>run tps_riposte_p_views.sh</u> <u>run tps_riposte_o_views.sql</u></p>
3	Recompile PL/SQL Modules and dependencies	<p>cd \$ORACLE_HOME/rdbms/admin</p> <p>sqlplus "/ as sysdba"</p> <p>@ utlrp.sql</p> <p>Note: There should be no other DDL on the database while running the script. Not following this recommendation may lead to deadlocks.</p>
4	Create database XML stock tables, users, roles, privileges	<p>CREATE TABLES</p> <p>TMS_RX_HNGX_MIGRATION_DAY, TMS_RX_HNGX_MIGRATION_PREP</p> <p><u>run tps_create_stock_tables.sql</u></p> <p>This script creates the XML stock tables</p> <p>Example:</p> <pre>create table tms_rx_hngx_migration_day (group_id number(6), trading_date date, insert_date date, migration_date, LOB (migration_data) STORE AS clob_store</pre>



[TITLE * MERGEFORMAT]
[SUBJECT * MERGEFORMAT]



		(TABLESPACE raw_data STORAGE (INITIAL 100K NEXT 100K PCTINCREASE 0) CHUNK 4 PCTVERSION 10 INDEX clob_index (TABLESPACE raw_index));
--	--	--

5.1 Shell Script – tps_mig_trans.sh

The shell script, tps_mig_trans.sh modifies selected table names. It performs the following:

1. scan all the tables for selected patterns of table names
2. for each table found modify it - add two new transaction column names

5.2 SQL Script – tps_create_stock_tables.sql

The SQL script, tps_create_stock_tables.sql creates the XML stock tables

6 Rollback Segments

The rollback of database transactions will use Oracle 10g AUTO Undo management, instead of Rollback Segments. This section describes and implements the steps to convert the rollback segments to AUTO Undo management.

1. Find all the rollback segments
2. Drop all the rollback segments except for system
3. drop rollback tablespace
4. create new undo tablespace
5. add undo_management = AUTO, undo_tablespace=UNDO_DRS to inittps.ora and comment out the rollback_segs param
6. bounce the database

'tps_app_ug.sh' shell script will perform 1 to 4 of the above steps and used the existing rollback segment datafiles for undo_tablespace

Task #	Task	Action
1	The rollback of database transactions will use Oracle 10g AUTO Undo management – Check and fix Oracle initialisation parameters, remove 'old' style rollback segment usage and Pro*C Modules usage HLD 4.1.3	Prepare application upgrade to HNGX 1) remove all rollback segments 2) drop the rollback segment tablespace 3) create Undo segment without datafile sizing (raw files): <u>run tps_app_ug.sh</u> This script prepares Undo Management to HNGX Check spfiletps.ora (init.ora) settings: undo_management=AUTO undo_tablespace=UNDO_TPS restart database



[TITLE * MERGEFORMAT]
[SUBJECT * MERGEFORMAT]



		remove EXEC SQL SET TRANSACTION USE ROLLBACK SEGMENT from the following source files: TPSC207.pc TPSC280.pc TPSC281.pc
2	Recompile Pro*C Modules	Compile the following modules: TPSC207 TPSC280 TPSC281

6.1 Shell Script – tps_app_ug.sh

The shell script, tps_app_ug.sh prepares automatic undo management upgrade to HNGX.

It performs the following:

1. removes all rollback segments
2. drop the rollback segment tablespace
3. create Undo segment