PHKEEP\_Specification&Unit Test Plan

06-06-2017

#### Document information

|  |  |  |
| --- | --- | --- |
| Document owner | Tom Parkin | |
| KeyedIn Project Reference | Not Applicable | |
| Prepared by | Ranjith Gopalankutty | |
| Contact details | [Ranjith.Gopalankutty@Boots.co.uk](mailto:Ranjith.Gopalankutty@Boots.co.uk) | |
| Document classification  (indicate with a X in the right hand column) | Key Project Document | X |
| Supporting Project Document |  |
| Process Document |  |
| Operating Procedure |  |
| Quality Standard |  |
| Document location and filename | T:\Boots The Chemists\IS & T\Service\Improvement\Technical\Development\SUTPs\PHKEEP | |
| Version number | 1.1 | |
| Document status | Revised after checking the status with inctactix | |

Document history

| **Version** | **Date** | **Change** | **Approved by** |
| --- | --- | --- | --- |
| V1.0 | 06-06-2017 | New program to housekeep planner files SRMOD,SRSXF and SRPDF | Tom Parkin |
| *V1.1* | *19-07-2017* | *Updated the program further after the meeting with Inctactix so that , it checks the SRPOG records against SRMAP. If its not present in SRMAP can delete it* | *Tom Parking* |

Distribution

| **Name** | **Title** |
| --- | --- |
| Ranjith Gopalankutty | **Developer / Author** |
|  | **Technical Lead** |
| Anne Greener | **Release manager** |
| Paul Bowers | **Retail solution architect** |

Related documents (if any)

| **Document name** | **Location** |
| --- | --- |
|  | *If appropriate add this as a hyperlink* |
|  |  |
|  |  |

Contents

[Document information 1](#_Toc384641664)

[Document history 1](#_Toc384641665)

[Distribution 2](#_Toc384641666)

[Related documents (if any) 2](#_Toc384641667)

[1. Preface 3](#_Toc384641668)

[1.1 Document Purpose 3](#_Toc384641669)

[1.2 Scope of document 3](#_Toc384641670)

[2. Overview 3](#_Toc384641671)

[2.1 Executive Summary 3](#_Toc384641672)

[2.2 Requirements Matrix 3](#_Toc384641673)

[2.3 Context Diagram 3](#_Toc384641674)

[3. Specification 3](#_Toc384641675)

[3.1 Overview 3](#_Toc384641676)

[3.2 Detail 3](#_Toc384641677)

[3.3 Error Reporting 3](#_Toc384641678)

[4. Unit Test Plan 3](#_Toc384641679)

[4.1 Overview 3](#_Toc384641680)

[4.2 SONAR QA Report 3](#_Toc384641681)

[4.3 Environment 3](#_Toc384641682)

[4.4 Unit Tests 3](#_Toc384641683)

1. Preface
   1. Document Purpose

The purpose of this document is to describe specification and unit test plan for the new bespoke housekeeping program PHKEEP.286 to housekeep expired planner records from SRMOD, SRPDF and SRSXF

* 1. Scope of document

This document will detail only the specification and unit test plan only for the new single moulded housekeeping program PHKEEP.286. No existing application or program will be changed as part of it. Also no other files will be changed as part of this program.

1. Overview
   1. Executive Summary

There was a defect with program SRP10. It supposed to house keep the expired planner records from SRPOG, SRMOD, SRSXF and SRPDF files. SRPOG is the master planner file SRMOD is module, SRSXF is notch data and SRPDF is the planner descriptor file. SRMOD, SRPDF and SRSXF files depends on the master planner file SRPOG to build the keys for them. The defect was that, SRP10 program was housekeeping only records from SRPOG and leaving behind other 3 files. Over time. Records will be house kept from SRPOG and Records will get piled up in other 3 and causing file full issues. This resulted in new planner update not able to add in to planner files. Defect has been found and fixed by AppsMgmt. Change is live in all stores, but one issue found afterwards that program was not able to delete the historical expired records from planner files. PHKEEP program is custom written to delete the expired records from 3 planner files SRMOD, SRPDF and SRSXF.

*Further enhancement done to the code so that instead of SRPOG it considers SRMAP as master file. After reading SRPOG it will check the same POGDB in SRMAP, if the record is not present in SRMAP it can be deleted. For any planners to be visible in store controllers, the SRMAP should have the POGDB for SRP04 to link the item code with planner (Live and pending). If it’s missing then we can ignore it and delete the same. By doing so, could see at least 300 to 400 planners are obsolete in most of the stores*

* 1. Requirements Matrix

* 1. Context Diagram

1. Specification
   1. Overview

Currently SRP10 application builds the housekeeping key from SRPOG, using the SRPOG key it builds the key for files SRMOD,SRPDF and SRSXF, Currently the expired records are only in SRMOD,SRPDF and SRSXF as it could not find the key, expired records are still lying in the controller files. To give a quantification when tested the application with pulling store 0023 files below is the difference

15:25:10-Read 6341 SRMOD records Written 6341 In to New file

15:25:10-Read 37558 SRSXF records Written 17054 In to New file

15:25:10-Read 5090 SRPDF records Written 2225 In to New file

*After the discussion with Inctactix , its been found that the number of planner records between inctactix and controller side are not matching. On further probe its been found that its due to the way how planner housekeeping is set in controller causing the issue plus not sending deletion records for obsolete planers from inctactix causing further issues. So instead of SRPOG, SRMAP file is being taken as the master file as it is important for any planner to be visible in controller to have a matching record in SRMAP*

* + 1. QA/Support Files Updated - Checklist

| **Description** | **Name** | **Updated** |
| --- | --- | --- |
| **Report file numbers used in error reporting** | REPORT.TXT & REPORT.NUM | No |
| **List of Pipe usage** | PIPES.TXT | No |
| **Event numbers used in error reporting** | EVENT.TXT | No |
| **List of Programs and description** | PROGRAMS.DOC | No |
| **Unit Test Plan** | PHKEEP.286 | Yes |

.

* 1. Detail

It’s been confirmed that, SRPOG file have no expired record by strolling through the expiry dates of the planner in some of the store where file full issue with other 3 files are happening. Also could confirm the same from inctactix. So the application is written considering SRPOG does not have any expired records

Program will read SRPOG using the block wise read method, while reading it will build an array of keys of planner records. Similarly the file read will happen against other 2 files except SRSXF. As SRSXF record count is normally huge, it is difficult to sort it using array and index. Instead a direct comparison of the records will happen between SRPOG and SRSXF and matching record will be written to a direct file in W: drive. For SRPDF and SRMOD similar arrays will be written with its keys. Once the reading is finished comparison operation will start, SRPOG array record will compared against SRMOD array and the matching records will be written to direct file. Similar operation will be done for SRPDF as well. Once comparison and writing operation has been finished. Direct files will be keyed up turbo utility.

After creating new keyed files original D:/ADX\_UDT3 files will backed in to D: drive and new files created will be placed in D:/ADX\_UDT3 with a distribution type of 3.

Program will update each step PHKEEP.LOG file and success cases will be updated in PHKEEP.OK with all EEEE flags, anything other.

*Program will read SRPOG as like in the original version of the code but will check the same records presence in SRMAP as well. If the planner record is not present in SRMAP then we can delete that from SRPOG. As if a planner is not present in SRMAP means its an obsolete planner.*

3.2.1 SEQUENTIAL.READ

All sequential read subroutine will perform block read of files and updates the key valued in the respective array counters except for SRSXF. In the case of SRSXF subroutine, saving the key value to array will not happen due the number of records.

*On top of the sequential read a comparison operation between SRPOG and SRMAP will happen to ensure there not more obsolete planners in a store.*

3.2.2 COMPARE

Compare subroutine will compare the records between SRPOG and other files. Matching records will be written to W:/ drive direct files by calling

* 1. Error Reporting

Error tracking will be done using two method any run time error will be captured using ERROR.DETECTED subroutine and event will be captured in event log using ERROR.LOG subroutine. Also anything other than E flag will be considered as failure from a store. So that analysis can be done on the same. File missing, file opening or read errors will be updated in PHKEEP.LOG sequential file in W: drive. Complete run status can be obtained from PHKEEP.ERR/PHKEEP.OK file. Unless until all files are completely successful E flags will not be written. Renaming for .ERR file to .OK will not happen.

1. Unit Test Plan
   1. Overview

Below unit test scripts will cover the test coverage required to make sure PHKEEP is executing the functionality as expected and not changing any other existing fi les or functionalities

* 1. SONAR QA Report

* 1. Environment
* 17A single site controller
* 17A MCF environment
* 17B single & multiple controller
* HHT (Batch & RF devices)
* CHKPLN.286 (utility to dissect the planner information correctly)
  1. Unit Tests

**Controller & HHT**

| **Test**  **No.** | **Test Description / Execution steps** | **Expected Result** | **Pass or**  **Actual Result if Fail** |
| --- | --- | --- | --- |
| 1 | | Step 1:  Navigate to c:/adx\_upgm  Step2:  Run CHKPLN in the directory by giving parameter as all  Step 3:  Once CHKPLN is completed save the files in to local desktop  Step 4:  Checks for the existence of planner related files in SRMOD,SRPDF and SRSXF | Application should complete fine and dump the results in a readable format in .CSV files | Pass |
| 2 | | Step 1:  Choose option number 1 from main menu  Step2:  Press F9 from the menu option  Step 3:  Select option 7 and press enter  Step 4:  Take live planners and sales plan | Planner results should display fine and should be able to navigate between different planner | Pass |
| 3 | | Step 1:  Place the PHKEEP program in C:/root  Step2:  Execute the program from the command prompt  Step 3:  Program execution should complete fine and should display a status like in the attached screen shot  Step 4:  Program will execute and display the status of read and write with a counter in the screen |  | Pass |
| 4 | | Step 1:  After executing PHKEEP successfully check the files in W: drive  Step 2:  There should be two files named PHKEEP.OK and PHKEEP.LOG  Step 3:  Read through PHKEEP.LOG | PHKEEP.LOG should display the execution status with total number of read and write operation performed on each file | Pass |
| 5 | | Step 1:  Once the PHKEEP executed successfully navigate in to W: drive  Step 2:  Check for the existence of PHKEEP.OK file  Step 3:  Try to open PHKEEP.OK file | PHKEEP.OK should be having the flag status as “EEEE” | Pass |
| 6. | | Step 1:Rename SRPOG.DAT file in D:/ADX\_UDT3 Step2:  Execute PHKEEP from C:/ DRIVE  Step 3:  Program should not complete and throw error message | Program will end in between and will throw appropriate error message in the screen | Pass |
| 8 | | Step1:  After getting the error please navigate to W: drive and check for file PHKEEP.LOG and PHKEEP.OK  Step 2:  Open PHKEEP.LOG and check if same error message is repeated in the log file | Program will write the error message in PHKEEP.LOG | Pass |
| 9 | | Step1:  After the failure check W: drive for PHKEEP.OK file  Step 2:  PHKEEP.OK file should not be present | PHKEEP.OK should not be present in the W: drive instead PHKEEP.ERR will be present in the file with an X flag writing at the first position | Pass |
| 10 | | Step 1:  Repeat the above steps for file SRMOD,SRPDF and SRSXF  Step 2:  Respective failure flags should be written in to respective positions in PHKEEP.ERR file  . | The screen display, PHKEEP.LOG and PHKEEP.ERR will display the error status accordingly. | Pass |
| 11 | | Step 1:  After executing PHKEEP run CHKPLN against the new set of planner files against SRPOG,SRMOD,SRPDF and SRSX  Step2:  Compare the .CSV files against the old CSV files generated before running PHKEEP  Step 3:  The planner files which is not present in SRPOG should not have any presence in other 3 files | Except for SRPOG all other files should be reduced in count in terms of number of records and that should be of expired planners | Pass |
| 12 | | Step 1:  After executing PHKEEP try to access planner files using 1-F9-7 ( any planner type)  Step 2:  Try to enter in to any of the planners  Step 3:  Try to search for planners or planner item | Results should be displayed without any issues. Planner search should work fine as like before | Pass |
| 13 | | Step 1:  After executing PHKEEP run CHKPLN against the new set of planner files against SRPOG,SRMOD,SRPDF and SRSX  Step2:  Compare the .CSV files against the old CSV files generated before running PHKEEP  Step 3:  The SRPOG planner count should not be changed and same number of planner should appear in controller files | As there is no issue with SRPOG | Pass |
| 14 | | Step 1:  Execute PHKEEP in an MCF environment  Step 2:  Make sure the program executes fine and writes PHKEEP.OK with all EEEE flags  Step 3:  Once program executed fine check alternate controller | Alternate controller should be updated with 4 files with latest time stamp | Pass |
| 15 | | Step 1:  On successful execution verify below checks  Step 2:  No other SR files exist in W: drive apart from PHKEEP.OK and PHKEEP.LOG | All intermediate and direct files should be deleted from W: drive | Pass |
| 16 | | Step 1:  Execute PHKEEP and on successful execution check D:/drive | Original D:/ADX\_UDT3 will be backed up in D: drive | Pass |
| 17 | | Step 1:  After executing the PHKEEP run CHKPLN and save the data in to .CSV fi les  Step2:  Compare 3 files data with old .CSV files | No valid planners should be deleted and any records which is present in 3 files should have master file reference in SRPOG | Pass |
| 18 | | Step1:  Navigate to C:/ADX\_UPGM  Step2:  Run the REFBUILD application | Planner files should be built correctly as before | Pass |
| 19 | | Step 1:  Once the planner files built for batch devices run a reference data reload process from a batch device | Reference data reload should complete fine and planner search should work fine from HHT device. | Pass |
| 20 | | Step 1:  Login to an RF device  Step 2:  Search for the planners as like in the controller and make sure the information is correct | There should not be any change between controller and HHT planner files | Pass |