HHT Date and Time sync issue (PRB0045752)

Specification & Unit test plan

27/10/2015

#### Document information

|  |  |  |
| --- | --- | --- |
| Document owner | Alam Khan | |
| KeyedIn Code | Not Applicable | |
| Prepared by | Arun Venugopalan | |
| Contact details | [Arun.Venugopalan@boots.co.uk](mailto:Arun.Venugopalan@boots.co.uk) | |
| Document classification  (indicate with a X in the right hand column) | Key Project Document |  |
| Supporting Project Document | X |
| Process Document |  |
| Operating Procedure |  |
| Quality Standard |  |
| Document location and filename | T:\Boots The Chemists\IS & T\Service\Improvement\Technical\Development\SUTPs\Problem & Investigations\HHT\PRB0045752 | |
| Version number | 1.0 | |
| Document status | Final | |

Document history

| **Version** | **Date** | **Change** | **Approved by** |
| --- | --- | --- | --- |
| 1.0 | 27/10/2015 | Final version |  |

Reviewer/Authorisation

| **Name** | **Title** | **Responsibilities** | **Authorisation Signature** |
| --- | --- | --- | --- |
| Alam Khan | **Service Manager, Retail Application Management** | Reviewer/Authoriser |  |
| Jamie Thorpe | **Solution Design** | Reviewer/Authoriser |  |
| Paul Bowers | **Manager Solution Design** | Reviewer/Authoriser |  |
| David Artiss | **Senior Technical Analyst** | Reviewer/Authoriser |  |
| Ranjith Gopalankutty | **Technical Analyst** | Reviewer/Authorizer |  |

Distribution

| **Name** | **Title** |
| --- | --- |
| Anne Greener | **Release Manager** |
| Mark Wood | **Project Manager** |
|  | **Test Manager** |

Related documents (if any)

| **Document name** | **Location** |
| --- | --- |
| Nil | Nil |

Contents

[Document information 1](#_Toc433731748)

[Document history 1](#_Toc433731749)

[Reviewer/Authorisation 2](#_Toc433731750)

[Distribution 2](#_Toc433731751)

[Related documents (if any) 2](#_Toc433731752)

[1. Preface 4](#_Toc433731753)

[1.1 Document Purpose 4](#_Toc433731754)

[1.2 Scope of document 4](#_Toc433731755)

[2. Overview 4](#_Toc433731756)

[2.1 Executive Summary 4](#_Toc433731757)

[2.2 Requirements Matrix 4](#_Toc433731758)

[2.3 Flow Diagram 5](#_Toc433731759)

[3. Specification 6](#_Toc433731760)

[3.1 Overview 6](#_Toc433731761)

[3.2 Detail 7](#_Toc433731762)

[3.3 Error Reporting 14](#_Toc433731763)

[4. Unit Test Plan 15](#_Toc433731764)

[4.1 Overview 15](#_Toc433731765)

[4.2 Environment 15](#_Toc433731766)

[4.3 Unit Tests 15](#_Toc433731767)

1. Preface
   1. Document Purpose

The purpose of this document is to describe the changes required in HHT and Transact application to resolve the HHT Date and Time synchronization issue (PRB0045752) with controller during cold reboot and battery drained out scenarios.

* 1. Scope of document

This document will detail only the changes to be done for HHT applications in class APPCONTAINER.VB and a new class SOCKETCONNECTIONMGR.VB will be added for socket communication and in Transact application in TRXBASE.C module will be changed to fix HHT Date and Time Sync issue (PRB0045752).

PPC and MC55RF devices are not in the scope of this project. Date/time sync issue is only affecting MC70 devices at the moment. RF devices are getting the latest date/time update using existing SM applications. As there is no concept of reference data download in RF, devices date/time mismatch for few seconds doesn’t affect the business functionalities, So any possible change for RF is ruled out.

1. Overview
   1. Executive Summary

**Overview of HHT Date and Time Sync issue (PRB0045752)**

HHT batch devices revert to the incorrect time and date after a cold reboot or battery has been drained down. This forces the stores to reload reference data. Also due to date/time mismatch reference data pulling is not perfect in those scenarios. As a result stores misses out SM activities in the morning time.

* 1. Requirements Matrix

| **Number** | **Requirement** | **Solution Reference** |
| --- | --- | --- |
| 1 | Automate the synchronisation of HHT Date and Time to avoid repeated reference data reload process | 3.2.1 |

* 1. Flow Diagram

McDowloader Triggered

HHT

Hard Rebooted / Reboot after battery drain out

Sends Sign On Request (SOR) message with dummy User ID send to Transact

(Hardcoded)

Transact receives the SOR message and checks the Application ID, if the Application ID is 6(hardcoded) skips the user authentication and builds the Sign on Response (SNR) message with controller current Date and Time.

SNR send back to HHT

HHT device uses the date and time in the SNR message and updates the HHT device date and time to the same.

Proposed solution

Checks the HHT device time and triggers the Reference Data download.

low be updated

1. Specification
   1. Overview

Currently the date/time update happens in HHT only with a valid sign on request (SOR) while logging into any of the applications i.e Goods In, Goods Out and Shelf Monitor. It is because the HHT application is designed in such a way that while logging into any of these applications from the Modal Day screen, the device sends a Sign On Request (SOR) to the TRANSACT. TRANSACT responds to the SOR message with a Sign On Response Message (SNR). It is along with this SNR message that the current time and date of the controller is sent to the HHT device. HHT device then uses the received time and date to update its time and date.

This method is proved to be not practical and ineffective, as the reference data reload from controller happens before logging into these Shelf monitoring activities. When date/time is wrong reference data won’t be updated, in those scenarios sign in to shelf monitoring won’t be perfect, so store users have to correct data/time at first and then go ahead with reference data reload process to get it corrected. The suggested solution is an automatic way of tackling it.

The recommended solution will update the time and date of batch devices (MC70, MC55), as soon as the device is Switched on. McDownloader.exe gets triggered every time the batch device gets booted up. A new time-check needs to be introduced in the McDownloader.exe, which will be triggered before the file version-check is executed. That should remove the issue of the files being reloaded when not required.

**Dimensions Stream**:

| **Name** | **Description/Summary of Change** | **Archived** | **Handover**  **Deployable** |
| --- | --- | --- | --- |
| TRANSACT.286 | Client/Server socket program | Yes | Yes |
| TRXBASE.C | Message handling file | Yes | No |

**SVN Details:**

| **Name** | **Description/Summary of Change** | **Archived** | **Handover**  **Deployable** |
| --- | --- | --- | --- |
| SocketConnectionMgr.vb | Socket connection class file for HHT | Yes | No |
| AppContainer.vb | Main Application container class | Yes | No |
| McDowloader.EXE | McDowloader application exe file | Yes | Yes |

* + 1. QA/Support Files Updated - Checklist

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

* 1. Detail

The problem fix includes changes to both HHT and Transact application.

* + 1. TRANSACT Changes

In transact part we need to change the TRXBASE.C file.

**TRXBASE.C Change:**

The SignOn() function in the TRXBASE.C file which is responsible receiving the SOR and generating the corresponding response via SNR, will be modified as part of the solution. The SOR request will contain a dummy User Id (XXX) and password (XXX). As it is a completely automated process so there will not be any user authentication process as considering to other SOR requests. When Transact receives a SOR message with dummy User Id and the Application ID holding the value "006" it will Skip the user authentication process and just build the SNR message with current controller date and time of the controller and will send back to the HHT device in the form of SNR message. All the other fields will be supplied with hardcoded dummy values as these values do not have any significance to this solution and will be ignored.

In the SOR function we need to change an IF condition as mentioned below:

**Current Code**:

if ((usrrc == RC\_OK) {

**Changed Code**:

To check if Application Id received is 6 no need to authorise the user just build an SNR with current controller Date and Time.

if ((usrrc == RC\_OK) || (wAppID == 6)) {

* + 1. HHT Changes

McDownloader.exe is the first application to be triggered when the batch devices is rebooted. The McDownloader.exe solution is responsible for triggering the Reference Data Reload process. A new GetTime () function will be introduced in the McDownloader.exe just before triggering the Reference Data Reload function. This function will be responsible for sending the SOR with new dummy USERID and PASSWORD will be send to Transact. SNR message will be received as the response from Transact, which will contain current date and time of the controller and this value is set as the HHT device's current date and time.

**Code Changes in HHT:**

AppContainer.vb class changes:

'Automated Date and Time function call

GetTime("XXX", "XXX")

**GetTime() function**: Function will create a new SOR message with dummy user name and password.

Private Function GetTime(ByVal UserName As String, ByVal Password As String) As Boolean

'Declaring the dummy SOR record varibales

Dim strFreeMem As String = Nothing

Dim strMacID As String = "000000000000"

Dim strDeviceType As String = "R"

Dim strAppId As String = "006"

Dim strIPAdd As String = "XXX.XXX.XXX.XXX"

Dim strAppVersion As String = "0001"

Dim strExportDataString As New System.Text.StringBuilder()

Dim SORRecords As String

strFreeMem = "00000000"

'Creating the dummy SOR message record

strExportDataString.Append("SOR")

strExportDataString.Append(UserName) ‘Username will be dummy value XXX

strExportDataString.Append(Password) ‘Password will be dummy value XXX

strExportDataString.Append(strAppId) ‘Application id will be 006

strExportDataString.Append(strAppVersion) ‘App version will be dummy value 0001

strExportDataString.Append(strMacID) ‘MAC id will be dummy value 000000000000

strExportDataString.Append(strDeviceType) ‘Device type id will be dummy value R

strExportDataString.Append(strIPAdd) ‘IP address will be dummy XXX.XXX.XXX.XXX

strExportDataString.Append(strFreeMem)’Free memory will be dummy 00000000

SORRecords = strExportDataString.ToString()

SendSOR(SORRecords, m\_ActBuildTime)

End Function

**SendSOR() function:** Function will check the SOR message was send to the controller correctly.

Public Function SendSOR(ByVal strRecords As String, ByRef strDateTime As String) As Boolean

If SendRecord(strRecords) Then

'Set controller date time

strDateTime = m\_ControllerDateTime

'insert parsing characters accordingly

strDateTime = strDateTime.Insert(4, "-")

strDateTime = strDateTime.Insert(7, "-")

strDateTime = strDateTime.Insert(10, " ")

strDateTime = strDateTime.Insert(13, ":")

strDateTime = strDateTime.Insert(16, ":")

strDateTime = strDateTime & "00"

'return the status

Return True

Else

Return False

End If

End Function

**SendRecord() function**: Function will send the SOR record to the controller.

Private Function SendRecord(ByVal strRecord As String) As Boolean

Dim m\_SendBytes As [Byte]() = Nothing

Dim m\_ReadBytes As [Byte]() = Nothing

Dim m\_Status As Boolean = Nothing

Dim m\_RetryWrite As Integer = 0

m\_SockectConnMgr = New SocketConnectionMgr()

Try

'Records sent to the controller.

obLogger.WriteAppLog("ExDataTransmitter: Record sent: ", Logger.LogLevel.RELEASE)

'new message format

strRecord = Chr(255) + (strRecord.Length + 5).ToString.PadLeft(4, "0") + strRecord

m\_SendBytes = Encoding.ASCII.GetBytes(strRecord.ToString())

m\_SendBytes(0) = &HFF

m\_RetryWrite = m\_Retry

' m\_SendBytes = Encoding.ASCII.GetBytes(strRecord)

'Read the rety attempt for writing data to the socket stream.

m\_RetryWrite = Macros.WRITE\_RETRY

'Send the record to the controller.

Do

If m\_SockectConnMgr.TransmitData(m\_SendBytes) Then

'Read the response stream from the client.

If m\_SockectConnMgr.ReadData(m\_ReadBytes) And \_

m\_ReadBytes.Length > 0 Then

'Return the response after parsing it.

Return ParseResponse( \_

Encoding.ASCII.GetString(m\_ReadBytes, \_

0, \_

m\_ReadBytes.Length))

Else

'Add the exception to the application log.

obLogger.WriteAppLog("ExDataTransmitter: Cannot " \_

& "read from socket.", \_

Logger.LogLevel.RELEASE)

'If reading response from the controller is failed.

MessageBox.Show("Failed")

Return False

End If

End If

m\_RetryWrite = m\_RetryWrite - 1

Loop Until m\_RetryWrite = 0

'If all the write attempt failed.

If m\_RetryWrite = 0 Then

'write the error message to the app log

obLogger.WriteAppLog( \_

"Unable to write record to the stream. Retry attempt" \_

& "failed for" & m\_RetryWrite & "times", \_

Logger.LogLevel.RELEASE)

Return False

End If

Catch ex As Exception

'Add the exception to the device log.

obLogger.WriteAppLog( \_

"Error in sending export data to controller:" \_

& ex.Message.ToString(), \_

Logger.LogLevel.RELEASE)

'incase of exception return false.

Return False

Finally

m\_SendBytes = Nothing

m\_ReadBytes = Nothing

End Try

End Function

**ParseResponse() function**: Function will check the recived message from the controller.

Private Function ParseResponse(ByVal m\_ResponseMessage As String) As Boolean

'new message format

m\_ResponseMessage = m\_ResponseMessage.Substring(5, m\_ResponseMessage.Length - 5)

'Response received from the controller.

obLogger.WriteAppLog("ExDataTransmitter: Response received: ", Logger.LogLevel.RELEASE)

MessageBox.Show("Checking received message")

'Based on the message type parse the response and return the value.

Select Case m\_ResponseMessage.Substring(0, 3)

Case "ACK"

Return True

Case "GIB"

Return True

Case "GIR"

Return True

Case "SNR"

'set the device date & time according to the date time received

'in the response message from controller.

m\_ControllerDateTime = m\_ResponseMessage.Substring(Macros.SNR\_DATETIME\_START\_INDEX, \_

Macros.SNR\_DATETIME\_LENGTH)

MessageBox.Show("Date & Time receiced:", +m\_ControllerDateTime)

Return SetDeviceDateTime(m\_ControllerDateTime)

Case "NAK"

Dim strNakMessage As String = ""

'Supress NAK String

strNakMessage = m\_ResponseMessage.Replace("NAK", "")

'Suppress NAKERROR string

strNakMessage = strNakMessage.Replace("NAKERROR", "")

'Display the recevied NAK message to the user.

MessageBox.Show("Received error from controller:" + strNakMessage, \_

"Error", MessageBoxButtons.OK, \_

MessageBoxIcon.Hand, MessageBoxDefaultButton.Button1)

'If response received is NAK

Return False

End Select

End Function

**SetDeviceDateTime() function:** Function will set the HHT date and time with the date and time received from controller.

Private Function SetDeviceDateTime(ByVal strDateTime As String) As Boolean

Dim objSysTime As SYSTEMTIME

'Get the device time.

GetSystemTime(objSysTime)

'Populate structure to update the table.

With objSysTime

.wYear = Convert.ToInt16(strDateTime.Substring(0, 4))

.wMonth = Convert.ToInt16(strDateTime.Substring(4, 2))

.wDay = Convert.ToInt16(strDateTime.Substring(6, 2))

.wHour = Convert.ToInt16(strDateTime.Substring(8, 2))

.wMinute = Convert.ToInt16(strDateTime.Substring(10, 2))

.wSecond = Convert.ToInt16(0)

End With

'Set the new time`

Return SetLocalTime(objSysTime)

End Function

**SocketConnectionMgr.vb:** Need to add a new socket connection manager class as there is no socket communication happening between HHT and Transact in McDownloader.exe. This socket class will be sending the SOR message to Transact and parsing back the SNR message received back. It is the same socket connection manager class used in Shelf monitoring activities.

Public Class SocketConnectionMgr

''' <summary>

''' Member variables

''' </summary>

''' <remarks></remarks>

Dim m\_HostName As String = Nothing

Dim m\_Port As Integer = Nothing

Dim m\_RecBufferSize As Integer = Macros.RECEIVE\_BUFFER

Public m\_TcpClient As TcpClient = Nothing

Dim m\_NetworkStream As NetworkStream = Nothing

Public objLogger As Logger = Nothing

''' <summary>

''' Gets or sets the connection status

''' </summary>

''' <value></value>

''' <returns></returns>

''' <remarks></remarks>

Public ReadOnly Property ConnectionStatus() As Boolean

Get

'Return m\_ConnStatus

Return m\_TcpClient.Client.Connected

End Get

End Property

''' <summary>

''' Constructor

''' </summary>

''' <remarks></remarks>

Sub New()

GetConnectionDetails()

InitialiseConnection()

End Sub

''' <summary>

''' Initialise a connection to the TRANSACT.

''' </summary>

''' <remarks></remarks>

Private Sub InitialiseConnection()

m\_TcpClient = New TcpClient()

Try

m\_TcpClient.Connect(m\_HostName, m\_Port)

m\_NetworkStream = m\_TcpClient.GetStream()

Catch ex As Exception

'Add the exception to the device log.

objLogger.WriteAppLog(ex.Message.ToString(), Logger.LogLevel.RELEASE)

End Try

End Sub

''' <summary>

''' Transmits record to the controller in byte stream.

''' </summary>

''' <param name="bExportData"></param>

''' <returns>Bool

''' True - If successfully transmitted the data to the controller.

''' False - If error in transmitting the data to the controller.

''' </returns>

''' <remarks></remarks>

Public Function TransmitData(ByVal bExportData As Byte()) As Boolean

'Check if the network stream is capable of writing.

If m\_NetworkStream.CanWrite Then

Try

m\_NetworkStream.Write(bExportData, 0, bExportData.Length)

Catch ex As Exception

'Add the exception to the device log.

objLogger.WriteAppLog( \_

ex.Message.ToString(), \_

Logger.LogLevel.RELEASE)

Return False

End Try

Return True

Else

Return False

End If

End Function

''' <summary>

''' Read record from the stream.

''' </summary>

''' <param name="bRespData"></param>

''' <returns>Bool

''' True - If successfully read the data from the stream.

''' False - If failed to read the data from the stream.

''' </returns>

''' <remarks></remarks>

Public Function ReadData(ByRef bRespData As Byte()) As Boolean

'Check if the network stream is capable of writing.

ReDim bRespData(m\_TcpClient.ReceiveBufferSize)

'Chec if network stream can read.

If m\_NetworkStream.CanRead Then

Try

m\_NetworkStream.Read(bRespData, 0, m\_RecBufferSize)

Catch ex As Exception

'Add the exception to the device log.

objLogger.WriteAppLog(ex.Message.ToString(), \_

Logger.LogLevel.RELEASE)

Return False

End Try

Return True

Else

Return False

End If

End Function

''' <summary>

''' Gets the controller credentials from the config file.

''' </summary>

''' <returns>Bool

''' True - If successfully obtained the setting from the config file.

''' False - Any error occured in obtaining the config file.

''' </returns>

''' <remarks></remarks>

Private Function GetConnectionDetails() As Boolean

Dim strActiveIP As String = "activeIP"

Dim strPort As String = "IPPort"

m\_HostName = GetIPParam(strActiveIP)

m\_Port = CInt(GetParam(strPort).ToString())

End Function

''' <summary>

''' Close the connection established with the TRANSACT.

''' </summary>

''' <returns>Bool

''' True - If successfully transmitted all the data.

''' False - Any failure in transmitting the export data.

''' </returns>

''' <remarks></remarks>

Public Function TerminateConnection() As Boolean

Try

If Not m\_TcpClient Is Nothing Then m\_TcpClient.Close()

If Not m\_NetworkStream Is Nothing Then m\_NetworkStream.Close()

Catch ex As Exception

'Add the exception to the device log.

objLogger.WriteAppLog( \_

ex.Message.ToString(), \_

Logger.LogLevel.RELEASE)

Return False

End Try

Return True

End Function

''' <summary>

''' Get the IP details from the IPCONFIG.XML file

''' </summary>

''' <param name="cKey">

''' key to be searched in the xml file

''' </param>

Public Function GetIPParam(ByVal cKey As String) As String

'xml document object

Dim xd As New Xml.XmlDocument

Dim value As String = Nothing

Try

If System.IO.File.Exists(Macros.IPCONFIG\_FILE\_PATH) Then

'load the xml file

xd.Load(Macros.IPCONFIG\_FILE\_PATH)

'query for a value

Dim Node As Xml.XmlNode = xd.DocumentElement.SelectSingleNode( \_

"/Configuration/IPDetails/add[@key=""" & cKey & """]")

'return the value or nothing if it doesn't exist

If Not Node Is Nothing Then

value = Node.Attributes.GetNamedItem("value").Value

End If

Else

System.Windows.Forms.MessageBox.Show("IP Config File does not exist", "Error")

End If

Return value

Catch ex As Exception

MessageBox.Show("Invalid Config Key for IP Address")

Return Nothing

End Try

End Function

Public Function GetParam(ByVal cKey As String) As String

'xml document object

Dim xd As New Xml.XmlDocument

Try

If System.IO.File.Exists(Macros.CONFIG\_FILE\_PATH) Then

'load the xml file

xd.Load(Macros.CONFIG\_FILE\_PATH)

'query for a value

Dim Node As Xml.XmlNode = xd.DocumentElement.SelectSingleNode( \_

"/configuration/appSettings/add[@key=""" & cKey & """]")

'return the value or nothing if it doesn't exist

If Not Node Is Nothing Then

Return Node.Attributes.GetNamedItem("value").Value

Else

Return Nothing

End If

Else

System.Windows.Forms.MessageBox.Show("Config File does not exist", "Error")

End If

Catch ex As Exception

MessageBox.Show("Invalid Config Key")

End Try

Return Nothing

End Function

End Class

#End If.

* 1. Error Reporting

There is no change in error reporting.

2. Unit Test Plan
   1. Overview

The fix has been included in the AppContainer.vb file and the new .exe for application (McDowloader) should be created for HHT device. And TRXBASE.C is changed in the controller side.

* ADXLXACN:C:\ADX\_UPGM\TRANSACT.286
* ADXLXACN: C:\MC70
* ADXLXACN: C:\MC55BH
  1. Environment
* MC70 Handheld Device
* MC55 Batch Handheld Device.
* IBM controller with OS needs to be 4690 V6 R3 with software build 15B
  1. Unit Tests
     1. Unit Test Summary

| **Test**  **No.** | **Test Description / Execution steps** | **Expected Result** | **Pass or**  **Actual Result if Fail** |
| --- | --- | --- | --- |
| **Automated Date and Time utility main test cases** | | | | |
| 1 | | Step 1:  Reboot the HHT device by pressing the small red button in the HHT device. | BOOTS application will be loaded correctly if HHT device time mismatches with controller then HHT time will be set correctly. | Pass |
| 2 | | Step 1:  Change the controller time to a new time:  Step 2:  Click Clt+print screen button. Select “C” from options.  Step 3:  Select option 5 “System functions” Press enter.  Step 4:  Select option 1 “Set System Date and time”  Step 5:  Set a new date and time. Then press enter  Step 6:  Now reboot the HHT device by pressing the small red button in the HHT device. | BOOTS application will be loaded again and the HHT device date and time will be reset to the current controller date and time. | Pass |
| 3 | | Step 1:  Remove the HHT battery and insert new battery.  Step2:  Reboot the HHT device  Step 3:  Make sure HHT time shows as 00:00  Step 4:  Wait for the boots application to load | HHT Device Date/time should be set according to the controller date/time without the need to login to any of the SM application | Pass |
| 4 | | Step 1:  Reboot the HHT device.  Step 2:  Wait for the boots application to load  Step 3:  Check if the HHT time is set as 00:00. | Boots application will load correctly and the reference data download will happen correctly. | Pass |
| 5 | | Step 1:  Remove the HHT device from cradle and wait until the battery gets fully drained out.  Step 2:  Put the device back in cradle for charging and let the device reboot.  Step 3:  Make sure HHT time shows as 00:0  Step 4:  Wait for the boots application to load | HHT Device Date/time should be set according to the controller date/time without the need to login to any of the SM application | Pass |
| 6. | | Step 1:Remove the HHT battery to perform a cold reboot. Step2:  Put back the battery and reboot the device.  Step 3:  Make sure HHT time shows as 00:0  Step 4:  Wait for the boots application to load | HHT Device Date/time should be set according to the controller date/time without the need to login to any of the SM application | Pass |
| **Transact Test cases** | | | | |
| 8 | | Step1:  Reboot the HHT device pressing the small red button in the HHT device. | Transact will work properly without any issue. | Pass |
| 9 | | Step1:  Reboot the HHT device pressing the small red button in the HHT device.  Step 2:  Now check the controller background task. | Transact will not show any errors. Also check the controller application logs for Transact events. | Pass |
| 10 | | Step 1:  Debug Transact when it receives a SOR message with dummy user id and password. | Transact will build SNR message with current controller date and time. | Pass |
| 11 | | Step 1:  Debug Transact when it receives a SOR message with dummy user id and password. | The application id in the message will be 6(new application id for Automated date & time) | Pass |
| 12 | | Step 1:  Stop the Transact in controller and reboot the HHT device. | HHT device will receive error message “Received error from controller” | Pass |
| **Goods In Testing** | | | | |
| **Goods In Main menu** | | | | |
| 13 | | Step 1:  Click on the Goods In icon from the main screen | Enter User Id screen will be displayed without any issue. | Pass |
| 14 | | Step 1:  Enter the user login credentials. | Goods In screen will be shown. | Pass |
| 15 | | Step 1:  Click on Stores Service Centre receiving option in Goods In screen. | Store service Centre processing page will open without any issue. | Pass |
| **Stores Service Centre receiving options** | | | | |
| 16 | | Step 1:  Click on the Goods In  Step 2:  Click on Stores Service receiving option. | Stores Service Receiving page will open without any issue. | Pass |
| 17 | | **Book in a new UOD.**  Step 1:  Click on the Goods In Icon from main menu.  Step 2:  Click on Stores Service receiving option.  Step 3:  Click on Book In Delivery then click Next button  Step 4:  Then Scan/Enter UOD barcode.  Step 5:  Then Scan/Enter Driver badge wait for Final confirmation. | UOD will be booked in without any issue. | Pass |
| 18 | | **Perform Audit UOD**  Step1:  Click on the Goods In Icon from main menu.  Step 2:  Click on Stores Service receiving option.  Step 3:  Click Audit UOD and Scan/Enter UOD barcode.  Step 4:  Then Scan/Enter Product code and Enter quantity and click next | Store service centre receiving exception report will be generated without any issue. | Pass |
| 19 | | **View the details of the UOD**  Step 1:  Click on the Goods In Icon from main menu.  Step 2:  Click on Stores Service receiving option.  Step 3:  Click on View UOD option and Click on Today  Step 4:  Give the UOD number from list and select the UOD | UOD details will be shown without any issue. | Pass |
| **Directs receiving options** | | | | |
| 20 | | **Book in a new Direct Receiving order**  Step1:  Click on the Goods In Icon from main menu.  Step2:  Click Directs receivings  Step3:  Select Book In Order.  Step 4:  Select supplier or scan/enter carton  Step 5:  Scan/Enter Carton Barcode  Step 6:  Scan/Enter Carton Barcode and Enter quantity | Direct Carton will be booked without any issue. | Pass |
| 21 | | **Perform Audit Carton**  Step1:  Click on the Goods In Icon from main menu.  Step2:  Click Directs receiving  Step3:  Select Audit Carton  Step 4:  Scan/Enter Carton Barcode  Step5:  Scan/Enter Item and enter quantity of item | File Carton Book In Report will be generated without any issue. | Pass |
| 22 | | **View the details of the Carton**  Step 1:  Click on the Goods In Icon from main menu.  Step 2:  Click Directs receiving  Step 3:  Select View carton  Step4:  Select supplier or scan/enter carton and select one carton | Carton details will be shown without any issue. | Pass |
| **Shelf Monitor Testing** | | | | |
| 23 | | Step 1:  Click on the Shelf Monitor icon from the main screen | Enter User Id screen will be displayed without any issue. | Pass |
| 24 | | Step 2:  Enter the user login credentials. | Shelf Monitor screen will be shown. | Pass |
|  | | **Perform a Shelf Monitor**  Step 1:  Click on the Shelf Monitor Icon from main menu.  Step 2:  Select Shelf Monitor option  Step 3  Enter the product code/Scan the item and click ok  Step 4:  Enter the quantity and click ok  Step 5  Enter fill quantity and click next button. | Shelf monitor summary will be shown correctly. | Pass |
| 25 | | **Perform Excess Stock Calculation**  Step 1:  Click on the Shelf Monitor Icon from main menu.  Step 2:  Click on Excess Stock Icon  Step 3:  Select the count location(Back Shop/offsite)  Step4:  Scan/Enter the Product code  Step 5:  Enter the quantity that has to send Back Shop/offsite) | Excess stock will be shown correctly. | Pass |
| 26 | | **Perform Price Check**  Step 1:  Click on the Shelf Monitor Icon from main menu.  Step 2:  Click on the Price check Icon  Step 3:  Scan/Enter the Product code  Step 4:  Scan the SEL(Shelf Edge Label) | Compares the Item price and SEL price. | Pass |
| 27 | | **Perform a Fast Fill**  Step 1:  Click on the Shelf Monitor Icon from main menu.  Step 2:  Click on Fast Fill Icon  Step 3:  Scan/Enter the Product code  Step 4:  Enter the fill quantity | Picking list will be created to take the item from back shop. | Pass |
| 28 | | **Perform Counts List**  Step 1:  Click on the Shelf Monitor Icon from main menu.  Step 2:  Click on Counts List Icon  Step 3:  Scan/Enter product code  Step 4:  Select the place to perform the count – Sales floor, Back shop or OSSR  Step 5:  Enter the quantity available in the selected location | Counting for that item will be completed successfully. | Pass |
| 29 | | **Perform a Picking List**  Step 1:  Click on the Shelf Monitor Icon from main menu.  Step 2:  Click on Picking List Icon  Step 3:  Scan/Enter the Product code  Step 4:  Enter Back shop Quantity  Step 5:  Enter the quantity required in the sales floor | If the items are present in the back shop then Goods will be send to the shop floor. | Pass |
| 30 | | Step 1:  Click on the Shelf Monitor Icon from main menu.  Step 2:  Click on item information icon  Step 3:  Enter the item code | Item information will be displayed. | Pass |
| 31 | | **Perform Stuff Your Shelves**  Step 1:  Click on the Shelf Monitor Icon from main menu.  Step 2:  Click on Stuff your Shelves Icon  Step 3:  Scan/Enter the bar code | Picking list will be created to transfer the back shop to the shop floor. | Pass |
| **Goods Out Testing** | | | | |
| **Goods Out Main menu** | | | | |
| 32 | | Step1:  Click on the Goods Out icon from main menu. | Enter User Id screen will be displayed without any issue. | Pass |
| 33 | | Step 1:  Enter the user login credentials | Goods Out screen will be shown. Goods Out, Credit Claiming and Recall options will be shown correctly. | Pass |
| **Goods Out Option** | | | | |
| 34 | | Step 1:  Click on the Goods Out option from main menu. | Goods Out options will be displayed correctly. | Pass |
| 35 | | **Perform an Item Return**  Step1:  Click on the Goods Out Icon from main menu.  Step2:  Click on the Return and Destroy option  Step 3:  Click on Return option | All the Returns options will be shown without issue. | Pass |
| 36 | | **Perform an Item Destroy**  Step 1:  Click on the Goods Out Icon from main menu.  Step 2:  Click on the Return and Destroy option  Step 3:  Click on Destroy option | All the item destroy options will be displayed without issue. | Pass |
| 37 | | Step 1:  Click on the Goods Out Icon from main menu.  Step 2:  Click on the Semi Centralised Returns | All the Semi Centralised options will be displayed without issue | Pass |
| 38 | | Step 1:  Click on the Goods Out Icon from main menu.  Step 2:  Click on the Direct Returns options | All the Direct Returns options will be displayed without issue | Pass |
| 39 | | Step 1:  Click on the Goods Out Icon from main menu.  Step 2:  Click on Centralised Returns | All the Centralised options will be displayed without issue | Pass |
| 40 | | **Perform a Inter Store Transfer**  Step 1:  Click on the Goods Out Icon from main menu.  Step 2:  Click on Inter Store Transfer option  Step 3:  Click Goods Out Transfer  Step 4:  Enter the Destination Store Number  Step 5:  Scan/Enter product code  Step 6:  Enter the Quantity of Goods to be send  Step 7:  Enter/Scan the orange badge | Details will be transferred to controller successfully. | Pass |
| 41 | | **Perform Pharmacy Special waste transfer**  Step 1:  Click on the Goods Out Icon from main menu.  Step 2:  Click on Pharmacy Special waste option  Step 3:  Scan/Enter the Label  Step 4:  Click on Despatch | Details will be transferred from the controller and items will be removed from the stock. | Pass |
| 42 | | Step 1:  Click on the Goods Out Icon  Step 2:  Click Credit Clamming option | All the credit claiming options will be displayed without issue. | Pass |
| 43 | | Step 1:  Click on the Goods Out Icon  Step 2:  Click on the Recall option | All Recall options will be displayed correctly. | Pass |
| 44 | | **Perform Company/HO Recall**  Step 1:  Click on the Goods Out Icon from main menu.  Step 2:  Click on Company/HO Recall  Step 3:  Open the Active recall list  Step 4:  Scan/Enter the product code  Step 5:  Select the item from list and enter the Quantity  Step 6:  Enter/Scan the label | Item will be returned to HO. | Pass |