# 

# HashCode Tester Documentation

#### Document History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Details |
| 1.0 | 14/11/2019 | Richard | Initial Draft |
|  |  |  |  |
|  |  |  |  |

#### Contents

[1. HashCode Tester Screen Design 2](#_Toc24969341)

[2. Functions 3](#_Toc24969342)

[3. Use of the HashCode Tester 5](#_Toc24969343)

[4. Functional Unit Tests 6](#_Toc24969344)

[5. For Future Use 11](#_Toc24969345)

### HashCode Tester Screen Design

| **Test No.** | **Ref** | **Test** | |
| --- | --- | --- | --- |
|  |  | HashCode Tester Screen Design | |
| **Expected Procedure & Outcome** | | | |
|  | | | |
| **Test No.** | **Ref** | **Test** | |
|  |  |  | |
| **Expected Procedure & Outcome** | | | **Pass** |
|  | | |  |

### Functions

| **Test No.** | **Ref** | **Test** | |
| --- | --- | --- | --- |
|  |  | File menu | |
| **Expected Procedure & Outcome** | | | |
| Added a File menu with the following items:   * New/Clear All (the keyboard shortcut is ‘Alt + C’) (This clear **all** textboxes of information) * Help (the keyboard shortcut is ‘F1’) (This displays the help file)   Exit (the keyboard shortcut is ‘Alt + X’) (This exits the HashCode Tester) | | | |
| **Test No.** | **Ref** | **Test** | |
|  |  | Picturebox | |
| **Expected Procedure & Outcome** | | | |
| Created a picturebox icon that displays the author & version of the program. | | | |
| **Test No.** | **Ref** | **Test** | |
|  |  | Frames | |
| **Expected Procedure & Outcome** | | | |
| Created 2 frames   1. Payment Details 2. Hash Code | | | |
| **Test No.** | **Ref** | **Test** | |
|  |  | Text boxes | |
| **Expected Procedure & Outcome** | | | |
| Created 6 text boxes:   1. The ‘BACS Random String’ textbox allows any character up to a maximum of 64. 2. The ‘Originator Sort Code’ textbox allows up to 6 numerical characters only, a warning message will be displayed if the digits are below 6. 3. The ‘Recipient Sort Code’ textbox allows up to 6 numerical characters only, a warning message will be displayed if the digits are below 6. 4. The ‘Payment Value’ textbox allows up to 12 numerical characters only, including negative values. 5. The ‘Hash Code Random Number’ textbox allows any character up to a maximum of 4. A warning message will be displayed if the digits are below 4. 6. The Hash Code textbox will display the Hash Code generated from once the above textboxes have been fully completed. In addition, this textbox allows any character up to a maximum of 64. | | | |
| **Test No.** | **Ref** | **Test** | |
|  |  | Buttons | |
| **Expected Procedure & Outcome** | | | |
| Added a ‘New/Clear All’ button to clear **all** textboxes of information  Added a ‘Generate’ button to generate the Hash-256 code  Added a ‘Compare’ button, to compare the 2 BACS Hash codes & added code to verify that the 2 hash codes do or do not match.   |  | | --- | | **Note:** Both the ‘Generate’ & ‘Compare’ buttons will be disabled until the details have been entered fully within the ‘Payment Details’ frame. | | | | |
| **Test No.** | **Ref** | **Test** | |
|  |  | Tooltips | |
| **Expected Procedure & Outcome** | | | |
| Added tooltips for all textboxes & buttons | | | |
| **Test No.** | **Ref** | **Test** | |
|  |  |  | |
| **Expected Procedure & Outcome** | | | **Pass** |
|  | | |  |

### Use of the HashCode Tester

| **Test No.** | **Ref** | **Test** | |
| --- | --- | --- | --- |
|  |  | How to Use (step-by-step guide) | |
| **Expected Procedure & Outcome**  **Pass** | | | |
| 1st step: Ensure that the employees are to be paid via BACS  2nd step: Print out a report displaying the employee’s pay details, including the Hash Code  3rd step: Submit your FPS  4th step: Open your send.xml file, usually located in the ‘xml’ directory of your installation folder  5th step: Locate the BACS random string that was created by your payroll product & paste that into the ‘BACS Random String’ textbox.  6th step: Enter the employer’s sort code in the format of nnnnnn [6n]  7th step: Enter the employee’s sort code in the format of nnnnnn [6n]  8th step: Enter the employee’s NET pay, up to a maximum value of 12 digits.  9th step: Enter the Hash Code generated by your payroll product  10th step: Once all the above has been entered, clicking the ‘Generate’ button will display the Hash Code.  11th step: Clicking the ‘Compare’ button will compare step 5 with step 10.  12th step: This will clear all data on the screen, ready for the whole process to start over.   |  | | --- | | **Note:** different payroll products may differ in the steps required. | | | | |
| **Test No.** | **Ref** | **Test** | |
|  |  |  | |
| **Expected Procedure & Outcome** | | | **Pass** |
|  | | |  |

### Functional Unit Tests

| **Test No.** | **Ref** | **Test** | |
| --- | --- | --- | --- |
|  |  | File menu | |
| **Expected Procedure & Outcome** | | | **Pass** |
| Check that a File menu has been added to the ‘HashCode tester’ with the following 3 items:   1. New/Clear All   Check that clicking ‘New/Clear All’ menu item must clear all textboxes of information without error.  Check that the keyboard shortcut of ‘Alt + C’ must behave in the same way as clicking the ‘New/Clear All’ menu item.  **HashCodeForm1.cs code lines: 402-417**   1. Help   Check that clicking ‘Help’ menu item must display the ‘Help’ file without error.  Check that the keyboard shortcut of ‘F1’ must behave in the same way as clicking the ‘Help’ menu item.  **HashCodeForm1.cs code lines: 398-401**   1. Exit   Check that clicking ‘Exit’ menu item must display a message asking if exiting the application is what you want to do, with ‘Yes/No’ buttons without error.  Check that clicking the ‘No’ button must return you back to the ‘HashCode Tester’ main screen without error.  Check that clicking the ‘Yes’ button must exit the application without error.  Check that the keyboard shortcut of ‘Exit’ must display a message asking if exiting the application is what you want to do, with ‘Yes/No’ buttons without error.  Check that clicking the ‘No’ button must return you back to the ‘HashCode Tester’ main screen without error.  Check that clicking the ‘Yes’ button must exit the application without error.  **HashCodeForm1.cs code lines: 176-179** | | |  |
| **Test No.** | **Ref** | **Test** | |
|  |  | Picturebox | |
| **Expected Procedure & Outcome** | | | **Pass** |
| Check that clicking the Picturebox icon (£) must display a [message](#Screen2) showing the version of the application & the author of the software.  **HashCodeForm1.cs code lines: 33-36** | | |  |
| **Test No.** | **Ref** | **Test** | |
|  |  | Frames | |
| **Expected Procedure & Outcome** | | | **Pass** |
| Check that there are 2 frames called:   1. Payment Details   **HashCodeForm1.cs code lines: 17-21**  Check that the following textboxes are contained within the above frame   * BACS Random String * Originator Sort Code * Recipient Sort Code * Payment Value * Hash Code Random Number  1. Hash Code   **HashCodeForm1.cs code lines: 24-26**  Check that a textbox is contained within the above frame, this is where the BACS Hash Code is displayed when the ‘Generate’ button is clicked & all pertinent information has been entered. | | |  |
| **Test No.** | **Ref** | **Test** | |
|  |  | Text Boxes | |
| **Expected Procedure & Outcome** | | | **Pass** |
| Check that there are 6 text boxes:  1. BACS Random String  Check that this textbox allows any character up to a maximum of 64.  Check that this textbox allows alphanumeric & special characters  **HashCodeForm1.cs code lines: 146-150**  2. Originator Sort Code  The ‘Originator Sort Code’ textbox allows up to 6 numerical characters only, a warning message will be displayed if the digits are below 6.  **HashCodeForm1.cs code lines: 160-164**   1. Recipient Sort Code   The ‘Recipient Sort Code’ textbox allows up to 6 numerical characters only, a warning message will be displayed if the digits are below 6.  **HashCodeForm1.cs code lines: 166-169**   1. Payment Value   The ‘Payment Value’ textbox allows up to 12 numerical characters only, including negative values.  **HashCodeForm1.cs code lines: 171-174**   1. Hash Code Random Number   The ‘Hash Code Random Number’ textbox allows any character up to a maximum of 4. A warning message will be displayed if the digits are below 4.  **HashCodeForm1.cs code lines: 93-102**   1. The Hash Code textbox will display the Hash Code generated, once the above textboxes have been fully completed. In addition, this textbox allows any character up to a maximum of 64.   **HashCodeForm1.cs code lines: 93-102**  Check that this textbox allows alphanumeric & special characters  The Hash Code textbox will display the Hash Code generated from once the above textboxes have been fully completed. In addition, this textbox allows any character up to a maximum of 64. | | |  |
| **Test No.** | **Ref** | **Test** | |
|  |  | Buttons | |
| **Expected Procedure & Outcome** | | | **Pass** |
| Check that there will be 3 buttons positioned at the bottom of the ‘HashCode Tester’ screen:   1. Generate   Check that the above button will be disabled, until the textboxes (i.e. ‘BACS Random String’, ‘Originator Sort Code’, ‘Recipient Sort Code’, ‘Payment Value’ & ‘Hash Code Random Number’) have been correctly completed.  Check that once all mentioned textboxes have been completed, the ‘Generate’ button will become enabled.  Check that clicking the ‘Generate’ button will create the BACS Hash-256 Code & display this code within the ‘Hash Code’ textbox.  **HashCodeForm1.cs code lines: 50-61**   1. New/Clear All   Check that the above button will always be enabled.  Check that clicking the above button will clear all textboxes of information, displaying a message asking if this is what you want to do, with ‘Yes/No’ buttons  Check that clicking ‘No’ will close the message taking you back to the ‘HashCode Tester’ screen, without error.  Check that clicking ‘Yes’ will close the message & delete all the information entered in the textboxes above, all without error.  **HashCodeForm1.cs code lines: 37-49**   1. Compare   Check that once the ‘BACS Random String’ has been copied/pasted from your payroll product, the ‘Generate button will automatically move from disabled to enabled.  **HashCodeForm1.cs code lines: 62-90**   |  | | --- | | **Note:** Both the ‘Generate’ & ‘Compare’ buttons will be disabled until the details have been entered fully within the ‘Payment Details’ frame. | | | |  |
| **Test No.** | **Ref** | **Test** | |
|  |  | Tooltips | |
| **Expected Procedure & Outcome** | | | **Pass** |
| * Check that a tooltip has been added to the ‘BACS Random String’ textbox & the tooltip will be:   ‘Copy & Paste the BACS hash code from your payroll product’  **HashCodeForm1.cs code lines: 293-294**   * Check that a tooltip has been added to the ‘Originator Sort Code’ textbox & the tooltip will be:   ‘Enter the Employers Bank Sort Code’  **HashCodeForm1.cs code lines: 297-298**   * Check that a tooltip has been added to the ‘Recipient Sort Code’ textbox & the tooltip will be:   ‘Enter the Employees Bank Sort Code’  **HashCodeForm1.cs code lines: 301-302**   * Check that a tooltip has been added to the ‘Payment Value’ textbox & the tooltip will be:   ‘Enter the Employees NET pay’  **HashCodeForm1.cs code lines: 305-306**   * Check that a tooltip has been added to the ‘Hash Code Random Number’ textbox & the tooltip will be:   ‘Enter the Hash Code’  **HashCodeForm1.cs code lines: 309-310**   * Check that a tooltip has been added to the ‘Hash Code’ textbox & the tooltip will be:   ‘Copy & Paste the 'BACS Hash Code' to Compare or click Generate’  **HashCodeForm1.cs code lines: 313-314**   * Check that a tooltip has been added to the ‘Generate’ button & the tooltip will be:   ‘Generate a hash code’  **HashCodeForm1.cs [Design] tooltip1**   * Check that a tooltip has been added to the ‘New/Clear All’ button & the tooltip will be:   ‘Clear all details’  **HashCodeForm1.cs [Design] tooltip2**   * Check that a tooltip has been added to the ‘Compare’ button & the tooltip will be:   ‘Compare hash codes’  **HashCodeForm1.cs [Design] tooltip3** | | |  |
| **Test No.** | **Ref** | **Test** | |
|  |  |  | |
| **Expected Procedure & Outcome** | | | **Pass** |
|  | | |  |

### For Future Use

| **Test No.** | **Ref** | **Test** | |
| --- | --- | --- | --- |
|  |  |  | |
| **Expected Procedure & Outcome** | | | **Pass** |
|  | | |  |
| **Test No.** | **Ref** | **Test** | |
|  |  |  | |
| **Expected Procedure & Outcome** | | | **Pass** |
|  | | |  |
| **Test No.** | **Ref** | **Test** | |
|  |  |  | |
| **Expected Procedure & Outcome** | | | **Pass** |
|  | | |  |
| **Test No.** | **Ref** | **Test** | |
|  |  |  | |
| **Expected Procedure & Outcome** | | | **Pass** |
|  | | |  |
| **Test No.** | **Ref** | **Test** | |
|  |  |  | |
| **Expected Procedure & Outcome** | | | **Pass** |
|  | | |  |
| **Test No.** | **Ref** | **Test** | |
|  |  |  | |
| **Expected Procedure & Outcome** | | | **Pass** |
|  | | |  |
| **Test No.** | **Ref** | **Test** | |
|  |  |  | |
| **Expected Procedure & Outcome** | | | **Pass** |
|  | | |  |
| **Test No.** | **Ref** | **Test** | |
|  |  |  | |
| **Expected Procedure & Outcome** | | | **Pass** |
|  | | |  |