Camera Functions Specifications

ORIGINATOR:

Name	Job Title	Signature	Date
Andrew Mason	Chief Technology Officer	*(signature)*	3/5/12

APPROVALS:

Name	Job Title	Signature	Date
Nancy Rizzo	Founder and CEO	*(signature)*	3/5/12.
Mike Stowell	V.P of Operations	*(signature)*	3/5/12
Brian Tompson	Software Engineer	*(signature)*	3/5/12
Andrea Miller	Quality Assurance Manager	*(signature)*	3/5/12

1.0   Purpose

The purpose of this document is to describe the specifications for the ClearView system calibration process.

2.0   General Requirements

Where appropriate, logic should be contained in try/except blocks and any exceptions should be logged using the standard logging mechanisms.

3.0   Specifications

Requirement #4.1: The calibration process must take 5 warm up images before capturing the 10 images to use in the actual calibration calculation process. The delay between the capture of these warm up images must be approximately 2 seconds.

Specification: The automated calibration routine will capture five energized images at the beginning of the process. The images are captured approximately 2 seconds apart. These five images will not be used in the calibration process itself and are not persisted, the sole intent of generating these images is to warm up the device.

Requirement #4.2: The calibration must be calculated using 10 sample images captured sequentially.

Specification: One the warm up images have been captured, the automated process will then continue to capture ten sequential images. The images are then saved as bitmap files to be used in the analysis process that follows. Each of the images being captured can be seen on the calibration page. The images are captured through the standard image capture algorithm of the system.

Requirement #4.3: The delay between capture should be approximately 2 seconds.

Specification: A delay of approximately two seconds exists between the capture of each image.

Requirement #4.4: The calibration process will be accessible to all users.

Specification: All user roles (Base User, EPIC Administrator,System Administrator and techadministrator) can all access the calibration function. The function is always

available to any user with the exception of when a user is performing a capture or some other system function.

Requirement #4.5: The button to access the calibration process must appear on the main menu.

Specification: The button to access the calibration function appears in the main menu of the application.

The button is always present once a user has logged in and is easily visible and discernible.

Requirement #4.6: A tolerance value for the number of pixels that may fail should be stored in the system and it should be configurable.

Specification: The value of total pixels that may be out of range before the image is considered failed is currently set as a constant in the source code itself. It is configurable but requires a code recompile (and hence a version change) before the change is realized. The range of values that are acceptable for each pixel is saved in a database table called 'CalibrationLimits'.

The value can be moved into a more configurable area if desired, but the current implementation seems as if it is the best overall option.

Requirement #4.7: Calibration is required before a scan can be performed under the following circumstances:

- ⚐ If 24 hours have elapsed since the last calibration process has been performed.

- ⚐ If more than 4 hours have elapsed since the device was last accessed (aka. a scan was performed).

Specification: A routine containing two separate checks is run before any scan can take place. The function "MustCalibrateCamera" which returns a Boolean indicating whether or not a calibration is needed performs the following checks in the following order.

- First determine if a successful calibration has been run within the past 24 hours. If one has not returned a true value indicating that a calibration must be run.

- Second, if a successful calibration has been run within the past 24 hours, determine if the device has been accessed within the past 4 hours. If the device has not, return a value of true indicating that the device must be calibrated.

If a true value is returned, logic in the application will inform the user of the need to calibrate and will prevent a scan from being performed until the process is successfully completed.

In addition to the above two checks a third check is performed to verify that the necessary Matlab and bitmap files are present in c:\temp\calib_results.

Requirement #4.8: The date and time of a successful calibration as well as the images associated with the calibration must be stored in the database. No record of an unsuccessful calibration needs to be maintained.

Specification: Upon a successful calibration an entry is placed in the table named 'Calibration', a description of the table is noted below:

dbo.Calibration

🔑	CalibrationId*	BIGINT
▦	CustomerId	INT
▦	DateCalibrated	TIMESTAMP
▦	ImageId	BIGINT
▦	Guid	UNIQUEIDENTIFIER
▦	CaptureDeviceId	INT

▸ **PK_Calibration**
▸ **IX_Calibration_CaptureDeviceId**
▸ **IX_Calibration_CustomerId**
▸ **IX_Calibration_DateCalibrated**
▸ **IX_Calibration_Guid**
▸ **IX_Calibration_ImageId**

Calibration

	Column Name	Data Type	Allow Nulls
🔑	CalibrationId	bigint	☐
	CustomerId	int	☐
	DateCalibrated	datetime	☐
	ImageId	bigint	☐
	Guid	uniqueidentifier	☐
	CaptureDeviceId	int	☑
	CreatedBy	varchar(80)	☑
			☐

All of the columns in the table are filled with data before the row is created, the column DateCalibrated contains the time and date that the successful calibration was run. The created by column, contains the user that ran the calibration process.

Requirement #4.9: The calibration process should be controlled by a single button available to all users labeled 'Auto Calibration'. Clicking this button should do the following without user input:

- ⚘ Collect 5 warm up images

- ⚘ Collect 10 images to use in the calibration process

- ⚘ Perform the calibration calculation on all images

- ⚘ Determine pass or fail status

    ⚜ Persist data if successful

Specification: The calibration dialog is shown below; the button to begin the calibration process is noted in the lower right corner of the dialog. Clicking this button begins the following sequence:

- The Auto Calibration button is disabled as well as the Close button.
- Five warm up images are captured.
- Ten calibration images are captured.
- The calculation (comparison) routine is run on all images.
- The status of pass or fail is determined.
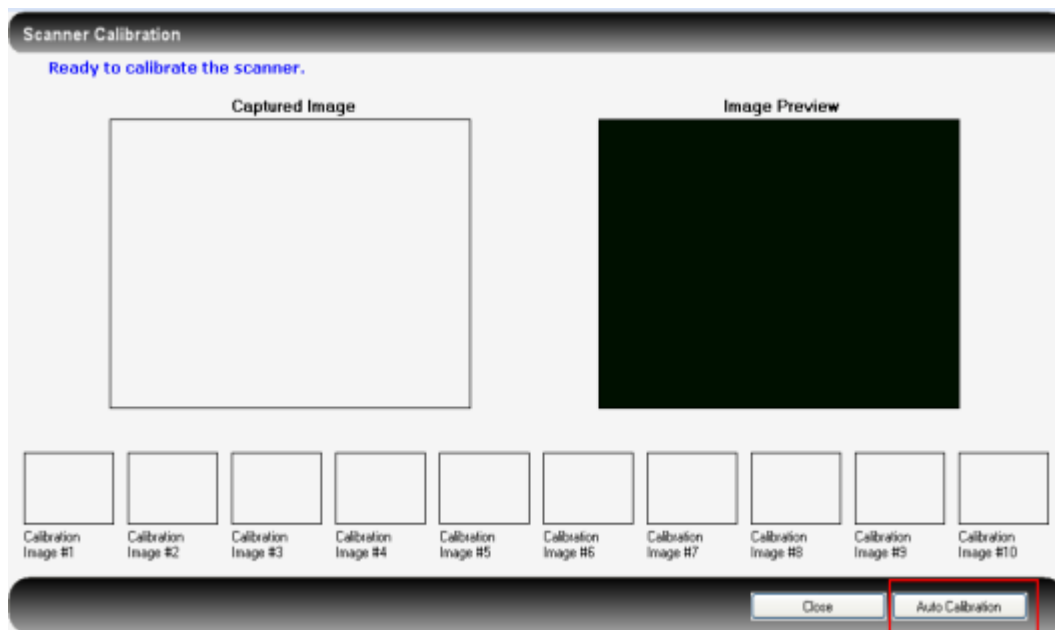
If all of the images pass the calibration check, they are written to the database as a single zip file containing all 20 images.

Requirement #4.10: Present the user with a progress indicator of some sort so they are aware of where they are in the process.

Specification: A progress indicator and progress message are displayed in the upper left corner of the dialog. The messages indicate what step of the process the routine is in.

Requirement #4.11: The process must prevent the user from exiting the process once it has begun until a final result has been reached.

Specification: Code has been implemented to cause the close button to become disabled during the calculation process. As soon as the Auto Calibration button is pressed and the process started, all buttons temporarily become disabled. The buttons are re-enabled as soon as the calibration process is complete.

If the user attempts to close the dialog using the X in the upper right corner of the window, the system will check to make sure that the calibration process is not running before allowing the user to close the dialog. If the process is running, the dialog will not be allowed to close.

Requirement #4.12: The administrator and techadministrator users only, should be able to manually load calibration images, bypassing the auto calibration process. Once the user has added the images, a button to perform the calibration calculations should be available..

Specification: If, and only if, the logged in user is administrator, or techadministrator, they can right click on any one of the image placeholders at the bottom of the dialog to cause a file load dialog to appear.
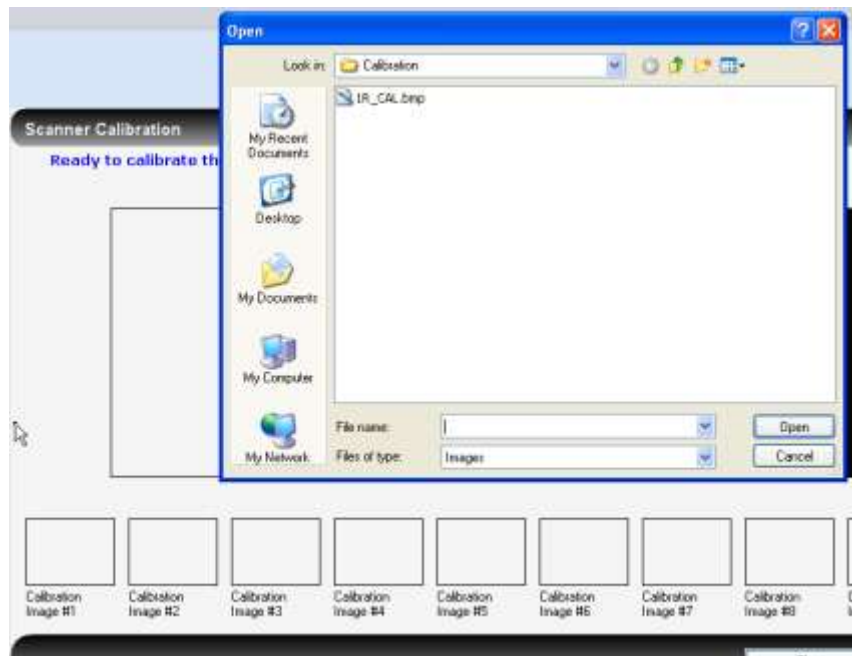
If any other user right clicks in the placeholders, noting will happen. The users can then browse and select a calibration image from the file system. When the image is selected it is added into the selected placeholder.

If an image is manually loaded, the "Auto Calibration" button changes to have a label of "Submit Calibration". The "Submit Calibration" button will verify that 10 images have been loaded before starting the actual calibration process.
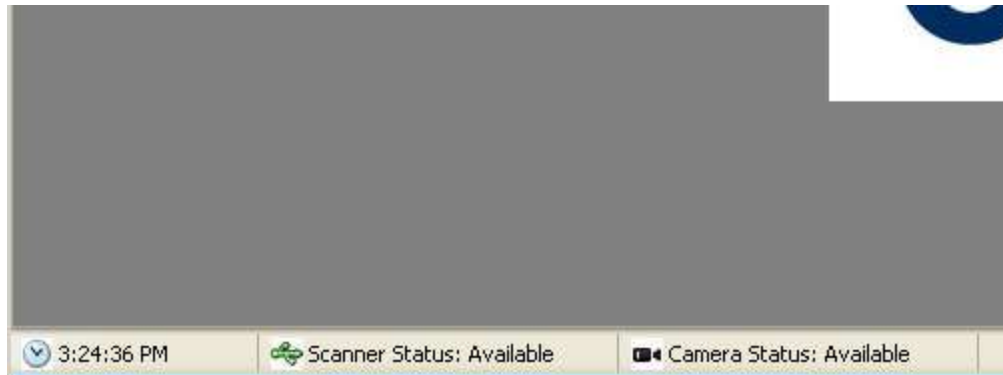
Requirement #4.13: The calibration process should not be accessible unless a scanner is attached to the system. If no scanner is found, a message should be displayed indicating such.

Specification: The system will check for the existence of a scanner prior to displaying any of the dialogs that perform the capture process. The check will be done by verifying that the device can be seen on the USB bus.

If the scanner cannot be detected, a message will be displayed informing the user and the capture process will not be allowed to continue.

In addition two visual indicators will be added to the ClearView application showing the status of both the scanner device and the camera within the scanner. These indicators will appear at the bottom of the application as shown below:

3:24:36 PM	Scanner Status: Available	Camera Status: Available

Requirement #4.14: The calibration dialog should allow the 10 captured images to be seen during the calibration process.

Specification: The calibration process displays the ten captured images in thumbnail for at the bottom of the dialog. The full size version of the image can be seen by clicking on the thumbnail.
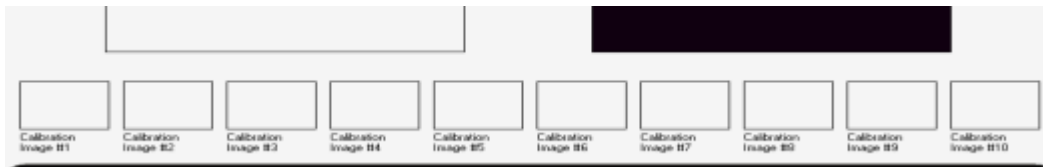
Requirement #4.15: Calibration can be re-run at any time and for as many times as the user desires. Only the last successful calibration will be used for analysis purposes.

Specification: The calibration process is always available on the menu bar. The only time calibration cannot be run is when the user is performing one of the other system functions.

Requirement #4.16: If a calibration process fails, a message should be displayed to the user. In addition the criteria that caused the failure should be placed in the alerts section of the application so they can be reviewed if necessary. The criteria would consist of the number of pixels that failed broken down by how many failed high and how many failed low.

Specification: The system will display a message indicating whether or not the image set passed calibration. In the event that the image set does not pass calibration, the system will add a message to the Alerts section of the website indicating what images failed calibration and what order of magnitude they failed by. The decision was made not to

display each pixel that failed since this number can sometimes be very large. Rather, the image name, the number of pixels that failed below the threshold and the number of pixels that failed above the threshold are shown

The calibration image analysis results are also stored in a local directory. The file is called 'CalibrationData.csv' and will be stored after each calibration run in the following location: c:\temp\calib_work. The file contains a table with the image number, number of pixels that fail high, number of pixels that fail low, and the pass/fail result. The file is overwritten on each calibration run; therefore, the user must save a copy of it after each run if the data is required to be retained.

A second more detailed data file will be written into the current ClearView home directory containing the following data for each pixel that fails:
- Image #
- X Coordinate
- Y Coordinate
- Measured Intensity
- Minimum Acceptable Intensity
- Maximum Acceptable Intensity

If any of the ten calibration images fail, the calibration set as a whole will be considered to be invalid.

Requirement #4.17: The intensity range for all pixels must be persisted in a table in the ClearView database. A row for each of the 320x240 pixels must exist. The high range and low range must also be contained in the table.

Specification: A table exists in the ClearView database to hold on to the calibration limits data. The table and its schema is shown below:

dbo.CalibrationLimits

🔑	XCoordinate*	SMALLINT
🔑	YCoordinate*	SMALLINT
▦	MinAcceptableValue	SMALLINT
▦	MaxAcceptableValue	SMALLINT
▦	CriticalPixel	SMALLINT
▶	**PK_CalibrationLimits**	

The critical pixel flag will be set to 0 and will not be used. This may be modified in a future release to refine the sensitivity of the calibration process.

Requirement #4.18: The calibration calculation is defined as follows:

4.18.1 The inner edge of the calibration image must be established to orient the pixels acceptance criteria.

4.18.2 For each pixel in the 320x240 matrix, compare it to the corresponding limits established in master calibration file. These limits were established based on the results documented in the EPIC ClearView Calibration Testing Modification Report, ENG-007. The intensity of the pixel must fall between the high and low range, if it does not, it is considered a failure.

4.18.3 The routine will keep track of how many pixels fail and whether or not they are critical.

4.18.4 If the total number of pixels that fail are equal to or greater than 6,528, the image will be considered invalid and the calibration process will fail.

4.18.5 The same process will be performed on all 10 images that have been collected.

4.18.6 If none of the 10 images fail, the calibration is considered a success.

Specification: Each of the points in the above requirement is described below.

⚐ Since the analysis of the images is predicated on the fact that the limits were created assuming a perfectly centered calibration image, it is necessary to determine the center of the image that is to be calibrated, as it may not be in exactly the center of the graphic. This is done using the standard center and radius MatLab routine that is incorporated throughout the calculations performed by ClearView. Once the center of the target image is determined, the limits array is adjusted to match the center of the target image, thus resulting in proper pixel comparison.

⼂ The calibration engine will determine the intensity of each of the 78,600 pixels in the image and compare that intensity to the high and the low specified in the configuration data for the given pixel location.

⼂ The routine tracks the count of pixels that fail (do not fall within the acceptable range for the specified location). When the count reaches the maximum number of pixels that can fail on an image, the calibration image is marked as fail.

⼂ The number of pixels that can fail before the image is considered invalid is 6528.

⼂ All ten images go through exactly the same process. If any one of the images fail the calibration process, the set as a whole is considered failed.

⼂ If all 10 images have a pixel failure rate less than 6528, the calibration set will pass.

Requirement #19: The calibration images and the calibration data should be able to be exported from the ClearView database. The export process will label the images with finger and hand references (1R).

Specification: The export process will create two folders in the main export folder called 'CalibrationImages' and 'CalibrationData'.

CalibrationImages will contain 10 .bmp files that are the calibration images that were collected. The naming convention will be {Finger Hand}_Calibration_Image_#{image number}.bmp

Calibrationdata will contain a CSV file that contains the number of pixels that failed for each calibration image as shown below:

	A	B	C	D	E	F
1	Image #1 passed with 3394 low pixels and 816 high pixels					
2	Image #2 passed with 3139 low pixels and 893 high pixels					
3	Image #3 passed with 3516 low pixels and 1020 high pixels					
4	Image #4 passed with 3434 low pixels and 792 high pixels					
5	Image #5 passed with 3117 low pixels and 778 high pixels					
6	Image #6 passed with 3072 low pixels and 912 high pixels					
7	Image #7 passed with 2969 low pixels and 984 high pixels					
8	Image #8 passed with 2905 low pixels and 836 high pixels					
9	Image #9 passed with 2870 low pixels and 1040 high pixels					
10	Image #10 passed with 2901 low pixels and 835 high pixels					

Requirement #4.20: When a successful calibration is obtained, the images and a reference to the calibration are stored in the database. All scans performed after the calibration will reference the calibration. An audit trail will be kept for each image stored in the database consisting of the following tracking points:

- ID of the user that performed the calibration.
- Date and time of the calibration down to the minute.

Specification: After a successful calibration, the images are stored in the database for later reference. The database table contains a location for the user that ran the calibration as well as a date and time that the calibration was run, stored in UTC.

Requirement #20: Upon a successful calibration, the images should be passed to the Matlab engine to generate the values necessary for the NS coefficient calculation.

Specification: Ten Matlab config files are created based on the data in the image. The config files are then used later in the analysis process.

Document Revision History

Version Number:	Description of Change:	Date:	Updated by:
000	Introduction	5/24/11	A. Mason
001	Update for changes to the calibration pass/fail criteria.	7/8/11	A. Mason
002	Update the specifications for requirements: • Requirement #1- Specify the 2 second delay as a requirement and correct the specification stated to match. • Requirement #16- Clarify how the number of failed pixels is reported and update the specification to match. Additionally, add the specifications for the local save of the calibration analysis results for each calibration image.	8/31/11	A. Mason
003	Performed a synchronization of the requirements document with this document. Major updates for this revision include the addition of the audit trail requirements (i.e., noting the user id, date and time of calibration image collection) and a modification to the file naming system to facilitate reload of images by EPIC Administrators.	3/5/12	A.Mason