Camera Functions- Specifications

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 #1: The calibration process must take 5 warm up images before capturing the 10 images to use in the actual calibration calculation process.

Specification: The automated calibration routine will capture five energized images at the beginning of the process. The images are captured approximately 1 second apart. These five images will not be used in the calibration process itself, the sole intent of generating these images is to warm up the camera.

Requirement #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 #3: The delay between capture should be approximately 2 seconds.

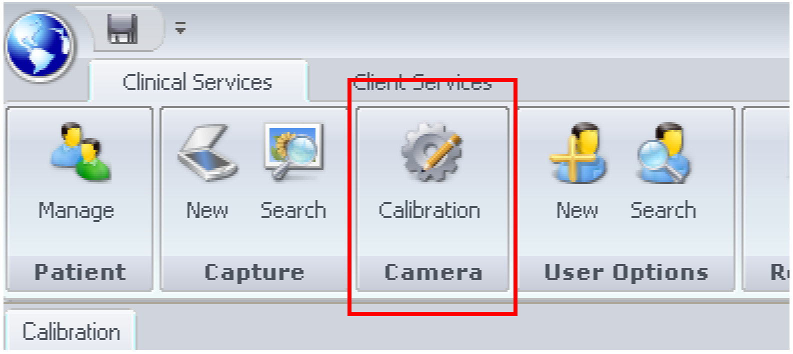
Specification: A delay of approximately two seconds exists between the capture of each image.

Requirement #4: The calibration process will be accessible to all users.

Specification: All user roles (Base User, EPIC Administrator and System Administrator) 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 #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 eacy visible and discernible.

Requirement #6: A tolerance value for the number of non-critical 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.

IT determined that storing it in the database or in a configuration file was not the best approach as these are relatively easily changed by the savvy user.

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 #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.

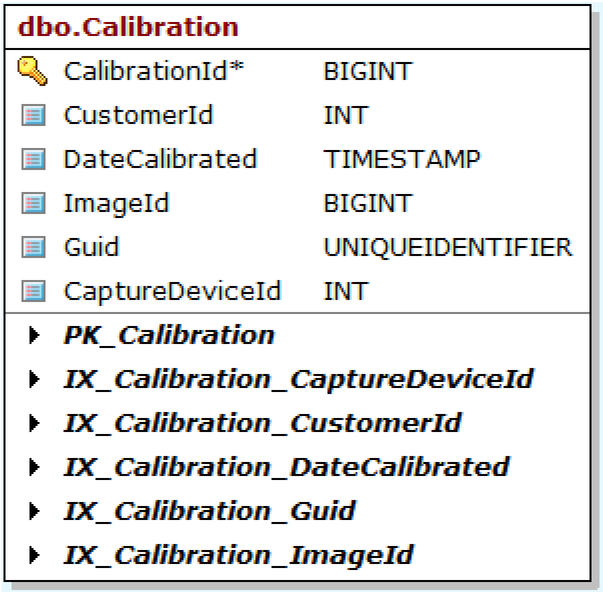
- First determine if a successful calibration has been run within the past 24 hours. If one has not returned a true value indicating that a calibration must be run.

- 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.

Requirement #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

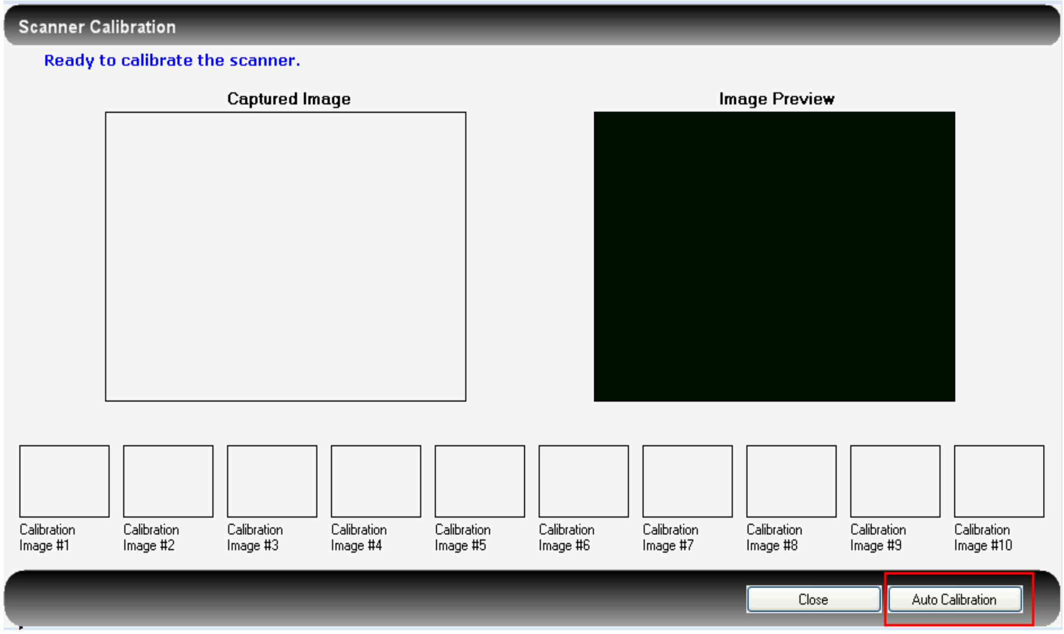
- *PK_Calibration*
- *IX_Calibration_CaptureDeviceId*
- *IX_Calibration_CustomerId*
- *IX_Calibration_DateCalibrated*
- *IX_Calibration_Guid*
- *IX_Calibration_ImageId*

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.

Requirement #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 once begins the calibration process that encompasses the five steps noted above.

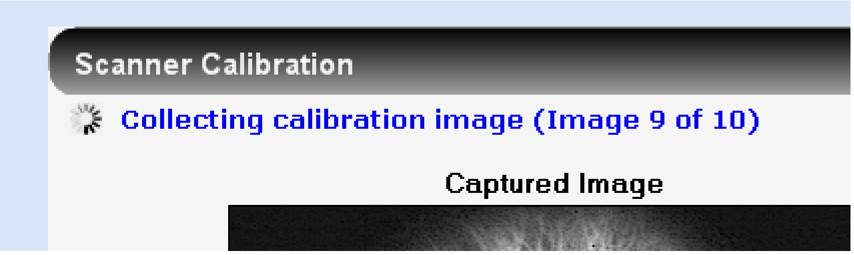


Scanner Calibration

Ready to calibrate the scanner.

Requirement #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.



Scanner Calibration

Collecting calibration image (Image 9 of 10)

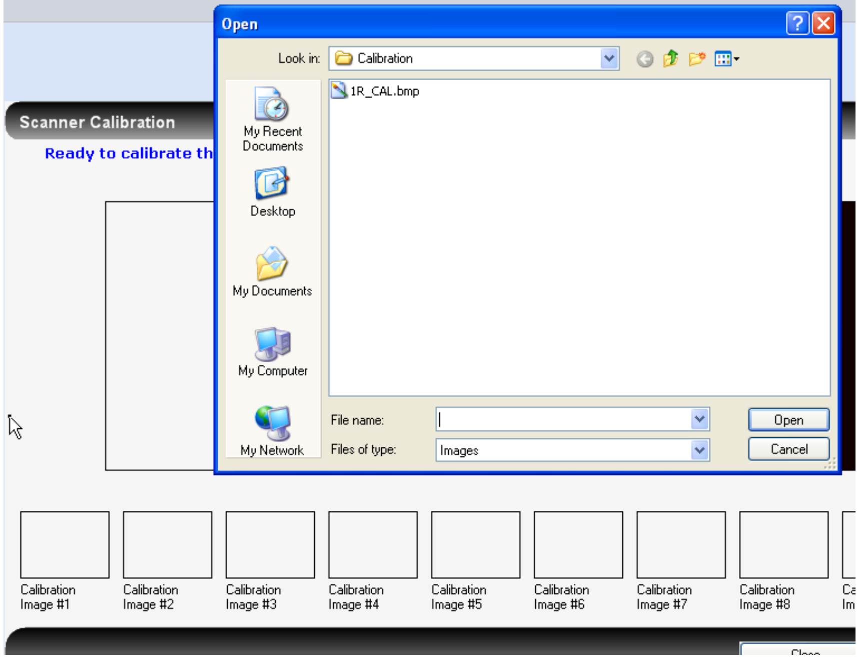
Requirement #11: The process must prevent the user from exiting the process once it has begun until a final result has been reached.

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 #12: The administrator user 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 the system administrator, 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.

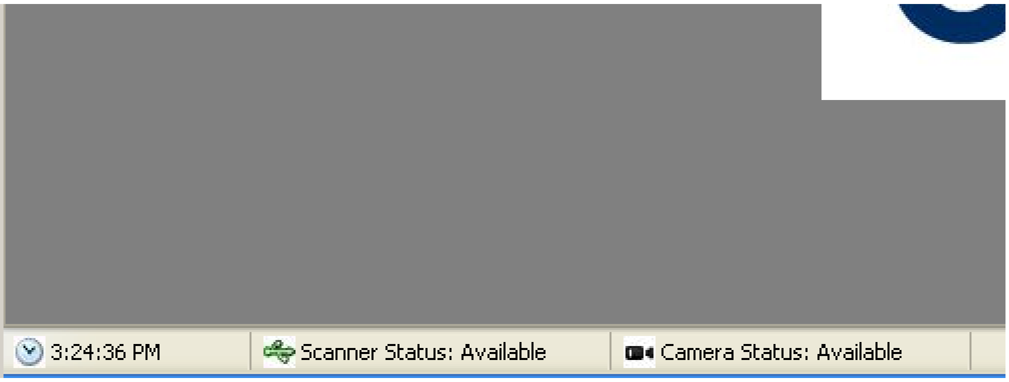
If an image is manually loaded, the “Auto Calibration” button changes to have a label of “Submit Calibration”. The “Submit Calibration” button will verify that 10 images have been loaded before starting the actual calibration process.

Requirement #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.

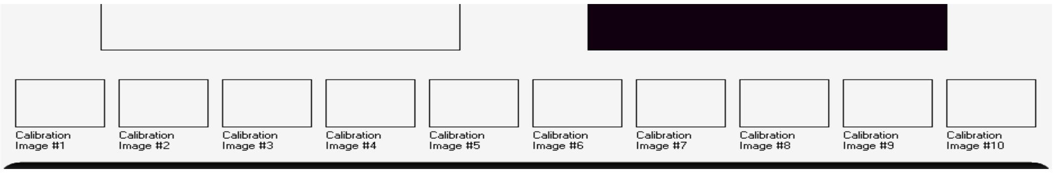
If the scanner cannot be detected, a message will be displayed informing the user and the capture process will not be allowed to continue.

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:



Requirement #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.

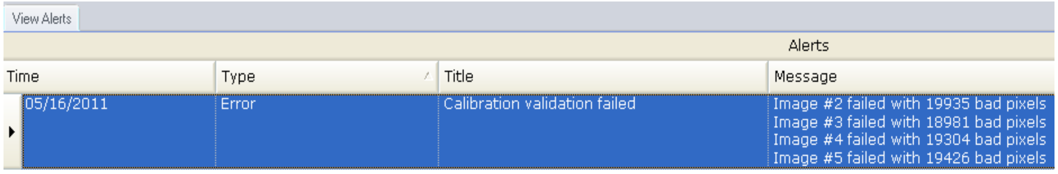


Requirement #15: Calibration can be re-run at any time and for as many times as the user desires. Only the last successful calibration will be used for analysis purposes.

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 #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 pixel location that failed and how it failed (high or 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.



View Alerts

Alerts			
Time	Type	Title	Message
05/16/2011	Error	Calibration validation failed	Image #2 failed with 19935 bad pixels Image #3 failed with 18981 bad pixels Image #4 failed with 19304 bad pixels Image #5 failed with 19426 bad pixels

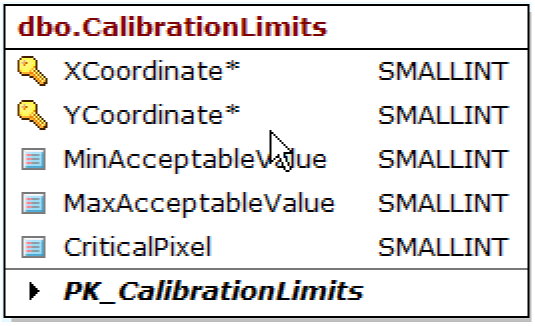
A second more detailed data file will be written into the current ClearView home directory containing the following data for each pixel that fails:

- 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 #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 as well as a critical bit.

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 #18: The calibration calculation is defined as follows:

- For each pixel in the 320x240 matrix, compare it to the corresponding limits established in the Calibration Pixel Limits csv file (dated 5/24/11) supplied by the Bio-statistics team. These limits were established based on the results documented in the EPIC ClearView Calibration Testing Report. The intensity of the pixel must fall between the high and low range, if it does not, it is considered a failure.

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

- If any of the critical pixels fail, the image will be considered invalid and the calibration process will fail.

- If there are failures in the non-critical areas of the images, a tolerance of 10 pixels will be allowed before the image will be considered a failure.

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

- 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.

- 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.

- 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 bad. The idea of critical pixels is not being used at the moment. Based on analysis documented in the EPIC ClearView Calibration Testing Report, there is currently no need to discern between types.

- Again based on the results documented in the EPIC ClearView Calibration Testing Report, the number of pixels that can fail before the image is considered invalid has changed from 10 to 7860. This is now the number that is being used in the calibration routine.

- 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.

- If all 10 images have a pixel failure rate less than 7860, the calibration set will pass.

Requirement #19: When a successful calibration is obtained, the images and a reference to the calibration is stored in the database. All scans performed after the calibration will reference the calibration.

Specification: After a successful calibration, the images are stored in the database for later reference.

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