



MPOST Java API Revision History

MPOST Java supports x86 and x64 architecture.

Version 2.80

- Added support for changes to BNF Status command. New property “GetLastBNFError()”.

Version 2.70

- Added support for new Audit commands. Host can retrieve the full audit data for all banknotes.

Version 2.60

- Added support for *Clear Audit* command. SC Advance units only.
- Ensure specific denominations can be enabled/disabled for Series 2000 products.

Version 2.50

- Ensure all documents are properly returned if they are inserted at the same time a disable command is sent.
- Capability for Flash download will now be properly set if the device powers up in the Flash Download state.
- Added support for Cashbox Cleanliness Reporting.

Version 2.40

- Ensure Stacked event comes before Power Up Complete Event if communications are lost during power up.
- Ensure the device is disabled immediately after the connection is closed.
- Improved device capabilities detection on start up with Series 2000 devices.
- Prevent Blue Screen issue when disconnecting device that is using Virtual Comm Port Drivers.

Version 2.30

- Updated the Open function to throw exceptions if the port is invalid for any reason.
- Ensure log file resources are properly released during the close function call.
- Events are now raised asynchronously to ensure thread safety between Host and API.
- Ensure message is properly built if received by serial port in multiple transactions.

Version 2.20

- <Version number skipped>

Version 2.10

- Improved the efficiency of the API to reduce general CPU consumption.
- Removed deprecation tag from the SetBezel Command.
- Optimized when some events are raised in exceptional cases.
- Added release history to the installation package
- Removed unnecessary logging system.

Version 2.00



MPOST Java API Revision History

- Updated Files – All files should be replaced to ensure compatibility.
 - MEI-java.jar
 - MEI-java-GUI.jar
 - Serial comm library (using version 2.1-7)
- Added several new events to the API.
 - PowerUpCompleteEvent – This event will fire after the event “OnPowerUp.” It signals that the device has completed all of its power up initialization and can now react normally to any user requests
 - SendMessageFailureEvent – This event is raised if there was a problem issuing a synchronous command to the device. For example, if a command to stack an escrowed note fails, the user would be informed through this event. The event will carry the Message that failed to be sent.
 - FailureDetectedEvent – This event is raised if any failure is detected.
 - FailureClearedEvent – This event is raised if a failure event has been corrected.
 - StackerFullClearedEvent – This event is designed to support older versions of Stackerless Code. It will fire when the Stacker Full condition was cleared.
 - NoteRetrievedEvent – This event is raised when the user removes a note from the rejected/returned position. This event is only supported on a certain device software revision and will not be raised unless supported.
- Deprecated several uncommonly used functions
 - SpecifyPupExtMode – This function was deprecated. It is not supported by any current device.
 - TransactionTimeout Property – The API will now fully manage the timing for internal messages.
 - DownloadTimeout Property – The API will now fully manage the timing during flash download.
 - SetBezel command – This function is not widely used and has been deprecated.
- New Properties/Methods
 - Overloaded the Open function. Now a device can be opened without having to specify a power up policy. The default will be the recommended PowerUP_A.
 - New Property called DisconnectTimeout can be set (in milliseconds) to specify the amount of time before the unit is considered “disconnected.” The Disconnected Event will fire if the device does not respond to the API after the specified amount of time (Default is 30 seconds). It is recommended not to set this value too low because the host could receive incorrect Disconnected Events. (Recommended minimum 4 seconds (4000ms))
 - New functions for Advanced Bookmark Feature. This feature is only supported by newer revisions of code. If the host calls “EnterAdvancedBookmarkMode()” the device will accept the next invalid document that is inserted (So long as it meets size restrictions 102mm > document > 160mm). This mode is exited as soon as a document is accepted or it can be canceled by calling “CancelAdvancedBookmarkMode()” Note – Valid notes that are inserted while this mode is active will automatically be returned.
- General fixes and improvements
 - Improved control of the order of events during power up and connection.
 - Important events will now be re-raised after power is lost or the connection is lost. These events include: Failure, Jammed, StackerFull, Escrow, Cashbox Removed.
 - Increased reliability of message handling within the API and added more robust retry ability.



MPOST Java API Revision History

- Fixed issue that caused barcode value to not be reported correctly.
- Fixed issue that caused Canadian units to not report the bill set correctly in series 2000 devices
- Function changes. Many functions will now throw an exception.
 - `getBill()` – Will now throw an exception if the `getDocType()` is not a bill.
 - `getCoupon()` – Will now throw an exception if the `getDocType()` is not a coupon.
 - `getBarcode()` – Will now throw an exception if the `getDocType()` is not a barcode.
 - `setEnableAcceptance()` – Will now throw an exception if the API is not connected.
 - `setAutoStack()` – Will now throw an exception if the API is not connected.
 - `setEnableBarCodes()` – Will now throw an exception if the API is not connected or the property is not supported.
 - `setEnableCouponExt()` – Will now throw an exception if the API is not connected or the property is not supported.
 - `setEnableNoPush()` – Will now throw an exception if the API is not connected or the property is not supported.
 - `setBillValueEnables()` – Will now throw an exception if the API is not connected or the list is not the correct size.