

BNR Tools - Release Notifications

BNR SDK for dotNet version 2.0.0

Release Date: July 2018

Software Part Number: 98602 2 000

The MEI® logo is a registered trademark.

Except as permitted under the relevant local legislation, no part of this publication may be copied, transmitted, transcribed or distributed in any form or by any means, or stored in a database or retrieval system, or translated into any language (natural or computer) without the prior written permission of Crane Payment Innovation (CPI).

Crane Payment Innovations reserves the right to change the product specifications at any time. While every effort has been made to ensure that the information in this publication is accurate, CPI disclaims any liability for any direct or indirect losses (howsoever caused) arising out of use or reliance on this information.

This document does not necessarily imply product availability.

Information in this document may change without notice.

Before You Begin

- By default, this SDK version works with USB driver version 2.0 and above
- Modify "BnrCtl.ini" file when the previous version driver is required

1. File Version Updates – 1.13.0 to 2.0.0

File Name	Current version	Previous version
BnrClientDotNet.exe	2.0.0.40000	1.13.0.40000
BnrClientDotNet40.exe	2.0.0.40000	1.13.0.40000
BnrCtlDotNet.dll	2.0.0.40000	1.13.0.40000
BnrCtlDotNet40.dll	2.0.0.40000	1.13.0.40000
BnrCtlW32.dll	2.0.0.40000	1.13.0.40000
BnrXmlRpc.dll	0.6.4.40000	0.6.4.40000
XmlParse.dll	0.6.5.40000	0.6.5.40000
XmlTok.dll	0.6.3.40000	0.6.3.40000
Bnr.inf	2.1.0.40000	2.1.0.40000
MegWinUsb.dll	1.1.0.40000	1.1.0.40000
BnrW32Serv.dll	1.0.0.40000	1.0.0.40000
BnrServ.exe	1.0.0.40000	1.0.0.40000
setupBNRWinUSBDriver.exe	1.1.3.40000	1.1.2.40000
BnrCtl.ini	1.13.0	1.10.0
All zipped into package: 98602 2 000- BNR dotNet SDK-2.0.0.zip	2.0.0	1.13.0

* Bold indicates a change since the previous release

2. Documentation Version

File Name	Current version	Previous version
BNR API For dotNet 67277-7-044 G8.pdf	G8	G7
BNR SDK For dotNet HTML Documentation	2.0.x	1.13.x

3. Compatibility - Minimum System Requirements

Windows Operating System	
	Windows XP Professional Edition SP2 (32-Bit or 64-Bit).
	Windows XP, 7, or 8 Embedded (32-Bit or 64-Bit) with WinUsb Driver package included.
	Windows 7 Professional (32-Bit or 64-Bit).
	Windows 8 Pro (32-Bit or 64-Bit).
	Windows 8.1 Pro (32-Bit or 64-Bit).
	Windows 10 Pro (32-Bit or 64-Bit).

Runtime		Version	
	.NET Framework	2.0	4.0
	VC++	9.0	10.0

PC platform	
	Celeron 600 MHz or better processor
	20 MB of system memory available for the API
	30 MB of free hard-disk space for full installation
	USB 1.1, 2.0 and 3.0

Development platform	Version
Visual C++ 2008 SP1 (VC++ 9.0 runtime) for .NET framework prior to 4.0	9.0
Visual C++ 2010 SP1 (VC++ 10.0 runtime) for .NET framework 4.0 and later	10.0

4. Compatibility - BNR firmware requirements

The BNR SDK for dotNet version 2.0.0 is backwards compatible with all BNR firmware versions package 1.0.x or higher. To enable all functionality added in dotNet 2.0.0, Firmware 2.0.0 or higher must be used.

5. Compatibility - Special instructions for upgrade

Applications running with BNR SDK for dotNet of prior versions:

In order to work with BNR SDK for dotNet 2.0.0, the Host application must be recompiled to take the new classes, properties and fields into account even if the new functionalities introduced in the 2.0.0 version are not used.

6. Compatibility - Functional

Functional compatibility is preserved relatively to previous BNR SDK for dotNet version.

7. Compatibility – With other tools

Tool name	Language / Platform	Versions
BnrClientDotNet.exe	Windows	2.0.0
BnrFlash for Windows	Windows	2.0.0

8. Reason for the modification (summarized)

- Support new features introduced in BNR Firmware version 2.0.0
- methods that give an easy way to:
 - Reset Cashbox cash units counts
 - Set Loader cash units counts.
- USB driver improvement
- WinUsb Driver Certificate
- Use of Win32 API 2.0.0

9. Software Package 1.13.0 to 2.0.0 – Change list (detailed)

Support of BNR Advance											
Change Type: Product Enhancement											
Reason for Change: To support new features introduced in BNR Firmware version 2.0.0, related to BNR Advance device.											
<p>Description:</p> <p>Compared to BNR, BNR Advance device has been added new electro-mechanical elements, that need to be presented to Host application and tools such as BNR Support Tools through API. Hereafter are listed these new elements and impact on the SDK.</p> <p>New electro-mechanical element: Spine Interface Diverter.</p> <p>New constants and types introduced:</p> <pre>Mei::Bnr::Module::Element::SpineInterfaceDiverter::Status</pre> <p>Properties</p> <table border="1"> <tbody> <tr> <td>SpineInterfaceDiverter::FunctionalStatus</td><td>FunctionalStatus [get] Get SpineInterfaceDiverter FunctionalStatus.</td></tr> <tr> <td>SpineInterfaceDiverter::ErrorCode</td><td>ErrorCode [get] Get SpineInterfaceDiverter ErrorCode.</td></tr> <tr> <td>T_ModuleId</td><td>Id [get] Get Element Id.</td></tr> <tr> <td>OperationalStatus</td><td>OperationalStatus [get] Get Element OperationalStatus.</td></tr> </tbody> </table> <p>Spine Interface Diverter status information is obtained when reading MainModule::Status property.</p> <p>A new attribute to retrieve Spine Interface Diverter cycle count has been added to MainModule::UseHistory.</p> <table border="1"> <tbody> <tr> <td>T_LongAbsoluteCount</td><td>Mei::Bnr::Module::MainModule::UseHistory::SpineInterfaceDiverterCycleCount</td></tr> </tbody> </table> <p>Get The number of cycles of the Spine Interface Diverter.</p>		SpineInterfaceDiverter::FunctionalStatus	FunctionalStatus [get] Get SpineInterfaceDiverter FunctionalStatus.	SpineInterfaceDiverter::ErrorCode	ErrorCode [get] Get SpineInterfaceDiverter ErrorCode.	T_ModuleId	Id [get] Get Element Id.	OperationalStatus	OperationalStatus [get] Get Element OperationalStatus.	T_LongAbsoluteCount	Mei::Bnr::Module::MainModule::UseHistory::SpineInterfaceDiverterCycleCount
SpineInterfaceDiverter::FunctionalStatus	FunctionalStatus [get] Get SpineInterfaceDiverter FunctionalStatus.										
SpineInterfaceDiverter::ErrorCode	ErrorCode [get] Get SpineInterfaceDiverter ErrorCode.										
T_ModuleId	Id [get] Get Element Id.										
OperationalStatus	OperationalStatus [get] Get Element OperationalStatus.										
T_LongAbsoluteCount	Mei::Bnr::Module::MainModule::UseHistory::SpineInterfaceDiverterCycleCount										
Case fixed: n/a											

Simplified Cashbox and Loader Cash Units management

Change Type: Product Enhancement

Reason for Change: To provide methods that give an easy way to:

- Reset Cashbox cash units counts
- Set Loader cash units counts.

Description:

New functions have been introduced to allow update of Cashbox and Loader cash unit counts in one single call:

- `Bnr::ResetCashboxCuContent`
- `Bnr::SetLoaderCuContent`

Case fixed: n/a

WinUsb Driver Installer

Change Type: Bug Fix

Reason for Change: In previous delivered version, USB driver presence checking was faulty on some operating systems, and prevented correct BNR USB driver installation.

Description:

Driver Installation failed with "WinUSB is not present on your system..." message on Windows XP and Windows 8.1 while WinUsb driver is available. Way of checking WinUSB driver presence has been fixed.

Also readme.txt file has been reworked to make better difference between change history of driver itself vs. change history of the installer software.

Case fixed: 140031 - Difference and impact of the new BNR Win32 SDK drivers installer

WinUsb Driver Certificate

Change Type: Product Enhancement

Reason for Change: To provide a signed driver with new certificate.

Description:

Certificate used to sign the driver had reached his expiration date, so it was necessary to sign it again with a new valid certificate (new expiration date: 26/04/2021).

This has no impact on functionality.

Case fixed: n/a

Use of Win32 API 2.0.0

Version 2.0.0 of Win32 API for BNR brings additional changes to support BNR Advance, and version 2.0.0 of BNR .Net SDK takes advantage of them.

This includes ability to handle more than 60 denominations in BNR. Indeed .Net API now internally uses new functions:

- `bnr_QueryDenominationsEx()`
- `bnr_UpdateDenominationsEx()`
- `bnr_QueryBillsetIdsEx()`
- `bnr_QueryCashUnitEx()`
- `bnr_UpdateCashUnitEx()`
- `bnr_ConfigureCashUnitEx()`
- `bnr_GetBillAcceptanceHistoryEx()`

See BNR SDK for Win32 2.0.0 release notes for more information.

10. History of changes

----- Software Package 1.10.0 to 1.13.0 -----

Support for New Capabilities
Change Type: Product Enhancement
Reason for Change: To provide API support for new features that were introduced in BNR Firmware versions 1.11.1/2/3.
Replication Instructions: N/A
Several special configuration options were introduced in BNR Firmware versions 1.11.1/2/3. At the time, these capabilities could only be enabled using the "Special Configuration" option. dotNET API 2.0.0 introduces class <code>Mei::Bnr::Capabilities</code> to access these features:
<i>reportUsbConsumption</i> : Specifies whether real max USB line consumption is reported on usb configuration descriptor instead of 0mA. Default value is set to FALSE. (FW 1.11.1)
<i>autoRetract</i> : Specifies whether bill will be automatically retracted to positioner when jam occurred during bill presenting at inlet or outlet. Default value is set to FALSE. (FW 1.11.2)
<i>rejectViaOutlet</i> : Specifies whether unknown or inhibited notes are rejected via the BNR's Outlet instead of the Inlet. Default value is set to FALSE. (FW 1.11.3)
NOTE: For more detailed descriptions of the features above, please refer to BNR Firmware 2.0.0 Release Notes (MEI-BNR-RNF-043).

Preventative Maintenance Information Stored in Memory
Change Type: Product Enhancement
Reason for Change: Allows the customer to retrieve preventative maintenance (PM) information.
Replication Instructions: N/A
The following information is provided with this new feature:
<ul style="list-style-type: none"> • The level of maintenance that has been performed (from 1 to 3) • The date and SystemCycle count at the time maintenance was performed
The information is saved to the BNR memory and can be retrieved to check that the preventative maintenance has been completed, based on the recommendations published by CPI. This information can be obtained from the following:
<i>Mei::Bnr::PreventativeMaintenance;</i>
This read-only property returns the current BNR PM information, shown below:
<pre>public class PreventativeMaintenance { public DateTime CurrentDateTime { get; } public DateTime LastL1Date { get; } public LongAbsoluteCount LastL1SystemCycleCount { get; } public DateTime LastL2Date { get; } public LongAbsoluteCount LastL2SystemCycleCount { get; } public DateTime LastL3Date { get; } public LongAbsoluteCount LastL3SystemCycleCount { get; } public LongAbsoluteCount SystemCycleCount { get; } }</pre>

Mei::Bnr::PreventativeMaintenanceCompleted(PreventativeMaintenanceLevel pmLevel);

This method updates the BNR preventative maintenance level information corresponding to the level passed as parameter:

```
public enum PreventativeMaintenanceLevel
{
    Level1 = 0, /**< PM Level 1: Basic cleaning of device without parts removal, done on-site.*/
    Level2 = 1, /**< PM Level 2: Thorough cleaning of device with parts removal, done in workshop.*/
    Level3 = 2, /**< PM Level 3: Thorough cleaning of device with parts removal and replacement, done in workshop.*/
}
```

NOTE: In the BNR firmware, preventative maintenance information is currently stored under the form of a string in Main Module UserData. This implies that when an application implements Preventative Maintenance features, it must not directly modify this attribute (e.g. do not use Module.Identification.UserData on Main Module) when employing the SDK feature. More details are provided in application note "AN-06 Preventative Maintenance Integration".

winUSB Driver Re-signed

Change Type: Product Enhancement

Reason for Change: The certificate used to sign the USB driver has reached its expiration date. It was necessary to resign the driver with a new valid certificate.

Replication Instructions: N/A

This change has no impact on functionality.

winUSB Driver Installer

Change Type: Product Enhancement

Reason for Change: Depending on the machine, it is possible that lower-level drivers are not present. The new installer checks that these drivers are present before allowing installation of the BNR winUSB driver.

Replication Instructions: N/A

Some customers deploying their application on Windows Embedded 7 and 8, produce customized Windows images by selecting the needed components through the Microsoft Image Configuration Editor tool. If WinUsb.sys low level driver is not included in this image, the communication with the BNR cannot be established. To address the missing driver, installation of the BNR WinUsb driver now fails if WinUsb.sys is not found.

Support for Windows 8 and 10

Change Type: Product Enhancement

Reason for Change: To validate proper API functionality with newer Windows operating systems

Replication Instructions: N/A

Tests have been conducted under Windows 8 and 10 environments with no impact on functionality of the BNR API.

Loader Exchange History

Change Type: Bug Fix

Reason for Change: Compared to underlying Win32 API, class definition of Mei::Bnr::Module::Loader::BillCassetteExchangeHistory was missing read-only property BillInvalidCount.

Replication Instructions: N/A

The property has been added.

----- Software Package 1.9.0 to 1.10.0 -----

New read-only Bnr:BillsetId property added	
Change Type:	New Functionality
Reason for Change:	A variety of new legal requirements and certification processes are being adopted by governments to control the methods used by bank note validators to authenticate banknotes. In some cases this is driving CPI to develop new country specific Recognition Algorithms and billsets (software). I.e. Russia is the first country to enforce unique requirements with what is branded as the Russian 4 of 6 certification process. To ensure billset compatibility, CPI has expanded the BNR/BNR6 architecture.
Replication Instructions:	N/A
<p>Description:</p> <p>The BNR now supports two types of billsets. BNRB, the global standard, and BNRU, which supports UV recognition (currently required to support the Russian 4 of 6 certification)</p> <p>In Firmware 1.12.0 the new command bnr.QueryBillsetIds() allows BNR Flash to validate compatible billsets on a unit. BNRB and BNRU DLD files cannot coexist on a single unit for the same country code. However, mixed currency DLD files can be of different recognition types. I.e. BNRB Euro DLD can coexist with BNRU Russian DLD in the same Main Module. On the other hand, BNRB Russian DLD cannot coexist with a BNRU Russian DLD in the same Main Module.</p> <p>Additionally, all variants must be of the same version (4 or 5).</p> <p>With BNR .Net API, the BillsetId list can be retrieved using the new read-only property named Bnr::QueryBillsetIds</p>	

BNA6 with Spine Electronics	
Change Type:	New Functionality
Reason for Change:	BNA6 was initially deployed without electronics to detect if the Spine was in the open or closed position. As a product enhancement, CPI added electronics to support a Smart Spine. These electronics detect the position of the Spine door (open or closed). The architecture of the Firmware and API have been expanded to enable this new capability.
Replication Instructions:	N/A
<p>Firmware 1.11.0 introduces the parameter smartSpineActivation for the commands SetSystemConfiguration and GetSystemConfiguration.</p> <p>This parameter can be set to On (1) and AutoDetect (0). For complete definitions of On and AutoDetect please refer to the API HTML documentation</p> <p>In BNR .Net API, an additional read/write property was added to Bnr:SystemConfiguration to retrieve and/or modify the smartSpineActivation parameter in the BNR.</p>	

----- Software Package 1.7.1 to 1.9.0 -----

BNA6F Configuration Added	
Change Type:	Product Enhancement
Reason for Change:	Preparation for future firmware capabilities
Replication Instructions:	N/A
<p>Description:</p> <p>The value BNA_6F has been added to the enumeration BnrType. This enables configuring the unit for the BNA6F type. Please note that FW 1.9.0 does not support this new value.</p>	

Module.GetMaintenanceInfo Updated
--

Change Type:	Product Enhancement
Reason for Change:	Handling new functionality of the BNR firmware version 1.9.0
Replication Instructions:	N/A
<p>Description:</p> <p>To support the new features of the firmware 1.9.0, the class Mei.Bnr.Module.MaintenanceInfo returned by the command module.MaintenanceInfo has been expanded to report a larger set of information about the need for maintenance of the BNR. With respect to the previous API 1.7.1, the following properties have been added:</p> <ul style="list-style-type: none"> • A list of (one or more) reasons that caused the need for maintenance. The reasons can be one of the following: <ul style="list-style-type: none"> ○ At least one of the sensor is dirty. ○ The number of cycles since the last maintenance has passed its limit. ○ The bundler is slipping (valid only for the bundler). • A list storing the status of the sensors, expressed as a percentage from 0% (dirty) to 100%. • Slippage of the bundler from 0% (i.e. not slipping) to 100%. This field is valid only for the bundler. <p>The history reports section of the Audit Files has been extended consequently. Please refer to the html documentation of the code to get more details about these new properties and their types.</p>	

----- Software Package 1.7.0 to 1.7.1 -----

Diagnostics File Generation	
Change Type:	Bug Fix
Reason for Change:	The previous release of the API/BNR Flash did not correctly download the History Reports content of the diagnostics file when the bnr.generateAuditReport() command was issued
Replication Instructions:	N/A
<p>Description:</p> <ul style="list-style-type: none"> • The generateAuditReport command now successfully downloads the entire contents of the diagnostics file to ensure all relevant data is collected. 	

----- Software Package 1.6.0 to 1.7.0 -----

SystemFailureHistory additions	
Change Type:	New Functionality
Reason for Change:	Handling new functionality of the BNR firmware version 1.8.1
Replication Instructions:	N/A
<p>Description:</p> <ul style="list-style-type: none"> • Audit fields have been added to the SystemFailureHistory to indicate <ul style="list-style-type: none"> ○ Number of times the reset command is sent to the BNR while any cover (Positioner, RecognitionSensor or Spine) is open ○ Number of times the reset command is sent to the BNR while the Interlock is open ○ Number of times by section a jam has occurred in the Main Module ○ Number of times by section where jam recovery started 	

BillDispenseHistory update	
Change Type:	Product enhancement
Reason for Change:	Handling new functionality of the BNR firmware version 1.8.1
Replication Instructions:	N/A
Description:	
<ul style="list-style-type: none"> Audit field has been added to BillDispenseHistory to indicate the number of times bnr.Dispense has been called when a mix algorithm was specified. Existing fields in BillDispenseHistory were only incremented when calling bnr.Denominate and not bnr.Dispense. This has been corrected such that bnr.Dispense will now increment those counters. For legacy purposes, the bnr.DenominateCount field will remain, but the meaning will change for the reasons mentioned earlier. 	

Acceptance History addition	
Change Type:	New Functionality
Reason for Change:	Handling new functionality of the BNR firmware version 1.8.1
Replication Instructions:	N/A
Description:	
<ul style="list-style-type: none"> In order to help our customers get a more accurate BNR configuration, an audit counter notesStackedWhileStackerFull will be added to indicate the number of notes that under normal conditions are routed to the recycler, but were stacked because the recycler was full. This gives our customer valuable information to know, by denomination, the excess routed to the cashbox. 	

Module.setIdentification now correctly writes space ' ' characters to module	
Change Type:	Bug fix
Originating Report:	CCR FXDU-9CSJHY
Affected API Versions:	1.6.0 and earlier
Affected FW Versions:	Not applicable
Symptoms	
When issuing module.SetIdentification, any space characters ' ' in the userData was erroneously replaced by a printable ASCII character '!'. This has been corrected and now any printable ASCII character can be written in the userData of module.SetIdentification.	
Impact:	
In the affected API version, one would not be able to write spaces to the module userData.	
Workaround prior to release API version:	
When writing user data via module.SetIdentification, choose a different character besides ' ' to separate strings.	
Errant comma ',' in BnrPropW32.h removed	
Change Type:	Bug fix
Originating Report:	CCR LAFR-9BPVMA
Affected API Versions:	1.6.0 and earlier
Affected FW Versions:	Not applicable
Symptoms	
A compilation error (unexpected symbol) was caused when an application linked with the affected version of the API. This was due to an extra comma on line 2214 of the file BnrPropW32.h. This comma has been deleted in this release.	
Impact	
Would not be able to compile against the affected API version(s).	
Workaround	

In affected API version(s), delete the comma at the end of line 2214 in BnrPropW32.h, save and recompile.

API Documentation Addition for bnr.ConfigureCashUnit

Change Type: Information Update

Originating Report: CCR BMES-9AZQMU

Affected API Versions: 1.6.0 and earlier

Affected FW Versions: Not applicable

Details

When adding primary/secondary cash types (denominations) to a recycler of varying lengths, it was not documented how much the note length of the same country and value could vary between subsequent series of notes. This has been documented in the HTML documentation.

Impact

There was no explanation for this case in the HTML documentation for bnr.ConfigureCashUnit for the error XFS_E_CDR_CASH_UNIT_ERROR. This is now well documented. Ultimately it cues our customers to check the differences in length between series of notes added as primary/secondary cash types to the same recycler or loader. This information can be found in the identification cards for the currency in question.

Workaround

None as this is information

----- 1.5.0 to 1.6.0 -----

Use of Win32 API 1.6.0

Change Type: Product enhancement / New functionality

Reason for Change: Handling new functionality of the BNR firmware version 1.7.0

Replication Instructions: N/A

Version 1.6.0 of the Win32 API for BNR brings the use of a new USB driver, a 32-bit and 64-bit version of each library, preventative maintenance functionality, history reports functionality, new counters, etc... Please refer to the Release Notes of the BNR SDK for Win32 1.6.0 for more information.

Availability of .Net 4.0

Change Type: Product enhancement

Reason for Change: Compatibility with .NET framework 4.0

Replication Instructions: N/A

- BnrCtlDotNet.dll and BnrClientDotNet.exe are available and compatible with a framework version between 2.0 and 3.5.
- In addition, two new files, BnrCtlDotNet40.dll and BnrClientDotNet40.exe, are available in the latest API. They maintain the functionality of BnrCtlDotNet.dll and BnrClientDotNet.exe, but are compiled for .NET framework 4.0.

Preventative Maintenance

Change Type: New functionality

Reason for Change: Handling new functionality of the BNR firmware version 1.7.0

Replication Instructions: N/A

In BNR firmware version 1.7.0, a new feature has been added for preventative maintenance. New commands have been added to the API to access this feature:

- `Module::MaintenanceInfo` retrieves the maintenance information related to a particular module.
- `Module::MaintenanceInfo.MaintenanceInterval` allows the Host to update the maintenance interval for a particular module (i.e. the number of cycles after which a maintenance operation will be recommended). This value is persistent over power cycles, default value is 50000 except for Loader where it is 20000.
- `Module::MaintenanceDone` allows the Host to clear the number of cycles since last maintenance for a particular module, and thus remove it from the list of the modules for which a maintenance operation is recommended if needed

These properties and method are available from BNR firmware version 1.7.0. Using them with an older firmware version will lead to a `BnrXfsException` with Code set to `NotSupported`.

A new preventative maintenance status event has been implemented

`MaintenanceStatusChanged`.

This new event returns a list of modules for which a maintenance operation is recommended.

This event is fired by the BNR each time a module is added or removed to the list.

Note: For compatibility reasons, this event is disabled by default; to enable it, the Host must use the new property `Bnr::EnabledStatusEvents` (see below).

Status Event enabling

Change Type: New functionality

Reason for Change: Handling new functionality of the BNR firmware version 1.7.0

Replication Instructions: N/A

In BNR firmware version 1.7.0, a new feature has been added to enable or disable individual status events that are or are not desired by the host.

A new property has been added to the API to access this feature:

- `Bnr::EnabledStatusEvents` - When read, this property retrieves the list of the Status events that are enabled. When written, it allows the Host to send to the BNR the list of the Status events that should be enabled

To be compatible with current integrations, at power-up, the enabled status events are those existing prior to firmware version 1.7.0:

- `CashUnitChanged`
- `CashUnitConfigurationChanged`
- `CashUnitThreshold`
- `DeviceStatusChanged`
- `CashTaken`
- `CashAvailable`

Please note `MaintenanceStatusChanged` is disabled by default.

This property is available from BNR firmware version 1.7.0. Using it with an older firmware version will lead to a `BnrXfsException` with Code set to `NotSupported`.

Revert to Boot (Firmware)	
Change Type:	New functionality
Reason for Change:	Handling new functionality of the BNR firmware version 1.7.0
Replication Instructions:	N/A
<p>In BNR firmware version 1.7.0, a new feature has been added to erase the application component of the module and, in case of the MainModule, to erase also the billset and the Logical Cash Units.</p> <p>New method Module::RevertToBoot() has been added to the API to access this feature.</p> <p>By using this method, production and/or customers have the possibility to deliver a "Vanilla" BNR configuration, independently of the current firmware package revision used during CTO.</p> <p>This method is available from BNR firmware version 1.7.0. Using it with an older firmware version will lead to a BnrXfsException with Code set to NotSupported.</p> <p>Because all modules in the BNR ultimately depend on the Main Module and/or the spine, the order in which this command is called should be</p> <ol style="list-style-type: none"> 1. Recyclers 2. Cashbox (if Smart Cashbox Only, otherwise XFS_E_FAILURE is returned) 3. Loader (if present, otherwise XFS_E_FAILURE is returned) 4. Barcode Reader (if present, otherwise XFS_E_FAILURE will be returned) 5. Spine 6. Main Module <p>Please consult with MEI Tech Support/Product Management when implementing</p>	

Firmware package version	
Change Type:	New functionality
Reason for Change:	Handling new functionality of the BNR firmware version 1.7.0
Replication Instructions:	N/A
<p>In BNR firmware version 1.7.0, a new feature has been added to the firmware to allow the host to query the BNR firmware package version, and indicate if each module application is up-to-date regarding this version.</p> <p>New property Bnr::FirmwarePackageVersion has been added to the API to access this feature.</p> <p>This property will return the main module application version in any case and will set isUpToDate to True if all other modules contain the respective application versions contained in the package version. Otherwise isUpToDate will be set to False</p> <p>This property is available from BNR firmware version 1.7.0. Using it with an older firmware version will lead to a BnrXfsException with Code set to NotSupported.</p> <p>Example:</p> <ul style="list-style-type: none"> • BNR with FW 1.7 returns 1.7.0 and isUpToDate set to TRUE • BNR with FW 1.7 with a loader that has an earlier firmware version returns 1.7.0 with isUpToDate set to FALSE 	

History reports	
Change Type:	New functionality
Reason for Change:	Handling new functionality of the BNR firmware version 1.7.0
Replication Instructions:	N/A
<p>In BNR firmware version 1.7.0, a new feature has been added to help analyzing BNR performance using snapshots of BNR history counters, taken under different conditions. These snapshots are called History Reports</p> <p>New methods and properties have been added the API to access this feature:</p> <ul style="list-style-type: none"> • Bnr::SaveHistoryReport() allows the Host to ask the BNR to save a new report, it accepts a string parameters of 20 characters to add a short description of the report • Bnr::HistoryReportList retrieves the history reports saved in the BNR, if no report exists in the BNR, a BnrXfsException with Code set to NoExist is raised. <p>These methods and properties are available from BNR firmware version 1.7.0. Using them with an older firmware version will lead to a BnrXfsException with Code set to NotSupported.</p>	

Audit Report Contents	
Change Type:	New functionality
Reason for Change:	Handling new functionality of the BNR firmware version 1.7.0
Replication Instructions:	N/A
<p>In BNR firmware version 1.7.0, BNR HistoryReports are captured in the AuditReport when calling the method BNR::GenerateAuditReport</p> <p>To offset the increase in size of the generated AuditReport due to the HistoryReports, billTransportEvent reports are no longer captured in the file by default (low priority reports that are rarely used).</p> <p>To capture the billTransportEvent the BnrCtl.ini file must be modified to set ENABLE_FULL_AUDIT_REPORT = 1</p> <p>Please note billTransportEvent are stored in the BNR regardless, this only pertains to the generated AuditReport</p>	

BNR Reporting Mode	
Change Type:	New functionality
Reason for Change:	Handling new functionality of the BNR firmware version 1.7.0
Replication Instructions:	N/A
<p>In BNR firmware version 1.7.0, a new reporting mode has been added to the BNR. This mode is configured using the BNR::Capabilities class and setting the property ReportingMode:</p> <p>ReportingMode defines if the BNR creates a SimpleFailureReport (Normal) or a BillTransportErrorReport (Full) in case of failure detection when no bill is being transported. (BillTransportErrorReports contain more details than a SimpleFailureReport).</p> <p>Default and recommended value is Normal</p> <p>FullReportingMode should be used only in cases where it is necessary to have logs related to unexplained failures in the field that occur when there is no note movement.</p> <p>Please refer to the Release Notes of the BNR SDK for Win32 1.6.0 for more information.</p>	

AuditReport (Diagnostics File) Compression	
Change Type:	Product enhancement
Reason for Change:	To reduce the size of the AuditReport
Replication Instructions:	N/A
<p>The ability to compress diagnostics files has been added to the method BNR::GenerateAuditReport. By default the diagnostics file will not be compressed.</p> <p>To enable compression the BnrCtl.ini file must be modified to:</p> <p>ENABLE_AUDIT_REPORT_FILE_COMPRESSION = 1</p> <p>Please note that the compression occurs in the API and not in the BNR itself</p>	

Disabling of Front USB Service Port	
Change Type:	New functionality
Reason for Change:	Security Option
Replication Instructions:	N/A
<p>In BNR firmware version 1.7.0 the ability to disable the use of the front USB service port was added. By default the service port is enabled.</p> <p>To disable the service port use the class BNR::Capabilities and set the property</p> <p>allowUsbFrontSwitch = False</p>	

Error counters	
Change Type:	New functionality
Reason for Change:	Handling new functionality of the BNR firmware version 1.7.0
Replication Instructions:	N/A
<p>In BNR firmware version 1.7.0, new history counters have been added:</p> <ul style="list-style-type: none"> noFeedCount, badFeedCount, badExtractCount in Loader::FailureHistory:: TransportErrorDetails class NotArmedCount in Cashbox::FailureHistory class MissingModuleCount , IncompatibleSoftwareCount and ConfigurationErrorCount in Bnr::SystemFailureHistory 	

Documentation	
Change Type:	Product enhancement
Reason for Change:	BMES-8Y2QZM: API Documentation Correction Needed for CashInEnd
Replication Instructions:	N/A
<p>Html documentation and "Programmers guide" have been updated for new commands and properties.</p>	

Other	
Change Type:	Correction
Reason for Change:	Handling new functionality of the BNR firmware version 1.7.0
Replication Instructions:	N/A
<p>The value StaysOpened in Flap::ErrorCode has been added for consistency with other APIs</p>	

BNA6		
Change Type:	Product enhancement	
Reason for Change:	New exceptions for handling BNA6	
Replication Instructions:	N/A	
<p>In BNR firmware version 1.7.0, full support for the new BNA6 model has been implemented. The following differences between the BNR and the BNA6 should be taken into account:</p> <ul style="list-style-type: none">• The escrow limit for the BNA6 becomes 20. Please note that a BNA6 Main Module contains hardware differences that allow the 20 note escrow. The BNR max escrow size remains 15.• The commands methods BNR::Denominate, BNR::Dispense, BNR::Present, and BNR::BillDispenseHistory are not supported by BNA6 and calls to these commands will return an exception. All other BNR API functions are valid for BNA6.• The method BNR::Capabilities returns different results, defined in the table below, depending on the model of the Main Module, BNA6 Vs Standard BNR. Please note only differences are shown:		
	BNRx	BNA6
Dispense	True	False
EscrowSize	15	20
MaxOutBills	15	20

<p style="text-align: center;">----- 1.4.0 to 1.5.0 -----</p> <p>USB Communication :</p> <ul style="list-style-type: none"> Use of Win32 API 1.5.0 for compatibility with the new USB driver <p>Ini file</p> <ul style="list-style-type: none"> New option "USE_LEGACY_DRIVER" to use the old USB driver. Not recommended for new integration. <p>Documentation</p> <ul style="list-style-type: none"> Updated description for CashUnit and related classes (CCR BMES-8M7J9U) <p>Other</p> <ul style="list-style-type: none"> New button "CancelWaitingCashTaken" in the BnrClientDotNet application. 	
<p style="text-align: center;">----- 1.3.0 to 1.4.0 -----</p> <p>Barcode reader support</p> <ul style="list-style-type: none"> SystemConfiguration class has been updated with new field for barcode reader activation Module::Status and Bnr::SystemStatus have been updated with barcode reader status. A new BarcodeReader class has been defined. New intermediate event BccInserted is defined to communicate the barcode number to the host New write-only property Bnr::RecognitionResult, allows the host to communicate the recognition result for a coupon to the BNR. <p>Other new methods</p> <ul style="list-style-type: none"> Bnr::Eject() has been added to allow the host to eject bills waiting at bezel. Bnr::CancelWaitingCashTaken() allows to the host to cancel an asynchronous command while BNR is waiting for the user to remove bills waiting at bezel (after event CashAvailable has been received). <p>Documentation</p>	

- Html documentation and "Programmers guide" have been updated with barcode reader status and more explanation about the synchronous and asynchronous commands.

Use of Win32 API 1.4.0

- Version 1.4.0 of Win32 API for BNR is necessary to allow the barcode reader functionality, the Bnr::Eject and Bnr::CancelWaitingCashTaken methods

----- 1.2.1 to 1.3.0 -----

- **Cash management:**
New Optimum change algorithm for Denominate or Dispense commands available from BNR FW version 1.4.0:
The BNR chooses the banknotes to be distributed in order to obtain the total amount in a way that slows down cashbox filling. As long as the low denomination Recyclers are not near to full, change is determined like with the MinBills algorithm. But when a Recycler becomes nearly full (5/6 of Full threshold), this algorithm will try to put more of this denomination in the change so that the Recycler doesn't become full and this denomination doesn't start to be cashed.
- **CashUnit:**
Fix confusion in extended counters update using the Bnr ::UpdateCashUnit method. Happened when trying to set extended counters to a value different than zero.
- **Documentation:**
Html documentation and "Programmers guide" have been updated with new change algorithm.
- **Use of Win32 API 1.3.0:**
Version 1.3.0 of Win32 API for BNR fixes few issues about USB communication on special situations and erroneous date on Audit Reports.

----- 1.1.0 to 1.2.1 -----

- **Use of Win32 API 1.2.1**
Version 1.2.1 of Win32 API for BNR embeds all USB recoveries. The API signals a USB error after having tried every possible operation to recover from USB failure.
Ini filename has been changed with this release. It is now called "BnrCtl.ini". Old filename is deprecated even if it can still be used for compatibility
- **Documentation**
Html documentation and "Programmers guide" have been updated. Additional information for Bnr::SelfTest() command and how to manage USB recoveries.
- **Events**
Instance of class CashUnit available on CashUnitChanged and CashUnitConfigurationChanged events (data parameter).
- **Retract / reject commands**
Fix for crash on Bnr::Retract and Bnr::Reject commands.
(CCR # EXBD-8BRETG)
- **Property**
Fix for Bnr::Status property issue concerning the positionStatusList field. Its length was sometimes equal to 0 or 1 instead of 2.

----- 1.0.0 to 1.1.0 -----

- New Reject and Retract command:
The new commands retract and reject are now available for handling cash. See html documentation for more details.
(Available in BNR from future FW version 1.3.0)
- Throwing exceptions:
Some Bnr commands were not throwing Bnr exceptions on error.
- Audit report:
Added new data available with BNR firmware 1.1.x.
- Supporting Usb 2.0 hub:
Modified for USB2.0 compliance and fully tested with an USB 2.0 hub.
- Bnr.UpdateCashUnit() fix:
ThresholdMode field not transmitted by the Bnr.UpdateCashUnit() function provoked sometimes the BNR to reject this command.
- Api.log fixes:
The "Cash Unit Changed" notification on Status occurred event was not logged into the Api.log.
A Cash Unit structure was logged into Api.log on QueryCashUnit returning "busy".
- Api Win32 1.1.0 compliance:
Api .Net updated to support ApiWin32 1.1.0.
- Documentation:
New html documentation and "Programmers guide" provided

----- 1.0.0 -----

First official release for customers