



Date: 2005-08-23

## [Interface Specification]

# BDI FRONT END SOFTWARE INTERFACE FOR THE LEIR ORBIT MEASUREMENT SYSTEM [BPNCO]

### *Abstract*

As part of the Beam Instrumentation Software Common Tools and Interfaces project (BISCoTI) [1], this document presents and explains the LEIR Orbit Front End Software Interface defined and implemented using FESA [2] (BPNCO class) following the standard interface for beam position measurements (BdiStdPosition) [3] and the document "Guidelines and conventions for defining interfaces of equipment developed using FESA" [4].

### *Prepared by :*

Danil Korchagin [AB/BDI]  
Michael Ludwig [AB/BDI]

### *Checked by :*

Jean-Jacques Gras [AB/BDI]  
Michael Ludwig [AB/BDI]

### *Approval List:*

Ana Guerrero Ollacarizqueta [AB/BDI]  
Michael Ludwig [AB/BDI]  
Lars Soby [AB/BDI]  
Jorg Wenninger [AB/OP]  
Fabio Follin [AB/OP]  
Vito Baggiolini [AB/CO]



## **Table of Contents**

<b>1. INTRODUCTION.....</b>	<b>4</b>
<b>2. CONVENTIONS .....</b>	<b>4</b>
<b>3. OVERVIEW .....</b>	<b>5</b>
<b>4. INTERFACE INHERITANCE.....</b>	<b>5</b>
<b>5. CUSTOM TYPES.....</b>	<b>6</b>
5.1    CONSTANTS .....	6
5.2    ENUM TYPES.....	7
5.3    BIT ENUM TYPES .....	11
<b>6. PROPERTIES.....</b>	<b>14</b>
6.1    DESCRIPTION .....	14
6.2    INITALLDEV.....	15
6.3    INIT .....	15
6.4    STATUS .....	16
6.5    RESET.....	16
6.6    SETTING .....	17
6.7    EXPERTSETTING.....	18
6.8    GURUSETTING .....	19
6.9    ACQUISITION .....	20
6.10   INTERMEDIATEACQUISITION .....	21
6.11   SUMMARYACQUISITION .....	22
6.12   EXPERTACQUISITION.....	23
6.13   SINGLEACQUISITION .....	24
6.14   EXPERTCALIBRATION.....	25
<b>7. REFERENCES.....</b>	<b>26</b>

## 1. INTRODUCTION

As part of the Beam Instrumentation Software Common Tools and Interfaces project (BISCoTI) [1], this document presents and explains the LEIR Orbit Front End Software Interface defined and implemented using FESA [2] (BPNCO class).

Every BDI instrument interfaces will inherit from 'virtual' interfaces corresponding to its main measured beam observable. The considered observables are the following:

- Beam Position [3],
- Beam Intensity,
- Beam Profile,
- Beam Losses,
- Tune & Chromaticity,
- ...

These virtual interfaces will themselves implement some standard custom types, properties and property fields relevant to every observable. These items will be grouped in the 'BDI standard interface' that will be inherited by every 'virtual' interface.

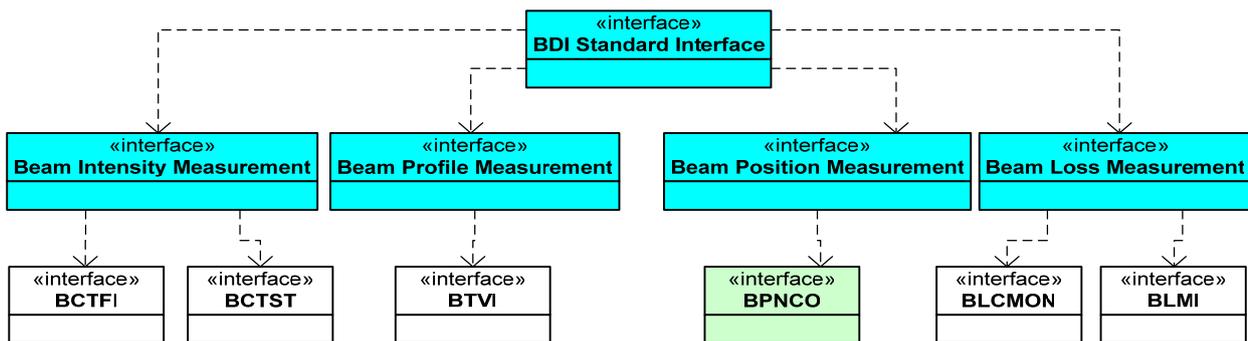


Figure 1: Inheritance Tree. Blue Boxes represent 'Virtual' interfaces. Green Box represents BPNCO interface.

This document describes the Front End Software Interface implemented by LEIR Orbit measurement system.

## 2. CONVENTIONS

The device names implementing this interface follow the conventions described in [4].

The properties defined start with a capital letter. To separate meaningful parts of the name a capital letter is used, not underscores. All properties are composite to insure data integrity and coherence. We expect the foreseen controls middle-tier to extract the so-called parameters (see [4]) from these properties.

The fields within each property start with a small letter. To separate meaningful parts of the name a capital letter is used, not underscores.

Every constant is in capital letter. To separate meaningful parts of the name an underscore is used in constants.

### 3. OVERVIEW

Most Beam Instruments are able to make several measurements during a given machine cycle. Beam Intensity Monitors in a ring could for instance acquire the total intensity every milliseconds between 'beam in' and 'beam out'. In a transfer line, they will measure every injection corresponding to a given cycle. In this document, we will refer to this as 'Number of Measurements' implicitly meaning per cycle.

Some Beam Instruments are also able to 'cut' the beam into slices and acquire bunch or batch intensity for instance. In this case, each of the previous measurements will consist of an intensity value per gate (bunch, batch, slice). In this document, we will refer to this as 'Number of Gates' implicitly meaning per measurement.

Every Beam Instruments will publish its result at the end of their measurements, i.e. at beam out for an instrument in the ring or last injection/extraction for an instrument in a transfer line. The instrument will then publish 3 properties (See chapter 6 for details): *SummaryAcquisition*, *Acquisition*, *ExpertAcquisition*. These properties will give results covering the entire cycle, i.e. all measurements. The BPNCO class will not provide standard *IntermediateAcquisition* and *Reset* properties. Two additional specific properties are implemented to the BPNCO class: *SingleAcquisition*, *ExpertCalibration*.

### 4. INTERFACE INHERITANCE

Table 1 gives the standard interface, if any, inherited by the interface presented in this document. By inheritance, we mean here that every FESA [5] custom types, properties and corresponding property fields of the inherited interface are integrated in the presented one. Inherited constants and fields are 'yellow highlighted' in the signature tables given in this document. They are given here for information but will obviously not be updated automatically when the inherited interface is modified. Reader should always refer to the corresponding document to get the latest description of this inherited interface. This document reference is also given in the following table:

Inherited Standard Interface	Corresponding FESA Virtual Class	Corresponding Documentation
BDI Common Interface	BdiStdIntRef	EDMS: <a href="#">532775</a>
BDI Position Interface	BdiStdPosition	EDMS: <a href="#">630857</a>

Table 1: Inherited Interface

## 5. CUSTOM TYPES

This chapter describes the custom types used in this interface.

We will group these constants in the 3 categories available in the FESA model:

- The **Constants** essentially used here to define array dimensions
- The **Enum Types**, used to select/reflect a single option/state within a list.
- The **Bit Enum Types**, used to select/reflect one or several options/states within a list.

### 5.1 CONSTANTS

Constant Name	Type	Value	Constant Usage
<b>NAME_SIZE</b>	long	64	Used to define string length for names.
<b>MSG_SIZE</b>	long	256	Used to define string length for messages.
<b>MAX_NB_OF_GATES</b>	long	1	Maximum number of Gates.
<b>MAX_NB_OF_MEAS</b>	long	360	Maximum number of Measurements.
<b>MAX_NB_OF_GATES_AND_MEAS</b>	long	360	Maximum number of Gates and Measurements.
<b>MAX_NB_OF_GAINS</b>	long	3	Maximum number of Gains.
<b>MAX_ERR</b>	long	32	Used to define the buffer of active error messages.
<b>CHANNEL_NAME_SIZE</b>	long	20	Used to define string length for channel names.
<b>NB_OF_CHANNELS</b>	long	32	Number of channels used.
<b>MAX_NB_OF_ADCS</b>	long	1	Maximum number of ADCs.
<b>MAX_NB_OF_ADC_CHANNELS</b>	long	32	Maximum number of Channels per ADC.
<b>TIMING_RATE</b>	long	10	Used to define ADC timing rate in ms.

## 5.2 ENUM TYPES

### 5.2.1 UNITS

FESA does not provide currently the means to define units associated to fields. Our standard interface integrates some constants and fields to handle this.

Enum Name	Enum Type	Enum Value Usage
<b>UNITS</b>	long	Used to define the unit of a given field.
Enum Identifier	Enum Value	Enum Identifier Usage
NO_UNIT	0	
AMP	1	
VOLT	2	
METER	3	
SECOND	4	
EVOLT	5	
NB_OF_CHARGES	6	
SIGMA	7	
PERCENT	8	
TURNS	9	
HERTZ	10	
WATT	11	
JOULE	12	
DEGREES	13	
RADIAN	14	
DECIBEL	15	
OHM	16	
FARAD	17	
HENRY	18	
DEGREES_CELSIUS	19	
KELVIN	20	
PASCAL	21	
BAR	22	
TORR	23	
INJECT_EXTRAC_INDEX	100	
SLICE_INDEX	101	
BATCH_ID	102	
BUNCH_ID	103	
BUCKET_INDEX	104	

### 5.2.2 BDI\_PROPERTY

BDI built standard Java viewers on top of its standard Acquisition interface. The following enum type allows these viewers to identify the type of Acquisition property they should display.

Enum Name	Enum Type	Enum Value Usage
<b>BDI_PROPERTY</b>	long	Used by the standard Java viewers to identify the type of standard Acquisition type.
Enum Identifier	Enum Value	Enum Identifier Usage
OTHER	0	
SUMMARY_ACQUISITION	1	
ACQUISITION	2	
EXPERT_ACQUISITION	3	
INTERMEDIATE_ACQUISITION	4	

### 5.2.3 DEVICE\_STATUS

Mutually exclusive values to describe the device status.

Enum Name	Enum Type	Enum Value Usage
<b>DEVICE_STATUS</b>	long	Used to describe the device status.
Enum Identifier	Enum Value	Enum Identifier Usage
UNKNOWN	0	The device status is unknown.
OK	1	The device is in a fully operational state.
WARNING	2	The device is not fully operational; A device in WARNING state can still be used operationally, but the operator should be informed of a problem that might become worse. Details are explained in the errorMsg field.
ERROR	3	The device is in a fault state. Details are explained in the errorMsg field.

### 5.2.4 DEVICE\_MODE

Mutually exclusive values to describe the operational mode of the device.

Enum Name	Enum Type	Enum Value Usage
<b>DEVICE_MODE</b>	long	Used to define the mode of a device.
Enum Identifier	Enum Value	Enum Identifier Usage
UNKNOWN	0	It is not possible to determine the mode.
ON	1	The device is in a fully operational state.
OFF	2	The device is turned off.
STANDBY	3	The device is in a stand-by mode. This mode is a sort of "parking mode" in which the device can stay for hours or even days. It is defined by the following characteristics: it is safe, it does not wear out, it consumes little energy. Furthermore, it takes a short time to go from STANDBY to ON mode.
INITIALIZING	4	The device is initializing. This field is activated when the Reset property has been Set, and remains active until the mode has reached one of the other values (e.g. STANDBY).
OTHER	127	The device is in another, equipment-specific mode, to be looked up in the field detailedMode.

### 5.2.5 DEVICE\_MODE\_SETTING

An enum type used to control the operational mode of the device. Its values are a subset of those in the DEVICE\_MODE type.

Enum Name	Enum Type	Enum Value Usage
<b>DEVICE_MODE_SETTING</b>	long	Used to control the mode of a device.
Enum Identifier	Enum Value	Enum Identifier Usage
ON	1	Used to set the device to mode=ON.
OFF	2	Used to set the device to mode=OFF.
STANDBY	3	Used to set the device to mode=STANDBY.

### 5.2.6 DEVICE\_CONTROL

Mutually exclusive values to describe the control mode of a device. Currently only two control modes (LOCAL, REMOTE) are defined.

Enum Name	Enum Type	Enum Value Usage
<b>DEVICE_CONTROL</b>	long	Used to describe the control mode of a device.
Enum Identifier	Enum Value	Enum Identifier Usage
REMOTE	5	The device can be controlled normally through the control system.
LOCAL	6	The device can be controlled only locally (but it can be accessed in read-only mode via the control system).

### 5.2.7 GAIN\_MODE

Mutually exclusive values to describe the gain mode of a device. Currently only three gain modes (LOW, MEDIUM, HIGH) are defined.

Enum Name	Enum Type	Enum Value Usage
<b>GAIN_MODE</b>	long	Used to describe the gain mode of a device.
Enum Identifier	Enum Value	Enum Identifier Usage
LOW_GAIN	0	Used to set the device to gain=LOW.
MEDIUM_GAIN	1	Used to set the device to gain=MEDIUM.
HIGH_GAIN	2	Used to set the device to gain=HIGH.

### 5.2.8 CALIBRATING\_MODE

Mutually exclusive values to describe the calibrating mode of a device. Currently only six calibrating modes (NO\_CALIBRATION, POSITIVE\_BEAM, CENTERED\_BEAM, NEGATIVE\_BEAM, SEMIAUTOMATIC, AUTOMATIC) are defined.

Enum Name	Enum Type	Enum Value Usage
<b>CALIBRATING_MODE</b>	long	Used to describe the calibrating mode of a device.
Enum Identifier	Enum Value	Enum Identifier Usage
NO_CALIBRATION	0	Used to set the device to operational mode.
POSITIVE_BEAM	1	Used to set the device to positive calibrating mode for chosen gain.
CENTERED_BEAM	2	Used to set the device to centered calibrating mode for chosen gain.
NEGATIVE_BEAM	3	Used to set the device to negative calibrating mode for chosen gain.
SEMIAUTOMATIC	4	Used to set the device to semiautomatic calibrating mode for chosen gain.
AUTOMATIC	5	Used to set the device to automatic calibrating mode for all gains.

### 5.2.9 CLEARING\_VOLTAGE\_MODE

Mutually exclusive values to describe the clearing voltage mode of a device. Currently only two modes (OFF, ON) are defined.

Enum Name	Enum Type	Enum Value Usage
<b>CLEARING_VOLTAGE_MODE</b>	long	Used to describe the clearing voltage mode of a device.
Enum Identifier	Enum Value	Enum Identifier Usage
OFF	0	Used to switch off clearing voltage.
ON	1	Used to switch on clearing voltage.

## 5.3 BIT ENUM TYPES

### 5.3.1 BDI\_OBSERVABLES

This set of documents describes BDI standard interfaces for BDI instrument families measuring the same beam observable. The aim is to be able to build generic viewers and recorders for a given observable (beam position, size, intensity...) able to monitor all related instrument based on different hardware and timing constraints. In order to select the right display, the application has to know the observable which will be part from the Result property. This bit enum type has been added for this purpose.

Bit Enum Name	Bit Enum Type	Enum Value Usage
<b>BDI_OBSERVABLES</b>	long	Used to define the different beam observables measured by the instrument. Selected observable(s) also mean(s) that the corresponding standard interface(s) is implemented by the instrument.
Bit Enum Identifier	Bit Enum Value	Enum Identifier Usage
OTHER	2 <sup>0</sup>	
INTENSITY	2 <sup>1</sup>	
POSITION	2 <sup>2</sup>	Instruments implementing the interface described in this document will have this bit set.
PROFILE	2 <sup>3</sup>	
LOSSES	2 <sup>4</sup>	
IMAGE	2 <sup>5</sup>	
LUMINOSITY	2 <sup>6</sup>	
TUNE	2 <sup>7</sup>	
CHROMATICITY	2 <sup>8</sup>	
MOMENTUM	2 <sup>9</sup>	

### 5.3.2 ACQ\_STATUS

Possible values to describe the acquisition status of a field (in the `_status` suffix). If this suffix is missing, it means that no additional status information is provided for the corresponding field. If all bits are 0, it means that the corresponding field is OK.

Only the lower 16 bits are standardized, the upper 16 bits can be defined by the equipment specialist.

Bit Enum Name	Bit Enum Type	Enum Value Usage
<b>AQN_STATUS</b>	long	Used to qualify the acquisition in every Acquisition property. Additional information about problems or anomalies in the corresponding main value that must be taken into account when interpreting the main value.
Bit Enum Identifier	Bit Enum Value	Enum Identifier Usage
NOT_OK	$2^0$	Some problem occurred that is not represented by the other bits. This property is called NOT_OK so that it is not mixed up with ERROR or WARNING in the Status property.
BAD_QUALITY	$2^1$	The value was acquired with a degraded quality. This is typically used for measurements.
DIFFERENT_FROM_SETTING	$2^2$	Different from the requested control value (for discrete values) or out of tolerance (for continuous values).
OUT_OF_RANGE	$2^3$	The value is out of the normal range (e.g. a temperature is too high or too low).
BUSY	$2^4$	The property value is changing in response to receiving a new control value (e.g. moving to a new position, charging a capacitor, ...). If the value change does not reach the requested new value within the maximum timeout, the BUSY bit should remain=1 and the TIMEOUT bit should be turned on.
TIMEOUT	$2^5$	A timeout occurred, because the property did not reach the requested new control value within the maximum allowable time. A timeout normally indicates a problem to be addressed by the equipment specialist. This is typically used for slow changing control values that are BUSY while they change.
<reserved>	$2^{6-15}$	Reserved for future standardization.
TOO_LOW	$2^{16}$	The signal is too low (reported by hardware).
TOO_HIGH	$2^{17}$	The signal is too high (reported by hardware).
CALIBRATION	$2^{18}$	The device is in calibration mode, no acquisition is done.
NO_CLEARING_VOLTAGE	$2^{19}$	The clearing voltage is off (reported by hardware).
NOT_BUNCHED	$2^{20}$	The beam is not bunched (reported by hardware).
WAIT_ACQ_EVT	$2^{21}$	
WAIT_PREPARE_EVT	$2^{22}$	
NOISY	$2^{23}$	
VMOD_CAL_MINUS	$2^{24}$	The negative calibrating bit is activated (reported by hardware).
VMOD_CAL_PLUS	$2^{25}$	The positive calibrating bit is activated (reported by hardware).
INCOHERENT_ACQ	$2^{26}$	
NO_SIGNAL	$2^{27}$	
FAULTY_DEVICE	$2^{28}$	
SIMULATED_TIMING	$2^{29}$	
SIMULATED_HW	$2^{30}$	

### 5.3.3 TOL\_CHECK\_MODE

This constant defines possible modes to check whether a control value is inside the tolerance values.

Bit Enum Name	Bit Enum Type	Enum Value Usage
<b>TOL_CHECK_MODE</b>	long	Used to give information on how the tolerance fields should be used to calculate the xxx_status information.
Bit Enum Identifier	Bit Enum Value	Enum Identifier Usage
ABS	2 <sup>0</sup>	Use the absolute tolerance _tolAbs.
REL	2 <sup>1</sup>	Use the relative tolerance _tolRel.

### 5.3.4 DEVICE\_DETAILED\_MODE

Bit Enum Name	Bit Enum Type	Enum Value Usage
<b>DEVICE_DETAILED_MODE</b>	long	Used to qualify the device detailed mode in 'Status' property.
Bit Enum Identifier	Bit Enum Value	Enum Identifier Usage
...	2 <sup>0-&gt;30</sup>	<i>Instrument developers will add here Identifiers for instrument specific modes.</i>

### 5.3.5 DEVICE\_DETAILED\_STATUS

Bit Enum Name	Bit Enum Type	Enum Value Usage
<b>DEVICE_DETAILED_STATUS</b>	long	Used to qualify the device detailed status in 'Status' property.
Bit Enum Identifier	Bit Enum Value	Enum Identifier Usage
...	2 <sup>0-&gt;30</sup>	<i>Instrument developers will add here Identifiers for instrument specific statuses.</i>

### 5.3.6 INIT\_DEV\_STATE

Bit Enum Name	Bit Enum Type	Enum Value Usage
<b>INIT_DEV_STATE</b>	long	Used to qualify device initialisation state in 'Status' property.
Bit Enum Identifier	Bit Enum Value	Enum Identifier Usage
NOT_DONE	2 <sup>0</sup>	
FAILED	2 <sup>1</sup>	
IN_PROGRESS	2 <sup>2</sup>	

### 5.3.7 INIT\_ALL\_DEV\_STATE

Bit Enum Name	Bit Enum Type	Enum Value Usage
<b>INIT_ALL_DEV_STATE</b>	long	Used to qualify class initialisation state in 'Status' property.
Bit Enum Identifier	Bit Enum Value	Enum Identifier Usage
NOT_DONE	2 <sup>0</sup>	
FAILED	2 <sup>1</sup>	
IN_PROGRESS	2 <sup>2</sup>	

## 6. PROPERTIES

The following 12 properties will be implemented and 2 properties will not be implemented (marked as 'grey').

Some properties need some context or **multiplexing** criteria. The following are currently supported:

- **None:** Meaning that no multiplexing criteria are required.
- **User:** Meaning that the cycle name (SFTPRO, EASTA...) is requested.
- **User/None:** Meaning that the cycle name (SFTPRO, EASTA...) is in principle requested but that None is supported for subscription, which will then return every acquisition on every User.

Some properties are **Subscribable** (like Acquisition), others not (like Description).

Property may have different scopes:

- **Class:** meaning that they represent/affect every devices of this class on this FEC.
- **Device:** meaning that they only represent/affect the targeted device.

Property may be targeted to specific usages or users:

- **Standard Operation:** Properties used during daily operation.
- **Hardware Expert or Machine Development:** Properties used by hardware expert to commission and tune the instrument and during machine development periods to push the instrument towards its limits.
- **Software Expert:** Properties used to develop, test and diagnose the instrument. These properties are not intended to be used by Operation.

Errors are part of the interface. Each property describes the error/s it can return in this document.

### 6.1 DESCRIPTION

#### 6.1.1 DETAILED DESCRIPTION

The '**Description**' property gives class and device description, the observables covered, unit information and other class related static information.

#### 6.1.2 SIGNATURE

Property Name : Description				
Multiplexing	Subscribable	Scope	Usage	Get/Set
None	No	Class	Standard Operation	Get
Data Field	Data Field Type	Data Field Description		
observables	BDI_OBSERVABLES	Main BDI Observables covered by this Class.		
classDesc	char[MSG_SIZE]	Static Class Description.		
classUrl	char[MSG_SIZE]	Web Page Address Describing This Class.		
deviceDesc	char[MSG_SIZE]	Static device description.		
Error Message		Problem Description		

## 6.2 INITALLDEV

### 6.2.1 DETAILED DESCRIPTION

The '**InitAllDev**' property initialises all devices of this class in this FEC.

### 6.2.2 SIGNATURE

Property Name: InitAllDev				
Multiplexing	Subscribable	Scope	Usage	Get/Set
None	No	Class	Standard Operation	Get
Data Field	Data Field Type	Data Field Description		
initAllDevState	INIT_ALL_DEV_STATE	Last Crate Initialization Status Register.		
initAllDevMsg	char[MSG_SIZE]	Last Crate Initialization Report with Time Stamp.		
Error Message		Problem Description		

## 6.3 INIT

### 6.3.1 DETAILED DESCRIPTION

The '**Init**' property initialises the device.

### 6.3.2 SIGNATURE

Property Name: Init				
Multiplexing	Subscribable	Scope	Usage	Get/Set
None	No	Device	Standard Operation	Get
Data Field	Data Field Type	Data Field Description		
initDevState	INIT_DEV_STATE	Last Device Initialization Status Register.		
initDevMsg	char[MSG_SIZE]	Last Device Initialization Report with Time Stamp.		
Error Message		Problem Description		

## 6.4 STATUS

### 6.4.1 DETAILED DESCRIPTION

The '**Status**' property gives Device and Crate Status.

### 6.4.2 SIGNATURE

Property Name: Status				
Multiplexing	Subscribable	Scope	Usage	Get/Set
User	Yes	Device	Standard Operation	Get
Data Field	Data Field Type	Data Field Description		
observables	BDI_OBSERVABLES	Main BDI Observables covered by this Class.		
isConnected	boolean	Device connection state.		
deviceMsg	char[MSG_SIZE]	Device Status Report with Time Stamp.		
initDevState	INIT_DEV_STATE	Last Device Initialization Status Register.		
initAllDevState	INIT_ALL_DEV_STATE	Last Crate Initialization Status Register.		
initDevMsg	char[MSG_SIZE]	Last Device Initialization Report with Time Stamp.		
initAllDevMsg	char[MSG_SIZE]	Last Crate Initialization Report with Time Stamp.		
status	DEVICE_STATUS	Summary of the device status.		
errorMsg	char[MAX_ERR][MSG_SIZE]	Active error messages (if the status field flags an error or a warning), true if an external condition is active.		
detailedStatus	DEVICE_DETAILED_STATUS	Equipment-specific, detailed information about the status. (Not mutually exclusive). This value is used to complement the information contained in the status field.		
detailedMode	DEVICE_DETAILED_MODE	Equipment-specific, detailed information about the mode. (Not mutually exclusive). This value is used to complement the information in the mode field. Normally, the mode field will have the value "OTHER".		
externalCondition	boolean	External condition.		
mode	DEVICE_MODE	Used to control the device mode.		
control	DEVICE_CONTROL	Device control mode, describes whether the device can be controlled remotely.		
Error Message		Problem Description		

## 6.5 RESET

### 6.5.1 DETAILED DESCRIPTION

The '**Reset**' property SET one or several operation settings depending on the actual sent items. This property will not be implemented.

### 6.5.2 SIGNATURE

Property Name: Reset				
Multiplexing	Subscribable	Scope	Usage	Get/Set
User	No	Device	Standard Operation	Set
Data Field	Data Field Type	Data Field Description		
reset	boolean	Used to reset the device.		
Error Message		Problem Description		

## 6.6 SETTING

### 6.6.1 DETAILED DESCRIPTION

The '**Setting**' property GET/SET one or several operation settings depending on the actual sent items.

During a GET every setting (i.e. item) will be updated. During a SET, only settings sent will be touched.

We do not plan to support subscription on the Settings, which often do not reflect the actual instrument status but just the user request for a given cycle. We expect people to subscribe instead to the 'Status' to get the current values or Acquisition properties were the operational settings active during the acquisition, which can differ from the newly requested ones, will be added.

### 6.6.2 SIGNATURE

Property Name: Setting				
Multiplexing	Subscribable	Scope	Usage	Get/Set
User	No	Device	Standard Operation	Get and Set
Data Field	Data Field Type	Data Field Description		
acqStamp	longlong	Acquisition Time Stamp (UTC in nanoseconds as double).		
mode	DEVICE_MODE	Used to control the device mode.		
gain	GAIN_MODE	Gain used (LOW/MEDIUM/HIGH).		
Error Message		Problem Description		

## 6.7 EXPERTSETTING

### 6.7.1 DETAILED DESCRIPTION

The '**ExpertSetting**' property GET/SET one or several expert settings depending of the given selector. These settings are relevant for HW experts and machine development period.

During a GET every setting (i.e. data field) will be updated. During a SET, only settings sent will be touched.

### 6.7.2 SIGNATURE

Property Name: ExpertSetting				
Multiplexing	Subscribable	Scope	Usage	Get/Set
None	No	Device	HW Expert	Get and Set
Filter Field	Filter Field Type	Filter Field Description		
gateSelector	short	Which gate to select (if applicable).		
Data Field	Data Field Type	Data Field Description		
isConnected	boolean	Device connection state.		
control	DEVICE_CONTROL	Device control mode, describes whether the device can be controlled remotely.		
calibratingMode	CALIBRATING_MODE	Calibrating mode: no calibration (standard mode), maximum positive beam displacement, centered beam, maximum negative beam displacement, semi-automatic calibration (for chosen gain), automatic calibration (for all gains).		
calibratingDelay	long	Time in milliseconds to start measuring calibrating factors after switch on HW calibration.		
calibratingFactorZero	float [MAX_NB_OF_GAINS] [NB_OF_CHANNELS]	Calibrating factors for centered beam (cal0, pos = unitFactor * (k * (V - cal0) + offset)), 2D array (MAX_NB_OF_GAINS x NB_OF_CHANNELS) of raw MPV908 data).		
calibratingFactorMinus	float [MAX_NB_OF_GAINS] [NB_OF_CHANNELS]	Calibrating factors for maximum negative beam displacement (cal-, k = 2 * a1 / (cal+ - cal-) [mm/V], 2D array (MAX_NB_OF_GAINS x NB_OF_CHANNELS) of raw MPV908 data).		
calibratingFactorPlus	float [MAX_NB_OF_GAINS] [NB_OF_CHANNELS]	Calibrating factors for maximum positive beam displacement (cal+, k = 2 * a1 / (cal+ - cal-) [mm/V], 2D array (MAX_NB_OF_GAINS x NB_OF_CHANNELS) of raw MPV908 data).		
calibratingSelectionSize	long	Selection size for averaging calibrating factors.		
calibratingTime	char[MSG_SIZE]	Calibrating Time (local as String), last time when scaling factors and / or calibrating factors for centered beam / maximum negative beam displacement / maximum positive beam displacement were updated.		
channelNames	char[NB_OF_CHANNELS] [CHANNEL_NAME_SIZE]	A 2D array of the channel names.		
clearingVoltage	CLEARING_VOLTAGE_MODE	Clearing voltage mode used (ON/OFF).		
coefficientsPUTime	char[MSG_SIZE]	PU Coefficients Time (local as String), last time when PU sensitivities and / or position offsets were updated.		
gain	GAIN_MODE	Gain used (LOW/MEDIUM/HIGH).		
offset	float[NB_OF_CHANNELS]	Position offsets (electrical offsets + mechanical offsets, offset, pos = unitFactor * (k * (V - cal0) + offset)), 1D array (NB_OF_CHANNELS)).		
scalingFactor	float [MAX_NB_OF_GAINS] [NB_OF_CHANNELS]	Scaling factors (k, pos = k * (V - cal0) + offset), k = 2 * a1 / (cal+ - cal-) [mm/V], 2D array (MAX_NB_OF_GAINS x NB_OF_CHANNELS)).		
sensitivityPU	float [MAX_NB_OF_GAINS] [NB_OF_CHANNELS]	PU sensitivities (a1, k = 2 * a1 / (cal+ - cal-) [mm/V], 2D array (MAX_NB_OF_GAINS x NB_OF_CHANNELS)).		
Error Message		Problem Description		

## 6.8 GURUSETTING

### 6.8.1 DETAILED DESCRIPTION

The '**GuruSetting**' property GET/SET one or several software expert settings depending of the given selector. These settings are not intended to be used by Operation.

During a GET every setting (i.e. data field) will be updated. During a SET, only settings sent will be touched.

### 6.8.2 SIGNATURE

Property Name: GuruSetting				
Multiplexing	Subscribable	Scope	Usage	Get/Set
None	No	Device	SW Expert	Get and Set
Data Field	Data Field Type	Data Field Description		
simulatedTiming	boolean	Timing connection state.		
simulatedHW	boolean	Devices connection state.		
pickupAngle	long[NB_OF_CHANNELS]	Angles of pickups (in degrees, 1D array (NB_OF_CHANNELS)).		
channelsMap	short[NB_OF_CHANNELS]	Channels map.		
exactTagging	boolean	Used to specify retrieving data mechanism from continuous acquisition (exact or ignore last one).		
Error Message		Problem Description		

## 6.9 ACQUISITION

### 6.9.1 DETAILED DESCRIPTION

The '**Acquisition**' property gets Nominal Acquisition Results. This property is targeted to the operational GUI's. This property will be published once every cycle, usually at beam out in rings and after the last injection/extraction in transfer lines. This property will give results covering the entire cycle, i.e. all measurements. Operational GUI's requiring intermediate results should subscribe to the following **IntermediateAcquisition** property instead in cases where it is supported by / relevant to the instrument.

### 6.9.2 SIGNATURE

Property Name: Acquisition				
Multiplexing	Subscribable	Scope	Usage	Get/Set
User	Yes	Device	Standard Operation	Get
Filter Field	Filter Field Type	Filter Field Description		
gateSelector	short	Which gate to select (if applicable).		
startTime	long	Time in milliseconds to start measuring from in the cycle.		
Data Field	Data Field Type	Data Field Description		
deviceName	char[NAME_SIZE]	Name of the device.		
cycleName	char[NAME_SIZE]	Name of the cycle.		
observables	BDI_OBSERVABLES	Used to select display panel in generic applications.		
propType	BDI_PROPERTY	Used to select the standard viewer.		
acqDesc	char[MSG_SIZE]	Dynamic acquisition description.		
superCycleNb	long	SPS Super Cycle Number.		
beamID	long	Unique identifier of a beam type.		
cycleTime	char[NAME_SIZE]	Cycle Time (local as String).		
acqTime	char[NAME_SIZE]	Acquisition Time (local as String).		
cycleStamp	longlong	Cycle Time Stamp (UTC in nanoseconds).		
acqStamp	longlong	Acquisition Time Stamp (UTC in nanoseconds).		
acqMsg	char[MSG_SIZE]	Acquisition Status Report with Time Stamp.		
acqState	AQN_STATUS	Acquisition Status Register.		
nbOfChannels	long	Number of channels.		
intensity	float[NB_OF_CHANNELS] [MAX_NB_OF_MEAS]	N intensity measurements for all channels.		
channelNames	char[NB_OF_CHANNELS] [CHANNEL_NAME_SIZE]	A 2D array of the channel names.		
channelStatus	short[NB_OF_CHANNELS]	State of the channels (connected, error, etc).		
gain	GAIN_MODE	Gain used (LOW/MEDIUM/HIGH).		
measurementTimeShift	float[MAX_NB_OF_MEAS]	Time shifts between adjoining measurements.		
measurementTimeShift_unit	UNITS	Measurement time shift unit.		
measurementTimeShift_unitExponent	long	Measurement time shift unit exponent.		
measurementTimeShift_unitFactor	double	Measurement time shift unit factor.		
nbOfMeasurements	long	Number of measurements.		
pickupAngle	long[NB_OF_CHANNELS]	Angles of pickups (in degrees, 1D array).		
position	float[NB_OF_CHANNELS] [MAX_NB_OF_MEAS]	N position measurements for all channels.		
position_unit	UNITS	Position unit.		
position_unitExponent	long	Position unit factor.		
position_unitFactor	double	Position unit exponent.		
Error Message		Problem Description		

## 6.10 INTERMEDIATEACQUISITION

### 6.10.1 DETAILED DESCRIPTION

The '**IntermediateAcquisition**' property gets last intermediate results. This property is targeted to the operational GUI's that do not want to wait till the end of the cycle or till the last injection to get intermediate results. This property will be published several times during the cycle depending of the instrument type. This property only covers one measurement (the last one) of the current cycle. This property will not be implemented.

### 6.10.2 SIGNATURE

Property Name: IntermediateAcquisition				
Multiplexing	Subscribable	Scope	Usage	Get/Set
User	No	Device	Standard Operation	Get
Data Field	Data Field Type	Data Field Description		
deviceName	char[NAME_SIZE]	Name of the device.		
cycleName	char[NAME_SIZE]	Name of the cycle.		
observables	BDI_OBSERVABLES	Used to select display panel in generic applications.		
propType	BDI_PROPERTY	Used to select the standard viewer.		
acqDesc	char[MSG_SIZE]	Dynamic acquisition description.		
superCycleNb	long	SPS Super Cycle Number.		
beamID	long	Unique identifier of a beam type.		
cycleTime	char[NAME_SIZE]	Cycle Time (local as String).		
acqTime	char[NAME_SIZE]	Acquisition Time (local as String).		
cycleStamp	longlong	Cycle Time Stamp (UTC in nanoseconds as double).		
acqStamp	longlong	Acquisition Time Stamp (UTC in nanoseconds as double).		
acqMsg	char[MSG_SIZE]	Acquisition Status Report with Time Stamp.		
acqState	AQN_STATUS	Acquisition Status Register.		
gain	GAIN_MODE	Gain used (LOW/MEDIUM/HIGH).		
Error Message		Problem Description		

## 6.11 SUMMARYACQUISITION

### 6.11.1 DETAILED DESCRIPTION

The '**SummaryAcquisition**' property gets Summary Acquisition Results. This property is targeted to the operational Black Boxes or summary displays. It will be published once every cycle, usually at beam out in rings and after the last injection/extraction in transfer lines. This property will give results covering the entire cycle, i.e. all measurements.

### 6.11.2 SIGNATURE

Property Name: SummaryAcquisition				
Multiplexing	Subscribable	Scope	Usage	Get/Set
User	Yes	Device	Standard Operation	Get
Filter Field	Filter Field Type	Filter Field Description		
startTime	long	Time in milliseconds to start measuring from in the cycle.		
Data Field	Data Field Type	Data Field Description		
deviceName	char[NAME_SIZE]	Name of the device.		
cycleName	char[NAME_SIZE]	Name of the cycle.		
observables	BDI_OBSERVABLES	Used to select display panel in generic applications.		
propType	BDI_PROPERTY	Used to select the standard viewer.		
acqDesc	char[MSG_SIZE]	Dynamic acquisition description.		
superCycleNb	long	SPS Super Cycle Number.		
beamID	long	Unique identifier of a beam type.		
cycleTime	char[NAME_SIZE]	Cycle Time (local as String).		
acqTime	char[NAME_SIZE]	Acquisition Time (local as String).		
cycleStamp	longlong	Cycle Time Stamp (UTC in nanoseconds as double).		
acqStamp	longlong	Acquisition Time Stamp (UTC in nanoseconds as double).		
acqMsg	char[MSG_SIZE]	Acquisition Status Report with Time Stamp.		
acqState	AQN_STATUS	Acquisition Status Register.		
nbOfChannels	long	Number of channels.		
averagedIntensity	float[NB_OF_CHANNELS]	An average of the intensities from the given time, over the given time period.		
channelNames	char[NB_OF_CHANNELS][CHANNEL_NAME_SIZE]	A 2D array of the channel names.		
channelStatus	short[NB_OF_CHANNELS]	State of the channels (connected, error, etc). ** Note that this will be an enumerated array eventually.		
gain	GAIN_MODE	Gain used (LOW/MEDIUM/HIGH).		
measurementQuality	short[NB_OF_CHANNELS]	What is the quality of the measurement? Is it saturated for example? ** Note this should eventually be an array of enumerated shorts.		
averagedPosition	float[NB_OF_CHANNELS]	An average of the positions from the given time, over the given time period.		
position_unit	UNITS	Position unit.		
position_unitExponent	long	Position unit factor.		
position_unitFactor	double	Position unit exponent.		
Error Message		Problem Description		

## 6.12 EXPERTACQUISITION

### 6.12.1 DETAILED DESCRIPTION

The '**ExpertAcquisition**' property gets detailed Acquisition Results. This property is targeted to the expert GUI's. It will be published once every cycle, usually at beam out in rings and after the last injection/extraction in transfer lines. This property will give results covering the entire cycle, i.e. all measurements.

### 6.12.2 SIGNATURE

Property Name: ExpertAcquisition				
Multiplexing	Subscribable	Scope	Usage	Get/Set
User	Yes	Device	Standard Operation	Get
Filter Field	Filter Field Type	Filter Field Description		
startTime	long	Time in milliseconds to start measuring from in the cycle.		
Data Field	Data Field Type	Data Field Description		
deviceName	char[NAME_SIZE]	Name of the device.		
cycleName	char[NAME_SIZE]	Name of the cycle.		
observables	BDI_OBSERVABLES	Used to select display panel in generic applications.		
propType	BDI_PROPERTY	Used to select the standard viewer.		
acqDesc	char[MSG_SIZE]	Dynamic acquisition description.		
superCycleNb	long	SPS Super Cycle Number.		
beamID	long	Unique identifier of a beam type.		
cycleTime	char[NAME_SIZE]	Cycle Time (local as String).		
acqTime	char[NAME_SIZE]	Acquisition Time (local as String).		
cycleStamp	longlong	Cycle Time Stamp (UTC in nanoseconds as double).		
acqStamp	longlong	Acquisition Time Stamp (UTC in nanoseconds as double).		
acqMsg	char[MSG_SIZE]	Acquisition Status Report with Time Stamp.		
acqState	AQN_STATUS	Acquisition Status Register.		
nbOfChannels	long	Number of channels.		
nbOfGates	long	Number of gates in this measurement.		
gateByGateIntensity	float[NB_OF_CHANNELS] [MAX_NB_OF_GATES_AND_MEAS]	X Intensity measurements over Y gates. Note the data is encoded for each channel as Meas1:Gate1, ... Meas1:GateY, MeasX:Gate1, ... MeasX:GateY.		
channelNames	char[NB_OF_CHANNELS] [CHANNEL_NAME_SIZE]	A 2D array of the channel names.		
channelStatus	short[NB_OF_CHANNELS]	State of the channels (connected, error, etc). ** Note that this will be an enumerated array eventually.		
gain	GAIN_MODE	Gain used (LOW/MEDIUM/HIGH).		
nbOfMeasurements	long	Number of measurements.		
pickupAngle	long[NB_OF_CHANNELS]	Angles of pickups (in degrees, 1D array (NB_OF_CHANNELS)).		
gateByGatePosition	float[NB_OF_CHANNELS] [MAX_NB_OF_GATES_AND_MEAS]	X Position measurements over Y gates. Note the data is encoded for each channel as Meas1:Gate1, ... Meas1:GateY, MeasX:Gate1, ... MeasX:GateY.		
position_unit	UNITS	Position unit.		
position_unitExponent	long	Position unit factor.		
position_unitFactor	double	Position unit exponent.		
Error Message		Problem Description		

## 6.13 SINGLEACQUISITION

### 6.13.1 DETAILED DESCRIPTION

The '**SingleAcquisition**' property gets Single Acquisition Results. This property is targeted to the operational GUI's. This property will be published once every cycle, usually at beam out in rings and after the last injection/extraction in transfer lines. This property will give results at predefined time.

### 6.13.2 SIGNATURE

Property Name: SingleAcquisition				
Multiplexing	Subscribable	Scope	Usage	Get/Set
User	No	Device	Standard Operation	Get
Filter Field	Filter Field Type	Filter Field Description		
gateSelector	short	Which gate to select (if applicable).		
startTime	long	Time in milliseconds to start measuring from in the cycle.		
Data Field	Data Field Type	Data Field Description		
deviceName	char[NAME_SIZE]	Name of the device.		
cycleName	char[NAME_SIZE]	Name of the cycle.		
observables	BDI_OBSERVABLES	Used to select display panel in generic applications.		
propType	BDI_PROPERTY	Used to select the standard viewer.		
acqDesc	char[MSG_SIZE]	Dynamic acquisition description.		
superCycleNb	long	SPS Super Cycle Number.		
beamID	long	Unique identifier of a beam type.		
cycleTime	char[NAME_SIZE]	Cycle Time (local as String).		
acqTime	char[NAME_SIZE]	Acquisition Time (local as String).		
cycleStamp	longlong	Cycle Time Stamp (UTC in nanoseconds).		
acqStamp	longlong	Acquisition Time Stamp (UTC in nanoseconds).		
acqMsg	char[MSG_SIZE]	Acquisition Status Report with Time Stamp.		
acqState	AQN_STATUS	Acquisition Status Register.		
nbOfChannels	long	Number of channels.		
intensity	float[NB_OF_CHANNELS] [MAX_NB_OF_MEAS]	N intensity measurements for all channels.		
channelNames	char[NB_OF_CHANNELS] [CHANNEL_NAME_SIZE]	A 2D array of the channel names.		
channelStatus	short[NB_OF_CHANNELS]	State of the channels (connected, error, etc).		
gain	GAIN_MODE	Gain used (LOW/MEDIUM/HIGH).		
nbOfMeasurements	long	Number of measurements.		
pickupAngle	long[NB_OF_CHANNELS]	Angles of pickups (in degrees, 1D array (NB_OF_CHANNELS)).		
position	float[NB_OF_CHANNELS] [MAX_NB_OF_MEAS]	N position measurements for all channels.		
position_unit	UNITS	Position unit.		
position_unitExponent	long	Position unit factor.		
position_unitFactor	double	Position unit exponent.		
Error Message		Problem Description		

## 6.14 EXPERTCALIBRATION

### 6.14.1 DETAILED DESCRIPTION

The '**ExpertCalibration**' property GET several expert settings / SET acquired calibrating and scaling factors to default ones. These settings are relevant for HW experts and machine development period.

During a GET every setting (i.e. data field) will be updated. During a SET, only settings sent will be touched.

### 6.14.2 SIGNATURE

Property Name: ExpertCalibration				
Multiplexing	Subscribable	Scope	Usage	Get/Set
None	No	Device	HW Expert	Get and Set
Filter Field	Filter Field Type	Filter Field Description		
gateSelector	short	Which gate to select (if applicable).		
Data Field	Data Field Type	Data Field Description		
calibratingMode_acquired	CALIBRATING_MODE	Acquired step of calibrating mode: no calibration (standard mode), maximum positive beam displacement, centered beam, maximum negative beam displacement, semi-automatic calibration (for chosen gain), automatic calibration (for all gains).		
calibratingFactorMinus_acquired	float [MAX_NB_OF_GAINS] [NB_OF_CHANNELS]	Acquired calibrating factors for maximum negative beam displacement (cal-, $k = 2 * a1 / (cal+ - cal-)$ [mm/V], 2D array of raw MPV908 data (averaging-out of chosen number values)).		
calibratingFactorMinus_current	float [MAX_NB_OF_GAINS] [NB_OF_CHANNELS]	Calibrating factors for maximum negative beam displacement (cal-, $k = 2 * a1 / (cal+ - cal-)$ [mm/V], 2D array of raw MPV908 data).		
calibratingFactorPlus_acquired	float [MAX_NB_OF_GAINS] [NB_OF_CHANNELS]	Acquired calibrating factors for maximum positive beam displacement (cal+, $k = 2 * a1 / (cal+ - cal-)$ [mm/V], 2D array of raw MPV908 data (averaging-out of chosen number values)).		
calibratingFactorPlus_current	float [MAX_NB_OF_GAINS] [NB_OF_CHANNELS]	Calibrating factors for maximum positive beam displacement (cal+, $k = 2 * a1 / (cal+ - cal-)$ [mm/V], 2D array of raw MPV908 data).		
calibratingFactorZero_acquired	float [MAX_NB_OF_GAINS] [NB_OF_CHANNELS]	Acquired calibrating factors for centered beam (cal0, pos = unitFactor * ( $k * (V - cal0) + offset$ )), 2D array of raw MPV908 data (averaging-out of chosen number values)).		
calibratingFactorZero_current	float [MAX_NB_OF_GAINS] [NB_OF_CHANNELS]	Calibrating factors for centered beam (cal0, pos = unitFactor * ( $k * (V - cal0) + offset$ )), 2D array of raw MPV908 data).		
calibratingTime_acquired	char[MSG_SIZE]	Acquired Calibrating Time (local as String), last time when scaling factors and / or calibrating factors for centered beam / maximum negative beam displacement / maximum positive beam displacement were acquired.		
calibratingTime_current	char[MSG_SIZE]	Calibrating Time (local as String), last time when scaling factors and / or calibrating factors for centered beam / maximum negative beam displacement / maximum positive beam displacement were updated.		
channelNames	char[NB_OF_CHANNELS] [CHANNEL_NAME_SIZE]	A 2D array of the channel names.		
control	DEVICE_CONTROL	Device control mode, describes whether the device can be controlled remotely.		
scalingFactor_acquired	float [MAX_NB_OF_GAINS] [NB_OF_CHANNELS]	Acquired scaling factors (k, pos = $k * (V - cal0) + offset$ ), $k = 2 * a1 / (cal+ - cal-)$ [mm/V], 2D array).		
scalingFactor_current	float [MAX_NB_OF_GAINS] [NB_OF_CHANNELS]	Scaling factors (k, pos = $k * (V - cal0) + offset$ ), $k = 2 * a1 / (cal+ - cal-)$ [mm/V], 2D array).		
Error Message		Problem Description		

## **7. REFERENCES**

- [1] **Beam Instrumentation Software Common Tools and Interfaces** Web Site  
[<http://project-biscoti.web.cern.ch/project-biscoti>]
- [2] **Front End Software Architecture** Web Site [<http://project-fesa.web.cern.ch/project-fesa>]
- [3] **BDI front end software Standard interface For Beam position Measurements**  
[EDMS: 630857]
- [4] **Guidelines and conventions for defining interfaces of equipment developed using FESA** [EDMS: 581892]