

# MOST

Media Oriented Systems Transport

**Multimedia and Control  
Networking Technology**

MOST FBlock Enhanced Testability

**Rev 2.6.1**

**01/2009**



## Legal Notice

### COPYRIGHT

© Copyright 1999 - 2009 MOST Cooperation. All rights reserved.

### LICENSE DISCLAIMER

Nothing on any MOST Cooperation Web Site, or in any MOST Cooperation document, shall be construed as conferring any license under any of the MOST Cooperation or its members or any third party's intellectual property rights, whether by estoppel, implication, or otherwise.

### CONTENT AND LIABILITY DISCLAIMER

MOST Cooperation or its members shall not be responsible for any errors or omissions contained at any MOST Cooperation Web Site, or in any MOST Cooperation document, and reserves the right to make changes without notice. Accordingly, all MOST Cooperation and third party information is provided "AS IS". In addition, MOST Cooperation or its members are not responsible for the content of any other Web Site linked to any MOST Cooperation Web Site. Links are provided as Internet navigation tools only.

MOST COOPERATION AND ITS MEMBERS DISCLAIM ALL WARRANTIES WITH REGARD TO THE INFORMATION (INCLUDING ANY SOFTWARE) PROVIDED, INCLUDING THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT. Some jurisdictions do not allow the exclusion of implied warranties, so the above exclusion may not apply to you.

In no event shall MOST Cooperation or its members be liable for any damages whatsoever, and in particular MOST Cooperation or its members shall not be liable for special, indirect, consequential, or incidental damages, or damages for lost profits, loss of revenue, or loss of use, arising out of or related to any MOST Cooperation Web Site, any MOST Cooperation document, or the information contained in it, whether such damages arise in contract, negligence, tort, under statute, in equity, at law or otherwise.

### FEEDBACK INFORMATION

Any information provided to MOST Cooperation in connection with any MOST Cooperation Web Site, or any MOST Cooperation document, shall be provided by the submitter and received by MOST Cooperation on a non-confidential basis. MOST Cooperation shall be free to use such information on an unrestricted basis.

### TRADEMARKS

MOST Cooperation and its members prohibit the unauthorized use of any of their trademarks. MOST Cooperation specifically prohibits the use of the MOST Cooperation LOGO unless the use is approved by the Steering Committee of MOST Cooperation.

### SUPPORT AND FURTHER INFORMATION

For more information on the MOST technology, please contact:

**MOST Cooperation**

Administration  
Bannwaldallee 48  
D-76185 Karlsruhe  
Germany

Tel: (+49) (0) 721 966 50 00

Fax: (+49) (0) 721 966 50 01

E-mail: [contact@mostcooperation.com](mailto:contact@mostcooperation.com)

Web: [www.mostcooperation.com](http://www.mostcooperation.com)



© Copyright 1999 - 2009 MOST Cooperation.  
All rights reserved.

MOST is a registered trademark

## Contents

<b>1</b>	<b>INTRODUCTION.....</b>	<b>8</b>
<b>2</b>	<b>FUNCTION CATALOG.....</b>	<b>8</b>
2.1	EnhancedTestability (FBlockID=0x0F) .....	8
2.1.1	Attenuation (0x200) .....	10
2.1.2	AutoWakeup (0x201) .....	11
2.1.3	DiagTimeout (0x202) .....	14
2.1.4	DiagResult (0x203) .....	15
2.1.5	Shutdown (0x204).....	16
2.1.6	ShutdownSuspendMode (0x205) .....	17
2.1.7	NetInterfaceState (0x206).....	18
2.1.8	SendMessage (0x207).....	19
2.1.9	EchoMessage (0x208).....	20
2.1.10	MessageBufSize (0x209).....	21
2.1.11	ResetTests (0x20E) .....	23
2.1.12	CodingErrors (0x20F) .....	24
2.1.13	VoltageLevels (0x210) .....	26
2.1.14	Reset (0x211) .....	27
2.1.15	CentralRegistrySize (0x212).....	28
2.1.16	NotificationMatrixSize (0x213).....	29
2.1.17	ManufacturerTimings (0x214).....	30
2.1.18	ActivateSlaveMode (0x215).....	31
2.1.19	SystemState (0x216) .....	32
2.1.20	Void (0x3C8) .....	33
2.1.21	DSIDSOCcount (0x3FD) .....	34
2.1.22	DSO (0x3FE) .....	35
2.1.23	DSIHold (0x3FF) .....	38
2.1.24	DSI (0x400).....	39
2.1.25	Version (0xF00) .....	41

## Document References

All documents, which this MOST document have references to, are listed here with the actual revision this document is referring to.

Number	Document	Revision
[1]	MOST Specification	2.5

## Bibliography MOST Function Catalog

This is a list of released FBlocks at the release time of this specification. FBlocks, which are released later, are not reflected in this list.

FBlockID	FunctionBlock
-	GeneralFBlock
-	GeneralPlayer
0x01	NetBlock
0x02	NetworkMaster
0x03	ConnectionMaster
0x06	Diagnosis
0x0E	Tool
0x0F	Enhanced Testability
0x22	AudioAmplifier
0x24	AuxIn
0x26	MicrophoneInput
0x30	AudioTapePlayer
0x31	AudioDiskPlayer
0x34	DVDVideoPlayer
0x40	AmFmTuner
0x41	TMCTuner
0x42	TVTuner
0x43	DABTuner
0x44	SDARS
0x50	Telephone
0x51	GeneralPhoneBook
0x60	GraphicDisplay
-	Unique Functions

## Document History

### Changes EnhancedTestability FBlock Rev. 2.6 to EnhancedTestability FBlock Rev. 2.6.1

Change Ref.	FktID	Changes
2V61-001	General	Minor editorial corrections.
2V61-002	0x000	Function FktIDs no longer referenced from GeneralFBlock.
2V61-003	0x203	Added description for code 0x40 to DiagResult.
2V61-004	0x3FD	Added new function DSIDSOCcount.

### Changes EnhancedTestability FBlock 2V5 to EnhancedTestability FBlock 2V6

Change Ref.	FktID	Changes
2V6-001	General	General: Corrections of clerical errors and unification of spelling of MOST terms.
2V6-002	General	Added reference to GeneralFBlock 2.5.1, which covers FktIDs (0x000) and Version (0x010)
2V6-003	0x000	Removed function FktIDs
2V6-004	0x200	Function becomes optional. Not applicable for ePhy.
2V6-005	0x201	AutoWakeup: – Function not used for ePhy – Replaced AbilityToWake by CapabilityToWake resp. PermissionToWake – Improved description of parameter "Duration" and "Delaytime": wakeup condition – Figures added to explain "wakeup condition" and "timing"
2V6-006	0x202	DiagTimeout: Function not used for ePhy
2V6-007	0x203	DiagResult: Function not used for ePhy
2V6-008	0x204	Shutdown: Improved description in case neither PM nor temp. management supported.
2V6-009	0x206	Function becomes optional.
2V6-010	0x207	Function becomes optional.
2V6-011	0x209	MessageBufSize : – Changed unit for MessageLengthRx and MessageLengthTx from none to Byte – Description improved: consideration of "parallel message buffers"
2V6-012	0x20A	SendViaMHP: Deleted function.
2V6-013	0x20B	EchoViaMHP: Deleted function.
2V6-014	0x20C	MHPData: Deleted function.
2V6-015	0x20D	MamacPing: Deleted function.
2V6-016	0x211	Reset: – Function not used for ePhy – Description improved: Function has to reset NetInterface
2V6-017	0x214	ManufacturerTimings: Modified Enum description.
2V6-018	0x216	SystemState: New function.
2V6-019	0x3FE	DSO: New function.
2V6-020	0x3FF	DSIHold: New function.
2V6-021	0x400	DSI: New function.
2V6-022	0xF00	Marked Version as deprecated.

Change Ref.	FktID	Changes
2V6-023	-	Removed empty "Dynamic Specification" chapter.

**Changes EnhancedTestability FBlock 2V4 to EnhancedTestability FBlock 2V5**

Change Ref.	Section	Changes
2V5-001	General	Using new MOST_document_template for better FCat consistency.
2V5-002	General	Change of erroneous Section types.
2V5-003	General	Revised descriptions.
2V5-004		Added Introduction regarding ErrorCode and ErrorInfo.
2V5-005		Added parameters Hysteresis and Timeout to function CodingErrors (0x20F). Changed description.
2V5-006		Added function ActivateSlaveMode (0x215).
2V5-007		Added function Void (0x3C8).

**Changes EnhancedTestability FBlock 2V3 to EnhancedTestability FBlock 2V4**

Change Ref.	Section	Changes
2V4-001	0x200	Inverted parameter Enabled.
2V4-002	0x201	Added parameter Duration for Set.
2V4-003	0x201	Added parameter Duration for SetGet.
2V4-004	0x201	Added parameter Duration for Status.
2V4-005	0x209	Changed description for parameter MessageLengthRx.
2V4-006	0x209	Changed description for parameter MessageLengthTx.
2V4-007	0x20D	Changed unit for parameter Timeout to ms.
2V4-008	0x20F	Added function CodingErrors.
2V4-009	0x210	Added function VoltageLevels.
2V4-010	0x211	Added function Reset.
2V4-011	0x212	Added function CentralRegistrySize.
2V4-012	0x213	Added function NotificationMatrixSize.
2V4-013	0x214	Added function ManufacturerTimings.
2V4-014	0xF00	Added function Version.

# 1 Introduction

A MOST Function Catalog is a collection of MOST Function Blocks (FBlocks).

This document contains the specification of an FBlock. MOST FBlocks are standardized and maintained by MOST workgroup Device Architecture (WG\_DA). In order to speed up the process of making new FBlocks available, every FBlock will be updated individually as required.

## 2 Function Catalog

### 2.1 EnhancedTestability (FBlockID=0x0F)

This FBlock is used to trigger sequences which have to be tested in the MOST Compliance Test but which are normally triggered by a project specific, sometimes complicated, mechanism. Due to the nature of this FBlock neither notification nor processing messages will be implemented.

The FBlock should be initialized every time the NetOn state is reached. The FBlock is only available during NetOn. All properties are reset to their default state when entering NetOn, if not mentioned otherwise.

The functions in this FBlock describe a general interface for starting functionality partly implemented in the application, partly in the Network Service.

If an application callback returns wrong or unexpected values the FBlock sends a "device malfunction" error message (code 0x0B).

FBlock EnhancedTestability philosophy: This FBlock generates or suppresses signals. It resides between application and the Netinterface. It should have no impact on the devices overall behavior, especially when the device is not under test condition. FBlock EnhancedTestability does not store any values beyond power down.

In addition to the functions contained in this document, the following functions are also part of the EnhancedTestability FBlock. They exist in the GeneralFBlock template Rev. 2.5.1 and are included here by reference:

FktID	Function name
0x010	Version



Function Overview		
FktID	Name	Section Type
0x200	<a href="#">Attenuation</a>	Mandatory
0x201	<a href="#">AutoWakeup</a>	Mandatory
0x202	<a href="#">DiagTimeout</a>	Mandatory
0x203	<a href="#">DiagResult</a>	Mandatory
0x204	<a href="#">Shutdown</a>	Mandatory
0x205	<a href="#">ShutdownSuspendMode</a>	Mandatory
0x206	<a href="#">NetInterfaceState</a>	Mandatory
0x207	<a href="#">SendMessage</a>	Mandatory
0x208	<a href="#">EchoMessage</a>	Mandatory
0x209	<a href="#">MessageBufSize</a>	Mandatory
0x20E	<a href="#">ResetTests</a>	Mandatory
0x20F	<a href="#">CodingErrors</a>	Mandatory
0x210	<a href="#">VoltageLevels</a>	Mandatory
0x211	<a href="#">Reset</a>	Mandatory
0x212	<a href="#">CentralRegistrySize</a>	Mandatory
0x213	<a href="#">NotificationMatrixSize</a>	Mandatory
0x214	<a href="#">ManufacturerTimings</a>	Mandatory
0x215	<a href="#">ActivateSlaveMode</a>	Mandatory
0x216	<a href="#">SystemState</a>	Mandatory
0x3C8	<a href="#">Void</a>	Mandatory
0x3FD	<a href="#">DSIDSOCCount</a>	Mandatory
0x3FE	<a href="#">DSO</a>	Mandatory
0x3FF	<a href="#">DSIHold</a>	Mandatory
0x400	<a href="#">DSI</a>	Extension
0xF00	<a href="#">Version</a>	Proprietary

## 2.1.1 Attenuation (0x200)

Section Type: Mandatory

Function may be implemented in nodes which handle the FOT unit. This function is optional and not applicable for ePhy.

This property may be used to enable and disable the -3 db attenuation during NetOn state.

### 2.1.1.1 Format of Function

**Function classes:** Unclassified Property

FBlock	Function	OPType	Parameter
EnhancedTestability (0x0F)	Attenuation (0x200)	Set	<a href="#">Enabled</a>
		Get	-
		SetGet	<a href="#">Enabled</a>
		Status	<a href="#">Enabled</a>
		Error	ErrorCode, ErrorInfo

### 2.1.1.2 Parameter

#### Enabled

Indicates if a device lowers optical output power.

Basis datatype	Bit #	Code	Description
Boolean	Bit 0	True	No -3dB attenuation
		False	-3dB attenuation activated

## 2.1.2 AutoWakeup (0x201)

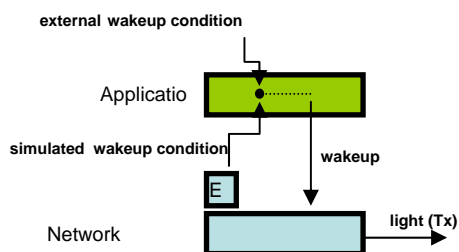
Section Type: Mandatory

Function has to be implemented in every node except ePhy nodes.  
 The option Diagnosis has only to be implemented in nodes which handle RBD.

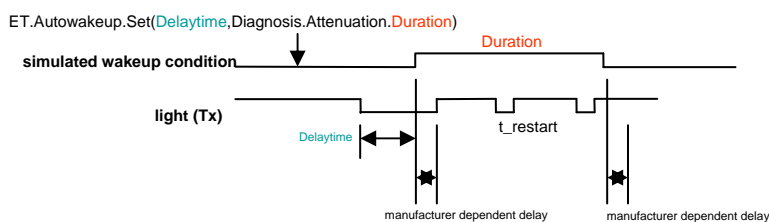
After the MOST network has been shutdown by the tester, the device will wakeup (optionally using ring break diagnosis) as if it was triggered by an external event. The result of a diagnosis will be stored in the property DiagResult until it is either overwritten by the next result or the device is disconnected from power. In normal wakeup scenario the property PermissionToWake of the NetBlock must be considered. If PermissionToWake is set to False, the device will not wakeup the network.

For further clarification, see following figures:

Wakeup condition triggered by FBlock ET



Timing example:



If the parameter Delaytime equals zero, the AutoWakeup is disabled.

The default state is Delaytime = 0, no diagnosis, no attenuation. When an AutoWakeup was performed the property returns to its default state.

When the sender sends Delaytime > 0 and Diagnosis = False and the NetBlock.PermissionToWake property is 0x00 (Off) or the device generally cannot wake up the network (CapabilityToWake), the device returns a "parameter not available" error message (code 0x07). The property shall be unchanged.

In case the NetBlock.PermissionToWake property is set to 0x00 after an AutoWakeup was scheduled and before the device shuts down it will not wakeup. In such a case no error message can be sent (not in NetOn state).

Diagnosis can be started even if PermissionToWake = False, Diagnosis = True. This implies that the result is not ErrorCode 0x07, but execution of diagnosis will be triggered.

If AutoWakeup is started with parameter Diagnosis = True and RBD is not finished automatically by the Network Service, then the application has to stop RBD after a device dependent timeout. In that case the function DiagTimeout (0x202) has to return the appropriate value.

**Note:** Calling *AutoWakeUp* triggers the start-up behavior of the device and consequently masks other wake-up signals of the application that might interfere. This is valid until next *NetOn*.

Diagnosis = false: Until next *NetOn*, diagnosis is forbidden.  
 Diagnosis = true: Until next *NetOn*, diagnosis must be performed.

Default-Behaviour: The application decides whether RBD is performed or not. The Default behavior is reached on every *NetOn* event.  
 This behavior is valid until EnhancedTestability *AutoWakeUp* is called.

### 2.1.2.1 Format of Function

**Function classes:** Unclassified Property

FBlock	Function	OPType	Parameter
EnhancedTestability (0x0F)	AutoWakeup (0x201)	Set	<a href="#">DelayTime</a> , <a href="#">Diagnosis</a> , <a href="#">Attenuation</a> , <a href="#">Duration</a>
		Get	-
		SetGet	<a href="#">DelayTime</a> , <a href="#">Diagnosis</a> , <a href="#">Attenuation</a> , <a href="#">Duration</a>
		Status	<a href="#">DelayTime</a> , <a href="#">Diagnosis</a> , <a href="#">Attenuation</a> , <a href="#">Duration</a>
		Error	ErrorCode, ErrorInfo

### 2.1.2.2 Parameter

#### DelayTime

Time the device waits after shutdown before it will initiate a wakeup event (starts after *NetInterfacePowerOff*).

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Byte	0	full range	1	s

#### Diagnosis

If the device should wake up using ring break diagnosis.

Basis datatype	Bit #	Code	Description
Boolean	Bit 0	True	Wakeup after Delaytime using ring break diagnosis.
		False	Wakeup after Delaytime, if NetBlock.AbilityToWake allows it.

#### Attenuation

Indicates if a device lowers optical output power while doing the ring break diagnosis. If *Diagnosis* is set to *False*, this parameter is ignored.

Basis datatype	Bit #	Code	Description
Boolean	Bit 0	True	-3dB attenuation activated
		False	No -3dB attenuation

---

## Duration

---

This parameter describes how long the simulated wakeup event (= wakeup condition) should be active. The default value is 0 and means infinite time.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Byte	0	full range	1	s

### 2.1.3 DiagTimeout (0x202)

Section Type: Mandatory

Function has to be implemented in nodes which handle RBD except ePhy nodes.

This is a read only property to retrieve the timeout value used by the device for ring break diagnosis.

**Note:** A master device will answer with *t\_diag\_master* while a slave device will answer with *t\_diag\_slave*.

#### 2.1.3.1 Format of Function

Function classes: Number

FBlock	Function	OPType	Parameter
EnhancedTestability (0x0F)	DiagTimeout (0x202)	Get	-
		Status	<a href="#">Timeout</a>
		Error	ErrorCode, ErrorInfo

#### 2.1.3.2 Parameter

##### Timeout

Timeout set in Network Service.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0	full range	1	ms

## 2.1.4 DiagResult (0x203)

Section Type: Mandatory

Function has to be implemented in nodes which handle RBD except ePhy nodes.

The result from a preceding ring break diagnosis is stored in this property and can be accessed in a following network session.

The content of the property is stored over a NetOff phase as long as power is connected. The content changes with every new result of a running RBD. The content may get lost if the device is disconnected from power. Due to this special need the value of DiagResult cannot be initialized.

In other words: You may only trust this value, if you know there was a ring break diagnosis since the device was connected to power!

### 2.1.4.1 Format of Function

Function classes: Number

FBlock	Function	OPType	Parameter
EnhancedTestability (0x0F)	DiagResult (0x203)	Get	-
		Status	<a href="#">Result</a>
		Error	ErrorCode, ErrorInfo

### 2.1.4.2 Parameter

#### Result

Result given by Network Service at the end of ring break diagnosis.

- A value from 0x00 to 0x3F indicates the relative position value in case of Ring-Break.
- A value of 0x40 indicates signal at input but stable lock was never gained.
- A value of 0xFC indicates network activity but no lock.
- A value of 0xFD indicates no Ring-Break and no TimingMaster exists (AllSlaveNetwork).
- A value of 0xFE indicates no Ring-Break and there is more than one TimingMaster in the network.
- A value of 0xFF indicates a fully operational network.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Byte	0	full range	1	none

## 2.1.5 Shutdown (0x204)

Section Type: Mandatory

Function has to be implemented in every node.

Type = 0x00 only has to be supported by nodes containing the PowerMaster (PM).

Type = 0x01,0x02 only have to be supported by nodes which handle the temperature management.

In case no PM implemented and no temperature management supported, a "function not available" error message (error code 0x03) shall be returned. This method is used to trigger different shutdown scenarios:

- \* Normal shutdown (PowerMaster only)
- \* Simulation of Shutdown
- \* Simulation of Dead

If a "Normal shutdown" is sent to a device, which is not PowerMaster, a "parameter not available" error message (code 0x07) shall be returned.

Pending or new wake-up conditions have to be ignored to allow Compliance tests, if the method is called with parameter Type = 0.

All signals from the application within the DUT which prevent PM to shutdown are ignored.

The MOST message NetBlock.Shutdown.Result(Suspend) has to be respected.

### 2.1.5.1 Format of Function

Function classes: Unclassified Method

FBlock	Function	OPType	Parameter
EnhancedTestability (0x0F)	Shutdown (0x204)	Start	Type
		Error	ErrorCode, ErrorInfo

### 2.1.5.2 Parameter

#### Type

If Type is 0x00 and the device is PowerMaster it will start the Shutdown sequence. Codes 0x01 and 0x02 refer to the over-temperature management as specified in MOST Specification 2.5, section 3.2.5.6.

Basis datatype	Range of values	Code	Description
Enum	0x00..0x02	0x00	Start shutdown sequence if PowerMaster
		0x01	Simulate Shutdown
		0x02	Simulate Dead



## 2.1.6 ShutdownSuspendMode (0x205)

Section Type: Mandatory

Function has to be implemented in every node.

This property can be set to On or Off. If set to "On" the device will respond with a report NetBlock.Shutdown.Result(Suspend) to each NetBlock.Shutdown.Start(Query) it receives. If set to "Off" no report NetBlock.Shutdown.Result(Suspend) will be sent. In "Default" mode, the application decides whether the sending of Suspend messages is required.

### 2.1.6.1 Format of Function

Function classes: Enumeration

FBlock	Function	OPType	Parameter
EnhancedTestability (0x0F)	ShutdownSuspend Mode (0x205)	Set	<a href="#">Suspend</a>
		Get	-
		SetGet	<a href="#">Suspend</a>
		Status	<a href="#">Suspend</a>
		Error	ErrorCode, ErrorInfo

### 2.1.6.2 Parameter

#### Suspend

Basis datatype	Range of values	Code	Description
Enum	0x00..0x02	0x00	Off / Do not send Suspend
		0x01	On / Always send Suspend
		0x02	Default behavior

## 2.1.7 NetInterfaceState (0x206)

Section Type: Mandatory

Function may be implemented in every node.

This read only property shows up the changes in the Network State while the device detects an undervoltage situation. The property reflects only two states:

\* NormalOperation: running network and application.

\* StandBy: the network is running, but the application is not active because of low voltage level.

The third network state PowerOff is not available through this property, because the network is not operable in this state.

### 2.1.7.1 Format of Function

Function classes: Enumeration

FBlock	Function	OPType	Parameter
EnhancedTestability (0x0F)	NetInterfaceState (0x206)	Get	-
		Status	<a href="#">NetState</a>
		Error	ErrorCode, ErrorInfo

### 2.1.7.2 Parameter

#### NetState

Status of network layer.

Basis datatype	Range of values	Code	Description
Enum	0x00..0x01	0x00	Device normal operation
		0x01	Device stand by

## 2.1.8 SendMessage (0x207)

Section Type: Mandatory

Function may be implemented in every node.

The device will answer to this function call by sending an application message via Control Channel. The maximum length the device can use for sending should determine the length of the complete message. This should be indicated by the parameter MessageLengthTx of the MessageBufSize property.

### 2.1.8.1 Format of Function

Function classes: Unclassified Method

FBlock	Function	OPType	Parameter
EnhancedTestability (0x0F)	SendMessage (0x207)	StartResult	-
		Result	<a href="#">Data</a>
		Error	ErrorCode, ErrorInfo

### 2.1.8.2 Parameter

#### Data

The payload of SendMessage.Result is a well-known pattern. The first data byte shall be 0x00; the second shall be 0x01 etc. The value is increased with every data byte. If 0xFF is reached the next data byte starts with 0x00 again ...

Basis datatype	Length	Condition	Description
Stream		-	

## 2.1.9 EchoMessage (0x208)

Section Type: Mandatory

Function has to be implemented in every node.

The device will answer to this request by sending back an application message via Control Channel that contains the same data as the request.

### 2.1.9.1 Format of Function

**Function classes:** Unclassified Method

FBlock	Function	OPType	Parameter
EnhancedTestability (0x0F)	EchoMessage (0x208)	StartResult	<a href="#">Data</a>
		Result	<a href="#">Data</a>
		Error	ErrorCode, ErrorInfo

### 2.1.9.2 Parameter

#### Data

Arbitrary data that shall be echoed by the device under test. If the device can receive messages that are longer than the maximum length that can be transmitted, the echoed data will be truncated in such way that the first n bytes of the data are discarded. Discarding first n bytes is used for a test scenario where TxBuf < RxBuf.

Basis datatype	Length	Condition	Description
Stream		-	

## 2.1.10 MessageBufSize (0x209)

Section Type: Mandatory

Function has to be implemented in every node.

Read only function to retrieve the number of message buffers available in parallel for message reception and their size in bytes from a device.

### 2.1.10.1 Format of Function

**Function classes:** Unclassified Property

FBlock	Function	OPType	Parameter
EnhancedTestability (0x0F)	MessageBufSize (0x209)	Get	-
		Status	<a href="#">BufferCountRx</a> , <a href="#">MessageLengthRx</a> , <a href="#">BufferCountTx</a> , <a href="#">MessageLengthTx</a> , <a href="#">Shared</a>
		Error	ErrorCode, ErrorInfo

### 2.1.10.2 Parameter

#### BufferCountRx

Maximum number of message buffers available in parallel for message reception. 0x0000 may be used to indicate that the number is not fixed (e.g., dynamic memory allocation).

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0	full range	1	none

#### MessageLengthRx

Maximum size of receivable application messages in Byte. This value counts only the payload bytes. Header informations like FBlockID InstID, Function, OpType, or additional infos like MessageHandle, Status, Priority, Target address are not included. 0x0000 may be used to indicate that the length is not fixed (e.g., dynamic memory allocation).

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0	full range	1	Byte

#### BufferCountTx

Maximum number of message buffers available for sending. 0x0000 may be used to indicate that the number is not fixed (e.g., dynamic memory allocation).

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0	full range	1	none

**MessageLengthTx**

Maximum size of transmittable application messages in Byte. This value counts only the payload bytes. Header informations like FBlockID InstID, Function, OpType, Length or additional infos like MessageHandle, Status, Priority, Target address are not included. 0x0000 may be used to indicate that the length is not fixed (e.g., dynamic memory allocation).

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0	full range	1	Byte

**Shared**

Indicates that RX and TX buffers share the same memory pool.

Basis datatype	Bit #	Code	Description
Boolean	Bit 0	True	RX and TX buffers share the same buffer pool.
		False	RX and TX own separate buffer pools.

## 2.1.11 ResetTests (0x20E)

Section Type: Mandatory

This method is used to reinitialize all the tests to their default states (e.g., deactivate AutoWakeUp, ShutdownSuspendMode).

### 2.1.11.1 Format of Function

**Function classes:** Trigger

FBlock	Function	OPType	Parameter
EnhancedTestability (0x0F)	ResetTests (0x20E)	Start	-
		Error	ErrorCode, ErrorInfo

## 2.1.12 CodingErrors (0x20F)

Section Type: Mandatory

Function has to be implemented in every node. For detection of errors in the data recovery of the device, it has to count these coding errors. With this function it is possible to activate/deactivate the counter, to read the values and to reset the counter. Default state is "deactivated". When the state is set to off, the counted value should stay valid. The counter is reset to zero after Get. Please note that the parameters Hysteresis and Timeout are optional.

### 2.1.12.1 Format of Function

**Function classes:** Unclassified Property

FBlock	Function	OPType	Parameter
EnhancedTestability (0x0F)	CodingErrors (0x20F)	Set	<a href="#">State, Data</a>
		Get	-
		Status	<a href="#">State, CounterValue</a>
		Error	ErrorCode, ErrorInfo

### 2.1.12.2 Parameter

#### State

State should only contain either "ON" or "OFF". Default value is "OFF".

Basis datatype	Range of values	Code	Description
Enum	0x00..0x01	0x00	OFF
		0x01	ON

#### Data

Please note that the parameters Hysteresis and Timeout are optional.

Basis datatype	Length	Condition	Description
Stream	4	-	Content: <a href="#">Hysteresis, Timeout</a>

#### Hysteresis

In this timeslot up to one coding error is counted. If multiple coding errors are detected within this timeslot, they are just ignored and only one error is counted. Typical value = 25 ms. If this optional timer is not used, all detected coding errors are counted.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0	25..10000	1	ms



## Timeout

---

This value determines the time of observation. On timeout the state changes to "OFF" automatically. If this value is set to 0xFFFF, the observation time is infinite. Typical value = 60000 ms. If this optional parameter is not used, the time is infinite.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Byte	0	25..65535	1	ms

## CounterValue

---

Contains the actual value of the counter.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Long	0	full range	1	none

## 2.1.13 VoltageLevels (0x210)

Section Type: Mandatory

Function has to be implemented in nodes which handle the device's power management.

This read only property is used to specify the voltage level values of the device.

### 2.1.13.1 Format of Function

Function classes: Unclassified Property

FBlock	Function	OPType	Parameter
EnhancedTestability (0x0F)	VoltageLevels (0x210)	Get	-
		Status	<a href="#">USuper</a> , <a href="#">UNormal</a> , <a href="#">UCritical</a> , <a href="#">ULow</a>
		Error	ErrorCode, ErrorInfo

### 2.1.13.2 Parameter

#### USuper

Voltage Level USuper as reported from DUT. The device is in a safe operation state, which must be defined for each device individually. A typical value for USuper is 16V.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Byte	-1	full range	1	V

#### UNormal

Voltage Level UNormal as reported from DUT. Device works normally, all functions are within the specified tolerances.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Byte	-1	full range	1	V

#### UCritical

Voltage Level UCritical as reported from DUT

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Byte	-1	full range	1	V

#### ULow

Voltage Level ULow as reported from DUT

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Byte	-1	full range	1	V

## 2.1.14 Reset (0x211)

Section Type: Mandatory

Function has to be implemented in every node except ePhy nodes.

The Reset function performs a reset of the NetInterface.

### 2.1.14.1 Format of Function

Function classes: Trigger

FBlock	Function	OPType	Parameter
EnhancedTestability (0x0F)	Reset (0x211)	Start	-
		Error	ErrorCode, ErrorInfo

## 2.1.15 CentralRegistrySize (0x212)

Section Type: Mandatory

Function has to be implemented in the node containing the NetworkMaster (NWM).

This Property indicates the size of the Central Registry. This Property is only used for devices with NetworkMaster functionality

### 2.1.15.1 Format of Function

Function classes: Number

FBlock	Function	OPType	Parameter
EnhancedTestability (0x0F)	CentralRegistry Size (0x212)	Get	-
		Status	<a href="#">Size</a>
		Error	ErrorCode, ErrorInfo

### 2.1.15.2 Parameter

#### Size

Parameter Size describes the maximum number of FBlocks the NWM can store in his registry. The value 0 describes a dynamic size.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0	full range	1	none

## 2.1.16 NotificationMatrixSize (0x213)

Section Type: Mandatory

Function has to be implemented in every node.

The NotificationMatrixSize property contains the minimum and maximum values of the size of the Notification Matrix across all FBlocks.

### 2.1.16.1 Format of Function

**Function classes:** Unclassified Property

FBlock	Function	OPType	Parameter
EnhancedTestability (0x0F)	NotificationMatrix Size (0x213)	Get	-
		Status	<a href="#">MinSize</a> , <a href="#">MaxSize</a>
		Error	ErrorCode, ErrorInfo

### 2.1.16.2 Parameter

#### MinSize

Describes the minimum size of all Notification Matrices. If at minimum one FBlock supports notification then MinSize is greater than 0.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Byte	0	full range	1	none

#### MaxSize

Describes the maximum size of all Notification Matrices. The Value 0xFF indicates that the device uses dynamic Notification Matrices

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Byte	0	full range	1	none

## 2.1.17 ManufacturerTimings (0x214)

Section Type: Mandatory

Function has to be implemented in every node.

For some tests there are individual values for timeouts necessary, dependant on the manufacturer. These values can here be queried directly from the device.

### 2.1.17.1 Format of Function

**Function classes:** Unclassified Property

FBlock	Function	OPType	Parameter
EnhancedTestability (0x0F)	Manufacturer Timings (0x214)	Get	<a href="#">TimingID</a>
		Status	<a href="#">TimingID, Value</a>
		Error	ErrorCode, ErrorInfo

### 2.1.17.2 Parameter

#### TimingID

This parameter describes the identification number of the timing considered

Basis datatype	Range of values	Code	Description
Enum	0x00..0x02	0x00	Time between (external) wakeup and MOST signal active. (timer obsolete; do not use. Substituted by t_deadlock_mid of MOST Core Compliance Test Specification)
		0x01	Time between detecting Boundary Descriptor value > 5 and entering Normal Operation (timer obsolete; since MOST Spec 2V4 this is t_boundary).
		0x02	Time between wakeup of PowerMaster and power down again (timer obsolete; do not use).

#### Value

This parameter contains the value of the queried timings

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0	full range	1	ms

## 2.1.18 ActivateSlaveMode (0x215)

Section Type: Mandatory

Function has to be implemented in the node configured as TimingMaster.

This method is exclusively used by the Limited Physical Compliance Test in case the DUT is a TimingMaster. As a reaction to this command the DUT will immediately switch off the MOST signal at its output. If the DUT is woken up again, it will configure as a TimingSlave. This configuration will persist until the DUT is disconnected from power.

If it is necessary that the DUT will wake up the network, then activation of the function AutoWakeup has to precede this method.

### 2.1.18.1 Format of Function

Function classes: Trigger

FBlock	Function	OPType	Parameter
EnhancedTestability (0x0F)	ActivateSlaveMode (0x215)	Start	-
		Error	ErrorCode, ErrorInfo

## 2.1.19 SystemState (0x216)

Section Type: Mandatory

Function has to be implemented in every node.

This function can be used to query the perceived System State of the device and the assumed logical node address of the NWM.

### 2.1.19.1 Format of Function

**Function classes:** Sequence Property

FBlock	Function	OPType	Parameter
EnhancedTestability (0x0F)	SystemState (0x216)	Get	-
		Status	<a href="#">SystemState</a> , <a href="#">NWMAddress</a>
		Error	ErrorCode, ErrorInfo

### 2.1.19.2 Parameter

#### SystemState

SystemState indicates the perceived System State of the device.

Basis datatype	Range of values	Code	Description
Enum	0x00..0x01	0x00	System State Ok
		0x01	System State NotOk

#### NWMAddress

NWMAddress indicates the assumed logical node address of the NWM.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0	full range	1	none



## 2.1.20 Void (0x3C8)

Section Type: Mandatory

Function must not be implemented by any node.

Function 0x3C8 shall be kept empty without implementation as a place holder for test purposes, so that, e.g., a Broadcast Error Test with a message like "EnhancedTestability.InstID.0x3C8.SetGet" might be established.

### 2.1.20.1 Format of Function

**Function classes:** Unclassified Method

<b>FBlock</b>	<b>Function</b>	<b>OPType</b>	<b>Parameter</b>
EnhancedTestability (0x0F)	Void (0x3C8)	Error	ErrorCode, ErrorInfo

## 2.1.21 DSIDSOCcount (0x3FD)

Section Type: Mandatory

Function has to be implemented in every node to indicate the maximum number of simultaneous MOST High Protocol connections (DSI / DSO).

In the case that no DSI is implemented, the parameter DSICount has to return "0x00". In the case that no DSO is implemented, the parameter DSOCcount has to return "0x00".

### 2.1.21.1 Format of Function

**Function classes:** Sequence Property

FBlock	Function	OPType	Parameter
EnhancedTestability (0x0F)	DSIDSOCcount (0x3FD)	Get	-
		Status	<a href="#">DSICount</a> , <a href="#">DSOCcount</a>
		Error	ErrorCode, ErrorInfo

### 2.1.21.2 Parameter

#### DSICount

Maximum number of simultaneous (external) receiving MOST High Protocol connections (DSI).

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Byte	0		1	none

#### DSOCcount

Maximum number of simultaneous (external) sending MOST High Protocol connections (DSO).

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Byte	0		1	none

## 2.1.22 DSO (0x3FE)

Section Type: Mandatory

Function has to be implemented in every node with Data source (DSO) functionality (conditional for MOST High source). The device will answer to this function call by opening a connection with MHP via the Packet Data Channel (target: TargetFBlockID, TargetInstID, TargetFktID and TargetOPType). The device has to send dummy data via MOSTHigh using suitable PacketSize and a well-known pattern. The first data byte shall be 0x00, the seconds shall be 0x01, etc. The value is increased with every data byte. If 0xFF is reached the next data byte starts with 0x00 again.

If the maximum number of connections is reached or there is already a connection, the method returns a 'function busy' (error code 0x40) error message. If a 'device malfunction' (error code 0x0B) error message is sent, there was not enough memory available to allocate the buffers (the application callback returned no memory). The operations StartAck, and ErrorAck of this method are transmitted via the Control Channel. "Prio" and "RevID" have to be chosen by DUT. The tester cannot give a priority to DUT.

### 2.1.22.1 Format of Function

**Function classes:** Unclassified Method

FBlock	Function	OPType	Parameter
EnhancedTestability (0x0F)	DSO (0x3FE)	StartAck	<a href="#">SenderHandle</a> , <a href="#">NumPackets</a> , <a href="#">NextPacketMethod</a> , <a href="#">AckMode</a> , <a href="#">MHP_ConID</a>
		ErrorAck	<a href="#">SenderHandle</a> , <a href="#">ErrorCode</a> , <a href="#">ErrorInfo</a>

### 2.1.22.2 Parameter

#### SenderHandle

Unique identifier of the 'send'-task within the device.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0	full range	1	none

#### NumPackets

Number of MOSTHigh packets that shall be sent. This is used for long-term and performance tests, as well as for testing the protocol handshaking, like the blocks hand-over and MOSTHigh connection establishing procedure.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0	full range	1	none

## NextPacketMethod

Determine what to do after a TX\_SUCCESS event.

Basis datatype	Range of values	Code	Description
Enum	0x00..0x02	0x00	Normal HOLD phase (will close after timeout).
		0x01	Terminate the connection and open a new one if necessary.
		0x02	Try to send next packet without a hold (optional).

## AckMode

Determine acknowledge method

Basis datatype	Range of values	Code	Description
Enum	0x00..0x01	0x00	use single frame acknowledge
		0x01	use block acknowledge

## MHP\_ConID

Determine target FBlockID, InstID, FktID and CmdOPType for MHP connection

Basis datatype	Length	Condition	Description
Stream		-	Content: <a href="#">TargetFBlockID</a> , <a href="#">TargetInstID</a> , <a href="#">TargetFktID</a> , <a href="#">TargetOPType</a>

## TargetFBlockID

Target FBlockID for MHP connection

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Byte	0		1	none

## TargetInstID

Target InstID for MHP connection

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Byte	0		1	none

## TargetFktID

Target FktID for MHP connection

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0		1	none

## TargetOPType

---

Target OPType (only commands allowed) for MHP connection.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Byte	0	0x0..0x8	1	none

## 2.1.23 DSIHold (0x3FF)

Section Type: Mandatory

Function has to be implemented in every node with Data sink (DSI) functionality (conditional for MOST High sink). This function is used by the tester to set certain DSI connections to hold.

If the parameter Hold is set to true, the application must stop servicing the receive buffer of the corresponding DSI connection.  
(Hint for implementation: the callback function has to indicate that no buffer memory is available).

### 2.1.23.1 Format of Function

Function classes: Sequence Method

FBlock	Function	OPType	Parameter
EnhancedTestability (0x0F)	DSIHold (0x3FF)	StartAck	<a href="#">SenderHandle</a> , <a href="#">Hold</a> , <a href="#">DSIIndex</a>
		ErrorAck	<a href="#">SenderHandle</a> , ErrorCode, ErrorInfo

### 2.1.23.2 Parameter

#### SenderHandle

Unique identifier of the 'send'-task within the device.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0		1	none

#### Hold

Switch to set hold of DSI connection on / off.

Basis datatype	Bit #	Code	Description
Boolean	Bit 0	True	Switch hold on
		False	Switch hold off

#### DSIIndex

Determines the DSI connection, the hold switch has to be applied to. The value is an offset to 0x400. E.g., DSIIndex = 0x01 addresses DSI function with FktID = 0x401.

The range of values depends on the number of supported simultaneous MHP connections of the device.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Byte	0		1	none

## 2.1.24 DSI (0x400)

Section Type: Extension

Function has to be implemented in every node with DSI functionality (conditional for MOSTHigh sinks). This function has to be called with MHP via Packet Data Channel. The device will answer to this function call with result, transmitted via Control Channel (no MHP used). With the result, the device has to return the checksum of the data, received during the function call via MHP).

**Note:** For checksum calculation, use CRC32 algorithm (according to IEEE 802.3). This is a standard cyclic redundancy check classified by the following properties:

- CRC polynomial is 0x04C11DB7 (bit sequence of coefficients)
- CRC is 32bit wide
- Initial value is 0xFFFFFFFF
- Result must be XOR-ed to 0xFFFFFFFF
- Reversing bit order on incoming values
- Reversing bit order on resulting data

The checksum must be calculated highest byte of Data first.

The FBlock EnhancedTestability has to provide as many 'copies' of function 'DSI' as the DUT supports simultaneous MOST High connections (as stated in the DUT Manufacturer Information List). This is necessary to check whether the DUT is able to handle the maximum amount of connections. For every connection, one FktID is used. The function DSI and its copies have to be implemented in contiguous FktID range starting from 0x400 to (at max.) 0x4FF.

### 2.1.24.1 Format of Function

**Function classes:** Unclassified Method

FBlock	Function	OPType	Parameter
EnhancedTestability (0x0F)	DSI (0x400)	StartResult Ack	<a href="#">SenderHandle</a> , <a href="#">Data</a>
		ErrorAck	<a href="#">SenderHandle</a> , <a href="#">ErrorCode</a> , <a href="#">ErrorInfo</a>
		ResultAck	<a href="#">SenderHandle</a> , <a href="#">crc32</a> , <a href="#">DataLength</a>

### 2.1.24.2 Parameter

#### SenderHandle

Unique identifier of the 'send'-task within the device.

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Word	0	full range	1	none

## Data

---

Arbitrary data that shall be used by the device to calculate the checksum.

Basis datatype	Length	Description
Stream		

## crc32

---

Checksum of received data

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Long	0		1	none

## DataLength

---

Length (in byte) of received data

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Long	0	full range	1	Byte



## 2.1.25 Version (0xF00)

Section Type: Proprietary

This function is deprecated. Please use function Version (0x010), which is referenced from FBlock GeneralFBlock.

Describes the Version of the FunctionBlock, divided into the Major, Minor and Build Values. E.g., Version 2.3.5.

### 2.1.25.1 Format of Function

**Function classes:** Unclassified Property

FBlock	Function	OPType	Parameter
EnhancedTestability (0x0F)	Version (0xF00)	Get	-
		Status	<a href="#">Major</a> , <a href="#">Minor</a> , <a href="#">Build</a>
		Error	ErrorCode, ErrorInfo

### 2.1.25.2 Parameter

#### Major

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Byte	0	full range	1	none

#### Minor

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Byte	0	full range	1	none

#### Build

Basis datatype	Exp.	Range of values	Step	Unit
Unsigned Byte	0	full range	1	none

Notes: