MOST

Media **O**riented **S**ystems **T**ransport

Multimedia and Control Networking Technology

MOST FBlock Cookbook
Rev 1.0
10/2007

MOSTCO CONFIDENTIAL

See page 3 for the terms of disclosure

Legal Notice

COPYRIGHT

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
Web: www.mostcooperation.com

Contents

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
2	MOST Specification	3.0

Document History

Changes in MOST FBlock Cookbook 1V0-00

Change Ref.	Section	Changes
-	-	Initial Revision

1 Introduction

This document can be used for developing MOST FBlocks, Function Catalogs or single functions, based on either MOST Specification Rev 2.5 or MOST Specification Rev 3.0.
It also can be used as guideline for expanding existing FBlocks, definition of new FBlocks, and for building of Function Catalogs on basis of existing FBlocks.

Specific parts of this document that require at least MOST Specification Rev 3.0 are highlighted.

2 Definitions

This chapter provides an overview of some elementary terms, which will be used in conjunction with FBlocks and Function Catalogs.

The MOST Cooperation provides a “MOST Cooperation FBlock Library”, which is the basis for the creation of Function Catalogs. The FBlock library contains FBlocks and FBlock templates.
FBlock templates do not have an FBlockID.

2.1 Sections of FktIDs

The FktID address range is divided into several sections. Every function of an FBlock is assigned to a section, depending on its FktID:

Section	Address Range	Definition
Coordination	0x000 - 0x1FF	Functions for administrative purposes in a function block ***A MOST device has to implement a coordinative FktID, if the respective function is used by the MOST device.***
Mandatory	0x200 - 0x3FF	Functions that are mandatory for the application of the FBlock, like the basic drive in all FBlocks describing drives
Extensions (Optional)	0x400 - 0x9FF	Extensions are optional functions
Unique	0xA00 - 0xBFF	Functions that are defined unambiguously in the entire system (to be coordinated by the system integrator).
Proprietary (System specific)	0xC00 - 0xEFF	Functions, which can be used by any system integrator (e.g., car manufacturer). They are specific for a system and are coordinated between the suppliers developing devices for this system. Proprietary functions may have additional attributes “coordination”, “mandatory” “extension” or “unique” (defined by System Integrator). These attributes are not defined by additional address range but by description of the relevant function.
Proprietary (Supplier specific)	0xF00 - 0xFFE	Functions, which can be used by suppliers for any proprietary purpose. The device has to be able to perform its normal tasks without using these functions.

Table 1: Sections of FktIDs (Overview)

Note: Do not consider the content of the DTD attribute “Function Section” because it is redundant information and will no longer be relevant.

MOST3

The FktID sections “Mandatory” and “Extension” are replaced by section “Application” that covers the FktID range from 0x200 to 0x9FF.

Independent from the FktID section, a new attribute is introduced for every function to indicate the type of the function. The type of a function could be set to “Mandatory”, “Extension” or “Conditional”). This new property could be named “type”.

A function with property “type” set to **“Mandatory”** must be implemented (equivalent to definition of FktID section “Mandatory” in MOST Specification Rev 2.5.)
A function with property “type” set to **“Extension”** is an optional function (equivalent to definition of FktID section “Extension” in MOST Specification Rev 2.5.)
A function with property “type” set to **“Conditional”** can become “Mandatory” in some cases (e.g., function “SourceInfo” is only mandatory for source devices).

2.2 FBlock

A MOST FunctionBlock is a collection of standardized functions for a dedicated component (e.g., tuner), provided by the MOST Cooperation.

Additionally, system integrator or device manufacturer may specify proprietary FBlocks.

For the MOST FBlock Library, each FunctionBlock consists of a bundle of four files:

XML file	The XML file is the main description file for any FBlock. It can be edited with the “MOST Editor”. It includes all necessary information about the static portion of the FBlock interface (including the change list). Together with the MSC file, the XML file is the master document of the FBlock.
MSC file	Additionally to the XML file that describes the static portion of the interface, the MSCs describe the dynamic behavior of the FBlock (exemplary implementation). Together with the XML file, the MSC file is the master document of the FBlock.
HTML file	The HTML file will be generated from the XML file, e.g., by the MOST Editor. It is a human readable representation of the static FBlock definition.
PDF file	At least one PDF file has to be distributed with the FBlock. It contains the history of the FBlock (change list), the static description and the dynamic specification (MSCs) of the FBlock. This file is based on the XML file and the MSC file.

If new FBlocks are defined, the bundle of all four files has to be provided.

The structure of the XML file is defined by an additional DTD file (Document Type Definition). This DTD file defines the elements, attributes used by the XML file. The MOST Cooperation maintains this DTD file to ensure consistency.
Refer to the “MOST DTD Cookbook” for detailed information about the DTD file and additional rules for creating FBlock specifications in XML.

MOST3

For MOST3, a seamless migration from DTD to XML schema will be supported.

2.3 Function Catalog

A MOST Function Catalog is a collection of MOST FBlocks. All functions or a subset of functions from the FBlock can be used for the Function Catalog. Additionally, proprietary functions as part of an FBlock or whole FBlocks can be added to the Function Catalog.

For description of the Function Catalog, an XML file is used. This XML file contains all relevant FBlocks with all relevant functions. When using the Function Catalog, it has to be ensured that it is associated with the correct DTD version.

Additionally, the Function Catalog contains information of the system integrator that are not covered by the FBlock specification (e.g., which function uses notification, communication via control channel or synchronous channel) or restriction of the FBlock specification for the specific system (e.g., reduced range of values). This results in additional effort for the system integrator.

2.3.1 Example of a Function Catalog

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE FunctionCatalog (View Source for full doctype...)>
<FunctionCatalog>
      <CatalogVersion> ... </CatalogVersion>
      <!-- FBlock: AmFmTuner -->
      <FBlock>
            <Function> ... </Function>
            <Function> ... </Function>
            <Function> ... </Function>
      </FBlock>
      <Definition> ... </Definition>
</FunctionCatalog>
```

When creating a new Function Catalog, the structure from the MOST Cooperation Intranet should be used as basis. New FBlocks can be inserted as follows:

```
<!-- FBlock: X -->
<FBlock>
      <Function> ... </Function>
      <Function> ... </Function>
      <Function> ... </Function>
</FBlock>
<!-- FBlock: Y -->
<FBlock>
      <Function> ... </Function>
      <Function> ... </Function>
      <Function> ... </Function>
</FBlock>
```

Insert the new FBlock sections before the tags <Definition>.
By doing this until all desired FBlocks are in place, a new Function Catalog has been created.

3 General structure of FBlocks

Any FBlock is specified by an XML file that describes the static interface and by MSCs, which describe the dynamic behavior.

3.1 Static Interface

The static interface specifies all functions of the FBlock and their format, according to MOST Specification.
To ensure interoperability, all mandatory functions have to be implemented.

3.1.1 Version of FBlock

Any FBlock has to implement function “Version” (FktID 0x010) that contains the version of the FBlock.

MOST3

In addition, any FBlock has to implement function “FBlock Info” (FktID 0x011) that contains the FBlock name, the name of the instance, the corresponding MOST Specification version, and the version of the FBlock itself.

With every change, the FBlock gets a new version. Versioning of an individual FunctionBlock is done in accordance with MOST Cooperation organizational procedures:

- First number equals not backward compatible changes
- Second number equals enhanced functionality
- Third number equals typographical changes as well as bug fixes.

3.1.2 Unique functions

The FktID range 0xA00 to 0xBFF is reserved for unique functions.

MOST3

Unique functions are not covered by a special FBlock. Former FBlock UniqueFunctions is obsolete.

3.2 Dynamic Behavior

MOST3

With introduction of MOST Specification Rev 3.0, any FBlock Specification contains - additionally to static specification of included functions - a dynamic specification in terms of MSCs.

The purpose of the dynamic specification is:

- The exemplary MSCs support developers of MOST devices. In addition to the static specification, it provides additional information about sequences and dependencies.
- The exemplary MSCs define the minimum implementation. Even if an FBlock has proprietary extensions, it has to support all mechanisms, specified by the dynamic specification. This ensures operability between different systems for the basic functions of any FBlock.

The dynamic specification includes the minimum of functionality, based on mandatory functions of the FBlock. If an MSC uses optional functions, the part of the MSC, using this function (or the whole MSC) has to be marked as optional.

4 How to build a new FBlock

4.1 Definition of new FBlocks, based on the GeneralFBlock

MOST3

In MOST3, “Function Templates” substitute the GeneralFBlock. The “Function Templates” are defined and maintained by the MOST Cooperation.
The working groups use the functions from the “Function Templates” and define “Abstract FBlocks” for dedicated profiles (e.g., Player with data source; AuxIn). The related working group maintains the “Abstract FBlocks”.
The system integrator specifies its FBlocks by using at least the coordination and mandatory subset of the “Abstract FBlocks”, provided by the working groups and adding its own, proprietary functions. These FBlocks are called “Customized FBlocks” and are basis for the Function Catalog of the system integrator.

The following rules apply:

- New FBlocks have to be based on the GeneralFBlock template.
- All relevant functions with FktID in range “coordination” and all functions with FktID in range “mandatory” must be referenced in the new FBlock.
- When compiling the Function Catalog, all referenced functions are copied to the Function Catalog.
- If required, functions with FktID in range “extension” can be referenced by the new FBlock.
- It is not allowed to add new OPTypes to or delete existing OPTypes from functions of FBlocks of the MOST Cooperation.
- If SetGet is specified, Set has to be specified as well. SetGet has not to be specified necessarily, if OPType Set is specified because of the following exception: Set is possible, but not SetGet in case of properties, which behave like methods (e.g. “Notification”), i.e., whenever the parameter list of “Set” is not identical with parameter list of “Status”.

Any implemented function that is based on an original FBlock of “MOST Cooperation FBlock Library” (independent whether the function is mandatory or optional) has to be operative without usage of proprietary functions. This ensures interoperability of all implemented functions, defined by the FBlock Library of the MOST Cooperation.

Any FBlock specification has to contain

- References to relevant MOST Specification and GeneralFBlock
- Static description of all specified functions
- Dynamic description (MSC) of core functions as exemplary implementation

Additional guidelines:

- New functions should be mapped to the appropriate function class. Unclassified Properties should be avoided as far as possible.
- Use Methods to trigger a process and use Properties to query data or obtain updates through notification.

How to compile new function blocks using GeneralFBlock (exemplary; MOST3)

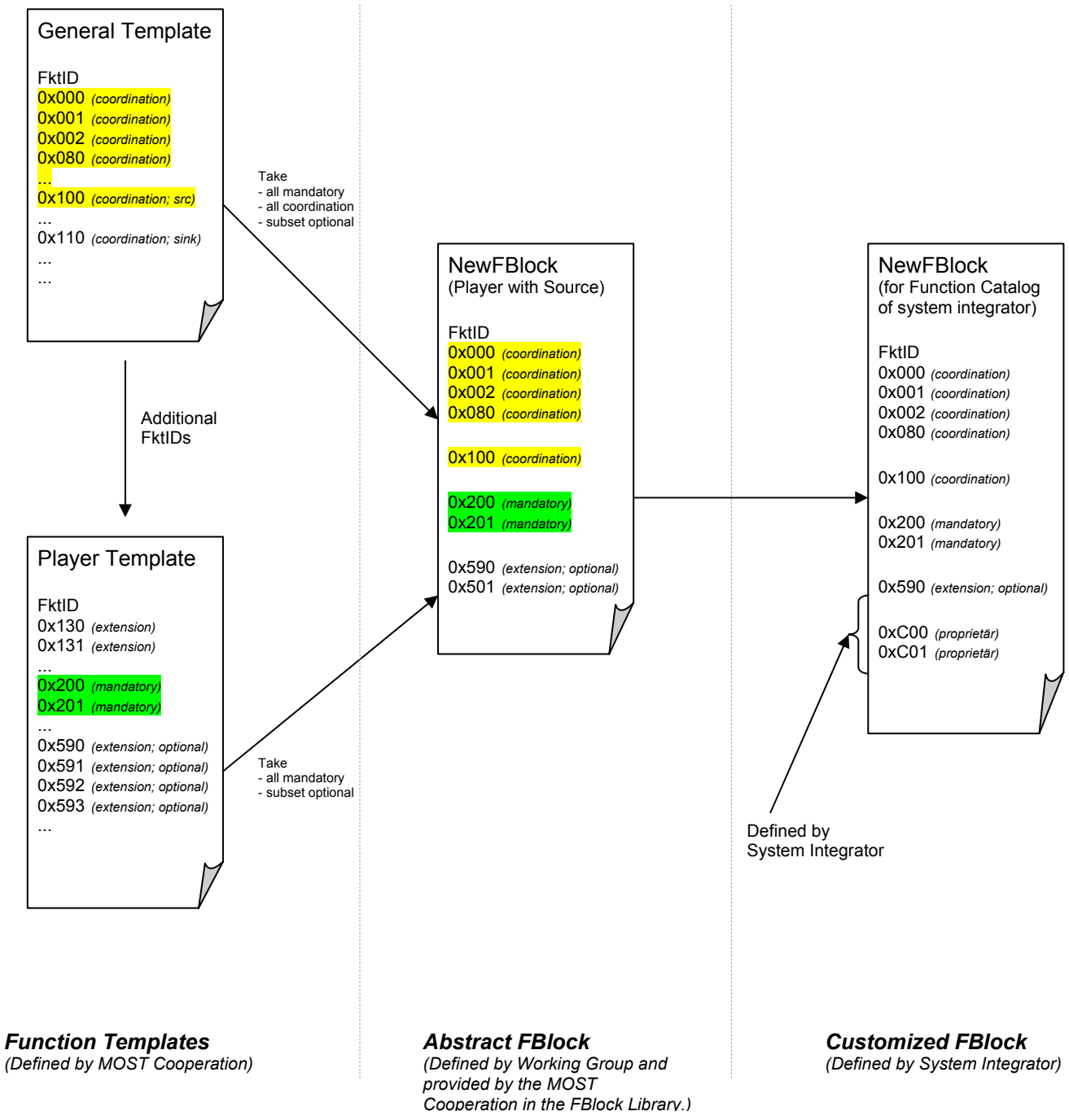

Figure 1: Compiling a new FBlock

During specification of a new FBlock, the following aspects have to be considered:

- Always use the latest version of the GeneralFBlock; add reference to used General FBlock(s); by specification of a General FBlock add reference to MOST Specification the FBlock is based on.
- To increase harmonization:
  The system integrator should inform the responsible WG about proprietary functions for potential consideration in future versions of the MOST Cooperation FBlock Library.

In some situations, the functions defined by FBlocks of the MOST Cooperation are not sufficient for system integrators to implement all necessary features and mechanisms. In that case, the system integrator is able to introduce new functions, new parameters or new Enum values by consideration of the following:

- If possible, proprietary extensions should be avoided. Before specification of proprietary extension, it has to be checked whether the desired mechanism could be achieved with usage of functions, specified by the MOST Cooperation. Even if increased communication effort between MOST devices is the result, MOST Cooperation functions should be preferred.
- Wherever applicable, introduction of a new FktID within the proprietary FktID range should be preferred instead of extension of an existing FktID that was specified by the MOST Cooperation. The original function should be copied to proprietary FktID range and the extensions could be defined there. Additionally, the original function as defined by the MOST Cooperation should be implemented for interoperability. The core functionality, based on reference implementation of dynamic specification has to be supported. (The core functions have to be defined by the MOST Cooperation)
  A controller has to use the MOST Cooperation function if the proprietary function is missing to ensure basic functionality.
- When introducing, e.g., new Enums, a gap between existing values and the new one should be reserved for future extensions by the MOST Cooperation. Additionally, the MOST Cooperation has introduced a predefined range of Enums that could be used for proprietary extensions of the system integrators, beginning from 0xA0.
- An expanded function must not return values, which exceed the range specified by the MOST Cooperation for the relevant function.
- If Enums are extended, a new function has to be defined. This function has to be located in proprietary FktID address range.
- Function and parameter names
  - shall contain no characters other than letters, numbers, and underscores
  - shall not start with a number
  - shall not exceed a certain length (e.g., 26 characters)

Design rules for notification-supporting functions

- It is recommended that an event is designed in a way that it contains all information about the changes.
- It is not recommended to signal just the occurrence of an event, which requires a new request by the controller to retrieve additional information about the details. This would lead to inconsistencies.

4.2 Definition of new FBlock “from scratch”

Even if the new FBlock is not based on any existing FBlock, provided by the MOST Cooperation, some guidelines have to be considered.

- Supplier specific FBlocks will not be reported via FBlockIDs.Status.
- System specific FBlocks will be reported via FBlockIDs.Status
- It is not possible to define proprietary FBlocks as “mandatory”.

The following table regulates who is authorized to use certain FBlockID/FktID combinations.

FktID / FBlockID	Coordination	Mandatory/ Extension	Unique	Proprietary/ System Specific	Proprietary/ Supplier Specific
MOST Co.	MOST Co.	MOST Co.	MOST Co.	System Integrator	Supplier
System Specific	MOST Co.	System Integrator	MOST Co.	System Integrator	Supplier
Supplier Specific	MOST Co.	Supplier	MOST Co.	Supplier	Supplier

Table 2: Responsibilities for FBlockID and FktID ranges

4.2.1 Version upgrade of FBlocks

Upgrading the version of an FBlock, the FktIDs ***should*** not be changed. Any function that exists in both versions should have the same FktID.
In case the FktID is changed, the main version of the FBlock needs to be changed.
This helps avoiding version conflicts, e.g., if two versions of an FBlock exists within one MOST system.

If a function will be deleted during a version upgrade, the FktID should not be used for a new introduced function. It should remain unused for at least one or two version upgrades.

5 How to compile a Function Catalog (System Integrator)

This chapter describes all necessary steps for a system integrator to create a Function Catalog.

The Function Catalog is a collection of all relevant functions of a MOST system, defined by the system integrator. Any MOST devices, dedicated for this system integrator have to implement a subset of functions from the defined Function Catalog.
The Function Catalog can contain functions that are not used in the MOST System.

To compile the Function Catalog, the system integrator has to copy all used FBlocks with all used functions into the Function Catalog-XML file.

Proprietary (supplier specific) functions (FktIDs 0xF00-0xFFE) have to be included into the Function Catalog, too.

The Function Catalog provides an overview of all utilized FBlocks and functions of a particular MOST system. It does not provide information about which FBlock and function has to be implemented into each single MOST device of the system.

As every single FBlock already contains references to relevant version of MOST Specification, the Function Catalog itself must not contain reference to any MOST Specification. All FBlocks within one Function Catalog should base on the same General FBlock and on the same MOST Specification. With generating the Function Catalog, all references to FBlocks of MOST Cooperation will be deleted. However, the version of the MOST Specification the Function Catalog is based on has to be stored in the Function Catalog.

The system integrator may add additional (system specific) information to the Function Catalog, not covered by FBlock specification (e.g., which function uses notification, communication via control channel or synchronous channel) or restrict presettings of the FBlock specification for the specific system (e.g., reduced range of values).

6 Appendix A: List of Figures

7 Appendix B: List of Tables

Notes:

Notes:

Notes:

