INTERNATIONAL STANDARD

ISO/IEC 15938-1

First edition
2002-07-01

AMENDMENT 2
2006-04-01

Information technology — Multimedia content description interface —

Part 1: **Systems**

AMENDMENT 2: Fast access extension

Technologies de l'information — Interface de description du contenu multimédia —

Partie 1: Systèmes

AMENDEMENT 2: Extension d'accès rapide

Reference number
ISO/IEC 15938-1:2002/Amd.2:2006(E)

Documents
Document Sharing Hub
© ISO/IEC 2006

Full document details: ISO_IEC 15938-1_2002_Amd 2_2006

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

© ISO/IEC 2006

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

© ISO/IEC 2006 – All rights reserved

Documents
Document Sharing Hub

Full document details: ISO_IEC 15938-1_2002_Amd 2_2006

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Amendment 2 to ISO/IEC 15938-1:2002 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

© ISO/IEC 2006 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO_IEC 15938-1_2002_Amd 2_2006

Documents
Document Sharing Hub

Full document details: ISO_IEC 15938-1_2002_Amd 2_2006

Information technology — Multimedia content description interface —

Part 1:
Systems

AMENDMENT 2: Fast access extension

This document preserves the sectioning of ISO/IEC 15938-1. The text and figures given below are additions and/or modifications to those corresponding sections in ISO/IEC 15938-1. All figures and tables shall be renumbered due to the addition of several figures and tables.

Add the following definitions to subclause 3.2 (keep alphabetical order), then renumber all definitions in subclause 3.2:

path index key
value representing the path to the element to be indexed/located, and the relative path to the fields to be keyed/searched.

value index key
set of encoded field values to be keyed/searched.

index stream
set of Index Access Units which together form the whole of the indexing data,

index decoder init
initialisation data for an index stream.

index access unit
index access unit header and associated structures forming a logical unit of access.

index access unit header
list of structures contained within this Index Access Unit

path index
structure allowing path index key to value index reference lookup.

value index
structure allowing value index key to value sub index reference lookup.

value sub-index
structure allowing value index key to BiM stream reference lookup.

node reference
reference from one node to another within a list or B-Tree structure.

data repository reference
reference to data entry within the binary or string data repository structures.

BiM stream reference
reference to a BiM encoded fragment within a BiM stream.

© ISO/IEC 2006 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO_IEC 15938-1_2002_Amd 2_2006

local access unit
default Access Unit associated with a Path Index.

local BiM stream reference
reference to a BiM encoded fragment contained within the local access unit.

remote BiM stream reference
reference to a BiM encoded fragment contained within the BiM stream, where the access unit is specified by an access unit ID.

value index reference
reference to a value index structure.

value sub-index reference
reference to a sub value index structure.

position codes reference
reference to a position codes entry within a position codes structure.

position code
location of an XML element within its parent element.

position codes
set of position code values for all elements within a context path.

BTree
binary decision tree, where each node can have multiple keys.

BTree order
specifies the maximum number of child nodes of a node in a BTree.

indexed element
XML element to which an index refers.

BiM stream reference format
specifies the format of the BiM stream reference.

value encoding
specifies how data has been encoded in value index key.

Add the following subclause 5.9.1:

5.9.1 General Description

Using the ISO/IEC 15938-1 index encoding, only fragments of the description that are of immediate interest to the terminal can be selectively acquired and combined with the current description tree. The terminal can search the index information to determine which fragments contain a node at a given location which has a related node with a given value within the description. Additionally the terminal may search for fragments containing nodes which have related node values falling within a given range.

© ISO/IEC 2006 – All rights reserved
Documents
Document Sharing Hub
Full document details: ISO_IEC 15938-1_2002_Amd 2_2006

The index information can also be compiled to allow the terminal to search for fragments containing nodes with multiple given related node locations, and respective values, within the description. This can allow the terminal to perform searches with multiple conditions, without needing to consolidate multiple result sets. As the indexing stream is optional a stream may consist of either

A DecoderInit and a description stream

A DecoderInit, an IndexDecoderInit, an index stream, and a description stream.

Before an index stream can be queried both the DecoderInit for the BiM stream to which the index stream belongs, and the IndexDecoderInit for the index stream must be acquired. However acquiring fragments from the description stream, without querying the index stream, only requires the DecoderInit to be acquired.

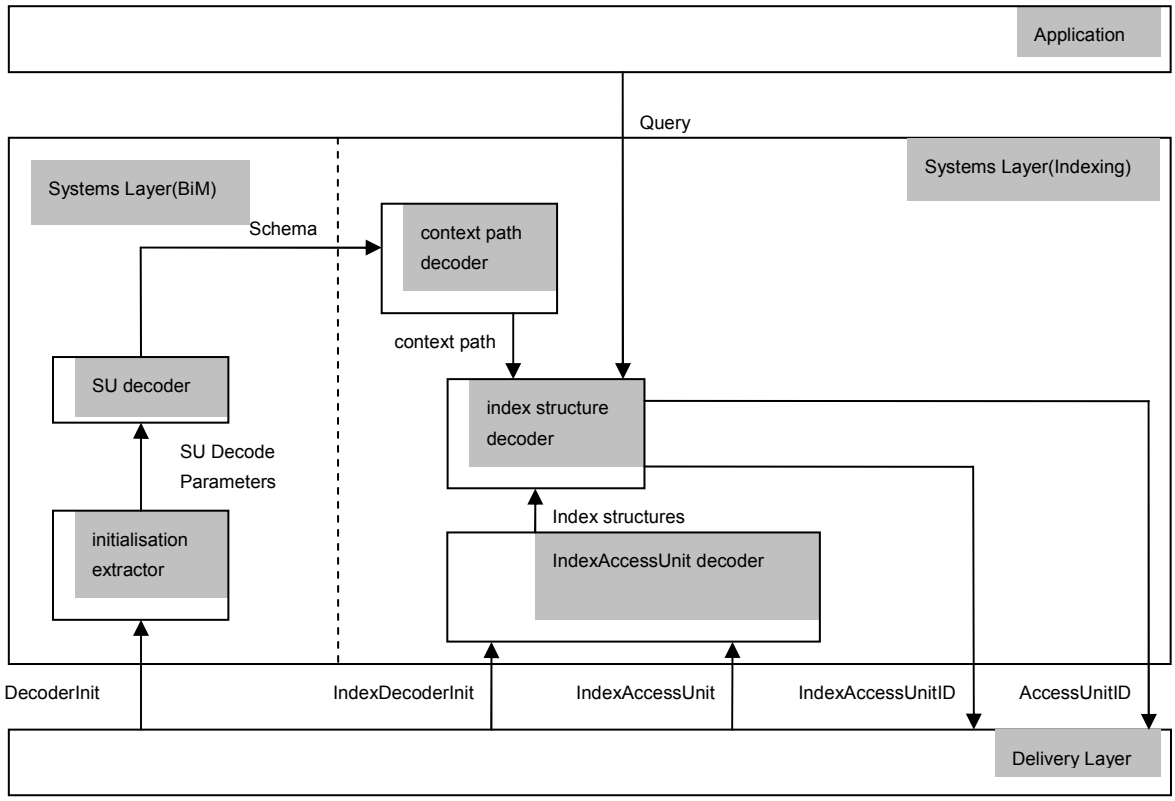

Figure Amd2.1 — Indexing Enabled Terminal Architecture Extension.

All components of Systems Layer(Indexing) section are non-normative

The Terminal Architecture for a BiM enabled terminal may be extended to support indexing, as shown in Figure Amd2.1. Figure Amd2.1 shows only the extensions to the Terminal Architecture, and not the complete architecture. The components shown in the Systems Layer(BiM) section of Figure Amd2.1 are the existing components of the standard Terminal Architecture.

Add the following subclause 5.9.2:

5.9.2 Options for multi criteria query

There are two main methods of querying for fragments when there are multiple criteria, multi-value indexing, and multi-stage indexing. These different methods are distinct and offer two complementary optimizations.

Multi-value indexing allows the whole data set to be indexed in a very compact manner. The size of the index stream and the number of comparisons of values is minimized, allowing the index to perform a multi criteria query using the smallest amount of index data and queries possible. Partitioning of indices with larger data

© ISO/IEC 2006 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO_IEC 15938-1_2002_Amd 2_2006

sets allows maximum IndexAccessUnit sizes to be imposed. This is useful when the underlying transport layer has an imposed, or preferred size for a unit of access, as might be the case with network packets, or transport layer data buffers. It is usual for several IndexAccessUnits to be required to complete a query, hence the IndexAccessUnit is not independent.

Multi-stage indexing allows the data set to be sectioned into multiple smaller index stream segments. This increases both the overall size of the index stream and the number of value comparisons by a moderate amount. However, multi-stage indexing can facilitate more efficient use of resources in client terminal devices where caching of stream index data is necessary but the size of the stream index data prohibits caching all of it. Caching is desirable where either the index stream is not always available or the acquisition time of Index Access Units from the index stream is significant. In a multi-stage index the data size of each independent segment of the index stream is reduced, allowing a whole segment of the index to be cached from the index stream into memory and searched independent of the Index Stream. This is often a better optimisation than attempting to cache a portion of an index arranged as a single segment.

Multi-value indexing is intended to be used in situations where efficient index size is a priority and there are no significant restraints imposed by client terminal resources. There are two formats of multi-value indexing, composite value, and hierarchical single value indexing. In composite value indexing the values are stored as an N column table, where N is the number of related values. In hierarchical single value indexing the values are stored as an N level tree, with the tree's nodes containing 1 column tables (Lists). The composite value index is intended for use where there is unlikely to be multiple instances of the related node with the same value, whilst the hierarchical single value indexing is intended for use where there are many instances of the related node with the same value in the BiM fragments to be indexed. It is important to take care to choose the order of the related nodes carefully, as the related node with most common values placed at the highest level of the hierarchy will usually produce the smallest index stream and the minimum number of value compares when querying.

Multi-stage indexing is intended for use in situations where client terminals with limited resources must access a very large amount of BiM fragments by index, and for which there are a small number of search criteria common for most queries. The common criteria can be used to segment the index stream into a collection of index stream segments, each segment then being accessed and searched independently of other segments. This allows searches to start on an initial segment stored in a cache and then progress to the relevant follow-on segment which, if not cached, must be acquired from the index stream. The search is therefore completed with a minimum number of acquisitions and without requiring the whole multi-stage index to be cached.

Note that the method of caching and cache maintenance is implementation dependent, and not defined in this specification.

Note - Another situation where multi stage indexing can be used, is when consolidating multiple independent index streams. This may be the case if there are multiple providers of BiM fragments each with an associated index stream. Consolidation of the index streams can be achieved easily by modifying each index stream to be a second stage segment. A first stage index would then be created to associate a provider to a second stage segment within the index stream.

Change the following sentence at the end of subclause 7.1 as indicated:

Several other coding modes are initialised in the DecoderInit related to the features used by the binary description stream: the insertion of elements, the transmission of schema information, references to fragments and a fixed length context path.

And add the following sentence at the end of subclause 7.1:

The fixed length context path mechanism provides a simplified addressing of nodes for usage scenarios where only a limited number of nodes need to be addressed. This is done by a table that uniquely maps fixed length codes to full context paths.

© ISO/IEC 2006 – All rights reserved
Documents
Document Sharing Hub
Full document details: ISO_IEC 15938-1_2002_Amd 2_2006

In subclause 7.2.2, insert grey marked rows at the position indicated:

If (! NoAdvancedFeatures) {		
AdvancedFeatureFlags_Length	8+	vluimsbf8
/** FeatureFlags **/		
InsertFlag	1	bslbf
AdvancedOptimisedDecodersFlag	1	bslbf
AdditionalSchemaFlag	1	bslbf
AdditionalSchemaUpdatesOnlyFlag	1	bslbf
FragmentReferenceFlag	1	bslbf
MPCOnlyFlag	1	bslbf
HierarchyBasedSubstitutionCodingFlag	1	bslbf
ContextPathTableFlag	1	bslbf
ReservedBitsZero	FeatureFlags_Length*8-8	bslbf
/** FeatureFlags end **/		

If (**ContextPathTableFlag**) {		
ContextPathTable()		
}		
*/** FUUConfig - Advanced optimised decoder framework **/*		
If (AdvancedOptimisedDecodersFlag) {		

ContextPathTable {		
ContextPathTable_Length	8+	vluimsbf8
ContextPathCode_Length	8+	vluimsbf8
NumberOfContextPaths	8+	vluimsbf8
CompleteContextPathTable	1	bslbf
for(i=0;i<NumberOfContextPaths;i++){		
ContextPath_Length[i]	5+	vluimsbf5
ContextPath()[i]	ContextPath_Length[i]	
If(!CompleteContextPathTable){		
ContextPathCode[i]	ContextPathCode_Length	bslbf
}		
}		
nextByteBoundary()		
}		

© ISO/IEC 2006 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO_IEC 15938-1_2002_Amd 2_2006

In subclause 7.2.3, insert,

ContextPathTableFlag	Signals the presence of a context path table in the decoder init.
ContextPathTable_Length	Defines the number of bytes used for the indication of the ContextPathTable. Note – This length provides a simple framework to skip the table.
ContextPathCode_Length	Signals the length of the context path codes in number of bits.
NumberOfContextPaths	Signals the number of ContextPaths contained in the ContextPathTable.
CompleteContextPathTable	Signals if the ContextPathTable is complete and ordered according to the assignment of ContextPathCodes. If CompleteContextPathTable is set to '1' the ContextPathCodes are assigned in the order the ContextPaths are specified in the ContextPathTable starting from '1'. If CompleteContextPathTable is set to '0' the ContextPathCodes are assigned explicitly. The ContextPathCode '0' is reserved.
ContextPath_Length	Signals the number of bits used for the following ContextPath[i]()
ContextPath[i]()	Signals the ContextPath as specified in subclause 7.6.2 with the following restrictions: - ContextModeCode is set to '001' - PositionCode() is an empty bitfield
ContextPathCode[i]	Signals the ContextPathCode of .ContextPath[i]

© ISO/IEC 2006 – All rights reserved
Documents
Document Sharing Hub
Full document details: ISO_IEC 15938-1_2002_Amd 2_2006

In subclause 7.6.2, insert grey marked rows at the position indicated:

FragmentUpdateContext () {	**Number of bits**	**Mnemonic**
SchemaID	ceil(log2(`NumberOfSchemas`))	uimsbf
ContextModeCode	3	bslbf
If (ContextModeCode=='101'){		
ContextPathCode	ContextPathCode_Length	bslbf
for (i=0; i < TBC_Counter(ContextPathCode); i++) {		
PositionCode()		
}		
}		
else {		
ContextPath()		
}		
}		

In subclause 7.6.4, insert grey marked rows at the position indicated:

Code	**Context Mode**
…	
101	Navigate in "Absolute addressing mode" from the selector node to the node specified by the Context Path signaled by the ContextPathCode.
110-111	Reserved

Add the following clause 10:

10 Indexing

10.1 Overview

The index is provided to support fast random access into a BiM stream. The index allows access to the FUU's, within the BiM stream, containing a desired XML node, either element or attribute, based on the specification of one or more related node, element or attribute, values. For instance, determining FUU's which contain "Car" elements, who's "Color" attribute is "Red". The search specification uses the context path of the desired XML node, and the relative context paths of the related nodes, this specification is termed the PathIndexKey. This allows a flexible PathIndexKey, which is capable of indexing complex type elements, simple type elements, and attributes according to one or more criteria.

This section gives a general overview of the structure and functionality of the index, and how it is accessed to arrive at a resultant set of FUU's matching a given search criteria.

© ISO/IEC 2006 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO_IEC 15938-1_2002_Amd 2_2006

The index is composed of two parts, the Path Index, and the Value Index. The Path Index allows the particular Value Index relating to the PathIndexKey, specified in the search criteria, to be located. The Value Index allows the set of BiM stream references which contain the desired XML node, with values for the PathIndexKey's related nodes meeting the search criteria. The indexing technologies specified here, allow a Value Index to be partitioned into smaller Value Sub Index structures. This partitioning allows a Value Index to contain an unlimited number of entries without increasing the resource requirements, in particular working memory, of the client.

The data structures in this specification allow for the Value Index to be created with the values for the PathIndexKey's related nodes, to be represented as a hierarchical index, or as a consolidated index. The hierarchical index can offer much smaller index structures, and faster searches, if the PathIndexKey's related nodes values contain significant repetition, as this allows repeated values to be grouped together and entered into the Value Index only once. If there is not significant repetition, then the Value Index can be represented in a consolidated index, which allows all of the values for the PathIndexKey's related nodes to be consolidated and represented within a single index structure. A search on the Value Index will result in a set of zero or more BiM stream references. The BiM Stream references locate the FUUs which contain the desired XML nodes with related node values satisfying the search criteria. The BiM Stream references can optionally specify the position of the desired XML node within each FUU, in addition to the FUU reference itself.

To demonstrate the searching process, consider the example where a simple, single criteria, search is being made for a "Picture" node, whose related node, "Subject", has a value of "Winter"

© ISO/IEC 2006 – All rights reserved
Documents
Document Sharing Hub
Full document details: ISO_IEC 15938-1_2002_Amd 2_2006

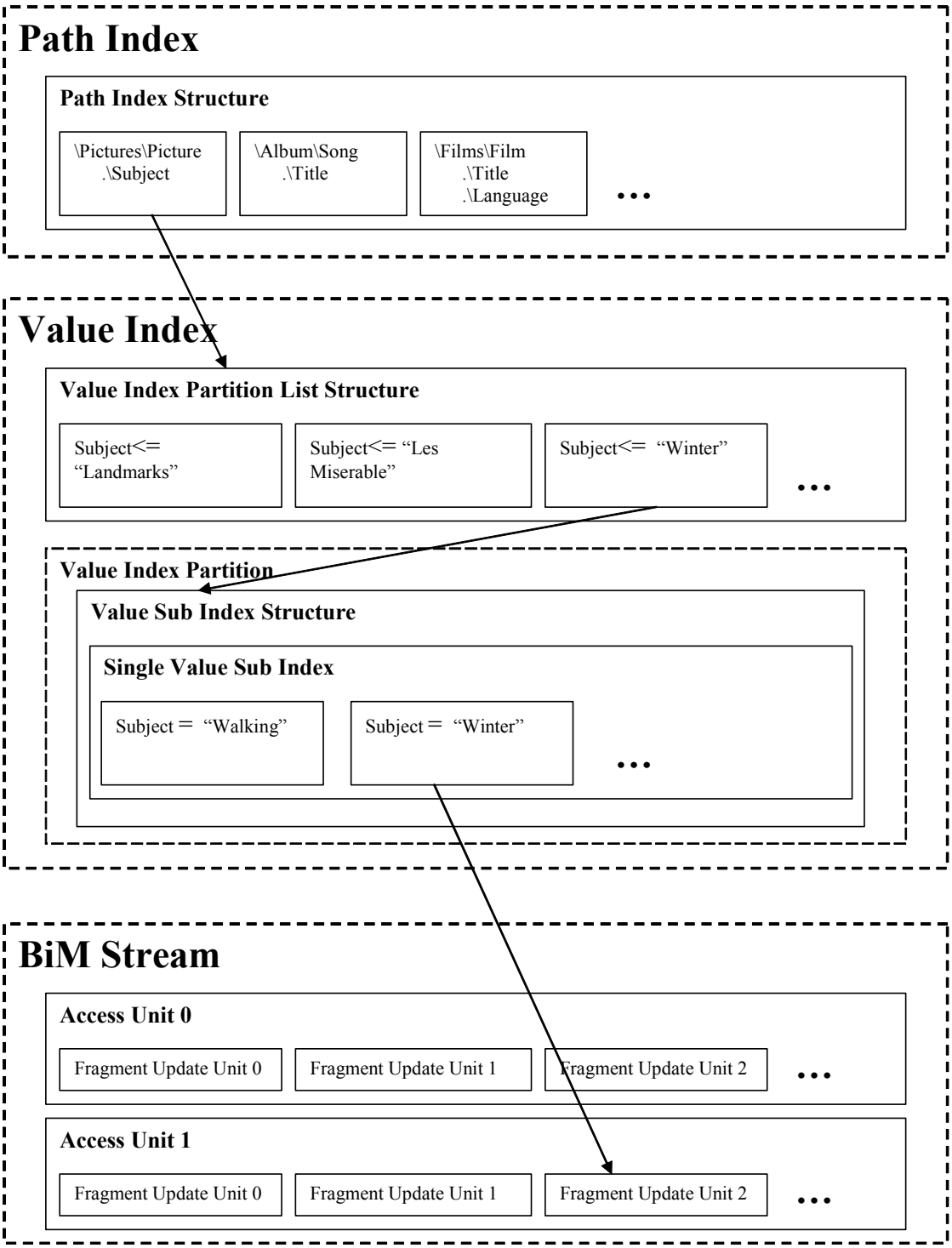

Figure Amd2.2 — Block diagram of the BiM Index structure

The Path Index is first searched to locate the relevant Value Index. This is achieved by scanning the Path Index structure. Once the Value Index has been determined, its Value Index Partition List structure is used to determine which of the Value Index partitions will contain the "Subject" being searched for, in this case "Winter". Now the Value Sub Index structure can be accessed, which in this case contains a Single Value Index, for the values of "Subject" in this partition. These values are then searched to determine the BiM Stream references which match the search criteria.

© ISO/IEC 2006 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO_IEC 15938-1_2002_Amd 2_2006

The next example explains how the Compound Value Index can be used to search for the “Film” element with two related elements “Title” and “Language”

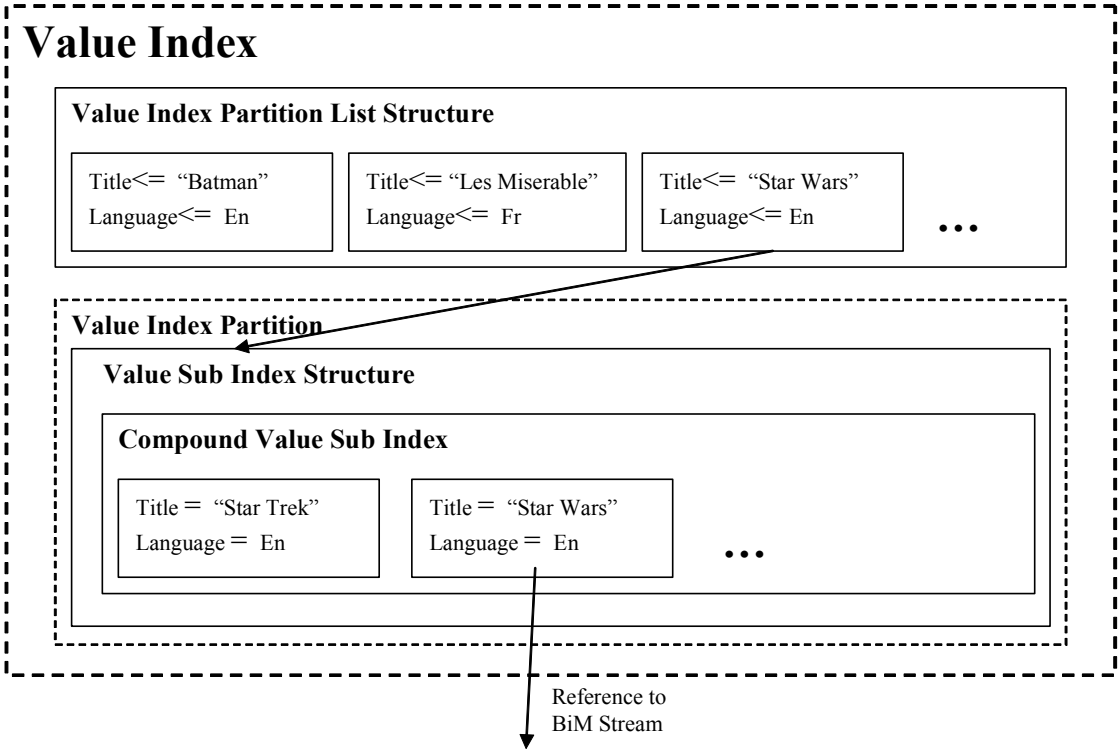

Figure Amd2.3 — Block diagram of the compound value index structure

The process is the same as in the first example, except that the Compound Value Index contains values for both related nodes.

The final example, is the same as the previous example, except that a hierarchy of Single Value Index entries has been used rather then the Compound Value Index.

© ISO/IEC 2006 – All rights reserved
Documents
Document Sharing Hub
Full document details: ISO_IEC 15938-1_2002_Amd 2_2006

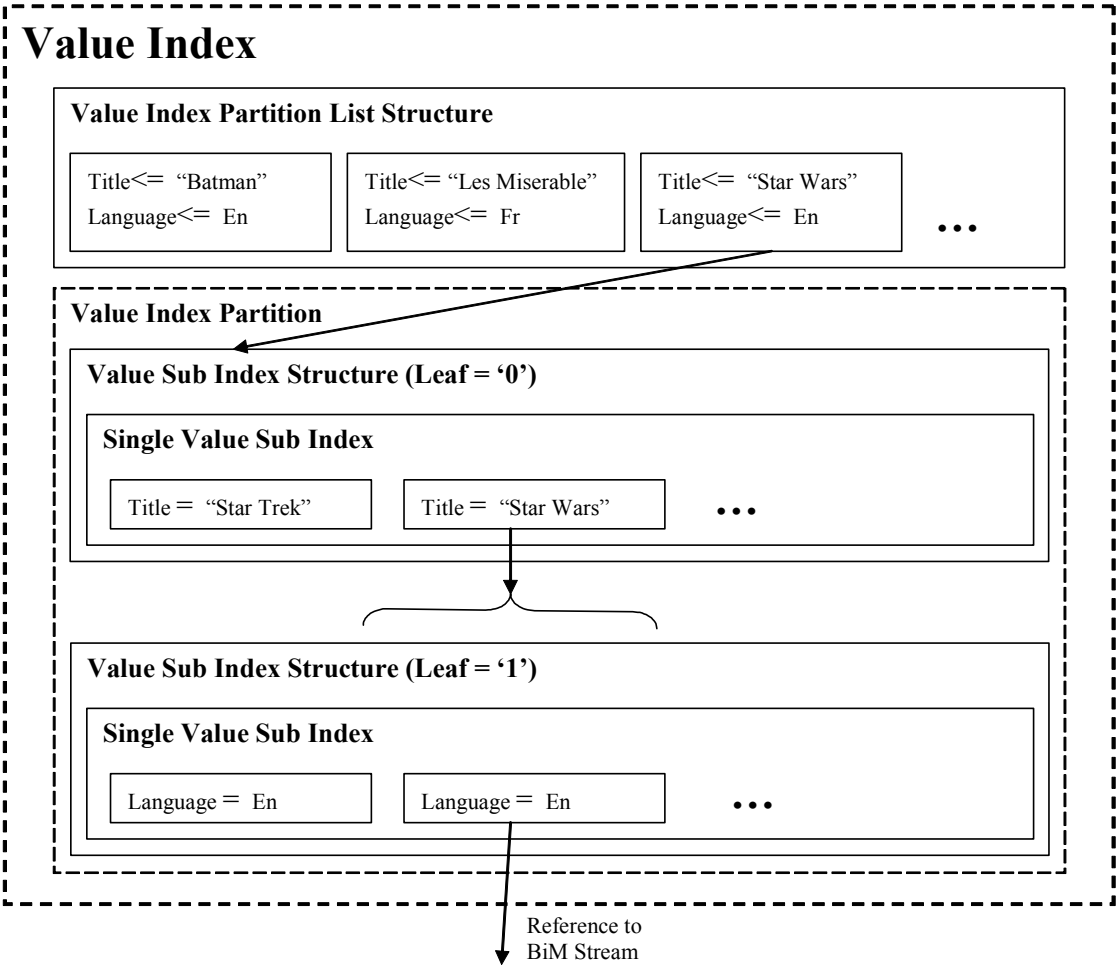

Figure Amd2.4 — Block diagram of the hierarchical single field index structure

This shows that the first related node, "Title" is searched first, but instead returning the BiM stream references, it returns a range to search in the child Sub Value Index. The child is then searched for the correct value for the "Language" element to determine the BiM stream references.

10.2 Characteristics of the delivery layer

The delivery layer is an abstraction that includes functionalities for the synchronization, framing and multiplexing of indexing streams with other data streams. Index streams may be delivered independently or together with the associated Description Stream. No specific delivery layer is specified or mandated by ISO/IEC 15938.

A delivery layer (DL) suitable for conveying ISO/IEC 15938 index streams shall have the following properties in addition to the properties defined for decoding of description streams:

— The DL shall provide a mechanism to communicate an index stream from its producer to the terminal.

— The DL shall provide a mechanism by which a random access point to the index stream can be identified.

— The DL shall provide a suitable random access mechanism allowing access to an IndexAccessUnit by use of a 16 bit IndexAccessUnit identifier.

— The DL shall provide a default 16 bit IndexAccessUnit identifier for each PathIndex in the index stream.

© ISO/IEC 2006 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO_IEC 15938-1_2002_Amd 2_2006