INTERNATIONAL STANDARD

ISO/IEC 23007-1

First edition
2010-11-01

Information technology — Rich media user interfaces —

Part 1: **Widgets**

Technologies de l'information — Interfaces d'utilisateur au support riche —

Partie 1: Widgets

Reference number
ISO/IEC 23007-1:2010(E)

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.

COPYRIGHT PROTECTED DOCUMENT

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

Contents

Page

Foreword ... v
Introduction ... vi
1 Scope ... 1
2 Normative references ... 1
3 Terms and definitions ... 1
4 Abbreviations and symbols ... 3
5 Conventions ... 3
6 Architectures ... 3
6.1 Widget manager architecture ... 3
6.2 Widget architecture ... 4
7 Widget management ... 5
7.1 Widget composition ... 5
7.2 Widget life cycle ... 5
7.3 Widget communication ... 7
7.4 Widget context ... 8
8 Widget packaging and delivery ... 9
8.1 Overview ... 9
8.2 Unpackaged delivery ... 9
8.3 Packaged delivery using the ISOFF ... 9
9 Widget API ... 10
9.1 Overview ... 10
9.2 The MPEGwidget interface ... 10
9.3 The InterfaceHandler interface ... 11
10 Widget manifest syntax ... 13
10.1 Overview ... 13
10.2 The <widget> element ... 13
10.3 The <name> element ... 13
10.4 The <description> element ... 13
10.5 The <author> element ... 14
10.6 The <license> element ... 14
10.7 The <icon> element ... 14
10.8 The <feature> element ... 14
10.9 The <param> element ... 14
10.10 The <preference> element ... 14
10.11 The <content> element ... 14
10.12 The <mw:contextConfiguration> element ... 15
10.13 The <mw:preferenceConnect> element ... 16
10.14 The <mw:interfaces> element ... 16
10.15 The <mw:interface> element ... 17
10.16 The <mw:messageIn> element ... 18
10.17 The <mw:messageOut> element ... 20
10.18 The <mw:input> element ... 22
10.19 The <mw:output> element ... 22
10.20 The <mw:component> element ... 23
10.21 The <mw:requiredInterface> element ... 24
10.22 Predefined interfaces and communications ... 25
10.23 Connection between scene and interface constructs ... 27

11 Widget context information syntax 28
11.1 Overview 28
11.2 The <ci:contextInformation> element 28

12 Security considerations 29
12.1 Widget security 29
12.2 Communication security 29

Annex A (normative) MIME type registration for application/mw-manifest+xml 30

Annex B (normative) MIME type registration for application/mw-context+xml 32

Annex C (informative) Widget manifest example 34

Annex D (informative) MPEG-U Use Cases and Implementation 36

Annex E (informative) Relationship with W3C 40

Bibliography 41

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.

ISO/IEC 23007-1 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 23007 consists of the following parts, under the general title *Information technology — Rich media user interfaces*:

— *Part 1: Widgets*

— *Part 3: Conformance and reference software*

Advanced user interaction interface will form the subject of a future Part 2.

Introduction

User interface represents a crucial feature for many consumer devices and services. User interfaces have recently evolved to support more media types including audio, video, 2D or 3D graphics and rich media functionalities. User interfaces are also evolving towards flexible and composite collections of small dedicated applications retrieved from different sources and aggregated into an effective and user friendly interface. Such applications are generally called widgets, a widget being a self-contained entity, with an interactive and dynamic visualization.

Additionally, more and more devices are capable of displaying rich media user interfaces, from desktop computers, to mobile devices, to home appliances, including TV sets. In this heterogeneous environment, users expect a homogeneous, unified experience when interacting with their devices.

The objective of this part of ISO/IEC 23007 is to provide normative interfaces between widgets and widget managers, to allow widgets from different service providers to run, communicate and be transferred within a unique framework.

In this part of ISO/IEC 23007, widgets can be processed by entities running on different devices, called widget managers, in charge of processing and managing the life cycle of the widgets supporting communications with other entities locally or remotely deployed and enabling widget mobility across devices.

This part of ISO/IEC 23007 is also known as “MPEG-U”. This part of ISO/IEC 23007 addresses the normative aspects of the MPEG-U widgets. In particular, it specifies widget packaging formats, aspects for widget communications with external entities and for widget mobility. It also contains a technical annex describing a list of use cases and examples to address such use cases. ISO/IEC 23007-2 will specify advanced user interaction interfaces to support various advanced user interaction devices. ISO/IEC 23007-3 addresses reference software and conformance aspects.

This part of ISO/IEC 23007 builds upon the W3C specification for widgets, packaging and configuration:

- to ensure that the widget packaging format and configuration documents are compatible with the MPEG media types which can be used to describe widgets (e.g. 2D or 3D content, MPEG-4 BIFS or MPEG-4 LASeR). For restricted profiles of these languages, this implies in particular the ability to create meaningful widgets which do not rely on scripting languages.
- to ensure that widgets can be transported on any existing transport mechanisms, in particular those defined by MPEG (e.g. ISO base media file format and the MPEG-2 Transport Stream).
- to ensure that it is targeted for domains in addition to Web-connected devices, e.g. broadcast, mobile or home networking domains.
- to enable interoperable communications between a widget and other entities (including widgets), these entities being remote (e.g. UPnP services [4]) or local services, or other widgets running in the same environment.
- to enable MPEG-specific requirements, such as the ability to dynamically update the widget presentation or to display a widget in a dynamic and interactive simplified representation.
- to enable widgets, mobility across devices while maintaining the state of the widget.

A general description of the architectures of this part of ISO/IEC 23007 is provided first, to clearly identify normative and non-normative entities.

This is followed by descriptions of behaviour and syntax of the normative elements, and syntax and examples of the normative elements for which a definition of new XML syntax is needed, namely widget manifest and widget API.

Annex A and Annex B provide registration forms for two media types defined within this part of ISO/IEC 23007. A complete example of the standardized technology can be found in Annex C. Examples of use cases which can be realized using MPEG-U are provided in Annex D. Finally, a description of the relationship of this specification with the W3C widgets family of specifications is provided in Annex E.

Information technology — Rich media user interfaces —

Part 1: **Widgets**

1 Scope

This part of ISO/IEC 23007 defines a specification for the exchange, the control and the communication of widgets with other entities, a widget being a self-contained living entity, with an interactive and dynamic visualization.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 14496-12, *Information technology — Coding of audio-visual objects — Part 12: ISO base media file format (technically identical with ISO/IEC 15444-12)*

W3C WPC "Widgets 1.0: Packaging and Configuration", W3C Working Draft 24 February 2009, available at http://dev.w3.org/2006/waf/widgets

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

3.1
communication entity
device or widget, local or remote to a widget manager, providing services

3.2
context information
set of data needed to reproduce a state and preferences of a widget, should it be deactivated and reactivated, possibly in a different widget manager

3.3
device
combination of hardware and software or just an instance of software that allows a user to perform actions

3.3
full representation
description of the widget appearance and behaviour given in a scene description language which represents the complete version of the widget with its maximum complexity and behaviour

3.4
icon
interactive, possibly animated and/or scripted, raster image or vector graphics that can be used to graphically represent the widget before the full representation is loaded

3.5
iconic representation

NOTE See simplified representation.

3.6
locale
set of information which defines local variants of widgets, elements for a particular country or region or language

3.7
manifest
XML description of the widget containing all the information necessary for the widget manager to process the widget

3.8
presentation engine
entity processing the scene description of the widget to provide its animated and interactive behaviour through composition and rendering

3.9
resource
part of a widget, in the form of either a file or a stream, which is needed by the widget manager or presentation engine to process and present the widget

3.10
scene description
description defining an audiovisual presentation for the widget in terms of spatiotemporal layout, and interactions by using text, graphics, animations, images, videos, sounds, etc.

3.11
service
system supporting interaction, local or over a network, by means of message exchanges (e.g. UPnP service or Web service)

3.12
simplified representation
description of the widget appearance and behaviour given in a scene description language which represents a version of the widget with a reduced complexity and/or behaviour compared to the full representation

3.13
widget
self-contained entity, with extensive communication capabilities, within a rich media user interface, composed of a manifest and associated resources, including scene descriptions for the full and simplified representations and context information

3.14
widget manager
user agent, processing widgets, in particular for communication between the widget and other entities

3.15
widget package
collection of the widget manifest and associated resources in a particular format used for delivery and storage

4 Abbreviations and symbols

ISOFF	ISO Base Media File Format
BIFS	BInary Format for Scene
LASeR	Lightweight application scene representation
SVG	Scalable Vector Graphics
UPnP	Universal Plug and Play
W3C WPC	Widgets 1.0: Packaging and Configuration
W3C WAE	The widget Interface
WLC	Widget life cycle
WM	Widget manager
XML	eXtensible Markup Language

5 Conventions

XML element and attribute names are written using this `style` in the text.

XML examples are written as follows:

```
    <mw:messageOut name="Search">
     <mw:output name="searchString" scriptParamType="string"/>
     <mw:input name="URL" scriptParamType="string"/>
    </mw:messageOut>
```

Script code examples are described as follows:

```
function myFunction { … }
```

6 Architectures

6.1 Widget manager architecture

The architecture of the different elements specified in this part of ISO/IEC 23007 is depicted in Figure 1.

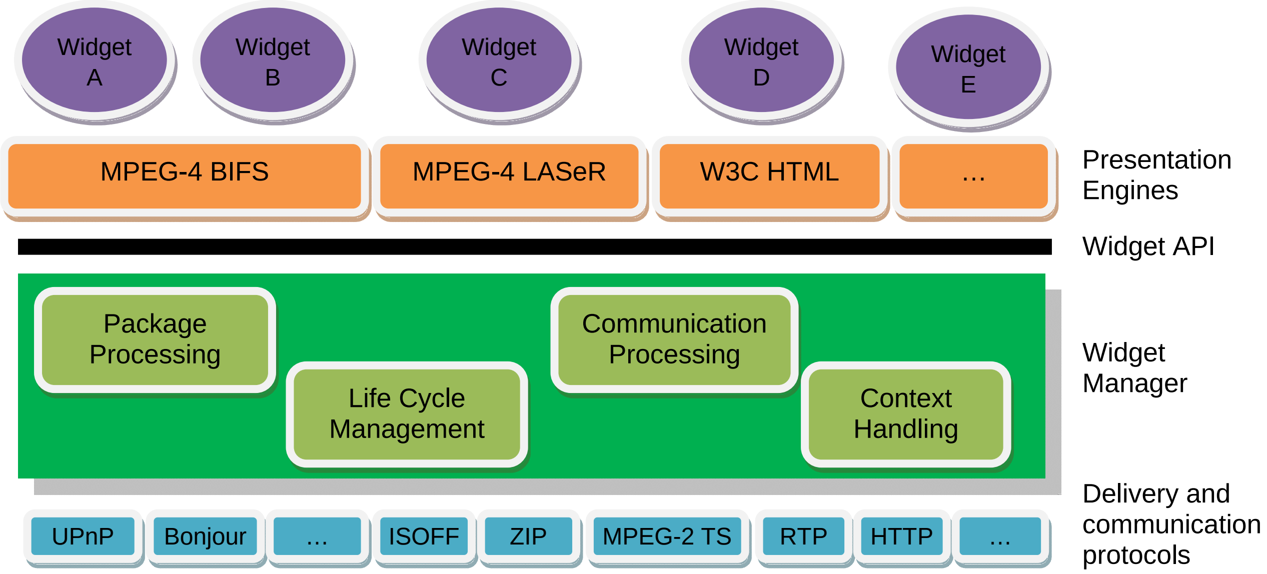

Figure 1 — Architecture of the rich media user interface standard

On client devices, an entity, called widget manager, shall provide the following functionalities:

— Processing widget packages, as defined in W3C WPC,

— Managing the life cycle of widgets, as described in 7.2,

— Managing the communication between widgets and external entities, as described in 7.3,

— And managing context information for widget, as described in 7.4.

Additionally, widget managers may support one or more of the widget delivery mechanisms described in Clause 8.

The presentation of a scene description of a widget is demanded to a presentation engine, running on the client device, supporting the scene description language. The presentation engine is an external entity interfaced with the widget manager. The widget manager may provide support for the API defined in Clause 9 to the presentation engine.

6.2 Widget architecture

In this part of ISO/IEC 23007, a widget shall be made of:

a) the manifest, i.e. an XML description which serves as an entry point for the widget and provides:

i) metadata about the widget (e.g. author name, short title, …),

ii) the format and the URL of the scene descriptions for the full and simplified representation,

iii) description of the communication capabilities of the widget,

iv) a set of preferences that needs to be restored and an indication whether they need to be saved as part of the context information;

b) optional context information to be used for restoration on a same or different widget manager;

c) the full representation of widget, i.e. a set of resources (e.g. scene description data, images, text content, …), allowing a full featured presentation of the widget;

d) one or more simplified representations, also called icons, i.e. an optional set of resources (possibly the same as the first one), allowing the retrieval or presentation of the widget in a simple state or way.

The simplified representation may be a different scene from the full representation, or may be a fragment of the full representation. Using a different scene may allow a fast presentation of the widget when the full representation is not yet available. This may also allow a simplified presentation when the full processing is not required, or too demanding.

7 Widget management

7.1 Widget composition

The spatial, temporal and interactive composition of the simplified or full representation of a widget shall be as specified by this representation. If the full and simplified representation both point to the same resource, a single instance of that resource may be used and the simplified and full rendering may work on the same synchronized compositing.

The spatial and temporal composition of the widgets together with other widgets or applications is not specified.

EXAMPLE In case of widgets being displayed on a TV at the same time as a TV program, widgets may be displayed on top of the TV program, or the TV program may be resized to show the widgets on the side, or any other paradigm may be used.

7.2 Widget life cycle

The widget life cycle (WLC) represents the set of states and transitions that a widget can be in during its lifetime. The WLC is depicted in Figure 2.

The widget manager shall apply the steps specified in Section 9 of W3C WPC. If this process fails, the widget is placed in the invalid state and the widget manager shall not process this widget. If the process is successful the widget manager shall place the widget in the validated state and provide the information from the manifest to the presentation engine. The widget manager shall not offer a widget for activation until all interfaces having a required attribute set true are ready to be bound.

NOTE 1 The icon or a text string may be used to show to the user that the widget is validated. This exact behaviour is implementation specific. A validated widget can then be selected for presentation. This is done either automatically or by user interaction, and its full representation and/or its simplified representation is started. The selection mechanism is also implementation specific. When a scene representation of a validated widget is selected for presentation, this representation is loaded by the presentation engine which processes it as defined in the related scene representation specification.

EXAMPLE If the widget manager and the presentation engines are interfaced to propose a widget dock, the presentation engine may allow the user to choose which simplified representation is to be used in the dock.

If the loading of the scene representation fails, the widget manager shall place the widget into the invalid state and shall not process it. If the loading is successful, the widget state becomes active and the widget manager shall apply the communication and context management behaviours as defined in 7.3 and 7.4.

NOTE 2 The presentations of the different widget representations are not exclusive. The simplified and full representations may be presented together by the widget manager, e.g. in a dock for the simplified presentation and in the main area for the full representation of the same widget. The transitions (if any) between the presentation of the full and simplified representation are implementation specific. Additionally, each representation of a widget may be shown or hidden.

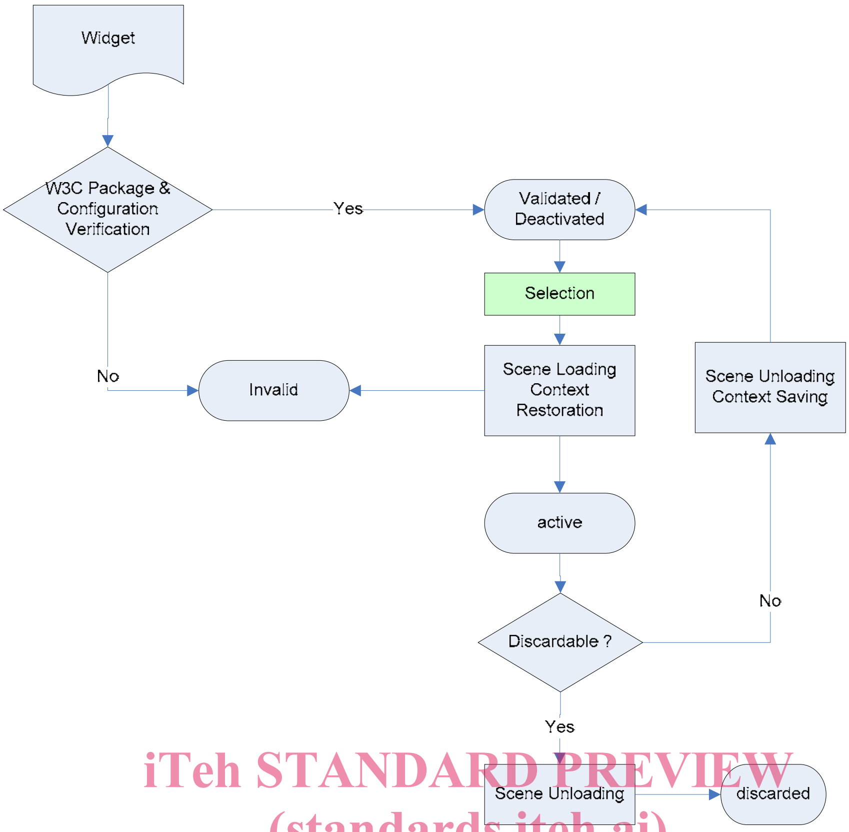
Widget
W3C Package & Configuration Verification
Yes
Validated / Deactivated
Selection
No
Invalid
Scene Loading Context Restoration
Scene Unloading Context Saving
active
No
Discardable ?
Yes
Scene Unloading
discarded

Figure 2 — MPEG-U widget life cycle

There are four events relevant to the life of each representation as specified in Table 1. These events may be used in particular to distinguish the substates of the active state of the widget life cycle when a single resource is used to describe both the full and simplified representations. These events are communicated to the widget using the mechanism described in 7.3.

Table 1 — Widget life cycle events

activateSimple	The simplified representation has been loaded and is now active and shown
activateFull	The full representation has been loaded and is now active and shown
showFull	The full representation was hidden and has been shown
showSimple	The simplified representation was hidden and has been shown
hideFull	The full representation was shown and has been hidden
hideSimple	The simplified representation was shown and has been hidden
deactivateFull	The full representation is going to be removed from the widget manager in the near future.
deactivateSimple	The simplified representation is going to be removed from the widget manager in the near future.

7.3 Widget communication

7.3.1 Overview

As regular scenes, widgets may communicate with any external entities using communications means supported by the presentation engine (e.g. Web servers and XMLHttpRequest object [2]; or streaming server and the RTSP protocol [3]). However, the widget manager offers means for widgets to communicate with entities for which URL are not known at widget authoring time nor at widget delivery time but when the widget enters the active state. This is the case of widgets communicating with devices in a home network (e.g. UPnP media server [4]) or the case of widgets communicating with local resources (e.g. battery status) or the case of widgets communicating with other widgets running in the same widget manager. For that purpose, widget authors may use the `MPEGwidget` script interface defined in Clause 9 to determine the address of the linked external communication entity.

In this Part of ISO/IEC 23007, we assume that the type of each communication entity (external or internal), or of its services (remote or local), can be identified by a unique string (e.g. using URN) and that a description of the communication capabilities of that entity is available. This description lists the messages that the entity can emit or receive. Each message has a name and a list of parameters. Each parameter carries a value and is identified by a name.

7.3.2 Matching services and interfaces

The widget manager shall match the available services with the interfaces of the activated widgets. This matching process is normative and may happen at any time, possibly at multiple times, when the widget is active. The matching process is as follows.

NOTE The widget manager is responsible for discovering available services which may come from local or remote communication entities, like devices or widgets. Services may become available or unavailable at any time.

First, the widget manager shall check the type of the widget interface and compare it with the string identifying the type of a service (e.g. the URN of the UPnP service). The exact comparison algorithm is service specific.

Then, if the types match, the widget manager shall compare the definition of the interface with the description of the service capabilities (e.g. UPnP actions and events) at the message level. For each message in the widget interface, there shall be a message in the service description with the same name. If this is not the case, the matching process fails.

Finally, for each message, the widget manager shall check:

- If, for each input in the widget message declaration, there is an output parameter in the service description, with the same name. If this fails, the widget manager shall check that there is a default value for this input. If this is not the case, the matching process fails. If after processing all input in the widget message declaration, some output parameters in the service description are not matched, the widget manager shall proceed with the assumption that these output parameters are not used in the widget.

- If for each output in the widget message declaration, there is an input parameter in the service description, with the same name. If this fails, the matching process fails.

In case of success of the matching process, the widget manager shall check that nodes or functions referenced in the widget interface declaration (see 10.23) are present in the widget representation when the matching process is performed. If this is not the case, the matching fails.

If the matching is successful, the interface is considered bound and the widget manager shall notify it to the widget either using the `mw:bindAction` attribute (see 10.15) or using the `onInterfaceBind` callback in the API (see 9.2). In case of failure of the matching, the widget manager shall not bind the interface.