ETSI TS 102 822-3-4 V1.1.1 (2006-01)

Technical Specification

Broadcast and On-line Services: Search, select, and rightful use of content on personal storage systems ("TV-Anytime"); Part 3: Metadata; Sub-part 4: Phase 2 - Interstitial metadata

European Broadcasting Union   Union Européenne de Radio-Télévision

EBU·UER

Reference
DTS/JTC-TVA-PH1-16-03-04

Keywords
broadcasting, content, data, TV, video

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00   Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

Individual copies of the present document can be downloaded from:
http://www.etsi.org

The present document may be made available in more than one electronic version or in print. In any case of existing or
perceived difference in contents between such versions, the reference version is the Portable Document Format (PDF).
In case of dispute, the reference shall be the printing on ETSI printers of the PDF version kept on a specific network drive
within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status.
Information on the current status of this and other ETSI documents is available at
http://portal.etsi.org/tb/status/status.asp

If you find errors in the present document, please send your comment to one of the following services:
http://portal.etsi.org/chaircor/ETSI_support.asp

Copyright Notification

ETSI

Contents

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://webapp.etsi.org/IPR/home.asp).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Technical Specification (TS) has been produced by Joint Technical Committee (JTC) Broadcast of the European Broadcasting Union (EBU), Comité Européen de Normalisation ELECtrotechnique (CENELEC) and the European Telecommunications Standards Institute (ETSI).

NOTE: The EBU/ETSI JTC Broadcast was established in 1990 to co-ordinate the drafting of standards in the specific field of broadcasting and related fields. Since 1995 the JTC Broadcast became a tripartite body by including in the Memorandum of Understanding also CENELEC, which is responsible for the standardization of radio and television receivers. The EBU is a professional association of broadcasting organizations whose work includes the co-ordination of its members' activities in the technical, legal, programme-making and programme-exchange domains. The EBU has active members in about 60 countries in the European broadcasting area; its headquarters is in Geneva.

European Broadcasting Union
CH-1218 GRAND SACONNEX (Geneva)
Switzerland
Tel: +41 22 717 21 11
Fax: +41 22 717 24 81

The present document is part 3, sub-part 4 of a multi-part deliverable covering Broadcast and On-line Services: Search, select and rightful use of content on personal storage systems ("*TV-Anytime*"), as identified below:

Part 1: "Benchmark Features";

Part 2: "System description";

Part 3: "Metadata";

Sub-part 1: "Phase 1 - Metadata schemas";

Sub-part 2: "System aspects in a uni-directional environment";

Sub-part 3: "Phase 2 - Extended Metadata Schema";

Sub-part 4: "Phase 2 - Interstitial metadata";

Part 4: "Content referencing";

Part 5: "Rights Management and Protection (RMP)";

Part 6: "Delivery of metadata over a bi-directional network";

Part 7: "Bi-directional metadata delivery protection";

Part 8: "Phase 2 - Interchange Data Format";

Part 9: "Phase 2 - Remote Programming".

Introduction

The present document is based on a submission by the *TV-Anytime* forum (http://www.TV-Anytime.org).

"TV-Anytime" (TVA) is a synchronized set of specifications established by the *TV-Anytime* Forum. TVA features enable the search, selection, acquisition and rightful use of content on local and/or remote personal storage systems from both broadcast and online services.

TS 102 822-1 [1] and TS 102 822-2 [2] set the context and system architecture in which the standards for Metadata, Content referencing, Bi-directional metadata and Metadata protection are to be implemented in the *TV-Anytime* environment. TS 102 822-1 [1] provides benchmark business models against which the *TV-Anytime* system architecture is evaluated to ensure that the specification enable key business applications. TS 102 822-2 [2] presents the *TV-Anytime* System Architecture. These two documents are placed ahead of the others for their obvious introductory value. Note that these first two documents are largely informative, while the remainder of the series is normative.

The features are supported and enabled by the specifications for Metadata (TS 102 822-3-1 [3], TS 102 822-3-2 [4], TS 102 822-3-3 [5] and TS 102 822-3-4), Content Referencing (TS 102 822-4 [6]), Rights Management (TS 102 822-5-1 [7] and TS 102 822-5-2 [8]), Bi-directional Metadata Delivery (TS 102 822-6-1 [9], TS 102 822-6-2 [10] and TS 102 822-6-3 [11]) and Protection (TS 102 822-7 [12]), Interchange Data Format (TS 102 822-8 [13]) and Remote Programming (TS 102 822-9 [14]). This list of Features is to be used as guidance to manufacturers, service providers and content providers regarding the implementation of the Phase 1 and Phase 2 *TV-Anytime* specifications.

Although each in the series of documents is intended to stand alone, a complete and coherent sense of the *TV-Anytime* system standard can be gathered by reading all the specification documents in numerical order.

TV-Anytime Phase 1 Metadata specification address a data model that allowed a broadcaster to describe the content available within the broadcast system and to therefore "attract" a user to acquire and consume the content.

TV-Anytime Phase 2 specifications target more advanced concepts such as the ability to perform interstitial replacement at playback time based on a number of criteria. The criteria to be used for the control of what content should replace what, may be explicitly declared using the schemas defined within the present document.

The specification does not attempt to defines all the possible ways in which a broadcaster may wish to control their system, but provides a generic framework into which a broadcaster can define their own platform specific rules, which are used for interstitial replacement control.

1   Scope

The present document is one in a series of Technical Specification documents produced by the *TV-Anytime* Forum. These documents establish the fundamental specifications for the services, systems and devices that will conform to the *TV-Anytime* standard, to a level of detail that is implementable for compliant products and services.

TS 102 822-1 [1] and TS 102 822-2 [2] set the context and system architecture in which the standards for Metadata, Content referencing, Bi-directional metadata and Metadata protection are to be implemented in the *TV-Anytime* environment. TS 102 822-1 [1] provides benchmark business models against which the *TV-Anytime* system architecture is evaluated to ensure that the specification enable key business applications. TS 102 822-2 [2] presents the *TV-Anytime* System Architecture and the relationship between Phase 1 and Phase 2 technologies. These first two documents are largely informative, while the remainder of the series is normative.

The Phase 2 *TV-Anytime* Metadata schema is a backwards-compatible extension of the Phase 1 schema. It extends Phase 1 datatypes for content description and user description and makes use of imported datatypes from MPEG-21 to enable new areas of functionality. It also extends the *TV-Anytime* root document type, TVAMainType, to enable publication of metadata described using the new datatypes.

The present document has been developed during the second phase of TV-Anytime and deals with the metadata necessary to enable Interstitial replacement, within a PDR device.

The set of metadata described in the present document was selected in order to satisfy the usage scenarios listed in TS 102 822-1 [1]. The formal definitions of metadata schemas should be read in conjunction with the system specification (TS 102 822-2 [2]) defining how they could be used in an end-to-end system.

TV-Anytime only defines the format for metadata that may be exchanged between various entities such as between content providers and consumers, among consumers, or between a third-party metadata provider and consumers.

XML is the "representation format" used to define schemas in TV-Anytime specifications. Although XML Schema is used to define how metadata is represented in XML, it can also be used to describe equivalent, non-XML representations of the same metadata.

2   References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication and/or edition number or version number) or non-specific.

- For a specific reference, subsequent revisions do not apply.

- For a non-specific reference, the latest version applies.

Referenced documents which are not found to be publicly available in the expected location might be found at http://docbox.etsi.org/Reference.

[1]   ETSI TS 102 822-1: "Broadcast and On-line Services: Search, select, and rightful use of content on personal storage systems ("TV-Anytime"); Part 1: Benchmark Features".

[2]   ETSI TS 102 822-2: "Broadcast and On-line Services: Search, select, and rightful use of content on personal storage systems ("TV-Anytime"); Part 2: System description".

[3]   ETSI TS 102 822-3-1: "Broadcast and On-line Services: Search, select, and rightful use of content on personal storage systems ("TV-Anytime");
Part 3: Metadata; Sub-part 1: Phase 1 - Metadata schemas".

[4]   ETSI TS 102 822-3-2: "Broadcast and On-line Services: Search, select, and rightful use of content on personal storage systems ("TV-Anytime");
Part 3: Metadata; Sub-part 2: System aspects in a uni-directional environment".

[5]   ETSI TS 102 822-3-3: "Broadcast and On-line Services: Search, select, and rightful use of content on personal storage systems ("TV-Anytime"); Part 3: Metadata; Sub-part 3: Phase 2 - Extended Metadata Schema".

[6]   ETSI TS 102 822-4: "Broadcast and On-line Services: Search, select, and rightful use of content on personal storage systems ("TV-Anytime"); Part 4: Content referencing".

[7]   ETSI TS 102 822-5-1: "Broadcast and On-line Services: Search, select, and rightful use of content on personal storage systems ("TV-Anytime"); Part 5: Rights Management and Protection (RMP) Sub-part 1: Information for Broadcast Applications".

[8]   ETSI TS 102 822-5-2: "Broadcast and On-line Services: Search, select, and rightful use of content on personal storage systems ("TV-Anytime"); Part 5: Rights Management and Protection (RMP) Sub-part 2: RMPI binding".

[9]   ETSI TS 102 822-6-1: "Broadcast and On-line Services: Search, select, and rightful use of content on personal storage systems ("TV-Anytime "); Part 6: Delivery of metadata over a bi-directional network; Sub-part 1: Service and transport".

[10]   ETSI TS 102 822-6-2: "Broadcast and On-line Services: Search, select, and rightful use of content on personal storage systems ("TV-Anytime"); Part 6: Delivery of metadata over a bi-directional network; Sub-part 2: Phase 1 - Service discovery".

[11]   ETSI TS 102 822-6-3: "Broadcast and On-line Services: Search, select, and rightful use of content on personal storage systems ("TV-Anytime"); Part 6: Delivery of metadata over a bi-directional network; Sub-part 3: Phase 2 - Exchange of Personal Profile".

[12]   ETSI TS 102 822-7: "Broadcast and On-line Services: Search, select, and rightful use of content on personal storage systems ("TV-Anytime Phase 1"); Part 7: Bi-directional metadata delivery protection".

[13]   ETSI TS 102 822-8: "Broadcast and On-line Services: Search, select, and rightful use of content on personal storage systems ("TV-Anytime"); Part 8: Phase 2 - Interchange data format".

[14]   ETSI TS 102 822-9: "Broadcast and On-line Services: Search, select, and rightful use of content on personal storage systems ("TV-Anytime"); Part 9: Phase 2 - Remote Programming".

[15]   XML Schema, W3C Recommendations (version 20010502).

NOTE: Available at:
         http://www.w3.org/TR/2001/REC-xmlschema-0-20010502,
         http://www.w3.org/TR/2001/REC-xmlschema-1-20010502,
         http://www.w3.org/TR/2001/REC-xmlschema-2-20010502.

[16]   ISO/IEC 15938-5 (2003): "Information Technology - Multimedia content description interface - Part 5: Multimedia Description Schemes".

[17]   ISO/IEC 15938-2 (2002): "Information Technology - Multimedia content description interface - Part 2: Description Definition Language".

3   Definitions and abbreviations

3.1   Definitions

For the purposes of the present document, the following terms and definitions apply:

application: specific set of functions running on the PDR

NOTE: Some applications use metadata, either automatically or under consumer control.

attractor: metadata element that is accessible by the consumer in order to aid in the content selection process, thus attracting the consumer

    EXAMPLE: Title and name of an actor in a television programme.

avail: individual interstitial item

    EXAMPLE: A commercial break (also commonly known as a 'Spot').

    NOTE: This term is used in the United States. In the UK an avail is used to refer to an interstitial break.

content creator: producer of the content

content item: entity that can be acquired as a single unit

    EXAMPLE: AV file, Audio stream.

content package: collection of content items, which may be consumed as a whole or individually

content provider: entity that acts as the agent for and is the prime exploiter of the content

content reference: pointer to a specific content item

description scheme: a formal definition of a metadata schema written in the MPEG-7 Description Definition Language (DDL), see ISO/IEC 15938-2.

descriptor: metadata element, such as an attractor or other information about content such as the key frame index of a piece of video

enhanced TV: television programme that includes additional information and/or applications related to content

    NOTE: Enhanced TV does not make use of a return path.

interactive TV: television programme that includes additional information and/or applications related to content and which takes advantage of a return path

interstitial: additional content that may be inserted within, at the start, or at the end of the primary content item

    NOTE: This additional content includes e.g. advertising spots, station idents, promos, graphics.

interstitial break: group of interstitials shown together

    NOTE: This term is used in the United Kingdom.

life Cycle: process of creation, usage, storage, and deletion of metadata

location Resolution: process of establishing the address (location and time) of a specific content instance from its CRID

metadata: data about content

    EXAMPLE: The title, genre and summary of a television programme.

    NOTE: In the context of *TV-Anytime*, metadata also includes consumer profile and history data.

metadata schema: identifier associated with a set of XML schemas that globally identifies those schemas so that they can be referenced externally

    NOTE: A globally unique namespace ensures that the names of types defined by schemas in that namespace do not conflict with types of the same name defined elsewhere.

metadata system: set of rules describing the syntax and semantics of metadata

MPEG-7: generic name of the ISO-IEC 15938 standard set of content-related metadata applicable to a broad range of applications

namespace: collection of components that allows the end-to-end operation of the *TV-Anytime* metadata solution

pod: set of avails or spots that form a commercial break

spot: individual content item within an interstitial break or "pod"

>   NOTE: Also commonly referred to as an Avail.

3.2   Abbreviations

For the purposes of the present document, the following abbreviations apply:

>   ARIB   Association of Radio Industries and Businesses
>
>   NOTE: Japanese Standardization Body.
>
>   ATSC   Advanced Television Systems Committee
>
>   NOTE: This committee is for the digital broadcast of multimedia and applications in North America and other
>   regions.
>
>   CRID   Content Reference IDentifier
>
>   NOTE: An identifier for content that is independent of its location specified by TS 102 822-4.
>
>   DDL   Description Definition Language
>   DVB   Digital Video Broadcasting project
>
>   NOTE: European project providing technical specifications for publication by ETSI.
>
>   IPR   Intellectual Property Rights
>   ISO/IEC JTC1   ISO/IEC Joint Technical Committee 1
>
>   NOTE: A joint effort of the ISO (International Organization for standardization) and IEC (International
>   Electrotechnical Commission).
>
>   PDR   Personal Digital Recorder
>   XML   eXtensible Markup Language

4   Use of CRID

As is the case with the *TV-Anytime* Phase 1 specifications, the CRID (TS 102 822-4 [6]) forms a key role within the present document. It is used to identify an individual interstitial item i.e. a content item. Still as is the case in Phase 1, a CRID shall be assigned to each unique piece of content. The present document does not try to define what a "unique piece of content" is, e.g. using globally unique identifiers, but assumes that an instance of a content item having the same CRID, may be treated the same.

5   Identification of Pods

Many of the business models supported by the data model defined below require that a "pod" has a unique identifier. The present document assumes that this identifier takes the form of an URI, but the detailed format of this URI is not defined in the present document.

No mechanism is provided by *TV-Anytime* for signalling the identification of a "Pod" within the broadcast stream. It is assumed that this is transport dependent, and should therefore be defined by other standards bodies such as DVB, ATSC or ARIB. For signalling the identification of a "pod" within downloaded content where no specific interstitial contents are pre-inserted, SegmentGroupInformation with GroupType of value "insertionPoints" defined in the TS 102 822-3-1 [3] may be used. In this case, the identifier of a "pod" takes the form of "CRID". Further explanation of the use of TVA metadata for the identification of "pod" insertion points is given in TS 102 822-2 [2].

6   Targeting Controlled Terms and Rules

The present document provides a framework for the broadcaster to control the playback/replacement of interstitial content. Within current day interstitial scheduling systems extensive use of "rules" are used to automatically select where a "spot" should be placed within a "pod" relative to other content and spots. The rules used may change over time, due to, a number of factors. For example:

- The introduction of new legislation.

- To support new business models.

These rules may also be dependent on the geographically region in which the system is deployed. Due to the dynamic nature and diversity of these rules, *TV-Anytime* has chosen to specify a well-defined data model (schema) for describing both simple and complex rules in a generic form.

The solution enables a broadcaster to define a set of "TargetingControlledTerms", to be used within the system. Where each entry within the "TargetingControlledTerms" has the following fields:

- a URI that uniquely identifies the term;

- the data type of the data returned when the term is evaluated. e.g. string, Integer, double, enumeration etc;

- an optional field which defines the data type of the input parameter.

These "TargetingControlledTerms" are used within a "rule" declaration to:

- inform the terminal of the context of the rule;

- and the data format of the rules input parameter, which is required to enable:

    - the efficient representation of the MethodParameter element in a binary form;

    - the correct evaluation of the rule based on the test directive of the rule i.e. "less_than", "greater_than";

    - the data format of the rules return value. This is required to enable:

    - the correct evaluation of the rule based on the test directive of the rule i.e. "less_than", "greater_than";

- the efficient representation of the rules corresponding ConstantValue (if any),

- as well as validate the value declared within the "ConstantValue" element.

7 Metadata Definitions

For the purpose of interoperability, *TV-Anytime* has adopted XML as the common representation format for metadata. XML offers many advantages: it allows for extensibility, supports the separation of data from the application, and is widely used. XML schema is mainly used to represent the data model. *TV-Anytime* descriptions may however be instantiated in a format other than textual. TV-Anytime has described some of these mechanisms such as binary encoding such as defined in TS 102 822-3-2 [4].

7.1 Use of MPEG-7

A *metadata schema* is the formal definition of the structure and type of metadata. *TV-Anytime* uses the MPEG-7 Description Definition Language (DDL) [17] to describe metadata structure as well as the XML encoding of metadata. DDL is based on XML schema as recommended by W3C in [15].

TV-Anytime uses several MPEG-7 datatypes as collected in the MPEG7 stub attached to the present document.

7.2 TV-Anytime Metadata Namespace

TV-Anytime interstitial metadata description schemes are associated with the *TV-Anytime* interstitial metadata XML namespace, defined as:

```
xmlns:int="urn:tva:metadata:interstitial:2005"
```

TV-Anytime interstitial metadata includes description schemes defined by XML as included in the XML stubs attached to the present document.

```
xmlns="http://www.w3.org/2001/XMLSchema"

<import namespace="http://www.w3.org/XML/2001/namespace" schemaLocation="xml.xsd"/>
```

TV-Anytime also includes description schemes defined by MPEG-7 as included in the MPEG7 stub attached to the present document, which use the MPEG-7 namespace as described in [16].

```
xmlns:mpeg7="urn:tva:mpeg7: 2005"

<import namespace="urn:tva:mpeg7: 2005" schemaLocation="tva_mpeg7.xsd"/>
```
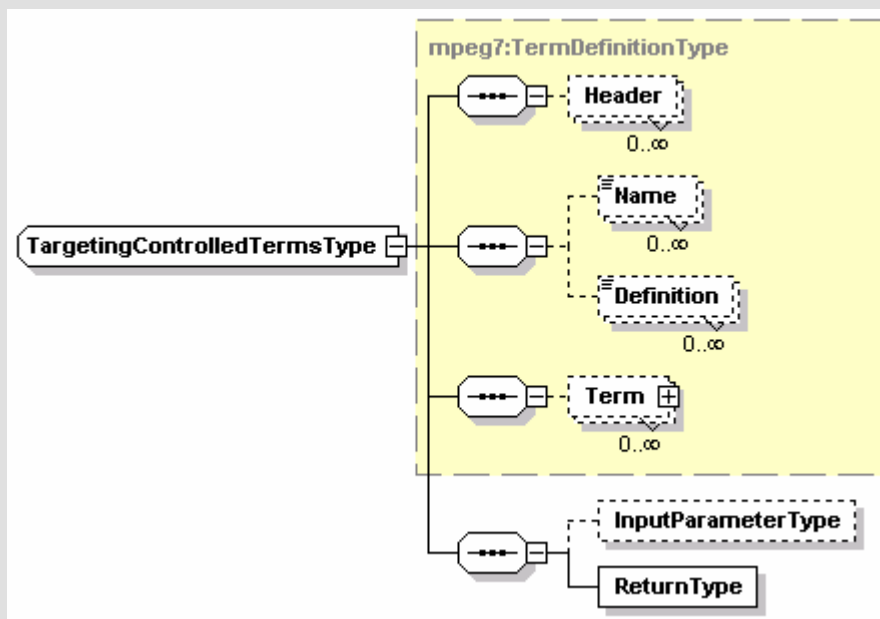
All TVA metadata documents using interstitial metadata must be fully namespace qualified and must declare the TVA interstitial metadata namespace.

7.3 TargetingControlledTerms

The following defines the syntax and semantics for the `TargetingControlledTerms`. These `TargetingControlledTerms` are used within the MPEG-7 `ClassificationSchemeType` to provide a set of terms for use within the `rule` complex type.

```
<complexType name="TermConstraintsType">
    <attribute name="simpleType" type="xsd:string" use="required"/>
    <attribute name="namespace" type="xsd:anyURI" use="optional"/>
</complexType>
```



```
<complexType name="TargetingControlledTermsType">
    <complexContent>
        <extension base="mpeg7:TermDefinitionType">
            <sequence>
                <element name="InputParameterType" type="int:TermConstraintsType"
                minOccurs="0"/>
                <element name="ReturnType" type="int:TermConstraintsType" />
            </sequence>
        </extension>
    </complexContent>
</complexType>
```

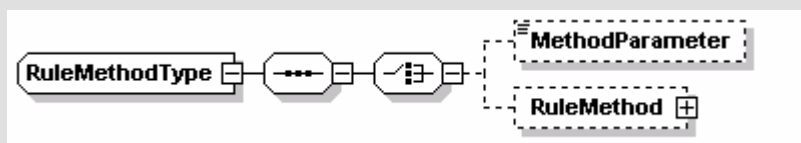| Name	Definition
TermConstraintsType	An complex type which is used to represent the data type associated with a Targeting Term.
simpleType	The datatype of the value associated with this term. This shall not have a namespace prefix.
namespace	A full declaration of the namespace that the simpleType is a member of. If this attribute is not present, it is assumed that the data type declared using the simpleType attribute, is a member of the XML schema namespace i.e. http://www.w3.org/2001/XMLSchema.
TargetingControlledTermsType	A complex type which declares a term.
Name	The name associated with the term.
Preferred	An optional attribute to indicate that the given controlled term is the preferred instance e.g. in a list.
Definition	A definition of the term.
termId	A string used to identify the term. This shall be unique within the scope of the set of ControlledTerms.
relation	Not used in the present document. If present, it shall be ignored by the terminal.
InputParameterType	An optional complex type that is used to signal the data type of the input parameter, used in the evaluation of the term, within a rule.
ReturnType	A complex type that is used to signal the data type returned when the term is evaluated.

The following snippet is an example instantiation of a set of Targeting Controlled terms.

```
<?xml version="1.0" encoding="UTF-8"?>

<tva:TVAMain xmlns="urn:tva:metadata:extended:2005"
xmlns:int="urn:tva:metadata:interstitial:2005" xmlns:mpeg7="urn:tva:mpeg7:2005"
xmlns:tva="urn:tva:metadata:2005" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:tva:metadata:extended:2005 tva2_metadata_3-3_v111.xsd"
xsi:type="ExtendedTVAMainType">
   <tva:ClassificationSchemeTable>
      <tva:ClassificationScheme uri="urn:Foo.com:Interstitial:ExampleTerms">
         <mpeg7:Term termID="SysTime" xsi:type="int:TargetingControlledTermsType">
            <mpeg7:Name>SystemTime</mpeg7:Name>
            <mpeg7:Definition>This signals to the terminal that it should retrieve
            the current System time</mpeg7:Definition>
            <int:ReturnType namespace="http://www.w3.org/2001/XMLSchema"
            simpleType="time"/>
         </mpeg7:Term>
         <mpeg7:Term termID="NumPlays" xsi:type="int:TargetingControlledTermsType">
            <mpeg7:Name>Number of Plays</mpeg7:Name>
            <mpeg7:Definition>This signals to the terminal that it should retrieve the
            Number of Times that a particular item of content has been
            played</mpeg7:Definition>
            <int:InputParameterType namespace="urn:tva:metadata:2005"
            simpleType="CRIDType"/>
            <int:ReturnType namespace="http://www.w3.org/2001/XMLSchema"
            simpleType="integer"/>
         </mpeg7:Term>
      </tva:ClassificationScheme>
   </tva:ClassificationSchemeTable>
</tva:TVAMain>
```
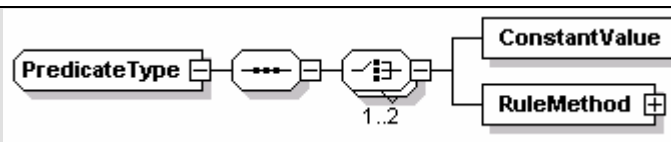
7.4 Expressing Rules

The ability to express rules by which interstitial content items are selected is a key part of the present document. A rule is composed of a set of Unary and Binary Predicates which can be combined together within a Predicate Bag using the AND and OR clauses. This is very much like the WHERE declaration within an SQL SELECT statement.

7.4.1 PredicateTypes


```
<complexType name="RuleMethodType">
   <sequence>
      <choice>
         <element name="MethodParameter" type="anySimpleType" minOccurs="0" maxOccurs="1"
            />
         <element name="RuleMethod" type="int:RuleMethodType" minOccurs="0" maxOccurs="1"
            />
      </choice>
   </sequence>
   <attribute name="methodName" type="mpeg7:termReferenceType" use="required"/>
</complexType>

<complexType name="ConstantValueType" >
   <attribute name="value" type="anySimpleType" use="required" />
</complexType>
```

```
<complexType name="PredicateType">
   <sequence>
      <choice minOccurs="1" maxOccurs="2">
         <element name="ConstantValue" type="int:ConstantValueType" minOccurs="1"
             maxOccurs="1" />
         <element name="RuleMethod" type="int:RuleMethodType" minOccurs="1" maxOccurs="1"
             />
      </choice>
   </sequence>
   <attribute name="negate" type="boolean" default="false"/>
   <attribute name="test" type="int:PredicateTestType" default="equals"/>
</complexType>

<simpleType name="PredicateTestType">
   <restriction base="string">
      <enumeration value="equals"/>
      <enumeration value="not_equals"/>
     <enumeration value="contains"/>
      <enumeration value="greater_than"/>
      <enumeration value="greater_than_or_equals"/>
      <enumeration value="less_than"/>
      <enumeration value="less_than_or_equals"/>
      <enumeration value="exists"/>
   </restriction>
</simpleType>
```

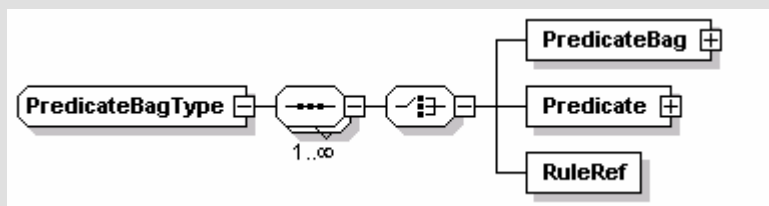| Name	Definition
RuleMethodType	A complex tType that is used to represent a method that when called will return a data value that is then used within the Predicate.
MethodParameter	An optional simple type which declares the input parameter into the Method to be evaluated. The simple type should match that declared in the classification scheme.
RuleMethod	An optional method which shall be evaluated, where the return value of the method, forms the input into this method.
methodName	A required attribute that defines the method being evaluated (e.g. System time, Number of plays, etc.). This attribute identifies a targeting controlled term, which has been previously defined.
ConstantValueType	A complex type that is used to represent a constant value against which a rule is being tested.
value	A required attribute that defines the value being tested.
PredicateType	A complex type that represents a boolean test that is to be evaluated. The type of evaluation performed depends on the corresponding RuleMethod element.
ConstantValue	An element which is used to declare a Constant value against which the rule is being tested.
RuleMethod	An element which is used to represent a method, where the results of performing this method are used within the test of the Predicate.
negate	This attribute reverses the Boolean evaluation (true/false) of the Predicate
test	The test being performed between a RuleMethod and a ConstantValue or another RuleMethod. This attribute can take the value "equals", "not_equals", "contains", "greater_than", "greater_than_or_equal", "less_than", "less_than_or_equals" or "exists".

7.4.2   PredicateBagType

```
<complexType name="RuleRefType">
   <attribute name="ref" type="tva:TVAIDRefType" use="required"/>
</complexType>


<simpleType name="PredicateBagTypeType">
   <restriction base="string">
     <enumeration value="AND"/>
     <enumeration value="OR"/>
   </restriction>
</simpleType>
```



```
<complexType name="PredicateBagType">
   <sequence maxOccurs="unbounded">
     <choice>
        <element name="PredicateBag" type="int:PredicateBagType"/>
        <element name="Predicate" type="int:PredicateType"/>
        <element name="RuleRef" type="int:RuleRefType"/>
     </choice>
   </sequence>
   <attribute name="negate" type="boolean" default="false"/>
   <attribute name="type" type="int:PredicateBagTypeType"/>
</complexType>
```

Name	Definition
RuleRefType	A complex type which provides the ability to reference a previously declared rule.
ref	The value of the id attribute of rule which is being referenced.
PredicateBagType	A PredicateBagType contains one or more Predicate, or PredicateBag children, and is used to express logical relationships between these children.
PredicateBag	A PredicateBag containing one or more Predicate, PredicateBag or PredicateRef children, and is used to express logical relationships between these children.
Predicate	A boolean test that can be evaluated.
RuleRef	A reference to a previous declared Rule which is to be evaluated within the context of this PredicateBag.
negate	This attribute reverses the Boolean evaluation (true/false) of the PredicateBag. If there are two or more children predicates, it is evaluated after the children predicates have been combined; i.e. the type attribute has tighter precedence.
type	This attribute expresses the logical relationship between the children of this PredicateBag. It can take the values "AND" or "OR". The attribute is required when there are two or more children. The attribute is meaningless if the PredicateBag contains a single element.

The following is an example of use of a PredicateBag. The example instance snippet, first, checks to see if the "spot" "crid://foo.com/persil" is contained within the "pod" (this behaviour is expressed by the methodName). If this evaluates to "true" then the terminal will next check to see how many times the "spot" "crid://foo.com/persil" has been played. If it has been played more than 3 times, the rule evaluates to "true".

```
<rule rule_id="example1">
   <PredicateBag type="AND" >
      <Predicate test="exists">
        <RuleMethod methodName="urn:foo:Interstitials:SpotInPod">
           <MethodParameter>crid://foo.com/soap</MethodParameter>
        </RuleMethod>
      </Predicate>
      <Predicate test="greater_than">
        <RuleMethod methodName="urn:foo:Interstitials:NoOfPlays">
           <MethodParameter>crid://foo.com/soap</MethodParameter>
        </RuleMethod>
        <ConstantValue value="3" />
      </Predicate>
   </PredicateBag>
</rule>
```

7.4.3   RuleType

```
<complexType name="OtherRuleLanguageType">
   <attribute name="ruleLanguage" type="mpeg7:termReferenceType" use="required" />
   <attribute name="href" type="anyURI" use="required"/>
</complexType>
```



```
<complexType name="RuleType">
   <sequence>
      <choice minOccurs="0" maxOccurs="unbounded">
         <element name="PredicateBag" type="int:PredicateBagType"/>
         <element name="Predicate" type="int:PredicateType"/>
         <element name="RuleRef" type="int:RuleRefType"/>
      </choice>
      <element name="OtherRuleLanguage" type="int:OtherRuleLanguageType" minOccurs="0"
      maxOccurs="unbounded"/>
   </sequence>
   <attribute name="rule_id" type="tva:TVAIDType" use="optional"/>
   <attributeGroup ref="tva:fragmentIdentification"/>
</complexType>

<complexType name="RulesType">
   <sequence>
      <element name="rule" type="int:RuleType" maxOccurs="unbounded"/>
   </sequence>
</complexType>
```
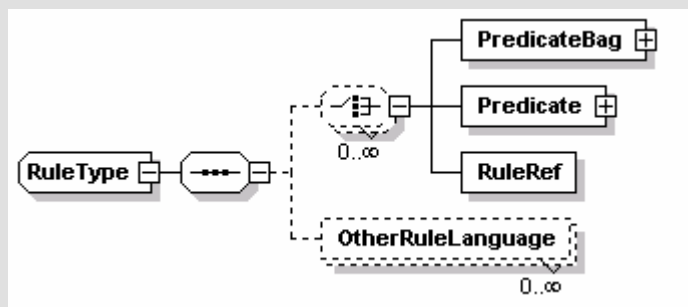
Name	Definition
OtherRuleLanguageType	A Complex Type that enables alternative rule expression languages to be used to express interstitial replacement logic.
ruleLanguage	A `termReferenceType` that signals the rule expression language used.
href	A URI which signals where the rule can be found.
RuleType	A complex type which encapsulates a single `Rule`, by means of one or more logical predicates. All children of this type shall be evaluated as if it was a single rule.
PredicateBag	A `PredicateBag` containing one or more `Predicate`, `PredicateBag` or `PredicateRef` children, and is used to express logical relationships between these children.
Predicate	A Boolean test that can be evaluated.
RuleRef	A reference to a previously declared rule.
OtherRuleLanguage	An element that enables the expression of the rule in an alternative language.
rule_id	An optional identifier for the rule. This id can be referenced by other rules and also by a `Condition` element to enable the inclusion of the referenced rules logic.
fragmentIdentification	An attributeGroup used to identify a metadata fragment.
RulesType	A complex type that encapsulates a number of rules.
rule	An instantiation of a rule.

7.5 Interstitial Framework

The interstitial framework enables an operator to describe the conditions under which interstitial replacement should take place and what replacements should take place.

7.5.1 Core Schema Types

7.5.1.1 ConditionType

The ConditionType is used extensively within the present document to define conditions that must be met, before a statement is "true". For example it is used to define what rules must be met before a particular substitution can take place.

A `Condition` element represents a condition. As such, it denotes the parent element as being conditional on a set of predicate tests. The `require` attribute lists the set of predicates that must become true, and the `except` attribute lists the set of predicates that must become "false", in order for the condition to be satisfied. Each predicate is identified by the value of:

- The `rule_id` attribute in a rule element.

A set of condition elements defines a boolean combination of predicate tests. Multiple tests within a condition are combined as a conjunction (an AND relationship). Multiple `Condition` elements within a given parent are combined as a disjunction (an OR relationship).

Validation Rules:

Each id value specified in the require and except attributes must match a rule_id attribute value defined in a rule element, located within the instance document.

```
<complexType name="ConditionType">
     <attribute name="require" type="tva:TVAIDRefsType"/>
     <attribute name="except" type="tva:TVAIDRefsType"/>
</complexType>
```

Name	Definition
ConditionType	A complex type that describes a condition that must be met, before it evaluates to "true".
require	An optional attribute which contains a list of rule_id references. All referenced predicates must evaluate to true for the condition to be equal to "true".
except	An optional attribute which contains a list of rule_id references. All references must evaluate to "false" for the condition to be equal to "true".

7.5.1.2   ContentRefType
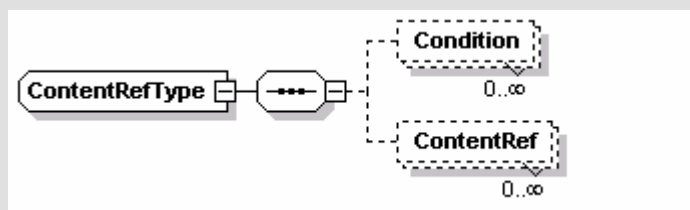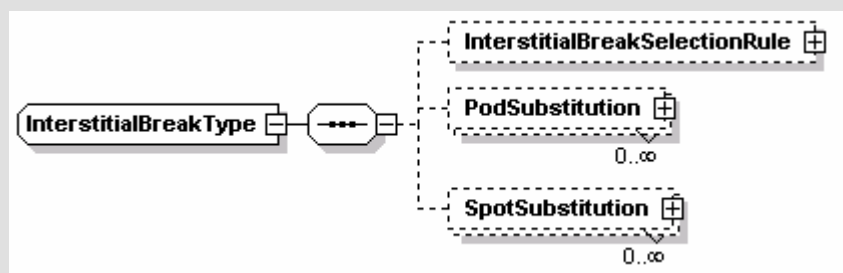


```
<complexType name="ContentRefType">
   <sequence>
     <element name="Condition" type="int:ConditionType" minOccurs="0"
     maxOccurs="unbounded"/>
     <element name="ContentRef" type="tva:CRIDRefType" minOccurs="0"
     maxOccurs="unbounded" />
   </sequence>
</complexType>
```

Name	Definition
ContentRefType	A complex type that represents a Content Selection, where its associated Conditions must be met before the Content is selected.
Condition	The element represents the Conditions under which the content identified within the ContentRef element are selected. Multiple Condition elements are combined as disjunction (an OR relationship).
ContentRef	A leaf CRID that identifies a Content item that should be acquired for later consumption or to form a substitution (defending on the context).

7.6 Interstitial Break and cascading

7.6.1 Interstitial Break Type


```
<complexType name="InterstitialBreakType">
   <sequence>
      <element name="InterstitialBreakSelectionRule" type="int:RuleType" minOccurs="0"
         maxOccurs="1"/>
      <element name="PodSubstitution" type="int:PodSubstitutionType" minOccurs="0"
         maxOccurs="unbounded"/>
      <element name="SpotSubstitution" type="int:SpotSubstitutionType" minOccurs="0"
         maxOccurs="unbounded"/>
   </sequence>
   <attributeGroup ref="tva:fragmentIdentification"/>
</complexType>
```

Name	Definition
InterstitialBreakType	A complex type that represents an interstitial break i.e. a "pod".
InterstitialBreakSelectionRule	Zero or more rules that are used to determine if PodSubstitution, and/or SpotSubstitution are valid within this interstitialBreak. The absence of any rule signals that this possible substitution is relevant for any interstitial break. For example this could be used to only allow interstitial break manipulation if the recorded content in which the break is contained within, has been resident on the terminal for more than "N" days.
PodSubstitution	An element that expresses that the complete interstitial break or "pod" should be replaced with that evaluated within this element.
SpotSubstitution	An element that allows the definition of what spots should be replaced by what depending on a set of rules.
fragmentIdentification	An attributeGroup used to identify a metadata fragment.

7.6.2 Substitution Cascading

With the framework defined within the present document it is possible to cascade substitutions. For example the conditions declared may cause a "pod" substitution to occur. It is then possible via one or more SpotSubstitution(s) to change the "spots" that will form the final "pod" substitution.

The following snippet is an example of cascade substitutions.

```
<InterstitialTable>
   <InterstitialBreak>
      <InterstitialBreakSelectionRule>
         <Predicate test="equals">
            <ConstantValue value="123"/>
            <RuleMethod methodName="urn:foo:Interstitials:targetPodID"/>
         </Predicate>
      </InterstitialBreakSelectionRule>
      <PodSubstitution>
```

```
            <ReplacementPod>
              <Condition require="PeakTime"/>
              <Spot crid="crid://foo.com/soap"/>
              <Spot crid="crid://foo.com/car"/>
            </ReplacementPod>
            <ReplacementPod>
              <Condition require="BreakfastTime"/>
              <Spot crid="crid://foo.com/HealthyBreakfast"/>
              <Spot crid="crid://foo.com/Cereal"/>
            </ReplacementPod>
            <ReplacementPod>
              <Condition require="PremiumSubscriber"/>
            </ReplacementPod>
            <ReplacementPod podId="89658">
              <Condition except="PeakTime BreakfastTime PremiumSubscriber" />
              <Spot crid="crid://foo.com/JoesCarLot"/>
              <Spot crid="crid://foo.com/AmateurDramatics"/>
            </ReplacementPod>
        </PodSubstitution>
        <SpotSubstitution>
          <SpotSelectionRule>
            <PredicateBag>
               <Predicate ruleTerm="urn:foo:Interstitials:SpotId"
               ruleValue="crid://foo.com/JoesCarLot" test="equals"/>
            </PredicateBag>
          </SpotSelectionRule>
          <ReplacementSpot>
            <Condition require="ParentalControl" />
            <ContentRef crid="crid://foo.com/GamesPromotion"/>
          </ReplacementSpot>
          <ReplacementSpot>
            <Condition require="EasterTime" />
            <ContentRef crid="crid://foo.com/ChocolatePromotion"/>
          </ReplacementSpot>
        </SpotSubstitution>
      </InterstitialBreak>
    </InterstitialTable>
</int:TargetingTable>
```

In this example, first of all the `PodSubstitution` declarations will be processed and will result in the original "pod" being replaced with a new one. e.g. the "pod" with a podId of 89658 is selected as the substitution.

Following the `PodSubstitution` declarations occurs a `SpotSubstitution` declaration which has a `SpotSelectionRule` which says that these following possible substitutions are valid if the "pod" contains a "spot" with a CRID equal to "crid://foo.com/JoesCarLot". Because the system previously selected the `ReplacementPod` with a podId of 89658 (which has a "spot" with a CRID equal to "crid://foo.com/JoesCarLot") the subsequent `ReplacementSpot` options are valid.

It is recommended that "substitutions" are validated in the order in which they are declared. Therefore, as soon as a "PodSubstitution" has been found that meets the set of declared rules, the system shall select this "Pod". The system shall continue to evaluate the remaining "PodSubstitution" but these are now evaluated against the newly selected "PodSubstitution". In the case of "PodSubstitutions" the system shall evaluate all entries, building up the final playout list as it goes through the set of possible "PodSubstitutions". For example, if a "Spot" is replaced, this replacement may subsequently be replaced itself by another "Spot".
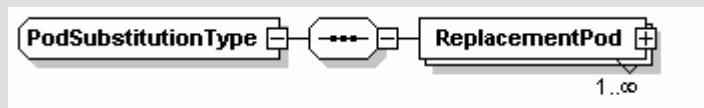
7.7 PodSubstitution Type
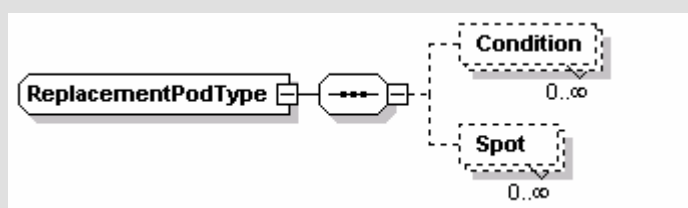
```
<complexType name="ReplacementPodType">
   <sequence>
     <element name="Condition" type="int:ConditionType" minOccurs="0"
         maxOccurs="unbounded" />
     <element name="Spot" type="tva:CRIDRefType" minOccurs="0" maxOccurs="unbounded"/>
   </sequence>
   <attribute name="podId" type="tva:TVAIDType"/>
</complexType>
```



```
<complexType name="PodSubstitutionType">
   <sequence>
     <element name="ReplacementPod" type="int:ReplacementPodType"
         maxOccurs="unbounded"/>
   </sequence>
</complexType>
```

Name	Definition
ReplacementPodType	A Complex type that enables the instantiation of a Pod.
Condition	The element represents the Conditions under which the ReplacementPod should be selected. Multiple Condition elements are combined as disjunction (an OR relationship).
Spot	An instantiation of a "spot". The "spots" shall be played out in increasing child order i.e. First instantiation will be played first, followed by the next instantiation. This element is of type CRIDRefType for which the semantics can be found in TS 102 822-3-1 [3].
podId	An optional attribute which uniquely identifies the Pod. This can then be used later on within a SpotSubstitution to replace specific "spots" within the ReplacementPod.
PodSubstitutionType	A complex type that describes the conditions under which a "pod" substitution should take place and the "pod" that should be substituted.
ReplacementPod	An element that describes a "pod". It has a number of child Spot elements that define the content that should be played and the order in which they are to be played. It is possible for there to be '0' Spot elements, which is used to signal that the original "pod" should be skipped.

In the situation where more than one ReplacementPod entries meet the substitution criteria i.e. the rules identified within the require, except attributes evaluate to "true", the behaviour is implementation dependent. It is recommended that this situation be avoided, if possible.

The following is an example of how PodSubstitution can be used to substitute the original broadcast "pod" with one of three alternatives. The "pod" that is substituted is selected based upon a number of previously declared rules, which are not included in the snippet to aid clarity).

A receiver would parse the document and shall at first evaluate the InterstitialBreakSelectionRule to determine if this PodSubstitution declaration is valid for this "pod". In this example a "pod" substitution shall only take place if the "id" of the "pod" in the original content is equal to "1234". If it is, then the system shall make a single selection from the set of possible ReplacementPod declarations, but evaluating the ReplacementPods conditions.

In the example if "PremiumSubscriber" evaluates to "true", the system shall Skip playout of the original "pod" and replace it with nothing, as no "spot" elements are instantiated.
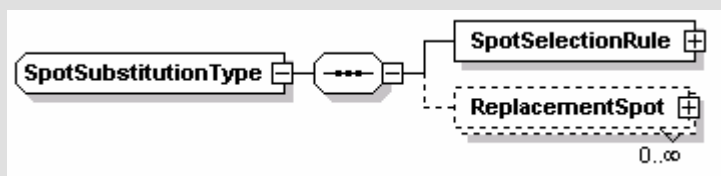
If no condition evaluated to "true", the last "pod" will be selected, as it has a `Condition` which means that it is chosen if neither the "PeakTimePod" "BreakfastTimePod" and "PremiumSubscriber" are selected.

```
<InterstitialBreak>
   <InterstitialBreakSelectionRule>
      <Predicate test="equals">
         <ConstantValue value="123" />
         <RuleMethod methodName="urn:foo:Interstitials:targetPodID" />
      </Predicate>
   </InterstitialBreakSelectionRule>
   <PodSubstitution>
      <ReplacementPod>
         <Condition require="PeakTime" />
         <Spot crid="crid://foo.com/soap"/>
         <Spot crid="crid://foo.com/car"/>
      </ReplacementPod>
      <ReplacementPod>
         <Condition require="BreakfastTime" />
         <Spot crid="crid://foo.com/HealthyBreakfast"/>
         <Spot crid="crid://foo.com/Cereal"/>
      </ReplacementPod>
      <ReplacementPod>
         <Condition require="PremiumSubscriber" />
      </ReplacementPod>
      <ReplacementPod>
         <Condition except="PeakTime BreakfastTime PremiumSubscriber" />
         <Spot crid="crid://foo.com/JoesCarLot"/>
         <Spot crid="crid://foo.com/AmateurDramatics"/>
      </ReplacementPod>
   </PodSubstitution>
</InterstitialBreak>
```

7.8   SpotSubstitutionType


```
<complexType name="SpotSubstitutionType">
   <sequence>
      <element name="SpotSelectionRule" type="int:RuleType" minOccurs="1"/>
      <element name="ReplacementSpot" type="int:ContentRefType" minOccurs="0"
         maxOccurs="unbounded"/>
   </sequence>
</complexType>
```

Name	Definition
SpotSubstitutionType	A complex type that describes the conditions under which a "spot" substitution takes place and the "spot" that should be substituted in its place.
SpotSelectionRule	A rule that is used to select which "spot" within the identified "pod" should be substituted. For example this could be used to see if a particular content item exists within the "pod", or it could just see if a content item from a particular promotion is present within Pod.
ReplacementSpot	An element that identifies the "spot" that will form the substitute. The ReplacementSpot selected will be dependent on the evaluation of the identified rule(s) that must be asserted for this condition to evaluate to "true". The absence of a ContentRef element within the ReplacementSpot element indicates that the "spot" should be skipped.

The following example of SpotSubstitution that substitutes the "spot" having a content item identified with a CRID "crid://foo.com/Bread", with a "Carpet World" promotion if viewing is undertaken during the promotional period, or a "Carpet World end of sale" promotion if viewing is undertaken near the end of the promotional period. If it is viewed outside the promotional period then the original "spot" is not substituted, and so the original "spot" (within the recorded content) shall be played.

```
<InterstitialTable>
   <InterstitialBreak>
      <InterstitialBreakSelectionRule>
         <Predicate test="equals">
            <ConstantValue value="1234" />
            <RuleMethod methodName="urn:foo:Interstitials:targetPodID" />
         </Predicate>
      </InterstitialBreakSelectionRule>
      <SpotSubstitution>
         <SpotSelectionRule>
            <PredicateBag>
               <Predicate test="equals">
                  <ConstantValue value="crid://foo.com/Bread"/>
                  <RuleMethod methodName="urn:foo:Interstitials:SpotID"/>
               </Predicate>
            </PredicateBag>
         </SpotSelectionRule>
         <ReplacementSpot>
            <Condition require="CarpetWorldPromotion">
            <ContentRef crid="crid://foo.com/CarpetWorld" />
         </ReplacementSpot>
         <ReplacementSpot>
            <Condition require="CarpetWorldPromotionEndofSale">
            <ContentRef crid="crid://foo.com/CarpetWorldEndOfSale" />
         </ReplacementSpot>
      </SpotSubstitution>
   </InterstitialBreak>
</InterstitialTable>
```

8 Recording Cache

To enable interstitial replacement to take place the receiver must have the appropriate content items available to enable the substitution to take place. These content items may for example be automatically pushed to the device during a quiet period in the broadcast schedule, or they may be available for OnDemand download by the receiver.

In some instances it may be desirable for the terminal to select which content items it should acquire based on a set of rules. For example, a broadcaster may make available a content item which requires a particular set of functionality within the receiver to view. It is therefore only appropriate for receivers having this capability to acquire the content item.

Overtime the interstitial content cache will consume a large amount of storage, and so it is important for the service provider to be able to expire content from a receiver.

It is also a requirement to be able to pull (stop playout) a content item for other reasons, like it has caused offence to a large percentage of the viewers and so it has decided to stop airing the content.

The following schema types enable this functionality.

```
<complexType name="ContentListType">
   <sequence>
      <element name="Item" type="int:ContentRefType" minOccurs="1"
      maxOccurs="unbounded"/>
   </sequence>
   <attributeGroup ref="tva:fragmentIdentification"/>
</complexType>
```
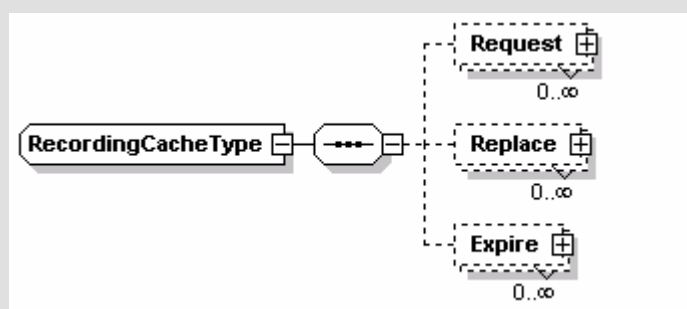


```
<complexType name="RecordingCacheType">
   <sequence>
      <element name="Request" type="int:ContentListType" minOccurs="0"
      maxOccurs="unbounded"/>
      <element name="Replace" type="int:ContentListType" minOccurs="0"
      maxOccurs="unbounded"/>
      <element name="Expire" type="int:ContentListType" minOccurs="0"
      maxOccurs="unbounded"/>
   </sequence>
</complexType>
```

Name	Definition
ConditionListType	A complex type that is used to instantiate a list of content items, where each content item has an optional require and/or except attribute to specify a condition that must be met for the item to be selected.
Item	An interstitial content item, which may have a optional require and/or except attribute to specify a condition that must be met for the item to be selected.
RecordingCacheType	A complex type that is used to inform the receiver about what content to: - request for acquisition - replace with a new version - expire from the system
Request	A list of content items that the receiver should acquire for use, as potential "spot" substitutions.
Replace	A list of content items that the receiver should reacquire as the content has changed.
Expire	A list of content items that the receiver shall remove from its content cache, as they are no longer needed i.e. The advertising campaign has finished.

The following example is an instantiation of the RecordingCache functionality. It shows how to instruct the receiver to acquire the "Carpet World" content items if the "Carpet World" promotion is currently active. It also instructs the receiver to unconditionally acquire the *TV-Anytime* promotion.

One of the "Carpet World" content items has had to be changed, which is relayed to the terminal using the `Replace` request.

In addition, this recording cache instance also defined when the "Carpet World" content items should be removed from the system cache, as they are no longer required.

```
<TargetingTable ..... >
   <RulesTable>
      <rule rule_id="CarpetWorldPromotion">
         <Predicate test="less_than">
            <RuleMethod methodName="urn:foo:Interstitials:SystemDateTime"/>
            <ConstantValue value="2004-09-01T00:00:00Z"/>
         </Predicate>
      </rule>
      <rule rule_id="CarpetWorldPromotionEndofSale">
         <Predicate test="less_than">
            <RuleMethod methodName="urn:foo:Interstitials:SystemDateTime"/>
            <ConstantValue value="2004-10-01T00:00:00Z" />
         </Predicate>
         <Predicate test="greater_than">
            <RuleMethod methodName="urn:foo:Interstitials:SystemDateTime"/>
            <ConstantValue value="2004-09-29T00:00:00Z" />
         </Predicate>
      </rule>
      <rule rule_id="ParentalControl">
         <Predicate test="less_than">
            <RuleMethod methodName="urn:foo:Interstitials:ParentalControl" />
            <ConstantValue value="16" />
         </Predicate>
      </rule>
   </RulesTable>
   <RecordingCache>
      <Request>
         <Item>
            <Condition require="CarpetWorldPromotion" />
            <ContentRef crid="crid://foo.com/CarpetWorldSale" />
            <ContentRef crid="crid://foo.com/CarpetWorldEndofSale" />
         </Item>
         <Item>
            <ContentRef crid="crid://foo.com/TV-Anytime/Promotion" />
         </Item>
      </Request>
      <Replace>
         <Item>
            <Condition require="CarpetWorldPromotion" />
            <ContentRef crid="crid://foo.com/CarpetWorldSale" />
         </Item>
      </Replace>
      <Expire>
         <Item>
            <Condition except="CarpetWorldPromotion" />
            <ContentRef crid="crid://foo.com/CarpetWorldEndofSale" />
            <ContentRef crid="crid://foo.com/CarpetWorldSale" />
         </Item>
      </Expire>
   </RecordingCache>
<TargetingTable>
```

9   Root Types

The following data type forms the root of an interstitial instance document, and is used to encapsulate the set of rules used within the document, the "recording cache" and the "interstitial substitution" controls.

```
< complexType name="InterstitialTableType">
   < sequence>
     < element name="InterstitialBreak" type="int:InterstitialBreakType" minOccurs="0"
maxOccurs="unbounded"/>
   </ sequence>
</ complexType>
```
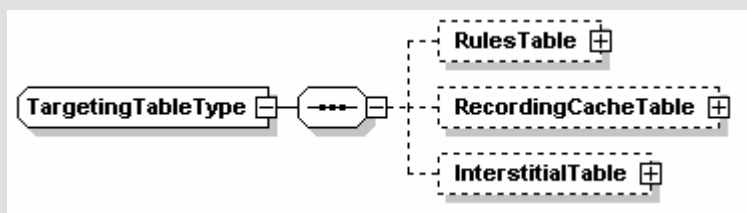


```
<complexType name="TargetingTableType">
   <sequence>
     <element name="RulesTable" type="int:RulesType" minOccurs="0"/>
     <element name="RecordingCache" type="int:RecordingCacheType"
     minOccurs="0"/>
     <element name="InterstitialTable" type="int:InterstitialTableType"
     minOccurs="0"/>
   </sequence>
</complexType>
```

Name	Definition
InterstitialTableType	A complex type that wraps zero or more interstitial substitution declarations.
InterstitialBreak	An element that allows the declaration of a possible interstitial substitution.
TargetingTableType	A complex type that forms the root of an instance document. It forms a wrapper for all the types previously defined within the present document. This data type is declared as a child element of TVAMain (TS 102 822-3-3 [5]).
RulesTable	An element which provides a wrapper for the set of rules expressed within this instance document.
RecordingCache	An element which provides a wrapper for the recording cache elements. See clause 8.
InterstitialTable	An element that provides a root for all interstitialBreak elements that define how and what content shall be substituted.

Annex A (informative):
TV-Anytime Phase 2 Interstitial Metadata Schemes

The *TV-Anytime* interstitial metadata scheme listed in the present document has been aggregated into an xsd file identified "tva_interstitial_3-4_v111.xsd", forming the reference documentation, contained in archive ts_1028220304v010101p0.zip, which accompanies the present document.

The interstitial metadata scheme imports other files that need to be present in order to be valid:

- "tva_mpeg7.xsd" that is available in archive ts_1028220301v010301p0.zip accompanying TS 102 822-3-1 [3].

- "tva2_metadata_3-3_v111.xsd" that is available in archive ts_1028220303v010101p0.zip accompanying TS 102 822-3-3 [5].

- "tva_metadata_3-1_v131.xsd" that is available in archive ts_1028220301v010301p0.zip accompanying TS 102 822-3-1 [3].

- "xml.xsd" that is available in archive ts_1028220301v010301p0.zip accompanying TS 102 822-3-1 [3].

History

Document history		
V1.1.1	January 2006	Publication