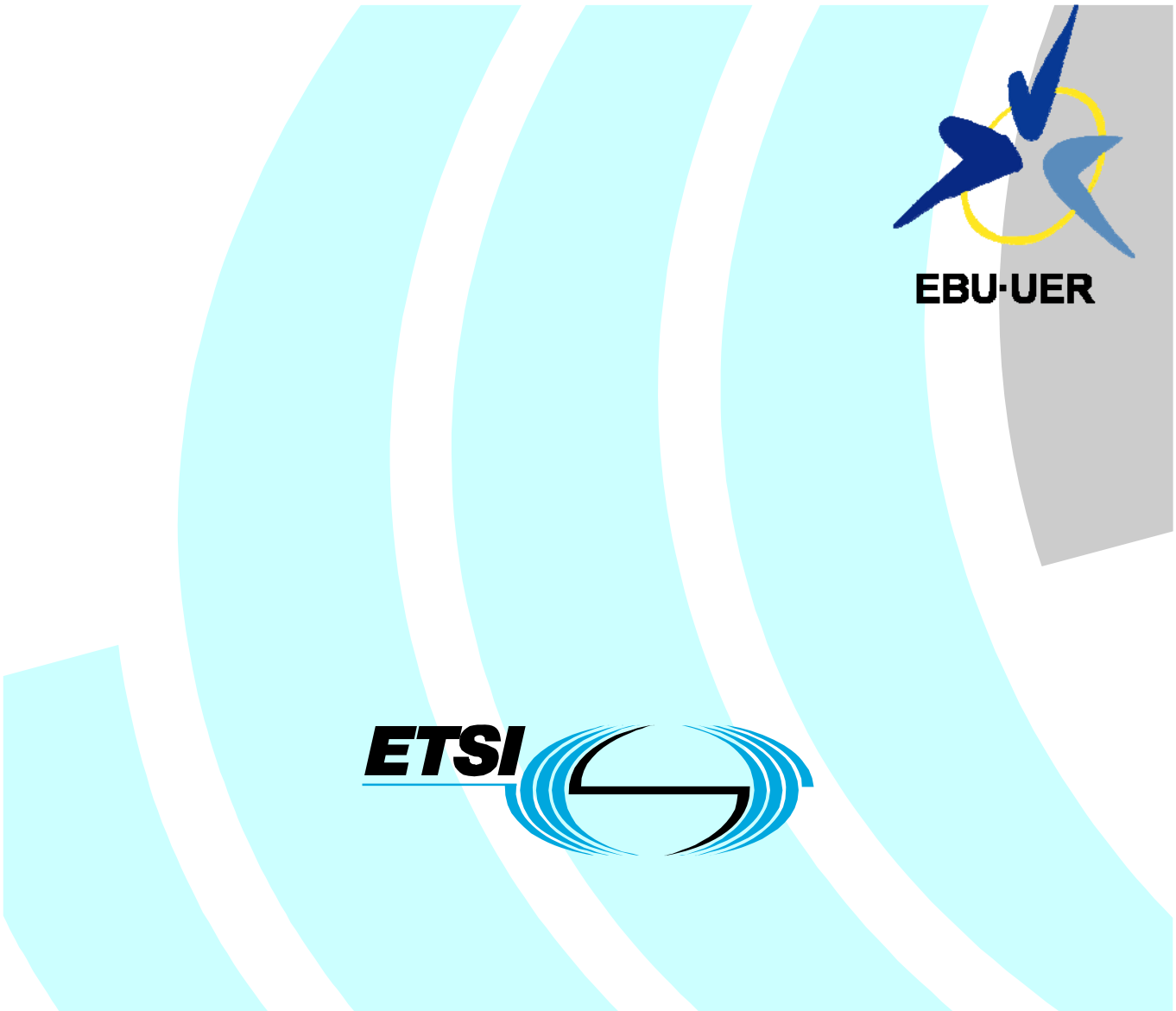
ETSI TS 102 822-9 V1.4.1 (2010-07)

Technical Specification

Broadcast and On-line Services: Search, select, and rightful use of content on personal storage systems ("*TV-Anytime*");
Part 9: Phase 2 - Remote Programming



Reference
RTS/JTC-TVA-PH2-54-09

Keywords
broadcasting, content, system, 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

No part may be reproduced except as authorized by written permission.
The copyright and the foregoing restriction extend to reproduction in all media.

© European Telecommunications Standards Institute 2010.
© European Broadcasting Union 2010.
All rights reserved.

DECT™, **PLUGTESTS**™, **UMTS**™, **TIPHON**™, the TIPHON logo and the ETSI logo are Trade Marks of ETSI registered for the benefit of its Members.
3GPP™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners.
LTE™ is a Trade Mark of ETSI currently being registered
for the benefit of its Members and of the 3GPP Organizational Partners.
GSM® and the GSM logo are Trade Marks registered and owned by the GSM Association.

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 9 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: "Phase 1 - System description";

Part 3: "Metadata";

Part 4: "Phase 1 - 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 enables 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 [6]), Content Referencing (TS 102 822-4 [7]), Rights Management (TS 102 822-5-1 [8] and TS 102 822-5-2 [9]), Bi-directional Metadata Delivery (TS 102 822-6-1 [10], TS 102 822-6-2 [11] and TS 102 822-6-3 [12]) and Protection (TS 102 822-7 [13]), Interchange Data Format (TS 102 822-8 [14]) and Remote Programming (TS 102 822-9). 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.

The *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.

The present document defines methods for remotely controlling personal and network digital recorders (PDRs and NDRs).

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 enables 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 present document has been developed during the second phase of *TV-Anytime* and covers the definition of the interchange data format for the delivery of *TV-Anytime* metadata and content referencing information from different data sources.

The *TV-Anytime* Forum has defined a number of data types that can be exchanged between *TV-Anytime* devices. These include program metadata, content referencing information, and user-centric metadata.

There is an interest to remotely control a PDR from other devices and locations such as a personal computer in the office or a PDA on the move. The present document defines how a mailing service may be used to remotely control a home PDR from outside.

There is also an interest to make use of an NDR (Network Digital Recorder) when the home PDR is not in a position to record content the end-user wants to obtain. The present document defines how to declare an NDR service, how to discover it and how to make use of it to record content to be delivered to the end-user.

2 References

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the reference document (including any amendments) applies.

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

NOTE: While any hyperlinks included in this clause were valid at the time of publication ETSI cannot guarantee their long term validity.

2.1 Normative references

The following referenced documents are necessary for the application of the present document.

[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: Phase 1 - 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 102 822-3-4: "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".

[7] 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: Phase 1 - Content referencing".

[8] 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".

[9] 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".

[10] 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".

[11] 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".

[12] 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".

[13] 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".

[14] 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".

[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] Namespaces in XML, W3C Recommendation, 14 January 1999 T. Bray, D. Hollander, A. Layman.

NOTE: Available at: http://www.w3.org/TR/1999/REC-xml-names-19990114.

[17] IETF RFC 1945: "Hypertext Transfer Protocol, HTTP/1.0".

NOTE: Available at: http://www.ietf.org/rfc/rfc1945.txt.

[18] Simple Object Access Protocol (SOAP) 1.1, W3C Note, 8 May 2000.

NOTE: Available at: http://www.w3.org/TR/2000/NOTE-SOAP-20000508/.

[19] Universal Description Discovery & Integration (UDDI), Version 3.0.

NOTE: Available at: http://uddi.org/pubs/uddi-v3.00-published-20020719.htm.

[20] Web Services Description Language, Version 1.1, W3C Note 15 March 2001.

NOTE: Available at: http://www.w3.org/TR/2001/NOTE-wsdl-20010315.

[21] Web Services Inspection Language, Version 1.0.

NOTE: Available at: http://www.ibm.com/developerworks/webservices/library/ws-wsilspec.html.

[22] The WS-Inspection and UDDI Relationship.

NOTE: Available at: http://www-106.ibm.com/developerworks/webservices/library/ws-wsiluddi.html.

[23] IETF RFC 822: "ARPA Internet Text Messages".

NOTE: Available at: http://www.faqs.org/rfcs/rfc822.html.

2.2 Informative references

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

Not applicable.

3 Definitions and abbreviations

3.1 Definitions

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

acquisition: process of retrieving selected content

application: specific set of functions running on the PDR

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

authority: organization that generates CRIDs

bi-directional network: network that supports two way, point-to-point, one-to-many, and many-to-many data delivery

NOTE: Internet is an example of such a network. A PDR may access a bi-directional network using its return path.

capture: process of storing the acquired content (e.g. to local storage)

client: personal digital recorder or any device connected to the Internet

content: audio, video or other types of material the viewer would like to access

NOTE: Movies, games, TV programmes, radio programmes, etc.

content reference: pointer to a specific content item

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

locator: time and place from where content can be acquired

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 service: service that provides *TV-Anytime* data via a server on a bi-directional network using the data and protocols defined in the present document

programme: editorially coherent piece of content that is acquired by a PDR as a whole

return path: path of the bi-directional distribution system from a consumer to a service provider

service provider: aggregator and supplier of content or metadata which may include gateway and management roles

3.2 Abbreviations

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

CRID Content Reference IDentifier

NOTE: An identifier for content that is independent of its location specified by TS 102 822-4 [7].

HTTP	HyperText Transfer Protocol
IMI	Instance Metadata Identifier
IP	Internet Protocol
MIME	Multipurpose Internet Mail Extension
NDR	Network Digital Recorder
PC	Personal Computer
PDA	Personal Digital Assistant
PDR	Personal Digital Recorder
SOAP	Simple Object Access Protocol
TVA	*TV-Anytime*
UDDI	Universal Description Discovery and Integration
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
W3C	World Wide Web Consortium
WSDL	Web Services Description Language
WS-Inspection	Web Services Inspection
XML	eXtensible Markup Language

4 Remote PDR access

4.1 Remote PDR programming (metadata delivery)

There is a desire to be able to remotely control a PDR, primarily to allow remote booking of recordings.

Remote booking will enable a user to program their PDR from their office PC, from their mobile, whilst on holiday etc. In addition it will also provide a mechanism for a metadata provider to recommend programs to be booked. This could be achieved by a PDR user subscribing to a recommendation engine. The recommendation engine would then periodically mail the PDR to book content to be recorded depending on the user's profile.

The namespace for *TV-Anytime* remote PDR programming has been defined as:

```
urn:tva:ndr:2010
```

4.1.1 Remote PDR programming by e-mail

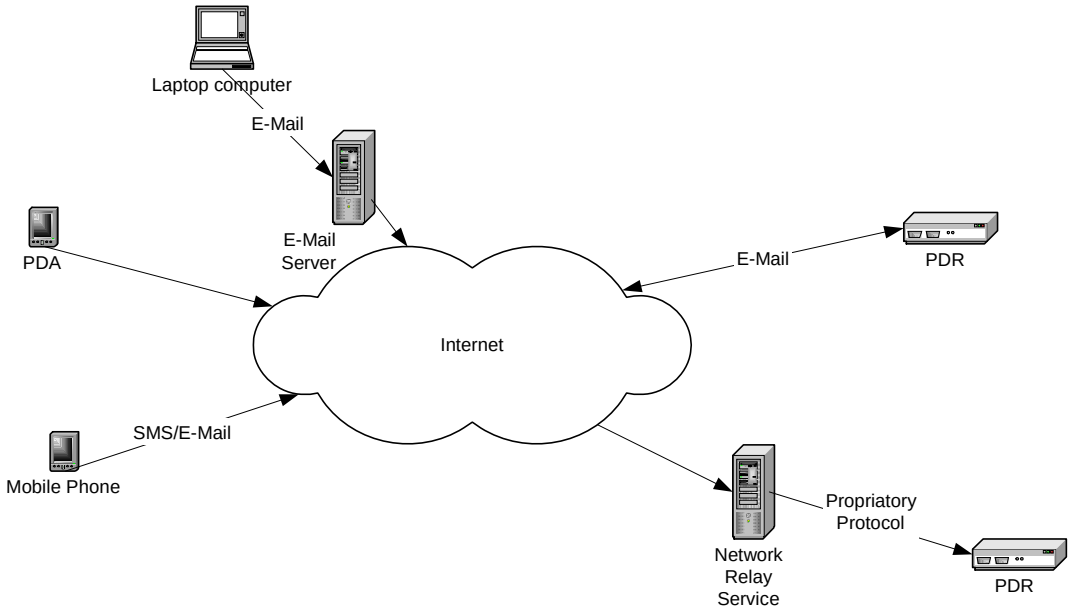


Figure 1: Architecture

The e-mail shall conform to RFC 822 (ARPA Internet Text Messages) [23].

The Subject line shall contain the CRID [7] or CRIDs to be recorded. When multiple CRIDs are provided they shall be separated by one or more space characters e.g. CRID://foo.co.uk/Westenders.

Optionally the e-mail message body may contain an XML instance as specified in TS 102 822-8 [14]. When included the Content-Type parameter within the e-mail header shall be set to "text/xml".

The "core_data" instance allows you to define the CRIDs to be booked. When these are included they take precedence over those contained within the subject line of the e-mail message header.

The actual RemoteBooking message shall form the "body" of the message.

The following is an example of an e-mail using only the Subject line.

```
From: user@mailserver.com
To: Bookings.user2pdr@pdr.com
Subject: CRID://foo.co.uk/Westenders
Date: Thu, 3 Feb 2009 13:20:00
Message-Id:
```

The following is an example of an e-mail using the message body:

```
From: user@mailserver.com
To: Bookings.user2pdr@pdr.com
Subject: Recommended content
Date: Thu, 3 Feb 2009 13:20:00
Message-Id:
Content-Type: text/xml; charset=UTF-8
<?xml version="1.0" encoding="UTF-8"?>
<CoreData xmlns="urn:tva:CoreData:2010" xmlns:tva="urn:tva:metadata:2010"
xmlns:CR="urn:tva:ContentReferencing:2010"
xmlns:xsi:schemaLocation="urn:tva:CoreData:2010 tva_core_data_8_v141.xsd">
  <SelectedContent id="CRID://foo.co.uk/Westenders">
    <tva:TVAMain version="03" xml:lang="en" publisher="..." publicationTime="2001-04-
    05T21:00:00.00+01:00">
      <tva:CopyrightNotice>...</tva:CopyrightNotice>
      <tva:ProgramDescription>
        <tva:ProgramInformationTable>
          <tva:ProgramInformation programId="CRID://foo.co.uk/Westenders">
```

```
            <tva:BasicDescription>
              <tva:Title>Eastenders</tva:Title>
              <tva:Keyword>Soap</tva:Keyword>
            </tva:BasicDescription>
          </tva:ProgramInformation>
        </tva:ProgramInformationTable>
      </tva:ProgramDescription>
    </tva:TVAMain>
    <WSIFServerAddress>http://www.TV-AllChannels.com</WSIFServerAddress>
    <Action>
      <Type href="urn:tva:CoreData:cs:CoreDataActionTypeCS:2005:5">
        <tva:Name xml:lang="en">recommend</tva:Name>
        <tva:Definition xml:lang="en">preselection coming from me or suggestion coming
        from any user</tva:Definition>
      </Type>
    </Action>
  </SelectedContent>
</CoreData>
```

4.1.2 Other remote PDR programming solutions

Other mechanisms may be considered to program a PDR from a remote device and location (e.g. a Web site or an SMS service) that are not defined in the present document.

4.2 Remote PDR management

Although certain remote PDR management functions such as "list content", "search", "delete", "show recording list", "transfer content to and from another device" are useful, the present document only defines the "recording request" function and the "related status request" function.

5 NDR operation

5.1 NDR remote programming scenarios

The use of an NDR is of interest in a number of situations where:

- the end-user is outside their home with no possibility to access his home PDR;
- content the end-user is interested in, is out of reach of his PDR (content broadcast on a channel not delivered to the end-user's receiver, conflict between recording requests, conflict between live event viewing and a recording request on a limited reception resource, etc.).

Different NDR usage scenarios illustrated in figure 2:

- recording request from a PDR for a content to be delivered to the PDR;
- recording request from a PDR for a content to be delivered to a PC or PDA;
- recording request from a PC or PDA for a content to be delivered to a PDR;
- recording request from a PC or PDA for a content to be delivered to the same equipment.

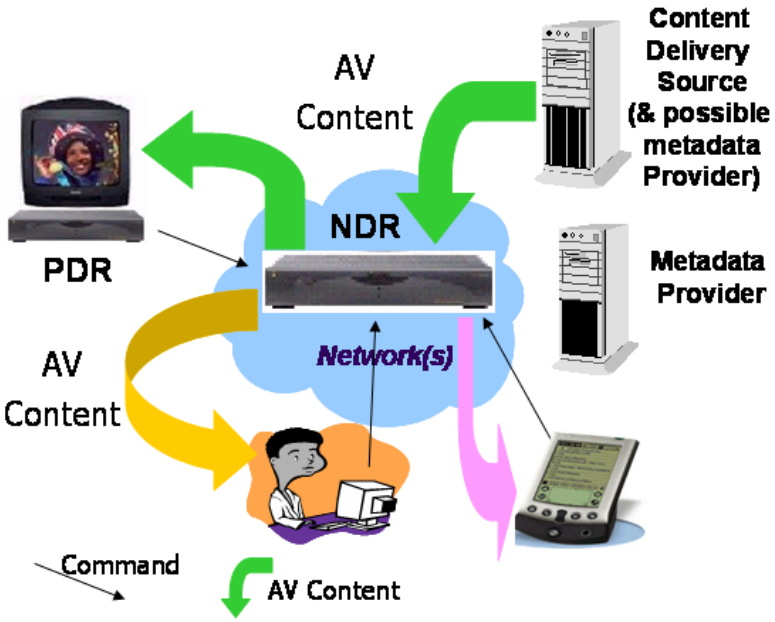


Figure 2: Remote programming scenarios

5.2 NDR control service and transport

5.2.1 Introduction

The following service description is based on the "NDR Service operation and transport" structure defined in TS 102 822-6-1 [10], and on the "NDR Service discovery" mechanisms defined in TS 102 822-6-2 [11].

5.2.1.1 Types and Functionalities of NDR services

TV-Anytime NDR services specified are all request-response based. The network transaction is always point-to-point (client to server), and the transaction is always initiated by the client.

NDR control:

NDR control occurs when a client wishes to record broadcast content.

For the *TV-Anytime* control_NDR operation that is provided, there is a corresponding operation, describe_control_NDR. These two operations (an operation, *X*, and its corresponding *describe_X* operation) together form a port (see annex A). Since a port always has a single binding and endpoint, the URL of the operation, *X*, and its corresponding *describe_X* operation must be the same. A "describe" operation is responsible for returning a capability description for the corresponding operation of the same port. A "describe" operation has no input parameters.

5.2.1.2 NDR service capability descriptions

In order to exploit usefully the NDR service described in the previous clause, the client needs information about the nature of the NDR service being offered. This is because different NDR services will provide different functionalities and optional features. For example, some NDR services may offer just content recording, whilst others may provide transrating or transcoding.

To address this issue, each NDR service provides, on request from a client, a capability description. This capability description allows a client to know about the features supported by the NDR service.

Furthermore, it allows NDR service providers to introduce new features and to make this known by end users.

5.2.2 NDR service types

TV-Anytime defines one type of NDR web service, which can be thought of as a remote procedure with a well-defined set of inputs and outputs and a well-defined behaviour. In WSDL [20] terminology, this remote procedure is known as an "operation" (see annex A). The NDR control operation is called the control_NDR operation.

The types used in the requests and responses to *TV-Anytime* NDR services are defined in the target namespace [16]: "urn:tva:ndr:2005". This allows Schema aware tools to validate the various messages. The types defined in the metadata specification TS 102 822-3-1 [3] and content referencing specification TS 102 822-4 [7] schemas are referenced in the transport namespace (using XML Schema's import mechanism [15]).

The schema fragments in clause 5.2.2 are all defined within this namespace. The corresponding namespace qualifier used in these schema fragments is "ndr". The complete XML schema file (tva_ndr_9_v111.xsd) may be found attached in annex C.

5.2.2.1 "control_NDR" operation

The control_NDR operation allows a client to query a server in order to control the hosted NDR function. The following list gives a few examples of the types of functionality that a *TV-Anytime* NDR provider can offer using a control_NDR operation:

- operation that takes a certain channel and a command Record to get the content currently broadcast on this channel recorded from now on till the end and returns a positive response;
- operation that takes a certain channel and a command Record to get the content currently broadcast on this channel recorded from now until the end and returns a negative response (unknown channel);
- operation that takes a CRID and an IMI and a command Record to get the corresponding content recorded and returns an availability date and time to ask for content availability;
- operation that takes an accepted requestId and a command Cancel to cancel a previously accepted record request;
- operation that takes an accepted requestId and a command Status and returns the availability of the content or the date and time to ask again for content availability if the content is not yet available.

A control_NDR operation can support all these types of queries.

5.2.2.2 Request format

The request format allows the client to specify four types of commands. The semantics and structure of these commands are explained in more detail in clause 5.2.2.3.

```
<complexType name="Control_NDRType">
  <choice>
    <element name="RecordRequest" type="ndr:RecordRequestType"/>
    <element name="RecordStatus" type="ndr:RecordStatusType"/>
    <element name="RecordCancel" type="ndr:RecordCancelType"/>
    <element name="RecordListRequest" type="ndr:RecordListRequestType"/>
  </choice>
</complexType>
<element name="Control_NDR" type="ndr:Control_NDRType"/>
```

Name	Definition
Control_NDRType	A complex type to describe all the requests which can be sent to an NDR.
RecordRequest	An element used to request the recording of a specific content currently broadcast or to be broadcast.
RecordStatus	An element used to ask for the status of a previously accepted Record Request.
RecordCancel	An element used to cancel a previously accepted record request.
RecordListRequest	An element used to ask for the list of previously accepted record requests arisen by a user.
Control_NDR	An element used to contain a request to an NDR.

5.2.2.3 RecordRequest parameters

5.2.2.3.1 Introduction

The RecordRequest command may contain a number of parameters to allow several types of record request:

- a record request for the content currently broadcast on a channel (no ContentId element);
- a record request for future broadcast content (all elements may be needed);
- any record request may include a request for format conversion, bit rate conversion or use of a specific delivery protocol as allowed by the NDR capability description.

If a CRID is supplied, it is a request for a complete recording and allows the NDR to resolve it.

A partial locator contains only the serviceURL.

A full locator contains a serviceURL, start time and duration.

We do not allow for a mix of partial and full locators in a record request.

- Case A: A partial locator (immediate recording and results in partial recording).
- Case B: CRID plus a list of partial locators (results in complete recording).
- Case C: A full locator (partial or complete recording).
- Case D: A list of full locators (partial or complete recording).
- Case E: A CRID plus a list of full locators (complete recording).
- Case F: A CRID plus a list of IMI(s) (complete recording).

A Record request is defined as follows.

```
<complexType name="ContentIdType">
  <sequence>
    <element name="InstanceMetadataId" type="tva:InstanceMetadataIdType"
    minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
  <attribute name="CRID" type="tva:CRIDType" use="required"/>
</complexType>
<complexType name="DeliveryProtocolType">
  <complexContent>
    <extension base="tva:ControlledTermType">
      <attribute name="version" type="float" use="optional"/>
    </extension>
  </complexContent>
</complexType>
<complexType name="ControlProtocolType">
  <complexContent>
    <extension base="tva:ControlledTermType">
      <attribute name="version" type="float" use="optional"/>
    </extension>
  </complexContent>
</complexType>
<complexType name="ProtocolSetType">
  <sequence>
    <element name="DeliveryProtocol" type="ndr:DeliveryProtocolType" minOccurs="0"/>
    <element name="ControlProtocol" type="ndr:ControlProtocolType" minOccurs="0"/>
  </sequence>
</complexType>
<complexType name="RecordRequestType">
  <sequence>
    <element name="SubscriptionId" type="string"/>
    <element name="ContentId" type="ndr:ContentIdType" minOccurs="0"/>
    <element name="Locator" type="cr:LocatorType" minOccurs="0" maxOccurs="unbounded"/>
    <element name="DeliveryMediaFormat" type="tva:AVAttributesType" minOccurs="0"/>
```

```
      <element name="ProtocolSet" type="ndr:ProtocolSetType" minOccurs="0"/>
      <element name="StartTime" type="dateTime" minOccurs="0"/>
      <element name="EndTime" type="dateTime" minOccurs="0"/>
      <element name="FilteringAndSearchPreferences"
      type="mpeg7:FilteringAndSearchPreferencesType" minOccurs="0"/>
    </sequence>
  </complexType>
```

Name	Definition
ContentIdType	A complex type identifying content to be recorded.
InstanceMetadataId	This element containing a reference to a specific instance of the content to be recorded.
CRID	An attribute containing the Content Reference IDentifier (CRID [7]) of the selected content to be recorded.
DeliveryProtocolType	A complex type to identify a delivery protocol supported by the NDR.
version	An optional attribute to identify the version of the delivery protocol. The default value is the initial version of the delivery protocol.
ControlProtocolType	A complex type to identify a control protocol supported by the NDR.
version	An optional attribute to identify the version of the supported control protocol. The default value is the initial version of the control protocol.
ProtocolSetType	A complex type to develop a delivery protocol and the associated control protocol if any.
DeliveryProtocol	An element providing the identification of a delivery protocol supported by the NDR.
ControlProtocol	An element providing the identification of the control protocol supported by the NDR associated to the delivery protocol if any. E.g. There is no control protocol associated with the FTP delivery protocol. The RTSP control protocol can be associated to RTP, UDP or HTTP.
RecordRequestType	A complex type used to define an NDR record request.
SubscriptionId	This element identifies the subscription made by the end-user. This SubscriptionId is obtained by the end-user as the result of subscription made over the Internet with the subscription server indicated by the "SubscriptionURL" element located in the NDR service capability description.
ContentId	This element identifies the content to be recorded. If this element is missing, the content to be recorded is the content currently broadcast on the channel indicated by the element "Locator".
Locator	This element defined in TS 102 822-4 [7] identifies the channel from which the selected content should be recorded. It might include in addition date and time and IMI.
DeliveryMediaFormat	An optional element allowing to specify content to be recorded in an encoding format different than the one used by the broadcaster and/or a different bit rate for the encoding of the recorded content before delivery to the end-user. Default value is delivery media format identical to source media format.
ProtocolSet	An optional element indicating the protocols to be used for the delivery and the control of the content to be recorded, among all values defined in the DeliveryProtocolType classification scheme given in clause C.2. (DeliveryProtocolTypeCS.xml) and in the ControlProtocolType classification scheme given in clause C.3 (ControlProtocolTypeCS.xml). The default values are "rtp" and "rtsp".
StartTime	An optional element indicating the time at which the recording should start.
EndTime	An optional element indicating the time at which the recording should end.
FilteringAndSearchPreferences	An element indicating the presence of FilteringAndSearchPreferences to be used by the NDR Service, in case there is a need to resolve a CRID.

5.2.2.3.2 RecordStatus parameters

When a PDR or end-user wants to know if a previously accepted record request is completed or not, it issues a RecordStatus command, as follows.

```
<!-- ########### Schema of the status of a record request ########### -->
 <complexType name="RecordStatusType">
  <attribute name="requestId" type="string" use="required"/>
 </complexType>
```

Name	Definition
RecordStatusType	A complex describing a record status.
requestId	An attribute containing the record request identifier returned by the NDR service when a record Request is accepted.

5.2.2.3.3 RecordCancel parameters

When a PDR or end-user wants to cancel a previously accepted record request, it issues a RecordCancel command as follows.

```
<!-- ########### Schema of the cancel of a record request ########### -->
 <complexType name="RecordCancelType">
  <attribute name="requestId" type="string" use="required"/>
 </complexType>
</schema>
```

Name	Definition
RecordCancelType	A complex type describing a record cancel request.
requestId	An attribute containing the record request identifier returned by the NDR service when a record request is accepted.

5.2.2.3.4 RecordListRequest parameters

When a PDR or end-user wants to know the list of previously accepted record requests which he generated, it issues a RecordListRequest command, as follows.

```
<!-- ######## Schema of the list of previously accepted record requests ######## -->
  <complexType name="RecordListRequestType">
    <sequence>
      <element name="SubscriptionId" type="string"/>
    </sequence>
  </complexType>
```

Name	Definition
RecordListRequestType	A complex type used to define an NDR record list request. As a result, it provides the list of previously accepted record requests arisen by the user with ID, SubscriptionId.
SubscriptionId	This element identifies the subscription made by the end-user. This SubscriptionId is obtained by the end-user as the result of subscription made over the Internet with the subscription server indicated by the "SubscriptionURL" element located in the NDR service capability description.

5.2.2.4 Response format

A control_NDR response to a request contains a TVAMain XML instance document specific to each request.

```
<complexType name="ControlNDRResultType">
  <choice>
    <element name="RecordRequestResult" type="ndr:RecordRequestResultType"/>
    <element name="RecordStatusResult" type="ndr:RecordStatusResultType"/>
    <element name="RecordCancelResult" type="ndr:RecordCancelResultType"/>
    <element name="RecordListRequestResult" type="ndr:RecordListRequestResultType"/>
  </choice>
</complexType>
<element name="Control_NDR_Result" type="ndr:ControlNDRResultType"/>
```

Name	Definition
ControlNDRResultType	A complex type that defines the responses to the different requests that can be sent to an NDR service provider.
RecordRequestResult	An element provided by the NDR Service that contains the response to a record request command.
RecordStatusResult	An element provided by the NDR Service that contains the response to a record status command.
RecordCancelResult	An element provided by the NDR Service that contains the response to a record cancel command.
RecordListRequestResult	An element to collect the results of a request to identify a record list.
Control_NDR_Result	An element describing an NDR response to a request.

5.2.2.4.1 RecordRequestResult parameters

```
<!-- ############ Description of the response to a record request ############ -->
<complexType name="RecordRequestOKType">
  <sequence>
    <element name="RequestId" type="string"/>
    <element name="Time2Call" type="dateTime"/>
    <element name="RecordingCharge" type="tva:PriceType" minOccurs="0"/>
    <element name="ConservationDelay" type="duration" minOccurs="0"/>
  </sequence>
</complexType>
<simpleType name="RecordRequestErrorType">
  <restriction base="string">
    <enumeration value="unknownSubscriptionId"/>
    <enumeration value="unknownCRID"/>
    <enumeration value="unavailableServiceURL"/>
    <enumeration value="unavailableNDRService"/>
    <enumeration value="unknownDeliveryProtocol"/>
    <enumeration value="unknownControlProtocol"/>
    <enumeration value="unsupportedDeliveryProtocol"/>
    <enumeration value="unsupportedControlProtocol"/>
    <enumeration value="unknownOriginalMediaFormat"/>
    <enumeration value="unsupportedOriginalMediaFormat"/>
    <enumeration value="unknownDeliveryMediaFormat"/>
    <enumeration value="unsupportedDeliveryMediaFormat"/>
  </restriction>
</simpleType>
<complexType name="RecordRequestResultType">
  <sequence>
    <choice>
      <element name="RecordRequestOK" type="ndr:RecordRequestOKType"/>
      <element name="RecordRequestError" type="ndr:RecordRequestErrorType"/>
    </choice>
  </sequence>
  <attribute name="serviceVersion" type="unsignedInt" use="required"/>
</complexType>
```

Name	Definition
RecordRequestOKType	A complex type providing additional information when the request can be fulfilled.
RequestId	An element containing the record request identifier returned by the NDR service when a record request is accepted.
Time2Call	This element provided by the NDR service contains the date and time at which it is likely that the selected content will be available for delivery to the end-user using the selected delivery protocol.
RecordingCharge	This element provided by the NDR service contains the charge to be applied to the requested recording.
ConservationDelay	This element provided by the NDR service contains the life time of the selected content recording.
RecordRequestErrorType	A simple type describing possible error messages to be returned when a record request cannot be fulfilled: unknownCRID: this error appears when the CRID provided in the record request does not exist; unavailableServiceURL: this error appears when the NDR Service has no access to the TV Channel indicated by "ServiceURL"; unavailableNDRService: this error appears when the NDR Service is not able to accept the request at this time.
RecordRequestResultType	A complex type describing the results of a record request.
RecordRequestOK	When the record request is accepted by the NDR service, an element provides the necessary information about the progress of the recording, to the attention of the user.
RecordRequestError	An element providing the reason why a record request could not be fulfilled.
serviceVersion	An attribute also available in the NDR service capability description. It indicates the version of the NDR service capability description. When the NDR service makes a modification to the NDR service capability description, it increases the serviceVersion. This allows the client to discover that a new version of the NDR service capability description is available. It should make a request to update its own version of the NDR service capability description.

5.2.2.4.2 RecordStatusResult parameters

```
<!-- ############ Description of the response to a record status request  ######### -->
<simpleType name="FailureType">
  <restriction base="string">
    <enumeration value="unknownRequest"/>
    <enumeration value="cancelledBroadcast"/>
    <enumeration value="cancelledByNDR"/>
  </restriction>
</simpleType>
<complexType name="RecordStatusResultType">
  <choice>
    <element name="Running">
      <complexType>
        <sequence>
          <element name="Time2Call" type="dateTime"/>
        </sequence>
      </complexType>
    </element>
    <element name="ContentAvailable">
      <complexType>
        <sequence>
          <element name="ContentURL" type="anyURI"/>
          <element name="ConservationDeadline" type="dateTime" minOccurs="0"/>
        </sequence>
      </complexType>
    </element>
    <element name="Failure" type="ndr:FailureType"/>
  </choice>
  <attribute name="requestId" type="string" use="required"/>
</complexType>
```

Name	Definition
FailureType	A simple type to express the reason why the record request referred by the record status request has failed.
RecordStatusResultType	A complex type describing the status of a record request.
Running	The presence of this element indicates that the content is not yet recorded.
Time2Call	An element indicating at what time the content is likely to be available for delivery to the end user.
ContentAvailable	The presence of this element indicates that the content is available but maybe not yet completely recorded. This depends on the protocols to be used for control and delivery to the end-user.
ContentURL	This element contains the URI to be used to get the recorded content from the NDR Service.
ConservationDeadline	This element provided by the NDR Service contains the deadline to get the selected content recording.
Failure	This optional element contains the reason why the referred record request has failed.
requestId	An attribute containing the record request identifier returned by the NDR service when a record request is accepted.

5.2.2.4.3 RecordCancelResult parameters

```
  <!-- ########### Description of the response to a cancel request  ########### -->
  <simpleType name="RecordCancelResultType">
    <restriction base="string">
      <enumeration value="unknownRequest"/>
      <enumeration value="OK"/>
      <enumeration value="removedAfterRecording"/>
    </restriction>
  </simpleType>
```

Name	Definition
RecordCancelResultType	A complex type used to describe the result of a cancellation request.

5.2.2.4.4 RecordListRequestResult parameters

```
<complexType name="RecordListRequestResultType">
    <sequence>
      <element name="RecordRequestStatus"
      type="ndr:RecordStatusResultType" minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
    <attribute name="serviceVersion" type="unsignedInt" use="required"/>
</complexType>
```

Name	Definition
RecordListRequestResultType	A complex type describing the results of a record list request.
RecordRequestStatus	An element containing information on the status of a request.
serviceVersion	An attribute also available in the NDR service capability description. It indicates the version of the NDR service capability description. When the NDR service makes a modification to the NDR service capability description, it increases the serviceVersion. This allows the client to discover that a new version of the NDR service capability description is available. It should make a request to update its own version of the NDR service capability description.

5.2.3 Transport protocol

5.2.3.1 Introduction

SOAP [18] and HTTP [17] are used for delivering *TV-Anytime* XML data over the IP networks, since this combination is very well suited to the point-to-point, request-response nature of the *TV-Anytime* operations. The exact usage of SOAP and HTTP is given in the next two clauses. Figure 3 provides a semantic representation of the network stack.

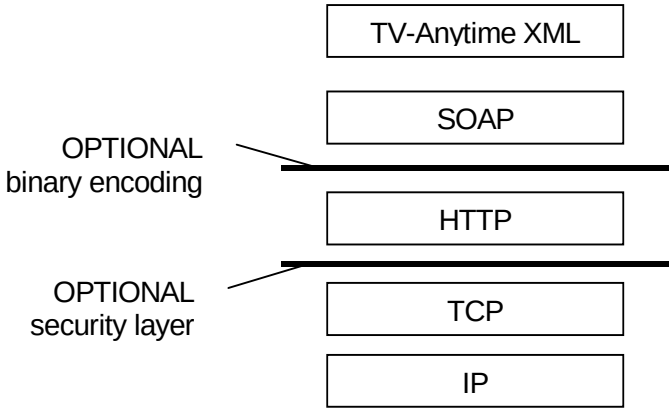


Figure 3: The bi-directional network transport stack

The architecture depicted in figure 3 results in HTTP requests of the following form.

```
POST /tva/md-service HTTP/1.0
Host: www.example.com
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn
Accept-Encoding: deflate
SOAPAction: "control_NDR"

<?xml version="1.0" encoding="UTF-8"?>
<Envelope xmlns="http://schemas.xmlsoap.org/soap/envelope/">
  <Body>
   <ndr:Control_NDR xmlns:NDR="urn:tva:ndr:2010" xmlns:tva="urn:tva:metadata:2010"
xmlns:mpeg7="urn:tva:mpeg7:2008" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:tva:ndr:2010 tva_ndr_9_v141.xsd">
    <RecordRequest>
     <SubscriptionId>3456-4567-5677-4321</SubscriptionId>
     <ContentId CRID="crid://www.broadcaster.com/ajcnd"/>
     <Locator>dvb://1.4ee2.3fa;4f5</Locator>
     <ProtocolSet>
      <DeliveryProtocol href="urn:tva:ndr:cs:DeliveryProtocolTypeCS:2005:ftp"/>
     </ProtocolSet>
    </RecordRequest>
   </ndr:Control_NDR>
  </Body>
</Envelope>
```

The architecture depicted in figure 3 results in HTTP responses of the following form.

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn
Content-Encoding: deflate

<?xml version="1.0" encoding="UTF-8"?>
<Envelope xmlns="http://www.w3.org/2002/06/soap-envelope">
  <Body>
<ndr:Control_NDR_Result xmlns:NDR="urn:tva:ndr:2010" xmlns:tva="urn:tva:metadata:2010"
xmlns:mpeg7="urn:tva:mpeg7:2008" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:tva:ndr:2010 tva_ndr_9_v141.xsd">
  <RecordRequestResult serviceVersion="12">
    <RecordRequestOK>
      <RequestId>45U753-452</RequestId>
      <Time2Call>2009-07-15T20:35:00.00</Time2Call>
      <RecordingCharge currency="EUR">2</RecordingCharge>
      <ConservationDelay>PT12H</ConservationDelay>
    </RecordRequestOK>
  </RecordRequestResult>
</ndr:Control_NDR_Result>
  </Body>
</Envelope>
```

5.2.3.2 SOAP

The following usage of SOAP [18] is mandated:

- *TV-Anytime* metadata services will provide an HTTP binding, and may support other transport bindings where appropriate.
- SOAP supports different messaging styles, but is most commonly used for remote procedure calls. *TV-Anytime* does not use remote procedure call messaging style, since this implies that every parameter must be included in a procedure call, which is not appropriate for *TV-Anytime* operations that have several optional parameter types. However, the remote procedure call convention of naming the root element in the SOAP body according to the name of the operation is followed.
- SOAP encoding is not used in the request or the response (i.e. the element in the root of the SOAP body will belong to the *TV-Anytime* transport types namespace (urn:tva:transport:2005). Servers will reject any request that arrives with a SOAP encoding attribute with the appropriate SOAP Fault.
- The SOAP Actor feature is not supported and servers will reject any request that arrives with a SOAP Actor attribute in the SOAP Header with the appropriate SOAP Fault.
- The following clause defines application specific fault conditions to be used in the SOAP Fault element.

This usage of SOAP is more formally defined using the WSDL [20] interface definition that can be found in annex A.

5.2.3.3 Error codes

The first line of error reporting is governed by the SOAP specification [18]. SOAP fault reporting and fault codes will be returned for most invalid requests or any request where the intent of the caller cannot be determined.

In a manner consistent with the SOAP processing rules, HTTP [17] status codes will be used for communicating status information in HTTP. As is the case for SOAP, success reporting will use a 200-status code to indicate that the client's request including the SOAP component was successfully processed.

The ErrorReport element is defined to allow servers to report application-level errors that are specific to *TV-Anytime* metadata services. In accordance with the SOAP specification, if the content of the SOAP message's body cannot be processed successfully, the SOAP fault must contain a detail element (which in turn contains an ErrorReport). The inability to process a well-formed Body element is also termed an "application-level error", since the error cannot be detected by the SOAP processor and instead relies on application-level knowledge. The error report contains error information that includes descriptions and a code that can be used to determine the cause of the error. Errors that arise due to problems with in the HTTP layer or SOAP layer (e.g. the SOAP message does not conform to the SOAP specification) should be reported using the error mechanisms provided by those layers and MUST NOT be reported inside a SOAP fault's detail element.

TV-Anytime application-level errors should be conveyed using standard HTTP status codes, where a 500-level code indicates a server-induced error. In such cases, the metadata service must issue an HTTP 500 "Internal Server Error" response and return an ErrorReport inside a SOAP fault report.

Any errors detected in the request will invalidate the entire request, and cause an ErrorReport to be generated within a SOAP fault as described below. A server may report multiple errors, although there is no requirement for the metadata service to continue processing the request after detecting the first error. In accordance with the SOAP specification, additional application response elements should not be included in the Body of the SOAP request. In other words, it is not possible for the server to indicate an error condition and also make a best-effort at providing a response.

The ErrorReport element takes the following form.

```
<!-- ############ Description of the error reports  ############ -->
<simpleType name="errorCodeType">
  <restriction base="string">
    <enumeration value="FatalError"/>
    <enumeration value="InvalidRequest"/>
    <enumeration value="Unsupported"/>
    <enumeration value="UnrecognizedVersion"/>
    <enumeration value="UnspecifiedError"/>
  </restriction>
</simpleType>
<complexType name="ErrorReportType">
  <sequence>
    <element name="Error" type="ndr:ErrorType" maxOccurs="unbounded"/>
  </sequence>
</complexType>
<complexType name="ErrorType">
  <sequence>
    <element name="Reason" type="mpeg7:TextualType" minOccurs="0"
    maxOccurs="unbounded"/>
  </sequence>
  <attribute name="errorCode" type="ndr:errorCodeType" use="required"/>
</complexType>
<element name="ErrorReport" type="ndr:ErrorReportType"/>
```

Name	Definition
errorCodeType	A simple type defining a required string that precisely defines the nature of the error.
ErrorReportType	A complex type to enumerate the application level errors that have occurred as a result of invoking an NDR service.
Error	This element describes a single NDR-level error.
ErrorType	A complex type defining the type of the error reported in the error report.
Reason	An optional human meaningful description of the error.
errorCode	An attribute describing the error.
ErrorReport	An element describing errors when sending requests to an NDR service.

The following general error conditions are defined as error codes that may be returned by invoking any of the operations defined in the present document:

- **FatalError:** Signifies that a serious technical error has occurred whilst processing the request.
- **InvalidRequest:** The query is well-formed but not valid according to the schema defined by the present document.
- **Unsupported:** Signifies that the NDR service does not support an optional feature that is required in order to correctly process the request. A possible reason for this error is that the client has assumed a functionality that is at odds with the functionality described in the capability description.
- **UnrecognizedVersion:** Signifies that the namespace of the child element inside the Body element of the request (e.g. urn:tva:ndr:2004 instead of urn:tva:ndr:2005) is not supported by this NDR service.
- **UnspecifiedError:** Signifies any other error.

5.2.4 NDR service capability description control

For each *TV-Anytime* Control_NDR operation that is provided, there is a corresponding operation, describe_control_NDR. This operation (an operation, *X*, and its corresponding *describe_X* operation) together form a port (see annex A). Since a port always has a single binding and endpoint, the URL of the operation, *X*, and its corresponding *describe_X* operation must be the same. A "describe" operation is responsible for returning a service capability description for the corresponding operation of the same port. A "describe" operation has no input parameters.

The describe_control_NDR service capability description provides the following type of information about the operation:

- Human-readable descriptive information about the operation.
- If media bit rate conversion capability is available.
- If media format conversion capability is available.
- If several delivery and control protocols are available.

The definition of the NDR service capability description is as follows.

```
<complexType name="ServiceURLListType">
  <sequence>
    <element name="ServiceURL" type="anyURI" maxOccurs="unbounded"/>
  </sequence>
</complexType>

<complexType name="ConversionOfferType">
  <sequence>
    <element name="OriginalMediaFormat" type="tva:AVAttributesType" minOccurs="0"/>
    <element name="DeliveryMediaFormat" type="tva:AVAttributesType" minOccurs="0"/>
  </sequence>
</complexType>

<complexType name="ConversionListType">
  <sequence>
    <element name="ConversionOffer" type="ndr:ConversionOfferType"
    maxOccurs="unbounded"/>
  </sequence>
</complexType>

<complexType name="ProtocolSetListType">
  <sequence>
    <element name="ProtocolSet" type="ndr:ProtocolSetType" maxOccurs="unbounded"/>
  </sequence>
</complexType>
<complexType name="describe_control_NDR_ResultType">
  <sequence>
    <element name="Name" type="mpeg7:TextualType" minOccurs="0" maxOccurs="unbounded"/>
```

```
      <element name="Description" type="mpeg7:TextualType" minOccurs="0"
      maxOccurs="unbounded"/>
      <element name="SubscriptionURL" type="anyURI" minOccurs="0"/>
      <element name="ServiceURLList" type="ndr:ServiceURLListType"/>
      <element name="ConversionList" type="ndr:ConversionListType" minOccurs="0"/>
      <element name="ProtocolSetList" type="ndr:ProtocolSetListType"
      minOccurs="0"/>
      <element name="PlayWhileRecording" type="boolean" minOccurs="0"/>
      <element name="SupportForFilteringAndSearchPreferences" type="boolean"
      minOccurs="0"/>
    </sequence>
    <attribute name="serviceVersion" type="unsignedInt" use="required"/>
  </complexType>
  <element name="describe_control_NDR_Result"
  type="ndr:describe_control_NDR_ResultType"/>
```

Name	Definition
ServiceURLListType	A complex type to develop a list of service URLs.
ServiceURL	This element contains an URL which allows the receiver to identify the associated physical service. This element should be consistent with the possible BroadcastURL in events that reference this ServiceInformation element.
ConversionOfferType	A complex type describing a conversion mode that can be performed by an NDR.
OriginalMediaFormat	An optional element describing the original media format in which content is broadcast by the source.
DeliveryMediaFormat	An optional element describing the media format for delivery if the NDR is able to convert the source broadcast media format in this other format.
ConversionListType	A complex type to develop a list of conversions that an NDR can perform.
ConversionOffer	Description of a media conversion the NDR is able to do.
ProtocolSetListType	A complex type to develop a list of possible delivery and control protocols from the NDR.
ProtocolSet	An element providing the identification of an association of a control protocol and a delivery protocol that can be used by the NDR for later delivery of the recorded content, e.g. rtsp with http, ftp, rtsp with rtp, etc. All the possible values are identified in the protocol type classification schemes (DeliveryProtocolTypeCS.xml and ControlProtocolTypeCS.xml) located in clauses C.2 and C.3.
describe_control_NDR_ResultType	A complex type defining the relevant parameters describing NDR service capability.
Name	An optional element giving the name of the NDR service, in a human readable form.
Description	An optional element giving a textual description of the NDR service, in a human readable form. This allows an application exploiting the NDR service to indicate to the user information about the NDR service that he is using (e.g. "Record all French channels").
SubscriptionURL	An optional element with a URL from where the user can obtain information on subscription conditions to a NDR service.
ServiceURLList	A required element with a list of serviceURL the NDR is able to record for the user.
ConversionList	An optional element providing a list of conversion capabilities. This element can be omitted if no conversion is provided.

Name	Definition
ProtocolSetList	An optional element providing a list of protocols that can be used for later delivery of the recorded content. The default value is "rtp with rtsp".
PlayWhileRecording	An optional element allowing the NDR to declare that it is able to deliver the content while the content is still broadcast. Default value is "false".
SupportForFilteringAndSearchPreferences	An optional element This allows the NDR to declare that it is able to support FilteringAndSearchpreferences to select a specific instance of the content to be recorded. Default value is "false".
serviceVersion	A required attribute containing a number that must equal the version number returned as part of the corresponding operation's result. The intention of this number is to make the client aware when an operation has been upgraded (e.g. can be searched on new channels or for new features) and to refresh the cached capability description information (if any).
describe_control_NDR_Result	An element providing information on the NDR service capability.

5.3 NDR service discovery methods

NDR service discovery is the process by which a client establishes a URL where a *TV-Anytime* NDR service can be found. There are a number of ways this process can occur, but only the last method (clause 5.4) is addressed by the present document.

5.3.1 Non-standardized discovery

A number of methods exist for discovering the URLs of NDR services that are not standardized by *TV-Anytime*. The following list gives some examples:

- the client might be pre-programmed with a set of URLs that refer to one or more NDR services. This will typically be useful in a vertical market, or tightly controlled horizontal market;
- a user might manually enter a URL of a new NDR service he is interested in, using some means of text input;
- a client software may be updated using software updates delivered via a unidirectional broadcast, or over the return channel.

5.3.2 Client-initiated discovery using a bi-directional network

This mode of NDR service discovery involves using the bi-directional network to access a "Yellow Pages" of *TV-Anytime* NDR services. The mechanism is based upon W3C standards for web service discovery (UDDI [19] and WS-Inspection [21]), the use of which is standardized by *TV-Anytime*, according to the rules given in clause 5.4. Support for these discovery techniques by clients and servers is optional.

5.4 NDR service discovery using web service discovery

The present document describes how standard web service discovery techniques can be used to allow PDRs and other *TV-Anytime* clients to discover *TV-Anytime* NDR services. The relevant standards are UDDI [19] and WS-Inspection [21] that enable different but complementary modes of discovery. A provider can choose to enable neither, both, or only one of these mechanisms.

For a useful overview of these two standards please refer to: "The WS-Inspection and UDDI Relationship" [22].

5.4.1 Universal Description, Discovery and Integration (UDDI)

UDDI [19] allows PDRs, and other clients with Internet connectivity, to discover *TV-Anytime* NDR services without the client requiring any prior knowledge of the NDR service, nor any information from a unidirectional metadata service. NDR service providers may publish details of their *TV-Anytime* NDR service(s) to the UDDI Business Registry. Any client can then use a node in the UDDI Business Registry (which have well-known addresses) to browse and locate *TV-Anytime* NDR services.

The present document relies on the use of version 3 of the UDDI specification. Clients wishing to discover *TV-Anytime* web services using UDDI must conform to the behaviour described in the UDDI specification [19]. To assist NDR service providers in describing and categorizing their services, *TV-Anytime* has selected the UDDI tModel described in clauses 5.4.1.1 and 5.4.1.2 (see clause 1.6.4 of the UDDI specification [19] for a definition of "tModel").

5.4.1.1 *TV-Anytime* web service tModel for NDR services

This tModel is used when a business publishes details of their bindingTemplate structures to indicate an NDR web service compliant with the present document. Clients issuing UDDI inquiry requests can use this tModel key to find only web services that are *TV-Anytime* NDR services.

This following technical model represents a "*TV-Anytime* control_NDR tModel port" as described in clause 4.1.

Name: TV-Anytime-org:control_NDR_v10.

Description: *TV-Anytime* WSDL interface for control_NDR port.

UDDI Key (V3): uddi:TV-Anytime.org:control_NDR_v10.

Categorization: Specification, xmlSpec, soapSpec, wsdlSpec.

```
<tModel tModelKey="uddi:TV-Anytime.org:control_NDR_v10">
  <name>TV-Anytime-org:control_NDR_v10</name>
  <description xml:lang="en">TV-Anytime WSDL interface for control_NDR
  port</description>
  <overviewDoc>
    <overviewURL useType="text">
       http://Location_At_ETSI_Website/Filename_for_TS102822-9
    </overviewURL>
  </overviewDoc>
  <categoryBag>
    <keyedReference keyName="uddi-org:types:wsdl" keyValue="wsdlSpec"
       tModelKey="uddi:uddi.org:categorization:types"/>
    <keyedReference keyName="uddi-org:types:soap" keyValue="soapSpec"
       tModelKey="uddi:uddi.org:categorization:types"/>
    <keyedReference keyName="uddi-org:types:xml" keyValue="xmlSpec"
       tModelKey="uddi:uddi.org:categorization:types"/>
    <keyedReference keyName="uddi-org:types:specification"
 keyValue="specification" tModelKey="uddi:uddi.org:categorization:types"/>
  </categoryBag>
</tModel>
```

5.4.1.2 *TV-Anytime* categorization tModel for NDR services

These tModels allow an NDR service provider to categorize their services. The NDR service provider assigns the categories at the point of publication (see clause 5.4.1.3). This enables clients to issue more refined UDDI searches for NDR services. By categorizing an NDR service as richly and accurately as possible, an NDR service provider maximizes the possibility of the NDR service being discovered using UDDI.

The use of this taxonomy is mandatory.

This taxonomy does not make strong guarantees about the behaviour of any services that use them. Therefore, having discovered an NDR service, a client should always retrieve a service capability description of that service. In some cases, this involves no extra steps as the capability description will be included inside the bindingTemplate for that operation.

5.4.1.2.1 *TV-Anytime* serviceURL categorization system

The serviceURL tModel is used to represent the content delivery services (e.g. channels) for which this NDR service provides content recording.

This serviceURL tModel is already defined in TS 102 822-2 [2].

Name: TV-Anytime-org:serviceURL.

Description: Category system for the content services handled by an NDR service.

UDDI Key (V3): uddi:TV-Anytime.org:serviceURL.

Valid values: A valid value must comply with the rules defined in the Metadata Specification [12] for the serviceURL element in the ServiceInformationTable.

Example usage: A client searches for *TV-Anytime* recording capability on a particular content service.

```
  <tModel tModelKey="uddi:TV-Anytime.org:serviceURL">
    <name>TV-Anytime-org:serviceURL</name>
    <description xml:lang="en">Category system for the content services handled by a NDR
service</description>
    <overviewDoc>
      <overviewURL useType="text">
http://Location_At_ETSI_Website/Filename_for_TS102822-9
      </overviewURL>
    </overviewDoc>
    <categoryBag>
      <keyedReference keyName="uddi-org:types:categorization"
  keyValue="categorization" tModelKey="uddi:uddi.org:categorization:types"/>
      <keyedReference keyName="uddi-org:types:unchecked"
  keyValue="unchecked" tModelKey="uddi:uddi.org:categorization:types"/>
    </categoryBag>
  </tModel>
```

5.4.1.2.2 Other categorizations

The other categorizations provided by Universal Business Registry are not used for NDR Service discovery.

5.4.1.3 Publishing a *TV-Anytime* NDR service

A *TV-Anytime* NDR service provider can publish details of their service to any node in the UDDI Business Registry. The manner in which this is done will depend upon the operator of that node (see the UDDI specification [19]).

An example of the publication process can be found in annex A.

A businessService is created for each NDR service that needs to be registered by that business. The businessService element contains a bindingTemplate for the binding offered by that NDR service (control_NDR).

When publishing a control_NDR operation, it is recommended that the instanceParms element (inside the tModelInstanceInfo) contains a service capability description. This allows the client to acquire the service capability description of the NDR service (and so determine its usefulness), without having to issue a describe_*X* request. Since the size of the instanceParms element is restricted, the service capability description will sometimes need truncating, in which case the capability description must remain schema valid.

5.4.2 Web Services Inspection language (WS-Inspection)

A *TV-Anytime* web server may declare the presence of its NDR service using WS-Inspection [21]. This allows clients to discover service descriptions (i.e. WSDL implementation definitions) for the web services available on that website. The WS-Inspection file may also lead to the discovery of other types of web services, as well as *TV-Anytime* NDR services available on other web sites.

It is recommended that each description element use a WSDL extensibility reference as follows:

- the endpointPresent attribute should be set to "true" (since a client is looking for existing services, and not abstract service definitions);
- an implementedBinding element should be included for each portType offered by the *TV-Anytime* service. In this way, the client can establish whether the corresponding service actually offers *TV-Anytime* ports and, if so, which portTypes are present, without having to download and parse the WSDL implementation description.

An example WS-Inspection file, along with its corresponding WSDL implementation definition, can be found in annex B.

To assist a client in finding a WS-Inspection file, clause 6.1 of the WS-Inspection specification [21] states that the document may have a well-known name (inspection.wsil) and be placed at a "common entry point" of the web-site. Because the term "common entry point" is vague, *TV-Anytime* has defined the following rule to make it easier for embedded clients to retrieve a WS-Inspection document.

- An NDR service provided by a web server with machine name <hostname>, which wishes to provide a WS-Inspection file, should place the document at the root of its web server. As a consequence, an HTTP GET request to http://<hostname>/inspection.wsil will retrieve the file if it exists.

5.5 NDR subscription

The "SubscriptionURL" in the control_NDR service capability description provides a way to access a server where the client can get information about the NDR service conditions such as storage capacity, recording allowance, storage duration and so on.

In addition to this, the subscription server allows the client to subscribe to the NDR service.

After NDR service description the client is invited to click on a link to get a file containing subscription information and in particular the "SubscriptionId" which is to be used later in the "RecordRequest" commands.

The "Content-Type" in the HTTP header of the response shall contain the following MIME type to launch the TVA recorder program: "application/x-tva-NDR-subscription".

The file extension "nst" is allocated to the NDR Subscription table.

The structure of the NDR Subscription table is as follows.

```
<xs:element name="NDR_Subscription_data" type="ndr:NDR_Subscription_dataType">
</xs:element>
<xs:complexType name="NDR_Subscription_dataType">
  <xs:sequence>
    <xs:choice>
      <xs:element name="Name" type="mpeg7:TextualType" maxOccurs="unbounded"/>
      <xs:element name="SubscriptionURL" type="anyURI"/>
    </xs:choice>
    <xs:element name="SubscriptionId" type="string"/>
  </xs:sequence>
</xs:complexType>
```

Name	Definition
NDR_Subscription_data	An element describing the subscription.
NDR_Subscription_dataType	A complex type describing the subscription.
Name	Same element as in the NDR service capability description.
SubscriptionURL	Same element as in the NDR service capability description.
SubscriptionId	This mandatory element contains the identification of the subscription made by the client to the referred NDR Service. This "SubscriptionId" is to be used in the "RecordRequest" commands.

Annex A (normative): Formal definition of NDR services

This annex provides a WSDL [20] interface definition for all *TV-Anytime* NDR services. WSDL defines a number of terms used to describe web services. The way in which these relate to *TV-Anytime* NDR services is given below:

- **Operation.** All *TV-Anytime operations* are request-response based, so can be thought of as a type of remote procedure call. An example of a *TV-Anytime operation* is control_NDR or describe_control_NDR.
- **PortType.** A collection of *operations*. When given a *binding* and a concrete endpoint the *portType* is known as a *port*. All *operations* in a given *port* must be present (i.e. it is not possible to offer only some of the *operations* in a *port*) and must have the same *binding*. The *TV-Anytime* Forum defines a *portType* for the NDR control (control_NDR), which has two *operations*, the basic functionality (control_NDR) and the corresponding describe *operation* (describe_control_NDR).
- **Binding.** A particular protocol binding (e.g. SOAP or HTTP GET) for a *portType*. There may be more than one *binding* for each *portType*. Each one is a different *port* and offers an alternative means for accessing the same *portType*. A *TV-Anytime* server could choose to provide other *bindings*, such as an HTTP POST implementation. By mandating a SOAP *binding*, a minimum level of interoperability is guaranteed.
- **Service.** A family of WSDL *ports* that are related in some way. A *TV-Anytime* NDR service contains a control_NDR port. In practice, most *TV-Anytime* servers will have just one WSDL *service*.

The following document is a WSDL interface definition that defines the behaviour of all *TV-Anytime* defined NDR web services. It plays two roles:

- The WSDL interface formally specifies the inputs, outputs, encodings and transport bindings used by all *TV-Anytime* NDR web services. The definitions given correspond to the specification of *TV-Anytime* NDR services in the present document.
- NDR service providers wishing to provide WSDL implementation definitions for their NDR services can import this WSDL interface definition. Annex F gives an example of how such a WSDL implementation can be referenced from a WS-Inspection [21] description.

```
<definitions targetNamespace="urn:tva:transport_wsdl:2010"
xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tns="urn:tva:transport_wsdl:2010" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:ndr="urn:tva:ndr:2010" elementFormDefault="qualified"
attributeFormDefault="unqualified">
  <documentation>WSDL for TV-Anytime Recording Service</documentation>
  <import namespace="urn:tva:ndr:2010" location="tva_ndr_9_v141.xsd"/>
  <message name="control_NDR_Result">
    <part element="ndr:Control_NDR_Result" name="body"/>
  </message>
  <message name="control_NDR">
    <part element="ndr:Control_NDR" name="body"/>
  </message>
  <message name="describe_control_NDR_Result">
    <part element="ndr:describe_control_NDR_Result" name="body"/>
  </message>
  <message name="ErrorReportMessage">
    <part element="ndr:ErrorReport" name="body"/>
  </message>
  <message name="describe_control_NDR">
    <part element="ndr:describe_control_NDR" name="body"/>
  </message>
  <portType name="control_NDR_Port">
    <operation name="control_NDR">
      <input message="tns:control_NDR"/>
      <output message="tns:control_NDR_Result"/>
      <fault message="tns:ErrorReportMessage" name="error"/>
    </operation>
```

```
    <operation name="describe_control_NDR">
      <input message="tns:describe_control_NDR"/>
      <output message="tns:describe_control_NDR_Result"/>
      <fault message="tns:ErrorReportMessage" name="error"/>
    </operation>
  </portType>
  <binding name="control_NDR_SOAP" type="tns:control_NDR_Port">
    <documentation>TV Anytime control_NDR binding</documentation>
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="control_NDR">
      <soap:operation soapAction="control_NDR"/>
      <input>
        <soap:body parts="body" use="literal"/>
      </input>
      <output>
        <soap:body parts="body" use="literal"/>
      </output>
      <fault name="error">
        <soap:fault use="literal"/>
      </fault>
    </operation>
    <operation name="describe_control_NDR">
      <soap:operation soapAction="describe_control_NDR"/>
      <input>
        <soap:body parts="body" use="literal"/>
      </input>
      <output>
        <soap:body parts="body" use="literal"/>
      </output>
      <fault name="error">
        <soap:fault use="literal"/>
      </fault>
    </operation>
  </binding>
</definitions>
```

Annex B (informative): Examples of control_NDR operation's service capability descriptions

This annex gives some examples of the different types of functionality a control_NDR operation might offer in real-life *TV-Anytime* NDR service deployments.

B.1 Simple NDR service

This clause contains the service capability description of an NDR Service provider which provides a simple store-and-forward content recording service.

```
<?xml version="1.0" encoding="UTF-8"?>
<ndr:describe_control_NDR_Result xmlns:NDR="urn:tva:ndr:2010"
xmlns:tva="urn:tva:metadata:2010" xmlns:mpeg7="urn:mpeg:mpeg7:2008"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:tva:ndr:2010 tva_ndr_9_v141.xsd" serviceVersion="148">
  <Name xml:lang="en">Voila Recording Service</Name>
  <Description xml:lang="fr">Le magnetoscope réseau de Voila.fr</Description>
  <SubscriptionURL>http://www.voila.fr/subscription.php</SubscriptionURL>
  <ServiceURLList>
    <ServiceURL>dvb://1.2.a</ServiceURL>
    <ServiceURL>dvb://1.2.f</ServiceURL>
  </ServiceURLList>
</ndr:describe_control_NDR_Result>
```

B.2 NDR service with different conversion and delivery capabilities

This clause contains the service capability description of an NDR Service provider which provides content recording service with media format conversion features and different delivery protocols to be used to deliver the recorded contents.

```
<?xml version="1.0" encoding="UTF-8"?>
<ndr:describe_control_NDR_Result xmlns:ndr="urn:tva:ndr:2010"
xmlns:tva="urn:tva:metadata:2010" xmlns:mpeg7="urn:tva:mpeg7:2008"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:tva:ndr:2010 tva_ndr_9_v141.xsd" serviceVersion="148">
  <Name xml:lang="en">Voila Recording Service</Name>
  <Description xml:lang="fr">Le magnetoscope réseau de Voila.fr</Description>
  <SubscriptionURL>http://www.voila.fr/subscription.php</SubscriptionURL>
  <ServiceURLList>
    <ServiceURL>dvb://1.2.a</ServiceURL>
    <ServiceURL>dvb://1.2.f</ServiceURL>
  </ServiceURLList>
  <ConversionList>
    <ConversionOffer>
      <DeliveryMediaFormat>
        <tva:BitRate variable="false">3500000</tva:BitRate>
        <tva:AudioAttributes>
          <tva:Coding href="urn:mpeg:mpeg7:cs:AudioCodingFormatCS:2001:6">
            <tva:Name xml:lang="en">AMR</tva:Name>
          </tva:Coding>
        </tva:AudioAttributes>
        <tva:VideoAttributes>
          <tva:Coding href="urn:mpeg:mpeg7:cs:VisualCodingFormatCS:2001:3.1">
            <tva:Name xml:lang="en">MPEG-4 Visual Simple Profile</tva:Name>
```

```
          </tva:Coding>
        </tva:VideoAttributes>
      </DeliveryMediaFormat>
    </ConversionOffer>
  </ConversionList>
  <ProtocolSetList>
    <ProtocolSet>
      <DeliveryProtocol href="urn:tva:ndr:cs:DeliveryProtocolTypeCS:2005:udp"/>
      <ControlProtocol href="urn:tva:ndr:cs:ControlProtocolTypeCS:2005:rtsp"/>
    </ProtocolSet>
    <ProtocolSet>
      <DeliveryProtocol version="1.1"
href="urn:tva:ndr:cs:DeliveryProtocolTypeCS:2005:http"/>
    </ProtocolSet>
  </ProtocolSetList>
  <PlayWhileRecording>true</PlayWhileRecording>
  <SupportForFilteringAndSearchPreferences>true</SupportForFilteringAndSearchPreferences>
</ndr:describe_control_NDR_Result>
```

Annex C (normative): XML Schema for NDR operation

In this annex, we define the normative "Control_NDR", "Control_NDR_Result" and "NDR_Subscription" schemas. Instances of this schema are used to subscribe to NDR services and to control them.

C.1 NDR subscription and operation schema

This clause contains the NDR subscription and operation schema (file "tva_ndr_9_v121.xsd").

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="urn:tva:ndr:2010" xmlns:ndr="urn:tva:ndr:2010"
xmlns:mpeg7="urn:tva:mpeg7:2008" xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:tva="urn:tva:metadata:2010" xmlns:cr="urn:tva:ContentReferencing:2010"
elementFormDefault="qualified" attributeFormDefault="unqualified">
  <import namespace="urn:tva:metadata:2010" schemaLocation="tva_metadata_3-1_v161.xsd"/>
  <import namespace="urn:tva:mpeg7:2008" schemaLocation="tva_mpeg7_2008.xsd"/>
  <import namespace="urn:tva:ContentReferencing:2010"
schemaLocation="tva_content_referencing_4_v151.xsd"/>
  <annotation>
    <documentation xml:lang="en">NDR schema</documentation>
  </annotation>
  <!-- ############
  Description of the requests from client to NDR and responses from NDR to client
   ############ -->
  <!-- ############ Description of the record request ############ -->
  <complexType name="ContentIdType">
    <sequence>
      <element name="InstanceMetadataId" type="tva:InstanceMetadataIdType"
      minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
    <attribute name="CRID" type="tva:CRIDType" use="required"/>
  </complexType>
  <complexType name="DeliveryProtocolType">
    <complexContent>
      <extension base="tva:ControlledTermType">
        <attribute name="version" type="float" use="optional"/>
      </extension>
    </complexContent>
  </complexType>
  <complexType name="ControlProtocolType">
    <complexContent>
      <extension base="tva:ControlledTermType">
        <attribute name="version" type="float" use="optional"/>
      </extension>
    </complexContent>
  </complexType>
    <complexType name="ProtocolSetType">
    <sequence>
      <element name="DeliveryProtocol" type="ndr:DeliveryProtocolType" minOccurs="0"/>
      <element name="ControlProtocol" type="ndr:ControlProtocolType" minOccurs="0"/>
    </sequence>
  </complexType>
  <complexType name="RecordRequestType">
    <sequence>
      <element name="SubscriptionId" type="string"/>
      <element name="ContentId" type="ndrR:ContentIdType" minOccurs="0"/>
      <element name="Locator" type="cr:LocatorType" minOccurs="0" maxOccurs="unbounded"/>
      <element name="DeliveryMediaFormat" type="tva:AVAttributesType" minOccurs="0"/>
      <element name="ProtocolSet" type="ndr:ProtocolSetType" minOccurs="0"/>
      <element name="StartTime" type="dateTime" minOccurs="0"/>
      <element name="EndTime" type="dateTime" minOccurs="0"/>
```

```
      <element name="FilteringAndSearchPreferences"
type="mpeg7:FilteringAndSearchPreferencesType" minOccurs="0"/>
    </sequence>
  </complexType>
  <!-- ########### Description of the status request for a previously accepted record
request ########### -->
  <complexType name="RecordStatusType">
    <attribute name="requestId" type="string" use="required"/>
  </complexType>
  <!-- ########### Description of the cancel request for a previously accepted record
request  ########### -->
  <complexType name="RecordCancelType">
    <attribute name="requestId" type="string" use="required"/>
  </complexType>
  <element name="Control_NDR_Result" type="ndr:ControlNDRResultType"/>
  <complexType name="ControlNDRResultType">
    <choice>
      <element name="RecordRequestResult" type="ndr:RecordRequestResultType"/>
      <element name="RecordStatusResult" type="ndr:RecordStatusResultType"/>
      <element name="RecordCancelResult" type="ndr:RecordCancelResultType"/>
      <element name="RecordListRequestResult" type="ndr:RecordListRequestResultType"/>
    </choice>
  </complexType>
  <complexType name="Control_NDRType">
    <choice>
      <element name="RecordRequest" type="ndr:RecordRequestType"/>
      <element name="RecordStatus" type="ndr:RecordStatusType"/>
      <element name="RecordCancel" type="ndr:RecordCancelType"/>
      <element name="RecordListRequest" type="ndr:RecordListRequestType"/>
    </choice>
  </complexType>
  <element name="Control_NDR" type="ndr:Control_NDRType"/>

  <complexType name="describe_control_NDRType"/>
  <element name="describe_control_NDR" type="ndr:describe_control_NDRType"/>
  <annotation>
    <documentation xml:lang="en"> Control_NDR_Result schema </documentation>
  </annotation>
  <!-- ######## Description of the list of previously accepted record requests ######## -
->
    <complexType name="RecordListRequestType">
        <sequence>
            <element name="SubscriptionId" type="string"/>
        </sequence>
    </complexType>
    <complexType name="RecordListRequestResultType">
        <sequence>
            <element name="RecordRequestStatus"
                type="ndr:RecordStatusResultType" minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
        <attribute name="serviceVersion" type="unsignedInt" use="required"/>
    </complexType>
  <!-- ########### Description of the response to a record request ########### -->
  <complexType name="RecordRequestOKType">
    <sequence>
      <element name="RequestId" type="string"/>
      <element name="Time2Call" type="dateTime"/>
      <element name="RecordingCharge" type="tva:PriceType" minOccurs="0"/>
      <element name="ConservationDelay" type="duration" minOccurs="0"/>
    </sequence>
  </complexType>
  <simpleType name="RecordRequestErrorType">
    <restriction base="string">
      <enumeration value="unknownSubscriptionId"/>
      <enumeration value="unknownCRID"/>
      <enumeration value="unavailableServiceURL"/>
      <enumeration value="unavailableNDRService"/>
      <enumeration value="unknownDeliveryProtocol"/>
      <enumeration value="unknownControlProtocol"/>
```

```
      <enumeration value="unsupportedDeliveryProtocol"/>
      <enumeration value="unsupportedControlProtocol"/>
      <enumeration value="unknownOriginalMediaFormat"/>
      <enumeration value="unsupportedOriginalMediaFormat"/>
      <enumeration value="unknownDeliveryMediaFormat"/>
      <enumeration value="unsupportedDeliveryMediaFormat"/>
    </restriction>
  </simpleType>
  <complexType name="RecordRequestResultType">
    <sequence>
      <choice>
        <element name="RecordRequestOK" type="ndr:RecordRequestOKType"/>
        <element name="RecordRequestError" type="ndr:RecordRequestErrorType"/>
      </choice>
    </sequence>
    <attribute name="serviceVersion" type="unsignedInt" use="required"/>
  </complexType>
  <!-- ############ Description of the response to a record status request  ############ --
>
  <simpleType name="FailureType">
    <restriction base="string">
      <enumeration value="unknownRequest"/>
      <enumeration value="cancelledBroadcast"/>
      <enumeration value="cancelledByNDR"/>
    </restriction>
  </simpleType>
  <complexType name="RecordStatusResultType">
    <choice>
      <element name="Running">
        <complexType>
          <sequence>
            <element name="Time2Call" type="dateTime"/>
          </sequence>
        </complexType>
      </element>
      <element name="ContentAvailable">
        <complexType>
          <sequence>
            <element name="ContentURL" type="anyURI"/>
            <element name="ConservationDeadline" type="dateTime" minOccurs="0"/>
          </sequence>
        </complexType>
      </element>
      <element name="Failure" type="ndr:FailureType"/>
    </choice>
    <attribute name="requestId" type="string" use="required"/>
  </complexType>
  <!-- ############ Description of the response to a cancel request  ############ -->
  <simpleType name="RecordCancelResultType">
    <restriction base="string">
      <enumeration value="unknownRequest"/>
      <enumeration value="OK"/>
      <enumeration value="removedAfterRecording"/>
    </restriction>
  </simpleType>
  <!-- ############ Description of the general error reports  ############ -->
  <simpleType name="errorCodeType">
    <restriction base="string">
      <enumeration value="FatalError"/>
      <enumeration value="InvalidRequest"/>
      <enumeration value="Unsupported"/>
      <enumeration value="UnrecognizedVersion"/>
      <enumeration value="UnspecifiedError"/>
    </restriction>
  </simpleType>
  <complexType name="ErrorType">
    <sequence>
      <element name="Reason" type="mpeg7:TextualType" minOccurs="0"
maxOccurs="unbounded"/>
```

```
    </sequence>
    <attribute name="errorCode" type="ndr:errorCodeType" use="required"/>
  </complexType>
  <element name="ErrorReport" type="ndr:ErrorReportType"/>
  <complexType name="ErrorReportType">
    <sequence>
      <element name="Error" type="ndr:ErrorType" maxOccurs="unbounded"/>
    </sequence>
  </complexType>
  <!-- ############
  Description of the network digital recorder capabilities
  ############ -->
  <complexType name="ServiceURLListType">
    <sequence>
      <element name="ServiceURL" type="anyURI" maxOccurs="unbounded"/>
    </sequence>
  </complexType>
  <complexType name="ConversionListType">
    <sequence>
      <element name="ConversionOffer" type="ndr:ConversionOfferType"
maxOccurs="unbounded"/>
    </sequence>
  </complexType>
  <complexType name="ConversionOfferType">
    <sequence>
      <element name="OriginalMediaFormat" type="tva:AVAttributesType" minOccurs="0"/>
      <element name="DeliveryMediaFormat" type="tva:AVAttributesType" minOccurs="0"/>
    </sequence>
  </complexType>
  <complexType name="ProtocolSetListType">
    <sequence>
      <element name="ProtocolSet" type="ndr:ProtocolSetType" maxOccurs="unbounded"/>
    </sequence>
  </complexType>
  <complexType name="describe_control_NDR_ResultType">
    <sequence>
      <element name="Name" type="mpeg7:TextualType" minOccurs="0" maxOccurs="unbounded"/>
      <element name="Description" type="mpeg7:TextualType" minOccurs="0"
maxOccurs="unbounded"/>
      <element name="SubscriptionURL" type="anyURI" minOccurs="0"/>
      <element name="ServiceURLList" type="ndr:ServiceURLListType"/>
      <element name="ConversionList" type="ndr:ConversionListType" minOccurs="0"/>
      <element name="ProtocolSetList" type="ndr:ProtocolSetListType" minOccurs="0"/>
      <element name="PlayWhileRecording" type="boolean" default="false" minOccurs="0"/>
      <element name="SupportForFilteringAndSearchPreferences" type="boolean"
default="false" minOccurs="0"/>
    </sequence>
    <attribute name="serviceVersion" type="unsignedInt" use="required"/>
  </complexType>
  <element name="describe_control_NDR_Result"
type="ndr:describe_control_NDR_ResultType"/>
  <!-- ############
  Description of the subscription structure returned from the NDR Web site after
subscription to an NDR Service
   ############ -->
  <complexType name="NDR_Subscription_dataType">
    <sequence>
      <choice>
        <element name="Name" type="mpeg7:TextualType" maxOccurs="unbounded"/>
        <element name="SubscriptionURL" type="anyURI"/>
      </choice>
      <element name="SubscriptionId" type="string"/>
    </sequence>
  </complexType>
  <element name="NDR_Subscription_data" type="ndr:NDR_Subscription_dataType"/>
</schema>
```

C.2 Classification scheme for DeliveryProtocolType (DeliveryProtocolTypeCS.xml)

```
<?xml version="1.0" encoding="UTF-8"?>
<ClassificationScheme uri="urn:tva:ndr:cs:DeliveryProtocolTypeCS:2005">
  <annotation>Terms for recorded content delivery protocol</annotation>
  <Term termID="rtp">
    <Name xml:lang="en">rtp</Name>
    <Definition>Real-Time Transport Protocol</Definition>
  </Term>
  <Term termID="udp">
    <Name xml:lang="en">udp</Name>
    <Definition>User Datagram Protocol</Definition>
  </Term>
  <Term termID="http">
    <Name xml:lang="en">http</Name>
    <Definition>Hypertext Transfer Protocol</Definition>
  </Term>
  <Term termID="ftp">
    <Name xml:lang="en">ftp</Name>
    <Definition>File Transfer Protocol</Definition>
  </Term>
</ClassificationScheme>
```

C.3 Classification scheme for ControlProtocolType (ControlProtocolTypeCS.xml)

```
<?xml version="1.0" encoding="UTF-8"?>
<ClassificationScheme uri="urn:tva:ndr:cs:ControlProtocolTypeCS:2005">
  <annotation>Terms for recorded content control protocol</annotation>
  <Term termID="rtsp">
    <Name xml:lang="en">rtsp</Name>
    <Definition>Real Time Streaming Protocol</Definition>
  </Term>
</ClassificationScheme>
```

Annex D (informative): Examples of Control_NDR requests

This annex gives some examples of how the different NDR Service requests can be used to control the NDR operation.

D.1 Record request for a currently broadcast Content on a specific channel

To obtain content currently broadcast over a TV channel, the following command might be used.

```
<RecordRequest>
  <SubscriptionId>3456-4567-5677-4321</SubscriptionId>
  <Locator>dvb://1.4ee2.3fa;4f5</Locator>
</RecordRequest>
```

The response might be negative if the NDR service has no access to the TV channel indicated by "Locator".

```
<RecordRequestResult serviceVersion="12">
  <RecordRequestError>unavailableServiceURL</RecordRequestError>
</RecordRequestResult>
```

The response might be positive as follows.

```
<RecordRequestResult serviceVersion="12">
  <RecordRequestOK>
    <RequestId>45U753-452</RequestId>
    <Time2Call>2009-07-15T20:35:00.00</Time2Call>
    <RecordingCharge currency="EUR">2</RecordingCharge>
    <ConservationDelay>PT12H</ConservationDelay>
  </RecordRequestOK>
</RecordRequestResult>
```

D.2 Record request for content identified by a CRID

To obtain content recorded by an NDR service limiting the CRID resolution to a set of specific channels, the following command might be used.

```
<RecordRequest>
  <SubscriptionId>3456-4567-5677-4321</SubscriptionId>
  <ContentId CRID="crid://broadcaster.com/ajcnd"/>
  <Locator>dvb://1.4ee2.3fa;4f5</Locator>
  <Locator>dvb://1.3fc4.431;9bC</Locator>
</RecordRequest>
```

The response might be negative if the NDR service cannot find a location for the requested content on the specified TV channel(s).

```
<RecordRequestResult serviceVersion="12">
  <RecordRequestError>unknownCRID</RecordRequestError>
</RecordRequestResult>
```

D.3 Requesting status of a previously accepted record request

To obtain the status of a previously accepted record request, the following command might be used.

```
<RecordStatus  requestId="45U753-452"/>
```

Here is a possible response when the requested content is available from the NDR Service.

```
<RecordStatusResult requestId="45U753-452">
 <ContentAvailable>
  <ContentURL>ftp://ccett.fr/content1.mp2</ContentURL>
  <ConservationDeadline>2009-07-17T20:30:00.00</ConservationDeadline>
 </ContentAvailable>
</RecordStatusResult>
```

Here is a possible response when the requested content is not yet available.

```
<RecordStatusResult requestId="45U753-452">
 <Running>
  <Time2Call>2009-07-15T22:30:00.00</Time2Call>
 </Running>
</RecordStatusResult>
```

Here is a possible response when the Record Request Identifier is unknown.

```
<RecordStatusResult requestId="45U753-452">
 <UnknownRequest/>
</RecordStatusResult>
```

Here is a possible response when the content broadcast has been cancelled.

```
<RecordStatusResult requestId="45U753-452">
 <CancelledBroadcast/>
</RecordStatusResult>
```

Here is a possible response when the NDR Service has not been able to execute the previously accepted Record Request.

```
<RecordStatusResult requestId="45U753-452">
 <CancelledByNDR/>
</RecordStatusResult>
```

D.4 Requesting cancellation of a previously accepted record request

To cancel a previously accepted record request, the following command might be used.

```
<RecordCancel requestId="45U753-452"/>
```

The response might be negative if the Record Request Identifier provided in the record cancel request is unknown.

```
<RecordCancelResult>unknownRequest</RecordCancelResult>
```

The response might be positive as follows.

```
<RecordCancelResult>OK</RecordCancelResult>
```

Annex E (informative): Example of UDDI usage

E.1 Example publication of control_NDR operation

The NDR service provider registers the new operation using the UDDI save_binding publication API (assuming that the appropriate parent businessEntity and businessService structures have already been registered).

```
<save_binding xmlns="urn:uddi-org:api_v3">
  <bindingTemplate>
    <description xml:lang="en">Recording service</description>
    <accessPoint useType="endPoint">
            http://www.tf1.fr/services/enregistrement</accessPoint>
    <tModelInstanceDetails>
      <tModelInstanceInfo tModelKey="uddi:TV-Anytime.org:control_NDR">
        <instanceDetails>
          <instanceParms><![CDATA[
                <?xml version="1.0" encoding="utf-8"?>
                  <describe_control_NDR_Result serviceVersion="3"
                        xmlns="urn:tva:transport:2005">
                    <!-- see describe_control_NDR_result -->
                  </describe_control_NDR_Result>
                ]]></instanceParms>
        </instanceDetails>
      </tModelInstanceInfo>
    </tModelInstanceDetails>
    <categoryBag>
      <keyedReference tModelKey="uddi:TV-Anytime.org:serviceURL"
                      keyValue="dvb://1.2.a"/>
      <keyedReference tModelKey="uddi:TV-Anytime.org:serviceURL"
                      keyValue="dvb://1.2.f"/>
    </categoryBag>
  </bindingTemplate>
</save_binding>
```

The bindingTemplate structure contains a reference to the tModel for the control_NDR operation. In this way, the tModel behaves as a technical fingerprint that formally indicates the *TV-Anytime* compliance of the web service.

The categorization information allows a client to establish the following:

- The NDR service can record contents available on channels dvb://1.2.a and dvb://1.2.f.

E.2 Example search for *TV-Anytime* NDR service

Consider the example of an end-user with a newly purchased PDR trying to get a movie broadcast tonight on channel A while a football match is broadcast on channel B.

The PDR wishes to discover NDRs able to record contents broadcast on channel A.

The following search could be used to find appropriate bindings.

```
<find_binding xmlns="urn:uddi-org:api_v3">
  <tModelBag>
    <tModelKey>uddi:TV-Anytime.org:control_NDR_v10</tModelKey>
  </tModelBag>
  <categoryBag>
    <keyedReference tModelKey="uddi:TV-Anytime.org:serviceURL"
                    keyValue="dvb://1.2.a"/>
    <keyedReference tModelKey="uddi:TV-Anytime.org:serviceURL"
                    keyValue="dvb://1.2.b"/>
    <keyedReference tModelKey="uddi:TV-Anytime.org:serviceURL"
                    keyValue="dvb://1.2.c"/>
  </categoryBag>
</find_binding>
```

The data structure returned to the device will contain a list of bindingTemplate elements that satisfy the above query. The list can then be refined by the user (based on brand preferences, recommendations, languages used, etc.), or automatically by the PDR (based on the capability description, and other taxonomies provided in each bindingTemplate element).

TV-Anytime clients may also register their interest in a particular type of NDR service, by registering with a node using the subscription API (see clause 5.5 of the UDDI specification [19]). In this case, the same find_binding element above could be used in the subscriptionFilter of the subscription message, thus defining the types of services that the PDR is interested in being notified about.

Annex F (informative): Referencing a WSDL implementation description using WS-Inspection

The following WS-Inspection [21] file contains a reference to a *TV-Anytime* NDR service providing a control_NDR port.

www.dummy.com/inspection.wsil

```
<inspection xmlns="http://schemas.xmlsoap.org/ws/2001/10/inspection/"
    xmlns:wsdl="http://schemas.xmlsoap.org/ws/2001/10/inspection/wsdl/"
    xmlns:tns="urn:tva:transport_wsdl:2010">
 <service>
  <description referencedNamespace="http://schemas.xmlsoap.org/wsdl/"
           location="http://www.channel1.com/services/record/record.wsdl">
   <wsdl:reference endpointPresent="true">
    <wsdl:implementedBinding>tva:control_NDR_SOAP</wsdl:implementedBinding>
   </wsdl:reference>
  </description>
 </service>
</inspection>
```

The location attribute in the above description allows a client to download a WSDL implementation description.

http://www.dummy.com/services/record/record.wsdl

```
<definitions targetNamespace="http://example.com/tva"
    xmlns:tns="urn:tva:transport_wsdl:2010"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns="http://schemas.xmlsoap.org/wsdl/">
  <import namespace="urn:tva:transport_wsdl:2010"/>
      location=" http://www.channel1.com/services/record/TVAGeneralRecord.wsdl">
  <service name="Channel1RecordingService">
    <port name="control_NDR_Port" binding="tva:control_NDR_SOAP">
      <soap:address location="http://www.channel1.com/services/record"/>
    </port>
  </service>
</definitions>
```

The referenced WSDL implementation definition is simple, and allows a client to establish the URL of the *TV-Anytime* control_NDR port. Also, the technical version of the port is indicated via the namespace of the fully qualified binding name.

Annex G (normative): *TV-Anytime* description schemes and classification schemes

The *TV-Anytime* DSs listed in the present document have been aggregated into several **xsd files identified by the Description Schemes' names**, forming the reference documentation, contained in archive ts_10282209v010401p0.zip which accompanies the present document:

- tva_ndr_9_v141.xsd is the NDR subscription and operation schema defined in clause C.1.
- DeliveryProtocolTypeCS.xml is the Delivery Protocol Type Classification Scheme defined in clause C.2.
- ControlProtocolTypeCS.xml is the Control Protocol Type Classification Scheme defined in clause C.3.

In order to validate, tva_ndr_9_v141.xsd imports several description schemes from other parts of the *TV-Anytime* specification:

- tva_metadata_3-1_v161.xsd and tva_mpeg7_2008.xsd and can be found in ts_1028220301v010601p0.zip, which is a file attached to ts_1028220301v010601.
- tva_content_referencing_4_v151.xsd can be found in ts_10282204v010501p0.zip, which is a file attached to ts_10282204v010501.

Annex H (informative): Bibliography

- The Platform for Privacy Preferences 1.0 (P3P1.0) Specification, M. Marchiori et. al. http://www.w3.org/TR/P3P/
- IETF RFC 2119: "Key words for use in RFCs to Indicate Requirement Levels", S. Bradner. http://www.ietf.org/rfc/rfc2119.txt
- IETF RFC 2396: "Uniform Resource Identifiers (URI): Generic Syntax", T. Berners-Lee, R. Fielding, L. Masinter. http://www.ietf.org/rfc/rfc2396.txt
- Unicode Collation Algorithm, Unicode Technical Report #10, M. Davis, K. Whistler. http://www.unicode.org/unicode/reports/tr10
- Unicode Normalization Forms, Unicode Standard Annex #15, M. Davis, M. Dürst. http://www.unicode.org/unicode/reports/tr15

List of figures

History

Document history		
V1.1.1	January 2006	Publication
V1.2.1	November 2007	Publication
V1.3.1	May 2009	Publication
V1.4.1	July 2010	Publication