

1 Scope

This document is a companion to ABNT NBR 25608:2025 and its annexes, providing complementary implementation guidelines intended to clarify architectural intent and operational behavior. It does not define normative requirements, but describes recommended practices, supports consistent interpretation of the standard, and promotes interoperable TV 3.0 Application-oriented Platform (AoP) implementations.

2 References

The following documents are cited in the text in such a way that their contents, in whole or in part, constitute requirements for this document. For dated references, only the editions cited apply. For undated references, the most recent editions of that document (including amendments) apply.

ABNT NBR 15603-2:2023, *Digital terrestrial television – Multiplexing and service information (SI) – Part 2: Data structure and definition of basic information of SI*

ABNT NBR 15603-2:2026, *Digital terrestrial television – Multiplexing and service information (SI) – Part 2: Data structure and definition of basic information of SI (under revision)*

ABNT NBR 15606-2, *Digital terrestrial television – Data coding and transmission specification for digital broadcasting – Part 2: Ginga-NCL for fixed and mobile receivers – XML application language for application coding*

ABNT NBR 15606-7, *Digital terrestrial television – Data coding and transmission specification for digital broadcasting – Part 7: Ginga-NCL – Operational Guidelines for ABNT NBR 15606-2 and ABNT NBR 15606-5*

ABNT NBR 15606-11, *Digital terrestrial television – Data coding and transmission specification for digital broadcasting – Part 11: Ginga CC WebServices – Ginga Common Core WebServices specification*

ABNT NBR 15606-12, *Digital terrestrial television – Data coding and transmission specification for digital broadcasting – Part 12: Ginga CC WebServices – Operational guidelines for ABNT NBR 15606-11*

ABNT NBR 25602, *TV 3.0 – Transport Layer*

ABNT NBR 25608:2025, *TV 3.0 – Application Coding*

IETF RFC 8216, *HTTP Live Streaming*

ISO/IEC 23009-1, *Information technology — Dynamic adaptive streaming over HTTP (DASH) — Part 1: Media presentation description and segment formats*

3 Terms and Definitions

For the purposes of this Document, the terms and definitions specified by ABNT NBR 25608:2025 apply. Additionally, the following terms and definitions apply.

No further terms and definitions are included in this Document.

4 Abbreviations

For the purposes of this Document, the following abbreviations apply.

ABNT	<i>Associação Brasileira de Normas Técnicas</i>
AMM	<i>Audience Measurement Manager</i>
AoP	<i>Application-oriented Platform</i>
API	<i>Application Programming Interface</i>
ATSC	<i>Advanced Television Systems Committee</i>
BALD	<i>Broadcaster Application Location Description</i>
BAM	<i>Bootstrap Application Manifest</i>
BAMT	<i>Bootstrap Application Manifest Table</i>
DASH	<i>Dynamic Adaptive Streaming over HTTP</i>
DPV	<i>Data Privacy Vocabulary</i>
DTV	<i>Digital TeleVision</i>
ECG	<i>Electronic Content Guide</i>
EIT	<i>Event Information Table</i>
EPG	<i>Electronic Programming Guide</i>
HLS	<i>HTTP Live Streaming</i>
HTML5	<i>HyperText Markup Language version 5</i>
HTTP	<i>HyperText Transfer Protocol</i>
HTTPS	<i>HTTP Secure</i>
MPD	<i>Media Presentation Description</i>
MPEG	<i>Moving Picture Experts Group</i>
NCL	<i>Nested Context Language</i>
NRT	<i>Non-Real Time</i>
PBDS	<i>Private Base Data Structure</i>
PD	<i>Personal Data Categories</i>
PII	<i>Personal Identifiable Information</i>

PMP	<i>Persistent Media Player</i>
PRP	<i>Proprietary Receiver Platform</i>
PRRD	<i>Privacy Record Request Description</i>
ROUTE	<i>Real-Time Object Delivery over Unidirectional Transport</i>
RP	<i>Receiver Platform</i>
SBTVD	<i>Sistema Brasileiro de TV Digital</i>
SDT	<i>Service Description Table</i>
SLS	<i>Service Layer Signaling</i>
SVG	<i>Scalable Vector Graphics</i>
UI	<i>User Interface</i>
UX	<i>User eXperience</i>
URI	<i>Uniform Resource Identifier</i>
URL	<i>Uniform Resource Locator</i>
UTC	<i>Coordinated Universal Time</i>
W3C	<i>World Wide Web Consortium</i>
XML	<i>eXtensible Markup Language</i>
XSD	<i>XML Schema Definition</i>

5 Introduction

This document is a companion to ABNT NBR 25608:2025 and its annexes, providing complementary implementation guidelines intended to clarify architectural intent and operational behavior. It does not define normative requirements, but describes recommended practices, supports consistent interpretation of the standard, and promotes interoperable TV 3.0 platform implementations.

The operational guidelines addressing the architectural aspects of TV 3.0 Application Coding and its Application-Oriented Platform are presented in the main body of this document. Guidelines related to specific technologies and regulatory aspects employed in TV 3.0 Application Coding are provided in the annexes that follow.

- Annex A contains the guidelines on NCL 4.0 and TV 3.0 Ginga-NCL;
- Annex B contains the guidelines on TV 3.0 Ginga-HTML5;
- Annex C contains the guidelines on TV 3.0 WebServices;
- Annex D contains the guidelines on NCLua for TV 3.0 Ginga-NCL; and
- Annex E contains the guidelines on Regulatory Compliance Requirements for Brazil.

6 TV 3.0 application coding architecture

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section 6 apply.

7 TV 3.0 Ginga Common Core

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section 7 apply.

7.1 Over-the-air physical layer information

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section 7.1 apply.

7.2 Transport layer metadata

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section 7.2 apply.

7.3 DASH streams

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section 7.3 apply.

7.4 HLS streams

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section 7.4 apply.

7.5 Non-Real Time Files

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section 7.5 apply.

7.6 Persistent Media Player

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section 7.6 apply.

The Persistent Media Player (PMP) is a core component of the TV 3.0 AoP, defined in ABNT NBR 25608:2025, Section 7.6. Its control APIs are specified in TV 3.0 WebServices (Annex C of ABNT NBR 25608:2025).

In a TV 2.X/TV 3.0 Receiver Platform (RP), Bootstrap Applications associated with TV 2.X services may need to instruct the PMP to start playing content, as required by BpAFR-07 of ABNT NBR 25608:2025, Subsection 8.4.2. The implementation is responsible for ensuring that the PMP provides appropriate support for both TV 2.X and TV 3.0 services, while maintaining proper environment isolation as described in Section 11 of this Document. Broadcaster Applications transmitted as part of TV 2.X services should access PMP functionality exclusively through TV 2.X-compliant interfaces, without requiring modifications to the application code or awareness of TV 3.0-specific features.

7.7 Platform capabilities and attributes

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section 7.7 apply.

7.8 Sensory Effect Renderers

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section 7.8 apply.

7.9 Interaction Modules

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section 7.9 apply.

7.10 Audience Measurement Manager

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section 7.10 apply.

7.10.1 General aspects

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section 7.10.1 apply.

As established in ABNT NBR 25608:2025, Section 7.10.1, the use of the audience measurement data collection API depends on the viewer's prior agreement, expressed through the privacy settings, for a purpose declared in the PRRD with `purpose@type="dpv:ServiceUsageAnalytics"` and applicable lawful basis. During the audience measurement session creation, this precondition is enforced by the Audience Measurement Manager (AMM). A session for which the required agreement cannot be established does not proceed to active data collection (see ABNT NBR 25608:2025, Figure 3).

This guideline describes platform-level behavior established by the standard. It does not make any determination regarding the personal or non-personal nature of audience measurement data, which remains the responsibility of the broadcaster under the applicable data protection legislation.

7.10.2 Audience measurement data specification for a data collection session

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section 7.10.2 apply.

AMM Report: TV30AoP-foregroundAppId property

The `TV30AoP-foregroundAppId` property, defined in `TV30AoPStatus` category (see ABNT NBR 25608:2025 Table 4), identifies the TV 3.0 application or AoP component currently running in the foreground during the audience measurement session. For interoperability purposes in AMM reports, implementations should adopt recommended identifiers for `TV30AoP-foregroundAppId` values.

The recommended format is harmonized with existing identifier conventions already adopted in TV 3.0 Application Coding specifications (e.g., NCL, NCL360, NCLua, and WebServices APIs). Therefore, AoP

component identifiers should use lowercase strings with hyphen-separated tokens. The prefix "aop-" should be used for TV 3.0 AoP component identifiers. Table 7.1 lists the recommended identifier values:

Table 7.1 Recommended TV30AoP-foregroundAppId identifiers

Foreground Application / Component	Recommended TV30AoP-foregroundAppId Identifier
Bootstrap Applications	aop-bootstrap-application-<BAM@globalServiceId>
Broadcaster Applications	aop-broadcaster-application-<BALD.EntryPackage@appId>
Application Catalog	aop-application-catalog
Access Panel	aop-access-panel
Privacy Manager	aop-privacy-manager
EPG	aop-epg
ECG	aop-ecg

Additional AoP component identifiers should follow the same naming convention.

AMM Report: TV30AoP-pmplayer.contentID property

The `TV30AoP-pmplayer.contentID` property defined in the `TV30AoPStatus` category (see ABNT NBR 25608:2025, Table 4) identifies the content associated with the PMP during the audience measurement session. For content associated with a TV 3.0 Service Guide entry, its value is obtained from the corresponding `Content` fragment of the Service Guide. The TV 3.0 Service Guide data model is defined in accordance with ATSC A/332:2025-10, as specified in ABNT NBR 25602:2025, Annex D, and its information is made available through the ESG associated with the DTV service, which Broadcaster Applications can access through the TV 3.0 WebServices APIs defined in ABNT NBR 25608:2025, including the one specified in its Table C.23.

The receiver identifies the applicable `Content` fragment through the association between the `Service`, `Schedule` and `Content` fragments: the `Schedule` fragment references the service through `ServiceReference/@idRef` and the content through `ContentReference/@idRef`, the latter resolving to the `Content/@id` attribute of the corresponding `Content` fragment. Two identifiers are defined for that fragment:

- `Content@id` identifies the `Content` fragment itself, is used to establish references between Service Guide fragments;
- `Content@globalContentID` identifies the content described by the `Content` fragment and is optional in the Service Guide data model.

The value reported in `TV30AoP-pmplayer.contentID` should be selected from these two identifiers according to their availability in the applicable `Content` fragment. When `Content@globalContentID` is present, its value should be used, as it identifies the content itself and is suitable for aggregation across services and providers. When `Content@globalContentID` is absent, the value of `Content@id` should be used instead. Although the two identifiers have different semantics, both are defined as globally unique. Moreover, `Content@id` is assigned by the entity that provides the Service Guide, so reporting `Content@id` still allows that entity to relate the audience measurement records to its own content. Consequently, a fixed default value should not be used whenever a `Content` fragment applicable to the content being presented can be resolved. In this arrangement, the entity that provides the Service Guide is responsible for supplying the content identifiers, and the receiver is responsible for resolving the applicable `Content` fragment and reporting the corresponding identifier.

7.11 Private Base Manager

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section 7.11 apply.

8 Application Catalog

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section 8 apply.

8.1 General aspects

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section 8.1 apply.

8.2 Functional requirements

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section 8.2 apply.

ACFR-08 is intended to prevent the Application Catalog from removing, hiding, or reordering Bootstrap Application icons as a result of the receiver's own autonomous decisions, whether based on usage criteria such as access frequency or usage time, or when the RF channel carrying a Bootstrap Application was not detected during the most recent signal rescan. This reading is coherent with ACFR-21, under which the Bootstrap Applications of RF channels not detected during a rescan are retained, and with ACFR-28, under which, when the limit established in ACFR-27 is exceeded, Bootstrap Applications whose signaling ceased to be detected are given lower priority for retention.

A distinct situation arises during a scan or rescan when an RF channel remains present and is positively detected, but the broadcaster has deliberately changed the bootstrap signaling of its services, for example by changing the number of category 30 services (see ABNT NBR 25602) or by replacing them, which also changes the set of icons to be presented. Because the channel is detected, this case does not correspond to the "channel not detected" condition addressed by ACFR-21, and the retention established there is not triggered. Since a Bootstrap Application is, by definition, what a DTV service announces and parameterizes through the corresponding transport layer metadata, during a scan or rescan the Application Catalog is expected to reflect the bootstrap configuration currently detected on each transport, adding and removing Bootstrap Applications accordingly. Being driven by the broadcaster's signaling rather than by an autonomous decision of the receiver, this update does not fall under the restriction of ACFR-08. Consequently, the third and fourth priorities of ACFR-28 are understood to apply to Bootstrap Applications retained under ACFR-21, and not to those deliberately withdrawn by the broadcaster from a transport that is still detected.

As established in ACFR-37, the Application Catalog persists the information of each BAM, which may become outdated relative to the current signaling. Therefore, during a scan or rescan, whenever the current BAMT is received for a detected transport, it is recommended that the TV 3.0 AoP reconcile the persisted entries for that transport with the currently signaled Bootstrap Applications, refreshing the entries that are still present from their corresponding BAM, while preserving viewer-defined preferences such as favorite status and manual ordering, and removing the entries that are no longer carried. This ensures that, after a scan or rescan, the Application Catalog does not retain stale or non-functional entries and that persisted information, such as media URLs, names, icons, or grouping information, reflects the Bootstrap Application configuration currently signaled by the broadcaster.

8.3 Non-functional requirements

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section 8.3 apply.

8.4 Bootstrap Applications

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section 8.4 apply.

In TV 2.X/TV 3.0 receivers processing TV 2.X transmissions, when a BAMT is detected, as defined in ABNT NBR 15603-2:2026, the information contained in this table should be used to derive the Bootstrap Application Manifest data, as described in ABNT NBR 25608:2025 (Section 8.4.4).

When no BAMT is present in a TV 2.X transmission, the Bootstrap Application Manifest data is derived according to the following recommended rules:

- `BAMT@groupingType`: set to "mandatoryGrouping", corresponding to the default value defined in BAMT XML schema included in the compressed ZIP folder associated with ABNT NBR 25608:2025.
- `BAM@appName`: obtained from Service Descriptor in SDT, as specified in ABNT NBR 15603-2:2023, Section 8.3.13.
- `BAM@appDescription`: set to an empty string.
- `BAM@backgroundColor`: defined by the implementer of the standard.
- `BAM@foregroundColor`: defined by the implementer of the standard.
- `BAM/appIcon`: defined by the implementer of the standard.
- `BAM/bannerIcon`: defined by the implementer of the standard.

For additional guidelines on Bootstrap Application Manifests under provisions specific to the Brazilian regulatory context, see Section E.1.

8.4.1 Introduction

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section 8.4.1 apply.

8.4.2 Functional requirements

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section 8.4.2 apply.

8.4.3 Non-functional requirements

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section 8.4.3 apply.

8.4.4 Bootstrap Application Manifest

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section 8.4.4 apply.

In ABNT NBR 25608:2025, Table 6, the description for the `fallbackMediaURLs` element contains the following residual text related to the former semantics of `initialMediaURLs`, which is now disregarded:

“List of default linear service URLs to be used if the Bootstrap Application is launched without the `initialMediaURL` parameter. In this case, the Bootstrap Application instructs the Persistent Media Player to play the first functional linear service URL in the list order.”

The Portuguese version is not affected.

8.5 Electronic Programming Guide (EPG)

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section 8.5 apply.

For operational guidelines regarding EPG harmonization of TV 2.X/TV 3.0 receiver platform (RP), refer to Section 11 of this Document.

8.6 Electronic Content Guide (ECG)

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section 8.6 apply.

8.7 Profile Manager

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section 8.7 apply.

ABNT NBR 25608:2025, Section 8.7 and Table 7 define the basic attributes of the TV viewer profile. The profile identifier (`id`), a UUID that uniquely identifies the profile within the receiver, is not editable by the viewer and is assigned automatically by the TV 3.0 AoP. The `nickname` is a human-readable label and is not required to be unique. When the viewer does not set one, the AoP provides a default value that is distinct within the receiver (for example, "Viewer 1"), so that profiles created without viewer personalization remain distinguishable to the viewer. The remaining basic attributes carry the default values defined in ABNT NBR 25608:2025 Table 7 (for example, `parentalControl` and `closedCaptioning` default to false).

As a result, a valid and fully functional default viewer profile exists even when the viewer performs no personalization. For the purposes of this Document, the "default" term denotes a profile populated solely with the default attribute values of ABNT NBR 25608:2025 Table 7, with `id` and `nickname` assigned by the AoP. It denotes the absence of viewer-provided personalization and not the anonymization of the profile, since the profile still carries an `id`, which is used as the `viewer-id` in audience measurement (see ABNT NBR 25608:2025, Table 4) and as the `piiPrincipalId` in privacy records (see Section 8.8.3 of this Document).

From the perspective of the TV 3.0 WebServices APIs, a default profile is not distinguishable from a personalized one. In particular, the privacy management workflow and the AMM precondition described in this Document (see 7.10.1) apply identically regardless of whether the active profile has been personalized. The degree of personalization of the active profile does not, by itself, change the privacy or audience measurement behavior established by the standard. The maintenance of an active viewer profile is presupposed elsewhere in the standard, for example in the profile selection performed whenever the viewer activates the TV 3.0 AoP (see ABNT NBR 25608:2025, Annex A).

8.8 Privacy Manager

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section 8.8 apply.

8.8.1 General aspects

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section 8.8.1 apply.

8.8.2 Privacy record request

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section 8.8.2 apply.

The Privacy Record Request Description (PRRD) is the mechanism through which DTV services formally declare their intended processing of viewers' personal data. PRRD is defined as an XML fragment embedded in the Service Layer Signaling (SLS) (see ABNT NBR 25602), following the fragment-based signaling model established in ATSC A/331 and adopted within the TV 3.0 architecture. As a signaling fragment, PRRD is transmitted as part of the broadcast flow and interpreted directly by TV 3.0 AoP, independently of Broadcaster Applications' logic. All elements and attributes discussed in this guideline therefore correspond to XML constructs defined within the PRRD fragment.

Broadcasters that do not engage in the processing of viewers' personal data are not required to signal a Privacy Record Request Description (PRRD). In such cases, the absence of a PRRD implicitly indicates that no personal data is collected, accessed, or processed within the scope of the associated TV 3.0 service. This determination is made by the broadcaster based on its own interpretation of the applicable data protection legislation, the nature of the service offered, and the categories of data involved. This exemption simplifies signaling and platform behavior for services that rely exclusively on broadcast delivery and do not invoke sensitive APIs. When no PRRD is announced, TV 3.0 AoP does not invoke the Privacy Manager workflow for that service. Should a broadcaster later introduce any form of personal data processing, the corresponding PRRD should then be signaled to accurately declare the new processing context and enable viewer awareness and control.

The PRRD specification builds upon established international standards rather than redefining privacy semantics. ISO/IEC 27560 [1] provides the conceptual foundation for structured records and receipts documenting data processing decisions, while the W3C Data Privacy Vocabulary (DPV) [2] and Personal Data Categories (PD) [3] vocabularies provide standardized identifiers for lawful bases, purposes, personal data categories, and actor roles. These vocabularies are reused as semantic building blocks and integrated into DTV service signaling and platform-level interpretation. This approach ensures clarity and machine-readability while addressing the specific requirements of personalized experiences under TV 3.0.

At a conceptual level, PRRD represents a service-scoped, machine-readable declaration of data processing intentions issued by the broadcaster. Its role is to establish the processing context that allows the Privacy Manager to determine whether viewer interaction is required, how privacy management interfaces shall be rendered, and which enforcement rules shall apply once decisions are expressed. PRRD precedes and conditions the generation of privacy records and privacy receipts; it does not record viewer choices, but instead defines the purposes, lawful bases, and data categories against which those choices will later be expressed and enforced.

This declarative role is concretely realized through explicit identification and versioning at the PRRD level. Each PRRD instance is uniquely identified by the `schemaVersion` attribute of the root PRRD element. The Privacy Manager uses this identifier to distinguish between new declarations and previously processed ones. Any modification to declared purposes, lawful bases, personal data categories, recipients, retention parameters, or other semantically relevant elements should result in a new version identifier. This mechanism ensures that

viewer decisions are always associated with the exact processing context in effect at the time of interaction and provides a deterministic trigger for re-evaluating privacy choices whenever that context changes.

An illustrative PRRD instance demonstrating multilingual declarations, multiple purposes with different lawful bases, associated personal data categories, and participating parties is provided in Listing 1. This example is cited throughout the present guideline solely to support understanding of PRRD structure and semantics. The specific values used in the example do not prescribe or recommend particular processing practices.

PRRD organizes its content through one or more `piiProcessing` elements, which serve as the top-level containers for viewer-facing privacy information. Each `piiProcessing` element corresponds to a specific language, identified by `piiProcessing@language`, and encapsulates all textual and presentational content required to render the privacy management interface in that language. This includes references to the applicable privacy notice (`piiProcessing@privacyNotice`), a broadcast-accessible copy of that notice (`piiProcessing@broadcastNotice`), the secure endpoint for privacy record delivery (`piiProcessing@deliveryUrl`), and the initial disclaimer presented in simplified privacy views (`initialDisclaimer`). All `piiProcessing` elements within a PRRD are expected to be semantically equivalent, differing only in language, so that the same processing intentions are consistently expressed across locales, as illustrated in Listing 1. Additional guidelines regarding the `piiProcessing@broadcastNotice` attribute can be found in Section 8.8.7.

Within each `piiProcessing` block, processing intentions are declared through one or more `purpose` elements. Each `purpose` represents a distinct data processing activity and is uniquely identified by the `purpose@id` attribute. The textual content of the `purpose` element describes the processing activity in natural language, in the language specified by the enclosing `piiProcessing` element. This description should be sufficiently detailed and unambiguous to enable the viewer to understand the intended processing of their personal data. Multiple purposes may be declared within the same PRRD, and purposes do not share a single lawful basis.

Each `purpose` is semantically classified through the `purpose@type` attribute, which adopts terms from the W3C Data Privacy Vocabulary. Examples of DPV purpose types include, but are not limited to, `dpv:ServiceUsageAnalytics`, `dpv:ServiceOptimisation`, `dpv:PersonalisedAdvertising`, `dpv:TargetedAdvertising`, `dpv:AccountManagement`, `dpv:AgeVerification`, `dpv:PoliticalCampaign`, and `dpv:CommercialPurpose`. The purpose type complements the natural-language description and enables consistent interpretation, interface behavior, and enforcement logic at the platform level. The example in Listing 1 illustrates the use of different purpose types across consent-based and non-consent-based processing.

The legal justification for each `purpose` is declared exclusively through the `purpose@lawfulBasis` attribute. This attribute also adopts DPV terms, such as `dpv:Consent`, `dpv:LegitimateInterest`, `dpv:Contract`, `dpv:LegalObligation`, `dpv:PublicInterest`, `dpv:VitalInterest`, and `dpv:OfficialAuthorityOfController`. The distinction between consent-based and non-consent-based processing is expressed solely through the value of this attribute. The choice of lawful basis is the responsibility of the broadcaster and reflects its own legal interpretation, organizational practices, and the nature of the data involved. The PRRD mechanism itself is agnostic to particular lawful bases and does not impose legal interpretation, a principle reflected in the example provided in Listing 1.

For each declared `purpose`, PRRD specifies the personal data categories involved through one or more `pii` elements within the `piiInformation` structure. Each `pii` element combines a human-readable description with a standardized classification expressed via the `pii@type` attribute, which adopts terms from the W3C Personal Data Categories vocabulary. Examples include `pd:Profile`, `pd:DeviceBased`, `pd:UID`, `pd:Location`, `pd:TVViewingBehavior`, `pd:Behavioral`, `pd:Tracking`, and `pd:Preference`. The optionality of each data category is explicitly indicated (`pii@optional`), allowing broadcasters to distinguish between mandatory and optional data for a given purpose, as illustrated in Listing 1.

The `piiInformation` structure also includes contextual attributes relevant for transparency and rights awareness, such as storage locations, retention periods, applicable jurisdiction, third-party recipients, withdrawal mechanisms, and supervisory authorities. These elements should accurately reflect the broadcaster's data handling practices and provide viewers with meaningful contextual information regarding the processing of their personal data.

All entities involved in the declared processing are identified in the `partyIdentification` section. Each party is assigned a role using DPV actor types such as `dpv:DataController`, `dpv:DataProcessor`, `dpv:Recipient`, `dpv:ServiceConsumer`, or `dpv:DataProtectionAuthority`. Identifiers declared in this section are referenced throughout PRRD to associate purposes and personal data categories with the corresponding actors, ensuring consistency between declared processing intentions and responsible entities. The structure and use of party identifiers are exemplified in Listing 1 too.

```
<?xml version="1.0" encoding="UTF-8"?>
<PRRD xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="tag:sbtvd.org.br,2025:XMLSchemas/TV30/AppSignaling/PRRD/1.0/PRRD-1.0-20250711.xsd"
  xmlns="tag:sbtvd.org.br,2025:XMLSchemas/TV30/AppSignaling/PRRD/1.0/"
  xmlns:pd="https://w3id.org/dpv/pd#" xmlns:dpv="https://w3id.org/dpv#"
  schemaVersion="PRRD-1.0-ABC-20251013">
  <!-- Privacy Record Request for broadcaster ABC, version 20251013 -->
  <!-- =====
  Portuguese version (pt-BR)
  ===== -->
  <piiProcessing privacyNotice="https://www.abc.tv/privacy/br/notice_20251013.html"
    language="pt-BR" broadcastNotice="http://tv30.abc.tv/appCtx1/privacy/notice_20251013br.html"
    deliveryUrl="https://privacy.abc.tv/record/submit">
  <initialDisclaimer>
    Este serviço de TV 3.0 processa dados pessoais para oferecer funcionalidades personalizadas.
    Revise as finalidades e selecione suas preferências.
  </initialDisclaimer>
  <purposes>
    <!-- Purpose 1: Consent-based (Advertising Personalization) -->
    <purpose id="br_p1" type="dpv:PersonalisedAdvertising" lawfulBasis="dpv:Consent">
      Exibição de anúncios personalizados com base no histórico de visualização.
      <piiInformation>
        <pii type="pd:TVViewingBehaviour" optional="true">Histórico de programas assistidos</pii>
        <pii type="pd:DeviceBased" optional="true">Identificadores do dispositivo</pii>
        <pii type="pd:UID" optional="true">Identificador publicitário</pii>
        <pii type="pd:Location" optional="true">Localização aproximada do espectador</pii>
        <piiControllers>
          <partyId>ABC-TV</partyId>
        </piiControllers>
        <storageLocations>BR</storageLocations>
        <retentionPeriod>P90D</retentionPeriod>
        <jurisdiction>BR</jurisdiction>
        <recipientThirdParties>
          <partyId>AdPartner123</partyId>
        </recipientThirdParties>
        <withdrawalMethod>TV30AoP-privacyManager</withdrawalMethod>
        <authorityParty>ANPD</authorityParty>
      </piiInformation>
    </purpose>
    <!-- Purpose 2: Legitimate interest (Service analytics) -->
    <purpose id="br_p2" type="dpv:ServiceOptimisation" lawfulBasis="dpv:LegitimateInterest">
      Coleta de métricas de uso para melhoria do serviço e estabilidade de transmissão.
      <piiInformation>
        <pii type="pd:DeviceBased" optional="false">
          Identificadores, configurações e telemetria do dispositivo</pii>
        <piiControllers>
          <partyId>ABC-TV</partyId>
        </piiControllers>
        <storageLocations>BR</storageLocations>
```

```

    <retentionPeriod>P180D</retentionPeriod>
    <jurisdiction>BR</jurisdiction>
    <recipientThirdParties>
      <partyId>AnalyticsProviderXYZ</partyId>
    </recipientThirdParties>
    <withdrawalMethod>TV30AoP-privacyManager</withdrawalMethod>
    <authorityParty>ANPD</authorityParty>
  </piiInformation>
</purpose>
</purposes>
</piiProcessing>
<!-- =====
English version (en-US)
===== -->
<piiProcessing privacyNotice="https://www.abc.tv/privacy/en/notice_20251013.html"
  language="en-US" broadcastNotice="http://tv30.abc.tv/appCtx1/privacy/notice_20251013en.html"
  deliveryUrl="https://privacy.abc.tv/record/submit">
  <initialDisclaimer>
    This TV 3.0 service processes personal data to provide personalized features.
    Review the purposes and select your preferences.
  </initialDisclaimer>
  <purposes>
    <!-- Purpose 1: Consent-based (Advertising Personalization) -->
    <purpose id="en_p1" type="dpv:PersonalisedAdvertising" lawfulBasis="dpv:Consent">
      Display of personalized advertisements based on viewing history.
      <piiInformation>
        <pii type="pd:TVViewingBehaviour" optional="true">Television viewing history</pii>
        <pii type="pd:DeviceBased" optional="true">
          Device identifiers used for profile association</pii>
        <pii type="pd:UID" optional="true">Advertising identifier used for targeted ads</pii>
        <pii type="pd:Location" optional="true">Approximate viewer location</pii>
        <piiControllers>
          <partyId>ABC-TV</partyId>
        </piiControllers>
        <storageLocations>BR</storageLocations>
        <retentionPeriod>P90D</retentionPeriod>
        <jurisdiction>BR</jurisdiction>
        <recipientThirdParties>
          <partyId>AdPartner123</partyId>
        </recipientThirdParties>
        <withdrawalMethod>TV30AoP-privacyManager</withdrawalMethod>
        <authorityParty>ANPD</authorityParty>
      </piiInformation>
    </purpose>
    <!-- Purpose 2: Legitimate interest (Service analytics) -->
    <purpose id="en_p2" type="dpv:ServiceOptimisation" lawfulBasis="dpv:LegitimateInterest">
      Collection of usage metrics to improve service quality and transmission stability.
      <piiInformation>
        <pii type="pd:DeviceBased" optional="false">
          Device identifiers, settings and telemetry for diagnostics</pii>
        <piiControllers>
          <partyId>ABC-TV</partyId>
        </piiControllers>
        <storageLocations>BR</storageLocations>
        <retentionPeriod>P180D</retentionPeriod>
        <jurisdiction>BR</jurisdiction>
        <recipientThirdParties>
          <partyId>AnalyticsProviderXYZ</partyId>
        </recipientThirdParties>
        <withdrawalMethod>TV30AoP-privacyManager</withdrawalMethod>
        <authorityParty>ANPD</authorityParty>
      </piiInformation>
    </purpose>
  </purposes>

```

```

</piiProcessing>
<!-- =====
Party Identification
===== -->
<partyIdentification>
  <party id="ABC-TV" type="dpv:DataController">
    <partyName>ABC Televisão S.A.</partyName>
    <partyAddress>Avenida das Comunicações, 1000 – São Paulo, SP – Brasil</partyAddress>
    <partyContact>mailto:privacidade@abc.tv</partyContact>
  </party>
  <party id="AdPartner123" type="dpv:DataProcessor">
    <partyName>AdPartner Brasil Ltda.</partyName>
    <partyAddress>Rua dos Anunciantes, 500 – Rio de Janeiro, RJ – Brasil</partyAddress>
    <partyContact>mailto:privacy@adpartner.com</partyContact>
  </party>
  <party id="AnalyticsProviderXYZ" type="dpv:DataProcessor">
    <partyName>Analytics Provider XYZ Ltda.</partyName>
    <partyAddress>Rua das Métricas, 42 – Curitiba, PR – Brasil</partyAddress>
    <partyContact>mailto:lgpd@analyticsxyz.com</partyContact>
  </party>
  <party id="ANPD" type="dpv:DataProtectionAuthority">
    <partyName>Autoridade Nacional de Proteção de Dados (ANPD)</partyName>
    <partyAddress>
      Setor Comercial Norte – SCN, Quadra 6, Conjunto A,
      Edifício Venâncio 3000, Bloco A, 9º andar – CEP 70716-900 – Brasília, DF – Brasil
    </partyAddress>
    <partyContact>mailto:protocolo@anpd.gov.br</partyContact>
  </party>
</partyIdentification>
</PRRD>

```

Listing 1. Example of a Privacy Record Request Description (PRRD) instance illustrating multilingual purpose declarations and lawful basis attribution.

Further guidance on data access, data minimization, and privacy-related practices in the TV 3.0 environment is provided in the *Data Access and Privacy Guide* [4], produced by the Intellectual Property Module of the SBTVD Forum. That guide elaborates recommended practices for data processing, user profile handling, and privacy management, with the objective of ensuring that only data strictly necessary for service operation and personalization is processed.

8.8.3 Privacy record

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section 8.8.3 apply.

8.8.4 Privacy receipt

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section 8.8.4 apply.

8.8.5 Privacy management graphical views

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section 8.8.5 apply.

The privacy management graphical views render information derived from the PRRD to support the viewer's understanding of the intended processing of PII and, when applicable, the viewer's data processing choices. The information displayed by the Privacy Manager is not expected to reproduce the full PRRD structure. Instead, implementations should render a representative subset of the privacy metadata that is sufficient to support the

interaction flows and the functional requirements defined in ABNT NBR 25608:2025, 8.8.5 and 8.8.6, respectively.

This guidance does not prescribe a specific layout, visual style, or complete set of information to be displayed. Implementations may render additional PRRD information, reorganize how the information is presented, summarize metadata, or adopt different UI and UX strategies, provided that the resulting behavior remains compliant with the requirements in ABNT NBR 25608:2025, 8.8.6 and the local regulation on privacy protection.

For the simplified privacy management view defined in PRMFR-03, applicable when the PRRD contains no consent-based purposes, the Privacy Manager should display the `piiProcessing.initialDisclaimer` description according to the current language, a link to the full privacy terms derived from `piiProcessing@privacyNotice`, and information identifying the current DTV service.

For the privacy management view defined in PRMFR-04, applicable when the PRRD contains at least one consent-based purpose, the Privacy Manager should display `piiProcessing.initialDisclaimer`, a link to the full privacy terms derived from `piiProcessing@privacyNotice`, and information identifying the current DTV service. The Privacy Manager uses `purposes.purpose@lawfulBasis` to determine whether consent-based interaction controls are required, including the “Accept All” and “Manage” viewers’ decision options.

For the detailed privacy management view defined in PRMFR-05, the Privacy Manager should list all declared purposes. Each purpose should be treated as the main unit of interaction, since viewer choices are associated with the concerned purpose in the resulting privacy record or privacy receipt. Each rendered purpose item should present a representative subset of information derived from the corresponding `purposes.purpose` element and its associated `purpose.piiInformation`.

A purpose item typically presents the purpose description, a list of the `piiInformation.pii` descriptions processed for that purpose, and a summarized indication of involved third-party recipients. Considering Listing 1, the purpose “Display of personalized advertisements based on viewing history” may be rendered with an indication that it uses “television viewing history”, “device identifiers used for profile association”, “advertising identifier used for targeted ads”, and “approximate viewer location”, together with a “Third parties” summary (e.g. the involvement of third parties or not, the number of third parties, etc) derived from the valid `partyId` values declared under `recipientThirdParties` for that purpose. The purpose “Collection of usage metrics to improve service quality and transmission stability” may be rendered with an indication that it uses “device identifiers, settings and telemetry for diagnostics”, together with the corresponding summarized indication of involved third-party recipients. Therefore, the main PRRD elements and attributes expected to be useful for each rendered purpose item are:

- `purpose`
- `purpose@lawfulBasis`
- `purpose.piiInformation.pii`
- `purpose.piiInformation.recipientThirdParties`

When the Privacy Manager is invoked to review or modify privacy settings, as described in PRMFR-08, PRMFR-09 and PRMFR-10, the same PRRD-derived information is used to render the corresponding privacy management view.

8.8.6 Functional requirements

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section 8.8.6 apply.

8.8.7 Delivery of privacy notice via broadcast

The `PRRD.piiProcessing@broadcastNotice` is defined as `xs:anyURI` and described in ABNT NBR 25608:2025, Table 8, as the location (a URI) of a privacy notice copy in the broadcast service for offline viewing. Under the current specification, however, the receiver cannot deterministically resolve this URI in the broadcast environment, because access to NRT files depends on a base URI correspondent to an Application Context, and no such context is defined for this purpose in the PRRD processing model. As a result, the current specification does not provide a reliable way to locate broadcaster privacy terms through `broadcastNotice` in offline scenarios. Operationally, implementers should recognize that, with the specification as it stands, privacy terms referenced through `broadcastNotice` may be unavailable to receivers that are not connected to the Internet, and no interoperable broadcast-only resolution mechanism should be assumed until a normative solution is defined.

8.8.8 Privacy-related XML schema files

The supplementary files referenced in ABNT NBR 25608:2025 contain a set of metadata and information associated with the standard and constitute an integral part of its normative content. Privacy-related XML schema files are included.

There is a mismatch between a normative statement in ABNT NBR 25608:2025, Section 8.8.2, and the current XML Schema definition (`PRRD-1.0-202511.xsd`) of the `recipientThirdParties` element. ABNT NBR 25608:2025 Table 8 indicates that, when no *Personal Identifiable Information* (PII) is shared, `recipientThirdParties` should contain no child `partyId` elements; however, the normative XSD defines `partyId` with `minOccurs="1"`, which means that an empty element does not pass schema validation. This reflects ABNT NBR 25608:2025 definition of `partyId` cardinality (1..N) also in ABNT NBR 25608:2025 Table 8. To preserve compliance with the current XSD while maintaining alignment with ISO/IEC 27560, Privacy Manager implementations should require the presence of `recipientThirdParties` and, when no third-party recipients exist, accept a declared `partyId` element with empty content.

As of the publication date of these operational guidelines, the `privacyRecord-1.0-202511.xsd` file has been corrected so that the identifier included as a child element of `event` is `entityId`, in accordance with ABNT NBR 25608:2025, Table 9. Implementations should therefore use `entityId` and disregard `eventId` in earlier versions of the schema file.

8.9 Access Panel

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section 8.9 apply.

9 Remote devices

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section 9 apply.

With regard to the remote device classes, these are not necessarily limited to those currently defined in ABNT NBR 25608. Applications may define any classes for remote devices, which affords considerable flexibility to services employing additional devices connected to the receiver via the home network.

10 TV 3.0 AoP general aspects

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section 10 apply.

10.1 Broadcaster Application Location Description (BALD)

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section 10.1 apply.

10.2 Graphic overlays

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section 10.2 apply.

10.3 Memory requirements

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section 10.3 apply.

10.4 TV 3.0 HTTP User-Agent header field

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section 10.4 apply.

10.5 APIs IDs and versioning

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section 10.5 apply.

10.6 Remote-control functions

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section 10.6 apply.

Table 11 of ABNT NBR 25608:2025 lists the remote-control functions applicable to TV 3.0, most of which are mandatory. These functions can be made available through physical keys on the remote control or other equivalent mechanisms, such as on-screen buttons. An exception applies to the “TV +”, “TV -” and “TV 3.0” functions, which, according to ABNT NBR 25609, 13.2, shall be implemented as physical keys.

It is expected that mandatory remote-control functions are always available to the TV 3.0 AoP component or the application (Bootstrap Application or Broadcaster Application) that is activated by them. Thus, if the option is to implement functions as on-screen buttons instead of physical keys, those not applicable in a given context can be omitted, while those applicable are expected to be displayed.

Specifically regarding the “Audio Settings”, “Closed Captioning”, “Audio Description”, “Closed Signing”, and “Dialog Enhancement” functions, it should be noted that there may be functional overlap with the configuration of audio, captioning and accessibility features provided by the Access Panel, as specified in requirement APFR-04 of ABNT NBR 25608:2025. In such cases, the simultaneous presentation of two distinct visual features that lead to the same configuration operation tends to be redundant and potentially confusing for the viewer. Thus, when such features are intended for the PMP, it is understood that it is sufficient to make them available through a single interface mechanism, either as on-screen buttons or as items in the Access Panel.

However, in the specific case of an active Broadcaster Application, when the PMP is inactive (i.e., when there is no audiovisual content being played by it) and these remote control functions are passed to the Broadcaster Application as key events, it is recommended to use on-screen buttons instead of the Access Panel items,

favoring the viewer's perception that they are intended for the application, which, in turn, can provide its own interface features for the viewer to make the related configurations.

10.7 Virtual keyboard

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section 10.7 apply.

10.8 Broadcaster Applications fonts and character encoding

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section 10.8 apply.

11 Harmonization of TV 2.X/TV 3.0 support

11.1 Application Execution Environment

In a TV 2.X/TV 3.0 receiver platform (RP) receiving a TV 2.X service, Broadcaster Applications associated with that service should be executed strictly within the TV 2.X application environment, compliant with the ABNT NBR 15606 series. The application environment behavior exposed to the application should be equivalent to that of a TV 2.X-only environment.

TV 2.X/TV 3.0 RPs are expected to be compliant with the ABNT NBR 15606 series when receiving TV 2.X services. Features mandated by the TV 2.X standards – such as support for the EXIT key or the image graphic plane – should be available in this context.

Broadcaster Applications associated with TV 2.X services should not assume any form of forward compatibility when executed on a TV 3.0 AoP. Capabilities introduced by TV 3.0 standards, such as additional remote-control keys or APIs should not be visible or accessible to Broadcaster Applications transmitted as part of a TV 2.X service.

Remote control key registration and handling for Broadcaster Applications associated with TV 2.X services should follow the semantics and return conditions defined in ABNT NBR 15606-2, ABNT NBR 15606-7, ABNT NBR 15606-11 and ABNT NBR 15606-12. Only keys defined by the TV 2.X specifications should be exposed to the application.

11.2 Electronic Programming Guide (EPG)

In a TV 2.X/TV 3.0 RP, specific limitations apply to the EPG. For TV 2.X services, the EPG should be populated based on information carried in the Event Information Table (EIT), in accordance with the following guidelines:

- Only program information obtained from EIT should be displayed in the EPG.
- The programming period presented by the EPG should be limited to the one defined by the EIT data received
- Functional requirement EPGFR-01 (ABNT NBR 25608:2025, Section 8.5.2) should not be considered applicable when the program information provided by EIT does not span the period specified in EPGFR-01.

- Functional requirement EPGFR-02 (ABNT NBR 25608:2025, Section 8.5.2) should not be considered applicable, as on-demand content identified by URL is not supported in the EPG for TV 2.X services.
- When a program entry is not specified in the EIT, the corresponding EPG item should be presented as empty.

12 Guidelines for building Receiver Platform UIs

12.1 Introduction

The RP needs to handle a series of UIs to provide viewers with the various TV 3.0 features, from PRP UIs, which add some of these features to those developed by manufacturers, to specific TV 3.0 AoP UIs, which provide the entire experience of accessing broadcasters' audiovisual content and additional resources through applications (Bootstrap Applications and Broadcaster Applications). Thus, it is important to define a set of guidelines to enhance the viewer's experience with TV 3.0, considering the variety of implementations by different manufacturers and ensuring visual harmony in the composition of its visual elements with those of broadcasters, while preserving as much as possible the flexibility for manufacturers to apply their best practices in interface design and user experience.

The following subsections provide a set of general and specific recommendations for the TV 3.0-related visual elements to be presented on the PRP home screen and in the TV 3.0 AoP components. Some subsections reference the "DTV+ Visual Identity Manual", a brand usage manual provided by the SBTVD Forum to authorized licensees. It is written in Portuguese under the title "DTV+ Manual de Identidade Visual".

12.2 General recommendations for the PRP home screen and the TV 3.0 AoP components

12.2.1 Bootstrap Application icon for the PRP home screen, the Application Catalog, the EPG, the ECG and the UI of this application

For the Bootstrap Application icon, which is displayed on the PRP home screen, in the Application Catalog, in the EPG, in the ECG and in the UI of this application, the manufacturer can freely choose between the SVG images defined in the `appIcon` element (square format) or `bannerIcon` element (16:9 format) of the BAM element of the BAMT (see ABNT NBR 25608:2025, Table 6). It can even choose different formats for each case.

Since it is not simply a logo but an icon, this icon, in its source file, should have a completely opaque background covering its entire area, with no transparent or semi-transparent regions. In addition, as there are no variations for different color display conditions (such as light and dark modes), it is recommended that the shade of the general background, defined by the manufacturer, be darker and the shade of the icon background, defined by the broadcaster, be lighter, or at least not excessively dark, in order to ensure sufficient contrast and clear visual distinction.

For the general background, it is recommended to use a solid color or a simple gradient, either in grayscale or in tones freely chosen by the manufacturer. Similar guidance applies to the icon background, avoiding excessive details or a complex gradient that may generate disproportionate prominence compared to the icons of other broadcasters.

Regarding the elements of the icon image other than the background, i.e., the essential elements of the icon, these may include the graphic symbol of the broadcaster's DTV service presented alone or this symbol plus the name of the service in its characteristic typography when this name is considered necessary by the broadcaster to facilitate recognition by the viewer. The inclusion of a slogan in the icon image is not recommended, in favor of greater clarity and simplicity of the whole.

In addition, it is expected that the broadcaster positions the essential elements of the icon within an internal safe area, defined by a margin of 15% of the width and height of the icon in relation to its borders, so that the manufacturer, aiming for compliance with its design system, can eventually apply rounded corners up to the limit of this area, avoiding cutting, distortion or loss of legibility of the essential elements.

12.2.2 Texts of the TV 3.0 AoP components

The manufacturer may apply the fonts from its design system to the texts of the various TV 3.0 AoP UIs, with the exception of those associated with the DTV+ brand (see 12.4.2). Thus, the definition of these fonts is free, covering all typographic aspects, such as font family (e.g., Helvetica, Open Sans, Roboto), weight (e.g., normal, light, bold), style (e.g., normal, italic, oblique) and width (e.g., normal, condensed, expanded).

12.3 Specific recommendations for the PRP home screen

12.3.1 DTV+ icon

It is recommended that the icon for accessing the Application Catalog from the PRP home screen presents the DTV+ brand in accordance with the guidelines of the DTV+ Visual Identity Manual. Considering that this manual provides multiple color combinations for the logo and its background, in order to ensure a consistent viewer experience with TV 3.0 across different manufacturers' implementations, it is recommended to adopt the logo in the color white (#FFFFFF) on a background with linear gradient from color #1D1D1B to #1649CC.

To fit the manufacturer's design system, this icon can adopt a square or rectangular format, with straight or rounded corners, and can even reach a circular shape when maximum rounding is applied to the square format. Its size can be adjusted to maintain proportion with the other icons displayed together, preserving visual harmony.

12.4 Specific recommendations for the Application Catalog

12.4.1 DTV+ logo

As it is the main entry point for TV 3.0 AoP, it is recommended that the Application Catalog displays the DTV+ logo at the top of the screen, above the list of Bootstrap Applications and groupings. It can be aligned to the left, right or center and can share this upper region with other UI elements, such as navigation buttons and the viewer profile's avatar.

Considering the general dark background in the Application Catalog due to the Bootstrap Applications and groupings icons (see 12.2.1), it is recommended, among the options provided in the DTV+ Visual Identity Manual, to adopt the DTV+ logo in the color white (#FFFFFF).

12.4.2 Texts associated with the DTV+ brand

The DTV+ Visual Identity Manual presents the typography to be used with the DTV+ brand and a screen example of the Application Catalog using this typography. However, it is important to note that the texts subject to the manual's rules are those associated with the brand and not the common visual elements of the catalog, such as the names of Broadcaster Applications and groupings, button labels etc. (see 12.2.2).

If the manufacturer chooses to display these texts associated with the brand, it is recommended that it uses the fonts indicated in the manual. In addition, it should be noted that there may be fonts that cannot be used free of charge in commercial products and therefore requires the manufacturer to purchase the appropriate license.

12.4.3 Icon of the Bootstrap Application or grouping

The ACFR-29 requirement of ABNT NBR 25608:2025 establishes that the Application Catalog shall maximize the size of the icons and names of Bootstrap Applications in order to promote better use of the display area. In this context, considering the possibility of choosing between the 2 SVG image formats for the Bootstrap Application icon (square or 16:9), as defined in 12.2.1, the manufacturer may switch between them depending on the total number of applications in order to meet this objective.

The Application Catalog can present individual Bootstrap Application icons and grouping icons mixed together in the same display grid. While the former allows direct access to the application, the latter leads to a list of Bootstrap Applications, requiring more than one step of access. Thus, it is necessary for grouping icons to adopt a graphic representation that clearly distinguishes them from individual application icons, in order to convey the idea of a collection accessible from a single icon.

According to the ACFR-33 requirement of ABNT NBR 25608:2025, the cover of this graphic representation is the Bootstrap Application icon associated with the DTV service with the lowest virtual channel number in the grouping. There is no specific normative indication regarding additional graphic elements that refer to the idea of a collection. Thus, the way of differentiating between individual Bootstrap Application and grouping icons is at the manufacturer's discretion and can be done, for example, by means of a graphic indication of stacked layers.

12.4.4 Text of the Bootstrap Application or grouping name

According to ABNT NBR 25608:2025, Table 10, the text of the Bootstrap Application name presented in the **BAM** element of **BAMT** – and used in the Application Catalog – can contain up to 20 characters. In addition, the ACFR-30 requirement of the same standard establishes that the catalog shall ensure a combined total of 40 icons – comprising Bootstrap Application and grouping icons – on a single screen without the need for scrolling.

Considering the limited space available, especially when there are a large number of icons, it is recommended that the text associated with the Bootstrap Application or grouping in the Application Catalog be displayed in a maximum of 2 lines, in order to preserve the proper highlighting of the icons.

It is not necessary for the Bootstrap Application icon image to contain a name text, but if it does, it is recommended that the same text be adopted as the application name, with a possible variation in the use of upper and lower case letters, but preserving the spelling in terms of spaces and the use of diacritical marks.

12.4.5 DTV+ badge for the Bootstrap Application or grouping icon

In the Application Catalog, the DTV+ badge is used to differentiate Bootstrap Applications and grouping icons transmitted in TV 3.0 from those transmitted in TV 2.X. It is specified in the ACFR-13 requirement of ABNT NBR 25608:2025.

It is recommended that this badge be composed of the DTV+ logo in the color white (**#FFFFFF**) on a background in a solid color among those provided in the DTV+ Visual Identity Manual that allow combination with white. In addition, it is recommended that the badge be overlaid on the corresponding icon, positioned next to its lower border and aligned to the left, right or center, without necessarily covering the entire width of the icon and with either straight or rounded corners.

Furthermore, it is recommended to take advantage of the safe area corresponding to a margin of 15% of the width and height intended for the positioning of the essential elements of the icon, in order to keep them fully visible, as is desired in the possible rounding of the corners of the icon for display (see 12.2.1).

When resizing the set of icons in the Application Catalog based on their total number (requirement ACFR-29 of ABNT NBR 25608:2025), it is recommended that the size of the badge maintain the same proportion with the size of the icon.

12.4.6 “Favorite” badge for the Bootstrap Application or grouping icon

In the Application Catalog, the “favorite” badge is a visual element associated with Bootstrap Applications icons and groupings, intended to indicate those marked by the viewer as favorites. This badge is specified in the ACFR-09 requirement of ABNT NBR 25608:2025, and its presence, together with repositioning to the top of the list, as defined in the same requirement, allows immediate identification of this configuration and facilitates access to the viewer's preferred applications.

The choice of the graphic element representing the “favorite” is up to the manufacturer.

It is recommended that the badge be overlaid on the Bootstrap Application or grouping icon. It may even occupy the icon's essential elements area (i.e. it is not restricted to the safe area, as is the case for the DTV+ badge, see 12.4.5), but outside the area where the DTV+ badge may appear. When there is interference with the essential elements, it is recommended that transparency be applied to the “favorite” graphic element.

For visual consistency, it is recommended that the same graphic element applied to the Bootstrap Application icon also be applied to the grouping icon, allowing minimal displacement for better adaptation to the graphic representation indicative of grouping. For example, the badge can be positioned in the cover area of the grouping icon that displays the icon image provided by the broadcaster for the application associated with the DTV service with the lowest virtual channel number (see 12.4.3). In this case, it is recommended that the size of the badge not be changed.

Regarding the resizing of the icon set (requirement ACFR-29 of ABNT NBR 25608:2025), it is recommended that the favorite badge be resized in the same proportion, similar to what occurs with the DTV+ badge (see 12.4.5).

Annex A

NCL 4.0 and TV 3.0 Ginga-NCL

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Annex A apply.

A.1 Introduction

This Annex details and explains recommendations and behavior of NCL 4.0 APIs and TV 3.0 Ginga-NCL presentation engine, as supported by the TV 3.0 Application-oriented Platform defined in ABNT NBR 25608:2025. It provides operational guidelines for an implementation of TV 3.0 Ginga-NCL according to ABNT NBR 25608:2025, Annex A.

A.2 NCL as glue language

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section A.2 apply.

A.3 NCL 4.0 specification

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section A.3 apply.

A.3.1 General aspects

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section A.3.1 apply.

A.3.2 NCL specification summary

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section A.3.2 apply.

A.3.3 NCL Structure API

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section A.3.3 apply.

A.3.4 NCL Layout API

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section A.3.4 apply.

A.3.5 NCL Components API

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section A.3.5 apply.

With regard the dynamically available variables, identified by the respective `<class_id>`, the variables `<class_id>.screenSize(i)` and `<class_id>.screenGraphicSize(i)` should not have their value defined if the corresponding device does not provide the corresponding information.

A.3.6 NCL Interfaces API

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section A.3.6 apply.

As stated in Section A.3.6 of ABNT NBR 25608:2025, except for `<media>` elements of type `application/x-vinga-time`, the begin and end attributes of an `<area>` element are specified using one of the following syntaxes:

1. Local time of day with decimal fraction representation, as defined in ISO 8601-1;
2. Internet Date/Time Format, as defined in IETF RFC 9557;
3. “`<seconds>s`”, where `<seconds>` is a positive integer.

When the “`<seconds>s`” syntax (option 3) is used, the specified time is interpreted as relative to the beginning of the presentation of the associated `<media>` element. That is, it represents an offset, in seconds, from the instant at which the `<media>` presentation event enters the occurring state. This option can be applied to any type of `<media>` content.

When the Internet Date/Time Format syntax (option 2), as defined in RFC 9557, is used, the specified time represents an absolute instant on the UTC timeline, including both date and time information. This representation is appropriate when the `<media>` content is delivered using real-time or live streaming mechanisms whose synchronization model is explicitly based on UTC and whose signaling allows receivers to map wall-clock time to the media timeline. Examples include ROUTE/DASH, MPEG-DASH, and HLS services that provide UTC-aligned timing metadata. If the Internet Date/Time Format syntax (option 2) is used when the `<media>` content is not delivered using real-time or live streaming mechanisms, implementations are recommended to ignore the corresponding `<area>` element.

When the local time of day with decimal fraction syntax defined in ISO 8601-1 (option 1) is used, the specified time includes only hours, minutes, and seconds, without any date component. In this context, the time is interpreted as relative to the beginning of the presentation of the associated `<media>` element, rather than as a wall-clock time of day. It therefore represents an offset from the instant at which the `<media>` presentation event enters the occurring state. This syntax can be understood as an alternative representation to option 3 and may be preferred when expressing large offsets, as it can improve readability and comprehension. This option can be applied to any type of `<media>` content.

A.3.7 NCL Presentation Specification API

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section A.3.7 apply.

A.3.8 NCL Linking API

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section A.3.7 apply.

A.3.9 NCL Connectors API

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section A.3.9 apply.

A.3.10 NCL Presentation Control API

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section A.3.10 apply.

A.3.11 NCL Timing API

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section A.3.11 apply.

A.3.12 NCL Reuse API

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section A.3.12 apply.

A.3.13 NCL Navigational Key API

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section A.3.13 apply.

A.3.14 NCL Animation API

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section A.3.14 apply.

A.3.15 NCL Transition Effects API

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section A.3.15 apply.

A.3.16 NCL Metainformation API

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section A.3.16 apply.

A.3.17 NCL Users API

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section A.3.17 apply.

A.3.18 NCL Multimodal Interaction API

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section A.3.18 apply.

A.3.19 NCL Sensory Effects API

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section A.3.19 apply.

A.3.20 NCL Multi-device API

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section A.3.20 apply.

A.3.21 NCL 360° VR API

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section A.3.21 apply.

A.3.22 NCL Editing Commands API

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section A.3.22 apply.

Table A.38 of ABNT NBR 25608:2025 defines two sets of application management commands. The `addDocument`, `removeDocument`, and `saveDocument` commands operate on NCL documents and are retained for compatibility with TV 2.X Ginga-NCL. The `addApplication`, `removeApplication`, and `saveApplication` commands operate on TV 3.0 Ginga-NCL applications, identified by the `appId` attribute informed via BALD (see ABNT NBR 25608:2025, 10.1). TV 3.0 DTV services should use `addApplication`, `removeApplication`, and `saveApplication`.

A.4 TV 3.0 Ginga-NCL presentation engine

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section A.4 apply.

A.5 Harmonization of TV 2.X/TV 3.0 support

A.5.1 NCL Preparation Event

In TV 3.0 Ginga-NCL, the handling of preparation events differs intentionally from the behavior implicitly allowed in previous TV 2.5 implementations based on ABNT NBR 15606-2. This change was introduced to remove

ambiguities identified during the interpretation and implementation of preparation event semantics, particularly regarding the conditions under which a preparation can be considered successfully completed.

In earlier specifications (see ABNT NBR 15606-2 Section 7.2.8), the state transition associated with the end of a preparation event was the same (`stops` transition) regardless of whether the transition resulted from the natural completion of preparation or from an explicit termination triggered by the application (`stopPreparation` action role). Because both situations relied on the same transition, implementations could infer (or not) completion even when the preparation had been prematurely stopped, which in turn led to inconsistent interpretations of readiness for presentation and to fragmented middleware behavior.

For TV 3.0, this ambiguity was addressed by eliminating the possibility of explicitly terminating a preparation event through an application-driven `stopPreparation` action (see ABNT NBR 25608:2025 Section A.3.9). As a result, a preparation event now reaches its `prepared=true` state only as a consequence of its natural conclusion under middleware control. This ensures that the completion of preparation consistently reflects the middleware's internal assessment of readiness rather than an external decision made by a Broadcaster Application.

This clarification directly strengthens the semantic reliability of preparation completion, particularly with respect to the expectation of immediate presentation start. When a preparation event concludes in TV 3.0 with its `prepared` attribute set to `true`, implementations can safely assume that the preparation has progressed to a level deemed sufficient by the middleware, without the risk that completion was forced while internal conditions were still incomplete.

Interruption of preparation remains possible, but only through abrupt termination semantics (`abortPreparation` action role), which are clearly distinct from successful completion and do not imply readiness for immediate presentation.

Although this change represents a departure from behaviors that may be tolerated by TV 2.5 implementations, it aligns the preparation model more closely with its original intent and with the TV 3.0 operational requirements. Future revisions of TV 2.5 specifications may incorporate similar clarifications in order to harmonize behavior across system generations.

Annex B

TV 3.0 Ginga-HTML5

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Annex B apply.

B.1 Introduction

This Annex details and explains recommendations and behavior of TV 3.0 HTML5, as supported by the TV 3.0 Application-oriented Platform defined in ABNT NBR 25608:2025. It provides operational guidelines for an implementation of TV 3.0 Ginga-HTML5 according to ABNT NBR 25608:2025, Annex B.

B.2 Basic reference

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section B.2 apply.

B.3 Application identification

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section B.3 apply.

B.4 Presentation of visual elements

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section B.4 apply.

B.5 User input

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section B.5 apply.

B.6 Network

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section B.6 apply.

Annex C

TV 3.0 WebServices

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Annex C apply.

C.1 Introduction

This Annex details and explains recommendations and behavior of TV 3.0 WebServices APIs supported by the TV 3.0 Application-oriented Platform defined in ABNT NBR 25608:2025. It provides operational guidelines for an implementation of TV 3.0 WebServices according to ABNT NBR 25608:2025, Annex C.

C.2 TV 3.0 WebServices architecture

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section C.2 apply.

C.3 General rules and definitions

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section C.3 apply.

C.4 Security rules and definitions

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section C.4 apply.

C.5 TV 3.0 WebServices APIs groups

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section C.5 apply.

C.6 TV 3.0 WebServices APIs specification

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section C.6 apply.

C.6.1 APIs for client identification and authorization

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section C.6.1 apply.

C.6.2 Event notification APIs

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section C.6.2 apply.

C.6.2.1 Error codes upon registration

When using the API C.6.2.2 to register a listener for event notifications against any event class, an immediate failure to register the listener will return an HTTP response with error code and corresponding message. The error codes common for every possible <class-name> are listed in ABNT NBR 25608 Table C.5, at the row "Possible error codes". In addition to those common error codes, specific class names may have their own exclusive set of error codes. These additional error codes are listed in the row "class-dependent error codes" of each respective table in ABNT NBR 25608.

C.6.3 DTV context access APIs

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section C.6.3 apply.

C.6.4 APIs for communicating with TV 3.0 AoP

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section C.6.4 apply.

C.6.5 APIs for accessing broadcaster tables and metadata

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section C.6.5 apply.

C.6.6 APIs for accessing broadcast media content

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section C.6.6 apply.

C.6.6.1 API for obtaining available media players and API for obtaining information about a specific media player

This subsection establishes additional recommendations for APIs C.6.6.1 and C.6.6.2

C.6.6.1.1 closedSigning field

If the platform implements ABNT NBR 25606, the closedSigning field is expected to be set according to its implementation in the PMP. Support for this field in other media players depends on the platform implementation.

C.6.6.2 API for controlling a specific media player

This subsection establishes additional recommendations for API C.6.6.3.

C.6.6.2.1 playbackRate field

It is recommended that changes to the playbackRate field take into account the type of content currently being played by the media player. For example, if the content being played does not support changes to the playbackRate field, such as live content, the API may return error 101. The same applies if the player does not support reverse playback, in which case a negative playbackRate value is provided.

C.6.6.2.2 Applicability of control actions by content type

The message body of API C.6.6.3 combines two families of parameters: presentation and rendering controls (`pos`, `vol`, `visible`, `ngaInteractivity`), which act on how the media is laid out and rendered, and time-domain controls (`currTime` for seeking, `playbackRate` for trick play, and the action state transitions), which act on the media timeline. The applicability of each time-domain control is governed by the temporal nature of the media addressed by the player. The paragraphs below describe the expected applicability in each scenario, and Table C.6.1 summarizes it.

The seekable range and the time-shift (DVR) window

For broadband content, the positions to which `currTime` may seek, and the range over which `playbackRate` may operate, are defined by the seekable range of the presentation. For on-demand content, the seekable range spans the whole presentation, from its start to its end. For live content, the seekable range is a time-shift window bounded on the future side by the live edge (the most recent position published by the server) and on the past side by the oldest position still retained. This past boundary, commonly referred to as the DVR window, is signaled by the manifest and is not a fixed property of "live": it may be absent (playback restricted to the live edge), bounded to a fixed depth, or effectively unbounded (seekable back to the start of the presentation), and it typically moves forward over time as new content is published and older content is retired.

The manifest signals this window differently in each streaming format supported by the PMP and additional media players:

- In MPEG-DASH (see ISO/IEC 23009-1), an on-demand presentation is indicated by `MPD@type="static"` and is seekable over its whole duration. A live presentation is indicated by `MPD@type="dynamic"`; the depth of its time-shift window is given by `MPD@timeShiftBufferDepth` (ISO/IEC 23009-1, subclause 5.3.1.2), which specifies the smallest time-shifting buffer guaranteed to be available. When `MPD@timeShiftBufferDepth` is not present, the window is unbounded and the presentation is seekable back to `MPD@availabilityStartTime`. Segments that fall outside the current window cease to be available, so the past boundary of the seekable range advances as the presentation progresses.
- In HLS (see IETF RFC 8216), the mutability of the media playlist defines the window. A playlist marked `EXT-X-PLAYLIST-TYPE:VOD`, or one carrying `EXT-X-ENDLIST`, denotes a complete presentation that is seekable over its whole duration. A live presentation is a playlist without `EXT-X-ENDLIST`: when marked `EXT-X-PLAYLIST-TYPE:EVENT`, segments are only appended and never removed, so the window grows and remains seekable back to the start of the event; when no playlist type is declared, the playlist is a sliding window from which older segments are removed, and the seekable range corresponds to the span of the segments currently listed. The later appearance of `EXT-X-ENDLIST` marks the end of the live presentation, after which it behaves as on-demand content.

In both formats the seekable range is the interval currently exposed by the manifest, and a seek is expected to succeed only for positions inside that interval and up to the live edge. This model is consistent with the behavior of widely deployed players for both formats, which expose the current seekable interval (for example, as a seekable time range or as a DVR window) and interpret seek positions on a live presentation relative to that interval, while exposing the whole duration for on-demand content.

Broadcast content played by the Persistent Media Player

When the PMP presents the current linear broadcast service, the content is inherently live and, in the general case, no receiver-side time-shift buffer is retained, so no seekable range is available other than the live edge. Under these conditions:

- `pos`, `vol` and `visible` are expected to remain applicable, since they act on the layout and audio level of the current program rather than on its timeline. `vol` applies to programs carrying an audio component; consistent with error 203, `vol` is not meaningful for video-only or caption media, and `pos` is not meaningful for audio-only media.
- `ngaInteractivity` is expected to be applicable when the current program carries an MPEG-H audio stream (see C.6.6.2 and ABNT NBR 25604) and does not apply otherwise.
- `currTime` (seek) and `playbackRate` (trick play) are, in the general case, not applicable, since there is no seekable buffer behind the live edge and playback cannot advance beyond it; a `currTime` set request on live broadcast content is expected to be rejected with error 203.
- Among the action transitions, `start`, `stop`, `unload`, `pause` and `resume` are expected to be applicable. Because the receiver does not retain a time-shift buffer for broadcast content, `pause` freezes the presentation on the last displayed frame without accumulating content behind the live edge, and `resume` does not continue from the position at which the presentation was paused: playback resumes from the current live edge, that is, from the most recent content being received over the air at the moment of resumption.

On-demand content

For on-demand content, delivered as a self-contained file or as a broadband presentation whose manifest denotes a complete presentation (a DASH "static" presentation, or an HLS playlist marked as VOD or carrying EXT-X-ENDLIST), the whole presentation timeline is seekable:

- `currTime` (seek) is expected to be applicable across the entire range `[0, duration]`, in the "playing", "paused" and "prepared" states, as specified in C.6.6.3.
- `playbackRate` is expected to be applicable across the range specified in C.6.6.3 (absolute values from 0.5 to 4, and negative values for reverse playback where the player supports it), enabling fast-forward, slow-motion and rewind.
- `pos`, `vol`, `visible` and `ngaInteractivity` (for MPEG-H audio) apply as for any A/V player.

Live broadband content

For live content delivered over broadband (a DASH "dynamic" presentation, or an HLS live playlist), the applicability of time-domain controls follows the time-shift window described above:

- `currTime` (seek) is expected to be applicable within the current window: backward to positions still retained, and forward up to the live edge. A seek to a position that has already fallen out of the window, or beyond the live edge, is not expected to be honored. When the presentation exposes no time-shift window (for example, a service kept at the live edge with no retained history), the behavior converges to that of live broadcast content, and a `currTime` set request is expected to be rejected with error 203.

- `playbackRate` is expected to be applicable for trick play within the window (for example, catch-up at rates above 1.0 that converge toward the live edge, or slow-motion and rewind toward the start of the window). Forward playback cannot progress beyond the live edge.
- `pause` and `resume` are expected to be applicable. When the presentation provides a time-shift window, `pause` freezes the presentation on the last displayed frame and `resume` continues from that position, which then trails the live edge within the window (if the pause lasts longer than the window depth, the paused position may fall outside the window, and playback resumes from the oldest position still retained). When the presentation provides no time-shift window, the behavior matches that of broadcast content, and `resume` continues from the current live edge rather than from the position at which the presentation was paused.
- When a live presentation is finalized (a DASH presentation that transitions to "static", or an HLS playlist to which `EXT-X-ENDLIST` is added), the finalized timeline becomes fully seekable and the content behaves as on-demand, subject to platform support.
- `pos`, `vol`, `visible` and `ngaInteractivity` (for MPEG-H audio) apply as for any A/V player.

Table C.6.1 Applicability of C.6.6.3 control actions by content type

Control action	Broadcast	On-demand	Live broadband
<code>pos</code>	Applicable	Applicable	Applicable
<code>vol</code>	Applicable (programs with audio)	Applicable	Applicable
<code>visible</code>	Applicable	Applicable	Applicable
<code>ngaInteractivity</code>	Applicable if MPEG-H audio	Applicable if MPEG-H audio	Applicable if MPEG-H audio
<code>currTime (seek)</code>	Not applicable (error 203)	Applicable over [0, duration]	Applicable within the time-shift window; not applicable when no window is retained (error 203)
<code>playbackRate (trick play)</code>	Not applicable	Applicable (0.5 to 4; reverse where supported)	Applicable within the time-shift window; cannot progress beyond the live edge
<code>action: start, stop, unload</code>	Applicable	Applicable	Applicable
<code>action: pause, resume</code>	Applicable; <code>resume</code> continues from the live edge	Applicable	Applicable; <code>resume</code> continues from the paused position within the time-shift window, or from the live edge when no window is retained

"Applicable" indicates that the action is meaningful for the content type; actual support in a given receiver depends on platform capabilities and may be reported through error 202 or 203 when a specific action or parameter cannot be honored.

C.6.7 APIs for integration with the platform and home environment

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section C.6.7 apply.

C.6.7.1 API for registering filters for target applications

This subsection establishes additional recommendations for API C.6.7.3.

C.6.7.1.1 Removal of target applications

It's up to the AoP implementer to define the removal and timing of deregistering an application registered by a stand-alone local client, as well as error handling. The implementation may return error 101 if it can verify that the request to this API is not authentic, that is, if it verifies that the "canonicalName" and/or "targetId" sent are valid, but the target application is no longer available for execution.

C.6.7.1.2 Parameter repetitions

In general, target applications under registration may encounter parameters that are already registered by another application; therefore, the displayName should not generate an error in case of repetitions. It's up to the AoP implementer to validate if the registration parameters of an application are valid and associated with an application in the current service and may optionally return error 101 (Illegal argument value) for invalid values. To manage conflicts with stand-alone local clients, refer to ABNT NBR 25608:2025 C.6.7.3.5.

C.6.8 Broadcaster security APIs

For the purposes of this part of the Document, the terms and definitions of ABNT NBR 25608:2025, Section C.6.8 apply.

C.6.8.1 Registering a DTV service context for the use of TV 3.0 WebServices APIs

This subsection establishes additional recommendations for API C.6.8.2.

C.6.8.1.1 Harmonization with TV 2.X

If the bind-token specified in the call to this API is associated with TV 2.X services, these services should also be listed in the JSON response. In this case, for these services, the validity check of the bind-token signature is not performed, and therefore error code 101 does not apply.

C.6.9 API for geolocation

For the purposes of this part of the Document, the terms and definitions of ABNT NBR 25608:2025, Section C.6.9 apply.

C.6.10 APIs for closed signing

For the purposes of this part of the Document, the terms and definitions of ABNT NBR 25608:2025, Section C.6.10 apply.

C.6.11 Application Context Cache APIs

For the purposes of this part of the Document, the terms and definitions of ABNT NBR 25608:2025, Section C.6.11 apply.

C.6.11.1 Caching broadband DASH media segment files

This subsection establishes additional recommendations for API C.6.11.3.

C.6.11.1.1 Use of the DASH Media Segment Caching API

The API specified in Section C.6.11.3 of ABNT NBR 25608:2025 is intended for downloading media segment files referenced by an XML fragment corresponding to a DASH Period. Given its parameter structure – a single `sourceURL` applied as a prefix to relative URLs within the `Period` – and its underlying model of bulk segment enumeration, the API has the following applicability scope.

C.6.11.1.2 Applicable segment addressing model

The API is directly applicable to Periods that use the `<SegmentTemplate>` addressing model (ISO/IEC 23009-1, Clause 5.3.9.4), in which segment URLs are constructed by substituting identifier templates (`$Number$`, `$Time$`, `$RepresentationID$`, `$Bandwidth$`, and `$SubNumber$`) into the attributes of the `<SegmentTemplate>` element.

The `<SegmentTemplate>` attributes that are expected to be processed by the API as references to files to be downloaded are:

- `SegmentTemplate@initialization`: yields the relative URL of the Initialization Segment for each Representation;
- `SegmentTemplate@media`: yields, for each valid combination of identifier values, the relative URL of each Media Segment for the Representation.

The attributes `SegmentTemplate@index` and `SegmentTemplate@bitstreamSwitching` are only relevant when present and when they reference distinct files (i.e., when they contain `$Number$` or `$Time$` templates, or when used in conjunction with the corresponding `@bitstreamSwitching` flag).

For each relative URL derived from the identifier expansion described above, the final download URL is obtained by direct concatenation of `sourceURL` + relative URL, following the URL resolution rules of ISO/IEC 23009-1, Clause 5.6.

C.6.11.1.3 Limitations of the Current API

a) **MPDs with multiple addressing models.** The API does not cover the `<SegmentList>` (ISO/IEC 23009-1, Clause 5.3.9.3) or `<SegmentBase>` with `<BaseURL>` (ISO/IEC 23009-1, Clause 5.3.9.2) addressing models. Applications that need to cache segments from Periods using these models are expected to use alternative mechanisms rather than this API.

b) **`<BaseURL>` elements within the Period fragment.** The API uses a single `sourceURL` as prefix. `<BaseURL>` elements present at the `<Period>`, `<AdaptationSet>`, or `<Representation>` levels within the fragment passed in the `Period` parameter have no defined processing in the current API design. It is recommended that applications remove any `<BaseURL>` elements from the fragment before submitting it to the API and, where necessary, pre-compose the effective prefix – including any contribution from MPD-level `<BaseURL>` elements – into the `sourceURL` parameter.

c) **`<BaseURL>` at the `<MPD>` level.** Since the API receives only the Period fragment rather than the complete MPD, any `<BaseURL>` element at the MPD level is not visible to the API. Applications are expected to compute the effective prefix and supply it in `sourceURL` prior to invoking the API.

d) **Server redundancy.** The API does not support multiple `<BaseURL>` elements for redundant origins; only a single `sourceURL` is accepted per request. Applications requiring server-side redundancy are expected to handle failover logic independently of this API.

e) **Absolute URLs within the fragment.** The API is designed for Periods containing only relative URLs, as stated in the standard. Periods that include absolute URLs in any URL attribute or in <BaseURL> elements are outside the scope of this API.

f) **Period@duration attribute and finite Period scope.** The @duration attribute on the <Period> element is required for the API to function, since it is used to determine the number of Media Segments to be enumerated and downloaded for each Representation (computed as $N = \text{ceil}(\text{Period@duration} \times \text{SegmentTemplate@timescale} / \text{SegmentTemplate@duration})$). When this attribute is absent from the original MPD but the duration of the Period is implicitly determined – for instance, by the @start of the following <Period> or by MPD@mediaPresentationDuration – applications are expected to compute the effective duration and add it to the fragment before invoking the API.

g) **Live presentations and Periods of indefinite duration.** In dynamic presentations (MPD@type="dynamic"), the last <Period> of an MPD may have no explicit Period@duration, no following <Period>, and no MPD@mediaPresentationDuration. In such cases, the duration of the Period is undefined at the time of the API call and is only constrained by future MPD updates. This API is not designed to operate on Periods of indefinite duration: enumerating an unbounded set of Media Segments would result in a download job of unknown size, with potential exhaustion of cache storage and receiver memory. Accordingly, this API is intended for use with Periods whose duration is fully determined at the moment of invocation – typically Periods of static MPDs (MPD@type="static", on-demand or scheduled content), or Periods of dynamic MPDs whose duration has already been finalized and whose Media Segments are already available on the server. For live caching scenarios involving the live edge of a dynamic presentation, applications are expected to either (i) wait until the relevant Period has acquired a finite duration (e.g., when the presentation transitions to static, when a subsequent <Period> is signaled, or when MPD@mediaPresentationDuration is set), or (ii) use alternative mechanisms such as the Internet file caching API (ABNT NBR 25608:2025, Section C.6.11.2) with explicit, application-managed segment URL lists.

C.6.11.1.4 Usage Example

Consider the following <Period> fragment, derived from a static MPD for the *Big Buck Bunny* content (1-second segments, 20 Representations with distinct bandwidth values):

```
<Period duration="PT0H9M55.46S">
  <AdaptationSet segmentAlignment="true" group="1" maxWidth="480" maxHeight="360"
    maxFrameRate="24" par="4:3">
    <SegmentTemplate timescale="96"
      media="bunny_$Bandwidth$bps/BigBuckBunny_1s$Number$.m4s"
      startNumber="1" duration="96"
      initialization="bunny_$Bandwidth$bps/BigBuckBunny_1s_init.mp4" />
    <Representation id="320x240 47.0kbps" mimeType="video/mp4" codecs="avc1.42c00d"
      width="320" height="240" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="46980" />
    <Representation id="320x240 92.0kbps" mimeType="video/mp4" codecs="avc1.42c00d"
      width="320" height="240" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="91917" />
    <!-- remaining Representations omitted for brevity -->
  </AdaptationSet>
</Period>
```

Suppose the application wishes to cache this content from the server <https://cdn.media.com/bbb/> into the bbb-content/ directory of the Application Context Cache. The API call would be:

```
POST http(s)://<host>/tv3/current-service/cache/dash/download-jobs
```

Request body:

```
{
```

```
"sourceURL": "https://cdn.media.com/bbb/",
"targetURL": "bbb-content/",
"Period": "<Period duration='PT0H9M55.46S'><AdaptationSet ...><SegmentTemplate timescale='96'
media='bunny_$Bandwidth$bps/BigBuckBunny_1s$Number$.m4s' startNumber='1' duration='96'
initialization='bunny_$Bandwidth$bps/BigBuckBunny_1s_init.mp4'><Representation id='320x240 47.0kbps' bandwidth='46980'
...>...</AdaptationSet></Period>"
}
```

Applying the URL construction rules of ISO/IEC 23009-1, Clause 5.3.9.4.4, with `Period@duration=595.46s`, `SegmentTemplate@duration=96`, `@timescale=96`, and `@startNumber=1`, the result is 596 Media Segments per Representation, plus 1 Initialization Segment per Representation. For the Representation with bandwidth 46980, the derived URLs and corresponding cache storage locations would be:

File Type	Derived Relative URL	Full Download URL	Cache Location
Init	bunny_46980bps/BigBuckBunny_1s_init.mp4	https://cdn.media.com/bbb/bunny_46980bps/BigBuckBunny_1s_init.mp4	<base>/bbb-content/bunny_46980bps/BigBuckBunny_1s_init.mp4
Media (k=1)	bunny_46980bps/BigBuckBunny_1s1.m4s	https://cdn.media.com/bbb/bunny_46980bps/BigBuckBunny_1s1.m4s	<base>/bbb-content/bunny_46980bps/BigBuckBunny_1s1.m4s
Media (k=2)	bunny_46980bps/BigBuckBunny_1s2.m4s	https://cdn.media.com/bbb/bunny_46980bps/BigBuckBunny_1s2.m4s	<base>/bbb-content/bunny_46980bps/BigBuckBunny_1s2.m4s
...	...	...	...
Media (k=596)	bunny_46980bps/BigBuckBunny_1s596.m4s	https://cdn.media.com/bbb/bunny_46980bps/BigBuckBunny_1s596.m4s	<base>/bbb-content/bunny_46980bps/BigBuckBunny_1s596.m4s

(Where <base> is the base URL of the *Application Context Cache* directory, as obtained via API C.6.11.1.)

The same pattern applies to each of the remaining Representations, with `$Bandwidth$` substituted by the corresponding `Representation@bandwidth` value.

C.6.11.1.5 Guidance for Application Implementers

Before submitting the `Period` fragment to the API, applications are expected to verify that: (i) the `@duration` attribute is present on the `<Period>` element and represents a finite, fully determined duration; (ii) all URLs within the fragment are relative; (iii) the fragment uses the `<SegmentTemplate>` addressing model; and (iv) any `<BaseURL>` elements have been handled – either removed from the fragment and/or incorporated into `sourceURL`.

When the original MPD uses addressing models not covered by this API (`<SegmentList>`, `<SegmentBase>`), contains `<BaseURL>` elements that cannot be reduced to a single prefix, or describes a `Period` whose duration is not yet finalized (live edge of a dynamic presentation), the recommended approach is to use the Internet file caching API (ABNT NBR 25608:2025, Section C.6.11.2), constructing the list of download URLs by processing the MPD directly and managing segment availability over time at the application level.

C.6.12 Alternative MPD resolution APIs

For the purposes of this part of the Document, the terms and definitions of ABNT NBR 25608:2025, Section C.6.12 apply.

C.6.12.1 Resolution of Alternative MPDs

This subsection establishes additional recommendations for API C.6.12.2.

C.6.12.1.1 API design and alignment with ISO/IEC 23009-1

API C.6.12.2 (/tv3/current-service/alternative-mpd/resolve) implements, within the *TV 3.0 AoP* context, the extension point provided by the *Alternative MPD* mechanism specified in ISO/IEC 23009-1:2025 (§5.16). The two operations supported by the DASH standard – *Alternative MPD Insertion* (urn:mpeg:dash:event:alternativeMPD:insert:2025) and *Alternative MPD Replacement* (urn:mpeg:dash:event:alternativeMPD:replace:2025) – are the same ones referenced by ABNT NBR 25608:2025 Table C.63 of ABNT NBR 25608:2025 for the event classes notified to the broadcaster's application.

The reference processing model defined in ISO/IEC 23009-1, §5.16.2, establishes that, for each *Alternative MPD* event dispatched, the DASH client executes the `resolve(E)` function, which comprises:

- a) the download of the MPD from the URL provided in the `AlternativeMPDEventType@uri` attribute of the event in the main MPD;
- b) the parsing and validity check of the obtained MPD;
- c) if the obtained MPD is a *List MPD*, its further resolution as defined in §5.3.2.6.3.

API C.6.12.2 gives the broadcaster's application the opportunity to interpose itself in this process: upon receiving an *Alternative MPD* event notification (ABNT NBR 25608:2025 Table C.63), the application may choose to (i) let the *TV 3.0 AoP* autonomously proceed with `resolve(E)` (`status=0`); (ii) abort the process (`status=-1`); or (iii) override the result of `resolve(E)` by itself supplying the alternative MPD to be used by the *Persistent Media Player* (`status=1`).

When option (iii) is exercised, the content delivered by the application through the "mpd" or "mpdUrl" parameters occupies, from the *Persistent Media Player's* perspective, the same functional position as the document that would have been autonomously obtained from `AlternativeMPDEventType@uri`. In other words, the application provides the final artifact that feeds the preloading and presentation-switching pipeline described in §5.16.2.

C.6.12.1.2 Document type expected by the "mpd" and "mpdUrl" parameters

As a consequence of the alignment described in the previous section, the content of the "mpd" parameter, as well as the resource pointed to by the "mpdUrl" parameter, is a complete, well-formed DASH MPD document, conforming to the urn:mpeg:dash:schema:MPD:2011 schema (ISO/IEC 23009-1, Annex B). This document may contain one or multiple `<Period>` elements as children of the `<MPD>` root element and may be either a regular *Presentation MPD* or a *List MPD* (`MPD@type="list"`, urn:mpeg:dash:profile:list:2024), as appropriate to the use case.

An isolated `<Period>` fragment, or any other partial fragment of an MPD, does not constitute valid input for either parameter. The reason is that step (b) of the `resolve(E)` function requires that the received document be parseable and validatable as a complete MPD, with an `<MPD>` root element and its mandatory attributes (`@profiles`, `@type`, etc.). A standalone `<Period>` would fail at this step and would lead to the removal of the event from the execution queue, as specified in §5.16.2.2.5 of ISO/IEC 23009-1.

This guidance is consistent with EXAMPLE 1 of ABNT NBR 25608:2025 Table C.64, in which the value of the "mpd" parameter begins with the XML declaration followed by the root element `<MPD xmlns="urn:mpeg:dash:schema:MPD:2011" ...>`, containing the `<Period>` element(s) as its children, and not as the root element.

NOTE The presence of the "sourceURL" and "Period" parameters in API C.6.11.4 (download of content into the *Application Context Cache*) addresses a distinct use case, in which the application requests the receiver to pre-cache segments described in a standalone MPEG-DASH <Period> fragment. That design does not apply to API C.6.12.2, and the two sets of parameters are not interchangeable.

C.6.13 Audience measurement APIs

For the purposes of this part of the Document, the terms and definitions of ABNT NBR 25608:2025, Section C.6.13 apply.

C.6.14 TV 3.0 Users APIs

For the purposes of this part of the Document, the terms and definitions of ABNT NBR 25608:2025, Section C.6.14 apply.

C.6.15 Remote device communication APIs

For the purposes of this part of the Document, the terms and definitions of ABNT NBR 25608:2025, Section C.6.15 apply.

C.6.16 Sensory Effects APIs

For the purposes of this part of the Document, the terms and definitions of ABNT NBR 25608:2025, Section C.6.16 apply.

C.6.17 Filter APIs for local caching of files transmitted via NRT from TV 3.0

For the purposes of this part of the Document, the terms and definitions of ABNT NBR 25608:2025, Section C.6.17 apply.

C.7 Harmonization of TV 2.X/TV 3.0 support

In a TV 2.X/TV 3.0 receiver platform (RP), support for all features implemented by application engines like NCL, HTML5 and WebServices should follow the definitions presented in the published ABNT standards for that version.

For TV 3.0 WebServices, the premises and recommendations described in Section 11 of this document apply. In a TV 2.X/TV 3.0 RP, the TV 2.X Ginga CC WebServices module supports all definitions and APIs described in ABNT 15606-11 and ABNT 15606-12, while the TV 3.0 WebServices module supports all definitions and APIs described in ABNT NBR 25608:2025, Annex C.

In certain specific situations additional guidelines are necessary for the implementation of TV 3.0 WebServices APIs.

In the case of some TV 3.0 APIs, when the service-context-id used in these routes is in the TV 2.X format, the API response should follow the same response defined in ABNT 15606-11. In these cases, the TV30- prefix should be omitted.

Additionally, for APIs which support the reading and writing of persistent values of an application, when the service-context-id used in these routes is in the TV 2.X format they should allow the reading and writing of such values for all the groups supported by TV 2.X, namely the shared, channel and service groups.

Hence, these recommendations apply in the following cases:

- 1) API C.6.4.1, defined in ABNT NBR 25608:2025, C.6.4.1. When service-context-id is in TV 2.X format, response should follow the definitions in API 8.3.1 of TV 2.X Ginga CC WebServices, defined in ABNT 15606-11, 8.3.1.
- 2) API C.6.4.2, defined in ABNT NBR 25608:2025, C.6.4.2. When service-context-id is in TV 2.X format, response should follow the definitions in API 8.3.2 of TV 2.X Ginga CC WebServices, defined in ABNT 15606-11, 8.3.2.
- 3) API C.6.4.5, defined in ABNT NBR 25608:2025, C.6.4.5. When service-context-id is in TV 2.X format, response should follow the definitions in API 8.3.5 of TV 2.X Ginga CC WebServices, defined in ABNT 15606-11, 8.3.5.
- 4) API C.6.4.6, defined in ABNT NBR 25608:2025, C.6.4.6. When service-context-id is in TV 2.X format, response should follow the definitions in API 8.3.6 of TV 2.X Ginga CC WebServices, defined in ABNT 15606-11, 8.3.6.

Annex D

NCLua for TV 3.0 Ginga-NCL

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Annex D apply.

D.1 Introduction

This Annex details and explains recommendations and behavior of NCLua for TV 3.0 Ginga-NCL, as supported by the TV 3.0 Application-oriented Platform defined in ABNT NBR 25608:2025. It provides operational guidelines for an implementation of NCLua for TV 3.0 Ginga-NCL according to ABNT NBR 25608:2025, Annex D.

D.2 NCLua Introduction

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section D.2 apply.

D.3 Lua library specification

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section D.3 apply.

D.4 Canvas API

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section D.4 apply.

D.5 Event API

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section D.5 apply.

D.6 Settings Data API

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section D.6 apply.

D.7 Persistent Data API

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section D.7 apply.

D.8 Directory API

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section D.8 apply.

D.9 PBDS API

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section D.9 apply.

D.10 Charset API

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section D.10 apply.

D.11 ZIP API

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section D.11 apply.

D.12 Bit32 API

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section D.12 apply.

D.13 Buffer API

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section D.13 apply.

D.14 Network API

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section D.14 apply.

D.15 Lua 5.1 specification

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section D.15 apply.

D.16 Application program interface (API)

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section D.16 apply.

D.17 Auxiliary library

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section D.17 apply.

D.18 Standard libraries

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section D.18 apply.

D.19 Complete syntax of Lua

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section D.19 apply.

Annex E

Regulatory Compliance Requirements for Brazil

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Annex E apply.

E.1 Specific Provisions on the Application Catalog

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section E.1 apply.

In TV 2.X/TV 3.0 receivers processing TV 2.X transmissions, when a BAMT is detected, as defined in ABNT NBR 15603-2:2026, the `BAM@mandatoryVisible` attribute should be obtained from this table.

When no BAMT is present in a TV 2.X transmission, the `BAM@mandatoryVisible` attribute should be set to `false`, corresponding to the default value defined in ABNT NBR 25608:2025 Table E.1.

E.2 Common Platform for Public Communication and Digital Government on TV 3.0

For the purposes of this part of the Document, the requirements and definitions of ABNT NBR 25608:2025, Section E.2 apply.

E.3 Additional recommendations

Requirement ACFR-20 of ABNT NBR 25608:2025 establishes that the receiver shall perform dynamic updates of Bootstrap Applications periodically when the receiver is in standby mode, at a frequency to be defined by local implementation policies and guidelines.

For Brazil, it is recommended that the Bootstrap Applications from a newly launched TV 3.0 service be added to the Application Catalog at most one day after the service's launch. For this, the receiver can use periods when it is in standby mode to perform an automatic signal scan, considering the elapsed time of 24 hours since the last automatic scan performed under the same condition or since the manual scan initiated by the viewer via the Application Catalog.

If the receiver remains continuously turned on for more than 24 hours, it is not necessary for it to perform an automatic signal scan during this period to update the Bootstrap Applications in the Application Catalog. However, at the manufacturer's discretion and in order not to compromise the update process, the receiver may perform this scan if there is not a Bootstrap Application or Broadcaster Application consuming content from the tuner or if a second tuner is available for the operation.

In standby mode, the scan may be interrupted before completion if the receiver is turned on again. In this case, only the Bootstrap Applications identified up to the point of interruption should be added to the catalog; existing entries are retained in accordance with ACFR-21.

Bibliography

- [1] ISO/IEC 27560, *Privacy technologies – Consent record information structure*
- [2] W3C Community Group Final Report. Data Privacy Vocabulary (DPV) version 2.0. 2024-08-01. Available at <<https://www.w3.org/community/reports/dpvcg/CG-FINAL-dpv-20240801/>>
- [3] W3C Community Group Final Report. Personal Data Categories (PD) version 2.0. 2024-08-01. Available at <<https://www.w3.org/community/reports/dpvcg/CG-FINAL-pd-20240801/>>
- [4] SBTVD Forum. TV 3.0 – Data access and privacy guide. Available at <https://forumsbtvd.org.br/wp-content/uploads/2025/11/Guia-de-Acesso-a-Dados-e-Privacidade-Fórum-SBTVD_EN.pdf>

Version history

4th edition, October 2026. Adds guidelines on the following aspects:

- 7.10.1 General aspects (Audience Measurement Manager)
- 7.10.2 Audience measurement data specification for a data collection session
 - AMM Report: TV30AoP-pmplayer.contentID property
- 8.2 Functional requirements (Application Catalog)
- 8.7 Profile Manager
- 9 Remote devices
- A.3.5 NCL Components API
- A.3.22 NCL Editing Commands API
- C.6.2 Event notification APIs
- C.6.6.2 API for controlling a specific media player
 - Applicability of control actions by content type

3rd edition, July 2026. Adds guidelines on the following aspects:

- 7.10.2 Audience measurement data specification for a data collection session
 - AMM Report: TV30AoP-foregroundAppId property
- 8.4.4 Bootstrap Application Manifest
- 8.8.5 Privacy management graphical views
- 12 Guidelines for building Receiver Platform UIs
- C.6.8.1 Registering a DTV service context for the use of TV 3.0 WebServices APIs
- C.6.11.1 Caching broadband DASH media segment files
- C.6.12.1 Resolution of Alternative MPDs

2nd edition, April 2026. Adds guidelines on the following aspects:

- 7.6 Persistent Media Player
- 8.8.7 Delivery of privacy notice via broadcast
- 8.8.8 Privacy-related XML schema files
- 10.6 Remote-control functions
- A.3.6 NCL Interfaces API
- C.6.6.1 API for obtaining available media players and API for obtaining information about a specific media player
- C.6.6.2 API for controlling a specific media player
 - playbackRate field
- C.6.7 APIs for integration with the platform and home environment
- E.3 Additional recommendations

1st edition, January 2026. Initial text, includes guidelines on the following aspects:

- 8.4 Bootstrap Applications
- 8.8 Privacy Manager
- 11 Harmonization of TV 2.X/TV 3.0 support
- 11.1 Application Execution Environment
- 11.2 Electronic Programming Guide (EPG)
- A.5 Harmonization of TV 2.X/TV 3.0 support
- A.5.1 NCL Preparation Event
- C.7 Harmonization of TV 2.X/TV 3.0 support
- E.1 Specific Provisions on the Application Catalog