

***CONTENTS***

<b>I. pbcoreAssetType - New Element!</b>	<b>2</b>
<b>II. Instantiations no longer required!</b>	<b>4</b>
<b>III. Relationship Definitions – additional documentation</b>	<b>6</b>
<b>IV. Source ID – additional documentation (also Multi-source ID)</b>	<b>7</b>
<b>V. Use of controlled vocabularies in PBCore – additional documentation</b>	<b>8</b>
<b>VI. Genre – additional documentation</b>	<b>10</b>
<b>VII. Description – additional documentation</b>	<b>11</b>
<b>VIII. Tip - How to package PBCore Records</b>	<b>12</b>
<b>IX. Note: Changes to attributed Strings</b>	<b>12</b>

## I. pbcoreAssetType - New Element!

*(In response to Digital Dawn Requirement S01.01 Definition of an "Asset" and S01.02 Varied Content Types)*

**New Element:** pbcoreAssetType

No longer does your titleType, descriptionType, formatMediaType or your formatGenerations need to answer the question "what is this PBCore record as a whole?" Now you can use assetType to define it and leave those more specific "type" fields to do their more "specific" jobs!

---

### Documentation

**Element NAME:** pbcoreAssetType

**Attributes available:** Source (as attribute of the element),  
Version (as attribute of the element)

**DEFINITION:** The descriptor pbcoreAssetType indicates the broad editorial format of what is described by the asset's contents. pbcoreAssetType describes the PBCore record as a whole and at its highest level. Though a record may contain many instantiations of different formats and generations, for example, pbcoreAssetType may be used to indicate that they all represent a "program" or a "clip." In the language of "FRBR" or the Functional Requirements for Bibliographic Records, pbcoreAssetType would be used to describe an asset at the "work" level, whereas "formatMediaType" would describe the "item" level. (For more on FRBR see <http://www.ifla.org/en/publications/functional-requirements-for-bibliographic-records>).

This element is largely based on the EBUCore element ObjectType:  
[http://www.ebu.ch/metadata/cs/ebu\\_ObjectTypeCodeCS.xml](http://www.ebu.ch/metadata/cs/ebu_ObjectTypeCodeCS.xml)

#### REFINEMENTS & ENCODING SCHEMES:

(Adapted from EBUCore)

Brand	Clip
Program	Media Object
Program Group	Track
Collection	Project
Series	Number
Episode	Series Number
Item	Episode Number
Story	Scene Number
Scene	Take Number
Shot	Framework
Element	Event
Segment	Trailer
	Pilot

**OBLIGATION TO USE:** Recommended, optional

**REPEATABLE ELEMENT:** Apply once, pbcoreAssetType (can be applied 0-1 times per PBCore record)

**TYPE OF DATA ENTRY:** Text String

## PBCore 1.3 Documentation – 7.30.10

### *Show Me Examples*

```
<pbcoreAssetType source="PBCore AssetType Picklist" version="1.0">  
Program</pbcoreAssetType>
```

### *Other Attributes of this Element*

**ELEMENT LABEL:** pbcoreAssetType

**ELEMENT VERSION:** PBCore v1.3

**NAMESPACE IDENTIFIER:** Public Broadcasting Metadata Dictionary Project Namespace  
<http://www.pbcore.org/PBCore/PBCoreNamespace.html>

**REGISTRATION AUTHORITY:** PBMD Project [pending]

**LANGUAGE OF THE ELEMENT:** en

## II. Instantiations no longer required!

*(In response to Digital Dawn Requirement S01.03 Abstract Assets)*

In response to a call for the ability to create PBCore records that do not necessarily correspond to a physical or digital asset, the Instantiation content class, and therefore the pbcoreInstantiation container, is no longer required. By the trickle down effect, if you are *not* using the pbcoreInstantiation container, the elements contained by pbcoreInstantiation are no longer mandatory.

This allows for PBCore to be used as a solely descriptive metadata structure without technical requirements. This allows PBCore users to create relationships and express conceptual levels of hierarchy, such as using one PBCore asset to express metadata about a 'Series', with no instantiation, and relate that to other PBCore assets which represent the 'Episodes' of the 'Series, where each episode has one or many instantiations. In earlier versions of PBCore, expressing metadata about a series was challenging since each asset was required to contain instantiation data while 'Series' themselves do not typically have direct associations with Instantiations. Again, in FRBR terms, you can now create a work level record and you are not required to document a physical item within it.

However, if an instantiation is used, PBCore still has the same requirements within the instantiation container as in PBCore 1.2. Each instantiation used still must include formatIdentifier, formatIdentifierSource, and a formatLocation.

Here are two example related records. The first represents a series and does not have an instantiation. The second represents a program which is part of the series and records an instantiation on Digital Betacam.

### **Series: Lidia's Italy**

```
<PBCoreDescriptionDocument>
  <pbcoreAssetType source="PBCore AssetType Picklist 1.0">
    <assetType>Series</assetType>
  </pbcoreAssetType>
  <pbcoreIdentifier>
    <identifier>LDIT</identifier>
    <identifierSource>NOLA</identifierSource>
  </pbcoreIdentifier>
  <pbcoreTitle>
    <title>Lidia's Italy</title>
    <titleType>Series</titleType>
  </pbcoreTitle>
  <pbcoreDescription>
    <description>Pack your bags, because you're going to Italy — Lidia's Italy! Journey with Lidia to her ten favorite regions around Italy as she introduces you to friends and family and takes you to food markets, fishing villages and farms as you haggle over the price of fish and forage for the perfect truffle. Then, return back to Lidia's familiar kitchen to prepare a sumptuous meal as you, the viewer cook along.</description>
    <descriptionType>Series</descriptionType>
  </pbcoreDescription>
</PBCoreDescriptionDocument>
```

### **Program: 115 - Bracirole-a Cinnamon Roll**

```
<PBCoreDescriptionDocument>
  <pbcoreAssetType source="PBCore AssetType Picklist 1.0">
    <assetType>Program</assetType>
```

## PBCore 1.3 Documentation – 7.30.10

```
</pbcoreAssetType>
<pbcoreIdentifier>
  <identifier>LDIT 000115</identifier>
  <identifierSource>NOLA</identifierSource>
</pbcoreIdentifier>
<pbcoreTitle>
  <title>Braciolo-a Cinnamon Roll</title>
  <titleType>Program</titleType>
</pbcoreTitle>
<pbcoreDescription>
  <description>Venice, the Veneto, spices! It was the center of the spice trade as Lidia loves to
  tell, and in this episode Lidia visits the region and then comes back to the kitchen where she
  doesn't hold back at all-making mouth watering beef cutlets garnished with aromatic cinnamon
  and cloves.</description>
  <descriptionType>Program</descriptionType>
</pbcoreDescription>
<pbcoreRelation>
  <relationType>Is Part of</relationType>
  <relationIdentifier>LDIT</relationIdentifier>
</pbcoreRelation>
<pbcoreInstantiation>
  <pbcoreFormatID>
    <formatIdentifier>KQED348-209-115</formatIdentifier>
    <formatIdentifierSource>KQED Tape Number</formatIdentifierSource>
  </pbcoreFormatID>
  <pbcoreFormatPhysical>Betacam Digital (Digi Beta)</pbcoreFormatPhysical>
  <formatLocation>KQED Vault</formatLocation>
  <pbcoreFormatGeneration>Moving image/Screening tape</pbcoreFormatGeneration>
  <pbcoreFormatDuration>00:29:52</pbcoreFormatDuration>
</pbcoreInstantiation>
</PBCoreDescriptionDocument>
```

### **III. Relationship Definitions – additional documentation**

*(In response to Digital Dawn Requirement S02.02 Relationship Definitions)*

Did you know you can relate one PBCore record to another? Check out the pbcoreRelation container and the elements it contains – relationType and relationIdentifier. Use the element “relationType” to define the relationship you want to express, and “relationIdentifier” allows you to indicate the record to which you are “relating.”

For example, a television series usually contains one or more programs. A program may contain one or more stock footage elements. You can express the relationship between these three items using the PBCore relationTypes vocabulary or “picklist” like this:

...in the record for the Series - American Experience or “AMEX”

```
<pbcoreRelation>
  <relationType>Has Part</relationType>
  <relationIdentifier>AMEX 001102</relationIdentifier>
</pbcoreRelation>
```

...in the record for the Program - American Experience program 1102, “Race for the Superbomb” or “AMEX1102”

```
<pbcoreRelation>
  <relationType>Is Part Of</relationType>
  <relationIdentifier>AMEX</relationIdentifier>
</pbcoreRelation>
```

...in the record for the footage – Los Alamos or “Tape 239872”

```
<pbcoreRelation>
  <relationType>Is Part Of</relationType>
  <relationIdentifier>AMEX 001102</relationIdentifier>
</pbcoreRelation>
```

Relationships can go both ways. You could express the above set of records from the bottom up or the top down. In addition, records can have multiple relationships. So the last one – for the footage, could also be considered part of the series – you could add that relationship. How you express the relationships between your records depends on how you want to exploit the relationships in your use of PBCore.

## IV. Source ID – additional documentation (also Multi-source ID)

*(In response to Digital Dawn Requirement S04.01 Source ID and S04.02 Multi-Source Identifiers)*

PBCore is flexible in that it allows you to choose any term from any source or system for its identifier source fields. You just need to declare where you got the identifier. The identifier plus its source helps to create and or maintain a unique identifier. Identifiers occur several times within PBCore and must be accompanied by an “identifier source” element. Within a container, you therefore indicate the identifier for your asset within the first element (identifier) and then you indicate where you got the identifier in the “identifierSource” element.

Basically – wherever the value of an element depends on the language or syntax of an external system or source, you want to indicate where you got that value. In this way, future users can re-visit your sources and use the same language, or at the very least understand where you got your values.

You may also indicate the version of a certain source or system by utilizing the “version” attribute within your source field. Add the attribute “version” within the source tag (See examples below).

Identifiers are used in the following container/element sets – see examples in right column:

Container	Elements	Examples
pbcoreIdentifier	identifier identifierSource	<pre>&lt;pbcoreIdentifier&gt; &lt;identifier&gt;KUER.1992-05-01.53467&lt;/identifier&gt; &lt;identifierSource&gt;KUER-FM 90 Library&lt;/identifierSource&gt; &lt;/pbcoreIdentifier&gt;</pre> <p>Or</p> <pre>&lt;pbcoreIdentifier&gt; &lt;identifier&gt;AMEX 001102&lt;/identifier&gt; &lt;identifierSource version="1998"&gt;PBS NOLA Code&lt;/identifierSource&gt; &lt;/pbcoreIdentifier&gt;</pre>
pbcoreFormatID	formatIdentifier formatIdentifierSource	<pre>&lt;pbcoreFormatID&gt; &lt;formatIdentifier&gt;KUED-5689.45.mov&lt;/formatIdentifier&gt; &lt;formatIdentifierSource&gt;KUED-TV 7 file system&lt;/formatIdentifierSource&gt; &lt;/pbcoreFormatID&gt;</pre> <p>or</p> <pre>&lt;pbcoreFormatID&gt; &lt;formatIdentifier&gt;239876&lt;/formatIdentifier&gt; &lt;formatIdentifierSource&gt;WNET Barcode&lt;/formatIdentifierSource&gt; &lt;/pbcoreFormatID&gt;</pre>

## V. Use of controlled vocabularies in PBCore – additional documentation

*(In response to Digital Dawn requirement S04.03 Multiple Controlled Vocabularies)*

PBCore is flexible in that it allows you to choose any term from any authority or controlled vocabulary for its authority fields. You just need to declare where you got the term. Authorities are called for several times within PBCore and are usually accompanied by an “authority used” element. Within a container, you indicate the descriptor for your asset within the first element – the “what” about this record. Next, you indicate where you got that descriptor, or what “language” you’re using to describe the asset in the “authority used” element.

Basically – wherever you select a value from an external “list” of options or an external language, you want to indicate where you got that value. In this way, future users can revisit your sources and use the same language, or at the very least understand where you got your values.

You may also indicate the version of a certain authority or controlled vocabulary by utilizing the “version” attribute within your authority used field. Add the attribute “version” within the authority used tag (See examples below).

Authorities are used in the following container/element sets – see examples in right column:

Container	Elements	Examples
pbcoreSubject	subject subjectAuthorityUsed	<pre>&lt;pbcoreSubject&gt; &lt;subject&gt;Prehistoric Native American culture&lt;/subject&gt; &lt;subjectAuthorityUsed&gt;Library of Congress Subject Headings&lt;/subjectAuthorityUsed&gt; &lt;/pbcoreSubject&gt;</pre> <p>Or</p> <pre>&lt;pbcoreSubject&gt; &lt;subject&gt;Genetic Processes&lt;/subject&gt; &lt;subjectAuthorityUsed&gt;NLM Medical Subject Heading D039361&lt;/subjectAuthorityUsed&gt; &lt;/pbcoreSubject&gt;</pre>
pbcoreGenre	genre genreAuthorityUsed	<pre>&lt;pbcoreGenre&gt; &lt;genre&gt;Documentary&lt;/genre&gt; &lt;genreAuthorityUsed version="1.2"&gt;PBCore Genre Picklist&lt;/genreAuthorityUsed&gt; &lt;/pbcoreGenre&gt;</pre> <p>Or</p> <pre>&lt;pbcoreGenre&gt; &lt;genre&gt;Film posters&lt;/genre&gt; &lt;genreAuthorityUsed&gt;http://www.loc.gov/rr/print/tgm2/&lt;/genreAuthorityUsed&gt; &lt;/pbcoreGenre&gt;</pre>

In addition, you can repeat each of the above containers in order to include more than one value from the same or different authorities in one record. You can choose all of your subjects from one list, or mix and match depending on your needs and your application of PBCore.

## **PBCore 1.3 Documentation – 7.30.10**

**Finally, a note:** for many elements, PBCore includes a picklist of values. It is important to know that PBCore users are not required to use these picklists (except in the case of coverageType) – that they are recommendations only. If you have an alternative term, feel free to use it. If you would like to suggest the term to the PBCore team, or if you would like to suggest a new way of incorporating controlled vocabularies into PBCore elements please submit a change request.

## VI. Genre – additional documentation

*(In response to Digital Dawn requirement S05.01 Genre Element Set)*

“Genre” is a difficult thing to define and many communities use it in different ways. For PBCore, we have suggested an amalgamation of lists which contain both editorial and format descriptors. PBCore users are not required to use the PBCore genre pick list – it is merely a recommendation.

Therefore PBCore users have 4 options:

1. use a term from the PBCore Genre Picklist
2. use a different term and cite the source of your term in the genreAuthorityUsed element. (Eg. use MARC for library data exchange, Netflix for commercial purposes)
3. Ignore genre altogether – it is not a requirement for PBCore
4. Repeat genre – pbcoreGenre is repeatable and each genre can have it’s own authority so you can pick multiple genres. These genres may come from the same or different authorities.

Finally, should you cite the source of your genre in the genreAuthorityUsed element, you can also indicate the version of your authority by adding the version attribute. See the example below:

Repeating pbcoreGenre within one PBCore record:

```
<pbcoreGenre>  
  <genre>Documentary</genre>  
  <genreAuthorityUsed version="PBCore 1.2">PBCore Genre Picklist</genreAuthorityUsed>  
</pbcoreGenre>
```

```
<pbcoreGenre>  
  <genre>History</genre>  
  <genreAuthorityUsed version="PBCore 1.2"> PBCore Genre Picklist</genreAuthorityUsed>  
</pbcoreGenre>
```

```
<pbcoreGenre>  
  <genre>motion picture</genre>  
  <genreAuthorityUsed>MARC Genre Term List</genreAuthorityUsed>  
</pbcoreGenre>
```

```
<pbcoreGenre>  
  <genre>PBS Documentaries</genre>  
  <genreAuthorityUsed>Netflix Documentary Sub-Genres</genreAuthorityUsed>  
</pbcoreGenre>
```

## VII. Description – additional documentation

(in response to Digital Dawn requirement S05.02 Description Element Set)

PBCore is flexible in that you can tailor the description field for your particular use. While you are only allowed one description per pbcoreDescription container, the container is repeatable. Each description is defined by an optional descriptionType value, see PBCore’s recommended descriptionTypes with definitions.

Use this flexibility to create PBCore records that address multiple uses. For example, you may have a “summary” description for your web accessible record, a “rundown” description for your editor, and third, less polished “just the facts” version specifically for internal use. For example:

Repeating pbcoreDescriptions in a record for *Rick Steve’s Europe*: “Barcelona and Catalunya”

```
<pbcoreDescription>
  <description>The creative spirit of Spain's Catalunya — the land of Picasso, Gaudi, and Salvador Dali — is
    on a roll. We'll get caught up in the festivity of Barcelona, enjoying the vibrant street scenes, tasty
    tapas, and pedestrian-friendly Gothic Quarter. Then we'll take a scenic side trip to mountaintop
    Montserrat, and finish with an artist's pilgrimage along the Costa Brava to Salvador Dali country. In
    seaside Cadaques, we'll visit Dali's home, and in nearby Figures, we'll experience his playfully surreal
    mausoleum/museum.
  </description>
  <descriptionType>Abstract</descriptionType>
</pbcoreDescription>

<pbcoreDescription>
  <description>00:03:10 – MS Steves in front of Gaudi building
    00:03:15 – Tilt up Gaudi building
    00:03:23 – CU Steves, gives show intro
  </description>
  <descriptionType>Rundown</descriptionType>
</pbcoreDescription>

<pbcoreDescription>
  <description>Rick Steves visit Pinotxo Bar, Picasso Museum, Sagrada Familia (Holy Family Cathedral) and
    Dali Theater-Museum in Barcelona and Catalunya</description>
  <descriptionType>Just the facts</descriptionType>
</pbcoreDescription>
```

## **VIII. Tip - How to package PBCore Records**

*(In response to Digital Dawn Requirement S1.04 Feed Creation)*

To containerize PBCore records into a collection of records for delivery, we recommend using the BagIt file packaging format developed and implemented by the Library of Congress. The Library has an [introductory video on BagIt](#) and the full specification and tools for using BagIt are available here. (<https://confluence.ucop.edu/display/Curation/BagIt>)

Other delivery options include the creation of an ATOM feed.  
(<http://tools.ietf.org/html/rfc4287> )

## **IX. Note: Changes to attributed Strings**

All Strings that allow for "version" attributes now include an additional "source" attribute  
e.g. `<creatorRole version="some version">creatorRole</creatorRole>`  
is now:

```
<creatorRole source="some source" version="some version">creatorRole
</creatorRole>
```