

Title	View configuration TwonkyMedia server 5.x		Version	Release Date
Author(s): PV	PacketVideo		5.0	Jan 2009

Page 1 of 31



## View configuration TwonkyMedia server

Release Date:

**Jan 2009**

Author(s):

**Burkhard Gauss**

This document describes the configuration of the view engine used to create the navigation tree in TwonkyMedia 5.

**Copyright © 2009 by PacketVideo Corporation.**

This document and the information contained herein is CONFIDENTIAL INFORMATION of PacketVideo Corporation (PV) and any client that has been granted a license of this document by PV. This document shall not be used, published, disclosed, or disseminated outside of PV and licensed client in whole or in parts without written consent of PV.

Title	View configuration TwonkyMedia server 5.x		Version	Release Date
Author(s): PV	PacketVideo		5.0	Jan 2009

### **Revision History**

Version	Date	Author	Modifications
5.0.1	Jan 2009	Burkhard Gauss	TwonkyMedia 5.0.40 release
5.0.2	Feb 2009	Jim Pfeifer	Proof reading
5.0.3	Feb 2009	Christian Gran	Added XML files and fixed server nodes

Title	View configuration TwonkyMedia server 5.x		Version	Release Date
Author(s): PV	PacketVideo		5.0	Jan 2009

---

## **Table of Contents**

1	Introduction	4
2	Overview	5
2.1	Features and capabilities	5
2.2	Basic functionality	5
2.3	Adding a new view	6
2.4	Assigning a views to clients	6
3	Structure of the “.view.xml” file	7
3.1	The <view> node	8
3.2	The <include> node	8
3.3	The <navtree> node	8
3.4	The <container> nodes	9
3.4.1	Using <container> to specify tree structure	9
3.4.2	Using <container> to specify generated sub-trees	10
4	Building a simple navigation tree	15
4.1	The complete demo.view XML file	21
5	Database properties	23
6	Fixed ID strings	24
7	UPnP Classes	25
7.1	createClass	25
7.2	class	25
8	The default.view xml file from server/resources/views	26

---

Title	View configuration TwonkyMedia server 5.x		Version	Release Date
Author(s): PV	PacketVideo		5.0	Jan 2009

---

## 1 Introduction

With TwonkyMedia server version 5.0 the configuration mechanism for the navigation tree has been completely replaced. The navigation tree is no longer configured through the *.ini* configuration file. It is now configured by one or more XML documents located in the folder *resources/views*.

Each of these XML documents defines a view on the internal database of TwonkyMedia server. Based on these views TwonkyMedia server generates and maintains the different navigation tree structures.

This document covers the following:

- An overview of the capabilities of the view based navigation trees
- Details of the XML documents needed to define the views
- Provides a step-by-step tutorial on how to create a custom view

Section 2 provides an overview of how the views are configured and how additional views can be defined.

Section 3 provides a detailed explanation of the configuration properties that can be used to configure new views.

In Section 4 demonstrates a “step by step” tutorial on how to build a new view.

The last section gives a reference list of available database properties that may be used to build views.

---

Title	View configuration TwonkyMedia server 5.x		Version	Release Date
Author(s): PV	PacketVideo		5.0	Jan 2009

---

## 2 Overview

This section provides a brief overview on the capabilities of the view-based navigation tree configuration and explains how the view XML documents are structured.

### 2.1 Features and capabilities

The new features and capabilities of the view based navigation tree configuration are:

- A view can be restricted to contain only a subset of the available media items
- There are no limitations regarding the complexity of the views
- Almost all database properties can be used
- TwonkyMedia server supports any number of simultaneous views
- No database rebuilds are needed when switching views
- Views can be assigned on a per client-type basis

### 2.2 Basic functionality

On startup TwonkyMedia server searches the folder *resources/views* for files with the extension *view.xml*. Each of these files can define one or more views on the database. TwonkyMedia server creates a navigation tree for each of these views.

**Note: The views are updated when an item is added to, or removed from the database. If you change the *view.xml* document(s), you need to rebuild the database.**

All available views are listed in the drop down box on the *First Steps* configuration page of TwonkyMedia server.

The TwonkyMedia server comes with a default view file, *default.view.xml*, which contains the configuration for the three views:

*advanceddefault*, *ipodlike* and *simpledefault*

**Note: The *default.view.xml* should not be changed. Some functions of TwonkyMedia server are dependent on the proper setup of these three default views.**

---

Title	View configuration TwonkyMedia server 5.x		Version	Release Date
Author(s): PV	PacketVideo		5.0	Jan 2009

---

## 2.3 Adding a new view

To add an additional view:

- start by creating a new *view.xml* file
- copy it to the folder *resources/views* of TwonkyMedia server
- rebuild the database.

To activate changes made to a *view.xml* file you also need to rebuild the database.

## 2.4 Assigning a views to clients

Views can be assigned to classes of clients in the client adaptation layer of TwonkyMedia server.

To assign a view to a certain class or type of client you need to edit the file *resources/clients.db*. Search for the section that matches your client and add the line "DV:<view name>" to that section.

For example: To assign the view *simpledefault* to the client "Freecom MusicPal" search for the appropriate section and add the DV switch. The resulting section would look like this:

```
>>
NA:Freecom MusicPal
DD:MusicPal
AV:M
DB:FIX
DV:simpledefault
```

This will permanently assign the view *simpledefault* to all clients that are detected as "Freecom MusicPal", regardless of the view that is selected on the *First Steps* of the TwonkyMedia server configuration page.

---

Title	View configuration TwonkyMedia server 5.x		Version	Release Date
Author(s): PV	PacketVideo		5.0	Jan 2009

---

### 3 Structure of the “.view.xml” file

The basic structure of a *view.xml* file is:

```
<view name='simple' path='simple.view' >
  <include>
    <tree path='*' />
  </include>
  <navtree>
    <container> ... </container>
  </navtree>
</view>
```

#### <view>

The view node is the root node of each *view.xml* document.

#### <include>

The include section will be used in future versions to restrict the view on certain data sources. For now leave this unchanged.

#### <navtree>

The navtree node encloses the actual navigation tree setup. It resembles the root container (id=0) of the navigation tree.

#### <container>

The container nodes are used to build the structure of the navigation tree. Each container node will result in one or more folders in the navigation tree.

There are two types of container nodes:

1. Nodes using the **name** attribute: These nodes are simply used to build the structure of the tree. Each of these nodes resembles exactly one folder in the navigation tree.
  2. Nodes using the **buildon** attribute: These nodes resemble a list of folders in the navigation tree. These folders are created based upon the database property specified in the buildon attribute.
-

Title	View configuration TwonkyMedia server 5.x		Version	Release Date
Author(s): PV	PacketVideo		5.0	Jan 2009

A more detailed description of the nodes and the available properties follows in the next sections.

### 3.1 The <view> node

The <view> node is the root node of the view.xml document. It has two mandatory attributes and should contain only the <include> and the <navtree> subnode.

Supported attributes of the view node are:

Attribute	Supported values Data type	Description
name	String	Symbolic name of the view, must be unique.
Path	String	Folder name that TwonkyMedia uses to store the view data.
subviews	Optional, Comma separated list	Can be used to define multiple views in one XML file. See also visibility attribute of the container node.

### 3.2 The <include> node

This section of the XML document will be used in future releases of TwonkyMedia. Currently it should look like this:

```
<include>
  <tree path='*' />
</include>
```

### 3.3 The <navtree> node

The <navtree> node resembles the root container of the navigation tree. With the attribute *sortcriteria* the default sort order of the items on the top level of the navigation tree can be defined. This allows, together with the attribute *upnp:originalTrackNumber* of the container nodes, to define any order for the containers on the top level. (See also the <container> section.)

Attribute	Supported values Data type	Description
sortcriteria	String, Optional	Used to define the default sort order for the top level of the navigation tree. The syntax follows the UPnP conventions: + <i>propertyname</i> use this property to sort ascending - <i>propertyname</i> use this property to sort descending



Title	View configuration TwonkyMedia server 5.x		Version	Release Date
Author(s): PV	PacketVideo		5.0	Jan 2009

---

(If a client adds an own sort order string to its requests, then this is used.)

---

### 3.4 The <container> nodes

The <container> node can be used in two different ways:

#### 3.4.1 Using <container> to specify tree structure

If the *name* attribute is present in the container node, a single folder will be created based upon this node. This can be used to build the basic navigation hierarchy or to create simple containers that contain all items of a certain upnp class.

<i>Attribute</i>	<i>Supported values Data type</i>	<i>Description</i>
name	String	Name of the folder which is created based upon this node.
visibility	Optional, Comma separated list of view names	If multiple sub views are defined in the <view> node, this property can be used to restrict the container to one ore more of these sub views. If it is missing the container is added to all sub views.
sortcriteria	Optional, String	Used to define the default sort order for the children of this container. The syntax follows the UPnP conventions: + <i>propertyname</i> use this property to sort ascending - <i>propertyname</i> use this property to sort descending
Id	Optional, String	(If a client adds an own sort order string to its requests, then this is used.) A fixed id string that will be mapped to the folder resulting from this container node. This allows certain clients to directly access these folders. See appendix for a list of predefined id strings.
class	String	The upnp class that will be used for the folder created based upon this container node.
createClass	Comma separated list	List of the upnp classes of the items that are allowed below this container.

Title	View configuration TwonkyMedia server 5.x		Version	Release Date
Author(s): PV	PacketVideo		5.0	Jan 2009

---

upnp:originalTrackNumber	Optional, Numeric	Allows to add an artificial track number to the container. This can be used together with the sortcriteria attribute to create a custom sort order.
--------------------------	----------------------	---

---

Example:

```
<container
  name='music'
  id='music'
  sortcriteria='+dc:title'
  upnp:originalTrackNumber='1'
  class='object.container'
  createClass='object.item.audioItem.musicTrack'>
</container>
```

- This line will create a folder called “music” with the id “music”.
- The items/folders inside this folder will be sorted alphabetically in ascending order.
- The music folder itself will have the track number 1. If the parent container uses the default sortcriteria “+upnp:originalTrackNumber” the music folder will be the first one in the list.
- The upnp:class of the music folder will be “object.container”
- Only items with the upnp:class “object.item.audioItem.musicTrack” will be located in the music folder
- Since the container node has no children, all items matching the createClass attribute will be located directly in this folder. Which means: All the music tracks are located directly in the music folder.

### 3.4.1.1 Filtering

With the *filteron* property, items that are allowed within a named container can be restricted. In the example above *filteron='pv:highrated'* would limit the music tracks in the music folder to those that have a value in the database field *pv:highrated*.

### 3.4.2 Using <container> to specify generated sub-trees

If the *buildon* attribute is used in a container node, TwonkyMedia will dynamically generate a list of folders based upon the specified database property.

The supported attributes are:

<i>Attribute</i>	<i>Supported values Data type</i>	<i>Description</i>
buildon	String	Name of the database property that is used to create a list of folders.

---

Title	View configuration TwonkyMedia server 5.x		Version	Release Date
Author(s): PV	PacketVideo		5.0	Jan 2009

---

visibility	Optional, Comma separated list of view names	If multiple sub views are defined in the <view> node, this property can be used to restrict the container to one ore more of these sub views. If it is missing the container is added to all sub views.
sortcriteria	Optional, String	Used to define the default sort order for the children of this container. The syntax follows the UPnP conventions: <i>+propertyname</i> use this property to sort ascending <i>-propertyname</i> use this property to sort descending
id	Optional, String	(If a client adds an own sort order string to its requests, then this is used.) A fixed id string that will be mapped to the folder resulting from this container node. This allows certain clients to directly access certain folders. See appendix for a list of predefined id strings.
class	String	The upnp class that will be used for the folder created based upon this container node.
createClass	Comma separated list	List of the upnp classes of the items that are allowed below this container.
upnp:originalTrackNumber	Optional, Numeric	Allows to add an artificial track number to the container. This can be used together with the sortcriteria attribute to create a custom sort order.
albumart	Optional, Boolean	If the album art attribute is set to '1', then TwonkyMedia will use the albumart of the first item added to the container as albumart for the container itself.
restricted	Optional, Boolean	Allows setting of the restricted flag for the folder and all items below. If this is not set the value from the parent is used. If the whole view has now restricted flags set, the internal default restricted='1' is used.

---

Title	View configuration TwonkyMedia server 5.x		Version	Release Date
Author(s): PV	PacketVideo		5.0	Jan 2009

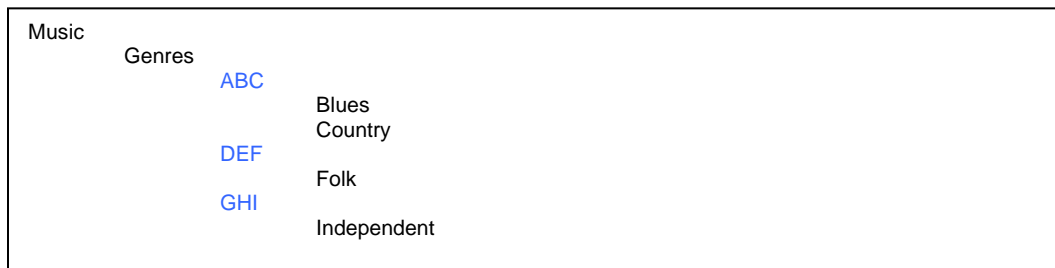
```
<container
  buildon='upnp:album'
  albumart='1'
  sortcriteria='+upnp:originalTrackNumber'
  createClass='object.item.audioItem.musicTrack'
  class='object.container.album.musicAlbum'
/>
```

This will result in a list of folders. The list will be created based on the database property 'upnp:album'. This means for each album found in the database a new folder will be created. TwonkyMedia then stores the items according to their album property in these folders.

### 3.4.2.1 Alpha-Grouping

If the resulting list of folders is too long, you can use the alpha-grouping feature to group these folders by their first character into a list of additional subfolders.

A navigation tree using alpha-grouping would look like this:

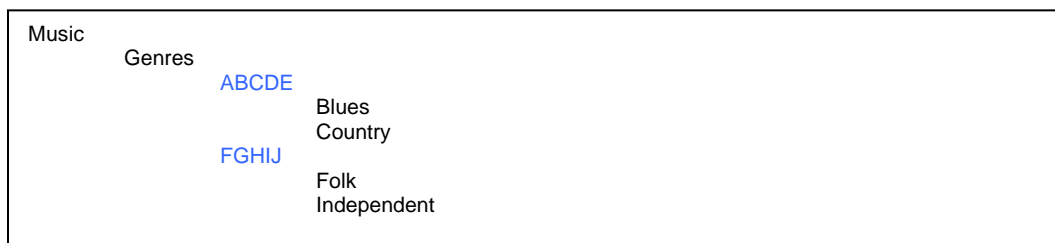


To enable alpha-grouping append a single digit in square brackets after the database property name in the buildon attribute.

The above example uses buildon='upnp:genre[3]'

The number defines the size of the groups.

For example buildon='upnp:genre[5]' would result in this tree:



Title	View configuration TwonkyMedia server 5.x		Version	Release Date
Author(s): PV	PacketVideo		5.0	Jan 2009

---

### 3.4.2.2 Substring Matching

A second way to automatically group a list of generated folders is the Substring matching feature. This feature works quite similar to the alpha-grouping, but it allows you to use any part of the string.

To enable substring matching append two comma separated numbers in square brackets after the database property name in the buildon attribute.

The first number is the starting point for the matching, the second number is the number of characters used for the matching.

For example: The default views are using this feature to create the date based navigation trees.

UPnP date strings are using this scheme to store a date YYYY-MM-DD

With buildon='dc:date[1:4]' the substring matching would match only on the YYYY part of the date strings. The result is an additional folder level based on the years in the date strings:



With buildon="dc:date[6:2]" the substring matching would only match the MM part of the date strings. Combined with the example from above this results in a navigation tree like this:

Title	View configuration TwonkyMedia server 5.x		Version	Release Date
Author(s): PV	PacketVideo		5.0	Jan 2009

,

---

Music					
	Date	2008			
			01	Artist	
				Album	
			04	Artist	
				Album	

---

Title	View configuration TwonkyMedia server 5.x		Version	Release Date
Author(s): PV	PacketVideo		5.0	Jan 2009

---

## 4 Building a simple navigation tree

This section provides a “step by step” tutorial showing how to configure a new view. The included file *demo.view.xml* contains the completed edits that are explained throughout this tutorial.

To start with a view, create the basic structure of the view.xml file:

```
<view name='demo' path='demo.view' >
  <include>
    <tree path='*' />
  </include>
  <navtree>
  </navtree>
</view>
```

Save this to the *resources/views* folder and restart the TwonkyMedia server. You will see the string “demo” in the drop down box for the navigation tree selection. If you select demo, the resulting navigation tree will be empty.

The next step is to add the three top-level containers for pictures, music and video files:

```
<view name='demo' path='demo.view' >
  <include>
    <tree path='*' />
  </include>
  <navtree sortcriteria='+upnp:originalTrackNumber'>
    <container name='music'
      id='music'
      sortcriteria='+dc:title'
      upnp:originalTrackNumber='3'
      createClass='object.item.audioItem.musicTrack'>
    </container>
    <container name='pictures'
      id='picture'
      sortcriteria='+dc:title'
      upnp:originalTrackNumber='2'
      createClass='object.item.imageItem.photo'
      class='object.container'>
    </container>
    <container name='videos'
      id='video'
      sortcriteria='+dc:title'
      upnp:originalTrackNumber='1'
      createClass='object.item.videoItem.movie'
      class='object.container'>
    </container>
  </navtree>
  ...
</view>
```

Title	View configuration TwonkyMedia server 5.x		Version	Release Date
Author(s): PV	PacketVideo		5.0	Jan 2009

The *sortcriteria* attribute of the navtree node, together with the *upnp:originalTrackNumber* attributes of the music, picture and video nodes are defining the default sort order of the resulting folders in the navigation tree:

Video, Pictures, Music

The attribute *createClass* ensures that each of these folders only contains items of the appropriate type. Since there are no further *container* nodes below the picture, music or video node, they will contain all items of the according type in one long list. The resulting navigation tree will look like this:



In the next step you will add an “All” folder to the music sub-tree, in addition to a folder that will contain an auto generated sub-tree based on the artist name.

```
<view name='demo' path='demo.view' >
  <include>
    <tree path='*' />
  </include>
  <navtree sortcriteria='+upnp:originalTrackNumber'>
    <container name='music'
      id='music'
      sortcriteria='+dc:title'
      upnp:originalTrackNumber='3'
      createClass='object.item.audioItem.musicTrack'
    >
      <container name='alltracks'
        id='music/all'
        sortcriteria='+dc:title'
        createClass='object.item.audioItem.musicTrack'
        class='object.container'
      />
      <container
        name='genre'
        id='music/genre'
        sortcriteria='+dc:title'
        class='object.container'
        createClass='object.item.audioItem.musicTrack'
      >
      <container name='artistindex'
        id='music/artistindex'
        sortcriteria='+dc:title'
        class='object.container'
        createClass='object.item.audioItem.musicTrack'
      >
      </container>
    </container>
  </navtree>
</view>
```



Title	View configuration TwonkyMedia server 5.x		Version	Release Date
Author(s): PV	PacketVideo		5.0	Jan 2009

This will create three additional folders below the music folder in the navigation tree. Each of these folders will contain all music tracks in a flat list:



The next step shows how to use the *buildon* attribute to generate two new sub-trees:

- First, we will use the *buildon* attribute to create one sub-tree based on the genre

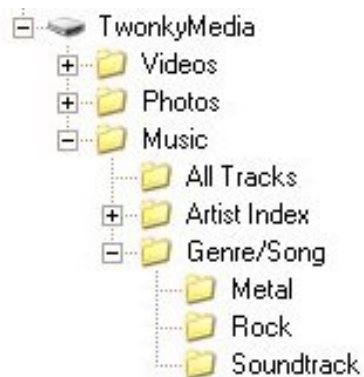
```

...
    <container
      name='genre'
      id='music/genre'
      sortcriteria='+dc:title'
      class='object.container'
      createClass='object.item.audioItem.musicTrack'
    >
      <container
        buildon='upnp:genre'
        sortcriteria='+dc:title'
        class='object.container.genre.musicGenre'
        createClass='object.item.audioItem.musicTrack'
      />
    </container>
...

```

This will add a list of folders based on the genres of your music items below the genre folder created in the previous step.

This picture below shows the resulting navigation tree:



Title	View configuration TwonkyMedia server 5.x		Version	Release Date
Author(s): PV	PacketVideo		5.0	Jan 2009

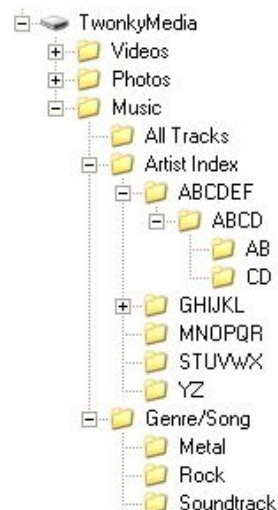
- Next, we will use the *buildon* attribute to create a sub-tree based on the artist name, using an alpha-index

```

...
    <container
      name='artistindex'
      id='music/artistindex'
      sortcriteria='+dc:title'
      class='object.container'
      createClass='object.item.audioItem.musicTrack'
    >
      <container
        buildon='upnp:artist[6]'
        sortcriteria='+dc:title'
        class='object.container'
        createClass='object.item.audioItem.musicTrack'
      >
        <container
          buildon='upnp:artist[4]'
          sortcriteria='+dc:title'
          class='object.container'
          createClass='object.item.audioItem.musicTrack'
        >
          <container
            buildon='upnp:artist[2]'
            sortcriteria='+dc:title'
            class='object.container'
            createClass=
              'object.item.audioItem.musicTrack'
          </>
        </container>
      </container>
    </container>
...

```

This adds three levels of artist name based alpha-index folders below the *artistindex* folder. On the first level a 6 character alpha-grouping is used for the artist name. To demonstrate a multi-level grouping, this example also includes a 4-character grouping (on the second level) and a 2-character grouping (on the third level). The resulting tree is shown below:



Title	View configuration TwonkyMedia server 5.x		Version	Release Date
Author(s): PV	PacketVideo		5.0	Jan 2009

The next step shows how to add a date-based navigation sub-tree to the pictures folder:

```

<container name='pictures'
  id='picture'
  sortcriteria='+dc:title'
  upnp:originalTrackNumber='2'
  createClass='object.item.imageItem.photo'
  class='object.container'>
  >
    <container
      name='allpictures'
      id='picture/all'
      sortcriteria='+dc:title'
      createClass='object.item.imageItem.photo'
      class='object.container'
    />
    <container
      name='bydate'
      id='picture/date'
      createClass='object.item.imageItem.photo'
      class='object.container'
    >
      <container
        buildon='dc:date[1:4]'
        createClass='object.item.imageItem.photo'
        class='object.container'
      >
        <container
          buildon='dc:date[6:2]'
          createClass='object.item.imageItem.photo'
          class='object.container.album.photoAlbum'
        />
      </container>
    </container>
  </container>
  ...

```

On the first level, this adds a simple folder to hold the date-based sub-tree to the pictures folder. On the next level, substring matching is used to build a list of folders from the first four characters of the date strings (this is the year). Below this level another list of folders is created from characters 6 and 7 of the date strings (this matches the month designation).



Title	View configuration TwonkyMedia server 5.x		Version	Release Date
Author(s): PV	PacketVideo		5.0	Jan 2009

The last step shows how to add a folder-based navigation sub-tree:

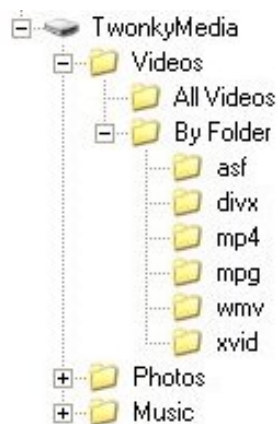
```

...
<container
  name='videos' id='video'
  sortcriteria='+dc:title'
  upnp:originalTrackNumber='1'
  createClass=
    'object.item.videoItem.movie,object.item.videoItem.online.movie'
  class='object.container'
>
  <container
    name='byfolder'
    id='video/folders'
    class='object.container.storageFolder'
    restricted='0'
    createClass='object.item.videoItem.movie'
  >
    <container
      buildon='res'
      sortcriteria='+dc:title'
      createClass='object.item.videoItem.movie'
      class='object.container.storageFolder'
    </container>
  </container>
</container>
...

```

This again uses a simple “*named*” container to create the Folder-folder that will contain the folder based sub-tree. TwonkyMedia handles the *buildon* directive on the *res* attribute a little different.

The *res* attribute contains the full path to the media item on disc. If the *buildon* is used together with the *res* attribute, TwonkyMedia splits the string stored in the *res* attribute along the path delimiters and builds the navigation tree on this.



Title	View configuration TwonkyMedia server 5.x		Version	Release Date
Author(s): PV	PacketVideo		5.0	Jan 2009

## 4.1 The complete demo.view XML file

```

<view name='demo' path='demo.view' >
  <include>
    <tree path='*' />
  </include>
  <navtree sortcriteria='+upnp:originalTrackNumber'>
    <container
      name='music'
      id='music'
      sortcriteria='+dc:title'
      upnp:originalTrackNumber='3'
      createClass=
        'object.item.audioItem.musicTrack,object.item.audioItem.online.musicTrack'
    >
      <container
        name='alltracks'
        id='music/all'
        sortcriteria='+dc:title'
        createClass='object.item.audioItem.musicTrack'
        class='object.container'
      />
      <container
        name='artistindex'
        id='music/artistindex'
        sortcriteria='+dc:title'
        class='object.container'
        createClass='object.item.audioItem.musicTrack'
      >
        <container
          buildon='upnp:artist[6]'
          sortcriteria='+dc:title'
          class='object.container'
          createClass='object.item.audioItem.musicTrack'
        >
          <container
            buildon='upnp:artist[4]'
            sortcriteria='+dc:title'
            class='object.container'
          >
            <container
              buildon='upnp:artist[2]'
              sortcriteria='+dc:title'
              class='object.container'
            >
              <container
                buildon='upnp:genre'
                sortcriteria='+dc:title'
                class='object.container.genre.musicGenre'
                createClass='object.item.audioItem.musicTrack'
              />
            </container>
          </container>
        </container>
      </container>
      <container
        name='genre'
        id='music/genre'
        sortcriteria='+dc:title'
        class='object.container'
        createClass='object.item.audioItem.musicTrack'
      >
        <container
          buildon='upnp:genre'
          sortcriteria='+dc:title'
          class='object.container.genre.musicGenre'
          createClass='object.item.audioItem.musicTrack'
        />
      </container>
    </container>
  </navtree>
</view>

```

Title	View configuration TwonkyMedia server 5.x		Version	Release Date
Author(s): PV	PacketVideo		5.0	Jan 2009

---

```

        <container
            name='pictures'
            id='picture'
            sortcriteria='+dc:title'
            upnp:originalTrackNumber='2'

createClass='object.item.imageItem.photo,object.item.imageItem.online.photo'
            class='object.container'
        >
            <container
                name='allpictures'
                id='picture/all'
                sortcriteria='+dc:title'
                createClass='object.item.imageItem.photo'
                class='object.container'
            />
            <container
                name='bydate'
                id='picture/date'
                createClass='object.item.imageItem.photo'
                class='object.container'
            >
                <container
                    buildon='dc:date[1:4]'
                    createClass='object.item.imageItem.photo'
                    class='object.container'
                >
                    <container
                        buildon='dc:date[6:2]'
                        createClass='object.item.imageItem.photo'
                        class='object.container.album.photoAlbum'
                    />
                </container>
            </container>
        </container>
        <container
            name='videos' id='video'
            sortcriteria='+dc:title'
            upnp:originalTrackNumber='1'

createClass='object.item.videoItem.movie,object.item.videoItem.online.movie'
            class='object.container'
        >
            <container
                name='allvideos'
                id='video/all'
                sortcriteria='+dc:title'
                createClass='object.item.videoItem.movie'
                class='object.container'
            />
            <container
                name='byfolder'
                id='video/folders'
                class='object.container.storageFolder'
                restricted='0'
                createClass='object.item.videoItem.movie'
            >
                <container
                    buildon='res'
                    sortcriteria='+dc:title'
                    createClass='object.item.videoItem.movie'
                    class='object.container.storageFolder'
                />
            </container>
        </container>
    </navtree>
</view>

```

---

Title	View configuration TwonkyMedia server 5.x		Version	Release Date
Author(s): PV	PacketVideo		5.0	Jan 2009

## 5 Database properties

A list of database properties that are supported in the view.xml files:

pv:avKeywords	Keywords
upnp:class	UpnP class of the item
dc:creator	Primary content creator or owner of the object
dc:date	The date of the item YYYY-MM-DD
dc:description	The description of the item
dc:title	Main title of the item
pv:orientation	Picture orientation
bitrate	Bitrate in bytes/seconds of the encoding of the resource
bitrateConstant	Flag to indicate constant or variable Bitrate
bitsPerSample	Encoding characteristic of the resource
chapterDuration	Chapter duration in ms
chapterFile	Filename of a DVD chapter (.vob)
chapterNumber	Chapter number
chapterOffset	Byte offset at the vob
colorDepth	Encoding characteristic of the resource.
duration	Time duration of the playback of the resource
pv:extension	File/url extension
contentType	The mime type of the item (if not auto detected)
nrAudioChannels	Number of audio channels
resolution	The resolution of the item (width x height)
sampleFrequency	Sample frequency of the audio in HZ.
size	Size in bytes of the resource
subtitle	If subtitle file (.srt/.sub/.txt/.smi/.ssa/.psb/.ass ) exists for this movie (same filename as movie file except extension) set this to the filename
pv:addedLast60	Set to 1, if added in last 60 days - auto calculated property!
pv:addedTime	Timestamp when the item was added (unix time)
upnp:albumArtURI	Object ID where the albumart is related - for containers only!
pv:albumArtLength	Length of album art - if there / for pictures used for thumbnails!
pv:albumArtOffset	Offset to album art - if there / for pictures used for thumbnails!
pv:albumArtResolution	XxY resolution of the albumArt in pixels (typically image item or video item).
pv:aspectRatio	The aspect ratio of the item
pv:compilation	Set to 1 if we have a compilation item
pv:sort	Default sort criteria for container
pv:highrated	Set if rating >4 - auto calculated property!
pv:lastPlayedTime	YYYY-MM-DD
pv:lastUpdated	The unix time when the db entry of the item was last updated
pv:modificationTime	The the modification time of the item (on disc) (Unix time)
pv:rating	Value between 1 (like) to 5 (don't like)
pv:playcount	Number of times an item was played
upnp:actor	Name of an actor appearing in a video item
upnp:albumArtist	Album Artist
upnp:album	Title of the album to which the item belongs.
upnp:artist	Name of an artist - always use with role=Performer here
upnp:author	Name of an author - always use role=Composer here
upnp:genre	Name of the genre to which an object belongs
upnp:originalTrackNumber	Original track number on an audio CD or other medium

Title	View configuration TwonkyMedia server 5.x		Version	Release Date
Author(s): PV	PacketVideo		5.0	Jan 2009

---

## 6 Fixed ID strings

music  
 music/all  
 music/playlists  
 music/genre  
 music/artists  
 music/albums  
 music/folders  
 music/rating  
 music/albumartists  
 music/composers  
 music/artistindex  
 music/genreindex  
 music/artistalbum  
 music/genreartistalbum  
 picture  
 picture/all  
 picture/playlists  
 picture/folders  
 picture/date  
 picture/albums  
 picture/keywords  
 picture/rating  
 video  
 video/all  
 video/playlists  
 video/genre  
 video/folders  
 video/rating

---



Title	View configuration TwonkyMedia server 5.x		Version	Release Date
Author(s): PV	PacketVideo		5.0	Jan 2009

---

## 7 UPnP Classes

### 7.1 createClass

For the createClass attribute these values are supported:

Videos: object.item.videoItem.movie

Online videos: object.item.videoItem.online.movie

Pictures: object.item.imageItem.photo

Online pictures: object.item.imageItem.online.photo

Audio: object.item.audioItem.musicTrack

Online audio: object.item.audioItem.online.musicTrack

### 7.2 class

For simple named containers: object.container

Folders that will hold pictures: object.container.album.photoAlbum

Use this when building a folder based tree: object.container.storageFolder

---

Title	View configuration TwonkyMedia server 5.x		Version	Release Date
Author(s): PV	PacketVideo		5.0	Jan 2009

## 8 The default.view xml file from server/resources/views

```

<view name='default' path='default.view'
subviews="advanceddefault,ipodlike,simpledefault">
    <include>
        <tree path='*' />
    </include>
    <navtree sortcriteria='+upnp:originalTrackNumber'>
        <container name='music' id='music' multiview='1' sortcriteria='+dc:title'
upnp:originalTrackNumber='1'
createClass='object.item.audioItem.musicTrack,object.item.audioItem.online.musicTrack'>
            <container name='alltracks' id='music/all' sortcriteria='+dc:title'
createClass='object.item.audioItem.musicTrack' class='object.container' />
            <container name='artistindex' visibility="advanceddefault"
id='music/artistindex' sortcriteria='+dc:title' class='object.container'
createClass='object.item.audioItem.musicTrack'>
                <container buildon='upnp:artist[3]' sortcriteria='+dc:title'
class='object.container' createClass='object.item.audioItem.musicTrack'>
                    <container buildon='upnp:artist'
sortcriteria='+dc:title' class='object.container.person.musicArtist'
createClass='object.item.audioItem.musicTrack'>
                        <container name='allname'
sortcriteria='+dc:title' createClass='object.item.audioItem.musicTrack'
class='object.container' />
                    <container buildon='upnp:album' albumart='1'
sortcriteria='+upnp:originalTrackNumber' createClass='object.item.audioItem.musicTrack'
class='object.container.album.musicAlbum' />
                </container>
            </container>
            <container name='artistalbum' visibility="advanceddefault"
id='music/artistalbum' sortcriteria='+dc:title'
class='object.container.person.musicArtist'
createClass='object.item.audioItem.musicTrack'>
                <container name='allname' buildon='upnp:album'
sortcriteria='+upnp:originalTrackNumber' createClass='object.item.audioItem.musicTrack'
class='object.container.album.musicAlbum' />
                <container buildon='upnp:artist' sortcriteria='+dc:title'
class='object.container.person.musicArtist'
createClass='object.item.audioItem.musicTrack'>
                    <container name='allname' sortcriteria='+dc:title'
createClass='object.item.audioItem.musicTrack' class='object.container' />
                    <container buildon='upnp:album' albumart='1'
sortcriteria='+upnp:originalTrackNumber' createClass='object.item.audioItem.musicTrack'
class='object.container.album.musicAlbum' />
                </container>
            </container>
            <container name='genre'
visibility="advanceddefault,ipodlike,simpledefault" id='music/genre'
sortcriteria='+dc:title' class='object.container'
createClass='object.item.audioItem.musicTrack'>
                <container buildon='upnp:genre' sortcriteria='+dc:title'
class='object.container.genre.musicGenre' createClass='object.item.audioItem.musicTrack'
/>
            </container>
            <container name='genreartist' visibility="advanceddefault"
id='music/genreartistalbum' sortcriteria='+dc:title' class='object.container'
createClass='object.item.audioItem.musicTrack'>
                <container buildon='upnp:genre' sortcriteria='+dc:title'
class='object.container.genre.musicGenre' createClass='object.item.audioItem.musicTrack'>
                    <container name='allname' sortcriteria='+dc:title'
createClass='object.item.audioItem.musicTrack' class='object.container' />
                    <container name='allname'
sortcriteria='+dc:title' createClass='object.item.audioItem.musicTrack'
class='object.container' />
            </container>
        </container>
    </navtree>
</view>

```

Title	View configuration TwonkyMedia server 5.x		Version	Release Date
Author(s): PV	PacketVideo		5.0	Jan 2009

```

        <container buildon='upnp:album' albumart='1'
sortcriteria='+upnp:originalTrackNumber' createClass='object.item.audioItem.musicTrack'
class='object.container.album.musicAlbum' />
    </container>
    <container buildon='upnp:artist'
sortcriteria='+dc:title' class='object.container'
createClass='object.item.audioItem.musicTrack'>
        <container name='allname'
sortcriteria='+dc:title' createClass='object.item.audioItem.musicTrack'
class='object.container' />
        <container buildon='upnp:album' albumart='1'
sortcriteria='+upnp:originalTrackNumber' createClass='object.item.audioItem.musicTrack'
class='object.container.album.musicAlbum' />
    </container>
</container>
<container name='artist' visibility="ipodlike,simpledefault"
id='music/artists' sortcriteria='+dc:title' class='object.container'
createClass='object.item.audioItem.musicTrack'>
    <container name='allname' sortcriteria='+dc:title'
createClass='object.item.audioItem.musicTrack' class='object.container' />
    <container buildon='upnp:artist' sortcriteria='+dc:title'
class='object.container.person.musicArtist' createClass='object.item.audioItem.musicTrack'
/>
</container>
<container name='album' visibility="ipodlike,simpledefault"
id='music/albums' sortcriteria='+dc:title' createClass='object.item.audioItem.musicTrack'
class='object.container' >
    <container buildon='upnp:album' albumart='1'
sortcriteria='+upnp:originalTrackNumber' createClass='object.item.audioItem.musicTrack'
class='object.container.album.musicAlbum' />
    </container>
    <container name='composer' visibility="ipodlike"
id='music/composers' sortcriteria='+dc:title'
createClass='object.item.audioItem.musicTrack' class='object.container' >
        <container buildon='upnp:author@role'
sortcriteria='+dc:title' createClass='object.item.audioItem.musicTrack'
class='object.container.album.musicAlbum' />
    </container>
    <container name='folder' visibility="simpledefault,advanceddefault"
id='music/folders' class='object.container.storageFolder'
createClass='object.item.audioItem.musicTrack'>
        <container buildon='res'
sortcriteria='+upnp:originalTrackNumber,+dc:title'
createClass='object.item.audioItem.musicTrack' class='object.container.storageFolder' />
    </container>
    <container name='playlists' id='music/playlists'
class='object.container'
createClass='object.item.audioItem.musicTrack,object.item.audioItem.online.musicTrack'
sortcriteria='+upnp:originalTrackNumber,+dc:title' restricted='0'
flags='pv:playlistRoot=Music' >
        <container name='playlistrecentlyadded'
filteron='pv:addedLast60' sortcriteria='-pv:addedTime'
createClass='object.item.audioItem.musicTrack' maxitems='500' upnp:originalTrackNumber='3'
restricted='1' class='object.container.playlistContainer' flags='pv:smartplaylist=1' />
        <container name='playlistmostplayed' filteron='pv:playcount'
sortcriteria='-pv:playcount' createClass='object.item.audioItem.musicTrack' maxitems='500'
upnp:originalTrackNumber='1' restricted='1' class='object.container.playlistContainer'
flags='pv:smartplaylist=1' />
        <container name='playlistlastplayed'
filteron='pv:lastPlayedTime' sortcriteria='-pv:lastPlayedTime'
createClass='object.item.audioItem.musicTrack' maxitems='500' upnp:originalTrackNumber='2'
restricted='1' class='object.container.playlistContainer' flags='pv:smartplaylist=1' />
        <container name='playlisthighlyrated'
filteron='pv:highrated' sortcriteria='-pv:rating'
createClass='object.item.audioItem.musicTrack' maxitems='500' upnp:originalTrackNumber='4'
class='object.container.playlistContainer' restricted='1' flags='pv:smartplaylist=1' />

```

Title	View configuration TwonkyMedia server 5.x		Version	Release Date
Author(s): PV	PacketVideo		5.0	Jan 2009

---

```

        <container buildon='upnp:playlist'
sortcriteria='+upnp:originalTrackNumber' upnp:originalTrackNumber='5'
createClass='object.item.audioItem.musicTrack,object.item.audioItem.online.musicTrack'
class='object.container.playlistContainer' />
    </container>
    <container buildon='pv:onlineServiceTitle'
sortcriteria='+pv:onlineServiceTitle'
createClass='object.item.audioItem.online.musicTrack' class='object.container'>
        <container name='allstations' sortcriteria='+dc:title'
createClass='object.item.audioItem.online.musicTrack' class='object.container' />
        <container name='genrestation' sortcriteria='+dc:title'
class='object.container' createClass='object.item.audioItem.online.musicTrack'>
            <container buildon='upnp:genre'
sortcriteria='+dc:title' class='object.container.genre.musicGenre'
createClass='object.item.audioItem.online.musicTrack' ></container>
            </container>
            <container name='stationindex' sortcriteria='+dc:title'
class='object.container' createClass='object.item.audioItem.online.musicTrack'>
                <container buildon='dc:title[3]'
sortcriteria='+dc:title' class='object.container'
createClass='object.item.audioItem.online.musicTrack'>
                    </container>
                </container>
            </container>
        </container>
    </container>
    <container name='pictures' id='picture' multiview='1'
sortcriteria='+dc:title' upnp:originalTrackNumber='2'
createClass='object.item.imageItem.photo,object.item.imageItem.online.photo'
class='object.container'>
        <container name='allpictures' id='picture/all'
sortcriteria='+dc:title' createClass='object.item.imageItem.photo'
class='object.container' />
        <container name='byfolder'
visibility="advanceddefault,simpledefault" id='picture/folders'
class='object.container.storageFolder' restricted='0'
createClass='object.item.imageItem.photo'>
            <container buildon='res' sortcriteria='+dc:title'
createClass='object.item.imageItem.photo' class='object.container.storageFolder' />
            </container>
            <container name='byyear' id='picture/year'
createClass='object.item.imageItem.photo' class='object.container'>
                <container buildon='dc:date[1:4]'
createClass='object.item.imageItem.photo' class='object.container.album.photoAlbum' />
                </container>
                <container name='bydate' id='picture/date'
createClass='object.item.imageItem.photo' class='object.container'>
                    <container buildon='dc:date[1:4]'
createClass='object.item.imageItem.photo' class='object.container'>
                        <container buildon='dc:date[6:2]'
createClass='object.item.imageItem.photo' class='object.container.album.photoAlbum' />
                        </container>
                    </container>
                </container>
            </container>
            <container name='album' visibility="ipodlike,simpledefault"
id='picture/albums' sortcriteria='+dc:title' createClass='object.item.imageItem.photo'
class='object.container' >
                <container buildon='upnp:album' albumart='1'
sortcriteria='+dc:title' createClass='object.item.imageItem.photo'
class='object.container.album.photoAlbum' />
                </container>
                <container name='keywords' visibility="ipodlike,simpledefault"
id='picture/keywords' sortcriteria='+dc:title' createClass='object.item.imageItem.photo'
class='object.container' >
                    <container buildon='pv:avKeywords' sortcriteria='+dc:title'
createClass='object.item.imageItem.photo' class='object.container.album.photoAlbum' />
                    </container>
                </container>
    </container>

```

---

Title	View configuration TwonkyMedia server 5.x		Version	Release Date
Author(s): PV	PacketVideo		5.0	Jan 2009

---

```

        <container name='rating' visibility='ipodlike,simpledefault'
id='picture/rating' sortcriteria='+dc:title' class='object.container'
createClass='object.item.imageItem.photo'>
        <container buildon='pv:rating' sortcriteria='+dc:title'
class='object.container.album.photoAlbum' createClass='object.item.imageItem.photo' />
    </container>
    <container name='slideshows' id='picture/playlists'
class='object.container'
createClass='object.item.imageItem.photo,object.item.imageItem.online.photo'
sortcriteria='+upnp:originalTrackNumber,+dc:title' restricted='0'
flags='pv:playlistRoot=Picture' >
        <container name='playlistrecentlyadded'
filteron='pv:addedLast60' sortcriteria='-pv:addedTime'
createClass='object.item.imageItem.photo' maxitems='500' upnp:originalTrackNumber='3'
restricted='1' class='object.container.playlistContainer' flags='pv:smartplaylist=1' />
        <container name='playlistmostviewed' filteron='pv:playcount'
sortcriteria='-pv:playcount' createClass='object.item.imageItem.photo' maxitems='500'
upnp:originalTrackNumber='1' restricted='1' class='object.container.playlistContainer'
flags='pv:smartplaylist=1' />
        <container name='playlistlastviewed'
filteron='pv:lastPlayedTime' sortcriteria='-pv:lastPlayedTime'
createClass='object.item.imageItem.photo' maxitems='500' upnp:originalTrackNumber='2'
restricted='1' class='object.container.playlistContainer' flags='pv:smartplaylist=1' />
        <container name='playlisthighlyrated'
filteron='pv:highrated' sortcriteria='-pv:rating'
createClass='object.item.imageItem.photo' maxitems='500' upnp:originalTrackNumber='4'
class='object.container.playlistContainer' restricted='1' flags='pv:smartplaylist=1' />
        <container buildon='upnp:playlist'
sortcriteria='+upnp:originalTrackNumber'
createClass='object.item.imageItem.photo,object.item.imageItem.online.photo'
upnp:originalTrackNumber='5' class='object.container.playlistContainer' />
    </container>
    <container buildon='pv:onlineServiceName'
sortcriteria='+pv:onlineServiceName' createClass='object.item.imageItem.online.photo'
class='object.container'>
        <container name='allname' sortcriteria='+dc:title'
createClass='object.item.imageItem.online.photo' class='object.container.album.photoAlbum'
/>
        <container buildon='pv:onlineServiceTitle'
sortcriteria='+pv:onlineServiceTitle' createClass='object.item.imageItem.online.photo'
class='object.container'>
            <container name='allname' sortcriteria='+dc:title'
createClass='object.item.imageItem.online.photo' class='object.container.album.photoAlbum'
/>
            <container name='keywords' sortcriteria='+dc:title'
createClass='object.item.imageItem.online.photo' class='object.container' >
                <container buildon='pv:avKeywords'
sortcriteria='+dc:title' createClass='object.item.imageItem.online.photo'
class='object.container.album.photoAlbum' />
            </container>
            <container name='album' sortcriteria='+dc:title'
createClass='object.item.imageItem.online.photo' class='object.container' >
                <container buildon='upnp:album'
sortcriteria='+dc:title' createClass='object.item.imageItem.online.photo'
class='object.container.album.photoAlbum' />
            </container>
            <container name='date'
createClass='object.item.imageItem.online.photo' class='object.container'>
                <container buildon='dc:date[1:4]'
createClass='object.item.imageItem.online.photo' class='object.container'>
                    <container name='allname'
sortcriteria='+dc:title' createClass='object.item.imageItem.online.photo'
class='object.container.album.photoAlbum' />
                    <container buildon='dc:date[6:2]'
createClass='object.item.imageItem.online.photo'
class='object.container.album.photoAlbum' />
                </container>
            </container>
        </container>
    </container>

```

---

Title	View configuration TwonkyMedia server 5.x		Version	Release Date
Author(s): PV	PacketVideo		5.0	Jan 2009

---

```

        </container>
        <container name='author' sortcriteria='+dc:title'
createClass='object.item.imageItem.online.photo' class='object.container' >
            <container buildon='upnp:author'
sortcriteria='+dc:title' createClass='object.item.imageItem.online.photo'
class='object.container.album.photoAlbum' />
        </container>
    </container>
</container>
</container>
    <container name='videos' id='video' multiview='1' sortcriteria='+dc:title'
upnp:originalTrackNumber='3'
createClass='object.item.videoItem.movie,object.item.videoItem.online.movie'
class='object.container'>
        <container name='allvideos' id='video/all' sortcriteria='+dc:title'
createClass='object.item.videoItem.movie' class='object.container' />
        <container name='album' visibility="ipodlike,simpledefault"
id='video/albums' sortcriteria='+dc:title' createClass='object.item.videoItem.movie'
class='object.container' >
            <container buildon='upnp:album' sortcriteria='+dc:title'
createClass='object.item.videoItem.movie' class='object.container' />
        </container>
        <container name='byfolder'
visibility="advanceddefault,simpledefault" id='video/folders'
class='object.container.storageFolder' restricted='0'
createClass='object.item.videoItem.movie'>
            <container buildon='res' sortcriteria='+dc:title'
createClass='object.item.videoItem.movie' class='object.container.storageFolder' />
        </container>
        <container name='byyear' id='video/year'
createClass='object.item.videoItem.movie' class='object.container'>
            <container buildon='dc:date[1:4]'
createClass='object.item.videoItem.movie' class='object.container' />
        </container>
        <container name='bydate' id='video/date'
createClass='object.item.videoItem.movie' class='object.container'>
            <container buildon='dc:date[1:4]'
createClass='object.item.videoItem.movie' class='object.container'>
                <container buildon='dc:date[6:2]'
createClass='object.item.videoItem.movie' class='object.container' />
            </container>
        </container>
        <container name='rating' id='video/rating'
visibility="ipodlike,simpledefault" sortcriteria='+dc:title' class='object.container'
createClass='object.item.videoItem.movie'>
            <container buildon='pv:rating' sortcriteria='+dc:title'
class='object.container' createClass='object.item.videoItem.movie' />
        </container>
        <container name='playlists' id='video/playlists'
class='object.container'
createClass='object.item.videoItem.movie,object.item.videoItem.online.movie'
sortcriteria='+upnp:originalTrackNumber,+dc:title' restricted='0'
flags='pv:playlistRoot=Video' >
            <container name='playlistrecentlyadded'
filteron='pv:addedLast60' sortcriteria='-pv:addedTime'
createClass='object.item.videoItem.movie' maxitems='500' upnp:originalTrackNumber='3'
restricted='1' class='object.container.playlistContainer' flags='pv:smartplaylist=1' />
            <container name='playlistmostviewed' filteron='pv:playcount'
sortcriteria='-pv:playcount' createClass='object.item.videoItem.movie' maxitems='500'
upnp:originalTrackNumber='1' restricted='1' class='object.container.playlistContainer'
flags='pv:smartplaylist=1' />
            <container name='playlistlastviewed'
filteron='pv:lastPlayedTime' sortcriteria='-pv:lastPlayedTime'
createClass='object.item.videoItem.movie' maxitems='500' upnp:originalTrackNumber='2'
restricted='1' class='object.container.playlistContainer' flags='pv:smartplaylist=1' />
            <container name='playlisthighlyrated'
filteron='pv:highrated' sortcriteria='-pv:rating'

```

---

Title	View configuration TwonkyMedia server 5.x		Version	Release Date
Author(s): PV	PacketVideo		5.0	Jan 2009

---

```

createClass='object.item.videoItem.movie' maxitems='500' upnp:originalTrackNumber='4'
restricted='1' class='object.container.playlistContainer' flags='pv:smartplaylist=1' />
    <container buildon='upnp:playlist'
sortcriteria='+upnp:originalTrackNumber'
createClass='object.item.videoItem.movie,object.item.videoItem.online.movie'
upnp:originalTrackNumber='5' class='object.container.playlistContainer' />
    </container>
    <container buildon='pv:onlineServiceName'
sortcriteria='+pv:onlineServiceName' createClass='object.item.videoItem.online.movie'
class='object.container'>
        <container buildon='pv:onlineServiceTitle'
sortcriteria='+pv:onlineServiceTitle' createClass='object.item.videoItem.online.movie'
class='object.container'></container>
        </container>
    </container>
</navtree>
</view>

```

---