



ContentDepotImport v. 9.20

1.0 Purpose

The ContentDepotImport(.exe) program will transfer .wav files from NPR's Content Depot appliance to AudioVAULT. The program polls the Content Depot to see if anything new has arrived, then imports the .wav files accordingly.

2.0 Requirements

ContentDepotImport.exe runs on a PC that has AudioVAULT 6.7 or better already installed.

3.0 Configuration

Obtain the program from Broadcast Electronics Digital Customer Service. Copy it into the C:\AUDIOVAU folder. Create shortcuts to it as needed (AudioVAULT, StartUp, etc).

The program's settings are stored in the AUDIOVAU.INI file under the section [ContentDepotImport].

The following settings are required:

- **Vault=** The HostName for the vault that you want to import the files to. For AV-100s this is something like PCR3. For Vault II this is something like PROD.
- **ContentDepotPath=** The UNC path (computer name and folder) to get .wav files from. This will be the Content Depot's computer name (AKA host name in UNIX speak), and the path where the .wav files are stored. The setting defaults to \\CDFile1\XDCache\CDLongname, where CDFile1 is the host name of the storage receiver and XDCache\CDLongname is the path to the .wav files. If the Content Depot has a static IP, you can use the IP vs. computer name if preferred. If the storage receiver's host name is not CDFile1, then an alternate form of ContentDepotPath= is possible; specify only the host name (in name or static IP format) as in ContentDepotPath=CDFile2 and the path is automatically appended internally.

3.1 Rules

Rules are used to configure ContentDepotImport to process the various audio programs that the Content Depot receives. Each rule must be given a unique name that a short description such as CarTalk, WaitWait, etc.

3.1.1 Rule Ordering

For each .wav file encountered on the Content Depot, the rules are checked in order, from top to bottom, for a CutID match. The first rule that includes the .wav file's CutID is used. For this reason, place the more specific rule.CutIDMask= values ahead of the more generic rules.

Correct order

[ContentDepotImport]
rule.CutIdMask=62671
rule.CutIdMask=6267#

[ContentDepotImport]
rule.CutIdMask=3267#
rule.CutIdMask=*

[ContentDepotImport]
rule.CutIdMask=74563
rule.CutIdMask=*

[ContentDepotImport]
rule.CutIdMask=77463-77467
rule.CutIdMask=*

[ContentDepotImport]
rule.CutIdMask=77463
rule.CutIdMask=7746[1-7]

[ContentDepotImport]
rule.CutIdMask=59201
rule.CutIdMask=59200-59209

Incorrect order

[ContentDepotImport]
rule.CutIdMask=6267#
rule.CutIdMask=62671

[ContentDepotImport]
rule.CutIdMask=*
rule.CutIdMask=3267#

[ContentDepotImport]
rule.CutIdMask=*
rule.CutIdMask=74563

[ContentDepotImport]
rule.CutIdMask=*
rule.CutIdMask=77463-77467

[ContentDepotImport]
rule.CutIdMask=7746[1-7]
rule.CutIdMask=77463

[ContentDepotImport]
rule.CutIdMask=59200-59209
rule.CutIdMask=59201

Outside of that, there is no restriction to the ordering of the rules found in [ContentDepotImport]. For example, you can put CarTalk.CutIdMask= either before or after WaitWait.CutIdMask=.

3.1.2 Duplicate Setting Names

Each setting name in [ContentDepotImport] (text to the left of the equal (=) sign) can be defined no more than once.

Correct

[ContentDepotImport]
Promos.CutIdMask=10000,10009
Promos.Gain=3

[ContentDepotImport]
Show1.CutIdMask=23450-23459
Show1.Category=SH1
Show2.CutIdMask=45929
Show2.Category=SH2

Incorrect

[ContentDepotImport]
Promos.CutIdMask=10000
Promos.Gain=3
Promos.CutIdMask=10009
Promos.Gain=3
;Incorrect, Promos.CutIdMask= and Promos.Gain= exist more than once

[ContentDepotImport]
Show1.CutIdMask=23450-23459
Category=SH1
Show2.CutIdMask=45929
Category=SH2
;Incorrect, Category= exists more than once.



3.1.3 Partitioning A Rule

When one or more individual .wav files in a rule require a setting that differs from the others in the rule, then create a new rule to handle the exception(s).

For example, if we have:

```
FreshAir.CutIdMask=62113-62117,62399-62403,62600-62604,62742-62746,61687-61691
FreshAir.FilePrefix=FA
```

And we decided that we want a different FilePrefix= per day, then we need to break that one all-encompassing rule into separate rules:

```
FreshAirMon.CutIdMask=62113-62117
FreshAirMon.FilePrefix=FAMON
FreshAirTue.CutIdMask=62399-62403
FreshAirTue.FilePrefix=FATUE
FreshAirWed.CutIdMask=62600-62604
FreshAirWed.FilePrefix=FAWED
FreshAirThu.CutIdMask=62742-62746
FreshAirThu.FilePrefix=FATHU
FreshAirFri.CutIdMask=61687-61691
FreshAirFri.FilePrefix=FAFRI
```

3.1.4 Rule Defaults

Many *rule.xyz=* values have a default that comes from a corresponding *xyz=* setting; the intention is to configure these generic *xyz=* settings once as dictated by the majority of the shows, then override them with a rule-specific setting, as needed. For example, if your plant's audio levels require a gain adjustment of +6 for the Content Depot-received cuts, then configure *Gain=6* and for each show that requires a different value, override the default with a new gain:

```
[ContentDepotImport]
Gain=6
Category=NPR
Show1.CutIdMask=23423-23425
Show2.CutIdMask=46564,47446
Show3.CutIdMask=57673,33536,23423
QuiteShow.CutIdMask=34567
QuiteShow.Gain=9
```

Default values are used when the rule-specific value is not defined. The following further demonstrates default values.

Settings	Comment
[ContentDepotImport]	
Gain=-8	
Show1.CutIdMask=10000	Rule uses Gain=-8
[ContentDepotImport]	
Show1.CutIdMask=10000	No gain specified for rule



[ContentDepotImport]

Gain=-8

Show1.CutIdMask=10000

Show1.**Gain=-6**

Rule uses Gain=-6, Gain=-8 is ignored by this rule

[ContentDepotImport]

Gain=-8

Show1.CutIdMask=10000

Show1.**Gain=-8**

Rule uses Show1.Gain=-8, Gain=-8 is ignored by this rule.
Avoid rule "clutter"

[ContentDepotImport]

Gain=-8

Show1.CutIdMask=10000

Show1.**Gain=**

Rule used Gain= while Gain=-8 is ignored

Pay special attention to this last example: If a rule defines a value equal to nothing, the default is not used. Defaults are only used when the rule doesn't declare the setting.

The various rule defaults include:

- o **Category=** The *rule.Category=* values default to this. Category= itself has no default. Example: Category=CDP
- o **NfsClass=** The *rule.NfsClass=* values default to this. NfsClass= proper defaults to blank (no class). Example: NfsClass=LOCAL
- o **FilePrefix=** The *rule.FilePrefix=* values default to this. FilePrefix= proper defaults to NPR. Example: FilePrefix=CDP
- o **Gain=** The *rule.Gain=* values default to this. Gain= proper defaults blank, meaning the cut plays back with no gain adjustment. Example: Gain=-8
- o **CheckVaultFileKillDate=** The *rule.CheckVaultFileKillDate=* values default to this. CheckVaultFileKillDate= proper defaults True, meaning the AudioVAULT file kill date is checked to ensure it is post-kill before it is overwritten by a new Content Depot file. False means the AudioVAULT file kill-date test is skipped. Example: CheckVaultFileKillDate=False
- o **CheckReceivedFileKillDate=** The *rule.CheckReceivedFileKillDate=* values default to this. CheckReceivedFileKillDate= proper defaults True, meaning the kill dates of new Content Depot files are checked to ensure the file is not already post-kill. True means the file is not imported if it is already post-kill. False means the received file post-kill test is skipped. Example: CheckReceivedFileKillDate=False
- o **NameMacro=** The *rule.NameMacro=* values default to this. NameMacro= itself defaults to {FilePrefix}{CutID}.
- o **KillDateMacro=** The *rule.KillDateMacro=* values default to this. KillDateMacro= itself has no default, meaning use the Producer-supplied end date.
- o **DescriptionMacro=** The *rule.DescriptionMacro=* values default to this. DescriptionMacro= itself defaults to {Title}. Example: DescriptionMacro={FileName}
- o **ClientIdMacro=** The *rule.ClientIdMacro=* values default to this. ClientIdMacro= itself defaults to {Artist}.
- o **CodesMacro=** The *rule.CodesMacro=* values default to this. CodesMacro= itself default to blank.



The newer description-based format is more flexible, easier to read and self documenting (you don't have to add comments), and is therefore preferred over the older category-based rule naming system.

New Format

CarTalk.Category=CTK
CarTalk.CutIdMask=10000-10007

Old Format

;Car Talk
CTK.CutIdMask=10000-10007

Note: AVMaint is used to add Category choices to the AudioVAULT system.

3.1.4 Rule Settings

The following settings can be used in each rule:

- o **rule.CutIdMask=** Required. .WAV files which Content Depot receives have a cart chunk. One field in the cart chunk is CutID, a 5-digit number. The PRSS website reports the CutIDs to you when registering to receive a show.

rule.CutIdMask= specifies which CutIDs the rule applies to (will try to import).

rule.CutIdMask= can be (a) a CutID such as 80201, or (b) a wildcard CutID such as 8020[1-6] or 802##, or (c) a range of CutIDs (smaller value listed first) such as 80201-80206.

If you need to list more than one CutID mask then separate each with a comma as in *rule.CutIdMask=80203,80201*. *rule.CutIdMask=* must have at least one mask; there is no default value. Comma-separated CutID masks need not be listed in any particular order. The length of this line may be up to 2000 characters.

Example: *Thistle.CutIdMask=60093-60095*

- o **rule.SegmentValue=** Optional. A comma-separated list of strings, one for each for each CutID in *CutIdMask=*. These strings are accessed within the various *xyzMacro=* settings via the {SegmentValue} curly brace special value. The intent of the *rule.SegmentValue=* setting is to allow a group of like shows to be handled under one rule, yet to visually differentiate each AudioVAULT cut in its name, description, etc.

The example that follows groups the five Writers Almanac shows into one rule named Writers. It uses *SegmentValue=* to name the cuts WRITERS-MON through WRITERS-FRI.

```
Writers.CutIdMask=61110-61114  
Writers.SegmentValue=MON,TUE,WED,THU,FRI  
Writers.NameMacro=WRITERS-{{SegmentValue}}
```

- o **rule.NfsClass=** Optional. The 1- to 7-letter class to assign to the cut. If not defined it defaults to the value of *NfsClass=*. Use AVMaint to add Class choices to the AudioVAULT system.



- **rule.FilePrefix=** Optional. As to avoid name conflicts between cuts imported from Content Depot and other AudioVAULT inventory, the resultant AudioVAULT cut name defaults to `<rule.FilePrefix> + <CartChunk.CutID>` (see `rule.NameMacro=`, below).

So, for example, if CutID in the cart chunk is 20805 and if `rule.FilePrefix=CTK`, the resultant vault file would be named CTK20805.

If not defined `rule.FilePrefix=` defaults to the value of `FilePrefix=`.

Example: `PianoJazz.FilePrefix=PIA`

- **rule.Gain=** Optional, a number or a negative number. The audio gain to assign to the cut, once imported. If not defined it defaults to the value of `Gain=`. `rule.Gain=` is expressed in dB, and can be greater than (louder) or less than (softer) 0. For example, `Gain=9` causes the cut to playback 9 dB louder, while `Gain=-6` plays back 6 dB softer.

Note that if the Producer of a show is inconsistent and the needed gain value varies from revision to revision of the show or even from segment to segment, then your only recourse is to contact that Producer and plead your case for standard, or at least consistent, levels.

Example: `AmericanRoutes.Gain=-10`

- **rule.CheckVaultFileKillDate=** Optional, True or False. When `ContentDepotImport` considers a Content Depot .wav file for import, several checks are made to the source and destination files. One check is to ensure the AudioVAULT file (in the case of an overwrite) is post-kill (is already beyond its kill date). This date originates from the Producer of a show. However, some Producers use a wide end/kill date. For example, a once-per-week show might have a kill date that is several weeks beyond its start date; a convention that isn't useful for controlling `ContentDepotImport` overwrites. You can disable this check if needed (set this option to False).

If `rule.CheckVaultFileKillDate=` is enabled, then remember to pay close attention to the kill/end date because you may have to specify `rule.KillDateMacro=` as well. If `rule.CheckVaultFileKillDate=` is not defined, it defaults to the value of `CheckVaultFileKillDate=`.

- **rule.CheckReceivedFileKillDate=** Optional, True or False. Another check is to ensure the Content Depot file is not already post-kill (is not yet beyond its kill date). You can disable this check by setting this option to False. True means the file is not imported if it is post-kill, while False means the post-kill test is skipped. If not defined, this option defaults to the value of `CheckReceivedFileKillDate=`.

The next group of settings can be used to adjust certain AudioVAULT cut meta data fields. This information normally originates from the Producer, and is delivered via the cart chunk within the .wav file. These settings allow a customized display of Content Depot-delivered cuts on your AudioVAULT. These settings usually includes certain special curly brace values such as {Title}, delineated in the next section. However, using these settings may come with some risk in that as Producers alter their meta data convention over time, you may have to likewise alter your setting.



In general, the more complex your value, the more susceptible it might be to misbehaving should meta data changes made by the Producer down the road. The KISS rule applies very nicely here. Like other rule.xyz settings, these default to the value of the corresponding xyz= value.

- o **rule.NameMacro=** The name to use when creating the AudioVAULT cut. If not defined it defaults to NameMacro=, which in turn defaults to {FilePrefix}{CutID}. Note that it is the intention of PRSS that automation systems name their files after CutID, possibly with a static file prefix prepended. See rule.FilePrefix=, above.

One common reason for configuring rule.NameMacro= is to assign the AudioVAULT cut a short, easy-to-remember, name such as <show abbreviation><segment number>, useful when the cut is often manually loaded into the automation system. Using Cart Talk as an example:

```
CarTalk.CutIdMask=51250-51256
```

```
CarTalk.NameMacro=CARTALK{Segment}
```

Here the resultant names for the 7 segments in each show would be CARTALK1, CARTALK2, ... CARTALK7.

Note that it is technically possible, but not necessarily recommended, to configure rule.NameMacro= such that a new AudioVAULT cut would result for each show (by using {Date}, for example); it is the responsibility of the customer to perform the needed file maintenance tasks such as deleting old, unneeded cuts, to ensure the system has sufficient disk space for the cuts, etc.

- o **rule.KillDateMacro=** Historically, Producers have used the kill date (i.e., end date) to indicate the number of days or weeks their show could run or re-run. But with the advent of electronic delivery via Content Depot, kill date may be better used to indicate the frequency of the show, such as once a week (see rule.CheckVaultFileKillDate=). The rule.KillDateMacro= setting specifies what kill date to apply to the AudioVAULT cut in lieu of what the Producer supplied via the cart chunk.

rule.KillDateMacro= must take on one of these formats:

- o **{StartDate}+<days>** The Producer's Start Date. If the start date of the show appears to be accurate, then you might consider setting the kill date relative to the start date. Ex: WaitWait.KillDateMacro={StartDate}+7
- o **{Date}+<days>** The date (when the import actually happens) from the PC's clock. To use this effectively, you need to have a knowledge base or history of when the show arrives on the Content Depot relative to when you air the show. Armed with that, plus knowing what day of the week you air the show, a value for <days> can be configured.

<days> is a number such as 7, etc. rule.KillDateMacro= defaults to KillDateMacro=, which in turn has no default (meaning use the Producer-supplied end date)

There are two reasons why you might choose to use rule.KillDateMacro=: **(a)** you want the cut to turn yellow in AudioVAULT and warn the operator when it is out of date, and/or **(b)** you want ContentDepotImport to be able to properly overwrite the vault file via kill date. For reason (b) be sure the corresponding rule.CheckVaultFileKillDate= is enabled as well.



Example: BealeStreet.KillDateMacro={Date}+5

The *rule*.CheckReceivedFileKillDate= behavior is unaffected by *rule*.KillDateMacro= since the first concerns the Content Depot file while the second concerns the AudioVAULT cut.

- o *rule*.DescriptionMacro= The AudioVAULT cut Description. *rule*.DescriptionMacro= defaults to DescriptionMacro=, which in turn defaults to {Title}.

Example: PriPromos.DescriptionMacro=PRI Promos - Fed {Date}

Cart chunk text fields can be up to 64 characters long, and a few Producers use every character of that in their title. However, some automation systems, AudioVAULT included, can't display all of those characters. *rule*.DescriptionMacro= allows you to reformat the description so that it displays reasonably on your AudioVAULT. A common value here is {FileName} because the Content Depot filename is short, concise, yet sufficiently descriptive. In fact some customers simply define DescriptionMacro={FileName} so that it applies to all rules.

If further customization is needed, then other options exist. Here is a hypothetical example; We subscribe to a show named Bike America and its title in the cart chunk meta data looks like: Bike Across America, September 20th, 2006, Segment number 3 of 4 (FYI: the cart chunk fields can be seen in the log file when TraceLevel= is 3 or more). Here it is again with a ruler. Bold indicates the portions that we would like in our final description:

```

      1           2           3           4           5           6
123456789 123456789 123456789 123456789 123456789 123456789 1234
Bike Across America, September 20th, 2006, Segment number 3 of 4
```

The desired effect can be achieved with something like:

```
rule.DescriptionMacro=Bike America, {Title.Mid(22,4)} {Title.Mid(32,2)}, Sgmt
{Title.Mid(59,1)}
```

which would result in the following:

```

      1           2
123456789 123456789 123456789
Bike America, Sept 20, Sgmt 3
```

- o *rule*.ClientIdMacro= The cut's AudioVAULT ClientID field. This defaults to ClientIdMacro=, which in turn defaults to {Artist}.
- o *rule*.CodesMacro= The cut's AudioVAULT Codes field. This defaults to CodesMacro=, which in turn defaults to blank.

3.1.4.1 Special {Curly Brace} Values

The *rule*.xyzMacro= values, above, (except for *rule*.KillDateMacro=) allow text substitutions using any of these special curly brace values.

- o **{Date}** The current date (when the .wav file is imported) from the PC's clock. This is formatted as yyyy/mm/dd.

Example: PriPromos.DescriptionMacro=PRI Promos - Fed {Date}



- **{Time}** The current time (when the .wav file is imported) from the PC's clock. This 24-hour time is formatted as hh:mm:ss.
- **{Weekday}** The current day of the week (when the .wav file is imported) as a number from 1 (for Sunday) - 7 (for Saturday).
- **{FileName}** The name of the .wav file, excluding the hostname, drive letter, path and extension. Example filenames are:
 - CarTalk_001_SGMT4
 - RadioRea_057_SGMT1
 - SoundSpi_001_SGMT3

This curly brace value is often used in *rule.DescriptionMacro=* or even *DescriptionMacro=*.

- **{Category}** The value of *rule.Category=*, above.
- **{FilePrefix}** The value of *rule.FilePrefix=*, above.
- **{RuleName}** The name of the rule (anything to the left of the period).
- **{Segment}** *rule.CutIdMask=* lists the CutIDs to import via that rule. **{Segment}** is a number that is the index into the list of *rule.CutIdMask=* values where the current show was found. For example, if we have *rule.CutIdMask=54013,92034,23247-23250* and if the cut's CutID is 92034 then **{Segment}** equals 2 because that CutID is the second CutID in the *rule.CutIdMask=* list. **{Segment}** is used in multi-segment shows to make user-friendly and easy-to-remember values via *rule.NameMacro=* or *rule.DescriptionMacro=*. Referring to *rule.CutIdMask=*, above, **{Segment}** is compatible with type (a) (static number) and (c) (range of static numbers), but not (b) (range via wild cards) such as 7434[56] nor 4358[1-9]. Likewise, it is compatible with comma-separated lists of (a) (static numbers) and (c) (range of static numbers).
- **{SegmentValue}** This is based on both **{Segment}** and the *rule.SegmentValue=* list, and provides a means to visually differentiate like AudioVAULT cuts in within a given rule. Where **{Segment}** is an index into the *rule.CutIdMask=* list, **{SegmentValue}** is an index into the *rule.SegmentValue=* list. See **{Segment}** and *rule.SegmentValue=*, above.
- **{Artist}** The Artist field from the cart chunk.
- **{ClientID}** The ClientID field from the cart chunk.
- **{CutID}** The CutID field from the cart chunk. See *rule.NameMacro=*, above.
- **{KillDate}** The End Date from the cart chunk. It is formatted as yyyy/mm/dd.
- **{KillTime}** The End Time from the cart chunk. This 24-hour time is formatted as hh:mm:ss.
- **{OutCue}** The Out Cue from the cart chunk.
- **{StartDate}** The Start Date from the cart chunk. It is formatted as yyyy/mm/dd.
- **{StartTime}** The Start Time from the cart chunk. This 24-hour time is formatted as hh:mm:ss.
- **{Title}** The Title from the cart chunk. See *rule.DescriptionMacro=*, above.



All of those special curly brace values offer the following variations:

- **{xyz.Left(#)}** .Left(#) is the left-most n characters of the value. For example, if {Date} was 2006/12/04, {Date.Left(4)} would be 2006.
- **{xyz.Right(#)}** .Right(#) is the right-most n characters of the value. For example, if {Date} was 2006/12/04, {Date.Right(2)} would be 04.
- **{xyz.Mid(##,##)}** .Mid(##,##) is a mid-portion of the value. The first number is the starting position, while the second is the number of characters to use. For example, if {Date} was 2006/12/04, {Date.Mid(6,2)} would be 12.

Note: Spaces in these variations are not permitted. Use xyz.Mid(2,4), not xyz.Mid(2 , 4)

3.2 Application Settings

There are other settings which default to reasonable values if absent from the section. You can change those if needed:

- **ScanInterval=** The interval in seconds to scan the Content Depot for new files. The default is 300 (5 * 60, 5 minutes). Note that the first scan after program start up is fixed at 10 seconds.
- **TraceLevel=** This controls how much detail regarding scanning & importing process gets logged. The default is 0 (just log imported filenames). When debugging, it can be increased (max. is 4) to get additional details. The log file is named c:\audiovau\log\ContentDepotImport\YMMH.log.
- **Uniqueld=** One possible need for running multiple instances of ContentDepotImport.exe on a single PC is when you want to import Content Depot files to multiple vaults. If you need to run multiple instances, then two changes are needed.

First, the shortcut's command line to the program needs to specify an alternative configuration name (default is ContentDepotImport). The configuration name is used as the INI section name and when naming the logging folder.

Secondly, when importing files, the audio is written to a temporary AudioVAULT file; that gets renamed to the final name at the end of the import. Uniqueld= is used in the algorithm for generating the temporary vault filename. Uniqueld= defaults to 1. If multiple instances are writing to a given vault, then Uniqueld must be set so that each instance's temporary vault filename is different/unique.

Use a single letter or number since vault filenames are limited in length.

- **ProducerAppIdMask=** Files that are received by Content Depot have a signature that is used to inform potential import utilities of the .wav file's origin. Various import utilities around the country can then be aware of the intention of the various .wav chunks and their fields. This special signature is Content Depot, and is what ProducerAppIdMask= defaults to. Files that lack this signature will not be imported. ProducerAppIdMask= uses VBA's LIKE syntax.
- **Width=** The default width of the program upon startup. There are about 1400 units per inch so some experimentation is needed to get the desired size/position.
- **Height=** See Width=
- **Left=** See Width=
- **Top=** See Width=



4.0 Import Restrictions

In an attempt to not import unwanted files, to not inadvertently overwrite up-coming soon-to-air content, etc, ContentDepotImport follows these restrictions when considering each file's import need:

- o The Content Depot file must be .wav
- o The Content Depot file must be a RIFF file with WAVE, fmt, cart and data chunks
- o The audio format (per the fmt chunk) must be supported by your AudioVAULT
- o The cart chunk's version and start date appear to be set
- o The cart chunk's CutID matches a rule.CutIdMask= INI setting
- o The cart chunk's ProducerAppID matches the ProducerAppIdMask= INI setting
- o The Content Depot file's cart chunk's kill date is not expired (see CheckReceivedFileKillDate=)
- o The Content Depot file's time stamp is newer than the corresponding AudioVAULT file (TLM Body)
- o The Content Depot's file's start/kill date is "better than" the AudioVAULT's start/kill date (see CheckVaultFileKillDate=)
- o The rule.CutIdMask= settings at the top of the INI section have higher priority over those at the bottom of the section. This only comes into play when the .wav's CutID is (accidentally or intentionally, explicitly or implicitly) included in multiple rule.CutIdMask= settings.

If the Content Depot has .wav files that ContentDepotImport is not importing the .wav files don't abide by one of these restrictions. Set TraceLevel=2 and inspect the log file to determine why the .wav file(s) don't import.

5.0 Timestamps & Manually Editing Cuts

Ideally, no human intervention would be needed to import and adjust Content Depot-received cuts. In practice, however, this may not always be possible. For example, if the Producer leaves lots of silence on the end of the cut then the cut will have to be manually trimmed. Or if a Producer is inconsistent with the audio level of a show from week to week, then the levels may have to be manually adjusted. ContentDepotImport examines the time stamp of the body (audio) of the AudioVAULT cut (in the case of an overwrite), and if the Content Depot file is newer then the file is considered for importation. As a result, it is safe to edit the meta data of the AudioVAULT cut, including gain, trim points, etc. But you should not modify the audio itself (Discard Scrap, re-record). Technically, the actual restriction is that you should not modify the AudioVAULT cut's audio after the next-to-air show arrives on the Content Depot.

Because the AudioVAULT cut is time stamped off of the Content Depot file, when there are multiple instances of a show on the Content Depot and that show is added to the ContentDepotImport configuration, multiple imports may take place initially. Ignoring kill dates for this discussion, we will assume that a show exists twice on the Content Depot, once with last week's date and again with yesterday's date. And let's say that ContentDepotImport, while scanning the Content Depot folder, first encounters last week's show. Since it is not yet on the AudioVAULT (it's a new show that we are adding), it is imported. Then scanning continues with the remaining files on the Content Depot, at which point the second instance is encountered. Since it is newer than that on the AudioVAULT, it too is imported. But from then on, because there is no longer a back log of files to import, there will be one import per instance that arrives on the Content Depot.



6.0 Examples

6.1 Example 1

```
[ContentDepotImport]
Vault=SVR
ContentDepotPath=192.168.1.101
Category=NPR
Default.CutIdMask=*
```

6.2 Real-World Example 1

```
[ContentDepotImport]
Vault=ENG
ContentDepotPath=\\192.168.1.101\xdcache\CDLongname
Category=BET
NfsClass=LOCAL
FilePrefix=BET
DescriptionMacro={FileName}
```

```
MinnOrch.CutIdMask=60326,60327
MinnOrch.FilePrefix=MNO
MinnOrch.Gain=-3
```

```
TTBOOK.CutIdMask=60963,60964,62654-62666
TTBOOK.FilePrefix=TBK
TTBOOK.Gain=-4
```

```
SoundandSpirit.CutIdMask=60711,60712,63736
SoundandSpirit.FilePrefix=SAS
SoundandSpirit.Gain=-3
```

```
MountainStage.CutIdMask=60085,60086,61554,61555,61556,61557,61558
MountainStage.FilePrefix=MST
MountainStage.Gain=-3
```

```
Symphonycast.CutIdMask=60114
Symphonycast.FilePrefix=SYM
```

```
Commonwealth.CutIdMask=60205
Commonwealth.FilePrefix=CCL
Commonwealth.GAIN=-4
```

```
FromtheTop.CutIdMask=60256,60257,64388
FromtheTop.FilePrefix=FTT
FromtheTop.Gain=-1
```

```
StPaulSun.CutIdMask=60106,60107
StPaulSun.FilePrefix=SPS
```



StPaulSun.Gain=1

Harmonia.CutIdMask=60141,60142,61145,61146,61147
Harmonia.FilePrefix=HAR

Women.CutIdMask=60113
Women.FilePrefix=WIM
Women.Gain=-2

SaysYou.CutIdMask=61123
SaysYou.FilePrefix=SAY
SaysYou.Gain=-5

Swingin.CutIdMask=60116-60118
Swingin.FilePrefix=SDL
Swingin.Gain=-8

PianoJazz.CutIdMask=60031
PianoJazz.FilePrefix=PIA
PianoJazz.Gain=-4

Tech.CutIdMask=60074
Tech.FilePrefix=TEK
Tech.Gain=-4

Broadway.CutIdMask=60183
Broadway.FilePrefix=BWY
Broadway.Gain=-1

Alternative.CutIdMask=60781
Alternative.FilePrefix=ALT

Echoes.CutIdMask=63676-63680,63683-63687
Echoes.FilePrefix=ECH
Echoes.Gain=-3

ClassicalGuitar.CutIdMask=60129
ClassicalGuitar.FilePrefix=CGA
ClassicalGuitar.Gain=-4

Schickele.CutIdMask=60614,60615,61585
Schickele.FilePrefix=SCH

NewLetters.CutIdMask=61334
NewLetters.FilePrefix=NLA
NewLetters.Gain=4

Thistle.CutIdMask=60093-60095



Thistle.FilePrefix=TNS
Thistle.Gain=-3

Shorts.CutIdMask=60007
Shorts.FilePrefix=SEL

LUSA.CutIdMask=60239,60240,64434-64437
LUSA.FilePrefix=LAT
LUSA.GAIN=-3

LOE.CutIdMask=61206-61209,62157-62159
LOE.FilePrefix=LOE
LOE.GAIN=-5

Default.CutIdMask=*

6.3 Real-World Example 2

[ContentDepotImport]
Vault=AVA
ContentDepotPath=\\10.0.10.52\xdcache\CDLongname
Category=CDP
NfsClass=ALL

PriPromos.CutIdMask=60060
PriPromos.Gain=-5
PriPromos.DescriptionMacro=PRI Promos - Fed {Date}
PriPromos.KillDateMacro={Date}+7

Conversations.CutIdMask=64360
Conversations.Gain=-6
Conversations.DescriptionMacro=Conversations from the World Cafe - Fed {Date}
Conversations.KillDateMacro={Date}+7

WorldCafe.CutIdMask=60068,60100,60131,60201,60420
WorldCafe.Gain=-3
WorldCafe.DescriptionMacro=World Cafe - Fed {Date}
WorldCafe.KillDateMacro={Date}+1

SoundOpinions.CutIdMask=60737-60741
SoundOpinions.Gain=-7
SoundOpinions.DescriptionMacro=Sound Opinions Seg #{Segment} - Fed {Date}
SoundOpinions.KillDateMacro={Date}+3

ETown.CutIdMask=60491-60492,61969-61971
ETown.Gain=-8
ETown.DescriptionMacro=E Town Seg #{Segment} - Fed {Date}
ETown.KillDateMacro={Date}+5



MountainStage.CutIdMask=60085-60086,61554-61558
MountainStage.Gain=-9
MountainStage.DescriptionMacro=Mountain Stage Seg #{Segment} - Fed {Date}
MountainStage.KillDateMacro={Date}+7

AmericanRoutes.CutIdMask=60618-60619,61491-61499
AmericanRoutes.Gain=-10
AmericanRoutes.DescriptionMacro=American Routes Seg #{Segment} - Fed {Date}
AmericanRoutes.KillDateMacro={Date}+5

BealeStreet.CutIdMask=60376-60379
BealeStreet.Gain=-11
BealeStreet.DescriptionMacro=Beale Street Caravan Seg #{Segment} - Fed {Date}
BealeStreet.KillDateMacro={Date}+5

Thistle.CutIdMask=60093-60095
Thistle.Gain=-12
Thistle.DescriptionMacro=Thistle and Shamrock Seg #{Segment} - Fed {Date}
Thistle.KillDateMacro={Date}+3

EchoesMon.CutIdMask=63676-63688
EchoesMon.Gain=-4
EchoesMon.DescriptionMacro=Echoes For Tuesday Seg #{Segment} - Fed {Date}
EchoesMon.KillDateMacro={Date}+1

EchoesTue.CutIdMask=63748-63760
EchoesTue.Gain=-4
EchoesTue.DescriptionMacro=Echoes For Wednesday Seg #{Segment} - Fed {Date}
EchoesTue.KillDateMacro={Date}+1

EchoesWed.CutIdMask=63820-63832
EchoesWed.Gain=-4
EchoesWed.DescriptionMacro=Echoes For Thursday Seg #{Segment} - Fed {Date}
EchoesWed.KillDateMacro={Date}+1

EchoesThu.CutIdMask=63848-63860
EchoesThu.Gain=-4
EchoesThu.DescriptionMacro=Echoes For Friday Seg #{Segment} - Fed {Date}
EchoesThu.KillDateMacro={Date}+1

EchoesFri.CutIdMask=63868-63880
EchoesFri.Gain=-4
EchoesFri.DescriptionMacro=Echoes For Saturday/Monday Seg #{Segment} - Fed {Date}
EchoesFri.KillDateMacro={Date}+1

Default.CutIdMask=*
Default.Gain=-8

