Manual - FeedGui

Manual - FeedGui

Overview :: What to do with the FeedGui?

The FeedGui is a graphical user interface to create, edit and validate openSDX feeds (structured data in the openSDX metadata format). The created feeds can be saved as xml file and be transferred to the receiver using the integrated feed beamer.

Prerequisites

1. You need java installed (Oracle-version - at least 1.6)
2. You have "JavaWebStart" appropriately enabled http://www.java.com/en/download/faq/java_webstart.xml
3. You have appropriately enabled JavaWebStart in your Browser and/or are happy to launch it by commandline/double-click the downloaded .jnlp-file
4. To create signatures of your feed data or to transfer your feed data to an openSDX fileserver you have to handle the following prerequisites:
    a. You have have already created a openSDX-keystore with at least one private-key using openSDX-SecGUI (or such) Manual - SecGui
    b. You have published your openSDX-(Public)Key to a well-known keyserver (such as keyserver.fnppl.org which comes with the standard-configuration of openSDX-SecGUI)
    c. You have told someone in possession of a openSDX-FileServer (if it is not yourself) to create an account for you (for that openSDX-Key and a username of your choice)
5. Please make sure, that you accept all permissions, the software is requesting (the software is trust-signed for that reason)
6. Please make sure, that no proxy/firewall is in-between/blocking the openSDX-FeedGUI

How to get the FeedGui started?

If you have Java-WebStart enabled in your browser, a click on this URL: https://simfy.finetunes.net/openSDX/opensdx_feedgui.jnlp should already launch the GUI.
Alternatively, you could download the .jnlp-file (just right-click on the previous link and select "save-as" or such) to your (e.g.) Desktop and

then either double-click it or launch it using the commandline (e.g.: "javaws opensdx_feedgui.jnlp").
You could also launch it using the commandline without prior downloading it (Java-WebStart will do it for you): "javaws https://simfy.finetunes.net/openSDX/opensdx_feedgui.jnlp"

The openSDX feed format

The most recent version of the openSDX-metadata format (proposal) for sending data is available here (and also on github): https://simfy.finetunes.net/example_feed.xml.
We will just try to summarize the main-sections of that "feed".
Basically, a feed is the "envelope" of sending information on one or more tracks or bundles. We think, it is advisable to send one bundle per feed (for error-isolation and faster workability), but the metadata-format allows more.
A detailed explanation of how the fields in the XML-feed shall be interpreted/used, can be found in a separate document called "openSDX_00-00-00-01.pdf" which can also be found on https://simfy.finetunes.net/openSDX_00-00-00-01.pdf.
This PDF is mainly constructed out of the existing .xsd and its annotations. Please note, that the example-feed specifically does NOT link to this .xsd, but has has commented-out line for simply enabling that.

feedinfo

On feedinfo-level there are the global information needed or at least valuable for ingesting / identifying the content sent.
It is defined, when the feed was created, when it shall be come effective, who created the feed and who is the receiver of the feed. Also the sender (which can deviate from the creator) is to be stated. The licensor and licensee are also to be stated (which in turn can also deviate from the creator and/or the sender).
There can be "actions" defined on the receiving-party's side which should be "done" when initially receiving this feed, or starting to process the feed for ingestion or finishing the feeds processing. Additionally when everything could be interpreted correctly (in the sense of the receiving party), a "full-success-action" could be issued; likewise if "some error" occurred while processing the feed, an "onerror-action" could be issued.
Those actions are initially defined to be email-notifications or http-calls; we also included some action to have a "registered letter" and/or "fax" to be sent; whether this is accepted/handled by the receiving party is to be dealt with contractually (we included a field for stating how much the sending party will cover the fee max.).

bundle

On bundle level, there are information on how to handle a collection of "items". This is mainly an album/ep/single. A bundle is identified by one unique identifier, but more unique identifiers could and should be transmitted as well (see below "ids"). Most notably on the bundle-level is the "bundle name" which is basically the conjunction of the "name"- and the "version"-field. Also to have this easy at hand, there should be the desired "display_artist"-string be present on this level. Of course, the receiver of the feed can still calculate the "correct" display_artist by evaluating the contributors (see below) for this.

ids

This element holds one or more identifiers for this bundle. The "ids"-element is also present on "item"-level (see below) and could also be present on "contributor"-level (in terms of publishing-authorities-id-system). A good unique identifier on bundle-level is the EAN/UPC-code; a good unique identifier on item-level is the ISRC resp. ISWC-code. But you could create your own identifier as well – a good idea could be to create a UUIDhttp://de.wikipedia.org/wiki/Universally_Unique_Identifier and to store that one in your database-system as well. In sales-reports all unique ids supplied by you, should be presented back to you as well; but the reporting party can still decide to "just" report on "the most significant" ids.

contributors

This element is rather abstract. Everyone and every company involved in this bundle (and its items placed here), must be denoted here. On item-level these contributors are therefore just referenced and if one (or more) contributor(s) is only "valid" on item-level (e.g. a conductor only conducting one item(track) on this release), you could/should mark that contributor (e.g. conductor) to be "item-level-only". Contributors can be of any type: Artist, Composer, Texter, Lyricist, Label, Publishing House, Publishing Authority etc... The .xsd states a discrete list for these, but if you have more information (=contributor-types) than we currently have in mind, please let us know - we totally do not want to lose any information aiming for total customer happiness and perfect administratabilty. For all - at least where available - you should be providing related information-sources (such as Website, Blog, youtube-Channel etc.).

information

On this level, there should also be some "narrative-information" on the bundle be present. This information can be put in different languages and should be (at least) the two "main" informations available: "Promo-Text" and "Release-Text".

related

This element holds all available information on related products/items/bundles for this bundle/its items. If there is your("a") youtube-link available for that, put it here, if you want to carry information on the orderability by the physical distributor of this product, put it here also.

license-basis

This element also contains the basic information of general (not specific to the targeted shop and not license-relevant as such!) "physical_release_datetime" and "digital_release_datetime". Btw. all "datetime"-fields are to be expressed in the GMT-denotation - e.g.: "1960-02-13 00:00:00 GMT+00:00" or "1960-02-13 00:00:00 GMT-07:00". This element then contains the license-basis-information such as time-frames, territorial allowances/restrictions and price-codes/price-levels.

license-specifics

This element also contains the "license-specific"-rule(s) which can be used to denote some deviation under some "specific circumstances" of that basic-license-information. Those rules are denoted in a "programmatic" style so that implementing should be straight-forward without losing human-readability. The license-specific-rules are to be interpreted in the order of their denotation in the xml (also indicated by the attribute "num" of each rule). The first rule matching, does NOT end the interpretation of those rules - if one rule "proclaims" the same information, the last rule proclaiming that is "winning". Rules can also state to "break" executing when they are matched. We do not have included "jump to rule num X" yet, but that is likely to come.

tags

Tags are either "flags" or categories (and sub-categories) imposed on this bundle/item. "tags" can/should also be present on item-level. Categories also contain "genres" as a quite useful categorization. The .xsd states the current genre-specifications available - if you need more, let us know! Flags are currently defined as "streaming_allowed" or "bundle_only" and also "main_language" and "origin_country".

reporting

This tag is not a must, but is of course very useful for implementing a real "live-reporting" via e.g. http-calls or mails-to-be-sent. You could also put some information here which should then be included in the "postponed" reporting. Whether this is interpreted/accepted by the target-shop is to be handled by contract. Of course, we think, it is advisable - especially in conjunction with signatures so that the licensor can have a valid⚠ and reliable source for realtime-reportings and there even could be no more need for a "postponed-reporting" then.

items

This element contains all items bundled in this bundle. Most probably those are tracks (either video, audio, flash, whatever).

item

An item's definition is quite straightforward from the bundle-level. In concern of the license-basis /-specifics, you most probably will denote "as_on_bundle_level", but you could define deviations from that here as well. Whether those are to be interpreted by the target-shop is to be dealt with contractually. On this level, there should also be some kind of audio-fingerprint-information available. This is desired, because we try to emphasize on a correct, ideal chain without any gap(s) of license-flow. If the target-shop (or its ingestion-provider) has to convert the audio you delivered to e.g. flac/mp3/different_bitrate, then the checksums will not match that converted file anymore - the target-shop (and/or its ingestion-provider) can also proof the license-chain when keeping the original-file in storage "forever" - this is not desired in the most cases. BUT an audio-fingerprint of an encoded-file is identifiable to the fingerprint of the original file. So, having the audio-fingerprint of the original file in the license-chain (namingly "the signed XML-document"), the license-chain stays intact on that.

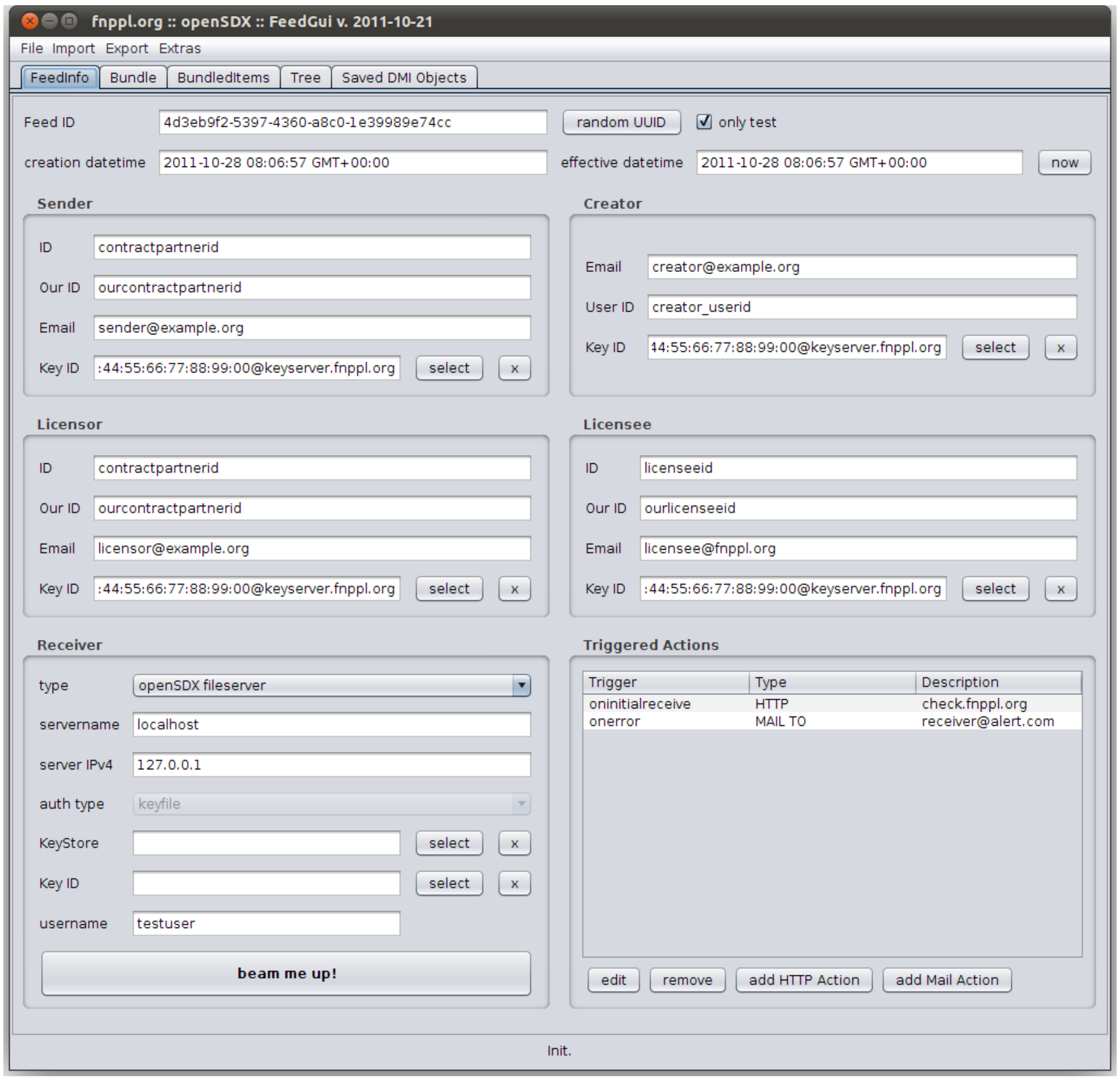
files

This element contains each file-element for each file accompanying this bundle/item. A file can be of type audio/video/flv/whatever. A file does not necessarily have to be delivered with the feed meaning, that a file could also be denoted to be "fetched on the fly" - e.g.: You send the delivery-feed denoting that files are available on YOUR ftp for download (with login/pass/within timeframe). This could also be used to make it easy to initially "ingest" all metadata to a target-shop to be able to be "live" there in a rather short time and then to have the target-shop "pull" the content on-demand or asynchronously. Whether this "on-demand-mechanism" is valid (and under which circumstances), is to be dealt with contractually (as well).

The User Interface

Main window

After starting the FeedGui you will see the following window:


fnppl.org :: openSDX :: FeedGui v. 2011-10-21
File Import Export Extras
FeedInfo Bundle BundledItems Tree Saved DMI Objects
random UUID
now
select
x
beam me up!
edit
remove
add HTTP Action
add Mail Action
Init.

Feed ID: 4d3eb9f2-5397-4360-a8c0-1e39989e74cc
☑ only test
creation datetime: 2011-10-28 08:06:57 GMT+00:00
effective datetime: 2011-10-28 08:06:57 GMT+00:00

Sender

ID	contractpartnerid
Our ID	ourcontractpartnerid
Email	sender@example.org
Key ID	:44:55:66:77:88:99:00@keyserver.fnppl.org

Creator

Email	creator@example.org
User ID	creator_userid
Key ID	44:55:66:77:88:99:00@keyserver.fnppl.org

Licensor

ID	contractpartnerid
Our ID	ourcontractpartnerid
Email	licensor@example.org
Key ID	:44:55:66:77:88:99:00@keyserver.fnppl.org

Licensee

ID	licenseeid
Our ID	ourlicenseeid
Email	licensee@fnppl.org
Key ID	:44:55:66:77:88:99:00@keyserver.fnppl.org

Receiver

type	openSDX fileserver
servername	localhost
server IPv4	127.0.0.1
auth type	keyfile
KeyStore	
Key ID	
username	testuser

Triggered Actions

Trigger	Type	Description
oninitialreceive	HTTP	check.fnppl.org
onerror	MAIL TO	receiver@alert.com

On top of the screen you will find the menu bar with the menu items *File*, *Import*, *Export* and *Extras*.

Below the menu bar in the centre of the screen you can choose from the following tabs:

- FeedInfo: input mask for the general feed data part of the feed
- Bundle: input mask for the data concerning the whole bundle of the feed (in the FeedGui you can only edit one bundle per feed)
- BundledItems: input mask for the data concerning all items in the bundle
- Tree: full view of feeds XML tree
- Saved DMI Objects: here you can see already saved information and add them to you feed, e.g. you can choose from a list of known previously saved receivers or your own previously saved sender, creator, actions, ... data. This is especially useful for not having to enter the same information in different feeds again and again.

In the following each tab will be described in detail.

FeedInfo

At the top of the *FeedInfo* tab you can set the id of the feed in *FeedID*. It is good practice to generate a UUID (by pressing the *random UUID* button) for every feed you create but certainly you can enter your own id.

editing text fields
When you type in a text field the background colour of the text will switch to yellow. The edited data will be applied by pressing the enter key and the background colour will turn white again. If you press the escape key (Esc) the previous data in the text field will be restored and the background colour will turn back to white.
Some text fields (e.g. fields with date/time information) have special format constraints. If the background colour turns red your data has the wrong format. In case of a data/time field you have to use the format:
yyyy-mm-dd HH:MM:SS GMT+hh:00, with

- yyyy: year (4 numerals)
- mm: month of the year (2 numerals)
- dd: day of the month (2 numerals)
- HH: hour of the day (2 numerals)
- MM: minute (2 numerals)
- SS: second (2 numerals)
- +hh or -hh: timezone offeset in hours (2 numerals)

On the right side of the UUID button you can select whether you feed should be a test or real data by checking / un-checking the *only test* checkbox. Next you can enter the creation datetime of the feed and the datetime when it should come effective (please follow the correct date/time format).

The *Sender* part should give information about the sender of the feed (typically you / your company), where the *Key ID* should be related to the key that will be used by the sender in further correspondence. The *ID* field should hold the id on the receiver side, the *Our ID* field should hold the ID of the sender side. Beside the sender you should give information about the *Creator*, the *Licensor*, the *Licensee* and the *Receiver* . The receiver information part is somehow special, because there is the possibility to send (beam up) the whole feed information including given media files to the receiver when you know his/her/its *openSDX fileserver* or *ftp* server account data. In case of an openSDX fileserver you have to enter the servername (e.g. simfy.finetunes.net) and the username and keyid you have previously agreed on with the operator of the server. The keyid can be selected out of all keys from a keystore file by first choosing the keystore that contains the desired key by pressing the select button next to the KeyStore text field. You can start the feed upload dialog by pressing the "beam me up!" button or selecting *File -> Upload Feed to Receiver ...* from the menu bar.

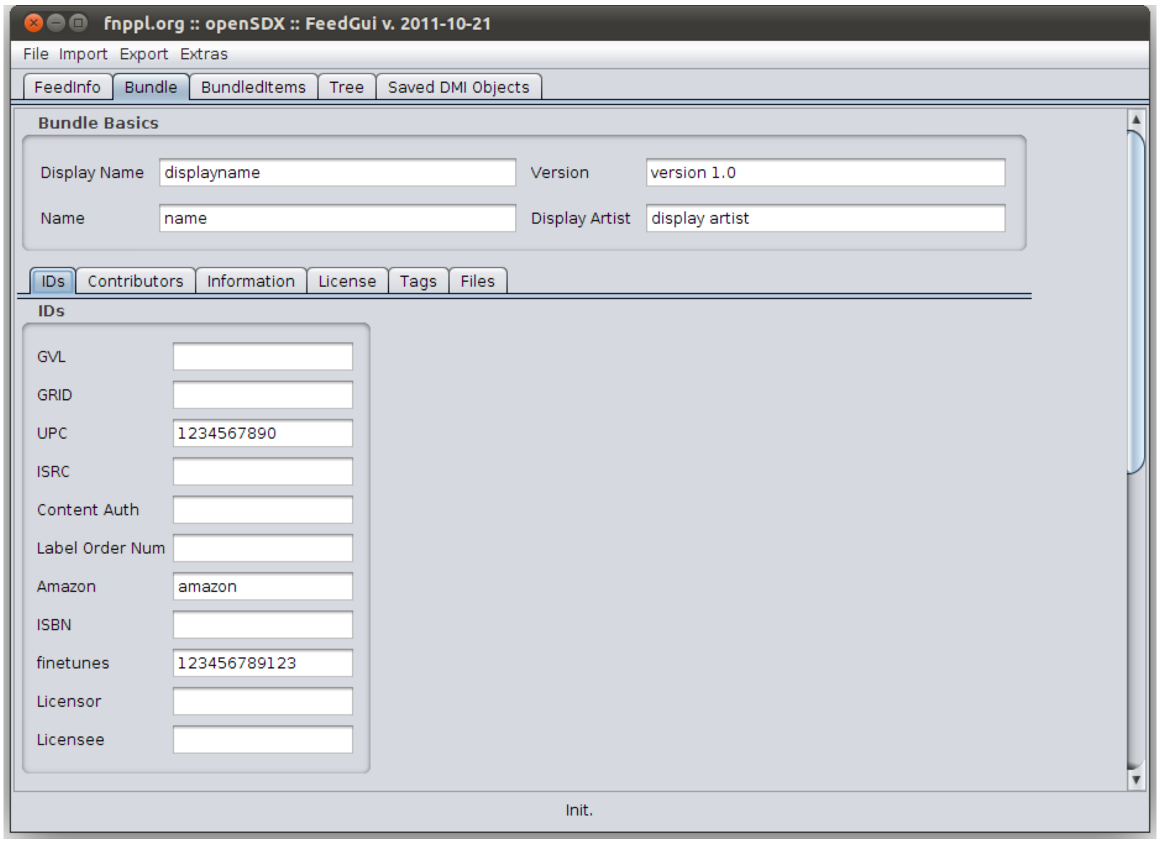
The last part of the *FeedInfo* tab is the *Triggerd Actions* section. Here you can add HTTP or email actions for the following events:

- on initial receive: triggered when the feed was received by the receiver
- on process start: triggered when the processing of the feed starts
- on process end: triggered when the processing of the feed ends
- on error: triggered when an error occurs during the processing of the feed
- on full success: triggered when the processing of the feed was successfully completed

A *HTTP action* will execute the given HTTP command and a mail action will send an email to the given receiver(s) with the given subject and text when the selected event is trigger.

Bundle

The *Bundle* tab shows all feed information on bundle level like bundle ids, a list of contributors, license information and so on.


fnppl.org :: openSDX :: FeedGui v. 2011-10-21

File Import Export Extras
FeedInfo Bundle BundledItems Tree Saved DMI Objects
Bundle Basics
IDs Contributors Information License Tags Files
IDs
Init.

Display Name	displayname	Version	version 1.0
Name	name	Display Artist	display artist

GVL	
GRID	
UPC	1234567890
ISRC	
Content Auth	
Label Order Num	
Amazon	amazon
ISBN	
finetunes	123456789123
Licensor	
Licensee	

At the top of the tab you can enter the basic bundle informations:

- display name: name that is displayed to consumers
- name: internal name
- version: bundle version
- display artist: name of the artist that is visible to consumers

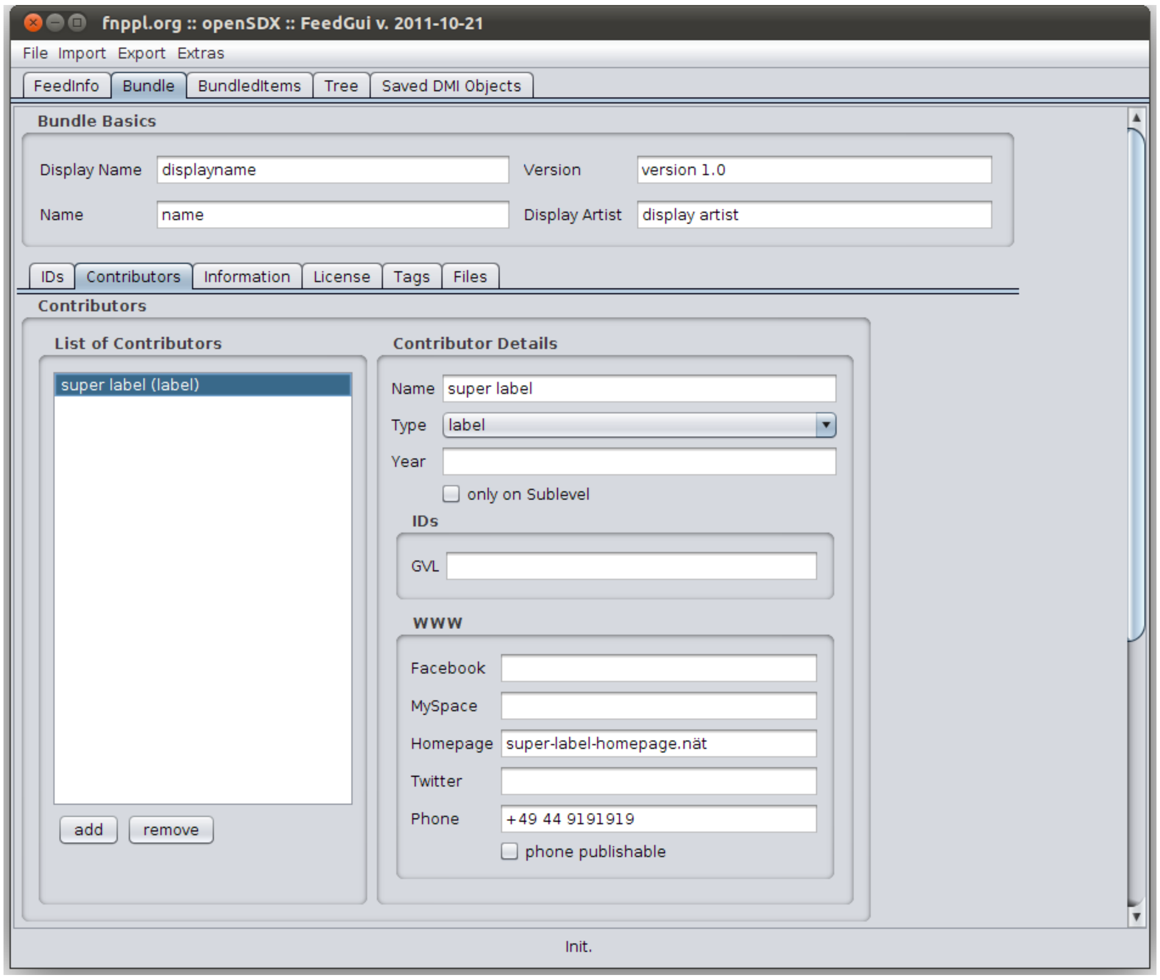
The other bundle data can be accessed in the following sub tabs:

Bundle :: IDs

The bundle IDs provide identifiers for the bundle. At least you should provide the UPC.

Bundle :: Contributors

In the *Contributors* sub tab you should provide the data of all peoples/companies/entities that contributed to the bundle in any way. The contributor can be of the following type:
label, performer, texter, editor, conductor, orchestra, display_artist, singer, composer, mixer, remixer, producer, featuring, with, DJ, versus, meets, presents, compilator, publisher, clearing house, copyright, production
A special case holds for the copyright and the production type. Here you should give the year value in the field below.
For every contributor further information about like ids (GVL) and internet appearance can be specified.



Bundle Basics

Display Name	displayname	Version	version 1.0
Name	name	Display Artist	display artist

Contributors

List of Contributors

super label (label)

Contributor Details

Name	super label
Type	label
Year	
	☐ only on Sublevel

IDs

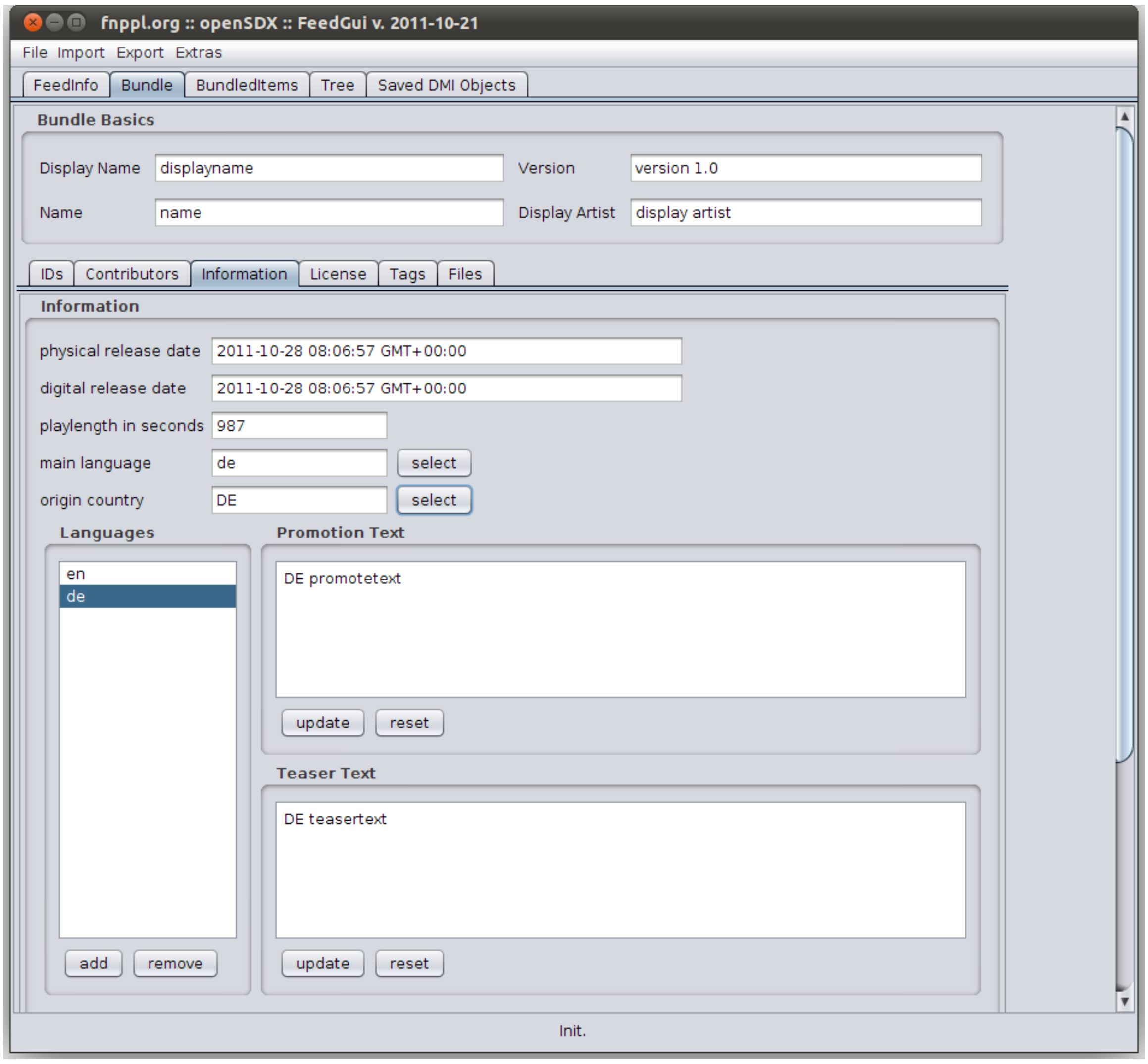
GVL	

WWW

Facebook	
MySpace	
Homepage	super-label-homepage.nät
Twitter	
Phone	+49 44 9191919
	☐ phone publishable

Bundle :: Information

In the bundle's *information* sub tab you can enter data about the physical and digital release date, the playlength of the complete bundle in seconds, the main language, the origin country and the promotion and teaser texts in different languages. To add a text for a new language please hit the *add* button below the Languages list first and select the language code. Then enter (or paste from clipboard with ctrl-v) the desired promotion and/or teaser text and don't forget to click on the update button after finishing the text inputs.


fnppl.org :: openSDX :: FeedGui v. 2011-10-21

File Import Export Extras

FeedInfo | Bundle | BundledItems | Tree | Saved DMI Objects

Bundle Basics

Display Name	displayname	Version	version 1.0
Name	name	Display Artist	display artist

IDs | Contributors | Information | License | Tags | Files

Information

physical release date	2011-10-28 08:06:57 GMT+00:00
digital release date	2011-10-28 08:06:57 GMT+00:00
playlength in seconds	987
main language	de
origin country	DE

Languages

en
de

Promotion Text

DE promotetext

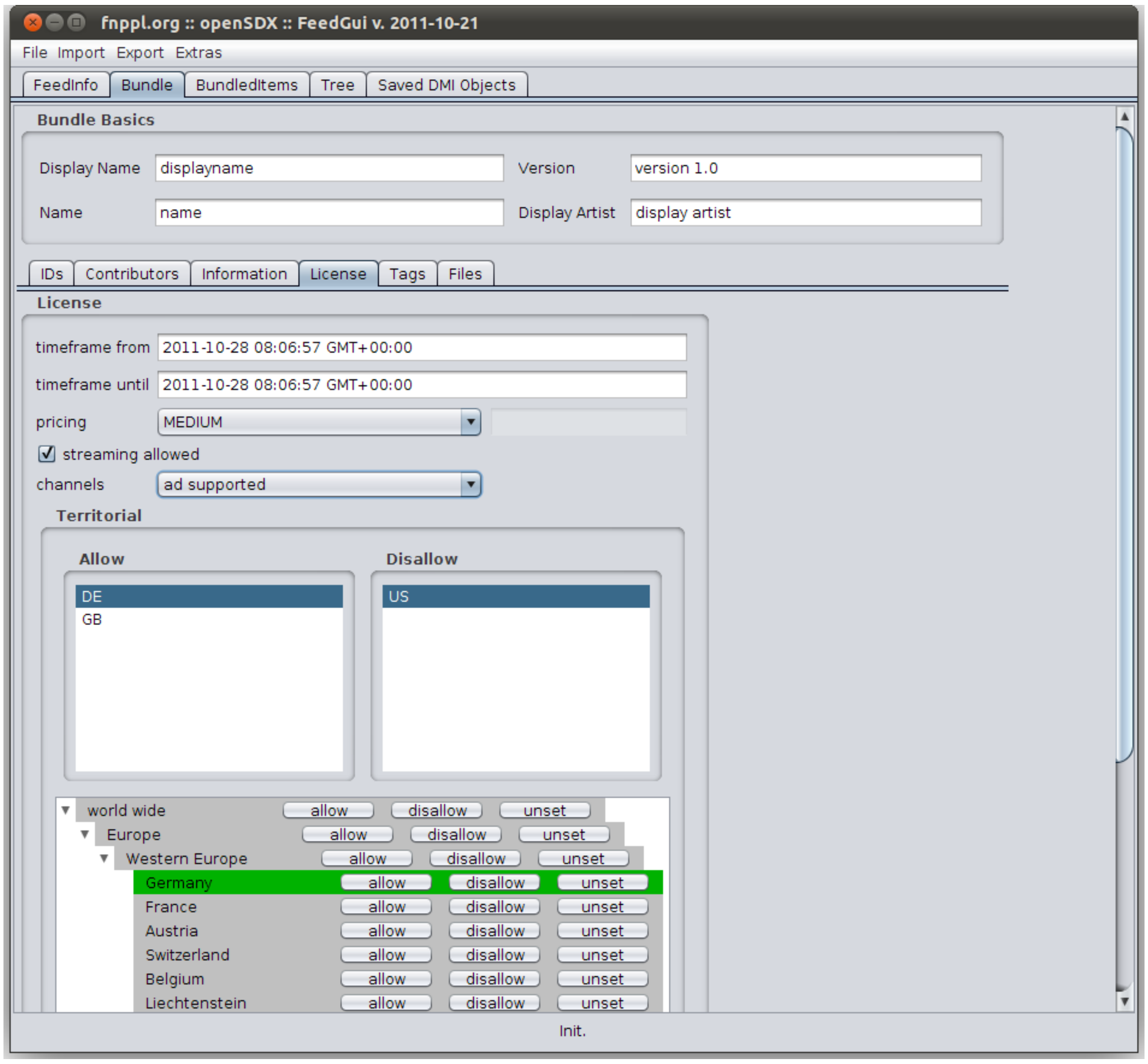
Teaser Text

DE teasertext

Init.

Bundle :: License

In the bundle's *License* sub tab you can specify the timeframe of the license in the from and until date/time fields. Moreover you can either set a pricing level (LOW, MEDIUM, HIGH, the details have to be specified in an external contract) or specify the price directly by selecting [other] and enter the price in the text field on the right side. You can allow or disallow streaming of the content by checking / unchecking the *streaming allowed* checkbox. If streaming is allowed you have to specify the channels by selecting *all*, *ad supported* or *premium* from the list below.
The territorial part shows the allowed and disallowed territories of content offers. The allowed territories are specified hierarchically by including and excluding whole regions or single countries. For example an allowance of WW and CA and dis-allowance of Americas and FJ would mean that the content can be offered all over the world except in America (North, Central, South and Caribbean) and on the Fiji Islands. Additionally Canada would be allowed (even if it is in America)



fnppl.org :: openSDX :: FeedGui v. 2011-10-21

File Import Export Extras

FeedInfo | Bundle | BundledItems | Tree | Saved DMI Objects

Bundle Basics

Display Name	displayname	Version	version 1.0
Name	name	Display Artist	display artist

IDs | Contributors | Information | License | Tags | Files

License

timeframe from	2011-10-28 08:06:57 GMT+00:00
timeframe until	2011-10-28 08:06:57 GMT+00:00
pricing	MEDIUM
✓ streaming allowed	
channels	ad supported

Territorial

Allow	Disallow
DE	US
GB	

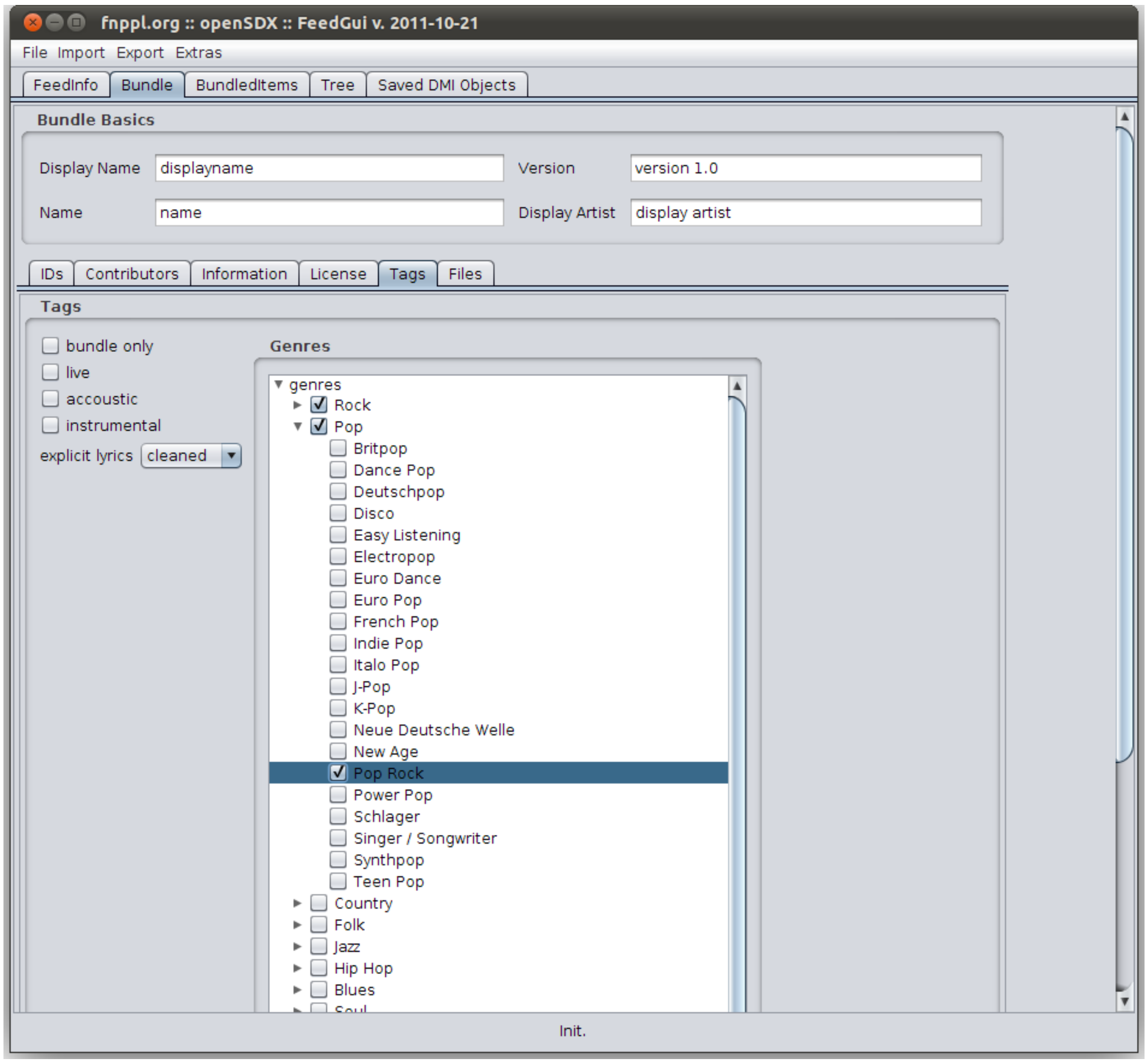
```
▼ world wide              allow   disallow   unset
  ▼ Europe                allow   disallow   unset
    ▼ Western Europe      allow   disallow   unset
        Germany           allow   disallow   unset
        France            allow   disallow   unset
        Austria           allow   disallow   unset
        Switzerland       allow   disallow   unset
        Belgium           allow   disallow   unset
        Liechtenstein     allow   disallow   unset
```

Init.

Bundle :: Tags

In the bundle's *Tags* sub tab you can set / unset set following tags:

- bundle only: set if the bundle's content should only be offered as a whole bundle and not as single items
- live: set if the bundle's content is a live recording
- acoustic: set if the bundle's content is acoustic media
- instrumental: set if the bundle's content are instrumental media
- explicit lyrics: select from: [not set], true, false or cleaned
- genres: specify the genres and sub-genres of your media


fnppl.org :: openSDX :: FeedGui v. 2011-10-21

File Import Export Extras

FeedInfo | Bundle | BundledItems | Tree | Saved DMI Objects

Bundle Basics

Display Name	displayname	Version	version 1.0
Name	name	Display Artist	display artist

IDs | Contributors | Information | License | Tags | Files

Tags

- [ ] bundle only
- [ ] live
- [ ] accoustic
- [ ] instrumental

explicit lyrics: cleaned

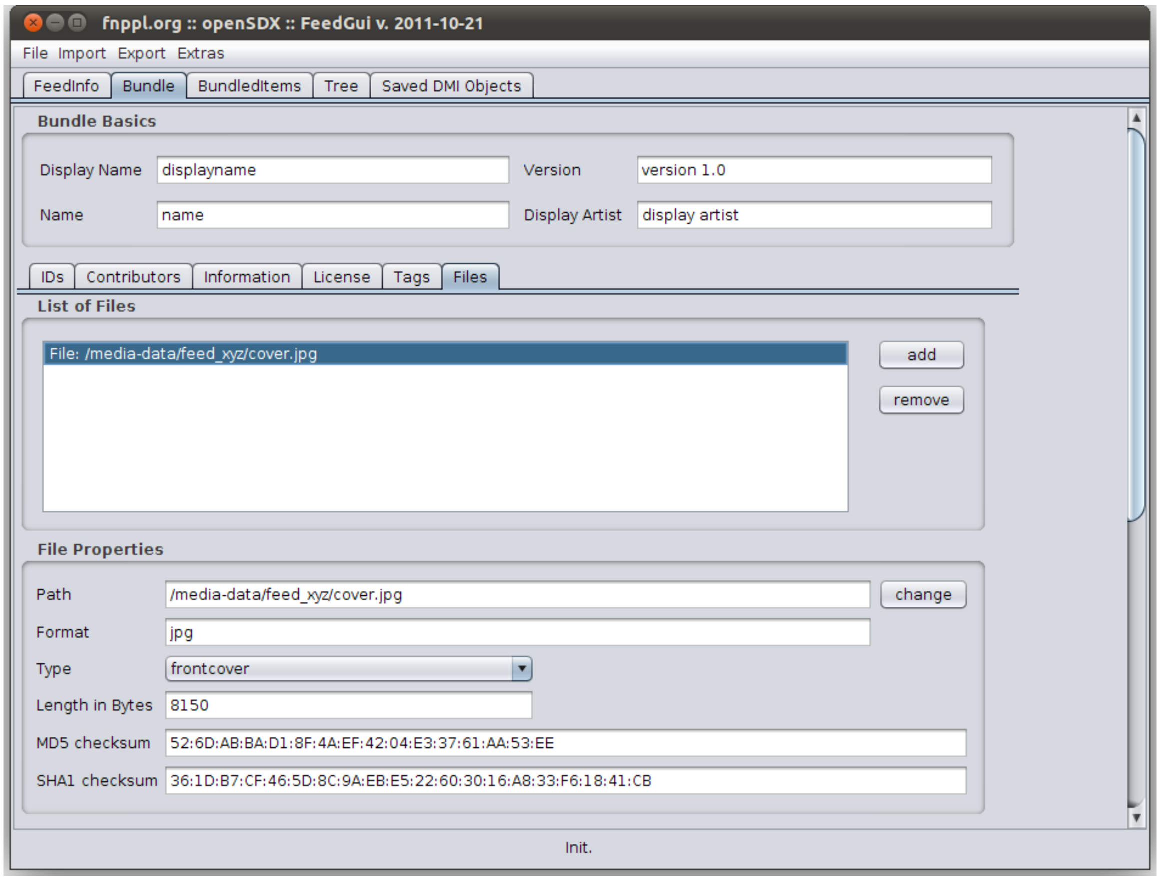
Genres

- genres
  - ✓ Rock
  - ✓ Pop
    - Britpop
    - Dance Pop
    - Deutschpop
    - Disco
    - Easy Listening
    - Electropop
    - Euro Dance
    - Euro Pop
    - French Pop
    - Indie Pop
    - Italo Pop
    - J-Pop
    - K-Pop
    - Neue Deutsche Welle
    - New Age
    - ✓ Pop Rock
    - Power Pop
    - Schlager
    - Singer / Songwriter
    - Synthpop
    - Teen Pop
  - Country
  - Folk
  - Jazz
  - Hip Hop
  - Blues

Init.

Bundle :: Files

In the bundle's *Files* sub tab you can add artwork file of the type: frontcover, backcover and / or booklet. Therefore you have to select the file by clicking the *add* button and then selecting the file from the file dialog. The type can be specified in the details view. The local path of the files will be transformed into the correct openSDX delivery format (filename of content := feedid_consecutive number_md5 checksum of the file) when the feed is uploaded by the *beam me up!* method in the *Receiver* section in the *FeedInfo* tab.



Bundle Basics

Display Name	displayname	Version	version 1.0
Name	name	Display Artist	display artist

List of Files

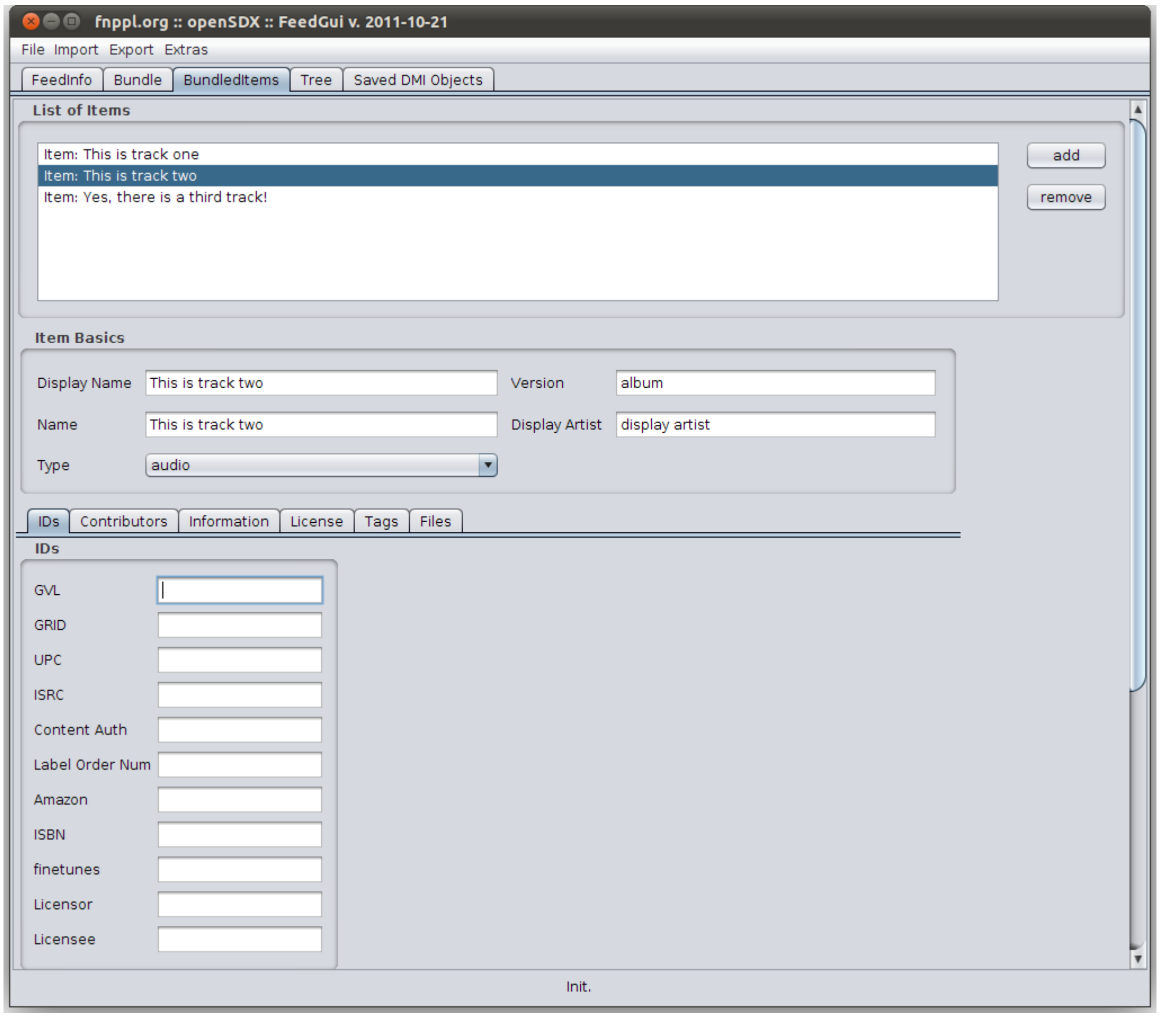
File: /media-data/feed_xyz/cover.jpg

File Properties

Path	/media-data/feed_xyz/cover.jpg
Format	jpg
Type	frontcover
Length in Bytes	8150
MD5 checksum	52:6D:AB:BA:D1:8F:4A:EF:42:04:E3:37:61:AA:53:EE
SHA1 checksum	36:1D:B7:CF:46:5D:8C:9A:EB:E5:22:60:30:16:A8:33:F6:18:41:CB

Bundled Items

The tab *BundledItems* is used to add your items (e.g. tracks / videos) to your bundle.


fnppl.org :: openSDX :: FeedGui v. 2011-10-21

File Import Export Extras

FeedInfo | Bundle | BundledItems | Tree | Saved DMI Objects

List of Items

- Item: This is track one
- Item: This is track two
- Item: Yes, there is a third track!

add | remove

Item Basics

Display Name	This is track two	Version	album
Name	This is track two	Display Artist	display artist
Type	audio		

IDs | Contributors | Information | License | Tags | Files

IDs

GVL	
GRID	
UPC	
ISRC	
Content Auth	
Label Order Num	
Amazon	
ISBN	
finetunes	
Licensor	
Licensee	

Init.

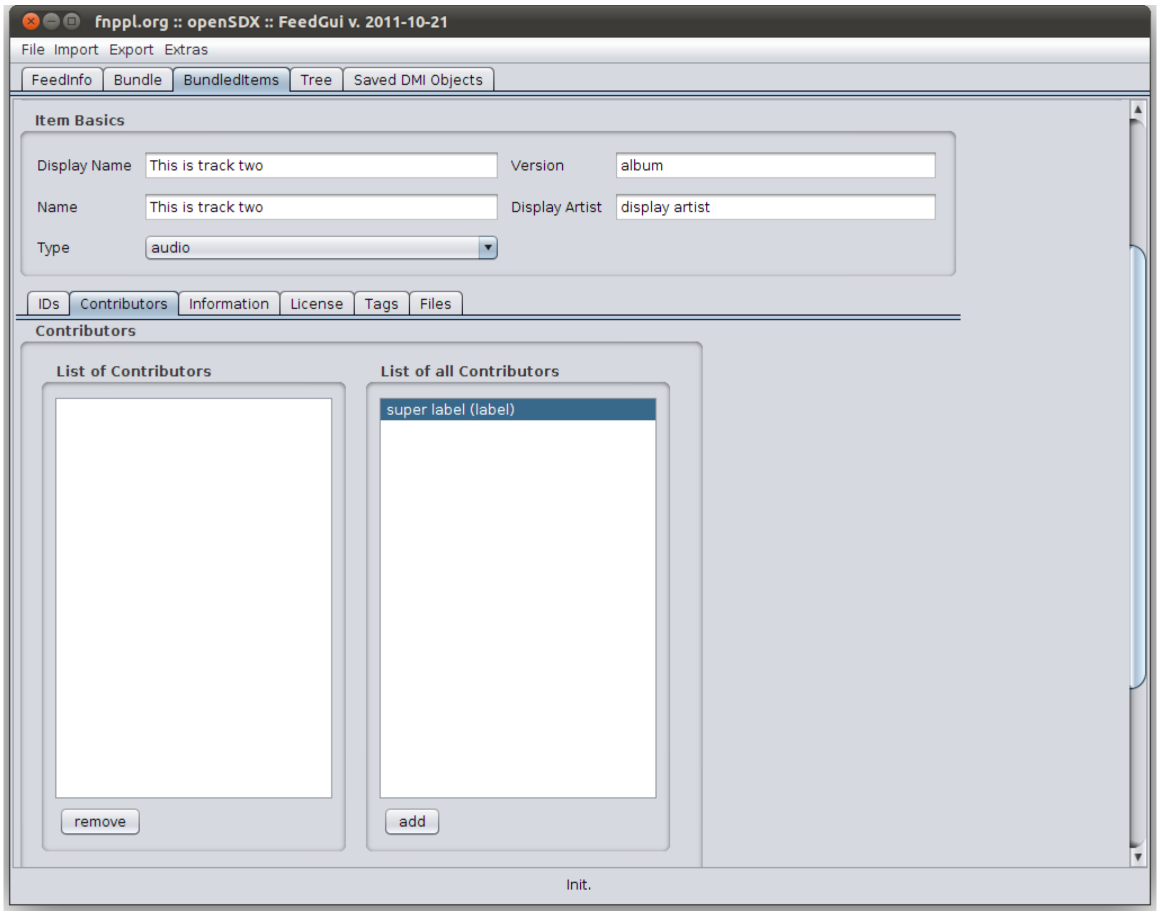
At the top of the tab you see the section *List of items*. You can add a new item to the list by pressing the *add* button on the right side and remove a selected item using the *remove* button. If you select an item in the list the details will be shown in the panels below. In the *Item Basics* section you can enter the *Display Name*, *Name*, *Type* (audio or video), the *Version* and the *Display Artist*. Below the *Item Basics* you see a number of sub tabs comparable to the sub tabs on bundle level. But as the context is a little bit different from the bundle level these sub tabs differ in details. They will be described in the following.

Item :: IDs

In the *IDs* sub tab you can enter your identifiers of the selected item.

Item :: Contributors

In the *Contributors* sub tab you can select all contributors that contribute to this item from the list of all contributors on bundle level. The selected (and added) contributors are listed on the left, all contributors on bundle level are listed on the right. You can remove a contributor by selecting the name in the left list and pressing the *remove* button below.



Item Basics

Display Name: This is track two
Version: album
Name: This is track two
Display Artist: display artist
Type: audio

Contributors

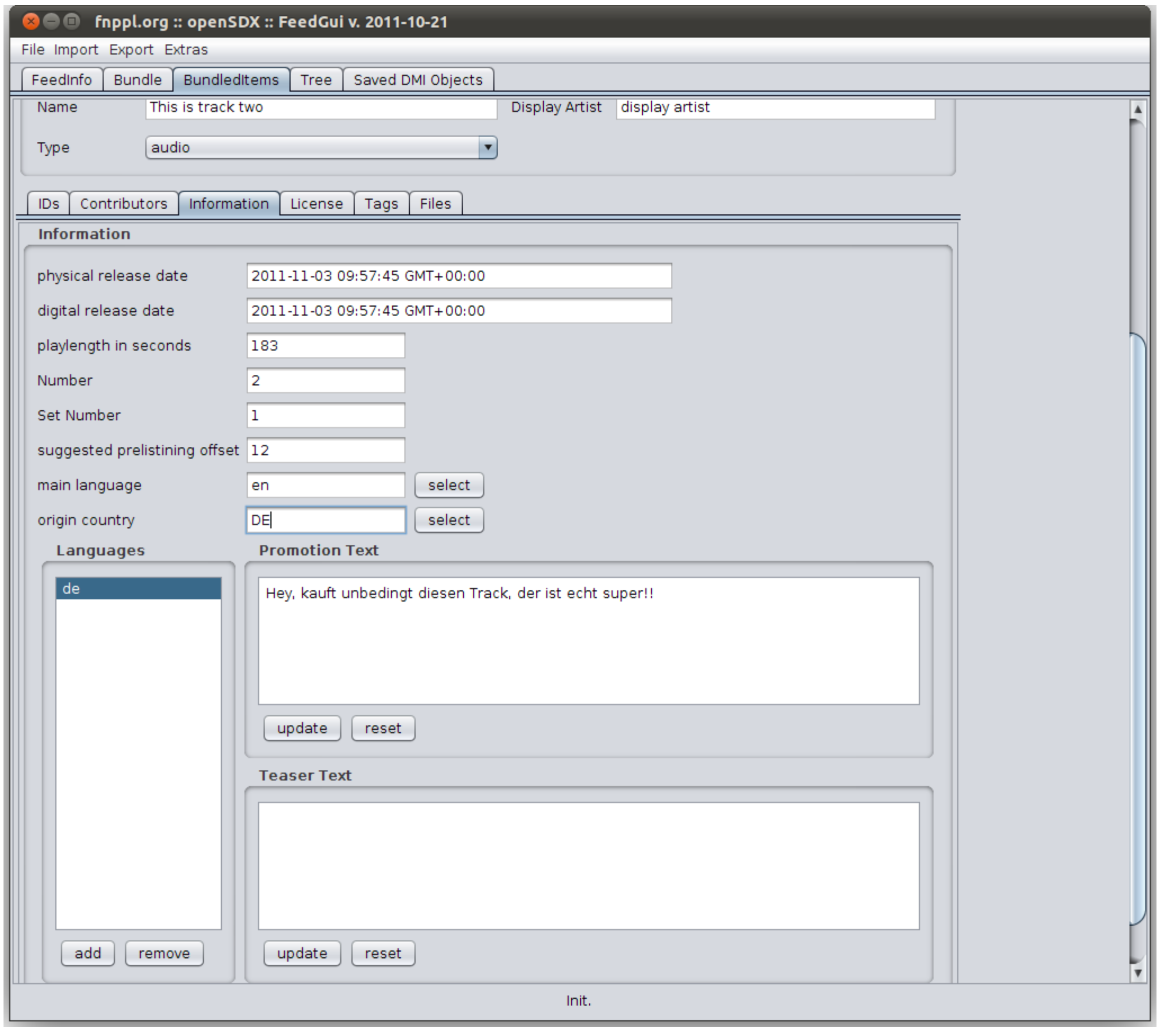
List of Contributors

List of all Contributors
super label (label)

Item :: Information

In the *Information* sub tab you can edit the following item related information:

- physical release date
- digital release date
- playlength in seconds
- number, e.g. the track number on the album
- set number, e.g. the number of the cd from a collection
- suggested prelistening offset (in seconds)
- main language
- origin country
- promotion and teaser texts for different languages (see: Bundle :: Information)


fnppl.org :: openSDX :: FeedGui v. 2011-10-21
File Import Export Extras
FeedInfo Bundle BundledItems Tree Saved DMI Objects
IDs Contributors Information License Tags Files
select
update reset
add remove
Init.

Name: This is track two
Display Artist: display artist
Type: audio

Information

physical release date: 2011-11-03 09:57:45 GMT+00:00
digital release date: 2011-11-03 09:57:45 GMT+00:00
playlength in seconds: 183
Number: 2
Set Number: 1
suggested prelistining offset: 12
main language: en
origin country: DE

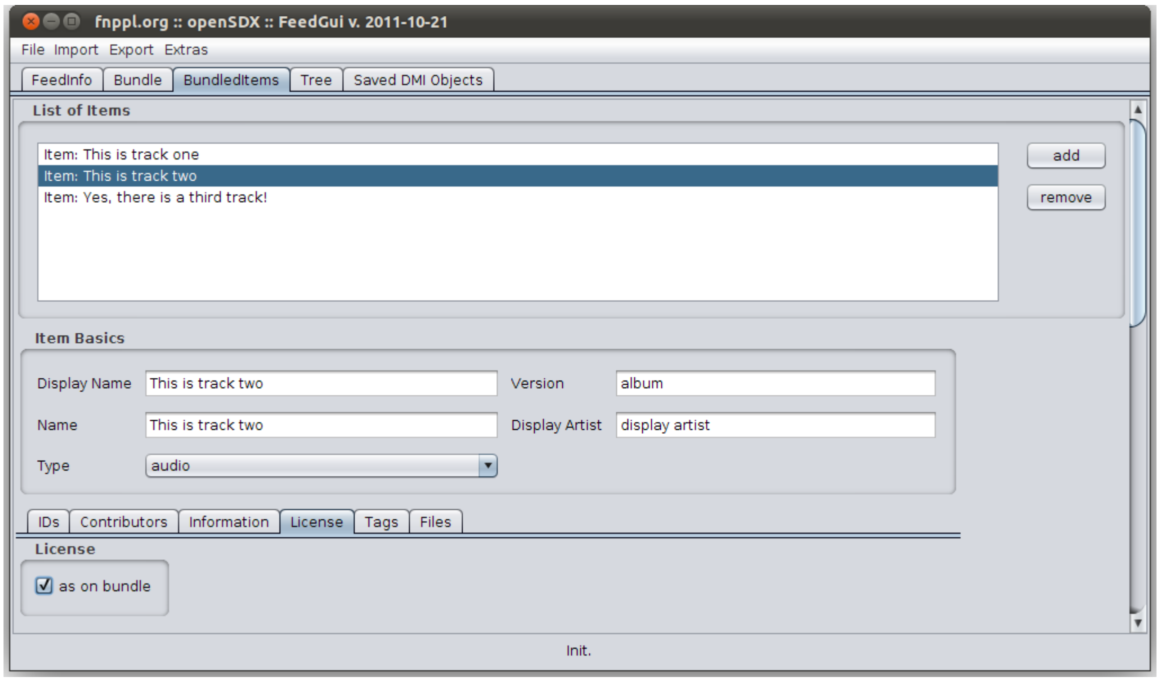
Languages: de

Promotion Text: Hey, kauft unbedingt diesen Track, der ist echt super!!

Teaser Text

Item :: License

In the *License* sub tab you either select to use the same licence as on bundle level by checking the *as on bundle* checkbox or you can define another license especially for this item (see: Bundle :: License)


fnppl.org :: openSDX :: FeedGui v. 2011-10-21
File Import Export Extras
FeedInfo Bundle BundledItems Tree Saved DMI Objects
add
remove
IDs Contributors Information License Tags Files
Init.

List of Items

Item: This is track one
Item: This is track two
Item: Yes, there is a third track!

Item Basics

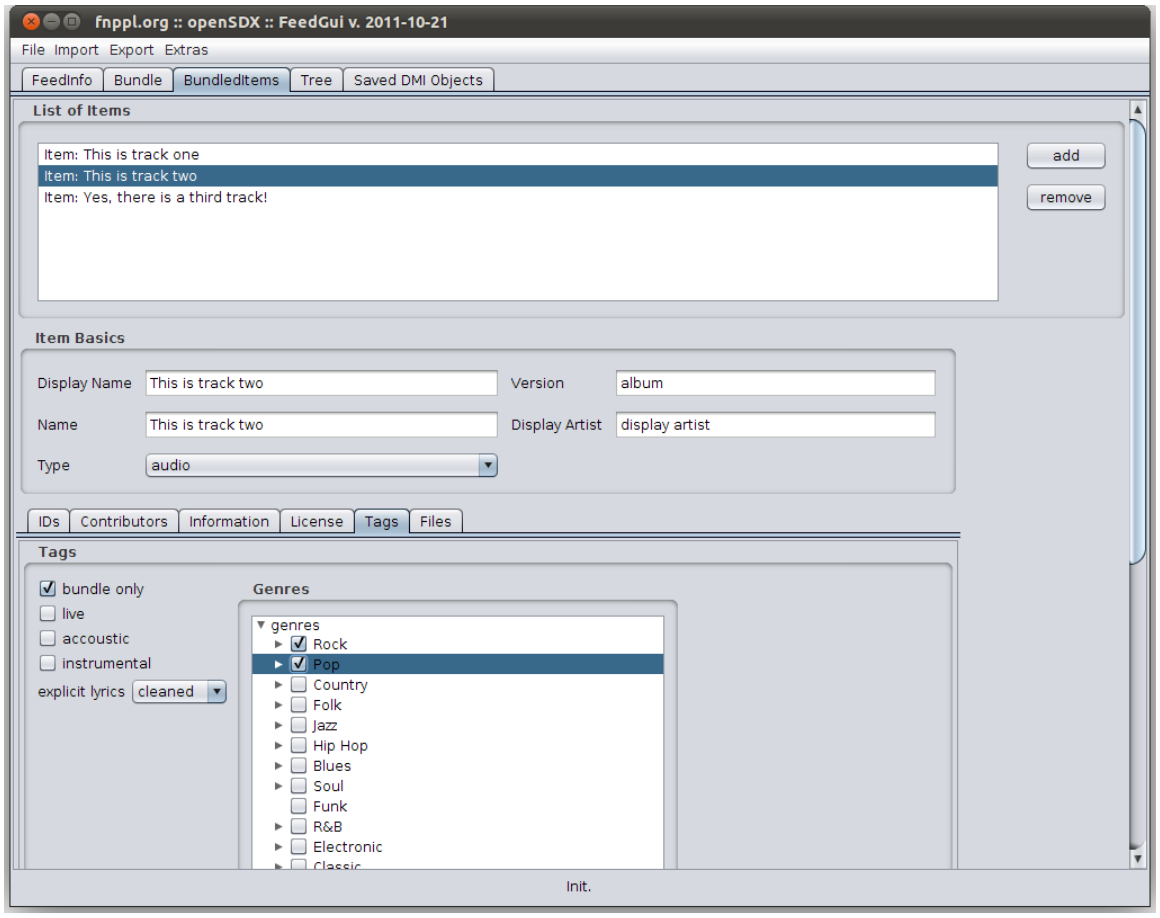
Display Name: This is track two
Version: album
Name: This is track two
Display Artist: display artist
Type: audio

License

✓ as on bundle

Item :: Tags

In the *Tags* sub tab you can set / unset the same tags on the item level as on bundle level (see: Bundle :: Tags)



List of Items

- Item: This is track one
- Item: This is track two
- Item: Yes, there is a third track!

Item Basics

Display Name	This is track two	Version	album
Name	This is track two	Display Artist	display artist
Type	audio		

Tags

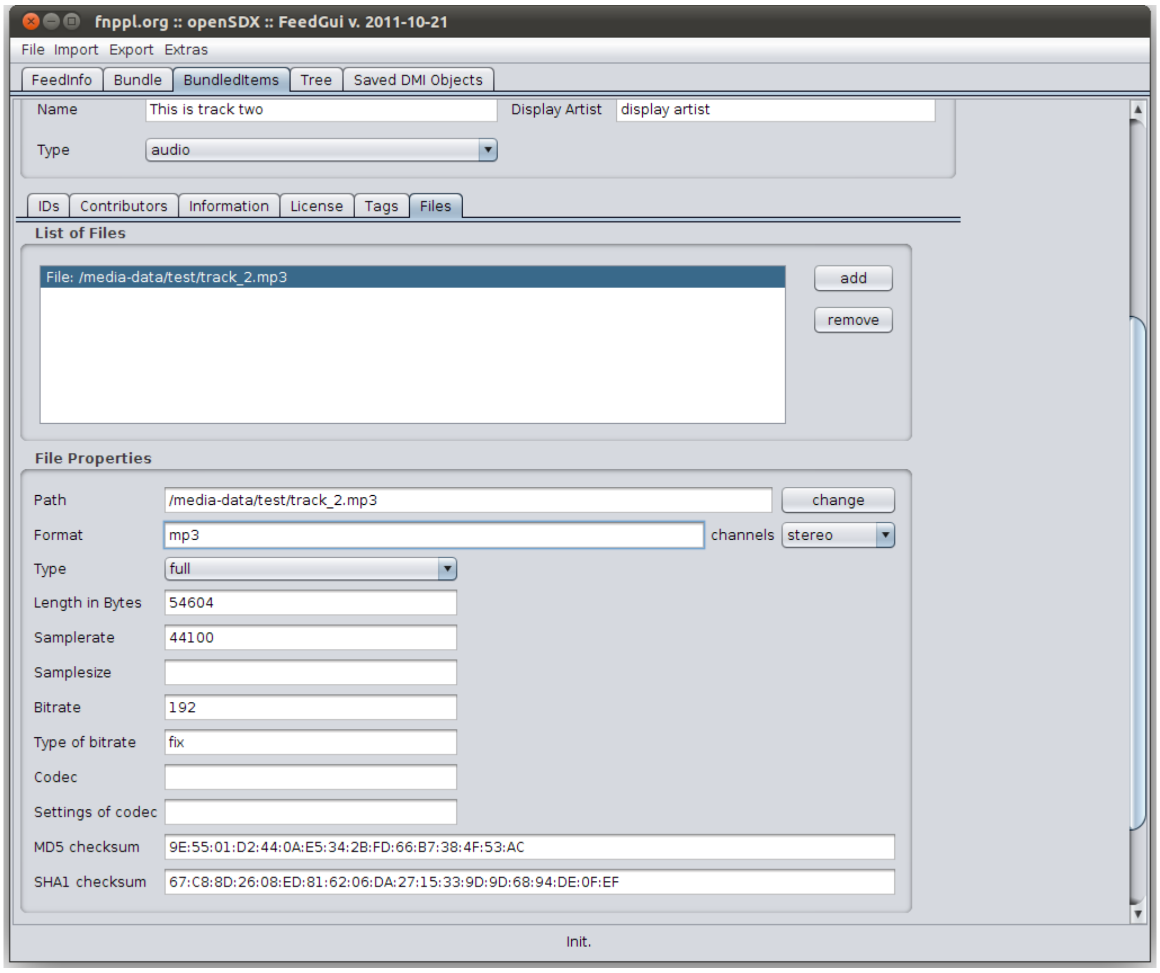
- [✓] bundle only
- [ ] live
- [ ] accoustic
- [ ] instrumental
- explicit lyrics: cleaned

Genres

- genres
  - [✓] Rock
  - [✓] Pop
  - [ ] Country
  - [ ] Folk
  - [ ] Jazz
  - [ ] Hip Hop
  - [ ] Blues
  - [ ] Soul
  - [ ] Funk
  - [ ] R&B
  - [ ] Electronic
  - [ ] Classic

Item :: Files

The media file or files that belong to the item can be set in the *files* sub tab. Typically you will select one media file per item (e.g. an audio file), but it is also possible to define one file with the full version (type = full) and a second file with a prelistening version (type = prelistening). The other fields from the *File Properties* section give information about the file and encoding format and the MD5 and SHA1 checksums will automatically be calculated.
What's the use of these checksums? When processing the feed on the receiver side, the related item files will most probably not be identified by their filenames but by their checksum values. This way one can be sure that the right files are used and that there was no modification (or missing part) in the files content.


fnppl.org :: openSDX :: FeedGui v. 2011-10-21
File Import Export Extras
FeedInfo Bundle BundledItems Tree Saved DMI Objects
IDs Contributors Information License Tags Files
add
remove
change
Init.

Name: This is track two
Display Artist: display artist
Type: audio

List of Files

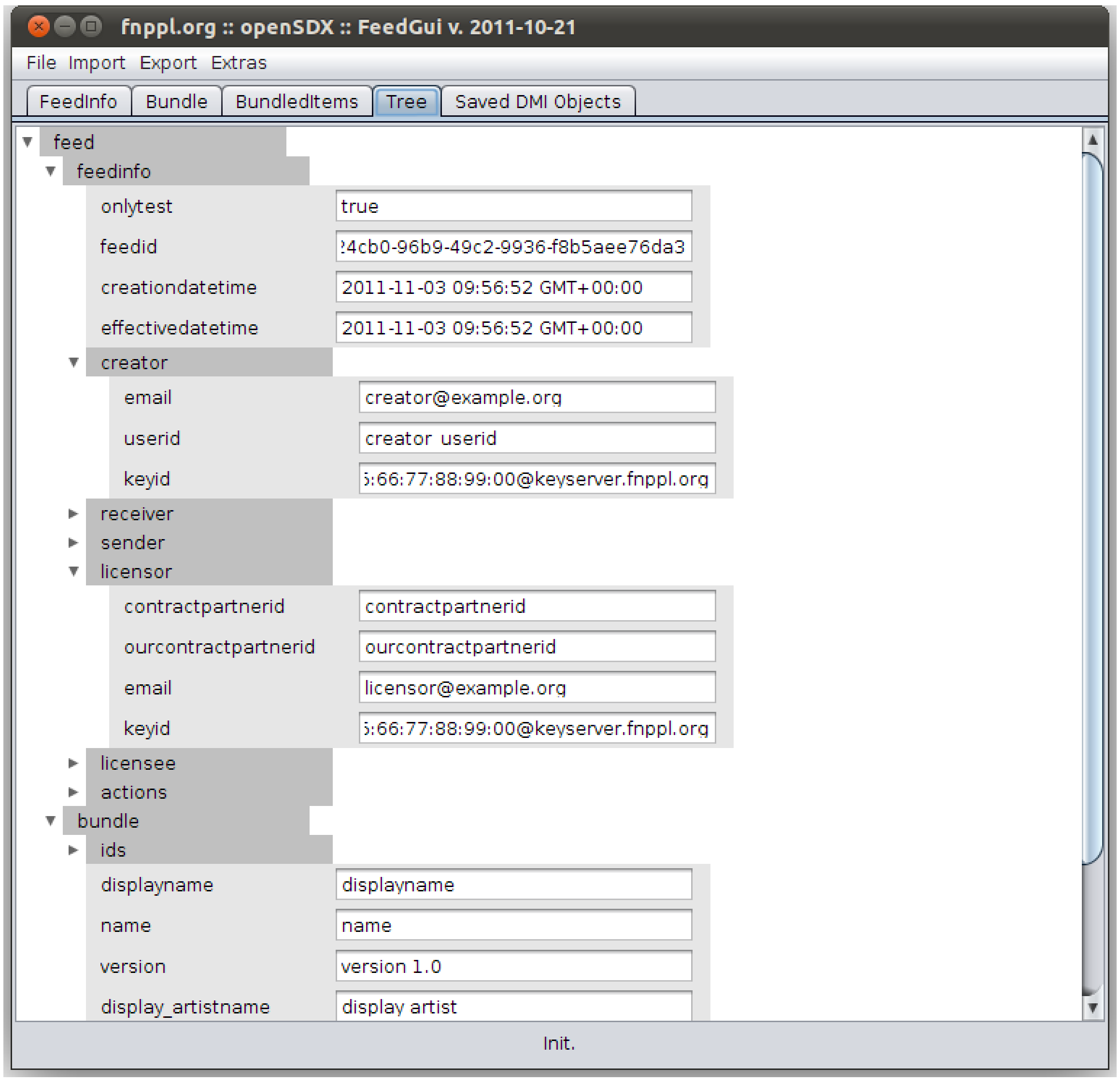
File: /media-data/test/track_2.mp3

File Properties

Path	/media-data/test/track_2.mp3
Format	mp3
channels	stereo
Type	full
Length in Bytes	54604
Samplerate	44100
Samplesize	
Bitrate	192
Type of bitrate	fix
Codec	
Settings of codec	
MD5 checksum	9E:55:01:D2:44:0A:E5:34:2B:FD:66:B7:38:4F:53:AC
SHA1 checksum	67:C8:8D:26:08:ED:81:62:06:DA:27:15:33:9D:9D:68:94:DE:0F:EF

Tree

In the *Tree* tab all feed data are represented in a tree structure which corresponds to the XML structure. On top level are the *feedinfo* and *bundle* parts, e.g. the bundled items can be found by the path: *bundle* -> *items* -> *item*. All data field can only be reviewed and not be edited.


fnppl.org :: openSDX :: FeedGui v. 2011-10-21
File Import Export Extras
FeedInfo Bundle BundledItems Tree Saved DMI Objects
Init.

feed
- feedinfo
  - onlytest: true
  - feedid: ?4cb0-96b9-49c2-9936-f8b5aee76da3
  - creationdatetime: 2011-11-03 09:56:52 GMT+00:00
  - effectivedatetime: 2011-11-03 09:56:52 GMT+00:00
  - creator
    - email: creator@example.org
    - userid: creator userid
    - keyid: 5:66:77:88:99:00@keyserver.fnppl.org
  - receiver
  - sender
  - licensor
    - contractpartnerid: contractpartnerid
    - ourcontractpartnerid: ourcontractpartnerid
    - email: licensor@example.org
    - keyid: 5:66:77:88:99:00@keyserver.fnppl.org
  - licensee
  - actions
- bundle
  - ids
  - displayname: displayname
  - name: name
  - version: version 1.0
  - display_artistname: display artist

Saved DMI Objects

You have the possibility to predefine a set of data objects that will be shown in the *FeedGui* and that can be used to fill some data parts in your feeds. For example your data for *sender*, *licensor* and *licensee* might be the same for a lot of feeds, so it would make sense to copy the data instead of entering it again for every new feed. The *Saved DMI Object* tab offers the possibility to browse your predefined data and add it to certain fields in the feed. At the top of the feed you see the *data path* field_ with the *select* and *read data* button. The *select* button opens a file dialog and lets you select a path where your predefined data is stored in XML formatted files. The default data directory is named *dmi_data* and is a subdirectory of *openSDX* in your local home directory. The *read data* function reads all .xml files in the given directory and extracts the useful information/data. An overview of all recognized entries will be shown in the table. If you click on an entry the data details will be shown in the panel below and one or more buttons (depending on the data type) at the bottom of the window will let you assign the data to a part of the feed.

To save your data in xml files you have to create UTF-8 encoded text files in xml format and with .xml file extension. For example you can copy the data of your choice out of a ready feed to a blank file with a texteditor. Please make sure you have entered/copied the right xml header at the beginning of the new file: <?xml version="1.0" encoding="UTF-8"?>

If you want to add multiple data chunks in one xml file, you can do that by starting the xml structure with a **<collection>** tag. (Don't forget to close this tag at the end of the file using **</collection>**)

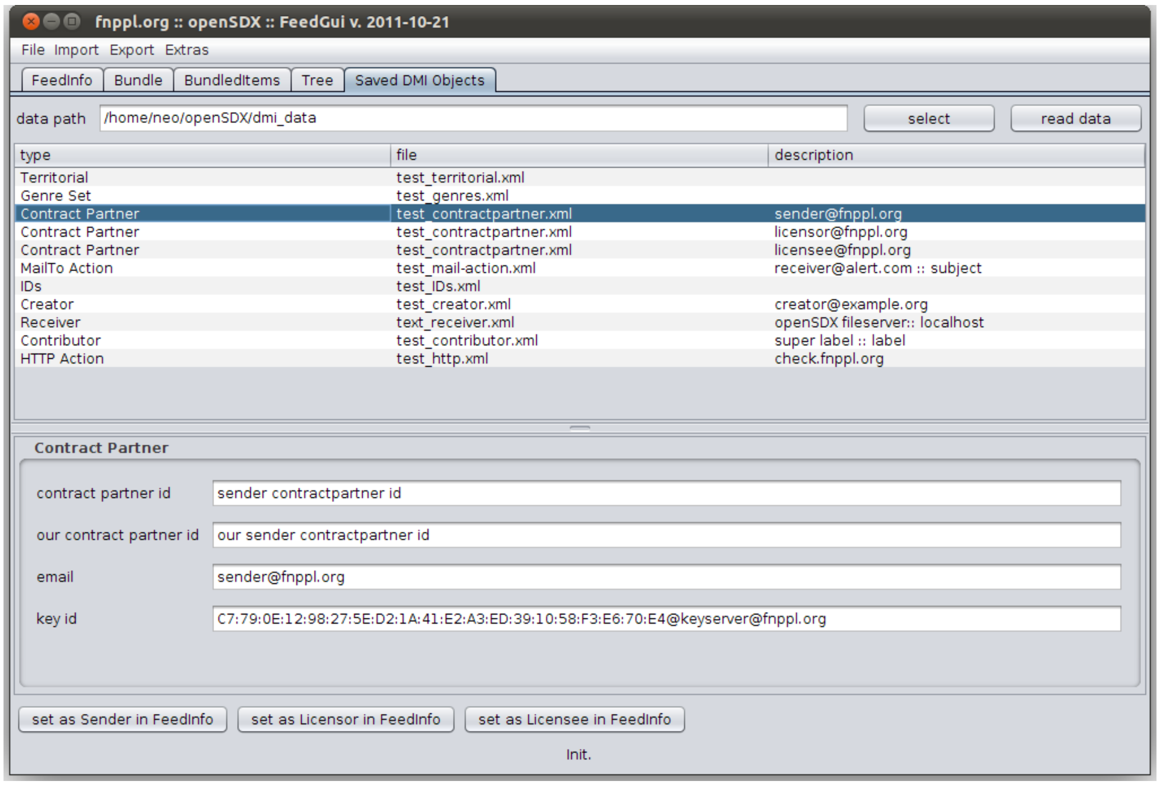
Example DMI Object Collection

```
<?xml version="1.0" encoding="UTF-8"?>
<collection>
   <sender>
      <contractpartnerid>sender contractpartner id</contractpartnerid>
      <ourcontractpartnerid>our sender contractpartner id</ourcontractpartnerid>
      <email>sender@fnppl.org</email>

<keyid>C7:79:0E:12:98:27:5E:D2:1A:41:E2:A3:ED:39:10:58:F3:E6:70:E4@keyserver@fnppl.org</keyid>
    </sender>

    <licensor>
      <contractpartnerid>licensor contractpartner id</contractpartnerid>
      <ourcontractpartnerid>our licensor contractpartnerid</ourcontractpartnerid>
      <email>licensor@fnppl.org</email>

<keyid>C8:79:0E:12:98:27:5E:D2:1A:41:E2:A3:ED:39:10:58:F3:E6:70:E4@keyserver@fnppl.org</keyid>
    </licensor>

    <licensee>
      <contractpartnerid>licensee contractpartner id</contractpartnerid>
      <ourcontractpartnerid>our licensee contractpartner id</ourcontractpartnerid>
      <email>licensee@fnppl.org</email>

<keyid>C9:79:0E:12:98:27:5E:D2:1A:41:E2:A3:ED:39:10:58:F3:E6:70:E4@keyserver@fnppl.org</keyid>
    </licensee>
</collection>
```

Since *sender*, *licensor* and *licensee* have the same data fields they are all shown as the type *Contract Partner*. All these contract partners could be assigned to the feedinfo part as sender, licensor or licensee using the buttons in the details view.

Other predefined data type are:

- receiver
- creator
- contributor
- sets of genres
- sets of territorial information
- IDs on bundle level
- HTTP and/or MailTo actions as triggered actions in the feedinfo part



fnppl.org :: openSDX :: FeedGui v. 2011-10-21
File Import Export Extras
FeedInfo Bundle BundledItems Tree Saved DMI Objects
select
read data
set as Sender in FeedInfo
set as Licensor in FeedInfo
set as Licensee in FeedInfo
Init.

data path /home/neo/openSDX/dmi_data

type	file	description
Territorial	test_territorial.xml	
Genre Set	test_genres.xml	
Contract Partner	test_contractpartner.xml	sender@fnppl.org
Contract Partner	test_contractpartner.xml	licensor@fnppl.org
Contract Partner	test_contractpartner.xml	licensee@fnppl.org
MailTo Action	test_mail-action.xml	receiver@alert.com :: subject
IDs	test_IDs.xml	
Creator	test_creator.xml	creator@example.org
Receiver	text_receiver.xml	openSDX fileserver:: localhost
Contributor	test_contributor.xml	super label :: label
HTTP Action	test_http.xml	check.fnppl.org

Contract Partner

contract partner id	sender contractpartner id
our contract partner id	our sender contractpartner id
email	sender@fnppl.org
key id	C7:79:0E:12:98:27:5E:D2:1A:41:E2:A3:ED:39:10:58:F3:E6:70:E4@keyserver@fnppl.org