
AIRS XML 2 iCarol Translation Documentation

Release 0.3.3

Chris Lambacher

June 05, 2015

1	AIRS XML 2 iCarol Translation	3
1.1	Features	3
2	Installation	5
3	Usage	7
4	Command Line Usage	9
5	Contributing	11
5.1	Types of Contributions	11
5.2	Get Started!	12
5.3	Pull Request Guidelines	12
5.4	Tips	13
6	Credits	15
6.1	Development Lead	15
6.2	Contributors	15
7	History	17
8	0.3.2 (2015-06-05)	19
9	0.3.2 (2015-05-25)	21
10	0.3.1 (2015-05-12)	23
11	0.3.0 (2015-05-12)	25
12	0.2.0 (2015-04-29)	27
13	0.1.0 (2015-04-10)	29
14	Indices and tables	31

Contents:

AIRS XML 2 iCarol Translation

A tool to convert AIRS XML export files into translation import files for iCarol

- Free software: Apache 2.0 License
- Documentation: <https://airs2icarol.readthedocs.org>.

1.1 Features

- Convert an AIRS XML v3.0 export into an iCarol translation file.
- Can use the command interface `airs2icarol` or as a Python module

Installation

At the command line:

```
$ pip install airs2icarol
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv airs2icarol  
$ pip install airs2icarol
```

Usage

To use AIRS XML 2 iCarol Translation in a project:

```
import airs2icarol

# get translations for selected language
translations = airs2icarol.get_stdtranslation(langauges = ['fr'])

airs2icarol.convert_xml('source.xml', 'dest.csv', 'fr', translations)
```

`source.xml` and `dest.csv` can optionally end in `.zip` they will then be treated as zip files containing a single file. The filename inside `source.zip` is ignored, and the filename in `dest.zip` will be `dest.csv`. The output CSV file will be encoded in UTF-8 prefixed with a UTF-8 BOM (Byte Order Mark).

Command Line Usage

When `airs2icarol` is installed, an `airs2icarol` command line application is also installed. This can be run as follows:

```
airs2icarol source.xml dest.csv fr
```

Where `source.xml` and `dest.csv` can optionally end in `.zip` they will then be treated as zip files containing a single file. The filename inside `source.zip` is ignored, and the filename in `dest.zip` will be `dest.csv`. The output CSV file will be encoded in UTF-8 prefixed with a UTF-8 BOM (Byte Order Mark). `fr` is the target translation culture code.

Contributing

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given. You can contribute in many ways:

5.1 Types of Contributions

5.1.1 Report Bugs

Report bugs at <https://github.com/lambacck/airs2icarol/issues>.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

5.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” is open to whoever wants to implement it.

5.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with “feature” is open to whoever wants to implement it.

5.1.4 Write Documentation

AIRS XML 2 iCarol Translation could always use more documentation, whether as part of the official AIRS XML 2 iCarol Translation docs, in docstrings, or even on the web in blog posts, articles, and such.

5.1.5 Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/lambacck/airs2icarol/issues>.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

5.2 Get Started!

Ready to contribute? Here's how to set up *airs2icarol* for local development.

1. Fork the *airs2icarol* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/airs2icarol.git
```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

```
$ mkvirtualenv airs2icarol
$ cd airs2icarol/
$ python setup.py develop
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

```
$ flake8 airs2icarol tests
$ python setup.py test
$ tox
```

To get flake8 and tox, just pip install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

5.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 2.6, 2.7, 3.3, and 3.4, and for PyPy. Check https://travis-ci.org/lambacck/airs2icarol/pull_requests and make sure that the tests pass for all supported Python versions.

5.4 Tips

To run a subset of tests:

```
$ python -m unittest tests.test_airs2icarol
```

Credits

6.1 Development Lead

- Chris Lambacher <chris@kateandchris.net>

6.2 Contributors

None yet. Why not be the first?

History

0.3.2 (2015-06-05)

- Internal Memo for Agency and Site

0.3.2 (2015-05-25)

- Handle contact XPATH depends on the language because the value for Type attribute changes based on language

0.3.1 (2015-05-12)

- Add missing Hours field for service
- Include translations in sdist

0.3.0 (2015-05-12)

- Make address line 1 join be newline instead of semi-colon

0.2.0 (2015-04-29)

- Fix French Translations
- Fix missing service description

0.1.0 (2015-04-10)

- First release on GitHub

Indices and tables

- `genindex`
- `modindex`
- `search`