

NAME

idl2wrs – CORBA IDL to Wireshark Plugin Generator

SYNOPSIS

idl2wrs <filename>

DESCRIPTION

idl2wrs is a program that takes a user specified **CORBA IDL** file and generates "C" source code for a **Wireshark** "plugin".

This resulting file can be compiled as a **Wireshark** plugin, and used to monitor **GIOP/IOP** traffic that is using this IDL.

idl2wrs is actually a shell script wrapper for two **Python** programs. These programs are:

- **wireshark_be.py** Contains the main IDL Visitor Class
- **wireshark_gen.py** Contains the Source Code Generator Class

idl2wrs supports heuristic dissection of GIOP/IOP traffic, and some experimental code for explicit dissection, based on Object Key \leftrightarrow Repository Id mapping. However, code for heuristic based plugins is generated by default, and users should consider this the preferred method unless you have some namespace collisions.

OPTIONS

Currently there are no options. **idl2wrs** can be invoked as follows.

1. To write the C code to stdout.

```
idl2wrs <your_file.idl>
```

```
eg: idl2wrs echo.idl
```

2. To write to a file, just redirect the output.

```
idl2wrs echo.idl > packet-test.c
```

ENVIRONMENT

idl2wrs will look for **wireshark_be.py** and **wireshark_gen.py** in `$PYTHONPATH/site-packages/` and if not found, will try the current directory `./`

The `-p` option passed to `omniidl` (inside **idl2wrs**) indicates where **wireshark_be.py** and **wireshark_gen.py** will be searched. This may need tweaking if you place these files somewhere else.

If it complains about being unable to find some modules (eg `tempfile.py`), you may want to check if `PYTHONPATH` is set correctly.

eg: `PYTHONPATH=/usr/lib/python1.5/`

SEE ALSO

wireshark(1), tshark(1)

NOTES

idl2wrs (including **wireshark_be.py** and **wireshark_gen.py**) are part of the **Wireshark** distribution. The latest version of **Wireshark** can be found at <https://www.wireshark.org>.

idl2wrs uses **omniidl**, an IDL parser, and can be found at <http://omniorb.sourceforge.net/>

TODO

Some of the more important things to do are:

- Improve Explicit dissection code.
- Improve command line options.
- Improve decode algorithm when we have operation name collision.

AUTHORS**Original Author**

Frank Singleton <frank.singleton[AT]ericsson.com>