

Accessing remote data

...and the tools used in an IDL project

Michael Galloy
Tech-X Corporation

Funding provided by NASA SBIR grants #NNX08CA99P and #NNX09CA72C

Data Access Protocol (DAP)

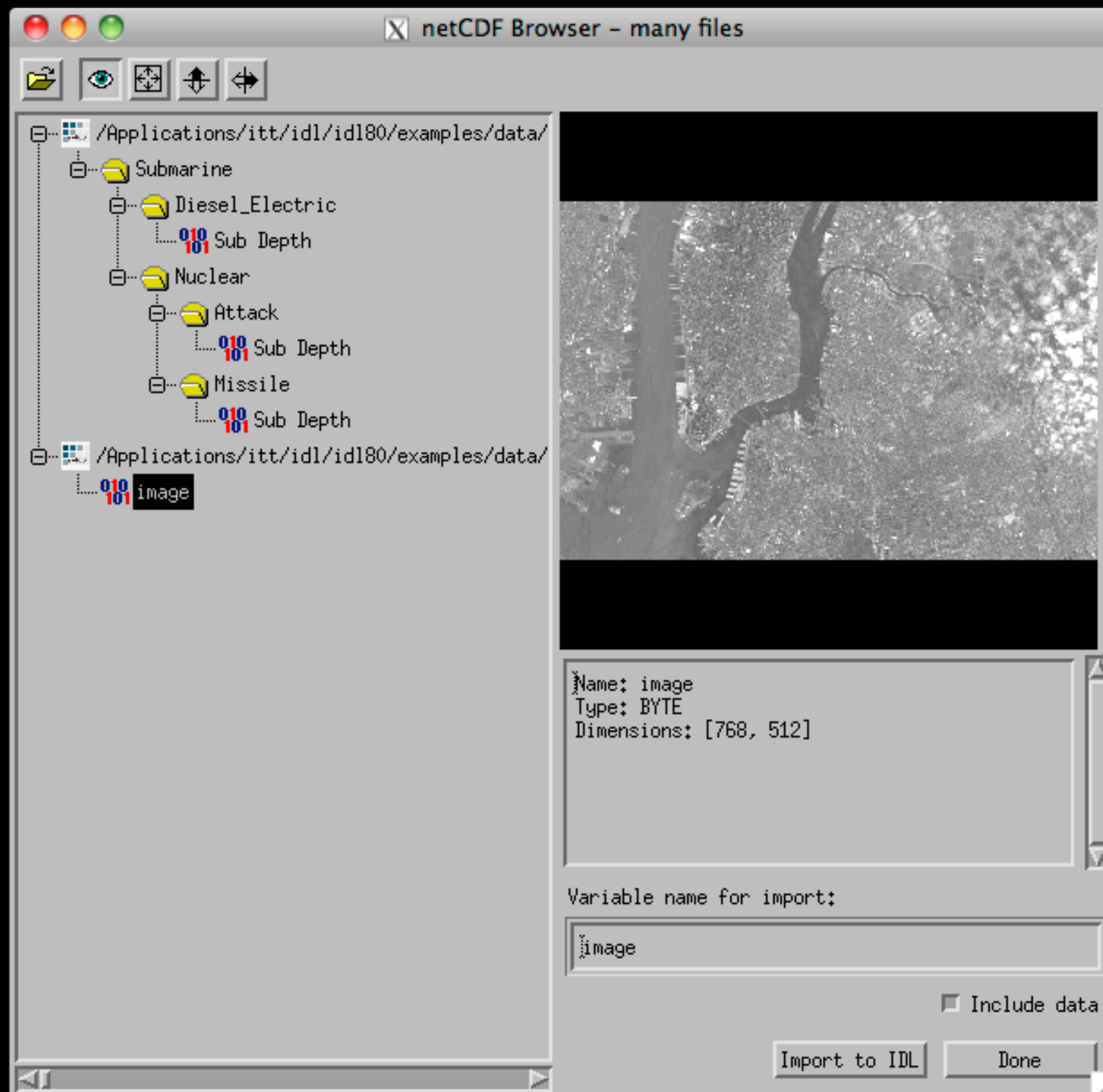
```
IDL> sample_filename = file_which('sample.nc')
IDL> tx_nc_dump, sample_filename
+ FILE </Applications/itt/idl/idl80/examples/data/sample.nc>
  . ATTRIBUTE TITLE = 'Incredibly Important Data'
  . ATTRIBUTE GALAXY = 'Milky Way'
  . ATTRIBUTE PLANET = 'Earth'
- VARIABLE bytarr(768, 512) image
  . ATTRIBUTE TITLE = 'New York City'
```

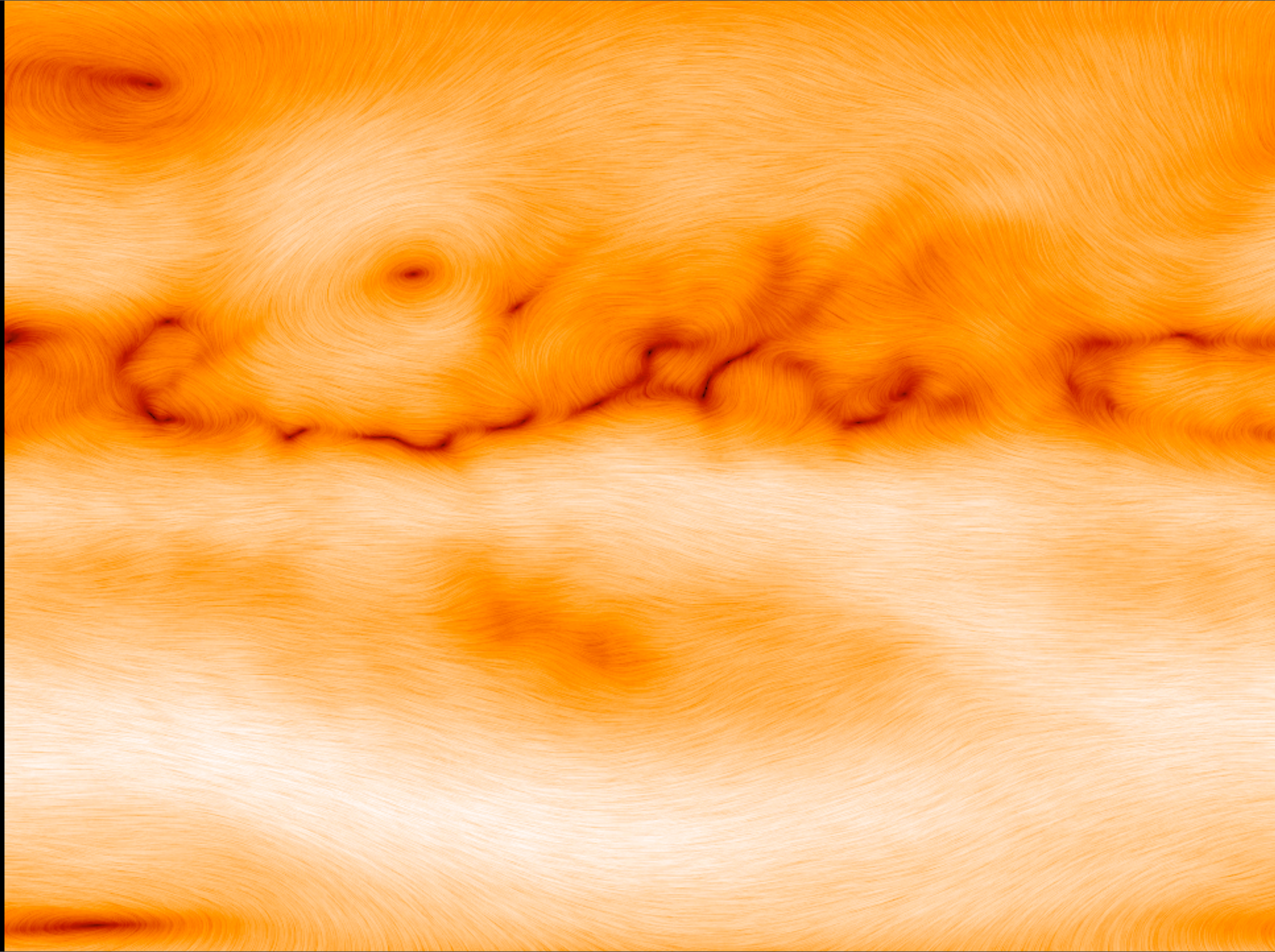
```
IDL> ncgroup_filename = file_which('ncgroup.nc')
IDL> tx_nc_dump, ncgroup_filename
+ FILE </Applications/itt/idl/idl80/examples/data/ncgroup.nc>
+ GROUP Submarine
+ GROUP Diesel_Electric
- VARIABLE intarr(2) Sub Depth
+ GROUP Nuclear
+ GROUP Attack
- VARIABLE intarr(4) Sub Depth
+ GROUP Missile
- VARIABLE intarr(3) Sub Depth
```

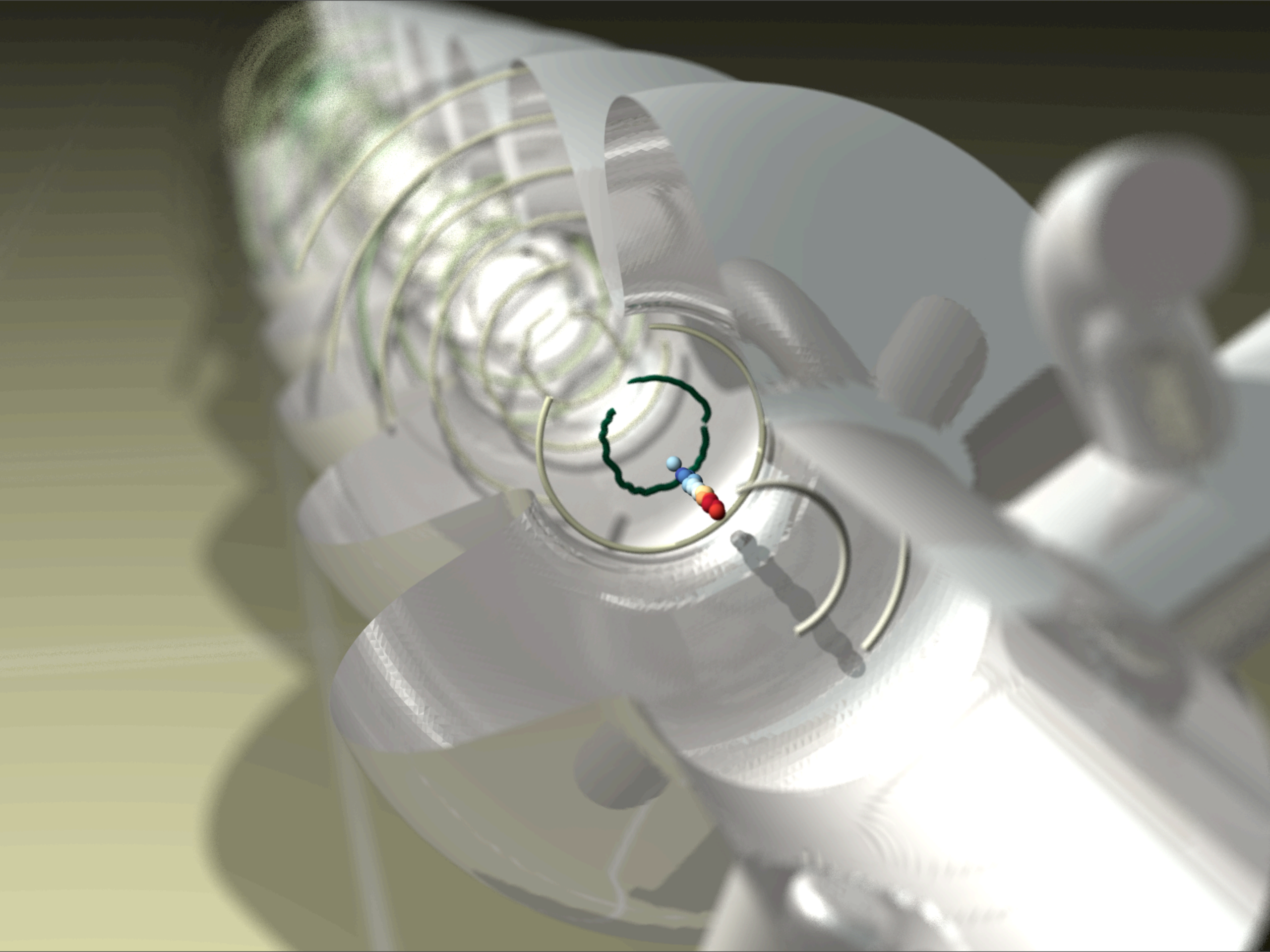
```
IDL> sample_filename = file_which('sample.nc')
IDL> im = tx_nc_getdata(sample_filename, '/image')
IDL> help, im
IM          BYTE          = Array[768, 512]
IDL> title = tx_nc_getdata(sample_filename, '/image.TITLE')
IDL> help, title
TITLE      STRING       = 'New York City'

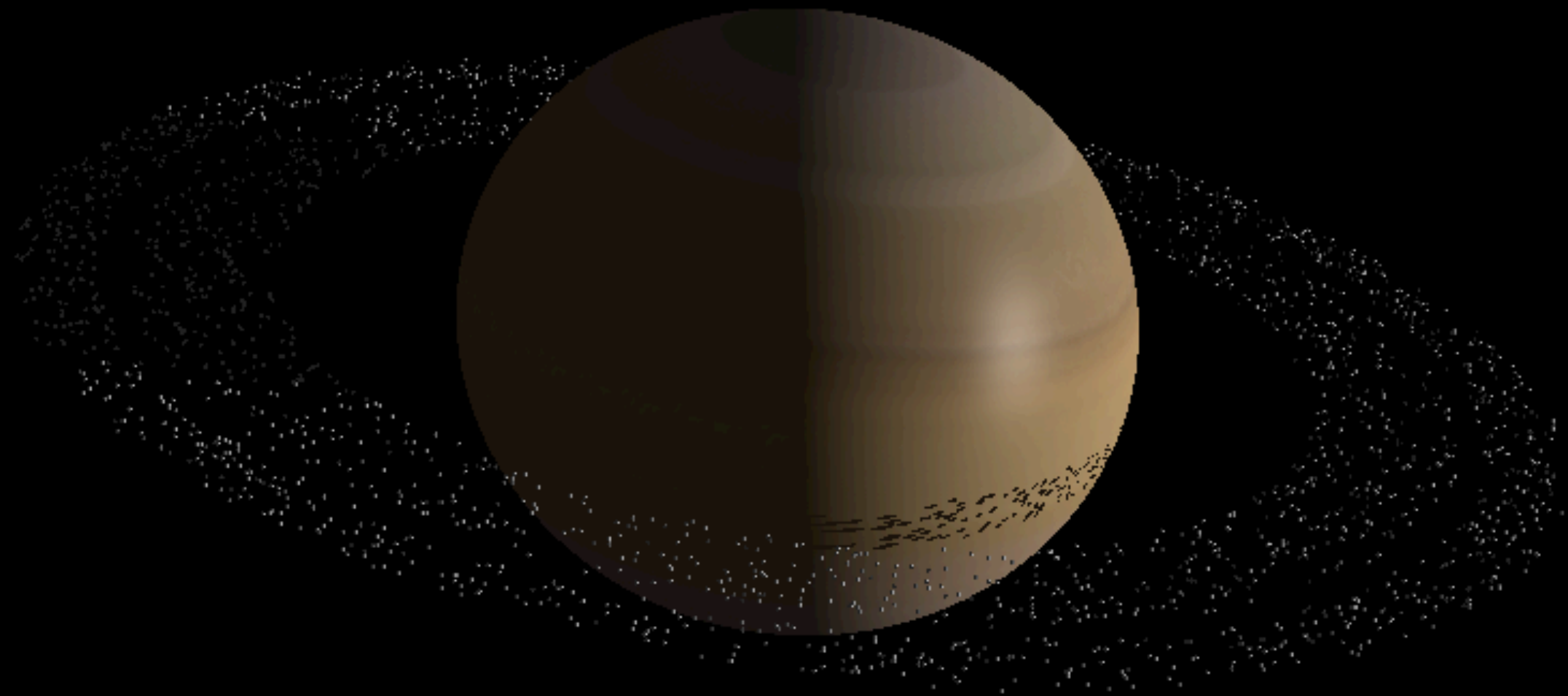
IDL> ncgroup_filename = file_which('ncgroup.nc')
IDL> variable = '/Submarine/Nuclear/Missile/Sub Depth'
IDL> depth = tx_nc_getdata(ncgroup_filename, variable)
IDL> print, depth
      410      447      304
```

```
IDL> f = TXffNCFfile(filename=file_which('sample.nc'))
IDL> print, f
+ FILE </Applications/itt/idl/idl80/examples/data/sample.nc>
  . ATTRIBUTE TITLE = 'Incredibly Important Data'
  . ATTRIBUTE GALAXY = 'Milky Way'
  . ATTRIBUTE PLANET = 'Earth'
- VARIABLE bytarr(768, 512) image
  . ATTRIBUTE TITLE = 'New York City'
IDL> help, f
F          NCFfile          = </Applications/itt/idl/idl80/
examples/data/sample.nc>
IDL> im = f['image']
IDL> help, im
IM          BYTE          = NCVariable:image[768, 512]
IDL> tv, im[*,*]
```









— Daniel Pomaredo, COAST: COmputational ASTrophysics in Saclay

Directories

[All files](#)

[lib/](#)

[netcdf/](#)

[txdap/](#)

[txdap/varobjects/](#)

[xdap/](#)

All files

.pro files 48 files

[electrons_plot_macro.pro](#)

[idl_object__define.pro](#)

[macro_test.pro](#)

[mg_src_root.pro](#)

[tx_arrayindtostr.pro](#)

[tx_base64decode.pro](#)

[tx_base64encode.pro](#)

[tx_braces2slash.pro](#)

[tx_converttype.pro](#)

[tx_hasneturl.pro](#)

[tx_index.pro](#)

[tx_isidlversion8.pro](#)

[tx_nc_browser.pro](#)

[tx_nc_dump.pro](#)

[tx_nc_getdata.pro](#)

[tx_newline.pro](#)

[tx_simplemap__define.pro](#)

[tx_slash2braces.pro](#)

[tx_structtoarray.pro](#)

[txcohashtable__define.pro](#)

[txdap__define.pro](#)

[txdap_atomic__define.pro](#)

[txdap_compile_opt.pro](#)

[txdap_convertbounds.pro](#)

IDL DAP bindings

Remote Data Exploration with IDL

[Overview](#) [Directory](#) [File](#) [Etc](#) [Categories](#) [Search](#) [Index](#) [Help](#)

[User documentation](#)

[single page](#) | [use frames](#) [directories](#) [project statistics](#)

Overview

The IDL source code for the RDL project. See the [TxDAP introduction](#) for more information about using the TxDAP bindings.

Directories

lib/	helper routines
netcdf/	routines/classes for accessing netCDF files and DAP data sets
txdap/	code for pure IDL DAP implementation
txdap/varobjects/	data objects for TxDAP
xdap/	GUI for exploring DAP servers

Project statistics

Directories:	5
.pro files:	48
.sav files:	0
Routines:	439

```
IDL> mgunit, 'txdaputnetcdftests_uts'  
"All tests" test suite starting (1 test suite/case, 8 tests)  
  "txdaputnetcdftests_uts" test suite starting (4 test suites/cases, 8 tests)  
    "tx_nc_getdata_ut" test case starting (2 tests)  
      test_group: passed (0.010783 seconds)  
      test_sample: passed (0.003909 seconds)  
    Results: 2 / 2 tests passed, 0 skipped  
    "txffncfile_ut" test case starting (2 tests)  
      test_group: passed (0.008113 seconds)  
      test_sample: passed (0.002207 seconds)  
    Results: 2 / 2 tests passed, 0 skipped  
    "txffncgroup_ut" test case starting (2 tests)  
      test_group: passed (0.008778 seconds)  
      test_sample: passed (0.001986 seconds)  
    Results: 2 / 2 tests passed, 0 skipped  
    "txffncvariable_ut" test case starting (2 tests)  
      test_group: passed (0.003976 seconds)  
      test_sample: passed (0.003071 seconds)  
    Results: 2 / 2 tests passed, 0 skipped  
  Results: 8 / 8 tests passed, 0 skipped  
Results: 8 / 8 tests passed, 0 skipped
```

```
; create the prefs object
prefs = obj_new('mgffprefs', author_name='txcorp', app_name= 'xdap')

; save preferences
self.prefs->save, 'url', _url
self.prefs->save, 'location', [event.x, event.y]

; relaunch
prefs = obj_new('mgffprefs', author_name='txcorp', app_name= 'xdap')

; restore preferences
location = self.prefs->get('location', found=locationFound)
```



```
mg_log, name='xdap', logger=logger
logger->setProperty, level=loggerLevel
logger->setProperty, filename=filepath('xdap.log', root=appdir), /append

catch, error
if (error ne 0L) then begin
    catch, /cancel
    mg_log, 'XDAP crash on startup', name='xdap', /critical, /last_error
endif

mg_log, 'Shutting down GUI interface', name='xdap', /informational

mg_log, 'Cleaning up XDAP object', name='xdap', /informational, /quit
```

```
Wed Mar 30 15:58:12 2011 INFORMATIONAL: XDAP::INIT: Starting GUI interface
Wed Mar 30 15:58:14 2011 INFORMATIONAL: XDAP::CLEANUPWIDGETS: Shutting down GUI interface
Wed Mar 30 15:58:14 2011 INFORMATIONAL: XDAP::CLEANUP: Cleaning up XDAP object
Wed Mar 30 15:59:10 2011 INFORMATIONAL: XDAP::INIT: Starting GUI interface
Wed Mar 30 15:59:20 2011 INFORMATIONAL: XDAP::CLEANUPWIDGETS: Shutting down GUI interface
Wed Mar 30 15:59:20 2011 INFORMATIONAL: XDAP::CLEANUP: Cleaning up XDAP object
Wed Mar 30 15:59:21 2011 INFORMATIONAL: XDAP::INIT: Starting GUI interface
Wed Mar 30 15:59:25 2011 INFORMATIONAL: XDAP::CLEANUPWIDGETS: Shutting down GUI interface
Wed Mar 30 15:59:25 2011 INFORMATIONAL: XDAP::CLEANUP: Cleaning up XDAP object
Wed Mar 30 15:59:27 2011 INFORMATIONAL: XDAP::INIT: Starting GUI interface
Wed Mar 30 15:59:29 2011 INFORMATIONAL: XDAP::CLEANUPWIDGETS: Shutting down GUI interface
Wed Mar 30 15:59:29 2011 INFORMATIONAL: XDAP::CLEANUP: Cleaning up XDAP object
Wed Mar 30 16:01:32 2011 CRITICAL: XDAP::INIT: XDAP crash on startup
Wed Mar 30 16:01:32 2011 DEBUG: Stack trace for error:
% Attempt to call undefined procedure/function: 'UNKNOWN_ROUTINE'.
% Execution halted at: XDAP::INIT          1359 /Users/mgalloy/projects/rdl/src/xdap/xdap.pro
%                      XDAP              1438 /Users/mgalloy/projects/rdl/src/xdap/xdap.pro
%                      $MAIN$
```

```
opts = obj_new('mg_options', app_name='mg_options_example',  
              version='1.0')  
opts->addOption, 'verbose', 'v', $  
              /boolean, $  
              help='set to print a verbose greeting'  
opts->addOption, 'name', 'n', help='name of user to greet',  
              default='Mike', metavar='user''s name'  
opts->parseArgs, error_message=errorMsg
```

```
$ idl mg_options_example -args --name=Mike
```

```
Hello, Mike!
```

```
$ idl mg_options_example -args --verbose --name=Mike
```

```
Greetings and salutations, Mike!
```

```
$ idl mg_options_example -args --help
```

```
usage: mg_options_example [options]
```

```
options:
```

```
--help, -h          display this help
```

```
--name=user's name, -n user's name
```

```
name of user to greet
```

```
--verbose, -v      set to print a verbose greeting
```

```
--version          display version information
```

```
d1m = mg_d1m(basename='idlgs1', $
             name='IDLGSL', $
             description='IDL bindings for GSL', $
             version='1.0', $
             source='Michael Galloy')
d1m->addInclude, ['gsl_math.h', 'gsl_sf_zeta.h', 'gsl_sf_ellint.h'], $
             header_directory='/usr/local/include/gsl', $
             lib_directory='/usr/local/lib', $
             lib_files='gsl'
d1m->addRoutinesFromHeaderFile, 'idlgs1_gsl_sf_ellint_bindings.h'
d1m->addRoutinesFromHeaderFile, 'idlgs1_gsl_sf_zeta_bindings.h'
d1m->addPoundDefineAccessor, 'M_EULER', type=5L
d1m->write
d1m->build, /show_all_output
```



```
// double gsl_sf_ellint_F(double phi, double k, unsigned int mode);
static IDL_VPTR IDL_gsl_sf_ellint_F(int argc, IDL_VPTR *argv, char *argk) {
    IDL_ENSURE_SCALAR(argv[0])
    MG_ENSURE_TYPE(argv[0], IDL_TYP_DOUBLE)
    IDL_ENSURE_SCALAR(argv[1])
    MG_ENSURE_TYPE(argv[1], IDL_TYP_DOUBLE)
    IDL_ENSURE_SCALAR(argv[2])
    MG_ENSURE_TYPE(argv[2], IDL_TYP_ULONG)
    double result = (double) gsl_sf_ellint_F(argv[0]->value.d,    // double phi
                                             argv[1]->value.d,    // double k
                                             argv[2]->value.ul); // unsigned int mode
    return MG_get_double(result);
}
```

rIDL 0.1.r172M: Really Interactive Data Language. [Build: Nov 29 2010]
IDL Version 8.0.1, Mac OS X (darwin x86_64 m64). (c) 2010, ITT Visual Information Solutions
Installation number: 209577.
Licensed for use by: Tech-X Corporation

```
[501]> :history 5 nonum
mgunit, 'tx_nc_getdata_ut'
mgunit, 'txdaputnetcdftests_uts'
mgunit, 'txdaputnetcdftests_uts'
.compile txdaputnetcdftests_uts__define
mgunit, 'txdaputnetcdftests_uts'
```

```
[501]> p
p_correlate      plot_field      poly            popd            print           ps_show_fonts
parse_url        ploterr         poly_2d         powell          printd          psafm
particle_trace  plots           poly_area      pref_commit     printf          pseudo
path_cache       pm              poly_fit       pref_get        product         ptr_free
path_sep         pnt_line       polyfill       pref_migrate    profile         ptr_new
pcomp            point_lun       polyfillv      pref_set        profiler        ptr_valid
plot             polar_contour  polyshade      prewitt         profiles        ptrarr
plot_3dbox       polar_surface  polywarp       primes          project_vol     pushd
[501]> p
```