NAME

sdl – the Shore Data Language Compiler

SYNOPSIS

```
sdl [ -s sourcefile ] [ -b modulename ] [ -l modulename ]
    [ -o outputfile ] [ -d pathname ] [ -S ]
    [ -B ] [ -L ] [ -v ] [ -f ] [ -c ]
```

DESCRIPTION

This is the Shore Data Language (SDL) compiler, linker, and C++ language binding generator. It performs one or more of these functions, determined by command-line arguments. **Sdl** is itself a Shore/SDL application program, and uses the Shore database to store information about the types it processes. Consequently, *you must have a Shore server running* when you use **sdl.**

Processing your SDL module descriptions

When compiling an SDL source file, **sdl** processes one or more SDL module definitions; for each such definition processed, it creates a *module object* within the Shore database. Each module object is a registered object in the Shore namespace. These module objects are used as the basis for further processing. **sdl** also creates a collection of other objects in the database which contain information about the types specified within the module; the *module object* is used as the root object for any access to these objects.

Linking your SDL modules objects together

If an SDL module specification uses types that are defined in a different module, the appropriate modules objects must be linked. That is, any intermodule references within a given module must be resolved with respect to other module objects within the Shore database. Linking resolves intermodule references. If a module definition makes no intermodule references, linking of the corresponding module object is not required.

Generating a language binding.

Language binding generation for a module object creates a C++ header file containing class definitions corresponding to each object type specified by that module,and other runtime support code used in creating objects of these types and moving these objects in and out of application program memory. It also allows operations on object types to be specified as member functions of the corresponding C++ class produced by the language binding.

Shore now has one target language - C++. For further information on use of the C++ language binding, see **intro(cxxlb).**

IMPORTANT NOTE

The Shore Server process, `shore`, must already be running when `sdl` is invoked.

Sdl reads the configuration file `.shoreconfig` in the current directory, if it exists, and if not, in `$HOME/.shoreconfig`, if it exists. If neither exists, **sdl** will fail to contact the server.

OPTIONS

-s sourcefile

Process the SDL source file *sourcefile*. If no errors occur, a module object is created for each SDL module definition in *sourcefile*, using the module name tag in the source file as the final component of the path name for the module object. The module objects are created within the first directory specified using the **-d** option described below; if no such option is used, the module object is created in the Shore directory `/types .` If the name of the file is '-', the standard input is used as source for the compilation. If standard input is used as source, it must be the first source file specified.

-S Use standard input as SDL source. This is equivalent to **-s** -.

-f Overwrite existing module if its reference count is zero. A module shows up in the name space in two places: as a pool and as another registered object, both with names derived from the eponymous module. If the **-f** flag is not set, the compilation will abort if either of the objects exists and the module has a zero reference count.

If the module has a non-zero reference count, the module will not be removed even if **-f** is used. Under these circumstances, you can remove the module objects and any remaining instances "by hand" (through a non-SDL path such as NFS or the server's Tcl interface), Be aware that this will make applications behave in an undefined manner when operating on instance objects that were built with the previous version of the module. If you destroy a module "by hand", you should destroy your database as well.

Reference counting is off by default; it can be turned on with the configuration options `oc_refcount` (see **options(oc);** *caveat emptor*).

-c Do a quick syntax check only on any SDL source. The Shore server need not be running if this flag is set; any operations other than compilation will fail if -c is used.

-v Verbose mode; print out command line arguments; then print what **sdl** thinks it is doing.

-l modulename
Link the named module object; attempt to resolve any intermodule references within *modulename*. Modules used in resolving these references will be any other modules specified named the **use** and **import** statements in the SDL specification of *modulename*; sdl will search for corresponding modules in the list of other modules specified using **-l** and in any directories specified using **-d**.

-L Link all modules created from SDL source in this run.

-b modulename
Produce a C++ language binding for the named module object; create a header file *modulename.h* which may be included by C++ application source. Any intermodule references within *modulename* must have been resolved via linking before a language binding may be produced.

-B Produce C++ language bindings for all modules created from source in this run. Normally there will be one header file per module, but see -o below.

-o outputfile
Send the C++ language binding output from any modules specified with **-b** or **-B** to the unix file *outputfile*. If the name of the file is '-', the standard output is used as the output file.

-d pathname
Insert the Shore directory *pathname* in the search path used for module name lookup during linking and language binding creation; the first directory specified is also used as the destination for any modules created during compilation. The Shore directory `/types` is the directory used if no "-d" arguments are given.

VERSION

This manual page applies to Version 1.1 of the Shore software.

SPONSORSHIP

The Shore project is sponsored by the Advanced Research Project Agency, ARPA order number 018 (formerly 8230), monitored by the U.S. Army Research Laboratory under contract DAAB07-91-C-Q518.

COPYRIGHT

SEE ALSO

intro(sdl), options(oc), and **intro(cxxlb)**

Also see **Shore Data Language Reference Manual,** for information on the definition of SDL, and **Getting Started with Shore** for a tutorial introduction to application writing using Shore and SDL.