

# ThriftFeatures

## Thrift Features

### Features

- interface description language - Everything is specified in an IDL file from which bindings for many languages can be generated. See [ThriftIDL](#)
- language bindings - Thrift is supported in many languages and environments
  - Actionscript 3 (AS3)
  - C (using GLib)
  - C++
  - C#
  - Cocoa (Objective-C on Mac OS and iOS)
  - Delphi
  - Erlang
  - Go
  - Haskell
  - Java
  - Javascript
  - OCaml
  - Perl
  - PHP
  - Python (including Twisted async support)
  - Ruby
  - Smalltalk
- namespaces - Each Thrift file is in its own namespace allowing you to use the same identifier in multiple Thrift files
- language namespaces - Per Thrift file you can specify which namespace should be used for each programming language
- base types - Thrift has a small set of base types. See [ThriftTypes](#)
- constants and enumerations - Constant values can be assigned logical names
- structs - Use structs to group related data. Structs can have fields of any type. See [ThriftTypes](#)
- sparse structs - Optional base fields that have not been set and reference fields that are `null` will not be sent across the wire
- struct evolution - The addition and removal of fields is handled without breaking existing clients by using integer identifiers for fields
- containers - You can use sets, lists and maps of any type: base types, structs and other containers. See [ThriftTypes](#)
- type definitions - Any type can be given a name that better describes it
- services - A service is a group of functions
- service inheritance - Subservices implement all functions of their base services and can have additional functions
- asynchronous invocations - Functions that do not return a result can be invoked asynchronously so the client is not blocked until the server has finished processing the request. The server may execute asynchronous invocations of the same client in parallel/out of order
- exceptions - If an error occurs a function can throw a standard or user-defined exception. See [ThriftTypes](#)

### Non-features

The following is not supported by Thrift:

- cyclic structs - Structs can only contain structs that have been declared before it. A struct also cannot contain itself
- struct inheritance - Use struct composition instead
- polymorphism - As there is no inheritance, polymorphism is also not supported
- overloading - All methods within a service must be uniquely named
- heterogeneous containers - All items in a container must be of the same type
- `null` return - `null` cannot be returned directly from a function. Use a wrapper struct or a marker value instead