```
// messenger.idl

interface Messenger {

   boolean send_message (

         in      string user_name,

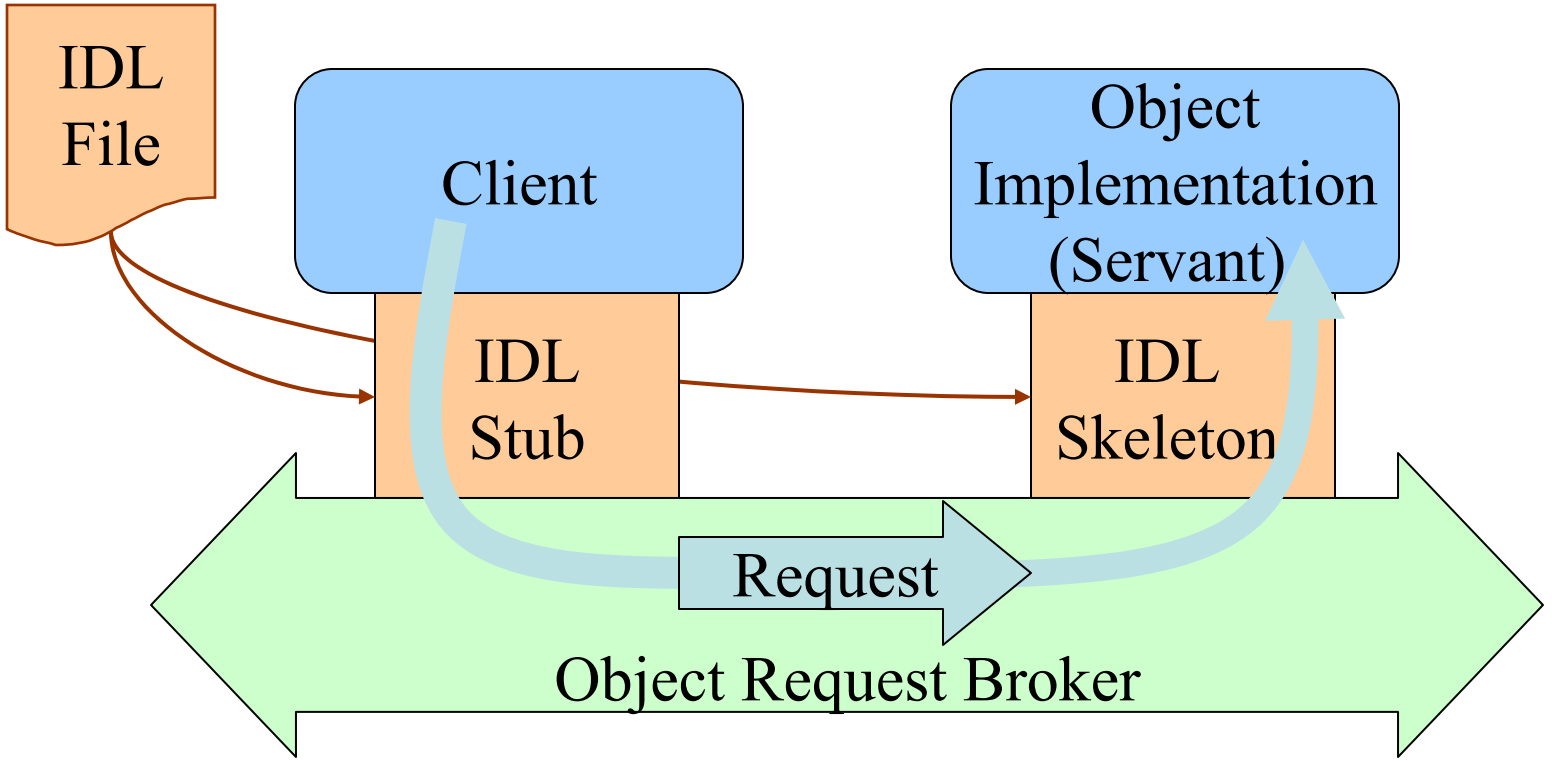
         in      string subject,

         inout   string message );

};
```

```
module BankExample {
    interface Account {
        exception BadCheck {
            float fee;
          };
        float deposit(in float amount);
        float writeCheck(in float amount)
            raises (BadCheck);
     };
    interface AccountManager {
        Account openAccount(in string name);
     };
};
```

CORBA Diagram



(c) 2001 A. David McKinnon, used
with permission

```
#include <Messenger_skel.h>

class Messenger_impl : public POA_Messenger,
   public PortableServer::RefCountServantBase
{
public:
  virtual bool send_message
       (const char* user_name,
        const char* subject,
        char*& message)
       throw(CORBA::SystemException);
};
```

```
int main(int argc, char *argv[]) {
  int status = EXIT_SUCCESS;
  CORBA::ORB_var orb;
  try {
    orb = CORBA::ORB_init(argc,argv);
    status = run(orb); }
  catch(const CORBA::Exception&) {
    status = EXIT_FAILURE;  }

  if (!CORBA::is_nil(orb)) {
    try {
      orb->destroy();    }
    catch(const CORBA::Exception&) {
      status = EXIT_FAILURE;    }  }
  return status;
}
```

```
int run(CORBA::ORB_ptr orb) {
  const char* refFile = "Messenger.ref";
  ifstream in(refFile);
  char s [2048];
  in >> s;
  in.close();
  CORBA::Object_var obj = orb->string_to_object(s);
  CORBA::String_var message = CORBA::string_dup("Howdy!");
  Messenger_var messenger = Messenger::_narrow(obj);

  if (messenger->send_message("hauser", "Test message", message.inout())) {
    cout << "server ok" << endl;
    cout << "Return message from server: " << message << endl;
  }
  else {
    cout << "server returned false" << endl;
    cout << "Return message from server: " << message << endl;
  }
  return 0;
}
```

```
int run(CORBA::ORB_ptr orb) {
  CORBA::Object_var poaObj = orb->resolve_initial_references("RootPOA");
  PortableServer::POA_var rootPoa =
                        PortableServer::POA::_narrow(poaObj);
  PortableServer::POAManager_var manager =
                        rootPoa->the_POAManager();
  Messenger_impl* messengerImpl = new Messenger_impl();
  PortableServer::ServantBase_var servant = messengerImpl;
  Messenger_var messenger = messengerImpl -> _this();

  CORBA::String_var s = orb -> object_to_string(messenger);
  const char* refFile = "Messenger.ref";
  ofstream out(refFile);
  out << s << endl;
  out.close();
  manager -> activate();
  orb->run();
  return EXIT_SUCCESS;
}
```

(c) 2001 A. David McKinnon, used with permission