Calling Haskell from C

Jump to: navigation, search

It is not uncommon to want to call a Haskell function from C code. Here's how to do that.

We define the fibonacci function in Haskell:

```
{-# LANGUAGE ForeignFunctionInterface #-}

    module Safe where

    import Foreign.C.Types

    fibonacci :: Int -> Int
    fibonacci n = fibs !! n
        where fibs = 0 : 1 : zipWith (+) fibs (tail fibs)

    fibonacci_hs :: CInt -> CInt
    fibonacci_hs = fromIntegral . fibonacci . fromIntegral

    foreign export ccall fibonacci_hs :: CInt -> CInt
```

Note the foreign export. When GHC sees this, it will generate stubs for C, to help it work out the Haskell types.

And call it from C:

```
#include <HsFFI.h>
    #ifdef __GLASGOW_HASKELL__
    #include "Safe_stub.h"
    extern void __stginit_Safe(void);
    #endif
    #include <stdio.h>

    int main(int argc, char *argv[]) {
       int i;
       hs_init(&argc, &argv);
    #ifdef __GLASGOW_HASKELL__
       hs_add_root(__stginit_Safe);
    #endif

       i = fibonacci_hs(42);
       printf("Fibonacci: %d\n", i);

       hs_exit();
       return 0;
    }
```

Now, first compile the Haskell file:

```
   $ ghc -c -O Safe.hs
```

Which creates Safe_stub.c, Safe_stub.o, Safe_stub.h, which you import into your C program. Now compile your C code with ghc (!), passing the Haskell objects on the command line:

```
   $ ghc -optc-O test.c Safe.o Safe_stub.o -o test
```

(Alternatively, fewer files to enumerate:

```
   $ ghc --make -no-hs-main -optc-O test.c Safe -o test
```

)

Then run your C code:

```
   $ ./test
   Fibonacci: 267914296
```

And that's it.

Category: Tutorials

Navigation

Haskell Wiki community Recent changes Random page

Toolbox

What links here Related changes Upload file Special pages Printable version Permanent link