Subject: How to call a C++ function in Witz template?
Posted by Peter on Thu, 21 Feb 2013 10:47:17 GMT

Hi.

According to Witz manual it's possible to call a C++ function in Witz template. Unfortunately there's no example showing how to do that. In one of my source files I defined a trivial function:

```
void fun(int& i)
{
   i++;
}
```

and Skylark handler:

```
SKYLARK(AAA, "aaa")
{
   http("MYVAR", Value(2));
   http("MYFUN", fun);
   http.RenderResult("Project_name/aaa");
}
```

The corresponding aaa.witz template looks like this:

```
<html>
<body>

$MYVAR
$MYFUN($MYVAR)
$MYVAR

</body>
</html>
```

I expected to see the following output: 2 3 (old and new value of MYVAR). Instead I got the following error message:

"function nor link not found 'MYFUN'".

I assumed I just had to pass a pointer to my function as a value of shared variable MYFUN to make it visible in Witz template, but apparently that's not how it works. What's the correct way to

do it?

Subject: Re: How to call a C++ function in Witz template?
Posted by dolik.rce on Thu, 21 Feb 2013 11:08:24 GMT

Hi Peter,

Peter wrote on Thu, 21 February 2013 11:47I assumed I just had to pass a pointer to my function as a value of shared variable MYFUN to make it visible in Witz template, but apparently that's not how it works. What's the correct way to do it?

It would be nice if it would work this way, but unfortunately it doesn't (at least not yet). You have to register the function and it has to have correct signature. The simple increment function from your example can be done like this (not tested):

```
Value Fun(const Vector<Value>& arg, const Renderer *)
{
 if(arg.GetCount() != 1 && !IsNumber(arg[0]))
  return "ERROR";
 return (int)arg[0] + 1;
}
```

```
INITBLOCK {
 Compiler::Register("fun", Fun);
};
```

This code should be placed in .icpp file, to make sure it is linked correctly. Now you can call the function in witz using $fun(MYVAR), it does not have to be passed to the Http.

Is that clearer now? Further examples can be found in Skylark/StdLib.icpp.

Best regards,
Honza