



AHELP for CIAO 3.4

is_struct_defined

Context: [varmm](#)

Jump to: [Description](#) [Examples](#) [Bugs](#) [See Also](#)

Synopsis

S-Lang function to see if a structure or field in a structure is defined

Syntax

```
Integer_Type is_struct_defined( String_Type sf )
```

Description

Determine if `sf` is "defined" as a field variable within a structure. The input argument should be the name of a structure or the name of the structure and field. The return value is 1 if the input structure or field exists, 0 otherwise. Note that this routine does not check whether the variable contains any data; for this you would use the `__is_initialized()` routine from the S-Lang Run-Time Library.

The variable containing the structure must be visible from other compilation units; ie it must not be declared as either static or private.

Example 1

With the following definition:

```
chips> variable s = struct { foo, bar }
```

then we find

```
chips> is_struct_defined( "s" )  
1  
chips> is_struct_defined( "s.foo" )  
1  
chips> is_struct_defined( "s.bar" )  
1
```

but

```
chips> is_struct_defined( "s.baz" )
0
```

Example 2

If we repeat the previous example, but let ChIPS automatically create the S–Lang variable for us, then we find that the routine will always return 0. This is because the variable is defined to have static linkage; see "ahelp _auto_declare" for more information.

```
chips> r = struct { baz }
chips> is_struct_defined( "r" )
0
chips> is_struct_defined( "r.baz" )
0
```

Example 3

Here we show the routine being used in S–Lang code run by slsh. If the file struct.sl contains:

```
require("varmm");
variable a = struct { one, two };
private variable b = struct { one, two };
vmessage( "Is a.one defined? %d", is_struct_defined("a.one") );
vmessage( "Is b.two defined? %d", is_struct_defined("b.two") );
```

then running the code with slsh would produce

```
unix% slsh struct.sl
Is a.one defined? 1
Is b.two defined? 0
```

since the variable b was declared as being a private variable.

Bugs

See the [bugs page for the Varmm library](#) on the CIAO website for an up-to-date listing of known bugs.

See Also

modules

[varmm](#)

varmm

[apropos](#), [clearstack](#), [dup_struct](#), [print](#), [reverse](#)