

```

bool_t
rpcb_gettime(host,timep)
    char *host = (char *)SvPV($arg,PL_na);
    time_t &timep = 0;
OUTPUT:
    timep

```

This should not be used to supply default values for parameters. One would normally use this when a function parameter must be processed by another library function before it can be used. Default parameters are covered in the next section.

If the initialization begins with =, then it is output in the declaration for the input variable, replacing the initialization supplied by the typemap. If the initialization begins with ; or +, then it is performed after all of the input variables have been declared. In the ; case the initialization normally supplied by the typemap is not performed. For the + case, the declaration for the variable will include the initialization from the typemap. A global variable, %v, is available for the truly rare case where information from one initialization is needed in another initialization.

Here's a truly obscure example:

```

bool_t
rpcb_gettime(host,timep)
    time_t &timep ; /* \${timep}=@{[\${timep}=$arg]} */
    char *host + SvOK($v{timep}) ? SvPV($arg,PL_na) : NULL;
OUTPUT:
    timep

```

The construct `\${timep}=@{[\${timep}=$arg]}` used in the above example has a two-fold purpose: first, when this line is processed by **xsubpp**, the Perl snippet `\${timep}=$arg` is evaluated. Second, the text of the evaluated snippet is output into the generated C file (inside a C comment)! During the processing of `char *host` line, `$arg` will evaluate to `ST(0)`, and `\${timep}` will evaluate to `ST(1)`.

68.1.16 Default Parameter Values

Default values for XSUB arguments can be specified by placing an assignment statement in the parameter list. The default value may be a number, a string or the special string `NO_INIT`. Defaults should always be used on the right-most parameters only.

To allow the XSUB for `rpcb_gettime()` to have a default host value the parameters to the XSUB could be rearranged. The XSUB will then call the real `rpcb_gettime()` function with the parameters in the correct order. This XSUB can be called from Perl with either of the following statements:

```

$status = rpcb_gettime( $timep, $host );

$status = rpcb_gettime( $timep );

```

The XSUB will look like the code which follows. A CODE: block is used to call the real `rpcb_gettime()` function with the parameters in the correct order for that function.

```

bool_t
rpcb_gettime(timep,host="localhost")
    char *host
    time_t timep = NO_INIT
CODE:
    RETVAL = rpcb_gettime( host, &timep );
OUTPUT:
    timep
    RETVAL

```