**Ofront+****  
Oberon-2 to C Translator, Version 1.0  
User Guide**

**SOFTWARE TEMPL OEG, 1995-2007**

**VEDAsoft Oberon Club, 2013-2022**

**SYSTEM flags**

# MODULE [foreign]

* No code C file is generated for a foreign module, only .oh header.
* Accordingly, a foreign module cannot contain tagged records that need type initialization. All tagged records declared in a foreign module will be untagged.
* Export mark "\*" may be omitted, all declarations are considered exported. You can use read-only exports (marked "-") as well as IN and OUT parameters.
* The tag of an array is its length field. Thus, arrays can be tagged. If you want an untagged array, declare it as ARRAY [notag]
* If you imported any modules from a foreign module (why?), they will not be initialized.

# MODULE [noinit]

* This type of module was designed as a lightweight module that plugs in without any overhead.
* No initialization procedure Module\_\_init is generated for a noinit module. Accordingly, there is no initialization of all modules imported from it.

# MODULE [main]

* This type of module was designed as main module (the analogue of the -m option).

# PROCEDURE [ccall]

* The cdecl (which stands for C declaration) is a calling convention for the C programming language and is used by many C compilers.
* In cdecl, function arguments are pushed on the stack in the right-to-left order, i.e. the last argument is pushed first. The caller cleans the stack after the function call returns.
* cdecl is used by Ofront+ as the default calling convention. That is, the presence of the system flag [ccall] and not specifying a system flag are equivalent.

# PROCEDURE [stdcall]

* The stdcall calling convention is a variation on the Pascal calling convention in which the callee is responsible for cleaning up the stack, but the parameters are pushed onto the stack in right-to-left order, as in the \_cdecl calling convention.
* The stdcall is the standard calling convention for the Microsoft Win32 API and for Open Watcom C++.

# PROCEDURE [fastcall]

* This calling convention usually involves passing parameters in registers. Use this in the way your C compiler supports.

# PROCEDURE [inline]

* An inline function is usually a function whose definition is small, and its body can be substituted wherever the function call occurs. The function substitution is not mandatory and is fully determined by a compiler.
* Use this in the way your C compiler supports.