

PIM4 vs GNU Modula-2 vs M2R10 vs TopSpeed V3 vs V3 — feature comparison

Date: 2026-10-24 · **V3 state:** open arrays, dialect library folders and separate compilation with M2 -I/incremental objects (suite 310/310, self-hosting fixpoint OK 3,845,439 bytes, corpus 521/612, 0 crashes).

Scope and sources.

- **PIM4** — Wirth's *Programming in Modula-2*, 4th edition (the language specification, not a compiler).
- **GNU Modula-2 (gm2)** — the GCC front end: PIM and ISO modes plus the extensions common in its testsuite.
- **M2R10** — the Kowarsch/Sutcliffe *Modula-2 R10* revision (2012). It is a deliberate redesign, **not source-compatible** with classic Modula-2. Cells follow the published R10 grammar and the document “*Differences between Modula-2 R10 and classic Modula-2*”. Some features are library-level or outside the base grammar; those are marked as such. A cell saying “not in the R10 grammar” means the classic keyword/operator is not in the base grammar, not necessarily that no library equivalent exists.
- **TopSpeed V3** — the JPI/TopSpeed (Clarion) V3 dialect as reconstructed in compiler/src/TopSpeed-V3-M2.atg and docs/00P.txt. This is a historical compiler dialect, not a standard; cells describe that grammar/dialect where it is documented.
- **V3** — this compiler: a PIM/ISO subset plus the gm2 compatibility it deliberately chose.

Legend: ✓ supported / standard · ⚡ partial or caveated · ✕ absent · ⚠ non-standard extension · n/a not applicable.

The tables describe common/default behaviour; exact flags and modes can change individual cells.

Standards & dialect

Feature	PIM4	GNU Modula-2	M2R10	TopSpeed V3	V3
PIM base language	✓ (definition)	✓ (PIM 2/3/4 modes)	⚡ modern revision, not source-compatible	✓ PIM-based dialect	✓ (PIM/ISO subset)
ISO 10514-1 base	✕ (predates ISO)	✓	✕ (independent R10 spec)	✕	⚡ mostly; see missing ISO rows
ISO 10514-2 generics	✕	✓	✕ (external template engine, not ISO generics)	✕ (not in the V3 dialect grammar)	✓ (refinement form)
ISO 10514-3 OO	✕	✕	✕	✕ (Clarion classes, not ISO OO)	✕ (CLASS is Clarion-form)
Declaration before use	✓ strict	⚡ multi-pass, accepts forward refs	✓ strict	✓ strict (single-pass)	✓ strict for data; ⚡ forward procedure calls
FORWARD declarations	✓	✓	✕ (not in the R10 grammar)	✓	✓

Feature	PIM4	GNU Modula-2	M2R10	TopSpeed V3	V3
EXPORT QUALIFIED / UNQUALIFIED	✗ (PIM export list only)	✓	✗ (EXPORT removed; import qualifiers/re-export)	✓ (EXPORT [QUALIFIED])	✓
Foreign modules (DEFINITION MODULE FOR "C")	✗	✓	△ FFI via pragma, not FOR "C"	✗ (not in the reconstructed grammar)	✓

Generics (explicit)

Dialect	Language-level generics
PIM4	✗ no generic modules
GNU Modula-2	✓ ISO 10514-2 generic modules (type/const/procedure parameters)
M2R10	✗ not in the language; R10 uses an external template engine (GENLIB/BLUEPRINT, ##...## placeholders) expanded before compilation
TopSpeed V3	✗ no GENERIC construct in the reconstructed V3 grammar
V3	✓ ISO 10514-2 refinement modules: type/value/procedure parameters, constraints, nested generics, generic classes (docs/analysis_generics.md)

Lexical

Feature	PIM4	GNU Modula-2	M2R10	TopSpeed V3	V3
Identifiers with _	✗ (classic names)	✓	✓	✓ (_ and @)	✓
Numeric bases	✓ octal B, hex H	✓ B/H (and more)	✓ C-style 0b/0x, U char-code suffix (octal removed)	✓ H/C/B suffixes + 0x	✗ (0x/0X only)
Nested (* *) comments	✓	✓	✓ (max 10 levels)	✓	✓
'...' and '..."	✓	✓	✓ + C-style escapes	✓	✓
Unicode	✗	✗	✓ UNICHAR	✗	✓ UCHAR/UString (extension)

Types

Feature	PIM4	GNU Modula-2	M2R10	TopSpeed V3	V3
Basic integer/real/bool/char types	✓	✓	✓ (+ OCTET, LONGCARD)	✓ PIM types	✓

Feature	PIM4	GNU Modula-2	M2R10	TopSpeed V3	V3
Subranges	✓	✓	✓ ([lo..hi] OF Base)	✓	✓ (typed CHAR/enum subranges)
Enumerations Sets (SET OF)	✓ ✓ any finite ordinal (impl. limit)	✓ ✓	✓ extensible ✗ SET OF <enumType> only	✓ ✓ range/base/enums bases	✓ ✗ bounded sets ≤ 1,000,000 bits; SET OF CARDINAL/INTEGER use a dynamic sorted range list over the full base range
PACKEDSET	✗ (ISO)	✓	✗	✗	✓ (1/2/4 bytes, span ≤ 32)
Fixed / multi-dimensional arrays	✓ [a], [b] OF	✓	✓ ARRAY count, count OF (zero-based)	✓	✓ both [a], [b] and [a, b]
Open arrays (1-D)	✓	✓	✓ (BARE ARRAY OF)	✓	✓
Multi-dimensional open arrays	✗ (ISO addition)	✓	✗ one-dimensional BARE arrays	✗	✓ (flat payload + per-level views; legacy 1-D ABI adapter)
Records	✓	✓	✓ extensible	✓	✓ (incl. self-referential pointer fields and flat multi-dimensional array fields)
Variant records	✓	✓	✗ removed (extensible records replace them)	✓	✓ (plus empty arms/ELSE)
Pointers, NIL, opaque types	✓	✓	✓ (OPAQUE keyword)	✓	✓ (incl. chained p^.next^.value)
ADDRESS ↔ pointer equality	n/a	✓	n/a	✓	✓ (ADR(p^) = p)
Procedure types	✓ (PROCEDURE)	✓ (PROCEDURE, PROC)	✓ (comma-separated formals)	✓	✓ (PROCEDURE, PROC; comma/un-named forms)
Complex types (COMPLEX, LONGCOMPLEX)	✗	✓ (ISO)	✗ (not in the base R10 grammar)	✗	✗ (COMPLEX64 is a REAL placeholder)

Feature	PIM4	GNU Modula-2	M2R10	TopSpeed V3	V3
SYSTEM sized types (WORD32, CARDINAL8, BITSET32, ...)	✗	✓	✗ OCTET (8-bit), LONGCARD, LONGBITSET	✗ SHORTINT/LONGINT family	✓ 8/16/32 INT/SHORTCARD/LONGCARD
BYTE granularity	✓ 8-bit	✓ 8-bit	✓ OCTET 8-bit	✓ (PIM BYTE)	✓ BYTE/SBYTE/WORD8/CARDINAL8 are 8-bit; ARRAY OF BYTE is a real buffer
SHORTREAL (32-bit)	✗	✓	✗ (REAL/LONGREAL)	⚠ dialect-dependent	✓ true 4-byte SHORTREAL/REAL32
ADDRESS arithmetic (ADDADR/SUBADR/DIFADR)	✗	✓	✗	✗	✗
Absolute variables VAR [addr]: T	✗	✓ (extension)	✓ (pragma ADDR)	✗	✗
Set ordering (<, <=, >, >=)	✗ (=/#/IN only)	✓ (extension)	✗	✗	✗

Statements & procedures

Feature	PIM4	GNU Modula-2	M2R10	TopSpeed V3	V3
IF/CASE/WHILE/REPEAT/LOOP/EXIT/WITH, RETURN, HALT	✓	✓	✗ no WITH/GOTO; LOOP/EXIT, FOR ... IN	✓ + IN/IS, <</>>	✓
FOR over any ordinal (CHAR/enum)	✓	✓	✗ FOR ... IN over ranges/iterables (loop variable declared in the header)	✓ classic FOR ... TO	✓ (since 2026-10-22)
GOTO / labels	✓	✓	✗ (not in the R10 grammar)	✓	✓
INC/DEC on enumerations	✗ (integer)	✓	✗ replaced by SUCC/PRED	✗ dialect-dependent	✓ (any ordinal)
Empty string ""/' ' as CHAR	✗/unclear (ISO rule)	✓	✗ (char literals compatible with ARRAY OF CHAR; empty-to-CHAR not explicit)	✓	✓ (ISO rule)
LENGTH	✗ (ISO adds it)	✓	✗ library/pervasive, not in the base grammar	✗	✓

Feature	PIM4	GNU Modula-2	M2R10	TopSpeed V3	V3
HIGH	✓	✓	✗ (not in the R10 grammar)	✓	✓ (including string CONST)
TSIZE/SIZE/ADR ✓		✓	↯ SYSTEM/library forms	✓	✓ array payload SIZE; multi/open ADR = payload base; fixed 1-D ADR keeps descriptor compatibility
Runtime subrange range checks	⚠ error per language; checks implementation-defined	✓	✓ (R10 safety)	↯	✓ assignment/value actual/RETURN/INC/DEC; catchable rangeException; FOR final increment exempt
SHIFT/ROTATE	⚠ implementation SYSTEM extension	✓	✗	✓ (<</>> operators)	✓
TBITSIZE	✗	✓	✗	✗	✗
ISO exceptions (EXCEPT, RETRY, EXCEPTIONS)	✗	✓	✗ (not in the base R10 grammar)	✗	↯ EXCEPT/RETRY/SYSTEM.TH EXCEPTIONS/M2EXCEPTION index/range/CASE/function traps, module FINALLY
Module finalization (FINALLY)	✗	✓	✗	✗	✓ normal termination, reverse init order; not on HALT/unhandled abort
__FILE__ / __LINE__ / __FUNCTION__	✗	✓	↯ COMPILER pseudo-module introspection	✗	✓ / ✓ / ✓
__BUILTIN__ / __ATTRIBUTE__ pragmas	✗	✓	✓ language pragmas < * ... * >	⚠ compiler directives	✓ (parsed, ignored)
Inline ASM	✗	✓	⚠ optional ASSEMBLER pseudo-module	✗ (not in the reconstructed grammar)	✗

Modules, compilation, libraries

Feature	PIM4	GNU Modula-2	M2R10	TopSpeed V3	V3
Definition / implementation / program modules	✓	✓	✓ (+ blueprints)	✓	✓
Local modules	✓	✓	✗ removed	✓	✓
Separate compilation	✓	✓ (object files/archives)	✓ compilation units	✓	✓ M2 -c -I per-unit objects + M2make --separate; one-shot remains default
Stable cross-object symbols / init order	n/a	✓	✓	✓	✓ <Module>__proc, module-qualified pools/vtables, dependency-ordered <Module>_init
Incremental object rebuild / ABI marker	n/a	✓	✓	✓	✓ stale mtime rebuilds, content ABI/flags stamp, compiler/shim relink triggers
Generics	✗	✓ ISO 10514-2	✗ external templates (GENLIB/BUEPRINT)	✗	✓ ISO 10514-2 refinement
Object orientation	✗	✗	✗ (extensible records only)	✓ Clarion-form CLASS, VIRTUAL, inheritance	✗ Clarion-form CLASS (single inheritance, virtual dispatch), not ISO 10514-3
PIM coroutines (PROCESS, NEWPROCESS, TRANSFER, IOTRANSFER)	✓	✓	✗ COROUTINE type/YIELD (R10 phase II)	✓ PIM-style	✓ (IOTRANSFER degrades to a transfer)
ISO COROUTINES module	✗	✓	✗	✗	✗ (PIM interface only)

Feature	PIM4	GNU Modula-2	M2R10	TopSpeed V3	V3
Standard library	✗ (none defined)	✓ PIM + ISO libraries	✓ redesigned R10 library	✓ TopSpeed libraries (historical)	✗ subset in lib/pim, lib/iso, lib/common, lib/gm2: ISO/PIM I/O, strings, storage, SysClock, EXCEPTIONS; no termination/coroutine libs
Self-hosting	n/a	✓ (part of GCC)	n/a (reference compiler under development)	n/a (historical commercial compiler)	✓ one-shot byte-identical 3-stage fixpoint + per-object stage2/stage3 .ssa gate
Backend / target	n/a	GCC (many targets)	reference compiler	x86/DOS/Windows (historical)	QBE → C (amd64 SysV today)

The practical differences

- **PIM4** is the clean specification: strict declaration order, no ISO exceptions/complex numbers/PACKEDSET/generics, no standard library, no sized SYSTEM types or address arithmetic.
- **GNU Modula-2** is the superset in practice: PIM and ISO modes, generics, exceptions, complex numbers, PACKEDSET, multi-dimensional open arrays, sized SYSTEM types, address arithmetic, absolute variables, ASM, pragmas, and many extensions — but deliberately permissive (multi-pass resolution accepts forward references).
- **M2R10** is a **redesign, not an extension**: no variant records, no local modules, no WITH, no EXPORT, no GOTO, no FORWARD; it adds OCTET, UNICHAR, LONGBITSET, ALIAS OF, OPAQUE, extensible records/enumerations, FOR ... IN, SUCC/PRED, CAST, structured literals, pragmas and a redesigned library. Its “generics” are an external template engine, not ISO generic modules.
- **TopSpeed V3** is a PIM-based **dialect with OO**: Clarion-form CLASS/VIRTUAL and inheritance are the main additions; it has its own lexical forms (_/@, H/C/B literals, <</>>, WITH/IN/IS). It has no language-level generics or ISO exceptions.
- **V3** is a PIM/ISO compiler with the opposite bias to gm2: it rejects gm2 leniencies (declaration order), implements the common ISO/PIM constructs (ISO 10514-2 generics, coroutines, foreign modules, ISO strings/SYSTEM/procedure types, empty string as CHAR), and documents its chosen gm2 compatibility (__FILE__/_LINE__/_FUNCTION__, scalar→ARRAY OF BYTE, same-size VAR, bare proc values, variant extensions, set CASE labels, libc prototypes). ISO exceptions are **mostly complete** (EXCEPT/RETRY/SYSTEM.THROW, EXCEPTIONS/ M2EXCEPTION, index/range/CASE/function traps, and module FINALLY). Fixed and open multi-dimensional arrays use a flat payload with per-level views; payload SIZE/ADR and row/subarray conversion are implemented. Its hard gaps are **complex arithmetic, finalization/protection, ISO OO** and parts of the **word model** (SHORTREAL, REAL width), plus a **1,000,000-bit bounded SET limit**, dynamic SET OF CARDINAL/INTEGER range lists and a **32-bit PACKEDSET limit**. Self-referential pointer records and chained pointer-field designators (p^.next^.value) are supported. **Separate compilation** is now available for normal modules: M2 -c emits one object per unit, library objects have no main, and M2make --separate links the objects; the one-shot mode remains the default for tests and self-hosting.

V3-specific deviations and limits

- **Strict data declaration order** (PIM-conforming); forward **procedure** calls are allowed where strict PIM would require FORWARD.
- **Word model:** BYTE/SBYTE/WORD8/CARDINAL8/INTEGER8 are true 8-bit types (SIZE 1), WORD16/CARDINAL16/INTEGER16 are true 16-bit types (SIZE 2); INTEGER/CARDINAL/WORD/WORD32 are 32-bit; LONGINT/LONGCARD are 64-bit; REAL/LONGREAL are 8 bytes, while SHORTREAL/REAL32 are true 4-byte floats; COMPLEX* is a placeholder.
- **Sets:** bases limited to bool/char/byte/16-bit/enum/small subranges; bounded SET spans up to 1,000,000 bits; SET OF CARDINAL and SET OF INTEGER use sorted range lists over the full base range. Dynamic SET CONST is not supported yet. PACKEDSET uses 1/2/4-byte storage with a 32-bit span limit (wider spans are a loud 230).
- **Runtime range checks** are emitted for subrange assignment, value actuals, RETURN and INC/DEC (QbeGen.CheckRange); they raise a catchable rangeException, with an unhandled exception aborting at rc 134. The post-increment of a FOR control variable is not checked: the final successor may legitimately exceed a full subrange.
- **Module FINALLY** runs on normal main return in reverse initialization order. HALT, unhandled aborts and the ISO TERMINATION module are not covered.
- **Multi-dimensional arrays** use a flat payload and per-level 40-byte views. A legacy 1-D {count,data} descriptor is still used at ABI boundaries; row/ARRAY OF BYTE conversions are copy-based there, so row ADR identity through an internal 1-D call remains a follow-up.
- **Separate compilation:** M2 -c emits a QBE object per unit and M2make --separate resolves interfaces and links with cc. Library objects have no main; every object exports <Module>_init and the program calls imported inits in dependency order. Stable symbols are <Module>__proc, <Module>__global, module-qualified pools and <Module>__<Class>__vt. M2 -I resolves interfaces transitively; M2make --separate rebuilds stale objects and relinks on source/ interface/compiler/shim changes. The one-shot mode remains the default for the suite/corpus. Type/value/procedure and generic-class instances, definition-only class vtables and per-object self-hosting are covered.
- **OO** is the Clarion-form CLASS dialect (single inheritance, virtual dispatch), not ISO 10514-3.

Sources

- V3: docs/STATUS.md, docs/features.md, docs/gaps/README.md, docs/analysis_gaps.md, docs/summary_session-2026-10-24-separate-compilation.md, docs/separate-compilation.md; the gates in this document were run at v3-separate-compilation.
- gm2: the GCC Modula-2 testsuite and gm2-libs/gm2-libs-iso definition modules in the sibling gcc-git tree.
- M2R10: the published R10 grammar (m2r10-grammar.gll) and “*Differences between Modula-2 R10 and classic Modula-2*” (Kowarsch/Sutcliffe, 2012).
- TopSpeed V3: compiler/src/TopSpeed-V3-M2.atg (reconstructed V3 grammar) and docs/00P.txt (Clarion class model).
- PIM4: N. Wirth, *Programming in Modula-2*, 4th edition.