Debugging Industrial Switch DML

The DML language and compiler is described in detail in the DML Reference Manual. See also the Device Modeling Guidelines document for a more high-level perspective on how to write device models.

This manual discusses programming navigational access to a CA IDMS database, programming CA IDMS applications in COBOL, PL/I, and Assembler, and testing and debugging.

dml-lang Device Modeling Language Github page DML 1.4 Reference Manual Chat on Zulip

This topic describes how you can customize your debug output using DML. For general information on enabling and using DML in the debuggers, see Using Debugger Markup Language.

Debugger commands can provide output in plain text or in an enhanced format that uses Debugger Markup Language (DML). Output that is enhanced with DML includes links.

The DML language is documented in the Simics simulator documentation, as well as on the github page. The Simics simulator installation contains a ready-built DML compiler, but you can ...

DML provides high-level abstractions suitable for functional device models, including constructs like register banks, registers, bit fields, event posting, interfaces between models, and logging.

This reference card covers the basics of Simics usage and programming in DML 1.4 language, using Simics script, and C/Python API. There are 2 equivalent visual representations available for download:

DML code is compiled by the DML Compiler (DMLC), producing C code with API calls tailored for a particular simulator. Currently, the compiler supports building models for the Intel® Simics® ...

The Simics Modeling Extension supports debugging Python and DML/C/C++ simultaneously. This can be very helpful to modelers who are writing their models in DML and the unit tests for the models in ...

Web: https://www.safireschools.co.za