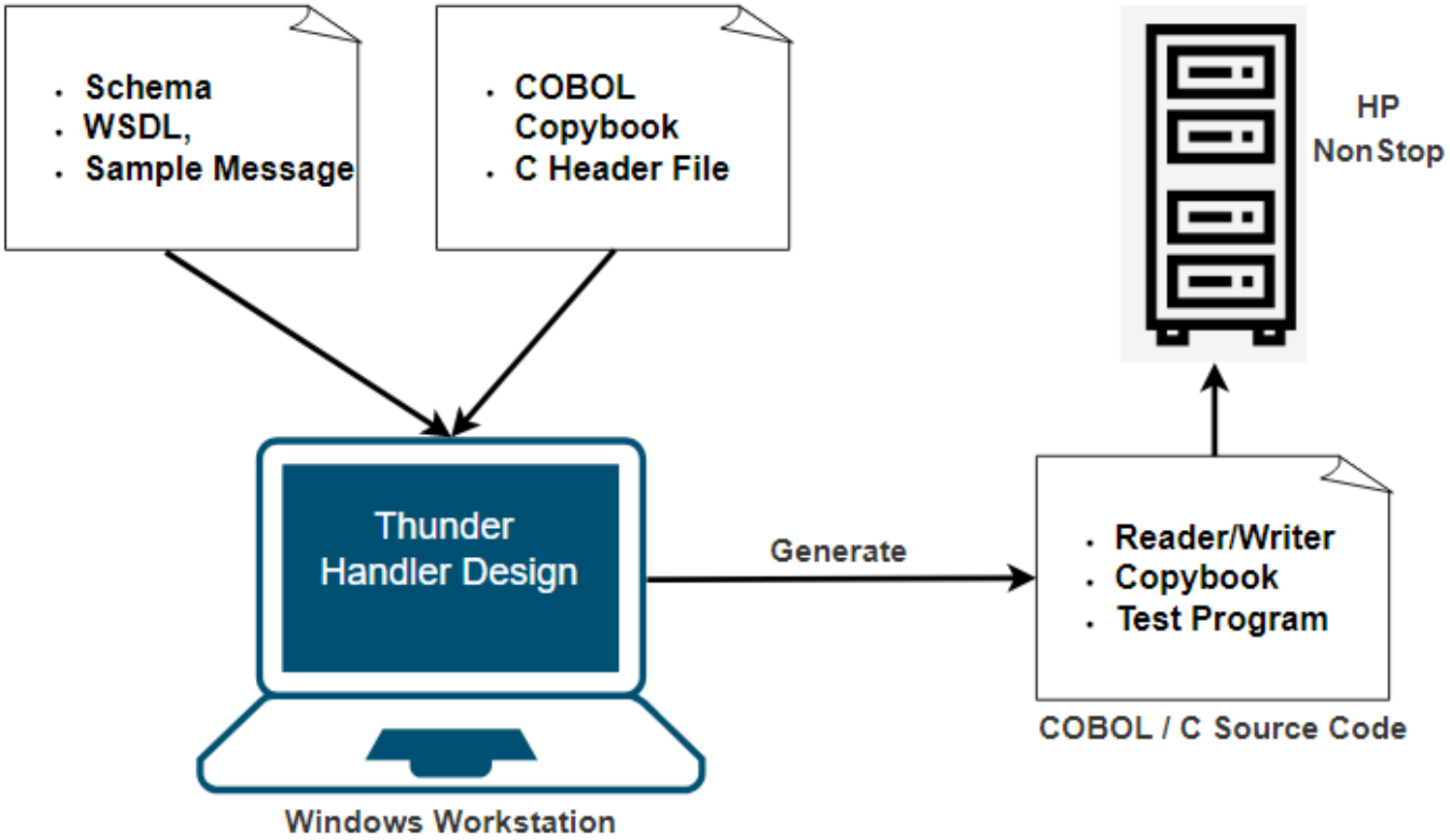
TICSOFTWARE.COM

Canam Software

XML Thunder Product Overview

Using the Thunder Toolset

Developer Workflow



XML Thunder enables COBOL, C and TAL programs to read and write XML documents so they can exchange data in XML format with other applications and companies.

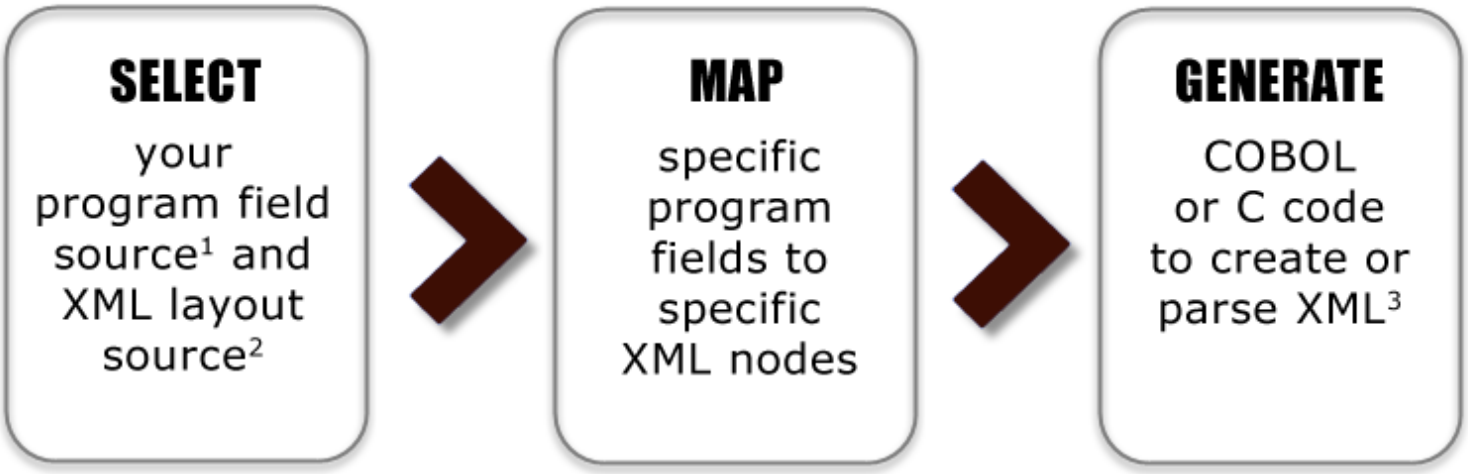
Read and Write XML Documents in COBOL, C and TAL

XML Thunder generates all of the COBOL or C source code needed to:

- read XML documents into COBOL, C and TAL structures.
- create XML documents from COBOL, C and TAL structures.
- validate XML documents to ensure only well-formed and valid documents are processed.

The code is generated as a program that can be called from existing COBOL, C and TAL programs.A sample COBOL or C calling program is provided to assist with developing new programs.

How it Works



[1]Program field source can be COBOL copybook, C header file or derived from the XML layout.
[2]XML source can be XSD, WSDL, sample XML message or derived from program data fields.
[3]The final output is a customized, callable program that can be used in existing applications. If you don't have an existing application, Thunder also provides a sample main (or calling) program that can serve as the starting point to developing it.

Model-Driven Development

XML Thunder uses a model-driven development approach enabling a faster time to market and the ability to respond to changes quickly.The Thunder toolset provides:

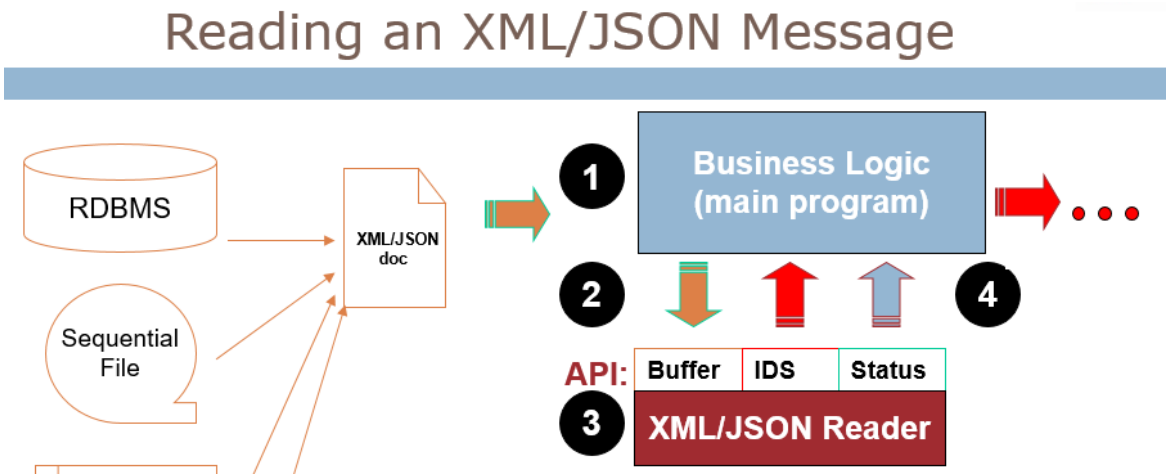
- A Visual Designer showing the program field layout, the XML message layout, and the mapping between the two.
- A full life cycle solution that is used in both development and maintenance.
- Automated source code generation.

Features

- High performance (not a generic parser!).
- Addresses the complexities of XML (e.g., choice, sequence, abstract types, etc.).
- Model-driven approach reducing development time and enabling developers to respond to changes quickly.
- Create Handler Designs from XML schemas, WSDLs, sample XML messages and COBOL/C data structures.
- Architecture and platform independent solution.
- Process messages of any size.
- Easy integration with existing programs

How the Generated Code is Used at Runtime.

Reading an XML Message



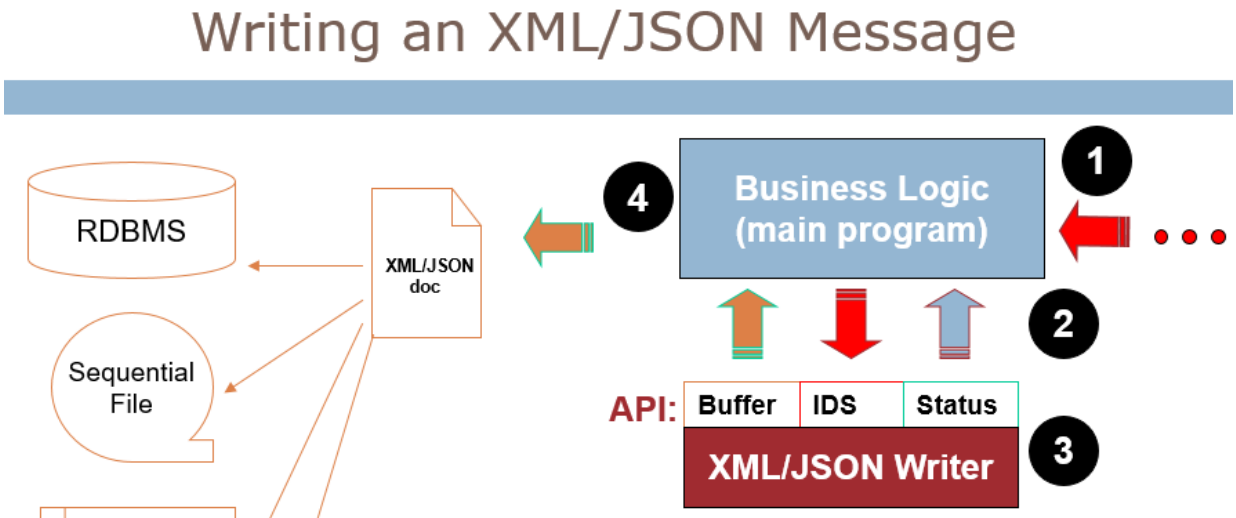
A. Business logic

1. Gets XML/JSON from any source desired
2. Calls Reader and passes XML/JSON instance via the buffer variable

B. XML/JSON Reader

3. Validates the XML/JSON and parses it into program fields (called the IDS) based on design time specification
4. Returns control and the populated IDS to the business logic for further processing

Writing an XML Message



A. Business logic

1. Gets data that will be used for the XML/JSON content
2. Passes data to the Writer via program fields (IDS)

B. XML/JSON Writer

3. Assembles a valid XML/JSON instance using the data received via the IDS
4. Returns XML/JSON instance in the buffer along with status information for further processing