Application of Extensible Markup Language to Data Acquisition

By Kristen Mark
August 2004

Dr. John Hoover
NRL-Chemistry Division

George Washington University’s SEAP Program
Problem

- Dependence on computers to acquire valuable data
- Files often unintelligible, creating a need for a standard format for data exchange
- Extensible Markup Language (XML) is a flexible standard, but can be time-consuming when programming in LabVIEW
LabVIEW

• Created for the demand of an easy-to-use programming environment
• Uses G (Graphical) programming language to create Virtual Instruments (VIs)
• Meets the needs, but can still be time-consuming to program for each experiment
LabVIEW Example
XML

• Used as a standard exchange
• Flexible, has a few, strict rules:
  – All XML documents must have a root element
  – All elements must be properly nested
  – All opening tags must have a corresponding end tag (these tags are case sensitive)
  – All attributes must be enclosed in quotes.
Original Water Mist Controller

- Before:
  - Data converted to strings to be concatenated
  - No labels
Modified Water Mist Controller

- After:
  - Data converted to string, given tags then concatenated
  - Standard format made information more intelligible
Output Data Flow Diagram
Generalized Output VI

• Needs to be able to write all data types
• Be able to use lower-level tags
• Use proper XML format
Output VI Flow Diagram
Generalized Input VI

• Needs to be able to find a lower level tag
• Needs to be able to access all the information between tags
Input VI Flow Diagram
Summary

- Modified the Water Mist Controller to add XML input/output
- Developed generic input/output VIs for XML
- Generic VIs can provide XML input and output for future LabVIEW applications