

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

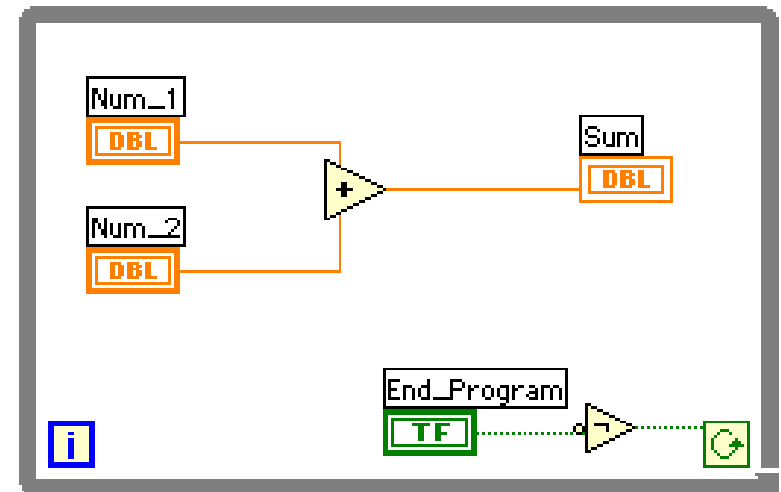
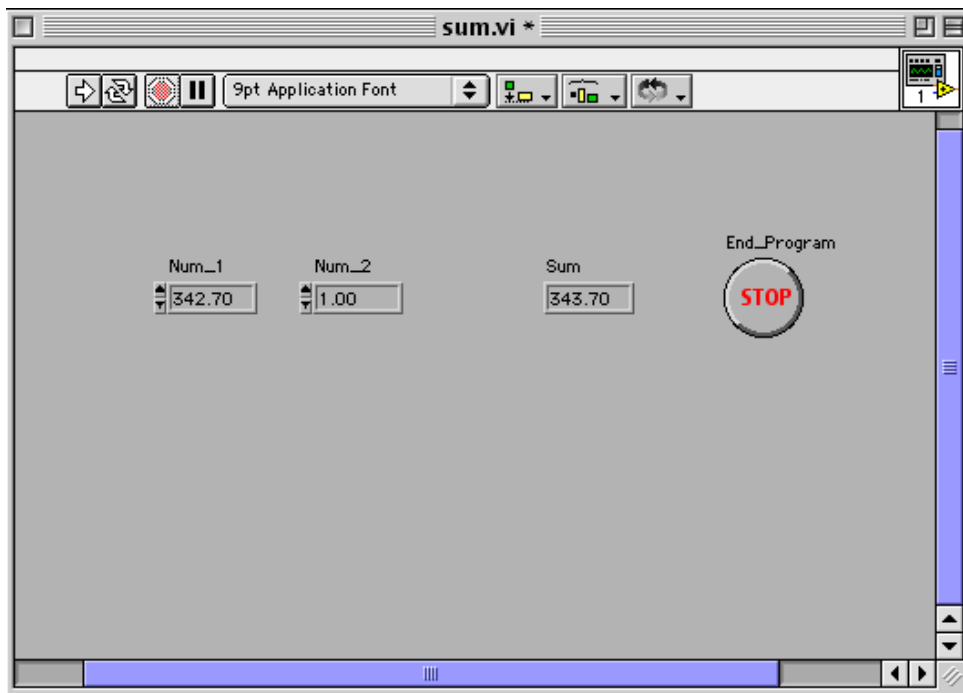
Problem

- 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

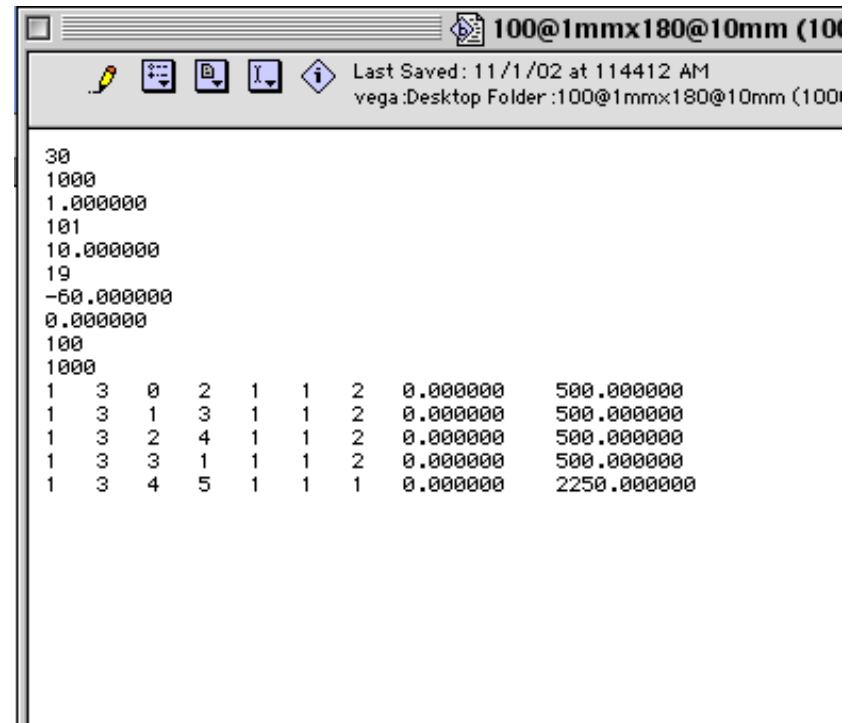


XML

- 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

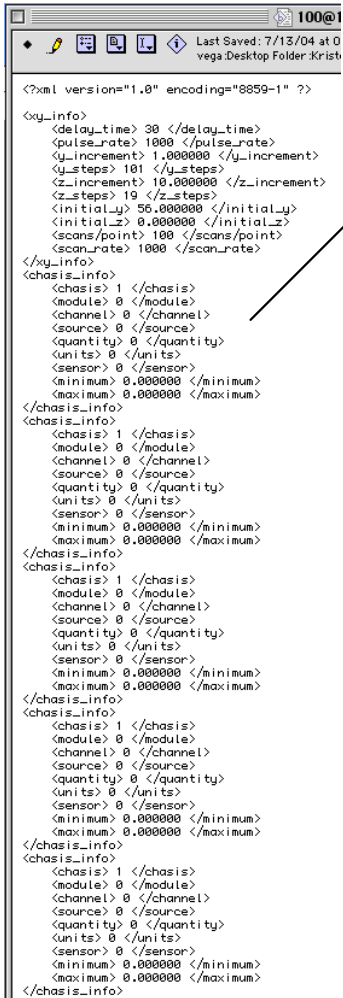
- Before:
 - Data converted to strings to be concatenated
 - No labels



The screenshot shows a text editor window with the following content:

```
30
1000
1.000000
101
10.000000
19
-50.000000
0.000000
100
1000
1 3 0 2 1 1 2 0.000000 500.000000
1 3 1 3 1 1 2 0.000000 500.000000
1 3 2 4 1 1 2 0.000000 500.000000
1 3 3 1 1 1 2 0.000000 500.000000
1 3 4 5 1 1 1 0.000000 2250.000000
```

Modified Water Mist Controller

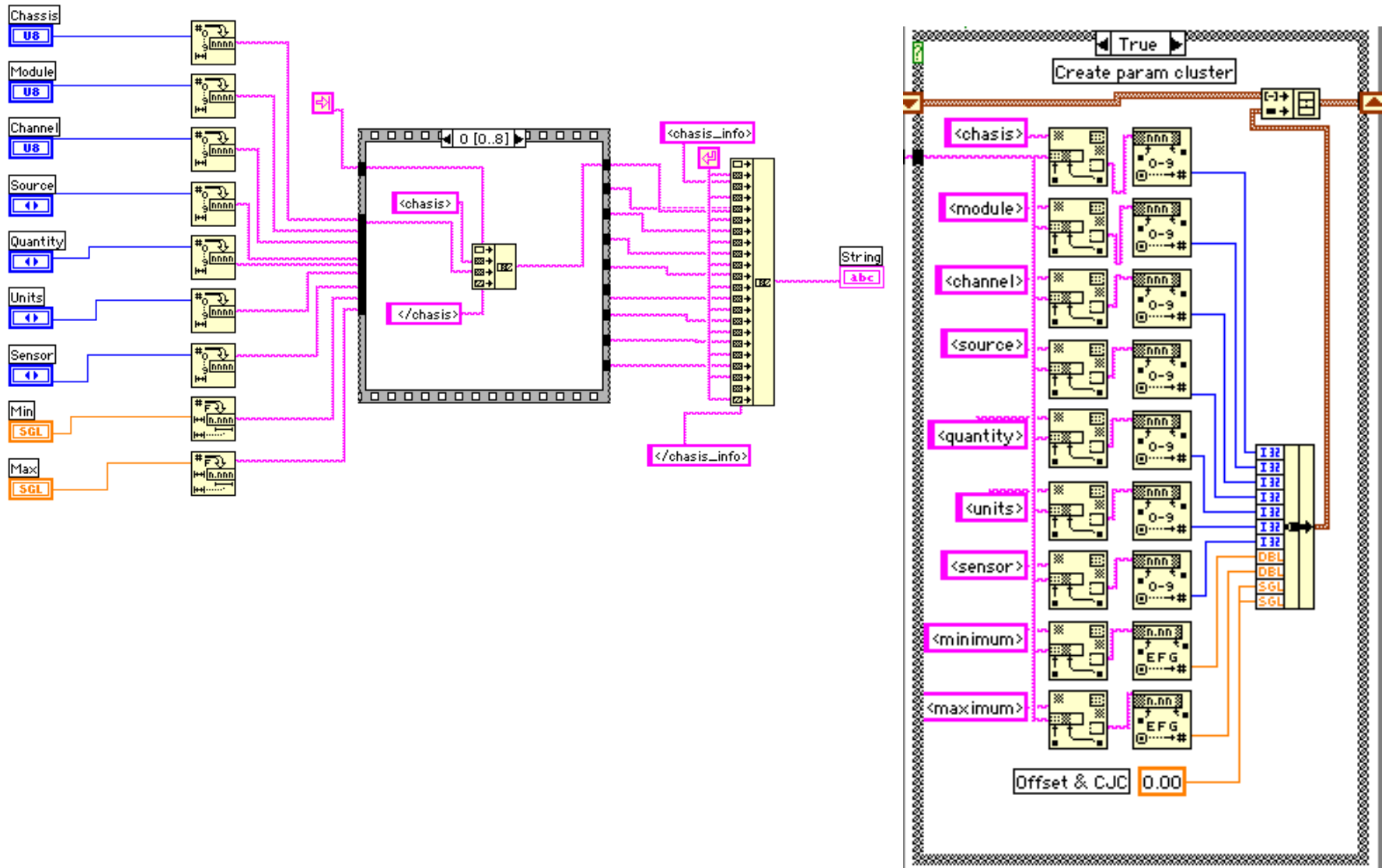


```
<?xml version="1.0" encoding="8859-1" ?>
<xy_info>
  <delay_time> 30 </delay_time>
  <pulse_rate> 1000 </pulse_rate>
  <y_increment> 1.000000 </y_increment>
  <y_steps> 101 </y_steps>
  <z_increment> 10.000000 </z_increment>
  <z_steps> 19 </z_steps>
  <initial_y> 56.000000 </initial_y>
  <initial_z> 0.000000 </initial_z>
  <scans/point> 100 </scans/point>
  <scan_rate> 1000 </scan_rate>
</xy_info>
<chasis_info>
  <chasis> 1 </chasis>
  <module> 0 </module>
  <channel> 0 </channel>
  <source> 0 </source>
  <quantity> 0 </quantity>
  <units> 0 </units>
  <sensor> 0 </sensor>
  <minimum> 0.000000 </minimum>
  <maximum> 0.000000 </maximum>
</chasis_info>
<chasis_info>
  <chasis> 1 </chasis>
  <module> 0 </module>
  <channel> 0 </channel>
  <source> 0 </source>
  <quantity> 0 </quantity>
  <units> 0 </units>
  <sensor> 0 </sensor>
  <minimum> 0.000000 </minimum>
  <maximum> 0.000000 </maximum>
</chasis_info>
<chasis_info>
  <chasis> 1 </chasis>
  <module> 0 </module>
  <channel> 0 </channel>
  <source> 0 </source>
  <quantity> 0 </quantity>
  <units> 0 </units>
  <sensor> 0 </sensor>
  <minimum> 0.000000 </minimum>
  <maximum> 0.000000 </maximum>
</chasis_info>
<chasis_info>
  <chasis> 1 </chasis>
  <module> 0 </module>
  <channel> 0 </channel>
  <source> 0 </source>
  <quantity> 0 </quantity>
  <units> 0 </units>
  <sensor> 0 </sensor>
  <minimum> 0.000000 </minimum>
  <maximum> 0.000000 </maximum>
</chasis_info>
```

```
<?xml version="1.0" encoding="UTF-8" ?>
<xy_info>
  <delay_time> 30 </delay_time>
  <pulse_rate> 1000 </pulse_rate>
  <y_increment> 1.000000 </y_increment>
  <y_steps> 101 </y_steps>
  <z_increment> 10.000000 </z_increment>
  <z_steps> 19 </z_steps>
  <initial_y> 56.000000 </initial_y>
  <initial_z> 0.000000 </initial_z>
  <scans/point> 100 </scans/point>
  <scan_rate> 1000 </scan_rate>
</xy_info>
<chasis_info>
  <chasis> 1 </chasis>
  <module> 0 </module>
  <channel> 0 </channel>
  <source> 0 </source>
  <quantity> 0 </quantity>
  <units> 0 </units>
  <sensor> 0 </sensor>
  <minimum> 0.000000 </minimum>
  <maximum> 0.000000 </maximum>
</chasis_info>
<chasis_info>
  <chasis> 1 </chasis>
  <module> 0 </module>
  <channel> 0 </channel>
  <source> 0 </source>
  <quantity> 0 </quantity>
  <units> 0 </units>
  <sensor> 0 </sensor>
  <minimum> 0.000000 </minimum>
  <maximum> 0.000000 </maximum>
</chasis_info>
<chasis_info>
  <chasis> 1 </chasis>
  <module> 0 </module>
  <channel> 0 </channel>
  <source> 0 </source>
  <quantity> 0 </quantity>
  <units> 0 </units>
  <sensor> 0 </sensor>
  <minimum> 0.000000 </minimum>
  <maximum> 0.000000 </maximum>
</chasis_info>
```

- After:
 - Data converted to string, given tags then concatenated
 - Standard format made information more intelligible

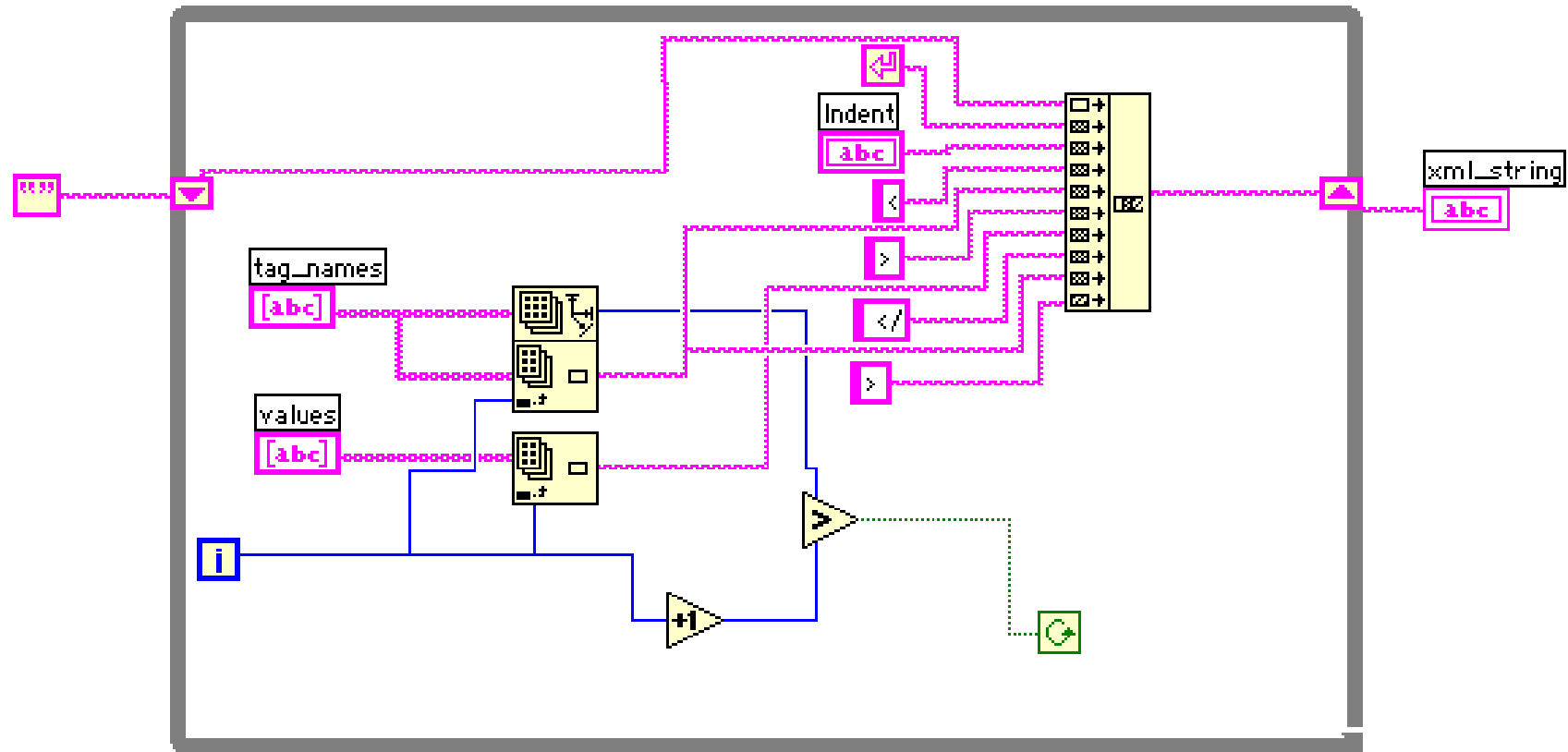
Output Data Flow Diagram



Generalized Output VI

- Needs to be able to write all data types
- Be able to use lower-level tags
- Use proper XML format

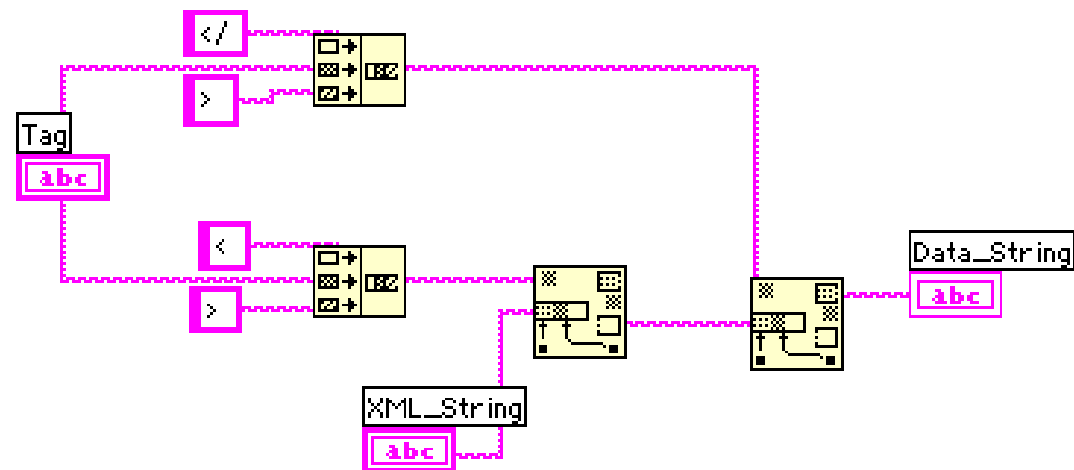
Output VI Flow Diagram



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