MultiSpeak Version 3.0 Interoperability Assertion

Statement of interoperable functionality between:

<table>
<thead>
<tr>
<th>Vendor(s)</th>
<th>Product</th>
<th>Product Version</th>
<th>Role</th>
<th>Batch Interface</th>
<th>Web Client Interfaces</th>
<th>Web Server Interfaces</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open Systems International, Inc. (OSI)</td>
<td>monarch™</td>
<td>34.002</td>
<td>SCADA</td>
<td></td>
<td></td>
<td>SCADA→EA</td>
</tr>
<tr>
<td>Milsoft Utility Solutions</td>
<td>WindMil</td>
<td>7.3</td>
<td>EA</td>
<td></td>
<td></td>
<td>SCADA→EA</td>
</tr>
</tbody>
</table>

Summary:

monarch™ is capable of providing analog and status point data to Milsoft’s WindMil® using MultiSpeak® web services.

WindMil is able to request a list of SCADA points from monarch. The WindMil user can then map each SCADA point to the appropriate corresponding circuit element in WindMil. This is necessary in order to ensure that any SCADA data WindMil receives from monarch is associated with the correct circuit elements in the electrical model.

The interesting data types that WindMil can request from monarch are: breaker operations (known as SCADA status) and substation voltages and currents (known as SCADA analogs). The WindMil user may request SCADA statuses or SCADA analogs from monarch for a particular circuit element. monarch sends the requested information to WindMil to be immediately available to the WindMil user for viewing, (in the case of fault current data) to run the Fault Locator utility, or (in the case of feeder current data) to run a Load Allocation study.

Prerequisites:

The monarch server must be accessible from WindMil and configured for communication. Additionally, the SCADA points must be associated with circuit model elements within WindMil. In order for the Fault Locator Utility to operate, fault currents must be supported by field devices and supplied by monarch as SCADA points. In order for Load Allocation studies to utilize SCADA data, feeder currents must be supplied as SCADA points.

For WindMil to access SCADA status and analog points hosted by monarch, they must first be added to monarch’s list of published status and analog points. The following configuration is available for each of the hosted SCADA points:

- Published Name (optional for status and analog points; defaults to monarch’s SCADA key)
- MultiSpeak Units (required for analogs)
- Status State Calculations (optional)
- Analog Deadbands (optional)
- Analog Scale Factors (optional)
Additional configuration options are available for translating monarch tags, limits, and qualities to the MultiSpeak OutOfService state and MultiSpeak qualities.

Specific Vendor Assertions:

1) WindMil can request a list of SCADA points from monarch.

_Importance to user:_ The SCADA points in the list can be mapped to WindMil circuit elements, allowing the user to specify the mapping between SCADA point ID’s and devices in the WindMil model.

_How Achieved:_ The WindMil user chooses “Get SCADA point list” from the real-time menu. WindMil then calls the GetAllSCADAPoints method on the monarch server. The resulting list of SCADA points is displayed to the WindMil user in a list. The SCADA point ID’s in this list can be dragged into the empty fields on the SCADA page of the circuit element editor for the appropriate WindMil circuit elements. This process creates the mapping between SCADA point ID’s and WindMil circuit elements.

2) WindMil can request SCADA analog values from monarch.

_Importance to user:_ The SCADA analog values such as fault currents and feeder currents can be used by the engineer for such applications as fault location and load allocation.

_How Achieved:_ After SCADA point ID’s have been assigned to their appropriate WindMil elements, the WindMil user can click the “Read SCADA analogs” or the “Read SCADA fault currents” button on the SCADA page of the circuit element editor of the device he/she wishes to read. WindMil then calls the GetSCADAAnalogBySCADAPointId method on the monarch server. The returned SCADA analog values are displayed in the circuit element editor.

3) WindMil can request SCADA status values from monarch.

_Importance to user:_ The status of a SCADA point tells the engineer whether a particular device is currently open or closed.

_How Achieved:_ After SCADA point ID’s have been assigned to their appropriate WindMil elements, the WindMil user can click the “Read SCADA status” button on the SCADA page of the circuit element editor of the device he wishes to read. WindMil then calls the GetSCADASTatusBySCADAPointId method on the monarch server. The returned SCADA statuses (possibly more than one if multiple phases are desired) are displayed in the circuit element editor.
### Summary of Interoperability Test Results

**Interface #8  SCADA → EA**

#### Table 1
Recommended MultiSpeak Methods

<table>
<thead>
<tr>
<th>Method Name</th>
<th>Importance to User</th>
<th>Supported by Server&lt;sup&gt;1&lt;/sup&gt; (SCADA)</th>
<th>Supported by Client&lt;sup&gt;2&lt;/sup&gt; (EA)</th>
<th>Verified Interoperable&lt;sup&gt;3&lt;/sup&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetMethods</td>
<td>Requests a list of methods supported by the server.</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>GetAllSCADAPoints</td>
<td>Returns a list of SCADA Point definitions.</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>GetPublishMethods</td>
<td>Requests list of methods to which this server can publish information.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GetRegistrationInfoByID</td>
<td>Requests the return of existing registration information (that is to say the details of what is subscribed on this subscription) for a specific registrationID.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GetSCADAAnalogBySCADAPointID</td>
<td>Returns a specific SCADA Analog by SCADAPointID.</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>GetSCADASTatusBySCADAPointID</td>
<td>Returns a specific SCADA Status by SCADAPointID.</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>PingURL</td>
<td>Verifies that the server is running and reachable.</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>RegisterForService</td>
<td>Establishes a subscription using a previously requested registrationID.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RequestRegistrationID</td>
<td>Requests of the publisher a unique registration ID that would subsequently be used to refer unambiguously to that specific subscription.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UnregisterForService</td>
<td>Deletes a previously established subscription (registration for service) that carries the registration identifier listed in the input parameter registrationID.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1) Supported by Server means that the server has demonstrated in some interoperability test (not necessarily with this client) that it can support the method.

2) Supported by Client means that the client has demonstrated in some interoperability test (not necessarily with this server) that it can call the method.

3) Verified Interoperable means that both the client and server have demonstrated in this interoperability test that they can usefully transfer data using this method.
Table 2
Optional MultiSpeak Methods

<table>
<thead>
<tr>
<th>Method Name</th>
<th>Importance to User</th>
<th>Supported by Server&lt;sup&gt;1&lt;/sup&gt; (SCADA)</th>
<th>Supported by Client&lt;sup&gt;2&lt;/sup&gt; (EA)</th>
<th>Verified Interoperable&lt;sup&gt;3&lt;/sup&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetAllSCADAAnalogs</td>
<td>Returns all SCADA Analogs.</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GetAllSCADAStatus</td>
<td>Returns all SCADA Status data.</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GetDomainMembers</td>
<td>Returns a list of members of a specific domain supported by the server.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GetDomainNames</td>
<td>Returns a list of domain names supported by the server.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GetModifiedSCADAPoints</td>
<td>Returns a list of SCADA Point definitions that have changed since the session identified by sessionID.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1) Supported by Server means that the server has demonstrated in some interoperability test (not necessarily with this client) that it can support the method.
2) Supported by Client means that the client has demonstrated in some interoperability test (not necessarily with this server) that it can call the method.
3) Verified Interoperable means that both the client and server have demonstrated in this interoperability test that they can usefully transfer data using this method.
Certified by:

For Open Systems International, Inc. (OSI):

__________________________ Vice President of Engineering
Name: Brian Rasfske
Title

5/7/2010
Date: ________________

For Milsoft Utility Solutions, Inc.

__________________________ Executive Vice President/CTO
Name: Luis R. Malavé
Title

5/7/2010
Date: ________________

Assertions Verified by:

____________________________ MultiSpeak Testing Agent
Name: ____________________
Title

UISOL, Inc.
Testing Agent
5/7/2010
Date: ________________

Disclaimer:

The assertions made in this document are statements of the vendors offering the two products listed above. The Testing Agent has observed the software performing the tasks described in these vendor assertions.
Neither NRECA, Cornice Engineering, Inc. (MultiSpeak Project Coordinator), nor __________, acting on behalf of NRECA, makes any warranty or guarantee that the software will perform as described in this assertion when installed at any specific utility. Furthermore, neither NRECA, Cornice Engineering, Inc., nor __________ makes any warranty or guarantee that the software described will be suitable for any specific purpose or need.

As used herein, the word verify shall mean an expression of the Testing Agent’s professional opinion to the best of its information, knowledge and belief, and does not constitute a warranty or guarantee by NRECA, Cornice Engineering, Inc., or the Testing Agent.