

MultiSpeak Version 3.0 Interoperability Assertion

Statement of Interoperable Functionality Between:

Vendor(s)	Product	Product Version	Role	Batch Interface	Web Client Interfaces	Web Server Interfaces
Survalent Technology Corporation	Survalent Windows SCADA	3.0	SCADA	-	OA→SCADA	SCADA→OA
Milsoft Utility Solutions	Milsoft Web Server	7.1	OA	-	SCADA→OA	OA→SCADA
Milsoft Utility Solutions	DisSPatch	7.1	OA	-	SCADA→OA	-

Summary:

Survalent SCADA is capable of publishing analog and status point data to Milsoft's DisSPatch Outage Management System using MultiSpeak web services.

The breaker operations reported by SCADA in this manner can be used by DisSPatch to automatically create verified SCADA outage/restored records for such events.

Declared outages for locations that are not monitored by SCADA can be requested from DisSPatch by SCADA. In this case, SCADA maps the device ID information to SCADA non-telemetered points and updates the SCADA points.

Milsoft's DisSPatch Outage Management System is an object oriented derived product from Milsoft's Engineering Circuit Model Engine. DisSPatch uses the same Engineering distribution model as WindMil; therefore DisSPatch is capable of functioning as an EA system and supports all the functions available in the EA interface without the use of Milsoft's WindMil product. DisSPatch is capable of requesting SCADA analogs and SCADA status from pre-mapped SCADA identified points. DisSPatch can request for all SCADA points and map these points to SCADA devices on the circuit model.

Prerequisites:

The SCADA points that are to be made available to DisSPatch are identified in the SCADA system's Data Exchange tables. In these tables, SCADA points can be assigned:

- alternate names for presentation to the client (optional for both analog and status points)
- unit codes (required for analog points)
- deadbands (optional for analog points)
- scale factors (optional for analog points)

Survalent SCADA does not require that DisSPatch post subscriptions for the desired points. By means of Autosubscribe checkboxes in the Data Exchange tables, the user can identify which points are to be published on establishment of a connection by DisSPatch (as detected by detection of PingURL requests). Frequency of updates is directly controlled by user-settable parameters in the Data Exchange table, and for analog points, indirectly by means of deadbands specific to this interface (as opposed to deadbands at the source such as the RTU).

Points that are not subscribed are not published, but can be read on demand via the GetSCADAAnalogBySCADAPointID and GetSCADAStatusBySCADAPointID methods.

Specific Vendor Assertions:

- 1) **Breaker operations reported by SCADA can be used by DisSPatch to automatically create verified SCADA outage/restored records.**

Importance to user: The time elapsed between the actual event and the generation of an outage case record is greatly reduced. Human error is also reduced.

How Achieved: When the SCADA system detects changes in the points specified in the Data Exchange table, it calls the SCADAStatusChangedNotification method in the OA->SCADA interface to publish the changed points to DisSPatch.

- 2) **DisSPatch can request a list of SCADA points from the Survalent SCADA server.**

Importance to user: The SCADA points in the list can be mapped to DisSPatch circuit elements, allowing the user to specify which SCADA points should be associated with which devices in the circuit model.

How Achieved: The DisSPatch user chooses “Get Real-time SCADA Points” from the SCADA menu. DisSPatch then calls the GetAllSCADAPoints method on the Survalent server. The resulting list of SCADA points is displayed to the DisSPatch user in a list. The SCADA point ids in this list can be dragged into the empty fields on the SCADA page of the Circuit Element Editor for the appropriate DisSPatch circuit elements. This process creates the mapping between SCADA point IDs and DisSPatch circuit elements.

- 3) **DisSPatch can request SCADA analog values from the Survalent SCADA server.**

Importance to user: The SCADA analog values such as fault currents and feeder currents can be used by the dispatcher to locate the fault.

How Achieved: After SCADA point IDs have been assigned to their appropriate DisSPatch elements, the DisSPatch 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 they wish to read. DisSPatch then calls the GetSCADAAnalogBySCADAPointId method on the Survalent SCADA server. The returned SCADA analog values are displayed in the Circuit Element Editor. If fault current analog data is available from SCADA, then a fault current analysis study can be performed using these values in order to locate the suspected fault.

- 4) **DisSPatch can request SCADA status values from the Survalent SCADA server.**

Importance to user: The status of a SCADA point tells the dispatcher whether some particular device is currently open or closed.

How Achieved: After SCADA point IDs have been assigned to their appropriate DisSPatch circuit elements, the DisSPatch user can click the “Read SCADA Status” button on the SCADA page of the Circuit Element Editor of the device they wish to read. DisSPatch then calls the GetSCADAStatusBySCADAPointId method on the Survalent SCADA 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 #9 SCADA→OA

**Table 1
Recommended MultiSpeak Methods**

Method Name	Importance to User	Supported by Server ¹ (SCADA)	Supported by Client ² (OA)	Verified Inter-operable ³
GetMethods	Requests a list of methods supported by the server.	X	X	X
PingURL	Verifies that the server is running and reachable.	X	X	X
GetAllSCADAPoints	Returns a list of SCADA Point definitions.	X	X	X
GetSCADAAnalogBySCADAPointID	Returns a specific SCADA Analog by SCADAPointID.	X	X	X
GetSCADAStatusBySCADAPointID	Returns a specific SCADA Status by SCADAPointID.	X	X	X

**Table 2
Optional MultiSpeak Methods**

Method Name	Importance to User	Supported by Server ¹ (SCADA)	Supported by Client ² (OA)	Verified Inter-operable ³
GetAllSCADAAnalog	Returns all SCADA Analogs.			
GetAllSCADAStatus	Returns all SCADA Status data.			
GetDomainMembers	Returns a list of members of a specific domain supported by the server.			
GetDomainNames	Returns a list of domain names supported by the server.			
GetModifiedSCADAPoints	Returns a list of SCADA Point definitions that have changed since the session identified by sessionID.			
OutageEventChangedNotification	OA Notifies SCADA of a change in OutageEvent by sending an array of changed OutageEvent objects.	X		

- 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.

**Summary of Interoperability Test Results
Interface #9 OA→SCADA**

**Table 3
Recommended MultiSpeak Methods**

Method Name	Importance to User	Supported by Server ¹ (OA)	Supported by Client ² (SCADA)	Verified Inter-operable ³
GetMethods	Requests a list of methods supported by the server.	X	X	X
PingURL	Verifies that the server is running and reachable.	X	X	X
SCADAPointChangedNotification	SCADA Notifies OA of changes in SCADA point definitions by sending an array of changed scadaPoint objects.			
SCADAStatusChangedNotification	SCADA Notifies OA of changes in point status by sending an array of changed scadaStatus objects.	X	X	X

- (i) In order for new or changed point definitions to be available to the server, the SCADA side of the interface must be told to reload its Data Exchange tables. This is accomplished by a user command. The SCADA system itself does not need to be reset.

**Table 4
Optional MultiSpeak Methods**

Method Name	Importance to User	Supported by Server ¹ (OA)	Supported by Client ² (SCADA)	Verified Inter-operable ³
GetDomainMembers	Returns a list of members of a specific domain supported by the server.			
GetDomainNames	Returns a list of domain names supported by the server.			
GetActiveOutages	Returns the outageEventIDs for all active outage events.	X		
GetOutageEventStatus	Returns the current status of an outage event, given the outage event ID.	X		
GetOutageEventStatusByOutageLocation	Returns the current status of an outage event, given the outage location.	X		
SCADAAnalogChangedNotification	SCADA Notifies OA of changes in analog values by sending an array of changed scadaAnalog objects.		X	
SCADAAnalogChangedNotificationByPointID	SCADA Notifies OA of changes in a specific analog value, chosen by scadaPointID, by sending a changed scadaAnalog object.			
SCADAAnalogChangedNotificationForPower	SCADA Notifies OA of changes in power analogs, by sending an array of changed scadaAnalog objects.			

SCADAAnalogChangedNotificationForVoltage	SCADA Notifies OA of changed voltage analogs, by sending an array of changed scadaAnalog objects.			
SCADAPointChangedNotificationForAnalog	SCADA Notifies OA of changes in SCADA point definitions, limited to Analog points, by sending an array of changed scadaPoint objects.			
SCADAPointChangedNotificationForStatus	SCADA Notifies OA of changes in SCADA point definitions, limited to Status points, by sending an array of changed scadaPoint objects.			
SCADAStatusChangedNotificationByPointID	SCADA Notifies OA of changes in the status of a specific point, chosen by PointID, by sending a changed scadaStatus object.			

- 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 Survalent Technology Corporation:



VP Software Development

Name: Cornel Munteanu

Date: July 18, 2006

For Milsoft Utility Solutions, Inc.:



Executive Vice President/CTO

Name: Luis Malave

Date: July 18, 2006

Assertions Verified by:



Name: Hannu Huhdanpaa

UISOL, Inc.

Testing Agent

Date: July 18, 2006

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 UISOL, 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 UISOL makes any warranty or guarantee that the software described will be suitable for any specific purpose or need.

June 11, 2006

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.