MultiSpeak Version 3.0 Interoperability Assertion – Outage Analysis (OA) / Outage Detection (OD)

Vendor: Milsoft Utility Solutions Milsoft Web Server and OpMoSys Mobile Alerts

Interfaces: Outage Analysis (OA), Outage Detection (OD)

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
OpMoSys	Mobile Alerts	2.0	OD		OD→OA	
Milsoft Utility Solutions	Milsoft Web Server	8.1	OA			OA→OD

Summary:

OpMoSys Mobile Alerts is capable of sending Milsoft outages detection events as reported by SMS subscribers. OpMoSys Mobile Alerts can also use the interface to determine SMS subscriber's current outage status.

Prerequisites:

For this interface to be useful, the OpMoSys system must be able to identify the SMS subscriber's meter number/numbers that correspond to the service location in the Milsoft model. This is used for querying outage status as well as reporting outages. Additionally, Milsoft Web Server must be accessible via the internet (https) from the machine on which OpMoSys Mobile Alerts is running.

Specific Vendor Assertions:

1) OpMosys SMS Subscribers can report outages via text to Milsoft OMS

Importance to user: SMS subscribers can use their mobile phone to report an outage of power to their utility company using text messaging.

How Achieved: A utility customer with a mobile device that supports SMS must be registered with the OpMoSys Mobile Alerts system. The registration process will associate the mobile device with the outage location(s) and or meter number(s) for the utility customer. This data will be stored within OpMoSys systems. When the utility customer has an outage, the utility customer will send SMS to a short code that is provided. OpMoSys Mobile Alerts will then use the ODEventNotification method using the outage location(s) and or meter number(s) that were previously associated with the mobile device sending the SMS.

2) OpMoSys will exchange outage status information on specific customers.

Importance to user: When an SMS subscriber has a meter number involved in an outage, OpMoSys can send a text to the subscriber with the outage details.

How Achieved: OpMoSys will call the GetActiveOutages method which will return a list of outage objectID's that are currently active on the OMS system. For each outage object ID OpMoSys will call GetOutageEventStatus to get the current status of the outage. For each status that has changed since the last GetOutageEventStatus call, OpMoSys will call GetCustomersAffectedByOutage to get the list of affected meter numbers. For each meter number returned that is associated with an SMS subscriber OpMoSys will send a text to the SMS subscriber with the outage status information.

Products: Milsoft Web Server and OpMoSys Mobile Alerts

Summary of Interoperability Test Results
Interface #5
OA → OD

Table 1
Recommended MultiSpeak Methods

Method Name	Importance to User	Supported by Server[1] (OA)	Supported by Client[2] (OD)	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
GetActiveOutages	Returns the outage Event IDs for all active outage events.	X	X	X
GetCustomersAffectedByOutage	Returns all customers that are affected by a specfic outage of interest, given the outageEventID. The outageEventID is the objectID for the outageEvent object.	X	X	X
GetOutageEventStatus	Returns the current status of an outage event, given the outage event ID.	X	X	X
ODEventNotification	Notifies a change in outage detection events	X	X	X

Table 2
Optional MultiSpeak Methods

Method Name	Importance to User	Supported by Server[1] (OA)	Supported by Client[2] (OD)	Verified Inter-operable[3]
AVLChangedNotification	Publisher notifies subscriber of new AVL events by sending an AVLMessage object. DGV returns information on failed transactions by returrning an array of errorObjects. The message header attribute 'registrationID' should be added to all publish messages to indicate to the subscriber under which registrationID they received this notification data.	X		
AddRemarkToOutage	Allows a system operator to add a remark to an outage event. OA returns information on failed transactions by returrning an array of errorObjects.			
AssignCrewsToOutage	Assigns crews to an outage given the outage event ID. OA returns information on failed transactions by returrning an array of errorObjects.	X		
AssessmentLocationChangedNotification	Publisher notifies Subscriber of new assessmentLocation(s). The transactionID calling parameter can be used to link this action with an asynchronous request, if any. If this transaction fails, Subscriber returns information about the failure in an array of errorObject(s). The message header attribute 'registrationID' should be added to all publish messages to indicate to the subscriber under which registrationID they received this notification data.	X		
GetAllCircuitElements	Returns all circuit elements.	X		
CDStateChangedNotification	Publisher notifies subscriber of state change for a connect/disconnect device By meterNumber and loadActionCode. The transactionID calling parameter can be			

	used to link this action with an InitiateConectDisconnect call. If this transaction fails, CB returns information about the failure in a SOAPFault. The message header attribute 'registrationID' should be added to all publish messages to indicate to the subscriber under which registrationID they received this notification data.			
CDStateNotification	CD notifies CB of state of a connect/disconnect device. The transactionID calling parameter can be used to link this action with an InitiateCDStateRequest call. If this transaction fails, CB returns information about the failure in a SOAPFault. The message header attribute 'registrationID' should be added to all publish messages to indicate to the subscriber under which registrationID they received this notification data.			
CDStatesChangedNotification	CD notifies CB of state change(s) for connect/disconnect device(s). The transactionID calling parameter can be used to link this action with an InitiateConectDisconnect call. If this transaction fails, CB returns information about the failure in an array of errorObject(s). The message header attribute 'registrationID' should be added to all publish messages to indicate to the subscriber under which registrationID they received this notification data.			
CDStatesNotification	Publisher notifies Subscriber of state of connect/disconnect device(s). The transactionID calling parameter can be used to link this action with an InitiateCDStateRequest call. If this transaction fails, CB returns information about the failure in an array of errorObject(s). The message header attribute 'registrationID' should be added to all publish messages to indicate to the subscriber under which registrationID they received this notification data.			
CHEventNotification	Publisher notifies OA of non-outage events by sending the a customerCall object. The message header attribute 'registrationID' should be added to all publish messages to indicate to the subscriber under which registrationID they received this notification data.	X		
CloseCalls	Publisher notifies OA of a list of customer calls to close out. OA returns status of failed transactions in an array of errorObjects.	X		
ConnectDisconnectChangedNotification	Publisher notifies OA of changes in connect disconnect event object(s) by sending the changed connectDisconnectEvent object(s). OA returns information about failed transactions in an array of errorObjects. The message header attribute 'registrationID' should be added to all publish messages to indicate to the subscriber under which registrationID they received this notification data.	X		
CustomerChangedNotification	Publisher Notifies subscriber of a change in the Customer object by sending the changed customer object(s). Subscriber returns information about failed transactions using an array of errorObjects. The message header attribute 'registrationID' should be added to all publish messages to indicate to the subscriber under which registrationID they received this notification data.	X		
DiscardOutage	This method allows a dispatcher or operator to discard an outage that has been created erroneously or which was generated for training purposes. Subscriber can return	X		

	information about erros using an errorObject.			
GetChildCircuitElements	Returns circuit elements immediately fed by the given line section or node (eaLoc).	X		
DomainMembersChangedNotification	This method permits a client to have changed information on domain members published to it using a previously arranged subscription, set up using the RegisterForServiceMethod. The client should first obtain a registrationID and then register for service, including the DomainMembersChangedNotification as one of the methods in the list of methods to which the client has subscribed. The server shall include the registrationID for the subscription in the message header so that the client can determine the source of the domainMember information.			
DomainNamesChangedNotification	This method permits a client to have changed information on domain names published to it using a previously arranged subscription, set up using the RegisterForServiceMethod. The client should first obtain a registrationID and then register for service, including the DomainNamesChangedNotification as one of the methods in the list of methods to which the client has subscribed. The server shall include the registrationID for the subscription in the message header so that the client can determine the source of the domainName information.			
GetActiveAssessmentLocations	Returns the assessmentLocationIDs for all of the active locations for workers to assess storm damage. The calling parameter lastReceived is included so that large sets of data can be returned in manageable blocks. lastReceived should carry an empty string the first time in a session that this method is invoked. When multiple calls to this method are required to obtain all of the data, the lastReceived should carry the objectID of the last data instance received in subsequent calls.			
GetAllActiveCalls	Returns all active calls that have been processed by the outage management system in the form of an outageDetectionLogList. The calling parameter lastReceived is included so that large sets of data can be returned in manageable blocks. lastReceived should carry an empty string the first time in a session that this method is invoked. When multiple calls to this method are required to obtain all of the data, the lastReceived should carry the objectID of the last data instance received in subsequent calls. If the sessionID parameter is set in the message header, then the server shall respond as if it were being asked for a GetModifiedXXX since that sessionID; if the sessionID is not included in the method call, then all instances of the object shall be returned in response to the call.	X		
GetAllActiveOutageEvents	Returns all of the outageEvent(s) for all active outages. The calling parameter lastReceived is included so that large sets of data can be returned in manageable blocks. lastReceived should carry an empty string the first time in a session that this method is invoked. When multiple calls to this method are required to obtain all of the data, the lastReceived should carry the objectID of the last data instance received in subsequent calls. If the sessionID parameter is set in the message header, then the server shall respond as if it were being asked for a GetModifiedXXX since that sessionID; if the sessionID is not	X		

	included in the method call, then all instances of the object shall be returned in response to the call.			
GetAllCircuitElements	Returns all circuit elements. The calling parameter lastReceived is included so that large sets of data can be returned in manageable blocks. lastReceived should carry an empty string the first time in a session that this method is invoked. When multiple calls to this method are required to obtain all of the data, the lastReceived should carry the objectID of the last data instance received in subsequent calls.	X		
GetAllConnectivity	Returns all information for all elements in the connectivity model. The calling parameter lastReceived is included so that large sets of data can be returned in manageable blocks. lastReceived should carry an empty string first time in a session that this method is invoked. When multiple calls to this method are required to obtain all of the data, the lastReceived should carry in subsequent calls the index number provided by the server as being the lastSent.	X		
GetAllCrews	Returns all active crews that are available for dispatching if the parameter activeOnly is set to be true, otherwise all crews are returned. The calling parameter lastReceived is included so that large sets of data can be returned in manageable blocks. lastReceived should carry an empty string the first time in a session that this method is invoked. When multiple calls to this method are required to obtain all of the data, the lastReceived should carry the objectID of the last data instance received in subsequent calls. If the sessionID parameter is set in the message header, then the server shall respond as if it were being asked for a GetModifiedXXX since that sessionID; if the sessionID is not included in the method call, then all instances of the object shall be returned in response to the call.	X		
GetAssessmentLocations	Returns the locations for storm assessment, given a list of assessmentLocation objectIDs. The calling parameter lastReceived is included so that large sets of data can be returned in manageable blocks. lastReceived should carry an empty string the first time in a session that this method is invoked. When multiple calls to this method are required to obtain all of the data, the lastReceived should carry the objectID of the last data instance received in subsequent calls.			
GetCallsReceivedOnOutage	Returns all calls that have been processed by the outage management system in the form of an outageDetectionLogList. The calling parameter lastReceived is included so that large sets of data can be returned in manageable blocks. lastReceived should carry an empty string the first time in a session that this method is invoked. When multiple calls to this method are required to obtain all of the data, the lastReceived should carry the objectID of the last data instance received in subsequent calls.			
GetChildCircuitElements	Returns circuit elements immediately fed by the given line section or node (eaLoc).	X		
GetChildConnectivity	Returns all information for circuit elements immediately fed by the given line section or node (eaLoc).	X		
GetCircuitElementNearLatLong	Returns an array of circuitElements that lie within the distance tolerance of the location expressed in latitude and longitude. If	X		

	many circuitElements are found within the distance tolerance, then the server shall return the maximum number of circuitElements expressed as numCEs. If it is necessary to drop some circuitElements, the server should drop those furthest from the latitude/longitude location specified in the calling parameter list.			
GetCircuitElementStatus	Returns the outage event, if any, associated with a circuitElement given the objectRef of the circuitElement. The outageEventID is the objectID of an outageEvent obtained earlier by calling GetActiveOutages.			
GetCustomerCallHistory	Returns all calls that have been processed by the outage management system for a given customer account and service location in the form of an outageDetectionLogList.			
GetCustomerCallsOnServiceLocation	Returns all calls that have been processed by the outage management system for a given service location in the form of an outageDetectionLogList.			
GetCustomerOutageHistory	Returns all outage duration events that have been processed by the outage management system for a given customer account and service location in the form of an outageDurationEventList.	X		
GetDomainMembers	The client requests from the server a list of names of domains supported by the server.	X		
GetDomainNames	Requests the domains (lists of fixed information, such as the counties served, or the acceptable statusCodes for this installation).	X		
GetDownlineCircuitElements	Returns all circuit elements downline from the given circuit element.	X		
GetDownlineConnectivity	Returns all information for circuit elements fed by a given line section or node. The calling parameter lastReceived is included so that large sets of data can be returned in manageable blocks. lastReceived should carry an empty string first time in a session that this method is invoked. When multiple calls to this method are required to obtain all of the data, the lastReceived should carry in subsequent calls the index number provided by the server as being the lastSent.	X		
GetDownlineMeterConnectivity	Returns the meter connectivity for all meters down line from a given meter	X		
GetModifiedCircuitElements	Returns all circuit elements that have been modified since the previous session identified			
GetModifiedConnectivity	Returns circuit elements that have been modified since the time of a specifed sessionID. The calling parameter lastReceived is included so that large sets of data can be returned in manageable blocks. lastReceived should carry an empty string first time in a session that this method is invoked. When multiple calls to this method are required to obtain all of the data, the lastReceived should carry in subsequent calls the index number provided by the server as being the lastSent.	X		
GetOutageByCircuitElement	Returns the outage event, if any, associated with a circuitElement given the . The outageEventID is the objectID of an outageEvent obtained earlier by calling GetActiveOutages.			
GetOutageDurationEvents	Returns all outage duration events that have been processed by the outage management system for a given outage identified by outageEventID, returned in the form of an outageDurationEventList.	X		
GetOutageEvent	Returns the outageEvent for the given outageEventID. This can	X		

	be either an active or a historical outage event. The outageEventID is the objectID for the outageEvent object.			
GetOutageEventStatusByOutageLocation	Returns the current status of an outage event, given the outage location.	X		
GetParentCircuitElements	Returns circuit elements immediately upstream of the given line section or node (eaLoc).	X		
GetOutageHistoryOnServiceLocation	Returns all outage duration events that have been processed by the outage management system for a given service location in the form of an outageDurationEventList.	X		
GetOutageMessagePromptList	Returns the current outage message prompt list. The requester system can store these messages and play them back to callers on demand, based on the OutageEventID.	X		
GetOutageReasonCodes	Returns the list of outage reason codes used by the OMS implementation.	X		
GetOutageStatusByLocation	Returns the current outage status of a customer location, given the outageLocation. The outageLocation object includes the telephone number, service locationID, account number and/or meter number at the location of the outage.	X		
GetOutagedODDevices	Returns the outageDetectionDevices that are currently experiencing outage conditions.			
GetParentCircuitElements	Returns circuit elements immediately upstream of the given line section or node (eaLoc).	X		
GetParentConnectivity	Returns all information for circuit elements immediately upstream of the given line section or node (eaLoc).	X		
GetPublishMethods	Requester requests list of methods to which this server can publish information.			
GetRegistrationInfoByID	This method requests the return of existing registration information (that is to say the details of what is subscribed on this subscription) for a specific registrationID. The server should return a SOAPFault if the registrationID is not valid.			
GetSiblingMeterConnectivity	Returns all meters on the same transformer as the given meter.	X		
GetSubstationNames	Returns all substation names	X		
GetUplineCircuitElements	Returns circuit elements in the shortest route to source from the given line section or node (eaLoc).	X		
GetUplineConnectivity	Returns all information for circuit elements in the shortest route to source from the given line section or node (eaLoc).	X		
GetUplineMeterConnectivity	Returns all meters from the first up line distribution transformer.	X		
HistoryLogChangedNotification	Publisher notifies Subscriber of a change in the History Log by sending the changed historyLog object. MDM returns information about failed transactions in an array of errorObjects. The message header attribute 'registrationID' should be added to all publish messages to indicate to the subscriber under which registrationID they received this notification data.			
InitiateCut	Publisher notifies OA of power line cut that should be initiated. OA returns information on failed transactions by returrning an array of errorObjects.			
MeterAddNotification	Publisher notifies subscriber to add the associated meter(s). Subscriber returns information about failed transactions using an array of errorObjects. The message header attribute	X		

	'registrationID' should be added to all publish messages to indicate to the subscriber under which registrationID they received this notification data.			
MeterChangedNotification	Publisher Notifies subscriber of a change in the Meter object by sending the changed meter object(s). Subscriber returns information about failed transactions using an array of errorObjects.The message header attribute 'registrationID' should be added to all publish messages to indicate to the subscriber under which registrationID they received this notification data.	X		
MeterExchangeNotification	Publisher notifies subscriber that meter(s) have been deployed or exchanged. A meterExchange shall be a paired transaction of a meter being removed and a meter being installed in the same meter base. Subscriber returns information about failed transactions in an array of errorObjects. The message header attribute 'registrationID' should be added to all publish messages to indicate to the subscriber under which registrationID they received this notification data.	X		
MeterRemoveNotification	Publisher notifies subscriberto remove the associated meter(s). Subscriber returns information about failed transactions using an array of errorObjects. The message header attribute 'registrationID' should be added to all publish messages to indicate to the subscriber under which registrationID they received this notification data.	X		
MeterRetireNotification	Publisher notifies subscriber that the associated meter(s)have been retired from the system. Subscriber returns information about failed transactions using an array of errorObjects. The message header attribute 'registrationID' should be added to all publish messages to indicate to the subscriber under which registrationID they received this notification data.	X		
ODDeviceChangeNotification	Notifies of a change in outage detection events			
OutageReasonChangedNotification	Publisher notifies Subscriber of the causes and other information related to an outage event. Subscriber returns information about failed transactions using an array of errorObjects. The message header attribute 'registrationID' should be added to all publish messages to indicate to the subscriber under which registrationID they received this notification data.	X		
OutageReasonContainerChangedNotification	Publisher notifies Subscriber of changes in the cause codes that may be used to describe outage events at this installation. Subscriber returns information about failed transactions using an array of errorObjects. The message header attribute 'registrationID' should be added to all publish messages to indicate to the subscriber under which registrationID they received this notification data.			
PMChangedNotification	Publisher notifies OA of new power monitor output by sending the new PMChangedNotification. The message header attribute 'registrationID' should be added to all publish messages to indicate to the subscriber under which registrationID they received this notification data.			
RegisterForService	This method establishs a subscription using a previously requested registrationID. The calling parameter registrationInfo is a complex type that includes the following information:			

	registrationID - the previously requested registrationID obtained from the publisher by calling RequestRegistrationID, responseURL – the URL to which information should subsequently be published on this subscription, msFunction – the abbreviated string name of the MultiSpeak method making the subscription request (for instance, if an application that exposes the Meter Reading function has made the request, then the msFunction variable should include “MR”), methodsList – An array of strings that contain the string names of the MultiSpeak methods to which the subscriber would like to subscribe. Subsequent calls to RegisterForService on an existing subscription replace prior subscription details in their entirety - they do NOT add to an existing subscription.			
RequestRegistrationID	This service requests of the publisher a unique registration ID that would subsequently be used to refer unambiguously to that specific subscription. The return parameter is the registrationID, which is a string-type value. It is recommended that the server not implement registration in such a manner that one client can guess the registrationID of another. For instance the use of sequential numbers for registrationIDs is discouraged.			
ResolvedCaller	Publisher Notifies subscriber that an unresolved caller is now resolved by the dispatcher. Subscriber returns status of failed transactions in an array of errorObjects.	X		
RestoreCut	Publisher notifies OA of a cut that should be restored. OA returns information on failed transactions by returrning an array of errorObjects.			
RestoreOutage	Publisher notifies OA of an outage that should be denoted as being restored, given an outage event ID. OA returns information on failed transactions by returrning an array of errorObjects.	X		
SCADAAnalogChangedNotification	Publisher notifies OA of changes in analog values by sending an array of changed scadaAnalog objects. OA returns failed transactions using an array of errorObjects. The message header attribute 'registrationID' should be added to all publish messages to indicate to the subscriber under which registrationID they received this notification data.	X		
SCADAAnalogChangedNotificationByPointID	Publisher notifies OA of changes in a specific analog value, chosen by scadaPointID, by sending a changed scadaAnalog object. If this transaction fails, OA returns information about the failure in a SOAPFault. The message header attribute 'registrationID' should be added to all publish messages to indicate to the subscriber under which registrationID they received this notification data.	X		
SCADAAnalogChangedNotificationForPower	Publisher notifies OA of changes in a specific analog value, limited to power analogs, by sending an arrray of changed scadaAnalog objects. OA returns failed transactions using an array of errorObjects.The message header attribute 'registrationID' should be added to all publish messages to indicate to the subscriber under which registrationID they received this notification data.	X		
SCADAAnalogChangedNotificationForVoltage	Publisher notifies OA of changed analog values, limited to voltage analogs, by sending an array of changed scadaAnalog	X		

	objects. OA returns failed transactions using an array of errorObjects. The message header attribute 'registrationID' should be added to all publish messages to indicate to the subscriber under which registrationID they received this notification data.			
SCADAPointChangedNotification	Publisher notifies OA of changes in SCADA point definitions by sending an array of changed scadaPoint objects. OA returns failed transactions using an array of errorObjects. The message header attribute 'registrationID' should be added to all publish messages to indicate to the subscriber under which registrationID they received this notification data.			
SCADAPointChangedNotificationForAnalog	Publisher notifies OA of changes in SCADA point definitions, limited to Analog points, by sending an array of changed scadaPoint objects. OA returns failed transactions using an array of errorObjects. The message header attribute 'registrationID' should be added to all publish messages to indicate to the subscriber under which registrationID they received this notification data.			
SCADAPointChangedNotificationForStatus	Publisher notifies OA of changes in SCADA point definitions, limited to Status points, by sending an array of changed scadaPoint objects. OA returns failed transactions using an array of errorObjects. The message header attribute 'registrationID' should be added to all publish messages to indicate to the subscriber under which registrationID they received this notification data.			
SCADAStatusChangedNotification	Publisher notifies OA of changes in point status by sending an array of changed scadaStatus objects. OA returns failed transactions using an array of errorObjects. The message header attribute 'registrationID' should be added to all publish messages to indicate to the subscriber under which registrationID they received this notification data.	X		
SCADAStatusChangedNotificationByPointID	Publisher notifies OA of changes in the status of a specific point, chosen by PointID, by sending a changed scadaStatus object. If this transaction fails, OA returns information about the failure in a SOAPFault. The message header attribute 'registrationID' should be added to all publish messages to indicate to the subscriber under which registrationID they received this notification data.	X		
ServiceLocationChangedNotification	Publisher Notifies subscriber of a change in the Service Location object by sending the changed serviceLocation object(s). Subscriber returns information about failed transactions using an array of errorObjects. The message header attribute 'registrationID' should be added to all publish messages to indicate to the subscriber under which registrationID they received this notification data.	X		
SetOutageElementStatus	This method allows a dispatcher or operator to verify or restore any circuit element by phase. This controls how the outage analysis behaves if the outage is assumed. It also allows for connectivity changes or backfeeds by performing switching operations. It is recommended that the dispatcher verify current status using GetOutageElementStatus prior to calling this method to change the outage element status.			
UnassignCrewsFromOutage	Unassigns crew(s) from an outage given the outage event ID.	X		

	Server returns information on failed transactions by returrning an array of errorObjects.			
UnassignOutagesFromCrew	Unassigns outages(s) from a crew given the crew ID. Server returns information on failed transactions by returrning an array of errorObjects.	X		
UnregisterForService	This method deletes a previously established subscription (registration for service) that carries the registration identifer listed in the input parameter registrationID.			
UpdateMessageStatus	Publisher Notifies subscriber that a call message was listened. Subscriber returns status of failed transactions in an array of errorObjects.	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.

Certified by:

For OpMoSys, Inc.:

Name: Ayodale Cole

Title: CEO

Date: 10/7/2013

For Milsoft Utility Solutions, Inc.

Becky Paul

Name: Becky Paul

Title: Vice President Product Management

Date: 10/7/2013

Assertions Verified by:

Hannu Huhdanpaa

MultiSpeak Testing Agent

Name: Hannu Huhdanpaa

Title:

Title

Date:10/7/2013

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 Technical Coordinator), nor UISOL, Inc, 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, Inc. 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 or the Testing Agent.