MultiSpeak Version 3.0 Interoperability Assertion – Outage Analysis (OA) / Interactive Voice Response (IVR) (CH) (OD)

Vendor: Milsoft Utility Solutions DisSPatch Outage Management System and Milsoft's Porche Interactive Voice Response System (CH)

Interfaces: Outage Analysis (OA), Outage Detection (OD), Interactive Voice Response (IVR) Call Handling (CH)

MultiSpeak Version 3.0 Interoperability Assertion

Statement of Interoperable Functionality Between:

Vendors	Products	Product Version	Role	Web Client Interfaces	Web Server Interfaces
Milsoft Utility Solutions, Inc.	1. Milsoft Integration Server 2. Milsoft DisSPatch Outage Management System 3. Calls Manager	8.1	OA	OA→CH_Server OA→OD_Server	OD→OA_Server CH→OA_Server
Milsoft Utility Solutions, Inc. Porche IVR	Milsoft Porche IVR	7.34	CH OD	OD→OA_Server CH→OA_Server	OA→OD_Server OA→CH_Server

Summary:

Milsoft has chosen to implement MultiSpeak Version 3.0 web services in order to provide a coupling between utility operations and customers via Interactive Voice Recordings (IVR). The Milsoft Porche IVR is capable of handling incoming calls and initializing outage or non-outage events in Milsoft's DisSPatch system in real time to create more accurate predictions of outage extents. Calls that arrive in Porche IVR can be closed, resolved, or listened to in both system and the other will acknowledge the action appropriately.

Also when DisSPatch restores outages, it can create callback events for Porche IVR to verify a customer's power has been restored. It is also possible for DisSPatch users to create messages to be played for customers who call in and are believed to be part of a specific outage. Porche IVR is capable of requesting in real time the status of an outage location for its callers as DisSPatch predicts the extents of each outage. This automated feedback frees up the utility employees to handle outage restoration and customer service.

Prerequisites:

The Milsoft Integration Server and Porche IVR must be accessible to each other via web services. This prerequisite is achieved by exchanging the URLs of each server and account credentials. This is validated in MultiSpeak using the **PingURL** method to ensure the connection is established.

Prerequisites (Continued):

Both Milsoft DisSPatch and Milsoft IVR support the ability of listening and recording voice messages. Since transmitting all audio messages via a network would cause excessive congestion, a linked database outside of MultiSpeak must be maintained between both systems.

Service location and customer account data (phone number, account, address, etc.) must either be imported using MultiSpeak real time interfaces (not covered in this assertion) or through batch file imports from the system of record, in this case, the Customer Billing (CB) system. For real time import of customer billing data using MultiSpeak, see the assertion document reference at:

http://www.multispeak.org/utilities/ProductTesting/TestedProducts/Documents/NISCMilsoftCB_OA_EAMultiSpeakassertion.pdf

Both Milsoft IVR and DisSPatch use the customer information in order to determine the service location, account status, multiple accounts, non-pay status, account priority, etc. All kept up-to-date from either the real time interface referenced above or batch import performed on a schedule.

Milsoft provides the ability of creating voice recordings that are played back to the customer when they are reporting an outage. For manipulating voice recordings using text-to-speech, the Text-to-Speech Engine software must be configured and accessible for the DisSPatch Outage clients. This is an optional product. If the Text-to-Speech engine is not available, then the only option for performing voice recordings is through a voice recorder embedded in the software using a microphone.

Specific Vendor Assertions:

1) **Milsoft's IVR can determine if the caller is calling to report an outage.**

   Importance to the user: Calls that are reporting an outage are entered into the **Outage Detection** (**OD**) process, otherwise non-outage calls are routed to the **Call Handling** (**CH**) interface. Non-outage calls are processed and are available in both DisSPatch and Porche IVR for review and maintained for historical purpose. ***Note****: **Payment Processing (PP)**, account balance, payment extensions calls and all non-outage call types are all handled under the **Call Handling** (**CH)** interface with **Customer Billing** (**CB**) under the **CH-CB/PP-CB** assertion not covered under this, **Outage Detection** assertion.*

   How Achieved: Milsoft's IVR plays custom voice scripts that the customer interacts with either using speech recognition or touch tone responses in order to determine the type of call being processed. Non-outage calls are sent to DisSPatch using the **CHEventNotification** method.

2) **Milsoft's IVR can identify the customer calling and provide feedback to the customer that their account has been found.**

Importance to the user: This provides customer feedback that their account has been found and starts the outage detection (**OD**) process.

How Achieved: The caller ID is used to search by phone number for the unique service location belonging for this caller. This is a direct customer lookup from the database that is kept up-to-date from the **CB** system referenced under Prerequisites. A text-to-speech engine plays back to the customer their account address information in order to determine the customer's service location and fills in the MultiSpeak **outageCustomer** portion of the **outageDetectionEvent** object. For a positive find on a unique account, the **resolvedLevel** is set to **Meter** for the **outageDetectionEvent** object.

3) **Milsoft's IVR attempts to resolve the caller to a unique service location in the event that the customer's account is not found.**

Importance to the user: Milsoft feels that it is important to resolve a call while the call is in process in order to reduce the number of unresolved calls, improve the outage detection **OD** process and to provide better customer service (**Call Handling** process). This frees up having to redirect the call to a manual call operator.

How Achieved: A voice recorded script is played asking if the customer is calling from the service location that is experiencing an outage. If not, then the script request that the customer either say the phone number or enter the number using the touch tones for the service location experiencing an outage (typically a call from a cell phone not listed for the account). This is a direct customer lookup from the database that is kept up-to-date from the **CB** system referenced under Prerequisites. The **outageCustomer** object is filled in. Milsoft's IVR attempts to resolve the call to a unique meter using caller ID, phone number, meter and or address. The **resolveLevel** enumeration reflects the result of the process. **Note:** Milsoft's IVR allows the customer to configure to what level the IVR attempts to resolve a call. For instance, you can configure the IVR to request the customer enter their meter number.

4) **Milsoft's IVR can identify if the customer calling owns multiple accounts using the same phone number and attempts to resolve the caller to a unique service location.**

Importance to the user: This reduces the number of unresolved calls and improves the outage detection (**OD**) process providing customer feedback that their account has been found.

How Achieved: In the event that the caller ID or phone number results in multiple accounts, the system attempts to resolve the call to a unique service location by performing a look up of all accounts matching this call. Multiple addresses are played back to the customer using

text-to-speech in an attempt to resolve the call to a unique service location. The **resolveLevel** is set to either **Meter** if the lookup matched by meter number, to **Address** if the match was performed by address or to **Unresolved** if the system was unable to match to a unique service location. The **outageDetectionEvent** object is filled in with the results.

5) **Milsoft's IVR can request current outage status of a customer during a call and provide this feedback to the caller in real time.**

Importance to user: This provides the customer with current outage status information. If a specific outage message exists for the outage that the customer is associated with, the outage message is played to the customer.

How Achieved: The Milsoft IVR calls the **GetOutageEventStatusByOutageLocation** method while the voice scripts are being played to the customer using the information stored in **outageLocation**.

6) **Customer Requested Callback Notification. A customer can request that they be called back when the outage is restored.**

Importance to the user: This allows the customer reporting an outage to receive a callback notification when the outage is restored. This also allows for the customer to respond if they are still out of power (covered below under the section for callback list notification for customers reporting that they are still out of power).

How Achieved: A voice script is played asking the customer if they wish to be called back. The customer can either say the words yes or no, or use the touch tones on their phone. The result of the callback response is stored in the **outageCustomer callBackFlag** element object associated with the **outageDetectionEvent** for that call.

7) **Milsoft's IVR can record a customer's response in a voice recorded message.**

Importance to the user: Voice recordings from a customer can provide additional details regarding an outage that cannot be handled using touch tones on a phone. This provides a unique customer service experience for the customer that they are being heard even though they are being processed through an automated system.

How Achieved: The voice recording is stored in the database for later retrieval. Call records are marked with a voice recording icon so that a customer service representative and or dispatcher can listen to the voice recording in order to determine additional actions that can be performed. The priority can be elevated (**priority** element in **outageDetectionEvent** object). This priority level sets off an alarm that is handled by the DisSPatch system. The voice recording attributes is stored in the MultiSpeak **messageList** object inside of the **outageDetectionEvent** object. The individual **message** object contains the unique **recordID**

for this message and attributes that track the **eventTime**, **listenedOn**, **listenedBy** of the recorded voice message.

8) **When messages are listened to in either product, the other is kept in sync. The actual voice recording can be played back from any of the following Milsoft products; Calls Manager**, **DisSPatch Outage Management** and or the **Milsoft Porche IVR OCM** products.

Importance to the user: This allows for flexibility in distributing the work load of listening to voice-recorded messages which contain additional important outage details. Recorded messages are synchronized throughout all Milsoft products determining the status of the recorded message. The message status contains when the message was played back and by which employee listened to the message.

How Achieved: Each Call Event record processed from the **outageDetectionEvent** object is marked with a voice message icon letting the Dispatcher or Customer Service Representative know that there is one or more voice recordings associated with that call event. When a voice **message** is played back using any one of the Milsoft products listed above, the voice message icon changes to reflect that the recording was heard. This sets the **listenedOn** event to the date and time the recording was played and the **listenedBy** event to the name of the employee that listened to the **message**. A call to **UpdateMessageStatus** is performed to update the message status.

9) ***The DisSPatch Outage Analysis system processes the IVR calls, predicts the location of the outage and marks all outage calls associated with the prediction with the appropriate outage ID.***

Importance to the user: After all the requirements are met for the call being processed by the IVR system, it sends the outage call to the **Outage Analysis** (**OA**) system. This outage detection event is what starts the outage prediction process. See the Milsoft DisSPatch Outage Analysis assertion documents for details on outage analysis at:
http://www.multispeak.org/utilities/ProductTesting/TestedProducts/Pages/Versions30and4x.aspx

How Achieved: The Milsoft IVR sends the completed **outageDetectionEvent** object using the **ODEventNotification** to the **Outage Analysis** (**OA**) system.

10) **The dispatcher can either enter a text message or record an outage message that can be played to all new customers that call that are associated with a specific outage.**

Importance to the user: All new customers calling to report an outage using the IVR system that are associated with the outage event will hear the informative outage message played back. This improves customer service feedback regarding the outage status.

How Achieved: Using the Outage Manager in DisSPatch, the dispatcher can select a specific outage and press the **Record Message** button. The **Recording for Outage** dialog then allows the dispatcher to either record a voice message using a microphone or enter a text message that will be converted to speech using the Text-to-Speech engine. This **message** object is sent to the IVR system using the **OutageMessagePromptList** method.

11) **DisSPatch can send a list of callbacks of all customers that called affected by an outage.**

Importance to user: Upon restoring an outage in DisSPatch, the user may verify power restoration by initiating callbacks of all customers that called or just those that requested a callback on the Porche IVR.

How Achieved: Upon outage restoration, a new **callBackList** object is sent using the **CallBackListNotification**. The Milsoft IVR will call each individual number from the callback list and play a scripted voice message asking the customer to confirm if their power is restored.

12) **Milsoft's IVR, using results from the callback list campaign can determine if the customer responded that they are still out of power.**

Importance to the user: Milsoft's IVR can provide and confirm outage restoration by providing customer real time feedback. This feedback allows the dispatcher to determine if there are still customers out of power.

How Achieved: In the event a caller responds that they are still out of power, the **outageDetectionEvent** object is resent with the **callbackStatus** enumeration of **StillOff**. A new **ODEventNotification** method is sent to the outage analysis system which will result in a new outage prediction.

13) **The Milsoft DisSPatch system can notify the Milsoft IVR to close out a list of calls that have been processed.**

Importance to the user: Once the cycle of processing calls is complete, DisSPatch can keep the Porche IVR in sync by closing out the calls. These calls are then archived for historical purposes, analysis and reports.

How Achieved: Upon outage restoration or discarding an outage and calls, the DisSPatch sends a list of **customerCalls** using the **CloseCalls** method. Once issued, the calls are archived for historical and report purposes.

14) **Detecting unknown callers. The Porche IVR can record a voice message either providing additional information regarding an outage or a customer's outage location for unknown phone numbers.**

Importance to user: Milsoft feels it is important to provide positive customer service experience by providing a solution to resolving unknown outage calls to the appropriate account. Both the DisSPatch system and the Calls Manager software used by Customer Service Reps are notified of calls requiring review and of those, which call records are marked as Unresolved for those callers that cannot be resolved to a unique service location. DisSPatch users can listen to the voice recording. These voice recording may contain additional outage information.

How Achieved: In the event of an unknown account or a multi meter account, the Porche IVR will tag this call as an unresolved caller by setting the **resolvedLevel** element to **Unresolved** in the **outageDetectionEvent** object. The DisSPatch user can later listen to the voice recording containing the location of the outage call in order to resolve the call to a specific outage location using the **ResolvedCaller** method.

15) **DisSPatch can resolve calls that could not be automatically resolved by the IVR system.**

Importance to the user: A call can be resolved to a unique service location and update IVR to be able to better handle later calls from the same caller.

How Achieved: Calls can be resolved either in DisSPatch of in Calls Manager. The call records are marked as unresolved. The voice recorded message containing the address information can be played and the unique service location found that is associated with this unresolved call. The call is resolved using the **ResolvedCaller** method with the original **objectID** and the updated outage location identifier.

16) **DisSPatch can add call records taken manually from Customer Service Representatives (CSR) and send them to the IVR system.**

Importance to the user: If a caller is handled by a CSR, the IVR system can be notified of these manual calls in order to avoid duplicate outage tickets for the same location.

How Achieved: The DisSPatch system calls the **ManualCallList** method with the generated **objectID**.

Products: Milsoft Integration Server and Porche Web Server

Summary of Interoperability Test Results

Interface #6

CH_Server→OA_Server

Table 1

Required MultiSpeak Methods

Method Name	Importance to User	Supported by Server[1] (OA)	Supported by Client[2] (CH)	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

Table 2

Optional MultiSpeak Methods

Method Name	Importance to User	Supported by Server[1] (OA)	Supported by Client[2] (CH)	Verified Inter-operable[3]
CHEventNotification	CH Notifies OA of non-outage events by sending the customerCall object.	X	X	X
CloseCalls	CH Notifies OA of a list of customer calls to close out. OA returns status of failed transactions in an array of errorObjects.	X	X	X
GetDomainMembers	The client requests from the server the members that comprise a domain in the server. This method is used in conjunction with GetDomainNames to enable systems to exchange information about application-specific or installation-specific lists of information.	X	X	X
GetDomainNames	The client requests from the server a list of names of domains supported by the server. This method is used, along with the GetDomainMembers method to enable systems to exchange information about application-specific or installation-specific lists of information, such as the lists of counties for this installation or the list of serviceStatusCodes used by the server.	X	X	X
GetOutageEventStatus	Returns the current status of an outage event, given the outage event ID. The outageEventID is the objectID of an outageEvent obtained earlier using the CustomersAffectedByOutageNotification, CallBackListNotification, or the GetActiveOutages methods.	X		
GetOutageEventStatusByOutageLocation	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	X	X	X

	the outage.			
GetOutageMessagePromptList	Returns the current outage message prompt list. The CH system can store these messages and play them back to callers on demand, based on the OutageEventID returned by OA			
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		
ResolvedCaller	CH notifies OA of an unknown inbound call being resolved to a specific service location. The objectID of the customerCall is the callrecordID provided in the ODEventNotification.	X	X	X
UpdateMessageStatus	CH notifies OA that a specific message has been listened to on their system. This will update in OA when the message was last listened to and by whom.	X	X	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.

Products: Milsoft Integration Server and Porche Web Server

Summary of Interoperability Test Results
Interface #6
OA_Server → CH_Server
Table 3
Required MultiSpeak Methods

Method Name	Importance to User	Supported by Server[1] (CH)	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. (CH returns List of T1 errors as errorObjects where applicable)	X	X	X

Table 4
Optional MultiSpeak Methods

Method Name	Importance to User	Supported by Server[1] (CH)	Supported by Client[2] (OA)	Verified Inter-operable[3]
CallBackListNotification	OA Notifies CH of new call back list(s) by sending the new call back lists. CH returns status of failed transactions in an array of errorObjects.	X	X	X
CloseCalls	OA Notifies CH of a list of customer calls to close out. CH returns status of failed transactions in an array of errorObjects.	X	X	X
GetDomainMembers	The client requests from the server the members of a specific domain of information, identified by the domainName parameter, which are supported by the server. This method is used, along with the GetDomainNames method to enable systems to exchange information about application-specific or installation-specific lists of information, such as the lists of counties for this installation or the list of serviceStatusCodes used by the server.	X	X	X
GetDomainNames	The client requests from the server a list of names of domains supported by the server. This method is used, along with the GetDomainMembers method to enable systems to exchange information about application-specific or installation-specific lists of information, such as the lists of counties for this installation or the list of serviceStatusCodes used by the server.	X	X	X
ManualCallList	OA Notifies CH that a customer service representative took an outage call manually. CH returns status of failed transactions in an array of	X	X	X

	errorObjects.			
OutageMessagePromptList	OA Notifies CH of new outage messages by sending a list of outage messages. CH returns status of failed transactions in an array of errorObjects.	X	X	X
ResolvedCaller	OA Notifies CH that an unresolved caller is now resolved by the dispatcher. CH returns status of failed transactions in an array of errorObjects.	X	X	X
UpdateMessageStatus	OA Notifies CH that a call message was listened. CH returns status of failed transactions in an array of errorObjects.	X	X	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.

Products: Milsoft Integration Server and Porche Web Server

Summary of Interoperability Test Results

Interface #5

OA→OD

Table 1
Recommended MultiSpeak Methods

Method Name	Importance to User	Supported by Server[1] (OD)	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
GetAllOutageDetectionDevices	Returns all Outage Detection Devices.	X		
GetOutageDetectionDevicesByMeterNo	Returns an Outage Detection Device Associated with the Given Meter Number.			

Table 2
Optional MultiSpeak Methods

Method Name	Importance to User	Supported by Server[1] (OD)	Supported by Client[2] (OA)	Verified Inter-operable[3]
CancelODMonitoringRequestByObject	Cancel outage detection monitoring on the list of supplied circuit elements.			
DisplayODMonitoringRequests	Requests a list of circuit elements being monitored.			
GetDomainMembers	Requests the members of a given domain (type of fixed information, such as all of the counties in the database).		X	
GetDomainNames	Requests the domains (lists of fixed information, such as the counties served, or the acceptable status codes for this installation).		X	
GetOutageDetectionDevicesByStatus	Returns all outage detection devices with a given status.			
GetOutageDetectionDevicesByType	Returns all outage detection devices with a given type			
GetOutagedODDevices	Returns the outage detection devices that are currently experiencing an outage.		X	
InitiateODEventRequestByObject	Initiates an outage detection event request on service locations experiencing an outage downline from a circuit element.			
InitiateODMonitoringRequestByObject	Initiates an outage detection monitoring request on service locations downline from a circuit element at a given time interval.			
InitiageOutageDetectionEventRequest	Initiates an outage detection event request on the list of meter numbers.		X	
ModifyODDataForOutageDetectionDevice	Allow OA to Modify OD data for a specific Outage Detection Device 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.

Products: Milsoft Integration Server and Porche Web Server

Summary of Interoperability Test Results
Interface #5
OD → OA

Table 3
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
ODEventNotification	Notifies a change in outage detection events	X	X	X

Table 4
Optional MultiSpeak Methods

Method Name	Importance to User	Supported by Server[1] (OA)	Supported by Client[2] (OD)	Verified Inter-operable[3]
GetActiveOutages	Returns the outage Event IDs for all active outage events.	X		
GetAllCircuitElements	Returns all circuit elements.	X		
GetChildCircuitElements	Returns circuit elements immediately fed by the given line section or node (eaLoc).	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		
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	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	X	X
GetParentCircuitElements	Returns circuit elements immediately upstream of the given line section or node (eaLoc).	X		
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		
GetUplineMeterConnectivity	Returns all meters from the first up line distribution transformer.	X		
ODDeviceChangeNotification	Notifies of a change in outage detection events	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 Milsoft Utility Solutions, Inc.:

Name: Luis R. Malavé

Title: Executive Vice President

Date: 2/19/2013

For Milsoft Utility Solutions, Inc.
Porche IVR Division:

Name: Becky Paul

Title: Vice President, Product Management

Date: 2/19/2013

Assertions Verified by:

Name: Gary A McNaughton

Title: Vice President

NRECA/Cornice Engineering
Testing Agent

Date: 2/19/2013

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 nor Cornice Engineering, Inc. (MultiSpeak Project Coordinator) acting on behalf of NRECA, makes any warranty nor guarantee that the software will perform as described in this assertion when installed at any specific utility. Furthermore, neither NRECA, Cornice Engineering, Inc. makes any warranty nor 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.