

RNI 4.14 MultiSpeak v3.0 Gateway

SECTION 1: OVERVIEW

Copyright

This document, in whole or in part, ("Document") includes confidential and proprietary information belonging to Sensus USA Inc. and/or one of its subsidiaries or affiliates. Unauthorized use, reproduction, disclosure, distribution, or dissemination of this Document is strictly prohibited. No party may use, reproduce, disclose, distribute, or disseminate this Document for any purpose without express written authorization from Sensus USA Inc. Any use, reproduction, disclosure, distribution, or dissemination of this Document does not transfer title to, license, or grant any patent, copyright, trademark, or other intellectual property rights. This Document, and any copies or derivatives thereof, must be returned immediately on demand. This Document is subject to any applicable non-disclosure agreement(s). Information in this Document is subject to change without notice and does not represent a commitment on the part of Sensus.

© 2023, Sensus USA, Inc., a subsidiary of Xylem, Inc. All rights reserved.

FlexNet® and associated logos are trademarks of Sensus and its subsidiaries and affiliates. All other brand names may be trademarks of their respective owners.

Sensus
637 Davis Drive
Morrisville, NC 27560
1-800-638-3748
www.sensus.com

Document: MultiSpeak v3.0 Gateway Section 1: Overview

Contents

1	Overview	1
1.1	Introduction.....	1
1.1.1	Scope	1
1.1.2	Target audience	1
1.1.3	Terminology.....	1
1.1.4	Notation	2
1.2	General considerations	2
1.2.1	Web service protocols.....	2
1.2.2	URLs	2
1.2.3	Message exchange patterns	2
1.2.4	MultiSpeakMsgHeader	5
1.2.5	Error reporting	6
1.2.6	Meter identification	7
1.2.7	Security	8
1.2.8	Date and time formats	9
1.2.9	Configurations	9
1.3	Common MultiSpeak methods	10
1.3.1	PingURL	10
1.3.2	GetMethods	11

1 Overview

1.1 Introduction

The Sensus MultiSpeakv3.0 web service gateway implements a subset of the Meter Reading (interface 2A), Outage Detection (interface 5) and Connect-Disconnect (interface 2B) services from the MultiSpeak specification. This document presents the API for this gateway and is a guide for developing or customizing an application to integrate with it.

1.1.1 Scope

The scope of this document is the MultiSpeakv3.0 gateway in the RNI. All the web services adhere to the last *publicly available* MultiSpeak specification – 3.0 build j.

A MultiSpeak v4.1 gateway is also available that is documented separately.

1.1.2 Target audience

The target audience is software integration engineers looking to leverage the AMI capabilities of the RNI through web services.

1.1.3 Terminology

Term	Meaning
Customer Billing (CB) system	The system of record for customer information and meter deployments. This provides the Customer Billing function defined in the MultiSpeak specification.
Gateway	A collection of related web services deployed as a single application in the RNI.
Meter Reading system (MR)	The AMR system (i.e., the Sensus RNI in the context of this document). This provides the Meter Reading function as defined in the MultiSpeak specification.
MultiSpeak	Specification for utility information exchange developed by the MultiSpeak initiative (http://www.multispeak.org/).
Outage Analysis (OA) System	The integrating system that is interested in outage events in the meter network. Also referred to as an Outage Management System (OMS).
Outage Detection (OD) system	The RNI is the outage detection system. This provides the Outage Detection function as defined in the MultiSpeak specification.
Regional Network Interface (RNI)	The head end system for the Sensus FlexNet AMR/AMI.
SOAP	A data format for communicating with web services. Usually used with the HTTPS protocol as a transport.
Web service	A collection of related web methods.
Web method	An operation that can be invoked remotely.

1.1.4 Notation

Following are the conventions used in this document:

- Mandatory parts of the SOAP requests and responses are denoted in bold font and marked with an asterisk (e.g., **UserID***).
- *Italics* are used for emphasis.
- **Highlights** are used (sparingly) for statements of high importance.
- The Consolas font is used for words that must be typed as shown, wherever applicable.
- [RN] indicates a release specific note. This will also be otherwise highlighted.
- Sample requests/responses/notifications are informative only.

1.2 General considerations

1.2.1 Web service protocols

The MultiSpeak gateway supports SOAP 1.1 over HTTPS.

Support for HTTPS is through the Apache reverse proxy in the RNI.

1.2.2 URLs

The specific URLs for each service are listed in their respective sections. However, they follow the same general pattern [protocol]://[rni_web_server]/multispeak/[service] (e.g., <https://myrniweb.mycompany.com/multispeak/mrcb>).

1.2.3 Message exchange patterns

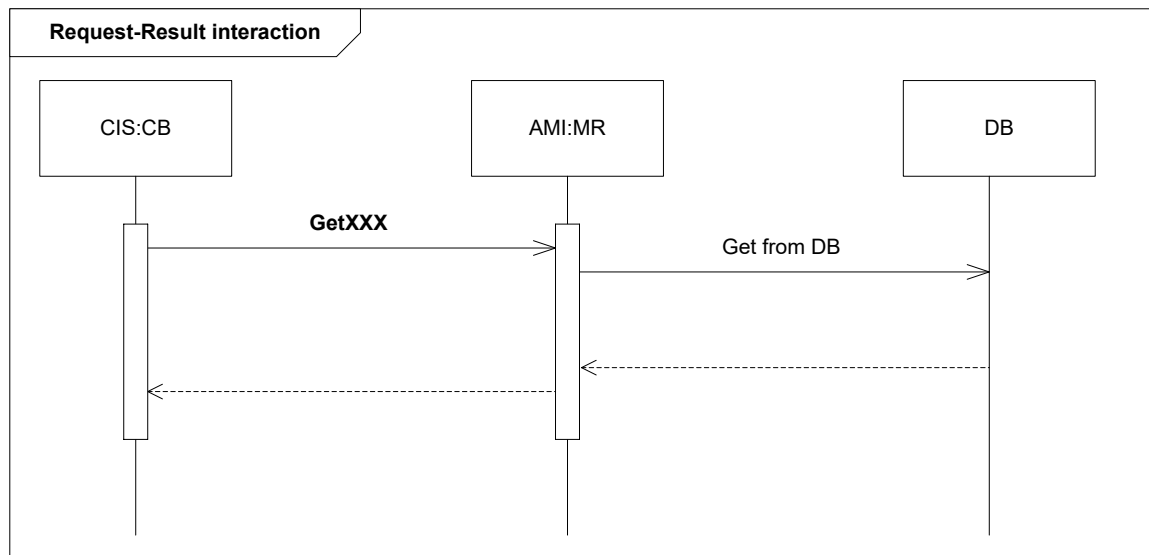
This section introduces the message exchange patterns implemented in the MultiSpeak gateway and the corresponding naming conventions.

Every SOAP message in MultiSpeak is technically a request and has a corresponding response. This response may or may not contain the *result* of the request. Since this is an intrinsic part of every interaction you will not see a Request-Response pattern called out below. Instead these patterns *focus on the exchange of business information*.

1.2.3.1 Request-result

This is a *synchronous* exchange in which the *result* of the request is contained in the response. The caller is expected to wait for the response. These are not expected to be long running operations.

The web methods are typically named as Get<...> e.g., GetAMRSupportedMeters.

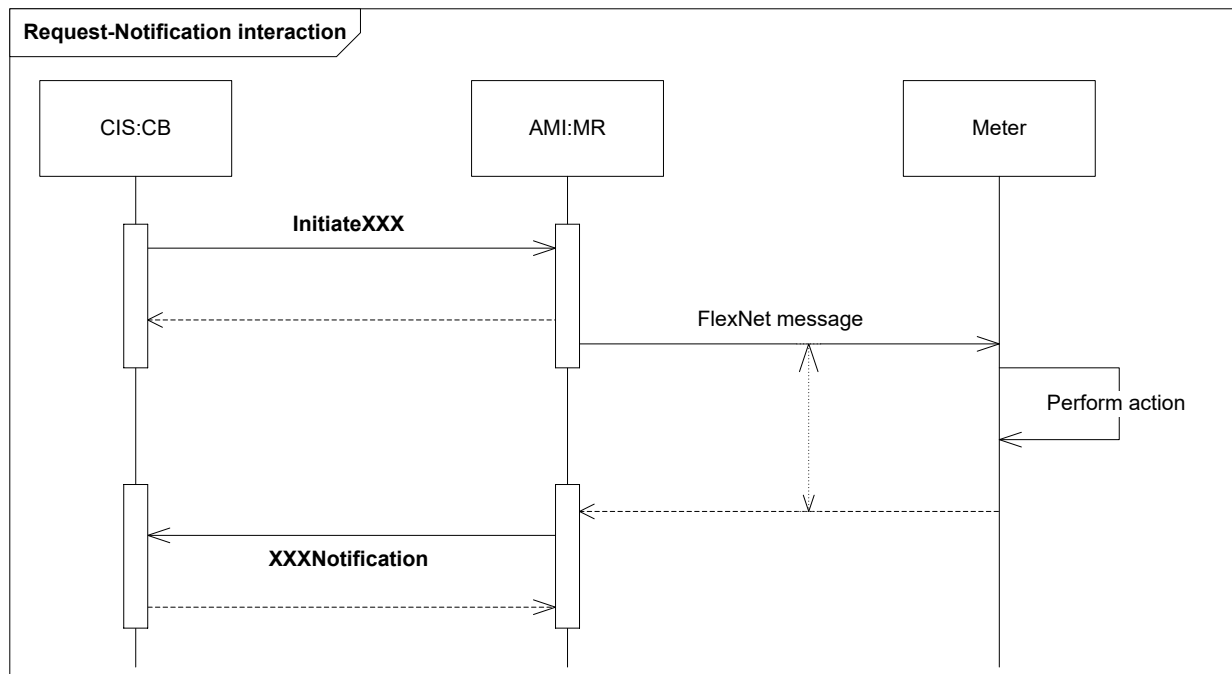


1.2.3.2 Request-notification

Long running operations follow an *asynchronous* model. The request triggers an operation and the result of the operation is later reported by publishing a SOAP message in the reverse direction. For this exchange, the gateway on the RNI is the server for the incoming message (request) and the client for the outgoing message (notification).

The web methods to trigger an action are typically named as `Initiate<...>`, and the methods to return the result of these operations are named as `<...>Notification` (e.g., `InitiateConnectDisconnect` and `CDStatesChangedNotification`).

It is common for the request and notification to share some context (for example, `transactionID`), which facilitates the correlation of the notification with the request on the integrating system(s). In addition to *request notification*, this type of message is also referred to as a *solicited notification* in Sensus documentation.

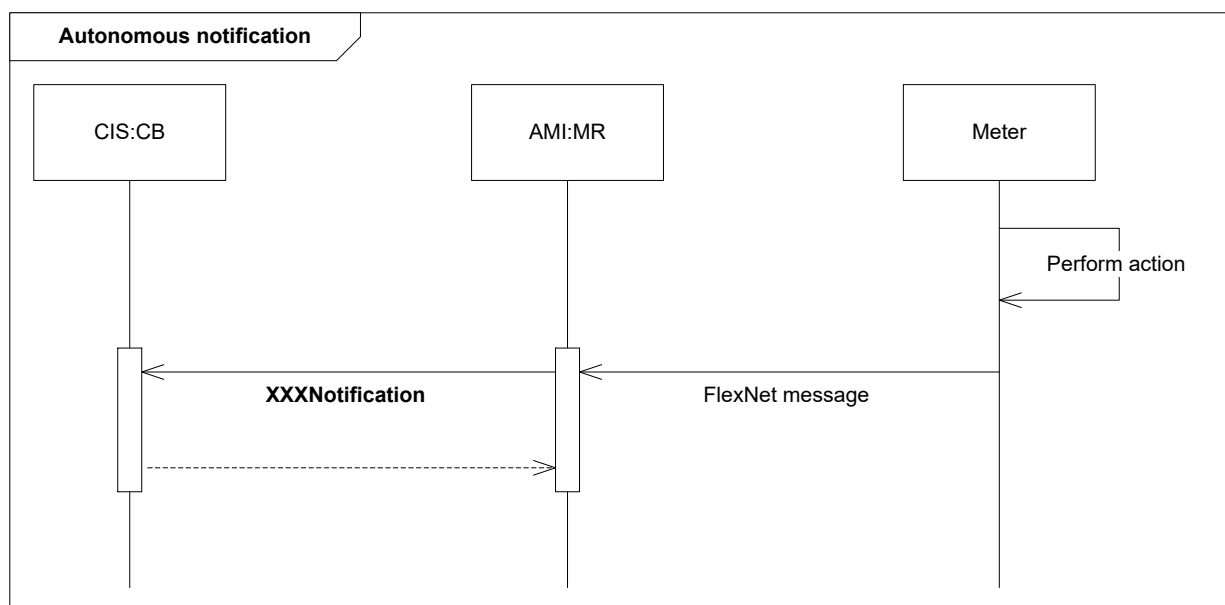


As mentioned previously, every MultiSpeak message has a response. In this pattern, the response for the incoming request can contain an error if the request was not accepted. When this occurs, as expected, the operation is not executed. Similarly, the response for the outgoing notification could contain an error if the receiver encountered a problem processing the notification.

1.2.3.3 Autonomous notification

The MultiSpeak gateway publishes notifications to report events of interest in the AMI system (e.g., `ODEventNotification`). These are typically the same notifications as in the Request-Notification pattern described previously.

As this type of message does not share context with a previous request, it is also referred to as an *unsolicited notification*.



The MultiSpeak gateway is also a receiver of notifications e.g., MeterAddNotification. When it receives a notification, the message is processed by the gateway and the response indicates whether the processing encountered any errors (i.e., the request-result pattern).

1.2.4 MultiSpeakMsgHeader

All MultiSpeak messages (requests, responses and notifications) must contain the MultiSpeakMsgHeader SOAP header. This header has many optional attributes and the Sensus MultiSpeak v3.0 gateway supports a subset of these attributes. Most of these are common to all methods and in the interest of saving space are shown in this section and referenced in all the method descriptions as “*Common request header attributes*,” “*Common response header attributes*,” or “*Common notification header attributes*.”

1.2.4.1 Common request header attributes

	Attribute/Element	Usage
HEADER	UserID*	User ID provisioned with RNI roles and permissions.
	Pwd*	Clear text password corresponding to the UserID.
	AppName	Web service client application name. This is only used for logging.
	AppVersion	Web service client application version. This is only used for logging.
	Company	Web service client application vendor. This is only used for logging.
	SessionID	Session identifier that is meaningful to the client. This is logged on the server to help with auditing.
	CustomerID	Sensus-specific extension for the utility code. Refer section 1.2.6.1.

1.2.4.2 Common response header attributes

	Attribute or Element name	Usage
HEADER	AppName	Name of the application sending the response e.g., RNI
	AppVersion	Version of the application sending the response e.g., 3.1
	Company	Fixed value "Sensus"
	SessionID	Session identifier from the request.

1.2.4.3 Common notification header attributes

	Attribute or Element name	Usage
HEADER	UserID*	User ID on the remote system.
	Pwd*	Clear text password corresponding to the UserID
	AppName	Name of the application sending the response e.g., RNI
	AppVersion	Version of the application sending the response e.g., 3.1
	Company	Fixed value "Sensus"
	SessionID	Session identifier for the notification

1.2.5 Error reporting

Errors encountered during request processing are reported to the client either through SOAP faults or by elements inside the response payload. The choice of construct is driven by the nature of the error and the definition of the responses in the WSDL.

For the Request-Notification message pattern, the SOAP fault is not an option so errors are reported via elements inside the notification payload.

1.2.5.1 SOAP fault

Errors that cause the entire request to fail will result in a SOAP fault. System and security errors fall into this category.

Some MultiSpeak method responses are defined without any elements to carry errors and in those cases SOAP faults will be used. These cases are called out specifically in the integration guide description of that method.

1.2.5.2 Error object

Business errors are related to the objects within the request and hence do not necessarily cause the entire request to fail. These are usually reported in errorObject elements.

	Attribute/Element	Usage
BODY	errorObject	Multiple elements, one per error
	errorObject.objectID*	The identifier for the object that the error is being reported against e.g., the MeterID for a meter
	errorObject.eventTime*	Timestamp of when the failure occurred on the server

Attribute/Element	Usage
errorObject.errorString*	Detailed (user friendly) error message
errorObject.nounType	The type of the object indicated by objectID e.g., Meter

1.2.5.3 Error string

Some messages are defined without errorObject elements in the payload but the objects have an errorString attribute. This is most common when a notification indicates an error in a prior request (see the Request-Notification message pattern).

1.2.6 Meter identification

All methods that deal with meters identify the meter using the meter number (a.k.a. meterNo). This value is determined by the utility and is usually but not necessarily the physical serial number of the meter.

1.2.6.1 The CustomerID attribute

Since the RNI can be a shared management platform (i.e., multi-tenant), each meterNo is resolved within the context of a *customerID* (a unique moniker for the utility that owns meters managed by the RNI). Thus when any request or data for a meter comes in, it must be identified by the customerID + meterID tuple.

Since the MultiSpeak 3.0j specification does not include such a data partitioning attribute, the Sensus implementation supports two options to indicate the customer ID.

1. The DefaultCustomerId system property can be set in the RNI DB and this will be used for all requests to the gateway. This is only suitable for a deployment in which the gateway only serves one utility. As of RNI 4.3, if the UserID identified in the request is assigned a single customer in the RNI, then this assigned customer will be used rather than the DefaultCustomerId.
2. The custom attribute CustomerID can be set in the MultiSpeak message header for every request. If provided, this supersedes the DefaultCustomerId setting.

The following sample shows the CustomerID attribute in a MultiSpeak message.

```
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ver="http://www.multispeak.org/Version_3.0"
  xmlns:sensus="http://www.sensus.com/multispeak3">
  <soapenv:Header>
    <ver:MultiSpeakMsgHeader Version="3.0"
      UserID="testuser" Pwd="testpass"
      AppName="myapp" AppVersion="1.5"
      Company="myco"
      sensus:CustomerID="ACME" />
  </soapenv:Header>
  <soapenv:Body>
    . . . . .
  </soapenv:Body>
</soapenv:Envelope>
```

Sidebar: Later releases of MultiSpeak do have a utility attribute to identify the utility that “owns” a meter. The MultiSpeak v4.1 gateway uses this utility attribute.

The use of the MultiSpeak v3.0 gateway in Multi Service Type deployments is *highly* discouraged as the v3.0 specification identifies meters using meterNo alone and does not provide a means to specify service type when selecting meters.

In the current implementation, if neither the CustomerID attribute in the request nor the DefaultCustomerId is set then the response will report a Missing header attribute CustomerID error message. This error implies a configuration problem and hence is only likely to be seen in a new deployment.

1.2.7 Security

1.2.7.1 Privacy

- The web service implementation does not support encryption in the SOAP payload. It relies on transport security (i.e., SSL) to guard against eavesdropping.

1.2.7.2 Authentication

- Supports per-request basic authentication as defined in the MultiSpeak message header.
- Supports authentication against an LDAP user store. This includes support for Active Directory.

1.2.7.3 Authorization

- Provides access control based on the *actions* that the user can perform and *restrictions* on the resources that the request is trying to use. Both levels of access are managed based on the user record in LDAP.
- The authorization enforces a fail-fast policy for access control. As soon as an illegal access is detected, the request is aborted. A missing meter is not considered a security violation and is instead reported as a business error.
- Notification destinations are defined per-customer in the RNI. Thus, notifications for a meter are only published to the consumer destination associated with that meter's customer ID (owner).

1.2.7.4 Common security errors

Any authentication or authorization violations from processing a MultiSpeak request will result in a system error and cause entire request to fail. The table below shows the error messages returned in the SOAP fault.

Since this behavior is common across all MultiSpeak messages, these error conditions will be referenced in method descriptions as “*Common security errors*.”

Error	Reasons
SYSTEM ERRORS	Userid and password are incorrect
	Credentials provided in the request header are not valid. Usually indicates a mismatch between the user store and the customer billing system.
	Userid and password attributes are not provided in the MultispeakMsgHeader.
	The UserId and/or Pwd attributes are not provided in the MultiSpeakMsgHeader. Authentication cannot proceed w/o these credentials.
	User is not allowed to perform the action on the specified resource
	The user does not have the appropriate permission in the user store or a meter in the request belongs to a customer ID that the user does not permissions to access.

1.2.8 Date and time formats

The date and time format used by this web service conforms to the W3C standard as specified by the schema. The **dateTime** data type in the schema follows the **ISO 8601** standard as described in the following link.

<http://www.w3.org/TR/NOTE-datetime>

The profile defines two ways of handling time zone offsets:

1. Times are expressed in UTC (Coordinated Universal Time), with a special UTC designator ("Z").

Example: 2011-05-11T17:23:00.000Z

Note: This format will be used in notifications published by the Sensus gateway.

2. Times are expressed in local time, together with a time zone offset in hours and minutes. A time zone offset of "+hh:mm" indicates that the date/time uses a local time zone which is "hh" hours and "mm" minutes ahead of UTC. A time zone offset of "-hh:mm" indicates that the date/time uses a local time zone which is "hh" hours and "mm" minutes behind UTC.

Example: 2011-05-11T13:23:00.000-04:00

1.2.9 Configurations

Many of the methods described in this document have some configurations for customizing the method behavior. This section introduces some conventions for how these configurations are used. Within each method description, the configurations are broadly categorized as "DB Configurations" or "File Configurations" based on where they are stored. The storage also has implications on how these configurations are viewed and modified.

1.2.9.1 DB configuration

DB configurations are saved in the RNI DB.

- Viewed/modified through the Configuration page in the RNI UI.
- Are preserved across upgrades.
- Settable per-customer or system-wide.
- Generally, read-on-demand. However, in order to improve performance an application level cache provides short term storage. The caching duration is controlled by the `systemconfig.cacheExpiration` file configuration shown in the following table.

	Property name	Usage
FILE	systemconfig.cacheExpiration	Configuration value cache expiration time in seconds. Configuration values that can be changed at run time will be cached for the configured amount of time. The default is 60 seconds.

1.2.9.2 File configuration

File configurations are saved in the `customApplication.properties` file that is included with each gateway component. These are behaviors that are expected to change rarely, if ever, once the system is set up.

- Viewed and modified using a text editor
- Are preserved across upgrades
- Only application-wide
- Always read-on-startup

1.3 Common MultiSpeak methods

Each service in the MultiSpeak v3.0 gateway implements the following methods:

- PingURL
- GetMethods

1.3.1 PingURL

This method can be used to verify that the particular web service (e.g., ODOA) is available. In the current implementation a successful ping *only* indicates that the MultiSpeak gateway received the request and user credentials were valid.

1.3.1.1 Sample request

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ver="http://www.multispeak.org/Version_3.0">
  <soapenv:Header>
    <ver:MultiSpeakMsgHeader Version="3.0"
      UserID="testuser" Pwd="testpass"
      AppName="myapp" AppVersion="1.5"
      Company="myco"/>
  </soapenv:Header>
  <soapenv:Body>
    <ver:PingURL/>
  </soapenv:Body>
</soapenv:Envelope>
```

1.3.1.2 Sample response

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ns2="http://www.multispeak.org/Version_3.0">
  <soapenv:Header>
    <ns2:MultiSpeakMsgHeader AppName="RNIGateway" AppVersion="1.0"
      CSUnits="feet" Company="Sensus" SessionID="779"/>
  </soapenv:Header>
  <soapenv:Body>
    <ns2:PingURLResponse>
      <ns2:PingURLResult/>
    </ns2: PingURLResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

1.3.1.3 Request parameters

	Attribute/Element	Usage
HEADER	Common request headers	

1.3.1.4 Response parameters

	Attribute or Element name	Usage
HEADER	Common response headers	
BODY	PingURLResult	Always empty if present.

1.3.1.5 Error messages

	Error	Reasons
System errors	Userid and password are incorrect	Credentials provided in the request header are not valid. Usually indicates a mismatch between the user store and the customer billing system.

1.3.2 GetMethods

This method can be used to get the list of method *currently* supported by a MultiSpeak service (e.g., ODOA). The method names in the response are the unqualified names as defined in the WSDL (e.g., 5A_OD_OA.wsdl)

1.3.2.1 Sample request

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ver="http://www.multispeak.org/Version_3.0">
  <soapenv:Header>
    <ver:MultiSpeakMsgHeader Version="3.0"
      UserID="testuser" Pwd="testpass"
      AppName="myapp" AppVersion="1.5"
      Company="myco"/>
  </soapenv:Header>
  <soapenv:Body>
    <ver:GetMethods/>
  </soapenv:Body>
</soapenv:Envelope>
```

1.3.2.2 Sample response

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ns2="http://www.multispeak.org/Version_3.0">
  <soapenv:Header>
    <ns2:MultiSpeakMsgHeader AppName="RNIGateway" AppVersion="1.0"
      CSUnits="feet" Company="Sensus" SessionID="779"/>
  </soapenv:Header>
  <soapenv:Body>
    <ns2:GetMethodsResponse>
      <ns2:GetMethodsResult>
        <ns2:string>PingURL</ns2:string>
        <ns2:string>GetMethods</ns2:string>
        <ns2:string>InitiateOutageDetectionEventRequest</ns2:string>
      </ns2:GetMethodsResult>
    </ns2:GetMethodsResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

1.3.2.3 Request parameters

	Attribute/Element	Usage
HEADER	Common request header attributes.	

1.3.2.4 Response parameters

	Attribute or Element name	Usage
HEADER	Common response header attributes.	
BODY	GetMethodsResult	List of strings, one per method supported by the current implementation.

1.3.2.5 Error messages

	Error	Reasons
SYSTEM ERRORS	Userid and password are incorrect	Credentials provided in the request header are not valid. Usually indicates a mismatch between the user store and the customer billing system.

Xylem |'zīləm|

- 1) The tissue in plants that brings water upward from the roots;
- 2) a leading global water technology company.

We're a global team unified in a common purpose: creating advanced technology solutions to the world's water challenges. Developing new technologies that will improve the way water is used, conserved, and re-used in the future is central to our work. Our products and services move, treat, analyze, monitor and return water to the environment, in public utility, industrial, residential and commercial building services settings. Xylem also provides a leading portfolio of smart metering, network technologies and advanced analytics solutions for water, electric and gas utilities. In more than 150 countries, we have strong, long-standing relationships with customers who know us for our powerful combination of leading product brands and applications expertise with a strong focus on developing comprehensive, sustainable solutions.



Sensus
637 Davis Drive
Morrisville, NC 27560
Tel +1.800.638.3748
www.sensus.com

Sensus, the Sensus logo, FlexNet® and associated logos are trademarks of Sensus and its subsidiaries and affiliates.