User’s Guide
User’s Guide

Prepared By:
Gary A. McNaughton, P.E.
Warren P. McNaughton, P.E.
Cornice Engineering, Inc.
P.O. Box 2350
Pagosa Springs, Colorado 81147-2350

For:
Cooperative Research Network
National Rural Electric Cooperative Association
4301 Wilson Boulevard
Arlington, Virginia 22203-1860
The National Rural Electric Cooperative Association (NRECA) is the national service organization of more than 900 rural electric systems. These cooperatively owned utilities own and operate about 44% of the miles of distribution lines in the nation to provide power to less than 10% of the nation’s people, primarily in the sparsely populated, rural areas of 46 states.

NRECA was founded in 1942 to unite rural electric systems in a way that would permit them to develop the services and support they needed to properly serve rural America. NRECA is one of the largest, rural-oriented cooperative organizations in the United States.

The Cooperative Research Network, a service of NRECA that has supported this project, was created to conduct studies and carry out research of special interest to rural electric systems and their consumers.

©MultiSpeak® 2 User’s Guide
Copyright © 2003 by National Rural Electric Cooperative Association.
All rights reserved. Reproduction in whole or in part is strictly prohibited without prior written approval of the National Rural Electric Cooperative Association.

Legal Notice
WHILE THE INFORMATION IN THIS PUBLICATION IS BELIEVED TO BE ACCURATE, NEITHER NRECA NOR CORNICE ENGINEERING, INC. MAKE ANY REPRESENTATION OR WARRANTY, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. NEITHER NRECA NOR CORNICE ENGINEERING, INC. SHALL BE LIABLE FOR ERRORS CONTAINED HEREIN OR FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH THE FURNISHING, PERFORMANCE, OR USE OF THIS MATERIAL. THE INFORMATION CONTAINED IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE.

Prepared by:
Cornice Engineering, Inc.
P.O. Box 2350
Pagosa Springs, Colorado 81147-2350
(970) 264-6121
(970) 264-6124 (Fax)
E-mail: gmcnaughton@frontier.net
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>MultiSpeak Basics</th>
<th>Section 2</th>
<th>About MultiSpeak Version 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1. What is MultiSpeak?</td>
<td>11. How has MultiSpeak 2 changed from MultiSpeak Version 1.1?</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2. How was MultiSpeak developed?</td>
<td>12. Is MultiSpeak 2 backward compatible with MultiSpeak Version 1.1?</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3. What does MultiSpeak do?</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>4. What software applications does MultiSpeak cover?</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>5. What are MultiSpeak defined functions and how do they relate to my software products?</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>6. Is MultiSpeak a “plug-and-play” solution? Doesn’t this take care of all of my integration needs?</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>7. Is MultiSpeak a product?</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>8. How do I get MultiSpeak?</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>9. What does MultiSpeak cost?</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>10. Why is MultiSpeak important?</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Section 3</th>
<th>MultiSpeak Compliance</th>
</tr>
</thead>
<tbody>
<tr>
<td>13. What does it mean if a software product is “MultiSpeak compliant”?</td>
<td>11</td>
</tr>
<tr>
<td>14. How can I tell if my software application is MultiSpeak compliant?</td>
<td>11</td>
</tr>
<tr>
<td>15. How can I tell if two MultiSpeak-compliant applications will work together?</td>
<td>12</td>
</tr>
<tr>
<td>16. I have an existing software product that is MultiSpeak compliant, how can I determine whether potential applications will work with it?</td>
<td>13</td>
</tr>
<tr>
<td>17. Why are there are three different communications transfer options; doesn’t that make things more complicated?</td>
<td>1-14</td>
</tr>
<tr>
<td>18. Will compliance testing ensure that two software applications work together seamlessly at my utility?</td>
<td>1-14</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Section 4</th>
<th>Using MultiSpeak</th>
</tr>
</thead>
<tbody>
<tr>
<td>19. Is any hardware required to implement MultiSpeak?</td>
<td>17</td>
</tr>
<tr>
<td>20. What is a “MultiSpeak Participating Vendor”?</td>
<td>18</td>
</tr>
<tr>
<td>21. What is a “MultiSpeak Software Developer”?</td>
<td>18</td>
</tr>
<tr>
<td>22. The type of software product that I want to integrate is not covered by MultiSpeak; what can I do?</td>
<td>19</td>
</tr>
<tr>
<td>23. How does MultiSpeak evolve?</td>
<td>19</td>
</tr>
<tr>
<td>24. What else can I do to further the development of MultiSpeak?</td>
<td>19</td>
</tr>
</tbody>
</table>

<p>| Section 5 | For More Information |</p>
<table>
<thead>
<tr>
<th>Appendix A</th>
<th>MultiSpeak Functions and Software Applications</th>
<th>23</th>
</tr>
</thead>
<tbody>
<tr>
<td>Appendix B</td>
<td>MultiSpeak-Defined Interfaces</td>
<td>27</td>
</tr>
<tr>
<td>Appendix C</td>
<td>Supported Communications Transfer Options</td>
<td>33</td>
</tr>
<tr>
<td>Appendix D</td>
<td>How to Read a MultiSpeak Compliance Label</td>
<td>37</td>
</tr>
</tbody>
</table>
Purpose of this Document

The MultiSpeak Initiative is a collaborative effort between the National Rural Electric Cooperative Association and software vendors serving the electric utility industry. The initiative has developed a specification that defines standard data interfaces between software applications commonly used by small electric utilities.

The MultiSpeak Specification consists of the following documents:

1. MultiSpeak® Version 2.1 Specification, a document that more fully describes how the specification should be applied.
2. MultiSpeak.xsd, MultiSpeakMessaging.xsd, and mspGeometry.xsd, Extensible Markup Language (XML), schemas that implement the specification.
4. MultiSpeak® Version 2 File-Based Transfer Implementation Guidelines, a document giving additional information about implementation of MultiSpeak interfaces using file transfers.

This specification guide has been developed for utility personnel wishing to better understand the MultiSpeak® Version 2 specification and its role in improving integration among existing and planned software applications. The guide is intended to provide the information most often needed by utility personnel to determine how to specify MultiSpeak-compliant software applications as well as to evaluate the importance of MultiSpeak compliance in integrating automation systems. The guide is written in a question and answer format.
1. What is MultiSpeak?

MultiSpeak® is a specification for the exchange of data among software applications commonly applied in small electric utilities, such as electric cooperatives. The foundation of the specification is an agreement on the details of the data objects that need to be exchanged to more fully integrate disparate software applications.

The MultiSpeak specification is intended to help vendors to develop interfaces that enable software products from a variety of vendors to interoperate without the need for extensive custom interface development. As a result, it is expected that MultiSpeak will help make available cost-effective, integrated software applications to serve the business needs of utilities.

2. How was MultiSpeak developed?

MultiSpeak was developed by the MultiSpeak Initiative, a collaborative effort between the National Rural Electric Cooperative Association and software vendors serving the electric utility industry. Vendors, along with consultants hired by NRECA, have met over 15 times since October 1999 to design the required software interfaces. Significant funding for MultiSpeak has been provided by the Cooperative Research Network of NRECA. The vendor community has also provided substantial funding in the form of staff time, travel, and other development costs.

3. What does MultiSpeak do?

MultiSpeak defines data objects and specifies message structures so that vendors can write a single, common interface that facilitates communication with another type of software. The specification is designed so that the details of an interface should not vary substantially, regardless of the vendor of the application with which their product is to integrate. Similarly, changes in the structure of one vendor’s software should not require changes in the agreed upon interface. Since the participating vendors support a variety of hardware and software platforms, database
programs, and programming languages, the Initiative has developed an approach that is independent of platform or database structure, one that relies on common data definitions and agreed-upon data flows.

This approach does limit somewhat the means that can be used to achieve integration. Any scheme that requires native database access, such as structured query language (SQL) or open database connectivity (ODBC), will not achieve the goals of the group. The approach used in the specification is for each vendor to write and maintain an interface that marshals the required information from its native data structure, converts those data into extensible markup language (XML) data packets, and transports those packets in the form of predefined messages. The receiving application is responsible for unmarshaling the message, parsing the XML, and taking the appropriate action with the resulting data objects. Figure 1.1 illustrates this approach schematically for a customer information system (CIS) and a geographic information system (GIS) that exchange information from their native databases.

![Figure 1.1: MultiSpeak approach to common interfacing.](image)

### 4. What software applications does MultiSpeak cover?

The MultiSpeak Initiative initially focused on five back-office software applications: (i) customer information systems (CIS), (ii) geographic information systems (GIS), (iii) engineering analysis (EA), (iv) interactive voice response systems (IVR), and (v) automated staking. Version 1.1 of the MultiSpeak Specification defined seven interfaces and developed a data dictionary for the information that could meaningfully be exchanged among these applications. Figure 1.2 illustrates the applications, data flows and interfaces defined in the MultiSpeak Version 1.1 specification. Only batch file transfers were defined in Version 1.1. In addition, no provisions were made for incremental updates or for deletion of data.
MultiSpeak Version 2 (MultiSpeak 2) extends the coverage of the specification to include (i) supervisory control and data acquisition (SCADA), (ii) automated meter reading (AMR), (iii) outage management, (iv) load management (LM), and (v) customer relationship management (CRM) systems. MultiSpeak carries forward provisions for batch transfers where such transfers make sense to support utility business processes, but also includes a messaging framework to support near-real-time integration. Both the batch and real-time messaging capabilities can support incremental updates and deletions with a high degree of specificity. Figure 1.3 shows the MultiSpeak defined software functions that are covered by MultiSpeak Version 2 and the interfaces between them. The interfaces are identified on Figure 1.3 by numbers in ovals and are defined in greater detail in Appendix B.
FIGURE 1.3: MultiSpeak Version 2 process model.
5. What are MultiSpeak-defined functions and how do they relate to my software products?

Unlike MultiSpeak Version 1.1, MultiSpeak 2 is defined in terms of software functions (that is to say theoretical capabilities) rather than software applications (actual software products). Although this distinction may be initially confusing, it is necessary to address two problems that have arisen in the process of determining what data interchanges MultiSpeak should support: (i) different applications of the same type (e.g., automatic meter reading packages) support different sets of capabilities and (ii) the same capability (e.g., the ability to detect and report an outage condition) may be provided by multiple software application types. For a more detailed description of MultiSpeak-defined functions and how they relate to actual software products, see Appendix A.

6. Is MultiSpeak a “plug-and-play” solution? Doesn’t this take care of all of my integration needs?

Unfortunately, the requirements for integration of software applications in the utility industry are just too complicated and diverse for any single specification to provide a universal “plug-and-play” solution for every utility. The MultiSpeak specification has been designed from the ground up to address the majority of the common integration needs of small utilities; it isn’t guaranteed to solve all of your problems. You may still need to have your vendors create special interfaces or fine-tune their MultiSpeak interfaces to meet the specific needs of your utility, especially if you have any of the following circumstances:

- Not all of your software products support MultiSpeak 2.
- Your MultiSpeak-compliant software applications do not support all of the defined interfaces (see Appendices B and D for more information on this issue).
- Software programs that need to exchange data use significantly different versions of MultiSpeak or send their data using different communications transfer options (see Question 16 and Appendix C for more information on this issue).
- You want to send data that are not covered by MultiSpeak.

Your specific utility may also have made choices during implementation of your software solutions that make your particular installation nonstandard; such choices may also affect the interoperability of MultiSpeak interfaces. Specific circumstances that may affect the performance of MultiSpeak interfaces at your utility include:

- Use of nonstandard data fields to store critical data
- Poor-quality data or the use of unusually formatted data, especially if such data are used as an index or key value that creates the link between two different applications.
7. Is MultiSpeak a product?

**MultiSpeak is a specification that defines software interfaces; it is not a software product.** Software vendors may use MultiSpeak to develop compliant interfaces for their existing, or planned, software products.

8. How do I get MultiSpeak?

Utilities may obtain MultiSpeak-compliant interfaces directly from their current software vendors. If one of their current software vendors does not provide MultiSpeak-compliant interfaces, the utility may use the MultiSpeak specification to write its own compliant interface for that product or get a third party to write such an interface. The MultiSpeak specification is freely available to all interested parties from the MultiSpeak Initiative web site (www.multispeak.org).

9. What does MultiSpeak cost?

Users should be aware that vendors have invested significant time, energy, and capital in the development of MultiSpeak-compliant interfaces and most must recover these investments in some manner. Some vendors may choose to offer MultiSpeak interfaces as part of their core software package and recover the costs from all users; others may offer MultiSpeak interfaces as an add-on component and recover the costs for such add-ons from the specific users that request them. However, it should be noted that, as the vendors serving the electric utility industry increasingly adopt MultiSpeak-compliant interfaces, the overall cost of interoperable software should go down.

10. Why is MultiSpeak important?

MultiSpeak is important to both the user and the vendor communities.

For utilities, MultiSpeak offers the following advantages:

- Minimizes the need for expensive and maintenance-intensive custom interfaces
- Enables utilities to focus on “best-of-breed” software without the fear of lack of integration
- Reduces the deployment time and risk in purchasing new automation applications
- Reduces the cost and hassle of changing from one application to another of the same type or upgrading to a new version of the same type of software
For **software vendors**, MultiSpeak has the following advantages:

- Minimizes the need for developing multiple interfaces with other vendors’ software
- Releases programmers to improve existing products or develop new offerings
- Reduces complexity and time to market for new products
- Reduces support headaches
- Utilities will be more willing to invest in new applications if they have assurance that integration problems are minimized
11. How has MultiSpeak 2 changed from MultiSpeak Version 1.1?

MultiSpeak 2 is significantly improved and expanded over Version 1.1 in nearly all regards.

- It carries forward the interfaces that were defined in Version 1.1 and adds support for SCADA, AMR, outage management, load management, and the Key Accounts Database, which was developed by the Cooperative Research Network of NRECA, a type of customer relationship management software. Even the interfaces that were included in Version 1.1 have been significantly improved and made more comprehensive in Version 2.
- It continues support for file-based (batch) transfers, but adds support for real-time integration using a messaging framework. Messages can be sent using either (i) Simple Object Access Protocol (SOAP), a web-based means to send packets of XML data, or (ii) TCP/IP sockets, a robust means for programs to talk directly with each other.
- It provides support for incremental updates and deletions, capabilities missing in Version 1.1.
- Compliance testing will be based on a more comprehensive set of testing data in Version 2 than was the case in Version 1.1. The MultiSpeak 2 testing suite will be designed to replicate actual business processes in a utility, instead of requiring a minimal set of data, as was the case with Version 1.1.
- It makes use of a more flexible technology, XML schemas, for defining the data objects and message structures than the document type definition (DTD) files used in Version 1.1. The use of XML schemas makes it much easier for sets of vendors to expand the capabilities of MultiSpeak 2 to meet the specific needs of their utility customers.
- It makes use of a vendor-independent means to exchange graphical data, called Geometry Markup Language (GML). GML, a standard of the OpenGIS Consortium, Inc. (OGC), makes it possible for MultiSpeak interfaces to move graphics data in the form of XML text, without the need to resort to proprietary graphics exchange files.
12. Is MultiSpeak 2 backward compatible with MultiSpeak Version 1.1?

Unfortunately, because of the comprehensive nature of the improvements in Version 2, it will be difficult for vendors to ensure backward compatibility with interfaces that have been tested compliant with Version 1.1. The MultiSpeak Initiative has requested, but not required, that vendors consider providing some form of compatibility in their MultiSpeak 2 products with the earlier version, where the nature of the changes in the interface permits. Some interfaces, for instance staking-GIS, have changed so significantly that compatibility with Version 1.1 will not be possible. Utilities should discuss with their software vendors what backward compatibility features are available in specific software products of interest.
13. What does it mean if a software product is “MultiSpeak compliant”? 

The NRECA has sponsored an independent testing organization to evaluate compliance with MultiSpeak requirements and a testing protocol has been developed. The testing organization will certify that products submitted for test meet the specification.

Certification is for products, not vendors; certification is by interface. Thus a particular product is deemed to be, for example, compliant with the requirements of the customer billing-geographic information system (CB-GIS) interface. If the submitted product passes the certification test for at least one defined interface, it is declared to be compliant. **Note that the MultiSpeak-compliant designation does not ensure that the product is compliant with every applicable interface.** For instance, as shown in Figure 1.3, seven possible interfaces with a GIS are defined in MultiSpeak [the interfaces with staking (21), EA (22), OA (7), CB (16) SCADA (24), LM (30), and SGV (12)]; a GIS product could be designated as MultiSpeak compliant if it had passed testing for just one of these possible seven interfaces. The interfaces for which a specific product has passed testing are clearly identified in a MultiSpeak product compliance label (See Appendix D). Recognizing compliance on an interface-by-interface basis permits vendors to submit interfaces for test as they become available instead of waiting for all interfaces to be ready for test.

14. How can I tell if my software application is MultiSpeak compliant?

The best way to determine if a product is compliant, and the extent of that compliance, is to check the MultiSpeak Initiative web site ([www.multispeak.org](http://www.multispeak.org)).

It is updated regularly with the most recent product compliance status. You can be assured that if the product is listed as compliant on this web site, it has passed the necessary compliance testing.
15. How can I tell if two MultiSpeak-compliant applications will work together?

Users need to answer the following questions to determine if two software applications will exchange information in a MultiSpeak-compliant manner:

- Are both software packages compliant with the same version of MultiSpeak?
- Are the necessary MultiSpeak functions supported in each software package?
- Do the two packages support a mutual interface?
- Do they share a mutual communications transfer option (file-based, SOAP, or sockets)?

The MultiSpeak Initiative has developed a compliance label for each product that has passed compliance testing for Version 2 interfaces. The label communicates all of the information necessary to answer these questions. Appendix D gives additional information on how to read the MultiSpeak 2 compliance labels and gives examples of hypothetical compliance labels. The most recent information on compliance can be found on the MultiSpeak Initiative web site (www.multispeak.org).
16. I have an existing software product that is MultiSpeak compliant, how can I determine whether potential applications will work with it?

You need to complete the following steps:

**Step 1.** Determine which MultiSpeak functions are required. (See Figure 1.3 and Appendix A).

**Step 2.** Identify the required interfaces. (See Figure 1.3 and Appendix B).

**Step 3.** Identify which software products of the type you wish to apply are MultiSpeak compliant (see the list of compliant products on the MultiSpeak web site, www.multispeak.org).

**Step 4.** Confirm that compliant interfaces exist for the products that need to exchange data (download the product compliance labels for both products).

**Step 5.** Confirm that the products support a mutual communications transfer option using the compliance labels.

---

**Example**

You have an existing MultiSpeak-compliant customer information system (CIS) application, and you want to find an automated staking software product that exchanges data with it using a MultiSpeak compliant interface.

**Step 1.** In order for a CIS application to integrate with an automated staking program, the CIS must support the MultiSpeak finance and accounting (FA) function and the automated staking application must support the MultiSpeak staking function.

**Step 2.** The required interface is no. 11 (see Figure 1.3).

**Step 3.** After checking the web site you find that there are two possible compliant automated staking products.

**Step 4.** The product compliance labels for the two staking systems show that only one has tested compliant for interface 11. Continue to analyze this product.

**Step 5.** The product compliance labels for your CIS and the staking product you chose in Step 4 show that the products have both tested compliant by the SOAP communications transfer option. Hence these products should exchange data using MultiSpeak with SOAP. See Appendix D for more information on how to make this determination.
17. Why are there three different communications transfer options; doesn’t that make things more complicated?

MultiSpeak vendors make use of a wide variety of databases, software and hardware platforms, and programming languages. In order for the specification to be widely supported and available on all of the possible combinations of platforms, it is desirable to offer different options for transporting messages. However, this flexibility creates additional complication for a utility interested in specifying MultiSpeak-compliant applications. It is important that potential users of MultiSpeak 2-compliant applications understand the different kinds of communications transfer options that may be used and whether their software applications are compliant using the desired communications transfer option. See Appendix C for more information on the different potential communications options, and Appendix D for more information on how to determine, using the MultiSpeak compliance label, whether the desired communications transfer option is supported by a specific piece of software.

18. Will compliance testing ensure that two software applications work together seamlessly at my utility?

For a number of reasons, such as those listed in Question 6, compliance testing cannot provide complete assurance that two software applications will work together seamlessly at your specific installation.

Note that testing can be performed at three levels. Successive levels of testing provide increasing assurance that two applications will interoperate seamlessly.

**Level 1: Compliance testing**

Testing performed to assess whether the software meets the requirements of the specification. This is the type of testing that is performed by the NRECA-approved MultiSpeak testing laboratory on products submitted for MultiSpeak compliance. Compliance testing is the foundation for creating interoperable software.

**Level 2: Interoperability testing**

Tests performed between pairs of compliant software applications to ensure that they work seamlessly together. It is possible that two software applications could both be compliant and yet not work seamlessly together without minor modifications. *This type of testing is not performed by the MultiSpeak Initiative unless pairs of vendors specifically request, and pay for, such testing.*

**Level 3: Site conformance testing**

Tests performed at an individual utility site to ensure that pairs of products interoperate seamlessly given the utility’s specific data structures and application data. Even if two software applications have passed both compliance and interoperability testing, it is possible that choices of data structure made by a specific utility or poor quality data stored in one of the applications...
may result in interface errors. Such errors may require work on the part of the software vendors
or data scrubbing on the part of the utility. **If it is absolutely required that two software appli-
cations work together without error at a specific utility, site conformance testing is the only
means to ensure that result.** Utilities may request that an independent party perform site con-
formance testing. Utilities may wish to require site conformance testing as an acceptance criteri-
on when purchasing new software products.
19. Is any hardware required to implement MultiSpeak?

No special hardware is required to implement MultiSpeak-compliant interfaces.

The interface specification has been designed to enable compliant applications to exchange data directly, using all of the potential communications options without the need for a special integration server. However, some utilities may wish to implement MultiSpeak with a MultiSpeak-compatible integration server that provides enhanced messaging and application functionality not available without the addition of such equipment. Capabilities that may be available in a MultiSpeak-compatible integration server include:

- Message logging, auditing, and historical archiving.
- Message queuing, persistence, and assured delivery.
- Conversion of messages between incompatible communications transfer options (e.g., SOAP and sockets) where no match of transfer options exists for a pair of compliant software applications.
- Legacy adapters (i.e., custom interfaces with noncompliant applications).
- Business process management (for instance, automatically updating a GIS map and a CIS record as a single logical transaction).

Third-party vendors may provide MultiSpeak-compliant integration servers, although neither NRECA nor MultiSpeak provide such hardware.
20. What is a “MultiSpeak Participating Vendor”?

A MultiSpeak participating vendor is a company that supports the goals of the MultiSpeak Initiative. A participating vendor may, or may not, offer MultiSpeak-compliant products. Vendors that participate in the MultiSpeak Initiative may be identified by the following black and white logo:

![MultiSpeak Participating Vendor Logo]

21. What is a “MultiSpeak Software Developer”?

A MultiSpeak Software Developer is a company that provides software products that have been tested compliant to the MultiSpeak Version 2 specification. Just as compliant products can be identified by the use of a color “MultiSpeak Compliant” logo (see Question 14), a vendor of Version 2-compliant products may be identified by the use of a color “MultiSpeak Software Developer” logo, shown below:

![MultiSpeak Software Developer Logo]
22. The type of software product that I want to integrate is not covered by MultiSpeak; what can I do?

The MultiSpeak specification is intended to be a living document that is modified as necessary to meet the most common needs of small electric utilities. If there are common application types that are not covered by the specification—and for which interfaces with other types of software would be beneficial—utilities or vendors should suggest to the MultiSpeak User’s Group, to NRECA, or to the MultiSpeak Project Coordinator that these applications be added to the specification in the near future. For contact information see the Section 5 of this document, “For More Information.”

23. How does MultiSpeak evolve?

The MultiSpeak specification is a collaborative effort. Representatives of NRECA and software vendors meet as required (usually about 3–4 times each year) to address extensions, corrections, or enhancements to the specification. Vendors may bring unsolicited proposals to the Initiative at any regular meeting for changes to the specification. When a consensus has been reached on the topics of discussion, the required changes are reflected in the specification.

24. What else can I do to further the development of MultiSpeak?

There are three things that a utility can do to further the development of the MultiSpeak specification:

- The best way to influence the future direction of the specification is to attend the MultiSpeak User’s Group meeting. Input is solicited at such meetings from both the vendor and user communities for enhancements to the specification. Additional information about the User’s Group meetings may be found on the MultiSpeak Initiative web site (www.multispeak.org).
- Consider the benefits of MultiSpeak compliance when purchasing new or upgraded application software.
- Since vendors have limited development funds, and target those funds to the goals of their users, it is important that utilities communicate with vendors the importance of MultiSpeak compliance in vendor offerings.
If the information included in this specification guide doesn’t address your questions, you may obtain additional information from the following sources:

**MultiSpeak web site**
The MultiSpeak Initiative web site (www.multispeak.org) is the best source of up-to-date information about the MultiSpeak Initiative, the MultiSpeak specification, and the status of compliance of vendor software.

**MultiSpeak specification**
For all the details about how MultiSpeak interfaces work and what data items and software interfaces are covered, see the detailed specification. All portions of the specification are available from the MultiSpeak web site (www.multispeak.org).

**NRECA MultiSpeak Project Manager**
Martin E. Gordon, P.E.; phone 703 907 5840; e-mail meg0@nreca.org.

**MultiSpeak Project Coordinator**
Gary A. McNaughton, P.E.; phone 970 264 6121; e-mail info@multispeak.org.
The interfaces established in the MultiSpeak specification are defined on the basis of information flows between software functions, not individual software applications. This was done in order (1) to give flexibility to model a number of different product types and (2) to minimize duplication in the specification.

The first goal was to give flexibility to model a number of different product types in the specification. An example that illustrates this need for flexibility is the automated meter reading application. Some AMR systems have the capability to support two-way communications between the AMR server and a meter, and hence can query individual meters and implement service connect and disconnect remotely. AMR systems using one-way communications technologies usually cannot support such functions. However, both systems support meter reading functionality. Figure A.1 shows examples of AMR systems that support different combinations of software functions. In the figure, AMR Product A supports only the meter reading software function, whereas AMR Product B supports the meter reading, load profile, and remote connect and disconnect software functions. Thus MultiSpeak was defined to enable vendors to choose which software functions they wish to support in their applications and at the same time to give the utility software user a clear and unbiased means to evaluate the levels of compliance of competing products.
A second objective was to eliminate duplication of interface definitions. An example of this situation is detection of outages. Detected outages can be logged by using (1) a database application employed by customer service representatives to enter information taken from customer phone calls, (2) an interactive voice response system that logs customer phone calls, (3) meters reporting to an AMR system, or (4) a system that deploys dedicated outage detection modules installed at the customer location. Regardless of the source of the outage notification, the data that need to be communicated with another software application are identical. Hence, it would lead to redundant interface definitions if each of these systems had its own unique interface.

Table A.1 describes the software functions currently defined by MultiSpeak. Each function is referred to by a function name; for instance CB is the function name for the customer billing database function.

### TABLE A.1: MultiSpeak Functions and Function Definitions

<table>
<thead>
<tr>
<th>FUNCTION</th>
<th>FUNCTION DEFINITIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>CB</td>
<td><strong>Customer Billing Database.</strong> This function includes a customer information database, customer billing, and accounting for electrical usage.</td>
</tr>
<tr>
<td>CD</td>
<td><strong>Connect/Disconnect/Power Limitation.</strong> This function controls remote switches or power-limiting devices installed at customer services.</td>
</tr>
<tr>
<td>CH</td>
<td><strong>Call Handling.</strong> This function manages incoming and outgoing calls.</td>
</tr>
<tr>
<td>CRM</td>
<td><strong>Customer Relationship Management.</strong> This function tracks customer care activities for individual end-use customers.</td>
</tr>
<tr>
<td>DGV</td>
<td><strong>Dynamic GIS Viewer.</strong> This function accepts and displays dynamically changing data in a geographic context. It is the intention of a dynamic GIS viewer to display changes in the status of data with no intentional time delay. Examples of this function are automatic vehicle location and near-real-time display of outages.</td>
</tr>
<tr>
<td>EA</td>
<td><strong>Engineering Analysis.</strong> EA is treated as a single function, although it typically includes all engineering analysis functions, such as voltage drop and fault study capabilities.</td>
</tr>
<tr>
<td>FA</td>
<td><strong>Finance and Accounting.</strong> This function includes the corporate accounting, accounting for time and materials, and work order accounting.</td>
</tr>
<tr>
<td>GIS</td>
<td><strong>Geographic Information System.</strong> MultiSpeak treats GIS as a single function, although a GIS typically includes map editing/creation tools, printing tools, and database editing/query tools. Any given vendor’s GIS application may also include the capabilities of the dynamic GIS viewer function.</td>
</tr>
<tr>
<td>LM</td>
<td><strong>Load Management.</strong> The load management function accepts information about required power system curtailments and manages those curtailments by communicating with remote devices, such as load control switches.</td>
</tr>
</tbody>
</table>
### TABLE A.1: MultiSpeak Functions and Function Definitions (Continued)

<table>
<thead>
<tr>
<th>FUNCTION</th>
<th>FUNCTION DEFINITIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>LP</strong></td>
<td><strong>Load Profile.</strong> The load profile function exchanges information about how metered loads change over time by saving information about such loads on a periodic basis. The load profile function is considered to be an advanced function. Support for this function is not required for compliance with the MultiSpeak specification. It is recommended, however, if support for load profile is included in an AMR product, that interfaces with the load profile function be compliant with the MultiSpeak definitions.</td>
</tr>
<tr>
<td><strong>MR</strong></td>
<td><strong>Meter Reading.</strong> This function collects information from remote meters, typically revenue meters, and presents it to other systems for analysis.</td>
</tr>
<tr>
<td><strong>OA</strong></td>
<td><strong>Outage Analysis.</strong> The outage analysis function accepts outage information from Outage Detection (OD) sources. Such information is used to: (1) assist a human dispatcher to determine which power system devices have likely operated to create the observed pattern of outages, and (2) facilitate outage reporting.</td>
</tr>
<tr>
<td><strong>OD</strong></td>
<td><strong>Outage Detection.</strong> The outage detection function is broadly defined for the purposes of MultiSpeak; it includes any means by which information about outages is stored on a computerized server. Information for such systems may be collected by any of the following: (1) humans taking customer calls and entering the applicable information, (2) an IVR that collects outage information, (3) automated outage detection devices, or (4) an automated meter reading system.</td>
</tr>
<tr>
<td><strong>SGV</strong></td>
<td><strong>Static GIS Viewer.</strong> This function accepts and displays information in a geographic context. This function is used to display information that is not rapidly changing; hence real-time response is not necessary.</td>
</tr>
<tr>
<td><strong>SCADA</strong></td>
<td><strong>Supervisory Control and Data Acquisition.</strong> SCADA provides status detection, logging of analog information, and control of remote power system equipment.</td>
</tr>
<tr>
<td><strong>Staking</strong></td>
<td><strong>Automated Staking.</strong> This function provides field design and cost estimation capabilities.</td>
</tr>
</tbody>
</table>
Table A.2 includes six common software applications and the MultiSpeak functions that might be supported by those applications.

**TABLE A.2: Examples of Multifunction Applications**

<table>
<thead>
<tr>
<th>APPLICATION</th>
<th>FUNCTIONS POTENTIALLY INCORPORATED IN THIS APPLICATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMR</td>
<td>Automated Meter Reading System. An AMR consists of at least the meter reading (MR) function, but may also include connect/disconnect/power limitation (CD), outage detection (OD), and load profile (LP) functions.</td>
</tr>
<tr>
<td>CIS</td>
<td>Customer Information System. As applied in most cooperatives, the CIS consists of a suite of applications, typically provided and integrated by a single vendor. Usually included are the customer billing (CB) and finance and accounting (FA) functions defined above, however, an actual CIS in a co-op also may include outage detection (OD) (via an application to enter customer calls).</td>
</tr>
<tr>
<td>GIS</td>
<td>Geographic Information System. A GIS will include the GIS MultiSpeak function, but may also include the static GIS viewer (SGV) and/or dynamic GIS viewer (DGV) functions.</td>
</tr>
<tr>
<td>GIS Viewer</td>
<td>GIS Viewer. It is assumed that all GIS viewer applications will contain a static GIS viewer (SGV) function. They may also contain a dynamic GIS viewer (DGV) function if they are designed to reflect information that changes in response to real-time data input.</td>
</tr>
<tr>
<td>IVR</td>
<td>Interactive Voice Response. An interactive voice response system includes, at a minimum, the call handling (CH) function, but may also include the outage detection (OD) and outage analysis (OA) functions.</td>
</tr>
<tr>
<td>OM</td>
<td>Outage Management. Some outage management systems include both the outage analysis (OA) and outage detection (OD) functions, but some only supply one of these functions.</td>
</tr>
</tbody>
</table>

Compliance with the MultiSpeak specification is based on functions rather than application to allow for differences in bundling of functions in specific commercial products.
The MultiSpeak specification is based on a model of data ownerships and data exchanges that is intended to implement common business processes in small electric utilities. These relationships are outlined in Table B.1. The first column of Table B.1 (“Data Flow Number”) indicates a data flow (or interface) number. These interfaces are also illustrated in Figure B.1. The next column (“Linked Functions”) indicates which two functions are linked on this interface. The third column (“Data Exchanged”) summarizes the types of information that are to be exchanged. The column entitled “Direction” establishes the ownership of the data, that is to say, which function is the primary repository of the associated data and has the ultimate responsibility for its accuracy in the context of this defined interface. For instance, for interface 1, the owner of the customer data to be exchanged is the customer billing (CB) function and the consumer of the data is the load management (LM) function. This is implied by the designation “CB>LM” in the “Direction” column. The final column (“Communications Requirements”) outlines the modes of communication that are defined for each interface. Three modes are used, batch (B), request/response (RR), and publish/subscribe (PS).

**Batch communication** implies that the function that owns the data sends information periodically in groups. In this mode there usually is a delay between the time that new information is available and when it is sent. It should be noted that even batch messaging could appear to an end user be in “real time” if the messages are kept short and are sent sufficiently frequently. The frequency of batch messages to implement the defined interfaces is left entirely up to each vendor, but it is recommended that messages be kept small so as to enhance performance and not unduly burden the receiving application. An exception to this recommendation is a data dump, when all known information is sent in order to initially set up the database in the receiving application.

**Request/response communication** implies that a client (data consumer) requests that a specific action be taken by the server (data provider); the server responds with a message outlining the results of that action. For instance, the client could request the current contents of a specific data item to be sent by the server. The server would respond with the required information or an appropriate error message. Request/response messaging is often used by the client in lieu of maintaining a separate copy of the subject data.
Publish/subscribe communication implies that the server (data provider) makes available (or publishes) data stored in its system to one or more clients, which subscribe to such data. Although publish/subscribe systems are often implemented using message queue middleware that supports point-to-multipoint messaging, this is not a requirement of MultiSpeak. The cost and complexity of messaging middleware may be prohibitive in the small utility environment. Hence, MultiSpeak requires that a software function that acts as a publisher maintain a list of subscribers and send multiple point-to-point messages to implement the equivalent functionality that would be offered by the message queue middleware. If a publishing application wishes to accomplish the required functionality using middleware, it is permissible, provided the required messages are sent in accordance with the specification and the middleware is bundled with the software product when sold.

Note that each box in the process model (Figure B.1) represents a software function (as defined in Table A.1), not a software application or product. Any specific product may encompass multiple functions and hence full compliance with the MultiSpeak specification for that product would require support for all of the defined interfaces for all of the software functions implemented. The numbers listed in ovals are the defined interfaces shown in Table B.1. Shaded ovals indicate interfaces that were defined previously in the MultiSpeak Version 1.1 specification. Unshaded ovals denote interfaces that are new in this version. The designation “optional” in either Table B.1 or Figure B.1 indicates that the associated interface is not required for compliance with the MultiSpeak specification. Although support for an optional interface is not required, it is recommended that if such an interface is implemented, it be done in the defined manner.
<table>
<thead>
<tr>
<th>DATA FLOW NUMBER</th>
<th>LINKED FUNCTIONS</th>
<th>DATA EXchanged</th>
<th>DIRECTION</th>
<th>COMMUNICATIONS REQUIREMENTS*</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (optional)</td>
<td>CB-LM</td>
<td>Customer information</td>
<td>CB-LM</td>
<td>B or PS</td>
</tr>
<tr>
<td>2A</td>
<td>CB-MR</td>
<td>• Meter readings/characteristics, service status, phasing • Customer, service, meter, and location info</td>
<td>MR&gt;CB CB&gt;MR</td>
<td>B, PS, or RR</td>
</tr>
<tr>
<td>2B</td>
<td>CB-CD</td>
<td>• Connect/disconnect and power limitation commands; customer, meter, service • Acknowledge connect/disconnect, power limitation commands</td>
<td>CB&gt;CD CD&gt;CB</td>
<td>B, PS, or RR</td>
</tr>
<tr>
<td>2C (optional)</td>
<td>LP-CB</td>
<td>• Load profile data</td>
<td>LP&gt;CB</td>
<td>B or PS</td>
</tr>
<tr>
<td>3</td>
<td>SCADA-LM</td>
<td>• Load shed schedules/commands, power factor correction signal</td>
<td>SCADA-LM</td>
<td>PS</td>
</tr>
<tr>
<td>4</td>
<td>MR-EA</td>
<td>• Customer metered load, measurement, measurement device, phase (request by EA, respond by MR) • Connectivity (location of meter electrically)</td>
<td>MR&gt;EA EA&gt;MR</td>
<td>RR</td>
</tr>
<tr>
<td>5</td>
<td>OD-OA</td>
<td>• Status change event, device information • Status (request by OA, response by OD)</td>
<td>OD&gt;OA OD&gt;OA</td>
<td>PS RR</td>
</tr>
<tr>
<td>6</td>
<td>CH-OA</td>
<td>• Outage information, status, restoration information, estimated time to restoration (ETOR), call backs • Is this service part of an existing outage? (CH requests, OA responds) • Revised call-back status</td>
<td>OA&gt;CH OA&gt;CH CH&gt;OA</td>
<td>PS RR</td>
</tr>
<tr>
<td>7</td>
<td>GIS-OA</td>
<td>• Connectivity file</td>
<td>GIS&gt;OA</td>
<td>B</td>
</tr>
<tr>
<td>8</td>
<td>EA-SCADA</td>
<td>• Analogs (volts, amperes, megawatts, power factor), device status (request from EA, respond by SCADA), temperature, time • Connectivity</td>
<td>SCADA&gt;EA EA&gt;SCADA</td>
<td>RR B</td>
</tr>
<tr>
<td>9</td>
<td>SCADA-OA</td>
<td>• Device status and transitions, SCADA analogs • Presumed status of downstream devices</td>
<td>SCADA&gt;OA OA&gt;SCADA</td>
<td>PS or RR PS or RR</td>
</tr>
<tr>
<td>10</td>
<td>EA-OA</td>
<td>• Connectivity file</td>
<td>EA&gt;OA</td>
<td>B</td>
</tr>
<tr>
<td>11</td>
<td>FA-Staking</td>
<td>• Work orders, units, and materials • Revised work orders, staked assemblies, pick list (optional), and CPRs (optional)</td>
<td>FA&gt;Staking Staking&gt;FA</td>
<td>B or PS B or PS</td>
</tr>
<tr>
<td>12</td>
<td>GIS-SGV</td>
<td>• Existing facilities information and background graphics, customer, service</td>
<td>GIS&gt;SGV</td>
<td>B (required) PS (optional)</td>
</tr>
</tbody>
</table>

* Communications requirements defined in the specification are: B = batch, PS = publish/subscribe, RR = request/response.
### TABLE B.1: Version 2 MultiSpeak Process Model—Data Exchanges and Communications Requirements (Continued)

<table>
<thead>
<tr>
<th>DATA FLOW NUMBER</th>
<th>LINKED FUNCTIONS</th>
<th>DATA EXchanged</th>
<th>DIRECTION</th>
<th>COMMUNICATIONS REQUIREMENTS*</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td></td>
<td>This interface has been intentionally omitted</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>CB-CH</td>
<td>• Customer, service, network and connect/disconnect, meter</td>
<td>CB&gt;CH</td>
<td>B, PS, or RR</td>
</tr>
<tr>
<td>15</td>
<td>CB-OA</td>
<td>• Customer connect/disconnect, power limitation command&lt;br&gt;• Customer meter data, customer and service information</td>
<td>CB&gt;OA, CB&gt;OA</td>
<td>B or PS, B, PS, or RR</td>
</tr>
<tr>
<td>16</td>
<td>CB-GIS</td>
<td>• Customer, service, network, meter, transformer, pole, joint use, usage&lt;br&gt;• Transformer, pole, joint use, network</td>
<td>CB&gt;GIS, GIS&gt;CB</td>
<td>B, PS, or RR, B, PS, or RR</td>
</tr>
<tr>
<td>17</td>
<td></td>
<td>This interface has been intentionally omitted</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>OA-DGV</td>
<td>• Outages to display</td>
<td>OA&gt;DGV</td>
<td>PS</td>
</tr>
<tr>
<td>19</td>
<td>CB-EA</td>
<td>• Billing account load information&lt;br&gt;• Connectivity</td>
<td>CB&gt;EA, EA&gt;Cb</td>
<td>B or RR, B</td>
</tr>
<tr>
<td>20</td>
<td></td>
<td>This interface has been intentionally omitted</td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>Staking-GIS</td>
<td>• Existing facility information and background graphics&lt;br&gt;• Modified facility information and background graphics</td>
<td>GIS&gt;Staking, Staking&gt;GIS</td>
<td>B or RR, B, PS</td>
</tr>
<tr>
<td>22</td>
<td>EA-GIS</td>
<td>• Connectivity&lt;br&gt;• Load flow and short-circuit analysis results</td>
<td>GIS&gt;EA, EA&gt;GIS</td>
<td>B or RR, B, PS</td>
</tr>
<tr>
<td>23</td>
<td>SCADA-DGV</td>
<td>• Device status and SCADA analogs</td>
<td>SCADA&gt;DGV</td>
<td>PS or RR</td>
</tr>
<tr>
<td>24</td>
<td>GIS-SCADA</td>
<td>• Connectivity and coordinates of nodes</td>
<td>GIS&gt;SCADA</td>
<td>B</td>
</tr>
<tr>
<td>25</td>
<td>CB-OD</td>
<td>• Customer, meter, and service information</td>
<td>CB&gt;OD</td>
<td>B or PS</td>
</tr>
<tr>
<td>26</td>
<td>OD-DGV</td>
<td>• Outage detection events, outage detection devices</td>
<td>OD&gt;DGV</td>
<td>PS</td>
</tr>
<tr>
<td>27 (optional)</td>
<td>OA-MR</td>
<td>• Customers affected by outage, outage events</td>
<td>OA&gt;M</td>
<td>PS</td>
</tr>
<tr>
<td>28</td>
<td>CB-CRM</td>
<td>• Customers, services, transformers, meters, customer calls</td>
<td>CB&gt;CRM</td>
<td>B</td>
</tr>
<tr>
<td>29</td>
<td>FA-CRM</td>
<td>• Work orders</td>
<td>FA&gt;CRM</td>
<td>B</td>
</tr>
<tr>
<td>30</td>
<td>LM-GIS</td>
<td>• Load management devices</td>
<td>LM&gt;GIS</td>
<td>B, PS</td>
</tr>
</tbody>
</table>

* Communications requirements defined in the specification are: B = batch, PS = publish/subscribe, RR = request/response.
FIGURE B.1: MultiSpeak Version 2 process model.
Communications transport options are means by which information may be moved between two MultiSpeak-compliant applications. Three communications transfer options are currently supported by the MultiSpeak Initiative: (1) file based, (2) SOAP, and (3) sockets. The MultiSpeak Initiative supports these three communications transfer options since each option:

- Supports the integration of real-time information flows as necessary to achieve the goals of the MultiSpeak Initiative.
- Provides a solution that will be, to as great an extent as possible, independent of computer platform, operating system, database, or programming language.
- Permits simple, but reliable, implementations.
- Provides for low-cost deployment, without the requirement for expensive or proprietary third-party products.
- Provides for compatibility with other standards efforts, where consistent with the goals of the MultiSpeak Initiative.
- Permits the use of a consistently formatted data payload across the range of supported transport options, where it is possible to do so and is compatible with the other stated goals.

Table C.1 indicates which of the three communications transport options (file based, SOAP, and sockets) supports the three required communications modes defined in Appendix B (batch, request/response, and publish/subscribe).

### TABLE C.1: Communications Modes and Communications Transport Options

<table>
<thead>
<tr>
<th>Communications Mode</th>
<th>File Based</th>
<th>Sockets</th>
<th>SOAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Batch</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Request/Response</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Publish/Subscribe</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
Supported Transfer Options

**File-based communications.** File-based transfer, sometimes referred to as batch communications, occurs when one software application writes data in the form of a file stored in a file space accessible to another application. The receiving software program subsequently reads the data from that file and takes the required actions based on the contents of the file. In the case of MultiSpeak messages, the file contains the data packet that needs to be sent between the two systems.

File-based communications is analogous to an e-mail message. The communicating programs do not have to be talking directly at the same time, and no dedicated connection between them is required.

**SOAP communications.** Simple Object Access Protocol (SOAP) is a means to exchange XML-formatted data using standard Hypertext Transfer Protocol (HTTP) delivered over a TCP/IP network. The MultiSpeak-formatted data packet, like the packet sent using either the file-based or sockets options, is encapsulated in a text wrapper, called a SOAP envelope, and sent with the same technology used to implement web servers. SOAP supports real-time data transfers. It requires both applications to be running and communicating at the same time, but a permanent connection between the two systems is not necessarily required. SOAP is analogous to the communication that occurs between a web browser and a web server.

**Sockets communications.** TCP/IP sockets are a well-established means for two programs to communicate in real time using a persistent connection. The sockets option relies on TCP/IP network protocols to stream the MultiSpeak data packet across this connection. Sockets connections are analogous to telephone conversations; both parties must communicate simultaneously using a connection that is maintained until communication is terminated by one party. The data payload is identical to that used in the SOAP option and nearly identical to that used in the file-based option.

File-Based Versus Real-Time Communications

In the context of the MultiSpeak specification, a data transfer is termed real time if it is completed as soon as possible after a change in data occurs—that is to say, with no intentional time delay. Both the SOAP and sockets transfer options are considered to be real time. On the other hand, in file-based communications, data are typically buffered and sent in batches; hence it is sometimes called “batch” communications.

Although file-based communications is thought by some as being cumbersome or insufficiently frequent to meet modern data exchange requirements, it should be remembered that file-based transfers that are automatically scheduled so that no human intervention is necessary, and that happen sufficiently frequently are indistinguishable from data exchanged in “real time” to the end user of the information. File-based transfer is simple and reliable. It may serve the complete
needs for data exchange for some utilities, and is likely to play a role for some interfaces in most utilities.

This said, there are some business processes, for instance, outage management, that are inconsistent with the time delays inherent in file-based communications. For MultiSpeak interfaces that support such business processes, the real-time transfer options (SOAP or sockets) are more appropriate. Figure C.1 indicates (1) which interfaces are defined in MultiSpeak 2 for both batch and real time communications (those shown in solid lines in Figure C.1) and (2) which will support the underlying business processes only if sent in real time (those denoted by dashed lines), and hence are defined in MultiSpeak 2 for real-time transfers only.

Examples of business processes that may be adequately supported with batch communications include:

- Establishing a units and materials database in an automated staking system from a work order accounting system (MultiSpeak function = FA).
- Establishing an initial geographic display in a SCADA system database from a GIS.
- Updating a laptop GIS viewer from a GIS server.

Examples of business processes that require real-time transfers of data include:

- Displaying actual SCADA device status or outage status in a GIS.
- Intelligent outage and outage call handling, using SCADA device status.
- Maintaining up-to-date outage detection status for outage analysis.
- Performing engineering studies based on actual system conditions and loads.
- Querying AMR units to determine status of a customer service.
- Sending real-time signals to load management switches.
- Locating and dispatching crews.
FIGURE C.1: File-based versus real-time communications.
Users need to answer the following questions to determine if two software applications will exchange information in a MultiSpeak-compliant manner:

1. Are both software packages compliant with the same version of MultiSpeak?
2. Are the necessary MultiSpeak functions supported in each software package?
3. Do the two packages support a mutual interface?
4. Do they share a mutual communications option (file based, SOAP, or sockets)?

The MultiSpeak Initiative has developed a compliance label for each product that has passed compliance testing for Version 2 interfaces. The label communicates all of the information necessary to answer these questions. Figure D.1 illustrates a compliance label for Staking Pro, a hypothetical automated staking product.

1. Are both software packages compliant with the same version of MultiSpeak?

Figure D.2 highlights the location of the MultiSpeak version information, which is necessary to answer this question, as well as information about the software product. The product information includes (i) the product name and version number, (ii) the vendor of the package, (iii) the type of software application the product represents, and (iv) the MultiSpeak functions that are supported by the product.

2. Are the necessary MultiSpeak functions supported in each software package?

Figure D.2 indicates that Staking Pro supports two MultiSpeak defined functions, (i) staking and (ii) static GIS viewer (SGV). Further information about the supported software functions is highlighted in Figure D.3. Figures D.3 and D.4 indicate that the staking function potentially has interfaces with (i) the finance and accounting function (FA) using interface 11 and (ii) the GIS function using interface 21. The reader may refer to Table A.1 for the abbreviations for all of the MultiSpeak-defined functions and to Figure B.1 for the interface numbers of all of the defined interfaces. Similarly, Figure D.3 confirms that the Staking Pro product supports an interface with GIS from its static GIS viewer function (SGV) using interface 12.
### MultiSpeak® Version 2.0 Compliance Label

**Product:** Staking Pro V2.04  
**Vendor:** XYZ, Inc.

**Application Type:** Automated Staking  
**MultiSpeak Functions:** Staking, Static GIS Viewer (SGV)

<table>
<thead>
<tr>
<th>MultiSpeak Functions</th>
<th>Supported Interfaces</th>
</tr>
</thead>
</table>
| Staking              | FA (#11)  
| Static GIS Viewer    | GIS (#12)  

Legend:  
- File Based  
- SOAP  
- Sockets  
- This communication mode is not defined for this interface in MultiSpeak.

**FIGURE D.1:** Sample MultiSpeak compliance label.

---

**MultiSpeak® Version 2.0 Compliance Label**  
**Product:** Staking Pro V2.04  
**Vendor:** XYZ, Inc.

**Application Type:** Automated Staking  
**MultiSpeak Functions:** Staking, Static GIS Viewer (SGV)

<table>
<thead>
<tr>
<th>MultiSpeak Functions</th>
<th>Supported Interfaces</th>
</tr>
</thead>
</table>
| Staking              | FA (#11)  
| GIS (#21)            |  
| Static GIS Viewer    | GIS (#12)  

Legend:  
- File Based  
- SOAP  
- Sockets  
- This communication mode is not defined for this interface in MultiSpeak.

**FIGURE D.2:** Compliance label vendor information.
FIGURE D.3: Supported functions and interfaces.

FIGURE D.4: Staking function interfaces.
Figure D.5 clarifies that the staking function can exchange information with the GIS function using MultiSpeak defined interface 21.

Figure D.6 shows that for each defined interfaces, one or more potential communications options (file-based, sockets, or SOAP) are possible. The boxes associated with each potential communications option are shaded if the software application covered by the compliance label has passed compliance testing for that communications option over that interface. For instance, Figure D.6 indicates that for interface 21 (staking-GIS), all three communications options are defined in the MultiSpeak specification, and the Staking Pro software product has been tested compliant with the file-based and SOAP options, but not the sockets option. Note that for interface 12 (static GIS viewer-GIS), the boxes corresponding to the SOAP and sockets communications options have ghosted icons. This indicates that neither of these communications options is defined in the MultiSpeak specification for interface 12; only the file-based communications option is supported for this interface.
3. Do the two packages support a mutual interface?

Given this information, how can a software user tell if two MultiSpeak-compliant software applications have a potential communications match? Figure D.7 shows the compliance labels for the hypothetical Staking Pro staking application that was used for Figures D.1 through D.6, and a new hypothetical accounting software package, Account Master. It is clear from Figure D.7 that there is a MultiSpeak-defined interface, number 11, which is a potential means to exchange information between these two applications. The next task is to see if the two specific pieces of software are compliant using a common communications transfer option.

4. Do they share a mutual communications option (file based, SOAP, or sockets)?

Figure D.8 shows that, although Staking Pro is compliant on interface 11 using SOAP (SP), Account Master is not; hence this combination will not work. Similarly, Account Master is compliant on interface 11 using sockets (SO), but Staking Pro is not; thus this combination will not work. However, they both are compliant using file-based (FB) transfers on interface 11. Thus MultiSpeak-compliant communications is possible between the two programs using file-based transfers.
FIGURE D.7: Shared interface.

FIGURE D.8: Matched interfaces.

These applications share an interface, but can they talk?

Yes, they can talk, but only using file-based communications.