GRANTS.GOV℠

AGENCY INTEGRATION TOOLKIT DOCUMENT

Systems Integration

Version 3.0

July 9, 2003

Final

Table of Contents

Page

1. Record of Changes **4**
2. Introduction **5**
2.1 Purpose 5
2.2 Scope 6
2.3 Format Of The Agency Integration Toolkit 7
2.4 Summary Of The Agency Technology Survey 7
3. Proposed Integration Solutions **7**
3.1 Proposed Integration Technology: System-to-System Interfaces 8
3.1.1 Description Of Supporting Technology 8
3.1.2 Supported Platforms 10
3.2 Proposed Integration Technology: System-to-Human Interfaces 10
3.2.1 Description Of Supporting Technology 10
3.2.2 Supported Platforms 12
3.3 Tools, Utilities 12
3.3.1 Web Browsers 12
3.3.2 Web Servers 13
3.3.3 Web Services Frameworks 13
3.3.4 Web Application Servers 13
3.3.5 Web Services Integrated Development Environment (IDE) 15
3.3.6 Web Services Toolkits 15
3.3.7 XML Utilities 15
3.3.8 E-Form Utilities 16
3.3.9 File Archive Utilities 16
4. Selecting and Implementing an Integration Solution **17**
4.1 Solution Selection Criteria 17
4.2 System-to-System Interface Integration 18
4.2.1 Assumptions 18
4.2.2 System-to-System Integration Details 19
4.3 System-to-Human Interface Integration 21
4.3.1 Assumptions 21

4.3.2 GUI Interface Usage Details 22
5. Conclusion 27
6. Acronyms 29
Appendix A Web Services and Related Technologies Overview 31
Appendix B Application Retrieval Technical Details Skeleton....39
Appendix C SOAP Message Structure 41
Appendix D Skeleton Application Message 47
Appendix E SF-424 and Related Forms....48
Appendix F Application Schemas....50
Appendix G Data Element Descriptions....51
Appendix H Bibliography/Resources....56

List of Tables

Page

Table 3-1 System-to-System Supporting Technologies 8
Table 3-2 System-to-Human Supporting Technologies 10
Table 4-1 Selection Criteria 17
Table 4-2 Access Instructions....22
Table 4-3 Login Instructions....23
Table 4-4 Retrieve Submitted Applications Instructions....24
Table 4-5 Select Applications for Download Instructions....25
Table 4-6 Logout Instructions....26
Table B -1 Sample Skeleton Format....39
Table B -2 High Level Description of Grants.gov Application Retrieval 39
Table E- 1 SF-424 and Related Forms....48
Table F- 1 Application Schemas....50
Table G- 1 Common Grants.gov Data Element Names....51

List of Figures

Page

Figure 2-1 (2) Sided View of Grant Lifecycle....6
Figure 4-1 Structure of the Application Message 21

Figure 4-2 Grants.gov Welcome Page 22
Figure 4-3 Login Screen 23
Figure 4-4 Grantors only Screen 24
Figure 4-5 Retrieve Submitted Applications Screen 25
Figure 4-6 Logout Selection 26
Figure A- 1 Core Web Services Relationship 34
Figure A- 2 SOAP Message with Binary Attachments 35
Figure C- 1 SOAP Message Structure 42
Figure C- 2 Structure of the Message Header 43
Figure C- 3 Sample Message Header 45
Figure D- 1 Sample Skeleton Application Message 47

1. Record of Changes

Version Number	Date of Change	Made By	Description of Change
1.0	**6/06/2003**	System Integration Team	Original Draft
2.0	6/11/2003	System Integration Team	Table of Content reorganization, addition of Appendices, changes based upon feedback
2.0	6/16/2003	System Integration Team	Changes based upon feedback received from IV&V
2.0	6/27/2003	System Integration Team	SOAP message structure, Appendix.
2.0	7/2/2003	Yvonne Duncan	Formatting – Final
3.0	7/9/2003	Yvonne Duncan	Incorporated additional feedback.

2. Introduction

Grantee-organizations are often faced with overwhelming challenges when applying for federal government grants, regardless of which agency they address. Various program requirements and administrative differences necessitate the constant updating of procedures, and disparate data standards and business processes require redundant data entry, often resulting in inaccurate application submissions. Twenty-six agencies with approximately 900 programs collect grant information. However, as the grant information is typically not shared among the programs, the grantee is often left with a time-consuming follow-up through a multitude of databases, web sites, and telephone calls.

The Grants.gov initiative aims to produce a simple, unified "Storefront" for all customers of Federal grants to electronically find opportunities, apply, and manage grants, as well as facilitate the quality, coordination, effectiveness, and efficiency of operations for grant makers and grant recipients.

The four primary goals[1] of the Grants.gov initiative were defined by consensus among the grant-making agencies and include:

1. Eliminate the burden of redundant or disparate electronic and paper-based data collection requirements.
2. Define and implement simplified standard processes and standard data definitions for Federal grant customer interactions.
3. Protect the confidentiality, availability, and integrity of data.
4. Standardize the collection of financial and progress report data in support of audit and performance measurement activities.

The E-Grants initiative's overall objective is to deploy an electronic grants portal that is a "one-stop shop" for all electronic grants administration activities. Two initiatives have been selected as the first major steps in reaching the overall goal. The first initiative, the "Find" function, allows potential grantees to find all grant opportunities online through a single portal. The second initiative will allow the constituent to apply for any of those grants using the E-Grants Store Front "Apply" functionality.

The E-Grants Store Front initiative is designed to reduce existing inefficiencies with common data standards and computer-to-computer interchange via a model framework for defining and formatting grant applications. Potential grant recipients will receive full service grants administration from a one-stop, full-service electronic portal.

2.1 Purpose

The Agency Integration Toolkit is a key component in the adoption and use of the Grants.gov system by grant-making agencies. Its primary purpose is to aide grant-making agencies in their *selection of* and *preparation for implementing*[2] a Grants.gov-to-Agency integration solution. It provides agency IT and grants-management personnel an introduction to and overview of the technical information needed to interface Grants.gov with their existing or planned grants management systems and processes. It also provides guidance that will assist agency

[1] *http://www.grants.gov/docs/VisionAndGoals.pdf*

[2] *Developers may want to refer to the "Agency Integration Toolkit Design Specification" document, which contains the more technical details of how to design and implement the system-to-system interfaces.*

management personnel in scheduling and budgeting for the IT and process changes needed to support Grants.gov based electronic filing.

2.2 Scope

The Grants.gov Agency Integration Toolkit will describe the interfaces between the Grants.gov "Front Office" operations and an agency's "Back Office" operations; i.e., the interfaces between the grantee-facing and agency-facing aspects of the grant lifecycle. The following figure represents a general, two-sided view of the grant lifecycle that identifies some of the more common parts of the lifecycle that are grantee-facing ("Front Office") as well as those that are solely internal agency processes ("Back Office"):

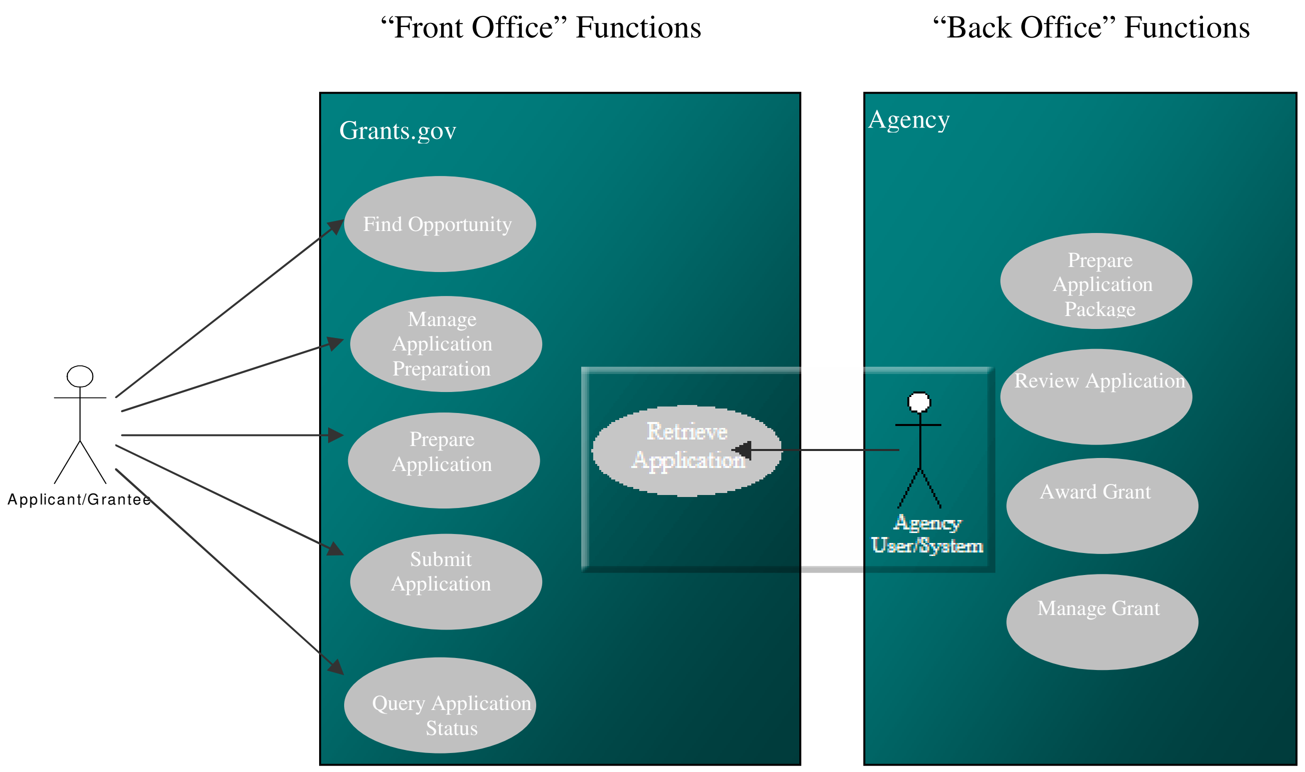


Figure 2-1 (2) Sided View of Grant Lifecycle

For the initial phase, the Toolkit will focus primarily on the *application retrieval* (Retrieve Application function) options available to agencies via the Agency Integration interfaces once an applicant has successfully submitted (filed) an application.

The Toolkit describes the one-stop set of interfaces where emerging e-business technologies and best practices are used to give agencies access to grantee applications gathered in a full service manner. The Agency Integration interfaces will be the single point of entry for agencies, offering both application data retrieval and secure e-business transaction processing.

2.3 Format Of The Agency Integration Toolkit

The Agency Integration Toolkit is an informational document that presents steps and guidelines that agency IT managers can use to begin to plan, schedule, and budget for Grants.gov integration. It contains a description of the proposed integration options, as well as criteria an agency can use in selecting an option that best fits its needs. A list of commercial and public domain tools is also provided. The appendices provide supplemental information, including a tutorial that provides an overview of various technologies that support the proposed application retrieval options.

2.4 Summary Of The Agency Technology Survey

A review of the agency technology survey[3] reveals that, in general, most agencies can be placed into one of four categories, as shown in the table below:

	Volume →	
Technology ↑	*Low-Volume / High-Automation*	*High-Volume / High-Automation*
	Low-Volume / Low-Automation	*High-Volume / Low-Automation*

The proposed integration solutions can accommodate agencies in any of these categories, as discussed in section, Solution Selection Criteria.

3. Proposed Integration Solutions

This section describes two proposed solutions to the question of how agencies electronically retrieve applications from Grants.gov:

- Grants.gov-to-agency integration via system-to-system interfaces; and,
- Grants.gov-to-agency integration via system-to-human interfaces.

A third option, Grants.gov-to-agency integration via automated, Secure File Transfer Protocol (FTP), was also considered. However, after some debate, it was determined that the initial version of the Agency Integration interfaces would *not* support FTP, even though this option is technically feasible and may be provided at a later time. One of the primary reasons for this decision was to facilitate the development of the core Grants.gov functionality by maintaining symmetry between the *inbound* and *outbound* Grants.gov interfaces.

Applicants can interact with Grants.gov via either human-to-system or system-to-system interfaces. For the purposes of this document, these will be referred to as *inbound* interfaces. Similarly, Grants.gov can interact with the Agencies in two ways, system-to-system or system-to-human interfaces. For the purposes of this document, these will be referred to as *outbound* interfaces.

[3] *"Integration Kit Survey Analysis for Grants.gov"*

A Human-to-System *inbound* interface is basically a set of Grants.gov provided Graphical User Interface (GUI's) via which an applicant can download a grant application, complete it, and submit (file) it to Grants.gov.

Similarly, a System-to-Human *outbound* interface is basically a set of Grants.gov provided Graphical User Interfaces (GUI's) via which an agency user can pull (download) the submitted application containing an application in a PDF or PureEdge format with any related attachments in their native format.

There are two options under the system-to-system *inbound* and *outbound* interfaces. The first option provides SOAP Client to Web Services exchange (inbound), and Web Services to SOAP Client exchange (outbound). The second option provides Web Services to Web Services exchange (inbound and outbound). Since Grants.gov system will provide Web Services (as opposed to just implementing a SOAP Client), any interfacing organization (applicant or agency) can use either of the above options. If an applicant or agency uses a SOAP Client, no exchange of messages will occur until a request is initiated from the applicant or agency (client). In the case of Web Services to Web Services interaction, both Grants.gov and Agency systems will be running and both can request and respond to messages. In either of these cases, once the message exchange "hand shake" is established, the transfer protocol and transfer of SOAP messages occurs in the same manner.

The proposed system-to-system and system-to-human interface technologies are discussed in turn below.

3.1 Proposed Integration Technology: System-to-System Interfaces

For the *outbound* system-to-system interfaces, Grants.gov-to-agency integration will be achieved via a message exchange that utilizes XML Web Services using SOAP v1.1, SOAP with Attachments, ebXML 2.0, HTTPS/SSL, and PDF.

3.1.1 Description Of Supporting Technology

The following table lists the technologies used, their purpose, and whether their use is mandatory or optional:

Table 3-1 System-to-System Supporting Technologies

Technology	Purpose/Description	Comments
ebXML v2.0	ebXML (Electronic Business using eXtensible Markup Language) is a XML based modular suite of specifications that enables enterprises of any size and in any geographical location to conduct business over the Internet.	Optional. Will be used as a guideline where appropriate.
HTTPS	HTTPS (Hypertext Transfer Protocol over Secure Socket Layer, or HTTP over SSL) is a Web protocol that encrypts and decrypts user page requests as well as the pages that are returned by the Web server. HTTPS is really just the use of SSL (Secure Socket Layer) as a	Required. The system-to-system message exchange will occur over HTTPS and will use digital certificates issued by Grants.gov. The authorization (i.e., roles) will be defined

	sub-layer under its regular HTTP application layering.	within the agency profile.
PDF	PDF (Portable Document Format) is a file format that captures all the elements of a printed document as an electronic image that can be viewed, navigated, printed, or forwarded to someone else. PDF files are created using Adobe Acrobat, Acrobat Capture, or similar products.	Required. Grants.gov will generate a PDF representation of the application once it has been successfully submitted (filed).
SOAP	SOAP (Simple Object Access Protocol) defines a standard data communications protocol for Web Services. SOAP provides a simple and consistent mechanism that allows one application to send an XML message to another application. SOAP specifies the structure of a message rather than how it is transported.	Required. Grants.gov will use SOAP as the messaging protocol for any system-to-system message exchanges.
SwA	SwA (SOAP with Attachments) describes a standard way to associate a SOAP message with one or more attachments in their native format in a multipart MIME structure for transport.	Required. Grants.gov will use SOAP with Attachments to package the main application data and any native format (PDF, Word, etc.) attachments in the same message.
UDDI	UDDI (Universal Description, Discovery and Integration) protocol provides a standard mechanism to publish and discover Web Services. Users can query the registry based on company name, industry category, service type, or other criteria. UDDI provides pointers to the WSDL document that describes a service and the access point of a service implementation.	Optional. Will not be implemented in the initial Grants.gov release, as it is unnecessarily complicated for a single-provider service like Grants.gov that would realize little benefit in the near term from dynamic binding.
Web Service	A Web Service represents an information resource or business process that can be accessed over the Web by another application via the use of XML messages. Although Web Services can be developed using a variety of XML-based communications protocols, the industry is converging on a core set of technologies to enable language- and platform-independence and to ensure multi-vendor interoperability. The core set of technologies includes Simple Object Access	Required. Grants.gov will publish its application retrieval interfaces as Web Service(s).

	Protocol (SOAP), Web Services Description Language (WSDL), and Universal Description, Discovery and Integration (UDDI).	
WSDL	WSDL (Web Services Description Language) defines a standard mechanism to describe a Web Service. A WSDL document describes what functionality a Web Service offers, how it communicates, and where it is accessible.	Optional. Grants.gov will provide a WSDL file as a means of concisely describing its Web Service interfaces. An agency may optionally choose to use this file (with or without the use of UDDI) to implement dynamic binding.
XML	XML (eXtensible Markup Language) is a meta-markup language that can be used to define data formats. XML data formats can be interpreted by any application, written in any language, running on any platform. XML data formats can also be transformed on the fly to adapt to the needs of a specific application.	Required. Grants.gov will use XML in any system-to-system message exchanges, as XML is the technology of choice in the present architecture of the Federal government for information exchange between systems.

A more detailed overview of the proposed technologies is provided in Appendix A Web Services and Related Technologies Overview. Additional resources are also listed in Appendix H Bibliography/Resources.

3.1.2 Supported Platforms

Due to the use of standards-based communications methods, Web Services can be developed in various programming languages, including Java, C++, VBScript, JavaScript, PHP, or PERL. More importantly, Web Services are "platform-independent"; i.e., Web Services can execute on virtually all platforms, including Linux, NT/2000/XP, HP-UX, and Solaris.

3.2 Proposed Integration Technology: System-to-Human Interfaces

For the *outbound* system-to-human interfaces, Grants.gov-to-agency integration will be achieved via interactive, web-based, graphical user interfaces (GUI's) accessible via HTTPS.

3.2.1 Description Of Supporting Technology

The following table lists the technologies used, their purpose, and whether their use is mandatory or optional:

Table 3-2 System-to-Human Supporting Technologies

Technology	Purpose/Description	Comments
HTTPS	HTTPS (Hypertext Transfer Protocol over Secure Socket Layer, or HTTP over SSL) is a Web protocol that encrypts and decrypts user	Required. Grants.gov web-based interfaces that support application retrieval (download)

	page requests as well as the pages that are returned by the Web server. HTTPS is really just the use of SSL (Secure Socket Layer) as a sub-layer under its regular HTTP application layering.	will be accessible via a Web browser that uses the HTTPS protocol. Grants.gov will issue digital certificates. The authorization (i.e., roles) will be defined within the agency profile.
PDF	PDF (Portable Document Format) is a file format that captures all the elements of a printed document as an electronic image that can be viewed, navigated, printed, or forwarded to someone else. PDF files are created using Adobe Acrobat, Acrobat Capture, or similar products.	Required. Grants.gov will generate and include a PDF representation of the application in the application download.
WinZip	WinZip is a Windows program that archives and compresses files so that they can be stored or distributed more efficiently. WinZip has a simple drag-and-drop interface that allows you to view individual files in a zip file without unzipping the file. WinZip will also launch installation programs from a zip file and automatically clean up after the installation. WinZip also supports other popular Internet file formats, including tar, gzip, Unix compress, Uuencode, BinHex, and MIME. ARJ, LZH, and ARC files are supported through other programs.	Required. Grants.gov will package an application and any supporting native format attachments into a single zip (.zip) file for download. WinZip (or some program that supports the zip file format) will need to be used to unzip the downloaded Grants.gov application.
Web Browser	A browser is an application program that provides a way to look at and interact with all the information on the World Wide Web. The word "browser" seems to have originated prior to the Web as a generic term for user interfaces that let you browse (navigate through and read) text files online. By the time the first Web browser with a graphical user interface was generally available (Mosaic, in 1993), the term seemed to apply to Web content, too. Technically, a Web browser is a client program that uses the Hypertext Transfer Protocol (HTTP) to make requests of Web servers throughout the Internet on behalf of	Required. Grants.gov web-based interfaces that support application retrieval (download) will be accessible via a Web browser that uses HTTPS.

	the browser user. A commercial version of the original browser, Mosaic, is in use. Many of the user interface features in Mosaic, however, went into the first widely used browser, Netscape Navigator. Microsoft followed with its Microsoft Internet Explorer. Today, these two browsers are the only two browsers that the vast majority of Internet users are aware of.	

3.2.2 Supported Platforms

Grants.gov web-based graphical user interfaces that support application retrieval (download) will be accessible via a Web browser that uses the HTTPS protocol. Grants.gov will support multiple browsers and versions, and will provide the same "look and feel" regardless of the browser. All industry standard platforms, including PC, Macintosh, Unix, and Linux will be supported.

3.3 Tools, Utilities

This section provides a *sample* list describing various Commercial Off The Shelf (COTS) and public domain tools and utilities, which agencies can use in their efforts to integrate with the Grants.gov application retrieval interfaces. A link to more information is also provided. The list has been divided into the following categories:

- Web Browsers
- Web Servers
- Web Services Frameworks
- Web Application Servers
- Web Services Integrated Development Environment (IDE)
- Web Services Toolkits
- XML Utilities
- E-Form Utilities

3.3.1 Web Browsers

Web Browser	Description
Internet Explorer (Microsoft)	Most widely used Web Browser today. Internet Explorer 6 is a set of core technologies in Microsoft Windows XP Home Edition and Windows XP Professional, and is available to users of Microsoft Windows® 98, Windows 98 Second Edition, Windows Millennium Edition (Windows Me), Microsoft Windows NT® Workstation 4.0, and Windows 2000 Professional operating systems. Internet Explorer 6 includes many new and enhanced features that can simplify daily tasks, while helping to maintain the privacy of personal information on the

	Web.
Netscape Navigator (Netscape)	The first widely used browser.

3.3.2 Web Servers

Web Server	Description
Apache (Apache)	Apache has been the most popular web server on the Internet since April of 1996. The June 2003 Netcraft Web Server Survey found that 63% of the web sites on the Internet are using Apache, thus making it more widely used than all other web servers combined.
Internet Information Server (Microsoft)	The second most widely used Web Server today. IIS 6.0 leverages the latest Web standards such as Microsoft ASP.NET, XML, and Simple Object Access Protocol (SOAP) for the development, implementation, and management of Web applications.

3.3.3 Web Services Frameworks

Toolkit	Description
.Net Framework (Microsoft)	The .NET Framework is the comprehensive tool set for rapidly building and integrating XML Web services, Windows-based applications, and Web solutions.
Java 2 Platform, Enterprise Edition (J2EE) (Sun Microsystems)	The J2EE platform is the foundation technology of the Sun ONE platform and Sun's Web services strategy.

3.3.4 Web Application Servers

Application servers offer developers an integrated Web development environment that allows them to connect and manage a variety of enterprise resources such as Web servers, databases, and legacy application systems. Application servers interface directly with the web server and the backend systems; they are where the business logic of web-based applications that access enterprise resources is embedded. The following tables list a *few* of the commercial and public domain application servers. See the Application Server Comparison Matrix provided by TheServerSide.com for a more comprehensive list.

Web Application Servers (Commercial)

Application Server	Description
BizTalk (Microsoft)	Offers an XML-based data integration platform. The server software includes tools to create and design XML definitions, map data from one definition to another, and manage process flow, document verification, and data exchange and processing. Only runs on Windows platform.
Oracle9i Application Server (Oracle)	Offers portal software, wireless and voice, Web page caching, business intelligence, complete integration, and more in a single product.
Sun ONE (Sun Microsystems)	Provides platform for developing and delivering Java web services. Delivers end-to-end high performance across a broad range of Web and enterprise application requirements.
WebLogic (BEA)	One of the most popular application servers. Visit the home page of BEA's WebLogic Server for an overview of the product, a list of benefits and profiles of customers who use WebLogic.
WebSphere (IBM)	Offers a cost-effective, integrated foundation platform for foundation messaging flows and applications.

Web Application Servers (Public Domain)

Application Server	Description
Tomcat (Apache)	The servlet container that is used in the official Reference Implementation for the Java Servlet and JavaServer Pages technologies.
JBoss (JBoss)	A Java application server developed in open source. Known for its ease of use, modularity and simplicity.
Caucho Resin	An XML Application Server. Features one of the fastest XML and XSL engines.

3.3.5 Web Services Integrated Development Environment (IDE)

An integrated development environment (IDE) is a programming environment that has been packaged as an application program, typically consisting of a code editor, a compiler, a debugger, and a graphical user interface (GUI) builder. The IDE may be a standalone application or may be included as part of one or more existing and compatible applications.

IDE	Description
WebLogic Workshop (BEA)	A visual development environment for building web services.
Sun ONE Studio 5 (Sun Microsystems)	Provides a powerful yet intuitive Integrated Development Environment (IDE) for Java, providing a comprehensive set of features and functionality which enables the development of applications ranging from desktop to standards based enterprise class applications and web services.

3.3.6 Web Services Toolkits

Toolkit	Description
Web Services Enhancements (Microsoft)	A free tool kit for developers building Web services on top of the .Net Framework and the Web services APIs in the .Net Framework.
Java Web Services Developer Pack (Sun Microsystems)	A free integrated toolkit that allows Java developers to build, test and deploy XML applications, Web services, and Web applications with the latest Web service technologies and standards implementations.

3.3.7 XML Utilities

Utility	Description
EPIC Editor (ArborText)	The Cadillac of tools included in this list. It offers the one of the nicest user interfaces, but has a learning curve and requires configuration to meet specific requirements. Includes a nice facility to bind the code to one or more specific DTDs to turn it into a "purpose-built" editor. EPIC has broad OS support and interoperability with document management systems.
XML Pro (Vervet Logic)	Offers visual editing that includes a drag-and-drop facility, and support for IBM's XML4J XML parser. Provides a clean, simple interface and easy-to-use controls.
XMLSpy (Altova)	A combination XML and DTD editor that offers great functionality. It also handles XML schemas and XSLT markup.

A more comprehensive list of editors can be found at http://www.xml.com/pub/pt/3.

3.3.8 E-Form Utilities

Utility	Description
Acrobat (Adobe)	Acrobat is a program from Adobe that allows documents to be captured electronically and then viewed it in its original format and appearance. Acrobat is ideal for making documents or brochures that were designed for the print medium, viewable electronically and capable of being shared with others on the Internet. To view an Acrobat document, which is called a Portable Document Format (PDF) file, Acrobat Reader is needed. The Reader is free and can be downloaded from Adobe. It can be used as a standalone reader or as a plug-in in a Web browser. Acrobat is actually a set of products. The latest version includes a "toolkit" that lets you scan in or otherwise capture documents created with Word, Pagemaker, and other desktop publishing products. The resulting PDF files can then be available for viewing either directly with the Reader or they can be viewed as embedded files within the browser.
PureEdge Viewer (PureEdge)	The PureEdge Viewer allows a user to capture, sign and submit XML e-forms for secure, tamper-proof transactions. Additionally, the form may begin as a template, but ends as a unique instance of data and presentation bound together. This creates a single XML file that is a complete record, (data, presentation, and logic) for archival and future reference. Its data can be extracted and used for transactional purposes such as integration with existing agency grant management systems. Each PureEdge e-form is self-contained. All form elements e.g., fields, labels, graphics, and all user-defined information e.g., text, attached files, electronic signatures, are contained within a single file. Users can open a blank e-form application template, complete and sign it, then submit it. At this point, the e-form is a unique object and a complete record of a user's transaction.

3.3.9 File Archive Utilities

Utility	Description
WinZip (WinZip Computing, Inc.)	WinZip is a Windows program that archives and compresses files so that they can be stored or distributed more efficiently.
WinRAR	An alternative to the WinZip utility. Offers consistently smaller archives, saving disk space and transmission costs. WinRAR provides complete support for RAR and ZIP archives and is able to unpack CAB, ARJ, LZH, TAR, GZ, ACE, UUE, BZ2, JAR, ISO archives.

4. Selecting and Implementing an Integration Solution

This section provides agency IT and grant-management personnel with information helpful in selecting and implementing one of two proposed Grants.gov-to-Agency integration solutions:

- Grants.gov-to-agency integration via system-to-system interfaces; and,
- Grants.gov-to-agency integration via system-to-human interfaces.

The following sections detail solution selection criteria and considerations, as well as technical and supporting information in the preparation to implement and use the system-to-system and system-to-human interfaces.

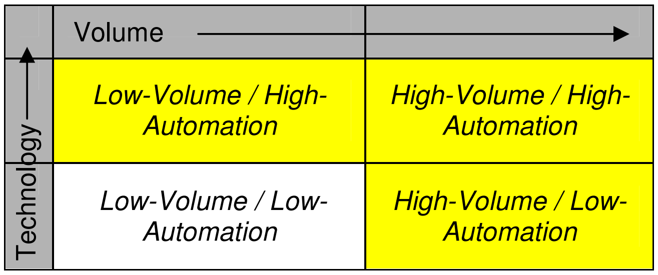
4.1 Solution Selection Criteria

This section provides criteria that will assist agency management personnel in selecting an integration solution. The table below summarizes many of the features, benefits, and costs associated with using and implementing the system-to-system and system-to-human integration interfaces.

Table 4-1 Selection Criteria

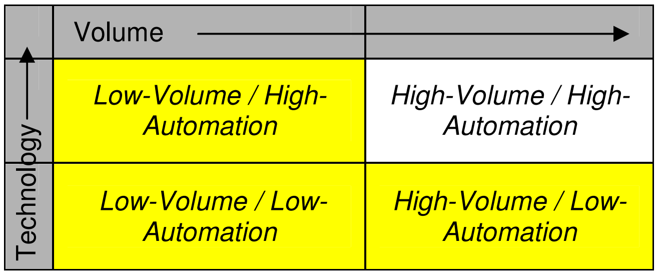
Integration Solution	Features/Benefits	Associated Costs
System-to-System	• Highly-automated: Capable of handling a large volume of application data without human intervention; • Helps eliminate the burden of redundant or disparate electronic and paper-based data collection requirements; • Supports the definition and implementation of simplified, standard processes and standard data definitions for Federal grant customer interactions; • Supports the standardization of the collection of financial and progress report data in support of audit and performance measurement activities;	Agency commitment, planning, and investment: • Cost of purchasing software and hardware required to implement the receiving end of the system-to-system message exchange. However, the use of open-source software, such as Java/J2EE, can mitigate much of the software cost; • Customized software development. • Support and maintenance of customized software once it has been deployed to production; • Training of staff and personnel to adopt new process changes and workflows
System-to-Human	• Ease of implementation: GUI interfaces provided by Grants.gov will allow interactive download of application data;	• Little or no IT investment: a browser and a network connection is virtually all that is required to interface with Grants.gov GUI interfaces

Based upon the information listed above, agencies within the highlighted quadrants below (i.e., agencies having a high application/submission volume *and/or* a highly automated system) may want to consider using the first integration solution, the use of system-to-system interfaces:



Volume
Technology
Low-Volume / High-Automation
High-Volume / High-Automation
Low-Volume / Low-Automation
High-Volume / Low-Automation

Agencies within the highlighted quadrants below (i.e., agencies having a low application/submission volume *and/or* little or no automation) may want to consider using the second integration solution, the use of system-to-human interfaces:



Volume
Technology
Low-Volume / High-Automation
High-Volume / High-Automation
Low-Volume / Low-Automation
High-Volume / Low-Automation

Integration and interface usage details for each of the two solutions are provided in the following sections.

4.2 System-to-System Interface Integration

This section details the use of a system-to-system message exchange in a highly automated manner to securely retrieve application data. The proposed technologies involved include XML Web Services using SOAP v1.1, SOAP with Attachments, ebXML 2.0, HTTPS/SSL, and PDF[4].

4.2.1 Assumptions

The success of the Grants.gov' system-to-system message exchange relies in great part upon the following fundamental assumptions about the process and the participants:

- Agencies selecting to participate in the electronic exchange have the technical capacity to do so.
- The information exchange is based on a system-to-system exchange, where each trading partner (or at least the Grants.gov system) has a device listening for messages at all times.

[4] *See Appendix Web Services and Related Technologies Overview for an overview of the proposed technologies.*

- The transfer of the body of the application, any other large document, or any message that contains a binary attachment is a controlled process primarily based on a retrieval request process that is initiated by an agency.
- The communication exchange is based on an agreed upon standard protocol and standard transaction definition. All trading partners will conform to these standards.
- The Grants.gov system has already received an application via either a system-to-system or human-to-system interaction;
- Grants.gov Web Services will process application retrieval requests one at a time; a separate retrieval request must be made for each application.
- The communication exchange will occur over HTTPS and will use digital certificates issued by Grants.gov. The authorization (i.e., roles) will be defined within the agency profile.

4.2.2 System-to-System Integration Details

Two versions of the message exchange model are being proposed: a *client-server* relationship model, and a *peer-to-peer* relationship model. In both applications, the manner in which Web Services are implemented within Grants.gov' does not change. The difference arises in the sequence of transactions and the degree of automation implemented by a given Agency.

In the first, less highly automated use of the electronic exchange model, agencies *will not* implement their own Web Services. Instead, agencies will simply implement SOAP clients that invoke the Grants.gov Web Services. In this usage of the exchange model, a *client-server* relationship has been established between an Agency and the Grants.gov systems, respectively.

In the second, more highly automated application of the electronic exchange model, agencies will implement not only SOAP clients, but also Web Service(s) that the Grants.gov system can invoke to notify the agency that new applications/submissions are available for download. In this usage of the exchange model, a *peer-to-peer* relationship has been established between an Agency and the Grants.gov systems.

The implications of each of the two uses of the exchange model is further detailed below.

Client-Server Relationship Model

In this model, the agency has *not* implemented any Web Services that the Grants.gov system can invoke. The implication is that an agency SOAP client *always initiates* the communication exchange with Grants.gov Web Services, and only the Grants.gov Web services will be up and running at all times (or within certain predetermined hours of the day), listening for any agency requests.

Peer-to-Peer Relationship Model

In this model, both the agency and Grants.gov will have implemented Web Services that can be invoked by either partner. The implication is that both Grants.gov and agency Web Services will need to be up and running at all times (or within certain predetermined hours of the day), listening for any messages. Also, either the Grants.gov system, or the agency system may *initiate* a message exchange.

Message Interaction Pattern

As not all agencies may choose to implement their own Web Services, any system-to-system message exchange between the Grants.gov and agency systems will be conducted (at least initially) using a *synchronous request/response interaction pattern*. A connection will be established directly to the ultimate recipient instead of using a provider that would store (queue) and forward messages until they can be delivered. Messages will *not* be persisted on the Grants.gov system. Record keeping of time and date of receipt of application and time and date of successful transfer to agency is a requirement of Grants.gov.

Message Exchange Sequence

Below is an example of a series of synchronous request-response message interactions that *might*[5] occur in a *Peer-to-Peer* relationship message exchange as part of the process of successfully retrieving an application from the Grants.gov system:

- The Grants.gov system invokes an agency's web service that is listening for any SOAP XML messages indicating that an application is ready to be retrieved.
- The agency system responds to this request with another SOAP XML message acknowledging receipt of the request.
- The agency system invokes the Grants.gov web service that is listening for any SOAP XML messages requesting a list of applications ready to be retrieved.
- The Grants.gov Web Service responds with another SOAP XML message containing a list of applications with their Grants.gov tracking numbers.
- The agency system invokes the Grants.gov Web Service that is listening for any SOAP XML messages requesting the download (pull) of a specific application.
- The Grants.gov system responds with another SOAP XML message that encapsulates the application, including any native documents in binary format.
- The agency system invokes the Grants.gov Web Service that is listening for any SOAP XML messages confirming the successful retrieval (download) of an application.
- The Grants.gov system responds with another SOAP XML message acknowledging the receipt of a successful application retrieval confirmation.

The sequence of message interactions in a Client-Server relationship model would be the same, with the exception of the absence of the first message interaction, as an agency would not have implemented their own Web Service(s) in this relationship model.

Message Structure Summary

This section provides a summary of the structure of the SOAP v1.1 compliant messages to be used by any system-to-system message exchange between the Grants.gov and agency systems, or vice-versa. A more detailed breakdown of the message is provided in Appendix C SOAP Message Structure.

Messages used in any system-to-system message exchange are formatted according to the SOAP with Attachments (SwA) specifications[6]. Some applications may not have any attachments;

[5] *Any request/response message exchange type is independent of any other; i.e., multiple application package download message exchanges can occur prior to a delivery confirmation message exchange occurring.*

however, the corresponding SOAP messages are still encapsulated in MIME via the transport protocol.

The following illustration depicts, at a high level, the structure of the application message, which is compliant with SOAP with Attachments specifications. This consists of a MIME envelope that embeds other MIME parts. The primary part of the multipart MIME message is the XML SOAP document itself; the subsequent parts contain any attached data:

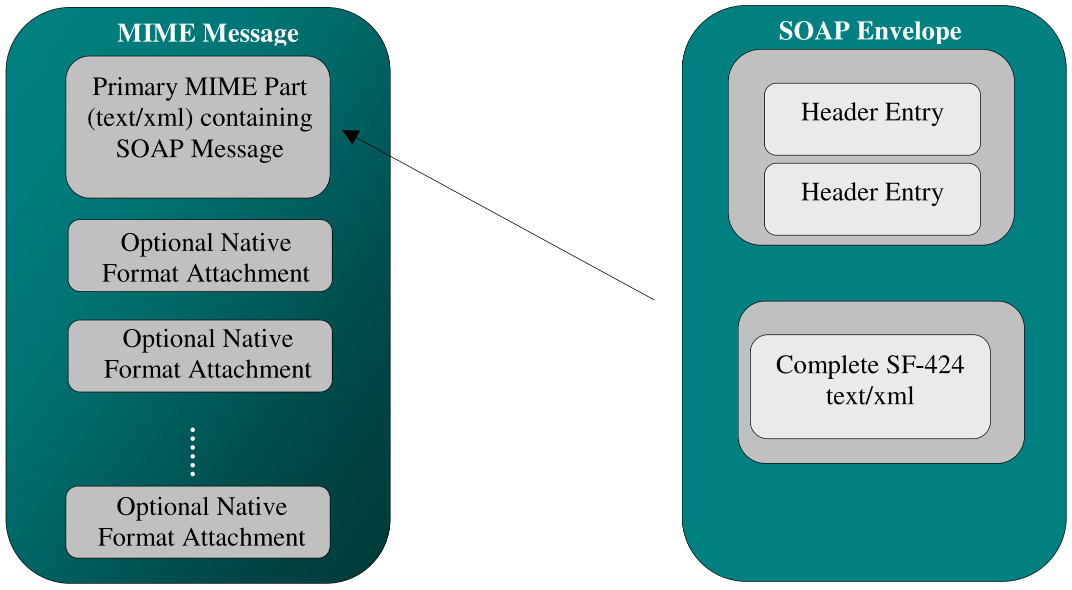

MIME Message
Primary MIME Part (text/xml) containing SOAP Message
Optional Native Format Attachment
Optional Native Format Attachment
Optional Native Format Attachment
SOAP Envelope
Header Entry
Header Entry
Complete SF-424 text/xml

Figure 4-1 Structure of the Application Message

A sample *skeleton* application message containing one PDF and one Word Document can be found in Appendix D Skeleton Application Message.

4.3 System-to-Human Interface Integration

This section details the use of Graphical User Interfaces (GUIs) by agency representatives to retrieve application data that has already been submitted to (filed with) the Grants.gov system.

4.3.1 Assumptions

- The agency user has the proper credentials (Username and Password identification) to login to Grants.gov.
- The agency user has been authorized (as defined in the agency profile) to retrieve (download) submitted applications from Grants.gov.
- The agency user has installed the necessary software on their workstations, such as WinZip, Adobe Acrobat/Reader, to unzip the downloaded file and view the applications in the format that has been specified in their agency's profile.

6 *http://www.w3.org/TR/SOAP-attachments*

4.3.2 GUI Interface Usage Details

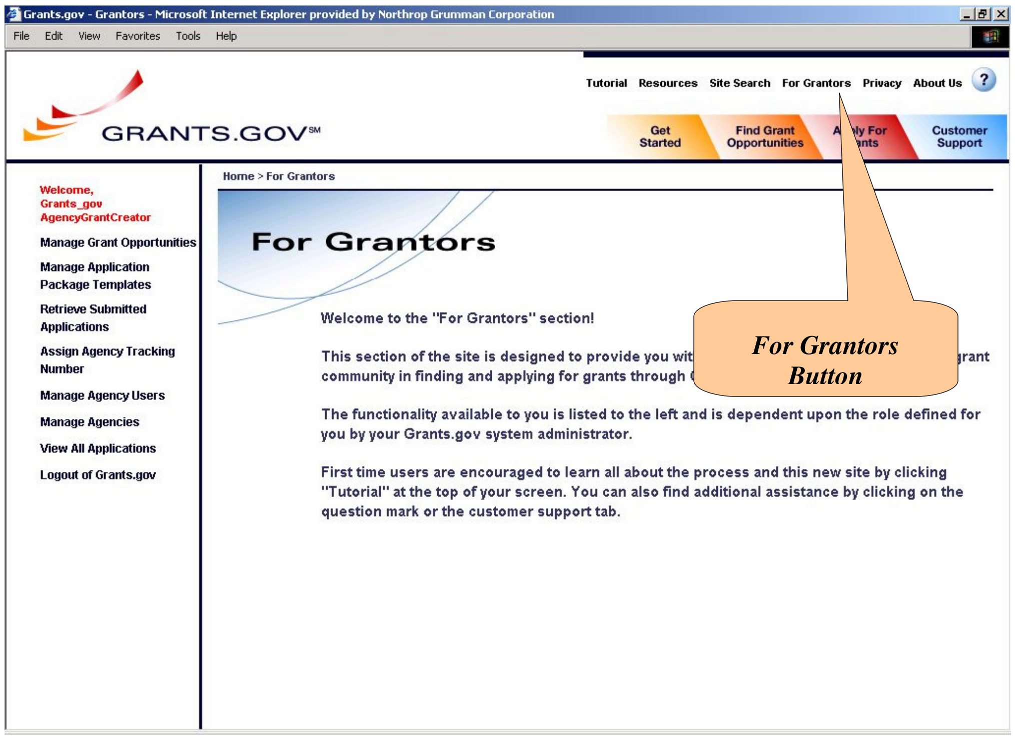
This is a System-to-Human interface where Grants.gov is accessed through a GUI via which an agency representative can pull (download) the submitted application in PDF format (plus attachments in their native format).

Each application will be available in a zip file format. The zip file (which will be named *agencynamedatetime.zip*) will contain the application in PDF format (plus attachments in their native format). If there are several applications ready for download, each application will be individually zipped, and then all applications will be zipped again into a single downloadable zip file. Once downloaded, the agency user will have to first unzip the main zip file and then each individual zip file.

The following pages display the details of this process:

Table 4-2 Access Instructions

Steps	***Action***	**Result**
1.	Agency user accesses Grants.gov web page HTTP://www.grants.gov.	Grants.gov Welcome Page is displayed. (*Figure 4-1 below*).
2.	User selects ***For Grantors*** option on the top navigation bar.	Logon Screen is displayed.


Grants.gov - Grantors - Microsoft Internet Explorer provided by Northrop Grumman Corporation
File Edit View Favorites Tools Help
Tutorial Resources Site Search For Grantors Privacy About Us
GRANTS.GOV℠
Get Started
Find Grant Opportunities
A ly For nts
Customer Support
Welcome, Grants_gov AgencyGrantCreator
Manage Grant Opportunities
Manage Application Package Templates
Retrieve Submitted Applications
Assign Agency Tracking Number
Manage Agency Users
Manage Agencies
View All Applications
Logout of Grants.gov
Home > For Grantors
For Grantors Button

For Grantors

Welcome to the "For Grantors" section!

This section of the site is designed to provide you wit rant community in finding and applying for grants through

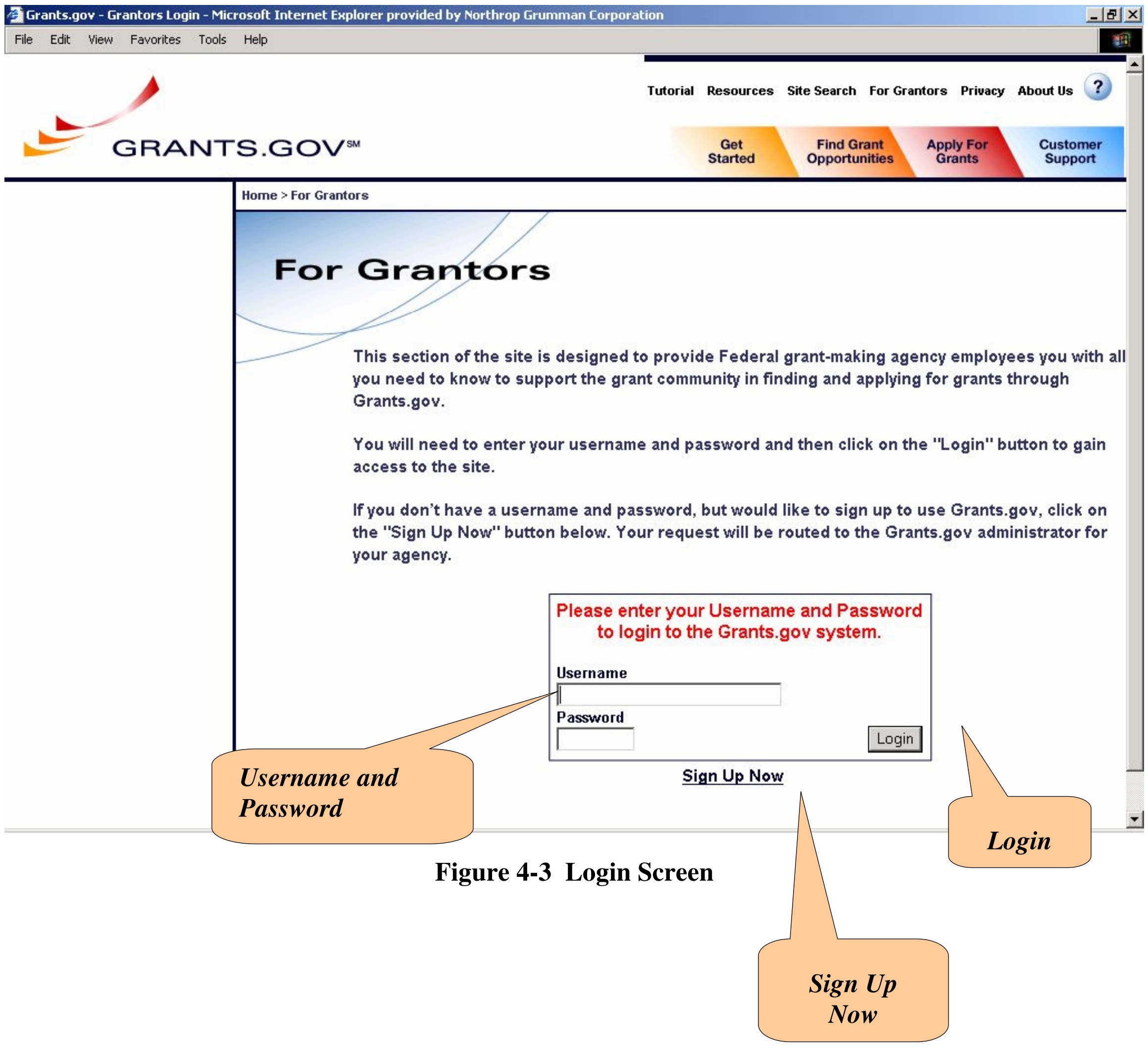
The functionality available to you is listed to the left and is dependent upon the role defined for you by your Grants.gov system administrator.

First time users are encouraged to learn all about the process and this new site by clicking "Tutorial" at the top of your screen. You can also find additional assistance by clicking on the question mark or the customer support tab.

Figure 4-2 Grants.gov Welcome Page

Table 4-3 Login Instructions

Steps	*Action*	**Result**
3.	Agency user enters his/her ***Username*** and ***Password.*** *(If you do not have a username and password select Sign Up Now to register for a username and password)*	User name appears in the name field. Asterisks appear in the Password field.
4.	User selects the ***login*** button.	The *GRANTORS only* screen is displayed.



Grants.gov - Grantors Login - Microsoft Internet Explorer provided by Northrop Grumman Corporation

File Edit View Favorites Tools Help

Tutorial Resources Site Search For Grantors Privacy About Us

GRANTS.GOV℠

Get Started | Find Grant Opportunities | Apply For Grants | Customer Support

Home > For Grantors

For Grantors

This section of the site is designed to provide Federal grant-making agency employees you with all you need to know to support the grant community in finding and applying for grants through Grants.gov.

You will need to enter your username and password and then click on the "Login" button to gain access to the site.

If you don't have a username and password, but would like to sign up to use Grants.gov, click on the "Sign Up Now" button below. Your request will be routed to the Grants.gov administrator for your agency.

Please enter your Username and Password to login to the Grants.gov system.

Username

Password

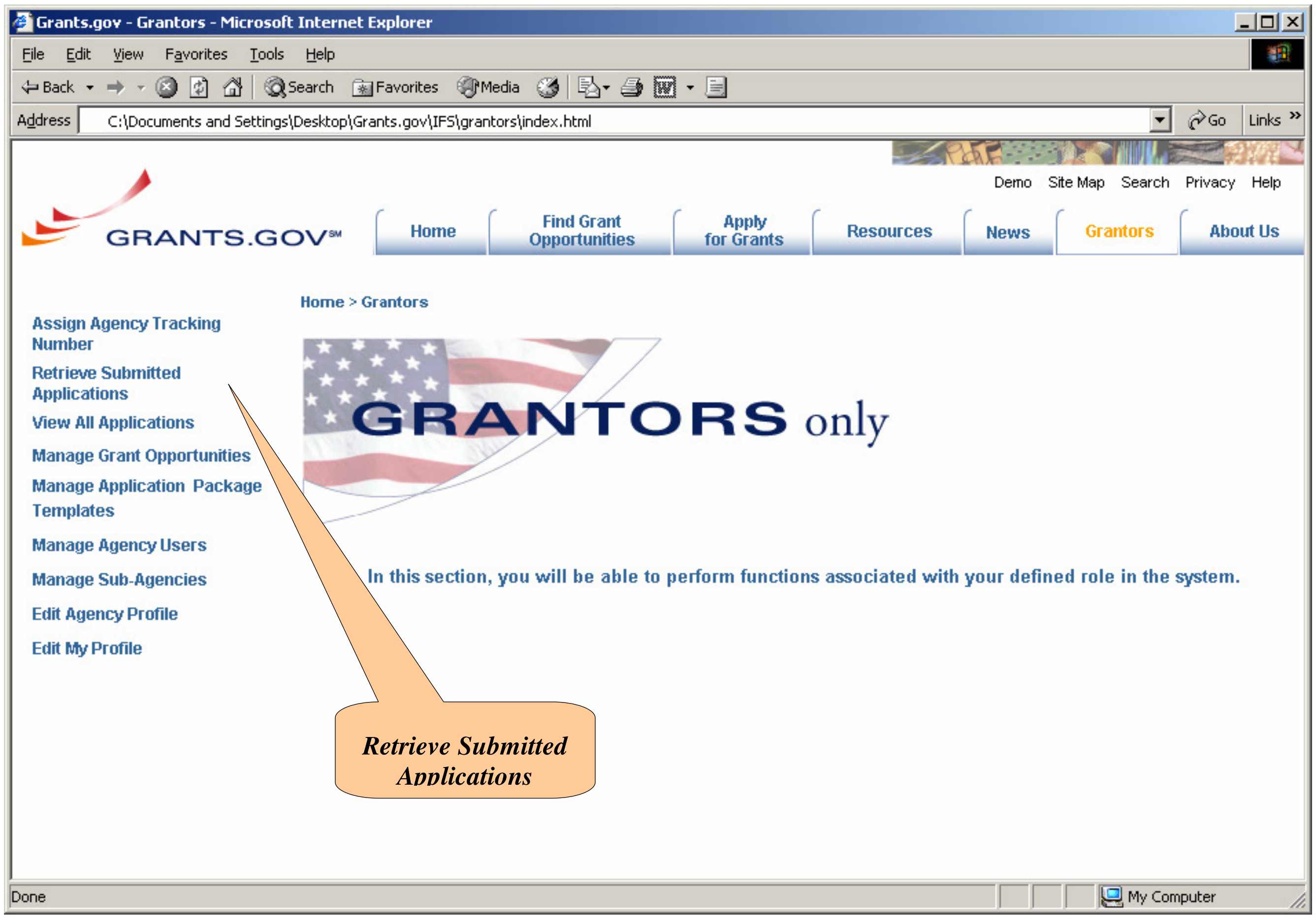
Login

Sign Up Now

Figure 4-3 Login Screen

Table 4-4 Retrieve Submitted Applications Instructions

Steps	***Action***	**Result**
5.	User selects ***Retrieve Submitted Applications*** from the options located along the left hand side of the screen.	The ***Retrieve Submitted Applications*** screen is displayed.



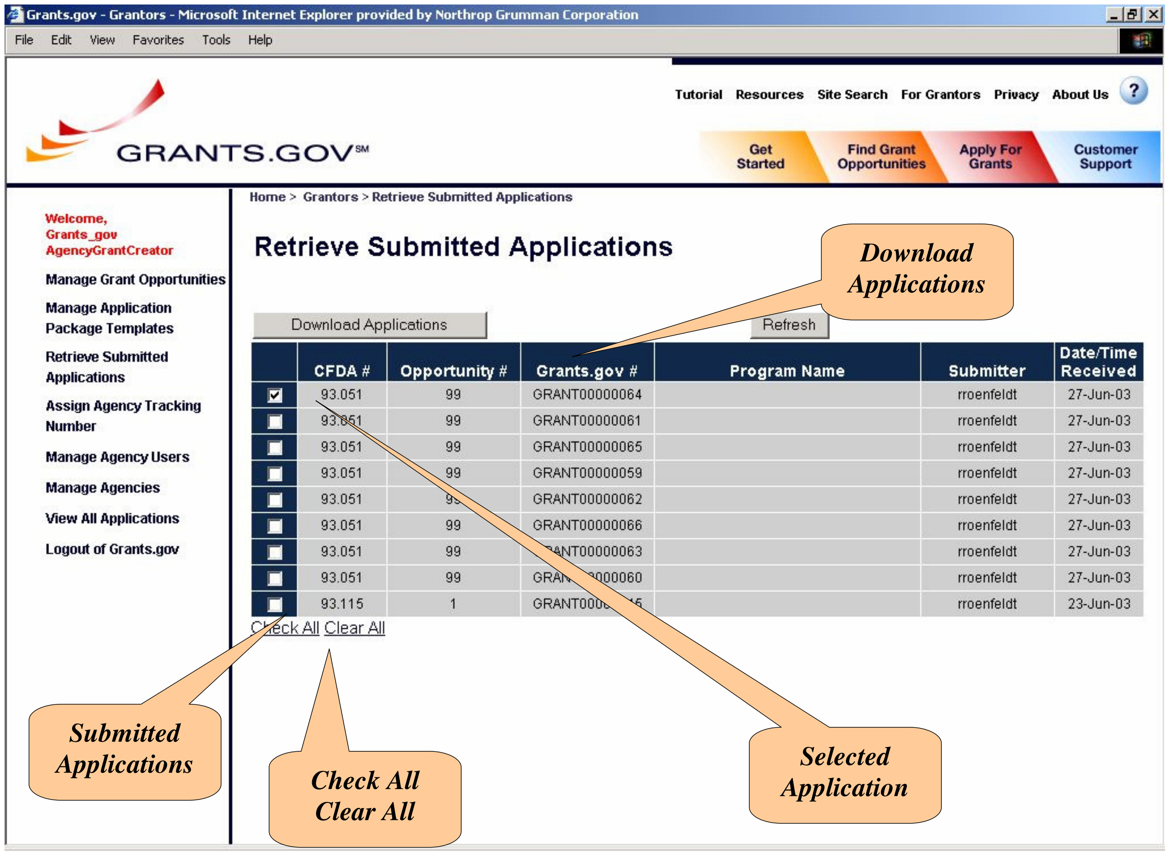
Grants.gov - Grantors - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address C:\Documents and Settings\Desktop\Grants.gov\IFS\grantors\index.html
Go Links
Demo Site Map Search Privacy Help
GRANTS.GOV℠
Home
Find Grant Opportunities
Apply for Grants
Resources
News
Grantors
About Us
Home > Grantors
Assign Agency Tracking Number
Retrieve Submitted Applications
View All Applications
Manage Grant Opportunities
Manage Application Package Templates
Manage Agency Users
Manage Sub-Agencies
Edit Agency Profile
Edit My Profile
GRANTORS only
Retrieve Submitted Applications
Done
My Computer

In this section, you will be able to perform functions associated with your defined role in the system.

Figure 4-4 Grantors only Screen

Table 4-5 Select Applications for Download Instructions

Steps	***Action***	**Result**
6.	User has the option to select (1) or all of the submitted applications that are ready to be downloaded. The system will ensure that only applications for which download is authorized can be accessed.	Check marks appear next to selections.
7.	Select ***Download Applications*** button.	Download begins. **Note:** The download process will be handled by the operating system where the system will prompt the user for name and location of the download file to be saved. The system will save the file as a *zip* file. Grants.gov will document by whom applications are downloaded.



Grants.gov - Grantors - Microsoft Internet Explorer provided by Northrop Grumman Corporation
File Edit View Favorites Tools Help
Tutorial Resources Site Search For Grantors Privacy About Us
Get Started
Find Grant Opportunities
Apply For Grants
Customer Support
Home > Grantors > Retrieve Submitted Applications
Welcome, Grants_gov AgencyGrantCreator
Manage Grant Opportunities
Manage Application Package Templates
Retrieve Submitted Applications
Assign Agency Tracking Number
Manage Agency Users
Manage Agencies
View All Applications
Logout of Grants.gov
Download Applications
Refresh
Check All Clear All
Download Applications
Submitted Applications
Check All Clear All
Selected Application

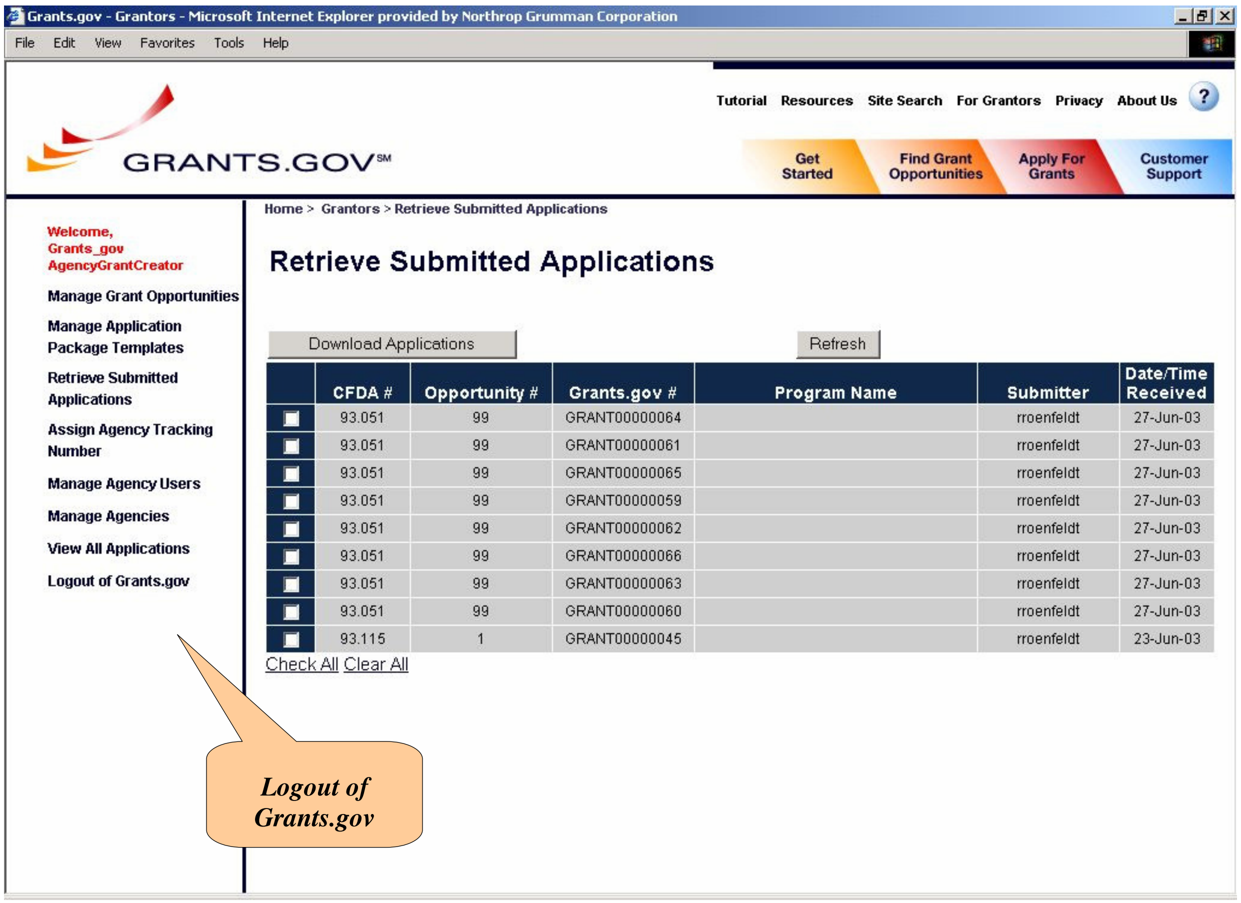
Retrieve Submitted Applications

	CFDA #	Opportunity #	Grants.gov #	Program Name	Submitter	Date/Time Received
✓	93.051	99	GRANT00000064		rroenfeldt	27-Jun-03
☐	93.051	99	GRANT00000061		rroenfeldt	27-Jun-03
☐	93.051	99	GRANT00000065		rroenfeldt	27-Jun-03
☐	93.051	99	GRANT00000059		rroenfeldt	27-Jun-03
☐	93.051	99	GRANT00000062		rroenfeldt	27-Jun-03
☐	93.051	99	GRANT00000066		rroenfeldt	27-Jun-03
☐	93.051	99	GRANT00000063		rroenfeldt	27-Jun-03
☐	93.051	99	GRANT00000060		rroenfeldt	27-Jun-03
☐	93.115	1	GRANT000[illegible]15		rroenfeldt	23-Jun-03

Figure 4-5 Retrieve Submitted Applications Screen

Table 4-6 Logout Instructions

Steps	*Action*	**Result**
8.	After the download process is complete, the system redisplays the **Retrieve Submitted Applications** screen.	**Note:** If all of the applications have already been downloaded, the applications list will be empty. If one application was selected the remaining applications will be displayed on the list.
9.	The user will locate the zipped file.	Unzip the file to retrieve one or all of the applications.
10.	Select ***Logout of Grants.gov*** to close your browser window when you are finished.	Grants.gov application will close.


Grants.gov - Grantors - Microsoft Internet Explorer provided by Northrop Grumman Corporation
File Edit View Favorites Tools Help
Tutorial Resources Site Search For Grantors Privacy About Us
GRANTS.GOV℠
Get Started
Find Grant Opportunities
Apply For Grants
Customer Support
Home > Grantors > Retrieve Submitted Applications
Welcome, Grants_gov AgencyGrantCreator
Manage Grant Opportunities
Manage Application Package Templates
Retrieve Submitted Applications
Assign Agency Tracking Number
Manage Agency Users
Manage Agencies
View All Applications
Logout of Grants.gov
Download Applications
Refresh
Check All Clear All
Logout of Grants.gov

Retrieve Submitted Applications

	CFDA #	Opportunity #	Grants.gov #	Program Name	Submitter	Date/Time Received
☐	93.051	99	GRANT00000064		rroenfeldt	27-Jun-03
☐	93.051	99	GRANT00000061		rroenfeldt	27-Jun-03
☐	93.051	99	GRANT00000065		rroenfeldt	27-Jun-03
☐	93.051	99	GRANT00000059		rroenfeldt	27-Jun-03
☐	93.051	99	GRANT00000062		rroenfeldt	27-Jun-03
☐	93.051	99	GRANT00000066		rroenfeldt	27-Jun-03
☐	93.051	99	GRANT00000063		rroenfeldt	27-Jun-03
☐	93.051	99	GRANT00000060		rroenfeldt	27-Jun-03
☐	93.115	1	GRANT00000045		rroenfeldt	23-Jun-03

Figure 4-6 Logout Selection

5. Conclusion

The Agency Integration Toolkit document has explained many of the challenges that Grantee-organizations face when applying for Federal grants. By implementing the Grants.gov Storefront we:

- Eliminate redundancy
- Simplify the process
- Protect confidentiality
- Standardize the collection of data

There are two proposed solutions for agencies to use to electronically retrieve grant applications from Grants.gov.

System-to-System Interfaces

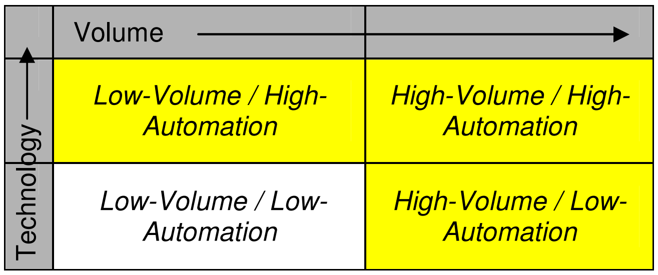
- For the *outbound* system-to-system interfaces, Grants.gov-to-agency integration via a message exchange that utilizes XML Web Services using SOAP v1.1, SOAP with Attachments, ebXML 2.0, HTTPS/SSL, and PDF.

System-to-Human Interfaces

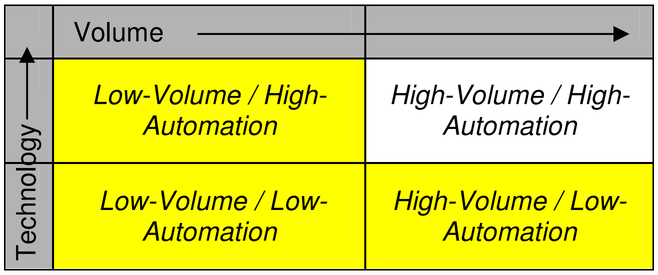
- For the *outbound* system-to-human interfaces, Grants.gov-to-agency integration is achieved via a set of Grants.gov provided Graphical User Interfaces (GUI's) by which an agency user can download the submitted application that contains an application in a PDF format along with any related attachments in their native format.

Section 4 (4.1) details the solution selection criteria and considerations, as well as technical and supporting information to implement and use the system-to-system and system-to-human interfaces.

Agencies within the highlighted quadrants below (i.e., agencies having a high application/submission volume *and/or* a highly automated system) may want to consider using the first integration solution, the use of system-to-system interfaces.


Volume
Technology
Low-Volume / High-Automation
High-Volume / High-Automation
Low-Volume / Low-Automation
High-Volume / Low-Automation

Agencies within the highlighted quadrants below (i.e., agencies having a low application/submission volume *and/or* little or no automation) may want to consider using the second integration solution, the use of system-to-human interfaces.


Volume
Technology
Low-Volume / High-Automation
High-Volume / High-Automation
Low-Volume / Low-Automation
High-Volume / Low-Automation

6. Acronyms

Acronym	Definition
CFDA	Catalog of Federal Domestic Assistance
COBRA	Common Object Request Broker Architecture
COTS	Common-Off-The-Shelf
DUNS	Data Universal Numbering System
ebXML	Electronic Business eXtensible Markup Language
FTP	File Transfer Protocol
GUI	Graphic User Interface
HTML	Hyper Text Markup Language
HTTP	Hyper Text Transfer Protocol
HTTPS	Hyper Text Transfer Protocol Secure Sockets
HTTP/SSL	Hyper Text Transfer Protocol Secure Socket Layer
IDE	Integrated Development Environment
IT	Information Technology
MIME	Multipurpose Internet Mail Extensions
OASIS	Organization for Advancement of Structured Information Standards
PDF	Portable Document Format
RMI	Remote Method Invocation
RPC	Remote Procedure Calls
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SSL	Secure Socket Layer
SwA	SOAP with Attachment

UDDI	Universal Description Discovery and Integration
URI	Uniform Resource Identifiers
URL	Uniform Resource Locator
WSDL	Web Services Description Language
XCBL	Extensible Common Business Language
XML	eXtensible Markup Language

Appendix A Web Services and Related Technologies Overview

Grants.gov proposes providing system-to-system interfaces that would provide Grants.gov-to-agency integration via a message exchange that utilizes XML Web Services using SOAP v1.1, SOAP with Attachments, ebXML 2.0, HTTPS/SSL, and PDF. This appendix provides an overview of these technologies.

System-to-System Exchange

To address the issue of large transaction peaks, the proposed system-system message exchange solution defines a message exchange with a controlled submission of large files. The first component in this solution is a system-to-system exchange; the second component is a set of rules by which the transactions themselves are exchanged.

To allow system-to-system transactions and submission of electronic applications, the concept of a system-to-system message exchange is proposed for the design of the Agency Integration application retrieval interfaces. The system-to-system message exchange allows Grants.gov and Agency systems to conduct predefined transactions *without* human intervention.

The system-to-system message exchange is not the same interface employed by a user on a PC or workstation entering and submitting data using a web page or sending E-Mail. The interaction with the Grants.gov application retrieval interfaces is done in the background via a system message consisting of an augmented XML file. The system-to-system communication is bi-directional. For example, the agency system might request a list of grant applications that are ready to be retrieved by sending a properly formatted XML message to the appropriate Agency Integration interface. This interface, in turn, would respond with another XML message containing the requested list.

A system-to-system message exchange implies that trading partners (the Grants.gov and Agency systems, in this case) have interfaces "listening" to the external world, ready to receive transactions at any time. It usually means that a server running specific software is accessible through firewalls to authorized external partners. The trading partner needs to participate in the SYSTEM-TO-SYSTEM MESSAGE exchange using the exact specifications stipulated for the Exchange. Also, the transport mechanism, transaction packaging, and, most importantly, the transaction content itself needs to be defined, agreed upon, and implemented at each node of the exchange. Finally, for security reasons, it also implies that the trading partner systems are identified, authenticated, and certified.

The next section briefly describes ebXML and its benefits.

ebXML

There are a number of system-to-system exchange protocols and models defined. Many are proprietary commercial standards and products. A few exchange standards, however, have been initiated by standard setting bodies. ebXML is one such standard.

The ebXML (Electronic Business using eXtensible Markup Language) standard resulted from a combined effort of the UN (UN/CEFACT) and OASIS (the Organization for Advancement of Structured Information Standards). The organization was formed to provide an open XML-based infrastructure to enable the global use of electronic business information. ebXML currently works on all aspects of a standard, including an implementation framework, vocabularies, data dictionaries, and processes. ebXML is a modular suite of specifications that enables enterprises of any size and in any geographical location to conduct business over the Internet.

The mission of ebXML is to "*provide an open XML-based infrastructure enabling the global use of electronics business information in an interoperable, secure and consistent manner by all parties*".

As described by Sean Gallagher[7], ebXML is a *"collection of standardized message structures and vocabularies for conducting automatic [system-to-system] transactions using the Internet or e-mail"*. It provides specifications for registering products and services; message formats and syntaxes; descriptions of how to use the products and services; and guidelines for setting up contracts and exchanging the messages themselves.

At the core of ebXML are four sets of standards, or specifications, designed for specific steps in the process. Each standard can be adopted *independently* of the others. The four steps (and related standards) include:

1. Describe the business processes you want to share, using XML or eXtensible Common Business Language (XCBL)
   Standard: Business Process Specification Schema/Core Components
2. Register descriptions of your business and of the processes you want to share in a common XML repository
   Standard: Registry Information Model
3. Explain your transaction system's messaging capabilities, how to use the processes you're sharing, and the contractual terms
   Standard: Collaboration Protocol Profile and Agreement Specification
4. Package and send the electronic business documents
   Standard: Message Service Specification

ebXML, together with Web Services, facilitates companies of any size, using any computing platform, to conduct automated business over the Internet. Automating transactions over the Internet instead of private, dedicated networks typically found in EDI exchanges, for example, can substantially reduce costs.

ebXML may be hard to put into operation in its entirety, and it may be more than the Grants.gov Agency Integration exchange needs. However, it does provide a recognized and well-documented framework to work with. Some components of ebXML, such as the Messaging Services Specification, offers recommendations for message packaging, reliable messaging, error handling, and security that provide a strong foundation for the creation of a SYSTEM-TO-SYSTEM MESSAGE exchange.

The solution described herein proposes using the ebXML framework as a guideline for the implementation of the Grants.gov system-to-system message exchange. If the standard is too complex for the size of the problem Grants.gov is trying to solve, only the appropriate subsets of the standards will be implemented.

The next section briefly describes Web Services and their benefits.

[7] *http://www.baselinemag.com/article2/0,3959,659101,00.asp*

Web Services

Simply put, a Web Service represents an information resource or business process that can be accessed over the Web by another application. Web Services are completely platform-independent. Web Services can be developed using any language, and they can be deployed on any platform, from the smallest device to the largest super computer.

Web Services offer valuable business benefits, such as increased agility, improved efficiency, and quantifiable cost savings. These business benefits derive from the fact that Web Services simplify the process of making applications talk to each other.

From a technical perspective, Web Services technology represents the convergence of three pre-existing technologies: service-oriented architecture (SOA), the Extensible Markup Language (XML), and the Internet.

SOA refers to distributed computing systems that expose data or business logic through a programming interface. Examples of earlier SOA systems include remote procedure calls (RPC), remote method invocation (RMI), and Common Object Request Broker Architecture (CORBA). A Web Service is a software resource that exposes its capabilities through a programming interface.

XML is a meta-markup language that can be used to define data formats. XML data formats can be interpreted by any application, written in any language, running on any platform. XML data formats can also be transformed on the fly to adapt to the needs of a specific application. A Web Service communicates using XML messages.

The **Internet** refers to an immense shared information space in which every information resource can be accessed by it uniform resource identifier (URI) over standard, pervasive Internet protocols. A Web Service can be accessed by a URI over Internet protocols. What distinguishes Web Services from other types of Web-based applications is that Web Services are designed to support system-to-system communication. Other Web applications support human-to-human communication (for example, e-mail and instant messaging) or human-to-system communication (Web browsers). Web Services are designed to allow applications to communicate without human assistance or intervention.

Core Web Services Technologies

The most basic technology that supports Web Services is XML. Web Services describe their interfaces using **XML**. Web Services communicate using XML messages. The message data are encoded as XML.

Although Web Services can be developed using a variety of XML-based communications protocols, the industry is converging on a core set of technologies to enable language- and platform-independence and to ensure multi-vendor interoperability. The core set of technologies includes Simple Object Access Protocol (SOAP), Web Services Description Language (WSDL), and UDDI (Universal Description, Discovery and Integration). Each is described briefly below.

SOAP

SOAP (Simple Object Access Protocol) defines a standard data communications protocol for Web Services. SOAP provides a simple and consistent mechanism that allows one application to send an XML message to another application. A Web Service's interface is defined by the SOAP messages that it supports. SOAP is designed to be a transport protocol (HTTP, HTTPS, or JMS, for example) independent. SOAP specifies the structure of a message rather than how it is transported.

WSDL (Web Services Description Language)

WSDL (Web Services Description Language) defines a standard mechanism to describe a Web Service. A WSDL document describes what functionality a Web Service offers, how it communicates, and where it is accessible. It is possible to compile a WSDL document into client and server skeleton code, or to use it at runtime to create a client proxy object that can call the Web Service using SOAP.

UDDI (Universal Description, Discovery and Integration)

UDDI (Universal Description, Discovery and Integration) protocol provides a standard mechanism to publish and discover Web Services. Users can query the registry based on company name, industry category, service type, or other criteria. UDDI provides pointers to the WSDL document that describes a service and the access point of a service implementation.

The following figure illustrates how the core Web Services technologies are related to each other:

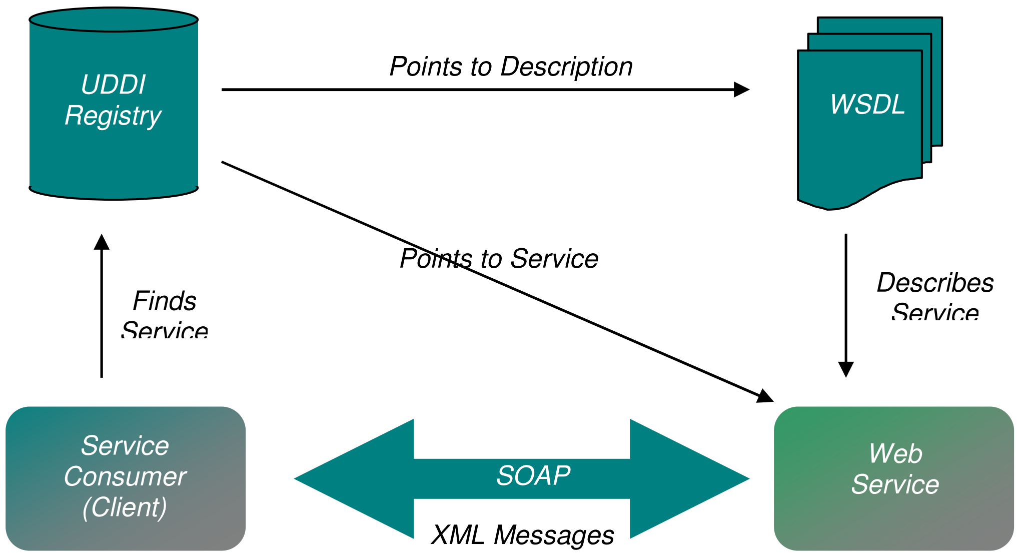


Figure A- 1 Core Web Services Relationship

When a service provider wants to make a service available to consumers, it describes the service using WSDL and registers the service in a UDDI registry. The UDDI registry then maintains pointers to the WSDL description and to the service. When a service consumer wants to use a service, it queries the UDDI registry to find a service that matches its needs and obtains the WSDL description of the service, as well as the access point of the service. The service consumer uses the WSDL description to construct a SOAP message with which to communicate with the service.

The next section describes the use of XML with binary attachments as a standard means of exchanging information between systems.

XML and Binary Attachments

The technology of choice in the present architecture of the Federal government is to use XML as the standard for information exchange between systems. The XML technology allows disparate systems to find a common information description and exchange standard. However, an XML file is a text file where each element of the information payload is preceded and followed by a descriptive tag. A formatted document is a binary stream with characters that may interfere with the tag scheme of the XML stream. To reliably transport a binary document with XML:

- The document must be transformed, or mapped, from its binary characters to the printable characters of the ASCII character set; or,
- The document must be transported as an external attachment to the XML stream.

There are numerous ways of solving this problem. As the transaction is expected to be one physical file for the XML stream and any attached documents, SOAP with attachments (SwA) is being proposed as the method for reliably transporting a binary document with XML.

The following sections provide an overview of SOAP with Attachments (SwA).

SOAP with Attachments (SwA)

The SOAP standard defines an `href` attribute within the SOAP body that allows pulling the embedded content out of the XML container, and replacing it with a link. Unfortunately, most binary documents that accompany a grant application will not be available to an agency via a URL; as a result, this approach will not be appropriate. In this case, the proposed solution is to use SOAP with Attachments (SwA), as discussed in the following section.

The following figure illustrates a SOAP message with binary attachments.

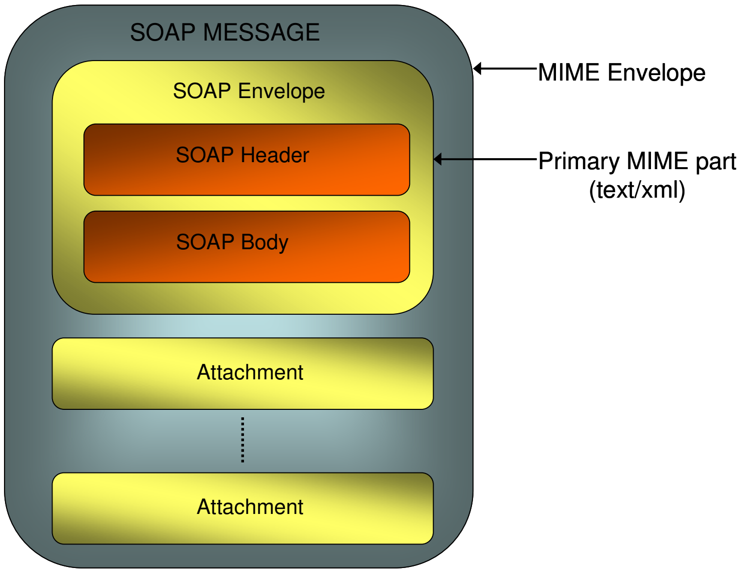


Figure A- 2 SOAP Message with Binary Attachments

SOAP Messages with Attachments (SwA) are basically composed of a primary, multipart MIME (Multipurpose Internet Mail Extension) message, which contains an XML SOAP document; the subsequent parts contain the attached binary data.

HTTP forms can be sent using MIME multipart; this means that all web servers should already have the proper MIME machinery built in.

The advantage of this implementation is that it is relatively simple to implement.

A drawback to SwA is that it cannot handle data streaming. This means that the client would have to buffer the entire attachment (for instance a multi-media data stream) before processing it.

Web Services and Security Considerations

Web Services need to be implemented securely; i.e., the data transferred over the Web Service needs to be safe from interception tampering, and unauthorized access.

The remainder of this section briefly outlines the threats to distributed systems, and how such risks can be mitigated.

Threats in a Distributed System

Some of the most common security threats to a distributed information system include:

- An authorized user of the system gaining access to information that should be hidden from him/her.
- A user masquerading as someone else, and so obtaining access to whatever that second user is authorized to do. Actions will then be attributed to the wrong person.
- Eavesdropping on a communication line, so gaining access to confidential data.
- Tampering with communication between objects: modifying, inserting, and deleting items.
- Lack of accountability due, for example, to inadequate identification of users. The most important problem is repudiation, meaning denial of responsibility for an action.

Security Services

The two most basic and fundamental elements of security are secure transport and authentication. Authentication mechanisms and particular secure transport protocols are often associated with one another.

Secure Transport refers to the encryption of sensitive data being transported. Sensitive data is encrypted to:

- Prevent interception of the data
- Prevent anyone from tampering with the data

Secure Socket Layer (SSL) is a commonly used protocol for managing the security of a message transmission on the Internet. SSL contains its own server-side authentication and an optional client authentication.

Authentication refers to the means used to confirm the identity of the client. Six common authentication mechanisms include:

1. **HTTP Basic**: The client gathers a user name and password and establishes the same with the service. Basic HTTP must be used with secure transport, to eliminate the risk of the password being intercepted.
2. **HTTP Digest**: The client gathers a user name and password. He uses these to generate a password digest, which is passed on to the service. The password cannot be reconstructed from the digest, reducing the risks resulting from possible interception.
3. **SSL Client Authentication**: The optional part of SSL protocol. In client authentication, the user owns a private key-certificate pair. In subsequent communication, the server verifies client identity based on the client's certificate, using asynchronous cryptography.
4. **Simple Public Key Mechanism (SPKM**): A security mechanism that always requires mutual client-server authentication. Using SPKM, multiple service identities are possible on the same server. It also automatically prevents repudiation, as every message contains a signature. Unlike HTTP- and SSL-based authentication, SPKM is not linked to any transport protocol.

5. **SOAP Digital Signature (SDS)**: A SOAP header entry carrying an XML digital signature.
6. **Kerberos**: A system developed at the Massachusetts Institute of Technology that uses passwords and symmetric cryptography (DES) to implement a ticket-based, peer-entity authentication service and an access control service distributed in a client-server network environment.

Transport-level Versus Message-level Security

Web Services security is still evolving, like Web Services infrastructure itself. Currently, Web Services are more tightly coupled to the network transport layer, using so-called transport layer security. There is, however, an evolution to message level security.

Transport level security using HTTP Basic / Digest and SSL, for example, is the usual "first line of defense", as securing the transport mechanism itself makes Web Services inherently secure. The trade-off is transport dependency.

Message level security, which can be as or more effective, has the added flexibility that the message can be sent over any transport.

Appendix B Application Retrieval Technical Details Skeleton

The following table provides a sample skeleton of the format that is used to summarize the technical details that will allow agency systems to bind to and invoke the Grants.gov application retrieval and associated Web services.

Table B -1 Sample Skeleton Format

Technical Detail	Value
Name of Service:	*Name of Service*
SOAP Endpoint URL:	*https://grants.gov/…*
SOAP Action:	*SOAP Action*
Method Namespace URI:	*Method Namespace URI*
Method Name(s):	• assignAgencyTrackingNumber() • confirmApplicationDelivery() • getApplication() • getApplicationList() • notifyApplicationAvailable()
WSDL URL:	*WSDL URL*

The following table provides a high-level description of the Grants.gov application retrieval and associated Web services method names. A detailed Web service interface specification is provided in the Agency Integration Toolkit Design Specifications document.

Table B -2 High Level Description of Grants.gov Application Retrieval

Method Name	Description	Parameters
assignAgencyTrackingNumber()	Used by an agency to notify Grants.gov of an agency's internal tracking number for a given application. Will be implemented by Grants.gov.	**Input**: • Grants.gov tracking number; • Agency tracking number **Output**: • None.
confirmApplicationDelivery()	Used by an agency to notify Grants.gov that it has successfully downloaded a given application. Can contain multiple confirmations. Will be implemented by Grants.gov	**Input:** • Grants.gov tracking number **Output**: • None.

getApplication()	Used by an agency to retrieve a specific application identified by a Grants.gov tracking number. Will be implemented by Grants.gov.	**Input**: • Grants.gov Tracking Number **Output**: • A message containing the application data.
getApplicationList()	Used by an agency to retrieve a list of applications that are ready to be downloaded. Will be implemented by Grants.gov	**Input**: • An indicator of whether the agency wants a list of all applications, only those applications that have not already been downloaded, or only those applications that have been already downloaded. **Output**: • A list containing mainly Header and Footer[8] information, such as Grants.gov Tracking number, download status, opportunity number, and CFDA number.
notifyApplicationAvailable():	Used by Grants.gov to notify an agency that a given application is ready to be downloaded. Must be implemented by any agency that wants Grants.gov to notify it that an application is ready to be downloaded. Otherwise, the agency will have to determine which applications are ready to be downloaded by calling the interface *getApplicationList*	**Input**: • None. **Output**: • Grants.gov Tracking Number.

[8] *See* *Appendix F Application Schemas* *for more information regarding the Header and Footer application data.*

Appendix C SOAP Message Structure

Messages used in the system-to-system message exchange are formatted according to the SOAP with Attachments (SwA) specifications[9]. Some applications may not have any attachments; however, the corresponding SOAP messages are still encapsulated in MIME via the transport protocol. The SOAP message structure is divided into four logical components:

- **Message Header:** This piece contains information that defines the nature and the context of the message. It is part of the SOAP message, which is a unique MIME part. The message header encapsulates meta-data about the application. For example, the system/user credentials are included in the message header. The `Content-Type` of this MIME part must be set to `text/xml; charset="UTF-8"`.
- **Message Body:** The message body is also part of the SOAP Message. This piece contains information intended for processing the request.
- **Structured Data for Message:** This optional section contains an XML file that describes the structured data for the application as defined by the application schema and related documentation. The `Content-Type` of this MIME part must be set to `application/xml`. Currently this section is only required for the application, and should not be included elsewhere.
- **Attachments:** This *optional* section includes attachments intended for the functionality of the message. Each part consists of a binary (native format) document, such as a PDF or Word document. The `Content-Type` for each of these MIME parts must be set to their corresponding content type, such as `application/pdf`, or `application/word`.

MIME View

The following illustration depicts, at a high level, the structure of the application message, which is compliant with SOAP with Attachments (SwA) specifications. This package consists of a MIME envelope that embeds other MIME parts. The first part of the multipart MIME message is the XML SOAP document; the subsequent parts contain any attached data.

9 *http://www.w3.org/TR/SOAP-attachments*

SOAP Message Structure

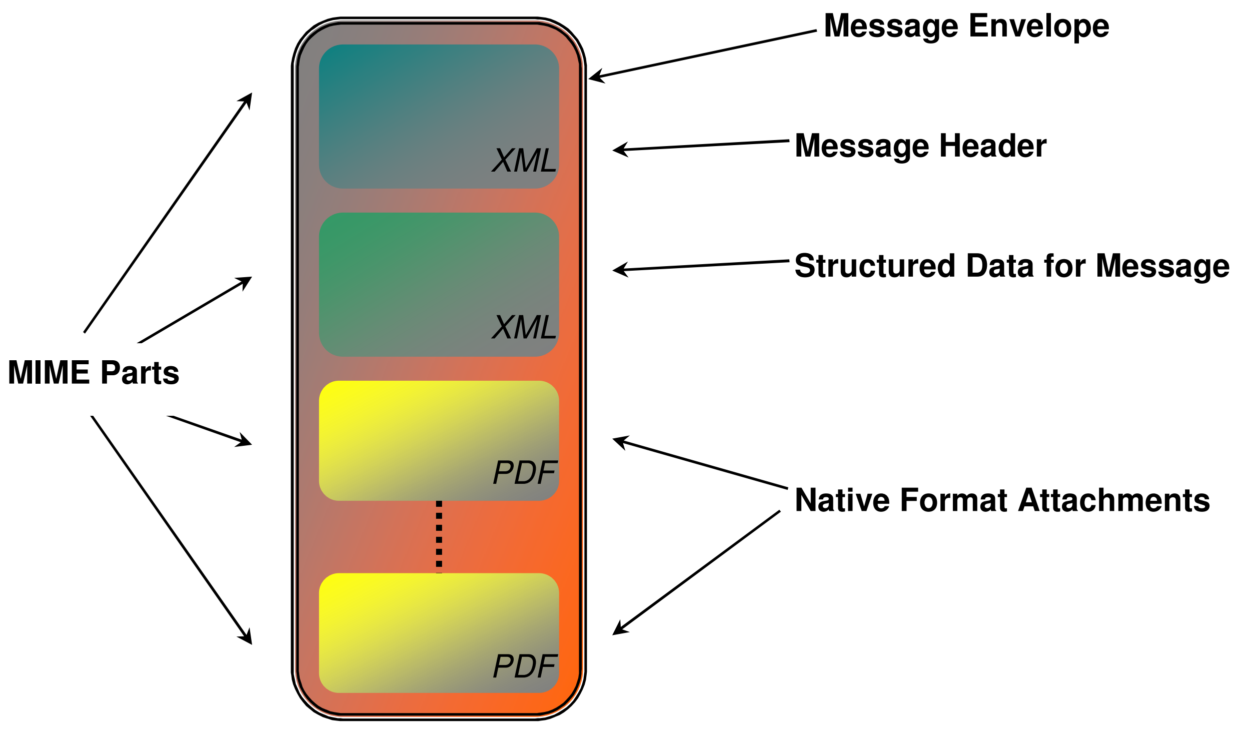


Figure C- 1 SOAP Message Structure

Mime Headers

The `Content-Type` header of the MIME envelope must be set to `multipart/related`. The `type` attribute must match the `Content-Type` of the Message Header body part. The `start` parameter (defined in the MIME Multipart/Related Content-type specifications **[RFC2387]**) will be present and will reference the body part that contains the Message Header. In addition, The `Content-Id` header is optional and is not interpreted. The following fragment illustrates this convention.

```
Content-Type: multipart/related; type = "text/xml";
boundary="boundaryValue"; start="MimeHeader"

--boundaryValue
Content-ID: <MimeHeader>
...
```

The MIME parts of the message must be described using the following MIME headers:

- **`Content-Id:`** This header attribute uniquely identifies MIME parts. It is necessary to differentiate parts and allow cross-referencing between parts. Each body part MUST be uniquely named.

- `Content-Type:` This header attributes are used to specify the media type and subtype of data in a message part. Only official types defined by IETF should be used.

Structure of the Message Header

The message header is the first body part to be interpreted in the Message Processor. The illustration below depicts its general structure. The MIME part contains the SOAP envelope. The envelope is comprised of the SOAP Header and the SOAP Body. The SOAP header contains the ebXML Header and the Application Header sections. Identically, the SOAP Body contains the ebXML Body and Application Body sections.

Structure of the Message Header

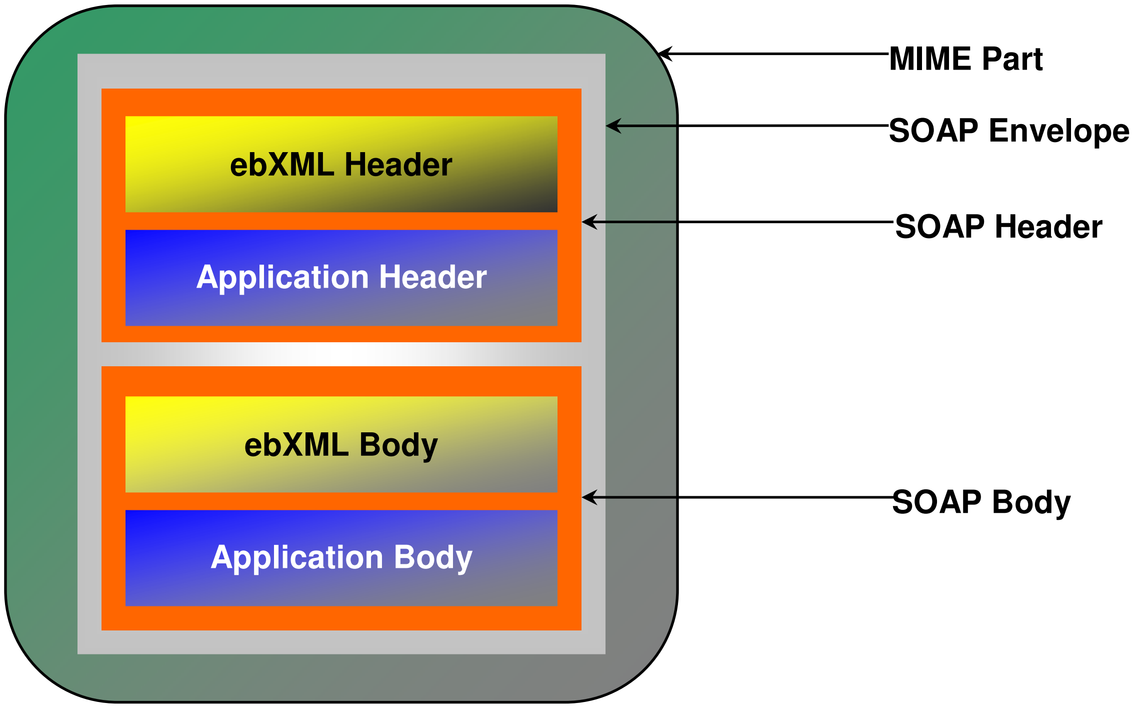


Figure C- 2 Structure of the Message Header

SOAP Envelope

The data contained in the MIME part MUST be an (XML) SOAP message, as defined in SOAP v1.1 specifications. The root element MUST correspond to the SOAP `envelope` element, which contains a `Header` and a `Body` element, as shown below.

ebXML Header

Since ebXML is part of the proposed solution, the following elements and attributes must be included to comply with the schema. The ebXML `MessageHeader` is the first element contained in the `SOAP:Header`. It has two attributes:

- `version:` Its value must be `2.0`. It refers to the version of the ebXML specifications used,

- `mustUnderstand:` Its value MUST be `1`. It means that the information contained in this element MUST be understood or rejected.

A `MessageHeader` element must encapsulate the following elements:

- `From:` Provides information about the sender of the message.
  It must contain the `PartyId` element with a `type` attribute. For the early implementation, the value of this attribute MUST be the URI `urn:x12.org:I05:01`. It is a custom-made URI that refers to the Interchange ID Qualifier `01` of the ANSI X12 I05 registries (described page 1635 of ASC X12 Standard 4030). In other words, it defines that the text value embedded in the PartyId element should be a DUNS number.
- `To:` Provides information about the intended recipient of the message.
  This element follows the same structure as the `From` element above.
- `CPAId:` This element references the partner agreement that defines interactions between the parties.

  At the time this document was written, the partner agreement piece had not been investigated. Any non-null values can be used and will be ignored for the initial implementation.
- `ConversationId:` from **[ebMS2]**

  The ConversationId element is a string identifying the set of related messages that make up a conversation between two Parties. It MUST be unique within the context of the specified CPAId. The Party initiating a conversation determines the value of the ConversationId element that will be reflected in all messages pertaining to that conversation. The ConversationId enables the recipient of a message to identify the instance of an application or process that generated or handled earlier messages within a conversation. It remains constant for all messages within a conversation.
- `Service:` This element identifies the service that acts on the message.
- `Action:` This element identifies a process within the `Service` that processes the message.
- `MessageData:` This element should contain two children:
  - `MessageId:` It should be a globally unique identifier conforming to MessageId **[RFC2822]**.
  - `Timestamp:` It should comply with the standard ISO 8601 and follow the `YYYY-MM-DDThh:mm:ssTZD` format (e.g., `2003-04-04T14:25:27-05:00`).

```
<MessageHeader version="2.0" mustUnderstand="1">
     <From>
          <PartyId type=" urn:x12.org:I05:01">Grants.gov DUNS
number
          </PartyId>
     </From>
     <To>
          <PartyId type=" urn:x12.org:I05:01"> Agency DUNS number
          </PartyId>
     </To>
     <CPAId> CPAId </CPAId>
     <ConversationId> 934523405 <ConversationId>
     <Service> urn of service </Service>
     <Action> Action </Action>
     <MessageData>
          <MessageId> mid:UUID-1 </MessageId>
          <Timestamp> 2003-04-04T14:25:27-05:00 </Timestamp>
     </MessageData>
</MessageHeader>
```

Figure C- 3 Sample Message Header

Detailed information about the sections related to ebXML can be found in **[ebMS2]**.

The schema that validates the ebXML elements is located at http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd.

ebXML Body

The schema that validates the ebXML elements is located at http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd.

For an application, the ebXML body consists of the `Manifest` element. It is included in the `SOAP:body` element.

The ebXML `Manifest` MUST have at least the attribute:

- `version:` its value MUST be `2.0`. It refers to the version of the ebXML specifications used,

The `Manifest` element describes the content of the payload. It MUST contain a list of `Reference` elements, which identifies all the other body parts of the application. A `reference` element should have three attributes (defined in **[XLink]** http://www.oasis-open.org/committees/ebxml/msg/schema/xlink.xsd):

- `type:` its value should be `simple`.
- `href:` it should follow the "cid" URI scheme and points to an existing part of the application.
- `role:` This attribute provides a semantic value to the document referred to.

Application XML Structure

The application XML structure varies depending on the grant opportunity. The various schemas that define the application XML structure can be found in Appendix F Application Schemas.

The actual Web services interface specification details are provided in the Agency Integration Toolkit Design Specification document.

Appendix D Skeleton Application Message

The following provides a *sample* Grants.gov application message with native format PDF and Word document attachments:

```
MIME-Version: 1.0
Content-Type: Multipart/Related; boundary=MIME_boundary; type=text/xml;
        start=”PrimaryMIMEPart”
Content-Description: an optional message description.

--MIME_boundary
Content-Type: text/xml; charset=UTF-8
Content-Transfer-Encoding: 8bit
Content-ID: <PrimaryMIMEPart>

<?xml version='1.0' ?>
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Body>
..
A single XML Document combining Core SF-424 and any related/supporting XML
documents such as SF-424A-D
..
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

--MIME_boundary
Content-Type: application/pdf
Content-Transfer-Encoding: base64
Content-ID: <Content Id>

...Actual PDF Binary Data...
--MIME_boundary-
Content-Type: application/word
Content-Transfer-Encoding: base64
Content-ID: <Content Id2>

...Actual Word Binary Data...
--MIME_boundary-
```

Figure D- 1 Sample Skeleton Application Message

In this example, the "Content-Type" header line has been continued across two lines so the example prints easily. SOAP message senders should send headers on a single long line.

Appendix E SF-424 and Related Forms

In support of the Grants.gov vision to "define and implement simplified standard processes and standard data definitions for Federal grant customer interactions", HHS has identified data standards it will use for grants forms government-wide. HHS and its partner agencies[10] have agreed to use SF-424, a standard form currently used by about 100 grant programs.

The SF-424 Form has various related components, including, SF-424A-D. The following table describes these components:

Table E- 1 SF-424 and Related Forms

Form Identifier	Form Name	Description
SF-424	Application for Federal Assistance (Cover Letter)	OMB Standard Form 424. SF-424 is a standard form used by applicants as a required *face sheet* for pre-applications and applications submitted for Federal assistance. It is used by Federal agencies to obtain applicant certification that States which have established a review and comment procedure in response to Executive Order 12372 and have selected the program to be included in their process, have been given an opportunity to review the applicant's submission.
SF-424A	Budget Information (Non-Construction)	OMB Standard Form 424A, OMB Approval Number 0348-0044. SF-424A is designed to capture budget information for *non-construction* awards from one or more grant programs. It is to be used for the following types of applications: (1) "New" (means a new [previously un-funded] assistance award); (2) "Continuation" (means funding in a succeeding budget period which stemmed from a prior agreement to fund); and (3) "Revised" (means any changes in the Federal Government's financial obligations or contingent liability from an existing obligation). If there is no change in the award amount, there is no need to complete this form. Certain Federal agencies may require only an explanatory letter to

10 *Some twenty-six (26) departments and agencies with approximately nine hundred (900) programs collect grant information today. Various program requirements and administrative differences necessitate the constant updating of procedures, and disparate data standards and business processes require redundant data entry, often resulting in inaccurate application submissions.*

The federal Grants.gov initiative was undertaken for the development of a one-stop electronic grant portal where potential grant recipients will receive full service electronic grant administration. Eleven of the twenty-six departments and agencies have been designated as supporting partners for the Grants.gov initiative, of which the Department of Health and Human Services (HHS) is the managing partner. For more information, see http://www.grants.gov/pmo.html.

		effect minor (no cost) changes. If you have questions, please contact the Federal agency.
SF-424B	Standard Assurances (Non-Construction Programs)	OMB Standard Form 424B, OMB Approval Number 0348-0040. SF-424B is a collection of required certifications and assurances. The form is used, for example, to certify that the applicant has the "legal authority to apply for Federal assistance, and the institutional, managerial and financial capability (including funds sufficient to pay the non-Federal share of project costs) to ensure proper planning, management and completion of the project described in this application"
SF-424C	Budget Information (Construction Programs)	OMB Standard Form 424C, OMB Approval Number 0348-0041. SF-424C is designed to capture budget information for construction awards from one or more grant programs. This form is used in place of SF-424A if a significant amount (equal to or greater than 50%) of the Federal funds requested are for construction projects (e.g., build-outs, alterations, upgrades and renovations of existing facilities)
SF-424D	Standard Assurances (Construction Programs)	OMB Standard Form 424D, OMB Approval Number 0348-0042. SF-424D is a collection of required certifications and assurances.

Copies of the SF-424A, SF-424B, SF-424C and SF-424D can be found on the OMB website: http://www.whitehouse.gov/omb/grants/grants_forms.html, as well as the Grant.gov *XML Schema and Implementation Guide* for more information.

Appendix F Application Schemas

Based on the SF-424 forms described in Appendix SF 424 and Related Forms, a corresponding set of SF 424 XML schemas has been developed by the Grants.gov data modeling team, as described below:

Table F- 1 Application Schemas

XML Schema Name	Description
SF-424 Awarding Agency Schema	This schema conforms to OMB Standard (Core) Form 424, E-Grants Pilot Version 4/03. It includes the SF-424 Schema for Submitting Organization, and adds several elements.
SF-424A Schema	This schema conforms to OMB Standard Form 424A, OMB Approval Number 0348-0044.
SF-424B Schema	This schema conforms to OMB Standard Form 424B, OMB Approval Number 0348-0040.
SF-424C Schema	This schema conforms to OMB Standard Form 424C, OMB Approval Number 0348-0041
SF-424D Schema	This schema conforms to OMB Standard Form 424D, OMB Approval Number 0348-0042
SF-424 Submitting Organization Schema	This schema contains data elements that identify a submitting organization.
Attachment Schema	This schema defines file attachments and narratives across all SF-424x forms. Rather than be included in the pertinent SF-424x XML documents, all file attachments and narratives will be placed in a single, separate XML document included in the overall application XML document. Named attachments include Project Narrative; Budget Narrative; and Debarment/Delinquency Explanation, and Other.
Global Schema	This schema contains common/shared XML constructs that are used by multiple SF-424x schemas, as well as the header/footer schemas.
Header Schema	Contains grant submission summary information, including Agency Name, CFDA Number, Activity Title, Opportunity ID, Opportunity Title, Competition ID, Opening Date, Closing Date, and Submission Title. Headers will be received from Submitting "Applicant" Organizations, as well as sent to Awarding Agencies.
Footer Schema	Contains tracking information such as received date and time, submitter name, and Grants.gov tracking number. Footers will ONLY be sent to Awarding Agencies.

Additional schemas will be created to describe any form(s) that capture agency specific data elements. Please refer to the latest version of the Grants.gov Core Data XML Schemas, as well as the Grant.gov *XML Schema and Implementation Guide* for more information.

Appendix G Data Element Descriptions

This section lists common Grants.gov data element names with their corresponding descriptions. Detailed data model information can be found on Grants.gov website at the following URL: http://www.grants.gov/docs/CoreDataElements.pdf.

Table G- 1 Common Grants.gov Data Element Names

#	Data Element Name	Data Element Description
1	Agency application download format	The specified format an agency wishes to download an application (e.g., PDF, XML etc.)
2	Agency CFDA prefix	Part of the CFDA # that is required by an agency to resister their profile
3	Agency code	The standard acronym for the agency and is a required field for the agency's profile registration
4	Agency e-mail notification to superuser guidance	A drop down selection during agency profile registration allowing a super user to receive notifications as to when grant applications are submitted
5	Agency login	The login name of an agency to register their profile and publish applications
6	Agency login password	Password to agency login
7	Agency name	The name of the agency as it is commonly referred to
8	Agency POC Address line 1	The agency's POC address line 1 during agency profile registration
9	Agency POC address line 2	The agency's POC address line 2 during agency profile registration
10	Agency POC city	The agency's POC city during agency profile registration
11	Agency POC e-mail	The agency's POC e-mail during agency profile registration
12	Agency POC name	The agency's POC name during agency profile registration
13	Agency POC state	The agency's POC state during agency profile registration
14	Agency POC tel #	The agency's POC telephone # during agency profile registration
15	Agency POC zip code	The agency's POC zip code during agency profile registration
16	Agency specific dataset (if required)	Is a special attachment created by an agency that covers additional applicant data required beyond the core data
17	Agency tracking #	Is the unique identifier created by the agencies that tracks applications submitted to an agency

18	AOR Confirmation (Y/N)	Is a confirmation field from BPN/CCR that confirms/denies that an application has the correct AOR assigned after submission to Grants.gov
19	Applicant authentication login	The e-authentication login name submitted by the applicant
20	Applicant authentication password	The e-authentication password submitted by the applicant
21	Applicant e-mail tracking ID	The unique ID generated in Grants.gov that tracks the e-mails used for informing applicants of application changes done by the agencies.
22	Application agency name	The name of the agency that assembles the application
23	Application CFDA #	The CFDA # associated with the application
24	Application CFDA title	The title (short description) of the CFDA # (or program)
25	Application close date	The close deadline to submit an application
26	Application contact information association	The link between an application and a person on the list of contacts within Grants.gov
27	Application funding opportunity ID	The Funding Opportunity ID that's associated with an application
28	Application funding opportunity title	A title or short description of the funding opportunity ID
29	Application instructions	A separate document that explains how to apply for a particular application
30	Application name	The name for the application
31	Application open date	The date that an agency will start accepting applications for review
32	Application POC address line 1	The 1st address line of the application's POC
33	Application POC address line 2	The 2nd address line of the application's POC
34	Application POC City	City of the application's POC
35	Application POC e-mail	e-mail of the application's POC
36	Application POC name	Name of the application's POC
37	Application POC state	State of the application's POC
38	Application POC Tel #	Tel # of the application's POC
39	Application POC Zip	Zip code of the application's POC
40	Attachment Types (Project Narrative, Budget Narrative, Debarment Explanation, Other) - if required	Is an optional part of the core data that consists of 4 forms: Budget Narrative, Project Narrative, Debarment Explanation, and Other.
41	CFDA #	is the unique identifier for an agency's grant program. (ex. 83.554 for Assistance to Firefighters Grant)
42	CFDA # confirmation field (Y/N)	The field confirming/denying whether the application has a valid CFDA # associated with it. This confirmation is retrieved from CCR/BPN.
43	CFDA title	The title (short description) of the CFDA # (or program)

44	Competition ID	Is a Grants.gov generated ID that allows further distinction of the funding opportunity ID. The funding opportunity - competition ID combination allows applications with the same funding opportunity # to be assigned unique identifiers. The competition ID allows unique identification of the application with different open and closed dates.
45	Credential provider ID	Uniquely identifies the Grants.gov approved credential provider to allow e-authentication.
46	Current receipt status	Is a status displayed to the grants applicant informing him or her the location of their grant application
47	Date/time of agency receipt	Is the date/time an agency has received the application submitted by the applicant
48	Date/Time of Grants.gov receipt	Is the date/time Grants.gov has received the application after applicant submission
49	DUNS #	This is the 9 digit unique identifier created by Dun and Bradstreet Inc. The D&B D-U-N-S Number is a unique nine-digit identification sequence, which provides unique identifiers of single business entities, while linking corporate family structures together. This field is a new field added to the SF-424.
50	DUNS # confirmation(Y/N)	Confirms or denies that the DUNS # submitted by the applicant exists within CCR/BPN
51	DUNS organization name	Is the DUNS registered name associated with the applicant's agency DUNS #
52	E-authentication confirmation field (Y/N)	Confirms or denies that an applicant's authentication login/password exists within an approved credential provider
53	Final applicant submission confirmation field (Y/N)	Confirms or denies that an applicant understood the application's funding opportunity # and title, CFDA # and title, and grant sponsoring agency prior to submission
54	SF-424 and form selections (SF-424A-D, Project Narrative, Budget Narrative, Debarment explanation, other, etc.)	These are the basic OMB approved forms for Application for Federal Assistance. SF-424 (as of 4/8/03) is the latest OMB approved form and will include the e-mail address, 9 digit DUNS #, fax, country, and Insurance Acceptance Checkmark. SF-424A addresses budget information for Non-construction programs. SF-424B addresses assurance for non-construction programs. SF-424C addresses budget information for construction programs and SF-424D addresses assurances for construction programs.
55	Funding opportunity ID	Is the unique identifier for grant announcements on FedGrants.gov. (ex. ED-GRANTS-040903-002).
56	Funding opportunity title	Is the name given to the Funding Opportunity when posted on Fedgrants.gov

57	Grants.gov tracking #	This is the unique identifier that is generated within Grants.gov and will be the unique identifier for all grant applications being processed through grants.gov
58	SF-424 Country (will be on new SF-424)	Country code of the applicant (new field on SF-424)
59	SF-424 Dataset - Fields 1 - 18c	SF-424 fields 1-18C
60	SF-424 e-mail (will be on new SF-424)	e-mail address of the application's POC from a applicant's perspective (new field on SF-424)
61	SF-424 Fax (will be on new SF-424)	fax number of the application's POC from an applicant's perspective (new field on SF-424)
62	SF-424 Assurance Acceptance Checkmark (will be on new SF-424)	Application checkmark that confirms that the applicant will comply with those requirements and other terms and conditions if it receives an award
63	SF-424 Fields 18d-e	Signature of authorized representative and date signed for the application
64	Sub-agency POC city (if required)	City of the sub-agency's POC
65	Sub-agency application download format (if required)	The specified format a sub- agency wishes to download an application (e.g., .pdf, XML etc.)
66	sub-agency CFDA prefix (if required)	The CFDA prefix of the sub-agency
67	Sub-agency code (if required)	The sub-agency code of the agency
68	Sub-agency e-mail notification to super-user guidance (if required)	e-mail notification guidance of the sub-agency
69	Sub-agency name (if required)	Name of the sub-agency
70	Sub-agency parent agency (if required)	Name of the sub-agency's parent organization
71	Sub-agency POC address line 1 (if required)	Address line one of the sub-agency's POC
72	Sub-agency POC address line 2 (if required)	Address line two of the sub-agency's POC
73	Sub-agency POC name (if required)	Point of contact name for a sub-agency
74	Sub-agency POC state (if required)	State of the sub-agency's POC
75	Sub-agency POC zip code (if required)	Zip code of the sub-agency's POC
76	Submission title	A descriptive title the applicant will create to represent their application
77	User’s address line 1	The agency user's address line 1
78	User’s address line 2	The agency user's address line 2
79	User’s agency	Name of the agency (or sub-agency) that the agency user belongs to
80	User’s city	The agency user's city
81	User’s e-mail	The agency user's e-mail address
82	User’s name	The agency user's name
83	User’s phone	The agency user's phone #
84	User’s role	The agency user's role
85	User’s state	The agency user's state
86	User’s zip code	The agency user's zip
87	Validation confirmation (Y/N)	A confirmation field submitted to the applicant informing him or her that their application has passed

		the validation process
88	Validation failure info (if needed)	Reason for failure codes and descriptions communicated to the applicant if the application failed the validation process

Appendix H Bibliography/Resources

The following list of writings provide additional Grants.gov specific and general, supplementary information:

Grants.gov Specific

- Grants.gov Vision and Goals
- Integration Kit Survey Analysis for Grants.gov
- Core Data XML Schemas
- XML Schema and Implementation Guide
- Agency Integration Toolkit Design Specifications

General

- What is SOAP? See a Flash Movie.
- [SOAP] SOAP (Simple Object Access Protocol) Version 1.1 (http://www.w3.org/TR/SOAP/)
- [XML] Extensible Markup Language (XML) 1.0 (http://www.w3.org/TR/1998/REC-xml-19980210)
- [ebXML] Electronic Business using Extensible Markup Language (ebXML) (http://www.ebxml.org)
- [ebMS2] ebXML messaging specification 2.0 http://www.ebxml.org/specs/ebMS2.pdf
- [MultipartRelated] The MIME Multipart/Related Content-type (http://www.ietf.org/rfc/rfc2387.txt)
- [MIME1] Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies (http://www.ietf.org/rfc/rfc2045.txt)
- [CID] Content-ID and Message-ID Uniform Resource Locators http://www.ietf.org/rfc/rfc2111.txt
- [URI] Uniform Resource Identifiers (URI): Generic Syntax (http://www.ietf.org/rfc/rfc2396.txt)
- [RFC2557] MIME Encapsulation of Aggregate Documents, such as HTML (MHTML) (http://www.ietf.org/rfc/rfc2557.txt)
- [XMLBASE] XML Base http://www.w3.org/TR/xmlbase
- [HTTP] Hypertext Transfer Protocol -- HTTP/1.1 (http://www.ietf.org/rfc/rfc2616.txt)
- Application Server Comparison Matrix