



System Integration Document

Grants.gov System

V 2.0

April 27, 2007

Table of Contents

1. Introduction.....	5
1.1 Purpose.....	5
1.2 Scope.....	5
2. Overview of the Grants.gov System-to-System Interface Protocol.....	6
2.1 System-to-System Integration Overview	7
2.1.1 Client-Server Relationship (Polling) Model.....	7
2.1.2 Message Interaction Pattern	7
2.2 Grants.gov Web Services Invocation Framework.....	8
3. System-to-System Interface Specifications	8
3.1 Applicant System-to-System Operations	8
3.1.1 Typical Applicant-Grants.gov Interaction Sequence.....	8
3.2 Applicant System-to-System WSDL Specifications	9
3.2.1 Applicant WSDL Schema	9
3.2.2 Applicant WSDL Directives – Schema Imports.....	10
3.2.3 Applicant WSDL Service.....	11
3.2.4 Applicant WSDL Port Types (Network Endpoints).....	12
3.2.5 Applicant Messages, Elements, and Types	12
3.3 Agency System-to-System Operations.....	16
3.3.1 Typical Agency-Grants.gov Interaction Sequence	16
3.4 Agency System-to-System WSDL Specifications	16
3.4.1 Agency WSDL Schema.....	16
3.4.2 Agency WSDL Directives – Schema Imports	17
3.4.3 Agency WSDL Service	18
3.4.4 Agency WSDL Port Types (Network Endpoints)	19
3.4.5 Agency Messages, Elements, and Types	19
3.5 Error Handling	23
3.5.1 SOAP Fault Structure.....	24
3.5.2 Grants.gov Error Codes and Messages.....	24
3.5.3 Sample SOAP Fault Message.....	25
4. Web Services Security	25
4.1 Secure Socket Layer (SSL) and Digital Certificates	25
4.2 Agency/Applicant System to Grants.gov Authentication.....	25
Appendices	27
Appendix A: Web Services Description Language (WSDL) Overview	27
Appendix B: WSDL Bindings with SOAP	28
Appendix C: SOAP Message Structure Overview.....	30
Appendix D: Sample SOAP Message with Attachment	32
Appendix E: Acronyms & Definitions.....	38

List of Tables

Table 1. SOAP Fault Sub-elements	24
Table 2. Code Values.....	24

Table 3. Code Values	24
Table 4. Code Values	27

List of Figures

Figure 1. Grant Lifecycle View	6
Figure 2. Grants.gov Interfaces.....	7
Figure 3. Typical Request/Response Interaction Between Applicant and Grants.gov System	9
Figure 4. Applicant WSDL Schema	10
Figure 5. Schema Imports Found Within the Applicant WSDL.....	11
Figure 6. Services.....	11
Figure 7. Port	12
Figure 8. Applicant Integration Port Type.....	12
Figure 9. GetApplicationListRequestMessage	13
Figure 10. GetApplicationListResponseMessage.....	13
Figure 11. GetApplicationStatusDetailRequestMessage	13
Figure 12. GetApplicationStatusDetailResponseMessage.....	13
Figure 13. GetOpportunityListRequestMessage.....	14
Figure 14. GetOpportunityListResponseMessage	14
Figure 15. SubmitApplicationRequestMessage.....	14
Figure 16. SubmitApplicationResponseMessage	14
Figure 17. ApplicantInformationType	15
Figure 18. GrantsGovApplicationStatusType.....	15
Figure 19. OpportunityInformationType	15
Figure 20. Typical Request/Response Interaction Between an Agency and Grants.gov System..	16
Figure 21. Agency WSDL Schema.....	17
Figure 22. Schema Imports Found Within the Agency WSDL	18
Figure 23. AgencyIntegrationServices.....	18
Figure 24. AgencyIntegrationSoapPort	19
Figure 25. Agency Integration Port Type	19
Figure 26. AssignAgencyTrackingNumberMessage	20
Figure 27. AssignAgencyTrackingNumberResponse Message.....	20
Figure 28. ConfirmApplicationDeliveryRequestMessage.....	20

Figure 29. ConfirmApplicationDeliveryResponseMessage	20
Figure 30. GetApplicationListRequestMessage	20
Figure 31. GetApplicationListResponsetMessage.....	21
Figure 32. GetApplicationRequestMessage.....	21
Figure 33. GetApplicationResponseMessage	21
Figure 34. GetApplicationZipRequestMessage	21
Figure 35. GetApplicationZipResponseMessage.....	21
Figure 36. Manifest Type.....	22
Figure 37. ApplicationInformation Type.....	22
Figure 38. Item Type.....	22
Figure 39. EncodingEnum Type	23
Figure 40. GrantsgovApplicationStatus Type.....	23
Figure 41. Main Structure of a WSDL Document.....	27
Figure 42. A Request-Response WSDL Using SOAP Binding.....	28
Figure 43. WSDL Binding with SOAP	29
Figure 44. SOAP Message Structure	30

1. Introduction

Applicant (Grantee) organizations are often faced with overwhelming challenges when applying for federal government grants, regardless of which Agency (Grantor) 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, the grant information is typically not shared among the programs; also, 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 of the Grants.gov initiative defined by consensus among the grant-making agencies 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.

Implementing the Grants.gov Storefront is only a part of the process of deploying an effective government-wide grants application system. Other significant challenges in ensuring system success involve supporting individuals and organizations at each end of Grants.gov external interfaces – the applicants and the grant-making agencies. Grants.gov must be adopted and used by these stakeholders if the system is to be successful.

1.1 Purpose

The purpose of the System Integration Document is to provide a description of the services, parameters, schemas, and protocols of the System-to-System interface used by agencies and applicants to exchange data with the Grants.gov system.

1.2 Scope

The System Integration Document describes the System-to-System interfaces between an Applicant organization and the Grants.gov “Front Office” operations, as well as the System-to-System interfaces between the Grants.gov “Front Office” operations and an agency’s “Back Office” operations; i.e., the interfaces between the Grants.gov system and the grantee and grantor facing aspects of the grant lifecycle.

The following figure represents a general, two-sided overview 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”).

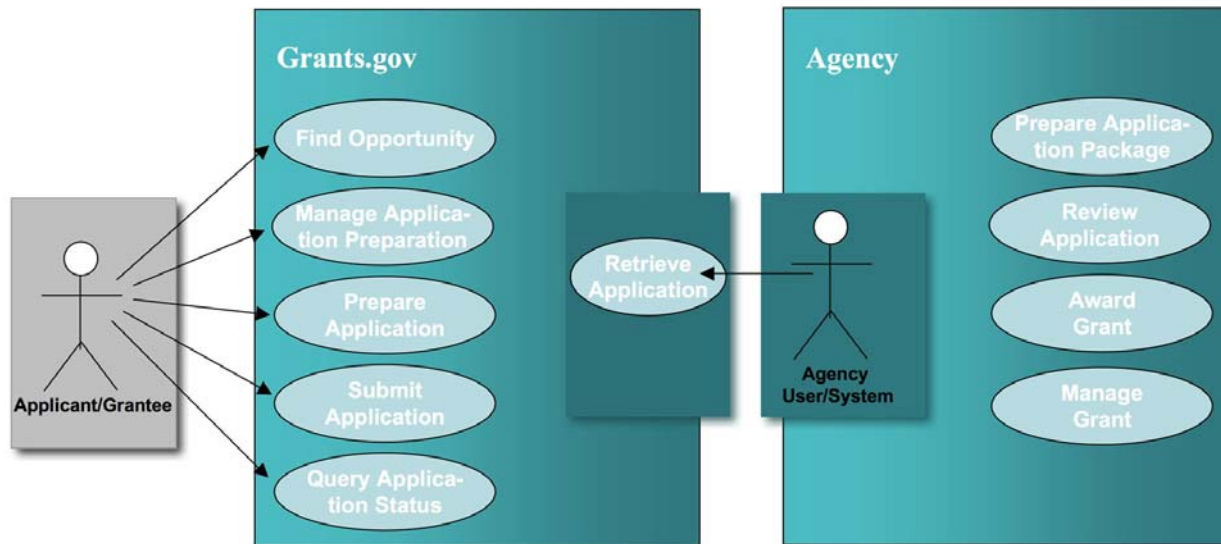


Figure 1. Grant Lifecycle View

While Applicant and Agency organizations can electronically interface with Grants.gov via either System-to-System or Person-to-System interfaces, this document is limited in Scope to a discussion of the System-to-System interfaces. An overview of the Grants.gov System-to-System Interface Protocol is provided in the next section.

2. Overview of the Grants.gov System-to-System Interface Protocol

Applicant and Agency organizations can electronically interface with Grants.gov via the following two methods:

- System-to-System interfaces
- Person-to-System interfaces

A further distinction is made between inbound and outbound interfaces. Applicants are said to interact with Grants.gov via inbound interfaces, while Agencies are said to interact with Grants.gov via outbound interfaces.

A Person-to-System inbound interface is basically a set of Grants.gov-provided Graphical User Interface (GUI's) through which an applicant can download a grant application, complete it, and submit (file) it to Grants.gov. Similarly, a Person-to-System outbound interface is basically a set of Grants.gov provided Graphical User Interfaces (GUI's) through which an agency user can pull (download) the submitted application in a PDF or Adobe format with any related attachments in their native format. Details of these interfaces have been provided in the Applicant and Agency Integration Toolkit Documents.

Both inbound and outbound System-to-System interfaces are SOAP Client to Web Services exchange (inbound), and Web Services to SOAP Client exchange (outbound), where Grants.gov has implemented the Web Services that Applicant and Agency systems can poll; i.e., all System-to-System interactions are initiated by either the Applicant or Agency system. Grants.gov does not initiate any of the interactions.

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. Once the SSL message exchange “hand shake” is established, the transfer protocol and transfer of SOAP messages occurs in the same manner.

Figure 2 below provides a high level depiction of the Grants.gov inbound and outbound interfaces.

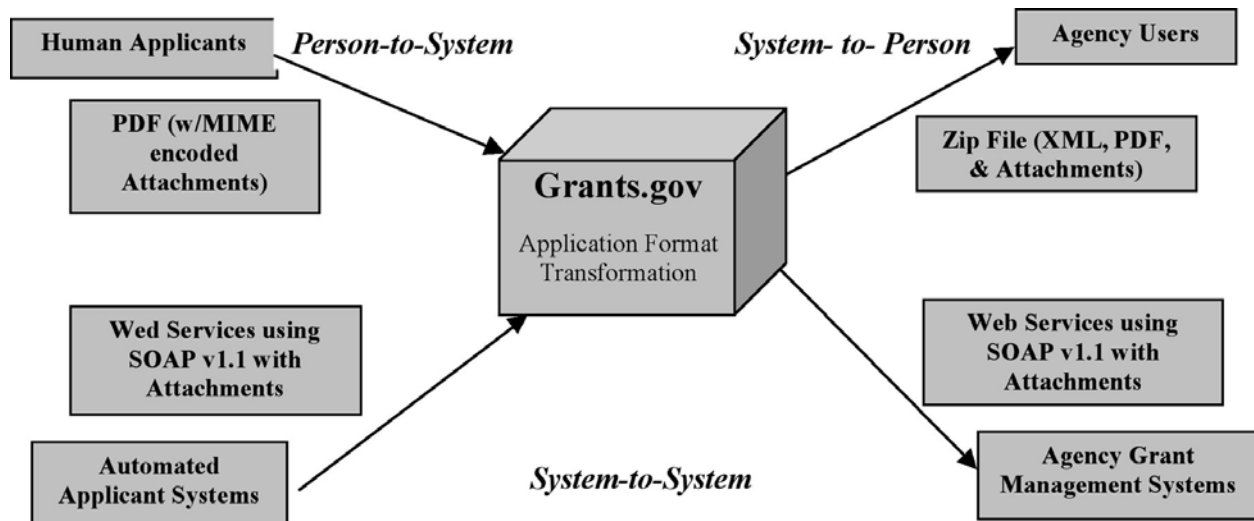


Figure 2. Grants.gov Interfaces

The following sections discuss detailed system specifications for both the Applicant and Agency System-to-System interfaces.

2.1 System-to-System Integration Overview

For the outbound System-to-System interfaces, Grants.gov-to-Agency integration is achieved via a message exchange that utilizes XML Web Services using SOAP v1.1 with Attachments over HTTP with Secure Socket Layer (SSL).

While two versions of the message exchange model were initially proposed (a client-server relationship (polling) model and a peer-to-peer relationship model), only the client-server (polling) model was implemented.

In the Client-Server model, agencies will not implement their own web services that the Grants.gov system can invoke. 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 Applicant or Agency system, and the Grants.gov system, respectively.

2.1.1 Client-Server Relationship (Polling) Model

In this model, the agency has not implemented any Web Services that the Grants.gov system can invoke. The implication is that an Applicant or 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 requests.

2.1.2 Message Interaction Pattern

Any System-to-System message exchange between Grants.gov and Applicant/Agency systems will be conducted using a synchronous request/response interaction pattern. A connection will be

established directly between an Applicant or Agency system, and the Grants.gov system, instead of using a provider that would store (queue) and forward messages until they can be delivered.

2.2 Grants.gov Web Services Invocation Framework

Grants.gov uses the Apache Axis framework to implement Remote Procedure Call (RPC) Style Web Services using the SOAP v1.1 with attachments protocol. Unlike Document-style Web Services that do not follow call/response semantics, RPC-style Web Services are synchronous (i.e., the client sends the request and waits for the response until the request is processed completely). The web service receives the entire message, processes it, and returns a response message. The SOAP Body of a Document-style carries one or more XML documents within its envelope and body. The protocol places no constraint on how that document needs to be structured, which is totally handled at the application level. Document-style web service follows asynchronous processing.

To show the capability and as a proof of concept, the Grants.gov System Integration team has developed the System-to-System integration functionality in a two phased, Applicant and Agency Reference Implementation.

In the first phase, an application was developed that consumed an XML document generated by Grants.gov; parsed the document, created Java object classes for the data elements, and inserted the data into a relational database. To demonstrate the capability in this phase, several Java sever pages were developed that showed the selection, import, and display of the parsed data. The standalone application can be downloaded from the Grants.gov web site.

In the second phase, another standalone application was developed that retrieved sample XML list and documents from the local file system via SOAP stubs to simulate the Grants.gov wrappers, parsed the documents, and loaded the parsed data into a relational database. Two instances of Tomcat application servers were installed on separate PCs to provide web services message exchange functionality. The applications perform the various Agency and Applicant System-to-System message requests. Both the Applicant and Agency reference implementations can be downloaded from the Grants.gov site.

3. System-to-System Interface Specifications

This section describes the services, parameters, and schemas of the System-to-System interface used by Applicant and Agency organizations to exchange data with the Grants.gov system.

3.1 Applicant System-to-System Operations

The Applicant to Grants.gov System-to-System interface consists of the following four (4) operations, listed in order of typical invocation:

- GetOpportunityList
- SubmitApplication
- GetApplicationList
- GetApplicationStatusDetails

3.1.1 Typical Applicant-Grants.gov Interaction Sequence

Figure 3, an interaction diagram, depicts a typical request/response interaction sequence between Applicant and Grants.gov systems:

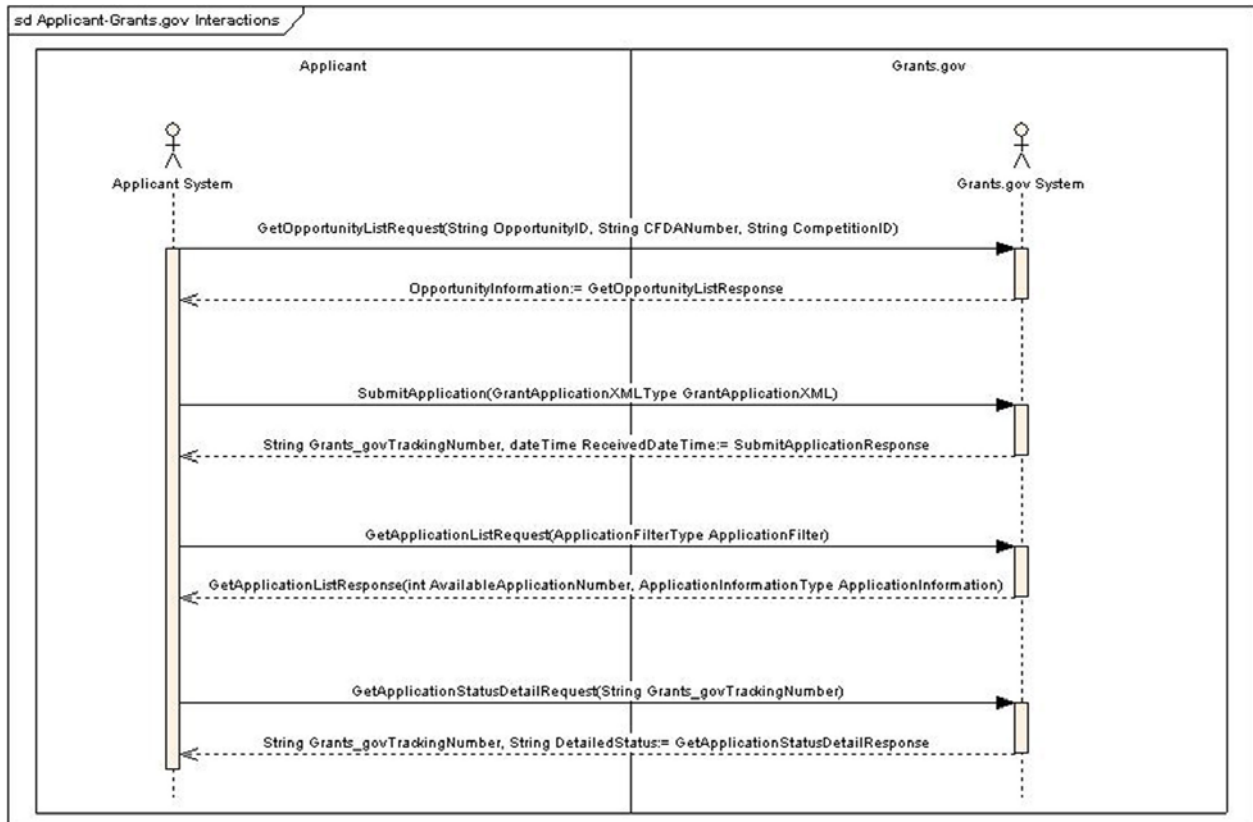


Figure 3. Typical Request/Response Interaction Between Applicant and Grants.gov System

The following sections provide further details for each of the Response/Request interactions outlined above.

3.2 Applicant System-to-System WSDL Specifications

This section provides additional Applicant System-to-System specifications via the use of the Web Services Description Language (WSDL), including an overview of the schema, schema imports, service, port types, messages, elements, and types.

For more information regarding the WSDL format and structure, please refer to the Web Services Description Language (WSDL) Overview section within the Appendix.

3.2.1 Applicant WSDL Schema

Figure 4 is a screenshot that provides an overview of the entire Applicant WSDL Schema.

Schema : <http://apply.grants.gov/WebServices/ApplicantIntegrationServices-V1.0>

Directives

- ← <http://schemas.xmlsoap.org/wsdl/> (<http://schemas.xmlsoap.org/wsdl/>)
- ← <http://schemas.xmlsoap.org/wsdl/soap/> (<http://schemas.xmlsoap.org/wsdl/soap/>)
- ← <http://apply.grants.gov/system/schemas/Global-V1.0.xsd> (<http://apply.grants.gov/system/Global-V1.0>)

Elements	Types
<ul style="list-style-type: none"> e ActivityTitle : StringMin1Max120Type e AgencyName : StringMin1Max60Type e AgencyTrackingNumber : StringMin1Max240Type e CFDANumber : StringMin1Max15Type e ClosingDate : date e CompetitionID : StringMin1Max100Type e GetApplicationListRequest e GetApplicationListResponse e GetApplicationStatusDetailRequest e GetApplicationStatusDetailResponse e GetOpportunityListRequest e GetOpportunityListResponse e Grants_govTrackingNumber : StringMin1Max240Type 	<ul style="list-style-type: none"> ApplicationInformationType OpportunityInformationType GrantsGovApplicationStatusType : xsd:string

Attributes	Groups

Figure 4. Applicant WSDL Schema

3.2.2 Applicant WSDL Directives – Schema Imports

Figure 5 is a screenshot that provides a listing of the schema imports found within the Applicant WSDL.

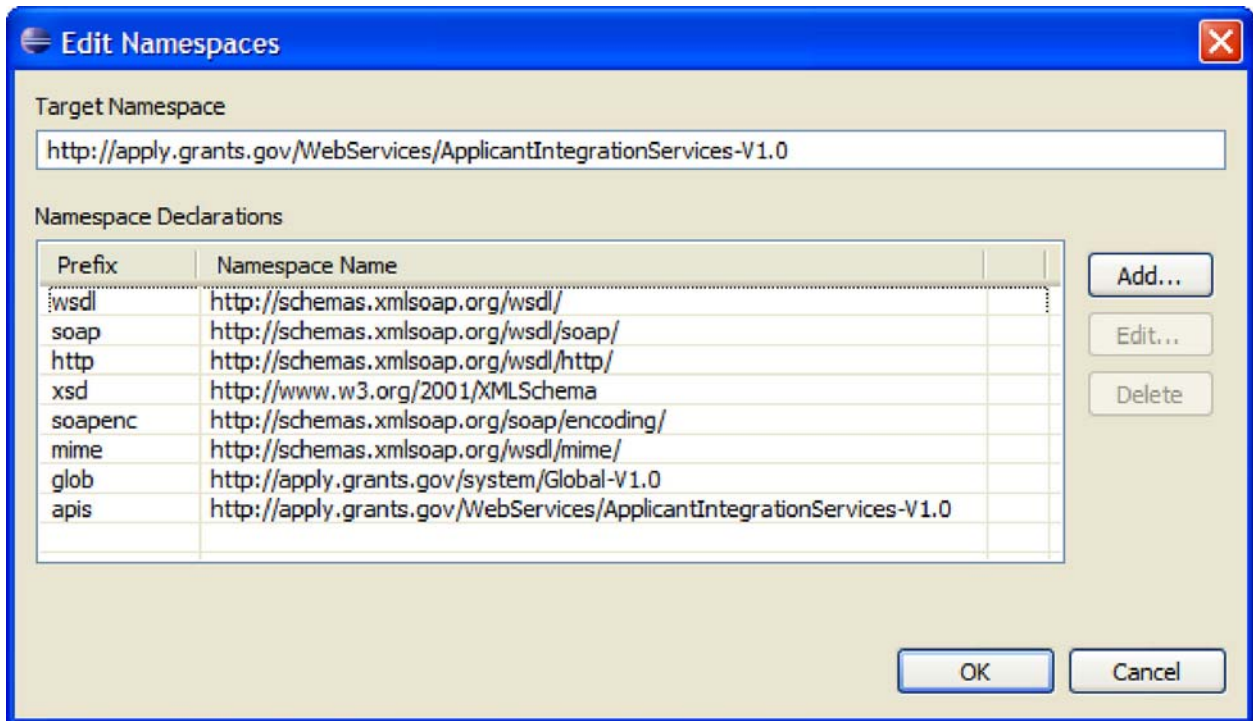


Figure 5. Schema Imports Found Within the Applicant WSDL

3.2.3 Applicant WSDL Service

The WSDL defines a Web Service as a collection of network endpoints, or ports.

Figures 6 and 7 provide details for the Applicant Web Service defined in the Applicant WSDL, including:

- Name
- Prefix
- Target Namespace
- Port Information



Figure 6. Services

port

Name:

Binding: ApplicantIntegrationSoapBinding ▼

Address:

Protocol: SOAP

Figure 7. Port

3.2.4 Applicant WSDL Port Types (Network Endpoints)

The WSDL defines a port as an association of a network address with a reusable binding. Port types, in turn, are abstract collections of supported operations.

Figure 8 provides a summary of the collection of network endpoints (and their messages) that make up the Applicant Web Service defined in the Applicant WSDL.

ApplicantIntegrationPortType		
GetOpportunityList		
➤ Input	GetOpportunityListRequest	ⓔ GetOpportunityListRequest
➤ Output	GetOpportunityListResponse	ⓔ GetOpportunityListResponse
SubmitApplication		
➤ Input	SubmitApplicationRequest	ⓔ SubmitApplicationRequest
➤ Output	SubmitApplicationResponse	ⓔ SubmitApplicationResponse
GetApplicationList		
➤ Input	GetApplicationListRequest	ⓔ GetApplicationListRequest
➤ Output	GetApplicationListResponse	ⓔ GetApplicationListResponse
GetApplicationStatusDetail		
➤ Input	GetApplicationStatusDetailRequest	ⓔ GetApplicationStatusDetailRequest
➤ Output	GetApplicationStatusDetailResponse	ⓔ GetApplicationStatusDetailResponse

Figure 8. Applicant Integration Port Type

3.2.5 Applicant Messages, Elements, and Types

The WSDL defines messages as abstract descriptions of the data being exchanged. The concrete protocol and data format specifications for a particular port type constitutes a reusable binding, where the messages and operations are then bound to a concrete network protocol and message format.

The diagrams in subsection 3.2.5.1 through 3.2.5.11 provide a summary of the messages and elements (including parameters and data types) that make up the Applicant Web Service defined in the Applicant WSDL.

3.2.5.1 GetApplicationListRequestMessage

The GetApplicationListRequest message shown in Figure 9 is used by an Applicant to request a list of submitted applications.



Figure 9. GetApplicationListRequestMessage

3.2.5.2 GetApplicationListResponseMessage

The GetApplicationListResponse message shown in Figure 10 is used by Grants.gov to return the list of submitted applications requested by an Applicant.

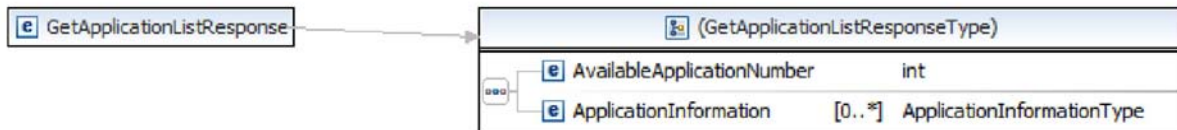


Figure 10. GetApplicationListResponseMessage

3.2.5.3 GetApplicationStatusDetailRequestMessage

An Applicant uses the GetApplicationStatusDetailRequest message (Figure 11) to request status information for a particular submitted application.



Figure 11. GetApplicationStatusDetailRequestMessage

3.2.5.4 GetApplicationStatusDetailResponseMessage

The GetApplicationStatusDetailResponse message as shown in Figure 12 is used by Grants.gov to return status detail information for a requested submitted application.

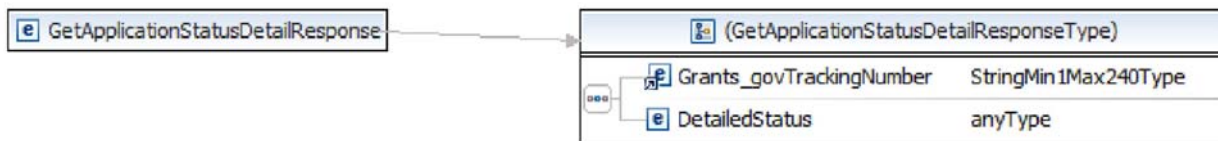


Figure 12. GetApplicationStatusDetailResponseMessage

3.2.5.5 GetOpportunityListRequestMessage

The GetOpportunityListRequest message (Figure 13) is used by an Applicant to request a list of open opportunities for which applications can be submitted. Applicants can search by OpportunityID, CFDA Number, and/or CompetitionID.

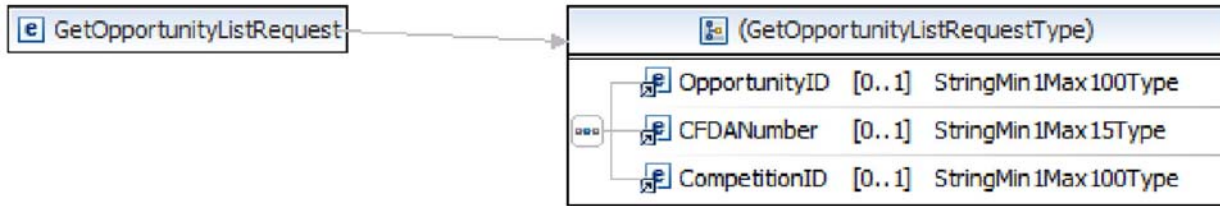


Figure 13. GetOpportunityListRequestMessage

3.2.5.6 GetOpportunityListResponseMessage

The GetOpportunityListResponse message (Figure 14) is used by Grants.gov to return a list of open opportunities for which applications can be submitted.



Figure 14. GetOpportunityListResponseMessage

3.2.5.7 SubmitApplicationRequestMessage

The SubmitApplicationRequest message (Figure 15) is used by an Applicant to submit an application for a given opportunity.

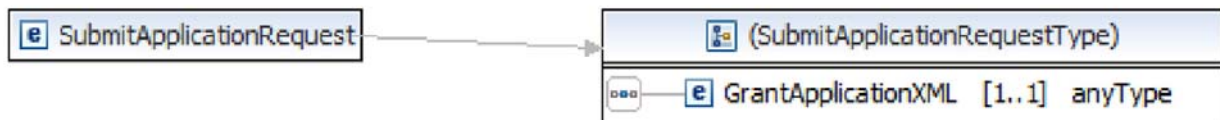


Figure 15. SubmitApplicationRequestMessage

3.2.5.8 SubmitApplicationResponseMessage

The SubmitApplicationResponse message is used by Grants.gov to indicate the result of a grant submission, as well as the Grants.gov Tracking Number for further application status inquiries and tracking as shown in Figure 16.

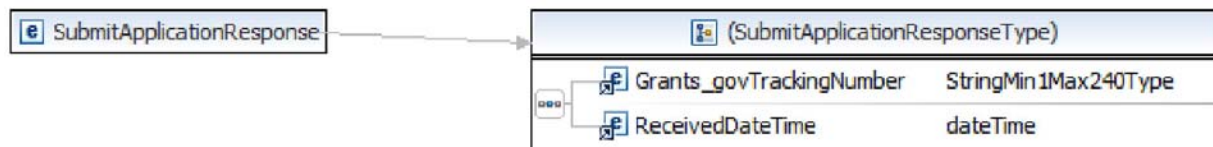


Figure 16. SubmitApplicationResponseMessage

3.2.5.9 ApplicantInformationType

The ApplicantInformation Type is used to specify the structure of the applicant information found in the application list returned following a call to the GetApplicationList operation (Figure 17).

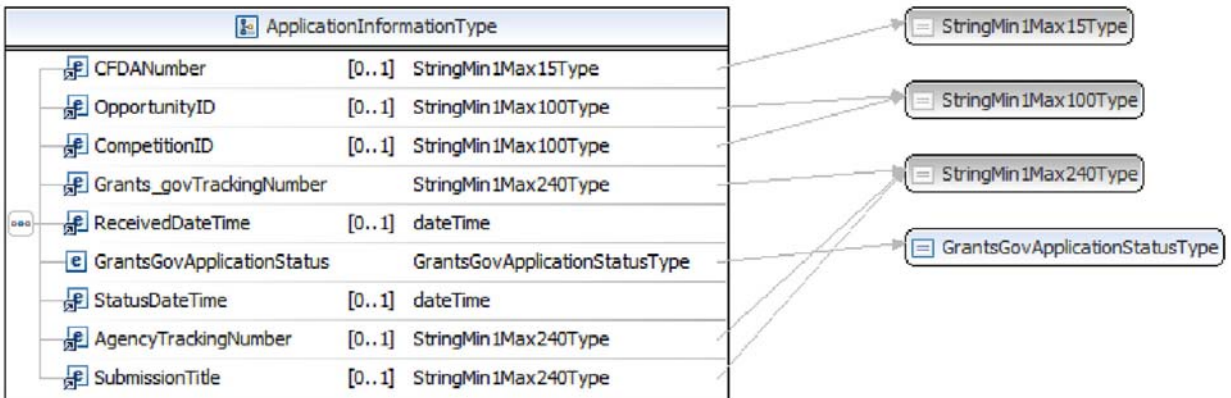


Figure 17. ApplicantInformationType

3.2.5.10 GrantsGovApplicationStatusType

The GrantsGovApplicationStatus Type is used to specify the structure of the application status detail returned when an Applicant invokes the GetApplicationStatusDetail operation. In this case, the GrantsGovApplicationStatus type is simply a string as shown in Figure 18.

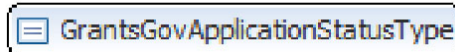
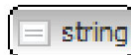


Figure 18. GrantsGovApplicationStatusType

3.2.5.11 OpportunityInformationType

The OpportunityInformation Type is used to specify the structure of the Opportunity information of the opportunity information found in the opportunity list returned following a call to the GetOpportunityList operation (Figure 19).

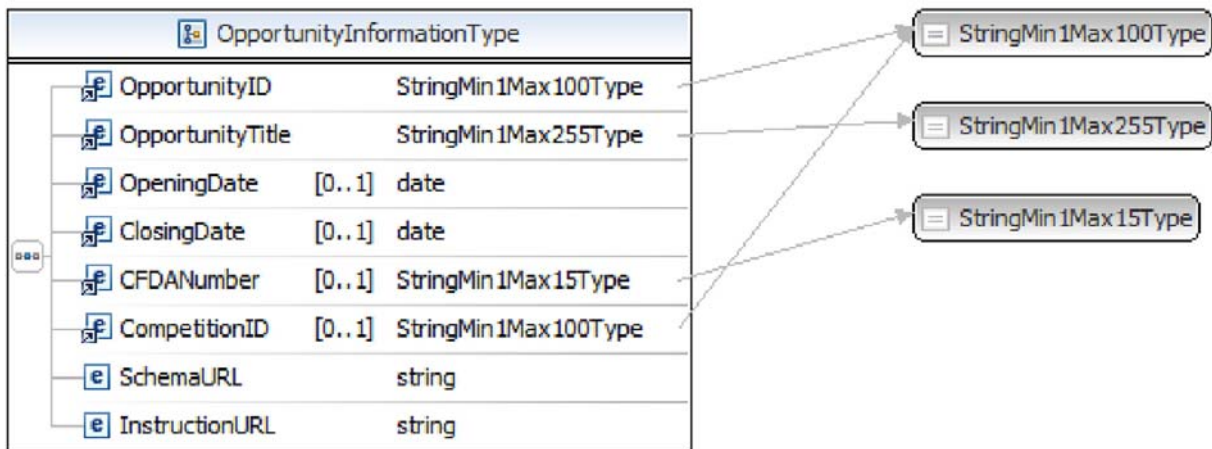


Figure 19. OpportunityInformationType

3.3 Agency System-to-System Operations

The Agency to Grants.gov System-to-System interface consists of the following five (5) operations, listed in order of typical invocation:

- GetApplicationList
- GetApplication
- GetApplicationZip
- ConfirmApplicationDelivery
- AssignAgencyTrackingNumber

3.3.1 Typical Agency-Grants.gov Interaction Sequence

Figure 20 depicts a typical request/response interaction sequence between an Agency and the Grants.gov systems:

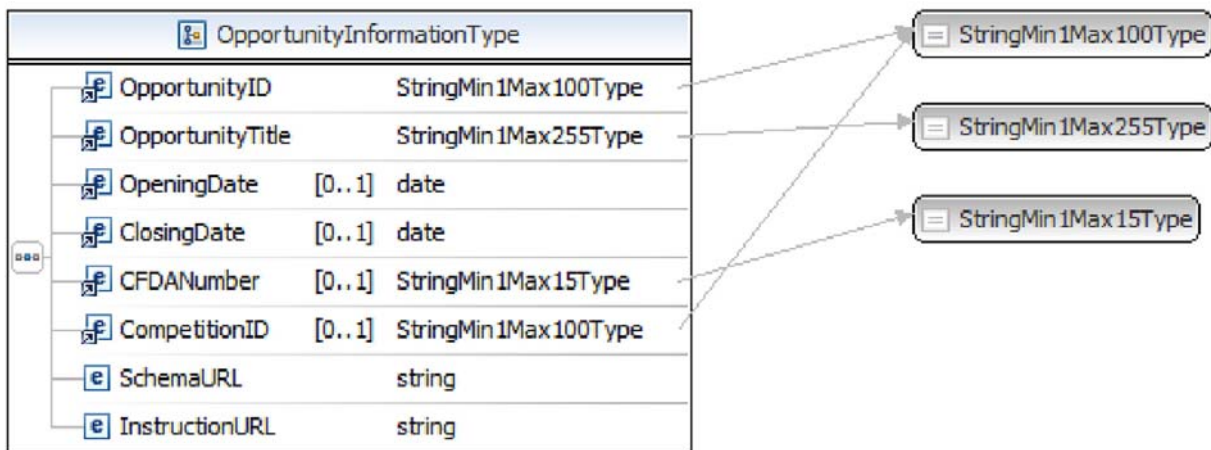


Figure 20. Typical Request/Response Interaction Between an Agency and Grants.gov System

Further details for each of the Response/Request interactions outlined above appear in following sections of the System Integration Document.

3.4 Agency System-to-System WSDL Specifications

This section provides additional Agency System-to-System specifications via the use of the Web Services Description Language (WSDL), including an overview of the schema, schema imports, service, port types, messages, elements, and types.

For more information regarding the WSDL format and structure, please refer to the Web Services Description Language (WSDL) Overview section within the appendix.

3.4.1 Agency WSDL Schema

Figure 21 is a screenshot that provides an overview of the entire Agency WSDL Schema.

Schema : <http://apply.grants.gov/WebServices/ApplicantIntegrationServices-V1.0>

Directives

- ← <http://schemas.xmlsoap.org/wsd/> (<http://schemas.xmlsoap.org/wsd/>)
- ← <http://schemas.xmlsoap.org/wsd/soap/> (<http://schemas.xmlsoap.org/wsd/soap/>)
- ← <http://apply.grants.gov/system/schemas/Global-V1.0.xsd> (<http://apply.grants.gov/system/Global-V1.0>)

Elements	Types
<ul style="list-style-type: none"> ActivityTitle : StringMin1Max120Type AgencyName : StringMin1Max60Type AgencyTrackingNumber : StringMin1Max240Type CFDANumber : StringMin1Max15Type ClosingDate : date CompetitionID : StringMin1Max100Type GetApplicationListRequest GetApplicationListResponse GetApplicationStatusDetailRequest GetApplicationStatusDetailResponse GetOpportunityListRequest GetOpportunityListResponse Grants_govTrackingNumber : StringMin1Max240Type 	<ul style="list-style-type: none"> ApplicationInformationType OpportunityInformationType GrantsGovApplicationStatusType : xsd:string
Attributes	Groups

Figure 21. Agency WSDL Schema

3.4.2 Agency WSDL Directives – Schema Imports

Figure 22 is a screenshot that provides a listing of the schema imports found within the Agency WSDL.

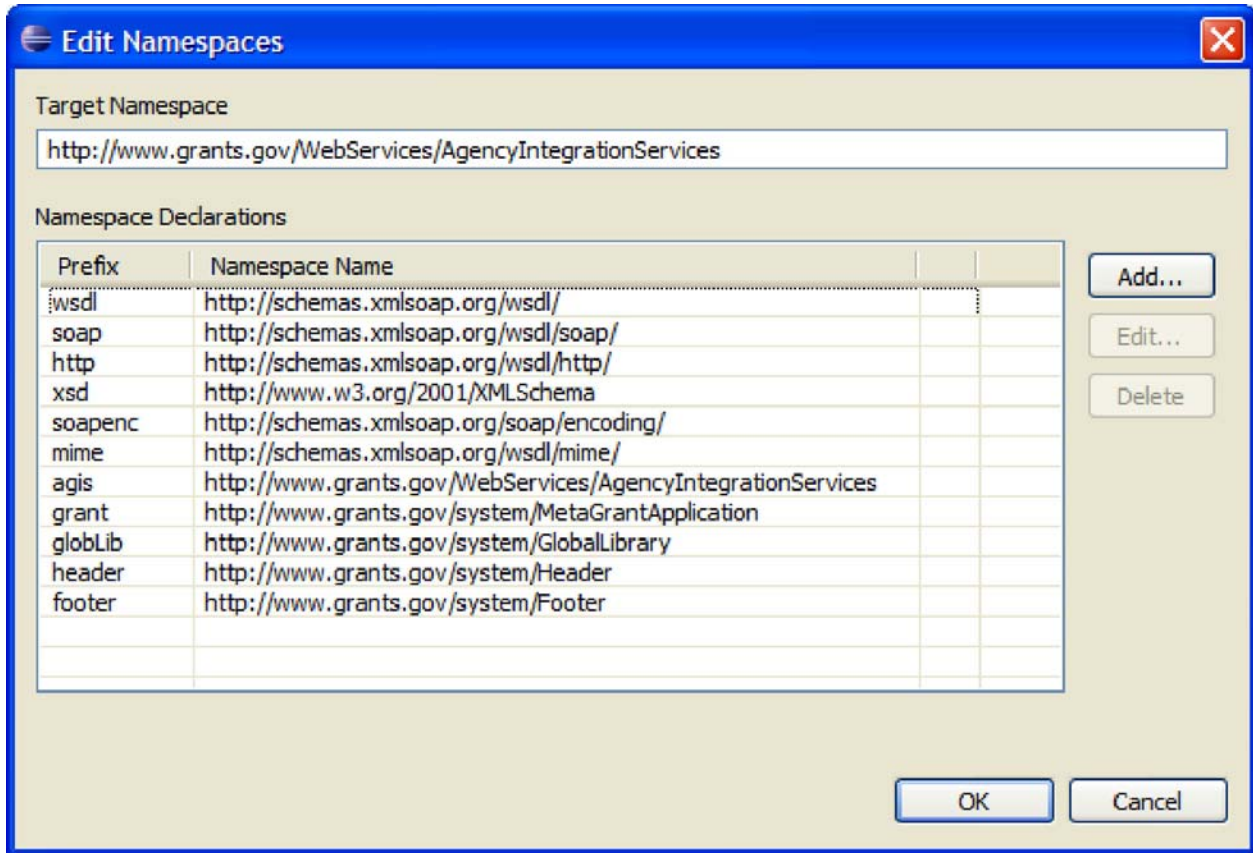


Figure 22. Schema Imports Found Within the Agency WSDL

3.4.3 Agency WSDL Service

The WSDL defines a Web Service as a collection of network endpoints, or ports. Figures 23 and 24 provide details for the Agency Web Service defined in the Agency WSDL, including:

- Name
- Prefix
- Target Namespace
- Port Information



Figure 23. AgencyIntegrationServices

port

Name: AgencyIntegrationSoapPort

Binding: AgencyIntegrationSoapBinding

Address: https://ws.grants.gov/AgencyIntegration

Protocol: SOAP

Figure 24. AgencyIntegrationSoapPort

3.4.4 Agency WSDL Port Types (Network Endpoints)

The WSDL defines a port as an association of a network address with a reusable binding. Port types, in turn, are abstract collections of supported operations.

Figure 25 provides a summary of the collection of network endpoints (and their messages) that make up the Applicant Web Service defined in the Agency WSDL.

ApplicantIntegrationPortType		
GetOpportunityList		
Input	GetOpportunityListRequest	GetOpportunityListRequest
Output	GetOpportunityListResponse	GetOpportunityListResponse
SubmitApplication		
Input	SubmitApplicationRequest	SubmitApplicationRequest
Output	SubmitApplicationResponse	SubmitApplicationResponse
GetApplicationList		
Input	GetApplicationListRequest	GetApplicationListRequest
Output	GetApplicationListResponse	GetApplicationListResponse
GetApplicationStatusDetail		
Input	GetApplicationStatusDetailRequest	GetApplicationStatusDetailRequest
Output	GetApplicationStatusDetailResponse	GetApplicationStatusDetailResponse

Figure 25. Agency Integration Port Type

3.4.5 Agency Messages, Elements, and Types

The WSDL defines messages as abstract descriptions of the data being exchanged. The concrete protocol and data format specifications for a particular port type constitutes a reusable binding, where the messages and operations are then bound to a concrete network protocol and message format.

The diagrams in sections 3.4.5.1 through 3.4.5.15 provide a summary of the messages and elements (including parameters and data types) that make up the Agency Web Service defined in the Agency WSDL.

3.4.5.1 AssignAgencyTrackingNumberMessage

The AssignAgencyTrackingNumberRequest message is used by an Agency to notify Grants.gov of the Agency’s internal tracking number for a given application.

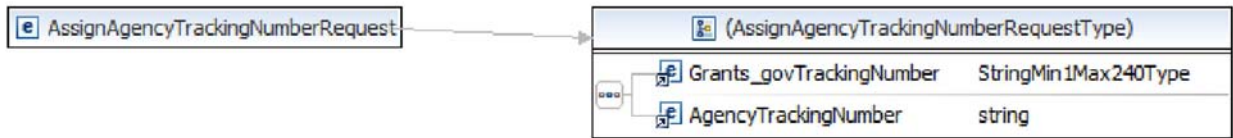


Figure 26. AssignAgencyTrackingNumberMessage

3.4.5.2 AssignAgencyTrackingNumberResponse Message

The AssignAgencyTrackingNumberResponse message is used by *Grants.gov* to acknowledge the receipt of the Agency Tracking Number for a specific application.

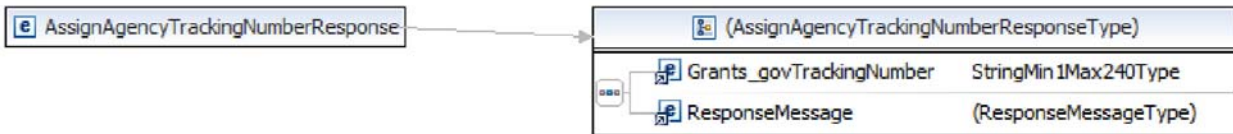


Figure 27. AssignAgencyTrackingNumberResponse Message

3.4.5.3 ConfirmApplicationDeliveryRequestMessage

The ConfirmApplicationDeliveryRequest message is used by an *Agency* to confirm the retrieval of an application.



Figure 28. ConfirmApplicationDeliveryRequestMessage

3.4.5.4 ConfirmApplicationDeliveryResponseMessage

The ConfirmApplicationDeliveryResponse message is used by an *Agency* to return status detail information for a requested submitted application.

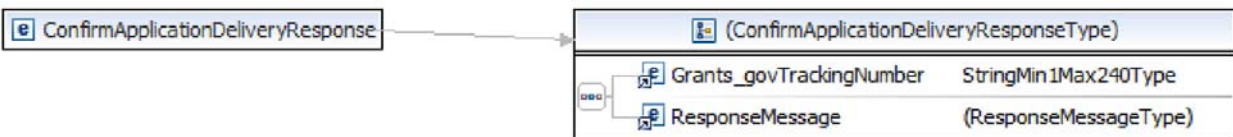


Figure 29. ConfirmApplicationDeliveryResponseMessage

3.4.5.5 GetApplicationListRequestMessage

The GetApplicationListRequest message is used by an *Applicant* to request a list of submitted applications.



Figure 30. GetApplicationListRequestMessage

3.4.5.6 GetApplicationListResponseMessage

The GetApplicationListResponse message is used by *Grants.gov* to return the list of submitted applications requested by an Applicant.

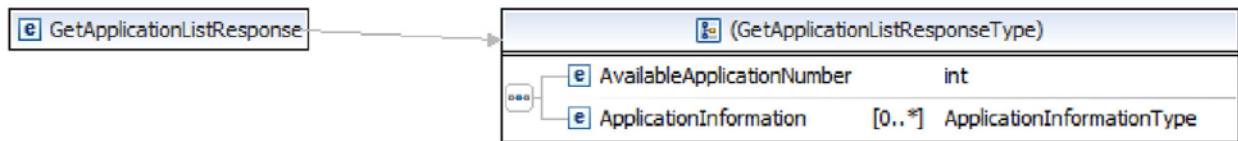


Figure 31. GetApplicationListResponseMessage

3.4.5.7 GetApplicationRequestMessage

The GetApplicationRequest message is used by an *Agency* to request an application.

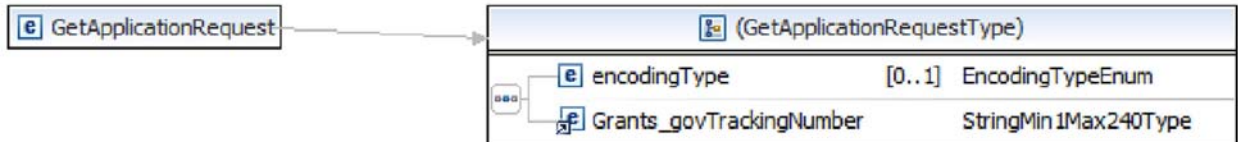


Figure 32. GetApplicationRequestMessage

3.4.5.8 GetApplicationResponseMessage

The GetApplicationResponse message is used by *Grants.gov* to provide the requested application.

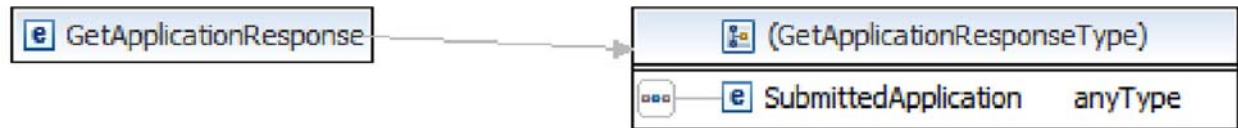


Figure 33. GetApplicationResponseMessage

3.4.5.9 GetApplicationZipRequestMessage

The GetApplicationZipRequest message is used by an *Agency* to request an application in Zip archive format.

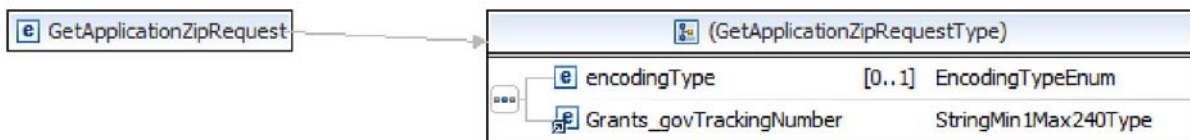


Figure 34. GetApplicationZipRequestMessage

3.4.5.10 GetApplicationZipResponseMessage

The GetApplicationZipResponse message is used by *Grants.gov* to provide the requested application in a Zip Archive format.

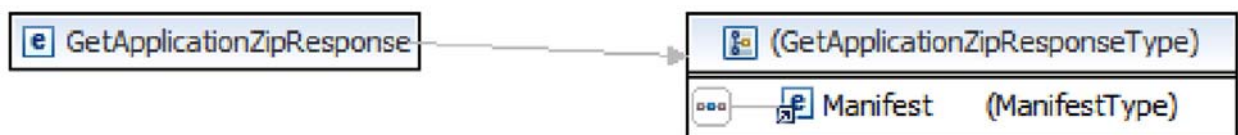


Figure 35. GetApplicationZipResponseMessage

3.4.5.11 Manifest Type

The Manifest type is used by *Grants.gov* to specify the structure of the items found within the returned Zip archive. See Item Type in Section 3.4.5.13 for more information.

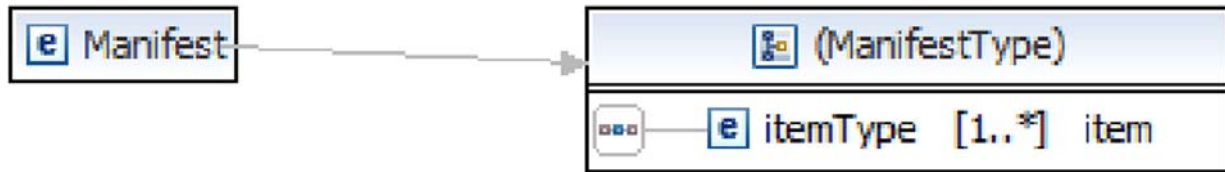


Figure 36. Manifest Type

3.4.5.12 ApplicationInformation Type

The ApplicationInformation type is used within the GetApplicationListResponseType by *Grants.gov* to specify the structure of each of the application information items returned within the list.

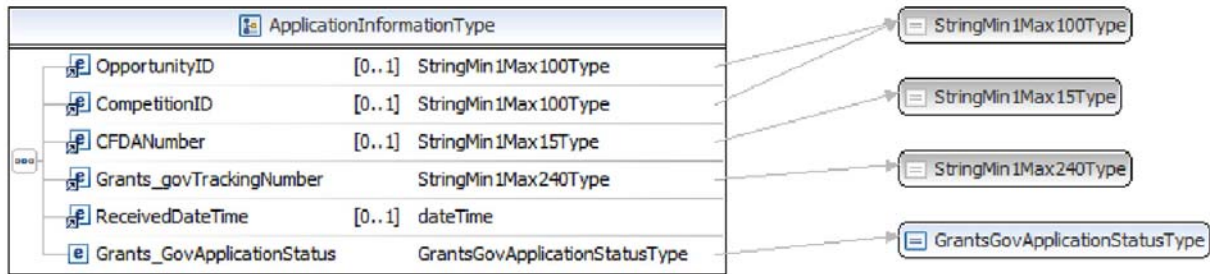


Figure 37. ApplicationInformation Type

3.4.5.13 Item Type

The Item Type is used within the GetApplicationZipResponseType message by *Grants.gov* to specify the structure of the Manifest type used within the returned Zip archive.

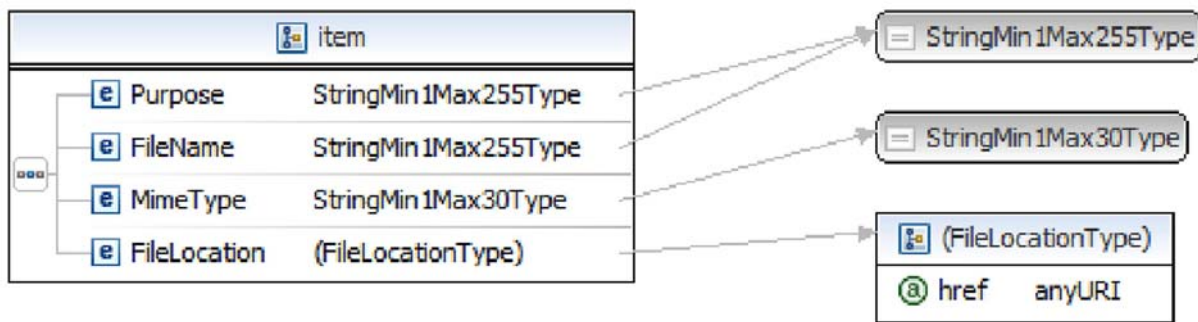


Figure 38. Item Type

3.4.5.14 EncodingEnum Type

The EncodingEnum type enumerates the valid encoding string values used by Agencies within each of the Request messages to specify the message encoding type.

Possible values currently include:

- MIME
- DIME

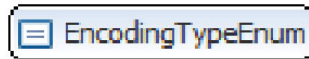
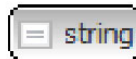


Figure 39. EncodingEnum Type

3.4.5.15 GrantsGovApplicationStatus Type

The Manifest type is used by *Grants.gov* to specify the structure of the items found within the returned Zip archive. See Item Type in Section 3.4.5.13 for more information.

The GrantsGovApplicationStatus type provides an enumeration of the valid application status values strings.

Possible values currently include:

- Receiving
- Received
- Processing
- Validated
- Rejected with Errors
- Received by Agency
- Agency Tracking Number Assigned




Figure 40. GrantsgovApplicationStatus Type

3.5 Error Handling

The errors that may be generated during a System-to-System message exchange between Grants.gov and an Agency system fall into one of two categories:

- Those that would be generated by a Grants.gov application, namely Grants.gov
- Those that would be generated by any of the framework API's used during the message exchange between systems.

In both instances, a SOAP Fault element will handle the transport of error messages.

The application-related errors occur when Grants.gov cannot fulfill a request from an Applicant or Agency system. For example, when an Agency system sends a request in the form of GetApplication

function through a SOAP exchange, it would need to provide a valid Grants.gov Tracking Number. If the tracking number is incorrect, or there are no applications associated with that tracking number, the Grants.gov would generate an error code for that incident. The error code would then be associated with a message, which will then be transported via a SOAP Fault element to the agency system.

In the case of System-to-System message exchange, an error could occur when a message cannot reach its destination. An example of that would be interruption in network, issue with the message structure, or message load, etc. In these instances, the web services framework will generate an error relevant to the incident and SOAP Fault element will transport the message to the agency system, if it can be delivered.

3.5.1 SOAP Fault Structure

As stated above, the SOAP Fault element is used to carry error and/or status information within a SOAP message. If a Fault element is present, it must appear as a child element of the Body element. A Fault element can only appear once in a SOAP message.

The SOAP Fault element has the sub elements shown in Table 1.

Table 1. SOAP Fault Sub-elements

Sub-element	Description
<faultcode>	A code for identifying the fault
<faultstring>	A human readable explanation of the fault
<faultfactor>	Information about who caused the fault to happen
<detail>	Holds application specific error information related to the Body element

The code values defined in Table 2 must be used in the faultcode element when describing faults:

Table 2. Code Values

Error	Description
VersionMismatch	Found an invalid namespace for the SOAP Envelope element.
MustUnderstand	An immediate child element of the Header element, with the mustUnderstand attribute set to "1", was not understood.
Client	The message was incorrectly formed or contained incorrect information
Server	There was a problem with the server so the message could not proceed.

3.5.2 Grants.gov Error Codes and Messages

Table 3 depicts the possible Grants.gov application error codes and associated messages that are currently in use with the System-to-System interfaces.

Table 3. Code Values

Error Code	Message	Where Used/Issued
01	Authentication: No Session	Generated whenever the HTTPSession is not available during the system authentication process. The HTTPSession is needed, as it contains the Agency/Applicant Certificate containing the serial number that represents the Agency or Applicant system.
02	Authentication: No User Certificate	Generated whenever an invalid certificate is found in the HTTPSession, during the system authentication process, or when the certificate is missing altogether. The certificate is also considered "invalid" if the serial number does not correspond to any registered user.
03	Authentication: No User ID	Generated whenever a certificate is found during the system

Error Code	Message	Where Used/Issued
	(serial #) in Certificate	authentication process, but the certificate serial number does not correspond to any registered user.
04	Authentication: Invalid Certificate	Generated whenever a certificate is found during the system authentication process, but the certificate is invalid (has expired) or has not been registered correctly.
05	Missing Agency Tracking Number	Generated during the Agency AssignAgencyTrackingNumber operation whenever the agency Tracking Number was not supplied.
06	Missing Grants.gov Tracking Number	Generated during the Agency AssignAgencyTrackingNumber or ConfirmApplicationDelivery operations whenever the Grants.gov Tracking Number was not supplied.
07	Authorization: Missing User ID	Generated during any of the 5 Agency operations whenever the user id represented by the certificate serial number does not have the authority to perform the operation.
08	Unable to Retrieve Application	Generated during the Agency GetApplication operation whenever an error was encountered while preparing the application for downloading.

3.5.3 Sample SOAP Fault Message

The following is a sample of code from Grants.gov message exchange web services using SOAP Fault with errors:

```
<soap-env:Envelope xmlns:soap-
env="http://schemas.xmlsoap.org/soap/envelope/"><soap-env:Header></soap-
env:Header><soap-env:Body><soap-
env:Fault><faultcode>Server</faultcode><faultstring>Error in processing or replying
to a message: javax.xml.bind.UnmarshalException: tag name
"att:GrantsNarrativeExplanations" is not allowed. Possible tag names are:
<Assurances>,<BudgetInformation>,<GrantSubmissionFooter>; -with
linked exception: [com.sun.msv.verifier.ValidityViolation: tag name
"att:GrantsNarrativeExplanations" is not allowed. Possible tag names are:
<Assurances>,<BudgetInformation>,<GrantSubmissionFooter>]</fau
ltstring></soap-env:Fault></soap-env:Body></soap-env:Envelope>
```

4. Web Services Security

4.1 Secure Socket Layer (SSL) and Digital Certificates

Grants.gov uses HTTP over SSL (HTTPS) to help encrypt and secure any communication between an Applicant or Agency System and the Grants.gov System. Mutual authentication is also employed in order to further enhance security. More information regarding how the mutual authentication handshake occurs and the messages are exchanged can be found at: http://en.wikipedia.org/wiki/Transport_Layer_Security#How_it_works.

4.2 Agency/Applicant System to Grants.gov Authentication

As mentioned in the Section 4.1, Grants.gov uses HTTP over SSL (HTTPS) with digital certificate to authenticate Agency and Applicant systems. Mutual authentication requires that Grants.gov install a server certificate, and that Agency and Applicant systems do the same. The certificates will be used during the mutual authentication process to establish an SSL connection between Agency systems and Grants.gov.

Detailed information regarding the setup and registration of digital certificates required for the mutual authentication to take place can be found in the following related guides, available as part of the Applicant and Agency Reference Implementation downloads:

- Applicant Web Services Security v1.0
- [Agency] Web Services Security v1.1

Appendices

Appendix A: Web Services Description Language (WSDL) Overview

A WSDL is an XML-based document that describes web services and how to access them. A WSDL document defines services as collections of network endpoints, or ports. In WSDL, the abstract definition of endpoints and messages is separated from their concrete network deployment or data format binding. A WSDL document defines a web service using the elements listed in Table 4.

Table 4. Code Values

Element	Defines
<portType>	The operations performed by the web service. The <portType> element is the most important WSDL element. It defines a web service, the operations that can be performed, and the messages that are involved. The <portType> element can be compared to a function library (or a module, or a class) in a traditional programming language.
<message>	The messages used by the web service. The <message> element defines the data elements of an operation. Each message can consist of one or more parts. The parts can be compared to the parameters of a function call in a traditional programming language.
<types>	The data types used by the web service. For maximum platform neutrality, WSDL uses XML Schema syntax to define data types.
<binding>	The communication protocols used by the web service. The <binding> element defines the message format and protocol details for each port.

The main structure of a WSDL document is as shown in Figure 41.

```

<definitions>
  <types>
    definition of types.....
  </types>

  <message>
    definition of a message....
  </message>

  <portType>
    definition of a port.....
  </portType>

  <binding>
    definition of a binding....
  </binding>

</definitions>

```

Figure 41. Main Structure of a WSDL Document

A WSDL document can also contain other elements, such as extension elements and a service element that makes it possible to group together the definitions of several web services in one single WSDL.

Appendix B: WSDL Bindings with SOAP

As WSDL describes the request-response interaction between systems, SOAP is the transport mechanism of message exchanges. Generally, a request-response WSDL using SOAP binding is constructed as shown in Figure 42.

```
<message name="getTermRequest">
  <part name="term" type="xs:string"/>
</message>

<message name="getTermResponse">
  <part name="value" type="xs:string"/>
</message>
<portType name="glossaryTerms">
  <operation name="getTerm">
    <input message="getTermRequest"/>
    <output message="getTermResponse"/>
  </operation>
</portType>
<binding type="glossaryTerms" name="b1">
  <soap:binding style="document"
transport="https://schemas.xmlsoap.org/soap/http" />
  <operation>
    <soap:operation
      soapAction="http://example.com/getTerm"/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
  </operation>
</binding>
```

Figure 42. A Request-Response WSDL Using SOAP Binding

The binding element has two attributes - the name attribute and the type attribute. The name attribute defines the name of the binding, and the type attribute points to the port for the binding, in this case the "glossary Terms" port.

The soap:binding element has two attributes - the style attribute and the transport attribute. The style attribute can be "rpc" or "document". In the above example, document attribute is used. The transport attribute defines the SOAP protocol to be used. The above example uses HTTPS.

The operation element defines each operation that the port exposes.

Figure 43 details the example that was shown earlier with WSDL terminology pointing to the various items that WSDL describes.

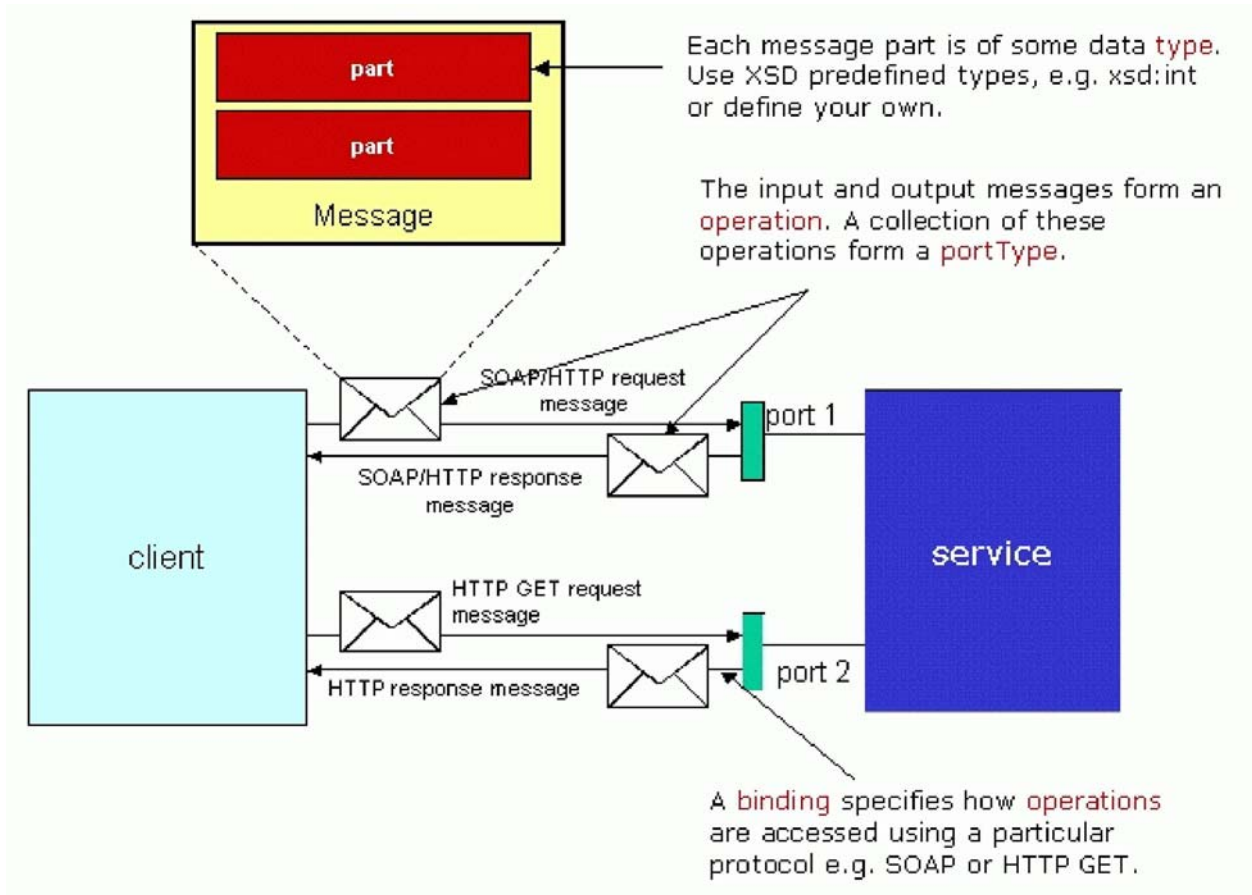


Figure 43. WSDL Binding with SOAP

Appendix C: SOAP Message Structure Overview

This section provides a brief summary of the structure of the SOAP v1.1 with Attachment 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 the Agency Integration Toolkit Document.

Messages used in Grants.gov System-to-System message exchange are formatted according to the SOAP with Attachments (SwA) specifications. Some applications may not have any attachments; however, the corresponding SOAP messages are still encapsulated in MIME via the transport protocol.

Figure 44 illustrates, at a high level, the structure of a SOAP message object, which is compliant with SOAP v1.1 with Attachments specifications. It 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.

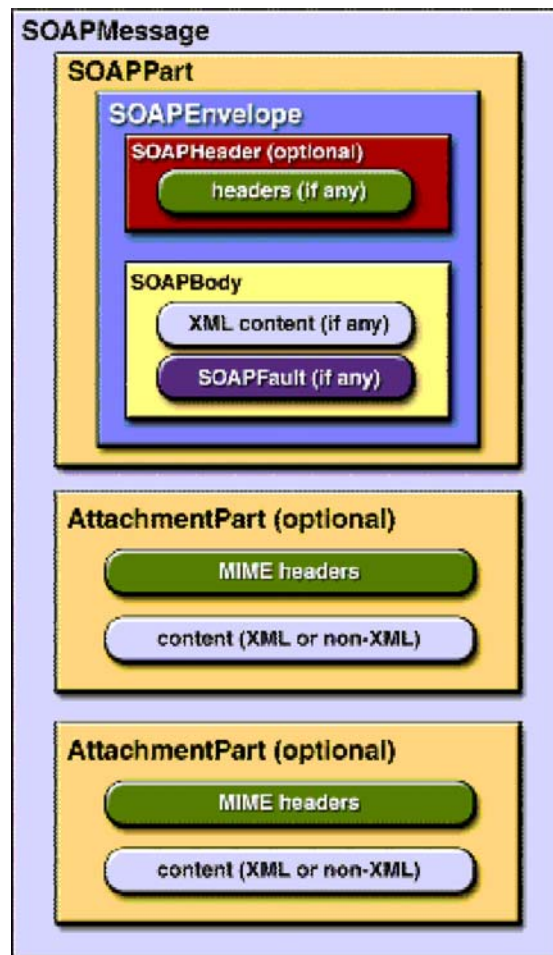


Figure 44. SOAP Message Structure

The grant applications that are received by Grants.gov may include attachments such as Budget Narrative, Project Narrative, and other user-selected attachments in their native format, as part of the application package. The subsequent XML documents that are produced by Grants.gov for delivery

to corresponding agencies in a System-to-System integration environment will have the following structure:

- SF424 or other Agency-related forms in text format
- A description of each attachment in the XML Document
- Attachments in their native binary format.

Appendix D: Sample SOAP Message with Attachment

Figure 45, which flows over the next 5+ pages, is a sample of a Grants.gov message using SOAP with Attachments representing an application containing forms SF424, and one PDF attachment. Please note that this is just a sample and does not represent the complete code and attachment data.

```

-----=_Part_19_5704740.1063741728366
Content-Type: text/xml
<?xml version="1.0" ?>
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"><SOAP-ENV:Header/><SOAP-
ENV:Body>
<grantapplication:GrantApplication
xsi:schemaLocation="http://www.grants.gov/grantapplication"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:grantapplication="http://www.grants.gov/grantapplication"
xmlns:core="http://www.grants.gov/core" xmlns:sf424="http://www.grants.gov/sf424"
xmlns:sys="http://www.grants.gov/system" xmlns:glob="http://www.grants.gov/global"
xmlns:sf424a="http://www.grants.gov/sf424a"
xmlns:sf424b="http://www.grants.gov/sf424b"
xmlns:sf424c="http://www.grants.gov/sf424c"
xmlns:sf424d="http://www.grants.gov/sf424d">
<sys:GrantSubmissionHeader xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:sys="http://www.grants.gov/system" xmlns:glob="http://www.grants.gov/global"
xsi:schemaLocation="http://www.grants.gov/header" glob:schemaVersion="1.0">
<glob:HashValue glob:hashAlgorithm="kkt1ff">LS0tLS0tLS0t</glob:HashValue>
<sys:AgencyName>HHS</sys:AgencyName>
<sys:CFDANumber>93.051</sys:CFDANumber>
<sys:ActivityTitle>Alzheimers Disease Demonstration Grants to States</sys:ActivityTitle>
<sys:OpportunityID>99</sys:OpportunityID>
<sys:OpportunityTitle>Test opp</sys:OpportunityTitle>
<sys:OpeningDate>2003-01-01</sys:OpeningDate>
<sys:ClosingDate>2003-12-31</sys:ClosingDate>
<sys:SubmissionTitle>Testing 3</sys:SubmissionTitle>
</sys:GrantSubmissionHeader>
<sf424:AwardingAgencyGrantApplication xmlns:fo="http://www.w3.org/1999/XSL/Format"
xmlns:glob="http://www.grants.gov/global" xmlns:sf424="http://www.grants.gov/sf424"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.grants.gov/sf424/ SF424.xsd"
glob:coreSchemaVersion="1.0" glob:agencySchemaVersion="1.0">
<glob:FormVersionIdentifier>04/03</glob:FormVersionIdentifier>
<sf424:SubmissionTypeCode>AN</sf424:SubmissionTypeCode>
<sf424:SubmittedDate>2003-07-01</sf424:SubmittedDate>
<sf424:ApplicationTypeCode>N</sf424:ApplicationTypeCode>
<sf424:Revision>
<sf424:RevisionCode1>A</sf424:RevisionCode1>
<sf424:RevisionCode2>A</sf424:RevisionCode2>
<sf424:RevisionOtherExplanation>Some Explanation</sf424:RevisionOtherExplanation>
</sf424:Revision>
<sf424:AgencyName>NIH</sf424:AgencyName>

```



```
<sf424:CFDANumber>93.051</sf424:CFDANumber>
<sf424:ActivityTitle>Alzheimers Disease Demonstration Grants to
States</sf424:ActivityTitle>
<sf424:SubmittingOrganization>
<sf424:OrganizationIdentifyingInformation>
<sf424:ApplicantTypeCode>Individual</sf424:ApplicantTypeCode>
<sf424:Organization>
<sf424:OrganizationName>Testing</sf424:OrganizationName>
<sf424:DUNSID>00-000-0000</sf424:DUNSID>
<sf424:EmployerID>12-1231231</sf424:EmployerID>
</sf424:Organization>
</sf424:OrganizationIdentifyingInformation>
<sf424:Address>
<sf424:Street>Test</sf424:Street>
<sf424:City>Test</sf424:City>
<sf424:County>Test</sf424:County>
<sf424:StateCode>MO</sf424:StateCode>
<sf424:ZipCode>32112</sf424:ZipCode>
</sf424:Address>
<sf424:DelinquentFederalDebtIndicator>N</sf424:DelinquentFederalDebtIndicator>
<sf424:CongressionalDistrict>23123123123</sf424:CongressionalDistrict>
</sf424:SubmittingOrganization>
<sf424:Project>
<sf424:ProjectTitle>test</sf424:ProjectTitle>
<sf424:Location>test</sf424:Location>
<sf424:ProposedStartDate>2003-10-01</sf424:ProposedStartDate>
<sf424:ProposedEndDate>2004-09-30</sf424:ProposedEndDate>
<sf424:CongressionalDistrict>312123132132</sf424:CongressionalDistrict>
</sf424:Project>
<sf424:Budget sf424:currencyCode="USD">
<sf424:FederalEstimatedAmount>5000000</sf424:FederalEstimatedAmount>
<sf424:ApplicantEstimatedAmount>100</sf424:ApplicantEstimatedAmount>
<sf424:StateEstimatedAmount>1500000</sf424:StateEstimatedAmount>
<sf424:LocalEstimatedAmount>0</sf424:LocalEstimatedAmount>
<sf424:OtherEstimatedAmount>0</sf424:OtherEstimatedAmount>
<sf424:ProgramIncomeEstimatedAmount>0</sf424:ProgramIncomeEstimatedAmount>
<sf424:TotalEstimatedAmount>6500100</sf424:TotalEstimatedAmount>
</sf424:Budget>
<sf424:Assurance>N</sf424:Assurance>
<sf424:Individual>
<sf424>Contact>
<sf424:GivenName1>test</sf424:GivenName1>
<sf424:FamilyName>test</sf424:FamilyName>
<sf424:TelephoneNumber>(703) 555-5555</sf424:TelephoneNumber>
<sf424:FaxNumber>(703) 555-5555</sf424:FaxNumber>
</sf424>Contact>
<sf424:AuthorizedRepresentative>
<sf424:GivenName1>Milton</sf424:GivenName1>
<sf424:FamilyName>Waddams</sf424:FamilyName>
<sf424:TelephoneNumber>(703) 555-5555</sf424:TelephoneNumber>
```

```
<sf424:RepresentativeTitle>Software Engineer</sf424:RepresentativeTitle>
</sf424:AuthorizedRepresentative>
</sf424:Individual>
<sf424:StateReviewCode>Not Reviewed</sf424:StateReviewCode>
</sf424:AwardingAgencyGrantApplication>
<sf424a:BudgetInformation xmlns:glob="http://www.grants.gov/global"
xmlns:sf424a="http://www.grants.gov/sf424a"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.grants.gov/sf424a SF424A.xsd"
sf424a:programType="Non-Construction" glob:coreSchemaVersion="1.0">
<glob:FormVersionIdentifier>XXXX</glob:FormVersionIdentifier>
<sf424a:BudgetSummary>
<sf424a:SummaryLineItem sf424a:activityTitle="activityTitle1">
<sf424a:CFDANumber>93.051</sf424a:CFDANumber>
<sf424a:BudgetFederalEstimatedUnobligatedAmount>5000000.00</sf424a:BudgetFedera
LEstimatedUnobligatedAmount>
<sf424a:BudgetNonFederalEstimatedUnobligatedAmount>1500100.00</sf424a:BudgetNo
nFederalEstimatedUnobligatedAmount>
<sf424a:BudgetTotalNewOrRevisedAmount>6500100.00</sf424a:BudgetTotalNewOrRevi
sedAmount>
</sf424a:SummaryLineItem>
<sf424a:SummaryLineItem sf424a:activityTitle="activityTitle2">
<sf424a:CFDANumber>93.051</sf424a:CFDANumber>
<sf424a:BudgetTotalNewOrRevisedAmount>0.00</sf424a:BudgetTotalNewOrRevisedAm
ount>
</sf424a:SummaryLineItem>
<sf424a:SummaryLineItem sf424a:activityTitle="activityTitle3">
<sf424a:CFDANumber>93.051</sf424a:CFDANumber>
<sf424a:BudgetTotalNewOrRevisedAmount>0.00</sf424a:BudgetTotalNewOrRevisedAm
ount>
</sf424a:SummaryLineItem>
<sf424a:SummaryLineItem sf424a:activityTitle="activityTitle4">
<sf424a:CFDANumber>93.051</sf424a:CFDANumber>
<sf424a:BudgetTotalNewOrRevisedAmount>0.00</sf424a:BudgetTotalNewOrRevisedAm
ount>
</sf424a:SummaryLineItem>
<sf424a:SummaryTotals>
<sf424a:BudgetFederalEstimatedUnobligatedAmount>5000000.00</sf424a:BudgetFedera
LEstimatedUnobligatedAmount>
<sf424a:BudgetNonFederalEstimatedUnobligatedAmount>1500100.00</sf424a:BudgetNo
nFederalEstimatedUnobligatedAmount>
<sf424a:BudgetFederalNewOrRevisedAmount>0.00</sf424a:BudgetFederalNewOrRevis
edAmount>
<sf424a:BudgetNonFederalNewOrRevisedAmount>0.00</sf424a:BudgetNonFederalNew
OrRevisedAmount>
<sf424a:BudgetTotalNewOrRevisedAmount>6500100.00</sf424a:BudgetTotalNewOrRevi
sedAmount>
</sf424a:SummaryTotals>
</sf424a:BudgetSummary>
<sf424a:BudgetCategories>
```

<sf424a:CategorySet sf424a:activityTitle="activityTitle5">
<sf424a:BudgetOtherRequestedAmount>6500100.00</sf424a:BudgetOtherRequestedAmount>
<sf424a:BudgetTotalDirectChargesAmount>6500100.00</sf424a:BudgetTotalDirectChargesAmount>
<sf424a:BudgetTotalAmount>6500100.00</sf424a:BudgetTotalAmount>
</sf424a:CategorySet>
<sf424a:CategorySet sf424a:activityTitle="activityTitle6">
<sf424a:BudgetTotalDirectChargesAmount>0.00</sf424a:BudgetTotalDirectChargesAmount>
<sf424a:BudgetTotalAmount>0.00</sf424a:BudgetTotalAmount>
</sf424a:CategorySet>
<sf424a:CategorySet sf424a:activityTitle="activityTitle7">
<sf424a:BudgetTotalDirectChargesAmount>0.00</sf424a:BudgetTotalDirectChargesAmount>
<sf424a:BudgetTotalAmount>0.00</sf424a:BudgetTotalAmount>
</sf424a:CategorySet>
<sf424a:CategorySet sf424a:activityTitle="activityTitle8">
<sf424a:BudgetTotalDirectChargesAmount>0.00</sf424a:BudgetTotalDirectChargesAmount>
<sf424a:BudgetTotalAmount>0.00</sf424a:BudgetTotalAmount>
</sf424a:CategorySet>
<sf424a:CategoryTotals>
<sf424a:BudgetPersonnelRequestedAmount>0.00</sf424a:BudgetPersonnelRequestedAmount>
<sf424a:BudgetFringeBenefitsRequestedAmount>0.00</sf424a:BudgetFringeBenefitsRequestedAmount>
<sf424a:BudgetTravelRequestedAmount>0.00</sf424a:BudgetTravelRequestedAmount>
<sf424a:BudgetEquipmentRequestedAmount>0.00</sf424a:BudgetEquipmentRequestedAmount>
<sf424a:BudgetSuppliesRequestedAmount>0.00</sf424a:BudgetSuppliesRequestedAmount>
<sf424a:BudgetContractualRequestedAmount>0.00</sf424a:BudgetContractualRequestedAmount>
<sf424a:BudgetConstructionRequestedAmount>0.00</sf424a:BudgetConstructionRequestedAmount>
<sf424a:BudgetOtherRequestedAmount>6500100.00</sf424a:BudgetOtherRequestedAmount>
<sf424a:BudgetTotalDirectChargesAmount>6500100.00</sf424a:BudgetTotalDirectChargesAmount>
<sf424a:BudgetIndirectChargesAmount>0.00</sf424a:BudgetIndirectChargesAmount>
<sf424a:BudgetTotalAmount>6500100.00</sf424a:BudgetTotalAmount>
<sf424a:ProgramIncomeAmount>0.00</sf424a:ProgramIncomeAmount>
</sf424a:CategoryTotals>
</sf424a:BudgetCategories>
<sf424a:NonFederalResources>
<sf424a:ResourceLineItem sf424a:activityTitle="activityTitle9">
</sf424a:ResourceLineItem>
<sf424a:ResourceLineItem sf424a:activityTitle="activityTitle10">
</sf424a:ResourceLineItem>

```
<sf424a:ResourceLineItem sf424a:activityTitle="activityTitle11">
</sf424a:ResourceLineItem>
<sf424a:ResourceLineItem sf424a:activityTitle="activityTitle12">
</sf424a:ResourceLineItem>
<sf424a:ResourceTotals>
</sf424a:ResourceTotals>
</sf424a:NonFederalResources>
<sf424a:BudgetForecastedCashNeeds>
<sf424a:BudgetFirstYearAmounts>
</sf424a:BudgetFirstYearAmounts>
<sf424a:BudgetFirstQuarterAmounts>
</sf424a:BudgetFirstQuarterAmounts>
<sf424a:BudgetSecondQuarterAmounts>
</sf424a:BudgetSecondQuarterAmounts>
<sf424a:BudgetThirdQuarterAmounts>
</sf424a:BudgetThirdQuarterAmounts>
<sf424a:BudgetFourthQuarterAmounts>
</sf424a:BudgetFourthQuarterAmounts>
</sf424a:BudgetForecastedCashNeeds>
<sf424a:FederalFundsNeeded>
<sf424a:FundsLineItem sf424a:activityTitle="activityTitle13">
</sf424a:FundsLineItem>
<sf424a:FundsLineItem sf424a:activityTitle="activityTitle14">
</sf424a:FundsLineItem>
<sf424a:FundsLineItem sf424a:activityTitle="activityTitle15">
</sf424a:FundsLineItem>
<sf424a:FundsLineItem sf424a:activityTitle="activityTitle16">
</sf424a:FundsLineItem>
<sf424a:FundsTotals>
</sf424a:FundsTotals>
</sf424a:FederalFundsNeeded>
<sf424a:OtherInformation>
</sf424a:OtherInformation>
</sf424a:BudgetInformation>
<sys:GrantSubmissionFooter xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:sys="http://www.grants.gov/system" xmlns:glob="http://www.grants.gov/global"
xsi:schemaLocation="http://www.grants.gov/footer" glob:schemaVersion="1.0">
<glob:HashValue glob:hashAlgorithm="kkt1ftf">LS0tLS0tLS0t</glob:HashValue>
<sys:ReceivedDateTime>2003-06-26T17:02:30</sys:ReceivedDateTime>
<sys:SubmitterName>Northrop Grumman</sys:SubmitterName>
<sys:Grants_govTrackingNumber>GRANT00000053</sys:Grants_govTrackingNumber>
</sys:GrantSubmissionFooter>
</grantapplication:GrantApplication>

</SOAP-ENV:Body></SOAP-ENV:Envelope>
-----=_Part_19_5704740.1063741728366
Content-Type: application/pdf
Content-Id: Data_Binding_With_JAXB.pdf
Content-Location: Data_Binding_With_JAXB.pdf
```

```
%PDF-1.3 %a«-4 0 obj << /Type /Info /Producer (null) >> endobj5 0 obj << /Length 700
/Filter /FlateDecode >> stream
xœiXBo>0 ~ç- ðÛZis±ÁüØÛ°-S«UêT!Uªúà€° ÎCEÓ,ÿÿœ@æ#0eš&EjQ^ü 1;ÿ¿Ï ßyâœ]x
^ú(TM:..š9 1æG ] 0Š,8Â4^PRçû“ \óSô†† vÑÉ$ 2¼šÛ «\i-°zww~Š-Pre\ f[ çÍ“ {„i
~ú-ù %jQi‘ k;ÍÁ“(äB”oR=Ö- íµ\*qjRañT,—3%,¶P/µT9/j è~ª”x8@ Ç0ç,&Ï|RâT-gY€Z -
ç‡Éä'çpÁ|ÿj Gl“Á^ç 1/ ,@Ä °u¾8çl“§ zo¾ZLB¿qžđl V'švü^šäU ,
<< /Type /Page /Parent 1 0 R /MediaBox [ 0 0 612 792 ] /Resources 3 0 R /Contents 5 0 R
/Annots 7 0 R >> endobj7 0 obj
%%EOF
```

Figure 45.

Appendix E: Acronyms & Definitions

Acronym	Definition
AP	Application Program(ming) Interface
CA	Certification Authority
CRL	Certificate Revocation List
GUI	Graphical User Interface
HTTP	Hyper Text Transfer Protocol
HTTPS	Hyper Text Transfer Protocol Secure Sockets
IT	Information Technology
JAXM	Java API for XML Messaging
MIME	Multipurpose Internet Mail Extensions
PAPIDF	Portable Document Format
POC	Point of Contact
RPC	Remote procedure call (RPC) is a technology that allows a computer program to cause a subroutine or procedure to execute in another address space (commonly on another computer) without the programmer explicitly coding the details for this remote interaction. That is, the programmer would write essentially the same code whether the subroutine is local to the executing program, or remote. When the software in question is written using object-oriented principles, RPC may be referred to as remote invocation or remote method invocation.
SOAP	Simple Object Access Protocol
SSL	Secure Socket Layer
SwA	SOAP with Attachment
URL	Uniform Resource Locator
WSDL	Web Services Description Language
XML	eXtensible Markup Language