

Federal Assistance Award Data Collection Prototype

Web Services API Interface Specifications Document

Version 3.0 – Final

Last updated: May 1st , 2018

Prepared by:



2300 Dulles Station Blvd
Herndon, VA 20171
United States

DOCUMENT HISTORY

| Version Number | Date | Responsible | Summary of Change |
|----------------|------------|-------------|--|
| 1.0 | 11/21/2017 | IBM | Draft of Baseline version ready for Product Owner review |
| 2.0 | 11/30/2017 | IBM | No changes requested. Updated title page and file name to reflect baseline delivery version. |
| 3.0 | 05/01/2018 | IBM | Added Delete API |
| | | | |
| | | | |
| | | | |
| | | | |

Table of Contents

| <u>SECTION</u> | <u>PAGE</u> |
|---|-------------|
| 1 INTRODUCTION..... | 1 |
| 2 SERVICE ORIENTED ARCHITECTURE | 1 |
| 3 FEDERAL ASSISTANCE SERVICE ARCHITECTURE..... | 1 |
| 4 ARCHITECTURAL GOALS AND CONSTRAINTS | 2 |
| 5 WSDL NOMENCLATURE | 2 |
| 6 API STANDARDS AND GENERIC DETAILS..... | 3 |
| 6.1 STANDARD METHOD SIGNATURES..... | 3 |
| 6.1.1 <i>Service Input Parameters</i> | 3 |
| Table 6-1. Service Input Parameters | 3 |
| Table 6-2. Data Operation Descriptions..... | 3 |
| 6.1.2 <i>Service Output Parameters</i> | 4 |
| Table 6-3. Service Output Parameters..... | 4 |
| Figure 6–1. Data Collection Business Services..... | 5 |
| 6.1.2.1 Output Parameter Types..... | 5 |
| Table 6-4. Output Parameter Types | 5 |
| 6.1.2.2 Sample Response Error Messages..... | 6 |
| 7 SERVICE API DETAILS..... | 8 |
| 7.1 BUSINESS SERVICES | 8 |
| 7.1.1 <i>Assistance Data Collection</i> | 8 |
| 7.1.1.1 Service meta-specifications in WSDL..... | 8 |
| 7.1.1.2 Assistance Service methods | 8 |
| 7.1.1.2.1 get..... | 9 |
| 7.1.1.2.2 getList..... | 10 |
| 7.1.1.2.3 create | 11 |
| 7.1.1.2.4 update | 11 |
| 7.1.1.2.5 isComplete..... | 12 |
| 7.1.1.2.6 approve | 14 |
| 7.1.1.2.7 exists..... | 15 |
| 7.1.1.2.8 correct..... | 16 |
| 7.1.1.2.9 isExistingAssistanceComplete..... | 16 |
| 7.2 GUI SERVICES | 17 |
| 7.2.1 <i>Assistance Data Collection</i> | 17 |
| 7.2.1.1 Service meta-specifications in WSDL..... | 17 |
| 7.2.1.2 Assistance Service methods | 18 |
| 7.2.1.2.1 getExistingAssistanceURL..... | 18 |
| 7.2.1.2.2 getNewAssistanceURL..... | 19 |
| APPENDIX A DEFINITION AND ACRONYMS..... | 20 |
| ACRONYMS..... | 20 |
| DEFINITIONS..... | 20 |
| APPENDIX B REFERENCES | 21 |
| NORMATIVE REFERENCES | 21 |
| INFORMATIVE REFERENCES..... | 21 |

1 Introduction

Under GSA's initiative and direction, the Federal Award Assistance Data Collection (FAADC) Prototype is being developed as a real-time federal enterprise information system to collect the Assistance data across the federal government. The advent of platform, language, vendor, and tool independent standards has enabled data processing and transport to be carried out seamlessly between heterogeneous systems.

Web services based on SOAP and XML, implemented using Java technologies, are used in Federal Assistance module to provide interoperability with various federal assistance systems.

2 Service Oriented Architecture

The Federal Assistance system architecture is based on a Service-Oriented Architecture (SOA) platform. The choice of a SOA is based on the requirement of GSA to produce a web service based application that will allow integration of the Assistance module with agency systems. All identifiable system functions are published as services that external systems invoke using open standards over a network. This architecture exposes all system functions including business logic and GUI screens making them all accessible to agency systems.

The value of a SOA-based approach is realized in the reusability of the components. Reusability offers the government tremendous savings of time and money as software development is leveraged by many systems without the need for additional development or redundant efforts.

SOA is the architectural structure underpinning web services and is developed to the J2EE standard. The technologies used to invoke web services promote interoperability. These technologies include:

- **XML**, which defines a universal way of representing data
- **SOAP**, which provides the transport mechanism for web services
- **WSDL**, which describes a web service definition

Table 2-1. Software Working Group (SAWG)

| Feature Rated | J2EE/Web Services | .NET/Web Services |
|--|-------------------|-------------------|
| Cross Platform Portability/OS Independent | ✓✓✓ | ✓ |
| Mature (not antiquated) Technology | ✓✓ | ✓ |
| Loose Integration of Heterogeneous Systems | ✓✓✓ | ✓✓✓ |
| Infrastructure Independence | ✓✓✓ | ✓✓✓ |
| Standards-Based | ✓✓✓ | ✓✓ |
| Non-proprietary Extensibility | ✓✓✓ | ✓ |
| Ease of Development / Integration | ✓✓ | ✓✓✓ |
| Application Interoperability | ✓✓✓ | ✓✓✓ |
| Final Analysis | 22 / 24 | 17 / 24 |

A standard catalog of Web services has been created across the business entities. System functions are categorized as described in the following sections.

3 Federal Assistance Service Architecture

This document introduces the web services system architecture that exposes one point of entry to Federal Assistance module. The web services APIs will act as the gateway to access all functionality on the server side. The following set of modules that belong to Federal Assistance use the web services APIs to achieve their functionality.

1. GUI services that allow creation or modification of Assistance data.
2. Business services that allow the integrators to launch the data collection screens from within the COTS/GOTS products

4 Architectural Goals and Constraints

Each Web Service API will follow the set of guidelines described below that are essential to any published set of APIs accessible from anywhere via the Internet.

1. **Simplicity:** An API addresses a simple business process and is atomic.
2. **Interoperability:** An API is platform independent. Web services have been the solution of choice throughout the industry to address heterogeneous distributed systems. They are designed to be platform independent in order to achieve maximum interoperability.
3. **Nomenclature Consistency:** An API follows a specific set of naming conventions that are used consistently.
4. **Functional Consistency:** An API behaves the same at all times for the same set of data inputs unless there are processing business logic and rules that are driven by factors like time, data history, etc.
5. **Macro Level API:** An API translates a business use case into one service that completes the business process in one transaction.
6. **Appropriate Payload Size:** List Retrieval API services limit the number of values returned, so that the payload is not exceeded beyond the limit the middle-tier can handle.
7. **Stateless:** An API service is stateless.
8. **Secure:** All API input contains the user and source data used for authentication.
9. **Error Processing:** The API returns a comprehensive and complete set of error codes and corresponding messages.
10. **Error Batching:** An API service encapsulates all errors during the service execution into a single response. This allows the service customer to send back the corrected request without running an iterative error correction process for each attribute or entity of the request.

5 WSDL Nomenclature

The following abstract from the *W3C March 2001 note 15* describes the WSDL:

WSDL is an XML format for describing network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information. The operations and messages are described abstractly, and then bound to a concrete network protocol and message format to define an endpoint. Related concrete endpoints are combined into abstract endpoints (services). WSDL is extensible to allow description of endpoints and their messages regardless of what message formats or network protocols are used to communicate.

Table 5-1 shows the Web nomenclature.

Table 5-1. Web Service Nomenclature

| WSDL Parameter | Value |
|--------------------|---------------------------|
| PortType | <DomainClassName>PortType |
| Binding | <DomainClassName>Binding |
| Soap-binding style | rpc |
| TargetNamespace | <DomainClassName>.wsdl |

All complex types specified by the Web Services include the targetNamespace in the corresponding WSDL and are named after the complex type or the domain level object, i.e., Assistance.xsd

The schemas are located and loadable from a public URL using the http protocol. Availability of the web services over other protocols such as ftp and SMTP is not supported due to security risks.

6 API Standards and Generic details

The web services APIs domain objects encompass the following standards:

- Service calls are authenticated by checking for valid User ID/Password.
- Service calls are checked for authorization before serving the request. For example, the create service will check if the user has the privilege to create an Assistance record.
- Web Services APIs include common business services such as create, get, update and delete.
- Service calls use a standard method signature. All the business classes in the system have the same method signature. Standardization involves the same set of input and output parameters and their generic structures for the web services.
- Service calls contain a logging and error mechanism. All the requests are logged in the underlying generic layer of the business classes.

6.1 Standard Method Signatures

All the methods use the following signatures:

- All the input and output parameters are in XML Format.
- The inputs to the service methods and the subsequent domain classes are encapsulated in the authentication key and the input parameters.

6.1.1 Service Input Parameters

Table 6-1 describes the service input parameter names and description.

Table 6-1. Service Input Parameters

| Parameter Name | Contents |
|-------------------|--|
| AuthenticationKey | UserID, Password, Source |
| InputXML | Contains an XML representation of the business object, search criteria, or key for the business object |

Table 6-2 describes the contents of the input XML based on the type of data operation.

Table 6-2. Data Operation Descriptions

| Operation | Input XML Data | Description |
|-----------|---|---|
| create | XML representation of the business object | XML is converted to the value object using JAXB and JDBC calls to perform DB operations |
| update | XML representation of the business object | Update errors, business validation errors and warning messages |
| correct | XML representation of the business object | Business validation errors and warning messages |
| get | XML representation of the key(s) | Retrieved based on the key values specified in the input |

| | | |
|---------|--|--|
| exists | XML representation of the key(s) | Retrieved based on the key values specified in the input |
| delete | XML representation of the key(s) | Deleted based on the key values specified in the input |
| getList | search criteria inputs serialized in predefined XML format | Query based on search criteria input |

6.1.2 Service Output Parameters

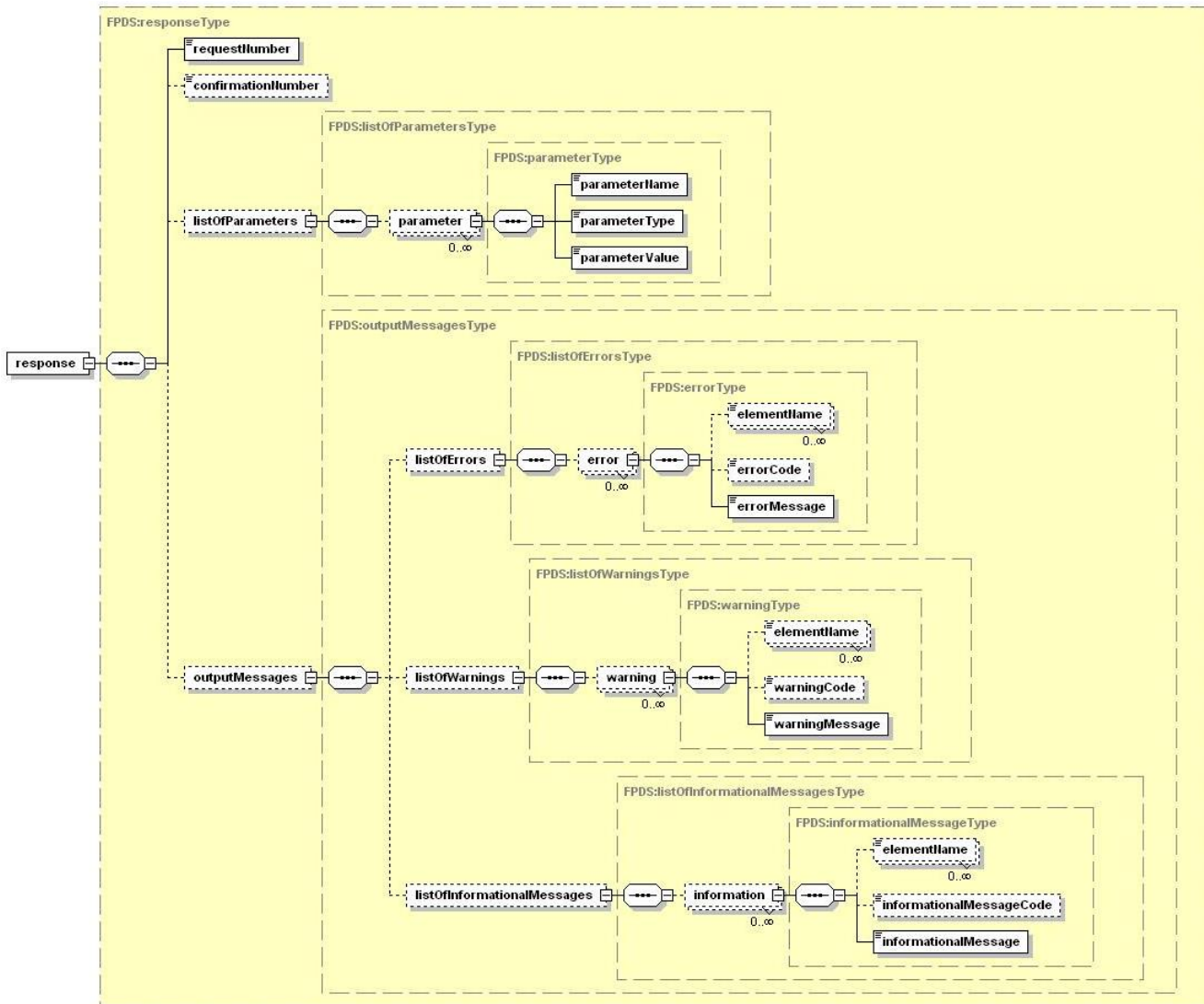
The service output parameters are represented in Meta XML strings. Table 6-3 describes the XML meta-response.

Table 6-3. Service Output Parameters

| Response XML Elements | Description |
|-----------------------|---|
| Request Number | Number returned to uniquely identify the request from the log store |
| Confirmation Number | Confirmation number for DML Transactions |
| Messages | List of error, warning and informational messages. |
| listOfParameters | List of return parameters |

The schematic representation of the output is shown in Figure 6–1.

Figure 6–1. Data Collection Business Services



6.1.2.1 Output Parameter Types

All unauthorized transaction operations are returned with Authentication Errors. Error messages are returned when the operation fails. Only a sample list of errors is provided for each operation. The output parameter types are described in Table 6-4.

Table 6-4. Output Parameter Types

| Operations | Confirmation Number | | Specific API Output Parameters | Example Errors and warnings |
|------------|---------------------|---------|--|---|
| | success | failure | | |
| get | Positive Integer | -1 | XML representation of the Business object | Not Found , mode errors |
| getList | Positive Integer | -1 | XML representation of the queried business objects in a list | Not Found , insufficient query information errors, query too broad warnings |

| | | | | |
|------------------------------|------------------|----|--|--|
| create | Positive Integer | -1 | Data created in the table | Data completion errors and warning messages |
| update | Positive Integer | -1 | Updated version of the business object | Update errors, business validation errors and warning messages |
| delete | Positive Integer | -1 | Boolean true / false response | Not Found, insufficient query information errors, business restriction errors. |
| isComplete | Positive Integer | -1 | Boolean true / false response | Business validation errors and warning messages |
| approve | Positive Integer | -1 | Boolean true / false response | Business validation errors and warning messages |
| exists | Positive Integer | -1 | Boolean true / false response | Not Found , insufficient query information errors |
| correct | Positive Integer | -1 | Corrected record | Business validation errors and warning messages |
| isExistingAssistanceComplete | Positive Integer | -1 | Boolean true / false response | Business validation errors and warning messages |

6.1.2.2 Sample Response Error Messages

```

<response>
<listOfErrors>
  <error>
    <elementName>searchCriteriaXML</elementName>
    <errorCode>10141</errorCode>
    <errorMessage>Service Unavailable Please try again later</errorMessage>
  </error>
  <error>
    <elementName>FAIN</elementName>
    <errorCode>10200</errorCode>
    <errorMessage>Cannot create assistance record. FAIN already exists in the
database</errorMessage>
  </error>
</listOfErrors>
<listOfWarnings>
  <warning>
    <elementName>countyCode</elementName>
    <warningCode>30501</warningCode>
    <warningMessage> countyCode is not required for when record type is 'Aggregate'. Ignoring the
value </warningMessage>
  </warning>
</listOfWarnings>
<listOfInformationalMessages>
  <informationalMessage>
    <elementName>None</elementName>
    <InformationalCode>73210</InformationalCode>
    <InformationalMessage>Congressional district code has been derived based on the zip code
information provided</InformationalMessage>
  </informationalMessage>
</listOfInformationalMessages>

```

</response>

7 Service API details

Below sections list the available integration services for the Federal Assistance module in FPDS-NG to collect assistance data.

7.1 Business Services

7.1.1 Assistance Data Collection

The Business Services WSDL specification lists all the available services, schema definitions, transport endpoints and name spaces to integrate and collect assistance data.

7.1.1.1 Service meta-specifications in WSDL

Below table gives the parameter names used in the Assistance WSDL file.

| WSDL Parameter | Value |
|--------------------|--------------------|
| PortType | AssistancePortType |
| Binding | AssistanceBinding |
| Soap-binding style | Rpc |
| TargetNameSpace | Assistance.wsdl |

7.1.1.2 Assistance Service methods

Below table gives all the services provided in the Federal Assistance business services module.

| Service Type/Name | Service Description |
|-------------------|--|
| get | The Assistance.get service finds an existing assistance record using the FAIN or URI information and then composes and returns the XML representation of the record |
| getList | The Assistance.getList service finds assistance records, which match the input selection criteria. Matching records are returned in the XML format declared in the Schema. |
| create | The Assistance.create service creates a new assistance record. This service utilizes the validate service to check record validity before inserting into the database. |
| update | The Assistance.update service updates an existing assistance record. This service utilizes the validate service to check record validity before updating the database. |
| delete | The Assistance.delete service deletes an existing assistance in the system. This service utilizes the exists service to check record existence before marking the record as deleted from the database. |
| isComplete | The Assistance.isComplete service does data validation and business rule validation. |
| approve | Assistance.approve approves the document after checking for validity and completeness of the document by using the isComplete service. If no error is returned, the status of the given record is changed from DRAFT mode to FINAL (approved) mode. If the record is not complete or a value is not valid, an error is returned. |
| exists | The Assistance.exists service checks for the existence of a given record. |

| Service Type/Name | Service Description |
|------------------------------|---|
| correct | The Assistance.correct service changes an existing assistance without creating a new record. |
| isExistingAssistanceComplete | The Assistance.IsExistingAssistanceComplete does the data validation and business rule validation for an already existing record. |

7.1.1.2.1 get

- The get service retrieves the existing Assistance information and returns the record in the XML format specified by the schema. If multiple records or no records are found for the given assistance Id, an error message is returned to the user.
- All users of the system are allowed to access the get service.

Input Parameters

| Parameter Name | Parameter Type | Name Space |
|-------------------|---------------------------------|----------------|
| authenticationKey | userAuthenticationKeyType | Service.xsd |
| assistanceID | complexType assistanceIDType | Assistance.xsd |

Output Parameters

| Parameter Name | Parameter Type | Name Space |
|-----------------------|--|-----------------|
| getAssistanceResponse | complexType getAssistanceResponseType | Assistance.wsdl |

The output parameters in the response contain the following:

Success Output

| Parameter Name | Parameter Type | Contents |
|----------------|----------------------------|---|
| assistance | complexType assistanceType | XML representation of the assistance record |

Example:

```

<getAssistanceResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <outputMessages>
    ...
  </outputMessages>
  <assistance>
    ...
  </assistance>
</getAssistanceResponse>
    
```

Failure Output

| Parameter Name | Parameter Type | Sample Error Contents |
|----------------|---------------------------------|---|
| ListOfErrors | complexType listOfErrorsType | 1. Assistance record does not exist 2. Multiple records exists for the given assistance Id. Please provide required information to retrieve the record |

7.1.1.2.2 getList

- The getList service retrieves the existing Assistance information that satisfies the specified criteria in the request..
- On successful retrieval, it returns the record in the XML format specified by the schema.

Input Parameters

| Parameter Name | Parameter Type | Name Space |
|--------------------------|---|----------------|
| AuthenticationKey | userAuthenticationKeyType | Service.xsd |
| assistanceSearchCriteria | complexType assistanceSearchCriteriaType | Assistance.xsd |

Output Parameters

| Parameter Name | Parameter Type | Name Space |
|---------------------------|--|-----------------|
| getListAssistanceResponse | complexType getListAssistanceResponseType | Assistance.wsdl |

The output parameters in the response contain the following:

Success Output

| Parameter Name | Parameter Type | Contents |
|---------------------------|--|----------------------------------|
| listOfAssistanceSummaries | complexType listOfAssistanceSummariesType | List of Assistance Summaries XML |

Example:

```
<getListAssistanceResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <listOfAssistanceSummaries>
  <outputMessages>
    ...
  </outputMessages>
  <assistance>
    ...
  </assistance>
</ listOfAssistanceSummaries>
</ getListAssistanceResponse>
```

Failure Output

| Parameter Name | Parameter Type | Sample Error Contents |
|-----------------------------|--|---|
| ListOfErrors | complexType listOfErrorsType | 1. Service unavailable 2. No records found. |
| ListOfWarnings | complexType listOfWarningsType | 1. Search Criteria too broad; enter specific values to search on. |
| ListOfInformationalMessages | complexType listOfInformationalMessagesType | 1. No assistance records available for search criteria. |

7.1.1.2.3 create

- The Assistance.create service creates a new assistance record in DRAFT status. The service is authenticated before creating the record.
- The document will be created if the user has ‘create’ privileges on Assistance module.

Input Parameters

| Parameter Name | Parameter Type | Name Space |
|-------------------|----------------------------|----------------|
| authenticationKey | userAuthenticationKeyType | Service.xsd |
| assistanceXML | complexType assistanceType | Assistance.xsd |

Output Parameters

| Parameter Name | Parameter Type | Name Space |
|--------------------------|---|-----------------|
| createAssistanceResponse | complexType createAssistanceResponse | Assistance.wsdl |

The output parameter in the response is wrapped as follows:

Success Output

| Parameter Name | Parameter Type | Contents |
|----------------|----------------------------|---|
| Assistance | complexType assistanceType | XML representation of the created assistance record |

Example:

```
<createAssistanceResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <outputMessages>
    ...
  </outputMessages>
  <assistance>
    ...
  </assistance>
</createAssistanceResponse >
```

Failure Output

| Parameter Name | Parameter Type | Sample Error Contents |
|----------------|---------------------------------|--|
| ListOfErrors | complexType listOfErrorsType | 1. An Assistance record with the FAIN already exists 2. Agency code is required |

7.1.1.2.4 update

- The Assistance.update service is the API to update and perform modifications to the existing assistance in draft mode.
- The Assistance.update method expects only the assistance ID and the required information to update the assistance information pertaining to the user.
- Any Awarding officer from the same awarding office is allowed to update the assistance record.

Input Parameters

| Parameter Name | Parameter Type | Name Space |
|-------------------|---------------------------|-------------|
| AuthenticationKey | userAuthenticationKeyType | Service.xsd |

| | | |
|------------|----------------------------|----------------|
| Assistance | complexType assistanceType | Assistance.xsd |
|------------|----------------------------|----------------|

Output Parameters

| Parameter Name | Parameter Type | Name Space |
|--------------------------|---|-----------------|
| updateAssistanceResponse | complexType updateAssistanceResponse | Assistance.wsdl |

The output parameters in the response contain the following:

Success Output

| Parameter Name | Parameter Type | Contents |
|----------------|-------------------------------|---|
| assistance | complexType assistanceType | XML representation of the updated assistance record |

Example:

```
<updateAssistanceResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <outputMessages>
    ...
  </outputMessages>
  <assistance>
    ...
  </assistance>
</updateAssistanceResponse>
```

Failure Output

| Parameter Name | Parameter Type | Sample Error Contents |
|----------------|---------------------------------|--|
| ListOfErrors | complexType listOfErrorsType | 1. Assistance for the FAIN not found. 2. User not authorized to update this contract. |

7.1.1.2.5 delete

- This service deletes the document
- This keeps track of an audit entry for all the FINAL deleted records

Input Parameters

| Parameter Name | Parameter Type | Name Space |
|-------------------|------------------------------|----------------|
| AuthenticationKey | userAuthenticationKeyType | Service.xsd |
| AssistanceID | complexType assistanceIDType | Assistance.xsd |

Output Parameters

| Parameter Name | Parameter Type | Name Space |
|--------------------------|--|-----------------|
| deleteAssistanceResponse | complexType deleteAssistanceResponse Type | Assistance.wsdl |

The output parameters in the response contain the following:

Success Output

| Parameter Name | Parameter Type | Contents |
|----------------|----------------|----------|
|----------------|----------------|----------|

| | | |
|-----------|-------------|--|
| isDeleted | xsd:boolean | True or false, representing whether the assistance is deleted or not |
|-----------|-------------|--|

Example:

```
<deleteAssistanceResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <outputMessages>
    ...
  </outputMessages>
  <isDeleted>
    true
  </isDeleted>
</ deleteAssistanceResponse >
```

Failure Output

| Parameter Name | Parameter Type | Sample Error Contents |
|----------------|---------------------------------|-----------------------------|
| ListOfErrors | complexType listOfErrorsType | 1. FAIN/URI does not exist. |

7.1.1.2.6 isComplete

- The isComplete service checks for the completeness of the assistance data.
- It checks for the existence of all mandatory fields in the Assistance, does data validation, applies business validations and returns success or failure to the user.
- In case of failure, the error information lists all the messages, codes and the data elements involved with the error.

Input Parameters

| Parameter Name | Parameter Type | Name Space |
|-------------------|----------------------------|----------------|
| AuthenticationKey | userAuthenticationKeyType | User.xsd |
| assistance | complexType assistanceType | Assistance.xsd |

Output Parameters

| Parameter Name | Parameter Type | Name Space |
|------------------------------|--|-----------------|
| isCompleteAssistanceResponse | complexType isCompleteAssistanceResponse Type | Assistance.wsdl |

The output parameters in the response contain the following:

Success Output

| Parameter Name | Parameter Type | Contents |
|----------------|----------------|---|
| isCompleted | xsd:boolean | True or false, representing whether the assistance is complete or otherwise |

Example:

```
<isCompleteAssistanceResponse>
  <requestNumber>549212</requestNumber>
```



```

    <confirmationNumber>329743</confirmationNumber>
    <outputMessages>
        ...
    </outputMessages>
    <isCompleted>
        true
    </isCompleted>
</ isCompleteAssistanceResponse >
    
```

Failure Output

| Parameter Name | Parameter Type | Sample Error Contents |
|----------------|---------------------------------|---|
| ListOfErrors | complexType listOfErrorsType | 1. <FAIN> must not contain special characters 2. CFDA number is invalid 3. Recipient DUNS specified for the assistance does not exist 4. Invalid CFDA Number |

7.1.1.2.7 approve

- This service approves the document, after doing a completeness check by using the isComplete service.
- Document must be in DRAFT status.
- In case of failure, the error response contains fields that were not filled in with values as well as fields that contain invalid data and business rule violations.

Input Parameters

| Parameter Name | Parameter Type | Name Space |
|-------------------|------------------------------|----------------|
| AuthenticationKey | userAuthenticationKeyType | Service.xsd |
| AssistanceID | complexType assistanceIDType | Assistance.xsd |

Output Parameters

| Parameter Name | Parameter Type | Name Space |
|---------------------------|--|-----------------|
| approveAssistanceResponse | complexType approveAssistanceResponse Type | Assistance.wsdl |

The output parameters in the response contain the following:

Success Output

| Parameter Name | Parameter Type | Contents |
|----------------|----------------|---|
| isApproved | xsd:boolean | True or false, representing whether the assistance is approved or not |

Example:

```

<approveAssistanceResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <outputMessages>
    ...
    
```

```

</outputMessages>
<isApproved>
  true
</isApproved>
</ approveAssistanceResponse >
    
```

Failure Output

| Parameter Name | Parameter Type | Sample Error Contents |
|----------------|---------------------------------|--|
| ListOfErrors | complexType listOfErrorsType | 2. FAIN must not contain special characters 3. Action Date cannot be later than tomorrows date. |

7.1.1.2.8 exists

- This service checks whether an assistance record, as per the criteria in the request, exists in the system and returns a success or failure response.
- The service also checks for valid authorization of the requesting user before sending back the response.

Input Parameters

| Parameter Name | Parameter Type | Name Space |
|-------------------|------------------------------|----------------|
| AuthenticationKey | userAuthenticationKeyType | Service.xsd |
| AssistanceID | complexType assistanceIDType | Assistance.xsd |

Output Parameters

| Parameter Name | Parameter Type | Name Space |
|--------------------------|--|-----------------|
| existsAssistanceResponse | complexType existsAssistanceResponse Type | Assistance.wsdl |

The output parameter in the response contains the output wrapped as follows:

Success Output

| Parameter Name | Parameter Type | Contents |
|----------------|----------------|--|
| isExists | xsd:boolean | True or false, representing whether the assistance exists or not |

Example:

```

<existsAssistanceResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <outputMessages>
    ...
  </outputMessages>
  <isExists>
    true
  </isExists>
</ existsAssistanceResponse >
    
```

Failure Output

| Parameter Name | Parameter Type | Sample Error Contents |
|----------------|---------------------------------|---|
| ListOfErrors | complexType listOfErrorsType | 1. Service Unavailable. 2. No records found. |

7.1.1.2.9 correct

- Assistance.correct service modifies the document, after doing a completeness check by using the isComplete service.
- Document must be in FINAL status.
- In case of failure, the error response contains fields that were not filled in with values as well as fields that contain invalid data and business rule violations. No changes to the document are performed
- This service should be used to correct any typographical errors or incorrect data in the system.

Input Parameters

| Parameter Name | Parameter Type | Name Space |
|-------------------|----------------------------|----------------|
| AuthenticationKey | userAuthenticationKeyType | Service.xsd |
| Assistance | complexType assistanceType | Assistance.xsd |

Output Parameters

| Parameter Name | Parameter Type | Name Space |
|---------------------------|--|-----------------|
| correctAssistanceResponse | complexType correctAssistanceResponse Type | Assistance.wsdl |

The output parameter in the response contains the following:

Success Output

| Parameter Name | Parameter Type | Contents |
|----------------|-------------------------------|---|
| assistance | complexType assistanceType | XML representation of the updated assistance record |

Example:

```
<correctAssistanceResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <outputMessages>
    ...
  </outputMessages>
  <assistance>
    ...
  </assistance>
</correctAssistanceResponse >
```

Failure Output

| Parameter Name | Parameter Type | Sample Error Contents |
|----------------|---------------------------------|--|
| ListOfErrors | complexType listOfErrorsType | 1. Assistance record not found in the System. 2. Action Date cannot be later than today's date. |

7.1.1.2.10 isExistingAssistanceComplete

- The Assistance. isExistingAssistanceComplete service checks for the completeness of an already saved assistance.
- It checks for the existence of all mandatory fields in the Assistance record, does data validation, applies business rules validations and returns success or failure to the user.

- In case of failure, the error information would list all the messages, codes and the data elements involved with the error.

Input Parameters

| Parameter Name | Parameter Type | Sample Error Contents |
|-------------------|---------------------------------|-----------------------|
| AuthenticationKey | userAuthenticationKeyType | Service.xsd |
| AssistanceID | ComplexType assistanceIDType | Assistance.xsd |

Output Parameters

| Parameter Name | Parameter Type | Name Space |
|--------------------------------------|---|-----------------|
| isExistingAssistanceCompleteResponse | complexType isExistingAssistanceCompleteResponse Type | Assistance.wsdl |

The output parameter in the response is wrapped as follows:

Success Output

| Parameter Name | Parameter Type | Contents |
|----------------|----------------|--|
| isCompleted | xsd:boolean | True or false, representing whether the existing assistance is complete or not |

Example:

```

< isExistingAssistanceCompleteResponse >
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <outputMessages>
    ...
  </outputMessages>
  <isCompleted >
    true
  </isCompleted >
</ isExistingAssistanceCompleteResponse >
    
```

Failure Output

| Parameter Name | Parameter Type | Sample Error Contents |
|----------------|---------------------------------|---|
| ListOfErrors | complexType listOfErrorsType | 1. Action Date cannot be later than tomorrow’s date. 2. CFDA Number is invalid |

7.2 GUI Services

7.2.1 Assistance Data Collection

The GUI services WSDL specification lists all the available services, schema definitions, transport endpoints and name spaces to integrate and collect assistance data.

7.2.1.1 Service meta-specifications in WSDL

Below table gives the parameter names used in the Assistance WSDL file

| WSDL Parameter | Value |
|----------------|-------|
|----------------|-------|

| | |
|--------------------|--------------------|
| PortType | AssistancePortType |
| Binding | AssistanceBinding |
| Soap-binding style | Rpc |
| TargetNameSpace | Assistance.wSDL |

7.2.1.2 Assistance Service methods

Below table gives all the services provided in the GUI services module.

| Service Type/Name | Service Description |
|--------------------------|--|
| getExistingAssistanceURL | Assistance.getExistingAssistanceURL service returns the FPDS-NG web page URL to be invoked for updating an existing assistance record from any external application. |
| getNewAssistanceURL | Assistance.getNewAssistanceURL service returns the FPDS-NG web page URL to be invoked from any external application. |

7.2.1.2.1 getExistingAssistanceURL

- The Assistance.getExistingAssistanceURL service gets the URL to launch the Assistance web page with the Assistance data existing in the database, identified by the ID provided in the request.
- When accessed by a web browser, the web page is pre-populated with the Assistance data fetched from the database using the assistance ID sent in the request.
- This is used by external systems that might want to perform assistance related transactions using the FPDS-NG system directly.

Input Parameters

| Parameter Name | Parameter Type | Name Space |
|-------------------|---------------------------------|----------------|
| AuthenticationKey | userAuthenticationKeyType | Servicexsd |
| assistanceID | complexType assistanceIDType | Assistance.xsd |

Output Parameters

| Parameter Name | Parameter Type | Name Space |
|----------------|---|-----------------|
| AssistanceURL | complexType getExistingAssistanceURLResponse Type | Assistance.wSDL |

The output parameters in the response contain the following:

Success Output

| Parameter Name | Parameter Type | Contents |
|----------------|----------------------------|---|
| assistance | complexType assistanceType | XML representation of the assistance record |

Example:

```

<getExistingAssistanceURLResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <outputMessages>
  ...
  </outputMessages>
  <assistanceURL>

```

```

    ...
  </ assistanceURL >
</ getExistingAssistanceURLResponse >

```

Failure Output

| Parameter Name | Parameter Type | Sample Error Contents |
|----------------|---------------------------------|---|
| ListOfErrors | complexType listOfErrorsType | User does not have authority for this function. |

7.2.1.2.2 getNewAssistanceURL

- The Assistance.getNewAssistanceURL service gets the URL to launch Assistance web page screen with the Assistance data provide in the request. The service is authenticated before sending the URL.
- When accessed from a browser, the web page shows up pre-populated with the Assistance data sent in the request.

Input Parameters

| Parameter Name | Parameter Type | Name Space |
|-------------------|----------------------------|----------------|
| AuthenticationKey | userAuthenticationKeyType | Service.xsd |
| Assistance | complexType assistanceType | Assistance.xsd |

Output Parameters

| Parameter Name | Parameter Type | Name Space |
|----------------|--|-----------------|
| assistanceURL | complexType getNewAssistanceURLResponse Type | Assistance.wsdl |

The output parameter in the response is wrapped as follows:

Success Output

| Parameter Name | Parameter Type | Contents |
|----------------|----------------------------|---|
| Assistance | complexType assistanceType | URL to invoke Assistance web page with contents filled in from the request. |

Example:

```

<getNewAssistanceURLResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <outputMessages>
    ...
  </outputMessages>
  <assistanceURL>
    ...
  </ assistanceURL >
</ getNewAssistanceURLResponse >

```

Failure Output

| Parameter Name | Parameter Type | Sample Error Contents |
|----------------|---------------------------------|--|
| ListOfErrors | complexType listOfErrorsType | 1. User does not have authority for this function. |

APPENDIX A Definition and Acronyms

All standard and non-standard terms and abbreviations used in this specifications document are explained in the following table.

Acronyms

| Acronym | Definition |
|------------|---------------------------------------|
| API | Application Programming Interface |
| CRU | Create, Read, Update |
| CRUD | Create, Read, Update, Delete |
| EJB | Enterprise Java Beans |
| FPDS-NG | Federal Procurement Data System |
| FTP | File Transfer Protocol |
| GUI | Graphical User Interface |
| HTTP | HyperText Transfer Protocol |
| HTTPS | Secure HyperText Transfer Protocol |
| MIME | Multipurpose Internet Mail Extensions |
| OLAP | On-Line Analytical Processing |
| PSC | Product Service Codes |
| RPC or rpc | Remote Process Call |
| SOAP | Simple Object Access Protocol |
| URL | Uniform Resource Locator |
| WSDL | Web Services Definition Language |
| XML | eXtensible Markup Language |
| XSD | XML Schema Definition |
| XSL | eXtensible Stylesheet Language |

Definitions

The following list contains definitions of the terms used in this document:

- **Port Type** – A Port type is an abstract set of operations supported by one or more web service providers (i.e., all of the web services available for an award).
- **Binding** – A concrete protocol and data format specification for a particular port type.
- **Types** – A container for data type definitions using some type system (such as XSD).
- **Service** – A collection of related endpoints
- **Target Name Space**. The target namespace serves to identify the namespace within which the association between the component and its name exists

Appendix B References

Normative References

[RFC 2119]

IETF "RFC 2119: Keywords for use in RFCs to Indicate Requirement Levels", S. Bradner, March 1997. (See <http://www.ietf.org/rfc/rfc2119.txt>.)

[RFC 2396]

IETF "RFC 2396: Uniform Resource Identifiers (URI): Generic Syntax", T. Berners-Lee, R. Fielding, L. Masinter, August 1998. (See <http://www.ietf.org/rfc/rfc2396.txt>.)

[RFC 2616]

IETF "RFC 2616: Hypertext Transfer Protocol -- HTTP/1.1", R. Fielding, J. Gettys, J. C. Mogul, H. Frystyk Nielsen, T. Berners-Lee, January 1997. (See <http://www.ietf.org/rfc/rfc2616.txt>.)

[XML Schema Part 1]

W3C Recommendation "XML Schema Part 1: Structures", Henry S. Thompson, David Beech, Murray Maloney, Noah Mendelsohn, 2 May 2001. (See <http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/>.)

[XML Schema Part 2]

W3C Recommendation "XML Schema Part 2: Datatypes", Paul V. Biron, Ashok Malhotra, 2 May 2001. (See <http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/>.)

[SOAP Part 0]

W3C Proposed Recommendation "SOAP Version 1.2 Part 0: Primer", Nilo Mitra, (see <http://www.w3.org/TR/soap12-part1>.)

Informative References

[WSDL 1.1]

Web Services Description Language (WSDL) 1.1 W3C Note 15 March 2001 (See <http://www.w3.org/TR/2001/NOTE-wsdl-20010315>).

[XML Schema Part 1]

W3C Recommendation "XML Schema Part 1: Structures", Henry S. Thompson, David Beech, Murray Maloney, Noah Mendelsohn, 2 May 2001. (See <http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/>.)

[XML Schema Part 2]

W3C Recommendation "XML Schema Part 2: Datatypes", Paul V. Biron, Ashok Malhotra, 2 May 2001. (See <http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/>.)

[SOAP Part 0]

W3C Proposed Recommendation "SOAP Version 1.2 Part 0: Primer", Nilo Mitra, (see <http://www.w3.org/TR/soap12-part1>.)