

V1.5 Federal Procurement Data System (FPDS) Data Collection

Web Services API Interface Specifications Document Version 1.5

Last updated: June 12, 2021

Prepared by:



2300 Dulles Station Blvd
Herndon, VA 20171
United States

Table of Contents

<u>SECTION</u>	<u>PAGE</u>
1 SCOPE	1
1.1 PURPOSE	1
2 CONCEPTUAL ARCHITECTURE	1
Figure 2–1. FPDS High Level Architecture	1
3 SERVICE ORIENTED ARCHITECTURE	1
4 FPDS SERVICE ARCHITECTURE	2
Figure 4–1. FPDS Service Architecture	3
4.1 DOCUMENT OVERVIEW	3
4.2 NEW VERSION UPDATES FOR 1.5	3
Table 4-1. Version 1.5 Changes	3
5 ARCHITECTURAL GOALS AND CONSTRAINTS	4
6 VERSION CONTROL	4
6.1 WEB SERVICE API VERSIONING	4
6.1.1 Implementation	4
6.1.2 Service Input	4
6.1.3 Service Output	4
6.1.4 Service Errors	5
6.1.5 Schema Changes	5
6.2 API VERSIONING CONVENTIONS	5
7 FPDS WEB SERVICES API.....	5
7.1 SERVICE OBJECT MODEL	5
Figure 7–1. FPDS Services Hierarchy	6
Figure 7–2 shows the business services available in FPDS.....	6
Figure 7–2. Data Collection Business Services.....	6
Figure 7–3. Reporting Services.....	7
<i>*Please note that FPDS no longer utilizes the Reporting services to provide reports to the user community. The above diagram remains for historical purposes.</i>	7
Figure 7–4 shows the system administration business services available in FPDS.....	7
Figure 7–4. System Administration Business Services	7
7.2 SERVICE DEFINITIONS AND USAGE	8
7.2.1 WSDL Specifications.....	8
8 API STANDARDS AND GENERIC DETAILS.....	8
8.1 STANDARD METHOD SIGNATURES.....	9
8.1.1 Service Input Parameters.....	9
Table 8-1. Service Input Parameters	9
Table 8-2. Data Operation Descriptions.....	9
8.1.2 Service Output Parameters.....	10
Table 8-3. Service Output Parameters.....	10
Figure 8–1. Data Collection Business Services.....	10
8.1.2.1 Output Parameter Types.....	10
Table 8-4. Output Parameter Types	11
8.1.2.2 Sample Response Error Messages.....	11
9 SPECIFIC API DETAILS.....	13
9.1 BUSINESS SERVICES	13
9.1.1 Data Collection.....	13
9.1.1.1 Contracts	13

9.1.1.1.1	Contract	13
9.1.1.1.1.1	Service meta-specifications in WSDL	13
9.1.1.1.1.2	Contract Service methods	13
9.1.1.1.2	get	13
9.1.1.1.2.1	getList	14
9.1.1.1.3	Award	15
9.1.1.1.3.1	Service meta-specifications in WSDL	15
9.1.1.1.3.2	Award Service methods	15
9.1.1.1.3.3	create	16
9.1.1.1.3.4	update	17
9.1.1.1.3.5	isComplete	18
9.1.1.1.3.6	approve	19
9.1.1.1.3.7	get	20
9.1.1.1.3.8	getList	21
9.1.1.1.3.9	exists	22
9.1.1.1.3.10	correct	23
9.1.1.1.3.11	isExistingAwardComplete	24
9.1.1.1.3.12	createFromTemplate	25
9.1.1.1.3.13	delete	26
9.1.1.1.4	IDV	27
9.1.1.1.4.1	Service meta-specifications in WSDL	27
9.1.1.1.4.2	IDV Service methods	27
9.1.1.1.4.3	Create	28
9.1.1.1.4.4	update	28
9.1.1.1.4.5	isComplete	29
9.1.1.1.4.6	approve	30
9.1.1.1.4.7	get	31
9.1.1.1.4.8	getList	32
9.1.1.1.4.9	exists	33
9.1.1.1.4.10	correct	34
9.1.1.1.4.11	isExistingIDVComplete	35
9.1.1.1.4.12	createFromTemplate	36
9.1.1.1.4.13	delete	37
9.1.1.1.5	OtherTransactionAward	38
9.1.1.1.5.1	Service meta-specifications in WSDL	38
9.1.1.1.5.2	OtherTransactionAward Service methods	38
9.1.1.1.5.3	create	39
9.1.1.1.5.4	update	40
9.1.1.1.5.5	isComplete	41
9.1.1.1.5.6	approve	42
9.1.1.1.5.7	get	43
9.1.1.1.5.8	getList	43
9.1.1.1.5.9	exists	45
9.1.1.1.5.10	correct	45
9.1.1.1.5.11	delete	46
9.1.1.1.6	OtherTransactionIDV	47
9.1.1.1.6.1	Service meta-specifications in WSDL	47
9.1.1.1.6.2	OtherTransactionIDV Service methods	48
9.1.1.1.6.3	Create	48
9.1.1.1.6.4	update	49
9.1.1.1.6.5	isComplete	50
9.1.1.1.6.6	approve	51
9.1.1.1.6.7	get	52
9.1.1.1.6.8	getList	53
9.1.1.1.6.9	exists	54
9.1.1.1.6.10	correct	55
9.1.1.1.6.11	delete	56
9.1.1.1.7	Close	57
9.1.1.1.7.1	Service meta-specifications in WSDL	57
9.1.1.1.7.2	Close Service methods	57
9.1.1.1.7.3	close	57
9.1.1.1.8	ContractSummary	58

- 9.1.1.1.8.1 Service meta-specifications in WSDL 58
- 9.1.1.1.8.2 ContractSummary Service methods 58
- 9.1.1.1.8.3 getList 58
- 9.1.1.1.8.4 getListWithHistory* 60
- 9.1.1.2 Organizations 63
 - 9.1.1.2.1 Department 63
 - 9.1.1.2.1.1 Service meta-specifications in WSDL 63
 - 9.1.1.2.1.2 Departments Service Methods 63
 - 9.1.1.2.1.3 create 63
 - 9.1.1.2.1.4 update 64
 - 9.1.1.2.1.5 get 65
 - 9.1.1.2.1.6 getList 66
 - 9.1.1.2.1.7 exists 67
 - 9.1.1.2.1.8 delete 68
 - 9.1.1.2.1.9 disable 69
 - 9.1.1.2.1.10 enable 70
 - 9.1.1.2.2 Agency 71
 - 9.1.1.2.2.1 Service meta-specifications in WSDL 71
 - 9.1.1.2.2.2 Agency Service methods 71
 - 9.1.1.2.2.3 create 71
 - 9.1.1.2.2.4 update 72
 - 9.1.1.2.2.5 get 73
 - 9.1.1.2.2.6 getList 74
 - 9.1.1.2.2.7 exists 75
 - 9.1.1.2.2.8 delete 76
 - 9.1.1.2.2.9 disable 76
 - 9.1.1.2.2.10 enable 77
 - 9.1.1.2.3 Contracting Office 78
 - 9.1.1.2.3.1 Service meta-specifications in WSDL 78
 - 9.1.1.2.3.2 Contracting Office Service methods 78
 - 9.1.1.2.3.3 create 79
 - 9.1.1.2.3.4 update 80
 - 9.1.1.2.3.5 get 81
 - 9.1.1.2.3.6 getList 82
 - 9.1.1.2.3.7 exists 83
 - 9.1.1.2.3.8 delete 84
 - 9.1.1.2.3.9 disable 85
 - 9.1.1.2.3.10 enable 85
 - 9.1.1.2.4 Government Office 86
 - 9.1.1.2.4.1 Service meta-specifications in WSDL 86
 - 9.1.1.2.4.2 Government Office Service Methods 87
 - 9.1.1.2.4.3 create 87
 - 9.1.1.2.4.4 update 88
 - 9.1.1.2.4.5 get 89
 - 9.1.1.2.4.6 getList 90
 - 9.1.1.2.4.7 exists 91
 - 9.1.1.2.4.8 delete 92
 - 9.1.1.2.4.9 disable 93
- 9.1.1.3 Vendors 93
 - 9.1.1.3.1 Vendor 93
 - 9.1.1.3.1.1 Service meta-specifications in WSDL 94
 - 9.1.1.3.1.2 Vendor Service methods 94
 - 9.1.1.3.1.3 get 94
 - 9.1.1.3.1.4 getList 95
 - 9.1.1.3.1.5 exists 96
- 9.1.2 *Product/Service Classifications* 97
 - 9.1.2.1 NAICS 97
 - 9.1.2.1.1 Service meta-specifications in WSDL 97
 - 9.1.2.1.1.1 NAICS Service methods 97
 - 9.1.2.1.1.2 create 97
 - 9.1.2.1.1.3 update 98
 - 9.1.2.1.1.4 get 99

- 9.1.2.1.1.5 getList 100
- 9.1.2.1.1.6 exists 101
- 9.1.2.1.1.7 delete..... 102
- 9.1.2.1.1.8 disable..... 103
- 9.1.2.1.1.9 enable..... 104
- 9.1.2.2 PSC 105
 - 9.1.2.2.1 Service meta-specifications in WSDL 105
 - 9.1.2.2.1.1 PSC Service methods 105
 - 9.1.2.2.1.2 create..... 105
 - 9.1.2.2.1.3 update..... 106
 - 9.1.2.2.1.4 get 107
 - 9.1.2.2.1.5 getList 108
 - 9.1.2.2.1.6 exists 109
 - 9.1.2.2.1.7 delete..... 110
 - 9.1.2.2.1.8 disable..... 111
 - 9.1.2.2.1.9 enable..... 112
- 9.1.2.3 Locations..... 113
 - 9.1.2.3.1 Country 113
 - 9.1.2.3.1.1 Service meta-specifications in WSDL 113
 - 9.1.2.3.1.2 Countries Service methods..... 113
 - 9.1.2.3.1.3 create..... 113
 - 9.1.2.3.1.4 update..... 114
 - 9.1.2.3.1.5 get 115
 - 9.1.2.3.1.6 getList 116
 - 9.1.2.3.1.7 exists 117
 - 9.1.2.3.1.8 delete..... 118
 - 9.1.2.3.1.9 disable..... 119
 - 9.1.2.3.1.10 enable..... 120
 - 9.1.2.3.2 Place 121
 - 9.1.2.3.2.1 Service meta-specifications in WSDL 121
 - 9.1.2.3.2.2 Place Service methods 121
 - 9.1.2.3.2.3 create..... 121
 - 9.1.2.3.2.4 update..... 122
 - 9.1.2.3.2.5 get 123
 - 9.1.2.3.2.6 getList 124
 - 9.1.2.3.2.7 exists 125
 - 9.1.2.3.2.8 delete..... 126
 - 9.1.2.3.2.9 disable..... 127
 - 9.1.2.3.2.10 enable..... 128
 - 9.1.2.3.2.11 State 128
 - 9.1.2.3.2.12 Service meta-specifications in WSDL 129
 - 9.1.2.3.2.13 State Service methods 129
 - 9.1.2.3.2.14 getList 129
 - 9.1.2.3.2.15 Zip..... 130
 - 9.1.2.3.2.16 Service meta-specifications in WSDL 130
 - 9.1.2.3.2.17 Zip Service methods 130
 - 9.1.2.3.2.18 getList 130
- 9.1.2.4 System Administration..... 132
 - 9.1.2.4.1 User 132
 - 9.1.2.4.1.1 Service meta-specifications in WSDL 132
 - 9.1.2.4.1.2 User Service methods 132
 - 9.1.2.4.1.3 create..... 133
 - 9.1.2.4.1.4 update..... 134
 - 9.1.2.4.1.5 get 135
 - 9.1.2.4.1.6 getList 135
 - 9.1.2.4.1.7 authenticate 137
 - 9.1.2.4.1.8 enable..... 138
 - 9.1.2.4.1.9 disable..... 139
 - 9.1.2.4.1.10 exists 140
 - 9.1.2.4.1.11 authorize 141
 - 9.1.2.4.1.12 getObjectPrivilege 142
 - 9.1.2.4.1.13 getForgotPassword..... 142

9.2 GUI SERVICES 143

 9.2.1 Data Collection..... 143

 9.2.1.1 Contracts 143

 9.2.1.1.1 Contract 143

 9.2.1.1.1.1 Service meta-specifications in WSDL 143

 9.2.1.1.1.2 Contract Service Methods 144

 9.2.1.1.1.3 getNewContractURL 144

 9.2.1.1.1.4 getExistingContractURL..... 145

 9.2.1.1.2 Award 146

 9.2.1.1.2.1 Service meta-specifications in WSDL 146

 9.2.1.1.2.2 Award Service methods 146

 9.2.1.1.2.3 getNewAwardURL 146

 9.2.1.1.2.4 getExistingAwardURL 147

 9.2.1.1.2.5 getNewAwardURLFromTemplate..... 148

 9.2.1.1.3 IDV..... 149

 9.2.1.1.3.1 Service meta-specifications in WSDL 149

 9.2.1.1.3.2 IDV Service methods 149

 9.2.1.1.3.3 getNewIDVURL..... 149

 9.2.1.1.3.4 getExistingIDVURL 150

 9.2.1.1.3.5 getNewIDVURLFromTemplate 151

 9.2.1.1.4 OtherTransactionAward 152

 9.2.1.1.4.1 Service meta-specifications in WSDL 152

 9.2.1.1.4.2 OtherTransactionAward Service methods..... 152

 9.2.1.1.4.3 getNewOtherTransactionAwardURL..... 152

 9.2.1.1.4.4 getExistingOtherTransactionAwardURL..... 153

 9.2.1.1.5 OtherTransactionIDV 154

 9.2.1.1.5.1 Service meta-specifications in WSDL 154

 9.2.1.1.5.2 OtherTransactionIDV Service methods 154

 9.2.1.1.5.3 getNewOtherTransactionIDVURL 155

 9.2.1.1.5.4 getExistingOtherTransactionIDVURL..... 156

 9.2.1.2 Organizations 156

 9.2.1.2.1 Department 156

 9.2.1.2.1.1 Service meta-specifications in WSDL 157

 9.2.1.2.1.2 Departments Service Methods 157

 9.2.1.2.1.3 getNewDepartmentURL 157

 9.2.1.2.1.4 getExistingDepartmentURL..... 158

 9.2.1.2.2 Agency..... 159

 9.2.1.2.2.1 Service meta-specifications in WSDL 159

 9.2.1.2.2.2 Agency Service methods..... 159

 9.2.1.2.2.3 getNewAwardURL 159

 9.2.1.2.2.4 getExistingAgencyURL..... 160

 9.2.1.2.3 Contracting Office 162

 9.2.1.2.3.1 Service meta-specifications in WSDL 162

 9.2.1.2.3.2 Contracting Office Service methods 162

 9.2.1.2.3.3 getNewContractingOfficeURL 162

 9.2.1.2.3.4 getExistingContractingOfficeURL 163

 9.2.1.2.4 Government Office 164

 9.2.1.2.4.1 Service meta-specifications in WSDL 164

 9.2.1.2.4.2 Government Office Service methods 164

 9.2.1.2.4.3 getNewGovernmentOfficeURL 164

 9.2.1.2.4.4 getExistingGovernmentOfficeURL 165

 9.2.1.3 Locations..... 166

 9.2.1.3.1 Country 166

 9.2.1.3.1.1 Service meta-specifications in WSDL 166

 9.2.1.3.1.2 Countries Service methods..... 166

 9.2.1.3.1.3 getNewCountryURL 167

 9.2.1.3.1.4 getExistingCountryURL 168

 9.2.1.3.2 Place 168

 9.2.1.3.2.1 Service meta-specifications in WSDL 169

 9.2.1.3.2.2 Place Service methods 169

 9.2.1.3.2.3 getNewPlaceURL 169

 9.2.1.3.2.4 getExistingPlaceURL..... 170

9.2.1.4 Product/Service Classifications 171

 9.2.1.4.1 NAICS 171

 9.2.1.4.1.1 Service meta-specifications in WSDL 171

 9.2.1.4.1.2 NAICS Service methods 171

 9.2.1.4.1.3 getNewNAICSURL 171

 9.2.1.4.1.4 getExistingNAICSURL 172

 9.2.1.4.2 PSC 173

 9.2.1.4.2.1 Service meta-specifications in WSDL 173

 9.2.1.4.2.2 PSC Service methods 173

 9.2.1.4.2.3 getNewPSCURL 173

 9.2.1.4.2.4 getExistingPSCURL 174

9.2.1.5 System Administration 175

 9.2.1.5.1 User 175

 9.2.1.5.1.1 Service meta-specifications in WSDL 175

 9.2.1.5.1.2 User Service methods 175

 9.2.1.5.1.3 getNewUserURL 175

 9.2.1.5.1.4 getExistingUserURL 176

APPENDIX A DEFINITION AND ACRONYMS 178

 ACRONYMS 178

 DEFINITIONS 178

APPENDIX B REFERENCES 179

 NORMATIVE REFERENCES 179

 INFORMATIVE REFERENCES 179

Revision History

Date	Version Number	Changed By	Change Summary
09/02/2003	1.0	GCE	Initial version
11/03/2003	1.1	GCE	Updates to include the version 1.1 web service changes <ol style="list-style-type: none"> 1. CCR Exceptions 2. Agency ID
06/01/2004	1.2	GCE	Updates to include the version 1.2 web service specification changes <ol style="list-style-type: none"> 1. Document Key changes 2. Award and IDV template functionality
03/21/2008	1.3	GCE	Updates to include the version 1.3 web service specification changes <ol style="list-style-type: none"> 1. Award and IDV Transfer functionality deprecated 2. Award and IDV PIID change functionality deprecated
05/27/2009	1.3	GCE	Updates to remove void service form the list of available services.
10/20/2016	1.5	IBM	Updated the document with version 1.5 specifications.
06/12/2021	1.5	IBM	Updated the document with the following: <ul style="list-style-type: none"> - Removed all references of '-NG' from 'FPDS-NG' throughout document

1 SCOPE

1.1 Purpose

Under GSA’s initiative and direction, Federal Procurement Data System (FPDS) has been reengineered as a real-time federal enterprise information system. 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 FPDS to provide interoperability with various federal procurement systems.

2 Conceptual Architecture

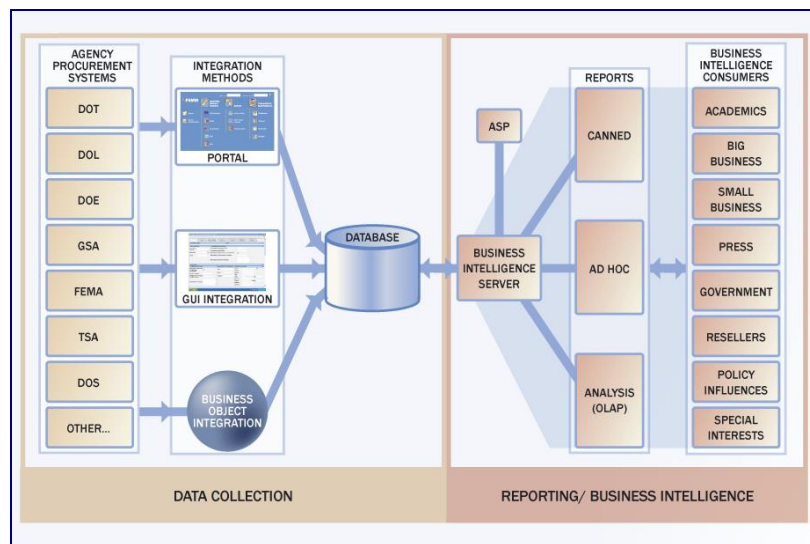
FPDS consists of two functional domains and one administrative domain. The two functional domains, Data Collection and Business Intelligence/Reporting, are depicted in Figure 2–1.

Data Collection: This domain provides multiple mechanisms to feed contract Award data from procurement systems throughout the federal government to FPDS. Emphasis is on real-time integration to shorten the lag time between contract Award and data availability in FPDS, and to increase data quality by removing batch interfaces and the need to re-key data into agency systems.

Business Intelligence/Reporting: This domain provides multiple mechanisms to report FPDS data to a wide spectrum of interested parties. The reporting mechanisms include canned, ad hoc, and OLAP analysis reporting which are delivered based on the format and schedule preferences established by the user.

System Administration: This domain manages user profiles, user authentication, reference tables, and other system functions such as purging old error records, and monitoring data quality.

Figure 2–1. FPDS High Level Architecture



3 Service Oriented Architecture

The FPDS system architecture, shown in Figure 2–1, 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 FPDS 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, GUI screens, and reports 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. Reusability also provides the government with the ability to construct authoritative services for vital information (e.g. NAICS codes, e data, etc).

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
- **UDDI** which allows users and applications to locate or publish web services in a registry.

Table 3-1. 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

The Federal Enterprise Architecture Program Management Office (FEA-PMO) recommends a J2EE/Web Services approach in *White Paper 1.3 published in October 2002*. The white paper provides a framework and guidance for the technology standards that support the 24 e-Government Initiatives, as well as other future efforts directed to reuse technology components across the federal government. Table 3-1 is an excerpt from White Paper 1.3 and shows the comparison ratings given by the Software Working Group (SAWG) for J2EE Web and .NET Services.

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

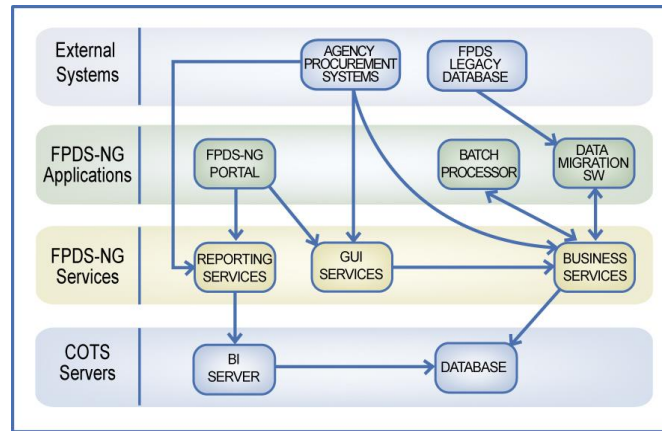
4 FPDS Service Architecture

Figure 4–1 describes the FPDS high level service architecture. It uses a building-block approach to maximize reusability. The FPDS service architecture contains several layers, each of which is fully reusable. For example, the business services are used by migration software, GUI services, and external procurement systems. The layers are:

- **COTS Layer:** This layer consists of the Oracle 10g database and the *Informatica*™ reporting server. The *Informatica*™ reporting server utilizes the Oracle 10g database for all data queries.
- **FPDS Services:** This layer consists of the GUI and business services which centralize all FPDS business logic. The GUI services layer represents all FPDS screens. The GUI screens use the business services to validate and post FPDS transactions.

- *FPDS Applications:* This layer represents FPDS software that employs reusable services. For example, the batch processor and migration software use the business services (Note: batch submission is no longer supported in Version 1.3)
- *External Systems:* This layer represents legacy systems and COTS products that wish to integrate with FPDS services. FPDS allows integrators use any of the business, GUI service or reporting services. For example, agencies may integrate with FPDS at the business services level or reuse the FPDS data collection screens.

Figure 4–1. FPDS Service Architecture



4.1 Document Overview

This document introduces the web services system architecture that exposes one point of entry to FPDS. 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 FPDS use the web services APIs to achieve their functionality.

1. GUI services that allow creation or modification of Awards, IDVs, reference and system administration data.
2. Business services that allow the integrators to launch the FPDS data collection screens from within the COTS/GOTS products
3. Reporting services that allow the integrators to launch FPDS reports from within the COTS/GOTS product

4.2 New Version updates for 1.5

With the release of Version 1.5 of the Business and GUI web services, it was necessary to make certain changes to the web services system architecture. These changes are listed in the table below, and included in the appropriate section throughout the document. The changes listed in this table are linked to the section in the document that addresses each particular change.

Table 4-1. Version 1.5 Changes

Service	Method	Change
BusinessServices - Close	close	This is a new method in this version and can be found in Section 9.1.1.1.6

5 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. The FPDS web services 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. **Flexibility:** An API is versioned, allowing clients to migrate to the next version when they are ready.
7. **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.
8. **Stateless:** An API service is stateless.
9. **Secure:** All API input contains the user and source data used for authentication.
10. **Error Processing:** The API returns a comprehensive and complete set of error codes and corresponding messages.
11. **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.

6 VERSION CONTROL

6.1 Web Service API Versioning

Business requirements that change an API will be appropriately collated and prioritized with the consent of GSA and new web services or new versions of existing web services will be engineered and deployed. Client systems can continue using the older version of the API until they are ready to migrate to the newer version.

The next sections describe version control of the Web Service API.

6.1.1 Implementation

The processing and business logic of the web services are artifacts of the web service that are prone to change, driven by business process modifications and enhancements.

When the Web Service API remains the same, but the business process carried out by the service changes, a new version is announced and released without affecting the current version.

6.1.2 Service Input

Changes to the input parameters of any particular web service are released as a new version.

6.1.3 Service Output

When the output parameters passed back to the caller as a web service response change, the changes will take effect in a separate, new version of the web service.

6.1.4 Service Errors

The errors section of the web service output is subject to change due to changes in the following:

- Input parameters
- Output parameters
- Business processes and rules underlying the web services
- Technical implementation changes to the web services.

The error changes will be made effective in the newer version of the Web Service API.

6.1.5 Schema Changes

The schema definitions for the domain input parameter(s) are modified to support changed or additional data. The method signature remains the same but a newer version of the Web Service API is published to process the new/deleted data in the schema.

6.2 API Versioning conventions

Newer versions of the API will be hosted under the URL, which can be located using the FPDS directory tree suffixed with the version number. For example:

Business Services WSDLs

<https://www.fpds.gov/FPDS/BusinessServices/DataCollection/contracts/1.5/Award.wsdl>

<https://www.fpds.gov/FPDS/BusinessServices/DataCollection/contracts/1.5/IDV.wsdl>

***NOTE:** For Beta testing integration, **www.fpds.gov** should be replaced with **beta.fpds.gov**

GUI Services WSDLs

<https://www.fpds.gov/FPDS/GUIServices/DataCollection/contracts/1.5/Award.wsdl>

<https://www.fpds.gov/FPDS/GUIServices/DataCollection/contracts/1.5/IDV.wsdl>

***NOTE:** For Beta testing integration, **www.fpds.gov** should be replaced with **beta.fpds.gov**

The schema namespaces referenced by the API follow the same naming convention. For example:

<https://www.fpds.gov/FPDS/schema/DataCollection/contracts/1.5/Award.xsd>

<https://www.fpds.gov/FPDS/schema/DataCollection/contracts/1.5/IDV.xsd>

All XSD schema files and versions are located at:

https://www.fpds.gov/fpdsng_cms/index.php/en/worksite.html

7 FPDS WEB SERVICES API

7.1 Service Object Model

The service object model depicts the data entities at the domain level as well as the lower level objects that are reused between the different modules or services of the system.

Figure 7–1 shows the service layer that caters to the operations on the objects. The operations include creating, retrieving, updating, validating, and other business layer activities that serve users through GUI services, batch processes, and reporting applications.

Figure 7-1. FPDS Services Hierarchy

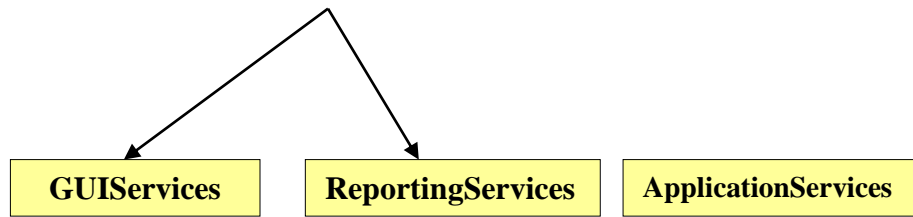
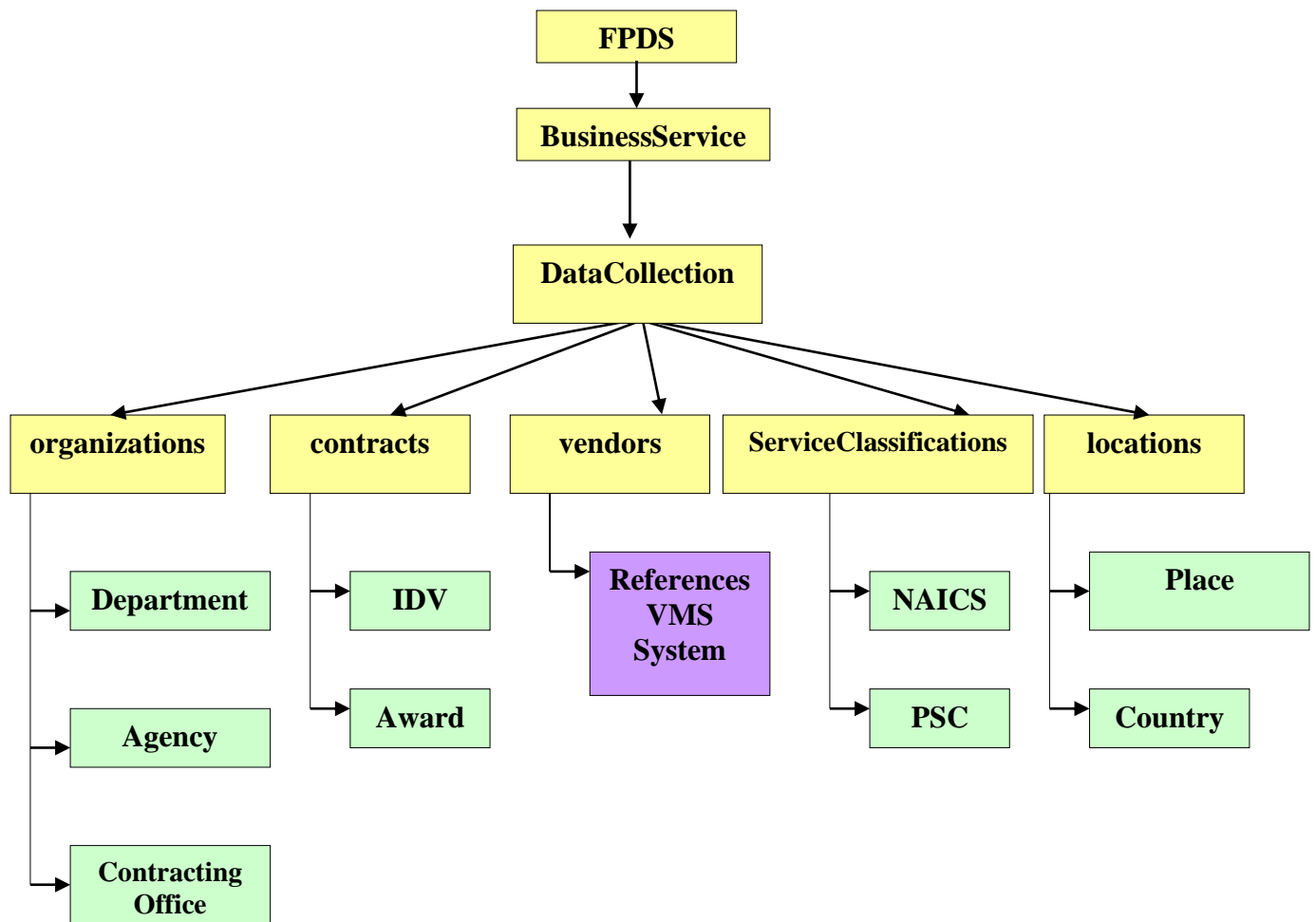


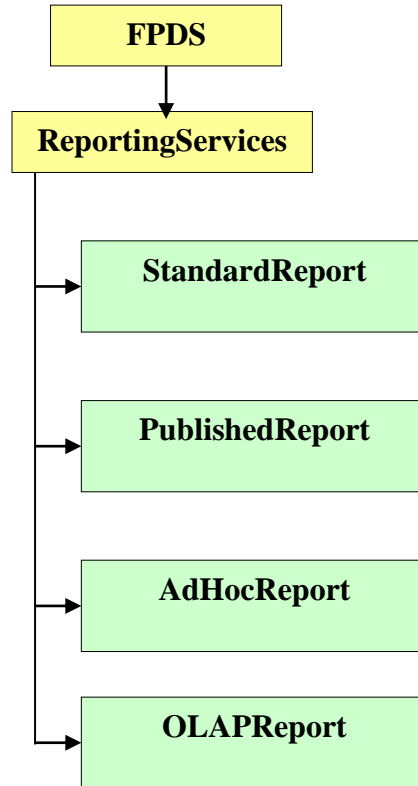
Figure 7-2 shows the business services available in FPDS.

Figure 7-2. Data Collection Business Services



Reporting services provide standard (canned), ad hoc, and OLAP analysis reports delivery based on the format and schedule preferences established by the user services.*

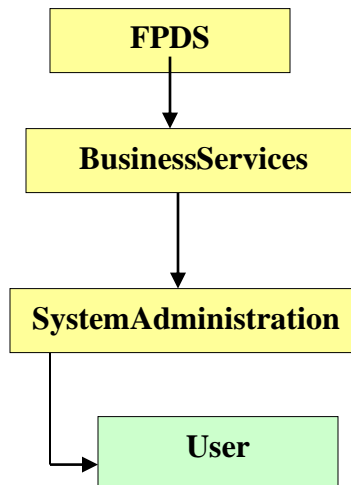
Figure 7–3. Reporting Services



**Please note that FPDS no longer utilizes the Reporting services to provide reports to the user community. The above diagram remains for historical purposes.*

Figure 7–4 shows the system administration business services available in FPDS

Figure 7–4. System Administration Business Services



7.2 Service Definitions and Usage

7.2.1 WSDL Specifications

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.

The bindings described in this document specifically deal with the use of WSDL in conjunction with SOAP 1.1, HTTP GET/POST, and MIME.

Table 7-1 shows the FPDS Web nomenclature. [Section 1.4.2 Definitions](#) contains definitions of the WSDL Parameters explained below.

Table 7-1. 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 FPDS Web Services include the targetNamespace in the corresponding WSDL and are named after the complex type or the domain level object, i.e., Award.xsd, Country.xsd, etc.

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.

8 API Standards and Generic details

The web services APIs of the FPDS domain objects encompass the following standards:

- Service calls in FPDS are authenticated by checking for valid User ID/Password.
- Service calls are checked for authorization before serving the request. For example, the ‘create’ service on Award will check if the user has the privilege to create an Award in FPDS.
- Web Services APIs supported for domain entities such as Award and IDV. This includes common business services such as create, get, update and delete.
- Service calls use a standard method signature. All the business classes in the FPDS 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.

8.1 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.

8.1.1 Service Input Parameters

Table 8-1 describes the service input parameter names and description.

Table 8-1. 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 8-2 describes the contents of the input XML based on the type of data operation.

Table 8-2. 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
isComplete	XML representation of the business object	XML is converted to the value object using JAXB and data is validated by the business rules engine. Returns isComplete errors, business validation errors and warning messages.
approve	XML representation of the key(s)	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)	Retrieved based on the key values specified in the input
getList	search criteria inputs serialized in predefined XML format	Query based on search criteria input
enable, disable	XML representation of the key(s)	Appropriate column flag value(s) are set / unset on the record specified by the key(s)

8.1.2 Service Output Parameters

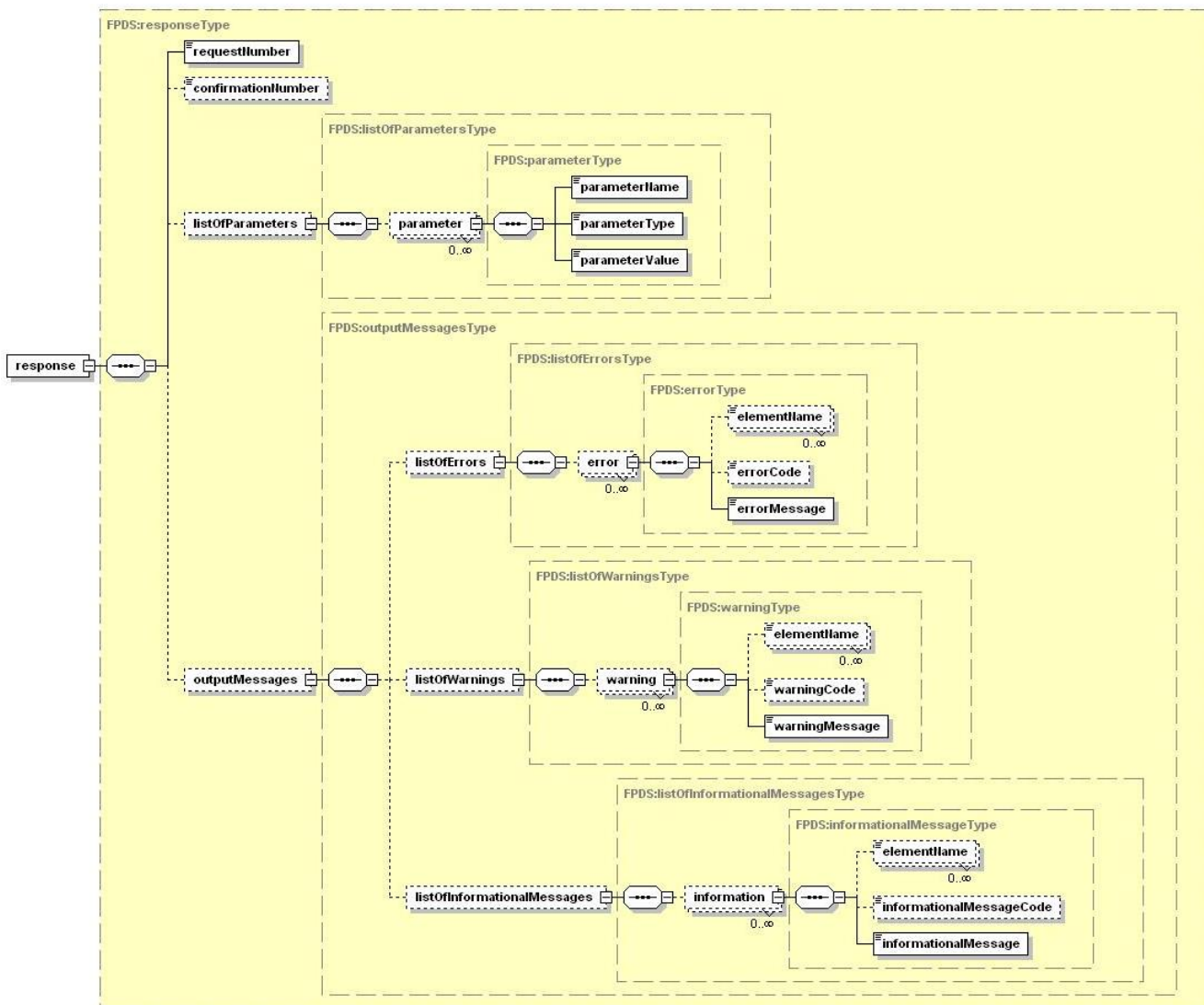
The service output parameters are represented in Meta XML strings. Table 8-3 describes the XML meta-response.

Table 8-3. 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 8-1.

Figure 8-1. Data Collection Business Services



8.1.2.1 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 Please refer to the FPDS Error Listings for a complete list of errors. The output parameter types are described in Table 8-4.

Table 8-4. Output Parameter Types

Operations	Confirmation Number		Specific API Output Parameters	Example Errors and Warnings
	Success	Failure		
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
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
correct	Positive Integer	-1	Corrected record	Business validation errors and warning messages
get	Positive Integer	-1	XML representation of the Business object	Not Found, mode errors
exists	Positive Integer	-1	Boolean true / false response	Not Found, insufficient query information errors
delete	Positive Integer	-1	Boolean true / false response	Not Found, insufficient query information errors, business restriction 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
enable, disable	Positive Integer	-1	XML representation of the enabled or disabled data	Not Found, insufficient query information errors

8.1.2.2 Sample Response Error Messages

```

<response>
<listOfErrors>
  <error>
    <elementName>searchCriteriaXML</elementName>
    <errorCode>10141</errorCode>
    <errorMessage>Service Unavailable Please try again later</errorMessage>
  </error>
  <error>
    <elementName>PIID</elementName>
    <errorCode>10200</errorCode>
    <errorMessage>Cannot create Award. PIID already exists in the database</errorMessage>
  </error>
</listOfErrors>
<listOfWarnings>
  <warning>

```

```
<elementName>BundledRequirement</elementName>
<warningCode>30501</warningCode>
<warningMessage> BundledRequirement is not required for Award type BPA call. Ignoring
the value.</warningMessage>
</warning>
</listOfWarnings>
<listOfInfoMessages>
<info>
<elementName>award/purchaserInformation/contractingOfficeAgencyID</elementName>
<infoCode>12050</infoCode>
<infoMessage>Contracting Office Agency ID is defaulted to the Agency ID from the User
Profile.</infoMessage>
</info>
</listOfInfoMessages>
</response>
```

9 Specific API details

9.1 Business Services

9.1.1 Data Collection

9.1.1.1 Contracts

9.1.1.1.1 Contract

9.1.1.1.1.1 Service meta-specifications in WSDL

WSDL Parameter	Value
PortType	ContractPortType
Binding	ContractBinding
Soap-binding style	rpc
TargetNameSpace	Contract.wsdl

9.1.1.1.1.2 Contract Service methods

Service Type/Name	Service Description
get	The Contract.get service finds an existing Award or IDV using the PIID information in FPDS and then composes and returns the XML representation of the Award or IDV.
getList	The Contract.getList service finds Award and IDV records, which match the input selection criteria. Matching records are returned in the XML format declared in the Schema.

9.1.1.1.2 get

- The get service retrieves the existing Award or IDV information in FPDS and returns the record in the XML format specified by the schema.
- All users of the system are allowed to access the get service. However, for public users, some fields like TIN etc. are not sent in the return Award or IDV data.
- Contracts in draft mode are retrievable only by the users of the same contracting office and agency, department administrators in the same organization hierarchy.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
contractID	complexType contractIDType	Contract.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
getContractResponse	complexType getContractResponseType	Contract.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
contract	complexType contractType	XML representation of the contract record

Example:

```

<getContractResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <award>
    ...
  </award>
  <outputMessages>
    ...
  </outputMessages>
</getContractResponse>
    
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	<ol style="list-style-type: none"> Contract for the PIID not found. Modification Number required to Uniquely identify the contract Entity specified for the contract does not exists Invalid NAICS code

9.1.1.1.2.1 getList

- The getList service retrieves the existing Award or IDV information that satisfies the specified criteria in the request from the FPDS system.
- 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	User.xsd
contractSearchCriteria	complexType contractSearchCriteriaType	Contract.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
getListContractResponse	complexType getListContractResponseType	Contract.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
listOfContracts	complexType listOfContractsType	List of Contracts XML

Example:

```

<getListContractsResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <listOfContracts>
    <award>
      ...
    </award>
  </listOfContracts>
  <outputMessages>
    ...
  </outputMessages>
</getListContractsResponse>
    
```

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. 2. Too many records retrieved, returning first 500.
ListOfInfoMessages	complexType listOfInfoMessagesType	1. No contracts available for search criteria.

9.1.1.1.3 Award

9.1.1.1.3.1 Service meta-specifications in WSDL

WSDL Parameter	Value
PortType	AwardPortType
Binding	AwardBinding
Soap-binding style	rpc
TargetNameSpace	Award.wSDL

9.1.1.1.3.2 Award Service methods

Service Type/Name	Service Description
create	The Award.create service creates a new Award in FPDS. This service utilizes the validate service to check record validity before inserting into the FPDS database.
createFromTemplate	The Award.createFromTemplate service creates a new Award in FPDS using an existing template.
update	The Award.update service updates an existing Award in FPDS. This service utilizes the validate service to check record validity before updating the FPDS database.

Service Type/Name	Service Description
delete	The Award.delete service deletes an existing Award in FPDS. This service utilizes the exists service to check record existence before marking the record as deleted from the FPDS database. If the document is in DRAFT mode, the document will be physically deleted. If the document is in a FINAL state, the document will be flagged as deleted.
validate	The Award.validate service validates the data for a given Award.
isComplete	The Award.isComplete service does data validation and business rule validation.
approve	Award.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 Award is changed from DRAFT mode to FINAL (approved) mode. If the Award is not complete or a value is not valid, an error is returned.
get	The Award.get service finds an existing Award in FPDS and then composes and returns the XML representation of the Award.
getList	The Award.getList service finds Awards records, which match the input selection criteria. Matching records are returned in the XML format declared in the Schema.
exists	The Award.exists service checks for the existence of a given Award.
correct	The Award.correct service changes an existing Award without creating a modification.
isExistingAwardComplete	The Award.isExistingAwardComplete does the data validation and business rule validation for an already existing Award.

9.1.1.1.3.3 create

- The Award.create method creates a new Award in DRAFT mode in FPDS. The service is authenticated before creating the Award.
- The document will be created if the user has ‘create’ privileges on Award in FPDS.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
AwardXML	complexType awardType	Award.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
createAwardResponse	complexType createAwardResponse	Award.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
Award	complexType awardType	XML representation of the created Award

Example:

```
<createAwardResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <award>
    ...
  </award>
  <outputMessages>
    ...
  </outputMessages>
</createAwardResponse>
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	1. An Award with the PIID already exists 2. Modification Number is required 3. PSC Code is not valid for the product

9.1.1.1.3.4 *update*

- The Award.update method is the API to update and perform modifications to the existing Award in draft mode.
- The Award.update method expects only the Award ID and the required information to uniquely identify the Award information pertaining to the user.
- Any Contracting Officer from the same contracting office is allowed to update the Award.
- Also, Administrators of the agency and department in the preparer’s hierarchy are also allowed to make modifications to the Award.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	FPDS:userAuthenticationKeyType	User.xsd
Award	complexType awardType	Award.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
updateAwardResponse	complexType updateAwardResponse	Award.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
award	complexType awardType	XML representation of the updated Award record

Example:

```
<updateAwardResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
```

```

<award>
  ...
</award>
<outputMessages>
  ...
</outputMessages>
</updateAwardResponse>
    
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	<ol style="list-style-type: none"> Contract for the PIID not found. User not authorized to update this contract. Entity specified for the contract does not exist Invalid NAICS code

9.1.1.1.3.5 *isComplete*

- The isComplete method checks for the completeness of the Award packet.
- It checks for the existence of all mandatory fields in the Award, does data validation, applies business rules validations and returns success or failure back 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
Award	complexType awardType	Award.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
iscompleteAwardResponse	complexType iscompleteAwardResponse Type	Award.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 Award is complete or otherwise

Example:

```

<iscompleteAwardResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <isCompleted>
    true
  </isCompleted>
  <outputMessages>
    
```

...
 </outputMessages>
 </iscompleteAwardResponse>

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	1. <PIID> must not contain special characters 2. Transaction Number <CG2003PR> must be unique within a <PIID> and <modification number> 3. <Contract Date Signed> <05/23/2003> cannot be later than today's date. 4. The sum of Dollars obligated all transactions with the same PIID must be less than or equal to Current Contract Value

9.1.1.1.3.6 *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	User.xsd
AwardID	complexType awardContractIDType	Award.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
approveAwardResponse	complexType approveAwardResponseType	Award.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 Award is approved or not

Example:

```
<approveAwardResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <isApproved>
    true
  </isApproved>
  <outputMessages>
```

```

...
</outputMessages>
</approveAwardResponse>
    
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	1. PIID must not contain special characters 2. Transaction Number must be unique within a PIID and Modification Number 3. Contract Date Signed cannot be later than today's date 4. The sum of the Dollars obligated for all transactions with the same PIID must be less than or equal to the Current Contract Value

9.1.1.1.3.7 *get*

1. The get service retrieves the existing Award information in FPDS and returns the record in the XML format specified by the schema.
2. All users of the system are allowed to access the get service. However, for public users, some fields like TIN are not sent in the return Award data.
3. Awards in DRAFT mode are retrievable only by the users of the same contracting office and agency or department administrators in the same organization hierarchy.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
AwardID	complexType awardContractIDType	Award.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
getAwardResponse	complexType getAwardResponseType	Award.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
award	complexType awardType	XML representation of the updated Award record

Example:

```

<getAwardResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <award>
    ...
  </award>
    
```

```
<outputMessages>
  </outputMessages>
</getAwardResponse>
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	<ol style="list-style-type: none"> 1. PIID must not contain special characters 2. Transaction Number must be unique within a PIID and Modification Number 3. Award not available for the specified PIID and Modification Number values.

9.1.1.1.3.8 *getList*

- The getList service retrieves the existing Award information that satisfies the specified criteria in the request from the FPDS system.
- On successful retrieval, it returns the records in the XML format specified by the schema.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
awardSearchCriteria	complexType awardSearchCriteriaType	Award.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
getListAwardResponse	complexType getListAwardResponseType	Award.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
listOfAwards	complexType listOfAwardsType	List of Awards XML

Example:

```
<getListAwardResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <listOfAwards>
    <award>
      ...
    </award>
    <award>
      ...
    </award>
    ...
  </listOfAwards>
</getListAwardResponse>
```

```

</listOfAwards>
<outputMessages>
...
</outputMessages>
</getListAwardResponse>
    
```

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
ListOfInfoMessages	complexType listOfInfoMessagesType	1. No Awards available for your search criteria

9.1.1.1.3.9 *exists*

- This service checks whether an Award record, as per the criteria in the request, exists in FPDS 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	User.xsd
AwardID	complexType awardContractIDType	Award.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
existsAwardResponse	complexType existsAwardResponseType	Award.wsdl

The output parameters in the response contain the following:

Success Output

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

Example:

```

<existsAwardResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <isExists>
    true
  </isExists>
  <outputMessages>
    
```

```

...
</outputMessages>
</existsAwardResponse>
    
```

Failure Output

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

9.1.1.1.3.10 *correct*

- The Award.correct gets the Award XML
- Award is in the FINAL state in FPDS
- Appropriate authentication is carried out before correcting the Award.

NOTE: Correcting a document signed prior to October 1, 2003 requires the entire XML to be sent.
The correction transaction logs the old data is in the appropriate transaction log.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	userAuthenticationKeyType	User.xsd
Award	complexType awardType	Award.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
correctAwardResponse	complexType correctAwardResponse Type	Award.wsdl

The output parameter in the response contains the following:

Success Output

Parameter Name	Parameter Type	Contents
award	complexType awardType	XML representation of the updated Award record

Example:

```

<correctAwardResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <isCorrected>
    true
  </isCorrected>
  <outputMessages>
    ...
  </outputMessages>
</correctAwardResponse>
    
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	<ol style="list-style-type: none"> 1. Award not found in the FPDS System 2. Transaction Number must be unique within a PIID and Modification Number 3. Contract Date Signed cannot be later than today's date 4. The sum of Dollars obligated all transactions with the same PIID must be less than or equal to Current Contract Value

9.1.1.1.3.11 isExistingAwardComplete

- The Award.isExistingAwardComplete method checks for the completeness of an already saved Award.
- It checks for the existence of all mandatory fields in the Award, does data validation, applies business rules validations and returns success or failure back 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	User.xsd
AwardID	complexType awardContractIDType	Award.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
isExistingAwardCompleteResponse	complexType isExistingAwardCompleteResponse Type	Award.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 existing Award is complete or not

Example:

```

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


Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	<ol style="list-style-type: none"> 1. The PIID must not contain special characters 2. Transaction Number must be unique within a PIID and Modification Number 3. Contract Date Signed cannot be later than today's date 4. The sum of Dollars Obligated all transactions with the same PIID must be less than or equal to Current Contract Value

9.1.1.1.3.12 *createFromTemplate*

- The Award.createFromTemplate method creates a new Award in DRAFT mode in FPDS. The service is authenticated before creating the Award.
- The document will be created by retrieving all the values from the template and
 1. Additional values will be filled from the Award XML input
 2. In case of redundant values, the values from Award template will be overwritten by the values from Award XML input.

Input Parameters

Parameter Name	Parameter Type	Name Space
authenticationKey	FPDS: User Authentication Key Type	User.xsd
createFromTemplateawardrequest	FPDS: creatFromTemplateAwardRequestType	Award.wsdl

Output Parameters

Parameter Name	Parameter Type	Name Space
createFromTemplateAwardResponse	FPDS:creatFromTemplateAwardResponse Type	Award.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
award	complexType awardType	XML representation of the created Award

Example:

```

<createFromTemplateAwardResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <award>
    ...
  </award>
  <outputMessages>
    ...
  </outputMessages>
</createFromTemplateAwardResponse>
    
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	<ol style="list-style-type: none"> 1. An Award with the PIID Already exists 2. Modification Number is required 3. PSC Code is not valid for the product

(Note: Only a sample list is provided: Please refer to FPDS Error Listings for complete list of errors)

9.1.1.1.3.13 delete

- The delete service deletes the existing Award information in FPDS and returns the Boolean value to indicate whether the document is successfully deleted or not.
- Not all government users have ‘delete’ privileges.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
AwardID	complexType awardContractIDType	Award.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
deleteAwardResponse	complexType deleteAwardResponse Type	Award.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 Award is deleted or not

Example:

```

<deleteAwardResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <isDeleted>true</isDeleted >
  <outputMessages>
  </outputMessages>
</delete:AwardResponse>
    
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	<ol style="list-style-type: none"> 1. The current base Award has modified documents and cannot be deleted 2. This entity modification has subsequent modifications and cannot be deleted.

		3. A record that is in final status and has the "Reason for Modification" as "Change PIID" cannot be deleted. Please contact your system administrator for further assistance.
--	--	--

9.1.1.1.4 IDV

9.1.1.1.4.1 Service meta-specifications in WSDL

WSDL Parameter	Value
PortType	IDVPortType
Binding	IDVBinding
Soap-binding style	rpc
TargetNameSpace	IDV.wsdl

9.1.1.1.4.2 IDV Service methods

Service Type/Name	Service Description
create	The IDV.create service creates a new IDV in FPDS. This service utilizes the validation services to check record validity before insertion into the FPDS database.
createFromTemplate	The IDV.createFromTemplate service creates a new IDV in FPDS using an existing template.
update	The IDV.update service updates an existing IDV in FPDS. This service utilizes the validation services to check record validity before updating the FPDS database.
delete	The IDV.delete service deletes an existing IDV in FPDS. This service utilizes the validation services to check record existence before marking the record as deleted from the FPDS database. Deletion does not delete the record physically from the table, but flags it as a deleted record. Only draft records in the suspense table are physically deleted.
isComplete	The IDV.isComplete service checks for data correctness and business rule validity.
approve	IDV approves the document after check for completeness (by using isComplete service). If no error is returned, the status of the given IDV is changed from the Suspense (Draft) mode to the Approved (Final) mode. If the IDV is not complete or a value is not valid, an error is returned.
Get	The IDV.get service finds an existing IDV in FPDS and then composes and returns the XML representation of the IDV.
getList	The IDV.getList service finds IDVs records, which match the input selection criteria. Matching records are returned in XML format declared in the Schema.
exists	The IDV.exists service checks for the existence of a given IDV code.
correct	The IDV.correct method makes changes to the IDV record without requiring the user to create modifications. Document must be in FINAL state for this action to be valid.
isExistingIDVComplete	The IDV.isExistingIDVComplete service checks for data correctness and business rule validity of an existing IDV.

9.1.1.1.4.3 Create

- The IDV.create method creates a new IDV in DRAFT mode in FPDS.
- User must have ‘create’ privileges on IDV object.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
IDV	complexType IDVType	IDV.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
createIDVResponse	complexType createIDVResponse Type	IDV.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
IDV	complexType IDVType	XML representation of the updated IDV record

Example:

```

<createIDVResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <IDV>
    ...
  </IDV>
  <outputMessages>
    ...
  </outputMessages>
</createIDVResponse>
    
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	<ol style="list-style-type: none"> 1. PIID contains special characters 2. Transaction Number must be unique within a PIID and Modification Number 3. Contract Date Signed cannot be later than today’s date.

9.1.1.1.4.4 update

- The IDV.update method updates and performs modifications to the existing IDV in DRAFT mode and corrects some errors that do not require journaling.
- Any Contracting Officer from the same contracting office is allowed to update the IDV.
- Also, Administrators of the agency and department in the same hierarchy as the contracting office are also allowed to make modifications to the IDV.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
IDV	complexType IDVType	IDV.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
updateIDVResponse	complexType updateIDVResponse Type	IDV.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
IDV	complexType IDVType	XML representation of the updated IDV record

Example:

```

<updateIDVResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <IDV>
    ...
  </IDV>
  <outputMessages>
    ...
  </outputMessages>
</updateIDVResponse>
    
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	<ol style="list-style-type: none"> 1. PIID must not contain special characters 2. Transaction Number must be unique within a PIID and Modification Number 3. Contract Date Signed cannot be later than today’s date 4. The sum of Dollars Obligated all transactions with the same PIID must be less than or equal to Current Contract Value

9.1.1.1.4.5 *isComplete*

- The isComplete method is the API to check for the completeness of the IDV data
- It checks for the existence of all mandatory fields, data validity and business rule validity, and return success or failure back to the user.
- In case of failure, the error information would list all the incomplete fields for which a value is mandatory.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	userAuthenticationKeyType	User.xsd
IDV	complexType IDVType	IDV.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
isCompleteIDVResponse	complexType isCompleteIDVResponse Type	IDV.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
isCompleteOutput	xsd:boolean	True or false, indicating whether the IDV is complete

Example:

```

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

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	<ol style="list-style-type: none"> 1. PIID must not contain special characters 2. Transaction Number must be unique within a PIID and Modification Number 3. Contract Date Signed cannot be later than today’s date 4. The amount obligated for all transactions with the same PIID must be less than or equal to the Current Contract Value

9.1.1.1.4.6 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	User.xsd

IDVID	complexType contractIDType	IDV.xsd
-------	----------------------------	---------

Output Parameters

Parameter Name	Parameter Type	Name Space
approveIDVResponse	complexType approveIDVResponse Type	IDV.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
isApproved	xsd:boolean	True or false, mentioning success or failure of approval

Example:

```

<approveIDVResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <isApproved>
    true
  </isApproved>
  <outputMessages>
    ...
  </outputMessages>
</approveIDVResponse>
    
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	<ol style="list-style-type: none"> 1. PIID must not contain special characters 2. Transaction Number must be unique within a PIID and Modification Number 3. Contract Date Signed cannot be later than today's date 4. The sum of Dollars Obligated for all transactions with the same PIID must be less than or equal to the Current Contract Value

9.1.1.1.4.7 get

- The get service retrieves the existing IDV information in FPDS and returns the record in the XML format specified by the schema.
- The get 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	User.xsd
IDVID	complexType contractIDType	IDV.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
----------------	----------------	------------

getIDVResponse	complexType getIDVResponse Type	IDV.wsdl
----------------	---------------------------------	----------

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
IDV	complexType IDVType	XML representation of the IDV record

Example:

```

<getIDVResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <IDV>
    ...
  </IDV>
  <outputMessages>
  </outputMessages>
</getIDVResponse>
    
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	<ol style="list-style-type: none"> 1. PIID must not contain special characters 2. Transaction Number must be unique within a PIID and Modification Number 3. IDV for the specified PIID and Modification Number does not exist.

9.1.1.1.4.8 *getList*

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

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	userAuthenticationKeyType	User.xsd
IDVSearchCriteria	complexType IDVSearchCriteriaType	IDV.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
getListIDVResponse	complexType getListIDVResponse Type	IDV.wsdl

Success Output

Parameter Name	Parameter Type	Contents
listOfIDVs	complexType	List of IDVs in XML format

	listOfIDVsType	
--	----------------	--

Example:

```

<getListIDVResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <listOfIDVs>
    <IDV>
      ...
    </IDV>
    <IDV>
      ...
    </IDV>
    ...
  </listOfIDVs>
  <outputMessages>
    ...
  </outputMessages>
</getListIDVResponse>
    
```

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
ListOfInfoMessages	complexType listOfInfoMessagesType	1. No IDVs available for your search criteria

9.1.1.1.4.9 *exists*

- The exists service returns true or false value, representing whether the IDV record exists in the database.
- Service implementation checks whether an IDV record, as per the criteria in the request, exists in FPDS and returns success or failure response.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	userAuthenticationKeyType	User.xsd
IDVID	complexType contractIDType	IDV.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
existsIDVResponse	complexType existsIDVResponse Type	IDV.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
isExists	xsd:boolean	True if record exists, or false otherwise.

Example:

```

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

9.1.1.1.4.10 correct

- The IDV.correct gets the IDV XML containing ID and elements that need to be corrected.
- This service updates the data sent in the request, keeping other data as is.
- The correction transaction that was carried out is also journalled in the appropriate transaction log.

NOTE: Correcting a document signed prior to October 1, 2003 requires the entire XML to be sent.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	userAuthenticationKeyType	User.xsd
IDV	complexType IDVType	IDV.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
correctIDVResponse	complexType correctIDVResponse Type	IDV.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
IDVXML	complexType IDVType	XML representation of the corrected IDV record

Example:

```

<correctIDVResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <isCorrected>
    true
  </isCorrected>
  <outputMessages>
    ...
  </outputMessages>
    
```

```

...
</outputMessages>
</correctIDVResponse>
    
```

9.1.1.1.4.11 *isExistingIDVComplete*

- The IDV.isExistingIDVComplete method checks for the completeness of an already saved IDV.
- It checks for the existence of all mandatory fields in the IDV, does data validation (by using validate service), applies business rules validations and returns success or failure back 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	User.xsd
IDVID	complexType contractIDType	IDV.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
isExistingIDVCompleteResponse	complexType isExistingIDVCompleteResponse Type	IDV.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 existing IDV is complete or not

Example:

```

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

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	<ol style="list-style-type: none"> 1. PIID must not contain special characters 2. Transaction Number must be unique within a PIID and Modification Number 3. Contract Date Signed cannot be later than today's date

Parameter Name	Parameter Type	Sample Error Contents
		4. The amount obligated for all transactions with the same PIID must be less than or equal to the Current Contract Value

9.1.1.1.4.12 *createFromTemplate*

- The IDV.createFromTemplate method creates a new IDV in DRAFT mode in FPDS. The service is authenticated before creating the IDV.
- The document will be created by retrieving all the values from the template and
 1. Additional values will be filled from the IDV XML input
 2. In case of redundant values, the values from IDV template will be overwritten by the values from IDV XML input.

Input Parameters

Parameter Name	Parameter Type	Name Space
authenticationKey	FPDS: User Authentication Key Type	User.xsd
createFromTemplateIDVRequestType	FPDS: createFromTemplateIDVRequestType	IDV.wsdl

Output Parameters

Parameter Name	Parameter Type	Name Space
createFromTemplateIDVResponse	FPDS:createFromTemplateIDVResponse Type	IDV.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
IDV	complexType IDVType	XML representation of the updated IDV record

Example:

```

<createFromTemplateIDVResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <IDV>
    ...
  </IDV>
  <outputMessages>
    ...
  </outputMessages>
</createFromTemplateIDVResponse>
    
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	1. PIID should not contain special characters 2. Transaction Number should be unique within a PIID and Modification Number

		3. Contract Date Signed cannot be later than today's date
--	--	---

9.1.1.1.4.13 delete

- The delete service deletes the existing IDV information in FPDS and returns the Boolean value to indicate whether the document is successfully deleted or not.
- Not all government users have 'delete' privileges.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
IDVID	complexType IDVIDType	IDV.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
deleteIDVResponse	complexType deleteIDVResponse Type	IDV.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 IDV is deleted or not

Example:

```
<deleteIDVResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <isDeleted>>true</isDeleted >
  <outputMessages>
  </outputMessages>
</delete:IDVResponse>
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	<ol style="list-style-type: none"> 1. This IDV is referenced by one or more Awards and cannot be deleted 2. IDV type FSS can only be created/updated/validated/approved/deleted/corrected by GSA/FAS or VA 3. This entity modification has subsequent modifications and cannot be deleted 4. A record that is in final status and has the "Reason for Modification" as "Change PIID" cannot be deleted. Please contact your system administrator

		for further assistance.
--	--	-------------------------

(Note: Only a sample list is provided: Please refer to FPDS Error Listings for complete list of errors)

9.1.1.1.5 OtherTransactionAward

9.1.1.1.5.1 Service meta-specifications in WSDL

WSDL Parameter	Value
PortType	OtherTransactionAwardPortType
Binding	OtherTransactionAwardBinding
Soap-binding style	rpc
TargetNameSpace	OtherTransactionAward.wsdl

9.1.1.1.5.2 OtherTransactionAward Service methods

Service Type/Name	Service Description
create	The OtherTransactionAward.create service creates a new OtherTransactionAward in FPDS. This service utilizes the validate service to check record validity before inserting into the FPDS database.
update	The OtherTransactionAward.update service updates an existing OtherTransactionAward in FPDS. This service utilizes the validate service to check record validity before updating the FPDS database.
delete	The OtherTransactionAward.delete service deletes an existing OtherTransactionAward in FPDS. This service utilizes the exists service to check record existence before marking the record as deleted from the FPDS database. If the document is in DRAFT mode, the document will be physically deleted. If the document is in a FINAL state, the document will be flagged as deleted.
isComplete	The OtherTransactionAward.isComplete service does data validation and business rule validation.
approve	OtherTransactionAward.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 OtherTransactionAward is changed from DRAFT mode to FINAL (approved) mode. If the OtherTransactionAward is not complete or a value is not valid, an error is returned.
get	The OtherTransactionAward.get service finds an existing OtherTransactionAward in FPDS and then composes and returns the XML representation of the OtherTransactionAward.
getList	The OtherTransactionAward.getList service finds OtherTransactionAwards records, which match the input selection criteria. Matching records are returned in the XML format declared in the Schema.

Service Type/Name	Service Description
exists	The OtherTransactionAward.exists service checks for the existence of a given OtherTransactionAward.
correct	The OtherTransactionAward.correct service changes an existing OtherTransactionAward without creating a modification.

9.1.1.1.5.3 create

- The OtherTransactionAward.create method creates a new OtherTransactionAward in DRAFT mode in FPDS. The service is authenticated before creating the OtherTransactionAward.
- The document will be created if the user has ‘create’ privileges on OtherTransactionAward in FPDS.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
OtherTransactionAwardXML	complexType OtherTransactionAwardType	OtherTransactionAward.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
createOtherTransactionAwardResponse	complexType createOtherTransactionAwardResponse	OtherTransactionAward.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
OtherTransactionAward	complexType OtherTransactionAwardType	XML representation of the created OtherTransactionAward

Example:

```
<createOtherTransactionAwardResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <OtherTransactionAward>
    ...
  </OtherTransactionAward>
  <outputMessages>
    ...
  </outputMessages>
</createOtherTransactionAwardResponse>
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	<ol style="list-style-type: none"> 1. An OtherTransactionAward with the PIID already exists 2. Modification Number is required 3. PSC Code is not valid for the product

9.1.1.1.5.4 update

- The OtherTransactionAward.update method is the API to update and perform modifications to the existing OtherTransactionAward in draft mode.
- The OtherTransactionAward.update method expects only the OtherTransactionAward ID and the required information to uniquely identify the OtherTransactionAward information pertaining to the user.
- The user should have OtherTransactionAward ‘update’ privilege to do this operation.
- Any Contracting Officer from the same contracting office is allowed to update the OtherTransactionAward.
- Also, Administrators of the agency and department in the preparer’s hierarchy are also allowed to make modifications to the OtherTransactionAward.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	FPDS:userAuthenticationKeyType	User.xsd
OtherTransactionAward	complexType OtherTransactionAwardType	OtherTransactionAward.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
updateOtherTransactionAwardResponse	complexType updateOtherTransactionAwardResponse	OtherTransactionAward.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
OtherTransactionAward	complexType OtherTransactionAwardType	XML representation of the updated OtherTransactionAward record

Example:

```

<updateOtherTransactionAwardResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <OtherTransactionAward>
    ...
  </OtherTransactionAward>
  <outputMessages>
    ...
  </outputMessages>
</updateOtherTransactionAwardResponse>
    
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	<ol style="list-style-type: none"> 1. Contract for the PIID not found 2. User not authorized to update this contract 3. Entity specified for the contract does not exists 4. Invalid NAICS code

9.1.1.1.5.5 *isComplete*

- The isComplete method checks for the completeness of the OtherTransactionAward data packet.
- It checks for the existence of all mandatory fields in the OtherTransactionAward, does data validation, applies business rules validations and returns success or failure back 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
OtherTransactionAward	complexType OtherTransactionAwardType	OtherTransactionAward.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
isCompleteOtherTransactionAwardResponse	complexType isCompleteOtherTransactionAwardResponse Type	OtherTransactionAward.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 OtherTransactionAward is complete or otherwise

Example:

```

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

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	<ol style="list-style-type: none"> 1. <PIID> must not contain special characters 2. Transaction Number <CG2003PR> must be unique within a <PIID> and <modification number> 3. <Contract Date Signed> <05/23/2003> cannot be later than today's date 4. The sum of Dollars obligated all transactions with the same PIID must be less than or equal to Current Contract Value

9.1.1.1.5.6 *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	User.xsd
OtherTransactionAwardID	complexType OtherTransactionAwardContractIDType	OtherTransactionAward.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
approveOtherTransactionAwardResponse	complexType approveOtherTransactionAwardResponse Type	OtherTransactionAward.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 OtherTransactionAward is approved or not

Example:

```

<approveOtherTransactionAwardResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <isApproved>
    true
  </isApproved>
  <outputMessages>
    ...
  </outputMessages>
</approveOtherTransactionAwardResponse>
    
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	<ol style="list-style-type: none"> 1. PIID must not contain special characters 2. Transaction Number must be unique within a PIID and Modification Number 3. Contract Date Signed cannot be later than today's date 4. The sum of the Dollars obligated for all transactions with the same PIID must be less than or equal to the Current Contract Value

9.1.1.1.5.7 *get*

1. The get service retrieves the existing OtherTransactionAward information in FPDS and returns the record in the XML format specified by the schema.
2. All users of the system are allowed to access the get service.
3. OtherTransactionAwards in DRAFT mode are retrievable only by the users of the same contracting office and agency or department administrators in the same organization hierarchy.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
OtherTransactionAwardID	complexType OtherTransactionAwardContractIDType	OtherTransactionAward.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
getOtherTransactionAwardResponse	complexType getOtherTransactionAwardResponse Type	OtherTransactionAward.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
OtherTransactionAward	complexType OtherTransactionAwardType	XML representation of the updated OtherTransactionAward record

Example:

```

<getOtherTransactionAwardResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <OtherTransactionAward>
    ...
  </OtherTransactionAward>
  <outputMessages>
  </outputMessages>
</getOtherTransactionAwardResponse>
    
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	<ol style="list-style-type: none"> 1. PIID must not contain special characters 2. Transaction Number must be unique within a PIID and Modification Number 3. OtherTransactionAward not available for the specified PIID and Modification Number values

9.1.1.1.5.8 *getList*

- The getList service retrieves the existing OtherTransactionAward information that satisfies the specified criteria in the request from the FPDS system.
- On successful retrieval, it returns the records in the XML format specified by the schema.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
OtherTransactionAwardSearchCriteria	complexType OtherTransactionAwardSearchCriteriaType	OtherTransactionAward.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
getListOtherTransactionAwardResponse	complexType getListOtherTransactionAwardResponse Type	OtherTransactionAward.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
listOfOtherTransactionAwards	complexType listOfOtherTransactionAwardsType	List of OtherTransactionAwards XML

Example:

```

<getListOtherTransactionAwardResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <listOfOtherTransactionAwards>
    <OtherTransactionAward>
      ...
    </OtherTransactionAward>
    <OtherTransactionAward>
      ...
    </OtherTransactionAward>
    ...
  </listOfOtherTransactionAwards>
  <outputMessages>
    ...
  </outputMessages>
</getListOtherTransactionAwardResponse>
    
```

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
ListOfInfoMessages	complexType	1. No OtherTransactionAwards available for

	listOfInfoMessagesType	your search criteria
--	------------------------	----------------------

9.1.1.1.5.9 *exists*

- This service checks whether an OtherTransactionAward record, as per the criteria in the request, exists in FPDS 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	User.xsd
OtherTransactionAwardID	complexType OtherTransactionAwardContractIDType	OtherTransactionAward.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
existsOtherTransactionAwardResponse	complexType existsOtherTransactionAwardResponse Type	OtherTransactionAward.wsdl

The output parameters in the response contain the following:

Success Output

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

Example:

```

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

Failure Output

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

9.1.1.1.5.10 *correct*

- The OtherTransactionAward.correct gets the OtherTransactionAward XML
- OtherTransactionAward is in the FINAL state in FPDS

- Appropriate authentication is carried out before correcting the OtherTransactionAward. The correction transaction logs the old data in the appropriate transaction log.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	userAuthenticationKeyType	User.xsd
OtherTransactionAward	complexType OtherTransactionAwardType	OtherTransactionAward.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
correctOtherTransactionAwardResponse	complexType correctOtherTransactionAwardResponse Type	OtherTransactionAward.wsdl

The output parameter in the response contains the following:

Success Output

Parameter Name	Parameter Type	Contents
OtherTransactionAward	complexType OtherTransactionAwardType	XML representation of the updated OtherTransactionAward record

Example:

```

<correctOtherTransactionAwardResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <isCorrected>
    true
  </isCorrected>
  <outputMessages>
    ...
  </outputMessages>
</correctOtherTransactionAwardResponse>
    
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	<ol style="list-style-type: none"> 1. OtherTransactionAward not found in the FPDS System 2. Transaction Number must be unique within a PIID and Modification Number 3. Contract Date Signed cannot be later than today's date 4. The sum of Dollars obligated all transactions with the same PIID must be less than or equal to Current Contract Value

9.1.1.1.5.11 delete

- The delete service deletes the existing OtherTransactionAward information in FPDS and returns the Boolean value to indicate whether the document is successfully deleted or not.
- Not all government users have ‘delete’ privileges.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
OtherTransactionAwardID	complexType OtherTransactionAwardIDType	OtherTransactionAward.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
deleteOtherTransactionAwardResponse	complexType deleteOtherTransactionAwardResponse Type	OtherTransactionAward.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 OtherTransactionAward is deleted or not

Example:

```

<deleteOtherTransactionAwardResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <isDeleted>true</isDeleted>
  <outputMessages>
  </outputMessages>
</deleteOtherTransactionAwardResponse>
    
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	<ol style="list-style-type: none"> 1. The current base Award has modified documents and cannot be deleted 2. This entity modification has subsequent modifications and cannot be deleted 3. A record that is in final status and has the "Reason for Modification" as "Change PIID" cannot be deleted. Please contact your system administrator for further assistance.

9.1.1.1.6 OtherTransactionIDV

9.1.1.1.6.1 Service meta-specifications in WSDL

WSDL Parameter	Value
PortType	OtherTransactionIDVPortType
Binding	OtherTransactionIDVBinding
Soap-binding style	rpc
TargetNameSpace	OtherTransactionIDV.wsdl

9.1.1.1.6.2 *OtherTransactionIDV Service methods*

Service Type/Name	Service Description
create	The OtherTransactionIDV.create service creates a new OtherTransactionIDV in FPDS. This service utilizes the validation services to check record validity before insertion into the FPDS database.
update	The OtherTransactionIDV.update service updates an existing OtherTransactionIDV in FPDS. This service utilizes the validation services to check record validity before updating the FPDS database.
delete	The OtherTransactionIDV.delete service deletes an existing OtherTransactionIDV in FPDS. This service utilizes the validation services to check record existence before marking the record as deleted from the FPDS database. Deletion does not delete the record physically from the table, but flags it as a deleted record. Only draft records in the suspense table are physically deleted.
isComplete	The OtherTransactionIDV.isComplete service checks for data correctness and business rule validity.
approve	OtherTransactionIDV.approve approves the document after check for completeness (by using isComplete service). If no error is returned, the status of the given OtherTransactionIDV is changed from the Suspense (Draft) mode to the Approved (Final) mode. If the OtherTransactionIDV is not complete or a value is not valid, an error is returned.
get	The OtherTransactionIDV.get service finds an existing OtherTransactionIDV in FPDS and then composes and returns the XML representation of the OtherTransactionIDV.
getList	The OtherTransactionIDV.getList service finds OtherTransactionIDV records, which match the input selection criteria. Matching records are returned in XML format declared in the Schema.
exists	The OtherTransactionIDV.exists service checks for the existence of a given OtherTransactionIDV code.
correct	The OtherTransactionIDV.correct method makes changes to the OtherTransactionIDV record without requiring the user to create modifications. Document must be in FINAL state for this action to be valid.

9.1.1.1.6.3 *Create*

- The OtherTransactionIDV.create method creates a new OtherTransactionIDV in DRAFT mode in FPDS.
- User must have ‘create’ privileges on OtherTransactionIDV object.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
OtherTransactionIDV	complexType OtherTransactionIDVType	OtherTransactionIDV.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
createOtherTransactionIDVResponse	complexType createOtherTransactionIDVResponse Type	OtherTransactionIDV.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
OtherTransactionIDV	complexType OtherTransactionIDVType	XML representation of the updated OtherTransactionIDV record

Example:

```

<createOtherTransactionIDVResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <OtherTransactionIDV>
    ...
  </OtherTransactionIDV>
  <outputMessages>
    ...
  </outputMessages>
</createOtherTransactionIDVResponse>
    
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	1. PIID contains special characters 2. Transaction Number must be unique within a PIID and Modification Number 3. Contract Date Signed cannot be later than today’s date

9.1.1.1.6.4 *update*

- The OtherTransactionIDV.update method updates and performs modifications to the existing OtherTransactionIDV in DRAFT mode and corrects some errors that do not require journaling.
- The user should have OtherTransactionIDV ‘update’ privilege to do this operation.

- Any Contracting Officer from the same contracting office is allowed to update the OtherTransactionIDV.
- Also, Administrators of the agency and department in the same hierarchy as the contracting office are also allowed to make modifications to the OtherTransactionIDV.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
OtherTransactionIDV	complexType OtherTransactionIDVType	OtherTransactionIDV.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
updateOtherTransactionIDVResponse	complexType updateOtherTransactionIDVResponse Type	OtherTransactionIDV.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
OtherTransactionIDV	complexType OtherTransactionIDVType	XML representation of the updated OtherTransactionIDV record

Example:

```

<updateOtherTransactionIDVResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <OtherTransactionIDV>
    ...
  </OtherTransactionIDV>
  <outputMessages>
    ...
  </outputMessages>
</updateOtherTransactionIDVResponse>
    
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	<ol style="list-style-type: none"> 1. PIID must not contain special characters 2. Transaction Number must be unique within a PIID and Modification Number 3. Contract Date Signed cannot be later than today’s date 4. The sum of Dollars Obligated all transactions with the same PIID must be less than or equal to Current Contract Value

9.1.1.1.6.5 *isComplete*

- The isComplete method is the API to check for the completeness of the OtherTransactionIDV data.

- It checks for the existence of all mandatory fields, data validity and business rule validity, and return success or failure back to the user.
- In case of failure, the error information would list all the incomplete fields for which a value is mandatory.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	userAuthenticationKeyType	User.xsd
OtherTransactionIDV	complexType OtherTransactionIDVType	OtherTransactionIDV.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
isCompleteOtherTransactionIDVResponse	complexType isCompleteOtherTransactionIDVResponse Type	OtherTransactionIDV.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
isCompleteOutput	xsd:boolean	True or false, indicating whether the OtherTransactionIDV is complete

Example:

```

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

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	<ol style="list-style-type: none"> 1. PIID must not contain special characters 2. Transaction Number must be unique within a PIID and Modification Number 3. Contract Date Signed cannot be later than today's date 4. The amount obligated for all transactions with the same PIID must be less than or equal to the Current Contract Value

9.1.1.1.6.6 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	User.xsd
OtherTransactionIDVID	complexType contractIDType	OtherTransactionIDV.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
approveOtherTransactionIDVResponse	complexType approveOtherTransactionIDVResponse Type	OtherTransactionIDV.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
isApproved	xsd:boolean	True or false, mentioning success or failure of approval

Example:

```

<approveOtherTransactionIDVResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <isApproved>
    true
  </isApproved>
  <outputMessages>
    ...
  </outputMessages>
</approveOtherTransactionIDVResponse>
    
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	<ol style="list-style-type: none"> 1. PIID must not contain special characters 2. Transaction Number must be unique within a PIID and Modification Number 3. Contract Date Signed cannot be later than today’s date 4. The sum of Dollars Obligated for all transactions with the same PIID must be less than or equal to the Current Contract Value

9.1.1.1.6.7 *get*

- The get service retrieves the existing OtherTransactionIDV information in FPDS and returns the record in the XML format specified by the schema.

- The get 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	User.xsd
OtherTransactionIDVID	complexType contractIDType	OtherTransactionIDV.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
getOtherTransactionIDVResponse	complexType getOtherTransactionIDVResponse Type	OtherTransactionIDV.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
OtherTransactionIDV	complexType OtherTransactionIDVType	XML representation of the OtherTransactionIDV record

Example:

```

<getOtherTransactionIDVResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <OtherTransactionIDV>
    ...
  </OtherTransactionIDV>
  <outputMessages>
  </outputMessages>
</getOtherTransactionIDVResponse>
    
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	<ol style="list-style-type: none"> 1. PIID must not contain special characters 2. Transaction Number must be unique within a PIID and Modification Number 3. OtherTransactionIDV for the specified PIID and Modification Number does not exist.

9.1.1.1.6.8 getList

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

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	userAuthenticationKeyType	User.xsd

OtherTransactionIDVSearchCriteria	complexType OtherTransactionIDVSearchCriteriaType	OtherTransactionIDV.xsd
-----------------------------------	--	-------------------------

Output Parameters

Parameter Name	Parameter Type	Name Space
getListOtherTransactionIDVResponse	complexType getListOtherTransactionIDVResponse Type	OtherTransactionIDV.wsdl

Success Output

Parameter Name	Parameter Type	Contents
listOfOtherTransactionIDVs	complexType listOfOtherTransactionIDVsType	List of OtherTransactionIDVs in XML format

Example:

```

<getListOtherTransactionIDVResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <listOfOtherTransactionIDVs>
    <OtherTransactionIDV>
      ...
    </OtherTransactionIDV>
    <OtherTransactionIDV>
      ...
    </OtherTransactionIDV>
    ...
  </listOfOtherTransactionIDVs>
  <outputMessages>
    ...
  </outputMessages>
</getListOtherTransactionIDVResponse>
    
```

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
ListOfInfoMessages	complexType listOfInfoMessagesType	1. No OtherTransactionIDVs available for your search criteria

9.1.1.1.6.9 *exists*

- The exists service returns true or false value, representing whether the OtherTransactionIDV record exists in the database.
- Service implementation checks whether an OtherTransactionIDV record, as per the criteria in the request, exists in FPDS and returns success or failure response.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	userAuthenticationKeyType	User.xsd
OtherTransactionIDVID	complexType contractIDType	OtherTransactionIDV.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
existsOtherTransactionIDVResponse	complexType existsOtherTransactionIDVResponse Type	OtherTransactionIDV.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
isExists	xsd:boolean	True if record exists, or false otherwise.

Example:

```

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

9.1.1.1.6.10 correct

- The OtherTransactionIDV.correct gets the OtherTransactionIDV XML containing ID and elements that need to be corrected.
- This service updates the data sent in the request, keeping other data as-is.
- The correction transaction that was carried out is also journalled in the appropriate transaction log.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	userAuthenticationKeyType	User.xsd
OtherTransactionIDV	complexType OtherTransactionIDVType	OtherTransactionIDV.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
correctOtherTransactionIDVResponse	complexType correctOtherTransactionIDVResponse Type	OtherTransactionIDV.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
OtherTransactionIDVXML	complexType OtherTransactionIDVType	XML representation of the corrected OtherTransactionIDV record

Example:

```
<correctOtherTransactionIDVResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <isCorrected>
    true
  </isCorrected>
  <outputMessages>
    ...
  </outputMessages>
</correctOtherTransactionIDVResponse>
```

9.1.1.1.6.11 delete

- The delete service deletes the existing OtherTransactionIDV information in FPDS and returns the Boolean value to indicate whether the document is successfully deleted or not.
- Not all government users have ‘delete’ privileges.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
OtherTransactionIDVID	complexType OtherTransactionIDVIDType	OtherTransactionIDV.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
deleteOtherTransactionIDVResponse	complexType deleteOtherTransactionIDVResponse Type	OtherTransactionIDV.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 OtherTransactionIDV is deleted or not

Example:

```
<deleteOtherTransactionIDVResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <isDeleted>true</isDeleted>
```



```
<outputMessages>
</outputMessages>
</deleteOtherTransactionIDVResponse>
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	<ol style="list-style-type: none"> 1. The current base Award has modified documents and cannot be deleted 2. This entity modification has subsequent modifications and cannot be deleted 3. A record that is in final status and has the "Reason for Modification" as "Change PIID" cannot be deleted. Please contact your system administrator for further assistance.

9.1.1.1.7 Close

9.1.1.1.7.1 Service meta-specifications in WSDL

WSDL Parameter	Value
PortType	ClosePortType
Binding	CloseBinding
Soap-binding style	rpc
TargetNameSpace	Close.wsdl

9.1.1.1.7.2 Close Service methods

Service Type/Name	Service Description
close	The Close.close service closes the contract family.

9.1.1.1.7.3 close

- This service closes the document, utilizing the provided as input in close 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.
- The document will be closed if the user has ‘close’ privileges.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
closeRequest	complexType closeRequestType	Close.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
closeResponse	complexType closeResponseType	Close.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
isClosed	xsd:Boolean	True or false, representing whether the Award is closed or not
closedDate	xsd:DateType	Date when the contract is closed

Example:

```

<closeResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <isClosed>true</isClosed>
  <closedDate>10/01/2017</closedDate>
  <outputMessages>
    ...
  </outputMessages>
</closeResponse>
    
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	1. Contract can not be closed

9.1.1.1.8 ContractSummary

9.1.1.1.8.1 Service meta-specifications in WSDL

WSDL Parameter	Value
PortType	ContractSummaryPortType
Binding	ContractSummaryBinding
Soap-binding style	rpc
TargetNameSpace	ContractSummary.wsdl

9.1.1.1.8.2 ContractSummary Service methods

Service Type/Name	Service Description
getList	The ContractSummary.getList service retrieves contract information in summary format (actions including modifications and novation)
getListWithHistory	The ContractSummary.getListWithHistory service retrieves contract information in summary format (Including sum of obligation amounts, novation, NAICS Codes, funding agencies and place of performance) for contracts worth >= 25k or <= -25k.

9.1.1.1.8.3 getList

- The getList service retrieves the existing contract information that satisfies the specified criteria in the request from the FPDS system.
- On successful retrieval, it returns the records in the XML format specified by the schema.

Input Parameters

Parameter Name	Parameter Type	Name Space
authenticationKey	userAuthenticationKeyType	User.xsd
contractSummaryDataSearchCriteria	contractSummaryDataSearchCriteriaType	ContractSummary.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
getListContractSummaryResponse	complexType getListContractSummaryResponseType	ContractSummary.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
getListContractSummaryResponse	complexType getListContractSummaryResponseType	List of Contract Summaries in XML format

Example:

Success:

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
  <soapenv:Body>
    <ns1:getListResponse soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:ns1="urn:FPDS.BusinessServices.DataCollection.contracts.ContractSummary"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
      <ns1:getListContractSummaryResponse xmlns:ns1="https://www.fpds.gov/FPDS">
        <ns1:requestNumber>2849461899</ns1:requestNumber>
        <ns1:confirmationNumber>2139362233</ns1:confirmationNumber>
        <ns1:outputMessages>
          <ns1:ListOfErrors/>
          <ns1:ListOfWarnings/>
          <ns1:ListOfInfoMessages/>
        </ns1:outputMessages>
        <ns1:ListOfContractSummaryData>
          <ns1:count>
            <ns1:total>25</ns1:total>
            <ns1:fetched>25</ns1:fetched>
          </ns1:count>
          <ns1:contractSummaryData>
            <ns1:awardOrIDV>AWARD</ns1:awardOrIDV>
            <ns1:agencyID>4740</ns1:agencyID>
            <ns1:PIID>GS09P06KSC3075</ns1:PIID>
            <ns1:contractSummaryDetails xsi:nil="true"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"/>
            <ns1:referencedIDVAgencyID/>
            <ns1:referencedIDVPIID/>
          </ns1:contractSummaryData>
        </ns1:ListOfContractSummaryData>
      </ns1:getListContractSummaryResponse>
    </ns1:getListResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

```

        <ns1:lastModifiedDate>2007-09-16 20:08:30.0</ns1:lastModifiedDate>
    </ns1:contractSummaryData>
    <ns1:contractSummaryData>
        <ns1:awardOrIDV>AWARD</ns1:awardOrIDV>
        <ns1:agencyID>8000</ns1:agencyID>
        <ns1:PIID>ARC00200309DNAS203144</ns1:PIID>
        <ns1:contractSummaryDetails xsi:nil="true"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"/>
        <ns1:referencedIDVAgencyID/>
        <ns1:referencedIDVPIID/>
        <ns1:lastModifiedDate>2007-09-16 20:13:15.0</ns1:lastModifiedDate>
    </ns1:contractSummaryData>
</ns1:listOfContractSummariesData>
</ns1:getListContractSummaryResponse>
</ns1:getListResponse>
</soapenv:Body>
</soapenv:Envelope>
    
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	1. The date entered is invalid

9.1.1.1.8.4 *getListWithHistory**

- The getListWithHistory service retrieves the existing contract information that satisfies the specified criteria in the request from the FPDS system. The response includes elements like Total Obligated Amount, Date Signed, Place of Performance, Vendor Information (a history of entities and associated Place of Performance).
- On successful retrieval, it returns the records in the XML format specified by the schema.
**Note: This service is no longer active. The service remains in this document for historical tracking purposes.*

Input Parameters

Parameter Name	Parameter Type	Name Space
authenticationKey	userAuthenticationKeyType	User.xsd
contractSummaryWithHistorySearch Criteria	complexType contractSummaryWithHistorySearchCriteriaType	ContractSummary.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
getListWithHistoryContract SummaryResponse	complexType getListWithHistoryContractSummaryResponseType	ContractSummary.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
listOfContractSummariesWithHistory	complexType	List of Contract

	listOfContractSummariesWithHistoryType	Summaries in XML format
--	--	-------------------------

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
  <soapenv:Body>
    <ns1:getListWithHistoryResponse
soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:ns1="http://contracts.DataCollection.BusinessServices.FPDS.gce2000.com"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
      <ns1:getListWithHistoryContractSummaryResponse xmlns:ns1="https://www.fpds.gov/FPDS">
        <ns1:requestNumber>2850866566</ns1:requestNumber>
        <ns1:confirmationNumber>2140437827</ns1:confirmationNumber>
        <ns1:outputMessages>
          <ns1:listOfErrors/>
          <ns1:listOfWarnings/>
          <ns1:listOfInfoMessages/>
        </ns1:outputMessages>
        <ns1:listOfContractSummariesWithHistory>
          <ns1:count>
            <ns1:total>1</ns1:total>
            <ns1:fetched>1</ns1:fetched>
          </ns1:count>
          <ns1:contractSummaryWithHistory>
            <ns1:departmentID>9700</ns1:departmentID>
            <ns1:departmentName>DEPT OF DEFENSE</ns1:departmentName>
            <ns1:agencyID>9700</ns1:agencyID>
            <ns1:agencyName>DEPT OF THE ARMY</ns1:agencyName>
            <ns1:PIID>0002</ns1:PIID>
            <ns1:referencedIDVAgencyID>9700</ns1:referencedIDVAgencyID>
            <ns1:referencedIDVPIID>W912ER04D0004</ns1:referencedIDVPIID>
            <ns1:contractActionType>Delivery Order</ns1:contractActionType>
            <ns1:obligatedAmount>20275148</ns1:obligatedAmount>
            <ns1:baseSignedDate>February 06, 2004</ns1:baseSignedDate>
            <ns1:transactionType>Contract</ns1:transactionType>
            <ns1:contractActionDescription>Services </ns1:contractActionDescription>
            <ns1:latestVendor>
              <ns1:vendorHeader>
                <ns1:vendorName>FLUOR INTERCONTINENTAL, INC</ns1:vendorName>
              </ns1:vendorHeader>
              <ns1:listOfVendorLocations>
                <ns1:vendorLocation>
                  <ns1:city>CHARLOTTE</ns1:city>
                  <ns1:state>NC</ns1:state>
                  <ns1:ZIPCode>NORTH CAROLINA</ns1:ZIPCode>
                  <ns1:countryCode>28210</ns1:countryCode>
                  <ns1:DUNSNumber>130757870</ns1:DUNSNumber>
                </ns1:vendorLocation>
              </ns1:listOfVendorLocations>
            </ns1:latestVendor>
          </ns1:contractSummaryWithHistory>
        </ns1:listOfContractSummariesWithHistory>
      </ns1:getListWithHistoryContractSummaryResponse>
    </ns1:getListWithHistoryResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

```

        <ns1:congressionalDistrict>09</ns1:congressionalDistrict>
    </ns1:vendorLocation>
</ns1:listOfVendorLocations>
</ns1:latestVendor>
<ns1:listOfVendors>
    <ns1:count>
        <ns1:total>0</ns1:total>
        <ns1:fetched>0</ns1:fetched>
    </ns1:count>
    </ns1:listOfVendors>
<ns1:listOfFundingAgencies>
    <ns1:fundingAgency>2100</ns1:fundingAgency>
</ns1:listOfFundingAgencies>
<ns1:listOfNAICS_CODES>
    <ns1:NAICS>
        <ns1:NAICSCode>236220</ns1:NAICSCode>
        <ns1:parentNAICSCode/>
        <ns1:NAICSName/>
    </ns1:NAICS>
</ns1:listOfNAICS_CODES>
<ns1:latestSummaryPlaceOfPerformance>
    <ns1:placeOfPerformanceStateCode/>
    <ns1:placeOfPerformanceCountryCode>UNITED
STATES</ns1:placeOfPerformanceCountryCode>
    <ns1:placeOfPerformanceCongressionalDistrictCode/>
</ns1:latestSummaryPlaceOfPerformance>
<ns1:listOfSummaryPlaceOfPerformance>
    <ns1:summaryPlaceOfPerformance>
        <ns1:placeOfPerformanceStateCode>IRQ</ns1:placeOfPerformanceStateCode>

<ns1:placeOfPerformanceCountryCode>IRQ</ns1:placeOfPerformanceCountryCode>
    <ns1:placeOfPerformanceCongressionalDistrictCode/>
    </ns1:summaryPlaceOfPerformance>
</ns1:listOfSummaryPlaceOfPerformance>
</ns1:contractSummaryWithHistory>
</ns1:listOfContractSummariesWithHistory>
</ns1:getListWithHistoryContractSummaryResponse>
</ns1:getListWithHistoryResponse>
</soapenv:Body>
</soapenv:Envelope>

```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	1. The Signed Date To is not in the right format(YYYY-MM-DD) 2. The Signed Date From is not in the right format(YYYY-MM-DD)

9.1.1.2 Organizations

9.1.1.2.1 Department

9.1.1.2.1.1 Service meta-specifications in WSDL

WSDL Parameter	Value
PortType	DepartmentPortType
Binding	DepartmentBinding
Soap-binding style	rpc
TargetNameSpace	Department.wsdl

9.1.1.2.1.2 Departments Service Methods

Service Type/Name	Service Description
create	The Department.create service creates a new Department in FPDS.
update	The Department.update service updates an existing Department in FPDS.
get	The Department.get service finds an existing Department record in FPDS and then composes and returns the XML representation of the Department information.
getList	The Department.getList service finds Department records, which match the input selection criteria. Matching records are returned in the defined XML format.
exists	The Department.exists service checks for the existence of a given Department code.
delete	The Depart.delete service deletes the given Department record if not referred by any other data.
disable	Department.disable service disables the Department record in the system.
enable	Department.enable service enables the Department record in the system.

9.1.1.2.1.3 create

- The Department.create method creates a new Department in FPDS.
- Only FPDS System Administrators are allowed to create a Department entry.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	userAuthenticationKeyType	User.xsd
Department	complexType departmentType	Department.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
createDepartmentResponse	complexType createDepartmentResponseType	Department.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
Department	complexType	XML representation of the created Department record

	departmentType	
--	----------------	--

Example:

```

<createDepartmentResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <department>
    ...
  </department>
  <outputMessages>
    ...
  </outputMessages>
</createDepartmentResponse>
    
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	<ol style="list-style-type: none"> Cannot create Department record. Input Department ID is empty. A Department with the same code already exists Cannot create Department record. Required Department ID is missing. Cannot create Department. End Date cannot be earlier than current date. Not authorized to create Department in the FPDS system.

9.1.1.2.1.4 update

- The Department.update method is the API that updates and performs modifications to an existing Department.
- Administrators who belong to the corresponding Department and administrators who can access the Department level object services are allowed to make modifications to the Department.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	userAuthenticationKeyType	User.xsd
Department	complexType departmentType	Department.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
updateDepartmentResponse	complexType updateDepartmentResponse Type	Department.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
Department	complexType departmentType	XML representation of the updated Department record

Example:

```
<updateDepartmentResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <department>
    ...
  </department>
  <outputMessages>
    ...
  </outputMessages>
</updateDepartmentResponse>
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	1. A Department with the same code already exists 2. Cannot update Department record. Start Date cannot be earlier than current date. 3. Cannot create Department. End Date cannot be earlier than current date. 4. Not authorized to create Department in the FPDS system

9.1.1.2.1.5 *get*

- The get service retrieves the existing Department information in FPDS and returns the record in the XML format specified by the schema.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
DepartmentID	complexType DepartmentIDType	Department.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
getDepartmentResponse	complexType getDepartmentResponse Type	Department.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
Department	complexType departmentType	XML representation of the updated Department record

Example:

```
<getDepartmentResponse>
  <requestNumber>549212</requestNumber>
```

```

    <confirmationNumber>329743</confirmationNumber>
    <department>
      ...
    </department>
    <outputMessages>
      ...
    </outputMessages>
  </getDepartmentResponse>

```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	1. Cannot get department record. Input Department ID is empty. 2. Department requested not found in the database

9.1.1.2.1.6 *getList*

- The getList service retrieves the existing Departments that satisfy the specified criteria in the request, from the FPDS system.
- On successful completion, it returns the record in the XML format specified by the schema.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
departmentSearchCriteria	complexType departmentSearchCriteriaType	Department.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
getListDepartmentResponse	complexType getListDepartmentResponse Type	Department.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
listOfDepartments	complexType listOfDepartmentsType	List of Departments XML

Example:

```

<getListDepartmentResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <listOfDepartments>
    ...
  </department>
  <department>
    ...

```

```

        </department>
        ...
    </listOfDepartments>
    <outputMessages>
        ...
    </outputMessages>
</getListDepartmentResponse>
    
```

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 Departments available for your search criteria

9.1.1.2.1.7 *exists*

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

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	userAuthenticationKeyType	User.xsd
DepartmentID	complexType DepartmentIDType	Department.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
existsDepartmentResponse	complexType existsDepartmentResponse Type	Department.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
isExists	xsd:boolean	True if record exists, or false otherwise.

Example:

```

<existsDepartmentResponse>
  <requestNumber>549212</requestNumber>
</existsDepartmentResponse>
    
```

```

    <confirmationNumber>329743</confirmationNumber>
    <isExists>
      true
    </isExists>
    <outputMessages>
      ...
    </outputMessages>
  </existsDepartmentResponse>

```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	1. Department requested does not exist in the database 2. Department ID entered is not valid
ListOfInformationalMessages	complexType listOfInformationalMessagesType	1. Department exists but is disabled in the database

9.1.1.2.1.8 delete

- The Department.delete method is the API to delete the Department and return success or failure back to the user.
- In case of failure, the validation results also specify the reason for the error.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
DepartmentID	complexType departmentIDType	Department.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
deleteDepartmentResponse	complexType deleteDepartmentResponseType	Department.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
isDeleted	xsd:boolean	True if record successfully deleted, false otherwise.

Example:

```

  <deleteDepartmentResponse>
    <requestNumber>549212</requestNumber>
    <confirmationNumber>329743</confirmationNumber>
    <department>
      ...
    </department>
  </deleteDepartmentResponse>

```

```

</department>
<outputMessages>
...
</outputMessages>
</deleteDepartmentResponse>
    
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	1. Cannot delete department record. Department ID is referenced by other records in the system. 2. Department requested for deletion not found in the database 3. Department ID entered is not valid

9.1.1.2.1.9 *disable*

- The disable service disables the existing Department record in FPDS and returns the success or failure of the disabling activity.
- The End Date of the Department record to the current date, on which the Department is disabled.
- FPDS System Administrator or Department Administrator can only perform this action.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
DepartmentID	complexType departmentIDType	Department.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
disableDepartmentResponse	complexType disableDepartmentResponse Type	Department.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
isDisabled	xsd:boolean	True if record successfully disabled, false otherwise.

Example:

```

<disableDepartmentResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <department>
    ...
  </department>
  <outputMessages>
    ...
  </outputMessages>
</disableDepartmentResponse>
    
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	1. Department requested for disabling not found in the database 2. Department ID entered is not valid

9.1.1.2.1.10 enable

- The enable service enables the existing Department record in FPDS and returns the success or failure of the enabling activity.
- The End Date of the Department is set to blank.
- FPDS System Administrator or Department Administrator can only perform this action.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
DepartmentID	complexType departmentIDType	Department.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
enableDepartmentResponse	complexType enableDepartmentResponse Type	Department.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
isEnabled	xsd:boolean	True if record successfully enabled false otherwise.

Example:

```

<enableDepartmentResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <department>
    ...
  </department>
  <outputMessages>
    ...
  </outputMessages>
</enableDepartmentResponse>
    
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	1. Department requested for enabling not found in the database 2. Department ID entered is not valid

9.1.1.2.2 Agency

9.1.1.2.2.1 Service meta-specifications in WSDL

WSDL Parameter	Value
PortType	AgencyPortType
Binding	AgencyBinding
Soap-binding style	rpc
TargetNameSpace	Agency.wsdl

9.1.1.2.2.2 Agency Service methods

Service Type/Name	Service Description
create	The Agency.create service creates a new Agency in FPDS. This service utilizes the validation services to check record validity before insertion into the FPDS database.
update	The Agency.update service updates an existing Agency in FPDS. This service utilizes the validation services to check record validity before updating the FPDS database.
get	The Agency.get service finds an existing Agency record in FPDS and then composes and returns the XML representation of the Agency record.
getList	The Agency.getList service finds Agency records, which match the input selection criteria. Matching records are returned in the defined XML format.
exists	The Agency.exists service checks for the existence of a given Agency code.
delete	The Agency.delete service deletes the given Agency in the FPDS System.
disable	Agency.disable service disables the Agency record in the system.
enable	Agency.enable service enables the Agency record in the system

9.1.1.2.2.3 create

- The Agency.create method creates a new Agency in FPDS.
- Only FPDS System Administrator or Department Administrator can perform this action

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	userAuthenticationKeyType	User.xsd
Agency	complexType AgencyType	Agency.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
createAgencyResponse	complexType createAgencyResponse Type	Agency.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
Agency	complexType agencyType	XML representation of the updated Agency record

Example:

```
<createAgencyResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <agency>
    ...
  </agency>
  <outputMessages>
    ...
  </outputMessages>
</createAgencyResponse>
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	1. Cannot create Agency record. Input Agency ID is empty. 2. An Agency with the same code already exists 3. Cannot create Agency record. Required Agency ID is missing. 4. Cannot create Agency. End Date cannot be earlier than current date. 5. Not authorized to create Agency in the FPDS system

9.1.1.2.2.4 update

- The Agency.update method is the API that updates and performs modifications to the existing Agency.
- Administrators who belong to the corresponding Agency and administrators who can access the Agency level object services are allowed to make modifications to the Agency.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	userAuthenticationKeyType	User.xsd
Agency	complexType AgencyType	Agency.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
updateAgencyResponse	complexType updateAgencyResponse Type	Agency.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
Agency	complexType agencyType	XML representation of the updated Agency record

Example:

```
<updateAgencyResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <agency>
    ...
  </agency>
  <outputMessages>
    ...
  </outputMessages>
</updateAgencyResponse>
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	1. An Agency with the same code already exists 2. Cannot update Agency record. Start Date cannot be earlier than current date. 3. Cannot create Agency. End Date cannot be earlier than current date. 4. Not authorized to create Agency in the FPDS system

9.1.1.2.2.5 *get*

- The get service retrieves the existing Agency information in FPDS and returns the record in the XML format specified by the schema.
- The get service also authenticates the requesting user before sending back the response.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
AgencyID	complexType agencyIDType	Agency.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
getAgencyResponse	complexType getAgencyResponse Type	Agency.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
agency	complexType AgencyType	Agency.xsd

Example:

```
<getAgencyResponse>
  <requestNumber>549212</requestNumber>
```

```

    <confirmationNumber>329743</confirmationNumber>
    <agency>
        ...
    </agency>
    <outputMessages>
        ...
    </outputMessages>
</getAgencyResponse>

```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	1. Cannot get Agency record. Input Agency ID is empty. 2. Agency requested not found in the database

9.1.1.2.2.6 *getList*

- The getList service retrieves the existing Agency information that satisfies the specified criteria in the request from FPDS.
- On successful completion, getList returns the record in the XML format specified by the schema.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
agencySearchCriteria	complexType agencySearchCriteriaType	Agency.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
getListAgencyResponse	complexType getListAgencyResponseType	Agency.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
ListOfAgencies	complexType listOfAgenciesType	List of Agencies XML

Example:

```

<getListAgencyResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <listOfAgencys>
    <agency>
      ...
    </agency>
    <agency>
      ...

```

```

        </agency>
        ...
    </listOfAgencys>
    <outputMessages>
        ...
    </outputMessages>
</getListAgencyResponse>
    
```

Failure Output

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

9.1.1.2.2.7 *exists*

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

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	userAuthenticationKeyType	User.xsd
AgencyID	complexType agencyIDType	Agency.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
existsAgencyResponse	complexType existsAgencyResponse Type	Agency.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
IsExists	xsd:Boolean	True if record exists, false otherwise.

Example:

```

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

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	<ol style="list-style-type: none"> Agency requested does not exist in the database Agency ID entered is not valid

9.1.1.2.2.8 *delete*

- The Agency.delete method is the API to delete the Agency and return success or failure back to the user.
- In case of failure, the validation results also specify the reason for the error.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
AgencyID	complexType agencyIDType	Agency.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
deleteAgencyResponse	complexType deleteAgencyResponse Type	Agency.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
isDeleted	xsd:boolean	True if record successfully deleted, false otherwise.

Example:

```

<deleteAgencyResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <agency>
    ...
  </agency>
  <outputMessages>
    ...
  </outputMessages>
</deleteAgencyResponse>
    
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	<ol style="list-style-type: none"> Cannot delete Agency record. Agency ID is referenced by other records in the system. Agency requested for deletion not found in the database Agency ID entered is not valid

9.1.1.2.2.9 *disable*

- The disable service disables the existing Agency record in FPDS and returns the success or failure of the disabling activity.
- The End Date of the Agency record to the current date, on which the Agency is disabled.
- FPDS System Administrator or Agency Administrator can only perform this action.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
AgencyID	complexType agencyIDType	Agency.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
disableAgencyResponse	complexType disableAgencyResponse Type	Agency.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
isDisabled	xsd:boolean	True if record successfully disabled, false otherwise.

Example:

```
<disableAgencyResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <agency>
    ...
  </agency>
  <outputMessages>
    ...
  </outputMessages>
</disableAgencyResponse>
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	1. Agency requested for disabling not found in the database 2. Agency ID entered is not valid

9.1.1.2.2.10 enable

- The enable service enables the existing Agency record in FPDS and returns the success or failure of the enabling activity.
- The End Date of the Department is set to blank.
- FPDS System Administrator or Department Administrator can only perform this action.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd

AgencyID	complexType agencyIDType	Agency.xsd
----------	--------------------------	------------

Output Parameters

Parameter Name	Parameter Type	Name Space
enableAgencyResponse	complexType enableAgencyResponse Type	Agency.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
isEnabled	xsd:boolean	True if record successfully enabled, false otherwise.

Example:

```

<enableAgencyResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <agency>
    ...
  </agency>
  <outputMessages>
    ...
  </outputMessages>
</enableAgencyResponse>
    
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	1. Agency requested for enabling not found in the database 2. Agency ID entered is not valid

9.1.1.2.3 Contracting Office

9.1.1.2.3.1 Service meta-specifications in WSDL

WSDL Parameter	Value
PortType	ContractingOfficePortType
Binding	ContractingOfficeBinding
Soap-binding style	rpc
TargetNameSpace	ContractingOffice.wsdl

9.1.1.2.3.2 Contracting Office Service methods

Service Type/Name	Service Description
create	The ContractingOffice.create service creates a new ContractingOffice in FPDS. This service utilizes the validation services to check record validity before insertion into the FPDS database.
update	The ContractingOffice.update service updates an existing Agency in FPDS. This service utilizes the validation services to check record validity before updating the FPDS database.

Service Type/Name	Service Description
get	The ContractingOffice.get service finds an existing ContractingOffice record in FPDS and then composes and returns the XML representation of the Contract Office.
getList	The ContractingOffice.getList service finds Contracting Office records, which match the input selection criteria. Matching records are returned in the defined XML format.
exists	The ContractingOffice.exists service checks for the existence of a given ContractingOffice code.
delete	The ContractingOffice.delete service deletes the given ContractingOffice in the FPDS System.
disable	ContractingOffice.disable service disables the ContractingOffice record in the system.
enable	ContractingOffice.enable service enables the ContractingOffice record in the system.

9.1.1.2.3.3 create

- The Contracting Office create method creates a new contracting office in FPDS.
- Only FPDS System Administrators are allowed to create a contracting office entry.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
ContractingOffice	complexType ContractingOfficeType	ContractingOffice.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
createContractingOfficeResponse	complexType createContractingOfficeResponse Type	ContractingOffice.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
ContractingOffice	complexType contractingOfficeType	XML representation of the updated ContractingOffice record

Example:

```

<createContractingOfficeResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <contractingOffice>
    ...
  </contractingOffice>
  <outputMessages>
    ...
  </outputMessages>
</createContractingOfficeResponse>
    
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	<ol style="list-style-type: none"> 1. Cannot create ContractingOffice record. Input ContractingOffice ID is empty. 2. A ContractingOffice with the same code already exists 3. Cannot create ContractingOffice record. Required ContractingOffice ID is missing. 4. Cannot create ContractingOffice. End Date cannot be earlier than current date. 5. Not authorized to create ContractingOffice in the FPDS system

9.1.1.2.3.4 update

- The ContractingOffice.update method is the API to update and perform modifications to the existing ContractingOffice.
- Administrators who belong to the corresponding ContractingOffice and administrators who can access the ContractingOffice level object services are allowed to make modifications to the ContractingOffice.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
ContractingOffice	complexType contractingOfficeType	ContractingOffice.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
updateContractingOfficeResponse	complexType updateContractingOfficeResponse Type	ContractingOffice.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
ContractingOfficeXML	complexType contractingOfficeType	XML representation of the updated ContractingOffice record

Example:

```

<updateContractingOfficeResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <contractingOffice>
    ...
  </contractingOffice>
  <outputMessages>
    ...
  </outputMessages>
</updateContractingOfficeResponse>
    
```


Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	<ol style="list-style-type: none"> 1. A Contracting Office with the same code already exists 2. Cannot update ContractingOffice record. Start Date cannot be earlier than current date. 3. Cannot create ContractingOffice. End Date cannot be earlier than current date. 4. Not authorized to create Contracting Office in the FPDS system

9.1.1.2.3.5 *get*

- The get service retrieves the existing ContractingOffice information in FPDS and returns the record in the XML format specified by the schema.
- The get 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	User.xsd
ContractingOfficeKey	complexType contractingOfficeKeyType	ContractingOffice.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
getContractingOfficeResponse	complexType getContractingOfficeResponse Type	ContractingOffice.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
ContractingOffice	complexType contractingOfficeType	XML representation of the updated ContractingOffice record

Example:

```

<getContractingOfficeResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <contractingOffice>
    ...
  </contractingOffice>
  <outputMessages>
  </outputMessages>
</getContractingOfficeResponse>
    
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	1. Cannot get Contracting Office record. Input Contracting Office ID is empty. 2. Contracting Office requested not found in the database

9.1.1.2.3.6 *getList*

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

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
contractingOfficeSearchCriteria	complexType contractingOfficeSearchCriteriaType	ContractingOffice.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
getListContractingOfficeResponse	complexType getListContractingOfficeResponse Type	ContractingOffice.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
listOfContractingOffices	listOfContractingOfficesType	List of ContractingOffices XML

Example:

```

<getListContractingOfficeResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <listOfContractingOffices>
    <contractingOffice>
      ...
    </contractingOffice>
    <contractingOffice>
      ...
    </contractingOffice>
    ...
  </listOfContractingOffices>
  <outputMessages>
    ...
  </outputMessages>
</getListContractingOfficeResponse>
    
```

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 ContractingOffices available for your search criteria

9.1.1.2.3.7 *exists*

- The exists returns true or false corresponding to the existence of the record in the database.
- It checks whether the Contracting Office record, as per the criteria in the request, exists in FPDS and returns success or failure response.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
ContractingOfficeKey	complexType contractingOfficeKeyType	ContractingOffice.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
existsContractingOfficeResponse	complexType existsContractingOfficeResponseType	ContractingOffice.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
isExists	xsd:Boolean	True if record exists, false otherwise.

Example:

```

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

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	<ol style="list-style-type: none"> Contracting Office requested does not exist in the database ContractingOffice ID entered is not valid

9.1.1.2.3.8 delete

- The ContractingOffice.delete method deletes a Contracting Office in FPDS.
- Only FPDS System Administrators are allowed to delete a Contracting Office entry.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
ContractingOfficeKey	complexType contractingOfficeKeyType	ContractingOffice.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
deleteContractingOfficeResponse	complexType deleteContractingOfficeResponse Type	ContractingOffice.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
isDeleted	xsd:boolean	True if record successfully deleted, false otherwise.

Example:

```

<deleteContractingOfficeResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <contractingOffice>
    ...
  </contractingOffice>
  <outputMessages>
    ...
  </outputMessages>
</deleteContractingOfficeResponse>
    
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	<ol style="list-style-type: none"> Cannot delete Contracting Office record. Contracting Office ID is referenced by other records in the system. Contracting Office requested for deletion not found in the database

		3. Contracting Office ID entered is not valid
--	--	---

9.1.1.2.3.9 *disable*

- The disable service disables the existing ContractingOffice record in FPDS and returns the success or failure of the disabling activity.
- The End Date of the ContractingOffice record is set to the current date, upon which the ContractingOffice is disabled.
- The service authentication input must contain credentials of a FPDS Administrator or Contracting Office Administrator who is authorized to disable this ContractingOffice.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
ContractingOfficeKey	complexType contractingOfficeKeyType	ContractingOffice.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
disableContractingOfficeResponse	complexType disableContractingOfficeResponse Type	ContractingOffice.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
isDisabled	xsd:boolean	True if record successfully disabled, false otherwise.

Example:

```

<disableContractingOfficeResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <contractingOffice>
    ...
  </contractingOffice>
  <outputMessages>
    ...
  </outputMessages>
</disableContractingOfficeResponse>
    
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	1. Contracting Office requested for disabling not found in the database 2. Contracting Office ID entered is not valid

9.1.1.2.3.10 *enable*

- The enable service enables the existing ContractingOffice record in FPDS and returns the success or failure of the enabling activity.

- The End Date of the ContractingOffice is set to blank.
- FPDS System Administrator or Contracting Office Administrator can only perform this action.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
contractingOfficeKey	complexType contractingOfficeKey KeyType	ContractingOffice.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
enableContractingOfficeResponse	complexType enableContractingOfficeResponse Type	ContractingOffice.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
IsEnabled	xsd:Boolean	True if record successfully enabled false otherwise.

Example:

```

<enableContractingOfficeResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <agency>
    ...
  </agency>
  <outputMessages>
    ...
  </outputMessages>
</enableContractingOfficeResponse>
    
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	1. Contracting Office requested for enabling not found in the database 2. Contracting Office ID entered is not valid

9.1.1.2.4 Government Office

9.1.1.2.4.1 Service meta-specifications in WSDL

WSDL Parameter	Value
PortType	GovernmentOfficePortType
Binding	GovernmentOfficeBinding
Soap-binding style	rpc
TargetNameSpace	GovernmentOffice.wsdl

9.1.1.2.4.2 Government Office Service Methods

Service Type/Name	Service Description
create	The GovernmentOffice.create service creates a new GovernmentOffice in FPDS. This service utilizes the validation services to check record validity before insertion into the FPDS database.
update	The GovernmentOffice.update service updates an existing GovernmentOffice in FPDS. This service utilizes the validation services to check record validity before updating the FPDS database.
get	The GovernmentOffice.get service finds an existing GovernmentOffice record in FPDS and then composes and returns the XML representation of the Contract Office.
getList	The GovernmentOffice.getList service finds GovernmentOffice records, which match the input selection criteria. Matching records are returned in the defined XML format.
exists	The GovernmentOffice.exists service checks for the existence of a given GovernmentOffice code.
delete	The GovernmentOffice.delete service deletes the given GovernmentOffice in the FPDS System.
disable	GovernmentOffice.disable service disables the GovernmentOffice record in the system.
enable	GovernmentOffice.enable service enables the GovernmentOffice record in the system.

9.1.1.2.4.3 create

- The GovernmentOffice.create method creates a new Government Office in FPDS.
- Only FPDS System Administrators are allowed to create a Government Office entry.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
governmentOffice	complexType GovernmentOfficeType	GovernmentOffice.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
createGovernmentOfficeResponse	complexType createGovernmentOfficeResponse Type	GovernmentOffice.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
GovernmentOfficeXML	complexType governmentOfficeType	XML representation of the updated GovernmentOffice record

Example:

```
<createGovernmentOfficeResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
```

```

    <governmentOffice>
      ...
    </governmentOffice>
    <outputMessages>
      ...
    </outputMessages>
  </createGovernmentOfficeResponse>

```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	1. Cannot create Government Office record. Input Government Office ID is empty. 2. A Government Office with the same code already exists 3. Cannot create Government Office record. Required Government Office ID is missing . 4. Cannot create Government Office. End Date cannot be earlier than current date. 5. Not authorized to create Government Office in the FPDS system

9.1.1.2.4.4 *update*

- The GovernmentOffice.update method is the API to update and perform modifications to the existing government office.
- Administrators who belong to the corresponding Government Office and administrators who can access the Government Office level object services are allowed to make modifications to the Government Office.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
GovernmentOffice	complexType governmentOfficeType	GovernmentOffice.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
updateGovernmentOfficeResponse	complexType updateGovernmentOfficeResponse Type	GovernmentOffice.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
GovernmentOffice	complexType governmentOfficeType	XML representation of the updated Government Office record

Example:

```

<updateGovernmentOfficeResponse>

```



```

<requestNumber>549212</requestNumber>
<confirmationNumber>329743</confirmationNumber>
<governmentOffice>
  ...
</governmentOffice>
<outputMessages>
  ...
</outputMessages>
</updateGovernmentOfficeResponse>
    
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	<ol style="list-style-type: none"> 1. A Government Office with the same code already exists 2. Cannot update Government Office record. Start Date cannot be earlier than current date. 3. Cannot create Government Office. End Date cannot be earlier than current date. 4. Not authorized to create Government Office in the FPDS system

9.1.1.2.4.5 *get*

- The get service retrieves the existing GovernmentOffice information in FPDS and returns the record in the XML format specified by the schema.
- The get 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	User.xsd
governmentOfficeKey	complexType governmentOfficeKeyType	GovernmentOffice.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
getGovernmentOfficeResponse	complexType getGovernmentOfficeResponse Type	GovernmentOffice.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
GovernmentOffice	complexType governmentOfficeType	XML representation of the updated Government Office record

Example:

```

<getGovernmentOfficeResponse>
  <requestNumber>549212</requestNumber>
    
```

```

    <confirmationNumber>329743</confirmationNumber>
    <governmentOffice>
      ...
    </governmentOffice>
    <outputMessages>
    </outputMessages>
  </getGovernmentOfficeResponse>

```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	1. Cannot get Government Office record. Input Government Office ID is empty. 2. Government Office requested not found in the database

9.1.1.2.4.6 *getList*

- The getList service retrieves the existing Government Offices’ information that satisfies the specified criteria in the request, from FPDS.
- On successful retrieval, it returns the records in the XML format specified by the schema.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
governmentOfficeSearchCriteria	complexType governmentOfficeSearchCriteriaType	GovernmentOffice.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
getListGovernmentOfficeResponse	complexType getListGovernmentOfficeResponse Type	GovernmentOffice.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
listOfGovernmentOffices	listOfGovernmentOfficesType	List of GovernmentOffices XML

Example:

```

<getListGovernmentOfficeResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <listOfGovernmentOffices>
    <governmentOffice>
      ...
    </governmentOffice>
    <governmentOffice>
      ...
    </governmentOffice>
  </listOfGovernmentOffices>
</getListGovernmentOfficeResponse>

```

```

    </governmentOffice>
    ...
</listOfGovernmentOffices>
<outputMessages>
    ...
</getListGovernmentOfficeResponse>
    
```

Failure Output

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

9.1.1.2.4.7 *exists*

- The exists returns true or false corresponding to the existence of the record in the database.
- It checks whether the Government Office record, as per the criteria in the request, exists in FPDS and returns success or failure response.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
governmentOfficeKey	complexType governmentOfficeKeyType	GovernmentOffice.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
existsGovernmentOfficeResponse	complexType existsGovernmentOfficeResponse Type	GovernmentOffice.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
isExists	xsd:Boolean	True if record exists, false otherwise.

Example:

```

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

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	<ol style="list-style-type: none"> Government Office requested does not exist in the database Government Office ID entered is not valid

9.1.1.2.4.8 delete

- The GovernmentOffice.delete method deletes a Government Office in FPDS.
- Only FPDS System Administrators are allowed to delete a Government Office entry.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
GovernmentOfficeKey	complexType governmentOfficeKeyType	GovernmentOffice.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
deleteGovernmentOfficeResponse	complexType deleteGovernmentOfficeResponseType	GovernmentOffice.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
isDeleted	xsd:boolean	True if record successfully deleted, false otherwise.

Example:

```

<deleteGovernmentOfficeResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <governmentOffice>
    ...
  </governmentOffice>
  <outputMessages>
    ...
  </outputMessages>
</deleteGovernmentOfficeResponse>
    
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	<ol style="list-style-type: none"> Cannot delete Government Office record. Government Office ID is referenced by other records in the system. Government Office requested for deletion not found in the database

		3. Government Office ID entered is not valid
--	--	--

9.1.1.2.4.9 *disable*

- The disable service disables the existing GovernmentOffice record in FPDS and returns the success or failure of the disabling activity.
- The End Date of the GovernmentOffice record is set to the current date, upon which the GovernmentOffice is disabled.
- The service authentication input must contain credentials of an FPDS Administrator or Government Office Administrator who is authorized to disable this GovernmentOffice.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
GovernmentOfficeKey	complexType governmentOfficeKeyType	GovernmentOffice.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
disableGovernmentOfficeResponse	complexType disableGovernmentOfficeResponse Type	GovernmentOffice.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
isDisabled	xsd:boolean	True if record successfully disabled, false otherwise.

Example:

```

<disableGovernmentOfficeResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <governmentOffice>
    ...
  </governmentOffice>
  <outputMessages>
    ...
  </outputMessages>
</disableGovernmentOfficeResponse>
    
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	1. Government Office requested for disabling not found in the database 2. Government Office ID entered is not valid

9.1.1.3 Vendors

9.1.1.3.1 Vendor

9.1.1.3.1.1 Service meta-specifications in WSDL

WSDL Parameter	Value
PortType	VendorPortType
Binding	VendorBinding
Soap-binding style	rpc
TargetNameSpace	Vendor.wsdl

9.1.1.3.1.2 Vendor Service methods

Service Type/Name	Service Description
get	The Vendor.get service finds an existing Vendor in FPDS and then composes and returns the XML representation of the IDV.
getList	The Vendors.getList service finds Vendor records, which match the input selection criteria. Matching records are returned in the defined XML format.
exists	The Vendor.exists service checks for the existence of a given Vendor code.

9.1.1.3.1.3 get

- The get service retrieves the existing Vendor information in FPDS and returns the record in the XML format specified by the schema.
- The get 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	User.xsd
vendorSearchCriteria	complexType VendorSearchCriteriaType	Vendor.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
getVendorResponse	complexType getVendorResponse Type	Vendor.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
Vendor	complexType vendorType	Vendor.xsd

Example:

```
<getVendorResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
```

```

<vendor>
...
</vendor>
<outputMessages>
</outputMessages>
</getVendorResponse>
    
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	1. Cannot get Vendor record. Input Vendor ID is empty. 2. Entity requested not found in the database

9.1.1.3.1.4 *getList*

- The Vendor.getList service retrieves the existing Vendor information that satisfies the specified criteria in the request, from FPDS.
- On successful retrieval, this service returns the records in the XML format specified by the schema.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
vendorSearchCriteria	complexType vendorSearchCriteriaType	Vendor.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
getListVendorResponse	complexType getListVendorResponseType	Vendor.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
listOfVendors	complexType listOfVendorsType	List of Vendors XML

Example:

```

<getListVendorResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <listOfVendors>
    <vendor>
      ...
    </vendor>
    <vendor>
      ...
    </vendor>
    ...
  </listOfVendors>
</getListVendorResponse>
    
```

```

</listOfVendors>
<outputMessages>
...
</outputMessages>
</getListVendorResponse>
    
```

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 Entities available for your search criteria

9.1.1.3.1.5 *exists*

- Check whether the Vendor record, as per the criteria in the request, exists in FPDS and returns success or failure response.
- The exists method also checks for valid authorization of the requesting user before sending back the response.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
VendorID	complexType VendorIDType	Vendor.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
existsVendorResponse	complexType existsVendorResponse Type	Vendor.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
isExists	xsd:boolean	True if record exists, or false otherwise.

Example:

```

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


</existsVendorResponse>

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	1. Entity requested does not exist in the database 2. Vendor ID entered is not valid
ListOfInformational Messages	complexType listOfInformationalMessagesType	1. Entity exists but is disabled in the database

9.1.2 Product/Service Classifications

9.1.2.1 NAICS

9.1.2.1.1 Service meta-specifications in WSDL

WSDL Parameter	Value
PortType	NAICSPortType
Binding	NAICSBinding
Soap-binding style	rpc
TargetNameSpace	NAICS.wsdl

9.1.2.1.1.1 NAICS Service methods

Service Type/Name	Service Description
create	The NAICS.create service creates a new NAICS record in FPDS. This service utilizes the validation services to check record validity before insertion into the FPDS database.
update	The NAICS.update service updates an existing NAICS in FPDS. This service utilizes the validation services to check record validity before updating the FPDS database.
get	The NAICS.get service finds an existing NAICS reference record in FPDS and then composes and returns the XML representation of the NAICS record.
getList	The NAICS.getList service finds NAICS records, which match the input selection criteria. Matching records are returned in the defined XML format.
exists	The NAICS.exists service checks for the existence of a given NAICS code
delete	The NAICS.delete service deletes the given NAICS code in the FPDS System.
disable	NAICS.disable service disables the NAICS record in the system.
enable	NAICS.enable service enables the NAICS record in the system.

9.1.2.1.1.2 create

- The NAICS.create method creates a new NAICS in FPDS.
- Only FPDS System Administrators may perform this functionality.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
NAICS	complexType NAICSType	NAICS.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
createNAICSResponse	complexType createNAICSResponse Type	NAICS.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
NAICS	complexType NAICSType	NAICS.xsd

Example:

```

<createNAICSResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <NAICS>
    ...
  </NAICS>
  <outputMessages>
    ...
  </outputMessages>
</createNAICSResponse>
    
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	<ol style="list-style-type: none"> Cannot create NAICS record. Input NAICS ID is empty. A NAICS with the same code already exists Cannot create NAICS record. Required NAICS ID is missing. Cannot create NAICS. End Date cannot be earlier than current date. Not authorized to create NAICS in the FPDS system

9.1.2.1.1.3 *update*

- The NAICS.update method is the API to update and perform modifications to an existing NAICS.
- Only FPDS System Administrators may perform this functionality.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	complexType UserAuthenticationKeyType	Common.xsd
NAICS	complexType NAICSType	NAICS.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
updateNAICSResponse	complexType updateNAICSResponse Type	NAICS.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
NAICS	complexType NAICSType	XML representation of the updated NAICS record

Example:

```
<updateNAICSResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <NAICS>
    ...
  </NAICS>
  <outputMessages>
    ...
  </outputMessages>
</updateNAICSResponse>
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	<ol style="list-style-type: none"> 1. A NAICS with the same code already exists 2. Cannot update NAICS record. Start Date cannot be earlier than current date. 3. Cannot create NAICS record. End Date cannot be earlier than current date. 4. Not authorized to create NAICS record in the FPDS system

9.1.2.1.1.4 *get*

- The get service retrieves the existing NAICS information in FPDS and returns the record in the XML format specified by the schema.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	complexType UserAuthenticationKeyType	Common.xsd

NAICSCode	complexType NAICSCodeType	NAICS.xsd
-----------	---------------------------	-----------

Output Parameters

Parameter Name	Parameter Type	Name Space
getNAICSResponse	complexType getNAICSResponse Type	NAICS.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
NAICS	complexType NAICSType	NAICS.xsd

Example:

```

<getNAICSResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <NAICS>
    ...
  </NAICS>
  <outputMessages>
  </outputMessages>
</getNAICSResponse>
    
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	1. Cannot get NAICS record. Input NAICS ID is empty. 2. NAICS requested not found in the database

9.1.2.1.1.5 *getList*

- The getList service retrieves the existing NAICS information that satisfies the specified criteria in the request, from the FPDS system.
- Upon successful retrieval, it returns the records in the XML format specified by the schema.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	complexType UserAuthenticationKeyType	Common.xsd
NAICSSearchCriteria	complexType NAICSSearchCriteriaType	NAICS.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
getListNAICSResponse	complexType getListNAICSResponse Type	NAICS.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
listOfNAICSs	listOfNAICSType	List of NAICSs XML

Example:

```

<getListNAICSResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <listOfNAICSs>
    <NAICS>
      ...
    </NAICS>
    <NAICS>
      ...
    </NAICS>
    ...
  </listOfNAICSs>
  <outputMessages>
    ...
  </outputMessages>
</getListNAICSResponse>
    
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	<ol style="list-style-type: none"> Service unavailable No records found

9.1.2.1.1.6 exists

The exists service checks whether the NAICS record, as per the criteria in the request, exists in FPDS and returns success or failure response.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
NAICSCode	complexType NAICSCodeType	NAICS.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
existsNAICSResponse	complexType existsNAICSResponse Type	NAICS.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
isExists	xsd:boolean	True if record exists, or false otherwise.

Example:

```

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

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	1. NAICS requested does not exist in the database 2. NAICS entered is not valid
ListOfInformationalMessages	complexType listOfInformationalMessagesType	1. NAICS exists but is disabled in the database

9.1.2.1.1.7 delete

- The delete method is the API to delete the NAICS and return success or failure back to the user.
- In case of failure, the validation results also specify the reason for the error.
- Only FPDS System Administrators may perform this functionality.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	complexType UserAuthenticationKeyType	Common.xsd
NAICSCode	complexType NAICSCodeType	NAICS.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
deleteNAICSResponse	complexType deleteNAICSResponse Type	NAICS.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
IsDeleted	xsd:Boolean	True if record successfully deleted, false otherwise.

Example:

```

<deleteNAICSResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <NAICS>
    ...
  </NAICS>
  <outputMessages>
    ...
  </outputMessages>
</deleteNAICSResponse>
    
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	<ol style="list-style-type: none"> Cannot delete NAICS record. NAICS is referenced by other records in the system. NAICS requested for deletion not found in the database NAICS entered is not valid

9.1.2.1.1.8 disable

- The disable service disables the existing NAICS record in FPDS and returns the success or failure of the disabling activity.
- The End Date of the NAICS record is set to the current date, upon which the NAICS is disabled.
- The service authentication input must contain credentials of a FPDS Administrator or NAICS Administrator who is authorized to disable this NAICS.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	complexType UserAuthenticationKeyType	Common.xsd
NAICSCode	complexType NAICSCodeType	NAICS.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
disableNAICSResponse	complexType disableNAICSResponseType	NAICS.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
IsDisabled	xsd:Boolean	True if record successfully disabled, false otherwise.

Example:

```
<disableNAICSResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <NAICS>
    ...
  </NAICS>
  <outputMessages>
    ...
  </outputMessages>
</disableNAICSResponse>
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	1. NAICS requested for disabling not found in the database 2. NAICS entered is not valid

9.1.2.1.1.9 enable

- The enable service enables the existing NAICS record in FPDS and returns the success or failure of the enabling activity.
- The End Date of the NAICS record is cleared, upon which the NAICS is enabled.
- The service authentication input must contain credentials of a FPDS Administrator or NAICS Administrator who is authorized to enable this NAICS.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	complexType UserAuthenticationKeyType	Common.xsd
NAICSCode	complexType NAICSCodeType	NAICS.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
enableNAICSResponse	complexType enableNAICSResponseType	NAICS.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
IsEnabled	xsd:Boolean	True if record successfully enabled, false otherwise.

Example:


```

<enableNAICSResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <NAICS>
    ...
  </NAICS>
  <outputMessages>
    ...
  </outputMessages>
</enableNAICSResponse>
    
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	1. NAICS requested for enabling not found in the database 2. NAICS entered is not valid

9.1.2.2 PSC

9.1.2.2.1 Service meta-specifications in WSDL

WSDL Parameter	Value
PortType	PSCPortType
Binding	PSCBinding
Soap-binding style	rpc
TargetNameSpace	PSC.wsdl

9.1.2.2.1.1 PSC Service methods

Service Type/Name	Service Description
create	The PSC.create service creates a new PSC in FPDS.
update	The PSC.update service updates an existing PSC in FPDS.
get	The PSC.get service finds an existing PSC record in FPDS and then composes and returns the XML representation of the PSC.
getList	The PSC.getList service finds PSC records, which match the input selection criteria. Matching records are returned in the defined XML format.
exists	The PSC.exists service checks for the existence of a given PSC code
delete	The PSC.delete service deletes the given PSC code in the FPDS System.
disable	PSC.disable service disables the PSC record in the system.

9.1.2.2.1.2 create

- The PSC.create method creates a new PSC in FPDS.
- Only FPDS System Administrators may perform this functionality.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	complexType UserAuthenticationKeyType	Common.xsd
PSC	complexType PSCType	PSC.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
createPSCResponse	complexType createPSCResponse Type	PSC.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
PSC	complexType PSCType	XML representation of the updated PSC record

Example:

```

<createPSCResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <PSC>
    ...
  </PSC>
  <outputMessages>
    ...
  </outputMessages>
</createPSCResponse>
    
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	<ol style="list-style-type: none"> 1. Cannot create PSC record. Input PSC ID is empty. 2. A PSC with the same code already exists 3. Cannot create PSC record. Required PSC ID is missing. 4. Cannot create PSC. End Date cannot be earlier than current date. 5. Not authorized to create PSC in the FPDS system

9.1.2.2.1.3 update

- The PSC.update method is the API to update and perform modifications to the existing PSC.
- Only FPDS System Administrators may perform this functionality.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	complexType	Common.xsd

Parameter Name	Parameter Type	Name Space
	UserAuthenticationKeyType	
PSC	complexType PSCType	PSC.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
updatePSCResponse	complexType updatePSCResponse Type	PSC.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
PSC	complexType PSCType	XML representation of the updated PSC record

Example:

```

<updatePSCResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <PSC>
    ...
  </PSC>
  <outputMessages>
    ...
  </outputMessages>
</updatePSCResponse>
    
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	<ol style="list-style-type: none"> 1. A PSC with the same code already exists 2. Cannot update PSC record. Start Date cannot be earlier than current date. 3. Cannot create PSC. End Date cannot be earlier than current date. 4. Not authorized to create PSC in the FPDS system

9.1.2.2.1.4 *get*

- The get service retrieves the existing PSC information in FPDS and returns the record in the XML format specified by the schema.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	complexType UserAuthenticationKeyType	Common.xsd
PSCCode	complexType PSCCodeType	PSC.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
getPSCResponse	complexType getPSCResponse Type	PSC.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
PSC	complexType PSCType	PSC.xsd

Example:

```

<getPSCResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <PSC>
    ...
  </PSC>
  <outputMessages>
  </outputMessages>
</getPSCResponse>
    
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	1. Cannot get PSC record. Input PSC ID is empty. 2. PSC requested not found in the database

9.1.2.2.1.5 getList

- The PSC.getList service retrieves the existing PSC information that satisfies the specified criteria in the request, from FPDS.
- On successful retrieval, returns the records in the XML format specified by the schema.

Input Parameter

Parameter Name	Parameter Type	Name Space
AuthenticationKey	complexType UserAuthenticationKeyType	Common.xsd
PSCSearchCriteria	complexType PSCSearchCriteriaType	PSC.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
getListPSCResponse	complexType getListPSCResponse Type	PSC.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
listOfPSCs	listOfPSCsType	List of PSCs XML

Example:

```

<getListPSCResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <listOfPSCs>
    <PSC>
      ...
    </PSC>
    <PSC>
      ...
    </PSC>
    ...
  </listOfPSCs>
  <outputMessages>
    ...
  </outputMessages>
</getListPSCResponse>
    
```

Failure Output

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

9.1.2.2.1.6 *exists*

- The exists service checks whether the PSC record, as per the criteria in the request, exists in FPDS and returns success or failure response.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
PSCCode	complexType PSCCodeType	PSC.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
existsPSCResponse	complexType existsPSCResponse Type	PSC.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
isExists	xsd:Boolean	True if record exists, or false otherwise.

Example:

```

<existsPSCResponse>
    
```

```

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

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	1. PSC requested does not exist in the database 2. PSC ID entered is not valid

9.1.2.2.1.7 delete

- The PSC.delete method is the API to delete the PSC and return success or failure back to the user.
- In case of failure, the validation results also specify the reason for the error.
- Only FPDS System Administrators may perform this functionality.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	complexType UserAuthenticationKeyType	Common.xsd
PSCCode	complexType PSCCodeType	PSC.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
deletePSCResponse	complexType deletePSCResponse Type	PSC.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
isDeleted	xsd:boolean	True if record successfully deleted, false otherwise.

Example:

```

<deletePSCResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <PSC>
    ...
  </PSC>
  <outputMessages>
    ...
  </outputMessages>
    
```

</deletePSCResponse>

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	<ol style="list-style-type: none"> Cannot delete PSC record. PSC ID is referenced by other records in the system. PSC requested for deletion not found in the database PSC ID entered is not valid

9.1.2.2.1.8 *disable*

- The disable service disables the existing PSC record in FPDS and returns the success or failure of the disabling activity.
- The End Date of the PSC record is set to the current date, upon which the PSC is disabled.
- The service authentication input must contain credentials of a FPDS Administrator or PSC Administrator who is authorized to disable this PSC.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	complexType UserAuthenticationKeyType	Common.xsd
PSCCode	complexType PSCCodeType	PSC.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
disablePSCResponse	complexType disablePSCResponse Type	PSC.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
isDisabled	xsd:boolean	True if record successfully disabled, false otherwise.

Example:

```

<disablePSCResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <PSC>
    ...
  </PSC>
  <outputMessages>
    ...
  </outputMessages>
</disablePSCResponse>
    
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
----------------	----------------	-----------------------

ListOfErrors	complexType listOfErrorsType	1. PSC requested for disabling not found in the database 2. PSC ID entered is not valid
--------------	---------------------------------	--

9.1.2.2.1.9 enable

- The enable service enables the existing PSC record in FPDS and returns the success or failure of the enabling activity.
- The End Date of the PSC record is cleared, upon which the PSC is enabled.
- The service authentication input must contain credentials of a FPDS Administrator or PSC Administrator who is authorized to enable this PSC.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	complexType UserAuthenticationKeyType	Common.xsd
PSCCode	complexType PSCCodeType	PSC.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
enablePSCResponse	complexType enablePSCResponse Type	PSC.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
isEnabled	xsd:Boolean	True if record successfully enabled, false otherwise.

Example:

```

<enablePSCResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <PSC>
    ...
  </PSC>
  <outputMessages>
    ...
  </outputMessages>
</enablePSCResponse>
    
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	1. PSC requested for enabling not found in the database 2. PSC ID entered is not valid

9.1.2.3 Locations

9.1.2.3.1 Country

9.1.2.3.1.1 Service meta-specifications in WSDL

WSDL Parameter	Value
PortType	CountryPortType
Binding	CountryBinding
Soap-binding style	rpc
TargetNameSpace	Country.wsdl

9.1.2.3.1.2 Countries Service methods

Service Type/Name	Service Description
create	The Country.create service creates a new Country in FPDS. This service utilizes the validation services to check record validity before insertion into the FPDS database.
update	The Country.update service updates an existing Country in FPDS. This service utilizes the validation services to check record validity before updating the FPDS database.
get	The Country.get service finds an existing Country Code record in FPDS and then composes and returns the XML representation of the Country.
getList	The Country.getList service finds Country records, which match the input selection criteria. Matching records are returned in the defined XML format.
exists	The Country.exists service checks for the existence of a given Country code.
delete	The Country.delete service deletes the given Country in the FPDS System.
disable	Country.disable service disables the Country record in the system.
enable	Country.enable service enables the Country record in the system.

9.1.2.3.1.3 create

- The Country.create method creates a new Country in FPDS.
- Only FPDS System Administrators may perform this functionality.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	complexType UserAuthenticationKeyType	Common.xsd
Country	complexType countryType	Country.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
createCountryResponse	complexType createCountryResponseType	Country.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
Country	complexType countryType	XML representation of the updated Country record

Example:

```
<createCountryResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <country>
    ...
  </country>
  <outputMessages>
    ...
  </outputMessages>
</createCountryResponse>
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	<ol style="list-style-type: none"> 1. Cannot create Country record. Input Country ID is empty. 2. A Country with the same code already exists 3. Cannot create Country record. Required Country ID is missing. 4. Cannot create Country. End Date cannot be earlier than current date. 5. Not authorized to create Country in the FPDS system

9.1.2.3.1.4 update

- The Country.update method is the API to update and perform modifications to the existing Country.
- Only FPDS System Administrators may perform this functionality.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	complexType UserAuthenticationKeyType	Common.xsd
Country	complexType countryType	Country.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
updateCountryResponse	complexType updateCountryResponseType	Country.wsdl

The output parameters in the response contain the following:

Success Output

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

Country	complexType countryType	XML representation of the updated Country record
---------	----------------------------	--

Example:

```
<updateCountryResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <country>
    ...
  </country>
  <outputMessages>
    ...
  </outputMessages>
</updateCountryResponse>
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	<ol style="list-style-type: none"> 1. A Country with the same code already exists 2. Cannot update Country record. Start Date cannot be earlier than current date. 3. Cannot create Country. End Date cannot be earlier than current date. 4. Not authorized to create Country in the FPDS system

9.1.2.3.1.5 *get*

- The get service retrieves the existing Country information in FPDS and returns the record in the XML format specified by the schema.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	complexType UserAuthenticationKeyType	Common.xsd
CountryKey	complexType CountryKeysType	Country.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
getCountryResponse	complexType getCountryResponseType	Country.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
Country	complexType countryType	Country.xsd

Example:

```

<getCountryResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <country>
    ...
  </country>
  <outputMessages>
  </outputMessages>
</getCountryResponse>
    
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	1. Cannot get Country record. Input Country ID is empty. 2. Country requested not found in the database.

9.1.2.3.1.6 *getList*

- The Country.getList service retrieves the existing Country information that satisfies the specified criteria in the request from FPDS.
- Upon successful retrieval, returns the records in the XML format specified by the schema.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	complexType UserAuthenticationKeyType	Common.xsd
countrySearchCriteria	complexType countrySearchCriteriaType	Country.xsd

Output Parameter

Parameter Name	Parameter Type	Name Space
getListCountryResponse	complexType getListCountryResponseType	Country.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
listOfCountries	complexType listOfCountriesType	List of Countries XML

Example:

```

<getListCountryResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <listOfCountries>
    <country>
      ...
    </country>
  </listOfCountries>
</getListCountryResponse>
    
```

```

    <country>
      ...
    </country>
    ...
  </listOfCountries>
  <outputMessages>
    ...
  </outputMessages>
</getListCountryResponse>

```

Failure Output

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

9.1.2.3.1.7 *exists*

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

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
CountryKey	complexType CountryKeysType	Country.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
existsCountryResponse	complexType existsCountryResponse Type	Country.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
isExists	xsd:Boolean	True if record exists, or false otherwise.

Example:

```

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

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	<ol style="list-style-type: none"> Country requested does not exist in the database Country ID entered is not valid

9.1.2.3.1.8 delete

- The Country.delete method is the API to delete the Country and return success or failure back to the user.
- In case of failure, the validation results also specify the reason for the error.
- Only FPDS System Administrators may perform this functionality.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	complexType UserAuthenticationKeyType	Common.xsd
CountryKey	complexType CountryKeysType	Country.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
deleteCountryResponse	complexType deleteCountryResponse Type	Country.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
isDeleted	xsd:Boolean	True if record successfully deleted, false otherwise.

Example:

```
<deleteCountryResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <delete>
    ...
  </delete>
  <outputMessages>
    ...
  </outputMessages>
</deleteCountryResponse>
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	1. Cannot delete Country record. Country ID is referenced by other records in the system. 2. Country requested for deletion not found in the database 3. Country ID entered is not valid

9.1.2.3.1.9 *disable*

- The disable service disables the existing Country record in FPDS and returns the success or failure of the disabling activity.
- The End Date of the Country record is set to the current date, upon which the Country is disabled.
- Only FPDS System Administrators may perform this functionality.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	complexType UserAuthenticationKeyType	Common.xsd
CountryKey	complexType CountryKeysType	Country.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
disableCountryResponse	complexType disableCountryResponse Type	Country.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
isDisabled	xsd:Boolean	True if record successfully disabled, false otherwise.

Example:

```
<disableCountryResponse>
```

```

<requestNumber>549212</requestNumber>
<confirmationNumber>329743</confirmationNumber>
<disable>
  ...
</disable>
<outputMessages>
  ...
</outputMessages>
</disableCountryResponse>
    
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	1. Country requested for disabling not found in the database 2. Country ID entered is not valid

9.1.2.3.1.10 *enable*

- The enable service enables the existing Country record in FPDS and returns the success or failure of the enabling activity.
- The End Date of the Country record is cleared, upon which the Country is enabled.
- Only FPDS System Administrators may perform this functionality.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	complexType UserAuthenticationKeyType	Common.xsd
CountryKey	complexType CountryKeysType	Country.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
enableCountryResponse	complexType enableCountryResponseType	Country.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
isEnabled	xsd:Boolean	True if record successfully enabled, false otherwise.

Example:

```

<enableCountryResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <enable>
    ...
  </enable >
  <outputMessages>
    ...
  </outputMessages>
</enableCountryResponse>
    
```



```

...
</outputMessages>
</enableCountryResponse>
    
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	1. Country requested for enabling not found in the database 2. Country ID entered is not valid

9.1.2.3.2 Place

9.1.2.3.2.1 Service meta-specifications in WSDL

WSDL Parameter	Value
PortType	PlacePortType
Binding	PlaceBinding
Soap-binding style	rpc
TargetNameSpace	Place.wsdl

9.1.2.3.2.2 Place Service methods

Service Type/Name	Service Description
create	The Place.create service creates a new Place in FPDS. This service utilizes the validation services to check record validity before insertion into the FPDS database.
update	The Place.update service updates an existing Place in FPDS. This service utilizes the validation services to check record validity before updating the FPDS database.
get	The Place.get service finds an existing Place in FPDS and then composes and returns the XML representation of the Place record.
getList	The Place.getList service finds Place records, which match the input selection criteria. Matching records are returned in the defined XML format.
exists	The Place.exists service checks for the existence of a given Place code.
delete	The Place.delete service deletes the given Place code in the FPDS System.
disable	Place.disable service disables the Place record in the system.
enable	Place.enable service enables the Place record in the system.

9.1.2.3.2.3 create

- The Place.create method creates a new Place in FPDS.
- Only FPDS System Administrators may perform this functionality.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	complexType UserAuthenticationKeyType	Common.xsd
Place	complexType PlaceType	Place.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
createPlaceResponse	complexType createPlaceResponse Type	Place.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
Place	complexType placeType	XML representation of the updated Place record

Example:

```

<createPlaceResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <place>
    ...
  </place>
  <outputMessages>
    ...
  </outputMessages>
</createPlaceResponse>
    
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	<ol style="list-style-type: none"> Cannot create Place record. Input Place ID is empty. A Place with the same code already exists Cannot create Place record. Required Place ID is missing. Cannot create Place. End Date cannot be earlier than current date. Not authorized to create Place in the FPDS system.

9.1.2.3.2.4 update

- The Place.update method is the API to update and perform modifications to the existing Place.
- The Place.update method expects only the Place ID, contract ID, and the required information to uniquely identify the Place information pertaining to the user.
- Only FPDS System Administrators may perform this functionality.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	complexType UserAuthenticationKeyType	Common.xsd
Place	complexType PlaceType	Place.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
updatePlaceResponse	complexType updatePlaceResponse Type	Place.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
Place	complexType placeType	XML representation of the updated Place record

Example:

```

<updatePlaceResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <place>
    ...
  </place>
  <outputMessages>
    ...
  </outputMessages>
</updatePlaceResponse>
    
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	<ol style="list-style-type: none"> 1. A Place with the same code already exists 2. Cannot update Place record. Start Date cannot be earlier than current date. 3. Cannot create Place. End Date cannot be earlier than current date. 4. Not authorized to create Place in the FPDS system

9.1.2.3.2.5 *get*

- The get service retrieves the existing Place information in FPDS and returns the record in the XML format specified by the schema.
- The get service also checks for valid authorization of the requesting user before sending back the response.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	complexType UserAuthenticationKeyType	Common.xsd
PlaceKey	complexType PlaceKeysType	Place.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
----------------	----------------	------------

getPlaceResponse	complexType getPlaceResponse Type	Place.wsdl
------------------	--------------------------------------	------------

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
Place	complexType PlaceType	Place.xsd

Example:

```

<getPlaceResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <place>
    ...
  </place>
  <outputMessages>
  </outputMessages>
</getPlaceResponse>
    
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	1. Cannot get Place record. Input Place ID is empty. 2. Place requested not found in the database

9.1.2.3.2.6 *getList*

- The Place.getList service retrieves the existing place information that satisfies the specified criteria in the request, from FPDS and returns the records in the XML format specified by the schema.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	complexType UserAuthenticationKeyType	Common.xsd
placeSearchCriteria	complexType placeSearchCriteriaType	Place.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
getListPlaceResponse	complexType getListPlaceResponseType	Place.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
listOfPlaces	complexType listOfPlacesType	List of Places XML

Example:

```

<getListPlaceResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <listOfPlaces>
    <place>
      ...
    </place>
    <place>
      ...
    </place>
    ...
  </listOfPlaces>
  <outputMessages>
    ...
  </outputMessages>
</getListPlaceResponse>

```

Failure Output

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

9.1.2.3.2.7 *exists*

- It checks whether the Place record, as per the criteria in the request, exists in FPDS and returns success or failure response.
- The exists method also checks for valid authorization of the requesting user before sending back the response.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
PlaceKey	complexType PlaceKeysType	Place.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
existsPlaceResponse	complexType existsPlaceResponse Type	Place.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
isExists	xsd:Boolean	True if record exists, or false otherwise.

Example:

```

<existsPlaceResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>

```

```

<isExists>
  true
</isExists>
<outputMessages>
  ...
</outputMessages>
</existsPlaceResponse>
    
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	<ol style="list-style-type: none"> Place requested does not exist in the database Place ID entered is not valid

9.1.2.3.2.8 *delete*

- The Place.delete method is the API to delete the Place and return success or failure back to the user.
- In case of failure, the validation results also specify the reason for the error.
- Only FPDS System Administrators may perform this functionality.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	complexType UserAuthenticationKeyType	Common.xsd
PlaceKey	complexType PlaceKeysType	Place.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
deletePlaceResponse	complexType deletePlaceResponse Type	Place.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
isDeleted	xsd:Boolean	True if record successfully deleted, false otherwise.

Example:

```

<deletePlaceResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <place>
    ...
  </place>
  <outputMessages>
    ...
  </outputMessages>
</deletePlaceResponse>
    
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	<ol style="list-style-type: none"> Cannot delete Place record. Place ID is referenced by other records in the system. Place requested for deletion not found in the database Place ID entered is not valid

9.1.2.3.2.9 *disable*

- The disable service disables the existing Place record in FPDS and returns the success or failure of the disabling activity.
- The End Date of the Place record to the current date, on which the Place is disabled.
- The service authentication input must contain credentials of a FPDS Administrator or Place Administrator who is authorized to disable this Place.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	complexType UserAuthenticationKeyType	Common.xsd
PlaceKey	complexType PlaceKeysType	Place.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
disablePlaceResponse	complexType disablePlaceResponse Type	Place.wsdl

Success Output

Parameter Name	Parameter Type	Contents
isDisabled	xsd:Boolean	True if record successfully disabled, false otherwise.

Example:

```

<disablePlaceResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <place>
    ...
  </place>
  <outputMessages>
    ...
  </outputMessages>
</disablePlaceResponse>
    
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType	1. Place requested for disabling not found in the

	listOfErrorsType	database 2. Place ID entered is not valid
--	------------------	--

9.1.2.3.2.10 enable

- The enable service enables the existing Place record in FPDS and returns the success or failure of the enabling activity.
- The End Date of the Place record is cleared, on which the Place is enabled.
- The service authentication input must contain credentials of a FPDS Administrator or Place Administrator who is authorized to disable this Place.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	complexType UserAuthenticationKeyType	Common.xsd
PlaceKey	complexType PlaceKeysType	Place.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
enablePlaceResponse	complexType enablePlaceResponse Type	Place.wsdl

Success Output

Parameter Name	Parameter Type	Contents
isEnabled	xsd:Boolean	True if record successfully enabled, false otherwise.

Example:

```

<enablePlaceResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <place>
    ...
  </place>
  <outputMessages>
    ...
  </outputMessages>
</enablePlaceResponse>
    
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	1. Place requested for enabling not found in the database 2. Place ID entered is not valid

9.1.2.3.2.11 State

9.1.2.3.2.12 Service meta-specifications in WSDL

WSDL Parameter	Value
PortType	StatePortType
Binding	StateBinding
Soap-binding style	rpc
TargetNameSpace	State.wsdl

9.1.2.3.2.13 State Service methods

Service Type/Name	Service Description
getList	The State.getList service finds State records which match the input selection criteria. Matching records are returned in the defined XML format.

9.1.2.3.2.14 getList

- The State.getList service retrieves the existing State information that satisfies the specified criteria in the request, from FPDS and returns the records in the XML format specified by the schema.

Input Parameters

Parameter Name	Parameter Type	Name Space
ServiceAuthenticationInput	complexType ServiceAuthenticationType	Common.xsd
searchCriteriaXML	complexType stateSearchCriteriaType	State.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
ResponseXML	complexType getListStateResponseType	State.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
listOfStatesXML	complexType listOfStatesType	List of States XML

Example:

```

<getListStateResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <listOfStates>
    <state>
      ...
    </state>
    <state>
      ...
    </state>
  </listOfStates>
</getListStateResponse>
    
```

```

...
</listOfStates>
<outputMessages>
...
</outputMessages>
</getListStateResponse>
    
```

Failure Output

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

9.1.2.3.2.15 Zip

9.1.2.3.2.16 Service meta-specifications in WSDL

WSDL Parameter	Value
PortType	ZipPortType
Binding	ZipBinding
Soap-binding style	rpc
TargetNameSpace	Zip.wsdl

9.1.2.3.2.17 Zip Service methods

Service Type/Name	Service Description
getList	The Zip.getList service finds Zip records, which match the input selection criteria. Matching records are returned in the defined XML format.

9.1.2.3.2.18 getList

- The Zip.getList service retrieves the existing Zip of performance information that satisfies the specified criteria in the request from FPDS and returns the records in the XML format specified by the schema.

Input Parameters

Parameter Name	Parameter Type	Name Space
ServiceAuthenticationInput	complexType ServiceAuthenticationType	Common.xsd
searchCriteriaXML	complexType ZipSearchCriteriaType	Zip.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
ResponseXML	complexType responseType	Zip.wsdl

The output parameters in the response contain the following:

Success Output

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

listOfZipsXML	complexType listOfZipsType	List of Zips XML
---------------	-------------------------------	------------------

Example:

```
<getListZipResponse>  
  <requestNumber>549212</requestNumber>  
  <confirmationNumber>329743</confirmationNumber>  
  <listOfZipCodes>  
    <Zip>  
      ...  
    </Zip>  
    <Zip>  
      ...  
    </Zip>  
    ...  
  </listOfZipCodes>  
  <outputMessages>  
    ...  
  </outputMessages>  
</getListZipResponse>
```

Failure Output

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

WSDL Parameter	Value
PortType	StatePortType
Binding	StateBinding
Soap-binding style	rpc
TargetNameSpace	State.wsdl

9.1.2.4 System Administration

9.1.2.4.1 User

9.1.2.4.1.1 Service meta-specifications in WSDL

WSDL Parameter	Value
PortType	UserPortType
Binding	UserBinding
Soap-binding style	rpc
TargetNameSpace	User.wsdl

9.1.2.4.1.2 User Service methods

Service Type/Name	Service Description
create	The User.create service creates a new User in FPDS. This service utilizes the validation services to check record validity before insertion into the FPDS database.
update	The User.update service updates an existing User in FPDS. This service utilizes the validation services to check record validity before updating the FPDS database.
get	The User.get service finds an existing User in FPDS and then composes and returns the XML representation of the User.
getList	The User.getList service finds User records, which match the input selection criteria. Matching records are returned in XML format declared in the Schema.
authenticate	The User.authenticate service authenticates the given user.
exists	The User.exists service checks for the existence of a given User code.
enable	User.enable service enables a User whose access is disabled.
disable	User.disable service disables a User's profile to deny access to the system, if required.
authorize	The user.authorize adds authorization capabilities to the User's existing privileges.
getObjectPrivilege	The User.getObjectPrivilege returns the contract privilege data for the given User and Object.
getForgotPassword	The User.getForgotPassword creates a temporary password for the given User.

9.1.2.4.1.3 create

- The User.create method creates a new user in FPDS.
- The service authentication must be a user who is authorized to create a new user.
- Returns success message if created.
- If the user already exists, returns failure message.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	complexType UserAuthenticationKeyType	Common.xsd
User	complexType User	User.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
createUserResponse	complexType createUserResponse Type	User.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
UserXML	complexType userType	XML representation of the updated User record

Example:

```

<createUserResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <user>
    ...
  </user>
  <outputMessages>
    ...
  </outputMessages>
</createUserResponse>
    
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	<ol style="list-style-type: none"> 1. Cannot create User record. Input User ID is empty. 2. A User with the same code already exists 3. Cannot create User record. Required User ID is missing. 4. Cannot create User. End Date cannot be earlier than current date. 5. Not authorized to create User in the FPDS system

9.1.2.4.1.4 update

- The User.update method is the API that updates and performs modifications to the existing User, corrects some errors that do not require journaling, or creates a new User as part of the contract information.
- The User.update method expects only the User ID and Agency ID to uniquely identify the User information pertaining to the User.
- The users are allowed to update their own profiles.
- After any update by the user, privilege changes have to be approved by any Administrator in the administrative chain to take effect.
- Administrators in the administrative chain of the agency and the department are also allowed to make updates to the User.
- In addition to the security check performed at the service controller level, without checking the type of request, this check ensures that authorized personnel make the modifications from the respective organizational hierarchy tree.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	complexType UserAuthenticationKeyType	Common.xsd
User	complexType User	User.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
updateUserResponse	complexType updateUserResponse Type	User.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
User	complexType userType	XML representation of the updated User record

Example:

```

<updateUserResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <user>
    ...
  </user>
  <outputMessages>
    ...
  </outputMessages>
</updateUserResponse>
    
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType	1. A User with the same code already exists

	listOfErrorsType	<ol style="list-style-type: none"> 2. Cannot update User record. Start Date cannot be earlier than current date. 3. Cannot create User. End Date cannot be earlier than current date. 4. Not authorized to create User in the FPDS system
--	------------------	--

9.1.2.4.1.5 *get*

- The get service retrieves the existing User information in FPDS and returns the record in the XML format specified by the schema.
- The get service also checks for valid authorization of the requesting user before sending back the response.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	complexType UserAuthenticationKeyType	Common.xsd
UserID	complexType userIDType	User.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
getUserResponse	complexType getUserResponse Type	User.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
User	complexType userType	XML representation of the updated User record

Example:

```

<getUserResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <user>
    ...
  </user>
  <outputMessages>
  </outputMessages>
</getUserResponse>
    
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	<ol style="list-style-type: none"> 1. Cannot get User record. Input User ID is empty. 2. User requested not found in the database

9.1.2.4.1.6 *getList*

- The getList service retrieves the existing User information that satisfies the specified criteria in the request from the FPDS system.
- On successful retrieval, it returns the records in the XML format specified by the schema.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	complexType UserAuthenticationKeyType	User.xsd
userSearchCriteria	complexType userSearchCriteriaType	User.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
getListUserResponse	complexType getListUserResponseType	User.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
listOfUsers	complexType listOfUsersType	List of Users XML

Example:

```

<getListUserResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <listOfUsers>
    <user>
      ...
    </user>
    <user>
      ...
    </user>
    ...
  </listOfUsers>
  <outputMessages>
    ...
  </outputMessages>
</getListUserResponse>
    
```

Failure Output

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

9.1.2.4.1.7 *authenticate*

- The User.authenticate method is the API to authenticate the User and return success or failure back.
- In case of failure, the error message just says “authentication failed”.
- The validation results must not return specific parameter errors that were the reason for the error.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	complexType UserAuthenticationKeyType	Common.xsd
UserID	complexType userIDType	User.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
authenticateUserResponse	complexType authenticateUserResponse Type	User.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
isAuthenticated	xsd:Boolean	True if authentication successful, false otherwise.

Example:

```
<authenticateUserResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <user>
    ...
  </user>
  <outputMessages>
  </outputMessages>
</authenticateUserResponse>
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	1. Invalid authentication credentials. User is not authorized.

9.1.2.4.1.8 *enable*

- The enable method is a macro level service method that essentially enables the User’s profile in FPDS.
- The service authentication input must contain credentials of a user who is authorized to enable this User.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	complexType UserAuthenticationKeyType	Common.xsd
UserID	complexType userIDType	User.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
enableUserResponse	complexType enableUserResponseType	User.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
isEnabled	xsd:Boolean	True if record enabled, false otherwise.

Example:

```
<enableUserResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <user>
```

```

...
</user>
<outputMessages>
...
</outputMessages>
</enableUserResponse>
    
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	1. User requested for enabling not found in the database 2. User ID entered is not valid

9.1.2.4.1.9 *disable*

- The disable service disables the existing User profile in FPDS and returns the success or failure of the disabling activity.
- The service authentication input must contain credentials of a user who is authorized to disable this User.
- The disable service also checks for valid authorization of the requesting user before sending back the response.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	complexType UserAuthenticationKeyType	Common.xsd
UserID	complexType userIDType	User.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
disableUserResponse	complexType disableUserResponse Type	User.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
isDisabled	xsd:Boolean	True if record is disabled successfully, false otherwise.

Example:

```

<disableUserResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <user>
    ...
  </user>
  <outputMessages>
    ...
    
```

```
</outputMessages>
</disableUserResponse>
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	1. User requested for disabling not found in the database 2. User ID entered is not valid

9.1.2.4.1.10 exists

1. The exists service checks for the existence of the User and sends a binary response back to the requestor.
2. The service authentication input must contain credentials of the user who is allowed to check for the existence of a User.
3. It checks whether a User record, as per the criteria in the request, exists in FPDS and returns success or failure response.
4. The service checks for valid authorization of the requesting user before sending back the response, unless called through another service implementation, namely, create or authenticate.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	complexType UserAuthenticationKeyType	Common.xsd
UserID	complexType userIDType	User.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
existsUserResponse	complexType existsUserResponse Type	User.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
isExists	xsd:Boolean	True if record exists, or false otherwise.

Example:

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

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	<ol style="list-style-type: none"> 1. User requested does not exist in the database 2. User ID entered is not valid
ListOfInformationalMessages	complexType listOfInformationalMessagesType	<ol style="list-style-type: none"> 1. User exists but is disabled in the database

9.1.2.4.1.11 authorize

1. The authorize service adds additional privileges to the User specified in the input.
2. The Service Authentication Input must contain the credentials of a user who is authorized to add additional privileges to the User in the input packet.
3. The service checks for valid authorization of the requesting user before sending back the response, unless called through another service implementation, namely, create or authenticate.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	complexType UserAuthenticationKeyType	Common.xsd
isAuthorizedUserInput	complexType isAuthorizedUserInputType	User.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
authorizeUserResponse	complexType authorizeUserResponse Type	User.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
isAuthorized	xsd:Boolean	True if User is authorized, false otherwise.

Example:

```

<authorizeUserResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <user>
    ...
  </user>
  <outputMessages>
    ...
  </outputMessages>
</authorizeUserResponse>
    
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	<ol style="list-style-type: none"> 1. User requested for authorizing not found in the database

9.1.2.4.1.12 *getObjectPrivilege*

1. The User.getObjectPrivilege returns the contract privilege data for the given User and Object.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	complexType userAuthenticationKeyType	User.xsd
getObjectPrivilegesInput	complexType getObjectPrivilegesInputType	User.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
getObjectPrivilegesResponse	complexType getObjectPrivilegesResponseType	User.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
listOfUserPrivileges	complexType listOfUserPrivilegesType	XML representation of the list of User Privileges.

Example:

```

<getObjectPrivilegesResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <listOfUserPrivileges>
    <userPrivilege>
      ...
    </userPrivilege>
    <userPrivilege>
      ...
    </userPrivilege>
    ...
  </listOfUserPrivileges>
  <outputMessages>
    ...
  </outputMessages>
</getObjectPrivilegesResponse>
    
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	1. Object requested for privileges not found in the database

9.1.2.4.1.13 *getForgotPassword*

1. The User.getForgotPassword creates a new password for the specified User and emails it to the User’s email address.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	complexType userAuthenticationKeyType	User.xsd
getForgotPasswordUserInput	complexType getForgotPasswordUserInputType	User.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
getForgotPasswordResponse	complexType getForgotPasswordResponse Type	User.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
*		

**Note: There is no web services output. Output will only display when the request fails, as shown in the next table.*

Example:

```

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

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	<ol style="list-style-type: none"> 1. The User ID is missing 2. Record does not exist 3. Email ID is not in the records

9.2 GUI Services

9.2.1 Data Collection

9.2.1.1 Contracts

9.2.1.1.1 Contract

9.2.1.1.1.1 Service meta-specifications in WSDL

WSDL Parameter	Value
PortType	ContractPortType
Binding	ContractBinding
Soap-binding style	rpc
TargetNameSpace	Contract.wsdl

9.2.1.1.1.2 Contract Service Methods

Service Type/Name	Service Description
getNewContractURL	Contract.getNewContractURL service gets the FPDS web page URL to be invoked from any external application.
getExistingContractURL	Contract.getExistingContractURL service gets the FPDS web page URL to be invoked for updating an existing Contract from any external application.

9.2.1.1.1.3 getNewContractURL

- The Contract.getNewContractURL method returns the URL to launch the Contract web page with the Contract data provided in the request.
- When accessed from a browser, the web page shows up pre-populated with the Contract data sent in the request.
- This is used by external systems that might want to perform Contract related transactions using the FPDS system directly.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
Contract	complexType ContractType	Contract.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
contractURL	complexType getNewContractURLResponse Type	Contract.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
Contract	complexType ContractType	Contract.xsd

Example:

```

<getNewContractURL>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <contractURL>
    ...
  </contractURL>

```



```
<outputMessages>
...
</outputMessages>
</getNewContractURL>
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	1. User not authorized to perform this function

9.2.1.1.1.4 *getExistingContractURL*

- The Contract.getExistingContractURL method returns the URL to launch the Contract web page screen with the Contract data existing in the database, identified using the ID provided in the request.
- When accessed by a web browser, the web page is pre-populated with the Contract data fetched from the database using the Contract ID send in the request.
- This is used by external systems that might want to perform Contract related transactions using the FPDS system directly.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
ContractID	complexType ContractIDType	Contract.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
ContractURL	complexType getExistingContractURLResponse Type	Contract.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
Contract	complexType ContractType	Contract.xsd

Example:

```
<getExistingContractURLResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <contractURL>
    ...
  </contractURL>
  <outputMessages>
    ...
  </outputMessages>
</getExistingContractURLResponse>
```

9.2.1.1.2 Award

9.2.1.1.2.1 Service meta-specifications in WSDL

WSDL Parameter	Value
PortType	AwardPortType
Binding	AwardBinding
Soap-binding style	rpc
TargetNamespace	Award.wsdl

9.2.1.1.2.2 Award Service methods

Service Type/Name	Service Description
getNewAwardURL	Award.getNewAwardURL service returns the FPDS web page URL to be invoked from any external application.
getExistingAwardURL	Award.getExistingAwardURL service returns the FPDS web page URL to be invoked for updating an existing Award from any external application.
getNewAwardURLFromTemplate	Award.getNewAwardURLFromTemplate service gets the URL to launch Award web page screen with the Award data provided in the request.

9.2.1.1.2.3 getNewAwardURL

- The Award.getNewAwardURL method returns the URL to launch the Award web page with the Award data provided in the request.
- When accessed from a browser, the web page shows up pre-populated with the Award data sent in the request.
- This is used by external systems that might want to perform Award related transactions using the FPDS system directly.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
Award	complexType awardType	Award.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
awardURL	complexType getNewAwardURLResponse Type	Award.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
AwardURL	complexType awardType	URL to invoke Award web page with contents filled in from the request.

Example:

```
<getNewAwardURLResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <awardURL>
    ...
  </awardURL>
  <outputMessages>
    ...
  </outputMessages>
</getNewAwardURLResponse>
```

Failure Output

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

9.2.1.1.2.4 *getExistingAwardURL*

- The Award.getExistingAwardURL method gets the URL to launch the Award web page with the Award 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 Award data fetched from the database using the Award ID sent in the request.
- This is used by external systems that might want to perform Award related transactions using the FPDS system directly.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
AwardID	complexType awardIDType	Award.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
AwardURL	complexType getExistingAwardURLResponse Type	Award.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
Award	complexType awardType	Award.xsd

Example:

```
<getExistingAwardURLResponse>
  <requestNumber>549212</requestNumber>
```

```

    <confirmationNumber>329743</confirmationNumber>
    <awardURL>
      ...
    </awardURL>
    <outputMessages>
      ...
    </outputMessages>
  </getExistingAwardURLResponse>

```

Failure Output

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

9.2.1.1.2.5 *getNewAwardURLFromTemplate*

- The Award.getNewAwardURLFromTemplate method gets the URL to launch Award web page screen with the Award data provided 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 template data and Award data sent in the request.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKeyXML	UserAuthenticationKeyType	User.xsd
getNewAwardURLFromTemplateRequest	complexType getNewAwardURLFromTemplate	Award.wsdl

Output Parameters

Parameter Name	Parameter Type	Name Space
ResponseXML	complexType responseType	Service.xsd

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
AwardURL	complexType awardType	URL to invoke Award web page with contents filled in from the request.

Example:

```

<getNewAwardURLFromTemplateResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <awardURL>
    ...
  </awardURL>
  <outputMessages>
    ...
  </outputMessages>
</getNewAwardURLFromTemplateResponse>

```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	<ol style="list-style-type: none"> 1. An Award with the PIID already exists 2. Modification Number is required 3. PSC Code is not valid for the product

(Note: Only a sample list is provided: Please refer to FPDS Error Listings for complete list of errors)

9.2.1.1.3 IDV

9.2.1.1.3.1 Service meta-specifications in WSDL

WSDL Parameter	Value
PortType	IDVPortType
Binding	IDVBinding
Soap-binding style	rpc
TargetNameSpace	IDV.wsdl

9.2.1.1.3.2 IDV Service methods

Service Type/Name	Service Description
getNewIDVURL	IDV.getNewIDVURL service gets the FPDS web page URL for a new IDV to be invoked from any external application.
getExistingIDVURL	IDV.getExistingIDVURL service gets the FPDS web page URL to be invoked for updating an existing IDV from any external application.
getNewIDVURLFromTemplate	IDV.getNewIDVURLFromTemplate service gets the URL to launch IDV web page screen with the IDV data provided in the request.

9.2.1.1.3.3 getNewIDVURL

- The IDV.getNewIDVURL method returns the URL to launch the IDV web page with the IDV data provided in the request.
- When accessed by a web browser, the web page is pre-populated with the IDV data sent in the request.
- This is used by external systems that might want to perform IDV related transactions using the FPDS system directly.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
IDV	complexType IDVType	IDV.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
IDVURL	complexType getNewIDVURLResponse Type	IDV.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
IDVURL	complexType IDVType	URL to invoke the web page with the IDV contents.

Example:

```

<getNewIDVURLResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <IDVURL>
    ...
  </IDVURL>
  <outputMessages>
    ...
  </outputMessages>
</getNewIDVURLResponse>
    
```

Failure Output

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

9.2.1.1.3.4 *getExistingIDVURL*

- The IDV.getExistingIDVURL method returns the URL to launch the IDV web page with the IDV data retrieved from the database, using the ID provided in the request.
- When accessed by a web browser, the web page is pre-populated with the IDV data fetched from the database using the IDV ID sent in the request.
- This is used by external systems that might want to perform IDV related transactions using the FPDS system directly.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
IDVID	complexType IDVIDType	IDV.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
IDVURL	complexType existsIDVURLResponse Type	IDV.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
IDVURL	complexType IDVType	URL to invoke the web page with the IDV contents pre-populated with values from the database

Example:

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

Failure Output

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

9.2.1.1.3.5 *getNewIDVURLFromTemplate*

- The IDV.getNewIDVURL method gets the URL to launch IDV web page screen with the IDV 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 template data and IDV data send in the request.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKeyXML	UserAuthenticationKeyType	User.xsd
getNewIDVURLFromTemplateRequest	complexType getNewIDVURLFromTemplate	IDV.wsdl

Output Parameters

Parameter Name	Parameter Type	Name Space
ResponseXML	complexType responseType	Service.xsd

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
IDVURL	complexType IDVType	URL to invoke IDV web page with contents filled in from the request.

Example:

```
<getNewIDVURLFromTemplateResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <IDVURL>
    ...
  </IDVURL>
```

```

<outputMessages>
  ...
</outputMessages>
</getNewIDVURLFromTemplateResponse>
    
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	<ol style="list-style-type: none"> 1. An IDV with the PIID Already exists 2. Modification Number is required 3. PSC Code is not valid for the product

(Note: Only a sample list is provided: Please refer to FPDS Error Listings for complete list of errors)

9.2.1.1.4 OtherTransactionAward

9.2.1.1.4.1 Service meta-specifications in WSDL

WSDL Parameter	Value
PortType	OtherTransactionAwardPortType
Binding	OtherTransactionAwardBinding
Soap-binding style	rpc
TargetNameSpace	OtherTransactionAward.wsdl

9.2.1.1.4.2 OtherTransactionAward Service methods

Service Type/Name	Service Description
getNewOtherTransactionAwardURL	OtherTransactionAward.getNewOtherTransactionAwardURL service returns the FPDS web page URL to be invoked from any external application.
getExistingOtherTransactionAwardURL	OtherTransactionAward.getExistingOtherTransactionAwardURL service returns the FPDS web page URL to be invoked for updating an existing OtherTransactionAward from any external application.

9.2.1.1.4.3 getNewOtherTransactionAwardURL

- The OtherTransactionAward.getNewOtherTransactionAwardURL method returns the URL to launch the OtherTransactionAward web page with the OtherTransactionAward data provided in the request.
- When accessed from a browser, the web page shows up pre-populated with the OtherTransactionAward data sent in the request.
- This is used by external systems that might want to perform OtherTransactionAward related transactions using the FPDS system directly.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
OtherTransactionAward	complexType OtherTransactionAwardType	OtherTransactionAward.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
OtherTransactionAwardURL	complexType getNewOtherTransactionAwardURLResponse Type	OtherTransactionAward.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
OtherTransactionAwardURL	complexType OtherTransactionAwardType	URL to invoke OtherTransactionAward web page with contents filled in from the request.

Example:

```

<getNewOtherTransactionAwardURLResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <OtherTransactionAwardURL>
    ...
  </OtherTransactionAwardURL>
  <outputMessages>
    ...
  </outputMessages>
</getNewOtherTransactionAwardURLResponse>
    
```

Failure Output

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

9.2.1.1.4.4 *getExistingOtherTransactionAwardURL*

- The OtherTransactionAward.getExistingOtherTransactionAwardURL method gets the URL to launch the OtherTransactionAward web page with the OtherTransactionAward 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 OtherTransactionAward data fetched from the database using the OtherTransactionAward ID sent in the request.
- This is used by external systems that might want to perform OtherTransactionAward related transactions using the FPDS system directly.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
OtherTransactionAwardID	complexType OtherTransactionAwardIDType	OtherTransactionAward.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
OtherTransactionAwardURL	complexType getExistingOtherTransactionAwardURLResponse Type	OtherTransactionAward.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
OtherTransactionAward	complexType OtherTransactionAwardType	OtherTransactionAward.xsd

Example:

```
<getExistingOtherTransactionAwardURLResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <OtherTransactionAwardURL>
    ...
  </OtherTransactionAwardURL>
  <outputMessages>
    ...
  </outputMessages>
</getExistingOtherTransactionAwardURLResponse>
```

Failure Output

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

9.2.1.1.5 OtherTransactionIDV

9.2.1.1.5.1 Service meta-specifications in WSDL

WSDL Parameter	Value
PortType	OtherTransactionIDVPortType
Binding	OtherTransactionIDVBinding
Soap-binding style	rpc
TargetNameSpace	OtherTransactionIDV.wsdl

9.2.1.1.5.2 OtherTransactionIDV Service methods

Service Type/Name	Service Description
getNewOtherTransactionIDVURL	OtherTransactionIDV.getNewOtherTransactionIDVURL service gets the FPDS web page URL for a new OtherTransactionIDV to be invoked from any external application.
getExistingOtherTransactionIDVURL	OtherTransactionIDV.getExistingOtherTransactionIDVURL service gets the FPDS web page URL to be invoked for

Service Type/Name	Service Description
	updating an existing OtherTransactionIDV from any external application.

9.2.1.1.5.3 *getNewOtherTransactionIDVURL*

- The OtherTransactionIDV.getNewOtherTransactionIDVURL method returns the URL to launch the OtherTransactionIDV web page with the OtherTransactionIDV data provided in the request.
- When accessed by a web browser, the web page is pre-populated with the OtherTransactionIDV data sent in the request.
- This is used by external systems that might want to perform OtherTransactionIDV related transactions using the FPDS system directly.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
OtherTransactionIDV	complexType OtherTransactionIDVType	OtherTransactionIDV.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
OtherTransactionIDVURL	complexType getNewOtherTransactionIDVURLResponse Type	OtherTransactionIDV.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
OtherTransactionIDVURL	complexType OtherTransactionIDVType	URL to invoke the web page with the OtherTransactionIDV contents.

Example:

```

<getNewOtherTransactionIDVURLResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <OtherTransactionIDVURL>
    ...
  </OtherTransactionIDVURL>
  <outputMessages>
    ...
  </outputMessages>
</getNewOtherTransactionIDVURLResponse>
    
```

Failure Output

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

9.2.1.1.5.4 *getExistingOtherTransactionIDVURL*

- The OtherTransactionIDV.getExistingOtherTransactionIDVURL method returns the URL to launch the OtherTransactionIDV web page with the OtherTransactionIDV data retrieved from the database, using the ID provided in the request.
- When accessed by a web browser, the web page is pre-populated with the OtherTransactionIDV data fetched from the database using the OtherTransactionIDV ID send in the request.
- This is used by external systems that might want to perform OtherTransactionAward related transactions using the FPDS system directly.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
OtherTransactionIDVID	complexType OtherTransactionIDVIDType	OtherTransactionIDV.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
OtherTransactionIDVURL	complexType existsOtherTransactionIDVURLResponse Type	OtherTransactionIDV.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
OtherTransactionIDVURL	complexType OtherTransactionIDVType	URL to invoke the web page with the OtherTransactionIDV contents pre-populated with values from the database

Example:

```

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

Failure Output

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

9.2.1.2 Organizations

9.2.1.2.1 Department

9.2.1.2.1.1 Service meta-specifications in WSDL

WSDL Parameter	Value
PortType	DepartmentPortType
Binding	DepartmentBinding
Soap-binding style	rpc
TargetNameSpace	Department.wsdl

9.2.1.2.1.2 Departments Service Methods

Service Type/Name	Service Description
getNewDepartmentURL	Department.getNewDepartmentURL service returns the FPDS web page URL to be invoked from any external application.
getExistingDepartmentURL	Department.getExistingDepartmentURL service returns the FPDS web page URL to be invoked for updating an existing Department from any external application.

9.2.1.2.1.3 getNewDepartmentURL

- The Department.getNewDepartmentURL method returns the URL to launch the Department web page screen with the Department data provided in the request.
- When accessed from a browser, the web page shows up pre-populated with the Department data sent in the request.
- This is used by external systems that might want to perform Department lookup data related transactions using the FPDS system directly.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
Department	complexType DepartmentType	Department.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
DepartmentURL	complexType getNewDepartmentURLResponse Type	Department.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
Department	complexType URLType	URL to invoke the Department web page with contents filled in from the request.

Example:

<getNewDepartmentURLResponse>

```

<requestNumber>549212</requestNumber>
<confirmationNumber>329743</confirmationNumber>
<departmentURL>
  ...
</departmentURL>
<outputMessages>
  ...
</outputMessages>
</getNewDepartmentURLResponse>
    
```

Failure Output

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

9.2.1.2.1.4 *getExistingDepartmentURL*

- The Department.getExistingDepartmentURL method returns the URL to launch the Department web page with the Department data existing in the database, identified using the ID provided in the request.
- When accessed by a web browser, the web page is pre-populated with the Department data fetched from the database using the Department ID sent in the request.
- This is used by the external systems that want to perform Department related transactions using the FPDS system directly.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
DepartmentID	complexType DepartmentIDType	Department.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
DepartmentURL	complexType getExistingDepartmentURLResponse Type	Department.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
DepartmentURL	complexType URLType	URL to invoke Department web page with contents filled in from the record fetched from the FPDS database

Example:

```

<getExistingDepartmentURLResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
    
```

```

<departmentURL>
...
</departmentURL>
<outputMessages>
...
</outputMessages>
</getExistingDepartmentURLResponse>
    
```

Failure Output

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

9.2.1.2.2 Agency

9.2.1.2.2.1 Service meta-specifications in WSDL

WSDL Parameter	Value
PortType	AgencyPortType
Binding	AgencyBinding
Soap-binding style	rpc
TargetNameSpace	Agency.wsdl

9.2.1.2.2.2 Agency Service methods

Service Type/Name	Service Description
getNewAgencyURL	Agency.getNewAgencyURL service returns the FPDS web page URL to be invoked from any external application.
getExistingAgencyURL	Agency.getExistingAgencyURL service returns the FPDS web page URL to be invoked for updating an existing Agency from any external application.

9.2.1.2.2.3 getNewAwardURL

- The Agency.getNewAgencyURL method returns the URL to launch Agency web page with the Agency data provided in the request.
- When accessed from a browser, the web page shows up pre-populated with the Agency data sent in the request.
- This is used by external systems that might want to perform Agency lookup data related transactions using the FPDS system directly.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
Agency	complexType AgencyType	Agency.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
AgencyURL	complexType getNewAgencyURLResponse Type	Agency.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
AgencyURL	complexType URLType	URL to invoke the Agency web page with the contents filled in from the request.

Example:

```

<getNewAgencyURLResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <agencyURL>
    ...
  </agencyURL>
  <outputMessages>
    ...
  </outputMessages>
</getNewAgencyURLResponse>
    
```

Failure Output

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

9.2.1.2.2.4 *getExistingAgencyURL*

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

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
AgencyID	complexType AgencyIDType	Agency.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
AgencyURL	complexType getExistingAgencyURLResponse Type	Agency.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
AgencyURL	complexType URLType	URL to invoke Agency web page with

		contents filled in from the record fetched from the FPDS database
--	--	---

Example:

```
<getExistingAgencyURLResponse>  
  <requestNumber>549212</requestNumber>  
  <confirmationNumber>329743</confirmationNumber>  
  <agencyURL>  
    ...  
  </agencyURL>  
  <outputMessages>  
    ...  
  </outputMessages>  
</getExistingAgencyURLResponse>
```

Failure Output

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

9.2.1.2.3 Contracting Office

9.2.1.2.3.1 Service meta-specifications in WSDL

WSDL Parameter	Value
PortType	ContractingOfficePortType
Binding	ContractingOfficeBinding
Soap-binding style	rpc
TargetNameSpace	ContractingOffice.wsdl

9.2.1.2.3.2 Contracting Office Service methods

Service Type/Name	Service Description
getNewContractingOfficeURL	ContractingOffice.getNewContractingOfficeURL service gets the FPDS web page URL to be invoked from any external application.
getExistingContractingOfficeURL	ContractingOffice.getExistingContractingOfficeURL service gets the FPDS web page URL to be invoked for updating an existing Contracting Office from any external application.

9.2.1.2.3.3 getNewContractingOfficeURL

- The ContractingOffice.getURLNew method gets the URL to launch the Contracting Office web page with the ContractingOffice data provided in the request.
- When accessed from a browser, the web page shows up pre-populated with the Contracting Office data sent in the request.
- This is used by external systems that might want to perform Contracting Office lookup data related transactions using the FPDS system directly.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
ContractingOffice	complexType ContractingOfficeType	ContractingOffice.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
ContractingOfficeURL	complexType getNewContractingOfficeURLResponse Type	ContractingOffice.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
ContractingOfficeURL	complexType URLType	URL to invoke ContractingOffice web page with contents filled in from the request.

Example:

```

<getNewContractingOfficeURLResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <contractingOfficeURL>
    ...
  </contractingOfficeURL>
  <outputMessages>
    ...
  </outputMessages>
</getNewContractingOfficeURLResponse>
    
```

Failure Output

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

9.2.1.2.3.4 *getExistingContractingOfficeURL*

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

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
ContractingOfficeKey	complexType ContractingOfficeKeyType	ContractingOffice.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
ContractingOfficeURL	complexType getExistingContractingOfficeURLResponse Type	ContractingOffice.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
ContractingOfficeURL	complexType URLType	URL to invoke Contracting Office web page with contents filled in from the record fetched from the FPDS database

Example:

```
<getExistingContractingOfficeURLResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <contractingOfficeURL>
    ...
  </contractingOfficeURL>
  <outputMessages>
    ...
  </outputMessages>
</getExistingContractingOfficeURLResponse>
```

Failure Output

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

9.2.1.2.4 Government Office

9.2.1.2.4.1 Service meta-specifications in WSDL

WSDL Parameter	Value
PortType	GovernmentOfficePortType
Binding	GovernmentOfficeBinding
Soap-binding style	rpc
TargetNameSpace	GovernmentOffice.wsdl

9.2.1.2.4.2 Government Office Service methods

Service Type/Name	Service Description
getNewGovernmentOfficeURL	GovernmentOffice.getNewGovernmentOfficeURL service gets the FPDS web page URL to be invoked from any external application.
getExistingGovernmentOfficeURL	GovernmentOffice.getExistingGovernmentOfficeURL service gets the FPDS web page URL to be invoked for updating an existing GovernmentOffice from any external application.

9.2.1.2.4.3 getNewGovernmentOfficeURL

- The GovernmentOffice.getNewGovernmentOfficeURL method gets the URL to launch the Government Office web page with the GovernmentOffice data provided in the request.
- When accessed from a browser, the web page shows up pre-populated with the Government Office data sent in the request.
- This is used by external systems that might want to perform Government Office lookup data related transactions using the FPDS system directly.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd

GovernmentOffice	complexType GovernmentOfficeType	GovernmentOffice.xsd
------------------	-------------------------------------	----------------------

Output Parameters

Parameter Name	Parameter Type	Name Space
GovernmentOfficeURL	complexType getNewGovernmentOfficeURLResponse Type	GovernmentOffice.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
GovernmentOfficeURL	complexType URLType	URL to invoke Government Office web page with contents filled in from the request.

Example:

```

<getNewGovernmentOfficeURLResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <governmentOffice>
    ...
  </governmentOffice>
  <outputMessages>
    ...
  </outputMessages>
</getNewGovernmentOfficeURLResponse>
    
```

Failure Output

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

9.2.1.2.4.4 *getExistingGovernmentOfficeURL*

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

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
GovernmentOfficeKey	complexType GovernmentOfficeKeyType	GovernmentOffice.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
GovernmentOfficeURL	complexType getExistingGovernmentOfficeURLResponse Type	GovernmentOffice.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
GovernmentOfficeURL	complexType URLType	URL to invoke Government Office web page with contents filled in from the record fetched from the FPDS database

Example:

```

<getExistingGovernmentOfficeURLResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <governmentOffice>
    ...
  </governmentOffice>
  <outputMessages>
    ...
  </outputMessages>
</getExistingGovernmentOfficeURLResponse>
    
```

Failure Output

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

9.2.1.3 Locations

9.2.1.3.1 Country

9.2.1.3.1.1 Service meta-specifications in WSDL

WSDL Parameter	Value
PortType	CountryPortType
Binding	CountryBinding
Soap-binding style	rpc
TargetNameSpace	Country.wsdl

9.2.1.3.1.2 Countries Service methods

Service Type/Name	Service Description
getNewCountryURL	Country.getNewCountryURL service gets the FPDS web page URL to be invoked from any external application.
getExistingCountryURL	Country.getExistingCountryURL service gets the FPDS web page URL to be invoked for updating an existing Country from any external application.

9.2.1.3.1.3 *getNewCountryURL*

- The Country.getNewCountryURL method gets the URL to launch Country web page with the Country data provided in the request.
- When accessed from a browser, the web page shows up pre-populated with the Country data sent in the request.
- This is used by external systems that might want to perform Country lookup data related transactions using the FPDS system directly.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
Country	complexType CountryType	Country.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
CountryURL	complexType getNewCountryURLResponse Type	Country.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
CountryURL	complexType URLType	URL to invoke Country web page with contents filled in from the request.

Example:

```

<getNewCountryURLResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <countryURL>
    ...
  </countryURL>
  <outputMessages>
    ...
  </outputMessages>
</getNewCountryURLResponse>
    
```

Failure Output

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

	listOfErrorsType	
--	------------------	--

9.2.1.3.1.4 *getExistingCountryURL*

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

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
CountryKey	complexType CountryKeysType	Country.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
CountryURL	complexType getExistingCountryURLResponse Type	Country.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
CountryURL	complexType URLType	URL to invoke Country web page with contents filled in from the record fetched from the FPDS database

Example:

```

<getExistingCountryURLResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <countryURL>
    ...
  </countryURL>
  <outputMessages>
    ...
  </outputMessages>
</getExistingCountryURLResponse>
    
```

Failure Output

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

9.2.1.3.2 Place

9.2.1.3.2.1 Service meta-specifications in WSDL

WSDL Parameter	Value
PortType	PlacePortType
Binding	PlaceBinding
Soap-binding style	rpc
TargetNameSpace	Place.wsdl

9.2.1.3.2.2 Place Service methods

Service Type/Name	Service Description
getNewPlaceURL	Place.getNewPlaceURL service gets the FPDS web page URL to be invoked from any external application.
getExistingPlaceURL	Place.getExistingPlaceURL service gets the FPDS web page URL to be invoked for updating an existing Place from any external application.

9.2.1.3.2.3 getNewPlaceURL

- The Place.getNewPlaceURL method gets the URL to launch Place web page with the Place data provided in the request.
- When accessed from a browser, the web page shows up pre-populated with the Place data sent in the request.
- This is used by external systems that might want to perform Place lookup data related transactions using the FPDS system directly.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
Place	complexType PlaceType	Place.xsd

Output Parameter

Parameter Name	Parameter Type	Name Space
PlaceURL	complexType getNewPlaceURLResponse Type	Place.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
PlaceURL	complexType URLType	URL to invoke Place web page with contents filled in from the request.

Example:

```
<getNewPlaceURLResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
```

```

    <placeURL>
      ...
    </placeURL>
    <outputMessages>
      ...
    </outputMessages>
  </getNewPlaceURLResponse>

```

Failure Output

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

9.2.1.3.2.4 *getExistingPlaceURL*

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

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
PlaceKey	complexType PlaceKeyType	Place.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
PlaceURL	complexType getExistingPlaceURLResponse Type	Place.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
PlaceURL	complexType URLType	URL to invoke Place web page with contents filled in from the record fetched from the FPDS database

Example:

```

<getExistingPlaceURLResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <placeURL>
    ...
  </placeURL>
  <outputMessages>
    ...

```

</outputMessages>
</getExistingPlaceURLResponse>

Failure Output

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

9.2.1.4 Product/Service Classifications

9.2.1.4.1 NAICS

9.2.1.4.1.1 *Service meta-specifications in WSDL*

WSDL Parameter	Value
PortType	NAICSPortType
Binding	NAICSBinding
Soap-binding style	rpc
TargetNameSpace	NAICS.wsdl

9.2.1.4.1.2 *NAICS Service methods*

Service Type/Name	Service Description
getNewNAICSURL	NAICS.getNewNAICSURL service gets the FPDS web page URL to be invoked from any external application.
getExistingNAICSURL	NAICS.getExistingNAICSURL service gets the FPDS web page URL to be invoked for updating an existing NAICS from any external application.

9.2.1.4.1.3 *getNewNAICSURL*

- The NAICS.getNewNAICSURL method gets the URL to launch NAICS web page with the NAICS data provided in the request.
- When accessed from a browser, the web page shows up pre-populated with the NAICS data sent in the request.
- This is used by external systems that might want to perform NAICS lookup data related transactions using the FPDS system directly.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
NAICS	complexType NAICSType	NAICS.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
NAICSURL	complexType getNewNAICSURLResponse Type	NAICS.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
NAICSURL	complexType URLType	URL to invoke NAICS web page with contents filled in from the request.

Example:

```
<getNewNAICSURLResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <NAICSURL>
    ...
  </NAICSURL>
  <outputMessages>
    ...
  </outputMessages>
</getNewNAICSURLResponse>
```

Failure Output

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

9.2.1.4.1.4 *getExistingNAICSURL*

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

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
NAICSCodeXML	complexType NAICSCodeType	NAICS.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
NAICSURL	complexType getExistingNAICSURLResponse Type	NAICS.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
NAICSURL	complexType URLType	URL to invoke NAICS web page with contents filled in from the record fetched

		from the FPDS database
--	--	------------------------

Example:

```

<getExistingNAICSURLResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <NAICSURL>
    ...
  </NAICSURL>
  <outputMessages>
    ...
  </outputMessages>
</getExistingNAICSURLResponse>
    
```

Failure Output

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

9.2.1.4.2 PSC

9.2.1.4.2.1 Service meta-specifications in WSDL

WSDL Parameter	Value
PortType	PSCPortType
Binding	PSCBinding
Soap-binding style	rpc
TargetNameSpace	PSC.wsdl

9.2.1.4.2.2 PSC Service methods

Service Type/Name	Service Description
getNewPSCURL	PSC.getNewPSCURL service gets the FPDS web page URL to be invoked from any external application.
getExistingPSCURL	PSC.getExistingPSCURL service gets the FPDS web page URL to be invoked for updating an existing PSC from any external application.

9.2.1.4.2.3 getNewPSCURL

- The PSC.getNewPSCURL method gets the URL to launch the PSC web page with the PSC data provided in the request.
- When accessed from a browser, the web page shows up pre-populated with the PSC data sent in the request.
- This is used by external systems that might want to perform PSC lookup data related transactions using the FPDS system directly.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd

PSC	complexType PSCType	PSC.xsd
-----	---------------------	---------

Output Parameters

Parameter Name	Parameter Type	Name Space
PSCURL	complexType getNewPSCURLResponse Type	PSC.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
PSCURL	complexType URLType	URL to invoke PSC web page with contents filled in from the request.

Example:

```

<getNewPSCURLResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <PSCURL>
    ...
  </PSCURL>
  <outputMessages>
    ...
  </outputMessages>
</getNewPSCURLResponse>
    
```

Failure Output

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

9.2.1.4.2.4 *getExistingPSCURL*

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

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
PSCCode	complexType PSCCodeType	PSC.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
PSCURL	complexType getExistingPSCURLResponse	PSC.wsdl

	Type	
--	------	--

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
PSCURL	complexType URLType	URL to invoke PSC web page with contents filled in from the record fetched from the FPDS database

Example:

```

<getExistingPSCURLResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <PSCURL>
    ...
  </PSCURL>
  <outputMessages>
    ...
  </outputMessages>
</getExistingPSCURLResponse>
    
```

Failure Output

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

9.2.1.5 System Administration

9.2.1.5.1 User

9.2.1.5.1.1 *Service meta-specifications in WSDL*

WSDL Parameter	Value
PortType	UserPortType
Binding	UserBinding
Soap-binding style	rpc
TargetNameSpace	User.wsdl

9.2.1.5.1.2 *User Service methods*

Service Type/Name	Service Description
getNewUserURL	User.getNewUserURL service gets the FPDS web page URL to be invoked from any external application.
getExistingUserURL	User.getExistingUserURL service gets the FPDS web page URL to be invoked for updating an existing User from any external application.

9.2.1.5.1.3 *getNewUserURL*

- The User.getNewUserURL method gets the URL to launch User web page with the User data provided in the request.
- When accessed from a browser, the web page shows up pre-populated with the User data sent in the request.
- This is used by external systems that want to perform User lookup data related transactions using the FPDS system directly.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
User	complexType UserType	User.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
UserURL	complexType getNewUserURLResponse Type	User.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
UserURL	complexType URLType	URL to invoke User web page with contents filled in from the request.

Example:

```

<getNewUserURLResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <UserURL>
    ...
  </UserURL>
  <outputMessages>
    ...
  </outputMessages>
</getNewUserURLResponse>
    
```

Failure Output

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

9.2.1.5.1.4 *getExistingUserURL*

- The User.getExistingUserURL method gets the URL to launch the User web page with the User data existing in the database, identified using the ID provided in the request.
- When accessed by a web browser, the web page is pre-populated with the User data fetched from the database using the User ID sent in the request.

- This is used by external systems that might want to perform User related transactions using the FPDS system directly.

Input Parameters

Parameter Name	Parameter Type	Name Space
AuthenticationKey	UserAuthenticationKeyType	User.xsd
UserID	complexType UserIDType	User.xsd

Output Parameters

Parameter Name	Parameter Type	Name Space
UserURL	complexType getExistingUserURLResponse Type	User.wsdl

The output parameters in the response contain the following:

Success Output

Parameter Name	Parameter Type	Contents
UserURL	complexType URLType	URL to invoke User web page with contents filled in from the record fetched from the FPDS database

Example:

```

<getExistingUserURLResponse>
  <requestNumber>549212</requestNumber>
  <confirmationNumber>329743</confirmationNumber>
  <UserURL>
    ...
  </UserURL>
  <outputMessages>
    ...
  </outputMessages>
</getExistingUserURLResponse>
    
```

Failure Output

Parameter Name	Parameter Type	Sample Error Contents
ListOfErrors	complexType listOfErrorsType	1. User does not have the 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	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
NAICS	North American Industry Classification System codes
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

[Data Element Dictionary]

GSA FPDSNG Data Element Dictionary #GS00M02PDR0008

[FPDS-RFP]

Request for Proposal (RFP) #GS00M02PDR0008

[FPDSGCE Proposal Volume I]

GCE Technical Proposal - Option 1, submitted to GSA.

[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/.](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.>)