Investment Adviser Public Disclosure
Representative Compilation Reports - XML Guide

Release 2022.5

Table of Contents

Schema Reference Legend

IAPD Individual Compilation Report Xml Schema Reference

Investment Adviser Representatives Report

The information that appears in this file is collected from individual investment adviser representatives, investment adviser firm(s), and/or securities regulators(s) as part of the securities industry's registration and licensing process. Details include each individual's professional background and conduct, current registrations, employment history, and an indication if a disclosure exists for certain disciplinary events involving the individual.

The report is available for download as an XML file. This document provides the description for all the fields/elements included in the XML schema.

Table of Contents

Example - IAPD Individual Compilation Report

```
<? xml version = " 1.0 " encoding = " iso-8859-1 " ?>
< IAPDIndividualReport GenOn = " 1967-08-13 " >
    < Indvls >
        < Indvl >
            < Info lastNm = " LAST NAME XXXX " firstNm = " FIRST NAME XXX " midNm = " MID NAME " sufNm = " JR " indvlPK = " 1234567891 " actvAGReg = " Y " link = " http://www.adviserinfo.sec.gov/IAPD/Individual/1234567891 " />
            < OthrNms >
                < OthrNm lastNm = " AAA " firstNm = " AAA " midNm = " AAA " sufNm = " A " />
            </ OthrNms >
            < CrntEmps >
                < CrntEmp orgNm = " XXXX LLC " orgPK = " 1234567891 " str1 = " 999 XXX " str2 = " XXXX " city = " XXX " state = " AK " cntry = " XXX " postlCd = " 12345 " >
                    < CrntRgstns >
                        < CrntRgstn regAuth = " AK " regCat = " RA " st = " APPROVED " stDt = " 1967-08-13 " />
                        < CrntRgstn regAuth = " TX " regCat = " RA " st = " APPROVED " stDt = " 1967-08-13 " />
                    </ CrntRgstns >
                    < BrnchOfLocs >
                        < BrnchOfLoc str1 = " 999 XXX " str2 = " XXX " city = " XXX " state = " AK " cntry = " XXX " postlCd = " 12345 " />
                        < BrnchOfLoc str1 = " 999 XXX " str2 = " XXX " city = " XXX " state = " TX " cntry = " XXX " postlCd = " 12345 " />
                    </ BrnchOfLocs >
                </ CrntEmp >
                < CrntEmp orgNm = " YYYY LLC " orgPK = " 1234567899 " str1 = " 999 XXX " str2 = " XXXX " city = " XXX " state = " AK " cntry = " XXX " postlCd = " 12345 " >
                    < CrntRgstns >
                        < CrntRgstn regAuth = " AK " regCat = " RA " st = " APPROVED " stDt = " 1967-08-13 " />
                    </ CrntRgstns >
                    < BrnchOfLocs >
                        < BrnchOfLoc str1 = " 999 XXX " str2 = " XXX " city = " XXX " state = " AK " cntry = " XXX " postlCd = " 12345 " />
                    </ BrnchOfLocs >
                </ CrntEmp >
            </ CrntEmps >
            < Exms >
                < Exm exmCd = " S63 " exmNm = " S63 XXXX " exmDt = " 1967-08-13 " />
                < Exm exmCd = " S66 " exmNm = " S66 XXXX " exmDt = " 1967-08-13 " />
            </ Exms >
            < Dsgntns >
                < Dsgntn dsgntnNm = " XXXXX " />
                < Dsgntn dsgntnNm = " XXXXX " />
            </ Dsgntns >
            < PrevRgstns >
                < PrevRgstn orgNm = " XXX LLC " orgPK = " 1234567891 " regBeginDt = " 1967-08-13 " regEndDt = " 1967-08-13 " >
                    < BrnchOfLocs >
                        < BrnchOfLoc city = " XXX " state = " AK " />
                        < BrnchOfLoc city = " XXX " state = " AL " />
                    </ BrnchOfLocs >
                </ PrevRgstn >
            </ PrevRgstns >
            < EmpHss >
                < EmpHs fromDt = " 99/9999 " toDt = " 99/9999 " orgNm = " XXX LLC " city = " XXXX " state = " AK " />
                < EmpHs fromDt = " 99/9999 " toDt = " 99/9999 " orgNm = " XXX LLC " city = " XXX " state = " AK " />
            </ EmpHss >
            < OthrBuss >
                < OthrBus desc = " XXXX " />
            </ OthrBuss >
            < DRPs >
                < DRP hasRegAction = " Y " hasCriminal = " N " hasBankrupt = " N " hasCivilJudc = " N " hasBond = " N " hasJudgment = " Y " hasInvstgn = " Y " hasCustComp = " N " hasTermination = " N " />
            </ DRPs >
        </ Indvl >
        < Indvl >
            < Info lastNm = " LAST NAME XXXX " firstNm = " FIRST NAME XXX " midNm = " MID NAME " sufNm = " JR " indvlPK = " 999999 " actvAGReg = " Y " link = " http://www.adviserinfo.sec.gov/IAPD/Individual/999999 " />
            < OthrNms >
                < OthrNm lastNm = " AAA " firstNm = " AAA " midNm = " AAA " sufNm = " A " />
            </ OthrNms >
            < CrntEmps >
                < CrntEmp orgNm = " XXXX LLC " orgPK = " 1234567891 " str1 = " 999 XXX " str2 = " XXXX " city = " XXX " state = " AK " cntry = " XXX " postlCd = " 12345 " >
                    < CrntRgstns >
                        < CrntRgstn regAuth = " AK " regCat = " RA " st = " APPROVED " stDt = " 1967-08-13 " />
                        < CrntRgstn regAuth = " TX " regCat = " RA " st = " APPROVED " stDt = " 1967-08-13 " />
                    </ CrntRgstns >
                    < BrnchOfLocs >
                        < BrnchOfLoc str1 = " 999 XXX " str2 = " XXX " city = " XXX " state = " AK " cntry = " XXX " postlCd = " 12345 " />
                        < BrnchOfLoc str1 = " 999 XXX " str2 = " XXX " city = " XXX " state = " TX " cntry = " XXX " postlCd = " 12345 " />
                    </ BrnchOfLocs >
                </ CrntEmp >
                < CrntEmp orgNm = " YYYY LLC " orgPK = " 1234567899 " str1 = " 999 XXX " str2 = " XXXX " city = " XXX " state = " AK " cntry = " XXX " postlCd = " 12345 " >
                    < CrntRgstns >
                        < CrntRgstn regAuth = " AK " regCat = " RA " st = " APPROVED " stDt = " 1967-08-13 " />
                    </ CrntRgstns >
                    < BrnchOfLocs >
                        < BrnchOfLoc str1 = " 999 XXX " str2 = " XXX " city = " XXX " state = " AK " cntry = " XXX " postlCd = " 12345 " />
                    </ BrnchOfLocs >
                </ CrntEmp >
            </ CrntEmps >
            < Exms >
                < Exm exmCd = " S63 " exmNm = " S63 XXXX " exmDt = " 1967-08-13 " />
                < Exm exmCd = " S66 " exmNm = " S66 XXXX " exmDt = " 1967-08-13 " />
            </ Exms >
            < Dsgntns >
                < Dsgntn dsgntnNm = " XXXXX " />
                < Dsgntn dsgntnNm = " XXXXX " />
            </ Dsgntns >
            < PrevRgstns >
                < PrevRgstn orgNm = " XXX LLC " orgPK = " 1234567891 " regBeginDt = " 1967-08-13 " regEndDt = " 1967-08-13 " >
                    < BrnchOfLocs >
                        < BrnchOfLoc city = " XXX " state = " AK " />
                        < BrnchOfLoc city = " XXX " state = " AL " />
                    </ BrnchOfLocs >
                </ PrevRgstn >
            </ PrevRgstns >
            < EmpHss >
                < EmpHs fromDt = " 99/9999 " toDt = " 99/9999 " orgNm = " XXX LLC " city = " XXXX " state = " AK " />
            </ EmpHss >
            < OthrBuss >
                < OthrBus desc = " XXXX " />
            </ OthrBuss >
            < DRPs >
                < DRP hasRegAction = " Y " hasCriminal = " N " hasBankrupt = " N " hasCivilJudc = " N " hasBond = " N " hasJudgment = " Y " hasInvstgn = " Y "
                hasCustComp = " N " hasTermination = " N " />
            </ DRPs >
        </ Indvl >
    </ Indvls >
</ IAPDIndividualReport >
```

Schema Reference Legend

How to Read Schema Reference Pages

The following "Address Type" schema element is provided with annotations that explain the layout and content of the actual schema elements and types in the remainder of the *Schema Reference* sections of this document.

Complex Type:	**BrnchOfLocType**
Schema Component Type	Schema Component Name

Super-types: None

Sub-types:

- BrnchOfLocType (by extension)

If this schema component is a type definition, its type hierarchy is shown in a gray-bordered box.

Name	BrnchOfLocType
Abstract	no

The table above displays the properties of this schema component.

XML Instance Representation

```
<...
 Str1="Str50[0..1]"
 Str2="Str50[0..1]"
 City="Str50[0..1]"
 State="StateCdType[0..1]"
 Cntry="Str50[0..1]"
 PostlCd="Str11[0..1]" />
```

The XML Instance Representation table above shows the schema component's content as an XML instance.

- The minimum and maximum occurrence of elements and attributes are provided in square brackets, e.g. [0..1].
- Model group information are shown in gray, e.g. Start Choice ... End Choice.
- For type derivations, the elements and attributes that have been added to or changed from the base type's content are shown in **bold**.
- If an element/attribute has a fixed value, the fixed value is shown in green.
- Otherwise, the type of the element/attribute is displayed.
    - If the element/attribute's type is in the schema, a link is provided to it.
    - For local simple type definitions, the constraints are displayed in angle brackets.

Schema Component Representation

```
<xsd:complexType name="BrnchOfLocType">
  <xsd:attribute name="Str1" type="Str50" use="optional"/>
  <xsd:attribute name="Str2" type="Str50" use="optional"/>
  <xsd:attribute name="City" type="Str50" use="optional"/>
  <xsd:attribute name="State" type="StateCdType" use="optional"/>
  <xsd:attribute name="Cntry" type="Str50" use="optional"/>
  <xsd:attribute name="PostlCd" type="Str11" use="optional"/>
</xsd:complexType>
```

The Schema Component Representation table above displays the underlying XML representation of the schema component. (Annotations are not shown.)

Table of Contents

IAPD Indvl Compilation Report Xml Schema Reference

Schema Properties

Documentation	©2013 Financial Industry Regulatory Authority, Inc. (FINRA). All rights reserved. Materials may not be reprinted or republished without the express permission of FINRA. This document contains FINRA Confidential and Proprietary information. FINRA provides this information for internal assessment or use of the Investment Adviser Compilation Reports available through IAPD. Any other use is strictly prohibited by FINRA. FINRA reserves the right to seek all injunctive and equitable relief available to it in the event FINRA Confidential or Proprietary information is released to a third party. Use of this document demonstrates acknowledgement that this document contains FINRA Confidential and Proprietary information, agreement that the user will not reprint, republish or otherwise disclose this information to any third party and agreement that FINRA may protect its rights, including but not limited to intellectual property rights. Version: Release 2020.03 (last updated on 09/05/2013)

Global Declarations

Element: IAPDIndividualReport

Name	IAPDIndividualReport
Type	IAPDIndividualReportType
Nillable	no
Abstract	no
Documentation	The compilation Report contains information for any individual that currently hold an approved equivalent registration status with at least one Regulator in the IAPD compilation.

XML Instance Representation

```
<IAPDIndividualReport

<!-- This node contains the report generated date information. -->
GenOn="xsd:date[1]">
   <Indvls> IndvlsType </Indvls> [1]
</IAPDIndividualReport>
```

Schema Component Representation

```
<xsd:element name="IAPDIndividualReport" type="IAPDIndividualReportType"/>
```

Table of Contents

Global Definitions

Complex Type: BrnchOfLocsType

Super-types:	None
Sub-types:	None

Name	BrnchOfLocsType
Abstract	no
Documentation	The Employment Location node contains a collection of Employment Location details for the individual.

XML Instance Representation

```
<...>
   <BrnchOfLoc> BrnchOfLocType </BrnchOfLoc> [0..*]
</...>
```

Schema Component Representation

```
<xsd:complexType name="BrnchOfLocsType">
   <xsd:sequence>
      <xsd:element name="BrnchOfLoc" type="BrnchOfLocType" minOccurs="0" maxOccurs="unbounded"/>
   </xsd:sequence>
</xsd:complexType>
```

Table of Contents

Complex Type: BrnchOfLocType

Super-types:	None
Sub-types:	None

Name	BrnchOfLocType
Abstract	no

XML Instance Representation

```
<...

<!-- This node has the firm's Branch Office address - street1. -->
str1="Str50[0..1]"

<!-- This node has the firm's Branch Office address - street2 -->
str2="Str50[0..1]"

<!-- This node has the firm's Branch Office address - City -->
city="Str50[0..1]"

<!-- This node has the firm's Branch Office address - State -->
state="StateCdType[0..1]"

<!-- This node has the firm's Branch Office address - Country -->
cntry="Str50[0..1]"

<!-- This node has the firm's Branch Office address - Postal Code -->
postlCd="Str11[0..1]"/>
```

Schema Component Representation

```
<xsd:complexType name="BrnchOfLocType">
   <xsd:attribute name="str1" type="Str50" use="optional"/>
   <xsd:attribute name="str2" type="Str50" use="optional"/>
   <xsd:attribute name="city" type="Str50" use="optional"/>
   <xsd:attribute name="state" type="StateCdType" use="optional"/>
   <xsd:attribute name="cntry" type="Str50" use="optional"/>
   <xsd:attribute name="postlCd" type="Str11" use="optional"/>
</xsd:complexType>
```

Table of Contents

Complex Type: CrntEmpsType

Super-types:	None
Sub-types:	None

Name	CrntEmpsType
Abstract	no
Documentation	The Current Employment node contains collection of active employment details

XML Instance Representation

```
<...>
   <!-- The Current Employment node contains a collection of active employment details -->
   <CrntEmp> CrntEmpType </CrntEmp> [0..*]
</...>
```

Schema Component Representation

```
<xsd:complexType name="CrntEmpsType">
   <xsd:sequence>
      <xsd:element name="CrntEmp" type="CrntEmpType" minOccurs="0" maxOccurs="unbounded"/>
   </xsd:sequence>
</xsd:complexType>
```

Table of Contents

Complex Type: CrntEmpType

Super-types:	None
Sub-types:	None

Name	CrntEmpType
Abstract	no

XML Instance Representation

```
<...

<!-- This node has the Firm's business name from the IARD composite record -->
orgNm="Str64[1]"

<!-- This node has the firm's CRD number. -->
orgPK="Int10[1]"

<!-- This node has the firm's address - street1. -->
str1="Str50[0..1]"

<!-- This node has the firm's address - street2 -->
str2="Str50[0..1]"

<!-- This node has the firm's address - City -->
city="Str50[0..1]"

<!-- This node has the firm's address - State -->
state="StateCdType[0..1]"

<!-- This node has the firm's address - Country -->
cntry="Str50[0..1]"

<!-- This node has the firm's address - Postal Code -->
postlCd="Str11[0..1]">
   <CrntRgstns> CrntRgstnsType </CrntRgstns> [0..1]
   <BrnchOfLocs> BrnchOfLocsType </BrnchOfLocs> [0..1]
</...>
```

Schema Component Representation

```
<xsd:complexType name="CrntEmpType">
   <xsd:sequence>
      <xsd:element name="CrntRgstns" type="CrntRgstnsType" minOccurs="0"/>
      <xsd:element name="BrnchOfLocs" type="BrnchOfLocsType" minOccurs="0"/>
   </xsd:sequence>
   <xsd:attribute name="orgNm" type="Str64" use="required"/>
   <xsd:attribute name="orgPK" type="Int10" use="required"/>
   <xsd:attribute name="str1" type="Str50" use="optional"/>
   <xsd:attribute name="str2" type="Str50" use="optional"/>
   <xsd:attribute name="city" type="Str50" use="optional"/>
   <xsd:attribute name="state" type="StateCdType" use="optional"/>
   <xsd:attribute name="cntry" type="Str50" use="optional"/>
   <xsd:attribute name="postlCd" type="Str11" use="optional"/>
</xsd:complexType>
```

Table of Contents

Complex Type: CrntRgstnsType

Super-types:	None
Sub-types:	None

Name	CrntRgstnsType
Abstract	no
Documentation	The CurrentRegistrations node contains a collection of registration details for the individual.

XML Instance Representation

```
<...>
   <CrntRgstn> CrntRgstnType </CrntRgstn> [0..*]
</...>
```

Schema Component Representation

```
<xsd:complexType name="CrntRgstnsType">
   <xsd:sequence>
      <xsd:element name="CrntRgstn" type="CrntRgstnType" minOccurs="0" maxOccurs="unbounded"/>
   </xsd:sequence>
</xsd:complexType>
```

Table of Contents

Complex Type: CrntRgstnType

Super-types:	None
Sub-types:	None

Name	CrntRgstnType
Abstract	no
Documentation	The CurrentRegistration node contains one of many registration details for the individual.

XML Instance Representation

```
<...

<!-- The regulatory Authority. -->
regAuth="StateCdType[1]"

<!-- The Registration Category. -->
regCat="registrationCategoryType[1]"

<!-- The current registration status. -->
st="registrationStatusCode[1]"

<!-- The date a status change was posted to the system. (YYYY-MM-DD) -->
stDt="xs:date[1]"/>
```

Schema Component Representation

```
<xsd:complexType name="CrntRgstnType">
   <xsd:attribute name="regAuth" type="StateCdType" use="required"/>
   <xsd:attribute name="regCat" type="registrationCategoryType" use="required"/>
   <xsd:attribute name="st" type="registrationStatusCode" use="required"/>
   <xsd:attribute name="stDt" type="xs:date" use="required"/>
</xsd:complexType>
```

Table of Contents

Complex Type: DRPsType

Super-types:	None
Sub-types:	None

Name	DRPsType
Abstract	no
Documentation	The DRPs node contains a collection of reportable and disclosable DRPs for the individual.

XML Instance Representation

```
<...>
   <!-- The OccurrenceDisclosureInformation node contains other information associated with the DRP. -->
   <DRP> DRPType </DRP> [0..9]
</...>
```

Schema Component Representation

```
<xsd:complexType name="DRPsType">
   <xsd:sequence>
      <xsd:element name="DRP" type="DRPType" minOccurs="0" maxOccurs="9"/>
   </xsd:sequence>
</xsd:complexType>
```

Table of Contents

Complex Type: DRPType

Super-types:	None
Sub-types:	None

Name	DRPType
Abstract	no
Documentation	The DRPs node contains one of many reportable and disclosable DRPs for the individual.

XML Instance Representation

```
<...

<!-- Flag to indicate the indvl has RegAction DRP. -->
hasRegAction="answerYNType[0..1]"

<!-- Flag to indicate the indvl has Criminal DRP. -->
hasCriminal="answerYNType[0..1]"

<!-- Flag to indicate the indvl has Bankrupt DRP. -->
hasBankrupt="answerYNType[0..1]"

<!-- Flag to indicate the indvl has CivilJudc DRP. -->
hasCivilJudc="answerYNType[0..1]"

<!-- Flag to indicate the indvl has Bond DRP. -->
hasBond="answerYNType[0..1]"

<!-- Flag to indicate the indvl has Judgment DRP. -->
hasJudgment="answerYNType[0..1]"

<!-- Flag to indicate the indvl has Invstgn DRP. -->
hasInvstgn="answerYNType[0..1]"

<!-- Flag to indicate the indvl has CustComp DRP. -->
hasCustComp="answerYNType[0..1]"

<!-- Flag to indicate the indvl has Termination DRP. -->
hasTermination="answerYNType[0..1]"/>
```

Schema Component Representation

```
<xsd:complexType name="DRPType">
   <xsd:attribute name="hasRegAction" type="answerYNType"/>
   <xsd:attribute name="hasCriminal" type="answerYNType"/>
   <xsd:attribute name="hasBankrupt" type="answerYNType"/>
   <xsd:attribute name="hasCivilJudc" type="answerYNType"/>
   <xsd:attribute name="hasBond" type="answerYNType"/>
   <xsd:attribute name="hasJudgment" type="answerYNType"/>
   <xsd:attribute name="hasInvstgn" type="answerYNType"/>
   <xsd:attribute name="hasCustComp" type="answerYNType"/>
   <xsd:attribute name="hasTermination" type="answerYNType"/>
</xsd:complexType>
```

Table of Contents

Complex Type: DsgntnsType

Super-types:	None
Sub-types:	None

Name	DsgntnsType
Abstract	no
Documentation	This node describes a collection of professional designations information for the individual.

XML Instance Representation

```
<...>
   <!-- Specifies the designation(s) an individual currently maintains. -->
   <Dsgntn> DsgntnType </Dsgntn> [0..*]
</...>
```

Schema Component Representation

```
<xsd:complexType name="DsgntnsType">
   <xsd:sequence>
      <xsd:element name="Dsgntn" type="DsgntnType" minOccurs="0" maxOccurs="unbounded"/>
   </xsd:sequence>
</xsd:complexType>
```

Table of Contents

Complex Type: DsgntnType

Super-types:	None
Sub-types:	None

Name	DsgntnType
Abstract	no
Documentation	This node describes a one of many designations information for the individual.

XML Instance Representation

```
<...

<!-- The designation code. -->
dsgntnNm="Str128[1]"/>
```

Schema Component Representation

```
<xsd:complexType name="DsgntnType">
   <xsd:attribute name="dsgntnNm" type="Str128" use="required"/>
</xsd:complexType>
```

Table of Contents

Complex Type: EmpHistsType

Super-types:	None
Sub-types:	None

Name	EmpHistsType
Abstract	no
Documentation	The EmploymentHistories node contains a collection of employment history details for the individual.

XML Instance Representation

```
<...>
   <EmpHs> EmpHistType </EmpHs> [0..*]
</...>
```

Schema Component Representation

```
<xsd:complexType name="EmpHistsType">
   <xsd:sequence>
      <xsd:element name="EmpHs" type="EmpHistType" minOccurs="0" maxOccurs="unbounded"/>
   </xsd:sequence>
</xsd:complexType>
```

Table of Contents

Complex Type: EmpHistType

Super-types:	None
Sub-types:	None

Name	EmpHistType
Abstract	no
Documentation	The EmploymentHistory node contains one of many employment history details for the individual.

XML Instance Representation

```
<...

<!-- The Employment Begin date. (MM/YYYY) -->
fromDt="Str7[0..1]"

<!-- The Employment End date. (MM/YYYY) -->
toDt="Str7[0..1]"

<!-- The organization name -->
orgNm="Str64[0..1]"

<!-- City of Employment. -->
city="Str50[0..1]"

<!-- State of Employment. -->
state="StateCdType[0..1]"/>
```

Schema Component Representation

```
<xsd:complexType name="EmpHistType">
   <xsd:attribute name="fromDt" type="Str7"/>
   <xsd:attribute name="toDt" type="Str7"/>
   <xsd:attribute name="orgNm" type="Str64"/>
   <xsd:attribute name="city" type="Str50"/>
   <xsd:attribute name="state" type="StateCdType"/>
</xsd:complexType>
```

Table of Contents

Complex Type: ExmsType

Super-types:	None
Sub-types:	None

Name	ExmsType
Abstract	no
Documentation	The Exams node contains a collection of exam nodes each describing details about an individual's Passed State Exams (S63,S64,S65,S66).

XML Instance Representation

```
<...>
   <Exm> ExmType </Exm> [0..*]
</...>
```

Schema Component Representation

```
<xsd:complexType name="ExmsType">
   <xsd:sequence>
      <xsd:element name="Exm" type="ExmType" minOccurs="0" maxOccurs="unbounded"/>
   </xsd:sequence>
</xsd:complexType>
```

Table of Contents

Complex Type: ExmType

Super-types:	None
Sub-types:	None

Name	ExmType
Abstract	no
Documentation	The Exam node contains one of many state exam details for the individual.

XML Instance Representation

```
<...

<!-- The Exam Code. -->
exmCd="examCodeType[1]"

<!-- The Exam Name. -->
exmNm="Str128[1]"

<!-- The date the exam was taken.(YYYY-MM-DD) -->
exmDt="xs:date[0..1]"/>
```

Schema Component Representation

```
<xsd:complexType name="ExmType">
   <xsd:attribute name="exmCd" type="examCodeType" use="required"/>
   <xsd:attribute name="exmNm" type="Str128" use="required"/>
   <xsd:attribute name="exmDt" type="xs:date"/>
</xsd:complexType>
```

Table of Contents

Complex Type: IAPDIndividualReportType

Super-types:	None
Sub-types:	None

Name	IAPDIndividualReportType
Abstract	no

XML Instance Representation

```
<...

<!-- This node contains the report generated date information. -->
GenOn="xsd:date[1]">
   <Indvls> IndvlsType </Indvls> [1]
</...>
```

Schema Component Representation

```
<xsd:complexType name="IAPDIndividualReportType">
   <xsd:sequence>
      <xsd:element name="Indvls" type="IndvlsType"/>
   </xsd:sequence>
   <xsd:attribute name="GenOn" type="xsd:date" use="required"/>
</xsd:complexType>
```

Table of Contents

Complex Type: IndvlsType

Super-types:	None
Sub-types:	None

Name	IndvlsType
Abstract	no
Documentation	The Individuals node contains a collection of individuals for the firm.

XML Instance Representation

```
<...>
   <Indvl> IndvlType </Indvl> [0..*]
</...>
```

Schema Component Representation

```
<xsd:complexType name="IndvlsType">
   <xsd:sequence>
      <xsd:element name="Indvl" type="IndvlType" minOccurs="0" maxOccurs="unbounded"/>
   </xsd:sequence>
</xsd:complexType>
```

Table of Contents

Complex Type: IndvlType

Super-types:	None
Sub-types:	None

Name	IndvlType
Abstract	no
Documentation	The Individual node contains Administrative and/or Disclosure information for one of many individuals requested by this report.

XML Instance Representation

```
<...>
   <Info> InfoType </Info> [1]
   <OthrNms> OthrNmsType </OthrNms> [0..1]
   <CrntEmps> CrntEmpsType </CrntEmps> [0..1]
   <Exms> ExmsType </Exms> [0..1]
   <Dsgntns> DsgntnsType </Dsgntns> [0..1]
   <PrevRgstns> PrevRgstnsType </PrevRgstns> [0..1]
   <EmpHss> EmpHistsType </EmpHss> [0..1]
   <OthrBuss> OthrBussType </OthrBuss> [0..1]
   <DRPs> DRPsType </DRPs> [0..1]
</...>
```

Schema Component Representation

```
<xsd:complexType name="IndvlType">
   <xsd:sequence>
      <xsd:element name="Info" type="InfoType"/>
      <xsd:element name="OthrNms" type="OthrNmsType" minOccurs="0"/>
      <xsd:element name="CrntEmps" type="CrntEmpsType" minOccurs="0"/>
      <xsd:element name="Exms" type="ExmsType" minOccurs="0"/>
      <xsd:element name="Dsgntns" type="DsgntnsType" minOccurs="0"/>
      <xsd:element name="PrevRgstns" type="PrevRgstnsType" minOccurs="0"/>
      <xsd:element name="EmpHss" type="EmpHistsType" minOccurs="0"/>
      <xsd:element name="OthrBuss" type="OthrBussType" minOccurs="0"/>
      <xsd:element name="DRPs" type="DRPsType" minOccurs="0"/>
   </xsd:sequence>
</xsd:complexType>
```

Table of Contents

Complex Type: InfoType

Super-types:	None
Sub-types:	None

Name	InfoType
Abstract	no
Documentation	The Info node contains basic information describing the individual.

XML Instance Representation

```
<...

<!-- Individual's Last Name -->
lastNm="Str25[0..1]"

<!-- Individual's First Name -->
firstNm="Str25[0..1]"

<!-- Individual's Middle Name -->
midNm="Str20[0..1]"

<!-- Individual's Suffix Name -->
sufNm="Str5[0..1]"

<!-- Individual's CRD # -->
indvlPK="Int10[1]"

<!-- Identifies if individual has Active AG Registration. -->
actvAGReg="answerYNType[1]"

<!-- Link URL to the individual's composite in IAPD. -->
link="Str128[0..1]"/>
```

Schema Component Representation

```
<xsd:complexType name="InfoType">
  <xsd:attribute name="lastNm" type="Str25"/>
  <xsd:attribute name="firstNm" type="Str25"/>
  <xsd:attribute name="midNm" type="Str20"/>
  <xsd:attribute name="sufNm" type="Str5"/>
  <xsd:attribute name="indvlPK" type="Int10" use="required"/>
  <xsd:attribute name="actvAGReg" type="answerYNType" use="required"/>
  <xsd:attribute name="link" type="Str128"/>
</xsd:complexType>
```

Table of Contents

Complex Type: OthrBussType

Super-types:	None
Sub-types:	None

Name	OthrBussType
Abstract	no
Documentation	This node describes a collection of other businesses for the individual.

XML Instance Representation

```
<...>
  <!-- This node identifies if the individual is engaged in any other business, either as a proprietor, partner,
  officer, director, employee, trustee, agent, or otherwise. -->
  <OthrBus> OthrBusType </OthrBus> [0..1]
</...>
```

Schema Component Representation

```
<xsd:complexType name="OthrBussType">
  <xsd:sequence>
    <xsd:element name="OthrBus" type="OthrBusType" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
```

Table of Contents

Complex Type: OthrBusType

Super-types:	None
Sub-types:	None

Name	OthrBusType
Abstract	no

XML Instance Representation

```
<...

<!-- Description for other business. -->
desc="Str4000[1]"/>
```

Schema Component Representation

```
<xsd:complexType name="OthrBusType">
   <xsd:attribute name="desc" type="Str4000" use="required"/>
</xsd:complexType>
```

Table of Contents

Complex Type: OthrNmsType

Super-types:	None
Sub-types:	None

Name	OthrNmsType
Abstract	no
Documentation	The OtherNames node contains a collection of names that an individual has used or is using, or by which they are known or have been known, other their legal name, since the age of 18. This would include, for example, nicknames, aliases, and names used before or after marriage.

XML Instance Representation

```
<...>
   <!-- The OtherName node contains one of many other names if Other Names have been provided. -->
   <OthrNm> OthrNmType </OthrNm> [0..*]
</...>
```

Schema Component Representation

```
<xsd:complexType name="OthrNmsType">
   <xsd:sequence>
      <xsd:element name="OthrNm" type="OthrNmType" minOccurs="0" maxOccurs="unbounded"/>
   </xsd:sequence>
</xsd:complexType>
```

Table of Contents

Complex Type: OthrNmType

Super-types:	None
Sub-types:	None

Name	OthrNmType
Abstract	no

XML Instance Representation

```
<...

<!-- Individual's Last Name of Other name -->
lastNm="Str25[0..1]"

<!-- Individual's First Name of Other name -->
firstNm="Str25[0..1]"

<!-- Individual's Middle Name of Other name -->
midNm="Str20[0..1]"

<!-- Individual's Suffix Name of Other name -->
sufNm="Str5[0..1]"/>
```

Schema Component Representation

```
<xsd:complexType name="OthrNmType">
   <xsd:attribute name="lastNm" type="Str25"/>
   <xsd:attribute name="firstNm" type="Str25"/>
   <xsd:attribute name="midNm" type="Str20"/>
   <xsd:attribute name="sufNm" type="Str5"/>
</xsd:complexType>
```

Table of Contents

Complex Type: PrevBrnchOfLocsType

Super-types:	None
Sub-types:	None

Name	PrevBrnchOfLocsType
Abstract	no
Documentation	The Employment Location node contains a collection of Employment Location details for the individual.

XML Instance Representation

```
<...>
   <BrnchOfLoc> PrevBrnchOfLocType </BrnchOfLoc> [0..*]
</...>
```

Schema Component Representation

```
<xsd:complexType name="PrevBrnchOfLocsType">
   <xsd:sequence>
      <xsd:element name="BrnchOfLoc" type="PrevBrnchOfLocType" minOccurs="0" maxOccurs="unbounded"/>
   </xsd:sequence>
</xsd:complexType>
```

Table of Contents

Complex Type: PrevBrnchOfLocType

Super-types:	None
Sub-types:	None

Name	PrevBrnchOfLocType
Abstract	no

XML Instance Representation

```
<...

<!-- This node has the firm's Branch Office address - City -->
city="Str50[0..1]"

<!-- This node has the firm's Branch Office address - State -->
state="StateCdType[0..1]"/>
```

Schema Component Representation

```
<xsd:complexType name="PrevBrnchOfLocType">
   <xsd:attribute name="city" type="Str50" use="optional"/>
   <xsd:attribute name="state" type="StateCdType" use="optional"/>
</xsd:complexType>
```

Table of Contents

Complex Type: PrevRgstnsType

Super-types:	None
Sub-types:	None

Name	PrevRgstnsType
Abstract	no
Documentation	The PreviousRegistration node contains registrations previously held by the individual.

XML Instance Representation

```
<...>
   <PrevRgstn> PrevRgstnType </PrevRgstn> [0..*]
</...>
```

Schema Component Representation

```
<xsd:complexType name="PrevRgstnsType">
   <xsd:sequence>
      <xsd:element name="PrevRgstn" type="PrevRgstnType" minOccurs="0" maxOccurs="unbounded"/>
   </xsd:sequence>
</xsd:complexType>
```

Table of Contents

Complex Type: PrevRgstnType

Super-types:	None
Sub-types:	None

Name	PrevRgstnType
Abstract	no
Documentation	The PreviousRegistration node contains one of many registration for the individual previously held.

XML Instance Representation

```
<...

<!-- This node has the Firm's business name from the IARD composite record -->
orgNm="Str64[1]"

<!-- This node has the firm's CRD number. -->
orgPK="Int10[1]"

<!-- The Registration Begin date. (YYYY-MM-DD) -->
regBeginDt="xs:date[0..1]"

<!-- The Registration End date. (YYYY-MM-DD) -->
regEndDt="xs:date[0..1]">
   <!-- The BranchOfficeLocs node contains one of many deficiencies for the individual registration. -->
   <BrnchOfLocs> PrevBrnchOfLocsType </BrnchOfLocs> [0..*]
</...>
```

Schema Component Representation

```
<xsd:complexType name="PrevRgstnType">
   <xsd:sequence>
      <xsd:element name="BrnchOfLocs" type="PrevBrnchOfLocsType" minOccurs="0" maxOccurs="unbounded"/>
   </xsd:sequence>
   <xsd:attribute name="orgNm" type="Str64" use="required"/>
   <xsd:attribute name="orgPK" type="Int10" use="required"/>
   <xsd:attribute name="regBeginDt" type="xs:date"/>
   <xsd:attribute name="regEndDt" type="xs:date"/>
</xsd:complexType>
```

Table of Contents

Simple Type: answerYNType

Super-types:	xsd:string < **answerYNType** (by restriction)
Sub-types:	None

Name	answerYNType
Content	• Base XSD Type: string • *value* comes from list: {'Y' | 'N'}

Schema Component Representation

```
<xsd:simpleType name="answerYNType">
   <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Y"/>
      <xsd:enumeration value="N"/>
   </xsd:restriction>
</xsd:simpleType>
```

Table of Contents

Simple Type: examCodeType

Super-types:	xs:string < **examCodeType** (by restriction)
Sub-types:	None

Name	examCodeType
Content	• Base XSD Type: string • *value* comes from list: {'S63' | 'S64' | 'S65' | 'S66'} • *length* <= 16
Documentation	The Exam Code. Allowed values(Code-Description) are: S63-Uniform Securities Agent State Law Examination

	S64-NASAA Real Estate Securities Exam S65-Uniform Investment Adviser Law Examination S66-Uniform Combined State Law Examination

Schema Component Representation

```
<xsd:simpleType name="examCodeType">
  <xsd:restriction base="xs:string">
    <xsd:maxLength value="16"/>
    <xsd:enumeration value="S63"/>
    <xsd:enumeration value="S64"/>
    <xsd:enumeration value="S65"/>
    <xsd:enumeration value="S66"/>
  </xsd:restriction>
</xsd:simpleType>
```

Table of Contents

Simple Type: Int10

Super-types:	xsd:integer < **Int10** (by restriction)
Sub-types:	None

Name	Int10
Content	- Base XSD Type: integer - *total no. of digits* = 10

Schema Component Representation

```
<xsd:simpleType name="Int10">
  <xsd:restriction base="xsd:integer">
    <xsd:totalDigits value="10"/>
  </xsd:restriction>
</xsd:simpleType>
```

Table of Contents

Simple Type: Int64

Super-types:	xsd:integer < **Int64** (by restriction)
Sub-types:	None

Name	Int64
Content	- Base XSD Type: integer - *total no. of digits* = 38

Schema Component Representation

```
<xsd:simpleType name="Int64">
  <xsd:restriction base="xsd:integer">
    <xsd:totalDigits value="38"/>
  </xsd:restriction>
</xsd:simpleType>
```

Table of Contents

Simple Type: Int8

Super-types:	xsd:integer < **Int8** (by restriction)
Sub-types:	None

Name	Int8

Content	• Base XSD Type: integer • *total no. of digits* = 8

Schema Component Representation

```
<xsd:simpleType name="Int8">
  <xsd:restriction base="xsd:integer">
    <xsd:totalDigits value="8"/>
  </xsd:restriction>
</xsd:simpleType>
```

Table of Contents

Simple Type: registrationCategoryType

Super-types:	xs:string < **registrationCategoryType** (by restriction)
Sub-types:	None

Name	registrationCategoryType
Content	• Base XSD Type: string • *value* comes from list: {'RA'} • *length* <= 16
Documentation	Representative Registration Category/Position with the Regulator. Allowed values(Code-Description) are: RA-Investment Advisor Representative

Schema Component Representation

```
<xsd:simpleType name="registrationCategoryType">
  <xsd:restriction base="xs:string">
    <xsd:maxLength value="16"/>
    <xsd:enumeration value="RA"/>
  </xsd:restriction>
</xsd:simpleType>
```

Table of Contents

Simple Type: registrationStatusCode

Super-types:	xs:string < **registrationStatusCode** (by restriction)
Sub-types:	None

Name	registrationStatusCode
Content	• Base XSD Type: string • *value* comes from list: {'ADMTERM' | 'APPRNT' | 'APPROVED' | 'APPROVED_RES' | 'APP_PEND_IARCE' | 'BAR' | 'CE2_YEAR_TRMD' | 'CE_INACTIVE' | 'DEFICIENT' | 'DENIED' | 'FTR' | 'INACTIVE_PRINTS' | 'MASS_TRNSF' | 'PENDING' | 'PURGED' | 'REJECTED' | 'REQUAL' | 'REQUEST_TERM' | 'REVOKED' | 'SUSPENSION' | 'TEMPREG' | 'TERMED' | 'T_NOREG' | 'T_NOREG_FTR' | 'T_NOREG_MT' | 'T_NOU5' | 'ABANDONED' | 'TEMP_WD' | 'T_NOMT' | 'T_NOMTNOREG' | 'APRSLTS' | 'TRANS_RQST' | 'TRANS_TERM' | 'TRANS_ERROR'} • *length* <= 16
Documentation	The registration status type. Allowed values(Code-Description) are: ABANDONED-Abandoned ADMTERM-Administrative Termination APPRNT-Approved Pending Prints APPROVED-Approved APPROVED_RES-Restricted Approval APP_PEND_IARCE-Approved - Pending IAR CE APRSLTS-Approved Pending Results BAR-Bar CE2_YEAR_TRMD-Termed CE Two Year Inactive CE_INACTIVE-Inactive - Continuing Education DEFICIENT-Deficient DENIED-Denied

	FTR-Terminated Failure to Renew INACTIVE_PRINTS-Inactive - Prints MASS_TRNSF-Termed Mass Transfer PENDING-Pending PURGED-Purged REJECTED-Rejected REQUAL-Requalification REQUEST_TERM-Termination Requested REVOKED-Revoked SUSPENSION-Suspended T_NOREG-Termed Without Registration T_NOREG_FTR-Terminated no registration Renewals T_NOREG_MT-Terminated no registration Mass Transfer T_NOU5-Terminated Without U5 T_NOMT-Termed-Did Not Mass Transfer T_NOMTNOREG-Terminated no reg.-Did Not Mass Transfer TEMPREG-Temporary Registration TEMP_WD-Temporary Withdrawal TERMED-Termed TRANS_RQST-Transition Requested TRANS_TERM-Transition Terminated TRANS_ERROR-Not Transitioned - Filed In Error

Schema Component Representation

```
<xsd:simpleType name="registrationStatusCode">
  <xsd:restriction base="xs:string">
    <xsd:maxLength value="16"/>
    <xsd:enumeration value="ADMTERM"/>
    <xsd:enumeration value="APPRNT"/>
    <xsd:enumeration value="APPROVED"/>
    <xsd:enumeration value="APPROVED_RES"/>
    <xsd:enumeration value="APP_PEND_IARCE"/>
    <xsd:enumeration value="BAR"/>
    <xsd:enumeration value="CE2_YEAR_TRMD"/>
    <xsd:enumeration value="CE_INACTIVE"/>
    <xsd:enumeration value="DEFICIENT"/>
    <xsd:enumeration value="DENIED"/>
    <xsd:enumeration value="FTR"/>
    <xsd:enumeration value="INACTIVE_PRINTS"/>
    <xsd:enumeration value="MASS_TRNSF"/>
    <xsd:enumeration value="PENDING"/>
    <xsd:enumeration value="PURGED"/>
    <xsd:enumeration value="REJECTED"/>
    <xsd:enumeration value="REQUAL"/>
    <xsd:enumeration value="REQUEST_TERM"/>
    <xsd:enumeration value="REVOKED"/>
    <xsd:enumeration value="SUSPENSION"/>
    <xsd:enumeration value="TEMPREG"/>
    <xsd:enumeration value="TERMED"/>
    <xsd:enumeration value="T_NOREG"/>
    <xsd:enumeration value="T_NOREG_FTR"/>
    <xsd:enumeration value="T_NOREG_MT"/>
    <xsd:enumeration value="T_NOU5"/>
    <xsd:enumeration value="ABANDONED"/>
    <xsd:enumeration value="TEMP_WD"/>
    <xsd:enumeration value="T_NOMT"/>
    <xsd:enumeration value="T_NOMTNOREG"/>
    <xsd:enumeration value="APRSLTS"/>
    <xsd:enumeration value="TRANS_RQST"/>
    <xsd:enumeration value="TRANS_TERM"/>
    <xsd:enumeration value="TRANS_ERROR"/>
  </xsd:restriction>
</xsd:simpleType>
```

Table of Contents

Simple Type: StateCdType

Super-types:	xsd:NMTOKEN < **StateCdType** (by restriction)
Sub-types:	None

Name	StateCdType
Content	- Base XSD Type: NMTOKEN

	• *value* comes from list: {'AK' | 'AL' | 'AR' | 'AZ' | 'CA' | 'CO' | 'CT' | 'DC' | 'DE' | 'FL' | 'GA' | 'GU' | 'HI' | 'IA' | 'ID' | 'IL' | 'IN' | 'KS' | 'KY' | 'LA' | 'MA' | 'MD' | 'ME' | 'MI' | 'MN' | 'MO' | 'MS' | 'MT' | 'NC' | 'ND' | 'NE' | 'NH' | 'NJ' | 'NM' | 'NV' | 'NY' | 'OH' | 'OK' | 'OR' | 'PA' | 'PR' | 'RI' | 'SC' | 'SD' | 'TN' | 'TX' | 'UT' | 'VA' | 'VT' | 'WA' | 'WI' | 'WV' | 'WY' | 'VI'} • *length* <= 2
Documentation	Elements of this type represent the allowed State Codes. Allowed values (code-Name) are: AK-Alaska AL-Alabama AR-Arkansas AZ-Arizona CA-California CO-Colorado CT-Connecticut DC-District of Columbia DE-Delaware FL-Florida GA-Georgia GU-Guam HI-Hawaii IA-Iowa ID-Idaho IL-Illinois IN-Indiana KS-Kansas KY-Kentucky LA-Louisiana MA-Massachusetts MD-Maryland ME-Maine MI-Michigan MN-Minnesota MO-Missouri MS-Mississippi MT-Montana NC-North Carolina ND-North Dakota NE-Nebraska NH-New Hampshire NJ-New Jersey NM-New Mexico NV-Nevada NY-New York OH-Ohio OK-Oklahoma OR-Oregon PA-Pennsylvania PR-Puerto Rico RI-Rhode Island SC-South Carolina SD-South Dakota TN-Tennessee TX-Texas UT-Utah VA-Virginia VT-Vermont WA-Washington WI-Wisconsin WV-West Virginia WY-Wyoming VI-Virgin Islands

Schema Component Representation

```
<xsd:simpleType name="StateCdType">
   <xsd:restriction base="xsd:NMTOKEN">
      <xsd:maxLength value="2"/>
      <xsd:enumeration value="AK"/>
      <xsd:enumeration value="AL"/>
      <xsd:enumeration value="AR"/>
      <xsd:enumeration value="AZ"/>
      <xsd:enumeration value="CA"/>
      <xsd:enumeration value="CO"/>
      <xsd:enumeration value="CT"/>
      <xsd:enumeration value="DC"/>
      <xsd:enumeration value="DE"/>
      <xsd:enumeration value="FL"/>
      <xsd:enumeration value="GA"/>
      <xsd:enumeration value="GU"/>
      <xsd:enumeration value="HI"/>
      <xsd:enumeration value="IA"/>
      <xsd:enumeration value="ID"/>
      <xsd:enumeration value="IL"/>
      <xsd:enumeration value="IN"/>
      <xsd:enumeration value="KS"/>
      <xsd:enumeration value="KY"/>
      <xsd:enumeration value="LA"/>
      <xsd:enumeration value="MA"/>
      <xsd:enumeration value="MD"/>
      <xsd:enumeration value="ME"/>
      <xsd:enumeration value="MI"/>
      <xsd:enumeration value="MN"/>
      <xsd:enumeration value="MO"/>
      <xsd:enumeration value="MS"/>
      <xsd:enumeration value="MT"/>
      <xsd:enumeration value="NC"/>
      <xsd:enumeration value="ND"/>
      <xsd:enumeration value="NE"/>
      <xsd:enumeration value="NH"/>
      <xsd:enumeration value="NJ"/>
      <xsd:enumeration value="NM"/>
      <xsd:enumeration value="NV"/>
      <xsd:enumeration value="NY"/>
      <xsd:enumeration value="OH"/>
      <xsd:enumeration value="OK"/>
      <xsd:enumeration value="OR"/>
      <xsd:enumeration value="PA"/>
      <xsd:enumeration value="PR"/>
      <xsd:enumeration value="RI"/>
      <xsd:enumeration value="SC"/>
      <xsd:enumeration value="SD"/>
      <xsd:enumeration value="TN"/>
      <xsd:enumeration value="TX"/>
      <xsd:enumeration value="UT"/>
      <xsd:enumeration value="VA"/>
      <xsd:enumeration value="VT"/>
      <xsd:enumeration value="WA"/>
      <xsd:enumeration value="WI"/>
      <xsd:enumeration value="WV"/>
      <xsd:enumeration value="WY"/>
      <xsd:enumeration value="VI"/>
   </xsd:restriction>
</xsd:simpleType>
```

Table of Contents

Simple Type: Str10

Super-types:	xsd:string < **Str10** (by restriction)
Sub-types:	None

Name	Str10
Content	• Base XSD Type: string • *length* <= 10

Schema Component Representation

```
<xsd:simpleType name="Str10">
   <xsd:restriction base="xsd:string">
      <xsd:maxLength value="10"/>
   </xsd:restriction>
</xsd:simpleType>
```

Table of Contents

Simple Type: Str11

Super-types:	xsd:string < **Str11** (by restriction)
Sub-types:	None

Name	Str11
Content	• Base XSD Type: string • *length* <= 11

Schema Component Representation

```
<xsd:simpleType name="Str11">
   <xsd:restriction base="xsd:string">
      <xsd:maxLength value="11"/>
   </xsd:restriction>
</xsd:simpleType>
```

Table of Contents

Simple Type: Str128

Super-types:	xsd:string < **Str128** (by restriction)
Sub-types:	None

Name	Str128
Content	• Base XSD Type: string • *length* <= 512

Schema Component Representation

```
<xsd:simpleType name="Str128">
   <xsd:restriction base="xsd:string">
      <xsd:maxLength value="512"/>
   </xsd:restriction>
</xsd:simpleType>
```

Table of Contents

Simple Type: Str20

Super-types:	xsd:string < **Str20** (by restriction)
Sub-types:	None

Name	Str20
Content	• Base XSD Type: string • *length* <= 20

Schema Component Representation

```
<xsd:simpleType name="Str20">
   <xsd:restriction base="xsd:string">
      <xsd:maxLength value="20"/>
   </xsd:restriction>
</xsd:simpleType>
```

Table of Contents

Simple Type: Str25

Super-types:	xsd:string < **Str25** (by restriction)
Sub-types:	None

Name	Str25
Content	• Base XSD Type: string • *length* <= 25

Schema Component Representation

```
<xsd:simpleType name="Str25">
   <xsd:restriction base="xsd:string">
      <xsd:maxLength value="25"/>
   </xsd:restriction>
</xsd:simpleType>
```

Table of Contents

Simple Type: Str4000

Super-types:	xsd:string < **Str4000** (by restriction)
Sub-types:	None

Name	Str4000
Content	• Base XSD Type: string • *length* <= 4000

Schema Component Representation

```
<xsd:simpleType name="Str4000">
   <xsd:restriction base="xsd:string">
      <xsd:maxLength value="4000"/>
   </xsd:restriction>
</xsd:simpleType>
```

Table of Contents

Simple Type: Str5

Super-types:	xsd:string < **Str5** (by restriction)
Sub-types:	None

Name	Str5
Content	• Base XSD Type: string • *length* <= 5

Schema Component Representation

```
<xsd:simpleType name="Str5">
   <xsd:restriction base="xsd:string">
      <xsd:maxLength value="5"/>
   </xsd:restriction>
</xsd:simpleType>
```

Table of Contents

Simple Type: Str50

Super-types:	xsd:string < **Str50** (by restriction)
Sub-types:	None

Name	Str50
Content	- Base XSD Type: string - *length* <= 50

Schema Component Representation

```
<xsd:simpleType name="Str50">
   <xsd:restriction base="xsd:string">
      <xsd:maxLength value="50"/>
   </xsd:restriction>
</xsd:simpleType>
```

Table of Contents

Simple Type: Str64

Super-types:	xsd:string < **Str64** (by restriction)
Sub-types:	None

Name	Str64
Content	- Base XSD Type: string - *length* <= 64

Schema Component Representation

```
<xsd:simpleType name="Str64">
   <xsd:restriction base="xsd:string">
      <xsd:maxLength value="64"/>
   </xsd:restriction>
</xsd:simpleType>
```

Table of Contents

Simple Type: Str7

Super-types:	xsd:string < **Str7** (by restriction)
Sub-types:	None

Name	Str7
Content	- Base XSD Type: string - *length* <= 7

Schema Component Representation

```
<xsd:simpleType name="Str7">
  <xsd:restriction base="xsd:string">
    <xsd:maxLength value="7"/>
  </xsd:restriction>
</xsd:simpleType>
```

Table of Contents

Schema Reference Glossary

Glossary

Abstract (Applies to complex type definitions and element declarations). An abstract element or complex type cannot used to validate an element instance. If there is a reference to an abstract element, only element declarations that can substitute the abstract element can be used to validate the instance. For references to abstract type definitions, only derived types can be used.

All Model Group Child elements can be provided *in any order* in instances. See: http://www.w3.org/TR/xmlschema-1/#element-all.

Choice Model Group *Only one* from the list of child elements and model groups can be provided in instances. See: http://www.w3.org/TR/xmlschema-1/#element-choice.

Collapse Whitespace Policy Replace tab, line feed, and carriage return characters with space character (Unicode character 32). Then, collapse contiguous sequences of space characters into single space character, and remove leading and trailing space characters.

Disallowed Substitutions (Applies to element declarations). If *substitution* is specified, then substitution group members cannot be used in place of the given element declaration to validate element instances. If *derivation methods*, e.g. extension, restriction, are specified, then the given element declaration will not validate element instances that have types derived from the element declaration's type using the specified derivation methods. Normally, element instances can override their declaration's type by specifying an `xsi:type` attribute.

Key Constraint Like Uniqueness Constraint, but additionally requires that the specified value(s) must be provided. See: http://www.w3.org/TR/xmlschema-1/#cIdentity-constraint_Definitions.

Key Reference Constraint Ensures that the specified value(s) must match value(s) from a Key Constraint or Uniqueness Constraint. See: http://www.w3.org/TR/xmlschema-1/#cIdentity-constraint_Definitions.

Model Group Groups together element content, specifying the order in which the element content can occur and the number of times the group of element content may be repeated. See: http://www.w3.org/TR/xmlschema-1/#Model_Groups.

Nillable (Applies to element declarations). If an element declaration is nillable, instances can use the `xsi:nil` attribute. The `xsi:nil` attribute is the boolean attribute, *nil*, from the *http://www.w3.org/2001/XMLSchema-instance* namespace. If an element instance has an `xsi:nil` attribute set to true, it can be left empty, even though its element declaration may have required content.

Notation A notation is used to identify the format of a piece of data. Values of elements and attributes that are of type, NOTATION, must come from the names of declared notations. See: http://www.w3.org/TR/xmlschema-1/#cNotation_Declarations.

Preserve Whitespace Policy Preserve whitespaces exactly as they appear in instances.

Prohibited Derivations (Applies to type definitions). Derivation methods that cannot be used to create sub-types from a given type definition.

Prohibited Substitutions (Applies to complex type definitions). Prevents sub-types that have been derived using the specified derivation methods from validating element instances in place of the given type definition.

Replace Whitespace Policy Replace tab, line feed, and carriage return characters with space character (Unicode character 32).

Sequence Model Group Child elements and model groups must be provided *in the specified order* in instances. See: http://www.w3.org/TR/xmlschema-1/#element-sequence.

Substitution Group Elements that are *members* of a substitution group can be used wherever the *head* element of the substitution group is referenced.

Substitution Group Exclusions (Applies to element declarations). Prohibits element declarations from nominating themselves as being able to substitute a given element declaration, if they have types that are derived from the original element's type using the specified derivation methods.

Target Namespace The target namespace identifies the namespace that components in this schema belongs to. If no target namespace is provided, then the schema components do not belong to any namespace.

Uniqueness Constraint Ensures uniqueness of an element/attribute value, or a combination of values, within a specified scope. See: http://www.w3.org/TR/xmlschema-1/#cIdentity-constraint_Definitions.