FGDC Document Number XX

United States Thoroughfare, Landmark, and Postal Address Data Standard (Draft)

Standards Working Group
Federal Geographic Data Committee

January 2010

Federal Geographic Data Committee

Established by Office of Management and Budget Circular A-16, the Federal Geographic Data Committee (FGDC) promotes the coordinated development, use, sharing, and dissemination of geographic data.

The FGDC is composed of representatives from the Departments of Agriculture, Commerce, Defense, Energy, Housing and Urban Development, the Interior, State, and Transportation; the Environmental Protection Agency; the Federal Emergency Management Agency; the Library of Congress; the National Aeronautics and Space Administration; the National Archives and Records Administration; and the Tennessee Valley Authority. Additional Federal agencies participate on FGDC subcommittees and working groups. The Department of the Interior chairs the committee.

FGDC subcommittees work on issues related to data categories coordinated under the circular. Subcommittees establish and implement standards for data content, quality, and transfer; encourage the exchange of information and the transfer of data; and organize the collection of geographic data to reduce duplication of effort. Working groups are established for issues that transcend data categories.

For more information about the committee, or to be added to the committee's newsletter mailing list, please contact:

Federal Geographic Data Committee Secretariat
c/o U.S. Geological Survey
590 National Center
Reston, Virginia 22092

Telephone: (703) 648-5514
Facsimile: (703) 648-5755
Internet (electronic mail): gdc@usgs.gov
Anonymous FTP: ftp://fgdc.er.usgs.gov/pub/gdc/
World Wide Web: http://fgdc.er.usgs.gov/fgdc.html

CONTENTS

1. Introduction

The Need for a Comprehensive Address Data Standard

Addresses are the location identifiers most widely used by the public and by state and local government. Addresses are critical information for administrative, emergency response, research, marketing, mapping, GIS, routing and navigation, and many other purposes. Because they have evolved over many decades, under the control of thousands of local jurisdictions, in many different record and database formats, and to serve many purposes, different address formats and types pose a number of complex geoprocessing and modeling issues. As a consequence, government agencies struggle with these issues as they seek to integrate large, mission-critical files into master address repositories.

Local governments must record and locate every address within and around their jurisdictions. Local governments must ascertain the location of every address that appears anywhere in their administrative records--every residence, business, public structure, building permit, emergency response site, voter, school child, and public service client, including addresses where no one resides and no mail is received. In many places addresses are also used to identify infrastructure facilities, including bus stops, fire hydrants, utility poles and meters, cell phone towers, manholes, and signs.

To organize, maintain, and provide address records, local address authorities must create master address repositories that replace the numerous isolated, incomplete departmental address data files with one authoritative, integrated geographic address database. The construction of master address repositories is of paramount importance at the local level, because it permits departments to integrate address-related records, and ultimately

operations, across department lines. The repository must include, not just the address itself, but its coordinate location, and documentation of where the address record originated and whether it is (or ever was) valid. To check validity and facilitate data maintenance, the repository must record the business rules by which addresses are assigned.

Emergency dispatchers in particular require accurate address locations. Emergency dispatchers must be able to route an emergency vehicle to any address in their response area, under circumstances when minutes matter. For emergency dispatchers, having well documented, standardized address data can mean the difference between life and death. Many 911 callers use cellphones, which report the callers' coordinates, but not their addresses. Emergency dispatchers must then infer the address from the coordinates. Translation from the coordinates to addresses is thus of increasing importance for dispatchers and first responders.

The USPS, commercial delivery services, and direct mail firms, before sending anything or attempting delivery, must verify the delivery address by standardizing it and matching it against a standardized master address list. Together they have, over several decades, worked out specifications for standardizing addresses and formatting mailing labels. The specifications are published in USPS Publication 28, "Postal Addressing Standards." The USPS maintains the nationwide master list of mailing addresses. Maintenance is complicated by the general lack of any local authority for address updates.

Government agencies require unambiguous ways to exchange address data among different units of government, both at the local level, e.g., city to city, or city to county,

and between different levels of government, e.g., from city or county to regional, state and federal agencies. The need is critical in times of emergency.

Finally, regional, state, and federal agencies (as well as private-sector firms) must aggregate local address files into state and national address lists. These include, most prominently, the USPS ZIP+4 and City State files, and Census Bureau MAF/TIGER files.

A comprehensive address data standard must serve the full range of these needs: postal delivery and census enumeration, local government administration and intergovernmental cooperation, emergency dispatch, the creation and administration of master address repositories by local address authorities, and the aggregation of local records into larger regional, state, and national address databases.

In sponsoring the creation of the *United States Thoroughfare, Landmark, and Postal Address Data Standard*, the Federal Geographic Data Committee (FGDC) has sought to convene, under the auspices of its Subcommittee on Cultural and Demographic Data, interested parties from among the local, state, Federal, and non-government sectors to resolve address data modeling and geoprocessing and to create a comprehensive address data standard, thereby helping to make our national spatial data infrastructure truly national.

1.1 Objective

The *United States Thoroughfare, Landmark, and Postal Address Data Standard* has been created to:

- Provide one standard that meets the diverse address data management requirements for local address administration, postal and package delivery, emergency response (and navigation generally), administrative recordkeeping, and address data aggregation.
- Support the use of best practices in address data management.
- Provide a systematic, consistent basis for recording all addresses in the United States.
- Define the elements needed to compose addresses and store them within relational databases and geographic information systems.
- Define the attributes needed for address documentation, mapping, and quality testing, including address ID's, coordinates, and linear reference locations.
- Provide a complete taxonomy (systematic classification) of US addresses that is useful to address data managers.
- Introduce the idea of the address reference system—the formal description of the local address assignment rules, both spatial and non-spatial—and define its elements and attributes, as a basis for address assignment and quality testing.
- Define tests and procedures for address data quality testing, error-trapping, and anomaly identification.
- Support seamless exchange of address information, and foster consistent implementation of this standard, by defining XML models for every address element, attribute, and class, integrated into a single XML Schema Document.
- Offer a migration path from legacy formats to standards-compliant ones.

- Recognize, as a practical matter, that different business purposes and different data sources will require different levels of complexity in address data records, files and repositories.
- Build on USPS Publication 28, the Census Bureau TIGER files, the FGDC Content Standard for Digital Geospatial Metadata, the FGDC's National Spatial Data Infrastructure (NSDI) Framework Data Content Standard, and previous FGDC address standard efforts.

1.2 Scope

1.2.1 Subject and Area

The *United States Thoroughfare, Landmark, and Postal Address Data Standard* covers thoroughfare, landmark, and postal addresses within the United States, including its outlying territories and possessions.

1.2.2 Structure: One Standard, Four Parts

This standard has been developed in conformance with the FGDC Standards Reference Model for data standards. It provides, in four separate parts, a data content, classification, quality, and exchange standard for thoroughfare, landmark, and postal addresses, and for address reference systems:

- Data Content standards provide semantic definitions of a set of objects. In this standard, the content part specifies and defines the data elements that may appear in or describe street, landmark, and postal addresses, and address reference systems.

- **Data Classification** standards provide groups or categories of data that serve an application. In this standard, the classification part defines classes of addresses according to their syntax, that is, their data elements and the order in which the elements are arranged.
- **Data Quality** standards describe how to express the applicability or essence of a data set or data element and include data quality, assessment, accuracy, and reporting or documentation standards. In this standard, the Data Quality part specifies tests and measures of address data quality.
- **Data Exchange** standards describe how to produce or consume packages of data, independent of technology and applications, to facilitate moving data between agencies and systems. In this standard, the Data Exchange part provides a complete XML schema description for exchange of address data.

The *United States Thoroughfare, Landmark, and Postal Address Data Standard* is thus one standard, comprised of four parts: Address Data Content, Address Data Classification, Address Data Quality, and Address Data Exchange.

1.2.3 Definition of “Address.”

This standard proposes a new definition of "address":

An address specifies a location by reference to a thoroughfare or a landmark; or it specifies a point of postal delivery.

This definition differentiates addressing from the two other types of spatial referencing systems, coordinate reference systems and linear reference systems. The difference rests, not on what the systems locate, but on what they refer to in order to specify a location.

Coordinate reference systems specify location by reference to a grid, spheroid, or geoid (and a datum). Linear reference systems specify location by reference to a route (and a beginning point). Within the context of this standard, coordinates and linear reference locations are treated as attributes of addresses, or, in the cases of certain postal delivery addresses, as inapplicable. This definition also excludes email and other computer system addresses.

This definition places address occupants and mail recipients (addressees) outside the scope of the standard. Many postal addressing standards include specifications for personal names, business names, and internal distribution points such as mailstops, particularly in the context of specifying formats for mailing labels. However, an addressee may have multiple addresses, and an address may have many occupants. For address data management, address and addressee should be treated as separate entities, and defined by separate standards.

1.2.4 Address Data Classification: A Syntactical Approach

The standard classifies addresses according to their syntax, that is, their address elements and the order in which the elements are arranged. Syntax determines the record structure needed to hold and exchange an address, and often it is all that is known about the addresses in a given file.

Classifying addresses by syntax rather than semantics (i.e., meaning) allows the users of the standard to focus on record structures, and to avoid the need for any assumptions about what kind of feature the address might identify. Classifying addresses by feature can be frustrating or impossible because:

1. Reliable information about an address may be unavailable.
2. Often, one address is used to identify several types of features (e.g., parcel, building, building entrance, utility meter, utility pole, incident location, etc.) at the same location.
3. A set of feature categories may be found to be ambiguous or incomplete when applied to a given address.

The Address Data Classification part of the standard classifies all US addresses into a simple, complete taxonomy of ten US address classes. Consistent with the principles of the General Information Model defined in the FGDC Framework Data Content Standard Base Part, each particular address class is a subclass of an abstract Address Class. The ten address classes are organized into three groups, plus a catch-all general class.

1.2.4.1 Thoroughfare Address Classes.

Thoroughfare addresses specify a location by reference to a thoroughfare. A thoroughfare is defined as a "road or part of a road or other access route along which a delivery point can be accessed"(UPU Publication S42-4 (sec. 5.2.9)). A thoroughfare is typically but not always a road — it may be, for example, a walkway, a railroad, or a river. The thoroughfare address classes are:

1. Numbered Thoroughfare Address ("123 Main Street")
2. Intersection Address ("Fifth Avenue and Main Street")

3. Two Number Address Range ("405-411 West Green Street")
4. Four Number Address Range ("900-962, 901-963 Milton Street")
5. Unnumbered Thoroughfare Address ("Forest Service Road 698")

1.2.4.2 Landmark Address Classes.

Landmark addresses specify a location by reference to a named landmark. A landmark is a relatively permanent feature of the manmade landscape that has recognizable identity within a particular cultural context" (definition adapted from U.S. Board on Geographic Names, 2003, p. 48).

6. Landmark Address ("Statue of Liberty")
7. Community Address ("123 Urbanizacion Los Olmos")

1.2.4.3 Postal Delivery Address Classes.

Postal delivery addresses specify points of postal delivery that have no definite relation to the location of the recipient, such as a post office box, rural route box, overseas military address, or general delivery office. The USPS specifies each class in detail in USPS Publication 28.

8. USPS Postal Delivery Box ("PO Box 16953")
9. USPS Postal Delivery Route ("RR 1, Box 100")
10. USPS General Delivery Office ("General Delivery")

1.2.4.4 General Address Class.

The General Address Class is for files that hold addresses from various classes, and for addresses (such as foreign addresses) that might not fit in any of the thoroughfare, landmark, or postal delivery classes.

1.2.5 Address Data Content: Elements

The Address Data Content part of the standard names and defines the simple and complex data elements needed to construct addresses, and for each one provides, among other information, its name, definition, data type, existing standards (if any), domain of values (if any), examples, and explanatory notes; XML tag, XML model, example, and notes; and data quality measures and notes. The elements are too numerous to list here, but they cover:

- Address numbers and their components
- Street names and their components
- Subaddresses (apartments, offices, suites, etc.) and their components
- Landmark names
- Larger areas (place names, states, ZIP Codes and ZIP+4, and country names)
- USPS postal address elements (PO Boxes, rural routes, overseas military addresses, general delivery, etc.)
- USPS address lines (Delivery Line and Last Line, as specified in USPS Publication 28)

1.2.6 Address Data Content: Attributes for Documentation, Mapping and Quality Control

The Address Data Content part of the standard also defines a number of attributes needed for address documentation, mapping, and quality control. For each attribute, the standard provides the same information that is provided for the address elements. Collectively the attributes constitute record-level metadata for each address. The attributes are too numerous to list here completely, but key attributes include:

- A unique identifier for each different address, to serve as a primary key in an address database.
- Geographic coordinates and linear referencing locations.
- Lifecycle status (potential, proposed, active, retired).
- Address Class (in terms of the taxonomy described above).
- Address feature type (the type of feature located by the address, e.g., parcel, building, entrance, subaddress, infrastructure component, etc.).
- Official status (official, alias, unofficial, etc.).
- Related address identifier and type of relation (to relate, say, an alias address to its official address, or a landmark address to its equivalent thoroughfare address, or a parcel address to the tax billing address).
- The address authority that assigned the address, the dataset where it is found, and the dates the address was created and retired.

- Various attributes that describe specific elements, such as address number parity, address range type, and place name type.

1.2.7 Address Reference System: The Local Framework for Address Assignment

The Address Data Content part of the standard introduces the concept of an address reference system and defines the elements needed to compose, describe and document it. An address reference system is the framework of local rules, both spatial and non-spatial, by which new addresses are assigned and old ones checked within a specific area. It may include rules for naming streets and for assigning address numbers along them, as well as a boundary defining the area within which the rules apply. The address reference system, in turn, is important to data quality testing.

1.2.8 Address Data Quality: A Complete Suite of Data Quality Tests

The Address Data Quality part of the standard provides a complete suite of data quality tests for all address elements, attributes, and classes. These tests measure how well a given set of address records conforms to this standard and the local address reference system. The tests are developed in terms consistent with the FGDC's "Content Standard for Digital Geospatial Metadata" (FGDC 1998) and subsequent SDTS and ISO standards of spatial data quality. Each test specification includes the scope, measure, and procedure of the test; an SQL pseudocode script; and parameters for calculating anomalies as a percentage of the data set.

1.2.9 Address Data Exchange: XML Schema Document (XSD), XML, and UML

The Address Data Exchange part of the standard includes an XSD that describes the XML elements, attributes, and classes, and the rules for assembling them. It also includes a UML metamodel. The XSD provides complete, open, standard XML data exchange templates for both monolithic and transactional data exchanges. XML is well-suited for this purpose (and required by FGDC exchange standards), because it supports seamless exchange between different users, while allowing for local variations on either end.

The XSD conforms to the W3C XML Core Working Group "Extensible Markup Language (XML) 1.0" (Third Edition, W3C Recommendation 4 February 2004). Geometry elements are defined and implemented following OGC's. "OpenGIS(R) Geography Markup Language (GML)" (Version: 3.1.1). These versions were chosen to provide consistency with the FGDC's Geographic Information Framework Data Content Standard. (See Appendix A for complete references.)

1.2.10 A Data Model, but Not a Database Model

The XSD defines an address data model. It states the rules for combining simple elements into complex elements, for composing addresses from simple and complex elements, and for using attributes to describe addresses and their elements.

However, the standard does not provide a database model with table structures or relationships. The standard does not prescribe one specific design for constructing complex elements from simple elements, or addresses from their complex and simple elements. It does

not specify, for example, how to relate address numbers to street names, or compose a master street name list, or geocode addresses, even though these and other tasks are crucial to the creation and maintenance of an address database.

There are many ways to accomplish these tasks. The standard accommodates a range of different design choices in composing, relating, and describing elements and addresses. The best way depends on local circumstances, rules, customs, and anomalies—and therefore cannot be prescribed in a standard. Instead, these choices are left as implementation matters to be decided locally.

1.2.11 A Few Basic Statements on Implementing This Standard

An implementation guide is well beyond the scope of this standard, but a few things can be stated here:

- The standard does not require parsing every address into its simplest elements, nor does it require creation of a complex, highly-normalized address data base. The standard recognizes and supports different levels of complexity, from the two-line format prescribed in USPS Publication 28 to a highly-parsed, fully-normalized database.
- By the same principle, the standard does not require incorporation of every element and attribute. Only the Address ID is required for every address record. From among the others, select only those needed for the purpose at hand, and omit the rest. For example, if none of the addresses in a given area have any Address Number Prefixes, that element may be omitted from the address records for that area. In another

example, the two-line USPS Publication 28 address format can be represented, if desired, by only two complex elements—or it can be composed from a more complex array of simple and complex elements.

- The standard does not require use of most of the address attributes. However, the Address ID is required, and several other attributes are essential for most purposes.

These choices, and others, will be dictated by the specific purpose for which the standard is applied, and the specific data to which it is applied.

1.2.12 Abbreviations in Addresses

Abbreviations are frequently used in addresses, and in particular the USPS abbreviations for street name directionals and types are widely used. However, this standard recognizes only two specific groups of abbreviations, both of which are unambiguous and used without variation:

- The two-letter abbreviations for the fifty states; the District of Columbia, US territories, possessions, and minor outlying islands; and USPS-designated overseas military and diplomatic "state" equivalents (AA, AE, AP)(see State Name element).
- Nine USPS abbreviations defined for postal delivery purposes and having no direct relation to any location (PO Box, PMB; RR, HC; PSC, CMR; APO, FPO, and DPO)(see USPS Postal Delivery Box and USPS Postal Delivery Route address classes).

No other abbreviations are recognized within the standard, for three reasons:

- The standard must serve a broad range of purposes, and no set of abbreviations is used for all those purposes. USPS abbreviations, for example, differ from emergency dispatch abbreviations and from other abbreviations in use.
- Abbreviations can create ambiguity. As an example, consider “N W Jones Tr.” Is it “Northwest Jones Tr,” “Noble Wimberly Jones Tr,” or “North William Jones Tr”? Does Tr stand for Terrace, Trail, or Trace? Abbreviations lose information about the full address, and thereby hamper data quality testing and data exchange. Time saved in data entry is lost in checking ambiguous addresses.
- Any list of standard abbreviations is bound to be incomplete. A few examples of street types missing from the most recent (2006) USPS list include: Alcove, Close, Connector, Downs, Exchange, and Promenade. In addition many applications such as 911 dispatch require specialized local abbreviations (e.g., “NCap” for North Capitol Street). Local abbreviations will not be clear to outsiders unless the complete form can be recovered from the master address record.

Therefore addresses should be stored unabbreviated in the master address record, and views or export routines should be used to meet the needs of E-911, mailing addresses, etc. If a link is preserved between the primary record and its recognized alternatives, abbreviations are unambiguously expandable when necessary -- as for instance when address information must be shared between two agencies that use different abbreviation rules.

This standard recognizes all USPS abbreviations and abbreviation rules within the Postal Addressing Profile. Additional profiles can be created if other needs warrant.

1.2.13 No Address Data Presentation Standard is Included

This standard does not specify how address data should be symbolized graphically or geographically. The appropriate representation depends on the purpose of the map creator, so no standard is warranted.

1.2.14 Language and Character Set

For English-language addresses, this standard can be implemented with the standard ASCII character set. To facilitate reproduction in the widest variety of media, the standard has been composed with the standard ASCII character set, even at the cost of simplifying the representation of certain non-English words. Other character sets, such as Unicode, are required to correctly represent addresses that use other languages. The character set should be specified in the file-level metadata for any address file.

1.3 Applicability

This standard is intended for use within and among federal, state, regional, local government agencies, nongovernmental sectors, and the general public.

1.4 Related Standards

This standard incorporates references to over 40 other standards and specifications. Appendix A (Informative) gives complete references to the standards and specifications cited, as well as to other standards and guidelines consulted in writing the standard.

This standard was written to conform to the FGDC *Standards Reference Model* (FGDC 1996). In the terms defined by that model, this standard is a data standard. Specifically, this standard has four parts: a data content standard (Part One), a data classification standard (Part Two), a data usability (Part Three), and a data transfer standard (Part Four). This standard does not include a data symbology or presentation standard.

This standard incorporates by reference, for address data files, the FGDC''s *Content Standard for Digital Geospatial Metadata* (CSDGM)(FGDC 1998). This standard extends the CSDGM by providing attributes for record-level address metadata. These attributes overlap to some extent with the CSDGM. If the values of these attributes are the same for all records in an address data file, the information can be omitted from the individual records and provided in the file-level metadata. If the values vary from record to record (e.g., in a file aggregated from multiple sources), the attributes can be included in the record-level metadata.

This standard is consistent with all parts of the FGDC's *Framework Data Content Standard of the National Spatial Data Infrastructure*. In particular, it conforms to all provisions of the Base part of the Framework Standard, which defines the abstract model that underlies and unifies the seven data themes. Appendix J shows this in detail. The address standard can therefore be used in conjunction with all of the National Spatal Data Infrastructure data themes. *Temporary Note to Reviewers: Consistency has not been verified by all framework maintenance committees. Certain inconstencies remain between this standard and the Transportation Part (Roads and Transit). These are listed in Appendix J.2.8*

USPS Publication 28, *Postal Addressing Standards*, is a foundational work for the Content and Classification Parts of this standard. USPS Publication 28 is the basis for the United States

profile of the template and rendition instructions in the Universal Postal Union *International postal address components and templates* (UPU 2008). The *Postal Addressing Profile* establishes the relationship between the FGDC standard and USPS Publication 28. The profile restricts this standard in some ways, and extends it in other ways, to incorporate the specific rules, abbreviations, and scope limitations of USPS Publication 28. Any address record that is standardized as defined within the terms of USPS Publication 28 is also compliant with the Postal Addressing Profile and, if altered according specific procedures described therein, will conform to this standard. *Temporary Note to Reviewers: As of 14 January 2010, the Postal Addressing Profile is under review by the USPS.*

This standard explicitly incorporates, as the Four Number Address Range class, the TIGER/Line file structure established by the U.S. Census Bureau for street segment address ranges (U.S. Census Bureau 2008).

During the time this standard has been developed, the National Emergency Number Association (NENA) has developed the *Next Generation 9-1-1 (NG9-1-1) Civic Location Data Exchange Format (CLDXF) Standard* to support the exchange of United States civic location address information about 9-1-1 calls. The CLDXF is the United States profile of the Internet Engineering Task Force (IETF) Presence Information Data Format – Location Object (PIDF-LO) civicAddress type. The FGDC and NENA working groups have aligned the two standards as closely as possible within the constraints of their respective purposes. To clarify the relation between the two standards, and to facilitate and standardize the conversion of address records between FGDC conformance and CLDXF conformance, the two committees have written the *Profile Reconciling the FGDC United States Thoroughfare, Landmark, and*

Postal Address Data Standard and the NENA Next Generation 9-1-1 (NG9-1-1) Civic Location Data Exchange Format (CLDXF) Standard. Temporary Note to Reviewers: As of 14 January 2010, the profile is under review by NENA.

The pseudocode for the data quality tests was written (with a few exceptions, all noted) using standard ISO/IEC 9075-1:2008 SQL. Spatial predicates used in the pseudocode are described in OGC's "OpenGIS Simple Features Specification for SQL" (Rev 1.1).

The XSD conforms to the W3C XML Core Working Group "Extensible Markup Language (XML) 1.0" (Third Edition, W3C Recommendation 4 February 2004). Geometry elements are defined and implemented following OGC's. "OpenGIS(R) Geography Markup Language (GML)" (Version: 3.1.1).

1.5 Standards development procedures

1.5.1 Antecedents

This standard builds on the Address Data Content Standard previously proposed by the FGDC (Public Review Draft, April 17, 2003).

1.5.2 The Address Standard Working Group (ASWG)

The FGDC efforts led the Urban and Regional Information Systems Association (URISA) to propose, with the support of the National Emergency Number Association (NENA) and the U.S. Census Bureau, the convening of an Address Standard Working Group (ASWG) to include representatives from a range of interested federal, state, regional, and local

government agencies, the private sector, and professional associations. The proposal was accepted by the FGDC Standards Working Group on April 13, 2005. The ASWG has worked under the authority of the Census Bureau, which chairs the FGDC Subcommittee on Cultural and Demographic Data (SCDD).

The ASWG prepared a draft standard, which was posted for public comment in August-September of 2005. A second draft was posted for public comment in December 2005 and January 2006. Since then, the ASWG has developed the standard further, by responding to additional comments and conference discussions, drafting additional material, integrating related standards, and preparing the final version for submittal to the FGDC.

1.5.3 Standard Development Process

Because addresses are created by such decentralized processes, and because the standard must satisfy such a wide range of requirements, the ASWG has sought by a variety of means to make the development process as open and broad-based as possible. This has involved:

1.5.3.1 Fostering Broad Awareness and Participation.

The ASWG has sought by various means to make the geospatial and addressing communities aware of the development of the standard and to involve as many as possible in the effort. The ASWG invited participation from and via professional associations representing geospatial professionals, local government officials, and emergency responders, including the National Association of Counties (NACO), GITA (Geospatial Information Technology Association), the American Association of Geographers (AAG), URISA, NSGIC (National States Geographic Information Council), and NENA (National Emergency Number Association).

The draft standards, when posted, were widely announced in the geospatial and standards online media. ASWG members have made numerous presentations on the standard at conferences and meetings. In addition, the ASWG has regularly briefed various federal groups, especially the FGDC and Census, about progress on the standard.

1.5.3.2 Using a Wiki Collaborative Website.

To encourage wide participation, the ASWG set up an interactive wiki web-site using free and open-source software (TWiki, from http://twiki.org/). Wiki software posts a draft document (in this case, the working draft of the standard) on a server and enables anyone to edit or comment on it via internet. Comments and changes, once saved, are immediately visible to all. Anyone can add comments and ideas, or join in discussions (and sometimes arguments) over various aspects of the standard.

The ASWG wiki site is open to anyone providing a name and a valid email to which to send a password. (The site is password protected only to keep out spam.) Over 400 individuals have signed up to view the site, provide comments, enter discussions and participate in the development of the standard. The wiki site has fostered discussion among widely scattered individuals, and proven useful in obtaining information and debating points of concept, practice, and actual address conditions.

1.5.3.3 Posting Drafts for Public Comment via Webform.

The ASWG posted a first draft on the standard two months after starting work, in the summer of 2005. It was posted on the URISA website, with copies available for download, and all comments were submitted via webform so that as many people as possible had access. Over

125 comments were received on this draft. A second draft was posted in December 2005, which received over 180 comments. The Committee has since made significant revisions to incorporate these comments, and to respond to issues that they raised. This is an unprecedented level of review for a standard that has not been officially submitted as a draft to FGDC. Wide early review has greatly improved the quality of the draft that will be formally submitted to FGDC, and, we hope, increased interest in reviewing the final draft.

1.5.3.4 Focusing on Practical Needs and Usefulness.

The ASWG's purpose has been to create a standard that will be useful and used. To be useful, the standard must reflect and build on the processes of address creation, management, and use. The standard must be developed by people who understand the local business work flows that utilize addresses in a real-time environment. Therefore the ASWG has sought advice and comment from a wide range of practitioners, including among others local government GIS managers, planners, assessors, emergency responders, school district officials, election officials, software developers, data aggregators, postal officials, census geographers, and a newspaper delivery manager, to name a few.

1.6 Maintenance authority

The Census Bureau will maintain the standard under the auspices of its duties as theme lead for the FGDC Subcommittee on Cultural and Demographic Data (SCDD), ensuring that the standard is revisited on the 5-year schedule as stipulated, or updating and revising as necessary. Direct any questions to Chief, Geography Division, U.S. Bureau of the Census.

1.7 Acronyms Used in the Standard

- AIS - Address Information System (USPS)
- ALI - Automatic Location Information
- ANSI - American National Standards Institute
- APO - Army Post Office
- ASWG - Address Standard Working Group
- CASS - Coding Accuracy Support System (USPS)
- CLDXF - Civic Location Data Exchange Format (NENA NG9-1-1 CLDXF)
- CMR - Common Mail Room
- CMRA - Commercial Mail Receiving Agency
- CRS - Coordinate Reference System
- CSDGM - Content Standard for Digital Geospatial Metadata (FGDC)
- DMM - Domestic Mail Manual (USPS)
- DPO - Diplomatic Post Office
    - EPSG Dataset - European Petroleum Survey Group Geodetic Parameter Dataset (OGP)
- EPA - Environmental Protection Agency
- ERD - Entity Relationship Diagram
- FGDC - Federal Geographic Data Committee
- FIPS - Federal Information Processing Standard
- FPO - Field Post Office, or Fleet Post Office
- GIS - Geographic Information System
- GML - Geography Markup Language (OGC)

- GNIS - Geographic Names Information System
- GPS - Global Positioning System
- GZD - Grid Zone Designation
    - HC - Contract Delivery Service Route (formerly Highway Contract Route, and still abbreviated as HC)(USPS)
- ID - Identifier
- IETF - Internet Engineering Task Force
    - INCITS L1- InterNational Committee for Information Technology Standards Technical Committee L1 (Geographic Information Systems) (accredited by ANSI)
- ISO - International Standards Organization
    - ITU-T - International Telecommunications Union Telecommunication Standardization Sector
- LRM - Linear Reference Method
- LRS - Linear Reference System
- MAF - Master Address File (Census Bureau)
- MSAG - Master Street Address Guide
- MGRS - Military Grid Reference System
- NAD83 - North American Datum of 1983
- NCITS - National Committee for Information Technology Standards
- NENA - National Emergency Number Association
- NG9-1-1 - Next-Generation 9-1-1
- NSDI - National Spatial Data Infrastructure

- PIDF-LO - Presence Information Data Format - Location Object (IETF)
- OGC - Open Geospatial Consortium
    - OGP - International Association of Oil and Gas Producers (the OGP Geodesy Subcommittee maintains and publishes EPSG Dataset)
- PMB - Private Mail Box
- PO Box - Post Office Box
- PSC - Postal Service Center
- RFC - Request for Comments (IETF)
- RR - Rural Route (USPS)
- SCDD - FGDC Subcommittee on Cultural and Demographic Data
- SDTS - Spatial Data Transfer Standard (FGDC and USGS)
- SWG - FGDC Standards Working Group
    - TIGER - Topologically Integrated Geographic Encoding and Referencing System (Census Bureau)
- UML - Unified Modeling Language
- UPU - Universal Postal Union
- URISA - Urban and Regional Information Systems Association
- USGS - United States Geological Survey
- USNG - United States National Grid
- USPS - United States Postal Service
- UTM - Universal Transverse Mercator
- UUID - Universally Unique Identifier
- XML - Extensible Markup Language

- XSD - XML Schema Document
- ZIP Code - Zoning Improvement Plan Code (USPS)

1.8 Trademark Acknowledgements

The following trademarks are owned by the United States Postal Service: CASS™, PO Box™, U.S. Postal Service®, United States Post Office®, United States Postal Service®, USPS®, ZIP + 4®, ZIP Code™, ZIP™

The following trademark is owned by the Open Geospatial Consortium: OpenGIS®

2. Address Data Content

2.1 Purpose

The content part defines address elements, address reference system elements, and their attributes.

2.2 Organization

The address elements are presented first, grouped according to the major components of an address, followed by the address reference system elements, and lastly the attributes, which are grouped by subject.

Address Elements

- Address Number Elements
- Street Name Elements
- Subaddress Elements
- Landmark Name Elements
- Place, State, and Country Name Elements
- USPS Postal Address Elements
- USPS Address Lines

Address Reference System Elements

Attributes

- Address ID
- Address Coordinates
- Address Parcel IDs
- Address Transportation Feature IDs
- Address Range Attributes
- Address Attributes
- Element Attributes
- Address Lineage Attributes

2.3 Simple Elements, Complex Elements, and Attributes

The content part defines simple elements, complex elements, and attributes.

- Simple elements are address components or address reference system components that are defined independently of all other elements
- Complex elements are formed from two or more simple or other complex elements
- Attributes provide descriptive information, including geospatial information, about an address, an address reference system, or a specific element thereof..

Appendix B: Table of Element Relationships provides a list of all the elements, and their relations to each other.

2.4 Element and Attribute Definitions and Descriptions

Each data element is defined and described by giving its:

- **Element name:** The name of the element.
- **Other common names for this element:** Common words or phrases having the same or similar meaning as the element name. Note:
  * "(USPS)" indicates terms used in USPS Publication 28.
  * "(Census TIGER)" indicates terms found in Census TIGER\Line Shapefile documentation.
  * Appendix A gives complete citations for both documents.
- **Definition:** The meaning of the element.
- **Syntax:** (For complex elements only) What component elements are required or permitted to construct the element, and the order in which they must appear. (For syntax notation, see below, "Notation for Constructing Complex Elements.")

- **Definition Source:** The source of the definition ("New" indicates that the definition is original.)
- **Data Type:** Whether the element is a characterString, date, dateTime, integer, real, or geometric (point, MultiCurve, or MultiSurface) (see "Element and Attribute Data Types" below for definitions)
- **Existing Standards for this Element:** Other standards that govern this element (if any).
- **Domain of Values for this Element:** The range or set of values (if any) to which the element is restricted.
- **Source of Values:** The source (if any) for the domain of values.
- **How Defined:** How the domain of values is defined.
- **Example:** Illustrative examples of the element.
- **Notes/Comments:** Notes and comments giving further explanation about the element.
- **XML Tag:** The XML tag for the element.
- **XML Model:** XML model of the element.
- **XML Example:** The XML model applied to a specific example of the element.
- **XML Notes:** Explanatory notes about the XML model.
- **Quality Measures:** Quality tests applied to the class.
- **Quality Notes:** Explanatory notes about the quality measures applied to this element.

2.5 Element and Attribute Data Types

Elements and attributes are either non--geometric, geometric, or abstract. Non-geometric data types include characterString, date, dateTime, integer, and real. Geometric data types include

point, MultiCurve, and MultiSurface. The abstract data type, as used in this standard, aggregates multiple elements of different data types, geometric and non-geometric.

The non-geometric data types are defined in the FGDC's "Framework Data Content Standard Part 0: Base Document" (section 7.8.2.2 (Table 4 - CodeList for DataType)) as follows:

1. **characterString:** "A CharacterString is an arbitrary-length sequence of characters including accents and special characters from repertoire of one of the adopted character sets"
2. **date:** "Values for year, month, and day"
3. **dateTime:** "A combination of year, month, and day and hour, minute, and second"
4. **integer:** "Any member of the set of positive whole numbers, negative whole numbers and zero"
5. **real:** "Real numbers are all numbers that can be written as a possibly never repeating decimal fraction"

The geometric data types are defined in the Open Geospatial Consortium's "OpenGIS(R) Geography Markup Language (GML)" version 3.1.1 (see Appendix A for a complete citation):

- **Point:** "...a single coordinate tuple." (Sec. 10.3.1)
- **MultiCurve:** "...a list of curves. The order of the elements is significant and shall be preserved..." (Sec. 11.3.3.1). (The MultiCurve replaced the MultiLinestring datatype defined in GML version 3.0)

- **MultiSurface:** "...a list of surfaces. The order of the elements is significant and shall be preserved..." (Sec 11.3.4.1). (The MultiSurface replaced the MultiPolygon datatype defined in GML version 3.0)

The abstract data type is defined in the FGDC's "Framework Data Content Standard Part 0: Base Document" (Annex B.2.2) as a "class, or other classifier, that cannot be directly instantiated." The abstract data type (used in this standard for the complex element Address Reference System) may aggregate multiple elements of different data types, geometric and non-geometric.

2.6 Notation for Constructing Complex Elements

The following notation is used to show how complex elements are constructed from simple or other complex elements:

{} enclose the name of an element.

***** indicates that the element is **required** to create the complex element. Otherwise the element may be omitted when desired.

+ indicates "and" (concatenation), with a space implied between each component unless stated otherwise.

2.7 XML and GML Standard

XML models and examples conform to the W3C XML Core Working Group's "Extensible Markup Language (XML) 1.0" (see Appendix A for a complete citation). Geometry elements

are defined and implemented following OGC's. "OpenGIS(R) Geography Markup Language (GML)" (Version: 3.1.1).

2.8 Address Elements

2.8.1 Address Number Elements

2.8.1.1 Address Number Prefix

Element Name	Address Number Prefix
Other common names for this element	Street Number Prefix, Building Number Prefix, House Number Prefix, Site Number Prefix, Structure Number Prefix
Definition	The portion of the Complete Address Number which **precedes** the Address Number itself.
Definition Source	New
Data Type	characterString
Existing Standards for this Element	None
Domain of Values for this Element	Can be created locally from existing values
Source of Values	Local
How Defined	Locally
Example	**N6W2** 3001 Bluemound Road **A** 19 Calle 117 **194-** 03 Fiftieth Avenue **Milepost** 1303 Alaska Highway
Notes/Comments	1. This element is not found in most Complete Address Numbers. When found, it should be separated from the Address Number so that the Address Number can be maintained as an integer for sorting and quality control tests. 2. Informally an Address Number and Address Number Prefix may be written with

	or without a space between them. Within this standard, the default assumption is that an empty space separates elements unless stated otherwise. The Attached Element can be used to indicate where the assumed space between the Address Number and Address Number Prefix has been omitted within an address file (see Attached Element for additional notes). 3. An Address Number Prefix is often separated from the Address Number by a hyphen. The hyphen may be included in the Address Number Prefix, or, alternatively, a Separator Element may be used to separate the Address Number from the Address Number Prefix in constructing the Complete Address Number (see Separator Element for additional notes). 4. Milepost numbers are often used to specify locations on limited-access roads such as interstate highways, and along highways and country roads where addressable features are too sparse to assign address numbers. Where it is useful to treat these as addresses, treat "Milepost" (or "Kilometer", in Puerto Rico) as an Address Number Prefix, and the milepost number as the Address Number.
XML Tag	<AddressNumberPrefix>
XML Model	<xsd:complexType name="AddressNumberPrefix_type"> <xsd:simpleContent> <xsd:extension base="xsd:string"> <xsd:attribute name="Separator" type="addr_type:Separator_type" /> </xsd:extension> </xsd:simpleContent> </xsd:complexType>
XML Example	<CompleteAddressNumber> **<AddressNumberPrefix Separator=" ">N6W2</AddressNumberPrefix>** <AddressNumber>3001</AddressNumber> </CompleteAddressNumber> <CompleteAddressNumber> **<AddressNumberPrefix Separator=" ">A</AddressNumberPrefix>** <AddressNumber>19</AddressNumber> </CompleteAddressNumber>
Quality Measures	Tabular Domain Measure Range Domain Measure Spatial Domain Measure Address Number Fishbones Measure
Quality Notes	Address number prefixes can include map-based information as grid coordinates, references to survey systems or references to sections of a subdivision or housing complex. Where a tabular domain of values are available the prefix can be tested against it. The measure chosen will depend on the type of domain involved. See the introduction to this section for a information on which measures to use.

796

797 **2.8.1.2 Address Number**

Element Name	AddressNumber
Other common names for this element	Street Number, Building Number, House Number, Site Number, Structure Number
Definition	The numeric identifier for a land parcel, house, building or other location along a thoroughfare or within a community.
Definition Source	New
Data Type	Integer
Existing Standards for this Element	None
Domain of Values for this Element	Can be created locally.
Source of Values	Local jurisdiction
Attributes Associated with this Element	Address Number Parity
How Defined	Based on local address ranges associated with individual streets and blocks.
Example	**123** Main Street
Notes/Comments	1. The Address Number is defined as an integer to support address sorting, parity (even/odd) definition, and in/out of address range tests. 2. The Address Number must be converted to a characterString when it is combined with the prefix and suffix into a Complete Address Number. 3. Some addresses may contain letters, fractions, hyphens, decimals and other non-integer content within the Complete Address Number. Those non-integer elements should be placed in the Address Number Prefix if they appear before the Address Number, or in the Address Number Suffix if they follow the Address Number.If necessary, the Separator Element can be used to separate the Address Number from the Address Number Prefix or Address Number Suffix elements in constructing the Complete Address Number. For example, if the New York City hyphenated address 194-03 ½ 50th Avenue, New York, NY 11365 were to be parsed rather than represented as a Complete Address Number: ---the Address Number Prefix would be "194", ---the Separator Element would be "-", ---the Address Number would be 3 (converted to "03" (text) in constructing the Complete Address Number), ---and the Address Number Suffix would be "1/2". 4. Special care should be taken with records where the Address Number is 0

	(zero). Occasionally zero is issued as a valid address number (e.g. Zero Prince Street, Alexandria, VA 22314) or it can be imputed (1/2 Fifth Avenue, New York, NY 10003 (for which the Address Number would be 0 and the Address Number Suffix would be "1/2")). More often, though, zero is shown because the Address Number is either missing or non-existent, and null value has been converted to zero.
XML Tag	<AddressNumber>
XML Model	<xsd:simpleType name="AddressNumber_type"> <xsd:restriction base="xsd:string"> <xsd:pattern value='[0-9]+' /> </xsd:restriction> </xsd:simpleType>
XML Example	<CompleteAddressNumber> **<AddressNumber>1234</AddressNumber>** </CompleteAddressNumber>
Quality Measures	Data Type Measure
Quality Notes	The Address Number element is specified as an integer. Data Type Measure is helpful when testing data held in staging tables with variable character fields. Additional tests for the address number require association with a street name.

2.8.1.3 Address Number Suffix

Element Name	**AddressNumberSuffix**
Other common names for this element	Street Number Suffix, Building Number Suffix, House Number Suffix, Fractional Street Number (USPS), Structure Number Suffix
Definition	The portion of the Complete Address Number which **follows** the Address Number itself.
Definition Source	New
Data Type	characterString
Existing Standards for this Element	None
Domain of Values for this Element	Can be created locally from existing values
Source of Values	Local
How Defined	Locally

Example	123 **1/2** Main Street 121 **E** E Street B317 **A** Calle 117 Milepost 34.4 (Address Number Suffix = decimal portion only)
Notes/Comments	1. This element is not found in most Complete Address Numbers. When found, it should be separated from the Address Number so that the Address Number can be maintained as an integer for sorting and quality control tests. 2. Informally an Address Number and Address Number Suffix may be written with or without a space between them. Within this standard, the default assumption is that an empty space separates elements unless stated otherwise. The Attached Element can be used to indicate where the assumed space between the Address Number and Address Number Suffix has been omitted within an address file (see Attached Element for additional notes). 3. An Address Number Suffix is often separated from the Address Number by a hyphen. The hyphen may be included in the Address Number Suffix, or, alternatively, a Separator Element may be used to separate the Address Number from the Address Number Suffix in constructing the Complete Address Number (see Separator Element for additional notes). 4. When milepost Complete Address Numbers include decimal fractions, the integer portion of the milepost number is treated as the Address Number, and the fraction (including the decimal point) is treated as an Address Number Suffix. (See Complete Address Number for additional notes on milepost address numbers.
XML Tag	<AddressNumberSuffix>
XML Model	<xsd:complexType name="AddressNumberSuffix_type"> <xsd:simpleContent> <xsd:extension base="xsd:string"> <xsd:attribute name="Separator" type="addr_type:Separator_type" /> </xsd:extension> </xsd:simpleContent> </xsd:complexType>
XML Example	<CompleteAddressNumber> <AddressNumber>123</AddressNumber> **<AddressNumberSuffix separator=" ">1/2</AddressNumberSuffix>** </CompleteAddressNumber> <CompleteAddressNumber> <AddressNumber>456</AddressNumber> **<AddressNumberSuffix separator=" ">B</AddressNumberSuffix>** </CompleteAddressNumber>

	<CompleteAddressNumber> <AddressNumber>317</AddressNumber> **<AddressNumberSuffix separator=" ">A</AddressNumberSuffix>** </CompleteAddressNumber>
Quality Measures	Tabular Domain Measure Spatial Domain Measure Address Number Fishbones Measure
Quality Notes	1. Address number suffixes can include references to sections of a subdivision or housing complex. Where a tabular domain of values are available the prefix can be tested against it. 2. When geometry for both the address point and an areal Address Number Suffix are available the Spatial Domain Measure can be used to measure tests whether the addressed location is within a polygon describing a map-based Address Number Suffix. 3. Use Address Number Fishbones Measure when geometry for both the address point and a linear spatial domain for Address Number Suffix are available. This measure tests whether the addressed location is along a line describing a map-based Address Number Suffix.

2.8.1.4 Separator Element

Element Name	**Separator Element**
Other common names for this element	
Definition	A symbol, word, or phrase used as a separator between components of a complex element or class. The separator is required for Intersection Addresses and for Two Number Address Ranges, and it may be used in constructing a Complete Street Name or a Complete Address Number.
Definition Source	New
Data Type	characterString
Existing Standards for this Element	None
Domain of Values for this Element	Typical values may include: 1. For Intersection Addresses: "and", "at", "@", "&", and "&&" "+","-", and "y" or "con" (Spanish) each having a space before and after. 2. For Two Number Address Ranges: - (hyphen)(spaces optional before or after) 3. If a Complete Street Name includes a prepositional phrase between the

	between a Street Name Pre Type and a Street Name, the prepositional phrase is treated as a separator: "of the", "de la", "des", etc. 4. Complete Address Numbers: - (hyphen)(spaces optional before or after)
Source of Values	New
How Defined (e.g., locally, from standard, other)	Locally.
Example	1. Intersection Address ("and"): Eighth Street **and** Pine Street. 2. Two Number Address Range (hyphen): **206-210** Fourth Street 3. Prepositional phrase between the Street Name Pre Type and the Street Name:("of the", "de las" and "des") Avenue **of the** Americas, Alameda **de las** Pulgas; Rue **des** Fleurs. 4. Complete Address Number (hyphen): **61-43** Springfield Boulevard
Notes/Comments	1. The default separator, an empty space, is implicit and is not shown in the syntaxes of complex elements and classes. 2. An explicit separator is required for Two Number Address Ranges and Intersection Addresses. It is sometimes required in constructing Complete Street Names. 3. For Complete Address Numbers, the separator is rarely needed and its use should be minimized. As an alternative, the separator symbol usually can be included with the Address Number Prefix or Address Number Suffix. 4. The Separator Element is not needed in creating fractions (1/2, etc.) for Address Number Suffixes. 5. Within a given dataset, one value should be used consistently within a given complex element. 6. Some address parsing software permits the use of ampersands ("&" or "&&") to signify intersection addresses. Be wary, though--in many programming languages, ampersands are reserved for other uses, which could complicate data exchange.
XML Tag	Separator
XML Model:	<xsd:simpleType name="Separator_type"> <xsd:restriction base="xsd:string"> <xsd:pattern value='.*' /> </xsd:restriction> </xsd:simpleType>
XML Example:	<IntersectionAddress **Separator="and"** > <CompleteStreetName> <StreetName>EIGHTH</StreetName> <StreetNamePostType>STREET</StreetNamePostType> </CompleteStreetName> <CompleteStreetName> <StreetName>PINE</StreetName>

	<StreetNamePostType>STREET</StreetNamePostType> </CompleteStreetName> <PlaceName PlaceNameType="USPSCommunity">ELLICOT CITY</PlaceName> <StateName>MD</StateName> <ZIPCode>21043</ZIPCode> </IntersectionAddress>
	<AddressNumberRange **Separator=" - "** > <CompleteAddressNumber> <AddressNumber>206</AddressNumber> </CompleteAddressNumber> <CompleteAddressNumber> <AddressNumber>210</AddressNumber> </CompleteAddressNumber> </AddressNumberRange>
	<CompleteStreetName> <StreetName>AVENUE</StreetName> <StreetNamePostType **Separator="of the"** >AMERICAS</StreetNamePostType> </CompleteStreetName>
	<CompleteStreetName> <StreetName>ALAMEDA</StreetName> <StreetNamePostType **Separator="de las"** >PULGAS</StreetNamePostType> </CompleteStreetName>
	<CompleteAddressNumber> <AddressNumber>61</AddressNumber> <AddressNumberSuffix **Separator="-"** >43</AddressNumberSuffix> </CompleteAddressNumber>
XML Notes:	This entity must be expressed as an empty string to indicate an empty string. Omitting the entity entirely indicates that a space is acceptable.
Quality Measures	Tabular Domain Measure
Quality Notes	If Separator Element entries are maintained within a database, rather than generated as part of a query, they may be tested with Tabular Domain Measure. Their use depends on other elements, and is tested at the classification level.

2.8.1.5 Complex Element: Complete Address Number

Element Name	Complete Address Number
Other common names for this element	Complete street number, full street number, Primary Address Number (USPS), Street Number (USPS), House Number (USPS, Census TIGER))
Definition	An Address Number, alone or with an Address Number Prefix and/or Address Number Suffix, that identifies a location along a thoroughfare or within a community.
Syntax	{ Address Number Prefix } + { Separator Element } + { Address Number *} + { Separator Element } + { Address Number Suffix }
Definition Source	New
Data Type	characterString
Existing Standards for this Element	Refer to component simple elements
Domain of Values for this Element	Refer to component simple elements
Source of Values	Refer to component simple elements
How Defined (e.g., locally, from standard, other)	Refer to component simple elements
Example	**123** Main Street **123 A** Main Street **123 1/2** Main Street **A 19** Calle 117 **0** Prince Street **0 1/2** Fifth Avenue **Milepost 240** Parks Highway Alaska **Milepost 72.9** Interstate 84, Wasco County, OR **Kilometer 0.5** Carretera 917, Urbanizacion April Gardens, Las Piedras PR 00771 **Kilometer 2 Hectometer 7** Carretera 175, Barrio San Antonio, Caguas, Puerto Rico 00725 **N89W16758** Appleton Avenue, Menomonee Falls, WI 53051 **W63N645** Washington Avenue, Cedarburg, WI 53012 **5-5415** Kuhio Highway, Hanalei, HI 96714 **194-03 1/2** 50th Avenue, New York, NY 11365
Notes/Comments	1. The Address Number element is required to compose a Complete Address Number. The other elements are optional. 2. The Address Number must be converted from integer to characterString

	when constructing the Complete Address Number. 3. The great majority of Complete Address Numbers are simple integers. Infrequently the integer is followed by an alphanumeric Address Number Suffix, typically a letter or a fraction. Even more rarely the integer is preceded by an alphanumeric Address Number Prefix. In addition to the typical numbering format, four special-case formats are found in the United States: Milepost addresses, grid-style address numbers, hyphenated address numbers, and other Address Number Prefix letters or symbols. 4. **Milepost** Complete Address Numbers**.** Road mileposts are sometimes used to specify locations along highways and similar roads. Mileposts are often used to locate, for example, crash sites, emergency call boxes, bridge locations, inspection stations, roadside rest stops, railroad crossings, highway exits, park and campground entrances, RV parks, and truck stops. Milepost addresses should be parsed as follows: ---"Milepost" (or equivalent word or phrase, such as "kilometer" or 'Mile Marker") is an Address Number Prefix ---The milepost number (integer part only) is an Address Number ---Tenths, if given, are an Address Number Suffix, including the decimal point. ---The road name or highway route number is a Complete Street Name, and parsed accordingly Note that, in Puerto Rico, road measurements are given in kilometers (km), which are sometimes divided into hectometers (hm). 5. **Grid-style** Complete Address Numbers**.** In certain communities in and around southern Wisconsin, Complete Address Numbers include a map grid cell reference preceding the Address Number. In the examples above, "N89W16758" should be read as "North 89, West 167, Address Number 58". "W63N645" should be read as "West 63, North, Address Number 645." The north and west values specify a locally-defined map grid cell with which the address is located. Local knowledge is needed to know when the grid reference stops and the Address Number begins. 6. **Hyphenated** Complete Address Numbers**.** In some areas (notably certain parts of New York City, southern California, and Hawaii), Complete Address Numbers often include hyphens. Hyphenated Complete Address Numbers should not be confused with Two Number Address Ranges. The former is a single Complete Address Number while the latter includes two Complete Address Numbers. 7. Hyphenated Complete Address Numbers can be parsed so that the number indicating the site or structure is the Address Number, and the remainder (including the hyphen) is the Address Number Prefix or Address Number Suffix. If necessary, the hyphen can be parsed as a Separator Element, to separate it from both the Address Number and the Address Number Prefix or Address Number Suffix. However, the Separator Element is rarely needed and its use should be minimized in constructing Complete Address Numbers. 8. In New York City, hyphenated Complete Address Numbers (the recommended format for storing complete address numbers in New York City)

	follow a more complex set of rules. The number to the left of the hyphen indicates the "block" (conceptually--the number does not always change at street intersections and sometimes it changes within a single block face). The number to the right of the hyphen indicates the site or house number within the "block". If the Address Number is less than ten, it is written with a leading zero, as in **194-03 1/2** above. Additional leading zeros may be added to either number to provide for correct sorting if the entire Complete Address Number is treated as a characterString with the hyphen included. Within the address standard, these numbers can be constructed and parsed as follows: a. The left-side number (**194**)) is the Address Number Prefix element (text), with leading zeros shown as needed. b. The hyphen is a Separator Element with no spaces inserted before or after the hyphen when constructing the Complete Address Number. c. The right-side number (**3**) is the Address Number (integer), converted to a characterString with leading the zero(s) added (**03**) upon conversion to Complete Address Number. d. The suffix, if any (such as the **"1/2"** in **194-03 1/2**), is an Address Number Suffix. **9. Other** Address Number Prefix **Letters or Symbols.** In Puerto Rico, Address Numbers are commonly preceded by an Address Number Prefix letter (e.g. "A 19"). In Portland, OR, negative Address Numbers have been assigned in an area along the west bank of the Willamette River. The minus sign is represented as a leading zero ("0121" and "121" are two different Complete Address Numbers). In such cases the leading zero should be treated as an Address Number Prefix. **10. Zero as a** Complete Address Number**.** Special care should be taken with records where the Address Number is 0 (zero). Occasionally zero is issued as a valid address number (e.g. 0 Prince Street, Alexandria, VA 22314) or it can be imputed (1/2 Fifth Avenue, New York, NY 10003, for which the Address Number would be 0 and the Address Number Suffix would be "1/2"). More often, though, the Address Number is either missing or non-existent, and null value has been converted to zero.
XML Tag	<CompleteAddressNumber>
XML Model	<xsd:complexType name="CompleteAddressNumber_type"> <xsd:sequence> <xsd:element name="AddressNumberPrefix" type="addr_type:AddressNumberPrefix_type" minOccurs="0" maxOccurs="1" /> <xsd:element name="AddressNumber" type="addr_type:AddressNumber_type" minOccurs="1" maxOccurs="1" /> <xsd:element name="AddressNumberSuffix" type="addr_type:AddressNumberSuffix_type" minOccurs="0" maxOccurs="1" /> </xsd:sequence>

	<xsd:attribute name="AddressNumberParity" type="addr_type:AddressNumberParity_type" /> <xsd:attribute name="AttachedElement" type="addr_type:AttachedElement_type" /> </xsd:complexType>
XML Exmample	<CompleteAddressNumber> <AddressNumber>55</AddressNumber> <AddressNumberSuffix Separator="">1/2</AddressNumberSuffix> </CompleteAddressNumber>
	<CompleteAddressNumber> <AddressNumberPrefix Separator="">MILEPOST</AddressNumberPrefix> <AddressNumber>72.9</AddressNumber> </CompleteAddressNumber>
Quality Measures	Pattern Sequence Measure
Quality Notes	

2.8.2 Street Name Elements

2.8.2.1 Street Name Pre Modifier

Element Name	**Street Name Pre Modifier**
Other common names for this element	Prefix Qualifier (Census TIGER)
Definition	A word or phrase that precedes the Street Name and is not a Street Name Pre Directional or a Street Name Pre Type.
Definition Source	New
Data Type	characterString
Existing Standards for this Element	No
Domain of Values for this Element	Can be created locally from existing values
Source of Values	Local
How Defined	Locally

Example	**Old** North First Street **The** Croft Lane
Notes/Comments	1. Census Bureau TIGER Technical Documentation (Appendix D) provides the following list of Street Name Pre Modifiers: Alternate, Business, Bypass, Extended, Historic, Loop, Old, Private, Public, Spur 2. Parsing rules allow some flexibility in deciding whether a Complete Street Name includes a Street Name Pre Modifier. In each of the examples above, for instance, the entire name could be treated as the Street Name element. If the Complete Street Name is parsed into components, the Street Name Pre Modifier provides a way to handle words that precede the Street Name and should be separated from it, or that are separated from the Street Name by a Street Name Pre Directional or a Street Name Pre Type. See Complete Street Name notes for a general discussion of Complete Street Name parsing principles.
XML Tag	<StreetNamePreModifier>
XML Model	<xsd:complexType name="StreetNameModifier_type"> <xsd:simpleContent> <xsd:extension base="xsd:string"> <xsd:attribute name="Separator" type="addr_type:Separator_type"></xsd:attribute> </xsd:extension> </xsd:simpleContent> </xsd:complexType>
XML Example	<CompleteStreetName> **<StreetNamePreModifier>OLD</StreetNamePreModifier>** <StreetName>FIRST</StreetName> <StreetNamePostType>STREET</StreetNamePostType> <StreetNamePostDirectional>SOUTHWEST</StreetNamePostDirectional> </CompleteStreetName>
Quality Measures	Tabular Domain Measure Spatial Domain Measure
Quality Notes	1. Where a specific set of premodifiers are specified for use, they may be maintained as a domain and tested with Tabular Domain Measure. 2. Where a schema may designate a particular area with a Street Name Pre Modifier the entries may be tested with Spatial Domain Measure.

2.8.2.2 Street Name Pre Directional

Element Name	**Street Name Pre Directional**
Other common	Predirectional (USPS), Prefix Direction (Census TIGER), Prefix Directional,

names for this element	Predir
Definition	A word preceding the street name that indicates the directional taken by the thoroughfare from an arbitrary starting point, or the sector where it is located.
Definition Source	New
Data Type	characterString
Existing Standards for this Element	USPS Publication 28 Section 233 and 294
Domain of Values for this Element	English: East, West, South, North, Northeast, Southeast, Southwest, Northwest Spanish: Este, Oeste, Sur, Norte; Noreste, Sureste, Suroeste, Noroeste Equivalent words in other languages
Source of Values	USPS Publication 28 Sections 233 and 294 (unabbreviated)
How Defined	As provided by USPS Publication 28 Section 233 and 294
Example	123 **North** Main Street 123 **West** North Street North Avenue (directional word is the Street Name) South Carolina Avenue (directional word is part of the Street Name)
Notes/Comments	1. USPS Publication 28 recommends abbreviating pre-directionals. The Standard requires storing pre-directionals fully spelled out, exactly as given by the local naming authority, to avoid confusion. For example: "N W Jones St": Is it Northwest Jones Street? Ned Walter Jones Street? North Walter Jones Street? The abbreviations create ambiguity. If stored unabbreviated, directionals can be exported as standard abbreviations as needed for mailing and other purposes. 2. USPS standard abbreviations are recognized within the Postal Addressing Profile of this standard. USPS Publication 28 sections 233, 294, and Appendix B provide the USPS abbreviations for directionals in English and Spanish. 3. Directional words are often used as or in the Street Name (e.g. North Avenue, West Virginia Avenue). The proper parsing must be inferred from the syntax and context of the street name. (For example, does West Virginia Avenue at some point change names and become East Virginia Avenue? Then perhaps "Virginia" is the Street Name, and East and West are Street Name Pre Directionals.) See Complete Street Name for a general discussion of street name parsing principles. 4. USPS Publication 28 (paraphrased to omit reference to abbreviations): *"233.21 Predirectional Field* -- When parsing the address from right to left, if a directional word is found as the first word in the street name and there is no other directional to the left of it, ...locate it in the predirectional field..." 5. See Street Name Post Directional for additional USPS Publication 28 notes that also apply to this element.
XML Tag	<StreetNamePreDirectional>

XML Model	<xsd:complexType name="StreetNameDirectional_type"> <xsd:simpleContent> <xsd:extension base="xsd:string"> <xsd:attribute name="Separator" type="addr_type:Separator_type"></xsd:attribute> </xsd:extension> </xsd:simpleContent> </xsd:complexType>
XML Example	<CompleteStreetName> **<StreetNamePreDirectional>NORTH</StreetNamePreDirectional>** <StreetName>MAIN</StreetName> <StreetNamePostType>STREET</StreetNamePostType> </CompleteStreetName>
Quality Measures	Tabular Domain Measure Spatial Domain Measure
Quality Notes	1. Tabular Domain Measure can test entries against a tabular domain. 2. In cases where an address scheme designates particular areas as corresponding with a given Street Name Pre Directional and the geometry for both the streets and the address scheme's spatial domain, Tabular Domain Measure can test the entries.

2.8.2.3 Street Name Pre Type

Element Name	**Street Name Pre Type**
Other common names for this element	Prefix type (Census TIGER), Street prefix type, Pre-type
Definition	The element of the complete street name **preceding** the street name element that indicates the type of street.
Definition Source	New
Data Type	characterString
Existing Standards for this Element	None (Appendix C1 of USPS Publication 28 provides a useful list of Street Suffixes, but does not recognize their use for Street Name Pre Types)

Domain of Values for this Element	Yes. Although not recognized as Street Name Pre Types, Appendix C1 of USPS Publication 28 contains a useful list of Street Suffixes. Development of a list of Street Name Pre Types can incorporate Street Suffixes from USPS Publication 28 Appendix C1 with local additions.
Source of Values	Although not recognized as Street Name Pre Types, Section 234 and Appendix C of USPS Publication 28 contains a useful list of Street Types. Development of a list of Street Name Pre Types can incorporate Street Types from USPS Publication 28 with local additions.
How Defined	By local addressing authority.
Example	**Avenue** A **Calle** Aurora **Avenue** of the Americas **Avenue** at Port Imperial **Alameda** de las Pulgas **Rue** d' Armour **Avenue** C Loop
Notes/Comments	1. Street Name Pre Types are recognized in this standard but not in USPS Publication 28. Within USPS Publication 28, Street Name Pre Types are combined into the USPS Primary Street Name. This practice is not recommended by the Address Standard as it complicates quality assurance testing of street names. USPS Publication 28 provides the most complete list of Street Suffixes, but it is not exhaustive. 2. USPS Publication 28 provides a standard list of street type abbreviations in Appendix C1 and Appendix H, and recommends their use. The Address Standard requires storing Street Name Pre Types and Street Name Post Types fully spelled out, exactly as given by the local naming authority, to avoid confusion. If stored unabbreviated, they can be exported as standard abbreviations as needed for mailing and other purposes. USPS Abbreviations are recognized within the Postal Addressing Profile of this standard. 3. Street Name Pre Types are much less common than Street Name Post Types in English. Street Name Pre Types are much more common in Spanish-, French-, and Italian-language street names. 4. If a prepositional phrase appears between the Street Name Pre Type and the Street Name, the prepositional phrase is considered a Separator Element: Avenue **of the** Americas, Alameda **de las** Pulgas. Such constructions are rare in English-language Complete Street Names, but they are common in Spanish, Italian and French. 5. A Complete Street Name usually includes either a Street Name Pre Type or a Street Name Post Type. Occasional Complete Street Names have neither ("Broadway") or both ("Avenue C Loop"). Parsing rules should be consistently applied. For example, if a jurisdiction parses "Avenue C" as a Street Name Pre Type plus a Street Name, then "Avenue C Loop" should be parsed as a Street Name Pre Type, Street Name, and Street Name Post Type. 6. See Complete Street Name notes for a general discussion of Complete

	Street Name parsing principles.
XML Tag	<StreetNamePreType>
XML Model	<xsd:complexType name="StreetNameType_type"> <xsd:simpleContent> <xsd:extension base="xsd:string"> <xsd:attribute name="Separator" type="addr_type:Separator_type"></xsd:attribute> </xsd:extension> </xsd:simpleContent> </xsd:complexType>
XML Example	<CompleteStreetName> **<StreetNamePreType>AVENUE</StreetNamePreType>** <StreetName>C</StreetName> <StreetNamePostType>LOOP</StreetNamePostType> </CompleteStreetName>
Quality Measures	Tabular Domain Measure Spatial Domain Measure
Quality Notes	1. Tabular Domain Measure can test entries against a tabular domain. 2. In cases where an Address Reference System designates particular areas as corresponding with a given Street Name Pre Type and the geometry for both the streets and the address scheme's spatial domain, Tabular Domain Measure can test the entries. 3. In some cases a jurisdiction may have associated specific Street Name Pre Type entries with functional aspects of the road that require additional local quality measures. For example, a court may be required to be a dead end, or a boulevard limited to streets divided by a median. While these associations are beyond the scope of the standard they should be considered in planning a quality program for local addresses.

2.8.2.4.Street Name

Element Name	Street Name
Other common names for this element	Primary Street Name, Base Name (Census TIGER)
Definition	Official name of a street as assigned by a local governing authority, or an alternate (alias) name that is used and recognized, excluding street types, directionals, and modifiers.
Definition Source	Adapted from FGDC Draft Address Data Content Standard v. 3 (citing

	Census)
Data Type	characterString
Existing Standards for this Element	Section 232 of USPS Publication 28
Domain of Values for this Element	Official list of street names maintained by local authority.
Source of Values	Local
How Defined	Defined by local ordinance
Example	**Central** Street Southwest **MacIntyre** Drive **Boston-Providence** Highway **Third** Avenue **3rd** Avenue
Notes/Comments	**1. Domain of Values:** Each jurisdiction should establish its own list of street names and use it as a domain of values to validate addresses. Alternate and Official names are distinguished by the Official Status attribute. **2. Use of Alternate or Alias Names:** If alternate or abbreviated versions of street names are needed for a specialized purpose such as mailing or emergency dispatch, they can be created in views or export routines. **3. Spelling Consistency: Internal Capitalization, Apostrophes, Hyphens, Spaces** Local addressing authorities are urged to follow consistent internal street naming practices, and to resolve internal street name inconsistencies, especially for internal capitalization ("McIntyre" or "Mcintyre" ?), hyphens, and apostrophes. **Example:** MacIntyre, McIntyre, Mc Intyre, Mcintyre **Example:** Smiths Lane, Smith's Lane **Example:** Boston Providence Highway; Boston-Providence Highway; **Rule:** Follow the spelling adopted by the local street naming authority. **Discussion:** This standard cannot specify local naming conventions. **4. Numbered Streets** **Examples:** Third Street, 3rd Street, 3 Street **Rule:** Use the name exactly as given by the local street naming authority. **Discussion:** This standard cannot specify local naming conventions. Different jurisdictions follow different practices for numbered street names. Pittsburgh spells out "First" through "Twelfth" and uses ordinal numbers ("13th", 14th, etc.) for higher numbers. Washington DC uses ordinal numbers only (1st, 2nd, etc.). Other jurisdictions have their own conventions. This is a matter for local authorities to decide.

	5. Parsing Ambiguous Complete Street Names: Some Complete Street Names can be parsed in more than one way. For example: -- **County Road 88** or County Road **88** -- **East River** Avenue or East **River** Avenue -- **The Croft** Lane or The **Croft** Lane -- **Boulevard of the Allies** or Boulevard of the **Allies** This standard accommodates any of the above choices. As a matter of guidance local authorities may prefer to parse the Complete Street Name so that the Street Name element can be used to create a sorted alphanumeric list of names. By this principle the first set of parsings would give the following sorted list: **-- Boulevard of the Allies** **-- County Road 88** -- **East River** Avenue -- **The Croft** Lane The second set of parsings would give a different list: -- **88,** County Road -- **Allies,** Boulevard of the -- **Croft** Lane, The -- **River** Avenue, East **6. Additional Discussion of Street Name Parsing:** See Complete Street Name for a general discussion of street name parsing principles.
XML Tag	<StreetName>
XML Model	<xsd:simpleType name="StreetName_type"> <xsd:restriction base="xsd:string"> <xsd:pattern value='.*' /> </xsd:restriction> </xsd:simpleType>
XML Example	<CompleteStreetName> **<StreetName>CENTRAL</StreetName>** <StreetNamePostType>STREET</StreetNamePostType> <StreetNamePostDirectional>SOUTHWEST</StreetNamePostDirectional> </CompleteStreetName> <CompleteStreetName> **<StreetName>BOSTON-PROVIDENCE</StreetName>** <StreetNamePostType>HIGHWAY</StreetNamePostType> </CompleteStreetName>
Quality Measures	Tabular Domain Measure

	Spatial Domain Measure
Quality Notes	In some cases a jurisdiction may have associated a given area with a type of street name: alpha characters, trees, flowers, birds, etc. Where such a scheme exists, along with the geometry for both the streets and the spatial domain, Spatial Domain Measure can be used to test conformance.

815

816 ### 2.8.2.5 Street Name Post Type

Element Name	**Street Name Post Type**
Other common names for this element	Street Type, Street Suffix, Street Suffix Type, Suffix (USPS), Suffix Type (Census TIGER)
Definition	The element of the complete street name **following** the street name element that indicates the type of street.
Definition Source	New
Data Type	characterString
Existing Standards for this Element	Section 234 and Appendix C of USPS Publication 28 with provision for local additions
Domain of Values for this Element	USPS Publication 28 Appendix C with provisions for local additions.
Source of Values	Section 234 and Appendix C of USPS Publication 28 with provision for local additions.
How Defined	Locally
Example	123 Central **Street** Southwest 123 MacIntyre **Drive** 123 Boston-Providence **Highway** 123 Third **Avenue** 123 3rd **Avenue** Avenue C **Loop**
Notes/Comments	1. USPS Publication 28 provides the most complete list of Street Name Post Types, but it is not exhaustive. Where a Street Name Post Type is not included in USPS Publication 28, the USPS requires that it be incorporated into the Street Name. This standard does not recommend following this practice. 2. USPS Publication 28 provides a standard list of street type abbreviations in Appendix C1 and Appendix H, and recommends their use. The Address Standard recommends storing Street Name Post Types fully spelled out, exactly as given by the local naming authority, to avoid confusion. If stored

	unabbreviated, they can be exported as standard abbreviations as needed for mailing and other purposes. USPS Abbreviations are recognized within the Postal Addressing Profile of this standard. 3. A Complete Street Name usually includes either a Street Name Pre Type or a Street Name Post Type. Occasional Complete Street Names have neither ("Broadway") or both ("Avenue C Loop"). Parsing rules should be consistently applied. For example, if a jurisdiction parses "Avenue C" as a Street Name Pre Type plus a Street Name, then "Avenue C Loop" should be parsed as a Street Name Pre Type, Street Name, and Street Name Post Type. 5. See Complete Street Name notes for a general discussion of Complete Street Name parsing principles.
XML Tag	<StreetNamePostType>
XML Model	<xsd:complexType name="StreetNameType_type"> <xsd:simpleContent> <xsd:extension base="xsd:string"> <xsd:attribute name="Separator" type="addr_type:Separator_type"></xsd:attribute> </xsd:extension> </xsd:simpleContent> </xsd:complexType>
XML Example	<CompleteStreetName> <StreetName>BOSTON-PROVIDENCE</StreetName> **<StreetNamePostType>HIGHWAY</StreetNamePostType>** </CompleteStreetName> <CompleteStreetName> <StreetNamePreType>AVENUE</StreetNamePreType> <StreetName>C</StreetName> **<StreetNamePostType>LOOP</StreetNamePostType>** </CompleteStreetName>
Quality Measures	Tabular Domain Measure Spatial Domain Measure
Quality Notes	1. Tabular Domain Measure can test entries against a tabular domain. 2. In cases where an Address Reference System designates particular areas as corresponding with a given Street Name Post Type and the geometry for both the streets and the address scheme's spatial domain, Tabular Domain Measure can test the entries. 3. In some cases a jurisdiction may have associated specific Street Name Post Type entries with functional aspects of the road that require additional local quality measures. For example, a court may be required to be a dead end, or a boulevard limited to streets divided by a median. While these associations are beyond the scope of the standard they should be considered in planning a

	quality program for local addresses.

2.8.2.6 Street Name Post Directional

Element Name	**Street Name Post Directional**
Other common names for this element	Postdirectional (USPS), Post Directional, Post-direction, Postdir, Suffix Directional, Suffix Direction (Census TIGER)
Definition	A word following the street name that indicates the directional taken by the thoroughfare from an arbitrary starting point, or the sector where it is located.
Definition Source	New
Data Type	characterString
Existing Standards for this Element	USPS Publication 28 Sections 233, 294 and Appendix B
Domain of Values for this Element	English: East, West, South, North, Northeast, Southeast, Southwest, Northwest Spanish: Este, Oeste, Sur, Norte; Noreste, Sureste, Suroeste, Noroeste Equivalent words in other languages
Source of Values	USPS Publication 28 Sections 233, 294 and Appendix B (unabbreviated)
How Defined	As provided by USPS Publication 28 Sections 233, 294 and Appendix B
Examples	Cherry Street **North** North Avenue **West**
Notes/Comments	1. USPS Publication 28 recommends abbreviating post-directionals. The Standard requires storing post-directionals fully spelled out, exactly as given by the local naming authority, to avoid confusion. For example: "N Avenue W"-- Is it "North Avenue W"? "N Avenue West"? "North Avenue West"? The abbreviations create ambiguity. If stored unabbreviated, directionals can be exported as standard abbreviations as needed for mailing and other purposes. 2. USPS standard abbreviations are recognized within the Postal Addressing Profile of this standard. USPS Publication 28 sections 233, 294, and Appendix B provide the USPS abbreviations for directionals in English and Spanish. 3. USPS Publication 28 Notes (paraphrased to omit reference to abbreviations): * *"233.22 Postdirectional Field* -- When parsing from right to left, if a directional word is located to the right of the street name and suffix, ..locate it

	in the postdirectional field. " * *"233.23 Two Directionals* -- When two directional words appear consecutively as one or two words, before the street name or following the street name or suffix, then the two words become either the pre- or the post-directionals. Exceptions are any combinations of NORTH-SOUTH or EAST-WEST as consecutive words. In these cases the second directional becomes part of the primary name and is spelled out completely in the street name element. * *"233.23 (Other Exception)* The other exception is when the local address information unit has determined that one of the directional letters (N, E, W, S) is used as an alphabet indicator and not as a directional." * *"233.3 Directional as Part of Street Name* -- If the directional word appears between the street name and the street type, then it should be considered part of the primary name and spelled out in that element. --Example: 12334 NORTH AVENUE (street name is "North"), --Example: 1234 WILD WEST STREET **SOUTH** (Street Name is "Wild West", "South" is a post-directional.) * *"233.3 (Alphabetical Indicators)* -- The exception is when the local AIS unit has determined that the letters (E, N, S, or W) are used as alphabet indicators and not as directionals [abbreviations]." --Example: "Avenue E". 4. In short, when parsing street names, types, directionals, and modifiers, the street name is required, and other elements are inferred from the context and syntax.
XML Tag	<StreetNamePostDirectional>
XML Model	<xsd:complexType name="StreetNameDirectional_type"> <xsd:simpleContent> <xsd:extension base="xsd:string"> <xsd:attribute name="Separator" type="addr_type:Separator_type"></xsd:attribute> </xsd:extension> </xsd:simpleContent> </xsd:complexType>
XML Example	<CompleteStreetName> <StreetName>CHERRY</StreetName> <StreetNamePostType>STREET</StreetNamePostType> **<StreetNamePostDirectional>NORTH</StreetNamePostDirectional>** </CompleteStreetName>

	<CompleteStreetName> <StreetName>NORTH</StreetName> <StreetNamePostType>AVENUE</StreetNamePostType> **<StreetNamePostDirectional>WEST</StreetNamePostDirectional>** </CompleteStreetName>
Quality Measures	Tabular Domain Measure Spatial Domain Measure
Quality Notes	1. Tabular Domain Measure can test entries against a tabular domain. 2. In cases where an address scheme designates particular areas as corresponding with a given Street Name Post Directional and the geometry for both the streets and the address scheme's spatial domain, Tabular Domain Measure can test the entries.

2.8.2.7 Street Name Post Modifier

Element Name	**Street Name Post Modifier**
Other common names for this element	Suffix Qualifier (Census TIGER)
Definition	A word or phrase that follows the Street Name but is not a Street Name Post Type or Street Name Post Directional.
Definition Source	New
Data Type	characterString
Existing Standards for this Element	No
Domain of Values for this Element	No
Source of Values	Local
How Defined (e.g., locally, from standard, other)	Locally
Example	East End Avenue **Extended** Grand Boulevard **Cutoff** Avenue A **Bypass** Concord Highway **Extension**

Notes/Comments	1. Census Bureau TIGER Technical Documentation (Appendix D) provides the following list of Street Name Post Modifiers: Access, Alternate, Business, Bypass, Connector, Extended, Extension, Loop, Private, Public, Scenic, Spur, Ramp, Underpass, Overpass. 2. Parsing rules allow some flexibility in deciding whether a Complete Street Name includes a Street Name Post Modifier. In each of the examples above, for instance, the entire name could be treated as the Street Name element. If the Complete Street Name is parsed into components, the Street Name Post Modifier provides a way to handle words that follow the Street Name and should be separated from it, or that are separated from the Street Name by a Street Name Post Directional or a Street Name Post Type. See Complete Street Name notes for a general discussion of Complete Street Name parsing principles.
XML Tag	<StreetNamePostModifier>
XML Model	<xsd:complexType name="StreetNameModifier_type"> <xsd:simpleContent> <xsd:extension base="xsd:string"> <xsd:attribute name="Separator" type="addr_type:Separator_type"></xsd:attribute> </xsd:extension> </xsd:simpleContent> </xsd:complexType>
XML Example	<CompleteStreetName> <StreetName>GRAND</StreetName> <StreetNamePostType>BOULEVARD</StreetNamePostType> **<StreetNamePostModifier>CUTOFF</StreetNamePostModifier>** </CompleteStreetName> <CompleteStreetName> <StreetName>CONCORD</StreetName> <StreetNamePostType>HIGHWAY</StreetNamePostType> **<StreetNamePostModifier>EXTENSION</StreetNamePostModifier>** </CompleteStreetName>
Quality Measures	Tabular Domain Measure Spatial Domain Measure
Quality Notes	1. Where a specific set of postmodifiers are specified for use, they may be maintained as a domain and tested with Tabular Domain Measure. 2. Where a schema may designate a particular area with a Street Name Post Modifier the entries may be tested with Spatial Domain Measure.

822 ### 2.8.2.8 Complex Element: Complete Street Name

Element Name	**Complete Street Name**
Other common names for this element	Street name, Road name, Full name (Census TIGER)
Definition	Official name of a street as assigned by a local governing authority, or an alternate (alias) name that is used and recognized.
Syntax	{ Street Name Pre Modifier } + { Street Name Pre Directional } + { Street Name Pre Type } + { Separator Element } + { Street Name *} + { Street Name Post Type } + { Street Name Post Directional } + { Street Name Post Modifier }
Definition Source	New
Data Type	characterString
Existing Standards for this Element	Refer to Component Elements
Domain of Values for this Element	Local domain of values for Complete Street Name. Refer to component elements for domains governing individual elements.
Source of Values	Locally determined
How Defined (e.g., locally, from standard, other)	Locally determined
Example	All of the following are complete street names: **Main Street** **North Main Street** **North Main Street Extended** **Avenue B** **Old Avenue B North** **Broadway** **Kentucky State Highway 67** **North Parkway** **Alameda de las Pulgas**
Notes/Comments	**1. Complete Street Name List.** Each jurisdiction should establish a domain of values for each street name element, and compose from that a lookup table of valid Complete Street Names, for use in validating addresses and diagnosing street name errors. Alternate and Official names are distinguished by the Official Status attribute. **2. Abbreviations.** All street name elements should be spelled out in full. If abbreviated versions of

street names are needed for a specialized purpose such as mailing or emergency dispatch, the variants can be created in views or export routines. USPS abbreviations for street types and directionals are recognized within the Postal Addressing Profile of this standard.

3. General Parsing Rules and Guides.
1. The Street Name element is required to compose a Complete Street Name. The other elements are optional and in some cases must inferred from the context and syntax.
2. Some Complete Street Names can be parsed in more than one way (see below for examples). In such cases, local authorities may prefer to parse the Complete Street Name so that the Street Name element can be used to create a sorted alphanumeric list of names.
3. It is permissible to parse the Complete Street Name in its entirety as a Street Name.

4. Parsing Ambiguous Complete Street Names:
Some Complete Street Names can be parsed in more than one way. For example:
-- **County Road 88** or County Road **88**
-- **East River** Avenue or East **River** Avenue
-- **The Croft** Lane or The **Croft** Lane
-- **Boulevard of the Allies** or Boulevard of the **Allies**

This standard accommodates any of the above choices. As a matter of guidance local authorities may prefer to parse the Complete Street Name so that the Street Name element can be used to create a sorted alphanumeric list of names. By this principle the first set of parsings would give the following sorted list:
-- **Boulevard of the Allies**
-- **County Road 88**
-- **East River** Avenue
-- **The Croft** Lane

The second set of parsings would give a different list:
-- **88,** County Road
-- **Allies,** Boulevard of the
-- **Croft** Lane, The
-- **River** Avenue, East

5. Special Case: Street Names Composed Entirely of Directional and Street Type Words
Examples: North Parkway; Avenue East; Court Place
Rule: Every Complete Street Name must include a Street Name element.
Discussion: In each Complete Street Name, at least one word must fill the Street Name element. "North Parkway", for example, could be handled four

	ways, one of which is invalid: * VALID: Street Name Pre Directional = “North”; Street Name = “Parkway” * VALID: Street Name = “North”; Street Name Post Type = “Parkway” * VALID: Street Name = “North Parkway” * **INVALID:** Street Name Pre Directional = “North”; Street Name = **null**; Street Name Post Type = “Parkway” **6. Special Case: Numbered Local Government, County, State, and U.S. Roads and Highways** **Examples:** Township Road 20; County Road 88; Kentucky State Highway 67; US Route 40 (see USPS Publication 28 Appendix F for additional examples) **Recommendation:** Use whatever parsing method is most convenient, but use one method consistently. **Discussion:** Within the structure of the standard, these cases could be handled in several ways. 1. Treat the entire name as a Street Name element: * Street Name = "Kentucky State Highway 67" 2. Parse the name into a Street Name Pre Type and a Street Name: * Street Name Pre Type = “Kentucky State Highway” * Street Name = “67” 3. Parse the name into a Street Name Pre Modifier and a Street Name: * Street Name Pre Modifier = “Kentucky State” * Street Name = “Highway 67” 4. Parse the name into a Street Name Pre Modifier, Street Name Pre Type, and Street Name: * Street Name Pre Modifier = “Kentucky State” * Street Name Pre Type = “Highway” * Street Name = “67” 7. The Separator Element may be used where a prepositional phrase such as "of the", "de", "de las", "d'" connects a Street Name Pre Type to a Street Name.
XML Tag	<CompleteStreetName>
XML Model	<xsd:complexType name="CompleteStreetName_type"> <xsd:sequence> <xsd:element name="StreetNamePreModifier" type="addr_type:StreetNameModifier_type" minOccurs="0" maxOccurs="1" /> <xsd:element name="StreetNamePreDirectional" type="addr_type:StreetNameDirectional_type" minOccurs="0" maxOccurs="1" /> <xsd:element name="StreetNamePreType" type="addr_type:StreetNameType_type" minOccurs="0" maxOccurs="1" /> <xsd:element name="StreetName" type="addr_type:StreetName_type" minOccurs="1" maxOccurs="1" />

	<xsd:element name="StreetNamePostType" type="addr_type:StreetNameType_type" minOccurs="0" maxOccurs="1" /> <xsd:element name="StreetNamePostDirectional" type="addr_type:StreetNameDirectional_type" minOccurs="0" maxOccurs="1" /> <xsd:element name="StreetNamePostModifier" type="addr_type:StreetNameModifier_type" minOccurs="0" maxOccurs="1" /> </xsd:sequence> <xsd:attribute name="AttachedElement" type="addr_type:AttachedElement_type" /> </xsd:complexType>
XML Example	<CompleteStreetName> <StreetNamePreDirectional>NORTH</StreetNamePreDirectional> <StreetName>MAIN</StreetName> <StreetNamePostType>STREET</StreetNamePostType> <StreetNamePostModifier>EXTENDED</StreetNamePostModifier> </CompleteStreetName> <CompleteStreetName> <StreetNamePreModifier>OLD</StreetNamePreModifier> <StreetNamePreType>AVENUE</StreetNamePreType> <StreetName>B</StreetName> <StreetNamePostDirectional>NORTH</StreetNamePostDirectional> </CompleteStreetName>
Quality Measures	Complete Street Name Tabular Domain Measure Duplicate Street Name Measure Pattern Sequence Measure
Quality Notes	Note that if tabular and/or domains are maintained for Complete Street Name elements at both levels, simple and complex, quality control checks should be run for simple element components before testing the complex element domain.

2.8.3 Subaddress Elements

2.8.3.1 Subaddress Type

Element Name	Subaddress Type
Other common	**Building:** Tower, Block, Terminal, Hangar, Pier

names for this element	**Multi-floor Part of a Building:** Wing, Tower **Floor:** Level, Story **Multi-unit Part of a Floor:** Corridor **Unit:** Apartment, Suite, Room, Unit, Office, Trailer, Space, Lot, Slip, Berth **Portion of a Unit:** Cubicle, Seat **PMB:** Private Mail Box **General:** Secondary Address Designator (USPS), Secondary Address Unit Designator (USPS); Secondary Unit Designator (USPS); Secondary Address Identifier (EPA); Generic Occupancy Type
Definition	The type of subaddress to which the associated Subaddress Identifier applies. (In the examples, Building, Wing, Floor, etc. are types to which the Identifier refers.) See Complete Subaddress for a definition of "subaddress."
Definition Source	New
Data Type	characterString
Existing Standards for this Element	None
Domain of Values for this Element	Can be created locally from existing values
Source of Values	Local
How Defined (e.g., locally, from standard, other)	Locally
Example	**Building** 4 **Wing** 7 **Floor** 6 **Corridor** Zero **Apartment** 2D **PMB** 596
Notes/Comments	1. The Subaddress Type is used with Subaddress Identifier to designate one of several structures, floors, corridors, units, etc. at a given site. It fits within the general USPS definition of a "secondary address designator" and EPA definition of a "secondary address identifier" 2. USPS Publication 28 Appendix C2 and Section 293 provide a list of common Subaddress Types with standard abbreviations. The FGDC Standard requires storing Subaddress Types fully spelled out, to avoid confusion. If stored unabbreviated, they can be exported as standard abbreviations as needed for mailing and other purposes. USPS Abbreviations are recognized within the Postal Addressing Profile of this standard. 3. PMB (Private mail box) is a special Subaddress Type. See Subaddress Element notes.

XML Tag	<SubaddressType>
XML Model	<xsd:simpleType name="SubaddressType_type"> <xsd:restriction base="xsd:string" /> </xsd:simpleType>
XML Example	<CompleteSubaddress> **<SubaddressElement Element Sequence Number="1" Subaddress Component? Order="1" >** **<SubaddressType>Building</SubaddressType>** <SubaddressIdentifier>A</SubaddressIdentifier> </SubaddressElement> <SubaddressElement Element Sequence Number="2" Subaddress Component? Order="2" > **<SubaddressType>Room</SubaddressType>** <SubaddressIdentifier>Empire</SubaddressIdentifier> </SubaddressElement> </CompleteSubaddress>
Quality Measures	Tabular Domain Measure
Quality Notes	Subaddress types may follow defined schemes for particular buildings or complexes. While these associations are beyond the scope of the standard they should be considered in planning a quality program for local addresses. Note that Subaddress Type entries must be associated with an address to test any spatial associations with particular buildings or complexes, and are therefore tested at the classification level.

2.8.3.2 Subaddress Identifier

Element Name	**Subaddress Identifier**
Other common names for this element	Building ID, Floor ID, Apartment Number, Suite Number; Secondary unit indicator (USPS), secondary number (USPS), secondary range (USPS)
Definition	The letters, numbers, words or combination thereof used to distinguish different subaddresses of the same type when several occur within the same feature. See Complete Subaddress for a definition of "subaddress."
Definition Source	New
Data Type	characterString
Existing Standards for this Element	None

Domain of Values for this Element	Can be defined locally from existing values.
Source of Values	Local
How Defined (e.g., locally, from standard, other)	Locally
Example	Building **4** Wing **7** Floor **6** Corridor **Zero** Apartment **2D** PMB **596** **Mezzanine** **Penthouse** **Basement**
Notes/Comments	1. The Subaddress Identifier, in combination with the Subaddress Type, is used to designate one of several subaddresses within or between structures at a given site. 2. See Subaddress Element and Complete Subaddress for additional notes.
XML Tag	<SubaddressIdentifier>
XML Model	<xsd:simpleType name="SubaddressIdentifier_type"> <xsd:restriction base="xsd:string" /> </xsd:simpleType>
XML Example	<CompleteSubaddress> <SubaddressElement Element Sequence Number="1" Subaddress Component Order="1" > <SubaddressType>Building</SubaddressType> **<SubaddressIdentifier>A</SubaddressIdentifier>** </SubaddressElement> <SubaddressElement Element Sequence Number="1" Subaddress Component Order="2" > <SubaddressType>Room</SubaddressType> **<SubaddressIdentifier>Empire</SubaddressIdentifier>** </SubaddressElement> </CompleteSubaddress>
Quality Measures	Range Domain Measure Tabular Domain Measure
Quality Notes	Subaddress identifiers may follow defined schemes for particular buildings or complexes. While these associations are beyond the scope of the standard they should be considered in planning a quality program for local addresses. Note that Subaddress Identifier entries must be associated with an address to

	test any spatial associations with particular buildings or complexes, and are therefore tested at the classification level

2.8.3.3 Complex Element: Subaddress Element

Element Name	**Subaddress Element**
Other common names for this element	Secondary address identifier (USPS, EPA)
Definition	A single combination of Subaddress Type and Subaddress Identifier (or, in some cases, a Subaddress Identifier alone), which, alone or in combination with other Subaddress Elements, distinguishes one subaddress within or between structures from another when several occur within the same feature. See Complete Subaddress for a definition of "subaddress."
Syntax	{ Subaddress Type } + { Subaddress Identifier* }
Definition Source	New
Data Type	characterString
Existing Standards for this Element	None
Domain of Values for this Element	No
Source of Values	N/A
How Defined (e.g., locally, from standard, other)	N/A
Attributes Associated with this Element	Subaddress Component Order
Example	**Building 4** **Wing 7** **North Tower** **Floor 6** **Sixth Floor** **Corridor Zero** **Apartment 2D** **PMB 596** **Empire Room**

	Penthouse
Notes/Comments	1. An Subaddress Element, alone or in combination with other Subaddress Elements, forms a Complete Subaddress. 2. In English, if the Subaddress Identifier is a name or an ordinal number, the Subaddress Identifier usually but not always precedes the Subaddress Type ("North Tower," "Sixth Floor," "Empire Room,"). If the Subaddress Identifier is a cardinal number, letter designator, or alphanumeric, it typically follows the Subaddress Type ("Building 4," "Apartment 2D", "Hanger A"). Common usage is loose, and there are numerous exceptions to both rules, and patterns differ in other languages. The Subaddress Component Order can be used to indicate the order in which the Subaddress Type and Subaddress Identifier should be written. 3. Some Subaddress Elements use only one word ("Mezzanine"). In such cases, by definition the word is considered an Subaddress Identifier, and the Subaddress Type is null. Other examples (all from USPS Publication 28 Appendix C2) are: Penthouse, Lobby, Basement, Front, Rear, Upper, Lower, Side. 4. **The Special case of PMB (Private Mail Box) Subaddresses.** Normally a PMB (Private Mail Box), like a mailstop code and other internal mail distribution codes, pertains to the recipient and is not part of the address. However, USPS Publication 28 Section 284 states, "Exception: When the CMRA [commercial mail receiving agency] mailing address contains a secondary address element (e.g. rural route box number, suite, # or other term), the CMRA customer must use Private Mail Box (PMB) when utilizing a three line address format. Examples: --RR 1 Box 12 **PMB 596** --10 Main Street Suite 11 **PMB 234** " The abbreviation "PMB" is recognized within this standard, along with a few others defined by the USPS for designating postal delivery boxes. PMB is the only Subaddress Type that can appear in the USPS Postal Delivery Box or USPS Postal Delivery Route address classes.
XML Tag	<SubaddressElement>
XML Model	<xsd:complexType name="SubaddressElement_type"> <xsd:sequence> <xsd:element name="SubaddressType" type="addr_type:SubaddressType_type" maxOccurs="1" minOccurs="0" /> <xsd:element name="SubaddressIdentifier" type="addr_type:SubaddressIdentifier_type" maxOccurs="1" minOccurs="1" /> </xsd:sequence> <xsd:attribute name="ElementSequenceNumber" type="addr_type:ElementSequenceNumber_type" /> <xsd:attribute name="SubaddressComponentOrder" type="addr_type:SubaddressComponentOrder_type" />

	<xsd:attribute name="Separator" type="addr_type:Separator_type" /> </xsd:complexType>
XML Example	<CompleteSubaddress> **<SubaddressElement** Element Sequence Number**="1"** Subaddress Component Order**="1" >** **<SubaddressType>Building</SubaddressType>** **<SubaddressIdentifier>A</SubaddressIdentifier>** **</SubaddressElement>** <SubaddressElement Element Sequence Number="2" Subaddress Component Order="1" > <SubaddressType>Floor</SubaddressType> <SubaddressIdentifier>7</SubaddressIdentifier> </SubaddressElement> </CompleteSubaddress>
Quality Measures	Pattern Sequence Measure
Quality Notes	Subaddress elements may follow defined schemes for particular buildings or complexes. While these associations are beyond the scope of the standard they should be considered in planning a quality program for local addresses. Note that Subaddress Element entries must be associated with an address to test any spatial associations with particular buildings or complexes, and are therefore tested at the classification level

2.8.3.4 Complex Element: Complete Subaddress

Element Name	**Complete Subaddress**
Other common names for this element	See Subaddress Element
Definition	One or more Subaddress Elements that identify a subaddress within an addressed feature. A subaddress is a separate, identifiable portion of a feature, the whole of which is identified by a: --- Complete Address Number and Complete Street Name (in the case of a Numbered Thoroughfare Address) --- Two Complete Address Numbers, separated by a hyphen, and followed by a Complete Street Name (in the case of a Two Number Address Range) --- Complete Street Name (in the case of an Unnumbered Thoroughfare Address) --- Complete Landmark Name (in the case of a Landmark Address) --- Complete Address Number and Complete Landmark Name or Complete Place Name (in the case of a Community Address)

	--- USPS Box or USPS Address (in the case of a USPS Postal Delivery Box or USPS Postal Delivery Route address; for these classes, PMB (private mail box) is the only Subaddress Type permitted.)
Syntax	A series of one or more Subaddress Elements. If more than one are listed, the Element Sequence Number can be used to show the order in which they should be listed.
Definition Source	New
Data Type	characterString
Existing Standards for this Element	None
Domain of Values for this Element	None
Source of Values	N/A
How Defined (e.g., locally, from standard, other)	N/A
Attributes Associated with this Element	Element Sequence Number
Example	1. 123 Main Street, **Apartment 101** 2. 1000 Aviation Road, **Building 4, Wing 7, Floor 6, Corridor Zero, Office 2B** 3. Metro Airport, **Terminal A, Gate C27** 4. Average Suburban Office Park, **Building 12, Mezzanine, Suite 200** 5. 800 West Mountain Road, **Building 6, Suite 450** 7. 740 Park Avenue, **Apartment 15/16B** 8. 1324-26 Calle Amapolas, **Apartamento 103** 9. Five-Star Hotel, **East Tower, Penthouse** 9. U.S. Dept. of Agriculture Building, **Wing 7, Room 324** 10. General Hospital, **Cardiac Wing, Room 224** 11. U.S. Department of Commerce Building, **Room 6056 (Floor 6, Corridor Zero, Room 56)** 12. Pentagon, **Room 3D126 (Third floor, D ring, First corridor, Room 26)** 13. RR 1 Box 12 **PMB 596** 14. 10 Main Street **Suite 11 PMB 234**
Notes/Comments	1. Complete Subaddresses and their component elements pertain a wide variety of residential, and commercial buildings, from single basement apartments to multi-stucture office parks, as well as countless specialized structures such as airports, piers, warehouses, manufacturing plants, and stadiums. Complete Subaddresses are typically designated by the property owner, and addressing authorities usually have no responsibility for compiling or verifiying them. However, this is changing as address

	verification becomes more important for government purposes such as security, emergency response, and verification of eligibility for voting, school attendance, and public services. 2. Usually Complete Subaddresses follow a pattern of Building-Floor-Room (or Doorway), but due to the wide variety of cases no general rule can be given. In composing the Complete Subaddress, the Subaddress Elements should be ordered from largest to smallest, or in the order one would encounter them in navigating from outside the site to the designated subaddress. If desired, use the Element Sequence Number to indicate the sequence in which the Subaddress Elements should be ordered.
XML Tag	<CompleteSubaddress>
XML Model	<xsd:complexType name="CompleteSubaddress_type"> <xsd:sequence> <xsd:element name="SubaddressElement" type="addr_type:SubaddressElement_type" minOccurs="1" maxOccurs="unbounded" /> </xsd:sequence> </xsd:complexType>
XML Example	<CompleteSubaddress> <SubaddressElement Element Sequence Number="1" Subaddress Component Order="1" > <SubaddressType>Building</SubaddressType> <SubaddressIdentifier>A</SubaddressIdentifier> </SubaddressElement> <SubaddressElement Element Sequence Number="2" Subaddress Component Order="1" > <SubaddressType>Floor</SubaddressType> <SubaddressIdentifier>7</SubaddressIdentifier> </SubaddressElement> </CompleteSubaddress>
Quality Measures	Repeated Element Uniqueness Measure Complex Element Sequence Number Measure
Quality Notes	This test for the Complete Subaddress assumes that quality tests have been run for supporting elements: Subaddress Type, Subaddress Identifier and Subaddress Element.

2.8.4 Landmark Name Elements

2.8.4.1 Landmark Name

Element Name	Landmark Name
Other common names for this element	Point of interest
Definition	The name of a relatively permanent feature of the manmade landscape that has recognizable identity within a particular cultural context.
Definition Source	Adapted from U.S. Board on Geographic Names, "Principles, Policies, Procedures," (Online Edition (revised), 2003, as posted May 17, 2006 at http://geonames.usgs.gov/docs/pro_pol_pro.pdf), p. 48, definition of "geographic name".
Data Type	characterString
Existing Standards for this Element	None, but see GNIS Feature ID
Domain of Values for this Element	Can be created locally from existing values.
Source of Values	Local
How Defined (e.g., locally, from standard, other)	Locally
Attributes Associated with this Element	Element Sequence Number, GNIS Feature ID
Examples	**U.S. Capitol Building** **Empire State Building** **Winona Park Elementary School** **Valley Mall** **Yosemite National Park**
Notes/Comments	1. A Landmark Name specifies a location by naming it. It does not relate the named feature to any thoroughfare system or coordinate reference system and therefore provides no information about where to find the feature. Many addresses include Landmark Names without any thoroughfare names, and as such Landmark Names form the basis for two address classes: Landmark Address and Community Address. 2. Landmark names are given to both natural and manmade features. In general, natural landmark names are not used in addresses and are therefore excluded from the scope of this standard. Thus "Yosemite National Park"

	could be part of an address, and therefore is within the scope of the standard, whereas "Yosemite Falls" and "Yosemite Valley" (naming the natural features) would not. 3. The difference between Landmark Name and a Place Name is not always clear and distinct. As a general principle, a landmark is under a single use or ownership or control, while places are not. Thus a landmark, even if it covers an extensive area, might be considered to be a single "master address" (often containing multiple subordinate addresses), while a place generally includes numerous separate addresses. These general principles apply to most cases and are useful as general distinctions, but exceptions and marginal cases are easily found. 4. Local address authorities may wish to compile a list of locally-recognized Landmark Names used as addresses for their convenience. Whether to do so, and if so what names to include, are implementation matters to be decided locally. 5. Most named landmarks that are used as addresses are also designated by one or more thoroughfare addresses. These should be cross-referenced to each other as Related Address I Ds, using the Address Relation Type attribute to record the relationship between them. 6. Landmark Name, as used in this standard, does not imply any officially-designated historic landmark status, nor is it restricted to features having such status. 7. The U.S. Board on Geographic Names has compiled and standardized names for many landmarks in the Geographic Names Information System (GNIS), each identified by a unique GNIS Feature ID. Local authorities are encouraged to review the GNIS Feature ID for more information on the use of the GNIS ID with Landmark Names. 8. The U.S. Board on Geographic Names has defined 65 classes of features for use in classifying features listed in GNIS. These classes, while neither exhaustive nor necessarily definitive for addressing purposes, may provide useful guidance in distinguishing Place Names, manmade Landmark Names, and natural landmark names. ---Manmade landmark classes (the names of these features are often used in addresses and therefore generally within the scope of this standard): airport, bridge, building, canal, cemetery, church, crossing, dam, harbor, hospital, levee, locale, military, mine, oilfield, park, post office, reserve, reservoir, school, tower, trail, tunnel, well. ---PlaceName classes (the names of these features are generally Place Names within this standard): Census, civil, populated place. ---Natural landmark classes (the names of these features are generally outside the scope of this standard): arch, area, arroyo, bar, basin, bay, beach, bench, bend, cape, cave, channel, cliff, crater, falls, flat, forest, gap, glacier, gut, island, isthmus, lake, lava, pillar, plain, range, rapids, ridge, sea, slope, spring, stream, summit, swamp, valley, woods. The complete feature class definitions can be found from the GNIS Domestic

	Names search page. See Appendix A (U.S. Geological Survey) for a complete citation.
XML Tag	<LandmarkName>
XML Model	<xsd:complexType name="LandmarkName_type"> <xsd:simpleContent> <xsd:extension base="xsd:string"> <xsd:attribute name="ElementSequenceNumber" type="addr_type:ElementSequenceNumber_type" /> </xsd:extension> </xsd:simpleContent> </xsd:complexType>
XML Example	<CompleteLandmark> **<LandmarkName>YOSEMITE NATIONAL PARK</LandmarkName>** </CompleteLandmark>
Quality Measures	Uniqueness Measure Tabular Domain Measure Spatial Domain Measure
Quality Notes	Some landmarks will be nested within a larger one, the latter constituting a spatial domain. Similarly, a tabular domain may be associated with an outer landmark.

2.8.4.2 Complex Element: Complete Landmark Name

Element Name	**Complete Landmark Name**
Other common names for this element	
Definition	One or more Landmark Names which identify a relatively permanent feature of the manmade landscape that has recognizable identity within a particular cultural context.
Syntax	A series of one or more Landmark Names. If more than one are listed, the Element Sequence Number can be used to show the order in which they should be listed.
Definition Source	Adapted from U.S. Board on Geographic Names, "Principles, Policies, Procedures," (Online Edition (revised), 2003, as posted May 17, 2006 at http://geonames.usgs.gov/docs/pro_pol_pro.pdf), p. 48, definition of "geographic name".

Data Type	characterString
Existing Standards for this Element	None, but see GNIS Feature ID
Domain of Values for this Element	Can be created locally from existing values
Source of Values	Local
How Defined (e.g., locally, from standard, other)	Locally
Examples	**University of Washington**, Seattle, WA **Suzallo Library, University of Washington**, Seattle, WA **Statue of Liberty**, New York, NY **Statue of Liberty, Liberty Island**, New York, NY **Yosemite National Park**, CA **Camp Curry, Yosemite National Park**, CA
Notes/Comments	1. Landmark names often refer to extensive areas, which may contain smaller named landmarks. In these cases the landmark name may function as a single "master address" containing multiple subordinate addresses. The Complete Landmark Name provides for the inclusion of multiple Landmark Names in an address. 2. Where multiple Landmark Names are given, they are typically ordered from smallest to largest. The Element Sequence Number can be used to indicate the sequence in which the Landmark Names should be ordered. 4. The U.S. Board on Geographic Names has compiled and standardized names for many landmarks in the Geographic Names Information System (GNIS). Local authorities are encouraged to review the GNIS Feature ID for more information on the use of the GNIS ID and Landmark Names. Where a complete landmark name consists of more than one landmark name, the GNIS Code for the smallest unit of the complete landmark name should be used to provide the most specific reference.
XML Tag	<CompleteLandmarkName>
XML Model	<xsd:complexType name="CompleteLandmarkName_type"> <xsd:sequence> <xsd:element name="LandmarkName" type="addr_type:LandmarkName_type" minOccurs="1" maxOccurs="unbounded" /> </xsd:sequence> <xsd:attribute name="Separator" type="addr_type:Separator_type" /> </xsd:complexType>
XML Example	<CompleteLandmark Separator=","> **<LandmarkName ElementSequenceNumber="1">CAMP**

	CURRY</LandmarkName> **<LandmarkName ElementSequenceNumber="2">YOSEMITE NATIONAL PARK</LandmarkName>** </CompleteLandmark>
Quality Measures	Repeated Element Uniqueness Measure Complex Element Sequence Number Measure
Quality Notes	

2.8.5 Place, State, and Country Name Elements

2.8.5.1 Place Name

Element Name	**Place Name**
Other common names for this element	**Unincorporated community or neighborhood:** Urbanization, urbanizacion place name, or barrio (Puerto Rico); borough (in, for example, New York City), community, neighborhood, subdivision, Census designated place, populated place (GNIS), locale (GNIS) **Incorporated local government:** Municipality, city, borough, town, village, township, actual city, location city, situs city, municipal place name, minor civil division, corporation, consolidated government, metropolitan government, unified government, populated place (GNIS), locale (GNIS) **USPS Post Office Name:** Post office, mailing city, city (as in "City, State, ZIP"), city name; APO, FPO, DPO (for overseas US military and diplomatic mail delivery) **County:** Parish (Louisiana); Census Area, City and Borough, and Unorganized Borough (Alaska), Municipality (Alaska and the Commonwealth of the Northern Mariana Islands), Municipio (Puerto Rico), City (Maryland, Missouri, Nevada, and Virginia), District (American Samoa), Island (American Samoa and U.S. Virgin Islands) **Region:** Metropolitan area, metropolitan statistical area (Census), consolidated metropolitan statistical area (Census), primary metropolitan statistical area (Census)
Definition	The name of an area, sector, or development (such as a neighborhood or subdivision in a city, or a rural settlement in unincorporated area); incorporated municipality or other general-purpose local governmental unit; county or county-equivalent; or region within which the address is

	physically located; or the name given by the U.S. Postal Service to the post office from which mail is delivered to the address.
Definition Source	New; partly adapted from: 1. FGDC's "Framework Data Content Standard Part 5: Governmental unit and other geographic area boundaries"; and, 2. USPS Publication 28, Section 292, "Urbanization".
Data Type	characterString
Existing Standards for this Element	No single controlling authority, but the Geographic Names Information System (GNIS) attempts to include and standardize the names of all populated places and incorporated local governments (see GNIS Feature ID). For USPS Post Office names, the controlling authority is the USPS "City State File" as referenced in Section 221 of USPS Publication 28
Domain of Values for this Element	None (but see existing standards above). Can be created locally from existing values.
Source of Values	Locally determined (but see existing standards above)
How Defined (e.g., locally, from standard, other)	Locally.
Attributes Associated with this Element	Place Name Type, Element Sequence Number, GNIS Feature ID
Examples	**Ajo**, AZ (unincorporated community in Pima County, AZ) **Urbanizacion Los Pinos** (Puerto Rico urbanization) **Jardine Fagota** (Puerto Rico urbanization) **Portola Valley**, CA (incorporated town) **Birmingham**, AL (city) **Salt Lake City**, UT (city) **Queens** (New York City borough) **Orleans Parish**, LA (county) **APO** AE (overseas military postal delivery) **FPO** AP (overseas military postal delivery) **DPO** AE (overseas US State Department postal delivery)
Notes/Comments	1. "Place name" can mean different things to different people in different contexts. It may name a community, an incorporated local government, a post office, a county, or a region. For many thoroughfare and landmark addresses, a different place name may be used by an emergency dispatcher directing an ambulance, a local government official assessing local taxes or eligibility for services, a postal clerk, or a business providing contact information on its website. 2. This standard provides the Place Name Type attribute to allow the use of different place names with the same address for different purposes. Five

	types are defined: unincorporated community or neighborhood, incorporated local governmnent, U.S. Post Office name, county, and region. Other types may be added. Additional explanation is given in the notes below and under Place Name Type. 3. The U.S. Board of Geographic Names has assigned GNIS Codes to all place names that have been registered and accepted by the Board. This standard provides the GNIS Feature ID attribute to accommodate those codes. For more information on GNIS, see GNIS Feature ID or http://geonames.usgs.gov/domestic/index.html . **Notes on Community Names:** 1. A community name refers to an area, sector, or development, such as a neighborhood or subdivision in a city, or a rural settlement in unincorporated area, that is not an incorporated general-purpose local government or county. The name may arise from official recognition or from popular usage. 2. Numerous different terms are used to denote different kinds of communities and community names, but the distinctions are not particularly significant in constructing addresses. An extensive list of terms and definitions can be found in "Framework Data Content Standard Part 5: Governmental unit and other geographic area boundaries," Tables 11 and 15. 3. "Urbanizacion" (community) names can be of particular importance in Puerto Rican addresses. Street names and address ranges are repeated in many cities, especially where a city has annexed older towns; these repeated addresses are distinguished by their urbanizacion name. (Certain other words can be used in place of "urbanizacion": extenciones, mansiones, reparto, villa, parque, jardine, altura, alturas, colinas, estancias, extension, quintas, sector, terraza, villa, villas.) For more information on Puerto Rican addressing conventions, see USPS Publication 28 Section 29, and USPS "Addressing Standards for Puerto Rico and the Virgin Islands". **Notes on Municipal and County Place Names:** 1. County and municipal names indicate the county and the general-purpose local government area (if any) in which the address is physically located. Local government types and terminologies vary substantially from state to state, but the distinctions are not particularly significant in constructing addresses. An extensive list of terms and definitions can be found in "Framework Data Content Standard Part 5: Governmental unit and other geographic area boundaries," Table 13. 2. Exact municipal and county names are required by public administrators for correct assessing local taxes, assignment of voting precinct, school enrollment, and provision of local government services. 3. Addresses in unincorporated portions of counties have no municipal place name by definition.

	4. Many governments have a legal name and a popular name ("Saint Paul" vs. "City of Saint Paul"). For addressing, the popular name is generally preferable if it is unique within the county and state. 5. "New York City" comprises five administrative "Boroughs" (Bronx, Brooklyn, Manhattan, Queens, and Staten Island). The Boroughs are legally distinct from the five Counties that are also subdivisions of New York City (Bronx, Kings, New York, Queens, and Richmond) even though the Boroughs and Counties have identical boundaries and two even share the same name. **Notes on USPS Place Names:** 1. The USPS place name is assigned to the post office from which the USPS delivers mail to the address. 2. USPS place names are preferred for postal operations. However, they are often not the best-suited place names for non-postal purposes such as navigation, public service delivery, and emergency response. 3. For postal purposes, the USPS strongly discourages the use of multiple place names in an address. For example, the USPS on-line ZIP finder will find a ZIP code for an address in ""Wailuku, HI," but not for "Wailuku, Maui, HI." 4. For overseas US military postal addresses, "APO" (Army Post Office) or "FPO" (Fleet Post Office) is used as the Place Name (see USPS Publication 28, Section 225.1 and 238.1). "DPO" (Diplomatic Post Office) is used as the Place Name for some overseas US State Department postal addresses (see USPS Pub 28 Sec. 239). **Notes on Regional Place Names:** 1. A region name refers to the region where the address is physically located. Typically this is name of the central city within the region. For precise, systematic terms, U.S. Census Bureau terms and definitions may be applied, but popular usage is often imprecise and to some extent subjective. Businesses and residents near a regional center often use the central-city name in their address, even if the address is located some distance outside the limits of the city itself.
XML Tag	<PlaceName>
XML Model	<xsd:complexType name="PlaceName_type"> <xsd:simpleContent> <xsd:extension base="xsd:string"> <xsd:attribute name="PlaceNameType" type="addr_type:PlaceNameType_type" /> <xsd:attribute name="ElementSequenceNumber" type="addr_type:ElementSequenceNumber_type" /> <xsd:attribute name="GNISFeatureID" type="addr_type:GNISFeatureID_type" />

	</xsd:extension> </xsd:simpleContent> </xsd:complexType>
XML Example	<PlaceName>ORLEANS PARISH</PlaceName>
Quality Measures	Uniqueness Measure Tabular Domain Measure Spatial Domain Measure
Quality Notes	Some place names will be nested within a larger one, the latter constituting a spatial domain. Similarly, a tabular domain may be associated with an outer place name.

2.8.5.2 Complex Element: Complete Place Name

Element Name	**Complete Place Name**
Other common names for this element	See Place Name
Definition	One or more Place Names which identify an area, sector, or development (such as a neighborhood or subdivision in a city, or a rural settlement in unincorporated area); incorporated municipality or other general-purpose local governmental unit; county; or region within which the address is physically located; or the name given by the U.S. Postal Service to the post office from which mail is delivered to the address.
Syntax	A series of one or more Place Names. If more than one is listed, the Place Name Type can be used to specify the type for each Place Name (e.g., community, municipal, postal, county, region) and the Element Sequence Number can be used to show the order in which they should be listed.
Definition Source	See Place Name
Data Type	characterString
Existing Standards for this Element	No single controlling authority, but the Geographic Names Information System (GNIS) attempts to include and standardize the names of all populated places and incorporated local governments (see GNIS Feature ID). For USPS Post Office names, the controlling authority is the USPS "City State File" as referenced in Section 221 of USPS Publication 28
Domain of Values for this Element	None (but see existing standards above)
Source of Values	Local (but see existing standards above)

How Defined (e.g., locally, from standard, other)	Locally.
Examples	**Ajo, Pima County**, AZ (unincorporated community in Pima County, AZ) **Jardines Los Almendros, Municipio Maunabo,** PR (Puerto Rico urbanization) **Portola Valley**, CA (incorporated town) **Birmingham**, AL (city) **Salt Lake City**, UT (city) **Queens, New York,** NY (New York City borough) **Orleans Parish**, LA (county) **FPO** AA (overseas military postal delivery) **New Hope Community, Shelby County**, AL (unincorporated community Shelby County, AL) **Capitol Hill, Washington**, DC (neighborhood in Washington, DC) **Wailuku, Maui**, HI **Edgewater Park, Bronx, New York**, NY (neighborhood in New York City)
Notes/Comments	1. "Place name" can mean different things to different people in different contexts. It may name a community, an incorporated local government, a post office, a county, or a region. For many thoroughfare and landmark addresses, a different place name may be used by an emergency dispatcher directing an ambulance, a local government official assessing local taxes or eligibility for services, a postal clerk, or a business providing contact information on its website. 2. For some purposes an address may require more than one place name (e.g., "Wailuku, Maui", or "New Hope, Shelby County"). This is discouraged in postal addresses, but it may necessary in other contexts, (e.g., to provide both the municipality and county for an address). The Complete Place Name provides for inclusion of multiple Place Names in the address. 3. Where multiple Place Names are given, they are typically ordered from smallest to largest. The Element Sequence Number can be used to indicate the sequence in which the Place Names should be ordered. 4. This standard provides the Place Name Type attribute to allow the use of different place names with the same address for different purposes. Five types are defined: community, municipal, postal, county, and regional. Others may be added. Additional explanation is given under Place Name and Place Name Type. 5. The difference between a place and a landmark is not always clear and distinct. As a general principle, a landmark is under a single use or ownership or control, while places are not. Thus a place generally includes numerous separate addresses, while a landmark, even if it covers an extensive area, might be considered to be a single "master address" (often

	containing multiple subordinate addresses). These general principles apply to most cases and are useful as general distinctions, but exceptions and marginal cases are easily found. 6. The U.S. Board of Geographic Names has assigned GNIS Feature ID's to all place names that have been registered and accepted by the Board. Within the address standard, GNIS Feature ID's may be associated with Place Names to facilitate standardization and unambiguous communication. See GNIS Feature ID for more information.
XML Tag	<CompletePlaceName>
XML Model	<xsd:complexType name="CompletePlaceName_type"> <xsd:sequence> <xsd:element name="PlaceName" type="addr_type:PlaceName_type" minOccurs="1" maxOccurs="unbounded" /> </xsd:sequence> <xsd:attribute name="Separator" type="addr_type:Separator_type" /> </xsd:complexType>
XML Example	<CompletePlaceName> <PlaceName Place Name Type="USPSPlaceName"> Ajo </PlaceName> </CompletePlaceName>
	<CompletePlaceName> <PlaceName Place Name Type="County" > Shelby </PlaceName> </CompletePlaceName>
	<CompletePlaceName> <PlaceName Place Name Type="USPS" > Washington </PlaceName> </CompletePlaceName>
	<CompletePlaceName> <PlaceName Place Name Type="Community" > Urbanizacion Los Olmos </PlaceName> </CompletePlaceName>
	<CompletePlaceName> <PlaceName Place Name Type="Community">Queens</PlaceName> <PlaceName Place Name Type="Municipal">New York</PlaceName> </CompletePlaceName>
Quality Measures	Repeated Element Uniqueness Measure Complex Element Sequence Number Measure

Quality Notes	

2.8.5.3 State Name

Element Name	**State Name**
Other common names for this element	State; Commonwealth (PA, MA, KY, VA, PR, MP); Territory (AS, GU, MP, PR, VI); District (DC); Minor Outlying Islands (UM); overseas military or diplomatic "state" (AA, AE, AP)
Definition	The names of the US states and state equivalents: the fifty US states, the District of Columbia, and all U.S. territories and outlying possessions. A state (or equivalent) is "a primary governmental division of the United States." The names may be spelled out in full or represented by their two-letter USPS or ANSI abbreviation.
Definition Source	Names and abbreviations: ANSI INCITS 38:200x, and USPS Publication 28 Appendix B Definition of 'state": Framework Data Content Standard Part 5: Governmental Unit and Other Geographic Area Boundaries," (Table 13).
Data Type	characterString
Existing Standards for this Element	ANSI INCITS 38:200x, and USPS Publication 28 Appendix B
Domain of Values for this Element	Yes
Source of Values	ANSI INCITS 38:200x, and USPS Publication 28 Appendix B
How Defined (e.g., locally, from standard, other)	ANSI INCITS 38:200x, and USPS Publication 28 Appendix B
Example	Chicago, **Illinois** Chicago **IL** Dover, **Delaware** Dover **DE** Hagatna, **Guam** Hagatna **GU** APO **AE** Wake Island **UM**
Notes/Comments	1. The State Name element follows the ANSI INCITS 38:200x standard (formerly the FIPS 5-2 standard) and USPS Publication 28 by including within the definition of State Name the fifty US states, the District of

	Columbia (DC), and US territories and possessions (AS, GU, MP, PR, and VI). In addition, USPS Publication 28 recognizes three overseas military and diplomatic State Name equivalents (AA, AE, and AP), which the ANSI standard does not; and the ANSI standard recognizes "UM" for US minor outlying islands, which USPS Publication 28 does not. 2. Within this standard State Names may be spelled out in full or they may be represented by their standard two-letter ANSI INCITS 38:200x or USPS abbreviations. 3. For overseas military and diplomatic postal addresses, "AE" or "AP" or "AA" is used as the State Name. "AE" is used for armed forces and certain diplomatic posts in Europe, the Middle East, Africa, and Canada; "AP" for the Pacific; and "AA" for the Americas excluding Canada (see USPS Publication 28, Section 225.1 and Appendix B). 4. The ANSI INCITS 38:200x standard abbreviations include the abbreviation UM for U.S. Minor Outlying Islands. These are nine small, remote islands or island groups that do not receive direct mail delivery: Midway Islands, Wake Island, Johnson Atoll, Kingman Reef, Palmyra Atoll, Jarvis Island, Howland Island, Baker Island, and Navassa Island. 5. In rare cases, the postal state and the physical location state of the address are not the same. This occurs in some communities on the borders of two states. In these cases, the physical address should be treated as the primary or official address, including the physical state name, while the postal address with its state name should be listed as an alias.
XML Tag	<StateName>
XML Model	<xsd:simpleType name="StateName_type"> <xsd:restriction base="xsd:token"> <!-- "US State and The District of Columbia" Abbreviations --> <xsd:pattern value='.*' /> </xsd:restriction> </xsd:simpleType>
XML Example	<StateName>VA</StateName> <StateName>VIRGINIA</StateName>
Quality Measures	Tabular Domain Measure
Quality Notes	

2.8.5.4 ZIP Code

Element Name	**ZIP Code**
Other common names for this element	ZIP5, Zone Improvement Plan
Definition	A system of 5-digit codes that identifies the individual Post Office or metropolitan area delivery station associated with an address.
Definition Source	USPS, "Quick Service Guide 800: Glossary of Postal Terms and Abbreviations in the DMM."
Data Type	characterString
Existing Standards for this Element	Yes
Domain of Values for this Element	Yes
Source of Values	USPS
How Defined (e.g., locally, from standard, other)	USPS is the sole source of this information.
Example	Birmingham, AL **35305** Webster Groves, MO **63119**
Notes/Comments	Strictly speaking a ZIPCode is not an area but a set of USPS delivery points served from the same post office. Delivery points with the same ZIP Code can encompass a a single building that has a very high mail volume; a portion of a city; all or parts of several municipalities; or even portions of more than more county (and, in a few cases, more than one state).
XML Tag	<ZIPCode>
XML Model	<xsd:simpleType name="ZIPCode_type"> <xsd:restriction base="xsd:string"> <xsd:pattern value='[0-9]{5}' /> </xsd:restriction> </xsd:simpleType>
XML Example	<ZIPCode>35305</Zipcode>
Quality Measures	Tabular Domain Measure Spatial Domain Measure
Quality Notes	

2.8.5.5 ZIP Plus 4

Element Name	ZIP Plus 4
Other common names for this element	ZIP+4
Definition	A 4-digit extension of the 5-digit ZIP Code (preceded by a hyphen) that, in conjunction with the ZIP Code, identifies a specific range of USPS delivery addresses.
Definition Source	Adapted from USPS, "Quick Service Guide 800: Glossary of Postal Terms and Abbreviations in the DMM."
Data Type	characterString
Existing Standards for this Element	Yes
Domain of Values for this Element	Yes
Source of Values	USPS is the sole source of this information.
How Defined (e.g., locally, from standard, other)	From USPS
Example	Birmingham, Alabama 35242 **-3426** Webster Groves, Missouri 63119 **-3212**
Notes/Comments	1. Strictly speaking, the ZIP Plus 4 consists of "the 5-digit ZIP Code and four additional digits that identify a specific range of USPS delivery addresses" (Quoted from USPS, "Quick Service Guide 800: Glossary of Postal Terms and Abbreviations in the DMM). However this standard separates the two components to facilitate data processing. 2. The ZIP Code and the ZIP Plus 4 are formatted with a hyphen between the two elements (see USPS Publication 28 Sections 343.1, 356 and Appendix A1). It is assumed in this standard that the hyphen is not stored with the ZIP Plus 4 value, but is added upon export for display.
XML Tag	<ZIPPlus4>
XML Model	<xsd:simpleType name="ZIPPlus4_type"> <xsd:restriction base="xsd:string"> <xsd:pattern value='[0-9]{4}' /> </xsd:restriction> </xsd:simpleType>
XML Example	<ZIPCode>35242</ZIPCode>

	<ZIPPlus4>3426</ZIPPlus4>
Quality Measures	Tabular Domain Measure
Quality Notes	

2.8.5.6 Country Name

Element Name	**Country Name**
Other common names for this element	Nation
Definition	The name of the country in which the address is located. A country is "an independent, self-governing, political entity."
Definition Source	Country Name: New Country: Framework Data Content Standard Part 5: Governmental Unit and Other Geographic Area Boundaries," (Table 13)
Data Type	characterString
Existing Standards for this Element	ISO 3166-1 Country Names (official short English version)
Domain of Values for this Element	Yes
Source of Values	ISO 3166-1 Country Names (official short English version)
How Defined (e.g., locally, from standard, other)	ISO 3166-1 Country Names (official short English version)
Example	1. **United States** 2. **Canada** 3. **Mexico**
Notes/Comments	1.Although the scope of this standard is restricted to US addresses, Country Name is included for two reasons: to facilitate reconciliation with address standards of other nations, and to accommodate files which mix addresses from the US and other countries. 2. ISO 3166-1 official short English names are specified because they a familiar and concise, and because ISO 3166-1 is specified in the UPU address standard. ISO 3166-1 also specifies a two-character abbreviations for each name, which are recognized within the postal profile of this standard. 3. The names and their abbreviations can be found at:

	http://www.iso.org/iso/en/prods-services/iso3166ma/02iso-3166-code-lists/list-en1.html
XML Tag	<CountryName>
XML Model	<xsd:simpleType name="CountryName_type"> <xsd:restriction base="xsd:string" /> </xsd:simpleType>
XML Example	<CountryName>CANADA</CountryName>
Quality Measures	Tabular Domain Measure
Quality Notes	

2.8.6 USPS Postal Address Elements

2.8.6.1 USPS Box Type

Element Name	**USPS Box Type**
Other common names for this element	PO Box; Box (Obsolete terms: Drawer, Lockbox, Bin, Caller, Firm Caller)
Definition	The name of the class of the container used for receipt of USPS mail. USPS Publication 28 requires the use of "PO Box" or "Box" for this element.
Definition Source	New
Data Type	characterString
Existing Standards for this Element	USPS Publication 28 sections 24, 25, and 28; section 238.1 (Military Addresses); and sections 293 and 295.6 (Puerto Rico Addresses)
Domain of Values for this Element	**PO Box** (if used in a USPS Postal Delivery Box address). **Box** (if used in a USPS Postal Delivery Route address
Source of Values	USPS Publication 28 sections 24, 25, and 28; section 238.1 (Military Addresses); and sections 293 and 295.6 (Puerto Rico Addresses)
How Defined (e.g., locally, from standard, other)	USPS Publication 28 sections 24, 25, and 28; section 238.1 (Military Addresses); and sections 293 and 295.6 (Puerto Rico Addresses)
Example	**PO Box** 6943 **PO Box** G **PO Box** 00145

	RR 4 **Box** 19-1A HC 68 **Box** 45
Notes/Comments	1. In USPS Postal Delivery Box addresses, "PO Box" is required for this element. "Post Office Box addresses are output as "PO Box NN" on the mailpiece." (USPS Publication 28 section 281). 2. In USPS Postal Delivery Route addresses, "Box" is required for this element. ---"Print rural route addresses on mailpieces as "RR N Box NN". (USPS Publication 28 section 241) ---"Print highway contract route addresses on mailpieces as "HC N Box NN". (USPS Publication 28 section 251) 3. The USPS Postal Delivery Box and USPS Postal Delivery Route address classes are defined in the Classification Part of this standard.
XML Tag	<USPSBoxType>
XML Model	<xsd:simpleType name="USPSBoxType_type"> <xsd:restriction base="xsd:string"> <xsd:pattern value='.*' /> </xsd:restriction> </xsd:simpleType>
XML Example	<USPSBox> **<USPSBoxType>PO Box</USPSBoxType>** <USPSBoxID>6943</USPSBoxId> </USPSBox>
Quality Measures	Tabular Domain Measure Range Domain Measure
Quality Notes	

855

856 ### 2.8.6.2 USPS Box ID

Element Name	**USPS Box ID**
Other common names for this element	PO Box Number; Box Number
Definition	The numbers or letters distinguishing one box from another within a post office or route.
Definition Source	New
Data Type	characterString

Existing Standards for this Element	USPS Publication 28 sections 24, 25, and 28; section 238.1 (Military Addresses); and sections 293 and 295.6 (Puerto Rico Addresses)
Domain of Values for this Element	Yes, within each post office
Source of Values	Local post office
How Defined (e.g., locally, from standard, other)	Local post office
Example	PO Box **6943** PO Box **G** PO Box **00145** RR 4 Box **19-1A** HC 68 Box **45**
Notes/Comments	1. USPS Box ID's may include numbers or letters, and may include a hyphen. 2. "Post Office Box numbers that are preceded by significant leading zeroes are identified in the ZIP+4 file by a hyphen (-) preceding the box number. Convert the hyphen into a zero on the output mailpiece." Example: Convert "PO BOX -0145" to "PO BOX 00145" on output from the ZIP+4 file. (USPS publication 28 Section 282)
XML Tag	<USPSBoxID>
XML Model	<xsd:simpleType name="USPSBoxId_type"> <xsd:restriction base="xsd:string"> <xsd:pattern value='.*' /> </xsd:restriction> </xsd:simpleType>
XML Example	<USPSBox> <USPSBoxType>PO Box</USPSBoxType> **<USPSBoxID>6943</USPSBoxId>** </USPSBox>
Quality Measures	Tabular Domain Measure Range Domain Measure
Quality Notes	

2.8.6.3 Complex Element: USPS Box

Element Name	USPS Box
Other common names for	PO Box, Box, Post Office Box

this element	(Obsolete terms: Lockbox, Drawer, Bin, Caller, Firm Caller)
Definition	A container for the receipt of USPS mail uniquely identified by the combination of a USPS Box Type and a USPS Box ID.
Syntax	{ USPS Box Type *} +{ USPS Box ID *}
Definition Source	New
Data Type	characterString
Existing Standards for this Element	USPS Publication 28 sections 24, 25, and 28; section 238.1 (Military Addresses); and sections 293 and 295.6 (Puerto Rico Addresses)
Domain of Values for this Element	See component elements.
Source of Values	See component elements.
How Defined (e.g., locally, from standard, other)	See component elements.
Example	**PO Box 246** Hillsdale, NJ 07642 **PO Box 1137** Saipan MP 96950-1137 RR 4 **Box 73** Grafton WV 26354 HC 4 **Box 100** Blanco TX 78606
Notes/Comments	A USPS Box location has no definite geographic relation to the location of the recipient of the mail.
XML Tag	<USPSBox>
XML Model	<xsd:complexType name="USPSBox_type"> <xsd:sequence> <xsd:element name="USPSBoxType" type="addr_type:USPSBoxType_type" maxOccurs="1" minOccurs="1"/> <xsd:element name="USPSBoxId" type="addr_type:USPSBoxId_type" maxOccurs="1" minOccurs="1"/> </xsd:sequence> </xsd:complexType>
XML Example	<USPSAddress> <USPSRoute> <USPSBoxGroupType>PSC</USPSGroupType> <USPSBOXGroupId>4</USPSGroupId> </USPSRoute> **<USPSBox>** **<USPSBoxType>BOX</USPSBoxType>** **<USPSBoxId>3</USPSBoxId>** **</USPSBox>** </USPSAddress>

Quality Measure	Pattern Sequence Measure
Quality Notes	

2.8.6.4 USPS Box Group Type

Element Name	**USPS Box Group Type**
Other common names for this element	See domain of values below.
Definition	A name for a type of postal delivery point or route containing a group of USPS Boxes.
Definition Source	New
Data Type	characterString
Existing Standards for this Element	USPS Publication 28 sections 24, 25, and 28; section 238.1 (Military Addresses); and sections 293, 295.6, and 295.7 (Puerto Rico Addresses)
Domain of Values for this Element	**RR** (Rural Route)(Obsolete terms: RD, RFD, Rural Delivery, Rural Free Delivery) **HC** (Contract Delivery Service Route) (Obsolete terms: Highway Contract Route, Star Route) **PSC** (Postal Service Center)(Overseas military postal address) **CMR** (Common Mail Room)(Overseas military postal address) **Unit** (Overseas military postal address)
Source of Values	USPS Publication 28 sections 24, 25, and 28; section 238.1 (Military Addresses); and sections 293, 295.6, and 295.7 (Puerto Rico Addresses)
How Defined (e.g., locally, from standard, other)	USPS Publication 28 sections 24, 25, and 28; section 238.1 (Military Addresses); and sections 293, 295.6, and 295.7 (Puerto Rico Addresses)
Example	1. **RR** 4, Box 10 2. **HC** 2, Box 7 3. **PSC** 4, Box 3 4. **CMR** 4, Box 2 5. **UNIT** 475, Box 690
Notes/Comments	1. This group includes rural routes, contract service delivery routes, postal service centers, overseas military common mail rooms and military unit numbers. 2. Contract Delivery Service Routes were formerly called Highway Contract Routes, and are still abbreviated "HC".

XML Tag	<USPSBoxGroupType>
XML Model	<xsd:simpleType name="USPSBoxGroupType_type"> <xsd:restriction base="xsd:string"> <xsd:pattern value='.*' /> </xsd:restriction> </xsd:simpleType>
XML Example	<USPSAddress> <USPSRoute> **<USPSBoxGroupType>PSC</USPSGroupType>** <USPSBOXGroupId>4</USPSGroupId> </USPSRoute> <USPSBox> <USPSBoxType>BOX</USPSBoxType> <USPSBoxId>3</USPSBoxId> </USPSBox> </USPSAddress>
Quality Measures	Tabular Domain Measure
Quality Notes	

2.8.6.5 USPS Box Group ID

Element Name	**USPS Box Group ID**
Other common names for this element	Rural route number; HC number; PSC/CMR/Unit Number
Definition	The numbers or letters distinguishing one route or distribution point from another route or distribution point of the same USPS Box Group Type.
Definition Source	New
Data Type	characterString
Existing Standards for this Element	USPS Publication 28 sections 24, 25, and 28; section 238.1 (Military Addresses); and sections 293, 295.6, and 295.7 (Puerto Rico Addresses)
Domain of Values for this Element	Yes
Source of Values	Local Post office
How Defined (e.g., locally, from standard, other)	Local Post office

Example	1. RR **4** Box 10 2. HC **2** Box 7 3. PSC **4** Box 3 4. CMR **4** Box 2 5. UNIT **475** Box 690
Notes/Comments	
XML Tag	<USPSBoxGroupID>
XML Model	<xsd:simpleType name="USPSBoxGroupId_type"> <xsd:restriction base="xsd:string"> <xsd:pattern value='.*' /> </xsd:restriction> </xsd:simpleType>
XML Example	<USPSAddress> <USPSRoute> <USPSBoxGroupType>PSC</USPSGroupType> **<USPSBOXGroupId>4</USPSGroupId>** </USPSRoute>* <USPSBox> <USPSBoxType>BOX</USPSBoxType> <USPSBoxId>3</USPSBoxId> </USPSBox> </USPSAddress>
Quality Measures	Tabular Domain Measure Range Domain Measure
Quality Notes	

2.8.6.6 Complex Element: USPS Route

Element Name	**USPS Route**
Other common names for this element	See component elements
Definition	A collection of boxes served from a single distribution point, and uniquely identified by a USPS Box Group Type and a USPS Box Group ID.
Syntax	{ USPS Box Group Type *} + { USPS Box Group ID *}
Definition Source	New
Data Type	characterString

Existing Standards for this Element	USPS Publication 28 sections 24, 25, and 28; section 238.1 (Military Addresses); and sections 293, 295.6, and 295.7 (Puerto Rico Addresses)
Domain of Values for this Element	See component elements
Source of Values	See component elements
How Defined (e.g., locally, from standard, other)	See component elements
Example	1. **RR 4** Box 10 2. **HC 2** Box 7 3. **PSC 4** Box 3 4. **CMR 4** Box 2 5. **Unit 475** Box 690
Notes/Comments	Unlike carrier routes and other USPS internal codes for mail sorting and delivery, the USPS Routes must be included in the address to provide sufficient information for delivery of mail.
XML Tag	<USPSRoute>
XML Model	<xsd:complexType name="USPSRoute_type"> <xsd:sequence> <xsd:element name="USPSBoxGroupType" type="addr_type:USPSBoxGroupType_type" maxOccurs="1" minOccurs="1"/> <xsd:element name="USPSBOXGroupId" type="addr_type:USPSBoxGroupId_type" maxOccurs="1" minOccurs="1"/> </xsd:sequence> </xsd:complexType>
XML Example	<USPSAddress> **<USPSRoute>** **<USPSBoxGroupType>PSC</USPSGroupType>** **<USPSBOXGroupId>4</USPSGroupId>** **</USPSRoute>** <USPSBox> <USPSBoxType>BOX</USPSBoxType> <USPSBoxId>3</USPSBoxId> </USPSBox> </USPSAddress>
Quality Measure	Pattern Sequence Measure
Quality Notes	USPS routes are locally determined. While these local routes are beyond the scope of the standard, they should be included in a local quality program.

2.8.6.7 Complex Element: USPS Address

Element Name	USPS Address
Other common names for this element	Postal Address
Definition	A USPS postal delivery point identified by a USPS Route and a USPS Box
Syntax	{ USPS Route *} + { USPS Box *}
Definition Source	New
Data Type	characterString
Existing Standards for this Element	USPS Publication 28 sections 24, 25, and 28; section 238.1 (Military Addresses); and sections 293, 295.6, and 295.7 (Puerto Rico Addresses)
Domain of Values for this Element	See Component Elements
Source of Values	USPS Publication 28 sections 24, 25, and 28; section 238.1 (Military Addresses); and sections 293, 295.6, and 295.7 (Puerto Rico Addresses)
How Defined (e.g., locally, from standard, other)	See component elements
Example	**RR 2 Box 223G** Dardanelle AR 72834 **HC 3 Box 330** Flasher, ND 58535 **PSC 802 Box 74** FPO AA 34058 **CMR 416 Box 100** APO AE 09140-0015 **Unit 2050 Box 4190** APO AP 96278-2050
Notes/Comments	
XML Tag	<UspsAddress>
XML Model	<xsd:complexType name="USPSAddress_type"> <xsd:sequence> <xsd:element name="USPSRoute" type="addr_type:USPSRoute_type" maxOccurs="1" minOccurs="1"/> <xsd:element name="USPSBox" type="addr_type:USPSBox_type" maxOccurs="1" minOccurs="1"/> </xsd:sequence> </xsd:complexType>
XML Example	<USPSAddress> <USPSRoute>

	<USPSBoxGroupType>PSC</USPSGroupType> <USPSBOXGroupId>4</USPSGroupId> </USPSRoute> <USPSBox> <USPSBoxType>BOX</USPSBoxType> <USPSBoxId>3</USPSBoxId> </USPSBox> </USPSAddress>
Quality Measure	Pattern Sequence Measure
Quality Notes	

867

868
2.8.6.8 USPS General Delivery Point

Element Name	**USPS General Delivery Point**
Other common names for this element	
Definition	A central point where mail may be picked up by the addressee. Two values are permitted: "General Delivery" (for post offices), and ship's names (for overseas military addresses).
Definition Source	New
Data Type	characterString
Existing Standards for this Element	Yes
Domain of Values for this Element	Yes
Source of Values	USPS
How Defined (e.g., locally, from standard, other)	USPS Publication 28 Section 26 (General Delivery Addresses); and section 238.1 (overseas military addresses)
Example	**General Delivery,** Tampa, FL 33602 **USCGC Hamilton**, FPO AP 96667-3931
Notes/Comments	For general delivery addresses, USPS Publication 28 section 261 specifies, "Use the words GENERAL DELIVERY, uppercase preferred, spelled out (no abbreviation), as the Delivery Address Line on the mailpiece. Each record will carry the 9999 add-on code.”

XML Tag	<USPSGeneralDeliveryPoint>
XML Model	<xsd:simpleType name="USPSGeneralDeliveryPoint_type"> <xsd:restriction base="xsd:string"> <xsd:pattern value='.*' /> </xsd:restriction> </xsd:simpleType>
XML Example	<USPSGeneralDeliveryPoint>USCGC Hamilton</USPSGeneralDeliveryPoint>
Quality Measures	Tabular Domain Measure
Quality Notes	

869

870 ### 2.8.7 USPS Address Lines

871 #### 2.8.7.1 Delivery Address

Element Name	**Delivery Address**
Other common names for this element	Delivery Address Line (USPS Publication 28); Location Address Text (EPA); Mailing Address Text (EPA)
Definition	The entire address, unparsed, except for the Place Name, State Name, ZIP Code, ZIP Plus 4, Country Name, and, optionally, Complete Subaddress elements.
Syntax	The Delivery Address syntax depends on the address class. Address class syntaxes are given in the Classification Part of this standard. The Delivery Address syntax is the same as the class syntax, except that the Delivery Address excludes the Place Name, State Name, ZIP Code, ZIP Plus 4, Country Name, and, optionally, Complete Subaddress elements.
Definition Source	New
Data Type	characterString
Existing Standards for this Element	USPS Publication 28
Domain of Values for this Element	No
Source of Values	NA
How Defined (e.g., locally, from	NA

standard, other)	
Attributes Associated with this Element	Delivery Address Type
Example	**123 Dartmouth College Highway, Suite 100,** Lyme, NH 03768 (Delivery Address Type = Subaddress Included) **123 Dartmouth College Highway,** Suite 100, Lyme, NH 03768 (Delivery Address Type = Subaddress Excluded)
Notes/Comments	1. The Delivery Address element corresponds to the Delivery Address Line defined in USPS Publication 28 (sec. 211, 231, 33, 341, and 343). 2. This element excludes Place Name, State Name, ZIP Code, and ZIP Plus 4 and Country Name, which together form the Place State ZIP complex element. 3. The Delivery Address typically includes the Complete Subaddress. However, there are sometimes reasons to omit or separate the Complete Subaddress from the Delivery Address. For example, the Complete Subaddress can hamper address geocoding, and contact lists often separate the Complete Subaddress from the rest of the feature address (see, e.g., the EPA Contact Information Data Standard). 4. The Delivery Address Type shows whether the Delivery Address includes or excludes the Complete Subaddress.
XML Tag	<DeliveryAddress>
XML Model	<xsd:complexType name="DeliveryAddress_type"> <xsd:extension base="xsd:string"> <xsd:attribute name="DeliveryAddressType" type="addr_type:DeliveryAddressType_type" /> </xsd:extension> </xsd:simpleContent> </xsd:complexType>
XML Example	<DeliveryAddress Delivery Address Type="Subaddress Included">123 Dartmouth College Highway, Suite 100</DeliveryAddress> <DeliveryAddress Delivery Address Type="Subaddress Excluded">123 Dartmouth College Highway, Suite 100</DeliveryAddress> <DeliveryAddress>123 Dartmouth College Highway, Suite 100</DeliveryAddress>
Quality Measures	Pattern Sequence Measure
Quality Notes	

2.8.7.2 Place State ZIP

Element Name	**Place State ZIP**
Other common names for this element	Last Line (USPS)
Definition	The combination of Complete Place Name, State Name, ZIP Code, ZIP Plus 4, and Country Name within an address. Complete Place Name and State Name are mandatory; the other elements are optional.
Syntax	{ Complete Place Name *} + { State Name *} + { ZIP Code } + { ZIP Plus 4 } + { Country Name }
Definition Source	New
Data Type	characterString
Existing Standards for this Element	Refer to component elements
Domain of Values for this Element	Refer to component elements
Source of Values	Refer to component elements
How Defined	Refer to component elements
Example	1. **Waterville ME 04901** 2. **Oxford MS 38655-4068** 3. **Florence, OR** 4. **Brattleboro, Windham County, VT**
Notes/Comments	1. Place State ZIP corresponds to the Last Line (or City, State, ZIP+4 line) as defined for postal addressing purposes in USPS Publication 28 (secs 211, 33, and 341). 2. ZIP Code and ZIP Plus 4 are recommended but not mandatory in the Place State ZIP element.
XML Tag	<PlaceStateZIP>
XML Model	<xsd:simpleType name="PlaceStateZip_type"> <xsd:restriction base="xsd:string"> <xsd:pattern value='.*' /> </xsd:restriction> </xsd:simpleType>
XML Example	<PlaceStateZIP>Brattleboro, Windham County, VT</PlaceStateZIP>

Quality Measures	Pattern Sequence Measure
Quality Notes	

2.9 Address Reference Systems

2.9.1 Address Reference Systems Introduction

An Address Reference System establishes the framework of rules, both spatial and non-spatial, adopted by an Address Authority for assigning addresses within the area it administers. The rules, in turn, provide the basis for address data quality tests that detect address anomalies and errors.

The Address Reference System includes, as needed, rules governing address numbering, street naming, block definition, subaddresses (suites, offices, apartments, etc.), and place names. The Address Reference System may also define address baselines, polylines, and breaklines to guide address numbering throughout the area. Finally, for identification and reference, an Address Reference System includes a name and identifier, the name of the Address Reference System Authority that administers it, the boundary of the area it administers, and reference to the official documents and maps where the rules are codified.

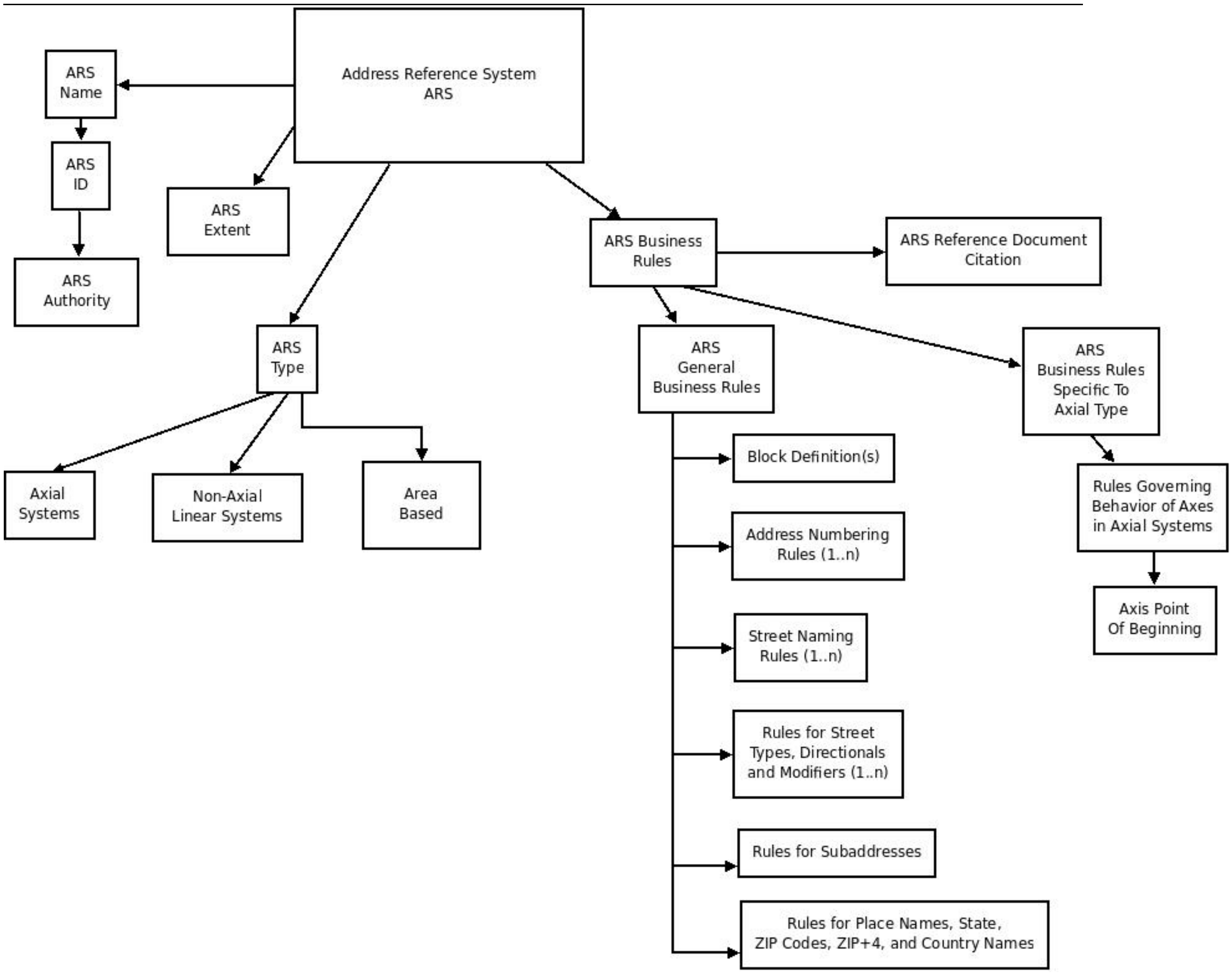
2.9.1.1 Working with Address Reference Systems

Address Reference Systems provide a framework for address assignment and for quality assurance of addresses. In order to use these within a Geographic Information System, the components of a system must be structured into a layer that includes the extent of the system

(Address Reference System Extent), and the reference grids, lines or points that govern address numbering throughout the area. In many cases, such grids have been constructed as graphic features that are not structured in a way to make them useful for developing Address Reference System Axis lines Address Reference System Axis Point Of Beginning locations, Address Reference System Reference Polylines, Address Reference System Range Breakpoints, Address Reference System Range Breaklines and for use in evaluating whether a specific address point falls in the correct place relative to the Address Reference System Rules. Thus it is important this the Address Reference System be created as intelligent geometry providing the tools needed to evaluate any address point found within the Address Reference System. It should also, where appropriate, utilize existing centerlines or other existing features so that exact matching is possible.

2.9.1.2 Types of Address Reference Systems

Address Reference Systems differ in detail from locality to locality, but in the United States all Address Reference Systems fit into one of three broad categories: axial, linear non-axial, and area non-axial. The categories differ fundamentally in whether and how the street system governs address numbering, and secondarily in the elements needed to compose them. Figure 1 diagrams the types and elements. Table 1 lists for each Address Reference System type, the elements required and permitted to compose it.



2.9.1.3 Axial Type Address Reference Systems

In axial Address Reference Systems, address numbering is organized around axes. The axes may be thoroughfares, rail lines, rivers, or imaginary lines (such as section lines in PLSS areas, lines of latitude and/or longitude, or arbitrarily drawn lines). Address axes typically extend from a common point of origin (the local "zero" point for address numbers), and all numbers increase with distance from the point of origin.

The axes, in turn, define the zero point for numbering along streets that cross the axes. Most commonly, axial system organize the streets and address numbering into a grid. In a simple case, if Main Street ran north-south from the town square, and State Street ran east-west, then:

- Address numbering for Main Street and State Street would increase as one proceeded away from the town square.
- Address numbering for other north-south streets would begin where they cross State Street and increase in parallel with Main Street.
- Address numbering for other east-west streets would begin where the cross Main Street and increase in parallel with State Street.

Often the geometric grid is interrupted or deformed by terrain, rivers, highways, rail lines, parks, or other major features. Occasionally there are more than four axes, or numbering does not begin at the same point for all axes.

2.9.1.4 Linear Non-Axial Address Reference Systems

In a linear non-axial Address Reference System, each thoroughfare is addressed independently of the other thoroughfares. There are no axes and there is no grid. Each thoroughfare has its own point of beginning for address numbering, and and numbers proceed according to an Address Reference System Numbering Rule from that point to the end of the thoroughfare or the boundary of the Address Reference System. Linear non-axial address reference systems are typically found in areas where the road network is sparse and intersections are few.

2.9.1.5 Area-Based Systems

In area-based Address Reference Systems, Complete Address Numbers are not assigned along a thoroughfare, but within an area denoted by a community name or a block number. Inside the area, address numbers might be assigned according to a spatial pattern (around the block, for example), or by parcel or lot numbers, or chronologically as the buildings are built.

Area-based Address Reference Systems are rare in the United States, but they may be found in gated communities, housing projects, Puerto Rican urbanizations, trailer courts, and similar developments that are built around interior walkways or roadways.

Table 1: Required, Optional, and Inapplicable Elements for Each Type of Address Reference System

Note: R - Required; O = Optional; NA = Not Applicable

Element name	**Axial**	**Linear Non-axial**	**Area Non-axial**
Address Reference System ID	R	R	R
Address Reference System Name	R	R	R
Address Reference System Authority	R	R	R
Address Reference System Extent	R	R	R
Address Reference System Type	R	R	R
Address Reference System Reference Document Citation	R	R	R
Address Reference System Rules	O	O	O
Address Reference System Numbering Rules	O	O	O
Address Reference System Block Rules	O	O	O
Address Reference System Street Naming Rules	O	O	O
Address Reference System Street Type Directional And Modifier Rules	O	O	O

Address Reference System Place Name State Country And ZIP Code Rules	O	O	O
Address Reference System Subaddress Rules	O	O	O
Address Reference System Axis	R	NA	NA
Address Reference System Axis Point Of Beginning	R	NA	NA
Address Reference System Reference Polyline	O	NA	NA
Address Reference System Range Breakpoint	O	NA	NA
Address Reference System Range Breakline	O	NA	NA
Address Reference System Range Polygon	O	NA	NA

2.9.2 Elements of an Address Reference System

2.9.2.1 Address Reference System Identification, Extent, and Authority

The general elements identify an Address Reference System and establish the source and extent of its authority. These elements are required for every Address Reference System. The general elements are: Address Reference System ID, Address Reference System Name, Address Reference System Authority, and Address Reference System Extent.

- The Address Reference System ID provides a unique identifier (typically an integer) for each Address Reference System administered by an Address Reference System Authority. This, plus the Address Reference System Authority, should be unique throughout the United States. Any Address Reference System Authority may administer multiple Address Reference Systems. For example, a county may have more than Address Reference System for unincorporated areas based on terrain changes, historical addressing patterns, or for other reasons. Cities may annex areas which have previously been addressed, and maintain the old Address Reference

System. Other Address Reference Systems may be established in the future as an area develops.

- The Address Reference System Name identifies the Address Reference System in a way that is meaningful to users.
- The Address Reference System Authority element identifies the agency and/or jurisdiction with administrative responsibility for the Address Reference System.
- The Address Reference System Extent defines the geographic boundaries of the area within which addressing is governed by the Address Reference System. The Address Reference System Extent may or may not follow jurisdictional boundaries. There may also be areas within an Address Reference System that are excluded from that Address Reference System because they are addressed according to different rules.
- The Address Reference System Reference Document Citation states where to find the authoritative documents that officially establish the Address Reference System. The documents may include a map of the reference system showing the extent, address numbering system, axes, and other features; a statement of the addressing rules described below; an addressing procedures manual and forms; and an address ordinance.

2.9.2.2. Address Reference System Rules

The remaining elements describe the types of rules that might be adopted by an Address Reference System Authority to govern addressing processes. Due to the variety of local conditions and preferences, not all elements will be applicable to any given system, and all of

these presented are optional elements. The rules are collected into the Address Reference System Rules, which incorporates the:

- Address Reference System Numbering Rules,
- Address Reference System Block Rules,
- Address Reference System Street Naming Rules,
- Address Reference System Street Type Directional And Modifier Rules,
- Address Reference System Place Name State Country And ZIP Code Rules,
- Address Reference System Subaddress Rules.

2.9.2.3 Address Numbering Rules

Address numbering rules specify how numbers are assigned along thoroughfares, including what features are numbered. They govern when numbers increase, assign even and odd numbers to sides of streets, and specify the beginning points for numbering. They may also specify if and how address ranges relate to blocks.

- What Features are Given Address Numbers?

  In addition to permanent primary structures, other features that can be numbered include vacant lots, secondary structures such as detached garages or farm outbuildings, temporary and seasonal structures, additional entrances of large buildings, non-structured uses such as open parking lots, infrastructure features such as cell towers, pump and metering stations, substations and transformers.

- Increase and Interval Rules for Address Numbering

  In the United States, address numbers increase according to one of three rules:

1. Distance rule - numbers are assigned according to distance along the thoroughfare (e.g., 1000 numbers per mile, 500 on either side, or 2 per 10.56 feet).
2. "Hundred block" Rule - where streets are laid out in a regular city grid, each block may be given a range of 100 numbers (50 per side), e.g. the 1400 block of Cherry Street. Within each block, numbers may be allocated by distance, or proportionally to the length of the block. If blocks have a fixed length (e.g ten per mile), then this rule can work just like a distance rule.
3. Sequentially - properties or buildings are numbered sequentially, regardless of distance or blocks. The numbers may increase by twos, or they may increase by a larger interval (4, 6, 8, 14, etc.) to leave intermediate numbers for future divisions of land.

- Parity Rules

Parity rules assign even numbers to one side of the thoroughfare and odd numbers to the other side.

- Point(s) of Beginning for Numbering

In axial address reference systems, numbering begins where a thoroughfare intersects (or would intersect) its axis. In non-axial systems, the point of beginning is defined separately for each thoroughfare. Many non-axial systems follow the federal and state highway milepost practice of starting numbering at the southern or western end of the thoroughfare (or boundary of a jurisdiction), and increasing numbers to the north or east.

- Block Rules and Address Range Rules

These rules derive from the increase and interval rules described above. The Address Reference System Block Rules define how the system is organized into blocks for addressing purposes, and whether blocks break at intersections and begin with a new series of numbers, or whether numbering is sequentially ordered along a street without regard to intersecting streets. Such rules also define what constitutes a block break, as many systems do not recognize alleys, or three-way (T) intersections as block breaks.

Address ranges are created using the low and high numbers for each block or other unit defined by the system. Rules pertaining to address ranges are contained with the Address Reference System Block Rules.

2.9.2.4 Street Naming Rules

Street naming rules define what Street Names may be allowed or prohibited, rules to prevent duplicate names, any language considerations, and whether Street Names must follow particular themes or orders (such as themes for names in subdivisions, or alphabetical or numerical orders).

2.9.2.5 Street Name Type, Directional, and Modifier Rules

The Address Reference System Street Type Directional And Modifier Rules govern the use of street types, directionals and quadrants, and modifiers in Complete Street Names. Street type rules might specify a limited list of approved types (such as the list in USPS Publication 28 Appendix C.2), whether the type must precede or follow the street name, and whether specific types are reserved for thoroughfares with specific functional characteristics. Directional rules

include whether a quadrant or cardinal direction (or rarely both) is required, optional or prohibited in an address, and, if so, whether it must precede or follow the street name and type. Modifier rules may to allow or prohibit Street Name Pre Modifiers or Street Name Post Modifiers, or specify which modifiers are permitted.

2.9.2.6 Subaddress Rules

These rules, if included, cover the naming and recording of any subaddresses within structures, such as apartments, office suites, campuses, mobile home parks, industrial plants, malls and retail centers with multiple tenants, etc.

2.9.2.7 Place Name, State, Country and ZIP Code Rules

These rules define the specific allowable combinations of a Place Name, State, and ZIP code in the Address Reference System, and provide input to checking these elements for quality. Unlike other elements of the address, which must be defined locally, State Name abbreviations and ZIP Codes are defined by the USPS, and Country Names are defined by international standard (ISO 3166-1).

2.9.2.8 Address Axis Rules

An Address Reference System Axis defines the points of beginning for address numbers for the streets that intersect it. The Address Reference System Axis pairs are often the "dividers" for quadrants, or directional designations. Finally, an Address Reference System Axis may also function as "rulers" to define block breaks and address ranges for thoroughfares with

similar directionality (e.g. north-south, or east-west streets) within the Address Reference System.

In theory, every street within an axial Address Reference System can be linked to an axis, either by intersection, or a virtual extension of the street centerline to the axis, or by interpolation (for streets that are set at an angle to the axes, and cannot be projected to intersect with only one of the axes). In practice, however, most jurisdictions with axial Address Reference System create a "grid" by using major through streets to create "blocks" of equal address ranges. For each Address Reference System Axis an Address Reference System Axis Point Of Beginning must be identified. These elements are used only within Axial systems.

2.9.2.9 Reference Polyline, Breakpoint, and Breakline and Polygon Elements

The Reference Polyline, Breakpoint, Breakline and Polygon elements are utilized primarily for quality assurance and address assignment purposes. These are optional elements used in Axial systems.

An address grid can be constructed by identifying the Address Reference System Range Breakpoints on a sufficient number of streets in the Address Reference System, and then joining equivalent breakpoints with an Address Reference System Range Breakline. By developing these breaklines, a set of areas are defined for each range of 100 (or some specified number of) numbers, and within them, shorter streets can be accurately addressed. If desired, the Address Reference System Range Breaklines can be used within a GIS environment to create polygons with equal address range values. These are then stored as Address Reference

System Range Polygon. Streets used for the development of the breakpoints and breaklines (including the Address Reference System Axis elements can be identified using the Address Reference System Reference Polyline element.

Together, Address Reference System Axis, Address Reference System Reference Polyline, Address Reference System Range Breakpoint, Address Reference System Range Breakline, Address Reference System Range Polygon form a geographic reference framework for the overall address numbering system within an axial Address Reference System. The framework guides assignment of new address numbers, and it provides the basis for important quality assurance tests.

2.9.3 Address Reference System Elements

2.9.3.1 Address Reference System ID

Element Name	Address Reference System ID
Other common names for this element	
Definition	A unique identifier of the Address Reference System for a specified area (Address Reference System Extent).
Definition Source	New
Data Type	Integer
Existing Standards for this Element	None
Domain of Values for this Element	Locally defined
Source of Values	Local
How Defined (e.g., locally,	Locally

from standard, other)	
Examples	For examples, see the Complex Element: Address Reference System.
Notes/Comments	The Address Reference System ID provides a reliable attribute to link an individual address record or a group of address records to a specific Address Reference System. This attribute identifies the specific rules that should be used in evaluating the address record. The Address Reference System ID must be unique to the Address Authority.
XML Tag	<AddressReferenceSystemId>
XML Model	<xsd:simpleType name="AddressReferenceSystemId_type"> <xsd:restriction base="xsd:integer" /> </xsd:simpleType>
XML Example	<AddressReferenceSystemId>55</AddressReferenceSystemId>
Quality Measures	Tabular Domain Measure
Quality Notes	Where geometry for the address reference system is available, the boundaries should be checked as well to support spatial queries.

2.9.3.2 Address Reference System Name

Element Name	**Address Reference System Name**
Other common names for this element	
Definition	The name of the address system used in a specified area (Address Reference System Extent).
Definition Source	New
Data Type	characterString
Existing Standards for this Element	None
Domain of Values for this Element	Locally defined
Source of Values	Local
How Defined (e.g., locally, from standard, other)	Locally
Examples	For examples, see the Complex Element: Address Reference System.

Notes/Comments	In some cases, the Address Reference System Name may simply be the city or county name, such as "Town of Fairplay Address Reference System." In other cases, it may provide a name for the address reference system for a smaller area within a jurisdiction, such as "Boulder County Mountain Addressing System."
XML Tag	<AddressReferenceSystemName>
XML Model	<xsd:simpleType name="AddressReferenceSystemName_type"> <xsd:restriction base="xsd:string" /> </xsd:simpleType>
XML Example	<AddressReferenceSystemName>Mountain Addressing Scheme</AddressReferenceSystemName> <AddressReferenceSystemName>pre-1990 System</AddressReferenceSystemName>
Quality Measures	Tabular Domain Measure
Quality Notes	Where geometry for the address reference system is available, the boundaries should be checked as well to support spatial queries.

1103

1104

2.9.3.3 Address Reference System Authority

Element Name	Address Reference System Authority
Other common names for this element	
Definition	The name of the authority or jurisdiction responsible for the creation and/or maintenance of an Address Reference System for a given area.
Definition Source	New
Data Type	characterString
Existing Standards for this Element	None
Domain of Values for this Element	None.
Source of Values	Local
How Defined	Defined locally
Example	**City of Orono**, ME;

	Commander, Bolling Air Force Base, Washington, DC
Notes/Comments	The agency responsible for creating or maintaining an Address Reference System may or may not be the same as the Address Authority responsible for assigning and maintaining the addresses in a given area.
XML Tag	<AddressReferenceSystemAuthority>
XML Model	<xsd:simpleType name="AddressReferenceSystemAuthority_type"> <xsd:restriction base="xsd:string" /> </xsd:simpleType>
XML Example	<AddressReferenceSystemAuthority>Commander, Bolling Air Force Base</AddressReferenceSystemAuthority> <AddressReferenceSystemAuthority>City of Orono</AddressReferenceSystemAuthority>
Quality Measure	Tabular Domain Measure
Quality Notes	

2.9.3.4 Address Reference System Extent

Element Name	**Address Reference System Extent**
Other common names for this element	
Definition	Boundary of the area(s) within which an Address Reference System is used.
Definition Source	New
Data Type	Geometry (Multisurface), as defined in the Open Geospatial Consortium's "OpenGIS(R) Geography Markup Language (GML)" version 3.1.1 (see Appendix A for a complete citation)
Existing Standards for this Element	NA
Domain of Values for this Element	Coordinate values within the geometric areal extent of the Address Reference System
Source of Values	Source of spatial data collection.
How Defined (e.g., locally, from standard, other)	Locally defined.

Examples	Address Reference System Extent: <gml:MultiSurface> <gml:surfaceMember> <gml:Polygon> <gml:exterior> <gml:LinearRing> <gml:posList>1000 1000 1000 25000 20000 1000 20000 25000 1000 1000</gml:posList> </gml:LinearRing> </gml:exterior> </gml:Polygon> </gml:surfaceMember> </gml:MultiSurface>
Notes/Comments	An Address Reference System may include the entire area of a city or county jurisdiction, or it may only include a portion thereof. Military bases, and some university campuses are addressed under Address Reference Systems that are maintained by the Base Commander for military bases, and by the State Department of Education (or the University system) for campuses. These often exist within the boundaries of a city, and are within county areas as well, but have their own schemes. Each Address Reference System is defined geographically, and should not (although many do so) overlap other Address Reference Systems that are in current use. Historical Address Reference System extents may be maintained, especially where an area under a county Address Reference System has been annexed into a city. The city may choose to maintain the county's numbering, and it will be useful, if additional development occurs, to have access to the previous Address Reference System to insure correct and consistent addressing with it.
XML Tag	<AddressReferenceSystemExtent>
XML Model	<xsd:complexType name="AddressReferenceSystemExtent_type"> <xsd:complexContent> <xsd:restriction base="gml:MultiSurfaceType"/ > </xsd:complexContent> </xsd:complexType>
XML Example	<AddressReferenceSystemExtent> <gml:MultiSurface> <gml:surfaceMember> <gml:Polygon>

	<gml:exterior> <gml:LinearRing> <gml:posList>1000 1000 1000 25000 20000 1000 20000 25000 1000 1000</gml:posList> </gml:LinearRing> </gml:exterior> </gml:Polygon> </gml:surfaceMember> </gml:MultiSurface> </AddressReferenceSystemExtent>
Quality Measures	None
Quality Notes	Check the boundary against the Address Reference System Description.

2.9.3.5 Address Reference System Type

Element Name	**Address Reference System Type**
Other common names for this element	
Definition	The category of address reference system in use. The type of reference system determines and guides the assignment of numbers within the Address Reference System Extent.
Definition Source	New
Data Type	characterString
Existing Standards for this Element	None
Domain of Values for this Element	Yes: Axial, Linear Non-Axial, Area Based
Source of Values	FGDC Address Data Content Standard, Part One
How Defined	Local determination
Example	The Address Reference System for the District of Columbia is an axial (grid) system.
Notes/Comments	1. An Address Reference System Type identifies the overall classification of the reference system. 2. The types include:

	a) Axial systems based on setting forth a framework consisting of streets, or other geometric lines to identify address numbering rules. Axial type systems include: i) grids based on either the street pattern, a geographic set of lines such as those forming the Public Land Survey System Grid, longitude and latitude lines or similar lines. ii) Radial patterns organized around primary arterial streets originating at a central point. b) Linear Non-axial systems, often found in areas of complex terrain where streets do not tend to travel in straight lines for any distance. i) Distance based systems in which each road has a defined starting point, and ii) Other types of linear organizational constructs that create a logical framework in which addresses are assigned. c) Area-based systems where the address numbers in a specified area are assigned by a non-geometric method, including chronological (where a number is assigned in the order in which a building or property is created regardless of its location), or by lot numbers (where these are not arranged in the usual sequential patterns found in axial and linear non-axial systems), or other means. 3. Some of these systems may have sub-types. In grid systems, some provide for 100 numbers per "block", others are numbered sequentially without regard for block breaks. In places with radial street patterns, axis streets or lines may originate at one or more places. In some cases a grid or radial pattern may extend beyond its original area, and be expanded in an outlying area using numbering that is continued from the original area. 4. The basis for numbering within any of these systems is created as an attribute of the system. Numbering rules are documented in the Address Reference System Numbering Rules element. It is expected to be applied consistently throughout the extent of the reference system, although in practice this is often not true. Additional information on Address Reference Systems may be found in the Address Reference Systems Introduction.
XML Tag	<AddressReferenceSystemType>
XML Model	<xsd:simpleType name="AddressReferenceSystemType_type"> <xsd:restriction base="xsd:string"> <xsd:enumeration value="Axial"></xsd:enumeration> <xsd:enumeration value="Grid"></xsd:enumeration> <xsd:enumeration value="Radial"></xsd:enumeration> <xsd:enumeration value="Linear Non-Axial"></xsd:enumeration> <xsd:enumeration value="Distance"></xsd:enumeration> <xsd:enumeration value="Area Based"></xsd:enumeration>

	</xsd:restriction> </xsd:simpleType>
XML Example	<AddressReferenceSystemType>Grid</AddressReferenceSystemType>
Quality Measure	Tabular Domain Measure
Quality Notes	

2.9.3.6 Complex Element: Address Reference System Rules

Element Name	**Address Reference System Rules**
Other common names for this element	Addressing Rules
Definition	The rules by which address numbers, street names and other components of a thoroughfare address are determined.
Definition Source	New
Data Type	characterString
Existing Standards for this Element	None
Domain of Values for this Element	Locally defined, see component elements
Source of Values	Local
How Defined	Defined locally, often by ordinance and encoded in terms of a spatial referencing system, described in the file-level metadata per FGDC's Content Standard for Digital Geospatial Metadata
Example	See component elements.
Notes/Comments	The rules are dependent upon the type of Address Reference System, and may also be explicitly provided in the component elements of Address Reference System Rules, or they may be referenced in the Address Reference System Reference Document Citation.
XML Tag	<AddressReferenceSystemRules>
XML Model	<xsd:complexType name="AddressReferenceSystemRules_type"> <xsd:sequence> <xsd:element name="AddressReferenceSystemBlockRules" type="addr_type:AddressReferenceSystemBlockRules_type" minOccurs="0" maxOccurs="unbounded"></xsd:element>

	<xsd:element name="AddressReferenceSystemNumberingRules" type="addr_type:AddressReferenceSystemNumberingRules_type" minOccurs="0" maxOccurs="unbounded"></xsd:element> <xsd:element name="AddressReferenceSystemStreetNamingRules" type="addr_type:AddressReferenceSystemStreetNamingRules_type" minOccurs="0" maxOccurs="unbounded"></xsd:element> <xsd:element name="AddressReferenceSystemStreetTypeDirectionalAndModfierRules" type="addr_type:AddressReferenceSystemStreetTypeDirectionalAndModifierRules_type" minOccurs="0" maxOccurs="unbounded"></xsd:element> <xsd:element name="AddressReferenceSystemPlaceNameStateCountyAndZipCodeRules" type="addr_type:AddressReferenceSystemPlaceNameStateCountryAndZipCodeRules_type" minOccurs="0" maxOccurs="unbounded"></xsd:element> <xsd:element name="AddressReferenceSystemSubaddressRules" type="addr_type:AddressReferenceSystemSubaddressRules_type" minOccurs="0" maxOccurs="unbounded"></xsd:element> </xsd:sequence> </xsd:complexType>
XML Example	
Quality Measures	Address Reference System Rules Measure
Quality Notes	

2.9.3.7 Address Reference System Block Rules

Element Name	Address Reference System Block Rules
Other common names for this element	
Definition	This element defines a block in an Address Reference System, and sets forth the rules for block ranges and block breaks.
Definition Source	New
Data Type	characterString
Existing Standards for this Element	None
Domain of Values for	Locally defined

this Element	
Source of Values	Local
How Defined	Defined locally, often by ordinance and encoded in terms of a spatial referencing systems, described in the file-level metadata per FGDC's Content Standard for Digital Geospatial Metadata
Example	1. "A block is defined as a street segment between its points of intersection with other street segments at either end." 2. A block shall contain 100 address numbers, and shall begin with the 00 value on one side, and the 01 value on the other side." 3. "A block shall be defined as one mile along a single street regardless of the intersection of the street with any other streets."
Notes/Comments	Parity, meaning the definition of which side of a street shall be given the odd numbers and which side the even numbers in a range is defined in the Address Range Parity element.
XML Tag	<AddressReferenceSystemBlockRules>
XML Model	<xsd:simpleType name="AddressReferenceSystemBlockRules_type"> <xsd:restriction base="xsd:string" /> </xsd:simpleType>
XML Example	<AddressReferenceSystemBlockRules>A block is defined as a street segment between its points of intersection with other street segments at either end.</AddressReferenceSystemBlockRules>
Quality Measures	See Address Reference System Rules Measure.
Quality Notes	

2.9.3.8 Address Reference System Numbering Rules

Element Name	Address Reference System Numbering Rules
Other common names for this element	
Definition	The rules for numbering along a thoroughfare, including parity (odd/even side definition), and numbering increment distance and value.
Definition Source	New
Data Type	characterString

Existing Standards for this Element	None
Domain of Values for this Element	Locally defined.
Source of Values	Local
How Defined	Defined locally
Example	Address Numbering Rules: Odd numbers are on the south and west, and even numbers on the north and east sides of all streets. There will be one address increment allocated every 5.28 feet, resulting in 1000 addresses possible in each mile of road. The addresses will increase by a value of one unit at each increment.
Notes/Comments	1. In assigning addresses it is important to know which side of a street should be assigned odd numbers, and which even. 2.Additionally, the distance between numbers should be specified. In some cases, this is given as a number of feet or meters, while in others, it is given as a number of addresses per block or per mile. 3. The amount by which the address number is to be increased at each increment should be defined. In many cases the next sequential number is used, e.g. 1, 3, 5, etc., while in other cases, the increment may be 2 units, 4 units or any other number determined appropriate by the Address Reference System Authority. 4. If any specific numbers are to be prohibited for local reasons, these should be identified here as well. 5. The rules for how blocks are numbered and where breaks occur are listed in the Address Reference System Block Rules element.
XML Tag	<AddressReferenceSystemNumberingRules>
XML Model	<xsd:simpleType name="AddressReferenceSystemNumberingRules_type"> <xsd:restriction base="xsd:string" /> </xsd:simpleType>
XML Example	<AddressReferenceSystemNumberingRules> 1. In assigning addresses it is important to know which side of a street should be assigned odd numbers, and which even. 2.Additionally, the distance between numbers should be specified. In some cases, this is given as a number of feet or meters, while in others, it is given as a number of addresses per block or per mile. 3. The amount by which the address number is to be increased at each increment should be defined. In many cases the next sequential number is used, e.g. 1, 3, 5,

	etc., while in other cases, the increment may be 2 units, 4 units or any other number determined appropriate by the Address Reference System Authority. </AddressReferenceSystemNumberingRules>
Quality Measures	See Address Reference System Rules Measure.
Quality Notes	

2.9.3.9 Address Reference System Street Naming Rules

Element Name	**Address Reference System Street Naming Rules**
Other common names for this element	
Definition	The rules for the selection and use of street names within an Address Reference System
Definition Source	New
Data Type	characterString
Existing Standards for this Element	None
Domain of Values for this Element	Locally defined
Source of Values	Local
How Defined	Defined locally, often by ordinance or regulation
Example	1. Street names shall not be duplicated within the extent of the City of Anywhere Address Reference System. 2. Streets running north-south shall be numbered, beginning at Main Street, and shall be called Avenues, while streets running east-west shall be given letter names (e.g. A, B, C) and shall be Streets. 3. Street names that are vulgar, profane, obscene, or contain racial, ethnic, religious or sexual terms shall not be permitted. 4. Streets within a subdivision shall have a theme, such as animals, birds, flowers, trees, etc. to unify the street naming and give the subdivision identify.
Notes/Comments	Specific street naming rules are helpful in maintaining unique street names and preserving existing patterns of street names that were historically

	established.
XML Tag	<AddressReferenceSystemStreetNamingRules>
XML Model	<xsd:simpleType name="AddressReferenceSystemStreetNamingRules_type"> <xsd:restriction base="xsd:string" /> </xsd:simpleType>
XML Example	<AddressReferenceSystemStreetNamingRules> 1. Street names shall not be duplicated within the extent of the City of Anywhere Address Reference System. 2. Streets running north-south shall be numbered, beginning at Main Street, and shall be called Avenues, while streets running east-west shall be given letter names (e.g. A, B, C) and shall be Streets. 3. Street names that are vulgar, profane, obscene, or contain racial, ethnic, religious or sexual terms shall not be permitted. 4. Streets within a subdivision shall have a theme, such as animals, birds, flowers, trees, etc. to unify the street naming and give the subdivision identify. </AddressReferenceSystemStreetNamingRules>
Quality Measures	See Address Reference System Rules Measure.
Quality Notes	See Address Reference System Rules Measure.

2.9.3.10 Address Reference System Street Type Directional And Modifier Rules

Element Name	**Address Reference System Street Type Directional And Modifier Rules**
Other common names for this element	
Definition	Rules pertaining to the use of street types (suffix and prefix), directionals (prefix and suffix), and modifiers (prefix and suffix) of street names.
Definition Source	New
Data Type	characterString
Existing Standards	None

for this Element	
Domain of Values for this Element	Locally defined
Source of Values	Local
How Defined	Defined locally, often by ordinance or regulation
Example	1. Only those street types included in the Anytown Address Reference System list of street types may be used in Anytown. 2. Prefix types may be used. 3. Only the words "Old" and "New" may be used as Pre-Modifiers. The words "Extended", "Bypass" and "Overpass" may be used as post-modifiers.
Notes/Comments	1. Many communities have specific rules about the street types that are permitted, and further rules about the functional classes of streets to which various types can be applied. For example, the type "Boulevard" may only be used with a primary arterial, while "Court" may only be used with a short (one block) cul-de-sac or dead-end road. Additionally, the use of prefix types (e.g. "Avenue B", or "Calle San Antonio") is regulated in some places. 2.The use of directionals is often complex. In some Axial Address Reference Systems, quadrants are defined for specific areas bounded by the Axes. In others, the part of the area in which a street is located is described by "North" or "West". The Address Reference System provides that these rules and the areas described for the use of directionals can be documented. 3. Modifiers are words that are separated from the name by either types or directionals. The use of these may be regulated by local rules which are documented in this element. 4. The U.S. Postal Service, in Publication 28 provides a list of recognized street types, and directional values. The USPS does not recognize prefix types, and includes them with the Street Name (not recommended by this Standard), and also requires that any street type not included in Appendix C of Publication 28 be incorporated into the Street Name (also not recommended by this Standard). Modifiers are also not recognized separately by the USPS. For mailing purposes, the Complete Street Name element concatenates all of the parts of a Street Name, and is compatible with USPS standards.
XML Tag	<AddressReferenceSystemStreetTypeDirectionalAndModifierRules>
XML Model	<xsd:simpleType name="AddressReferenceSystemStreetTypeDirectionalAndModifierRules_type">

	<xsd:restriction base="xsd:string" /> </xsd:simpleType>
XML Example	<AddressReferenceSystemStreetTypeDirectionalAndModifierRules> 1. Only those street types included in the Anytown Address Reference System list of street types may be used in Anytown. 2. Prefix types may be used. 3. Only the words "Old" and "New" may be used as Pre-Modifiers. The words "Extended", "Bypass" and "Overpass" may be used as post-modifiers. </AddressReferenceSystemStreetTypeDirectionalAndModifierRules>
Quality Measures	See Address Reference System Rules Measure.
Quality Notes	

2.9.3.11 Address Reference System Place Name State Country And ZIP Code Rules

Element Name	**Address Reference System Place Name State Country And ZIP Code Rules**
Other common names for this element	
Definition	This element contains rules for the use of place names, state names, country names, and ZIP Codes within the jurisdiction of an Address Authority.
Definition Source	New
Data Type	characterString
Existing Standards for this Element	Existing Rules for State Name abbreviations and Country Name abbreviations (see those elements for citations).
Domain of Values for this Element	Locally defined
Source of Values	Local
How Defined	Defined locally, often by ordinance and regulation
Example	1. "All addresses within the Extent of this Address Reference System shall have the Municipal Place Name of "Anytown" and the State Name of "OHIO"." 2. "The following community Place Names may be used within this Address

	Reference System Extent: New Hope, Pine Level, Red Oak Village. The areas of these communities are shown on the map attached to the Address Ordinance for Any County."
Notes/Comments	The combinations of place names with state names, and ZIP Codes are defined by the Address Authority for all areas within Address Reference System Extent. For all areas outside the Extent, which are found in the mailing addresses used by a local government, or other user, the USPS is usually the best source of the proper association of a place name (community, city or place) with a State Name, and ZIP Code. For Country Names, rules usually specify how a Country Name will be used (fully spelled out, abbreviated, etc.) may be documented here. Further information on the standards and rules that are applied to State Names and Country Names are found in the element descriptions.
XML Tag	<AddressReferenceSystemPlaceNameStateCountryAndZipCodeRules>
XML Model	<xsd:simpleType name="AddressReferenceSystemPlaceNameStateCountryAndZipCodeRules_type"> <xsd:restriction base="xsd:string" /> </xsd:simpleType>
XML Example	<AddressReferenceSystemPlaceNameStateCountryAndZipCodeRules> 1. "All addresses within the Extent of this Address Reference System shall have the Municipal Place Name of "Anytown" and the State Name of "OHIO"." 2. "The following community Place Names may be used within this Address Reference System Extent: New Hope, Pine Level, Red Oak Village. The areas of these communities are shown on the map attached to the Address Ordinance for Any County." </AddressReferenceSystemPlaceNameStateCountryAndZipCodeRules>
Quality Measures	See Address Reference System Rules Measure.
Quality Notes	

2.9.3.12 Address Reference System Subaddress Rules

Element Name	**Address Reference System Subaddress Rules**
Other common names for this element	
Definition	The rules that are applied to the addressing of areas within structures as subaddresses (units, suites, apartments, spaces, etc.) within a given Address

	Reference System
Definition Source	New
Data Type	characterString
Existing Standards for this Element	None
Domain of Values for this Element	Locally defined
Source of Values	Local
How Defined	Defined locally, often by ordinance or procedures manual.
Example	1. Apartments are addressed using a four digit number where the first number represents the building, the second number the floor on which the unit is located, and the third and fourth numbers the individual apartment unit. 2. In a multi-story building, suites will be numbered in a clockwise manner from the elevator lobby, using even numbers on the right hand side, and odd numbers on the left hand side of the hallway. If the hallway is a single corridor, then the numbers will be assigned from one end of the structure to the other, in the same direction as the addresses on the street on which the building is addressed.
Notes/Comments	The rules for subaddresses may include the methods by which subaddresses are applied in a given situation. The rules may also specify the words that are allowed to identify subaddress types, such as unit, suite, space, apartment, and to prohibit the use of others.
XML Tag	<AddressReferenceSystemSubaddressRules>
XML Model	<xsd:simpleType name="AddressReferenceSystemSubaddressRules_type"> <xsd:restriction base="xsd:string" /> </xsd:simpleType>
XML Example	<AddressReferenceSystemSubaddressRules> 1. Apartments are addressed using a four digit number where the first number represents the building, the second number the floor on which the unit is located, and the third and fourth numbers the individual apartment unit. 2. In a multi-story building, suites will be numbered in a clockwise manner from the elevator lobby, using even numbers on the right hand side, and odd numbers on the left hand side of the hallway. If the hallway is a single corridor, then the numbers will be assigned from one end of the structure to the other, in the same direction as the addresses on the street on which the building is addressed. </AddressReferenceSystemSubaddressRules>
Quality Measures	See Address Reference System Rules Measure.

Quality Notes	

2.9.3.13 Address Reference System Axis

Element Name	**Address Reference System Axis**
Other common names for this element	
Definition	The line that defines the points of origin for address numbering along thoroughfares that intersect it, or which are numbered in parallel to streets that intersect it. It may be a road, another geographic feature, or an imaginary line.
Definition Source	New
Data Type	Geometry (Multicurve), as defined in the Open Geospatial Consortium's "OpenGIS(R) Geography Markup Language (GML)" version 3.1.1 (see Appendix A for a complete citation)
Existing Standards for this Element	None
Domain of Values for this Element	Locally defined
Source of Values	Local
How Defined	Defined locally, often by ordinance and encoded in terms of a spatial referencing systems, described in the file-level metadata per FGDC's Content Standard for Digital Geospatial Metadata
Example	Address Reference System Axis: <gml:MultiCurve> <gml:curveMember> <gml:Curve> <gml:segments> <gml:LineStringSegment> <gml:posList>1000 15000 20000 15000 </gml:posList> </gml:LineStringSegment> </gml:segments>/gml:Curve> </gml:curveMember> </gml:MultiCurve>

Notes/Comments	1. An Address Reference System Axis creates the beginning point for assigning Complete Address Numbers to thoroughfares that cross it, and it may guide the assignment of Complete Address Numbers along parallel thoroughfares. 2. An Address Reference System Axis is typically a road, but it may also be a line derived from a Public Land Survey System (PLSS) grid or a river (common in riverfront cities), a rail line, or an imaginary line (e.g. the east-west centerline of the national mall in Washington, DC). 3. Axis lines may cross, radiate or branch. 4. It may also provide a "measuring device" for the extension of numbers along parallel streets, especially where there is a gap in development within a scheme. 5. Axis lines may also define quadrants or areas in which certain directionals may be required for street names and addresses.
XML Tag	<AddressReferenceSystemAxis>
XML Model	<xsd:complexType name="AddressReferenceSystemAxis_type"> <xsd:complexContent> <xsd:restriction base="gml:MultiCurveType"> </xsd:restriction> </xsd:complexContent> </xsd:complexType>
XML Example	<AddressReferenceSystemAxis> <gml:MultiCurve> <gml:curveMember> <gml:Curve> <gml:segments> <gml:LineStringSegment> <gml:posList>1000 15000 20000 15000 </gml:posList> </gml:LineStringSegment> </gml:segments>/gml:Curve> </gml:curveMember> </gml:MultiCurve> </AddressReferenceSystemAxis>
Quality Measures	Address Reference System Axes Point Of Beginning Measure
Quality Notes	

2.9.3.14 Address Reference System Axis Point Of Beginning

Element Name	Address Reference System Axis Point Of Beginning
Other common names for this element	Axis Origin Point
Definition	Coordinate location of the beginning point of address numbering along an Address Reference System Axis.
Definition Source	New
Data Type	Geometry (Point) as defined in the Open Geospatial Consortium's "OpenGIS(R) Geography Markup Language (GML)" version 3.1.1 (see Appendix A for a complete citation)
Existing Standards for this Element	N/A
Domain of Values for this Element	Coordinate location of the beginning point for address numbers along an address axis.
Source of Values	Source of spatial data collection.
How Defined (e.g., locally, from standard, other)	Point location defined locally, often by ordinance, and encoded in terms of a spatial referencing system, described in file-level metadata per FGDC's Content Standard for Geospatial Metadata.
Example	**Definition** For Washington DC: The **US Capitol Building** (point of origin for North, South, and East Capitol Streets and the Capitol Mall, which divide DC into four quadrants, NW, NE, SE, and SW). Address numbers increase along those four axes as one travels away from the Capitol Building, and all other streets are addressed more or less in parallel with one of the axis streets, and every address must include a quadrant designation. **Element** : <gml:Point> <gml:pos>15000,15000</gml:pos> </gml:Point> For additional examples, please see the Complex Element: Address Reference System
Notes/Comments	The origin point for an Address Reference System Axis may be the same or may differ from the origin point for other Address Reference System Axis

	lines in the same Address Reference System.
XML Tag	<AddressReferenceSystemAxisPointOfBeginning>
XML Model	<xsd:complexType name="AddressReferenceSystemAxisPointOfBeginning_type"> <xsd:complexContent> <xsd:extension base="gml:PointType"></xsd:extension> </xsd:complexContent> </xsd:complexType>
XML Example	<AddressReferenceSystemAxisPointOfBeginning> <gml:Point> <gml:pos>15000,15000</gml:pos> </gml:Point> </AddressReferenceSystemAxisPointOfBeginning>
Quality Measures	Address Reference System Axes Point Of Beginning Measure
Quality Notes	If the Address Reference System Description specifies that the Address Reference System Axis Point Of Beginning for one Address Reference System Axis is at the intersection of another Address Reference System Axis, then use Address Reference System Axes Point Of Beginning Measure.

2.9.3.15 Address Reference System Reference Polyline

Element Name	**AddressReferenceSystemReferencePolyline**
Other common names for this element	
Definition	A street, geometric line, or other line used to measure address number assignment intervals and ranges within an Address Reference System. The Address Reference System Reference Polyline may consist of a beginning point, one or more segments of a street centerline, geographically identified line, such as a line of latitude or longitude, a land-division based line, such as a township, range, or section line, or an imaginary line constructed for the purpose of allocating address ranges and address numbers.
Definition Source	New
Data Type	Geometry (Multicurve), as defined in the Open Geospatial Consortium's "OpenGIS(R) Geography Markup Language (GML)" version 3.1.1 (see Appendix A for a complete citation)
Existing Standards	None

for this Element	
Domain of Values for this Element	Can be created locally.
Source of Values	Local jurisdiction
Attributes Associated with this Element	Address Range Side, Address Range Parity, Address Range Span, Address Range Type, Address Reference System Range Breakpoint, Address Reference System Range Breakline
How Defined	Locally
Example	Address Reference System Reference Polyline: <gml:MultiCurve> <gml:curveMember> <gml:Curve> <gml:segments> <gml:LineStringSegment> <gml:posList>1000 15000 20000 15000 </gml:posList> </gml:LineStringSegment> </gml:segments>/gml:Curve> </gml:curveMember> </gml:MultiCurve>
Notes/Comments	Theoretically, every street or other access route to an address within an Address Reference System can be construed as an Address Reference System Reference Polyline. However, in practice, where a framework of axes exists, a selection of major through streets is often used to identify breaks in address ranges, and to assist in locating the correct Address Range for a given local street. Every Complete Address Number is related to an Address Reference System Reference Polyline. 1. In an axial type Address Reference System, all Address Reference System Reference Polylines are, or could, by extension, be connected to one of the Address Reference System Axis lines. Each of the Address Reference System Reference Polylines has its Point of Beginning at the vertex of its intersection with the axis. 2. In a non-axial Address Reference System, a specific Point of Beginning is defined by the Address Reference System Authority for each Address Reference System Reference Polyline at the point where numbering for that polyline is commenced.
XML Tag	<AddressReferenceSystemReferencePolyline>
XML Model	<xsd:complexType name="AddressReferenceSystemReferencePolyline_type">

	<xsd:complexContent> <xsd:restriction base="gml:MultiCurveType"></xsd:restriction> </xsd:complexContent> </xsd:complexType>
XML Example	<AddressReferenceSystemReferencePolyline> <gml:MultiCurve> <gml:curveMember> <gml:Curve> <gml:segments> <gml:LineStringSegment> <gml:posList>1000 15000 20000 15000 </gml:posList> </gml:LineStringSegment> </gml:segments>/gml:Curve> </gml:curveMember> </gml:MultiCurve> </AddressReferenceSystemReferencePolyline>
Quality Measures	See Address Reference System Rules Measure.
Quality Notes	

2.9.3.16 Address Reference System Range Breakpoint

Element Name	**AddressReferenceSystemRangeBreakpoint**
Other common names for this element	
Definition	A point along a street or other thoroughfare within an Address Reference System where an address range beginning and/or endpoint is located.
Definition Source	New
Data Type	Geometry (Point), as defined in the Open Geospatial Consortium's "OpenGIS(R) Geography Markup Language (GML)" version 3.1.1 (see Appendix A for a complete citation)
Existing Standards for this Element	None
Domain of Values for this Element	Can be created locally.
Source of Values	Local jurisdiction

Attributes Associated with this Element	Address Range Span, Address Range Side, Address Range Parity, Address Reference System Range Breakline
How Defined	By Address Reference System rules
Example	Address Reference System Range Breakpoint: <gml:Point> <gml:pos>15000,15000</gml:pos> </gml:Point>
Notes/Comments	1. Address Reference System Range Breakpoints may occur at intersections, or they may be defined by distances, or address number increments. They represent the point at which one address range is ended, and another begins. This is usually defined at the break from one series of 100 to the next, where ranges are defined as 100-199, 200-299, etc. In an axial type Address Reference System, where a grid of streets is formed, these breakpoint almost always occur at intersections. Where an axial system is based on other geometry, such as township/range/section lines, they may occur at the point where one unit ends and the next begins (e.g. a section line, or township or range line). In a non-axial system, ranges are normally based on distance (e.g. 1000 numbers per mile), and the breakpoints may be identified by their distance from the 0 point for the road. 2. Address Reference System Range Breakpoints may be connected within the Address Reference System Extent to other points having the same value (connecting all the points that represent the breakpoint between the 100-199 Address Range and the 200-299 Address Range) to create an Address Reference System Range Breakline. Such Address Reference System Range Breaklines are useful in assignment of new addresses, and in quality review of existing references to determine whether or not they fall within the Address Range with which they are associated. For further information on Address Reference System Range Breaklines, refer to the element.
XML Tag	<AddressReferenceSystemRangeBreakpoint>
XML Model	<xsd:complexType name="AddressReferenceSystemRangeBreakpoint_type"> <xsd:complexContent> <xsd:extension base="gml:PointType"> </xsd:extension> </xsd:complexContent> </xsd:complexType>
XML Example	<AddressReferenceSystemRangeBreakpoint> <gml:Point> <gml:pos>15000,15000</gml:pos> </gml:Point> </AddressReferenceSystemRangeBreakpoint>

Quality Measures	See Address Reference System Rules Measure.
Quality Notes	

2.9.3.17 Address Reference System Range Breakline

Element Name	**AddressReferenceSystemRangeBreakline**
Other common names for this element	
Definition	A line connecting the Address Reference System Range Breakpoints with the same value within an Address Reference System
Definition Source	New
Data Type	Geometry (Multicurve), as defined in the Open Geospatial Consortium's "OpenGIS(R) Geography Markup Language (GML)" version 3.1.1 (see Appendix A for a complete citation)
Existing Standards for this Element	None
Domain of Values for this Element	Based on range values in Address Reference System.
Source of Values	Local jurisdiction
Attributes Associated with this Element	
How Defined	
Example	Address Reference System Range Breakline: <gml:MultiCurve> <gml:curveMember> <gml:Curve> <gml:segments> <gml:LineStringSegment> <gml:posList>1000 15000 20000 15000 </gml:posList> </gml:LineStringSegment> </gml:segments>/gml:Curve> </gml:curveMember>

	</gml:MultiCurve>
Notes/Comments	The Address Reference System Range Breakline provides address assignment and quality assurance personnel with a means of identifying which ranges apply within a given area of an Address Reference System. In axial (or grid) type systems, with roughly rectangular blocks, these lines should be relatively straight and parallel. However, in less regular topography, or where the street pattern is more irregular, these lines may converge or diverge. They should not cross. The lines are constructed in an axial system by connecting all of the Address Reference System Range Breakpoints that have identical values (for example those that represent the beginning of the "1200" block, and where the low values are 1200 and 1201 for left low and right low.)
XML Tag	<AddressReferenceSystemRangeBreakline>
XML Model	<xsd:complexType name="AddressReferenceSystemRangeBreakline_type"> <xsd:complexContent> <xsd:restriction base="gml:MultiCurveType"> </xsd:restriction> </xsd:complexContent> </xsd:complexType>
XML Example	<AddressReferenceSystemRangeBreakline> <gml:MultiCurve> <gml:curveMember> <gml:Curve> <gml:segments> <gml:LineStringSegment> <gml:posList>1000 15000 20000 15000 </gml:posList> </gml:LineStringSegment> </gml:segments>/gml:Curve> </gml:curveMember> </gml:MultiCurve> </AddressReferenceSystemRangeBreakline>
Quality Measures	See Address Reference System Rules Measure.
Quality Notes	

2.9.3.18 Address Reference System Range Polygon

Element Name	**AddressReferenceSystemRangePolygon**

Other common names for this element	
Definition	A polygon created by connecting the Address Reference System Range Breaklines with the same value within an Address Reference System
Definition Source	New
Data Type	Geometry (Multisurface), as defined in the Open Geospatial Consortium's "OpenGIS(R) Geography Markup Language (GML)" version 3.1.1 (see Appendix A for a complete citation)
Existing Standards for this Element	None
Domain of Values for this Element	Based on range values in Address Reference System.
Source of Values	Local jurisdiction
Attributes Associated with this Element	Address Reference System Range Breakpoint, Address Reference System Range Breakline, Address Reference System Reference Polyline
How Defined	
Example	Address Reference System Range Polygon: <gml:MultiSurface> <gml:surfaceMember> <gml:Polygon> <gml:exterior> <gml:LinearRing> <gml:posList>1000 1000 1000 25000 20000 1000 20000 25000 1000 1000</gml:posList> </gml:LinearRing> </gml:exterior> </gml:Polygon> </gml:surfaceMember> </gml:MultiSurface>
Notes/Comments	The Address Reference System Range Polygon provides address assignment and quality assurance personnel with a means of identifying which ranges apply within a given area of an Address Reference System. In axial (or grid) type systems, with roughly rectangular blocks, these polygons should create an area of a long band where all of the addresses are or should be within a given block range. However, in less regular topography, or where the street pattern is more irregular, these polygons may be less coherent. They must not overlap.

	The lines are constructed in an axial system by connecting all of the Address Reference System Range Breaklines that have identical values and extending the polygon to the Address Reference System Range Breakline with the next higher value (for example those that represent the beginning of the "1200" block, and where the low values are 1200 and 1201 for left low and right low.)
XML Tag	<AddressReferenceSystemRangePolygon>
XML Model	<xsd:complexType name="AddressReferenceSystemRangePolygon_type"> <xsd:complexContent> <xsd:restriction base="gml:MultiSurfaceType"> </xsd:restriction> </xsd:complexContent> </xsd:complexType>
XML Example	<AddressReferenceSystemRangePolygon> <gml:MultiSurface> <gml:surfaceMember> <gml:Polygon> <gml:exterior> <gml:LinearRing> <gml:posList>1000 1000 1000 25000 20000 1000 20000 25000 1000 1000</gml:posList> </gml:LinearRing> </gml:exterior> </gml:Polygon> </gml:surfaceMember> </gml:MultiSurface> </AddressReferenceSystemRangePolygon>
Quality Measures	See Address Reference System Rules Measure.
Quality Notes	

2.9.3.19 Address Reference System Reference Document Citation

Element Name	**Address Reference System Reference Document Citation**
Other common names for this element	Address Ordinance, Address Manual
Definition	A bibliographic reference to an ordinance, map, manual, or other document in which the rules governing an Address Reference System are written.
Definition Source	New

Data Type	characterString
Existing Standards for this Element	None
Domain of Values for this Element	Locally defined
Source of Values	Local
How Defined	Defined locally
Example	"Rules for the Anytown Address Reference System are found in the Anytown Address Ordinance, Chapter 15, Sections 1-29, of the Anytown Municipal Code (www.ci.anytown.na.us)"
Notes/Comments	The citation should be used initially, until all of the rules are documented within the Address Reference System Rules elements. However, once all of the rules are documented, the citation must be maintained to provide valuable source information for users.
XML Tag	<AddressReferenceSystemReferenceDocumentCitation>
XML Model	<xsd:simpleType name="AddressReferenceSystemReferenceDocumentCitation_type"> <xsd:restriction base="xsd:string" /> </xsd:simpleType>
XML Example	<AddressReferenceSystemReferenceDocumentCitation> "Rules for the Anytown Address Reference System are found in the Anytown Address Ordinance, Chapter 15, Sections 1-29, of the Anytown Municipal Code (www.ci.anytown.na.us)" </AddressReferenceSystemReferenceDocumentCitation>
Quality Measures	See Address Reference System Rules Measure.
Quality Notes	

2.9.3.20 Complex Element: Address Reference System

Element Name	**Address Reference System**
Other common names for this element	Addressing system, address numbering system, address numbering grid, house numbering system, street numbering system
Definition	An Address Reference System is a set of rules and geometries that define how addresses are assigned along thoroughfares and/or within a given area (Address Reference System Extent). At minimum, an Address Reference

	System must specify where Complete Address Number sequences begin and how Complete Address Numbers are assigned along the length of thoroughfares governed by the Address Reference System within the Address Reference System Extent. Address Reference Systems typically provide rules governing left-right parity of Complete Address Numbers, assignment of Street Names and street types, use of directionals and quadrants, and other aspects of address assignment.An Address Reference System that is based on axis lines, an Address Reference System Axis defined for each axis used to define address assignment. Each Address Reference System Axis must have an identified Address Reference System Axis Point Of Beginning. An Address Reference System is known by its Address Reference System Name (required). Additional business rules for an Address Reference System are described in the Address Reference System Rules.
Definition Source	New
Data Type	Abstract
Existing Standards for this Element	Refer to Component Elements
Domain of Values for this Element	Refer to Component Elements
Source of Values	Refer to Component Elements
How Defined (e.g., locally, from standard, other)	Refer to Component Elements
Example	Address Reference System Name: Metro City Address Grid Address Reference System Axis Point Of Beginning: <gml:Point> <gml:pos>15000,15000</gml:pos> </gml:Point> Address Reference System Axis: <gml:MultiCurve> <gml:curveMember> <gml:Curve> <gml:segments> <gml:LineStringSegment> <gml:posList>1000 15000 20000 15000 </gml:posList> </gml:LineStringSegment> </gml:segments>/gml:Curve>

```
        </gml:curveMember>
    </gml:MultiCurve>
```

Address Reference System Axis Point Of Beginning:

```
    <gml:Point>
        <gml:pos>15000,15000</gml:pos>
    </gml:Point>
```

Address Reference System Axis:

```
    <gml:MultiCurve>
        <gml:curveMember>
            <gml:Curve>
                <gml:segments>
                    <gml:LineStringSegment>
                        <gml:posList>1000 15000 20000 15000
                    </gml:posList>
                </gml:LineStringSegment>
            </gml:segments>/gml:Curve>
        </gml:curveMember>
    </gml:MultiCurve>
```

Address Reference System Extent:

```
    <gml:MultiSurface>
        <gml:surfaceMember>
            <gml:Polygon>
                <gml:exterior>
                    <gml:LinearRing>
                        <gml:posList>1000 1000 1000 25000
                        20000 1000 20000 25000 1000
                        1000</gml:posList>
                    </gml:LinearRing>
                </gml:exterior>
            </gml:Polygon>
        </gml:surfaceMember>
    </gml:MultiSurface>
```

Address Reference System Rules: Written information about parity, street naming conventions, numbering intervals, grids, and other business rules. (Contains elements including Address Reference System Block Rules,

	Address Reference System Numbering Rules, Address Reference System Street Naming Rules, Address Reference System Street Type Directional And Modifier Rules, Address Reference System Place Name State Country And ZIP Code Rules Address Reference System Authority: Name of agency (municipality, county, other) that has authority over the scheme's business rules, extent and other parameters.
Notes/Comments	1. Address Reference System Extents may overlap. 2. There are three broad types of Address Reference Systems: Axial, linear non-axial and area based. * **Axial** The Address Reference System is based on streets or geometric lines whichform the basis for address numbering. The axes are often oriented more or less at 90 degrees to each other to define quadrants or directions. The grid may be deformed by topography, rivers, rail lines or other features. This is by far the most common type in the United States; Chicago is but one of many clear examples. * **Linear Non-axial.** Each thoroughfare has its own beginning point for Complete Address Numbers, independent of the other thoroughfares in the Address Reference System. This is common, for example, in rural areas where the road network is sparse and street segments are long. This term may also apply to places where the address numbers are not based on thoroughfares at all. * **Area-based.** An Address Reference System may not be based on street geometry, but number assignment is done according to chronology (when a structure was addressed), or parcel or lot numbers. 3. A jurisdiction may have more than one addressing scheme within its area, and its Address Reference System(s) may change over time. Occasionally addresses from different schemes are intermingled along the same block face, which complicates the assignment of an address range to that block face. This may be the result of annexation of developed properties with existing addresses from one jurisdiction to another.
XML Tag	<AddressReferenceSystem>
XML Model	<xsd:complexType name="AddressReferenceSystem" > <xsd:sequence> <xsd:element name="AddressReferenceSystemId" type="addr_type:AddressReferenceSystemId_type" maxOccurs="1" minOccurs="1"/> <xsd:element name="AddressReferenceSystemName" type="addr_type:AddressReferenceSystemName_type" maxOccurs="1" minOccurs="1"/> <xsd:element name="AddressReferenceSystemAuthority" type="addr_type:AddressReferenceSystemAuthority_type" maxOccurs="1" minOccurs="0"/> <xsd:element name="AddressReferenceSystemExtent"

	type="addr_type:AddressReferenceSystemExtent_type" maxOccurs="1" minOccurs="0"/> <xsd:element name="AddressReferenceSystemType" type="addr_type:AddressReferenceSystemType_type" maxOccurs="1" minOccurs="0"/> <xsd:element name="AddressReferenceSystemRules" type="addr_type:AddressReferenceSystemRules_type" maxOccurs="1" minOccurs="0"/> <xsd:element name="AddressReferenceSystemAxis" type="addr_type:AddressReferenceSystemAxis_type" maxOccurs="1" minOccurs="0"/> <xsd:element name="AddressReferenceSystemAxisPointOfBeginning" type="addr_type:AddressReferenceSystemAxisPointOfBeginning_type" maxOccurs="1" minOccurs="0"/> <xsd:element name="AddressReferenceSystemReferencePolyline" type="addr_type:AddressReferenceSystemReferencePolyline_type" maxOccurs="unbounded" minOccurs="0"/> <xsd:element name="AddressReferenceSystemRangeBreakpoint" type="addr_type:AddressReferenceSystemRangeBreakpoint_type" maxOccurs="1" minOccurs="0"/> <xsd:element name="AddressReferenceSystemRangeBreakline" type="addr_type:AddressReferenceSystemRangeBreakline_type" maxOccurs="unbounded" minOccurs="0"/> <xsd:element name="AddressReferenceSystemReferenceDocumentCitation" type="addr_type:AddressReferenceSystemReferenceDocumentCitation_type" maxOccurs="unbounded" minOccurs="0"/> </xsd:sequence> </xsd:complexType>
XML Example	<AddressReferenceStstem> <AddressReferenceSystemId>MCAG Unified</AddressReferenceSystemId> <AddressReferenceSystemName>Metro City Address Grid</AddressReferenceSystemName> <AddressReferenceSystemType>Grid</AddressReferenceSystemType> </AddressReferenceSystem>
Quality Measures	Address Reference System Rules Measure
Quality Notes	

2.10 Address Attributes

2.10.1 Address ID

2.10.1.1 Address ID

Element Name	Address ID
Other common names for this element	
Definition	The unique identification number assigned to an address by the addressing authority
Definition Source	New
Data Type	characterString
Existing Standards for this Element	None
Domain of Values for this Element	No
Source of Values	Primary key, issued locally
How Defined (e.g., locally, from standard, other)	Locally
Example:	Integer ID: **1243286** UUID: **550e8400-e29b-11d4-a716-446655440000**
Notes/Comments	1. The Address ID is a required element of an address data record. The ID must be unique for each address assigned by an Address Authority. The Address ID may be either a locally generated unique ID, or it may be a Universally Unique ID (UUID) which is machine-generated within the database environment. 2. IDs are almost always integers, and integer ID's are much easier to manage. However, some ID schemes use hyphens, leading zeros, or other non-integer characters, so the standard also accommodates alphanumeric IDs. **Notes and Reference Information on UUID** 1. A UUID is presented as a 16-byte (128-bit) number written in hexadecimal form computed according to a UUID algorithm. At least five algorithms have been developed. 2. UUIDs are documented in two standards, ITU-T X.667 and IETF RFC

	4122 (see Appendix A for complete references). The two standards are technically consistent. 3. The standard provides for a UUID as a means to identify an address while it is passed from the originating source through a chain of intermediaries to the end-user. The need arises because there exists within the United States no central coordinating body to identify and register addresses. There is not even a registry of the authorities empowered to create addresses, nor is one likely to be created. 4. "The intent of UUIDs is to enable distributed systems to uniquely identify information without significant central coordination. Thus, anyone can create a UUID and use it to identify something with reasonable confidence that the identifier will never be unintentionally used by anyone for anything else. Information labelled with UUIDs can therefore be later combined into a single database without need to resolve name conflicts." (quoted from Wikipedia, "Universally Unique Identifier", as posted 6 September 2009 at: http://en.wikipedia.org/wiki/Universally_Unique_Identifier)
XML Tag	<AddressID>
XML Model	<xsd:simpleType name="AddressId_type"> <xsd:restriction base="xsd:string"> <xsd:pattern value='.*' /> </xsd:restriction> </xsd:simpleType>
XML Example	<AddressID>550e8400-e29b-11d4-a716-446655440000</AddressID>
Quality Measures	Uniqueness Measure
Quality Notes	

1146

1147 **2.10.1.2 Address Authority**

Element Name	**Address Authority**
Other common names for this element	
Definition	The name of the authority (e.g., municipality, county) that created or has jurisdiction over the creation, alteration, or retirement of an address
Definition Source	New
Data Type	characterString
Existing Standards for this Element	None

Domain of Values for this Element	None
Source of Values	None
How Defined (e.g., locally, from standard, other)	Locally
Example	1. Florence County, SC 2. City of Boulder, CO 3. University of Georgia, Athens, GA (for addresses within the campus) 4. Hartsfield-Jackson International Airport, Clayton County, GA (for addresses within the airport) 5. Bolling Air Force Base, Washington, DC (for addresses within the base)
Notes/Comments	1. The Address Authority is the agency responsible for assigning and administering addresses in a given area. 2. The Address Authority is also responsible for providing unique Address IDs for the addresses it administers. Thus the Address Authority name plus the ID in combination are likely to be unique nationwide. 3. The Address Authority may or may not be the same as the municipal or postal jurisdiction noted for the address. In a given area, there may be multiple authorities, a single authority or no known authority with jurisdiction over address assignment. For example, a state agency may be the Address Authority for a university campus within the municipal boundaries of a city. 4. Contact information for Address Authority will be found in the dataset metadata.
XML Tag	<AddressAuthority>
XML Model	<xsd:simpleType name="AddressAuthority_type"> <xsd:restriction base="xsd:string"> <xsd:pattern value='.*' /> </xsd:restriction> </xsd:simpleType>
XML Example	<AddressAuthority>City of Boulder, CO</AddressAuthority> <AddressAuthority>University of Georgia, Athens, GA</AddressAuthority>
Quality Measures	Tabular Domain Measure
Quality Notes	

1148

1149 ### 2.10.1.3 Related Address ID

Element Name	Related Address ID
Other common names for this element	
Definition	The identifier of an address that is related to the identifier of another address.
Definition Source	New
Data Type	characterString
Existing Standards for this Element	None
Domain of Values for this Element	None
Source of Values	None
How Defined (e.g., locally, from standard, other)	Locally
Examples:	See examples under Address Relation Type
Notes/Comments	1. The Related Address ID is used to relate one address identifier to another address identifier. 2. In database terms, the Related Address ID is linked to the Address ID in a linking table or relationship table. Logically, a Related Address ID cannot exist unless it is associated with an Address ID. 3. In some cases, the Related Address ID designates an alternate address at the same location, for example, a Landmark Address associated with a Numbered Thoroughfare Address, or an official address with its alias, or a retired address in the same location as an active address. 4. In other cases, the Related Address ID designates an address at a different location, for example, the address of a property owner (if the owner does not live on the property), or a property's tax billing address (if it is sent to the mortgage holder). 5. The Address Relation Type attribute can be used to record **how** the address identified by the Related Address ID is related to the address identified by the Address ID. (See Address Relation Type example and notes for additional discussion of Related Address ID.)
XML Tag	<RelatedAddressID>
XML Model	<xsd:complexType name="RelatedAddressID_type"> <xsd:simpleContent> <xsd:extension base="addr_type:AddressID_type"> <xsd:attribute name="AddressRelationType"

	type="addr_type:AddressRelationType_type" /> </xsd:extension> </xsd:simpleContent> </xsd:complexType>
XML Example	<RelatedAddressID Address Relation Type="Historical Predecessor" >250</RelatedAddressID>
Quality Measures	Repeated Element Uniqueness Measure Related Not Null Measure Tabular Domain Measure
Quality Notes	

2.10.1.4 Address Relation Type

Element Name	**Address Relation Type**
Other common names for this element	
Definition	The manner in which an address identified by a Related Address ID is related to an address identified by an Address ID.
Definition Source	New
Data Type	characterString
Required Element	None.
Existing Standards for this Element	None
Domain of Values for this Element	May be created locally to standardize terms used to describe relationships.
How Defined (e.g., locally, from standard, other)	New
Example	1. **123 Main St** (Address ID = 1000) is also known as the **"Grand Old Office Building"** (a landmark name, Address ID = 5000). Then for: Related Address ID = 5000, Address ID = 1000, Address Relation Type = **Landmark Name Alias** Related Address ID = 1000, Address ID = 5000, Address Relation Type = **Official Street Address**

	2. Tax bills for 123 Main St (Address ID = 1000) should be sent to **PO Box 150080, Omaha, NE 68153** (Address ID = 8000). Correspondence for the owner should be sent to **108 East Burnside Street, Portland, OR 97214.** (Address ID = 10267). Then for: Related Address ID = 8000, Address ID = 1000, Address Relation Type = **Tax Billing** Related Address ID = 10267, Address ID = 1000, Address Relation Type = **Owner Mailing** 3. 123 Main Street was created years ago when **101 Main Street** (Address ID = 250) was subdivided into several properties. Then for: Related Address ID = 250, Address ID = 1000, Address Relation Type = **Historical Predecessor** 4. This particular part of Main Street is part of **State Route 88.** 123 Main Street (Address ID = 1000) is the official address, but 123 State Route 88 (Address ID = 8943) is also recognized. Then for: Related Address ID = 8943, Address ID = 1000, Address Relation Type = **Official Alias Address** Related Address ID = 1000, Address ID = 8943, Address Relation Type = **Official Address** 5. A large building occupies an entire square block in a downtown area. It has a main entrance to its public lobby at 123 Main Street. However, its loading dock, mail and goods receiving entry, and trash pickup location are on the "back" of the building, which faces Elm Street, and is given the address of 222 Elm Street. In this instance, the main entrance at 123 Main Street has Address ID = 456, while the service entrance at 222 Elm Street has Address ID = 789. The Relationship would be: Address ID = 456, Related Address ID = 789, Address Relation Type = Service Entrance, and conversely Address ID = 789, Related Address ID = 456, Address Relation Type = Official Street Address.
Notes/Comments	1. This element describes how two addresses, identified by their Related Address ID and Address ID respectively, are related. Relationships may be defined and described in any way, according to the needs of the user. To maximize efficiency and clarity, users should establish a limited, standard set of descriptors that meet local needs. 2. To minimize ambiguity, the descriptors should state how the Related Address ID is related to the Address ID, not the other way around. 3. To minimize clutter, short connector words such as "is", "are", "for", "of", etc. may be omitted from the descriptors if the meaning is otherwise clear. 4. Examples 1, 3, and 4 above show how Related Address ID can be used to link an address to its alias addresses or to its historical predecessor address. 5. Example 1 above shows that two addresses must have reciprocal relations, each being designated by the Address ID in one case and the Related Address

	ID in the other. 6. Example 5 shows how one feature (such as a large building) may have more than one address, each with a different purpose (official street address vs service entrance). 7. Example 2 above shows that Related Address ID may designate an address that is outside the control of, and perhaps distant from, the Address Authority that created the address it is related to. It is common, for example, for owners to live in different states from properties they own, or for tax bills to be sent to out-of-state mortgage service addresses.
XML Tag	AddressRelationType
XML Model	<xsd:simpleType name="AddressRelationType_type"> <xsd:restriction base="xsd:string"> <xsd:pattern value='.*' /> </xsd:restriction> </xsd:simpleType>
XML Example	<RelatedAddressID **AddressRelationType="Historical Predecessor"** >250</RelatedAddressID>
Quality Measures	Tabular Domain Measure
Quality Notes	

2.10.2 Address Coordinates

2.10.2.1 Address X Coordinate

Element Name	**Address X Coordinate**
Other common names for this element	
Definition	The X coordinate of the address location.
Definition Source	New
Data Type	Real
Existing Standards for this Element	Yes

Domain of Values for this Element	Spatial extent of the jurisdiction(s).
Source of Values	Source of spatial data collection.
How Defined (e.g., locally, from standard, other)	By reference to a coordinate reference system (see note below).
Example	750908.0469
Notes/Comments	Address X Coordinate values can be interpreted only if their coordinate system, datum, units of measure, and any other coordinate reference system parameters are provided. The parameters can be documented in the dataset metadata, per FGDC's Content Standard for Digital Geospatial Metadata, or by inclusion of the Address Coordinate Reference System Authority and Address Coordinate Reference System ID in each address record. See Address Coordinate Reference System Authority and Address Coordinate Reference System ID for more information.
XML Tag	<AddressXCoordinate>
XML Model	<xsd:simpleType name="AddressXCoordinate_type"> <xsd:restriction base="xsd:double"> </xsd:restriction> </xsd:simpleType>
XML Example	<AddressXCoordinate>750908.0469</AddressXCoordinate>
Quality Measures	XY Coordinate Completeness Measure XY Coordinate Spatial Measure
Quality Notes	

2.10.2.2 Address Y Coordinate

Element Name	**Address Y Coordinate**
Other common names for this element	
Definition	The Y coordinate of the address location.
Definition Source	New
Data Type	Real
Existing Standards for this Element	Yes

Domain of Values for this Element	Spatial extent of the jurisdiction(s).
Source of Values	Source of spatial data collection.
How Defined (e.g., locally, from standard, other)	By reference to a coordinate reference system.
Example	3740623.0628
Notes/Comments	Address Y Coordinate values can be interpreted only if their coordinate system, datum, units of measure, and any other coordinate reference system parameters are provided. The parameters can be documented in the dataset metadata, per FGDC's Content Standard for Digital Geospatial Metadata, or by inclusion of the Address Coordinate Reference System Authority and Address Coordinate Reference System ID in each address record. See Address Coordinate Reference System Authority and Address Coordinate Reference System ID for more information.
XML Tag	<AddressYCoordinate>
XML Model	<xsd:simpleType name="AddressYCoordinate_type"> <xsd:restriction base="xsd:double"> </xsd:restriction> </xsd:simpleType>
XML Example	<AddressYCoordinate>3740623.0628 </AddressYCoordinate>
Quality Measures	XY Coordinate Completeness Measure XY Coordinate Spatial Measure
Quality Notes	

2.10.2.3 Address Longitude

Element Name	Address Longitude
Other common names for this element	
Definition	The longitude of the address location, in decimal degrees.
Definition Source	New
Data Type	Real
Existing Standards for this Element	Adapted from FGDC, "Content Standard for Digital Geospatial Metadata (CSDGM)", which refers to the following standard: ANSI INCITS 61-1986

	(R2002), "Representation of Geographic Point Locations for Information Interchange".
Domain of Values for this Element	Spatial extent of the jurisdiction(s).
Source of Values	Source of spatial data collection.
How Defined (e.g., locally, from standard, other)	By reference to a coordinate reference system.
Example	-84.29049105
Notes/Comments	Address Longitude values can be interpreted only if their coordinate system, datum, units of measure, and any other coordinate reference system parameters are provided. The parameters can be documented in the dataset metadata, per FGDC's Content Standard for Digital Geospatial Metadata, or by inclusion of the Address Coordinate Reference System Authority and Address Coordinate Reference System ID in each address record. See Address Coordinate Reference System Authority and Address Coordinate Reference System ID for more information.
XML Tag	<AddressLongititude>
XML Model	<xsd:simpleType name="AddressLongitude_type"> <xsd:restriction base="xsd:double"> </xsd:restriction> </xsd:simpleType>
XML Example	<AddressLongitude>-84.29049105</AddressLongitude>
Quality Measures	XY Coordinate Completeness Measure XY Coordinate Spatial Measure
Quality Notes	

2.10.2.4 Address Latitude

Element Name	**Address Latitude**
Other common names for this element	
Definition	The latitude of the address location, in decimal degrees.
Definition Source	New
Data Type	Real

Existing Standards for this Element	Adapted from FGDC, "Content Standard for Digital Geospatial Metadata (CSDGM)", which refers to the following standard: ANSI INCITS 61-1986 (R2002), "Representation of Geographic Point Locations for Information Interchange".
Domain of Values for this Element	Spatial extent of the jurisdiction(s).
Source of Values	Source of spatial data collection.
How Defined (e.g., locally, from standard, other)	By reference to a coordinate reference system.
Example	33.77603207
Notes/Comments	Address Latitude values can be interpreted only if their coordinate system, datum, units of measure, and any other coordinate reference system parameters are provided. The parameters can be documented in the dataset metadata, per FGDC's Content Standard for Digital Geospatial Metadata, or by inclusion of the Address Coordinate Reference System Authority and Address Coordinate Reference System ID in each address record. See Address Coordinate Reference System Authority and Address Coordinate Reference System ID for more information.
XML Tag	<AddressLatitude>
XML Model	<xsd:simpleType name="AddressLatitude_type"> <xsd:restriction base="xsd:double"> </xsd:restriction> </xsd:simpleType>
XML Example	<AddressLatitude>33.77603207</AddressLatitude>
Quality Measures	XY Coordinate Completeness Measure XY Coordinate Spatial Measure
Quality Notes	

2.10.2.5 US National Grid Coordinate

Element Name	**US National Grid Coordinate**
Other common names for this element	USNG Coordinate
Definition	The USNG is an alphanumeric point reference system that overlays the

	Universal Transverse Mercator (UTM) numerical coordinate system. A USNG coordinate consists of **three parts**, the: 1. **Grid Zone Designation (GZD)** for worldwide unique geoaddresses (two digits plus one letter, developed from the UTM system). 2. **100,000-meter Square Identification** for regional areas (two letters). 3. **Grid Coordinates** for local areas (always an even number of digits between 2 and 10 depending upon precision).
Definition Source	Adapted from US National Grid, FDGC-STD-011-2001, Section 3.3 Quoted from: Tom Terry, "The United States National Grid." *Professional Surveyor Magazine*. Oct. 2004, p. 12.
Data Type	characterString
Required Element	No
Existing Standards for this Element	US National Grid, FGDC-STD-011-2001.
Domain of Values for this Element	No
Source of Values	
How Defined (from standard, other)	As prescribed in FGDC-STD-011-2001.
Example	**18SUJ2348306479** or **18S UJ 23483 06479** **18S** – Identifies a **GZD** **18S UJ** – Identifies a specific **100,000-meter square** in the specified GZD **18S UJ** 2 0 - Locates a point with a precision of 10 km **18S UJ** 23 06 - Locates a point with a precision of 1 km **18S UJ** 234 064 - Locates a point with a precision of 100 meters **18S UJ** 2348 0647 - Locates a point with a precision of 10 meters **18S UJ** 23483 06479 - Locates a point with a precision of 1 meter
Notes/Comments	1. USNG basic coordinate values and numbering are identical to Universal Transverse Mercator (UTM) coordinate values over all areas of the United States including outlying territories and possessions. The USNG is based on universally defined coordinate and grid systems and can, therefore, be easily extended for use world-wide as a universal grid reference system. 2. USNG coordinates shall be identical to the Military Grid Reference System (MGRS) numbering scheme over all areas of the United States including outlying territories and possessions. 3. While their coordinates are the same, the key difference between MGRS and USNG is in the organization of their 100,000-m Square Identification schemes. MGRS uses two 100,000-m Square Identification lettering schemes, depending on which datum is used, while USNG uses only the single scheme associated with NAD 83/WGS 84. When USNG values are referenced to NAD

	83/WGS 84, USNG and MGRS values are identical and MGRS can be used as a surrogate when software does not yet support USNG. 4. The USNG is not intended for surveying, nor is it intended to replace the coordinate reference system used for digital mapping by local authorities (typically, local or state plane coordinate systems). USNG provides a nationally consistent presentation format and grid for public safety, general public, and commercial activities that is user-friendly in both digital and hardcopy products. USNG values enable use of geocoded address point data with low cost consumer grade GPS receivers and properly gridded maps. 5. USNG provides a flexible numbering scheme to accommodate variable precision from tens of kilometers to one meter or higher.
XML Tag	<USNationalGridCoordinate>
XML Model	<xsd:simpleType name="LocationUSNG_type"> </xsd:annotation> <xsd:restriction base="xsd:string"> <xsd:pattern value='.*' /> </xsd:restriction> </xsd:simpleType>
XML Example	<USNationalGridCoordinate>18SUJ2348306479</USNationalGridCoordinate> <USNationalGridCoordinate>18S UJ 23483 06479</USNationalGridCoordinate>
Quality Measures	USNG Coordinate Spatial Measure
Quality Notes	There are a variety of ways to check USNG coordinate values. Due to the complexity of the USNG standard entire working functions are offered as examples, rather than pseudocode: coord2usng, converting Universal Transverse Mercator (UTM) coordinates to USNG, and usng2coord, converting USNG to UTM. 1. The coord2usng function requires both UTM and longitude latitude coordinates, and calculates the UTM zone on the fly. This method was chosen due to common confusion about zone numbers. There are a variety of other ways to structure the conversion. 2. Usng2coord requires only USNG, and is fairly straightforward.

2.10.2.6 Address Elevation

Element Name	**Address Elevation**

Other common names for this element	Altitude, height, Z-coordinate
Definition	Distance of the address in specified units above or below a vertical datum, as defined by a specified coordinate reference system.
Definition Source	New
Data Type	Real
Existing Standards for this Element	Yes
Domain of Values for this Element	None
Source of Values	Locally defined.
How Defined (e.g., locally, from standard, other)	By reference to a coordinate reference system.
Examples	**1023.0** (elevation in specified units above a specified vertical datum)
Notes/Comments	Address Elevation values can be interpreted only if their units of measure, vertical datum, and any other coordinate reference system parameters are be provided. The parameters can be documented in the dataset metadata, per FGDC's Content Standard for Digital Geospatial Metadata, or by inclusion of the Address Coordinate Reference System Authority and Address Coordinate Reference System ID in each address record. See Address Coordinate Reference System Authority and Address Coordinate Reference System ID for more information.
XML Tag	<AddressElevation>
XML Model	<xsd:simpleType name="AddressElevation_type"> <xsd:restriction base="xsd:double"> <xsd:pattern value='.*' /> </xsd:restriction> </xsd:simpleType>
XML Example	<AddressElevation>1023.0</AddressElevation>
Quality Measures	Address Elevation Measure
Quality Notes	

1167 ### 2.10.2.7 Address Coordinate Reference System ID

Element Name	Address Coordinate Reference System ID
Other common names for this element	Spatial Reference ID (SRID)
Definition	A name or number which, along with the Address Coordinate Reference System Authority, identifies the coordinate reference system to which Address X Coordinate and Address Y Coordinate. Address Latitude and Address Longitude, US National Grid Coordinate, or Address Elevation values are referenced.
Definition Source	New
Data Type	Integer
Existing Standards for this Element	Yes
Domain of Values for this Element	May be defined by the Address Coordinate Reference System Authority.
Source of Values	Address Coordinate Reference System Authority.
How Defined (e.g., locally, from standard, other)	Address Coordinate Reference System Authority.
Example	EPSG **2893** Wisconsin State Cartographer's Office, **"Dane County Coordinate System"**
Notes/Comments	1. A coordinate location cannot be determined without knowledge of the coordinate reference system (CRS) by which the specific coordinate values are defined. The CRS itself is defined by a set of geodetic parameters. The parameters vary according to the type of CRS, but may include, for example, datum, unit of measure, or projection. When the CRS and its geodetic parameters are known, the address location can be determined unambiguously from its coordinates. 2. The Address Coordinate Reference System ID, combined with the Address Coordinate Reference System Authority in the complex element Address Coordinate Reference System, identifies the CRS to which the Address X Coordinate and Address Y Coordinate, Address Latitude, Address Longitude, US National Grid Coordinate, or Address Elevation values are referenced. The Address Coordinate Reference System Authority and the Address Coordinate Reference System ID should refer interested persons to an authoritative source where the geodetic parameters can be found, or else complete reference information should be provided in the file-level metadata. 3. See Address Coordinate Reference System Authority for additional pertinent notes.

XML Model	<xsd:simpleType name="AddressCoordinateReferenceSystemID_type"> <xsd:restriction base="xsd:integer" /> </xsd:simpleType>
XML Example	<AddressCoordinateReferenceSystem> <AddressCoordinateReferenceSystemAuthority>EPSG Geodetic Parameter Dataset </AddressCoordinateReferenceSystemAuthority> **<AddressCoordinateReferenceSystemID>2893</AddressCoordinateReferenceSystemID>** </AddressCoordinateReferenceSystem>
Quality Measures	Tabular Domain Measure
Quality Notes	

2.10.2.8 Address Coordinate Reference System Authority

Element Name	**Address Coordinate Reference System Authority**
Other common names for this element	Spatial Reference System Authority
Definition	The Authority that assigns the unique Address Coordinate Reference System ID (number or name) to the Address Coordinate Reference System to which the Address X Coordinate and Address Y Coordinate, Address Latitude and Address Longitude, US National Grid Coordinate, or Address Elevation are referenced.
Definition Source	New.
Data Type	characterString
Existing Standards for this Element	No
Domain of Values for this Element	None
Source of Values	New
How Defined (e.g., locally, from standard, other)	Authority name defined by creator of base map
Examples	1. **EPSG Geodetic Parameter Dataset** 2. **Wisconsin State Cartographer's Office**
Notes/Comments	1. Coordinate values specify a location by reference to a grid, spheroid, or

	geoid. A coordinate location cannot be determined without knowledge of the coordinate reference system (CRS) by which the specific coordinate values are defined. The CRS itself is defined by a set of geodetic parameters. The parameters vary according to the type of CRS, but may include, for example, datum, unit of measure, or projection. When the CRS and its geodetic parameters are known, the address location can be determined unambiguously from its coordinates. 2. The Address Coordinate Reference System Authority, combined with the Address Coordinate Reference System ID in the complex element Address Coordinate Reference System, identifies the CRS to which the Address X Coordinate and Address Y Coordinate, Address Latitude, Address Longitude, US National Grid Coordinate, or Address Elevation values are referenced. The Address Coordinate Reference System Authority and the Address Coordinate Reference System ID should refer interested persons to an authoritative source where the geodetic parameters can be found, or else complete reference information should be provided in the file-level metadata. 3. The EPSG Geodetic Parameter Dataset, maintained and published by the Geodesy Subcommittee of the International Association of Oil and Gas Producers (OGP), is an extensive, authoritative, and public compilation of CRS, the geodetic parameters that define them, and conversion and transformation operations that allow coordinates to be changed from one CRS to another. Within the EPSG dataset, each CRS is identified by a COORD_REF_SYS_CODE. Although it is extensive, the EPSG dataset is not exhaustive. The OGC states, "The geographic coverage of the data is worldwide, but it is stressed that the dataset does not and cannot record all possible geodetic parameters in use around the world." 4. For examples of CRS not included in the EPSG dataset, see the Wisconsin State Cartographers Office's "Wisconsin Coordinate Systems." This publication gives the projection parameters and associated information for the Wisconsin Coordinate Reference Systems used by each of Wisconsin's 72 counties, identified by county name. The EPSG Dataset includes parameters for various versions of the Wisconsn State Plane Coordinate System, but not for each county CRS. 5. If all coordinate values in a dataset are referenced to the same CRS, the CRS should be described in the dataset-level metadata per FGDC's Content Standard for Digital Geospatial Metadata. The Address Coordinate Reference System Authority and Address Coordinate Reference System ID may then be omitted from the individual address records. 6. If the address data set includes Address X Coordinate and Address Y Coordinate, Address Latitude, Address Longitude, or Address Elevation values based on more than one CRS, each address record should include the Address Coordinate Reference System Authority and Address Coordinate Reference System ID to show which system applies to each value. 7. EPSG Guidance Note 7-1 ("Using the EPSG Geodetic Prameter Dataset") provides a clear, concise explanation of the concepts underlying coordinate

	reference systems, and of the EPSG dataset and its use. EPSG Guidance Note 7-1 can be found at www.epsg.org under "Guidance notes" or "Geodetic dataset". 8. The Wisconsin State Cartographers Office publication also includes a concise, clear explanation of the concepts underlying CRS.
XML Tag	<AddressCoordinateReferenceSystemAuthority>
XML Model	<xsd:simpleType name="AddressCoordinateReferenceSystemAuthority_type"> <xsd:restriction base="xsd:string" /> </xsd:simpleType>
XML Example	<AddressCoordinateReferenceSystem> **<AddressCoordinateReferenceSystemAuthority>EPSG Geodetic aParameter Dataset </AddressCoordinateReferenceSystemAuthority>** <AddressCoordinateReferenceSystemID>2893</AddressCoordinateReferenceSystemID> </AddressCoordinateReferenceSystem>
Quality Measure	Tabular Domain Measure
Quality Notes	

2.10.2.9 Complex Element: Address Coordinate Reference System

Element Name	**Address Coordinate Reference System**
Other common names for this element	
Definition	{ Address Coordinate Reference System Authority* } + { Address Coordinate Reference System ID* }
Data Type	characterString
Existing Standards for this Element	No
Domain of Values for this Element	No
Source of Values	
How Defined (e.g., locally, from standard, other)	From base mapping

Example	EPSG:12349
Notes/Comments	The Address Coordinate Reference System combines the Address Coordinate Reference System Authority and the Address Coordinate Reference System ID. Together they form a unique identifier for any coordinate reference system that might define the coordinate values associated with an address, whether an Address X Coordinate, Address Y Coordinate, Address Latitude, Address Longitude, or Address Elevation
XML Tag	<AddressCoordinateReferenceSystem>
XML Model	<xsd:complexType name="AddressCoordinateReferenceSystem_type"> <xsd:sequence> <xsd:element name="AddressCoordinateReferenceSystemAuthority" type="AddressCoordinateReferenceSystemAuthority_type" /> <xsd:element name="AddressCoordinateReferenceSystemID" type="AddressCoordinateReferenceSystemID_type"></xsd:element> </xsd:sequence> </xsd:complexType>
XML Example	<AddressCoordinateReferenceSystem> <AddressCoordinateReferenceSystemAuthority>EPSG Geodetic Parameter Dataset </AddressCoordinateReferenceSystemAuthority> <AddressCoordinateReferenceSystemID>2893</AddressCoordinateReferenceSystemID> </AddressCoordinateReferenceSystem>
QualityMeasures	Pattern Sequence Measure
QualityNotes	

2.10.3 Address Parcel IDs

2.10.3.1 Address Parcel Identifier Source

Element Name	**Address Parcel Identifier Source**
Other common names for this element	
Definition	The permanent identifier for the agency, organization, or jurisdiction that assigns and maintains the Address Parcel Identifier. Definition source: FGDC, May 2008. "Geographic Information Framework Data Content Standard Part 1: Cadastral." Section 4.7.

Data Type	characterString
Existing Standards for this Element	None.
Domain of Values for this Element	None.
Source of Values	None.
How Defined (e.g., locally, from standard, other)	By local government (typically county government) law or administrative procedure, as governed by state law.
Example	Chester County (PA) Tax Assessment Department Bureau of Land Records Wake County (NC) Revenue Department Delaware County (OH) Auditor's Office
Notes/Comments	1. The Address Parcel Identifier Source designates the agency, organization or jurisdiction that assigns and maintains the Address Parcel Identifier. 2. If known, give the full name of the agency (department, office, etc.) rather than just the jurisdiction name. 3. In giving a jurisdiction name, if possible follow known naming standards, such as the ANSI (formerly FIPS) names or codes for states and counties, or GNIS names or codes for minor civil divisions, populated places, and other features.
XML Tag	<AddressParcelIdentifierSource>
XML Model	<xsd:simpleType name="AddressParcelIdentifierSource_type"> <xsd:restriction base="xsd:string"> <xsd:pattern value='.*' /> </xsd:restriction> </xsd:simpleType>
XML Example	<AddressParcelIdentifierSource>Wake County (NC) Revenue Department </AddressParcelIdentifierSource>
Quality Measures	Tabular Domain Measure
Quality Notes	

2.10.3.2 Address Parcel Identifier

Element Name	Address Parcel Identifier
Other common names for this element	Parcel Identifier Number, PIN number

Definition	The primary permanent identifier, as defined by the Address Parcel Identifier Source, for a parcel that includes the land or feature identified by an address. A parcel is "a single cadastral unit, which is the spatial extent of the past, present, and future rights and interests in real property." Definition source for "parcel identifier": Adapted from FGDC, May 2008. "Geographic Information Framework Data Content Standard Part 1: Cadastral." Section 4.2. Definition source for "parcel": FGDC, May 2008. "Cadastral Data Content Standard for the National Spatial Data Infrastructure." Vesion 1.4 – Fourth Revision. p. 45. (Part 3.2 "Parcel)
Data Type	characterString
Existing Standards for this Element	Determined by local ordinance or procedure, or in some cases by state law.
Domain of Values for this Element	Determined by local procedure.
Source of Values	Address Parcel Identifier Source
How Defined (e.g., locally, from standard, other)	By local procedure, as it may be governed by local ordinance or state law.
Example	**5142301020000** (= the address identifies the land or a feature within parcel 5142301020000) **07660254993-000** (= the address identifies the land or a feature within parcel 07660254993-000) **176-N-075** (= the address identifies the land or a feature within parcel 176-N-075)
Notes/Comments	1. Parcels and addresses are created independently of each other. Some addresses locate features on one parcel only, and some addresses locate features that encompass multiple parcels. There are addresses that locate features that are not on tax parcels, but that are on ownership parcels such as federally-managed lands or public rights of way. Conversely there are parcels that have no address at all, parcels that have one address, and parcels that have many addresses (e.g. large parcels that front on or encompass more than one thoroughfare). 2. Thus no specific address-parcel relationship can be assumed. Addresses and parcels should be treated as independent of each other, and the relationship between should be treated, in relational database terms, as a many-to-many relationship. By providing an Address Parcel Identifier and an Address Parcel Identifier Source, the address standard provides a means to link an address with any number of parcels, and to link a parcel with any number of addresses. 3. The Address Parcel Identifier corresponds to the Parcel ID element in the

	Cadastral Standard. The Parcel ID is the primary key that identifies each record or occurrence in the Parcel entity. That, plus the Address Parcel Identifier Source, are the only parcel elements included or needed within the address standard. All other parcel elements are defined within the Cadastral Standard and need not be repeated here.
XML Tag	<AddressParcelIdentifier>
XML Model	<xsd:simpleType name="AddressParcelIdentifier_type"> <xsd:restriction base="xsd:string"> <xsd:pattern value='.*' /> </xsd:restriction> </xsd:simpleType>
XML Example	<AddressParcelIdentifier>07660254993-000</AddressParcelIdentifier>
Quality Measures	Uniqueness Measure Pattern Sequence Measure
Quality Notes	

2.10.4 Address Transportation Feature IDs

2.10.4.1 Address Transportation System Name

Element Name	**Address Transportation System Name**
Other common names for this element	Street centerline file, road network file, street network file, centerline network file
Definition	The name of the transportation base model to which the address is related.
Data Type	characterString
Existing Standards for this Element	1. There are no standards specifically for naming specific transportation base models. 2. The content requirements for transportation base models are set forth in: U.S. Federal Geographic Data Committee, "Framework Data Content Standard Part 7: Transportation base." 3. The Transportation base part is extended by the "Framework Data Content Standard Part 7c: Roads," which sets forth the requirements for road system models. 4. The Framework Data Content Standard Part 7: Transportation is incorporated into this standard by reference.

Domain of Values for this Element	None.
Source of Values	None.
How Defined (e.g., locally, from standard, other)	By Address Transportation System Authority
Example	DC Street Spatial Data Base TIGER/MAF File
Notes/Comments	1. The Transportation Standard base part "defines the data model for describing transportation systems components of transportation systems for the modes [Roads, rail, inland waterways, and transit] that compose the Transportation theme of the NSDI." ("Framework Data Content Standard Part 7: Transportation base", Section 1, "Scope."). 2. All thoroughfare addresses, by definition, are located by reference to a thoroughfare--that is, by reference to a component of the transportation system. In addition, many landmark addresses and some postal addresses may also be so located, by virtue of alias addresses, road frontages, etc. 3. To make explicit the relationship between addresses and transportation networks, to provide a foundation for Address Reference Systems, and to strengthen address data quality testing, the "Framework Data Content Standard Part 7: Transportation" is incorporated by reference into this standard. 4. A thoroughfare is defined in Part 2: Street Address Data Classification as follows: "...a road or other access route by which the addressed feature can be reached... A thoroughfare is typically but not always a road — it may be, for example, a walkway, a railroad, or a river. Most Address Reference Systems pertain only to road systems--addresses are rarely assigned along rail lines or waterways. 5. Where only roads are of concern, reference should also be made to the "Framework Data Content Standard Part 7c: Roads," which extends the Transportation Standard base part.
XML Tag	<AddressTransportationSystemName>
XML Model	<xsd:simpleType name="AddressTransportationSystemName_type"> <xsd:restriction base="xsd:string"> <xsd:pattern value='.*' /> </xsd:restriction> </xsd:simpleType>
XML Example	<AddressTransportationSystemName>TIGER/MAF File</AddressTransportationSystemName>
Quality Measures	Tabular Domain Measure
Quality Notes	

2.10.4.2 Address Transportation System Authority

Element Name	Address Transportation System Authority
Other common names for this element	Department of Transportation, Public Works Department, Roads Department, etc.
Definition	The authority that maintains the transportation base model specified by the Address Transportation System Name, and assigns Address Transportation Feature I Ds to the features it represents.
Data Type	characterString
Existing Standards for this Element	None.
Domain of Values for this Element	None.
Source of Values	None.
How Defined (e.g., locally, from standard, other)	NA
Example	**District of Columbia Department of Transportation** (Street Spatial Data Base) **U.S. Census Bureau** (TIGER/MAF file)
Notes/Comments	The authority is typically the office or agency responsible for opening, maintaining, and closing the transportation features represented in the transportation base model. In some cases, the data model may be maintained by a federal agency or a private-sector firm.
XML Tag	<AddressTransportationSystemAuthority>
XML Model	<xsd:simpleType name="AddressTransportationSystemAuthority_type"> <xsd:restriction base="xsd:string"> <xsd:pattern value='.*' /> </xsd:restriction> </xsd:simpleType>
XML Example	<AddressTransportationSystemAuthority>District of Columbia Department of Transportation</AddressTransportationSystemauthority>
Quality Measures	Tabular Domain Measure
Quality Notes	

1183 ## 2.10.4.3 Address Transportation Feature Type

Element Name	Address Transportation Feature Type
Other common names for this element	Point, centroid; node, intersection; line, arc, segment, edge; path, route
Definition	The type of transportation feature (TranFeature) used to represent an address.
Data Type	characterString
Existing Standards for this Element	For transportation features generally: U.S. Federal Geographic Data Committee, "Framework Data Content Standard Part 7: Transportation base." For roads features only: U.S. Federal Geographic Data Committee, "Framework Data Content Standard Part 7: Transportation base," as extended by "Framework Data Content Standard Part 7c: Roads."
Domain of Values for this Element	For transportation features generally: Point event, linear event, transportation point (TranPoint), transportation segment (TranSeg), or transportation path (TranPath) For road features only: RoadPointFeatureEvent, RoadLinearFeatureEvent, RoadPoint, RoadSeg, or RoadPath
Source of Values	U.S. Federal Geographic Data Committee, "Framework Data Content Standard Part 7: Transportation base." See especially Sections 5 (Terms and Definitions), and Section 7 (Requirements).
How Defined (e.g., locally, from standard, other)	For all transportation features: U.S. Federal Geographic Data Committee, "Framework Data Content Standard Part 7: Transportation base." For road features: "Framework Data Content Standard Part 7c: Roads."
Examples	**Point event:** parcel centroid, building centroid, etc., located along a thoroughfare. **Linear event:** parcel frontage, building frontage, etc. located along a thoroughfare **Transportation point:** Any Intersection Address **Transportation segment:** A length of road between two intersecting roads (First Street between A Street and B Street) **Transportation path:** A length of including multiple segments (First Street from beginning to end)
Notes/Comments	1. This element is meaningful only in the context of a transportation base model as defined in the FGDC's "Framework Data Content Standard Part 7." Transportation features are defined therein. 2. The type of transportation feature used to represent an address depends on: --a. the class of the address, and --b. (in some cases) how the address in mapped (i.e. as a point, line, or polygon). These relationships are explained more fully in Appendix C (Section 3) of this

	standard.
XML Tag	<AddressTransportationFeatureType>
XML Model	<xsd:simpleType name="AddressTransportationFeatureType_type"> <xsd:restriction base="xsd:string"> <xsd:pattern value='.*' /> </xsd:restriction> </xsd:simpleType>
XML Example	<AddressTransportationFeatureType>RoadPoint</AddressTransportationFeatureType>
Quality Measures	Address Completeness Measure Intersection Validity Measure Segment Directionality Consistency Measure XY Coordinate Completeness Measure XY Coordinate Spatial Measure
Quality Notes	

2.10.4.4 Address Transportation Feature ID

Element Name	**Address Transportation Feature ID**
Other common names for this element	
Definition	The unique identifier assigned to the particular feature that represents an address within a transportation base model.
Data Type	characterString
Existing Standards for this Element	U.S. Federal Geographic Data Committee, "Framework Data Content Standard Part 7: Transportation base." "Framework Data Content Standard Part 7c: Roads,"
Domain of Values for this Element	Constrained by reference transportation base model.
Source of Values	Reference transportation base model.
How Defined (e.g., locally, from standard, other)	Within reference transportation base model.
Example	9087456
Notes/Comments	1. The reference transportation base model might identify addresses by their

	Address ID, or it might assign a different identifier within the transportation base model. 2. If a different identifier is assigned within the transportation base model, then the Address Transportation Feature ID will serve, within the scope of the address record, as a foreign key to the transportation base model.
XML Tag	<AddressTransportationFeatureID>
XML Model	<xsd:simpleType name="AddressTransportationFeatureId_type"> <xsd:restriction base="xsd:string"> <xsd:pattern value='.*' /> </xsd:restriction> </xsd:simpleType>
XML Example	<AddressTransportationFeatureID>9087456</AddressTransportationFeatureID>
Quality Measures	Pattern Sequence Measure Uniqueness Measure
Quality Notes	

2.10.4.5 Related Transportation Feature ID

Element Name	**Related Transportation Feature ID**
Other common names for this element	
Definition	The unique identifier assigned (within the reference transportation base model) to a transportation feature to which an address is related.
Data Type	characterString
Existing Standards for this Element	U.S. Federal Geographic Data Committee, "Framework Data Content Standard Part 7: Transportation base." "Framework Data Content Standard Part 7c: Roads."
Domain of Values for this Element	Constrained by reference transportation base model.
Source of Values	Reference transportation base model.
How Defined (e.g., locally, from standard, other)	Within the reference transportation base model.
Example	786542
Notes/Comments	1. Thoroughfare addresses (other than Intersection Addresses) are

	represented within a transportation base model as point events or linear events, each with a unique Address Transportation Feature ID. These point events and linear events may, turn, be related to one or more transportation segments within the transportation base model. The transportation segment must have a Complete Street Name and an address range that includes the Complete Street Name and Complete Address Number of the address. 2. The Related Transportation Feature ID provides the ID, as assigned within the transportation base model, of the related segment. 4. Intersection Addresses are related to one or more transportation points within the transportation data model. For Intersection Addresses, the TranPoint ID would be placed within the Related Transportation Feature ID element.
XML Tag	<RelatedTransportationFeatureID>
XML Model	<xsd:simpleType name="RelatedTransportationFeatureId_type"> <xsd:restriction base="xsd:string"> <xsd:pattern value='.*' /> </xsd:restriction> </xsd:simpleType>
XML Example	<RelatedTransportationFeatureID>786542</RelatedTransportationFeatureID>
Quality Measures	Related Element Uniqueness Measure
Quality Notes	

2.10.5 Address Range Attributes

2.10.5.1 Address Range Type

Element Name	**Address Range Type**
Other common names for this element	
Definition	This attribute states whether an address range (either a Two Number Address Range or a Four Number Address Range) is actual or potential. **Actual range:** the low and high Complete Address Numbers are numbers that have been assigned and are in use along the addressed feature. **Potential range:** the low and high Complete Address Numbers are numbers that would be assigned if all possible numbers were in use along the

	addressed feature, and there were no gaps between the range and its preceding and following ranges.
Definition Source	New
Data Type	characterString
Existing Standards for this Element	None
Domain of Values for this Element	Actual, Potential, Unknown
Source of Values	New
How Defined (e.g., locally, from standard, other)	New
Example	**Actual** range
Notes/Comments	1. Ranges may be actual or potential. 2. Actual ranges give the lowest and highest Complete Address Numbers that have been assigned and are in use along the addressed featurem, excluding any addresses that are anomalies, especially with regard to parity or sequence. 3. Potential (or theoretical) ranges include all the numbers that could be assigned along the addressed feature based on the Address Reference System Numbering Rules. Potential ranges permit no numbering gaps between the range and its preceding and following ranges. Potential ranges are equal to or broader than actual ranges. 4. The Census Bureau uses theoretical ranges in its TIGER files, to ensure continuity from census to census. Potential ranges are also used in Googlemaps, Mapquest and other online road map and routing services, because they get their data originally from Census TIGER files. 5. Theoretical ranges are useful for software, such as some computer aided emergency dispatching applications, that requires continuous ranges along the length of a street. 6. Ranges are often used for geocoding, but point matches are preferable. 7. When constructing actual ranges, the lowest assigned Address Number and the highest assigned Address Number in use along a given segment are used. However, no Address Number which is an anomaly (as to range parity or side, or for any other reason) is to be used in constructing the actual address range.
XML Tag	<AddressRangeType>
XML Model	<xsd:simpleType name="AddressRangeType_type"> <xsd:annotation> <xsd:documentation xml:lang="en"> This attribute states whether an address range (either a Two Number Address

	Range or a Four Number Address Range) is actual or potential. </xsd:documentation> </xsd:annotation> <xsd:restriction base="xsd:string"> <xsd:enumeration **value="Actual"** > <xsd:annotation> <xsd:documentation>the low and high Complete Address Numbers are numbers that have been assigned and are in use along the addressed feature. </xsd:documentation> </xsd:annotation> </xsd:enumeration> <xsd:enumeration **value="Potential"** > <xsd:annotation> <xsd:documentation>The low and high Complete Address Numbers are numbers that would be assigned if all possible numbers were in use along the addressed feature, and there were no gaps between the range and its preceding and following ranges. </xsd:documentation> </xsd:annotation> </xsd:enumeration> <xsd:enumeration **value="Unknown"** > <xsd:annotation> <xsd:documentation>The relationship between the low and high Complete Address Numbers and the addressed feature is unknown. </xsd:documentation> </xsd:annotation> </xsd:enumeration> </xsd:restriction> </xsd:simpleType>
XML Example	<AddressRangeType>Actual</AddressRangeType>
Quality Measures	Tabular Domain Measure
Quality Notes	Address Range Type validation is completely dependent on Address Reference System in a given area, and will have to be formulated locally.

2.10.5.2 Address Range Parity

Element Name	**Address Range Parity**
Other common names for this element	

Definition	The set of Address Number Parity values specified in the Address Reference System Numbering Rules for the Address Numbers in an address range.
Definition Source	New
Data Type	characterString
Existing Standards for this Element	None
Domain of Values for this Element	Even, Odd, Both, None, Unknown
Source of Values	New
How Defined (e.g., locally, from standard, other)	**Odd** - All Address Numbers in the range have an Address Number Parity of "odd" **Even** - All Address Numbers in the range have an Address Number Parity of "even" **Both** - Both even and odd Address Numbers are found in the range **None** - No Address Number is found within the range **Unknown** - The parity of the Address Numbers in the range in not known.
Examples	**Odd** - 101 - 199 Main Street **Even** - 100 - 198 Main Street **Both** - 100 - 199 Main Street **None** - (null) - (null) Main Street (no address numbers assigned to that specific segment)
Notes/Comments	1. Odd and even Address Numbers are usually associated with opposite sides of a thoroughfare. For example, a jurisdiction may have rules within its Address Reference System Rules to consistently assign odd numbers to the "left" side of its thoroughfares and even numbers to the "right" side. (See Address Range Side for how "left" and "right" are defined). 2. The Address Range Parity is determined using the Address Reference System Numbering Rules. For theoretical type ranges, the low and high numbers are the lowest and highest numbers of the identified parity found within the identified block within the Address Reference System. For actual ranges, the lowest and highest Address Number in use for the selected block are identified and used. Anomalous addresses (e.g., those Address Numbers that have a parity that is not the same as the Address Range Parity are not used in creating the actual Address Range? or in determining the Address Range Parity. 3. The expected values for Address Range Parity depend on rules found in the Address Reference System Rules, and are associated with the Address Range Side. If the address range includes addresses from only one side of the thoroughfare, the Address Range Parity is typically but not always "odd" or "even". If the range covers both sides of the thoroughfare, then the Address Range Parity is typically "both" 4. If no addresses occur within a range, then the Address Range Parity is

	"none."
XML Tag	<AddressRangeParity>
XML Model	<xsd:simpleType name="AddressRangeParity_type"> <xsd:annotation> <xsd:documentation xml:lang="en"> The set of Address Number Parity values specified in the Address Reference System Numbering Rules for the Address Numbers in an address range. </xsd:documentation> </xsd:annotation> <xsd:restriction base="xsd:string"> <xsd:pattern value='.*' /> <xsd:enumeration **value="even"** > <xsd:annotation> <xsd:documentation> All Address Numbers in the range have an Address Number Parity of "even". </xsd:documentation> </xsd:annotation> </xsd:enumeration> <xsd:enumeration **value="odd"** > <xsd:annotation> <xsd:documentation> All Address Numbers in the range have an Address Number Parity of "odd". </xsd:documentation> </xsd:annotation> </xsd:enumeration> <xsd:enumeration **value="both"** > <xsd:annotation> <xsd:documentation> Both even and odd Address Numbers are found in the range. </xsd:documentation> </xsd:annotation> </xsd:enumeration> <xsd:enumeration **value="none"** > <xsd:annotation> <xsd:documentation> No Address Number is found within the range. </xsd:documentation> </xsd:annotation></xsd:enumeration> <xsd:enumeration **value="unknown"** > <xsd:annotation> <xsd:documentation>The parity of the Address Numbers in the range in not known. </xsd:documentation></xsd:annotation></xsd:enumeration> </xsd:restriction> </xsd:simpleType>

XML Example	<AddressRangeParity>odd</AddressRangeParity>
Quality Measures	Address Number Range Parity Consistency Measure
Quality Notes	

2.10.5.3 Address Range Side

Element Name	**Address Range Side**
Other common names for this element	
Definition	The side of a transportation segment (TranSeg) on which the address range is found (right, left or both).
Definition Source	New
Data Type	characterString
Existing Standards for this Element	None
Domain of Values for this Element	right, left, both, unknown
Source of Values	New
How Defined (e.g., locally, from standard, other)	New
Example	Elm Street runs south-to-north. For each block,the from-node is at the south end, and the to-node is at the north end. "Right" and "left" are defined by standing at the south (from) end, and facing the north (to) end. The "right" side is in this case the east side, and the "left" side is the west side. (If the from- and to- nodes were reversed, "left' and "right' would also be reversed.)
Notes/Comments	1. Address Range Side has nothing to do with traffic flow or compass direction. 2. Address Range Side states whether the range includes Complete Address Numbers on right side, left side, or both sides of the thoroughfare. 3. "Right" and "left" must be defined by reference to a specific transportation segment (or set of segments) in a particular transportation network model. By definition, every transportation segment has a from-node at one end and a to-node at the other end. The directionality, right side, and left side of the segment are determined by standing at the from-node and facing the to-node.

	Address Left Right Measure and Address Range Directionality Measure provide tools for determining "left", "right" and directionality. 4. Address Range Directionality can be defined only for a Two Number Address Range or a Four Number Address Range that has been related to a specific transportation segment (or set of segments) in a particular transportation network model. 5. Use the Address Transportation System Name, Address Transportation System Authority, Address Transportation Feature Type, Address Transportation Feature ID, and Related Transportation Feature ID attributes to relate a particular address range to a specific transportation segment (or set of segments) in a specific transportation network model. Transportation segments, and transportation network models generally, are defined and described in the FGDC's "Geographic Information Framework Data Content Standard Part 7: Transportation Base."
XML Tag	<AddressRangeSide>
XML Model	<xsd:simpleType name="AddressRangeSide_type"> <xsd:annotation> <xsd:documentation xml:lang="en"> The side of the transportation segment (right , left, both, none, unknown) on which the address range applies. </xsd:documentation> </xsd:annotation> <xsd:restriction base="xsd:string"> <xsd:pattern value='.*' /> <xsd:enumeration **value="right"** > <xsd:annotation> <xsd:documentation> The address is related to the right side of the street. </xsd:documentation> </xsd:annotation> </xsd:enumeration> <xsd:enumeration **value="left"** > <xsd:annotation> <xsd:documentation> The address is realted to the left side of the street. </xsd:documentation> </xsd:annotation> </xsd:enumeration> <xsd:enumeration **value="both">** <xsd:annotation> <xsd:documentation> The address pertains to both sides of the street. </xsd:documentation> </xsd:annotation>

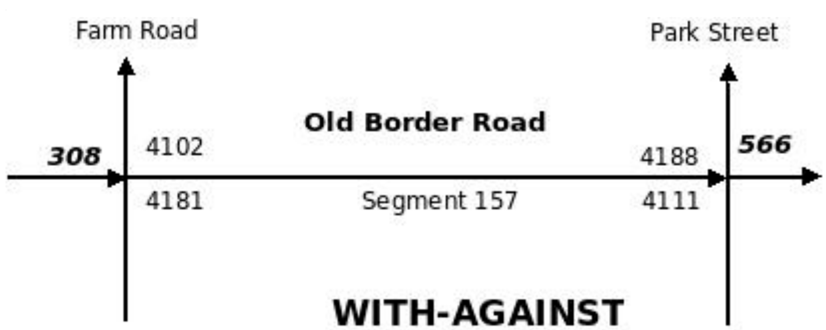
	</xsd:enumeration> <xsd:enumeration **value="none"** > <xsd:annotation> <xsd:documentation>The address is not on either or both sides of the street or the concept of side of street does not apply to the address. For instance an intersection address would have a Address Side Of Street of none. </xsd:documentation> </xsd:annotation></xsd:enumeration> <xsd:enumeration **value="unknown"** ></xsd:enumeration> </xsd:restriction> </xsd:simpleType>
XML Example	<AddressRangeSide>left</AddressRangeSide>
Quality Measures	Left Right Odd Even Parity Measure Address Left Right Measure
Quality Notes	Note that this measure checks the agreement of a an Address Range Side attribute with geometry, while Left Right Odd Even Parity Measure checks the agreement of an Address Number against an established local rule for associating address parity with the right or left side of the street when traveling away from the governing Address Reference System Axis Point Of Beginning.

2.10.5.4 Address Range Directionality

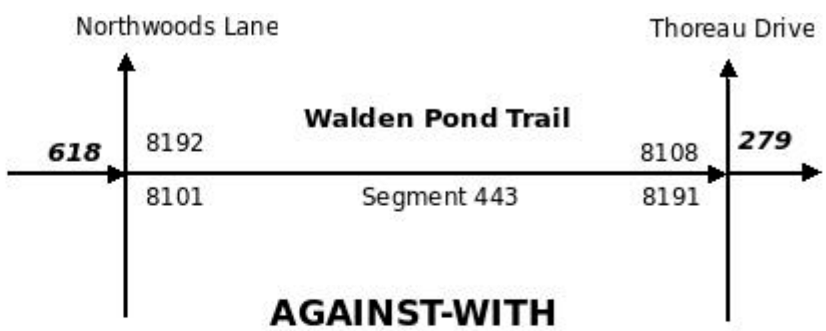
Element Name	**Address Range Directionality**
Other common names for this element	
Definition	Whether the low Complete Address Number of an address range is closer to the from-node or the to-node of the transportation segment(s) that the range is related to.
Definition Source	New
Data Type	characterString
Existing Standards for this Element	None
Domain of Values for this Element	**With** - The low address is nearer the from node; numbers ascend toward the to node. **Against** - The low address is nearer the to node; numbers descend toward the to node.

	With-Against - The numbers run in opposite directions on either side of the street. The low number on the left side is nearer the from node. The low number on the right side is nearer the to node. **Against-With** - The numbers run in opposite directions on either side of the street. The low number on the left side is nearer the to node. The low number on the right side is nearer the from node. **Null** - The address range has null values for the high and low Complete Address Numbers. **NA** - Does not apply (transportation segment directionality is inconsistent within the range). **Unknown** - The address range directionality is not known.
Source of Values	New
How Defined (e.g., locally, from standard, other)	New
Example	Smalltown has a digital street centerline network model. Each street is mapped as a series of segments that run from one intersection to another. 1. **With:** Segment 1 represents Main Street from First Street to Second Street. It runs from Node 1 to Node 2. (That is, From-node = Node 1; To-node = Node 2). Node 1 = Main and First; Node 2 = Main and Second. The Four Number Address Range along this segment is 100 - 198; 101 - 199 Main Street. 100 Main and 101 Main are both near Node 1 (First and Main); the high numbers are near Main and Second. The Address Range Directionality for this Four Number Address Range is **With** the segment directionality.  2. **Against:** Segment 25 represents Elm Street from Oak Street to Pine Street. Segment 25 From-node = Node 92; To-node = Node 77. Node 92 = Elm and Oak; Node 77 = Elm and Pine. The Four Number Address Range along this segment is 110 - 180; 111 - 187 Elm Street. 110 Elm and 111 Elm are both near Node 77 (Elm and Pine); the high numbers are near Elm and Oak. The Address Range Directionality for this Four Number Address Range is **Against** the segment directionality.

3. **Special Case: With - Against:** Segment 157 is unusual--the address numbers run in different directions on each side of the street. Segment 157 represents Old Border Road from Farm Road to Park Street. Segment 157 From-node = Node 308; To-node = Node 566. Node 308 = Old Border and Farm; Node 566 = Old Border and Park. The Four Number Address Range along this segment is 4102 - 4188; 4111 - 4181 Old Border Road. 4102 Old Border and 4181 Old Border are both near Node 308 (Old Border and Farm). 4188 Old Border and 4111 Old Border are both near Node 566 (Old Border and Park). The Address Range Directionality for this Four Number Address Range is **With - Against** the segment directionality.



4. **Special Case: Against - With:** This is the reverse of the previous case. Segment 443 also has address numbers that run in different directions on each side of the street. Segment 443 represents Walden Pond Trail from Northwoods Lane to Thoreau Drive. Segment 443 From-node = Node 618; To Node = 279. Node 618 = Walden Pond Trail and Northwoods Lane, and Node 279 = Walden Pond Trail and Thoreau Drive. The Four Number Address Range along this segment is 8108 - 8192; 8101 - 8191. 8192 Walden Pond Trail and 8101 Walden Pond Trail are near Node 618 (Walden Pond Trail and Northwoods Lane) while 8108 Walden Pond Trail and 8191 Walden Pond Trail are near Node 279 (Walden Pond Trail and Thoreau Drive). The Address Range Directionality for this Four Number Address Range is **Against - With**

	the segment directionality. 
Notes/Comments	1. Address Range Directionality has nothing to do with traffic flow or compass direction. 2. Address Range Directionality states whether the Complete Address Numbers ascend or descend as one proceeds from the from-node to the to-node of the transportation segments (TranSeg(s)) to which the range is related. 3. Address Range Directionality can be defined only for a Two Number Address Range or a Four Number Address Range that has been related to a specific TranSeg (or set of TranSegs) in a particular transportation network model. 4. By definition, TranSegs have a from-node and a to-node, which determine the TranSeg's directionality, right side, and left side. 5. If the low Complete Address Number of a range is closer to the from-node, and the high Complete Address Number is closer to the to-node, then the Complete Address Numbers ascend **With** the TranSeg directionality. 6. If the low Complete Address Number of a range is closer to the to-node, and the high Complete Address Number is closer to the from-node, then the Complete Address Numbers ascend **Against** the TranSeg directionality. 7. If the low and high Complete Address Numbers of a range are equal, or equidistant from the from-node and to-node, or if the from-node and the to-node are the same (a loop), then by definition the Complete Address Numbers are considered to ascend **With** the Tran Seg directionality. 8. If the two ranges of a Four Number Address Range have different Address Range Directionality, then give the left range directionality first, followed by the right range directionality: **"With - Against"** or **"Against - With"** 9. Special values apply in the following cases: **---Null** - the address range contains null values. **---Unknown** - the range directionality (or the relative locations of the low and high Complete Address Numbers) is unknown. **---NA (not applicable)** - the range covers multiple TranSegs, and the TranSegs have inconsistent segment directionality. 10. Use the Address Transportation System Name, Address Transportation System Authority, Address Transportation Feature Type, Address Transportation Feature ID, and Related Transportation Feature ID attributes to relate a particular address range to a specific transportation segment (or set of

	segments) in a specific transportation network model. TranSegs, and transportation network models generally, are defined and described in the FGDC's "Geographic Information Framework Data Content Standard Part 7: Transportation Base."
XML Tag	<AddressRangeDirectionality>
XML Model	<xsd:simpleType name="AddressRangeDirectionality_type"> <xsd:annotation> <xsd:documentation xml:lang="en"> Whether the low Complete Address Number of an address range is closer to the from-node or the to-node of the transportation segment(s) that the range is related to. </xsd:documentation> </xsd:annotation> <xsd:restriction base="xsd:string"> <xsd:enumeration value="With"> <xsd:annotation> <xsd:documentation>The low address is nearer the from node; numbers ascend toward the to node. </xsd:documentation> </xsd:annotation> </xsd:enumeration> <xsd:enumeration value="Against"> <xsd:annotation> <xsd:documentation>The low address is nearer the to node; numbers descend toward the to node. </xsd:documentation> </xsd:annotation> </xsd:enumeration> <xsd:enumeration value="With-Against"> <xsd:annotation> <xsd:documentation>The numbers run in opposite directions on either side of the street. The low number on the left side is nearer the from node. The low number on the right side is nearer the to node.</xsd:documentation></xsd:annotation></xsd:enumeration> <xsd:enumeration value="Against-With"> <xsd:annotation> <xsd:documentation>The numbers run in opposite directions on either side of the street. The low number on the left side is nearer the to node. The low number on the right side is nearer the from node. </xsd:documentation> </xsd:annotation> </xsd:enumeration> <xsd:enumeration value="Null"> <xsd:annotation>

	<xsd:documentation>The address range has null values for the high and low Complete Address Numbers. </xsd:documentation> </xsd:annotation> </xsd:enumeration> <xsd:enumeration value="NA"> <xsd:annotation> <xsd:documentation>Does not apply (transportation segment directionality is inconsistent within the range). </xsd:documentation> </xsd:annotation> </xsd:enumeration> <xsd:enumeration value="Unknown"> <xsd:annotation> <xsd:documentation>The address range directionality is not known. </xsd:documentation> </xsd:annotation> </xsd:enumeration> </xsd:restriction> </xsd:simpleType>
XML Example	<AddressRangeDirectionality>With-Against</AddressRangeDirectionality>
Quality Measures	Address Range Directionality Measure
Quality Notes	

2.10.6 Address Attributes

2.10.6.1 Address Classification

Element Name	**Address Classification**
Other common names for this element	Address Type, Address Class
Definition	The class of the address as defined in the Classification Part of this standard.
Definition Source	New
Data Type	characterString
Existing Standards for this Element	The Classification Part of this standard.

Domain of Values for this Element	Class names given in the Classification Part of this standard.
Source of Values	The Classification Part of this standard.
How Defined (e.g., locally, from standard, other)	In the Classification Part of this standard.
Examples	Numbered Thoroughfare Address Intersection Address Two Number Address Range Four Number Address Range Unnumbered Thoroughfare Address Landmark Address Community Address USPS Postal Delivery Box USPS Postal Delivery Route USPS General Delivery Office General Address Class
Notes/Comments	Address classes are defined and described in the Classification part of this standard.
XML Tag	<AddressClassification>
XML Model	<xsd:simpleType name="AddressClassification_type"> <xsd:restriction base="xsd:string"> <xsd:enumeration value="NumberedThoroughfareAddress"></xsd:enumeration> <xsd:enumeration value="IntersectionAddress"></xsd:enumeration> <xsd:enumeration value="TwoNumberAddressRange"></xsd:enumeration> <xsd:enumeration value="FourNumberAddressRange"></xsd:enumeration> <xsd:enumeration value="UnnumberedThoroughfareAddress"></xsd:enumeration> <xsd:enumeration value="LandmarkAddress"></xsd:enumeration> <xsd:enumeration value="CommunityAddress"></xsd:enumeration> <xsd:enumeration value="USPSPostalDeliveryBox"></xsd:enumeration> <xsd:enumeration value="USPSPostal Delivery Route?"></xsd:enumeration> <xsd:enumeration value="USPSGeneral Delivery Office?"></xsd:enumeration> <xsd:enumeration value="GeneralAddressClass"></xsd:enumeration> </xsd:restriction> </xsd:simpleType>
XML Example	<AddressClassification>IntersectionAddress<AddressClassification>
Quality Measures	Tabular Domain Measure Pattern Sequence Measure

Quality Notes	The Tabular Domain Measure checks on whether a classification entry actually exists. The Pattern Sequence Measure can be used to check whether the entry associated with the classification matches its description.

2.10.6.2 Address Feature Type

Element Name	**Address Feature Type**
Other common names for this element	
Definition	A category of real world phenomena with common properties whose location is specified by an address.
Definition Source	Adapted from FGDC Framework Data Content Standard, Part 0: Base Document, Section 5.22
Data Type	characterString
Existing Standards for this Element	None
Domain of Values for this Element	May be created locally
Source of Values	Local
How Defined (e.g., locally, from standard, other)	Locally
Example	Parcel, building, building entrance, service entrance, subaddress, power pole, cell tower
Notes/Comments	Initial list of feature types: Block, block face, intersection, parcel, building, entrance, subaddress. The list might be expanded indefinitely to include infrastructure and other features. An address may designate multiple Address Feature Types.
XML Tag	<AddressFeatureType>
XML Model	<xsd:simpleType name="AddressFeatureType_type"> <xsd:annotation> <xsd:documentation xml:lang="en"> The type of feature identified by the address Initial list of feature types: Building Utility Cabinet,

	Telephone Pole, Building, Street block, street block face, intersection, parcel, building, entrance, unit. The list might be expanded indefinitely to include infrastructure and other features. </xsd:documentation> </xsd:annotation> <xsd:restriction base="xsd:string"> <xsd:pattern value='.+*' /> </xsd:restriction> </xsd:simpleType>
XML Example	<AddressFeatureType>Cell Tower</AddressFeatureType>
Quality Measures	Tabular Domain Measure Address Reference System Description Address Completeness Measure
Quality Notes	Address Feature Type elements may be defined in the Address Reference System Description, and should be checked there. Address Completeness Measure checks whether all the addressable objects have assigned addresses.

2.10.6.3 Address Lifecycle Status

Element Name	**Address Lifecycle Status**
Other common names for this element	
Definition	The lifecycle status of the address.
Definition Source	New
Data Type	characterString
Existing Standards for this Element	None
Domain of Values for this Element	**Potential** = Address falls within a theoretical range (See Address Range Type), but has never been used; **Proposed** = Application pending for use of this address (e.g., address tentatively issued for subdivision plat that is not yet fully approved); **Active** = Address has been issued and is in use; **Retired** = Address was issued, but is now obsolete (e.g. street name has been changed, building was demolished, etc.)
Source of Values	New

How Defined (e.g., locally, from standard, other)	From this standard
Notes/Comments	1. An address should be assigned as early as possible in the development process, generally upon subdivision or issuance of the intial building permit. Long before occupancy, a site may require construction deliveries, emergency services, or mention in official records, all of which are facilitated if the address is assigned and known. 2. An address, once issued, should not be deleted from the records, even if it falls out of use. If an address becomes obsolete, its status should be changed from "active" to "retired".
XML Tag	<AddressLifecycleStatus>
XML Model	<xsd:simpleType name="AddressLifecycleStatus_type"> <xsd:annotation> <xsd:documentation xml:lang="en"> The life cycle status of the address. </xsd:documentation> </xsd:annotation> <xsd:restriction base="xsd:token"> <xsd:enumeration **value="Potential"** > <xsd:annotation> <xsd:documentation> Address falls within a theoretical range, but has never been used. </xsd:documentation> </xsd:annotation> </xsd:enumeration> <xsd:enumeration **value="Proposed"** > <xsd:annotation> <xsd:documentation> Application pending for use of this address (e.g., address tentatively issued for subdivision plat that is not yet fully approved). </xsd:documentation> </xsd:annotation> </xsd:enumeration> <xsd:enumeration **value="Active"** > <xsd:annotation> <xsd:documentation> Address has been issued and is in use. </xsd:documentation> </xsd:annotation> </xsd:enumeration> <xsd:enumeration **value="Retired"** > <xsd:annotation> <xsd:documentation>

	Address was issued, but is now obsolete (e.g. street name has been changed), building was demolished, etc. </xsd:documentation> </xsd:annotation> </xsd:enumeration> </xsd:restriction> </xsd:simpleType>
XML Example	<AddressLifecycleStatus>Proposed</AddressLifecycleStatus>
Quality Measures	Tabular Domain Measure Address Lifecycle Status Date Consistency Measure
Quality Notes	Each locality will have records describing conditions associated with a given lifecycle status. While the nature of these records and methods for checking correspondence with Address Lifecycle Status entries are beyond the scope of the standard, they may be considered in a local quality program.

1204

1205 ### 2.10.6.4 Official Status

Element Name	**Official Status**
Other common names for this element	Official address, legal address, alias address, alternate address, variant address
Definition	Whether the address, street name, landmark name, or place name is as given by the official addressing authority (official), or an alternate or alias (official or unofficial), or a verified error.
Definition Source	New
Data Type	characterString
Existing Standards for this Element	No
Domain of Values for this Element	1. Official 2. Alternate or Alias ---2.1 Official Alternate or Alias ------2.1.1 Alternate Established by an Official Renaming Action of the Address Authority ------2.1.2 Alternates Established by an Address Authority ---2.2 Unofficial Alternate or Alias ------2.2.1 Alternate Established by Colloquial Use ------2.2.2. Unofficial Alternate in Frequent Use ------2.2.3. Unofficial Alternate in Use by Agency or Entity

	------2.2.4. Posted or Vanity Address 3. Verified Invalid
Source of Values	New
How Defined (e.g., locally, from standard, other)	New
Example	See notes below.
Notes/Comments	**1. Official** The address or name as designated by the Address Authority. **2. Alternate or Alias** An alternate or alias to the official address or name that is also in official or popular use. The Related Address ID can be used to link an alternate or alias to the Address ID of the official address. There are two types of alternate or alias names, official and unofficial, each of which has subtypes. **2.1 Official Alternate or Alias:** These are alternate names designated by an official Address Authority. Subtypes include, but are not limited to: * *Official Renaming Action of the* Address Authority An Address Authority may replace one address or name with another, e.g. by renaming or renumbering. The prior, older address should be retained as an alias, to provide for conversion to the new address. . * *Alternates Established by an* Address Authority An Address Authority may establish a name or number to be used in addition to the official address or name. For example, a state highway designation (State Highway 7) may be given to a locally-named road, or a memorial name may be applied to an existing street by posting an additional sign, while the local or original name and addresses continue to be recognized as official. **2.2 Unofficial Alternate or Alias:** These are addresses or names that are used by the public or by an individual, but are not recognized as official by the Address Authority: Some examples include, but are not limited to: * *Alternates Established by Colloquial Use in a Community* An address or name that is in popular use but is not the official name or an official alternate or alias. * *Unofficial Alternates Frequently Encountered* In data processing, entry errors occur. Such errors if frequently encountered may be corrected by a direct match of the error and a substitution of a correct name. * *Unofficial Alternates In Use by an Agency or Entity* For data processing efficiency, entities often create alternate names or abbreviations for internal use. These must be changed to the official form for public use and transmittal to external users. * *Posted or Vanity Address* An address that is posted, but is not recognized by the Address Authority (e.g. a vanity address on a building); **3. Verified Invalid**

	An address that has been verified as being invalid, but which keeps appearing in address lists. Different from Unofficial Alternate Names in that these addresses are known not to exist.
XML Tag	<OfficialStatus>
XML Model	<xsd:simpleType name="OfficialStatus_type"> <xsd:annotation> <xsd:documentation xml:lang="en"> Whether the address, street name, landmark name, or place name is as given by the official addressing authority (official), or an alternate or alias (official or unofficial), or a verified error. </xsd:documentation> </xsd:annotation> <xsd:restriction base="xsd:string"> <xsd:pattern value='.*' /> <xsd:enumeration **value="Official"** > <xsd:annotation> <xsd:documentation> The address or name as designated by the Address Authority. </xsd:documentation> </xsd:annotation> </xsd:enumeration> <xsd:enumeration **value="Alternate or Alias"** > <xsd:annotation> <xsd:documentation> An alternate or alias to the official address or name that is also in official or popular use. The Related Address ID can be used to link an alternate or alias to the Address ID of the official address. There are two types of alternate or alias names, official and unofficial, each of which has subtypes. </xsd:documentation> </xsd:annotation> </xsd:enumeration> <xsd:enumeration **value="Official Alternate or Alias"** > <xsd:annotation> <xsd:documentation> These are alternate names designated by an official Address Authority. </xsd:documentation> </xsd:annotation> </xsd:enumeration> <xsd:enumeration **value="Official Renaming Action of the Address Authority"** > <xsd:annotation>

<xsd:documentation>An Address Authority may replace one address or name with another, e.g. by renaming or renumbering. The prior, older address should be retained as an alias, to provide for conversion to the new address.</xsd:documentation></xsd:annotation>
</xsd:enumeration>
<xsd:enumeration **value="Alternates Established by an Address Authority"** >
<xsd:annotation>
<xsd:documentation>An Address Authority may establish a name or number to be used in addition to the official address or name. For example, a state highway designation (State Highway 7) may be given to a locally-named road, or a memorial name may be applied to an existing street by posting an additional sign, while the local or original name and addresses continue to be recognized as official.</xsd:documentation></xsd:annotation>
</xsd:enumeration>
<xsd:enumeration **value="Unofficial Alternate or Alias"** >
<xsd:annotation>
<xsd:documentation>
These are addresses or names that are used by the public or by an individual, but are not
recognized as official by the Address Authority.
</xsd:documentation>
</xsd:annotation>
</xsd:enumeration>
<xsd:enumeration **value="Alternate Names Established by Colloquial Use in a Community"** >
<xsd:annotation>
<xsd:documentation>An address or name that is in popular use but is not the official name or an official alternate or alias.
</xsd:documentation></xsd:annotation>
</xsd:enumeration>
<xsd:enumeration **value="Unofficial Alternate Names Frequently Encountered"** >
<xsd:annotation>
<xsd:documentation>In data processing, entry errors occur. Such errors if frequently encountered may be corrected by a direct match of the error and a substitution of a correct name.
</xsd:documentation>
</xsd:annotation>
</xsd:enumeration>
<xsd:enumeration **value="Unofficial Alternate Names In Use by an Agency or Entity"** >
<xsd:annotation>
<xsd:documentation>For data processing efficiency, entities often create alternate names or abbreviations for internal use. These must be changed to the

	official form for public use and transmittal to external users. </xsd:documentation> </xsd:annotation> </xsd:enumeration> <xsd:enumeration **value="Posted or Vanity Address"** > <xsd:annotation> <xsd:documentation>An address that is posted, but is not recognized by the Address Authority (e.g. a vanity address on a building);</xsd:documentation> </xsd:annotation></xsd:enumeration> <xsd:enumeration **value="Verified Invalid"** > <xsd:annotation> <xsd:documentation> An address that has been verified as being invalid, but which keeps appearing in address lists. Different from Unofficial Alternate Names in that these addresses are known not to exist. </xsd:documentation> </xsd:annotation> </xsd:enumeration> </xsd:restriction> </xsd:simpleType>
XML Example	<OfficialStatus>Official Renaming Action of the Address Authority</OfficialStatus>
Quality Measures	Tabular Domain Measure Official Status Address Authority Consistency Measure
Quality Notes	Each locality will have records describing conditions associated with a given Official Status. While the nature of these records and methods for checking correspondence between entries are beyond the scope of the standard, they may be considered in a local quality program.

2.10.6.5 Address Anomaly Status

Element Name	**Address Anomaly Status**
Other common names for this element	
Definition	A status flag, or an explanatory note, for an address that is not correct according to the Address Reference System that governs it, but is nonetheless a valid address.

Definition Source	New
Data Type	characterString
Existing Standards for this Element	No
Domain of Values?	May be "yes" or "no", or may be an enumerated domain of anomaly types
How Defined (e.g., locally, from standard, other)	Locally
Example	An address that has an even Address Number Parity but is located on the odd-numbered side of the street.
Notes/Comments	This field may be used to identify the type of anomaly (e.g. wrong parity, out of sequence, out of range, etc.) rather than simply whether or not it is anomalous. Local jurisdictions may create specific categories for anomalies.
XML Tag	<AddressAnomalyStatus>
XML Model	<xsd:simpleType name="AddressAnomalyStatus_type"> <xsd:restriction base="xsd:string"></xsd:restriction> </xsd:simpleType>
XML Example	<AddressAnomalyStatus>yes</AddressAnomalyStatus>
Quality Measures	Tabular Domain Measure
Quality Notes	Validation tests for conditions described Address Anomaly Status values are entirely dependent on local conditions, and are beyond the scope of this standard. Some of the measures described in the standards may provide complete or partial solutions.

1208

1209
2.10.6.6 Address Side Of Street

Element Name	**Address Side Of Street**
Other common names for this element	
Definition	The side of the transportation segment (right , left, both, none, unknown) on which the address is located.
Data Type	characterString
Existing Standards for this Element	U.S. Federal Geographic Data Committee, "Framework Data Content Standard Part 7: Transportation base," sections 7.3.2 and B.3.6

Domain of Values for this Element	right, left, both, none, unknown
Source of Values	
How Defined (e.g., locally, from standard, other)	U.S. Federal Geographic Data Committee, "Framework Data Content Standard Part 7: Transportation base," Annex B.
Example	See domain of values above.
Notes/Comments	1. "Left" and "right" are defined by reference to the direction of the transportation segment to which the address is related. "The direction of a TranSeg is determined by its "from" and "to" TranPoints" (Transporation base standard, section 7.3.2). "Left" and "right" are defined by facing the "to" TranPoint. 2. Most addresses are located to the left or right of the segment. The value of "none" can be used only for Intersection Addresses, which by definition occur at the point of intersection of two or more street segments. An Intersection Address begins or ends a segment and so is not on either side of it. 3. If an addressed feature straddles the thoroughfare to which it is addressed (a rare occurence but it does happen), it should be given the Address Side Of Street value that corresponds to the correct side for the number that was assigned to the feature. 4. Address Side Of Street does not apply to address ranges. Use the the Address Range Side attribute to give the side of a Two Number Address Range or a Four Number Address Range.
XML Tag	<AddressSideOfStreet>
XML Model	<xsd:simpleType name="AddressSideOfStreet_type"> <xsd:restriction base="xsd:string"> <xsd:pattern value='.*' /> <xsd:enumeration **value="right"** > <xsd:annotation> <xsd:documentation> The address is related to the right side of the street. </xsd:documentation> </xsd:annotation> </xsd:enumeration> <xsd:enumeration **value="left"** > <xsd:annotation> <xsd:documentation> The address is realted to the left side of the street. </xsd:documentation> </xsd:annotation> </xsd:enumeration> <xsd:enumeration **value="both"** > <xsd:annotation>

	<xsd:documentation> The address pertains to both sides of the street. </xsd:documentation> </xsd:annotation> </xsd:enumeration> <xsd:enumeration **value="none"** > <xsd:annotation> <xsd:documentation>The address is not on either or both sides of the street or the concept of side of street does not apply to the address. For instance an intersection address would have a Address Side Of Street of none. </xsd:documentation> </xsd:annotation></xsd:enumeration> <xsd:enumeration **value="unknown"** ></xsd:enumeration> </xsd:restriction> </xsd:simpleType>
XML Example	<AddressSideOfStreet>both</AddressSideOfStreet>

2.10.6.7 Address Z Level

Element Name	**Address Z Level**
Other common names for this element	Floor, building level, story
Definition	Floor or level of the structure
Definition Source	New
Data Type	Integer
Existing Standards for this Element	N/A
Domain of Values for this Element	Positive integers
Source of Values	Field observations, building plans, or other source of spatial data collection.
How Defined (e.g., locally, from standard, other)	The lowest level of a building is 1, and ascending numbers are assigned in order to each higher level.
Examples	**1** (=lowest floor), **3** (the ground floor, if the structure has two below-ground floors)

Notes/Comments	1. This attribute is intended for use with multi-story buildings, where the [Subaddress Element] does not indicate the building level on which the subaddress is found. Common examples include hotel lobbies and mezzanines, named meeting rooms in conference centers, and multi-unit residential buildings whose unit identifiers do not indicate the building level ("Penthouse", "Basement"). 2. "Ground level" is often ambiguous (especially when the building itself is built on sloping ground), and floor designations often omit parking and basement levels at the base of the building. To avoid confusion in assigning [Address Z Level] values, 1 should be assigned to the lowest level of the building, and ascending numbers assigned in order to each higher level, regardless of how that level is named within the building floor plan. Use the [Subaddress Element] to record how a subaddress is named in the building floor plan.
XML Tag	<AddressZLevel>
XML Model	<xsd:simpleType name="AddressZLevel_type"> <xsd:restriction base="xsd:string"> <xsd:pattern value='.*' /> </xsd:restriction> </xsd:simpleType>
XML Example	<AddressZLevel>13</AddressZLevel>
Quality Measures	[Subaddress Element Z Level Measure] [Range Domain Measure]
Quality Notes	

1212

1213 ### 2.10.6.8 Location Description

[Element Name]	**[Location Description]**
Other common names for this element	Additional Location Information
Definition	A text description providing more detail on how to identify or find the addressed feature.
Definition Source	New
Data Type	characterString
Existing Standards for this Element	None
Domain of Values for this	No

Element	
Source of Values	None
How Defined (e.g., locally, from standard, other)	Locally
Example	"White house at intersection.", "400 yards west of water tank."
Notes/Comments	
XML Tag	<LocationDescription>
XML Model	<xsd:simpleType name="LocationDescription_type"> <xsd:restriction base="xsd:string"></xsd:restriction> </xsd:simpleType>
XML Example	<LocationDescription>White house at intersection</LocationDescription>
Quality Measures	Location Description Field Check Measure
Quality Notes	

1214

1215 ### 2.10.6.9 Mailable Address

Element Name	Mailable Address
Other common names for this element	
Definition	Identifies whether an addresses receives USPS mail delivery (that is, the address is occupiable, and the USPS provides provides on-premises USPS mail delivery to it).
Data Type	characterString
Existing Standards for this Element	None
Domain of Values for this Element	Yes, No, Unknown
Source of Values	New
How Defined (e.g., locally, from standard, other)	New definition
Example	1391 North Oak Street (apartment building): Mailable Address = **Yes** 645 Maine Avenue (vacant lot): Mailable Address = **No**

	701 Lee Street (business): Mailable Address = **Yes** 703 Lee Street (vacant storefront): Mailable Address = **Yes** 1440 Golden Gate Avenue (recreational field, no structures): Mailable Address = **No** 6813 Homestead Road (residence, in USPS home delivery area): Mailable Address = **Yes** 49984 Aspen Road (residence, outside USPS home delivery area): Mailable Address = **No**
Notes/Comments	1. The Mailable Address attribute indicates whether USPS mail will or will not be delivered to the address. This attribute is useful in determining where not to send notices or correspondence via USPS mail. 2. Postal Delivery Address Class addresses (e.g., PO Box, RD Route, and General Delivery addresses) all have a Mailable Address value = **Yes**, except in unusual circumstances such as the temporary closure of a Post Office. 3. There are many addressed, occupied features, including residences, businesses, and other features which have been addressed to facilitate the provision of E-911 and non-emergency services, and for other types of premises-based delivery services, but which are not served by premises-based USPS delivery. It is important that these location (situs) addresses not be confused with mailable addresses. The thoroughfare addresses assigned to these features, while appearing to be mailable, would be Mailable Address = **No**. 4. In verifying which addresses are not mailable, it should further be noted that the USPS ZIP+4 address validation service only validates street name and address range to a ZIP Code. Thus a vacant, addressed parcel would potentially validate as mailable if it fell within an address range on a street that was verified within the ZIP Code. 5. There are many addressed features where USPS mail cannot be delivered: vacant lots, pumping stations, parking lots, structures under construction or destroyed by disaster, and undeveloped parklands, for example. These addresses would have a Mailable Address = **No**. 6. In addition, many addresses are in areas where the USPS delivers mail to a PO Box, Rural Route Box, or General Delivery address, not to the premises address. These premise addresses also would have a Mailable Address = **No**. 7. The Mailable Address attribute can also be used to identify addresses where mail delivery has been temporarily suspended due to a large-scale natural disaster or other event. 8. The Mailable Address attribute is not intended for tracking normal vacancies due tenant turnover or change in ownership. It should be set to "No" only if mail cannot be delivered because of USPS delivery rules or long-term physical conditions at the address.
XML Tag	<MailableAddress>

XML Model	<xsd:simpleType name="MailableAddress_type"> <xsd:restriction base="xsd:string"> <xsd:pattern value='.*' /> <xsd:enumeration **value="Yes"** > <xsd:annotation> <xsd:documentation>The USPS delivers mail to this address.</xsd:documentation> </xsd:annotation></xsd:enumeration> <xsd:enumeration **value="No"** > <xsd:annotation> <xsd:documentation>The USPS does not deliver mail to this address.</xsd:documentation> </xsd:annotation></xsd:enumeration> <xsd:enumeration **value="Unknown"** > <xsd:annotation> <xsd:documentation>It is unknown whether the USPS delivers mail to this address.</xsd:documentation> </xsd:annotation></xsd:enumeration> </xsd:restriction> </xsd:simpleType>
XML Example	<MailableAddress>Yes</MailableAddress>

2.10.7 Element Attributes

2.10.7.1 Address Number Parity

Element Name	**Address Number Parity**
Other common names for this element	
Definition	The property of an Address Number with respect to being odd or even.
Definition Source	Adapted from MerriamWebster's Dictionary
Data Type	characterString
Existing Standards for this Element	NA
Domain of Values for this Element	"odd", "even"

Source of Values	NA
How Defined (e.g., locally, from standard, other)	Defined in integer mathematics.
Notes/Comments	1. Address Number Parity applies to individual Address Numbers only. Address Range Parity shows the Address Number Parity values for the Address Numbers within a range. 2. Odd and even addresses are usually associated with opposite sides of a street. For example, a jurisdiction may consistently assign odd numbers to the "left" side of its streets and even numbers to the "right" side. ("Left" and "right" would be defined with reference to the address schema.) 3. A Complete Address Number with an Address Number Suffix has the same parity as the Address Number alone. For example, 610 and 610A are both even; 611 and 611 1/2 are both odd.
XML Tag	AddressNumberParity
XML Model	<xsd:simpleType name="AddressNumberParity_type"> <xsd:restriction base="xsd:token"> <xsd:enumeration value="Even" /> <xsd:enumeration value="Odd" /> </xsd:restriction> </xsd:simpleType>
XML Example	<CompleteAddressNumber **AddressNumberParity="even"** > <AddressNumber>456</AddressNumber> <AddressNumberSuffix separator=" ">B</AddressNumberSuffix> </CompleteAddressNumber>
Quality Measure	Address Number Parity Measure
Quality Notes	

2.10.7.2 Attached Element

Element Name	**Attached Element**
Other common names for this element	
Definition	This attribute identifies when two or more Complete Address Number elements or two or more Complete Street Name elements have been combined without a space separating them.

Definition Source	New
Data Type	characterString
Required Element	No
Existing Standards for this Element	None
Domain of Values for this Element	Attached, Not Attached, Unknown
Source of Values	New
How Defined (e.g., locally, from standard, other)	New
Example	**121E** E Street (**Attached**) 121 **E** E Street (**Not Attached**) Banhoffstrasse (**Attached**) Banhoff Street (**Not Attached**)
Notes/Comments	1. The Attached Element attribute can be used to indicate that two or more Complete Address Number elements or two or more Complete Street Name elements have been combined with no space between them, so that the parsing and construction of the elements can be managed correctly. 2. Complete Address Numbers are often written with no space between the Address Number and the Address Number Prefix or Address Number Suffix (e.g., 121E E Street). The Attached Element can be used to indicate where the space is omitted as a standard practice. 3. German-language street names words are often written as a single word, combining the Street Name and Street Name Post Type (e.g., Banhoffstrasse). The Attached Element can be used to indicate such names. Attached Elements are rare in the United States street names, and normally this attribute will not be needed. In such cases the entire single word can be placed in the Street Name field, and the street type field can be left blank. This is typically done with the Street Name Street Name Post Type combination "Broadway".
XML Tag	AttachedElement
XML Model	<xsd:simpleType name="AttachedElement_type"> <xsd:restriction base="xsd:string"> <xsd:enumeration value="Attached"> <xsd:annotation> <xsd:documentation>The elements inside the Complete Address Number or Complete Street Name are attached and need special parsing rules.</xsd:documentation> </xsd:annotation></xsd:enumeration> <xsd:enumeration value="Not Attached"></xsd:enumeration>

	</xsd:restriction> </xsd:simpleType>
XML Example	<CompleteAddressNumber Address Number Parity="even" **AttachedElement="Attached"** > <AddressNumber>456</AddressNumber> <AddressNumberSuffix separator=" ">B</AddressNumberSuffix> </CompleteAddressNumber>
Quality Measures	Check Attached Pairs Measure Complete Street Name Tabular Domain Measure
Quality Notes	Check Attached Pairs Measure checks for adjacent pairs of attached attributes. The value of the street name as a whole, including the attached components are checked in the Complete Street Name Tabular Domain Measure.

2.10.7.3 Subaddress Component Order

Element Name	**Subaddress Component Order**
Other common names for this element	None
Definition	The order in which Subaddress Type and Subaddress Identifier appear within an Subaddress Element
Definition Source	New
Data Type	Integer
Existing Standards for this Element	None
Domain of Values for this Element	1 = Subaddress Type first, then Subaddress Identifier (or: Subaddress Element does not include an Subaddress Type). 2 = Subaddress Identifier first, then Subaddress Type. 3 = Not stated.
Source of Values	New
How Defined (e.g., locally, from standard, other)	Within this standard
Example	1. Room 212 (Subaddress Component Order = **1** = "Room" (the type) **precedes** "212" (the identifier)) 2. Empire Room (Subaddress Component Order = **2** = "Room" (the type)

	follows "Empire" (the identifier)) 3. Mezzanine (Subaddress Component Order = **1** = "Mezzanine" (the identifier) only; no type is given.) 4. Floor 5 (Subaddress Component Order = **1** = "Floor" (the type) **precedes** "5" (the identifier)) 5. Fifth Floor (Subaddress Component Order = **2** = "Floor" (the type) **follows** "Fifth" (the identifier)) 6. Terrace Ballroom (Subaddress Component Order = **2** --this would refer to a ballroom, the "Terrace" ballroom) 7. Ballroom Terrace (Subaddress Component Order = **2** --this would refer to a terrace, the "Ballroom" terrace)
Notes/Comments	1. This attribute tells data users how to construct an Subaddress Element from its component Subaddress Type and Subaddress Identifier. There are three possibilities, described below. The order is usually obvious for any given record, but if there are a large number of records it may not be feasible to examine each record individually. This attribute supports automated procedures for composing Subaddress Elements. 2. Usually a Subaddress Element is composed of a Subaddress Type followed by a Subaddress Identifier (e.g. "Room 212", "Floor 5") 3. However, if the Subaddress Identifier is a name or an ordinal number, it typically **precedes** the Subaddress Type (e.g. "Empire Room", "Fifth Floor") 4. Occasionally a Subaddress Element includes only a Subaddress Identifier (e.g. "Mezzanine", "Penthouse", "Rear"). These cases are grouped under Type 1. 5. Usually the component order is obvious upon examination, but ambiguous cases occur, such as "Terrace Ballroom" and "Ballroom Terrace" above. In these cases the order can be determined only by field examination or reference to authoritative records.
XML Tag	SubaddressComponentOrder
XML Model	<xsd:simpleType name="SubaddressComponentOrder_type"> <xsd:restriction base="xsd:integer"> <xsd:enumeration value="1"> <xsd:annotation> <xsd:documentation>SubaddressType first, then Subaddress Identifier (or: Subaddress Element does not include an Subaddress Type). Example: "Floor 7"</xsd:documentation> </xsd:annotation></xsd:enumeration> <xsd:enumeration value="2"> <xsd:annotation> <xsd:documentation>SubaddressIdentifier first, then Subaddress Type. Example: "Empire Room"</xsd:documentation> </xsd:annotation></xsd:enumeration> <xsd:enumeration value="3">

	<xsd:annotation> <xsd:documentation>Order is not known or unstated.</xsd:documentation> </xsd:annotation></xsd:enumeration> </xsd:restriction> </xsd:simpleType>
XML Example	<CompleteSubaddress> <SubaddressElement Element Sequence Number="1" **"SubaddressComponentOrder="1"** > <SubaddressType>Building</SubaddressType> <SubaddressIdentifier>A</SubaddressIdentifier> </SubaddressElement> <SubaddressElement Element Sequence Number="1" **SubaddressComponentOrder="2"** > <SubaddressType>Room</SubaddressType> <SubaddressIdentifier>Empire</SubaddressIdentifier> </SubaddressElement> </CompleteSubaddress>
Quality Measures	Tabular Domain Measure Subaddress Component Order Measure
Quality Notes	

2.10.7.4 Element Sequence Number

Element Name	**Element Sequence Number**
Other common names for this element	
Definition	The order in which the Subaddress Elements should be written within a Complete Subaddress; the order in which the Landmark Names should be written within a Complete Landmark Name; or the order in which the Place Names should be written within a Complete Place Name.
Definition Source	New
Data Type	Integer
Existing Standards for this Element	None
Domain of Values for this Element	Positive integers

Source of Values	Locally determined
How Defined (e.g., locally, from standard, other)	Locally
Example	For the Complete Place Name "Sun Valley, San Rafael, Marin County," the Place Name elements would have the following Element Sequence Numbers: Sun Valley: Element Sequence Number= **1** San Rafael: Element Sequence Number= **2** Marin County: Element Sequence Number= **3**
Notes/Comments	1. Complete Subaddresses, Complete Landmark Names, or Complete Place Names can include more than one element. When that occurs, the Element Sequence Number shows the order in which the components should be assembled. 2. If the Element Sequence Number is omitted, is the sequence presumed to be unknown or irrelevant.
XML Tag	ElementSequenceNumber
XML Model	<xsd:simpleType name="ElementSequenceNumber_type"> <xsd:restriction base="xsd:integer" /> </xsd:simpleType>
XML Example	<CompleteLandmark Separator=","> <LandmarkName **ElementSequenceNumber="1"** >CAMP CURRY</LandmarkName> <LandmarkName **ElementSequenceNumber="2"** >YOSEMITE NATIONAL PARK</LandmarkName> </CompleteLandmark>
Quality Measures	Related Element Uniqueness Measure Uniqueness Measure Element Sequence Number Measure
Quality Notes	

2.10.7.5 Place Name Type

Element Name	**Place Name Type**
Other common names for this element	Type of Place Name
Definition	The type of Place Name used in an Address

Definition Source	The element definition is new. The definitions of the specific examples given below (community, municipal, etc.) are new and partly adapted from: 1. FGDC's "Framework Data Content Standard Part 5: Governmental unit and other geographic area boundaries"; and, 2. USPS Publication 28, Section 292, "Urbanization".
Data Type	characterString
Existing Standards for this Element	None
Domain of Values for this Element	Community, Municipal, USPS, County, Region, Unknown. Additional values may be created as needed.
Source of Values	Locally determined
How Defined (e.g., locally, from standard, other)	**Community**: The name of an area, sector, or development, such as a neighborhood or subdivision in a city, or a rural settlement in an unincorporated area, that is not an incorporated general-purpose local government or county. The name may arise from official recognition or from popular usage. **Municipal**: The name of the general-purpose local government (if any) where the address is physically located. **USPS**: The name assigned to the post office from which the USPS delivers mail to the address. **County**: the county or county equivalent where the address is physically located. **Region**: The name of the region where the address is physically located. Typically this is name of the central city within the region. For precise, systematic terms, Census terms and definitions may be applied, but popular usage is often imprecise and to some extent subjective. **Unknown**: The Place Name Type is not known.
Example	A part of the Regent Square neighborhood is within Swissvale Borough, just outside the city limits of Pittsburgh, PA. It is served by the Wilkinsburg post office. The following place names might be used for this part of the neighborhood: **Community:** Regent Square **Municipal:** Swissvale **USPS:** Wilkinsburg **MSAG:** Swissvale **County:** Allegheny **Region:** Pittsburgh
Notes/Comments	1. Place Name Type is an attribute of the Place Name element. It is used to show what kind of place name is given for the address.
XML Tag	PlaceNameType
XML Model	<xsd:simpleType name="PlaceNameType_type">

```
<xsd:restriction base="xsd:string">
<xsd:enumeration value="Community" >
<xsd:annotation>
<xsd:documentation xml:lang="en">
The name of an area, sector, or development, such as a neighborhood or subdivision in a city, or a rural settlement in an unincorporated area, that is not an incorporated general-purpose local government or county. The name may arise from official recognition or from popular usage.
</xsd:documentation>
</xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="USPS" >
<xsd:annotation>
<xsd:documentation xml:lang="en">
The name assigned to the post office from which the USPS delivers mail to the address.
</xsd:documentation>
</xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="Municipal" >
<xsd:annotation>
<xsd:documentation xml:lang="en">
The name of the general-purpose local government (if any) where the address is physically located.
</xsd:documentation>
</xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="County" >
<xsd:annotation>
<xsd:documentation xml:lang="en">
the county or county equivalent where the address is physically located.
</xsd:documentation>
</xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="Region" >
<xsd:annotation>
<xsd:documentation xml:lang="en">
The name of the region where the address is physically located. Typically this is name of the central city within the region. For precise, systematic terms, Census terms and definitions may be applied, but popular usage is often imprecise and to some extent subjective.
</xsd:documentation>
</xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="Unknown" >
```

	<xsd:annotation> <xsd:documentation xml:lang="en"> The PlaceNameType is not known. </xsd:documentation> </xsd:annotation> </xsd:enumeration> <xsd:pattern value=".+"></xsd:pattern> </xsd:restriction> </xsd:simpleType>
XML Example	<PlaceName **PlaceNameType="County"** >Shelby</PlaceName> <PlaceName **PlaceNameType="USPS"** >Washington</PlaceName> <PlaceName **PlaceNameType="Community"** >Urbanizacion Los Olmos</PlaceName>
Quality Measures	To be written locally
Quality Measures	Place Name Type classifications are locally determined. Validation routines should be written to test against local rules.

2.10.7.6 GNIS Feature ID

Element Name	**GNIS Feature ID**
Other common names for this element	(Obsolete) FIPS Codes for populated places (FIPS 5-5), counties (FIPS 6-4), and states (FIPS 5-2) (all subsumed and superseded by GNIS Feature ID)
Definition	"A permanent, unique number assigned to a geographic feature for the sole purpose of uniquely identifying that feature as a record in any information system database, dataset, file, or document and for distinguishing it from all other feature records so identified. The number is assigned sequentially (highest existing number plus one) to new records as they are created in the Geographic Names Information System."
Definition Source	Geographic Names Project, USGS, 523 National Center, Reston, VA 20192-0523, as posted August 25, 2009 at: http://geonames.usgs.gov/domestic/metadata.htm "Feature Identifier"
Data Type	Integer
Existing Standards	U.S. Geological Survey, 19810501, U.S. Geographic Names Information

for this Element	System (GNIS): U.S. Geological Survey, Reston, VA.
Domain of Values for this Element	Integers from 1 to 9,999,999,999 inclusive.
Source of Values	U.S. Geological Survey, 19810501, U.S. Geographic Names Information System (GNIS): U.S. Geological Survey, Reston, VA. Accessible at: http://geonames.usgs.gov/domestic/index.html
How Defined (e.g., locally, from standard, other)	Assigned within U.S. Geographic Names Information System (GNIS)
Example	**531676** - United States Department of the Interior Building, Washington DC **1658360** - Curry Village, Yosemite National Park, CA (Old FIPS55 Place Code: 17638) **1248001** - Florence County, SC (Old FIPS55 Place Code: 99041)
Notes/Comments	1. “The Geographic Names Information System (GNIS) is the Federal and national standard for geographic nomenclature. The U.S. Geological Survey developed the GNIS in support of the U.S. Board on Geographic Names as the official repository of domestic geographic names data, the official vehicle for geographic names use by all departments of the Federal Government, and the source for applying geographic names to Federal electronic and printed products. “The GNIS contains information about physical and cultural geographic features of all types in the United States, associated areas, and Antarctica, current and historical, but not including roads and highways. The database holds the Federally recognized name of each feature and defines the feature location by state, county, USGS topographic map, and geographic coordinates. Other attributes include names or spellings other than the official name, feature designations, feature classification, historical and descriptive information, and for some categories the geometric boundaries. “… The GNIS collects data from a broad program of partnerships with Federal, State, and local government agencies and other authorized contributors, and provides data to all levels of government, to the public, and to numerous applications through a web query site, web map and feature services, file download services, and customized files upon request.” (Quoted August 25, 2009 from http://geonames.usgs.gov/domestic/index.html) 2. "The [GNIS Feature Identifier] number, by design, carries no information or association to the content of the feature record and therefore is not subject to change as attribute values change. Once assigned to a feature, the number is never changed or withdrawn, and never reassigned. The Feature ID can be applied in conjunction with system-unique record identifiers in any database or system, thus providing a national standard common reference identifier across multiple datasets. The Feature ID is stored in the GNIS database as an integer with a maximum of ten digits. (Source: Geographic Names Project, USGS, 523 National Center, Reston, VA 20192-0523.)" (Quoted August 25, 2009 from:

	http://geonames.usgs.gov/domestic/metadata.htm "Feature Identifier") 3. The Board of Geographic Names has set forth its principles, policies, and procedures for recognizing and standardizing domestic geographic names in its "Principles, Policies, and Procedures," posted at: http://geonames.usgs.gov/domestic/policies.htm 4. In the context of the address standard, GNIS Feature ID is applicable primarily to Landmark Names, Place Names and State Names. GNIS also includes the names of natural features, which are generally outside the scope of the address standard. 5. The Board of Geographic Names seeks to include in GNIS all feature names of public interest. Local authorities are encouraged to submit local feature names that are not already included in GNIS. 6. GNIS offers useful guidance to address authorities in selecting one name as a standard where several variants exist. GNIS Feature ID's, if assigned to Landmark Names or Place Names, can help reconcile minor name variations that can frustrate computer matches (e.g., DeKalb, Dekalb, De Kalb). GNIS Feature ID's also provide a way to link a preferred local variant name to a nationally-recognized standard. 7. GNIS provides a primary location point (x, y coordinate) for each feature. The GNIS primary point will in many cases differ from address coordinates assigned to the same feature by the addressing authority, due to differences in procedure and precision. GNIS procedures are described at: http://geonames.usgs.gov/domestic/metadata.htm "Primary Point".
XML Tag	GNISFeatureID
XML Model	<xsd:simpleType name="GNISFeatureID_type"> <xsd:restriction base="xsd:integer" /> </xsd:simpleType>
XML Example	<CompleteLandmark Separator=","> <LandmarkName ElementSequenceNumber="0" **GNISFeatureID="1658360"** >CURRY VILLAGE</LandmarkName> <LandmarkName Element Sequence Number="1">YOSEMITE NATIONAL PARK</LandmarkName> </CompleteLandmark>
Quality Measures	Related Not Null Measure
Quality Notes	

2.10.7.7 Delivery Address Type

Element Name	**Delivery Address Type**

Other common names for this element	
Definition	Whether the Delivery Address includes or excludes the Complete Subaddress.
Definition Source	New
Data Type	characterString
Existing Standards for this Element	None
Domain of Values for this Element	**Subaddress Included** - The Delivery Address **includes** the Complete Subaddress (if any) **Subaddress Excluded** - The Delivery Address **excludes** the Complete Subaddress (if any) **Unstated** - Not stated/no information (default value)
Source of Values	New
How Defined (e.g., locally, from standard, other)	Defined herein.
Example	Delivery Address = 123 Main Street, Apt. 1 (Delivery Address Type = **Subaddress Included**) Delivery Address = 123 Main Street Complete Subaddress = Apt. 1 (Delivery Address Type = **Subaddress Excluded**) Delivery Address = Ames High School, Room 12 (Delivery Address Type = **Subaddress Included**) Delivery Address = Ames High School Complete Subaddress = Room 12 (Delivery Address Type = **Subaddress Excluded**)
Notes/Comments	1. The Delivery Address typically includes the Complete Subaddress. However, there are sometimes reasons to omit or separate the Complete Subaddress from the Delivery Address. For example, the Complete Subaddress can hamper address geocoding, and contact lists often separate the Complete Subaddress from the rest of the feature address (see, for example, the EPA Contact Information Data Standard). 2. The Delivery Address Type shows whether the Delivery Address includes or excludes the Complete Subaddress. 3. If all the records in a file have the same Delivery Address Type, this information can be included in the file-level metadata. If records of different types are likely to be mixed together, the Delivery Address Type should be included in each record.
XML Tag	DeliveryAddressType
XML Model	<xsd:simpleType name="DeliveryAddressType_type">

	<xsd:restriction base="xsd:token"> <xsd:enumeration **value='SubAddress Included'** > <xsd:annotation> <xsd:documentation>The Delivery Address includes the Complete Subaddress (if any) </xsd:documentation></xsd:annotation></xsd:enumeration> <xsd:enumeration **value='SubAddress Excluded'** > <xsd:annotation> <xsd:documentation>The Delivery Address includes the Complete Subaddress (if any) </xsd:documentation></xsd:annotation></xsd:enumeration> <xsd:enumeration **value='Unstated'** > <xsd:annotation> <xsd:documentation>Not stated/no information (default value) </xsd:documentation> </xsd:annotation></xsd:enumeration> </xsd:restriction> </xsd:simpleType>
XML Example	<DeliveryAddress **DeliveryAddressType="Subaddress Included"** >123 Dartmouth College Highway, Suite 100</DeliveryAddress> <DeliveryAddress **DeliveryAddressType="Subaddress Excluded"** >123 Dartmouth College Highway, Suite 100</DeliveryAddress>
Quality Measures	Tabular Domain Measure Delivery Address Type Subaddress Measure
Quality Notes	

2.10.8 Address Lineage Attributes

2.10.8.1 Address Start Date

Element Name	**Address Start Date**
Other common names for this element	
Definition	The earliest date on which the address is known to exist.
Definition Source	New

Data Type	Date
Existing Standards for this Element	For representation of dates: YYYYMMDD (Year-month-date)(ISO 8601:2004 and FGDC CSDGM:1998).
Domain of Values for this Element	May be created locally
Source of Values	Local records
How Defined (e.g., locally, from standard, other)	Locally
Example	20050415
Notes/Comments	1, The Address Start Date is record-level metadata that should be stored for each address. 2. Changes to the Complete Address Number values or to the Complete Street Name values warrant retirement and a "new" address. 3. Changes to the values contained in Complete Subaddress, Place Name, and ZIP Code do not necessarily warrant a "new" address. 4, Therefore, the Complete Address Number and the Complete Street Name, and the Place Name, and ZIP Code elements should each have their own start dates and end dates, separate from the address start/end dates, and the dataset start/end dates. The simple elements that make up the Complete Address Number and Complete Street Name do not need to have individual start/end dates. 5. An address start date is not assigned until the Address Lifecycle Status is "proposed" or "active". The start date is generally the date on which the address authority assigns or reserves the address for use. As a rule this should be done as early as possible in the development process, generally upon subdivision or issuance of the intial building permit. 6. By definition, an address with a Address Lifecycle Status of "potential" has no Address Start Date. 7. Dates are stored in many different ways by various software programs, typically as an integer showing the number of days since some arbitrary beginning date, and converted upon display to a format that people can read. This standard does not prescribe how software should create or handle dates internally. However, for display and exchange of dates, this standard prescribes the YYYYMMDD format specified in ISO 8601:2004 and in the FGDC Content Standard for Digital Geospatial Metadata (v2, 1998). The standard is unambiguous and easily-understood, it is recognized nationally and internationally, and it can be extended if needed to include hours, minutes and seconds.
XML Tag	<AddressStartDate>
XML Model	<xsd:simpleType name="AddressStartdDate_type"> <xsd:restriction base="xsd:date" />

	</xsd:simpleType>
XML Example	<AddressStartDate>19950517</AddressStartDate>
Quality Measures	Start End Date Order Measure Future Date Measure
Quality Notes	

2.10.8.2 Address End Date

Element Name	**Address End Date**
Other common names for this element	
Definition	The date on which the address is known to no longer be valid.
Definition Source	New
Data Type	Date
Existing Standards for this Element	For representation of dates: YYYYMMDD (Year-month-date)(ISO 8601:2004 and FGDC CSDGM:1998).
Domain of Values for this Element	May be created locally
Source of Values	Local records
How Defined (e.g., locally, from standard, other)	Locally
Example	20080726
Notes/Comments	1. The Address End Date is record-level metadata that should be stored for each address. 2. Changes to the Complete Address Number value or to the Complete Street Name value warrant retirement and a "new" address. 3. Changes to the values contained in Complete Subaddress, Place Name, and ZIP Code do not necessarily warrant a "new" address. 4. Therefore, the Complete Address Number and the Complete Street Name, and the Place Name, and ZIP Code elements should have start dates and end dates for the element itself, separate from the dataset start/end dates. The simple elements that make up the Complete Address Number and Complete Street Name do not need to have individual start/end dates. 5. An address is given an end date when the Address Authority retires it. 6. If the Address Lifecycle Status is potential, proposed or active, then the

	Address End Date must be null. If the Address Lifecycle Status is retired, then the address or name must have an Address End Date. 7. Dates are stored in many different ways by various software programs, typically as an integer showing the number of days since some arbitrary beginning date, and converted upon display to a format that people can read. This standard does not prescribe how software should create or handle dates internally. However, for display and exchange of dates, this standard prescribes the YYYYMMDD format specified in ISO 8601:2004 and in the FGDC Content Standard for Digital Geospatial Metadata (v2, 1998). The standard format is unambiguous and easily-understood, it is recognized nationally and internationally, and it can be extended if needed to include hours, minutes and seconds.
XML Tag	<AddressEndDate>
XML Model	<xsd:simpleType name="AddressEndDate_type"> <xsd:restriction base="xsd:date" /> </xsd:simpleType>
XML Example	<AddressEndDate>19950517</AddressEndDate>
Quality Measures	Start End Date Order Measure Future Date Measure
Quality Notes	

1236

1237
2.10.8.3 Data Set ID

Element Name	Data Set ID
Other common names for this element	
Definition	An identifier in each record of a transmitted dataset, assigned by the sender or the receiver of the dataset, to link each record of the dataset to the file-level metadata that accompanies the dataset.
Definition Source	New
Data Type	characterString
Existing Standards for this Element	None
Domain of Values for this Element	Yes
Source of Values	Assigned by the sender or the receiver of a data set.

How Defined (e.g., locally, from standard, other)	Assigned by the sender or the receiver of a data set.
Example	Dataset ID **1475**
Notes/Comments	1. The content of the file-level metadata is specified in the FGDC's Content Standard for Digital Geospatial Metadata. 2. The ID may be assigned by the sender upon transmittal of the dataset or the recipient upon receipt. 3. Normally the identifier will be numeric, but the standard does not preclude alphanumeric identifiers.
XML Tag	<DataSetID>
XML Model	<xsd:simpleType name="DataSetID_type"> <xsd:restriction base="xsd:string"> <xsd:pattern value=".*"></xsd:pattern> </xsd:restriction> </xsd:simpleType>
XML Example	<DataSetID>1457</DataSetID>
Quality Measures	Related Not Null Measure
Quality Notes	

3. Address Data Classification

3.1 Introduction

3.1.1 Basis for Classification

The classification part of this standard classifies addresses according to their syntax, that is, their address elements and the order in which the elements are arranged. Syntax determines the record structure needed to hold and exchange an address, and often it is all that is known about the addresses in a given file.

Classifying addresses by syntax rather than semantics (i.e. meaning) allows the users of the standard to focus on record structures, and to avoid making any assumptions about what Address Feature Type the address might identify. Classifying addresses by Address Feature Type can be frustrating or impossible because:

1. Reliable information about an address may be unavailable.
2. Often, one address is used to identify several types of features (e.g., parcel, building, building entrance, utility meter, utility pole, incident location, etc.) at the same location.
3. Address Feature Type categories may be found to be ambiguous or incomplete when applied to a given address.

3.1.2 Organization

The classes are presented in four broad groups:

1. **Thoroughfare addresses** specify a location by reference to a thoroughfare.
2. **Landmark addresses** specify a location by reference to a named landmark.
3. **Postal delivery addresses** specify points of postal delivery which have no definite relation to the location of the recipient, such as post office boxes, rural route boxes, overseas military addresses, or general delivery offices.
4. The **general address class** may include addresses from any or all of the other classes, or addresses whose class is unknown, or whose syntax does not conform to any of the thoroughfare, landmark, and postal classes.

Each class is described by giving its:

1. **Name:** The name of the class.
2. **Syntax:** The address elements required and permitted in the class, and the order in which they are arranged.
3. **Defining Characteristics:** The elements and arrangement that distinguish this class from the other classes.
4. **Examples:** Illustrative examples of the class.
5. **Notes:** Explanatory notes about the class.
6. **XML Tag:** The XML tag for the class.
7. **XML Model:** XML model of the class.
8. **XML Example:** The XML model applied to a specific example of the class.
9. **XML Notes:** Explanatory notes about the XML model.
10. **Quality Measures:** Data quality tests applied to the class.
11. **Quality Notes:** Explanatory notes about the data quality measures applied to this class.

3.1.3 Formatting Conventions

Syntax and Formatting. The following notation is used to show how classes are constructed from elements:

{} enclose the name of an element.

***** indicates that the element is **required** in addresses of that class. Otherwise the element may be omitted when desired.

+ indicates "and" (concatenation), with a space implied between each component unless stated otherwise.

Example: { Complete Address Number *}+{ Complete Street Name *}+{ Complete Subaddress }

Complex Elements Include All Combinations of Their Component Elements. To avoid a multiplicity of insignificant permutations and combinations, complex elements are used to represent the various combinations of the simple elements that comprise them. Thus, for example, {CompleteAddressNumber} includes all of the following combinations:

- { Address Number *}
- { Address Number* } + { Address Number Suffix }
- { Address Number* } + { Separator Element } + { Address Number Suffix }
- { Address Number Prefix } + { Address Number *}
- { Address Number Prefix } + { Separator Element } + { Address Number *}
- { Address Number Prefix } + { Address Number *} + { Address Number Suffix }
- { Address Number Prefix } + { Separator Element } + { Address Number *} + { Address Number Suffix }
- { Address Number Prefix } + { Address Number *} + { Separator Element } + { Address Number Suffix }
- { Address Number Prefix } + { Separator Element } + { Address Number *} + { Separator Element } + { Address Number Suffix)

Place State ZIP **is Shown in Parsed Form.** In each class syntax pattern, the Complete Place Name, State Name, ZIP Code, ZIP Plus 4, and Country Name. are shown separately. They could also be shown in their unparsed form as the Place State ZIP element. However, the elements are shown separately in each syntax pattern, to emphasize that the importance of each separate element in the address.

XML Notation and Formatting. XML models and examples conform to the W3C XML Core Working Group's "Extensible Markup Language (XML) 1.0" (see Appendix A for a complete citation).

3.2 Address Classes

3.2.1 Thoroughfare Address Classes

A thoroughfare address specifies a location by reference to a thoroughfare. A thoroughfare in this context is a road or other access route by which the addressed feature can be reached (definition adapted from Universal Postal Union, *"International Postal Address Components and Templates"*, Publication S42-4 (approved July 6, 2004), section 5.2.9). A thoroughfare is typically but not always a road — it may be, for example, a walkway, a railroad, or a river. In most but not all addresses the thoroughfare is designated by a Complete Street Name and sites or features along the thoroughfare are designated in sequence by their Complete Address Number.

3.2.1.1 Numbered Thoroughfare Address

Syntax: { Complete Address Number *** } + {** Complete Street Name *** } + {** Complete Subaddress **} + {** Complete Place Name *** } + {** State Name *** } + {** ZIP Code **} + {** ZIP Plus 4 **} + {** Country Name **}**

Defining Characteristics:

1. Addresses of this class must include a Complete Address Number and a Complete Street Name.
2. In addition, all thoroughfare, landmark, and postal addresses must include a Place Name and a State Name. A ZIP Code is recommended but not mandatory.

Examples:

- 123 Main Street Buffalo Lake MN 55314
- 123 Main Street Apt 3A Buffalo Lake MN 55314
- 123 North Main Street Le Sueur MN 56058
- 123A North Main Street Le Sueur MN 56058
- 123 South Avenue C Cheyenne WY 82007
- A123 Calle B Ponce PR 00716-2525
- 123 Boulevard of the Allies Pittsburgh PA 15222-1613
- 123 Camino de la Placitas Taos NM 87571
- Mile Post 142.5, Sterling Highway, Happy Valley, AK 99639

- White House, 1600 Pennsylvania Avenue, Washington DC 20500
- Heinz Hall, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh PA 15217
- Standard Office Building, Suite 400, 600 North Milwaukee Street, Milwaukee, WI 53202
- Urbanizacion Las Gladiolas, 150 Calle A, San Juan PR 00926-3232
- Carver Park, 2730 Unwin Road, Cleveland, OH 44104

Notes:

1. Most business and residential addresses are Numbered Thoroughfare Addresses.
2. Numbered Thoroughfare Addresses are sometimes preceded by Complete Landmark Names. For example:
   - White House, 1600 Pennsylvania Avenue, Washington DC 20500
   - Heinz Hall, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh PA 15217
   - Standard Office Building, Suite 400, 600 North Milwaukee Street, Milwaukee, WI 53202
3. Less commonly, Numbered Thoroughfare Addresses are preceded by Complete Place Names, such as, for example, the name of a neighborhood, housing project, or Puerto Rican urbanization:
   - Urbanizacion Las Gladiolas, 150 Calle A, San Juan PR 00926-3232
   - Carver Park, 2730 Unwin Road, Cleveland, OH 44104

4. Strictly speaking, these are hybrid addresses. Logically they can each be decomposed to two related addresses, a Numbered Thoroughfare Address, and a Landmark Address or Community Address. For that reason, the Complete Landmark Name and Complete Place Name, although permitted, are not shown in the syntax of the Numbered Thoroughfare Address.
5. If the Complete Address Number is missing, then either the address is incomplete, or the address should be classified as an Unnumbered Thoroughfare Address.
6. In Puerto Rico it is common practice to name subdivisions and neigborhoods ("urbanizacions"), number the streets within them (Calle 1, Calle 2, etc.), and assign Complete Address Numbers that duplicate Complete Address Numbers in other nearby urbanizacions. As a result a jurisdicition or postal delivery area may contain duplicate Complete Street Names and address ranges. In these cases the urbanizacion name is required to tell the duplicates apart:
   - Urbanizacion Royal Oak, 123 Calle 1, Bayamon PR 00961-0123
   - Urbanizacion Hermosillo, 123 Calle 1, Bayamon PR 00961-1212
7. Some Puerto Rican urbanizacion addresses include Complete Street Names, and some do not. Urbanizacion addresses are classified as Numbered Thoroughfare Addresses if they include a thoroughfare name. Without a thoroughfare name, they are classified as Community Addresses:
   - (Numbered Thoroughfare Address): Urbanizacion Royal Oak, 123 Calle 1, Bayamon PR 00961-0123
   - (Community Address): 1234 Urbanizacion Los Olmos, Ponce PR 00731

8. For additional information on Puerto Rican addressing see USPS "Addressing Standards for Puerto Rico and the Virgin Islands" (page 1), and also USPS Publication 28, Section 29.

XML Tag: <NumberedThoroughfareAddress>

XML Model:

```
 <xsd:complexType name="NumberedThoroughfareAddress_type">
 <xsd:annotation>
  <xsd:documentation xml:lang="en">
  The Address Class
  associated with singular locations
  referenced off of a linear feature,
  having numeric
  identifiers.
  </xsd:documentation>
 </xsd:annotation>
 <xsd:sequence>
  <xsd:element name="CompleteAddressNumber"
type="addr_type:CompleteAddressNumber_type"
  minOccurs="1" maxOccurs="1" />
  <xsd:element name="CompleteStreetName"
type="addr_type:CompleteStreetName_type"
  minOccurs="1" maxOccurs="1" />
```

```
 <xsd:element name="CompleteSubaddress"
 type="addr_type:CompleteSubaddress_type" minOccurs="0"
 maxOccurs="1" />
 <xsd:group ref="addr_type:PlaceStateZip_group" minOccurs="0"
 maxOccurs="unbounded" />
 <xsd:group ref="addr_type:AddressAttributes_group"
 minOccurs="0" maxOccurs="1" />
</xsd:sequence>
<xsd:attribute name="action" type="addr_type:Action_type"
 use="optional" />
</xsd:complexType>
```

XML Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<addr:AddressCollection version="0.4" xmlns:addr="addr"
xmlns:addr_type="addr_type" xmlns:gml="http://www.opengis.net/gml"
xmlns:smil20="http://www.w3.org/2001/SMIL20/"
xmlns:smil20lang="http://www.w3.org/2001/SMIL20/Language"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xml="http://www.w3.org/XML/1998/namespace"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="addr addr.xsd ">
 <NumberedThoroughfareAddress>
```

```
 <CompleteAddressNumber>
  <AddressNumber>123</AddressNumber>
 </CompleteAddressNumber>
 <CompleteStreetName>
  <StreetName>Main</StreetName>
  <StreetNamePostType>Street</StreetNamePostType>
 </CompleteStreetName>
 <CompletePlaceName>
  <PlaceName>Buffalo Lake</PlaceName>
 </CompletePlaceName>
 <StateName>MN</StateName>
 <ZIPCode>55314</ZIPCode>
 </NumberedThoroughfareAddress>
</addr:AddressCollection>
```

XML Notes:

Quality Measures:

Address Completeness Measure

Left Right Odd Even Parity Measure

Address Number Fishbones Measure

Range Domain Measure

3.2.1.2 Intersection Address

Syntax: { Complete Street Name *** {** Separator Element *** } } (2..n) + {** Complete Place Name ***} + {** State Name ***} + {** ZIP Code **} + {** ZIP Plus 4 **} + {** Country Name **}**

Defining Characteristics:

1. Addresses of this class must include two or more Complete Street Names, separated by Separator Elements.
2. In addition, all thoroughfare, landmark, and postal addresses must include a Place Name and a State Name. A ZIP Code is recommended but not mandatory.

Examples:

- Boardwalk and Park Place, Atlantic City, NJ
- Hollywood Boulevard and Vine Street, Hollywood, CA 90028
- West Street & Main Street, Newtown CT 06470
- P Street && 19th Street && Mill Road, Ellicott City MD 21043
- Avenida Rosa y Calle 19, Bayamon PR 00961
- Memorial Park, Last Chance Gulch and Memorial Drive, Helena, MT 59601
- Phoenix Village, Scovill Avenue and East 59th Street, Cleveland, Ohio 44104

Notes:

1. Intersection addresses are useful for recording events occurring in the street, such as accidents, infrastructure locations, etc. However, when referring to a feature on

one corner of an intersection, the Numbered Thoroughfare Address for that corner is always preferable to the intersection address.

2. A Complete Landmark Name or Complete Place Name may precede an Intersection Address. For example:

- Memorial Park, Last Chance Gulch and Memorial Drive, Helena, MT 59601
- Phoenix Village, Scovill Avenue and East 59th Street, Cleveland, Ohio 44104

3. Strictly speaking, these are hybrid addresses. Logically they can each be decomposed to two related addresses, an Intersection Address, and a Landmark Address or Community Address. For that reason, the Complete Landmark Name and Complete Place Name, although permitted, are not shown in the syntax of the Intersection Address.
4. Intersections of more than two streets can be represented as one sequence of three or more street names, or as every pairwise combination of the names.
5. Separator values include " and ", “ at “, “ @ “, " & ", and " && " " + "," - ", and " y " or " con " (Spanish) each having a space before and after. Other values may also be in use.
6. Some address parsing software permits the use of ampersands (" & " or " && ") to signify intersection addresses, because the double ampersand does not occur in any street names, and ampersands rarely do. Be wary, though--in many programming languages, ampersands are reserved for other uses, which could complicate data exchange.

XML Tag: <IntersectionAddress>

XML Model:

```
<xsd:complexType name="IntersectionAddress_type">
 <xsd:sequence>
  <xsd:element name="CompleteStreetName"
  type="addr_type:CompleteStreetName_type" minOccurs="1"
  maxOccurs="1" />
  <xsd:group ref="addr:IntersectionAddress_StreetName_group"
minOccurs="1" maxOccurs="unbounded"/>
  <xsd:group ref="addr_type:PlaceStateZip_group" minOccurs="1"
  maxOccurs="1" />
  <xsd:group ref="addr_type:AddressAttributes_group"
  minOccurs="0" maxOccurs="1" />
 </xsd:sequence>
 <xsd:attribute name="action" type="addr_type:Action_type"
  use="optional" />
 </xsd:complexType>
```

XML Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<addr:AddressCollection version="0.4" xmlns:addr="addr"
xmlns:addr_type="addr_type" xmlns:gml="http://www.opengis.net/gml"
xmlns:smil20="http://www.w3.org/2001/SMIL20/"
xmlns:smil20lang="http://www.w3.org/2001/SMIL20/Language"
```

```
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xml="http://www.w3.org/XML/1998/namespace"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="addr addr.xsd ">
 <IntersectionAddress>
 <CompleteStreetName>
  <StreetName>Boardwalk</StreetName>
 </CompleteStreetName>
 <SeparatorElement>and</SeparatorElement>
 <CompleteStreetName>
  <StreetName>Park</StreetName>
  <StreetNamePostType>Place</StreetNamePostType>
 </CompleteStreetName>
 <CompletePlaceName>
  <PlaceName>Atlantic City</PlaceName>
 </CompletePlaceName>
 <StateName>NJ</StateName>
 </IntersectionAddress>
</addr:AddressCollection>
```

Quality Measures

Intersection Validity Measure

Pattern Sequence Measure

3.2.1.3 Two Number Address Range

Syntax: { Complete Address Number **(low) *} + {** Separator Element ***} + {** Complete Address Number **(high)*} + {** Complete Street Name *** } + {** Complete Place Name *** } + {** State Name *** } + {** ZIP Code **} + {** ZIP Plus 4 **} + {** Country Name **}**

Defining Characteristics:

1. Addresses of this class must include two Complete Address Numbers separated by a hyphen. The first Complete Address Number must be less than or equal to the second.
2. The two Complete Address Numbers must be followed by a Complete Street Name.
3. In addition, all thoroughfare, landmark, and postal addresses must include a Place Name and a State Name. A ZIP Code is recommended but not mandatory.

Examples:

- 401-418 Green Street, Flint MI 48503
- 1400-1420 Smith Street, West Monroe, LA 71292
- 13-25 Elm Street, Muncie, IN 47305
- 214-02 - 214-14 1/2 Evergreen Street, New York, NY 11364
- 55A - 55H Kelly Circle SW, Bolling Air Force Base, Washington, DC

- Quincy Market, 1-47 Faneuil Hall Market Place, Boston, MA 02109

Notes:

1. The Two Number Address Range includes a set of two Complete Address Numbers, which represent the low and high values of a continuous series of Complete Address Numbers. By convention, the first Complete Address Number represents the low end of the range, and the second represents the high end, and they are separated by a hyphen.
2. Generally, but not always, if a range refers to Complete Address Numbers on one side of a thoroughfare, the Complete Address Numbers in the range will all have the same parity, that is, they will all be either odd or even. However, mixed parities do occur in some places.
3. A range can begin or end with a Complete Address Number that has a suffix or prefix. USPS Publication 28 Appendix E contains instructive notes on the complexities of these address ranges.
4. Use the Address Range Type to show whether a Two Number Address Range represents an actual or potential range.
5. Use the Address Range Parity attribute to show whether a Two Number Address Range includes Complete Address Numbers that are odd, even, or both.
6. If a Two Number Address Range is related to a transportation segment (or set of segments) in a transportation network model, then:

- The Address Range Side attribute may be used to show if the Complete Address Numbers in the range are on the right side, left side, or both sides of the segment(s).
- The Address Range Directionality attribute may be used to show if the Complete Address Numbers in the range increase with or against the directionality of the segment(s).
- The Address Range Span attribute may be used to show whether the range spans a part of one segment, one entire segment, multiple segments, or the entire length of the thoroughfare.

7. Use the Address Transportation System Name, Address Transportation System Authority, Address Transportation Feature Type, Address Transportation Feature ID, and Related Transportation Feature ID attributes to relate a particular address range to a specific transportation segment (or set of segments) in a specific transportation network model. Transportation segments, and transportation network models generally, are defined and described in the FGDC's "Geographic Information Framework Data Content Standard Part 7: Transportation Base."
8. Ranges should not be confused with hyphenated address numbers that denote a single site. A range must be composed of two Complete Address Numbers. Certain areas of New York City, southern California, and Hawaii use hyphens in Complete Address Numbers. In the example above, "214-02 Evergreen St" would be one address, and "214-14 1/2 Evergreen Street" would be a second address, and neither one alone is an address range.

9. A Two Number Address Range may be preceded by a Complete Landmark Name or Complete Place Name that spans the range. (for example: "Quincy Market, 1-47 Faneuil Hall Market Place, Boston, MA 02109"). Strictly speaking, this is a hybrid address. Logically it could be decomposed to two related addresses, the Two Number Address Range, and a corresponding Landmark Address or Community Address. For that reason, the Complete Landmark Name and Complete Place Name, although permitted, are not shown in the syntax of the Two Number Address Range.

XML Tag: <TwoNumberAddressRange>

XML Model:

```
 <xsd:complexType name="TwoNumberAddressRange_type">
 <xsd:sequence>
  <xsd:element name="CompleteAddressNumber"
type="addr_type:CompleteAddressNumber_type"
  minOccurs="1" maxOccurs="1" />
   <xsd:element name="SeparatorElement" type="addr_type:Separator_type"
maxOccurs="1" minOccurs="1"/>
  <xsd:element name="CompleteAddressNumber"
type="addr_type:CompleteAddressNumber_type"
  minOccurs="1" maxOccurs="1" />
  <xsd:element name="CompleteStreetName"
type="addr_type:CompleteStreetName_type"
```

```
  minOccurs="1" maxOccurs="1" />
  <xsd:group ref="addr_type:PlaceStateZip_group" minOccurs="1"
  maxOccurs="1" />
  <xsd:group ref="addr_type:AddressAttributes_group"
  minOccurs="0" maxOccurs="1" />
 </xsd:sequence>
 <xsd:attribute name="action" type="addr_type:Action_type"
  use="optional" />
 </xsd:complexType>
```

XML Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<addr:AddressCollection version="0.4" xmlns:addr="addr"
xmlns:addr_type="addr_type" xmlns:gml="http://www.opengis.net/gml"
xmlns:smil20="http://www.w3.org/2001/SMIL20/"
xmlns:smil20lang="http://www.w3.org/2001/SMIL20/Language"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xml="http://www.w3.org/XML/1998/namespace"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="addr addr.xsd ">
 <TwoNumberAddressRange>
 <CompleteAddressNumber>
  <AddressNumber>401</AddressNumber>
```

```
    </CompleteAddressNumber>
    <SeparatorElement>-</SeparatorElement>
    <CompleteAddressNumber>
     <AddressNumber>418</AddressNumber>
    </CompleteAddressNumber>
    <CompleteStreetName>
     <StreetName>Green</StreetName>
     <StreetNamePostType>Street</StreetNamePostType>
    </CompleteStreetName>
    <CompletePlaceName>
     <PlaceName>Flint</PlaceName>
    </CompletePlaceName>
    <StateName>MI</StateName>
    <ZIPCode>48503</ZIPCode>
    </TwoNumberAddressRange>
</addr:AddressCollection>
```

Quality Measures

Address Number Fishbones Measure

Address Number Range Completeness Measure

Address Number Range Parity Consistency Measure

Address Number Range Sequence Measure

Low High Address Sequence Measure Overlapping Ranges Measure Range Domain Measure

3.2.1.4 Four Number Address Range

Syntax: { Complete Address Number ***(left low) } + {** Complete Address Number ***(left high) } + {** Complete Address Number *** (right low) } + {** Complete Address Number *** (right high) }+{** Complete Street Name *** } + {** Complete Place Name *** } + {** State Name *** } + {** ZIP Code **} + {** ZIP Plus 4 **} + {** Country Name **}**

Defining Characteristics:

1. Addresses of this class must include four Complete Address Numbers, representing respectively the left low, left high, right low, and right high four Complete Address Numbers for the block or transportation segment(s), followed by a Complete Street Name.
2. In addition, all thoroughfare, landmark, and postal addresses must include a Place Name and a State Name. A ZIP Code is recommended but not mandatory.
3. The Four Number Address Range syntax follows the structure established by the U.S. Census Bureau for TIGER/Line file street segment address ranges (see http://www.census.gov/geo/www/tiger/tgrshp2008/TGRSHP08.pdf ("All Lines Shapefile" attribute table layout)).

Examples:

- U.S. Census Bureau TIGER file formatted address ranges (left low, left high, right low, right high, street name) are the most widely-used examples of Four Number Address Ranges.

Notes:

1. Address ranges are important for municipal operations (such as snow plow dispatch), emergency dispatch, and geocoding.
2. A Four Number Address Range includes four Complete Address Numbers, representing, for each side of a block or transportation segment, the low and high end of the Complete Address Number range. By convention, based on the attribute structure established by the U.S. Census Bureau for the TIGER/Line files, the left-side low Complete Address Number is given first, followed by the left-side high Complete Address Number, followed by the right-side low and high Complete Address Numbers.
3. Generally, but not always, the left and right ranges will have different parities (even or odd). However, mixed parities do occur in some places.
4. A range can begin or end with a Complete Address Number that has a suffix or prefix. USPS Publication 28 Appendix E contains instructive notes on the complexities of these address ranges.
5. Use the Address Range Type attribute to show whether a Four Number Address Range represents an actual or potential range.
6. Use the Address Range Parity attribute to show whether a Four Number Address Range includes Complete Address Numbers that are odd, even, or both.

7. If a Four Number Address Range is related to a transportation segment (or set of segments)in a transportation network model, , then:
   - The Address Range Side attribute may be used to show if the Complete Address Numbers in the range are on the right side, left side, or both sides of the segment(s).
   - The Address Range Directionality attribute may be used to show if the Complete Address Numbers in the range increase with or against the directionality of the segment(s).
   - The Address Range Span attribute may be used to show whether the range spans a part of one segment, one entire segment, multiple segments, or the entire length of the thoroughfare.
8. Use the Address Transportation System Name, Address Transportation System Authority, Address Transportation Feature Type, Address Transportation Feature ID, and Related Transportation Feature ID attributes to relate a particular address range to a specific transportation segment (or set of segments) in a specific transportation network model. Transportation segments, and transportation network models generally, are defined and described in the FGDC's "Geographic Information Framework Data Content Standard Part 7: Transportation Base."
9. A Four Number Address Range may be preceded by a Complete Landmark Name or Complete Place Name that encompasses the range. Strictly speaking, this would be a hybrid address. Logically it could be decomposed to two related addresses, the Four Number Address Range, and a corresponding Landmark Address or Community Address. For that reason, the Complete Landmark Name and

Complete Place Name, although permitted, are not shown in the syntax of the Four Number Address Range.

XML Tag: <FourNumberAddressRange>

XML Model:

```
<xsd:complexType name="FourNumberAddressRange_type">
<xsd:annotation>
 <xsd:documentation> TIGER file ranges (left low, left high, right
 low, right high, street name) are the most widely-used example of
 block ranges Notes: Although they do not necessarily refer to one
 specific site, block addresses are important for municipal
 operations (such as snow plow dispatch), emergency dispatch, and
 geocoding. A block address range may be expressed by four numbers,
 representing the low and high end of the numeric range for each side
 of a block. By convention, the first number represents the low end
 of the range of addresses for the left side, the second number
 represents the high end of the range of addresses for the left side,
 the third number represents the low end of the range of addresses
 for the right side, and the fourth number represents the high end of
 the range for the right side. A block face is defined as one side of
 a thoroughfare between two intersecting street segments. Generally,
 but not always, a block face has addresses of a single parity, that
 is, either odd or even numbers. However, mixed parities do occur in
```

```
            some places. In other cases, where the numeric ranges on opposite
            sides of the same block are not within the same general range, it is
            preferable to express the range in terms of the left low-high, right
            low-high, or to provide individual block face ranges. A block range
            may refer to either a theoretical range (the possible range of
            addresses along that street segment) or to an actual or used range
            of addresses. These types (actual or theoretical) are distinguished
            by the range type attribute.
              </xsd:documentation>
           </xsd:annotation>
           <xsd:sequence>
            <xsd:element name="CompleteAddressNumber"
          type="addr_type:CompleteAddressNumber_type"
            minOccurs="1" maxOccurs="1" />
             <xsd:element name="SeparatorElement" type="addr_type:Separator_type"
          maxOccurs="1" minOccurs="1"/>
            <xsd:element name="CompleteAddressNumber"
          type="addr_type:CompleteAddressNumber_type"
            minOccurs="1" maxOccurs="1" />
            <xsd:element name="CompleteAddressNumber"
          type="addr_type:CompleteAddressNumber_type"
            minOccurs="1" maxOccurs="1" />
             <xsd:element name="SeparatorElement" type="addr_type:Separator_type"
```

```
maxOccurs="1" minOccurs="1"/>
  <xsd:element name="CompleteAddressNumber"
type="addr_type:CompleteAddressNumber_type"
  minOccurs="1" maxOccurs="1" />
  <xsd:element name="CompleteStreetName"
type="addr_type:CompleteStreetName_type"
  minOccurs="1" maxOccurs="1" />
  <xsd:group ref="addr_type:PlaceStateZip_group" minOccurs="1"
  maxOccurs="1" />
  <xsd:group ref="addr_type:AddressAttributes_group"
  minOccurs="0" maxOccurs="1" />
 </xsd:sequence>
 <xsd:attribute name="action" type="addr_type:Action_type"
  use="optional" />
 </xsd:complexType>
```

XML Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<addr:AddressCollection version="0.4" xmlns:addr="addr"
xmlns:addr_type="addr_type" xmlns:gml="http://www.opengis.net/gml"
xmlns:smil20="http://www.w3.org/2001/SMIL20/"
xmlns:smil20lang="http://www.w3.org/2001/SMIL20/Language"
xmlns:xlink="http://www.w3.org/1999/xlink"
```

```
xmlns:xml="http://www.w3.org/XML/1998/namespace"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="addr addr.xsd ">
 <FourNumberAddressRange>
 <CompleteAddressNumber>
  <AddressNumber>1900</AddressNumber>
 </CompleteAddressNumber>
 <SeparatorElement>-</SeparatorElement>
 <CompleteAddressNumber>
  <AddressNumber>1908</AddressNumber>
 </CompleteAddressNumber>
 <CompleteAddressNumber>
  <AddressNumber>1901</AddressNumber>
 </CompleteAddressNumber>
 <SeparatorElement>-</SeparatorElement>
 <CompleteAddressNumber>
  <AddressNumber>1909</AddressNumber>
 </CompleteAddressNumber>
 <CompleteStreetName>
  <StreetName>Bear</StreetName>
  <StreetNamePostType>court</StreetNamePostType>
 </CompleteStreetName>
 <CompletePlaceName>
```

```
 <PlaceName>Fort Collins</PlaceName>
</CompletePlaceName>
<StateName>CO</StateName>
<ZIPCode>80525</ZIPCode>
</FourNumberAddressRange>
</addr:AddressCollection>
```

Quality Measures

Address Number Fishbones Measure
Address Number Range Completeness Measure
Address Number Range Parity Consistency Measure
Address Number Range Sequence Measure
Left Right Odd Even Parity Measure
Low High Address Sequence Measure
Overlapping Ranges Measure Range Domain Measure

3.2.1.5 Unnumbered Thoroughfare Address

Syntax: { Complete Street Name *** } + {** Complete Subaddress **} + {** Complete Place Name *** } + {** State Name *** } + {** ZIP Code **} + {** ZIP Plus 4 **} + {** Country Name **}**

Defining Characteristics:

1. Addresses of this class must contain a Complete Street Name with no Complete Address Number preceding it.
2. In addition, all thoroughfare, landmark, and postal addresses must include a Place Name and a State Name. A ZIP Code is recommended but not mandatory.

Example:

- Ili'ili Airport Road, Ili'ili, AS
- East End Road, St. Croix, VI 00820
- Ilisagvik College, Stevenson Street, Barrow, AK 99723
- Orote Point Lighthouse, San Luis Drive, Santa Rita, GU

Notes:

1. In many areas no address numbers have been assigned, and addresses in those areas often include only the thoroughfare name. This class separates those addresses from addresses that include address numbers or cross-streets.
2. An Unnumbered Thoroughfare Address may be preceded by a Complete Landmark Name or Complete Place Name (for example, "Ilisagvik College, Stevenson Street, Barrow, AK 99723"). Strictly speaking, this would be a hybrid address. Logically it can be decomposed to two related addresses, the Unnumbered Thoroughfare Address, and a corresponding Landmark Address or Community Address. For that reason, the Complete Landmark Name and Complete Place Name, although permitted, are not shown in the syntax of the Unnumbered Thoroughfare Address.

XML Tag: <UnnumberedThoroughfareAddress>

XML Model:

```
<xsd:complexType name="UnnumberedThoroughfareAddress_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">
 The Address Class
 associated with singular locations
 referenced off of a linear feature,
 lacking numeric
 identifiers.
 </xsd:documentation>
</xsd:annotation>
<xsd:sequence>
 <xsd:element name="CompleteStreetName"
type="addr_type:CompleteStreetName_type"
 minOccurs="1" maxOccurs="1" />
 <xsd:element name="CompleteSubaddress"
type="addr_type:CompleteSubaddress_type"
 minOccurs="0" maxOccurs="1" />
 <xsd:group ref="addr_type:PlaceStateZip_group" minOccurs="1"
 maxOccurs="1" />
 <xsd:group ref="addr_type:AddressAttributes_group"
 minOccurs="0" maxOccurs="1" />
</xsd:sequence>
```

```
<xsd:attribute name="action" type="addr_type:Action_type"
 use="optional" />
</xsd:complexType>
```

XML Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<addr:AddressCollection version="0.4" xmlns:addr="addr"
xmlns:addr_type="addr_type" xmlns:gml="http://www.opengis.net/gml"
xmlns:smil20="http://www.w3.org/2001/SMIL20/"
xmlns:smil20lang="http://www.w3.org/2001/SMIL20/Language"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xml="http://www.w3.org/XML/1998/namespace"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="addr addr.xsd ">
 <UnnumberedThoroughfareAddress>
 <CompleteStreetName>
  <StreetName>Fagaima</StreetName>
  <StreetNamePostType>Road</StreetNamePostType>
 </CompleteStreetName>
 <CompletePlaceName>
  <PlaceName>Nu'uli</PlaceName>
 </CompletePlaceName>
 <StateName>AS</StateName>
```

```
 <ZIPCode>96799</ZIPCode>
 </UnnumberedThoroughfareAddress>
</addr:AddressCollection>
```

Quality Measures

Pattern Sequence Measure

3.2.2 Landmark Address Classes

A landmark address specifies a location by reference to a named landmark. A landmark is a relatively permanent feature of the manmade landscape that has recognizable identity within a particular cultural context (definition adapted from U.S. Board on Geographic Names, *"Principles, Policies, Procedures,"* (Online Edition (revised)), 2003, p. 48, definition of "geographic name").

3.2.2.1 Landmark Address

Syntax: { Complete Landmark Name *** } (1..n) + {** Complete Subaddress **} + {** Complete Place Name *** } + {** State Name *** } + {** ZIP Code **} + {** ZIP Plus 4 **} + {** Country Name **}**

Defining Characteristics:

1. Addresses of this class must include a Complete Landmark Name, with no Complete Address Number preceding it and no Complete Street Name following it.
2. In addition, all thoroughfare, landmark, and postal addresses must include a Place Name and a State Name. A ZIP Code is recommended but not mandatory.

Examples:

- Statue of Liberty, New York NY 10004
- Langston Housing Complex, Building 7, Apartment 290, Kansas City KS 66101
- Condominium Garden Hills Plaza, Torre 2, Apartamento 905, Mayaguez PR 00680-1233
- Condominium Del Mar, Apartamento 905, Ponce PR 00731
- Residencial Las Margaritas, Edificio 1, Apartamento 104, San Juan PR 00924

Notes:

1. This class includes the "condominium" addresses found in Puerto Rico, where a complex or building is known by name, without reference to a street.

XML Tag: <LandmarkAddress>

XML Model:

```
<xsd:complexType name="LandmarkAddress_type">
<xsd:sequence>
 <xsd:element name="CompleteLandmarkName"
```

```
type="addr_type:CompleteLandmarkName_type"
  minOccurs="1" maxOccurs="1" />
  <xsd:element name="CompleteSubaddress"
type="addr_type:CompleteSubaddress_type"
  minOccurs="0" maxOccurs="1" />
  <xsd:group ref="addr_type:PlaceStateZip_group" minOccurs="1"
  maxOccurs="1" />
  <xsd:group ref="addr_type:AddressAttributes_group"
  minOccurs="0" maxOccurs="1" />
 </xsd:sequence>
 <xsd:attribute name="action" type="addr_type:Action_type"
  use="optional" />
 </xsd:complexType>
```

XML Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<addr:AddressCollection version="0.4" xmlns:addr="addr"
xmlns:addr_type="addr_type" xmlns:gml="http://www.opengis.net/gml"
xmlns:smil20="http://www.w3.org/2001/SMIL20/"
xmlns:smil20lang="http://www.w3.org/2001/SMIL20/Language"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xml="http://www.w3.org/XML/1998/namespace"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```xml
xsi:schemaLocation="addr addr.xsd ">
 <LandmarkAddress>
 <CompleteLandmarkName>
  <LandmarkName>Condominium Garden Hills Plaza</LandmarkName>
 </CompleteLandmarkName>
 <CompleteSubaddress>
  <SubaddressElement SubaddressComponentOrder="1">
  <SubaddressType>Torre</SubaddressType>
  <SubaddressIdentifier>2</SubaddressIdentifier>
  </SubaddressElement>
  <SubaddressElement>
  <SubaddressType>Apartamento</SubaddressType>
  <SubaddressIdentifier>905</SubaddressIdentifier>
  </SubaddressElement>
 </CompleteSubaddress>
 <CompletePlaceName>
  <PlaceName>Mayaguez</PlaceName>
 </CompletePlaceName>
 <StateName>PR</StateName>
 <ZIPCode>00608</ZIPCode>
 <ZIPPlus4>1233</ZIPPlus4>
 </LandmarkAddress>
</addr:AddressCollection>
```

Quality Measures

Pattern Sequence Measure

3.2.2.2 Community Address

Syntax: { Complete Address Number *** } + {** Complete Landmark Name **or** Complete Place Name *** } + {** Complete Subaddress **} + {** Complete Place Name *** } + {** State Name *** } + {** ZIP Code **} + {** ZIP Plus 4 **} + {** Country Name **}**

Defining Characteristics:

1. Addresses of this class must include a Complete Address Number followed by a Complete Landmark Name or a Complete Place Name, and they must not include a Complete Street Name.
2. In addition, all thoroughfare, landmark, and postal addresses must include a Place Name and a State Name. A ZIP Code is recommended but not mandatory.

Examples:

1. 1234 Urbanizacion Los Olmos, Ponce PR 00731
2. A17 Jardine Fagota, Ponce PR 00731
3. B133 Urbanizacion Golden Gate, San Juan PR 00920
4. 23B Edgewater Park, Apartment 12, Bronx, NY 10465

Notes:

1. Community Addresses may be found in gated communities, housing projects, Puerto Rican urbanizations, trailer courts, and similar developments that are built around interior walkways or roadways. Their Complete Address Numbers refer to the community name, not to a thoroughfare. The community name might be a treated as a Landmark Name or Place Name--the distinction is often arbitrary or unclear for community names.
2. If there is no Complete Address Number preceding the urbanization name, the address fits into the Landmark Address class.
3. If the address includes both a Complete Street Name and a community name, it fits in the Numbered Thoroughfare Address class.
4. This class includes Puerto Rican urbanization addresses where the urbanization name is preceded by a number, and no street name is included. In Puerto Rico, an urbanization denotes an area, sector, or residential development within a geographic area. For more information on Puerto Rican addressing conventions, see USPS Publication 28 Section 29, and USPS “Addressing Standards for Puerto Rico and the Virgin Islands”. See also the notes under Numbered Thoroughfare Address.

XML Tag: <CommunityAddress>

XML Model:

```
<xsd:complexType name="CommunityAddress_type">
<xsd:annotation>
```

```
        <xsd:documentation> 1. Community Addresses are commonly used for
        housing projects, Puerto Rican urbanizations, trailer courts, and
        similar developments that are built around unnamed interior walkways
        or roadways. Their Complete Address Numbers refer to the community
        name, not to a thoroughfare. 2. A Community Address includes a
        Complete Address Number, a community name, and a Place Name. The
        address does not include a Complete Street Name. The community name
        might be a treated as a Landmark Name or Place Name--the distinction
        is often arbitrary or unclear for community names. 3. If there is no
        Complete Address Number preceding the urbanization name, the address
        fits into the Landmark Address class. 4. If the address includes
        both a Complete Street Name and a community name, it fits in the
        Landmark Site Address class. 5. This class includes Puerto Rican
        urbanization addresses where the urbanization name is preceded by a
        number, and no street name is included. In Puerto Rico, an
        urbanization denotes an area, sector, or residential development
        within a geographic area. For more information on Puerto Rican
        addressing conventions, see USPS Publication 28 Section 29, and USPS
        “Addressing Standards for Puerto Rico and the Virgin Islands”.
        </xsd:documentation>
       </xsd:annotation>
       <xsd:sequence>
        <xsd:element name="CompleteAddressNumber"
```

```
type="addr_type:CompleteAddressNumber_type"
  minOccurs="1" maxOccurs="1" />
  <xsd:choice>
  <xsd:element name="CompleteLandmarkName"
type="addr_type:CompleteLandmarkName_type"
   minOccurs="1" maxOccurs="1" />
  <xsd:element name="CompletePlaceName"
type="addr_type:CompletePlaceName_type"
   minOccurs="1" maxOccurs="1" />
  </xsd:choice>
  <xsd:element name="CompleteSubaddress"
type="addr_type:CompleteSubaddress_type"
  minOccurs="0" maxOccurs="1" />
  <xsd:group ref="addr_type:PlaceStateZip_group" minOccurs="1"
  maxOccurs="1" />
  <xsd:group ref="addr_type:AddressAttributes_group"
  minOccurs="0" maxOccurs="1" />
 </xsd:sequence>
 <xsd:attribute name="action" type="addr_type:Action_type"
  use="optional" />
 </xsd:complexType>
```

XML Example:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<addr:AddressCollection version="0.4" xmlns:addr="addr"
xmlns:addr_type="addr_type" xmlns:gml="http://www.opengis.net/gml"
xmlns:smil20="http://www.w3.org/2001/SMIL20/"
xmlns:smil20lang="http://www.w3.org/2001/SMIL20/Language"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xml="http://www.w3.org/XML/1998/namespace"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="addr addr.xsd ">
 <CommunityAddress>
 <CompleteAddressNumber>
  <AddressNumberPrefix>A</AddressNumberPrefix>
  <AddressNumber>17</AddressNumber>
 </CompleteAddressNumber>
 <CompleteLandmarkName>
  <LandmarkName>Jardine Fagota</LandmarkName>
 </CompleteLandmarkName>
 <CompletePlaceName>
  <PlaceName>Ponce</PlaceName>
 </CompletePlaceName>
 <StateName>PR</StateName>
 <ZIPCode>00731</ZIPCode>
 </CommunityAddress>
```

```
</addr:AddressCollection>
```

Quality Measures

Pattern Sequence Measure

3.2.3 Postal Delivery Address Classes

A postal delivery address specifies a point of postal delivery that has no definite relation to the location of the recipient, such as a post office box, rural route box, overseas military address, or general delivery office. The USPS specifies each class in detail in USPS Publication 28.

Postal addresses are often combined with thoroughfare and landmark addresses. Examples:

- Landmark-Postal Address: Wagon Wheel Ranch, RR1 Box 100, Pawhuska, OK
- Postal-Thoroughfare Address: 200 South Minnesota Avenue, PO Box 1304, Sioux Falls, SD
- Landmark-Postal-Thoroughfare Address: Twin Falls Extension Center, Evergreen Building, College of Southern Idaho, PO Box 1827, 315 Falls Avenue East, Twin Falls, ID

These potential classes are not recognized in this standard because the USPS strongly discourages their use (USPS Publication 28 sections 215, 245, 255, 295.6 and 295.7). Within the standard they can be handled in two ways:

1. Separate them into their component types, create records for each, and relate the records to show that they refer to the same location.

2. Treat the entire address as a General Address Class address.

3.2.3.1 USPS Postal Delivery Box

Syntax: { USPS Box*** } + {** Complete Subaddress **} + {** Complete Place Name *** } + {** State Name *** } + {** ZIP Code **} + {** ZIP Plus 4 **} + {** Country Name **}**

USPS Box Format: USPS Box = "PO Box"* + { USPS Box ID * }

In this address class, the phrase "PO Box" is the only permittted value for USPS Box Type.

Defining Characteristics:

1. Addresses of this class must include a USPS Box in the required format, and must not include a USPS Route.
2. In addition, all thoroughfare, landmark, and postal addresses must include a Place Name and a State Name. A ZIP Code is recommended but not mandatory.

Example:

- PO BOX 16943, New Orleans LA 70112
- PO BOX 1890, Kryton TN 38188-1890
- PO BOX G, Gabbs NV 89409
- PO BOX 159753 PMB 3571, Herndon VA 22071-2716

Notes:

1. This class is defined in USPS Publication 28, Sections 281-283. The phrase "PO Box" is mandatory as the USPS Box Type.

2. USPS Pub 28 Sec. 282: “Post Office Box numbers that are preceded by significant leading zeroes are identified in the ZIP+4 File by a hyphen (-) preceding the box number. Convert the hyphen into a zero on the output mailpiece.
    - ZIP+4 File: PO BOX -0145
    - Mailpiece: PO BOX 00145"
3. USPS Pub 28 Sec. 283: “PO Box addresses often appear with the word CALLER, FIRM CALLER, BIN, LOCKBOX, or DRAWER. Change these to PO BOX.”
    - Incorrect: DRAWER L
    - Correct: PO BOX L
4. The Complete Subaddress, if it appears at all, must have only one Subaddress Element, and that Subaddress Element must have a Subaddress Type of "PMB".
5. In USPS Postal Delivery Box addresses, the Complete Place Name element may include multiple Place Names, but the USPS strongly prefers that only the postal community name be used. Example:
    - **Preferred:** Wailuku, HI
    - **Acceptable:** Wailuku, Maui, HI

XML Tag: <USPSPostalDeliveryBox>

XML Model:

```
<xsd:complexType name="USPSPostalDeliveryBox_type">
<xsd:sequence>
 <xsd:element name="USPSBox" type="addr_type:USPSBox_type"
 minOccurs="1" maxOccurs="1" />
```

```
  <xsd:element name="CompleteSubaddress"
type="addr_type:CompleteSubaddress_type"
  minOccurs="0" maxOccurs="1" />
  <xsd:group ref="addr_type:PlaceStateZip_group" minOccurs="1"
  maxOccurs="1" />
  <xsd:group ref="addr_type:AddressAttributes_group"
  minOccurs="0" maxOccurs="1" />
 </xsd:sequence>
 <xsd:attribute name="action" type="addr_type:Action_type"
  use="optional" />
 </xsd:complexType>
```

XML Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<addr:AddressCollection version="0.4" xmlns:addr="addr"
xmlns:addr_type="addr_type" xmlns:gml="http://www.opengis.net/gml"
xmlns:smil20="http://www.w3.org/2001/SMIL20/"
xmlns:smil20lang="http://www.w3.org/2001/SMIL20/Language"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xml="http://www.w3.org/XML/1998/namespace"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="addr addr.xsd ">
 <USPSPostalDeliveryBox>
```

```
  <USPSBox>
   <USPSBoxType>PO BOX</USPSBoxType>
   <USPSBoxId>159753</USPSBoxId>
  </USPSBox>
  <CompleteSubaddress>
   <SubaddressElement>
   <SubaddressType>PMB</SubaddressType>
   <SubaddressIdentifier>3571</SubaddressIdentifier>
   </SubaddressElement>
  </CompleteSubaddress>
  <CompletePlaceName>
   <PlaceName>Herndon</PlaceName>
  </CompletePlaceName>
  <StateName>VA</StateName>
  <ZIPCode>22071</ZIPCode>
  </USPSPostalDeliveryBox>
</addr:AddressCollection>
```

Quality Measure

Pattern Sequence Measure

3.2.3.2 USPS Postal Delivery Route

Syntax: { USPS Address *** } + {** Complete Place Name *** } + {** State Name *** } + {** ZIP Code **} + {** ZIP Plus 4 **} + {** Country Name **}**

Type 1: Rural Route (RR):

USPS Address = "RR"* + { USPS Box Group ID *} + "BOX"* + (USPS Box ID * } + { Private Mail Box }

Syntax: { USPS Address* } + { Complete Place Name * } + { State Name *} + { ZIP Code } + { ZIP Plus 4 } + { Country Name }

Type 2: Highway Contract Route (HC):

USPS Address = "HC"* + { USPS Box Group ID * } + "BOX"* + (USPS Box ID * } + { Private Mail Box }

Syntax: { USPS Address* } + { Complete Place Name * } + { State Name *} + { ZIP Code } + { ZIP Plus 4 } + { Country Name }

Type 3: Overseas Military or Diplomatic Delivery:

USPS Address = "PSC" or "CMR" or "UNIT" * + { USPS Box Group ID * } + "BOX"* + (USPS Box ID * }

Syntax: { USPS Address* } + {"APO" or "FPO" or "DPO" *} + {"AE" or "AP" or "AA" *} + { ZIP Code } + { ZIP Plus 4 } + { Country Name }

Defining Characteristics:

1. Addresses of this class must include a USPS Address in the specified RR or HC or overseas military delivery format.
2. In addition, all thoroughfare, landmark, and postal addresses must include a Place Name and a State Name. A ZIP Code is recommended but not mandatory.

Notes and Examples

1. General. In RR and HC addresses, the Complete Place Name element may include multiple Place Names, but the USPS recommends that only the postal community name be used. Example:

- Preferred: Wailuku, HI
- Acceptable: Wailuku, Maui, HI

2. Rural Route Address Notes and Examples (per USPS Pub 28 sec. 24):

- USPS Pub 28 Sec. 241: "Print rural route addresses on mailpieces as: RR N BOX NN. Do not use the words RURAL, NUMBER, NO., or the pound sign (#)."
  - RR 2 BOX 152
  - RR 9 BOX 23A
- USPS Pub 28 Sec. 242: "A leading zero before the rural route number is not necessary."
  - Acceptable: RR03 BOX 98D
  - Preferred: RR 3 BOX 98D

- USPS Pub 28 Sec. 243: "Print hyphens as part of the box number only when they are part of the address in the ZIP+4 File."
    - RR 4 BOX 19-1A
- USPS Pub 28 Sec. 244: "Change the designations RFD and RD (as a meaning for rural or rural free delivery) to RR."
    - Incorrect: RFD ROUTE 4 #87A
    - Correct: RR 4 BOX 87A
- USPS Pub 28 Sec. 245: "There should be no additional designations, such as town or street names, on the Delivery Address Line of rural route addresses. Because street names used together with route and box numbers can create potential matching difficulty, mailers are encouraged to use only one style of addressing. If secondary name information is used, however, place it above the Delivery Address Line."
    - Incorrect: RR 2 BOX 18 BRYAN DAIRY RD
    - Correct: RR 2 BOX 18
- USPS Pub 28 Sec. 246: "When applying a ZIP+4 code to a rural address, an exact match is preferred. If a box number is included in the address, the mailpiece must bear the appropriate ZIP+4 code representing the range for that box number. When box number information is not available, the Rural Route base record must be used."

3. Highway Contract Route Address Notes and Examples (per USPS Pub 28 sec. 25)

- USPS Pub 28 Sec. 251: "Print highway contract route addresses on a mailpiece as: HC N BOX NN. Do not use the words HIGHWAY CONTRACT, ROUTE, NUMBER, NO., STAR ROUTE, or the pound sign (#).
    - Incorrect: HIGHWAY CONTRACT ROUTE 68 BOX 23A
    - Correct: HC 68 BOX 23A"
- USPS Pub 28 Sec. 252: "A leading zero before the highway contract route number is not needed.
    - Acceptable: HC068 BOX 98D
    - Preferred: HC 68 BOX 98D"
- USPS Pub 28 Sec. 253: "Print hyphens as part of the box number only when they are part of the address in the ZIP+4 File.
    - HC 68 BOX 19-2B "
- USPS Pub 28 Sec. 254: "Change the designation STAR ROUTE, which usually refers to highway contract route, to HC.
    - Incorrect: STAR ROUTE 68 BOX # 45
    - Correct: HC 68 BOX 45"
- USPS Pub 28 Sec. 255: "There should be no additional designations, such as town or street names, on the Delivery Address Line of highway contract route addresses. Street names used together with route and box numbers can create potential matching difficulty. Mailers are encouraged to use only one style of addressing. If secondary name information is used, however, place it above the Delivery Address Line.

- Incorrect: HC 72 BOX 18 BRYAN DAIRY RD
- Correct: HC 72 BOX 18"

- USPS Pub 28 Sec. 256: "When applying a ZIP+4 code to a highway contract route address, an exact match is preferred. If a box number is included in the address, the mailpiece must bear the appropriate ZIP+4 code representing the range for that box number. When box number information is not available, the highway contract base record must be used."

4. Overseas Military PSC, CMR, or UNIT Address Notes and Examples (per USPS Pub 28 sec. 225.1, 238.1, and 239)

- PSC stands for Postal Service Center. CMR stands for Common Mail Room.
- USPS Pub 28 Sec. 238.1: "The Delivery Address Line for all APO/FPO military mail must be standardized as follows:
  - PSC (CMR OR UNIT) NNNN
  - BOX NNNN
  - **Examples:**
  - CMR 830 BOX 51
  - PSC 1650 BOX 10
  - UNIT 908 BOX 111

- **APO, FPO; AA, AE, AP:** USPS Pub 28 Sec. 225.1 "Overseas military addresses must contain the APO or FPO designation along with a two-character “state” abbreviation of AE, AP, or AA and the ZIP Code or ZIP+4 code."
  - APO AE 09001-5275

    - FPO AP 96606-2783
    - APO AA 34035-4198
- APO stands for Army Post Office
- FPO stands for Field Post Office or Fleet Post Office
- AE is used for armed forces in Europe, the Middle East, Africa, and Canada;
- AP is for the Pacific; and
- AA is the Americas excluding Canada."
- **DPO:** USPS Pub 28 Sec. 239 The Delivery Address Line for DPO (Diplomatic Post Office) Department of State mail must be standardized to include the DPO designation and the appropriate two-letter abbreviation (AA, AE or AP), followed by the ZIP+4 or 5-digit ZIP Code.
- **Complete Address Examples:**
    - PSC 802 BOX 74 APO AE 09499-0074
    - UNIT 2050 BOX 4190 APO AP 96278-2050
    - UNIT 9900 DPO AE 09701-1000

XML Tag: <USPSPostalDeliveryRoute>

XML Model:

```
<xsd:complexType name="USPSPostalDeliveryRoute_type">
<xsd:sequence>
```

```
 <xsd:element name="USPSAddress" type="addr_type:USPSAddress_type"
 minOccurs="1" maxOccurs="1" />
 <xsd:group ref="addr_type:PlaceStateZip_group" minOccurs="1"
 maxOccurs="1" />
 <xsd:group ref="addr_type:AddressAttributes_group"
 minOccurs="0" maxOccurs="1" />
</xsd:sequence>
<xsd:attribute name="action" type="addr_type:Action_type"
 use="optional" />
</xsd:complexType>
```

XML Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<addr:AddressCollection version="0.4" xmlns:addr="addr"
xmlns:addr_type="addr_type" xmlns:gml="http://www.opengis.net/gml"
xmlns:smil20="http://www.w3.org/2001/SMIL20/"
xmlns:smil20lang="http://www.w3.org/2001/SMIL20/Language"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xml="http://www.w3.org/XML/1998/namespace"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="addr addr.xsd ">
 <USPSPostalDeliveryRoute>
 <USPSAddress>
```

```
<USPSRoute>
<USPSBoxGroupType>RR</USPSBoxGroupType>
<USPSBoxGroupId>2</USPSBoxGroupId>
</USPSRoute>
<USPSBox>
<USPSBoxType>Box</USPSBoxType>
<USPSBoxId>18</USPSBoxId>
</USPSBox>
</USPSAddress>
<CompletePlaceName>
<PlaceName>Largo</PlaceName>
</CompletePlaceName>
<StateName>FL</StateName>
<ZIPCode>33777</ZIPCode>
</USPSPostalDeliveryRoute>
</addr:AddressCollection>
```

Quality Measures

Pattern Sequence Measure

3.2.3.3 USPS General Delivery Office

Syntax: { USPS General Delivery Point * } + { Complete Place Name * } + { State Name * } + { ZIP Code } + { ZIP Plus 4 } + { Country Name }

Type 1: General Delivery: USPSGeneralDeliveryPoint = "GENERAL DELIVERY" *

Syntax: "GENERAL DELIVERY"* + { Complete Place Name * } + { State Name * } + { ZIP Code * } + "9999" + { Country Name }

Type 2: Overseas Military Address: USPSGeneralDeliveryPoint = SHIP'S NAME

Syntax: SHIP'S NAME* + { "APO" or "FPO" * } + { "AE" or "AP" or "AA" *} + { ZIP Code * } + { ZIP Plus 4 } + { Country Name }

Defining Characteristics:

1. Addresses of this class must include a USPS General Delivery Point in the specified format.
2. In addition, all thoroughfare, landmark, and postal addresses must include a Place Name and a State Name. A ZIP Code is recommended but not mandatory.

Notes and Examples

1. General. In General Delivery addresses, the Complete Place Name element may include multiple Place Names, but the USPS recommends that only the postal community name be used. Example:

- Preferred: Wailuku, HI
- Acceptable: Wailuku, Maui, HI

2. General Delivery Addresses Note and Example {per USPS Pub 28 sec. 26}

- USPS Pub 28 Sec. 261: “Use the words GENERAL DELIVERY, uppercase preferred, spelled out (no abbreviation), as the Delivery Address Line on the mailpiece. Each record will carry the 9999 add-on code.”

- **Complete Example:**

GENERAL DELIVERY
TAMPA FL 33602-9999

3. Overseas Military Addresses Notes and Examples {per USPS Pub 28 sec. 225.1 and 238.1}

- USPS Pub 28 Sec. 238.1: "The Delivery Address Line for all APO/FPO military mail must be standardized as follows:
  - SHIP’S NAME
  - **Example:**
  - USS SEA DEVIL SSN-664

- USPS Pub 28 Sec. 225.1 "Overseas military addresses must contain the APO or FPO designation along with a two-character “state” abbreviation of AE, AP, or AA and the ZIP Code or ZIP+4 code.
  - APO AE 09001-5275
  - FPO AP 96606-2783
  - APO AA 34035-4198
  - AE is used for armed forces in Europe, the Middle East, Africa, and Canada;
  - AP is for the Pacific; and
  - AA is the Americas excluding Canada."
  - **Complete Example:**

    USCGC HAMILTON

    FPO AP 96667-3931

XML Tag: <USPSGeneralDeliveryOffice>

XML Model:

```
<xsd:complexType name="USPSGeneralDeliveryOffice_type">
<xsd:sequence>
 <xsd:element name="USPSGeneralDeliveryPoint"
 type="addr_type:USPSGeneralDeliveryPoint_type" />
 <xsd:group ref="addr_type:PlaceStateZip_group" minOccurs="1"
 maxOccurs="1" />
```

```
 <xsd:group ref="addr_type:AddressAttributes_group"
 minOccurs="0" maxOccurs="1" />
</xsd:sequence>
<xsd:attribute name="action" type="addr_type:Action_type"
 use="optional" />
</xsd:complexType>
```

XML Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<addr:AddressCollection version="0.4" xmlns:addr="addr"
xmlns:addr_type="addr_type" xmlns:gml="http://www.opengis.net/gml"
xmlns:smil20="http://www.w3.org/2001/SMIL20/"
xmlns:smil20lang="http://www.w3.org/2001/SMIL20/Language"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xml="http://www.w3.org/XML/1998/namespace"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="addr addr.xsd ">
 <USPSGeneralDeliveryOffice>
 <USPSGeneralDeliveryPoint>General
Delivery</USPSGeneralDeliveryPoint>
 <CompletePlaceName>
  <PlaceName>Tampa</PlaceName>
 </CompletePlaceName>
```

```
 <StateName>FL</StateName>
 <ZIPCode>33602</ZIPCode>
 <ZIPPlus4>9999</ZIPPlus4>
 </USPSGeneralDeliveryOffice>
</addr:AddressCollection>
```

Quality Measures

Pattern Sequence Measure

3.2.4 General Address Class

The general address class handles all of the above classes, for files in which the various classes may be mixed together, and addresses that do not conform to any of the above classes.

3.2.4.1 General Address Class

Syntax:

Type 1: The complete address as a single unparsed string of text.

Type 2: The complete address with place, state and zip code parsed out to a single field

{ Delivery Address ***}** **+** **{** Place State ZIP ***}**

Type 3: The complete address with place, state and zip code parsed out to separate fields

{ Delivery Address *** }** **+** **{** Complete Place Name *** }** **+** **{** State Name *** }** **+** **{** ZIP Code **}** **+** **{** ZIP Plus 4 **}** **+** **{** Country Name **}**

Defining Characteristic: In addresses of this class the Delivery Address must be unparsed (except that in Types 2 and 3 the Complete Subaddress may be separated from the rest of the Delivery Address) and may contain thoroughfare, landmark, or postal syntaxes. This class may also include addresses that do not conform to any of the thoroughfare, landmark, or postal classes, including non-U.S. addresses.

Examples:

Type 1:

- Record 1: Address = **123 Main Street, Apt. 1, Ames, IA 50010**
- Record 2: Address = **Ames High School, Room 12, Ames, IA 50010**
- Record 3: Address = **PO Box 1511, Ames, IA 50010**

Type 2:

- Record 1: Delivery Address = **123 Main Street, Apt. 1**; Place State ZIP = **Ames, IA 50010**
- Record 2: Delivery Address = **Ames High School, Room 12**; Place State ZIP = **Ames, IA 50010**

- Record 3: Delivery Address = **PO Box 1511**; Place State ZIP = **Ames, IA 50010**

Type 3:

- Record 1: Delivery Address = **123 Main Street, Apt. 1**; Complete Place Name = **Ames**; State Name = **IA**; ZIP Code = **50010**
- Record 2: Delivery Address = **Ames High School, Room 12**; Complete Place Name = **Ames**; State Name = **IA**; ZIP Code = **50010**
- Record 3: Delivery Address = **PO Box 1511**; Complete Place Name = **Ames**; State Name = **IA**; ZIP Code = **50010**

Notes:

1. Address files often contain—and need to contain—street, landmark, and postal addresses mixed together. The general address class is intended to provide a basis for handling these kinds of files.
2. The general class provides a way to handle addresses that do not conform to any of the thoroughfare, landmark, or postal classes, including non-U.S. addresses.
3. In the general class, at minimum, the complex element Delivery Address is unparsed (except that in Types 2 and 3 the Complete Subaddress may be separated from the rest of the Delivery Address) and may contain thoroughfare, landmark, or postal syntaxes.
4. Within the general class, the three types differ as follows:

- In Type 1, the entire address is a single unparsed string of text.
- In Type 2, the Delivery Address line is separated from the Place State ZIP line.
- In Type 3, the Complete Place Name, State Name, ZIP Code, ZIP Plus 4, and Country Name are separated from each other.

5. In Types 2 and 3, if the Complete Subaddress is separated from the rest of the Delivery Address, then the Delivery Address Type value should be "Subaddress Excluded".

XML Tag: <GeneralAddressClass>

XML Model:

```
<xsd:complexType name="GeneralAddressClass_group">
<xsd:choice>
 <xsd:element name="GeneralAddress"
 type="addr_type:GeneralAddress_type" />
 <xsd:sequence>
 <xsd:element name="USPSGeneralDeliveryPoint"
  type="addr_type:USPSGeneralDeliveryPoint_type" />
 <xsd:group ref="addr_type:PlaceStateZip_group"
  minOccurs="1" maxOccurs="1" />
 <xsd:group ref="addr_type:AddressAttributes_group"
  minOccurs="0" maxOccurs="1" />
 </xsd:sequence>
```

```
 </xsd:choice>
 <xsd:attribute name="action" type="addr_type:Action_type" />
 </xsd:complexType>
```

XML Example: Type 1:

123 Main Street, Apt 1, Ames, IA 50010

Type 2:

```
<?xml version="1.0" encoding="UTF-8"?>
<addr:AddressCollection version="0.4" xmlns:addr="addr"
xmlns:addr_type="addr_type" xmlns:gml="http://www.opengis.net/gml"
xmlns:smil20="http://www.w3.org/2001/SMIL20/"
xmlns:smil20lang="http://www.w3.org/2001/SMIL20/Language"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xml="http://www.w3.org/XML/1998/namespace"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="addr addr.xsd ">
 <GeneralAddressClass>
  <DeliveryAddress>123 Main Street, Apt 1</DeliveryAddress>
    <CompletePlaceName>
```

```
    <PlaceStateZip>Ames, IA 50010</PlaceStateZip>
    </CompletePlaceName>
  </GeneralAddressClass>
</addr:AddressCollection>
```

Type 3:

```
<?xml version="1.0" encoding="UTF-8"?>
<addr:AddressCollection version="0.4" xmlns:addr="addr"
xmlns:addr_type="addr_type" xmlns:gml="http://www.opengis.net/gml"
xmlns:smil20="http://www.w3.org/2001/SMIL20/"
xmlns:smil20lang="http://www.w3.org/2001/SMIL20/Language"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xml="http://www.w3.org/XML/1998/namespace"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="addr addr.xsd ">
 <GeneralAddressClass>
   <DeliveryAddress>123 Main Street, Apt 1</DeliveryAddress>
    <CompletePlaceName>
 <PlaceName>Ames</PlaceName>
 </CompletePlaceName>
 <StateName>IA</StateName>
```

```
  <ZIPCode>50010</ZIPCode>
  </GeneralAddressClass>
</addr:AddressCollection>
```

Quality Measures

Pattern Sequence Measure

3.3 Abstract Address Feature Class and Address Collection

3.3.1 Abstract Address Feature Class

All of the address classes described above are specific implementations of an abstract Address Feature class. The Address Feature class is compatible with the abstract Feature class that is generally described in the FGDC Geographic Information Framework Data Content Standard, Base Part, section 7.8.1. The Address Feature is modelling concept used to bind together within the framework. It is described in more detail in Part 4 of this standard.

3.3.2 Address Collection

An Address Collection is an aggregation of address data with its associated metadata, which can then be transferred from one party to another. The Address Collection conforms to the Feature Collection construct that is generally described in the FGDC Geographic Information

Framework Data Content Standard, Base Part, section 7.8.1. The Address Collection is described in more detail in Part 4 of this standard.

4. Address Data Quality

4.1 Introduction

4.1.1 Purpose

The purpose of Part Three is to help users assess the quality of their address data. It provides ways to measure each element, attribute and classification. Some measures compare values to address assignment schemes or domains of values. Others check internal consistency, one of the most important aspects of addresses. Addresses are interdependent: the validity of all can be affected by some. Parity, for example, is an important part of address assignment. In most address schemes, even and odd addresses on the same side of the street disrupt normal address usage. While the assignment of each address is important, patterns make the system work. The methods describe ways to discover anomalies and how to report the quality of the data.

4.1.2 Quality definition

Measuring quality requires a definition for quality. There are definitions of quality in existing standards. Those describing addresses, however, describe the utility of a data set for specific purposes. The National Emergency Number Association (NENA), for example, has documented exchange standards that include a way to describe address quality relative to an

established Automatic Location Information (ALI) files. Similarly, the United States Postal Service (USPS) Postal Addressing Standards describe addresses as used for mailing. These application-specific assessments fulfill the purpose of their respective documents.

Assessing the quality of address data content independent of formats or specific uses, however, is a very different task. It requires information on each of the many aspects of address information. Evaluations for a specific purpose can be constructed from that information, with criteria varying according to each application. What is needed is a definition of the information required, a definition of the elements of address quality.

4.1.3 Elements of Address Quality.

Definitions for the quality of address data content are supplied by standards for spatial metadata. They have quality reporting requirements that describe the elements of address quality. These elements provide guidance for checking the various aspects of quality control, and a way of classifying the various measures listed in this section. If all of them are satisfied, a data set has been thoroughly checked. The measures are listed by quality element in Appendix G.

There is remarkable agreement among the documents on the elements of spatial data quality. Each of the standards that approach that question describes the same five core elements:

- Attribute (Thematic) Accuracy
- Logical Consistency
- Completeness
- Positional Accuracy

- Lineage

Even the names of the elements are essentially identical. CSDGM and SDTS discuss "Attribute Accuracy", while the same content is described as "Thematic Accuracy" in ISO 19115. ISO 19113 includes temporal accuracy as one of the data quality elements, defined as "accuracy of the temporal attributes and temporal relationships of features" (ISO 19113:2002(E), 5.2.1), and this definition remains constant throughout the ISO series. Temporal attributes are not separated from other types of attributes in the CSDGM, although various types of time period entries are listed throughout the standard.

Quality Elements	Standards			
	CSDGM	SDTS	ISO 19113	ISO 19115
Dataset Purpose	†	•	•	•
Dataset Use	†	•	•	•
Attribute (Thematic) Accuracy	•	•	•	•
Logical Consistency	•	•	•	•
Temporal Accuracy	†	†	•	•
Completeness	•	•	•	•
Positional Accuracy	•	•	•	•
Lineage	•	•	•	•

• Specified by name in the standard.

† Provision made for the information in the standard, but not specified by name.

4.2 Anomalies: Uncertainty and Addresses

Quality control for address content is unusual in that it is normal to carry inconsistencies in a data set indefinitely. Those inconsistencies are simply the result of the addressing process. A given locality may not have always applied its address scheme systematically, or the scheme

may have changed. Street names may have changed, or the ground conditions changed in some other way.

These inconsistencies are called "anomalies" throughout this document. It is difficult to call them errors: the conditions that create the anomalies will persist, and many of the individual inconsistencies will never be changed. Finally, address information is essential to core, shared databases in many enterprises. Sharing the information involves communications of all kinds, including face-to-face conversations. Plainly, the word "error" can be less than diplomatic in workplace discussions.

4.2.1 Using Address Anomaly Status

The Address Anomaly Status attribute provides a way of documenting and accounting for anomalies. This attribute should be assigned after an inconsistent address is researched. After it is assigned, records documented as anomalies can be excluded from quality testing. This practice reduces ambiguity, and prevents repeated research on the same addresses.

4.3 Measuring Address Quality

4.3.1 About the Measures

The quality control tests follow a simple recipe:

1. Compare address data to domains and specifications tailored for local use
2. Identify anomalies

Tests are designed to provide data quality element information as described in ISO 19115. The test specification includes:

- Scope: the elements, attributes or classifications to be tested
- Measure: a description of what the test measures.
- Procedure: a description of the test
- Pseudocode Script or Function: an example of the test, in SQL pseudocode. The pseudocode was written (except where noted otherwise) using standard ISO/IEC 9075-1:2008 SQL. Spatial predicates used in the pseudocode are described in OpenGIS Simple Features Specification for SQL.
- Parameters for calculating anomalies as a percentage of the data set

4.3.2 About Anomalies

Measures are described with the understanding that records with known anomalies are excluded from related tests. New anomalies discovered should be corrected or described with Address Anomaly Status attributes.

4.3.3 Calculating Conforming Records as a Percentage of the Data Set

Perc Conforming measures the results of tests for anomalies and describes the percentage of data elements that conform, that are not anomalies. Calculating the percentage of conformance requires inverting the query: the number of anomalies found is subtracted from the total number of records before calculating the percentage.

Pseudocode function: Calculating conforming records as % of data set	**Description** The function receives information directly from SQL statements including the standard COUNT aggregator and calcuates percentages. **Function** Function Perc Conforming(count_of_nonconforming_records, count_of_total_records) RETURNS numeric BEGIN SELECT INTO calc_perc ROUND((CAST((count_of_total_records - count_of_nonconforming_records) AS NUMERIC) / (CAST(count_of_total_records AS NUMERIC)) * 100), 2) RETURN(calc_perc) END
Generalized pseudocode example: Parameters in *italics*	**Query** SELECT Perc Conforming((*SELECT COUNT(*) FROM Address Collection element = nonconforming_condition*), (*SELECT COUNT(*) FROM Address Collection*)) as percent_conforming_to_condition **Successful result: 100% conforming** percent_conforming_to_condition -------------------------------- 100.00 **Unsuccessful result: 30% conforming** percent_conforming_to_condition -------------------------------- 30.00

4.3.4 Notation

The tests are described in pseudocode based on SQL. It uses SQL constructs and operators.

Operators used include:

Operator	Description	SQL Example Statement	Statement Result
||	concatenation	SELECT 'a' || 'b';	ab
%	modulo	SELECT 5 % 2;	1

4.4 Applying Measures to Domains of Values

Domains of values are an important tool for controlling values for address components. Measures used to test data conformance depend on the type of domain. The Content Standard for Digital Geospatial Metadata (CSDGM) classifies domains as enumerated,range, codeset and unrepresentable. The following table lists the CSDGM definition for each type of domain, as listed in *Section 5: Entity and Attribute Information*, along with the measures associated with each.

Domain Type	CSDGM Definition	Quality Measures	Quality Notes
codeset	"reference to a standard or list which contains the members of an established set of valid values."	Tabular Domain Measure Spatial Domain Measure	The U.S. Postal Service list of "Primary Street Suffix Names" is a familiar example of a codeset domain. In cases where specific street suffixes are associated with a given area in the Address Reference System, the association should also be checked with the Spatial Domain Measure.
enumerated	"the members of an established set of valid values."	Tabular Domain Measure Spatial Domain Measure	A local, validated street name list is an example of an enumerated domain. In cases where specific types of values are associated with a given area in the Address Reference System, the association should also be checked with the Spatial Domain Measure.
range	"the minimum and maximum values of a continuum of valid values."	Range Domain Measure	Range domain examples include such things as minimum and maximum Address Number values set by some jurisdictions, or a range of address values assigned to a given grid cell in

		Spatial Domain Measure Address Number Fishbones Measure	the Address Reference System. In the latter case, the Spatial Domain Measure would be required to validate the location of the grid cell. Many Address Number values are associated with a Two Number Address Range or a Four Number Address Range. In the latter case conformance can be checked with the Address Number Fishbones Measure.
unrepresentable	"description of the values and reasons why they cannot be represented."		

4.5 How to use the Measures in a Quality Control Program

4.5.1 Preparation

The measures assume a certain body of knowledge about local addresses. Preparation for a local quality control program largely consists of assembling that information. These include:

- Tabular domains of values for street name components
- Spatial domains: jurisdiction boundaries, address scheme boundaries and components
- Address schemes: geometry and rules.

Tabular domains are frequently difficult to complete, as no organized list of street names may be available. In the latter case all available sources of street names should be compiled and checked against source documents such as plats and ordinances where possible. Official Status attributes may be helpful in assembling and maintaining domains of values. It will be normal to find a variety of street name abbreviations in use: Doctor Martin Luther King Junior

Boulevard, MLK Boulevard, etc. Official Status attributes can help describe variations on street names that may appear in addresses assigned to the same location, or along the same street. As noted throughout the standard, the official name should be completely spelled out: Doctor Martin Luther King Junior Drive.

Address schemes or other local conventions may govern much more than street number assignment. There may be street classification requirements for specific Street Name Post Type names. For instance, some jurisdictions may require "Courts" to be deadends, and forbid "Boulevards" on the same deadends. Street names may conform to themes in particular areas: numbers, birds, trees and presidents are some examples. The latter rule may be satisfied by applying the Spatial Domain Measure, but the former will require locally formulated tests. In any case, local conditions will require attention in drawing up a complete list of standard and local quality measures.

4.5.2 Construction

Once all of the domains and rules have been assembled, use the guidance in Parts One and Two to assemble a list of measures for each aspect of your data. Informative Appendices D through G can be helpful in maintaining an overview of the process. Order the measures by their sequence: check simple elements first, then complex elements, then classifications. Where the conditions are entirely local, such as the "Courts" and "Boulevards" example given above, there may not be a standard measure available. In this case create, name and document your own test, taking case to chose a name that does not duplicate one in the standard. It will be important to have the method completely documented both for maintenance, and in order to convey complete quality information to address users.

4.5.3 Testing

Construct SQL statements specific to your system from the pseudocode given in the standard, and test them. Run all the measures on a test data set to make sure they produce believable results. Where known address problems are not discovered by the measures, review how the measures are applied and double check the SQL. Check to make sure that all measures of quality are thoroughly tested: attribute (thematic) accuracy, logical consistency, temporal accuracy, completeness, positional accuracy and lineage. Where there is insufficient information to check a given aspect of quality you may have a problem built into your address process.

4.5.4 Interpreting Results

The measures are written to produce sets of identifiers. In practice it's important to see the data you're examining in context. In a normalized relational database it is most often easiest to construct a view to display the data you want to see. For example, the table for Complete Street Name entries will most often be composed of foreign keys to other tables with domains of values for all the street name components. A view with concatenated components can help you see patterns of anomalies that you can handle in an organized, efficient way.

4.5.5 Implementation

Once a suite of measures has been constructed and tested, implementation consists of deciding when it will be run, and how to handle the results. Both depend on the process used to assemble a given data set. For example, where a number of datasets from separate organizations are assembled to create a master address repository, a complete suite of tests

may be run on each individual dataset before acceptance. The data may only be incorporated into the repository when the anomalies are either attributed and accepted as part of the data set, or resolved. Once the data are incorporated, it is risky to believe the combined data will test identically to each individual data set. While the street name components may be identical, other aspects may be affected by the inherent interdependence of addresses. The results of Address Number Fishbones Measure, for example, may be very different when new data are added. Quality control implementation, therefore, may require developing several suites of quality measures to support each part of the address process.

User confidence in a data set depends on an effective program. Test thoroughly, and document what you did. One simple thing to do is to record when each suite of tests was used. Users will question aspects of the data. Knowing the condition of the data over time simplifies your response, and increases both the reality and perception of the value of the data.

4.5.6 Maintenance

Addressing is a dynamic process. Just as the construction of testing suites is based on the process, testing suites need to be reexamined each time a process that produces data changes.

4.6 About Nodes for Quality Control

4.6.1 About Nodes

Nodes are the end points for each road segment. They are used throughout Address Data Quality in checking features at intersections. The pseudocode below shows how to create and fill one version of the tables required. There are a wide variety of variations that will work. For

example, in a more normalized database the Complete Street Name field may be replaced by a foreign key. This particular example is given in PostgreSQL/PostGIS. The specifics will vary across systems.

The tables are:

1. StreetsNodes, a table correlating nodes with the street names assigned to segments connecting at those nodes.
2. Nodes, a table to hold the nodes themselves.

4.6.2 StreetsNodes

The transaction below creates and fills the table.

```
begin;

create table StreetsNodes
(
  id serial primary key,
  nodesfk integer references nodes,
  transegfk integer references TranSeg,
  CompleteStreetName varchar(100),
  seg_end varchar(4)
)
;
```

```
select addgeometrycolumn( 'nodes', 'streets_nodes',
'geom',-1,'POINT',2);

insert into StreetsNodes( transegfk, CompleteStreetName, seg_end, geom )
(
  select
     id,
     CompleteStreetName,
     'from',
     st_startpoint( a.geom )
  from
     TranSeg
)
union
(
  select
     id,
     CompleteStreetName,
     'to',
     st_endpoint( a.geom )
  from
     TranSeg
)
```

```
;
end;
```

4.6.3 Nodes

Where street segments intersect, multiple nodes will have the same geometry. This table selects unique node points. The geometries are matched back to the street_nodes table so that each record has a node identifier referencing an unique geometry.

The following transaction creates and fills the table

```
begin;

create table Nodes
(
  id serial primary key
)
;

select addgeometrycolumn( 'nodes', 'nodes', 'geom',-1,'POINT',2);

insert into
  Nodes( geom )
select distinct
```

```
  geom
from
  StreetsNodes
;

end;
```

Finally, the statement below fills the nodesfk field in the StreetsNodes table.

```
update
  StreetsNodes
set
  nodesfk = foo.nodesfk
from
  (
    select
      a.id as nodesfk,
      b.id
    from
      Nodes a,
      StreetsNodes b
    where
      equals( a.geom, b.geom )
  ) as foo
```

```
where
  foo.id = StreetsNodes.id
;

end;
```

4.7 Quality Measures

4.7.1 Address Completeness Measure

Measure Name	Address Completeness Measure
Measure Description	Completeness: a comparison of the number of addressable objects with the address information recorded. There are a number of circumstances where more than one address is assigned to an addressable object. Addressable objects without addresses, however, are anomalies unless described by a domain of exceptions.
Report	Completeness
Evaluation Procedure	Compare the number of addressable objects with the address information recorded.
Spatial Data Required	Geometry describing addressable objects attributed with Address ID, and polygon(s) describing Address Reference System extent.
Pseudocode Example: Testing records	**Query** SELECT Addressable Objects. Address ID FROM AddressReferenceSystemExtent, Addressable Objects WHERE INTERSECTS(Address Reference System Extent.Geometry, Addressable Objects.Geometry)

	AND Addressable Objects. Address ID is null **Result Without Anomalies** AdddressID ------ **Anomalies** ID ------ 23 97 186 ....
Pseudocode Example: Testing the Conformance of a Data Set	**Function** See Perc Conforming for the sample query. **Function Parameters** • count_of_nonconforming_records *SELECT COUNT(Addressable Objects.ID)* *FROM Scheme Extent, Addressable Objects* *WHERE* *INTERSECTS(Scheme Extent.Geometry, Addressable Objects.Geometry)* *AND* *Addressable Objects.Address ID isnull* • count_of_total_records *SELECT COUNT(*)* *FROM Addressable Objects*
Result Report Example	Tested Address Completeness Measure at 98% conformance.

2860
4.7.2 Address Elevation Measure

Measure Name	Address Elevation Measure
Measure	This measure checks elevations against polygons created from

Description	contours of elevation.
Report	Attribute (Thematic) Accuracy
Evaluation Procedure	Check each elevation identified by the measure as outside the range defined by the polygons.
Spatial Data Required	Point locations for each address with an elevation, polygons created from contours of elevation.
Pseudocode Example: Testing Records	**Function** create or replace function addr_check_elev(int) returns boolean as $BODY$ declare addr_elev_id alias for $1; within_poly_values boolean; begin select into within_poly_values case when a.AddressElevation between b.min_elev and b.max_elev then TRUE else FALSE end as "within_poly" from AddressElevations a, ContourPolygons b where a.id = addr_elev_id and intersects(a.Geometry, b.Geometry) ; return (within_poly_values); end

	$BODY$ language 'plpgsql'; **Query** select id, addr_check_elev(id) from AddressElevations
Pseudocode Example: Testing the Conformance of a Data Set	**Function** See Perc Conforming for the sample query. **Function Parameters** • count_of_nonconforming_records *SELECT COUNT(*)* *FROM AddressElevationPoints* *WHERE addr_check_elev(id) = FALSE* • count_of_total_records *SELECT COUNT(*)* *FROM AddressElevationPoints*
Result Report Example	Tested Address Elevation Measure at 94% conformance.

4.7.3 Address Left Right Measure

Measure Name	Address Left Right Measure
Measure Description	Determine the side of the street centerline, left or right, on which each address is located. Left and right attributes are frequently entered by hand, an error-prone process. It is important to confirm the actual locations of the addresses. This information is central to confirming the conformance of the address assignment to the local Address Reference System. Note that the measure is constructed with overlapping ranges where an address is found precisely aligned with the road centerline. In these cases two records will be generated: one describing the point on the left side of the centerline, another describing it on the right. In these few cases it is simply practical to use

	the record that conforms to the Address Reference System and eliminate the other. AddressLeftRightMeasure is a prerequisite to Left Right Odd Even Parity Measure. An example of finding these duplicate records is included, as well as an example of comparing the mathematically determined sides against those recorded by hand. Remember when examining the results that the side is determined by the Address Range Directionality of the centerline geometry. If all the from ends of the centerline segments are at the low addresses and the to ends of the centerline segments at the high addresses, then the results will be consistent. In the latter case, you can evaluate whether the odd and even Address Number values are consistently on the left or right of the segment without also accounting for Address Range Directionality. Where Address Range Directionality is inconsistent, however, it must also factor into left/right evaluation.
Report	Logical Consistency
Evaluation Procedure	Determine the left/right status of the location of each address point. Where there is a value recorded in the database, check it against the side as calculated.
Spatial Data Required	Street centerline and address point locations. The identifier for the street segment associated with each address must be recorded with the address points, and the Address Range Directionality must be recorded with each segment.
Pseudocode Example: Assembling Data	**Query to determine left and right** -- -- The query below describes finding the left-right status of -- address points, storing the results in a table for use in -- later quality control processes. Each subquery is commented. -- -- This query requires pre-existing information about the street -- centerline segment to which each address point is related. -- -- The **select** query may be used on its own, or the **insert** -- clause pre-pended to it in order to save the information in a table. -- This example assumes you want to save quite a lot of information -- -- insert into AddressLeftRight (Address ID, StreetCenterlineID, AddressCompleteStreetNameID, StreetCenterlineAddressRangeDirectionality, StreetCenterlineCompleteStreetNameID,

```
        Point2LineDistance,
        Side,
        AddressPointGeometry
        )

--
-- Figure the angles to determine whether the point is to the left or right of
-- the portion of the line segment closest to the access point. Select
-- all the foreign keys, street name identifiers and distance between the
-- the access point and the street segment along with it. The results show
-- left-right parity, along with the information used to derive it.
-- Use select distinct to account for the adjacent points for which you are
-- generating measurement. When the address point is perfectly aligned with
-- the two point in the segment it will generate two records, one for right and
-- one for left. These can be discovered using the next query example, and
-- resolved individually in favor of the one that agrees with the
-- Address Reference System. There are usually very few of these in
-- any given jurisdiction.

select distinct

        bar.AddressID
        bar.StreetCenterlineID
        bar.AddressCompleteStreetNameID,
        bar.StreetCenterlineCompleteStreetNameID,
        bar.StreetCenterlineAddressRangeDirectionality,
        bar.Point2LineDistance,
        case
                when
                        (
                                degrees( azimuth( bar.Pt1, bar.AddressPointGeometry )
                                )
                                -
                                degrees( azimuth( bar.Pt1, bar.Pt2 ) )
                        ) between 0 and 180 then 'right'
                when
                        (
                                degrees( azimuth( bar.Pt1, bar.AddressPointGeometry )
                                )
                                -
                                degrees( azimuth( bar.Pt1, bar.Pt2 ) )
                        ) between 180 and 360 then 'left'
                when
                        (
                        degrees( azimuth( bar.Pt1, bar.AddressPointGeometry ) )
```

```
                    -
                    degrees( azimuth( bar.Pt1, bar.Pt2 ) )
                    ) between -180 and 0 then 'left'
                when
                    (
                    degrees( azimuth( bar.Pt1, bar.AddressPointGeometry ) )
                    -
                    degrees( azimuth( bar.Pt1, bar.Pt2 ) )
                    ) between -360 and -180 then 'right'
                end as "Side",
        bar.AddressPointGeometry

from
--
-- Select all the fields from the inner query, with the
-- exception of the selected line geometry. Adjacent sets
-- of points describing the line geometry are generated
-- instead. These adjacent points allow measurement of the
-- angle to the closest part of the line segment.
--

        ( select
                foo.AddressID,
                foo.StreetCenterlineID,
                foo.AddressCompleteStreetNameID,
                foo.StreetCenterlineCompleteStreetNameID,
                foo.Point2LineDistance,
                foo.AddressPointGeometry,
                foo.ClosestPointOnStreetCenterline,
                foo.!StreetCenterlineAddressRangeDirectionality,
                pointn( foo.st_geom, generate_series( 1, ( numpoints(
                foo.StreetCenterlineGeometry ) - 1 ) ) ) as Pt1,
                pointn( foo.st_geom, generate_series( 2, numpoints(
                foo.StreetCenterlineGeometry ) ) ) as Pt2
        from
                --
                -- Assemble the various identifiers and geometries for the
                -- query.
                --
                ( select
                    a.AddressID,
                    b.StreetCenterlineID
                    a.CompleteStreetNameID as AddressCompleteStreetNameID,
                    b.CompleteStreetNameID as
                    StreetCenterlineCompleteStreetNameID,
```

```
                distance( a.AddressLocationGeometry, b.locationGeometry)
                as Point2LineDistance,
                line_interpolate_point( b.locationGeometry, (
                line_locate_point( b.locationGeometry, a.locationGeometry )
                ) ) as ClosestPointOnStreetCenterline,
                a.AddressPointGeometry,
                b.!StreetCenterlineAddressRangeDirectionality,
                b.StreetCenterlineGeometry
        from
        --
        -- Select the access point foreign key,
        -- the primary address foreign key,
        -- and the street name foreign key
        -- from the access point layer,
        -- the primary address layer
        -- and the intersection table.
        --
                ( select
                        Address ID,
                        CompleteStreetNameID
                        AddressPointGeometry
                from
                        Address Collection
                ) as a,
                --
                -- Select the street centerline primary key,
                -- the street name foreign key,
                -- the Address Range Directionality,
                -- and the geometry as a linestring from a layer
                -- where the geometry is stored as multilinestrings.
                --
                ( select
                        StreetCenterlineID,
                        CompleteStreetNameID
                        StreetCenterlineAddressRangeDirectionality,
                        StreetCenterlineGeometry
                from
                        StreetCenterline
                ) as b
        where
                a.StreetCenterlineIDfk = b.StreetCenterlineID
        ) as foo
) as bar
```

	where expand(bar.ClosestPointOnStreetCenterline, 1) && makeline(bar.Pt1, bar.Pt2) ;
Pseudocode Example: Checking For Address Points with both Left and Right Records	-- This query should describe few, if any, records. They should be resolved individually -- before proceeding with queries based on left/right data select foo.AddressID, bar.Side from (select Address ID from AddressLeftRight group by Address ID having count(Address ID) > 1) as foo, AddressLeftRight as bar where foo.AddressID = bar.AddressID ;
Pseudocode Example: Checking existing attributes against data produced by the queries above	-- This query produces a list of Address ID values for which the left/right -- attribute can not be checked by the left/right information in the query results -- or where the left/right attribute conflicts with query results -- select a.AddressID from TableWithHandEnteredLeftRightAttributes a

	left join AddressLeftRight b on a.AddressID = b.AddressID where b.AddressID is null or (a.Side != b.Side and (b.StreetCenterlineAddressRangeDirectionality = 'With' or (a.Side = 'left' and b.StreetCenterlineAddressRangeDirectionality = 'With-Against') or (a.Side = 'right' and b.StreetCenterlineAddressRangeDirectionality = 'Against-With')) or (a.Side = b.Side and (b.StreetCenterlineAddressRangeDirectionality = 'Against' or (a.Side = 'left' and b.StreetCenterlineAddressRangeDirectionality = 'Against-With') or (a.Side = 'right' and b.StreetCenterlineAddressRangeDirectionality = 'With-Against')) ;
Pseudocode Example: Testing the conformance of a Data Set	**Function** See Perc Conforming for the sample query. **Function Parameters**

	• count_of_nonconforming_records **Result Without Anomalies** Address ID --------- **Anomalies** Address ID --------- 37 52 96 ...
Pseudocode Example: Testing the Conformance of a Data Set	**Function** See Perc Conforming for the sample query. **Function Parameters** • count_of_nonconforming_records • count_of_total_records *SELECT COUNT(*)* *FROM Address Collection*
Result Report Example	Tested Address Left Right Measure at 85% conformance.

2862
4.7.4 Address Lifecycle Status Date Consistency Measure

Measure Name	Address Lifecycle Status Date Consistency Measure
Measure Description	Test the agreement of the Address Lifecycle Status with the development process. This query produces a list of Address I Ds for which the Address Lifecycle Status attributes do not logically agree with the corresponding stages in the process. This query is far more conceptual than many other queries for the simple reason that the development process, and the data it generates, vary considerably from place to place. It is common to track the starting and ending dates of each Address Lifecycle Status value. Address Start Date and Address End Date are notably different, not directly attached to any given Address Lifecycle Status value. Checking the validity of any given Address Lifecycle Status requires checking both the

	Address Start Date and Address End Date values, and data from the development process.
Report	Temporal Accuracy/Logical Consistency
Evaluation Procedure	Use Simple Elements checks to validate Address Lifecycle Status entries against the domain. Check logical relationships between the status values and stages in the development process.
Spatial Data Required	None
Pseudocode Example: Testing records	**Query** -- -- In this example, the issuance of a building permit describes the transition of an address from potential -- or proposed to active. Any given development process is likely to have a longer, more complicated list -- of conditions. -- SELECT Address ID FROM Address Collection WHERE (Address Lifecycle Status = 'Potential' AND (BuildingPermit is not null OR Address Start Date is null OR Address End Date is not null)) OR (Address Lifecycle Status = 'Proposed' AND (BuildingPermit is not null OR Address Start Date is null OR Address End Date is not null)) OR (Address Lifecycle Status = 'Active' AND (Address Start Date is null or Address End Date is not null)) OR (Address Lifecycle Status = 'Retired'

<table>
<tr><td></td><td>AND Address End Date is null)
)

;</td></tr>
<tr><td>Pseudocode Example: Testing the Conformance of a Data Set</td><td>Function
See Perc Conforming for the sample query.

Function Parameters
count_of_nonconforming_records
SELECT COUNT(*)
FROM Address Collection
WHERE
(Address Lifecycle Status = 'Potential'
AND
(BuildingPermit is not null
OR Address Start Date is null
OR Address End Date is not null
)
)
OR
(Address Lifecycle Status = 'Proposed'
AND
(BuildingPermit is not null
OR Address Start Date is null
OR Address End Date is not null
)
)
OR
(Address Lifecycle Status = 'Active'
AND (Address Start Date is null or Address End Date is not null)
)
OR
(Address Lifecycle Status = 'Retired'
AND Address End Date is null)
)
count_of_total_records
SELECT COUNT(*)
FROM Address Collection</td></tr>
<tr><td>Result Report Example</td><td>Tested Address Lifecycle Status Date Consistency Measure at 100% conformance.</td></tr>
</table>

2863 ### 4.7.5 Address Number Fishbones Measure

Measure Name	Address Number Fishbones Measure
Measure Description	Generate lines between addressed locations and the corresponding locations along the matching Overlapping Ranges Measure to check the spatial sequence of Address Number locations. The pattern created by these lines frequently resembles a fishbone. Where there are sequence conflicts lines between those points will cross. This query is most often used where the Two Number Address Range or Four Number Address Range values are present and trusted. If those values are not present or are suspect the geocoded points may be produced without reference to the ranges. For example, points may be generated along the closest street centerline with a matching Complete Street Name value, directly opposite the addresses. This process, with the results diligently checked, allows the ranges themselves to be checked against an inventory of the address numbers actually located along the segment.
Report	Logical Consistency
Evaluation Procedure	Check for addresses having constructed lines that cross others.
Spatial Data Required	Addressed object location points attributed with Address ID, Address X Coordinate and Address Y Coordinate values; and points describing the same addresses geocoded on a street centerline with 0 offset distance attributed with corresponding, Address ID, TranSegId, Address X Coordinate and Address Y Coordinate values. Note that the query produces spatial data, and a table to collect those results is recommended.
Pseudocode Example: Testing records	The result table has to hold, at minimum, the Address ID, the TranSegId and the lines generated. This allows the fishbones to describe a relationship between the addressed feature and the street on which it is addressed in both geometry and attributes. It may be convenient to add additional information, such as the Address Number and/or Complete Street Name. In the latter case, both the query and the table should be altered to respectively create and hold the information. Note that the geomfromtext function, from the Open Geospatial Consortium's Simple Features for SQL Specification, requires an Address Coordinate Reference System ID. The example is the European Petroleum Survey Group (EPSG) identifier for "NAD83 / Colorado South (ftUS), listed in version 6.13 of the database. **Result Table**

<table>
<tr><td></td><td>CREATE TABLE site2geocode (

pkey SERIAL PRIMARY KEY,
Address ID INTEGER NOT NULL REFERENCES Address Collection,
TranSegID INTEGER NOT NULL REFERENCES TranSeg Collection,
geom geometry

)

Query
insert into site2geocode(Address ID, geom)
select

a.AddressID,
b.TranSegId,
st_makeline(st_makepoint(a.addr_x, a.addr_y), st_makepoint(b.geocode_x, b.geocode_y))

from

Address Point a,
Geocoded Point With 0 Offset b

where

a. Address ID = b. Address ID

;</td></tr>
<tr><td>Pseudocode Example: Testing the Conformance of a Data Set</td><td>Function
See Perc Conforming for the query example.

Function Parameters
count_of_nonconforming_records
SELECT COUNT(selected_id)
FROM site2geocode
WHERE crosses_id isnull
count_of_total_records
SELECT COUNT(Address Number)
FROM Address Collection</td></tr>
<tr><td>Result Report Example</td><td>Tested Address Number Fishbones Measure at 90% conformance.</td></tr>
</table>

Attribution	Address Number Fishbones Measure is based on the "fishboning" test developed by Richard Allen of MAGIC GIS, and is included with his permission.

4.7.6 Address Number Parity Measure

Measure Name	Address Number Parity Measure
Measure Description	Test agreement of the odd/even status of the numeric value of an address number with the Address Number Parity attribute. The arithmetic listed in the pseudocode substitutes for a modulo operator(%) that may be unfamiliar, and is not always available. An alternate WHERE clause with a modulo is: WHERE ((Address Number % 2) = 1 and Address Number Parity = 'even') or ((Address Number % 2) = 0 and Address Number Parity = 'odd');
Report	Logical Consistency
Evaluation Procedure	Compare the odd/even status of the numeric value of an address number with the Address Number Parity attribute.
Spatial Data Required	None.
Pseudocode Example: Testing records	**Query** SELECT Address ID, Address Number FROM Address Collection WHERE (Address Number - ((Address Number / 2) * 2)) = 0 and Address Number Parity = 'odd') or (Address Number - ((Address Number / 2) * 2)) = 1 and Address Number Parity = 'even') **Result Without Anomalies** Address Number Parity Measure -- **Anomalies** Address Number Parity Measure -- Address ID, Address Number 1 Address ID, Address Number 2 Address ID, Address Number 3

	....
Pseudocode Example: Testing the Conformance of a Data Set	**Function** See Perc Conforming for the sample query. **Function Parameters** • count_of_nonconforming_records *SELECT COUNT(Address ID)* *FROM Address Collection* *WHERE* *(Address Number - ((Address Number / 2) * 2)) = 0 and Address Number Parity = 'odd')* *or* *(Address Number - ((Address Number / 2) * 2)) = 1 and Address Number Parity = 'even')* • count_of_total_records *SELECT COUNT(Address ID)* *FROM Address Collection*
Result Report Example	Tested Address Number Parity Measure at 82% conformance.

4.7.7 Address Number Range Completeness Measure

Measure Name	Address Number Range Completeness Measure
Measure Description	Check for a low and high value in each Two Number Address Range or Four Number Address Range pair. This test assumes that you are checking ranges for which ranges should be complete in order to conform with the Address Reference System. Systems that use addresses, such as Computer Aided Dispatch (CAD), or any given Address Reference System may have requirements regarding null or zero numbers on ranges.
Report	Logical Consistency
Evaluation Procedure	Check for a non-zero value for both low and high each range.

Spatial Data Required	None.
Pseudocode Example: Testing records	**Query** -- -- The query below is identical for features using either Two Number Address Range or Four Number Address Range. -- One range type must, however, be used consistently throughout the query. -- When using Four Number Address Range each side must be checked independently, and remain constant -- throughout the query. Fill in the appropriate range values where you see -- [Two Number Address Range.Low | Four Number Address Range.Side.Low] -- -- Two Number Address Range | Four Number Address Range.Side test -- SELECT Two Number Address Range | Four Number Address Range.Side FROM Address Collection WHERE (Two Number Address Range.Low | Four Number Address Range.Side.Low is null OR Two Number Address Range.Low | Four Number Address Range.Side.Low = 0) OR (Two Number Address Range.High | Four Number Address Range.Side.High is null OR Two Number Address Range.High | Four Number Address Range.Side.High = 0) -- -- Two Number Address Range | Four Number Address Range.Side test -- *SELECT* *COUNT(* Two Number Address Range | Four Number Address Range*.Side)* *FROM* *Address Collection* *WHERE* *(* Two Number Address Range*.Low* | Four Number Address Range*.Side.Low is null OR* Two Number Address Range*.Low* |

	Four Number Address Range.Side.Low = 0) *OR* *(Two Number Address Range.High | Four Number Address Range.Side.High isnull OR Two Number Address Range.High | Four Number Address Range.Side.High = 0)* • count_of_total_records -- -- Two Number Address Range | Four Number Address Range.Side test -- *SELECT COUNT(Two Number Address Range | Four Number Address Range.Side)* *FROM Address Collection* --
Result Report Example	Tested Address Number Range Completeness Measure at 50% conformance.

2867 ### 4.7.8 Address Number Range Parity Consistency Measure

Measure Name	Address Number Range Parity Consistency Measure
Measure Description	Test agreement of the odd/even status of the numeric value of low and high address numbers. The arithmetic listed in the pseudocode substitutes for a modula operator(%) that may be unfamiliar, and is not always available. Alternate WHERE clauses with a modula are listed below. Two Number Address Range | Four Number Address Range.Side test: WHERE (Two Number Address Range.Low | Four Number Address Range.Side.Low % 2) != (Two Number Address Range.High | Four Number Address Range.Side.High % 2)
Report	Logical Consistency
Evaluation Procedure	Compare the odd/even status of the numeric value of each address number.
Spatial Data Required	None.
Pseudocode Example: Testing records	**Query** -- -- The query below is identical for features using either Two Number Address

	Range or Four Number Address Range. -- One range type must, however, be used consistently throughout the query. -- When using Four Number Address Range each side must be checked independently, and remain constant -- throughout the query. Fill in the appropriate range values where you see -- [Two Number Address Range.Low | Four Number Address Range.Side.Low] -- Two Number Address Range | Four Number Address Range.Side test: SELECT Two Number Address Range | Four Number Address Range.Side FROM Address Collection WHERE (Two Number Address Range.Low | Four Number Address Range.Side.Low - (truncate(Two Number Address Range.Low | Four Number Address Range.Side.Low / 2) * 2)) != (Two Number Address Range.High | Four Number Address Range.Side.High - (truncate(Two Number Address Range.High | Four Number Address Range.Side.High/ 2) * 2)) **Result Without Anomalies** Address Number Range Parity Consistency Measure -- **Anomalies** Two Number Address Range | Four Number Address Range.Side test: Address Number Range Parity Consistency Measure -- Two Number Address Range | Four Number Address Range.Side 1 Two Number Address Range | Four Number Address Range.Side 2 Two Number Address Range | Four Number Address Range.Side 3 ....
Pseudocode Example: Testing the Conformance of a Data Set	**Function** See Perc Conforming for the sample query. **Function Parameters** Two Number Address Range | Four Number Address Range.Side test: • count_of_nonconforming_records

	SELECT COUNT(Two Number Address Range | Four Number Address Range*.Side)* *FROM Address Collection* *WHERE (* Address Number*.Low - ((* Address Number*.Low / 2) *))* *!= (* Address Number*.High - ((* Address Number*.Low / 2) *))* • count_of_total_records Two Number Address Range | Four Number Address Range.Side test: *SELECT COUNT(* Two Number Address Range | Four Number Address Range*.Side)* *FROM Address Collection*
Result Report Example	Tested Address Number Range Parity Consistency Measure at 90% consistency.

4.7.9 Address Number Range Sequence Measure

Measure Name	Address Number Range Sequence Measure
Measure Description	The sequence of numbers where one non-zero Two Number Address Range or Four Number Address Range meets another. The example shown describes the direction of the segment geometry going from the low Address Number to the high Address Number. Where the direction of the geometry varies, the query will have to be altered accordingly. In cases where segment directionality may vary it is extremely helpful to describe that directionality in the database.
Report	Logical Consistency
Evaluation Procedure	Check ranges on each side of a common point.
Spatial Data Required	Street centerline.
Pseudocode Example: Testing records	**Query** -- The query below is identical for features using either Two Number Address Range or Four Number Address Range. -- One range type must, however, be used consistently throughout the query. -- When using Four Number Address Range each side must be checked independently, and remain constant -- throughout the query. Fill in the appropriate range values where you see -- [Two Number Address Range.Low | Four Number Address Range.Side.Low]

```
--
-- Two Number Address Range | Four Number Address Range.Side test
--

SELECT a.CompleteStreetName,

    a. Two Number Address Range.Low | Four Number Address
    Range.Side.Low,
    a. Two Number Address Range.High | Four Number Address
    Range.Side.High,
    b. Two Number Address Range.Low | Four Number Address
    Range.Side.Low,
    b. Two Number Address Range.High | Four Number Address
    Range.Side.High

FROM Address Collection as a,

    ( SELECT Complete Street Name,
        Two Number Address Range.Low | Four Number Address
        Range.Side.Low,
        Two Number Address Range.High | Four Number Address
        Range.Side.High
    FROM Address Collection )
    as b

WHERE

    a.SegmentDirection = 'low to high'
    AND
    b.SegmentDirection = 'low to high'
    AND
    st_equals( st_endpoint( a.Geometry ), st_startpoint( b.Geometry ) )
    AND
    a. Two Number Address Range.Low | Four Number Address
    Range.Side.Low >= b. Two Number Address Range.High | Four Number
    Address Range.Side.High
    AND
    a.CompleteStreetName = b.CompleteStreetName

GROUP BY a.CompleteStreetName,

    a. Two Number Address Range.Low | Four Number Address
    Range.Side.Low,
    a. Two Number Address Range.High | Four Number Address
    Range.Side.High,
```

	b. Two Number Address Range.Low | Four Number Address Range.Side.Low, b. Two Number Address Range.High | Four Number Address Range.Side.High **Result Without Anomalies** street | end_toleft | end_toright | start_fromleft | start_fromright -----------+----------------+-----------------+---------------+------------- **Result With Anomalies** street | end_toleft | end_toright | start_fromleft | start_fromright -----------+----------------+-----------------+---------------+------------- BIRD DRIVE | 1048 | 1049 | 1048 | 1049 BIRD DRIVE | 1248 | 1249 | 1148 | 1149
Pseudocode Example: Testing the Conformance of a Data Set	**Function** See Perc Conforming for the sample query. **Function Parameters** • count_of_nonconforming_records -- -- Two Number Address Range | Four Number Address Range.Side test -- *SELECT COUNT(*)* *FROM Address Collection as a,* *(SELECT* Complete Street Name, Four Number Address Range.*Side.Low,* Four Number Address Range.*Side.High* *FROM Address Collection)* *as b* *WHERE* *a.SegmentDirection = 'low to high'* *AND* *b.SegmentDirection = 'low to high'* *AND* *st_equals(st_endpoint(a.Geometry), st_startpoint(b.Geometry))* *AND* *a.FourNumberAddressRange.Side.Low >=* *b.FourNumberAddressRange.Side.High* *AND*

	a.CompleteStreetName = b.CompleteStreetName *GROUP BY a.CompleteStreetName,* *a.FourNumberAddressRange.Side.Low,* *a.FourNumberAddressRange.Side.High,* *b.FourNumberAddressRange.Side.Low,* *b.FourNumberAddressRange.Side.High* • *count_of_total_records* -- -- Two Number Address Range | Four Number Address Range.*Side test* -- *_SELECT COUNT(* Two Number Address Range | Four Number Address Range.*Side)* *FROM Address Collection* --
Result Report Example	Tested Address Number Range Sequence Measure at 85% conformance.

4.7.10 Address Range Directionality Measure

Measure Name	Address Range Directionality Measure
Measure Description	This measure derives Address Range Directionality values, allowing update to and/or checks of values stored in the database. It requires that Address Side Of Street be known, and checked with Address Left Right Measure.
Report	Logical Consistency
Evaluation Procedure	Determine the AddressRangeDirectionality value of each segment. Where there is a value recorded in the database, check it against the AddressRangeDirectionality as calculated.
Spatial Data Required	Street centerlines, address point locations, and fishbones (see Address Number Fishbones Measure).
Pseudocode Example	-- -- The example below determines the AddressRangeDirectionality value for a single segment. -- Fill in the TranSegId value for the segment where "[*fill in TranSegId*]" appears in the query. -- select

```
        boom.TranSegId,
        case
                when
                        boom.directionality_left = boom.directionality_right
                then
                        boom.directionality_left
                when
                        boom.directionality_left is not null
                        and
                        boom.directionality_right is null
                then
                        boom.directionality_left
                when
                        boom.directionality_left is null
                        and
                        boom.directionality_right is not null
                then
                        boom.directionality_right
                when
                        boom.directionality_left is not null
                        and
                        boom.directionality_right is not null
                        and
                        boom.directionality_left = boom.directionality_right
                then
                        boom.directionality_left || '-' || boom.directionality_right
        end as "AddressRangeDirectionality"

from

        (
                select
                        a.TranSegId,
                        bim.lf_id,
                        bim.lf_number,
                        bim.lt_id,
                        bim.lt_number,
                        case
                                when
                                        bim.lf_number is not null
                                        and
                                        bim.lt_number is not null
                                        and
                                        bim.lf_number = bim.lt_number
                                        and
```

```
                ( st_distance( st_startpoint( a.Geometry ),
                bim.lf.Geometry )
                  <
                  st_distance( st_endpoint( a.Geometry ),
                bim.lt.Geometry )
                )
        then
                'with'
        when
                bim.lf_number is not null
                and
                bim.lt_number is not null
                and
                bim.lf_number = bim.lt_number
                and
                ( st_distance( st_startpoint( a.Geometry ),
                bim.lf.Geometry )
                  >
                  st_distance( st_endpoint( a.Geometry ),
                bim.lt.Geometry )
                )
        then
                'against'
end as "directionality_left",
bam.rf_id,
bam.rf_number,
bam.rt_id,
bam.rt_number,
case
        when
                bam.rf_number is not null
                and
                bam.rt_number is not null
                and
                bam.rf_number = bam.rt_number
                T and
                ( st_distance( st_startpoint( a.Geometry ),
                bam.rf.Geometry )
                  <
                  st_distance( st_endpoint( a.Geometry ),
                bam.rt.Geometry )
                )
        then
                'with'
        when
```

```
                    bam.rf_number is not null
                    and
                    bam.rt_number is not null
                    and
                    bam.rf_number = bam.rt_number
                    and
                    ( st_distance( st_startpoint( a.Geometry ),
                    bam.rf.Geometry )
                      >
                      st_distance( st_endpoint( a.Geometry ),
                    bam.rt.Geometry )
                    )
                then
                    'against'
            end as "directionality_right"
        from
            TranSegCollection a
            left join
                ( select
                    foo.TranSegId,
                    b.AddressID as lf_id,
                    b.AddressNumber as lf_number,
                    b.Geometry as lf_geom,
                    d.AddressID as lt_id,
                    d.AddressNumber as lt_number,
                    d.Geometry as lt.Geometry
                from
                    AddressNumberFishbones a,
                    AddressCollection b,
                    AddressNumberFishbones c,
                    AddressCollection d,
                    (
                        select
                            [ fill in TranSegId ] as TranSegId,
                            min( c.AddressNumber ),
                            max( c.AddressNumber )
                        from
                            TransSegCollection a,
                            AddressNumberFishbones b,
                            AddressCollection c,
                            Address Side Of Street d
                        where
                            a.id = [ fill in TranSegId ]
                            and
                            a.id = b.TranSegId
```

```
                                        and
                                        b.AddressID = c.AddressID
                                        and
                                        c.AddressID = d.AddressID
                                        and
                                        d.AddressSideOfStreet = 'left'
                                ) as foo
                        where
                                a.TranSegId = foo.TranSegId
                                and
                                a.AddressID = b.AddressID
                                and
                                b.AddressNumber = foo.min
                                and
                                c.TranSegId = foo.TranSegId
                                and
                                c.AddressID = d.AddressID
                                and
                                d.AddressNumber = foo.max
                        ) as bim
                        on a.TranSegId = bim.TranSegId
        left join
                ( select
                        foo.TranSegId,
                        b.AddressID as rf_id,
                        b.AddressNumber as rf_number,
                        b.Geometry as rf_geom,
                        d.geodb_oid as rt_id,
                        d.AddressNumber as rt_number,
                        d.Geometry as rt.Geometry
                from
                        AddressNumberFishbones a,
                        AddressCollection b,
                        AddressNumberFishbones c,
                        AddressCollection d,
                        (
                                select
                                        [ fill in TranSegId ]::integer as
                                        TranSegId,
                                        min( c.AddressNumber ),
                                        max( c.AddressNumber )
                                from
                                        TranSegCollection a,
                                        AddressNumberFishbones b,
                                        AddressCollection c,
```

<table>
<tr><td></td><td>

```
                        AddressLeftRight d
                    where
                        a.geodb_oid = [ fill in TranSegId ]
                        and
                        a.geodb_oid = b.TranSegId
                        and
                        b.AddressID = c.geodb_oid
                        and
                        c.geodb_oid = d.AddressID
                        and
                        d.side = 'right'
                    ) as foo
                where
                    a.TranSegId = foo.TranSegId
                    and
                    a.AddressID = b.geodb_oid
                    and
                    b.AddressNumber = foo.min
                    and
                    c.TranSegId = foo.TranSegId
                    and
                    c.AddressID = d.geodb_oid
                    and
                    d.AddressNumber = foo.max
                ) as bam
            on a.geodb_oid = bam.TranSegId
        where
            a.TranSegId = [ fill in TranSegId ]
        ) as boom
    ;
```

</td></tr>
<tr><td>Pseudocode Example: Checking existing attributes against data produced by the query above</td><td>

--
-- Run the query above for each of the segments within the Address Reference System Extent
--

select

    a.TranSegId,
    a.AddressRangeDirectionality,
    b.AddressRangeDirectionality

from

</td></tr>
</table>

	Address Range Directionality Measure Results a left join Database AddressRangeDirectionality Values b on a.TranSegId = b.TranSegId where b.AddressRangeDirectionality is null or a.AddressRangeDirectionality = b.AddressRangeDirectionality ;
Pseudocode Example: Testing the conformance of a Data Set	**Function** See Perc Conforming for the sample query. **Function Parameters** • count_of_nonconforming_records *SELECT COUNT(a.TranSegId)* *from* Address Range Directionality Measure *Results a* *left join Database AddressRangeDirectionality Values b* *on a.TranSegId = b.TranSegId* *where* *b.AddressRangeDirectionality is null* *or* *a.AddressRangeDirectionality = b.AddressRangeDirectionality* *;* • count_of_total_records *SELECT COUNT(TranSegId)* *FROM TranSegIdCollection*
Result Report Example	Tested Address Range Directionality Measure at 94% conformance.

2870 ### 4.7.11 Address Reference System Axes Point Of Beginning Measure

Measure Name	Address Reference System Axes Point Of Beginning Measure
Measure Description	Check the location of the Address Reference System Axis Point Of Beginning against the intersection of the Address Scheme X Axis
Report	Logical Consistency

Evaluation Procedure	Make sure the axes actually intersect. If that intersection is also the location of the Address Reference System Axis Point Of Beginning, compare the point locations.
Spatial Data Required	Address Reference System Axis Point Of Beginning, Address Reference System Axis
Pseudocode Example: Locating Address Reference System Axis Point Of Beginning in use	**Description** The query assumes that the data are stored topologically, so each axis is split at the intersection. It makes, however, no assumptions about the directionality of the axis lines themselves. The query should return a single TRUE result. Once a TRUE result is achieved, the query may be altered to return the Address Reference System Axis Point Of Beginning if it does not already exist. **Query** select equals(foo.x_origin, foo.y_origin) from (select case when equals(east.start, west.start) then east.start when equals(east.start, west.end) then east.start when equals(east.end, west.start) then east.end when equals(east.end, west.end) then east.end end as "x_origin", case when equals(north.start, south.start) then north.start when equals(north.start, south.end) then north.start when equals(north.end, south.start) then north.end when equals(north.end, south.end) then north.end end as "y_origin"

	`` from (select st_startpoint(geom) as start, st_endpoint(geom) as end from axes where axis = 'north') as north, (select st_startpoint(geom) as start, st_endpoint(geom) as end from axes where axis = 'south') as south, (select st_startpoint(geom) as start, st_endpoint(geom) as end from axes where axis = 'east') as east, (select st_startpoint(geom) as start, st_endpoint(geom) as end from axes where axis = 'west') as west) as foo ; ```
Pseudocode Example: Checking Address Reference System Axis Point Of Beginning in use against the Address Reference System Axis Point Of Beginning	**Description** The query assumes that the previous query has been run successfully, describing the Address Reference System Axis Point Of Beginning in use. The query should return a single TRUE result.

of record	**Query**

Query

```
select

        equals( foo.x_origin,
        a.AddressReferenceSystemAxisPointOfBeginning )

from

        Address Reference System Axis Point Of Beginning a,
        (
          select
                case
                        when equals( east.start, west.start )
                        then east.start
                        when equals( east.start, west.end )
                        then east.start
                        when equals( east.end, west.start )
                        then east.end
                        when equals( east.end, west.end )
                        then east.end
                end as "x_origin",
                case
                        when equals( north.start, south.start )
                        then north.start
                        when equals( north.start, south.end )
                        then north.start
                        when equals( north.end, south.start )
                        then north.end
                        when equals( north.end, south.end )
                        then north.end
                end as "y_origin"
        from
                (
                  select
                        st_startpoint( geom ) as start,
                        st_endpoint( geom ) as end
                from
                        axes
                where
                        axis = 'north'
                ) as north,
                (
                  select
```

	st_startpoint(geom) as start, st_endpoint(geom) as end from axes where axis = 'south') as south, (select st_startpoint(geom) as start, st_endpoint(geom) as end from axes where axis = 'east') as east, (select st_startpoint(geom) as start, st_endpoint(geom) as end from axes where axis = 'west') as west) as foo ;
Pseudocode Example: Testing the Conformance of a Data Set	**Function** See Perc Conforming for the sample query. **Function Parameters** • count_of_nonconforming_records N/A. This measure produces a result that conforms 100% or 0%. • count_of_total_records N/A. This measure produces a result that conforms 100% or 0%.
Result Report Example	Tested Address Reference System Axes Point Of Beginning Measure at 100% conformance.

4.7.12 Address Reference System Rules Measure

Measure Name	Address Reference System Rules Measure
Measure Description	Address Reference System layers are essential for both address assignment and quality control, particularly in axial systems. The exact use is dependent on Address Reference System Rules and will be different for each locality. The example query given here describes checking one frequently used rule, that the beginning Address Number values for each street are determined by the grid cell in which the start point of the street is located. Local Address Reference System Rules will shape the final query or queries used.
Report	Logical Consistency. Given the variability involved in testing it will be important to report the queries actually used along with the results.
Evaluation Procedure	For the example, examine each Two Number Address Range or Four Number Address Range for streets where the lowest range numbers are not within the ranges described for the corresponding grid cell.
Spatial Data Required	For the example, street centerline with Two Number Address Range or Four Number Address Range values, and an Address Reference System layer are required. In the example below the Address Reference System layer has low values for east-west and north-south trending roads beginning within the area covered by each grid cell. For the purposes of this example, the east-west or north-south direction of each street segment is recorded in the database.
Pseudocode Example: Testing Records	select a.CompleteStreetName, a.TranSegId, b.GridCellId, a.[Two Number Address Range.Low | Four Number Address Range.Side.Low] b.EastWestLowRangeNumber, b.NorthSouthLowRangeNumber, case when ((a.[Two Number Address Range.Low | Four Number Address Range.Side.Low] = b.EastWestLowRangeNumber and a.GeometryDirection = 'east-west') or (a.[Two Number Address Range.Low | Four Number Address Range.Side.Low] =

	b.NorthSouthLowRangeNumber and a.GeometryDirection = 'north-south')) then 'ok' else 'anomaly' end as "RangeAnomaly" from TranSegCollection a, Address Reference System b, where intersects(st_startpoint(a.Geometry), b.Geometry) order by a.[Two Number Address Range.Low | Four Number Address Range.Side.Low] limit 1 ;
Pseudocode Example: Testing the Conformance of a Data Set	**Function** See Perc Conforming for the sample query. **Function Parameters** • count_of_nonconforming_records *SELECT COUNT(TranSegId)* *FROM TranSegCollection* *WHERE RangeAnomaly = 'anomaly'* • count_of_total_records *SELECT COUNT(TranSegId) FROM TranSegCollection*
Result Report Example	Tested Address Reference System Rules at 95% conformance.

4.7.13 Check Attached Pairs Measure

Measure Name	Check Attached Pairs Measure
Measure Description	This measure describes how to check Attached Element attributes set to "attached" for matching values describing adjacent street name components. By definition, if a component is "attached", one of the adjacent components must also be "attached".
Report	Logical Consistency
Evaluation	Run the query for each Attached Element attribute. Attached Element attributes will be present or absent according to the needs of each locality. If the query is successful it will return an empty result set. Anomalies returned should be researched and corrected. The value of the street name as a whole will be checked in the Complete Street Name Tabular Domain Measure.
Spatial Data Required	None
Pseudocode Example	SELECT Address ID FROM Street name component attributes WHERE attached = TRUE and ((PreviousStreetNameComponent.attached = FALSE or PreviousStreetNameComponent.attached is null) and (SucceedingStreetNameComponent.attached = FALSE or SucceedingStreetNameComponent.attached is null))
Pseudocode Example: Testing the Conformance of a Data Set	**Function** See Perc Conforming for the sample query. **Function Parameters** • count_of_nonconforming_records SELECT COUNT(Address ID) FROM Street name component attributes WHERE attached = TRUE

	and ((PreviousStreetNameComponent.attached = FALSE or PreviousStreetNameComponent.attached is null) and (SucceedingStreetNameComponent.attached = FALSE or SucceedingStreetNameComponent.attached is null)) ; • count_of_total_records SELECT COUNT(Address ID) FROM Street name component attributes ;
Result Report Example	Tested Check Attached Pairs Measure at 92% conformance.

4.7.14 Complete Street Name Tabular Domain Measure

Measure Name	Complete Street Name Tabular Domain Measure
Measure Description	Test the agreement of each Complete Street Name with the domain.
Report	Attribute (Thematic) Accuracy
Evaluation Procedure	Check the Complete Street Name against a domain of values. Creating a list of Complete Street Name values requires taking account of Attached Element attributes. Both the components and the Attached Element attributes may be selected to suit the needs of a particular locality. The function below is one example of a typical function for assembling components into Complete Street Name values. It checks for Attached Element values before putting a space between components. These attributes may describe any of the street name components but most often describe the relattionship between a Separator Element and a Street Name. The example function given here illustrates that condition. The query following the function allows you to check a list of street names compiled without parsing against those assembled using the function. If you maintain the street names only as parsed components, assemble the street names

	using the function and ask at least two people who did not enter them to check the list by hand.
Spatial Data Required	None
Pseudocode Example: Testing records	**Function** create or replace function concat_csn(varchar, varchar, varchar, varchar, boolean, varchar, boolean, varchar, varchar, varchar) returns varchar as $$ declare predir alias for $1; premod alias for $2; pretype alias for $3; separator alias for $4 separator_attached alias for $5; stname alias for $6; stname_attached alias for $7; postmod alias for $8; posttype alias for $9; postdir alias for $10; csn varchar; begin csn = ''; -- predir csn = csn || coalesce(predir, ''); if predir is not null then csn = csn || ' ' ; end if; -- premod csn = csn || coalesce(premod, ''); if premod is not null then csn = csn || ' ' ; -- pretype csn = csn || coalesce(pretype, ''); if pretype is not null then csn = csn || ' ' ; end if; -- separator csn = csn || coalesce(separator, '');

```
if separator is not null
and
(separator_attached isnull or separator_attached = FALSE )
then csn = csn || ' ' ;
end if;

-- stname
csn = csn || stname;
if posttype is not null or postdir is not null
then csn = csn || ' ' ;
end if;

-- postmod
csn = csn || coalesce( postmod, '' );
if postmod is not null
and
postdir is not null
then csn = csn || ' ' ;
end if;

-- posttype
csn = csn || coalesce( posttype, '' );
if posttype is not null
and
( postdir is not null )
then csn = csn || ' ' ;
end if;

-- postdir
csn = csn || coalesce( postdir, '' );

return trim( both csn );

end;

$$ language plpgsql;
```

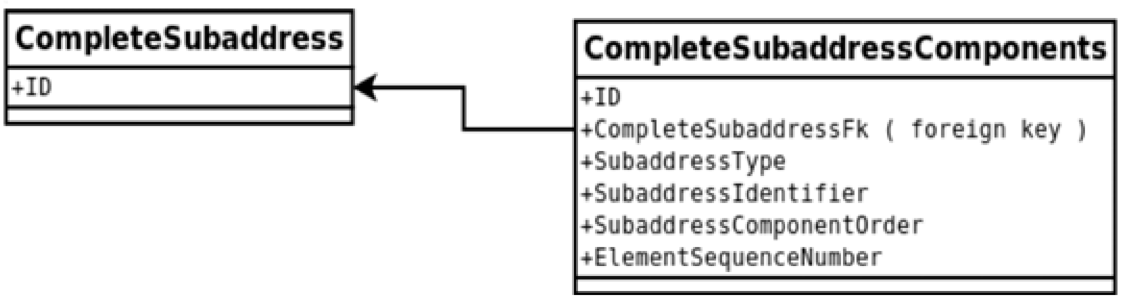
Query

```
SELECT
  CompleteStreetName As disagreeWithCompleteStreetNameDomain
FROM
  Address Collection
    LEFT OUTER JOIN Complete Street Name Domain ON
```

	CompleteStreetName = Complete Street Name Domain.Value WHERE CompleteStreetName Domain.Value isnull ; **Result Without Anomalies** disagreeWithCompleteStreetNameDomain ------------------------------------ **Anomalies** disagreeWithCompleteStreetNameDomain ------------------------------------ Complete Street Name 1 Complete Street Name 2 Complete Street Name 3 ....
Pseudocode Example: Testing the Conformance of a Data Set	**Function** See Perc Conforming for the sample query. **Function Parameters** • count_of_nonconforming_records *SELECT COUNT(Complete Street Name)* *FROM Address Collection* *LEFT OUTER JOIN Complete Street Name Domain* *ON Complete Street Name = Complete Street Name Domain Value* *WHERE Complete Street Name Domain Value isnull* • count_of_total_records *SELECT COUNT(Complete Street Name) FROM Address Collection*
Result Report Example	Tested Complete Street Name Tabular Domain Measure at 83% conformance.

2875 ### 4.7.15 Complex Element Sequence Number Measure

Measure Name	Complex Element Sequence Number Measure
Measure Description	This measure describes how to assemble a complex element in order by Element Sequence Number, and test it for completeness of the elements. It includes a function that can be used to assemble a complete complex element by sequence number. The

	example given is for Complete Subaddress elements, but can be applied to any series of elements ordered by an Element Sequence Number.
Report	Attribute (Thematic) Accuracy
Evaluation Procedure	Check measure results for missing strings. Some localities may have a protocol for the order in which elements of a complete complex element appear. While the various types of combinations are beyond the scope of the standard, they should be considered in a local quality program.
Spatial Data Required	None
Pseudocode Example	**Table Structure** The table structure below describes the structure for the example function and query. It is an example only: tables for any given jurisdiction will almost certainly differ.  **Function** create or replace function AssembleSubaddressString(int) returns varchar as $BODY$ declare id alias for $1; csa CompleteSubaddressComponents%rowtype; Complete Subaddress varchar; begin FOR csa in SELECT * FROM CompleteSubaddressComponents where CompleteSubaddressFk = id order by element_seq LOOP IF Element Sequence Number = 1 AND (Subaddress Component Order = 1 or Subaddress Component Order = 3) THEN Complete Subaddress := Subaddress Type || ' ' || Subaddress

```
        Identifier;

    ELSIF Element Sequence Number = 1
    AND
    Subaddress Component Order = 2

        THEN Complete Subaddress := Subaddress Identifier || ' ' || Subaddress
        Type;

    ELSIF Element Sequence Number > 1
    AND
    ( Subaddress Component Order = 1 OR Subaddress Component Order = 3 )

        THEN Complete Subaddress := Complete Subaddress || ', ' || Subaddress
        Type || ' ' || Subaddress Identifier ;

    ELSIF Element Sequence Number > 1
    AND
    Subaddress Component Order = 2

        THEN Complete Subaddress := Complete Subaddress || ', ' || Subaddress
        Identifier || ' ' || Subaddress Type ;

    END IF;

END LOOP;

RETURN Complete Subaddress ;
END
$BODY$
;
```

Query

```
SELECT DISTINCT
  b.CompleteSubaddressFk, AssembleSubaddressString( b.CompleteSubaddressFk )
FROM
  CompleteSubaddress a
   left join Subaddress b
     on a.PrimaryKey = b.CompleteSubaddress
WHERE
  AssembleSubaddressString( b.CompleteSubaddressFk ) is null
  or
  b.CompleteSubaddressFk is null
```

	;
Pseudocode Example: Testing the Conformance of a Data Set	**Function** See Perc Conforming for the sample query. **Function Parameters** • count_of_nonconforming_records SELECT DISTINCT b.CompleteSubaddressFk, AssembleSubaddressString(b.CompleteSubaddressFk) FROM CompleteSubaddress a left join Subaddress b on a.PrimaryKey = b.CompleteSubaddressFk WHERE AssembleSubaddressString(b.CompleteSubaddressFk) is null or AssembleSubaddressString(b.CompleteSubaddressFk) ; • count_of_total_records SELECT COUNT(*) FROM Complete Subaddress ;
Result Report Example	Tested Complex Element Sequence Number Measure at 93% conformance.

2876 **4.7.16 Data Type Measure**

Measure Name	Data Type Measure
Measure Description	This test uses pattern matching to test for data types. It is common for delimited text files to arrive with fields that appear to be one data type or another, but may have isolated anomalies buried somewhere it the file. In this case the data are frequently loaded to a staging table with all the fields defined as character varying (VARCHAR). Data types need to be evaluated before loading the data into a comprehensive database. For example, this standard defines the Address Number element as integer. This technique helps to locate and resolve types that don't match. Different database systems offer functions to replace one value with another given user-defined conditions. Data types for ASCII values can also be checked by trying to load them to a relational table. Data that do not conform to a given field definition they should

	fail to load. This method, however, leaves anomaly resolution to other systems. Using the staging table method allows for the data to be manipulated within the system where it will be permanently deployed, while allowing the original text file to remain in its original state. History and repeatability can be maintained by saving any queries required to alter values. Patterns are given here for integer and numeric values, as they are most often the data types that cause data loading failures. Other patterns may be added as necessary.
Report	Logical consistency
Evaluation Procedure	Test each value in the address collection for its data type. Any elements that do not agree with the specified data type are anomalies.
Spatial Data Required	None
Pseudocode Example: Testing Records	**Query** SELECT COUNT(value_type), value_type FROM (SELECT CASE WHEN coalesce(trim(value::varchar)) ~ '^[0-9]*$' then 'integer' WHEN coalesce(trim(value::varchar)) ~ '^[0-9]*.[0-9]{1,}$' then 'numeric' ELSE 'other' END AS value_type FROM table WHERE value::varchar ~ '[A-Za-z0-9]') AS foo GROUP BY value_type ; **Successful result example: integer required, integer found** Value Type ---------------------------- integer **Unsuccessful result example: integer required, not found** Value Type

	--------------------------- other
Pseudocode Example: Testing the Conformance of a Data Set	**Function** See Perc Conforming for the sample query. **Function Parameters** • count_of_nonconforming_records *SELECT COUNT(Domain.Values)* *FROM Domain* *LEFT OUTER JOIN Source ON Domain.Value = Source.Value* *WHERE Source.Value isnull* • count_of_total_records *SELECT COUNT(Domain.Values)* *FROM Domain*
Result Report Example	Tested Data Type Measure at 94% conformance.

4.7.17 Delivery Address Type Subaddress Measure

Measure Name	Delivery Address Type Subaddress Measure
Measure Description	Check for null values where the Delivery Address Type indicates an associated Complete Subaddress, and values present where the Delivery Address Type indicates otherwise.
Report	Logical consistency
Evaluation Procedure	Check results for inconsistencies.
Spatial Data Required	None
Pseudocode Example: Testing records	SELECT AddressID, DeliveryAddressType, CompleteSubaddressForeign Key FROM Address Collection WHERE (Delivery Address Type = 'Subaddress Included' and Complete Subaddress.Foreign Key is null) or (Delivery Address Type = 'Subaddress Excluded' and Complete Subaddress.Foreign Key is not null)

	;
Pseudocode Example: Testing the Conformance of a Data Set	**Function** See Perc Conforming for the sample query. **Function Parameters** • count_of_nonconforming_records SELECT COUNT(Address ID) FROM Address Collection WHERE (Delivery Address Type = 'Subaddress Included' and Complete Subaddress.Foreign Key is null) or (Delivery Address Type = 'Subaddress Excluded' and Complete Subaddress.Foreign Key is not null) ; • count_of_total_records SELECT COUNT(Address ID) FROM Address Collection ;
Result Report Example	Tested Delivery Address Type Subaddress Measure at 92% conformance.

2878
4.7.18 Duplicate Street Name Measure

Measure Name	Duplicate Street Name Measure
Measure Description	In many Address Reference Systems distantly disconnected street segments with the same names constitute an anomaly. This query returns TranSegId values for the ends of all disconnected segments. These will most often include results where the disconnected segments are close enough to mitigate the anomaly. The query may be customized with a local distance-based test or the results may be examined individually by Complete Street Name to create a final list of duplicate street names.
Report	Logical Consistency
Evaluation Procedure	Examine the segments included in the results by Complete Street Name, along with the entire set of segments with the same Complete Street Name from all the street segments within the Address Reference System Extent. Take appropriate action where duplicate street names constitute a threat to public safety.

Spatial Data Required	Street centerlines, Address Reference System Extent and nodes and StreetsNodes as described in About Nodes For Quality Control.
Pseudocode Example: Testing records	-- -- The following function and the query using it, as written, assume that there is an integer primary key -- (CompleteStreetNameID) for Complete Street Name values. -- **Function** create or replace function too_many_ends(integer) returns boolean as $BODY$ declare stid alias for $1; chk_dup boolean; begin select into chk_dup case when count(bim.CompleteStreetNameID) > 2 then TRUE else FALSE end as "check_for_duplicate_names" from (select distinct a.CompleteStreetNameID, a.TranSegId from StreetsNodes a, TranSegCollection b, Address Reference System Extent c, (select foo.nodesfk from (select nodesfk from StreetsNodes where CompleteStreetNameID = stid) as foo group by

```
                        foo.nodesfk
                  having
                        count( nodesfk ) = 1
                ) as bar
        where
                a.nodesfk = bar.nodesfk
                and
                CompleteStreetNameID = stid
                and
                a.TranSegId = b.TranSegId
                and
                ( c.AddressReferenceSystemName = 'This Jurisdiction'
                  and
                  intersects( b.Geometry, c.Geometry )
                )
        ) as bim

group by

        bim.CompleteStreetNameID

;

return chk_dup;
end
$BODY$
language 'plpgsql';
```

Query

--
-- This query is likely to require considerable hardware resources as written.
-- Depending on your individual situation, it may be better to run it individually by Complete Street Name
-- To do this, add a **" and** Complete Street Name **= 'Street Name' "** qualifier to the outer where clause.
--
--

```
select distinct

        a.CompleteStreetNameID,
        a.TranSegId

from
```

	StreetsNodes a, TranSegCollection b, (select foo.nodesfk from (select nodesfk from StreetsNodes where too_many_ends(CompleteStreetNameID) = TRUE) as foo group by foo.nodesfk having count(nodesfk) = 1) as bar where a.nodesfk = bar.nodesfk and a.TranSegId = b.geodb_oid ;
Pseudocode Example: Testing the Conformance of a Data Set	**Function** See Perc Conforming for the sample query. **Function Parameters** • count_of_nonconforming_records *SELECT COUNT(distinct a.TranSegId)* *from* *StreetsNodes a,* *TranSegCollection b,* *(* *select* *foo.nodesfk* *from* *(select* *nodesfk* *from* *StreetsNodes* *where*

	too_many_ends(CompleteStreetNameID) = TRUE *) as foo* *group by* *foo.nodesfk* *having* *count(nodesfk) = 1* *) as bar* *where* *a.nodesfk = bar.nodesfk* *and* *a.TranSegId = b.geodb_oid* • *count_of_total_records* *_SELECT COUNT(TranSegId) FROM TranSegCollection*
Result Report Example	Tested Duplicate Street Name Measure at 95% conformance.

4.7.19 Element Sequence Number Measure

Measure Name	Element Sequence Number Measure
Measure Description	Element Sequence Number values must begin at 1 and increment by 1. This measure generates a sequence of integers and checks the Element Sequence Number values against them.
Report	Attribute (Thematic) Accuracy
Evaluation Procedure	Examine Element Sequence Number values for sequences identified by the query.
Spatial Data Required	None
Pseudocode Example: Testing records	This example uses the same tables described for Complex Element Sequence Number Measure, but can be used for any complex element using Element Sequence Number values. The **nextval** construct is specific to PostgreSQL, and similar to the **nextval** used in Oracle. The SQL standard uses **NEXT VALUE FOR**. **Function** create function test_element_sequence_numbers(int) returns int as $BODY$ declare subaddr_id alias for $1;

```
    bum_seq int;

begin
create temporary sequence temp_seq;

select into anomaly_seq

    foo.CompleteSubaddressFk

from

    ( select
        CompleteSubaddressFk,
        nextval( 'temp_seq' ) as TestSequenceNumber,
        Element Sequence Number
     from
        CompleteSubaddressComponents
     where
        CompleteSubaddressFk = subaddr_id
    ) as foo

where

    foo.ElementSequenceNumber != TestSequenceNumber

;

drop sequence test_seq;

return ( anomaly_seq );

end
$BODY$
language 'plpgsql';
```

Query

```
select

    test_element_sequence_numbers( CompleteSubaddressFk ),
    Element Sequence Number

from
```

	CompleteSubaddressComponents where test_element_sequence_numbers(CompleteSubaddressFk) is not null order by test_element_sequence_numbers(CompleteSubaddressFk), Element Sequence Number ;
Pseudocode Example: Testing the Conformance of a Data Set	**Function** See Perc Conforming for the sample query. **Function Parameters** • count_of_nonconforming_records *select count(distinct CompleteSubaddressFk)* *from* *CompleteSubaddressComponents* *where* *test_element_sequence_numbers(CompleteSubaddressFk) is not null* *order by* *test_element_sequence_numbers(CompleteSubaddressFk),* Element Sequence Number ; • count_of_total_records SELECT COUNT(*) FROM Complete Subaddress ;
Result Report Example	Tested Element Sequence Number Measure at 93% conformance.

4.7.20 Future Date Measure

Measure Name	Future Date Measure
Measure Description	This measure produces a list of dates that are in the future.

Report	Temporal Accuracy
Evaluation Procedure	Check dates.
Spatial Data Required	None
Pseudocode Example: Testing records	**Query** SELECT AddressID, Address Start Date, Address End Date FROM Address Collection WHERE Address Start Date > now or Address End Date > now **Result Without Anomalies** Address ID | Address Start Date | Address End Date -----------------+---------------------------+---------------- **Anomalies** Address ID | Address Start Date | Address End Date -----------------+---------------------------+---------------- ID 1 | Start Date 1 | End Date 1 ID 2 | Start Date 2 | End Date 2 ID 3 | Start Date 3 | End Date 3 ....
Pseudocode Example: Testing the Conformance of a Data Set	**Function** See Perc Conforming for the sample query. **Function Parameters** • count_of_nonconforming_records *SELECT* *COUNT(* Address ID *)* *FROM* *Address Collection* *WHERE* Address Start Date > *now() or* Address End Date > *now()* • *count_of_total_records* *_SELECT COUNT(* Address ID *)* *FROM Address Collection*

Result Report Example	Tested Future Date Measure at 87% conformance

4.7.21 Intersection Validity Measure

<table>
<tr><th>Measure Name</th><td>Intersection Validity Measure</td></tr>
<tr><th>Measure Description</th><td>Check intersection addresses for streets that do not intersect in geometry.</td></tr>
<tr><th>Report</th><td>Logical Consistency</td></tr>
<tr><th>Evaluation Procedure</th><td>Check for the intersection of the geometry</td></tr>
<tr><th>Spatial Data Required</th><td>street centerline</td></tr>
<tr><th>Pseudocode Example: Preparing data</th><td>Prepare intersection geometry as described in About Nodes For Quality Control.

Intersection addresses frequently arrive as undifferentiated strings. It is helpful to separate the Complete Street Name elements in these strings in order to check them against the geometry. The exact methods for doing this will vary across database platforms. This simple example uses PostgreSQL.

1. Create a staging table with a primary key (ID) and a field for the intersection strings. It will look something like this:

create table intersection_address
(
id serial primary key,
intersection_address varchar(255)
);

2. Fill the table with your strings. Let the primary key increment automatically to create the intersection indentifiers.

id | intersection_address
1|Boardwalk and Park Place
2|Hollywood Boulevard and Vine Street
3|West Street & Main Street
4|P Street && 19th Street && Mill Road
5|Avenida Rosa y Calle 19
6|Memorial Park, Last Chance Gulch and Memorial Drive
7|Phoenix Village, Scovill Avenue and East 59th Street</td></tr>
</table>

3. Create a new table for the strings to be broken into separate Complete Street Name elements. Use a foreign key from the first table to link each Complete Street Name element to its corresponding intersection.

```
create table intersection_parsed
(
        id serial primary key,
        intersection_addressfk integer references intersection_address,
        completestreetname varchar(80)
);
```

4. Fill the new table

```
insert into intersection_parsed( intersection_addressfk, completestreetname )
select
        id,
        trim( both regexp_split_to_table( intersection_address , ',| and | && | & | y ') )
from
        intersection_address
        ;
```

5. Check your results. They should look something like this:

id	**intersection_addressfk**	**completestreetname**
1	1	Boardwalk
2	1	Park Place
3	2	Hollywood Boulevard
4	2	Vine Street
5	3	West Street
6	3	

	Main Street 7 4 P Street 8 4 19th Street 9 4 Mill Road 10 5 Avenida Rosa 11 5 Calle 19 12 6 Memorial Park 13 6 Last Chance Gulch 14 6 Memorial Drive 15 7 Phoenix Village 16 7 Scovill Avenue 17 7 East 59th Street (17 rows)
Pseudocode Example: Testing records	-- -- The example is shown for a single intersection -- Methods of repetition are left to the user -- **Query 1: List the streets at the intersection**

```
select

	Complete Street Name

from

	intersection_parsed

where

	intersection_addressfk = 2559

;
```

Result:

Complete Street Name
Elm Street Southwest
Oak Road Northwest
Oak Road Southwest

Query 2: Identify the corresponding intersections in the geometry

```
select

	a.nodesfk,
	a.CompleteStreetName

from

	StreetsNodes a,
	( select
		a.nodesfk
	from
		( select nodesfk from StreetsNodes where Complete Street Name =
		'Elm Street Southwest' ) as a,
		( select nodesfk from StreetsNodes where Complete Street Name =
		'Oak Road Northwest' ) as b,
		( select nodesfk from StreetsNodes where Complete Street Name =
		'Oak Road Southwest' ) as c
	where
		a.nodesfk = b.nodesfk
		and
		a.nodesfk = c.nodesfk
```

```
    ) as foo

where


    foo.nodesfk = a.nodesfk

order by

    Complete Street Name

;
```

Result:
Nodesfk | CompleteStreetName
31617 | Elm Street Southwest
31617 | Oak Road Northwest
31617 | Oak Road Southwest

Query 3: Test the Intersection Address

```
select

    foo.CompleteStreetName

from

    ( select
        Complete Street Name
    from
        intersection_parsed
    where
        intersection_addressfk = 2559
    ) as foo
    left join
        select
            a.nodesfk,
            a.CompleteStreetName
        from
            StreetsNodes a,
            ( select
                a.nodesfk
            from
                ( select nodesfk from StreetsNodes where Complete Street
                Name = 'Elm Street Southwest' ) as a,
                ( select nodesfk from StreetsNodes where Complete Street
                Name = 'Oak Road Northwest' ) as b,
```

	(select nodesfk from StreetsNodes where Complete Street Name = 'Oak Road Southwest') as c where a.nodesfk = b.nodesfk and a.nodesfk = c.nodesfk) as bar on foo.CompleteStreetName = bar.CompleteStreetName where bar.CompleteStreetName is null ; **Result Without Anomalies** Intersection Validity Measure ---------------------------- **Anomalies** Intersection Validity Measure ---------------------------- Complete Street Name 1 Complete Street Name 2 Complete Street Name 3 ....
Pseudocode Example: Testing the Conformance of a Data Set	**Function** See Perc Conforming for the sample query. **Function Parameters** • count_of_nonconforming_records *select* *count(foo.CompleteStreetName)* *from* *(select* Complete Street Name *from* *intersection_parsed* *where* *intersection_addressfk = 2559* *) as foo* *left join* *select*

```
                a.nodesfk,
                a.CompleteStreetName
            from
                StreetsNodes a,
                ( select
                    a.nodesfk
                from
                    ( select nodesfk from StreetsNodes where
                    Complete Street Name = 'Elm Street Southwest' )
                    as a,
                    ( select nodesfk from StreetsNodes where
                    Complete Street Name = 'Oak Road Northwest' )
                    as b,
                    ( select nodesfk from StreetsNodes where
                    Complete Street Name = 'Oak Road Southwest' )
                    as c
                where
                    a.nodesfk = b.nodesfk
                    and
                    a.nodesfk = c.nodesfk
                ) as bar
                    on foo.CompleteStreetName =
                    bar.CompleteStreetName
    where
        bar.CompleteStreetName is null
    ;
```

- *count_of_total_records*
  _SELECT COUNT(Intersection Address)
  FROM Address Collection

Result Report Example	Tested Intersection Validity Measure at 96% conformance.

4.7.22 Left Right Odd Even Parity Measure

Measure Name	Left Right Odd Even Parity Measure
Measure Description	Test association of odd and even values in each Block Face Range with the left and right side of the thoroughfare.
Report	Logical Consistency
Evaluation Procedure	Check the odd/even status of the numeric value of each address number for consistency with the established local rule for associating address parity with the

	right or left side of the street when traveling away from the address axis.
Spatial Data Required	Street centerline and spatially derived left and right attributes for each address. Depending on local rules, it may be necessary to derive the overall cardinal direction of the road as well. For example, a local parity rule may require odds to the north and west. In practice this translates to odd numbers to the left where roads run east-west, with the lowest address numbers on the west end, increasing to the east, and so on. Pseudocode listed below describes evaluating parity once the spatial information is in place. Establishing the left-right and directional attributes varies by system.
Pseudocode Example: Testing records	**Query** -- The query below is identical for features using either Two Number Address Range or Four Number Address Range. -- One range type must, however, be used consistently throughout the query. -- When using Four Number Address Range each side must be checked independently, and remain constant -- throughout the query. Fill in the appropriate range values where you see -- [TwoNumberAddressRange.Low | Four Number Address Range.Side.Low] -- SELECT Address ID FROM Address Collection WHERE Complete Address Number BETWEEN [TwoNumberAddressRange.Low | Four Number Address Range.Side.Low] AND [Two Number Address Range.High | Four Number Address Range.Side.High] AND Delivery Address. Complete Street Name = [TwoNumberAddressRange | Four Number Address Range]. Complete Street Name AND ((Address Number Parity = 'odd' AND Address Number Parity.localRule = 'odd addresses on right' AND ((AddressLeftRightMeasure = 'right' AND (Address Range Directionality = 'with' OR AddressRangeDirectionality = 'against-with'

```
                )
            )
            OR
            ( AddressLeftRightMeasure = 'left'
              AND
                ( Address Range Directionality = 'against'
                  OR
                  AddressRangeDirectionality = 'against-with'
                )
            ) )
          )
            OR
            (
                Address Number Parity = 'even'
                AND
                Address Number Parity.localRule = 'even addresses on right'
                AND
                    ( ( AddressLeftRightMeasure = 'right'
                      AND
                        ( Address Range Directionality = 'with'
                          OR
                          AddressRangeDirectionality = 'against-with'
                        )
                    )
                    OR
                    ( AddressLeftRightMeasure = 'left'
                      AND
                        ( Address Range Directionality = 'against'
                          OR
                          AddressRangeDirectionality = 'against-with'
                        )
                    ) )
                  )
            OR
                ( Address Number Parity = 'odd'
                AND
                Address Number Parity.localRule = 'odd addresses on left'
                AND
                ( ( AddressLeftRightMeasure = 'left'
                  AND
                    ( Address Range Directionality = 'with'
                      OR
                      AddressRangeDirectionality = 'with-against'
                    )
                )
```

```
                    OR
                    ( AddressLeftRightMeasure = 'right'
                      AND
                            ( Address Range Directionality = 'against'
                              OR
                              AddressRangeDirectionality = 'with-against'
                            )
                    ) )
                  ) OR
                    ( Address Number Parity = 'even'
                    AND
                    Address Number Parity.localRule = 'even addresses on left'
                    AND
                    ( ( AddressLeftRightMeasure = 'left'
                      AND
                            ( Address Range Directionality = 'with'
                              OR
                              AddressRangeDirectionality = 'with-against'
                            )
                    )
                    OR
                    ( AddressLeftRightMeasure = 'right'
                      AND
                            ( Address Range Directionality = 'against'
                              OR
                              AddressRangeDirectionality = 'with-against'
                            )
                    ) )
            )
)
```

Result Without Anomalies

Address ID

Anomalies

Address ID

    37
    52
    96
    ...

Pseudocode Example: Testing the Conformance of a Data Set	**Function** See Perc Conforming for the sample query. **Function Parameters** • count_of_nonconforming_records -- The query below is identical for features using either Two Number Address Range or Four Number Address Range. -- One range type must, however, be used consistently throughout the query. -- When using Four Number Address Range each side must be checked independently, and remain constant -- throughout the query. Fill in the appropriate range values where you see -- [TwoNumberAddressRange.Low | Four Number Address Range.Side.Low] -- *SELECT COUNT(* Address ID *)* *FROM Address Collection* *WHERE* Complete Address Number *BETWEEN* *[TwoNumberAddressRange.Low* | Four Number Address Range*.Side.Low] AND [* Two Number Address Range*.High* | Four Number Address Range*.Side.High]* *AND* Delivery Address. Complete Street Name = *[TwoNumberAddressRange* | Four Number Address Range *].* Complete Street Name *AND* *(* *(* Address Number Parity = *'odd'* *AND* Address Number Parity*.localRule = 'odd addresses on right'* *AND* *((AddressLeftRightMeasure = 'right'* *AND* *(* Address Range Directionality = *'with'* *OR* *AddressRangeDirectionality = 'against-with'* *)* *)*

	OR *(AddressLeftRightMeasure = 'left'* *AND* *(Address Range Directionality = 'against'* *OR* *AddressRangeDirectionality = 'against-with'* *)* *))* *)* *OR* *(* *Address Number Parity = 'even'* *AND* *Address Number Parity.localRule = 'even addresses on right'* *AND* *((AddressLeftRightMeasure = 'right'* *AND* *(Address Range Directionality = 'with'* *OR* *AddressRangeDirectionality = 'against-with'* *)* *)* *OR* *(AddressLeftRightMeasure = 'left'* *AND* *(Address Range Directionality = 'against'* *OR* *AddressRangeDirectionality = 'against-with'* *)* *))* *)* *OR* *(Address Number Parity = 'odd'* *AND* *Address Number Parity.localRule = 'odd addresses on left'* *AND* *((AddressLeftRightMeasure = 'left'* *AND* *(Address Range Directionality = 'with'* *OR*

	AddressRangeDirectionality = 'with-against' *)* *)* *OR* *(AddressLeftRightMeasure = 'right'* *AND* *(* Address Range Directionality *= 'against'* *OR* *AddressRangeDirectionality = 'with-against'* *)* *))* *) OR* *(* Address Number Parity *= 'even'* *AND* Address Number Parity*.localRule = 'even addresses on left'* *AND* *((AddressLeftRightMeasure = 'left'* *AND* *(* Address Range Directionality *= 'with'* *OR* *AddressRangeDirectionality = 'with-against'* *)* *)* *OR* *(AddressLeftRightMeasure = 'right'* *AND* *(* Address Range Directionality *= 'against'* *OR* *AddressRangeDirectionality = 'with-against'* *)* *))* *)* *)* • count_of_total_records *SELECT COUNT(*)* *FROM Address Collection*
Example: Determining left-right attributes	See Address Left Right Measure

Example: Determining road direction	See attached query -- -- The query below describes finding the overall direction of a road. -- Like the left-right query, it describes finding the directional status -- of a single road within a jurisdiction. The query is run in a loop -- to capture the status of each road. The query itself is complicated, -- so for this purpose the loop is not shown. Each subquery is commented. -- -- Also like the left-right query, the query is complicated by a -- relationship between the primary address and a separately recorded -- access point. In this rural area the access point is essential to -- emergency services. -- -- As presented, the query determines direction for a street name with -- a foreign key of 23. -- -- The query requires: -- * a street centerline layer -- * the foreign key for the street name (CompleteStreetNameID) -- * the foreign key for the segment along which the lowest -- address number is assigned (from_segfk) -- * the foreign key for the segment along which the highest -- address number is assigned (to_segfk) -- -- The query assumes that the street centerline segments are consistently -- named, and that the street names are stored in a table, with a foreign -- key in the street centerline spatial data table. -- -- Further, the query assumes that the street centerline segments are -- directionally consistent with the addresses. -- -- -- -- Store the results for use in the parity check -- insert into CompleteStreetNameDirections(CompleteStreetNameID, from_addressfk, from_segfk, to_addressfk, to_segfk, from_address, from_x, from_y, to_address,

```
        to_x,
        to_y,
        degrees,
        direction
)

        --
        -- Begin with the foreign key for the street name for which you want
        -- to determine the road direction. In this case it's 23.
        --
        -- Select foreign keys for the location of the lowest and highest
        -- address number for the given street name foreign key, and the
        -- foreign keys for the street segments along which those addresses are
        -- located.
        --
        -- Select the start point for the segment corresponding to the lowest
        -- address point, the end point for the segment corresponding to the
        -- highest address point and figure the azimuth. Convert to degrees and
        -- calculate the cardinal direction from the result.
        --
        select

23::integer as CompleteStreetNameID,
bar.pt_id as from_addressfk,
bar.seg_id as from_segfk,
bim.pt_id as to_addressfk,
bim.seg_id as to_segfk,
bar.address as from_address,
round( ( x( bar.geom ) )::numeric, 2 ) as from_x,
round( ( y( bar.geom ) )::numeric, 2 ) as from_y,
bim.address as to_address,
round( ( x( bim.geom ) )::numeric, 2 ) as to_x,
round( ( y( bim.geom ) )::numeric, 2 ) as to_y,
degrees( azimuth( bar.geom, bim.geom ) ),
case
        when
                ( degrees( azimuth( bar.geom, bim.geom ) ) between 0 and 44.99
                or
                degrees( azimuth( bar.geom, bim.geom ) ) between 134.99 and
                224.99
                or
                degrees( azimuth( bar.geom, bim.geom ) ) between 314.99 and
                359.99
                )
                and
```

```
            y( bar.geom ) < y( bim.geom )
        then
            'north'
        when
            ( degrees( azimuth( bar.geom, bim.geom ) ) between 0 and 44.99
            or
            degrees( azimuth( bar.geom, bim.geom ) ) between 134.99 and
            224.99
            or
            degrees( azimuth( bar.geom, bim.geom ) ) between 314.99 and
            359.99
            )
            and
            y( bar.geom ) > y( bim.geom )
        then
            'south'
        when
            ( degrees( azimuth( bar.geom, bim.geom ) ) between 45 and 135
            or
            degrees( azimuth( bar.geom, bim.geom ) ) between 225 and 315
            )
            and
            x( bar.geom ) < x( bim.geom )
        then
            'west'
        when
            ( degrees( azimuth( bar.geom, bim.geom ) ) between 45 and 135
            or
            degrees( azimuth( bar.geom, bim.geom ) ) between 225 and 315
            )
            and
            x( bar.geom ) > x( bim.geom )
        then
            'east'
    end as "direction"

       from
       --
       -- Select the lowest address number for the selected street,
       -- the foreign key for the primary address point, the
       -- foreign key for the street segment, and the geometry
       -- for the start point for the street segment.
       --

    ( select
```

```
        a.address,
        a.geodb_oid as pt_id,
        d.geodb_oid as seg_id,
        startpoint( d.geom ) as geom
from
        primary_address a,
        primary_address_has_access b,
        axess c,
        street_segments d,
        --
        -- Select the minimum address number for the
        -- specified street name
        --
        ( select
                min( address )
        from
                primary_address
        where
                streetnames_id = 23
        ) as foo
where
        a.streetnames_id = 23
        and
        a.address = foo.min
        and
        a.uniqueid = b.primaryaddressuniqueid
        and
        b.accessuniqueid = c.uniqueid
order by
        distance( c.geom, d.geom )
limit 1
) as bar,
--
-- Select the highest address number for the selected street,
-- the foreign key for the primary address point, the
-- foreign key for the street segment, and the geometry
-- for the end point for the street segment.
--
( select
        a.address,
        a.geodb_oid as pt_id,
        d.geodb_oid as seg_id,
        endpoint( d.geom ) as geom
from
        primary_address a,
```

	primary_address_has_access b, axess c, street_segments d, -- -- Select the maximum address number for the -- specified street name -- (select max(address) from primary_address where streetnames_id = 23) as foo where a.streetnames_id = 23 and a.address = foo.max and a.uniqueid = b.primaryaddressuniqueid and b.accessuniqueid = c.uniqueid order by distance(c.geom, d.geom) limit 1) as bim ;
Result Report Example	Tested Left Right Odd Even Parity Measure at 85% conformance.

4.7.23 Location Description Field Check Measure

Measure Name	Location Description Field Check Measure
Measure Description	Field check the location description
Report	Positional Accuracy/Lineage
Evaluation Procedure	Use the Location Description to navigate to the address, checking for discrepancies between the description and ground conditions. It can Note that additional information such as the date the Location Description was collected or last validated

	and/or the name of the people who collected or entered it can reinforce the lineage of the address.
Spatial Data Required	None

2886 ### 4.7.24 Low High Address Sequence Measure

Measure Name	Low High Address Sequence Measure
Measure Description	Confirm that the value of the low address is less than or equal to the high address.
Report	Logical consistency
Evaluation Procedure	Check the values for each range.
Spatial Data Required	None
Pseudocode Example: **Testing records**	-- Two Number Address Range test -- SELECT Two Number Address Range FROM Address Collection WHERE Two Number Address Range.Low < Two Number Address Range.High -- -- -- Four Number Address Range test -- Note that each side must be tested separately. -- Four Number Address Range.Side means either right or left.

	-- The same side should be used consistently throughout the query. -- SELECT Four Number Address Range.Side FROM Address Collection WHERE Four Number Address Range.Side.Low < Four Number Address Range.Side.High
Pseudocode Example: Testing the Conformance of a Data Set	**Function** See Perc Conforming for the sample query. **Function Parameters** • count_of_nonconforming_records -- -- Two Number Address Range test -- *SELECT* Two Number Address Range *FROM Address Collection* *WHERE* Two Number Address Range.*Low* < Two Number Address Range.*High* -- -- -- Four Number Address Range test -- Note that each side must be tested separately.

-- Four Number Address Range.Side means either right or left.

-- The same side should be used consistently throughout the query.

--

SELECT Four Number Address Range.*Side*

FROM Address Collection

WHERE Four Number Address Range.*Side.Low <* Four Number Address Range.*Side.High*

- count_of_total_records

--

-- Two Number Address Range test

--

SELECT COUNT(Two Number Address Range *)*

FROM Address Collection

--

--

-- Four Number Address Range test

-- Note that each side must be tested separately.

-- Four Number Address Range.Side means either right or left.

-- The same side should be used consistently

	throughout the query. -- *SELECT COUNT(Four Number Address Range.Side* *)* *FROM Address Collection*
Result Report Example	Test Low High Address Sequence Measure at 91% conformance.

2887 ### 4.7.24 Official Status Address Authority Consistency Measure

Measure Name	Official Status Address Authority Consistency Measure
Measure Description	Test logical agreement of the Official Status with the Address Authority.
Report	Logical Consistency
Evaluation Procedure	Use Simple Elements checks to validate Official Status entries against the domain. Check logical agreement between the status values and the Address Authority.
Spatial Data Required	None
Pseudocode Example: Testing records	**Description** This query produces a list with Official Status attributes requiring an Address Authority that lack a corresponding entry. **Query** SELECT AddressID FROM Address Collection WHERE (Address Authority is null AND (Official Status = 'Official' OR

	Official Status = 'Official Alternate or Alias' OR Official Status = 'Official Renaming Action of the Address Authority' OR Official Status = 'Alternates Established by an Address Authority')) OR (Address Authority is not null AND (Official Status = 'Unofficial Alternate or Alias' OR Official Status = 'Alternate Names Established by Colloquial Use in a Community' OR Official Status = 'Unofficial Alternate Names Frequently Encountered' OR Official Status = 'Unofficial Alternate Names In Use by an Agency or Entity' OR Official Status = 'Posted or Vanity Address' OR Official Status = 'Verified Invalid') **Result Without Anomalies** addressID --------- **Anomalies** addressID --------- 98 387 598 ....
Pseudocode Example:	**Function**

Testing the Conformance of a Data Set	See Perc Conforming for the sample query. **Function Parameters** • count_of_nonconforming_records *SELECT COUNT(*)* *FROM Address Collection* *WHERE* *(* *Address Authority is null* *AND (* Official Status = *'Official'* *OR* Official Status = *'Official Alternate or Alias'* *OR* Official Status = *'Official Renaming Action of the* Address Authority*'* *OR* Official Status = *'Alternates Established by an* Address Authority*'* *)* *)* *OR (* *Address Authority is not null* *AND (* Official Status = *'Unofficial Alternate or Alias'* *OR* Official Status = *'Alternate Names Established by Colloquial Use in a Community'* *OR* Official Status = *'Unofficial Alternate Names Frequently Encountered'* *OR* Official Status = *'Unofficial Alternate Names In Use by an Agency or Entity'* *OR* Official Status = *'Posted or Vanity Address'* *OR* Official Status = *'Verified Invalid'* *) _* • *count_of_total_records* *_SELECT COUNT(*)* *FROM Address Collection*

Result Report Example	Tested [Official Status Address Authority Consistency Measure] at 81% conformance.

4.7.25 Overlapping Ranges Measure

Measure Name	[Overlapping Ranges Measure]
Measure Description	Check for overlapping ranges with the same Complete Street Name. [Two Number Address Ranges] and [Four Number Address Ranges] are most often used as part of the [Address Reference System]. Overlapping ranges create multiple potential locations for individual addresses, creating ambiguity during the assignment process. Anomalies discovered by this measure should, therefore, be resolved.
Report	Logical Consistency
Evaluation Procedure	Select block ranges or block face ranges with the same Complete Street Name.
Spatial Data Required	Street centerlines with [Two Number Address Range] or [Four Number Address Range] values. Nodes and StreetsNodes as described in [About Nodes For Quality Control].
Pseudocode Example: Testing records	**Function** create or replace function overlapping_range_measure(integer, integer, integer, integer) returns boolean as $BODY$ declare low1 alias for $1; high1 alias for $2; low2 alias for $3; high2 alias for $4; overlap boolean; begin select into overlap case when low1 between low2 and high2 or high1 between low2 and high2 then TRUE else FALSE

```
          end as "overlap_case";

     return overlap;
     end
     $BODY$
     ;
```

Query

```
-- The query below is identical for features using either Two Number Address Range
or Four Number Address Range.
-- One range type must, however, be used consistently throughout the query.
-- When using Four Number Address Range each side must be checked
independently, and remain constant
-- throughout the query. Fill in the appropriate range values where you see
-- [TwoNumberAddressRange.Low | Four Number Address Range.Side.Low ]
--

--
-- Use the overlapping_range_measure function to identify adjoining segments with
overlapping ranges
--
select

     a.NodePointGeometryIdentifier,
     a.CompleteStreetName,
     a.LinestringGeometryIdentifier,
     [b.TwoNumberAddressRange.Low | b.FourNumberAddressRange.Side.Low ]
     [b.TwoNumberAddressRange.High | b.FourNumberAddressRange.Side.High]
     c.LinestringGeometryIdentifier
     [d.TwoNumberAddressRange.Low | d.FourNumberAddressRange.Side.Low ]
     [d.TwoNumberAddressRange.High | d.FourNumberAddressRange.Side.High]

from

     StreetsNodes a,
     StreetCenterlines b,
     StreetsNodes c,
     StreetCenterlines d,
     --
     -- Select only those nodes where two segments with the given street name meet
     --
     (
```

```
        select
                foo.NodePointGeometryIdentifier
        from
                --
                -- Select all the nodes on either end of segments with a given street
                name
                --
                ( select
                        NodePointGeometryIdentifier
                from
                        StreetsNodes
                where
                        a.CompleteStreetName = 'Elm Street'
                ) as foo
        group by
                foo.NodePointGeometryIdentifier
        having count( foo.NodePointGeometryIdentifier ) > 1
) as bar
where
                a.NodePointGeometryIdentifier = bar.NodePointGeometryIdentifier
                and
                a.CompleteStreetName = 'Elm Street'
                and
                a.LinestringGeometryIdentifier = b.LinestringGeometryIdentifier
                and
                a.NodePointGeometryIdentifier = c.NodePointGeometryIdentifier
                and
                c.CompleteStreetName = 'Elm Street'
                and
                c.LinestringGeometryIdentifier = d.LinestringGeometryIdentifier
                and
                b.LinestringGeometryIdentifier = d.LinestringGeometryIdentifier
                and
                b.Geometry.AddressDirectionality =
                d.Geometry.AddressDirectionality
                and
                [b.TwoNumberAddressRange.Low |
                b.FourNumberAddressRange.Side.Low]
                <
                [d.TwoNumberAddressRange.Low |
                d.FourNumberAddressRange.Side.Low]
                and
                overlapping_range_measure(
                        [b.TwoNumberAddressRange.Low |
                        b.FourNumberAddressRange.Side.Low],
```

**[b.TwoNumberAddressRange.High |
b.FourNumberAddressRange.Side.High],
[d.TwoNumberAddressRange.Low |
d.FourNumberAddressRange.Side.Low],
[d.TwoNumberAddressRange.High |
d.FourNumberAddressRange.Side.High]
) = TRUE**
;

Result Without Anomalies

**a.NodePointGeometryIdentifier
a.CompleteStreetName
a.LinestringGeometryIdentifier
[b.TwoNumberAddressRange.Low |
b.FourNumberAddressRange.Side.Low]
[b.TwoNumberAddressRange.High |
b.FourNumberAddressRange.Side.High]
c.LinestringGeometryIdentifier
[d.TwoNumberAddressRange.Low |
d.FourNumberAddressRange.Side.Low]
[d.TwoNumberAddressRange.High |
d.FourNumberAddressRange.Side.High]**

Anomalies

**a.NodePointGeometryIdentifier
a.CompleteStreetName
a.LinestringGeometryIdentifier
[b.TwoNumberAddressRange.Low |
b.FourNumberAddressRange.Side.Low]
[b.TwoNumberAddressRange.High |
b.FourNumberAddressRange.Side.High]
c.LinestringGeometryIdentifier
[d.TwoNumberAddressRange.Low |
d.FourNumberAddressRange.Side.Low]
[d.TwoNumberAddressRange.High |
d.FourNumberAddressRange.Side.High]**

58104
Elm Street
34004
1
1309

	29652 5 1309
Pseudocode Example: Testing the Conformance of a Data Set	Note that the function below calculates the percentage of conformance on the total set of records. The method of repetitively running the test is left to the user. **Function** See Perc Conforming for the sample query. **Function Parameters** • count_of_nonconforming_records -- This query counts the non-conforming records by street name. It should be run for each street name in the data set, and all of the -- results summed, to get the total number of non-conforming records. *SELECT* *COUNT(a.*)* *from* *StreetsNodes a,* *StreetCenterlines b,* *StreetsNodes c,* *StreetCenterlines d,* -- -- *Select only those nodes where two segments with the given street name meet* -- *(* *select* *foo.NodePointGeometryIdentifier* *from* -- -- *Select all the nodes on either end of segments with a given street name* -- *(select* *NodePointGeometryIdentifier* *from* *StreetsNodes* *where* *a.CompleteStreetName = 'Elm Street'* *) as foo* *group by* *foo.NodePointGeometryIdentifier* *having count(foo.NodePointGeometryIdentifier) > 1* *) as bar*

where

a.NodePointGeometryIdentifier = bar.NodePointGeometryIdentifier
and
a.CompleteStreetName = 'Elm Street'
and
a.LinestringGeometryIdentifier = b.LinestringGeometryIdentifier
and
a.NodePointGeometryIdentifier = c.NodePointGeometryIdentifier
and
c.CompleteStreetName = 'Elm Street'
and
c.LinestringGeometryIdentifier = d.LinestringGeometryIdentifier
and
b.LinestringGeometryIdentifier = d.LinestringGeometryIdentifier
and
b.Geometry.AddressDirectionality = d.Geometry.AddressDirectionality
and
[b.TwoNumberAddressRange.Low | b.FourNumberAddressRange.Side.Low]
<
[d.TwoNumberAddressRange.Low | d.FourNumberAddressRange.Side.Low]
and
overlapping_range_measure(
[b.TwoNumberAddressRange.Low | b.FourNumberAddressRange.Side.Low],
[b.TwoNumberAddressRange.High | b.FourNumberAddressRange.Side.High],
[d.TwoNumberAddressRange.Low | d.FourNumberAddressRange.Side.Low],
[d.TwoNumberAddressRange.High | d.FourNumberAddressRange.Side.High]
) = TRUE_

- *count_of_total_records*
  _SELECT COUNT([Two Number Address Range | Four Number Address Range*.Side]) FROM Address Collection*

Result Report Example	Tested Overlapping Ranges Measure at 73% conformance.

4.7.26 Pattern Sequence Measure

Measure Name	Pattern Sequence Measure
Measure Description	Test the sequence of values in each complex element for conformance to the pattern for the complex element. The query produces a list of complex elements in the address collection that do not match a sequence of simple elements. For those complex elements ordered by an Element Sequence Number please refer to Complex Element Sequence Number Measure
Report	Logical Consistency
Evaluation Procedure	Check each complex element value against the pattern defining it.
Spatial Data Required	None
Pseudocode Example: Testing records	**Query** SELECT Complex Element As disagreeWithSequence FROM Address Collection WHERE (Simple Element || Simple Element ...) != Complex Element Pattern **Result Without Anomalies** disagreeWithSequence ------------------ **Anomalies** disagreeWithSequence ------------------ complex element 1 complex element 2 complex element 3 ....
Pseudocode Example: Testing the Conformance of a	**Function** See Perc Conforming for the sample query.

Data Set	**Function Parameters** • count_of_nonconforming_records *SELECT COUNT(Complex Element)* *FROM Address Collection* *WHERE (Simple Element || Simple Element ...) != Complex Element Pattern* • count_of_total_records *SELECT COUNT(Complex Element) FROM Address Collection*
Result Report Example	Tested Pattern Sequence Measure at 95% conformance.

4.7.27 Range Domain Measure

Measure Name	Range Domain Measure
Measure Description	Test each domain for agreement with source ranges.
Report	Attribute (Thematic) Accuracy
Evaluation Procedure	Validate range domain values against low and high range values.
Spatial Data Required	None
Pseudocode Example: Testing records	**Description** This query produces a list of simple elements in the address collection that do not conform to a range domain. **Query** SELECT AddressNumber As disagreeWithSource FROM Address Collection WHERE NOT(Address Number BETWEEN [Two Number Address Range.Low | Four Number Address Range.Side.Low] AND [Two Number Address Range.High | Four Number Address Range.Side.High]) **Result Without Anomalies** disagreeWithSource ------------------

	Anomalies disagreeWithSource ------------------ simple element 1 simple element 2 simple element 3 ....
Pseudocode Example: Testing the Conformance of a Data Set	**Function** See Perc Conforming for the sample query. **Function Parameters** • count_of_nonconforming_records *SELECT COUNT(Address Number)* *FROM* *Address Collection* *WHERE* *NOT(Address Number BETWEEN [Two Number Address Range.Low* | *Four Number Address Range.Side.Low] AND [Two Number Address Range.High* | *Four Number Address Range.Side.High])* • *count_of_total_records* *_SELECT COUNT(Address Number)* *FROM Address Collection*
Result Report Example	Tested Range Domain Measure at 86% conformance.

2891
4.7.28 Related Not Null Measure

Measure Name	Related Not Null Measure
Measure Description	Check to make sure a data element referenced through a foreign key exists.
Report	Logical consistency
Evaluation Procedure	Check for references that don't actually exist.
Spatial Data Required	None
Pseudocode Example: Testing records	SELECT AddressID,

	Related Data Identifier FROM Address Collection LEFT JOIN Related Data on Address Collection.Related Data Identifier = Related Data.Identifier WHERE Related Data.Identifier is null ;
Pseudocode Example: Testing the Conformance of a Data Set	**Function** See Perc Conforming for the sample query. **Function Parameters** • count_of_nonconforming_records SELECT COUNT(Address ID) FROM Address Collection LEFT JOIN Related Data on Address Collection.Related Data Identifier = Related Data.Identifier WHERE Related Data.Identifier is null ; • count_of_total_records SELECT COUNT(Address ID) FROM Address Collection ;
Result Report Example	Tested Related Not Null Measure at 83% conformance.

2892
4.7.29 Related Element Uniqueness Measure

Measure Name	Related Element Uniqueness Measure
Measure Description	Check the uniqueness of the values related to a given element
Report	Attribute (Thematic) Accuracy
Evaluation Procedure	Check any duplicate values found.
Spatial Data Required	None
Pseudocode Example:	**Function**

Testing Records	-- The function for this example describes checking for unique Element Sequence Number values -- using the table structure diagrammed in Complex Element Sequence Number Measure. The same -- pattern can be followed for any related element. -- create or replace function related_element_uniq(int) returns integer as $BODY$ declare ele_id alias for $1; dup_element_seq int; begin select into dup_element_seq Element Sequence Number from Complete Subaddress a, CompleteSubaddressComponents b where a.id = ele_id and a.id = b.CompleteSubaddressFk group by Element Sequence Number having count(Element Sequence Number) > 1 order by Element Sequence Number ; return(dup_element_seq); end $BODY$ language 'plpgsql'; **Query** select id, related_element_uniq(id)

	from Complete Subaddress where related_element_uniq(id) is not null ; **Results without Anomalies** id | related_element_uniq -------------------------- **Results with Anomalies** id | related_element_uniq -------------------------- 2 | 1
Pseudocode Example: Testing the Conformance of a Data Set	**Function** See Perc Conforming for the sample query. **Function Parameters** • count_of_nonconforming_records *SELECT COUNT(Element)* *FROM Address Collection* *WHERE related_element_uniq(ElementID) is not null* • count_of_total_records *SELECT COUNT(Element) FROM Address Collection*
Result Report Example	Tested Related Element Uniqueness Measure at 95% conformance.

4.7.30 Repeated Element Uniqueness Measure

Measure Name	Repeated Element Uniqueness Measure
Measure	Check complex elements with repeated elements for the uniqueness of those elements within the complete element.
Report	Attribute (Thematic) Accuracy
Evaluation Procedure	Check for a repeated element that appears more than once.

<table>
<tr><td>Spatial Data Required</td><td>None</td></tr>
<tr><td>Pseudocode Example: Testing records</td><td>select
  count(subaddress_type || ' ' || subaddress_identifier),
  complete_subaddress_fk,
  subaddress_type || ' ' || subaddress_identifier as subaddress_element
from
  subaddress
group by
  complete_subaddress_fk,
  subaddress_type || ' ' || subaddress_identifier
having
  count(subaddress_type || ' ' || subaddress_identifier) > 1
order by
  complete_subaddress_fk
;</td></tr>
<tr><td>Pseudocode Example: Testing the Conformance of a Data Set</td><td>Function
See Perc Conforming for the sample query.

Function Parameters
count_of_nonconforming_records select
  count(subaddress_type || ' ' || subaddress_identifier),
  complete_subaddress_fk,
  subaddress_type || ' ' || subaddress_identifier as subaddress_element
from
  subaddress
group by
  complete_subaddress_fk,
  subaddress_type || ' ' || subaddress_identifier
having
  count(subaddress_type || ' ' || subaddress_identifier) > 1
order by
  complete_subaddress_fk
;
count_of_total_records select
  count(subaddress_type || ' ' || subaddress_identifier),
  complete_subaddress_fk,
  subaddress_type || ' ' || subaddress_identifier as subaddress_element
from
  subaddress
;</td></tr>
</table>

Result Report Example	Tested Repeated Element Uniqueness Measure at 94% conformance.

4.7.31 Segment Directionality Consistency Measure

Measure Name	Segment Directionality Consistency Measure
Measure Description	Check consistency of street segment directionality. This is important to do before assigning or using Two Number Address Range or Four Number Address Range values, especially when no locations for addresses along those segments yet exist. The test checks for segments with the same street name where more than one "from" or "to" ends meet at the same node. The directionality of the inconsistent segment may be reversed, the Two Number Address Range or Four Number Address Range reversed, or the directionality noted in the database so that the data may be handled consistently.
Report	Logical Consistency
Evaluation Procedure	Examine segments where the measure indicates inconsistent directionality and take appropriate action
Spatial Data Required	Nodes and StreetsNodes as described in About Nodes For Quality Control.
Pseudocode Example: Testing Records	-- -- Pseudocode below describes checking a single Complete Street Name. Methods of repetition left to the user. -- In this example the query is repeated, listing the TranSegId from each side of a given node on separate lines. -- The result, therefore, has a single column listing anomalous TranSegId values. This simplifies the task of producing -- a SELECT DISTINCT list of TranSegId values to check, and for calculating -- the conformance of the dataset. -- select foo.id, foo.CompleteStreetName, foo.TranSegId, foo.TranSegFromTo, foo.NodesFk

	from ((select a.id, a.CompleteStreetName, b.TranSegId, b.TranSegFromTo, b.NodesFk from Complete Street Name a, StreetsNodes b, StreetsNodes c where a.CompleteStreetName = [*fill in* Complete Street Name *value*] and a.id = b.streetnames_staging_allfk and a.id = c.streetnames_staging_allfk and b.NodesFk = c.NodesFk and b.TranSegId < c.TranSegId and b.TranSegFromTo = c.TranSegFromTo) union (select a.CompleteStreetName, c.TranSegId, c.TranSegFromTo, b.NodesFk from streetnames_staging.streetnames_staging_all_streetname a, nodes.streets_nodes b, nodes.streets_nodes c where a.CompleteStreetName = [*fill in* Complete Street Name *value*] and a.CompleteStreetName = b.CompleteStreetName and

	a.CompleteStreetName = c.CompleteStreetName and b.NodesFk = c.NodesFk and b.TranSegId < c.TranSegId and b.TranSegFromTo = c.TranSegFromTo)) as foo order by foo.NodesFk, foo.TranSegId ;
Pseudocode Example: Testing the Conformance of a Data Set	**Function** See Perc Conforming for the sample query. **Function Parameters** • count_of_nonconforming_records *select count(distinct TranSegId) from SegmentDirectionalityConsistencyMeasure Result ;* • count_of_total_records *SELECT COUNT(*) FROM TranSeg Collection ;*
Result Report Example	Tested Segment Directionality Consistency Measure at 55% conformance.

4.7.32 Spatial Domain Measure

Measure Name	Spatial Domain Measure
Measure Description	Test values of some simple elements constrained by domains based on spatial domains: ZIP codes, PLSS descriptions, etc. This is limited to domains that are identified by the simple element alone. Address numbers, for example, cannot be tested against centerline ranges because the street name is only identified in a

	complex element. The query produces a list of simple elements in the address collection that do not conform to a spatial domain.
Report	Positional Accuracy
Evaluation Procedure	Intersect the addressed spatial object with the corresponding location identified by the codeset.
Spatial Data Required	address collection geometry, spatial domain geometry
Pseudocode Example: Testing records	**Query** SELECT Simple Element As notWithinSpatialDomain FROM Address Collection, Spatial Domain WHERE NOT(INTERSECTS(Address Collection.Geometry, Spatial Domain.Geometry)) **Result Without Anomalies** notWithinSpatialDomain ---------------------- **Anomalies** notWithinSpatialDomain ------------------ simple element 1 simple element 2 simple element 3 ....
Pseudocode Example: Testing the Conformance of a Data Set	**Function** See Perc Conforming for the sample query. **Function Parameters** • count_of_nonconforming_records *SELECT* *COUNT(Simple Element)* *FROM* *Address Collection* *WHERE* *NOT(INTERSECTS(Address Collection.Geometry, Spatial*

	Domain.Geometry)) • count_of_total_records *SELECT COUNT(Simple Element)* *FROM Address Collection*
Result Report Example	Tested Spatial Domain Measure at 87% conformance

4.7.33 Start End Date Order Measure

Measure Name	Start End Date Order Measure
Measure	Test the logical ordering of the start and end dates.
Report	Temport Accuracy/Attribute (Thematic) Accuracy
Evaluation Procedure	Check the date order for all entries.
Spatial Data Needed	None
Pseudocode Example: Testing records	**Description** This query makes a list of the date values where the Address Start Date is after the Address End Date. **Query** SELECT Address Start Date, Address End Date FROM Address Collection WHERE Adress End Date is not null and Address Start Date > Address End Date **Result Without Anomalies** Address Start Date | Address End Date -------------------+--------------------- **Anomalies** Address Start Date | Address End Date ---------------------+--------------------- 2004-03-25 | 2004-03-24 2004-06-30 | 2004-05-30

	2005-08-14 | 2005-06-25 ... | ...
Pseudocode Example: Testing the Conformance of a Data Set	**Function** See Perc Conforming for the sample query. **Function Parameters** • count_of_nonconforming_records *SELECT COUNT(*)* *FROM Address Collection* *WHERE Address Start Date >= Address End Date* • count_of_total_records *SELECT COUNT(*)* *FROM Address Collection*
Result Report Example	Tested Start End Date Order Measure at 98% conformance.

2899 **4.7.34 Subaddress Component Order Measure**

Measure Name	Subaddress Component Order Measure
Measure Description	Test Subaddress Elements against the component parts in the order specified by the Subaddress Component Order Measure.
Report	Attribute (Thematic) Accuracy
Evaluation Procedure	Check complex element against concatenated simple elements for anomalies.
Spatial Data Required	None
Pseudocode Example: Testing records	SELECT Subaddress Element, Subaddress Type, Subaddress Identifier, Subaddress Component Order FROM Subaddress Collection WHERE ((Subaddress Element = Subaddress Type || ' ' || Subaddress Identifier or (Subaddress Element = Subaddress Identifier and Subaddress Type is null)

	) and Subaddress Component Order = 2) or (Subaddress Element = Subaddress Identifier || ' ' || Subaddress Type and Subaddress Component Order = 1) ;
Pseudocode Example: Testing the Conformance of a Data Set	**Function** See Perc Conforming for the sample query. **Function Parameters** • count_of_nonconforming_records SELECT COUNT(*) FROM Subaddress Collection WHERE ((Subaddress Element = Subaddress Type || ' ' || Subaddress Identifier or (Subaddress Element = Subaddress Identifier and Subaddress Type is null)) and Subaddress Component Order = 2) or (Subaddress Element = Subaddress Identifier || ' ' || Subaddress Type and Subaddress Component Order = 1) ; • count_of_total_records SELECT COUNT(*) FROM Subaddress Collection ;

Result Report Example	Tested Subaddress Component Order Measure at 96% conformance.

4.7.35 Subaddress Element Z Level Measure

Measure Name	Subaddress Element Z Level Measure
Measure	Check the consistency of the association between Subaddress Element entries associated with a given address. Create a table called **subaddr_zlevel_anom** to hold the anomaly records.
Report	Attribute (Thematic) Accuracy
Evaluation Procedure	Check for multiple Subaddress Element entries associated with a single Address Z Level and Address ID
SQL Example:Testing records	**Function** create or replace function subaddr_zlevel() returns varchar as $$ declare addr record; cnt integer; str varchar; begin cnt := 0; -- drop any pre-existing subaddr_zlevel_anom table drop table subaddr_zlevel_anom; FOR addr in select * from complete_subaddress LOOP IF cnt = 0 THEN -- Create the table with the first set of anomalies. -- Include the full set of subaddress information to -- make it easier to resolve the anomalies. create table subaddr_zlevel_anom as select bar.addressfk, a.complete_subaddress_fk,

```
  a.element_sequence_number,
  a.subaddress_type,
  a.subaddress_identifier,
  bar.z_level
from
  subaddress a,
  complete_subaddress b,
  (
-- Identify addresses where the zlevels conflict:
-- either a given zlevel is assigned to more than one
-- subaddress element or more than one zlevel
-- is assigned to a given subaddress element.
   select
     foo.addressfk,
      foo.z_level
   from
     ( select
      a.addressfk,
      b.complete_subaddress_fk,
      b.z_level
    from
      complete_subaddress a,
      subaddress b
    where
      a.addressfk = addr.addressfk
      and
      b.z_level is not null
    group by
      a.addressfk,
      b.complete_subaddress_fk,
      b.subaddress_type || ' ' || b.subaddress_identifier,
      b.z_level
    order by
      a.addressfk,
      b.complete_subaddress_fk,
      b.z_level
    ) as foo
    group by
      foo.addressfk,
       foo.z_level
    having
      count( z_level ) > 1
  ) as bar
where
  b.addressfk = bar.addressfk
  and
```

```
      b.pkey = a.complete_subaddress_fk
      and
      a.z_level = bar.z_level
    ;

-- Insert additional anomaly records into the existing table.
-- Use the same query as above.

ELSE
  insert into subaddr_zlevel_anom
  select
    bar.addressfk,
    a.complete_subaddress_fk,
    a.element_sequence_number,
    a.subaddress_type,
    a.subaddress_identifier,
    bar.z_level
  from
    subaddress a,
    complete_subaddress b,
    (
     select
       foo.addressfk,
    foo.z_level
     from
       ( select
       a.addressfk,
       b.complete_subaddress_fk,
       b.z_level
     from
       complete_subaddress a,
       subaddress b
     where
       a.addressfk = addr.addressfk
       and
       b.z_level is not null
     group by
       a.addressfk,
       b.complete_subaddress_fk,
       b.subaddress_type || ' ' || b.subaddress_identifier,
       b.z_level
     order by
       a.addressfk,
       b.complete_subaddress_fk,
       b.z_level
    ) as foo
```

```
        group by
          foo.addressfk,
        foo.z_level
        having
          count( z_level ) > 1
      ) as bar
    where
      b.addressfk = bar.addressfk
      and
      b.pkey = a.complete_subaddress_fk
      and
      a.z_level = bar.z_level
    ;
  END IF;

  -- keep a count of the anomaly records to report to the user.

  cnt = cnt + 1;

END LOOP;

-- report results to the user.

str = cnt::varchar || ' ' || 'anomaly records in table subaddr_zlevel_anom.';

return( str );

end;

$$ language plpgsql;
```

Queries

```
select subaddr_zlevel();
select * from subaddr_zlevel_anom;
```

Notes	Due to the complexity of the function, it is listed in its entirety as a complete example rather than as pseudocode.

4.7.36 Tabular Domain Measure

Measure Name	Tabular Domain Measure
Measure Description	Test each value for a simple element for agreement with the corresponding tabular domain. The query produces a list of simple elements in the address collection that do not conform to a domain.
Report	Attribute (Thematic) Accuracy
Evaluation Procedure	Check the value of each simple element against the tabular domain by which it is constrained.
Spatial Data Required	None
Pseudocode Example: Testing records	**Query** SELECT Simple Element As disagreeWithDomain FROM Address Collection LEFT OUTER JOIN Domain WHERE Domain isnull ; **Result Without Anomalies** disagreeWithDomain ------------------ **Result With Anomalies** disagreeWithDomain ------------------ simple element 1 simple element 2 simple element 3 ....
Pseudocode Example: Testing the Conformance of a gData Set	**Function** See Perc Conforming for the query example. **Function Parameters** • count_of_nonconforming_records *SELECT COUNT(Simple Element)* *FROM Address Collection* *LEFT OUTER JOIN Domain ON Simple Element.Field = Domain.Field* *WHERE Domain.Field isnull*

	• count_of_total_records *SELECT COUNT(Simple Element) FROM Address Collection*
Result Report Example	Tested Tabular Domain Measure at 100% conformance.

2904
4.7.37 Uniqueness Measure

Measure Name	Uniqueness Measure
Measure	Test uniqueness of a simple or complex value.
Report	Attribute (Thematic) Accuracy
Evaluation Procedure	Identify unique and duplicate values.
Spatial Data Required	None
Pseudocode Example: Testing records	**Description** This test returns duplicate elements, and the number of occurrences. **Query** SELECT COUNT(Element), Element FROM Address Collection GROUP BY Element HAVING COUNT(Element) > 1 ; **Result Without Anomalies** Count | Element ------+----------- **Anomalies** Count | Element ------+----------- 2 | Element 1 2 | Element 2

	5 | Element 3 .... | ....
Pseudocode Example: Testing the Conformance of a Data Set	**Function** See Perc Conforming for the query example. **Function Parameters** • count_of_nonconforming_records *SELECT COUNT(Element)* *FROM Address Collection* *WHERE Element* *IN (* *SELECT Element* *FROM Address Collection* *GROUP BY Element* *HAVING COUNT(Element) > 1* *)* • count_of_total_records *SELECT COUNT(*) FROM Address Collection*
Result Report Example	Tested Uniqueness Measure at 94% conformance.

4.7.38 USNG Coordinate Spatial Measure

Measure Name	USNG Coordinate Spatial Measure
Measure Description	Test agreement between the location of the addressed object and the area described by the US National Grid Coordinate. This test derives the USNG for a point geometry and compares it to the USNG coordinate
Report	Positional accuracy
Evaluation Procedure	If the derived USNG matches the recorded USNG the comparison is successful. The coord2usng function is an example written for PostgreSQL. Code for other systems may vary. An inverse function, converting USNGto UTM coordinates, is provided for convenience in an **Addendum** section.
Spatial Data Required	Address points, USNG coordinates
Pseudocode Example: Testing records	**Function** coord2usng create or replace function coord2usng(numeric, numeric, numeric, numeric, integer)

```
returns varchar as '

declare
utm_x alias for $1;
utm_y alias for $2;
dd_long alias for $3;
dd_lat alias for $4;
precision alias for $5;
utm_zone integer;
gzd_alpha char(1);
set integer;
e100k_grp1 varchar[8];
e100k_grp2 varchar[8];
e100k_grp3 varchar[8];
n100k_grp1 varchar[20];
n100k_grp2 varchar[20];
e_100k integer;
n_100k integer;
x_alpha_gsz char(1);
y_alpha_gsz char(1);
usng varchar;
x_grid_coord varchar;
y_grid_coord varchar;
num integer;

begin

--find utm zone
select into utm_zone
case
when dd_long between -180 and -174 then 1
when dd_long between -174 and -168 then 2
when dd_long between -168 and -162 then 3
when dd_long between -162 and -156 then 4
when dd_long between -156 and -150 then 5
when dd_long between -150 and -144 then 6
when dd_long between -144 and -138 then 7
when dd_long between -138 and -132 then 8
when dd_long between -132 and -126 then 9
when dd_long between -126 and -120 then 10
when dd_long between -120 and -114 then 11
when dd_long between -114 and -108 then 12
when dd_long between -108 and -102 then 13
when dd_long between -102 and -96 then 14
when dd_long between -96 and -90 then 15
```

	when dd_long between -90 and -84 then 16 when dd_long between -84 and -78 then 17 when dd_long between -78 and -72 then 18 when dd_long between -72 and -66 then 19 when dd_long between -66 and -60 then 20 when dd_long between -60 and -54 then 21 when dd_long between -54 and -48 then 22 when dd_long between -48 and -42 then 23 when dd_long between -42 and -36 then 24 when dd_long between -36 and -30 then 25 when dd_long between -30 and -24 then 26 when dd_long between -24 and -18 then 27 when dd_long between -18 and -12 then 28 when dd_long between -12 and -6 then 29 when dd_long between -6 and 0 then 30 when dd_long between 0 and 6 then 31 when dd_long between 6 and 12 then 32 when dd_long between 12 and 18 then 33 when dd_long between 18 and 24 then 34 when dd_long between 24 and 30 then 35 when dd_long between 30 and 36 then 36 when dd_long between 36 and 42 then 37 when dd_long between 42 and 48 then 38 when dd_long between 48 and 54 then 39 when dd_long between 54 and 60 then 40 when dd_long between 60 and 66 then 41 when dd_long between 66 and 72 then 42 when dd_long between 72 and 77 then 43 when dd_long between 78 and 84 then 44 when dd_long between 84 and 90 then 45 when dd_long between 90 and 96 then 46 when dd_long between 96 and 102 then 47 when dd_long between 102 and 108 then 48 when dd_long between 108 and 114 then 49 when dd_long between 114 and 120 then 50 when dd_long between 120 and 126 then 51 when dd_long between 126 and 132 then 52 when dd_long between 132 and 138 then 53 when dd_long between 138 and 144 then 54 when dd_long between 144 and 150 then 55 when dd_long between 150 and 156 then 56 when dd_long between 156 and 162 then 57 when dd_long between 162 and 168 then 58 when dd_long between 168 and 174 then 59 when dd_long between 174 and 180 then 60

```
end;

-- find grid zone character

select into gzd_alpha
case
when dd_lat between -80 and -72 then "C"
when dd_lat between -72 and -64 then "D"
when dd_lat between -64 and -56 then "E"
when dd_lat between -56 and -48 then "F"
when dd_lat between -48 and -40 then "G"
when dd_lat between -40 and -32 then "H"
when dd_lat between -32 and -24 then "J"
when dd_lat between -24 and -16 then "K"
when dd_lat between -16 and -8 then "L"
when dd_lat between -8 and 0 then "M"
when dd_lat between 0 and 8 then "N"
when dd_lat between 8 and 16 then "P"
when dd_lat between 16 and 24 then "Q"
when dd_lat between 24 and 32 then "R"
when dd_lat between 32 and 40 then "S"
when dd_lat between 40 and 48 then "T"
when dd_lat between 48 and 56 then "U"
when dd_lat between 56 and 64 then "V"
when dd_lat between 64 and 72 then "W"
when dd_lat between 72 and 84 then "X"
end;

-- derive set
if ( utm_zone <= 6 ) then <br /> set := utm_zone;
else
if ( utm_zone % 6 = 0 ) then
set := 6;
else
set := utm_zone % 6;
end if;
end if;

-- construct arrays describing grid zone squares
select into e100k_grp1 array["A","B","C","D","E","F","G","H"];
select into e100k_grp2 array["J","K","L","M","N","P","Q","R"];
select into e100k_grp3 array["S","T","U","V","W","X","Y","Z"];
select into n100k_grp1
array["A","B","C","D","E","F","G","H","J","K","L","M","N","P","Q","R","S","T"
,"U","V"];
```

	select into n100k_grp2 array["F","G","H","J","K","L","M","N","P","Q","R","S","T","U","V","A","B","C","D","E"]; -- get the digit for the 100K places (easting and northing) select into e_100k substring(utm_x from (length(trunc(utm_x)) - 5) for 1); n_100k = (floor(utm_y / 100000) % 20) + 1; -- get the grid select into x_alpha_gsz case when (set = 1 or set = 4) then e100k_grp1[e_100k] when (set = 2 or set = 5) then e100k_grp2[e_100k] when (set = 3 or set = 6) then e100k_grp3[e_100k] end; select into y_alpha_gsz case when (set = 1 or set = 3 or set = 5) then n100k_grp1[n_100k] when (set = 2 or set = 4 or set = 6) then n100k_grp2[n_100k] end; -- get coordinates select into x_grid_coord case when (precision = 10000) then substring(utm_x from (length(trunc(utm_x)) - 4) for 1) when (precision = 1000) then substring(utm_x from (length(trunc(utm_x)) - 4) for 2) when (precision = 100) then substring(utm_x from (length(trunc(utm_x)) - 4) for 3) when (precision = 10) then substring(utm_x from (length(trunc(utm_x)) - 4) for 4) when (precision = 1) then substring(utm_x from (length(trunc(utm_x)) - 4) for 5) end; select into y_grid_coord case when (precision = 10000) then

	substring(utm_y from (length(trunc(utm_y)) - 4) for 1) when (precision = 1000) then substring(utm_y from (length(trunc(utm_y)) - 4) for 2) when (precision = 100) then substring(utm_y from (length(trunc(utm_y)) - 4) for 3) when (precision = 10) then substring(utm_y from (length(trunc(utm_y)) - 4) for 4) when (precision = 1) then substring(utm_y from (length(trunc(utm_y)) - 4) for 5) end; -- assemble the USNG value usng := utm_zone || gzd_alpha || x_alpha_gsz || y_alpha_gsz || x_grid_coord || y_grid_coord; return(usng); end; ' language 'plpgsql'; **Query** SELECT Recorded USNG FROM Address Collection WHERE coord2usng(easting, northing, longitude, latitude, precision) != Recorded USNG ; **Result Without Anomalies** usng ---- **Result with Anomalies** usng ---------------

	18SUJ2348306479
Pseudocode Example: Testing the Conformance of a Data Set	Note that the function below calculates the percentage of conformance on the total set of records. The method of repetitively running the test is left to the user. **Function** See Perc Conforming for the query example. • count_of_nonconforming_records *SELECT COUNT(result)* • count_of_total_records *SELECT COUNT(usng) from Address Collection*
Result Report Example	Tested USNG Coordinate Spatial Measure at 96% conformance.
Addendum	**Function** usng2coord **Note:** This function returns a pair of coordinates at the center of the area described by the precision of the USNG grid reference. create or replace function usng2coord(varchar) returns varchar as ' declare usng alias for $1; zone integer; grid_zone char(1); set integer; offset_north numeric; x_alpha_gsz char(1); y_alpha_gsz char(1); e_coord integer; n_coord integer; e100k integer; n100k integer; e100k_grp1 varchar[8]; e100k_grp2 varchar[8]; e100k_grp3 varchar[8]; n100k_grp1 varchar[20]; n100k_grp2 varchar[20]; e_gsz integer;

```
n_gsz integer;
grid numeric;
precision numeric;
e_grid integer;
n_grid integer;
usng_coords varchar;
xmin numeric;
ymin numeric;
begin


-- parse UTM zone
select into zone cast( ( substring( usng from "^[[:digit:]]*") ) as integer );

-- derive set
if ( zone <= 6 ) then <br /> set := zone;
else
if ( zone % 6 = 0 ) then
set := 6;
else
set := zone % 6;
end if;
end if;

--- parse grid zone

select into grid_zone substring( usng from ( length(zone) + 1 ) for 1 );

-- parse grid zone squares
select into x_alpha_gsz substring( usng from ( length( zone) + 2 ) for 1 );
select into y_alpha_gsz substring( usng from ( length( zone ) + 3 ) for 1 );

-- calculate offset_north

select into offset_north
case
when ( grid_zone = "N" or grid_zone = "P" )
then 0
when ( grid_zone = "Q" and ( set % 2 ) = 1 and y_alpha_gsz <= "K" ) <br />
then 2000000
when ( grid_zone = "Q" and ( set % 2 ) = 1 and y_alpha_gsz >= "L" )
then 0
when ( grid_zone = "Q" and ( set % 2 ) = 0 and y_alpha_gsz >= "F" and
y_alpha_gsz <= "Q") <br /> then 2000000
when ( grid_zone = "Q" and ( set % 2 ) = 0 and ( y_alpha_gsz <= "E" or
```

```
y_alpha_gsz >= "R") )
then 0
when ( grid_zone = "R" )
then 2000000
when ( grid_zone = "S" and ( set % 2 ) = 1 and y_alpha_gsz <= "K" ) <br />
then 4000000
when ( grid_zone = "S" and ( set % 2 ) = 1 and y_alpha_gsz >= "L" )
then 2000000
when ( grid_zone = "S" and ( set % 2 ) = 0 and y_alpha_gsz >= "F" and
y_alpha_gsz <= "Q") <br /> then 4000000
when ( grid_zone = "S" and ( set % 2 ) = 0 and ( y_alpha_gsz <= "E" or
y_alpha_gsz >= "R") )
then 2000000
when ( grid_zone = "T" )
then 4000000
when ( grid_zone = "U" and ( set % 2 ) = 1 and y_alpha_gsz <= "C" ) <br />
then 6000000
when ( grid_zone = "U" and ( set % 2 ) = 1 and y_alpha_gsz >= "D" )
then 4000000
when ( grid_zone = "U" and ( set % 2 ) = 0 and y_alpha_gsz >= "F" and
y_alpha_gsz <= "H" ) <br /> then 6000000
when ( grid_zone = "U" and ( set % 2 ) = 0 and ( y_alpha_gsz <= "E" or
y_alpha_gsz >= "J" ) )
then 4000000
when ( grid_zone = "V" or grid_zone = "W" )
then 6000000
when ( grid_zone = "X" and ( set % 2 ) = 1 and y_alpha_gsz = "V" )
then 6000000
when ( grid_zone = "X" and ( set % 2 ) = 1 and y_alpha_gsz = "V" )
then 8000000
when ( grid_zone = "X" and ( set % 2 ) = 0 and y_alpha_gsz = "E" )
then 6000000
when ( grid_zone = "X" and ( set % 2 ) = 0 and y_alpha_gsz = "E" )
then 8000000
end;

-- construct arrays describing grid zone squares
select into e100k_grp1 array["A","B","C","D","E","F","G","H"];
select into e100k_grp2 array["J","K","L","M","N","P","Q","R"];
select into e100k_grp3 array["S","T","U","V","W","X","Y","Z"];
select into n100k_grp1
array["A","B","C","D","E","F","G","H","J","K","L","M","N","P","Q","R","S","T"
,"U","V"];
select into n100k_grp2
array["F","G","H","J","K","L","M","N","P","Q","R","S","T","U","V","A","B","C"
```

	,"D","E"]; -- derive X coordinate for grid zone square for e_gsz in 1 .. 8 loop if (set = 1 or set = 4) then if (x_alpha_gsz = e100k_grp1[e_gsz]) then e_coord := 100000 * e_gsz; exit; end if; elsif (set = 2 or set = 5) then if (x_alpha_gsz = e100k_grp2[e_gsz]) then e_coord := 100000 * e_gsz; exit; end if; else if (x_alpha_gsz = e100k_grp3[e_gsz]) then e_coord := 100000 * e_gsz; exit; end if; end if; end loop; -- derive Y coordinate for grid zone square for n_gsz in 1 .. 20 loop if (set = 1 or set = 3 or set = 5) then if (y_alpha_gsz = n100k_grp1[n_gsz]) then n_coord = 100000 * (n_gsz - 1); end if; elsif(set = 2 or set = 4 or set = 6) then if (y_alpha_gsz = n100k_grp2[n_gsz]) then n_coord = 100000 * (n_gsz - 1); end if; end if; end loop; -- derive grid coordinates and precision grid = substring(usng, "[[:digit:]]*$"); select into e_grid case when length(grid) = 2 then (cast(substring(grid from 1 for 1) as integer)) * 10000

	when length(grid) = 4 then (cast(substring(grid from 1 for 2) as integer)) * 1000 when length(grid) = 6 then (cast(substring(grid from 1 for 3) as integer)) * 100 when length(grid) = 8 then (cast(substring(grid from 1 for 4) as integer)) * 10 when length(grid) = 10 then cast(substring(grid from 1 for 5) as integer) end; select into n_grid case when length(grid) = 2 then (cast(substring(grid from 2 for 1) as integer)) * 10000 when length(grid) = 4 then (cast(substring(grid from 3 for 2) as integer)) * 1000 when length(grid) = 6 then (cast(substring(grid from 4 for 3) as integer)) * 100 when length(grid) = 8 then (cast(substring(grid from 5 for 4) as integer)) * 10 when length(grid) = 10 then cast(substring(grid from 6 for 5) as integer) end; select into precision case when length(grid) = 2 then 10000 when length(grid) = 4 then 1000 when length(grid) = 6 then 100 when length(grid) = 8 then 10 when length(grid) = 10 then 1 end; -- create usng coords xmin = round((e_coord + e_grid + (precision / 2)), 1); ymin = round((offset_north + n_coord + n_grid + (precision / 2)), 1); usng_coords = xmin || " " || ymin ;

	return (usng_coords); end; ' language 'plpgsql';

4.7.39 XY Coordinate Completeness Measure

Measure Name	XY Coordinate Completeness Measure
Measure Description	Checks for coordinates pairs with one member missing. The query produces a list of incomplete coordinates.
Report	Logical consistency
Evaluation Procedure	Query for null values.
Spatial Data Required	Address X Coordinate, Address Y Coordinate
Pseudocode Example: Testing records	**Query** SELECT Address X Coordinate, Address Y Coordinate FROM Address Collection WHERE Address X Coordinate isnull OR Address Y Coordinate isnull **Result Without Anomalies** Address X Coordinate | Address Y Coordinate -----------------+--------------------- **Anomalies** Address X Coordinate | Address Y Coordinate -----------------+--------------------- x1 | x2 | | y3 .... | ....
Pseudocode Example: Testing the Conformance of a Data Set	**Function** See Perc Conforming for the query example. **Function Parameters** • count_of_nonconforming_records

	*SELECT COUNT(*)* *FROM Address Collection* *WHERE Address X Coordinate isnull OR Address Y Coordinate isnull* • count_of_total_records *SELECT COUNT(*)* *FROM Address Collection*
Result Report Example	Tested XY Coordinate Completeness Measure at 93% conformance.

2907 ### 4.7.40 XY Coordinate Spatial Measure

Measure Name	XY Coordinate Spatial Measure
Measure Description	Compare the coordinate location of the addressed object with the coordinate attributes. The measure applies to both types of coordinate pairs listed in Part One: Address X Coordinate, Address Y Coordinate and Address Longitude, Address Latitude. The query produces a list of coordinate values in the address collection that do not conform to a spatial domain.
Report	Positional accuracy
Evaluation Procedure	Intersect the addressed spatial object with the coordinate attributes. Note that the spatial referencing system of the addressed spatial objects must match that of the coordinate attributes.
Spatial Data Required	Address X Coordinate, Address Y Coordinate
Pseudocode Example: Testing records	**Query** SELECT Address X Coordinate, Address Y Coordinate FROM Address Collection WHERE NOT(EQUALS(Addressed Object.Geometry, MakePoint(Address X Coordinate, Address Y Coordinate, Address Z Coordinate))) **Result Without Anomalies** Address X Coordinate | Address Y Coordinate -----------------+-------------------- **Anomalies**

	Address X Coordinate | Address Y Coordinate -----------------+--------------------- x1 | y1 x2 | y2 x3 | y3 .... | ....
Pseudocode Example: Testing the Conformance of a Data Set	**Function** See Perc Conforming for the query example. **Function Parameters** • count_of_nonconforming_records *SELECT COUNT(*)* *FROM Address Collection* *WHERE NOT(EQUALS(Addressed Object.Geometry, Point(Address X Coordinate, Address Y Coordinate)))* • count_of_total_records *SELECT COUNT(*)* *FROM Address Collection*
Result Report Example	Tested XY Coordinate Spatial Measure at 90% conformance.

5. Address Data Exchange

5.1 Introduction

The purpose of this section is three-fold: to provide a template for the XML documents and metadata that will move addresses from place to place, to provide information on preparing address data to be packaged, and to provide information on unpackaging address data that has been received.

Historically, the data format aspect of data exchange has impeded the flow of information. By providing a single and flexible data structure for exchanging street address data, the Address

Standard will simplify the implementation of data exchanges, making them more reliable and less likely to need small changes, especially over time. Local data processing systems and applications change over time and frequently data exchange programs and reports must be rewritten along with those changes. Such changes may be as seemingly minor as the renaming of a data element, shortening or extending the length of a field, or the addition or subtraction of a field. When new data sharing partners are identified, a data format for sharing data with that partner must be constructed and implemented by each party. The Address Standard aims to minimize local changes necessary when upgrading computer systems and to provide a structure that can be reused by all data sharing parties without their having to implement something new.

The data sharing benefits of the Address Standard will only be realized when local agencies have implemented both export and import engines to process exchanged street addresses. The initial implementation of these data engines or programs will provide a lasting benefit to implementers in that once created, the agency will never again need to be concerned with creating programs or engines to share data with any new data sharing partners that they identify in the future.

The Address Standard is designed to be flexible enough to fit within current data sharing methods. There are two basic forms of sharing data between parties:

- Monolithic, in which all records are in the exchange package.
- Transactional, in which the exchange package records include commands to add or remove a record from the local copy of all records.

The Standard supports both of these forms, using a slightly modified structure to enable transactional exchanges.

5.2 Structure of a Transfer Package

All packages of address data to be exchanged must include:

- FGDC Metadata, conforming to the FGDC-STD-001-1998 *Content Standard for Digital Geospatial Metadata*, Version 2.0 (see Appendix A for a complete citation).
- Address data, expressed as an XML document conforming to the AddrStd XML Schema.

5.2.1 FGDC Metadata

Metadata provides a common set of terminology and definitions for the documentation of digital geospatial data (CSDGM, Introduction). It is a required part of all Federal standards, and is required of all federally generated geospatial data per Executive Order 12906. The transfer of data always needs to be accompanied by copyright information, use restrictions, contact information, data lineage information, known data defects and a description of the geographic area that the data represents. The *Content Standard for Digital Geospatial Metadata* provides a uniform, consistent and well known way to express those things amongst others.

5.2.2 Address Data

The Address Standard XML schema is a way of packaging address data such that only fields meaningful to the particular data transfer package need to be included. The nature of XML is that only meaningful data is included but the meaning of everything that could be included is documented. In addition, XML data transfer implementations can be extended without breaking existing implementations. Existing implementations will not understand the extensions but by definition will ignore them.

Data is produced by agencies possessing address information and consumed by those receiving the address data. Many agencies will be both producers and consumers at different times. The roles of producer and consumer describe, respectively, the activity at hand when exporting or importing address data.

5.2.2.1 Exporting Data

A data producer will follow these basic steps while implementing an export engine:

- Construct a logical map of local data fields into the equivalent Address Standard Content and Classification elements.
- Write programs or subroutines to split local fields into the Address Standard elements if necessary.
- Collect support information required by the CSDGM metadata into a accessible place.
- Optionally, write programs or subroutines to automate the CSDGM "Data Quality" tests documented in the Data Quality section of this standard.

- Write programs or subroutines to include the CSDGM support data into a complete and valid CSDGM document.

A data producer will follow these basic steps while creating a package of address data:

- Run the Data Quality tests and collect the reports into the CSDGM metadata.
- Set the "Publication Date" element of the CSDGM metadata to be the time the package was created.
- Run the data remapping and splitting programs.
- Set the "DirectSource" element of the Address Standard to be the producer's ID.
- Set the "AddressID" and "AuthorityID" elements of the Address Standard for any addresses created by the producer.
- Export the data into the Address Standard XML format.
- Transfer both the Address Standard XML document and the CSDGM document to another party.

5.2.2.2 Importing data

A data consumer will follow these basic steps while implementing an export engine:

- Construct a logical map of local data fields into the equivalent Address Standard Content and Classification elements.
- Write programs or subroutines to combine Address Standard elements into local data fields, if necessary.

- Create a place to store the CSDGM data from received packages.
- Optionally write programs or subroutines to automate the CSDGM "Data Quality" tests documented in the Data Quality section of this standard on the received data.

A data consumer will follow these basic steps while importing a package of address data:

- Receive both the Address Standard XML document and the CSDGM document from another party.
- Parse the Address Standard XML document into a working area.
- Parse the CSDGM XML document into a working area.
- Run the Data Quality tests and compare to the report in the CSDGM metadata received.
- Run the data remapping and combining programs.
- Import from the working area to the local production database.

When mapping local data fields into the equivalent Address Standard Content and Classification elements, or the reverse, it is important to understand that the Address Standard is set up to allow address producers to directly and unequivocally express the taxonomy of their own addresses. The Content and Classification sections provide a taxonomy to help parse addresses into descriptive elements.

For example, given an address such as **225 North Avenue Northwest Atlanta GA 30318**, the Address Standard allows the the address producer to state that the word **North** is not a Street

Name Pre Directional but is actually a Street Name. When stated by the actual addressing authority, it should be taken as factual and not converted.

Within other agencies, database design requirements might cause the address to be stored differently, but they should record the **official** form somewhere within their databases.

It is important, if distributing data received from other address authorities, that their taxonomy or parsing of addresses into elements be maintained and be reproducible.

5.2.3 The Address Standard XSD Data Model (see Appendix H for the complete XSD document)

5.2.3.1 General Notes on the XML schema

Content and Classification use the word **element** in a way that differs slightly from its use in designing XML document schemes.

- Content and Classification use **element** to describe a taxonomy facet for parsing an address.
- XML Scheme Document (XSD) uses **element** to describe an XML tag.

Some Content and Classification **elements** become XSD elements and others become XSD attributes of other XSD elements.

The Address Standard XSD has been designed by creating a simple type for almost every thing in Content and Classification. The simple data type is a place to describe the form of data that populates the simple type. Many times no attempt to provide an automatable test for

correctness of form is given in the XSD. Local implementers may attempt such tests outside the scope of the Address standard.

From the simple types simple elements are created. Simple elements and some simple types cluster into complex elements. Finally, elements are gathered into the **global** elements that comprise the top level XML data types.

5.2.3.2 Relation of the Address Standard XSD data model to the Content and Classification parts.

A crosswalk chart relating the Content and Classification elements into XSD classes, types, elements and attributes.

Classes	XSD Type Name	Simple or Complex (in XSD terms)	Element or Attribute Name	XSD	Parent XSD class
NumbereredThoroughfareAddress	NumberedThoroughfareAddress_type	Complex	NumberedThoroughfareAddress	Global Element	Global, AddressCollection
IntersectionAddress	IntersectionAddress_type	Complex	IntersectionAddress	Global Element	Global, AddressCollection
TwoNumberAddressRang	TwoNumberAddressRange_type	Complex	TwoNumberAddress	Global Element	Global, AddressColl

e			Range		ection
FourNumberAddressRange	FourNumberAddressRange_type	Complex	FourNumberAddressRange	Global Element	Global, AddressCollection
Unnumbered Thoroughfare Address	UnnumberedThoroughfareAddress_type	Complex	UnnumberedThoroughfareAddress	Global Element	Global, AddressCollection
LandmarkAddress	LandmarkAddress_type	Complex	Landmark Address	Global Element	Global, AddressCollection
Community (Urbanization) Address	CommunityAddress_type	Complex	CommunityAddress	Global Element	Global, AddressCollection
USPS Postal Delivery Box	USPSPostalDeliveryBox_type	Complex	USPSPostalDeliveryBox	Global Element	Global, AddressCollection
USPS Postal Delivery Route	USPSPostalDeliveryRoute_type	Complex	USPSPostalDeliveryRoute	Global Element	Global, AddressCollection
USPS	USPSGeneralDeliveryAddress	Complex	USPSGene	Global	Global,

General Delivery Address	_type		ralDelivery Address	Element	AddressCollection
General Address	GeneralAddressClass_type	Complex	GeneralAddressClass	Global Element	Global, AddressCollection
AddressReferenceSystem	AddressReferenceSystem_type	Complex	AddressReferenceSystem	Global Element	Global, AddressCollection
	AddressCollection_type	Complex	AddressCollection	Global Element	
Complete Address Number	CompleteAddressNumber_type	Complex	CompleteAddressNumber	Element	Various, AddressNumberRange
Address Number Prefix	AddressNumberPrefix_type	Simple	Prefix	Element	CompleteAddressNumber, AddressNumberRange
Address Number	AddressNumber_type	Simple	Number	Element	CompleteAddressNumbe

					r, AddressNumberRange
Address Number Suffix	AddressNumberSuffix_type	Simple	Suffix	Element	CompleteAddressNumber, AddressNumberRange
SeparatorElement	SeparatorElement_type	Simple	SeparatorElement	Attribute	CompleteAddressNumber, AddressNumberRange
CompleteStreetName	CompleteStreetName_type	Complex	CompleteStreetName	Element	Global objects
Street Name Pre-modifier	StreetNameModifier_type	Simple	PreModifier	Element	CompleteStreetName
Street Name Pre-directional	StreetNameDirectional_type	Simple	PreDirectional	Element	CompleteStreetName
Street Pre-	StreetNameType_type	Simple	PreType	Element	CompleteStr

type					eetName
Street Name	StreetName_type	Simple	StreetName	Element	CompleteStreetName
Street Post-type	StreetNameType_type	Simple	PostType	Element	CompleteStreetName
Street Post-directional	StreetNameDirectional_type	Simple	PostDirectional	Element	CompleteStreetName
Street Name Post-modifier	StreetNameModifier_type	Simple	PostModifier	Element	CompleteStreetName
CompleteSubaddress	CompleteSubaddress_type	Complex	CompleteSubaddress	Element	Global objects
Subaddress Type	SubaddressType_type	Simple	Subaddress Type	Element	Building
Subaddress Identifier	SubaddressIdentifier_type	Simple	Subaddress Identifier	Element	Building
CompleteLandmarkName	Set the "AddressID" and "AuthorityID" elements of the Address Standard for any addresses created by the producer.ompleteLandmarkName_type	Simple	CompleteLandmarkName	Element	Global objects

Landmark Name	LandmarkName_type	Complex	Landmark Name	Element	LandmarkAddress
Community (Urbanization) Place Name	CommunityPlaceName_type	Simple	CommunityPlaceName	Element	PlaceName, LandmarkSiteAddress,CommunityAddress
CompletePlaceName	completePlaceName_type	Complex	CompletePlaceName	Element	Global objects
PlaceName	PlaceName_type	Simple	PlaceName	Element	CompletePlaceName
PlaceNameType	PlaceNameType_type	Simple	PlaceName Type	Attribute	PlaceName
County Name	CountyName_type	Simple	CountyName_type	Element	PlaceName
State	StateName_type	Simple	StateName	Element	Global objects
ZIP Code	ZIPCode_type	Simple	ZIPCode	Element	Global objects
ZIP+4 Code	ZIPPlus4_type	Simple	ZIPPlus4	Element	Global objects

CountryName	CountryName_type	Simple	CountryName	Element	Global objects
USPS Box Type	USPSBoxType_type	Simple	USPSBoxType	Attribute	USPSBox
USPS Box ID	USPSBoxId_type	Simple	USPSBoxId		USPSBox
	USPSBox_type	Complex	USPSBox	Element	UPSPPostal Delivery Classes
USPS Box Group Type	USPSBoxGroupType_type	Simple	USPSBox GroupType	Attribute	USPSBoxGroup
USPS Box Group ID	USPSBoxGroupId_type	Simple	USPSBox GroupId		USPSBoxGroup
	USPSBoxGroup_type	Complex	USPSBox Group	Element	UPSPPostal Delivery Classes
USPS General Delivery Point	USPSGeneralDeliveryPoint_type	Simple	USPSGeneralDelivery Point	Element	UPSPPostal Delivery Classes
DeliveryAddr	CompleteFeatureAddress_type	Simple	DeliveryAd	Element	GeneralAddr

ess			dress		ess
Place State ZIP	PlaceStateZip_type	Simple	PlaceState Zip	Element	GeneralAddr ess
Address ID	AddressId_type	Simple	AddressId	Element	Global objects
Address X Coordinate	AddressXCoordinate_type	Simple	X	Element	XYCoordina te
Address Y Coordinate	AddressXCoordinate_type	Simple	Y	Element	XYCoordina te
Address Longitude	AddressLongitude_type	Simple	Longitude	Element	LongLat
Address Latitude	AddressLatitude_type	Simple	Latitude	Element	LongLat
US National Grid Coordinate	LocationUSNG_type	Simple	USNGCoo rdinate	Element	Global objects
Address Z Value	AddressZValue_type	Simple	Zvalue	Element	Global objects
Address Classification		Internal		Internal to model	The XSD element

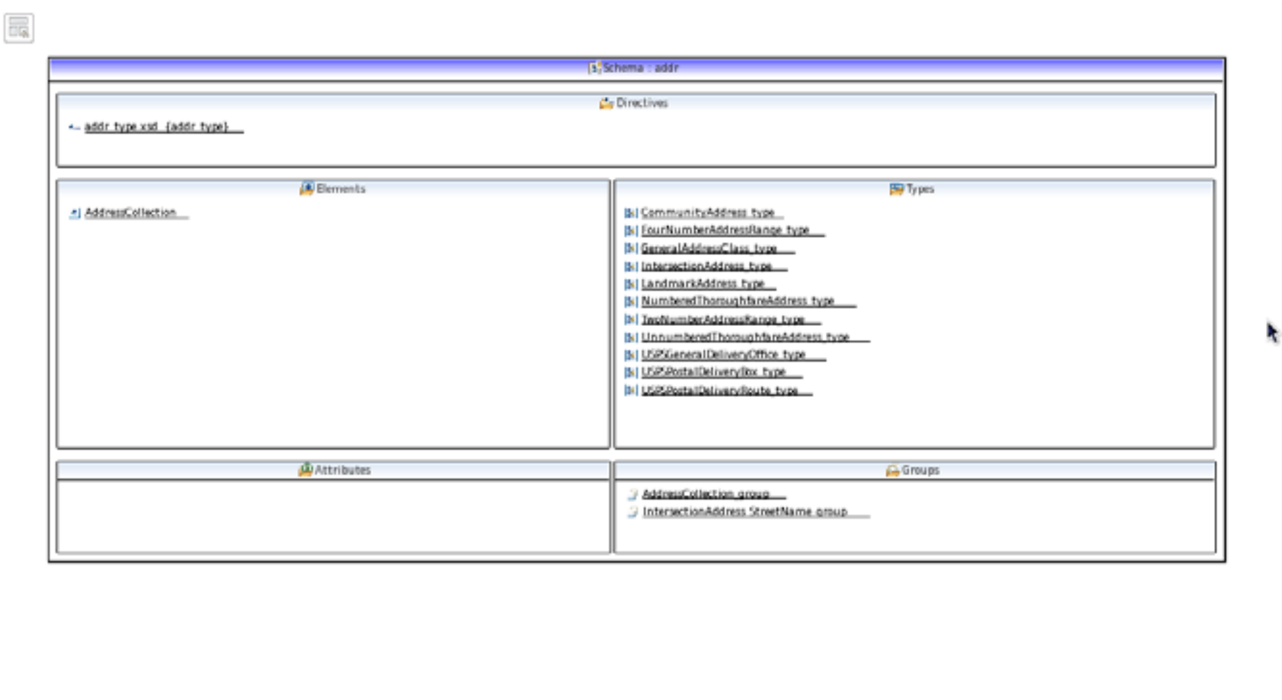
					name stores this information
Feature Type	FeatureType_type	Simple	AddressFeatureType	Element	Global objects
Address Lifecycle Status	AddressLifecycleStatus_type	Simple	AddressLifecycleStatus	Element	Global objects
Address Official Status	AddressOfficialStatus_type	Simple	OfficialStatus	Element	Global objects
Address Anomaly Status	AddressAnomalyStatus_type	Simple	AddressAnomalyStatus	Element	Global objects
Address Range Type	AddressRangeType_type	Simple	AddressRangeType	Element	Global objects
Location Description	LocationDescription_type	Simple	LocationDescription	Element	Global objects
Address Number Parity	AddressNumberParity_type	Simple	Parity	Attribute	TwoNumberAddressRange

Address ReferenceSystem Name	AddressReferenceSystemName_type	Simple	AddressReferenceSystemName_type	Element	Global objects
Address ReferenceSystem Axis	AddressReferenceSystemeAxes_type	Simple	AddressReferenceSystemAxis_type	Element	AddressReferenceSystem
Address ReferenceSystem DocumentCitation	AddressReferenceSystemDocumentCitation_type	Simple	AddressDocumentCitation_type	Element	AddressReferenceSystem
Address ReferenceSystem Origin	AddressReferenceSystemOrigin_type	Simple	AddressReferenceSystemOrigin_type	Element	AddressReferenceSystem
Address ReferenceSystem Extent	AddressReferenceSystemExtent_type	Simple	AddressReferenceSystemExtent_type	Element	AddressReferenceSystem

Address ReferenceSystem	AddressReferenceSystem_type	Complex	AddressReferenceSystem	Global Element	Global
Address Start Date	AddressStartDate_type	Simple	AddressStartDate	Element	Global objects
Address End Date	AddressEndDate_type	Simple	AddressEndDate	Element	Global objects
Address Direct Source	AddressDirectSource_type	Simple	AddressDirectSource	Element	Global objects
Address Authority	AddressAuthority_type	Simple	AddressAuthority	Element	Global objects
Address Authority Idenitifier	AddressAuthorityIdentifier_type	Simple	AddressAuthorityIdentifier	Element	Global objects

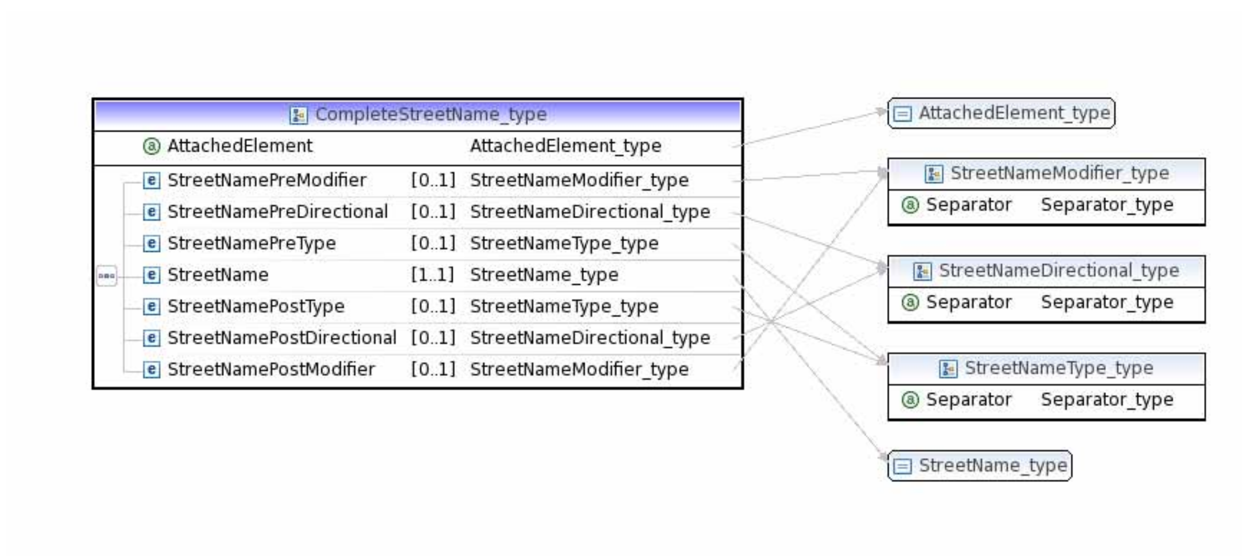
5.2.3.3 Diagrams of Elements of the XSD datamodel

- Data Model 0.4:

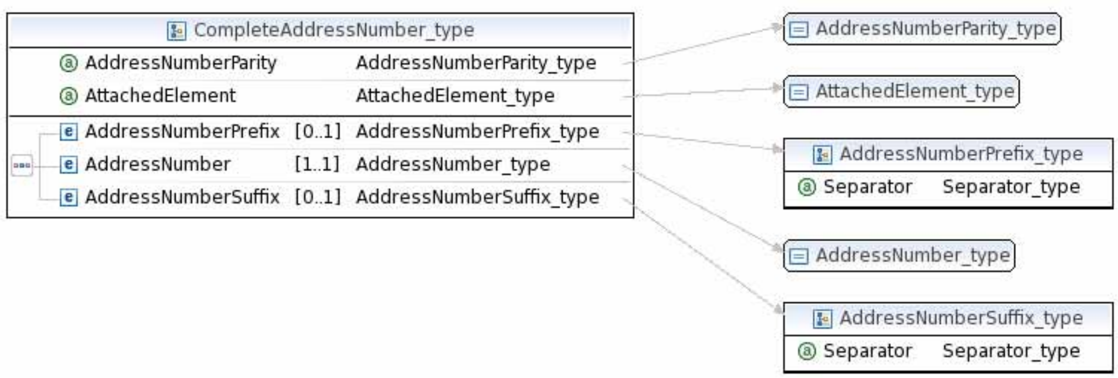


Complex Types

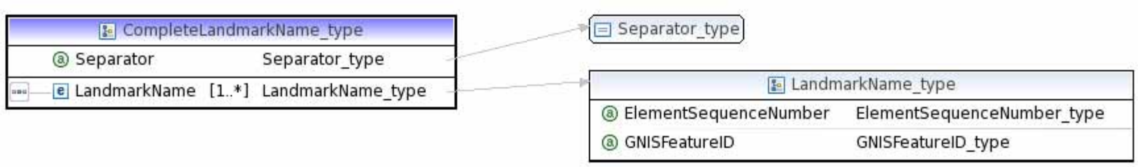
* [Complete Street Name] v0.4:



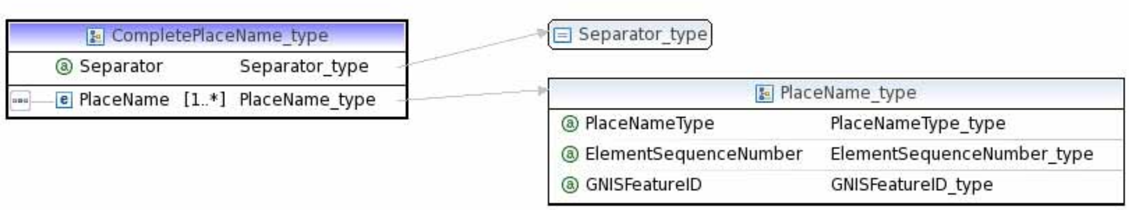
- Complete Address Number v0.4:



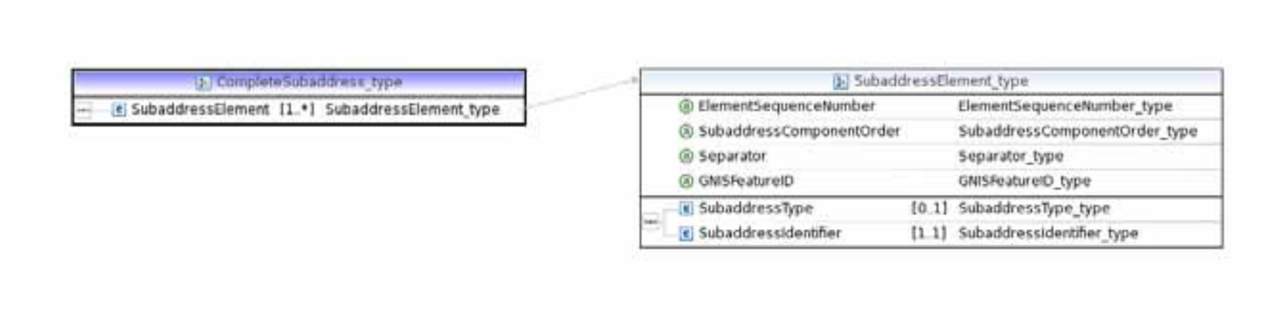
- Complete Landmark Name v0.4:



- Complete Place Name v0.4:

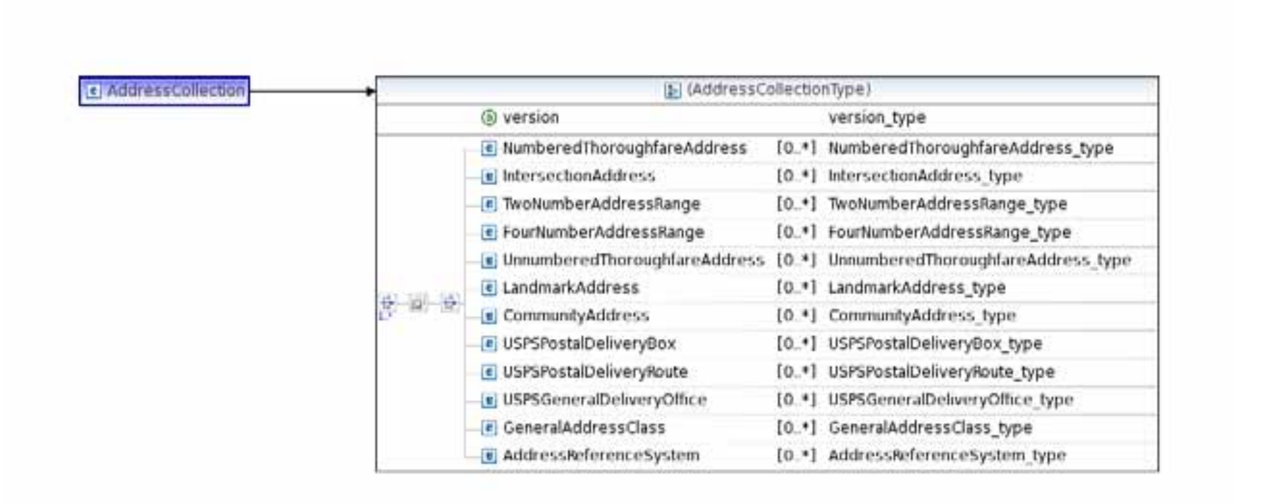


- Complete Subaddress v0.4:



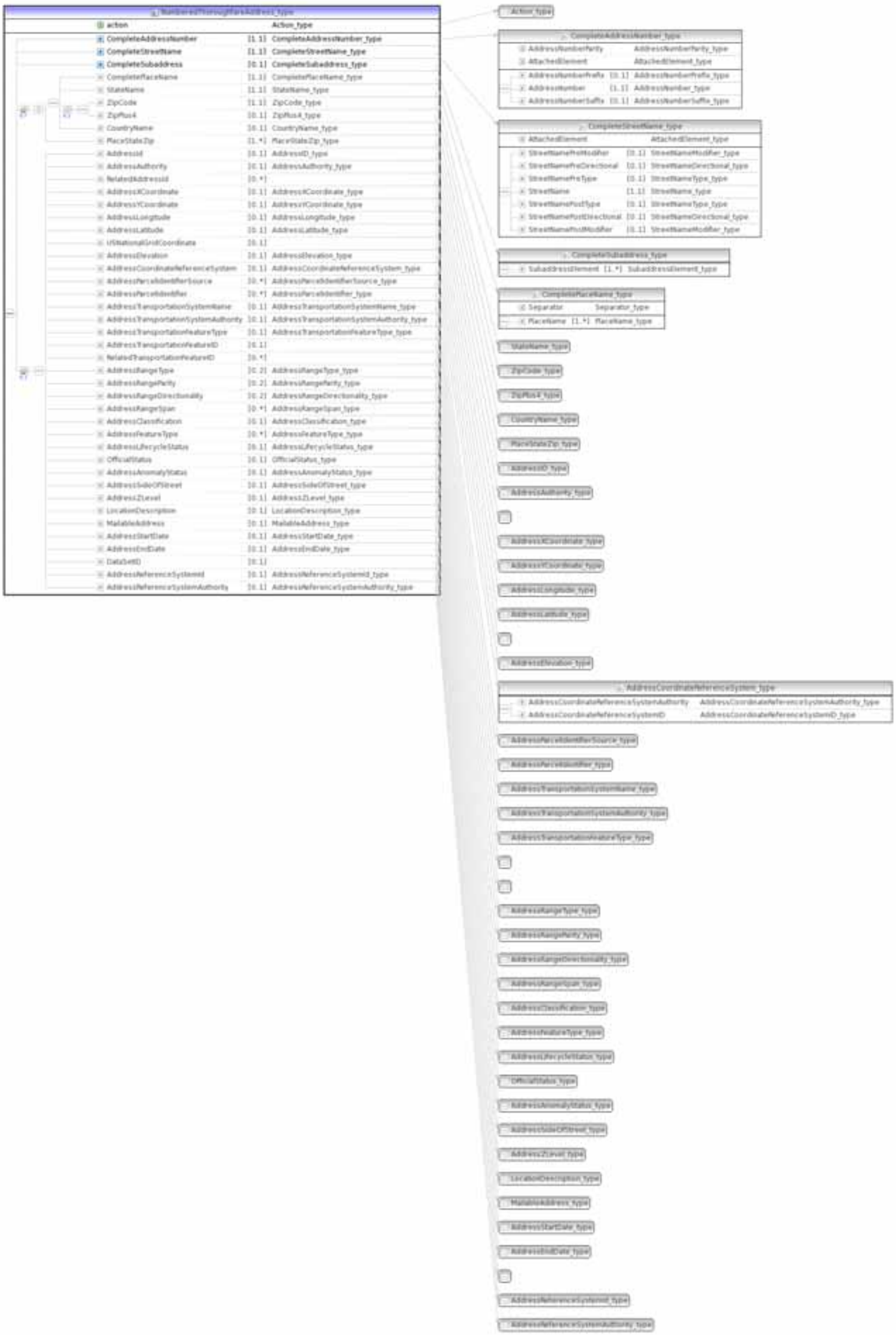
- Address Referencesystem? v0.4:

- Address Collection v0.4:

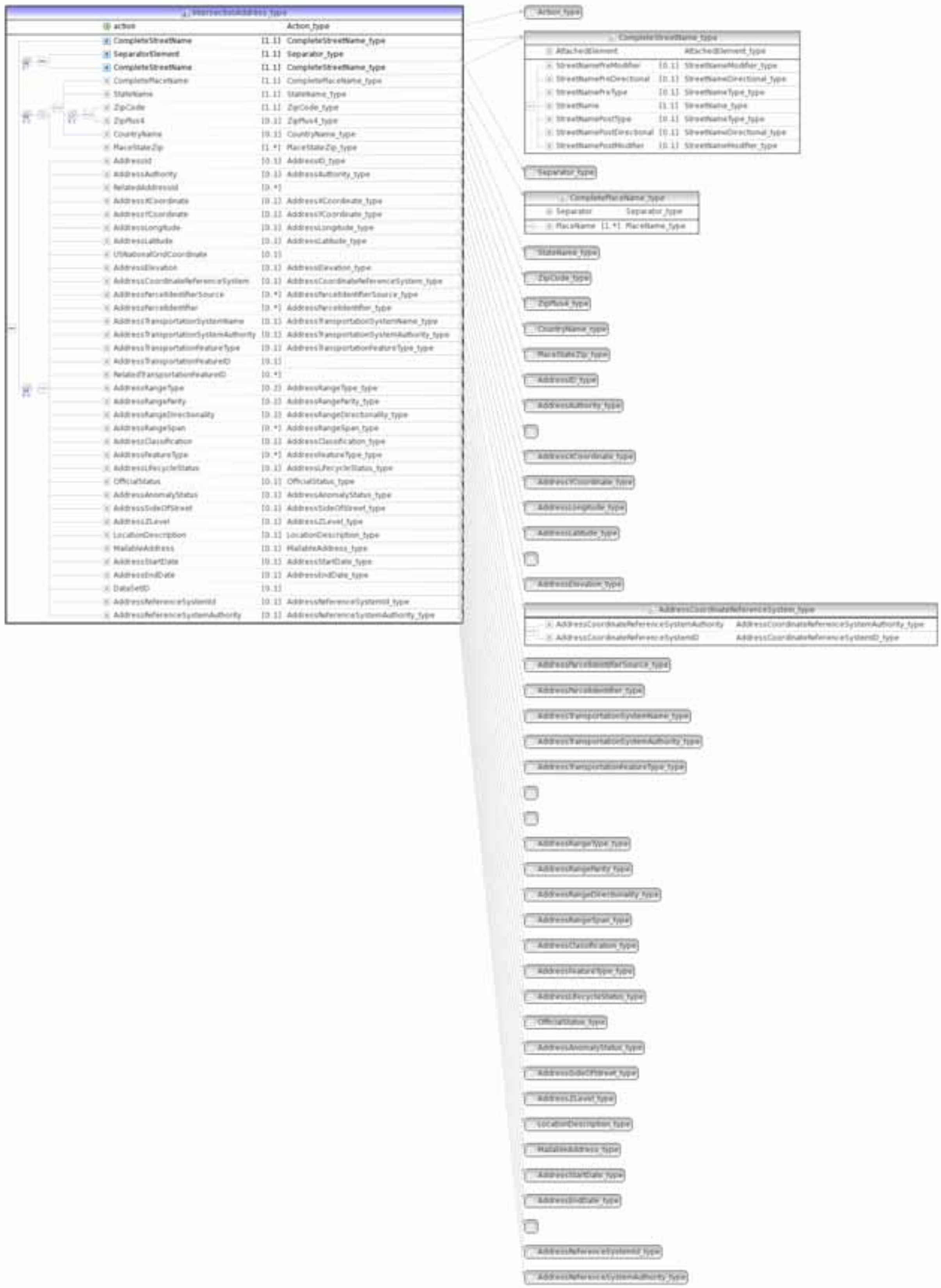


Thoroughfare Address Classes

Numbered Thoroughfare Address v0.4:

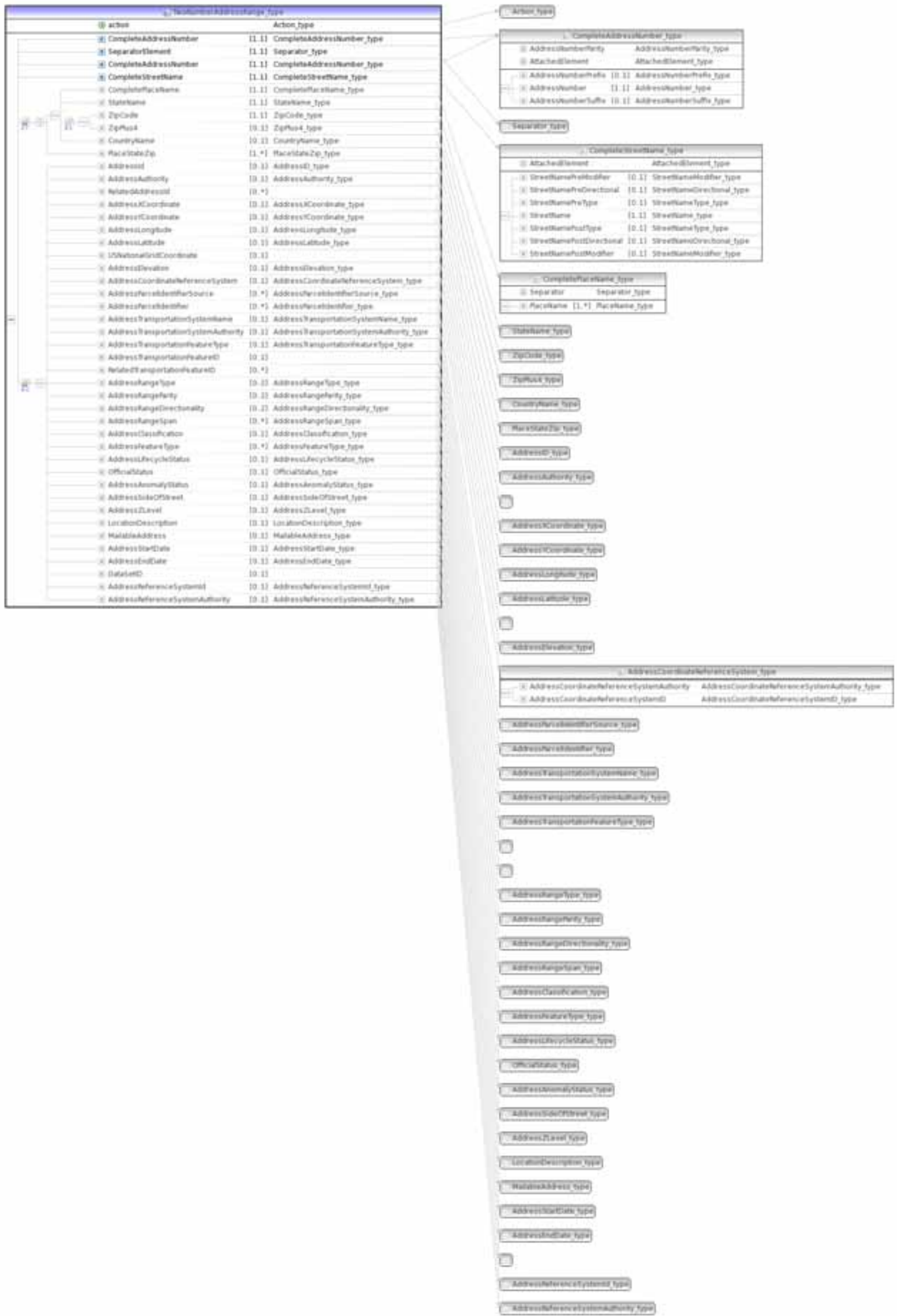


- Intersection Address v0.4:



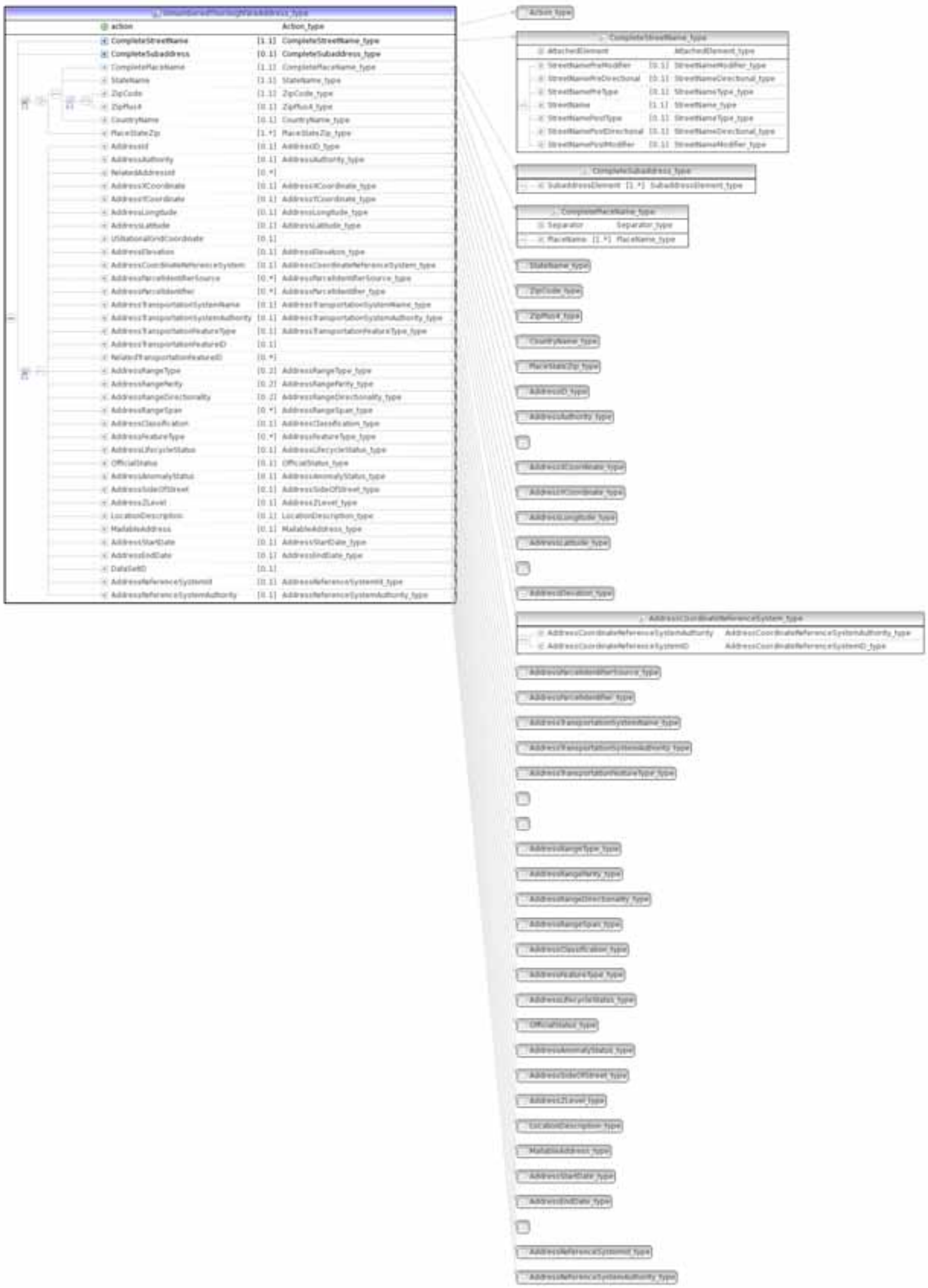
3071

- Two Number Address Range:



- Four Numbe Address Range? v0.4:

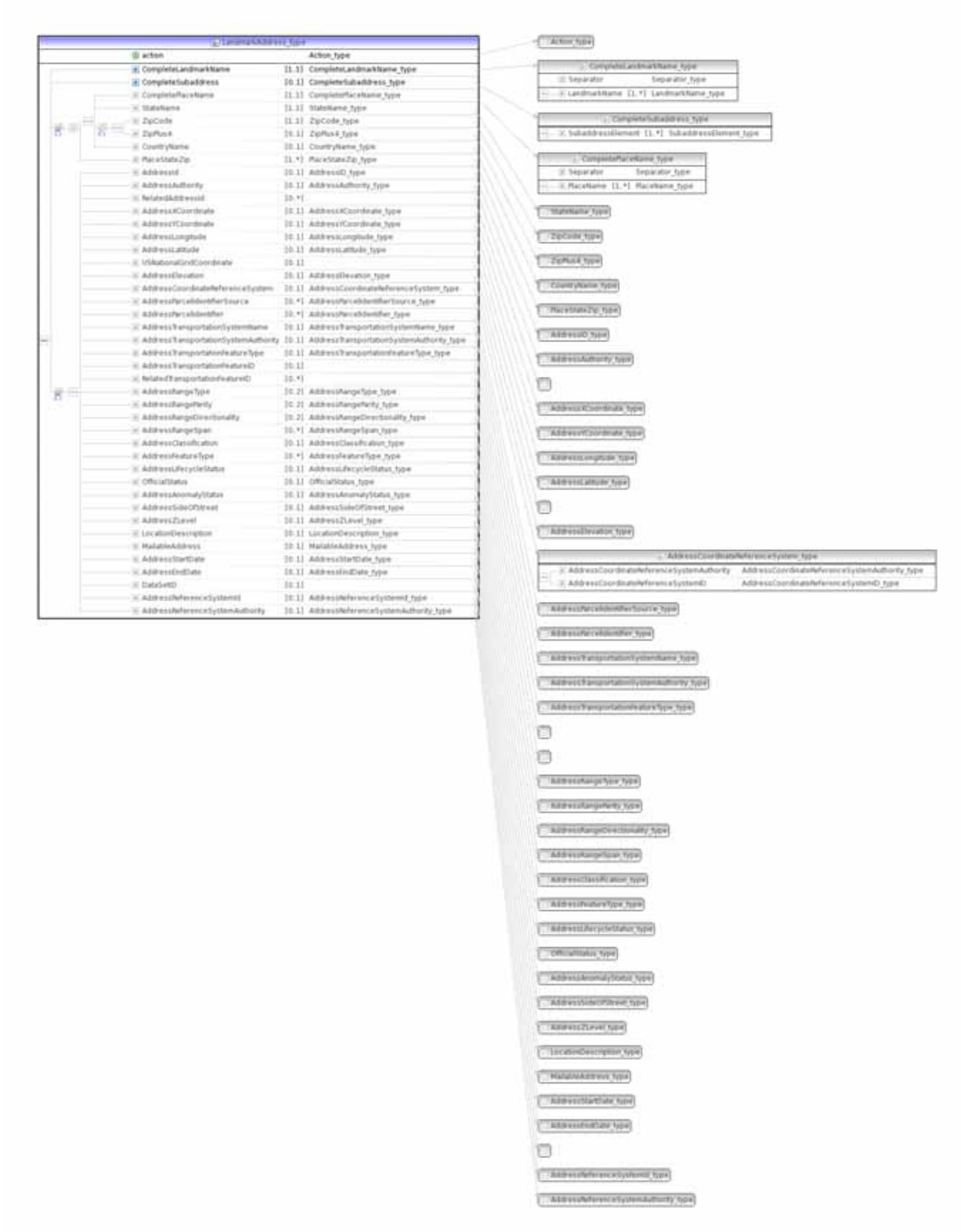
- Unnumbered Thoroughfare Address:



3080

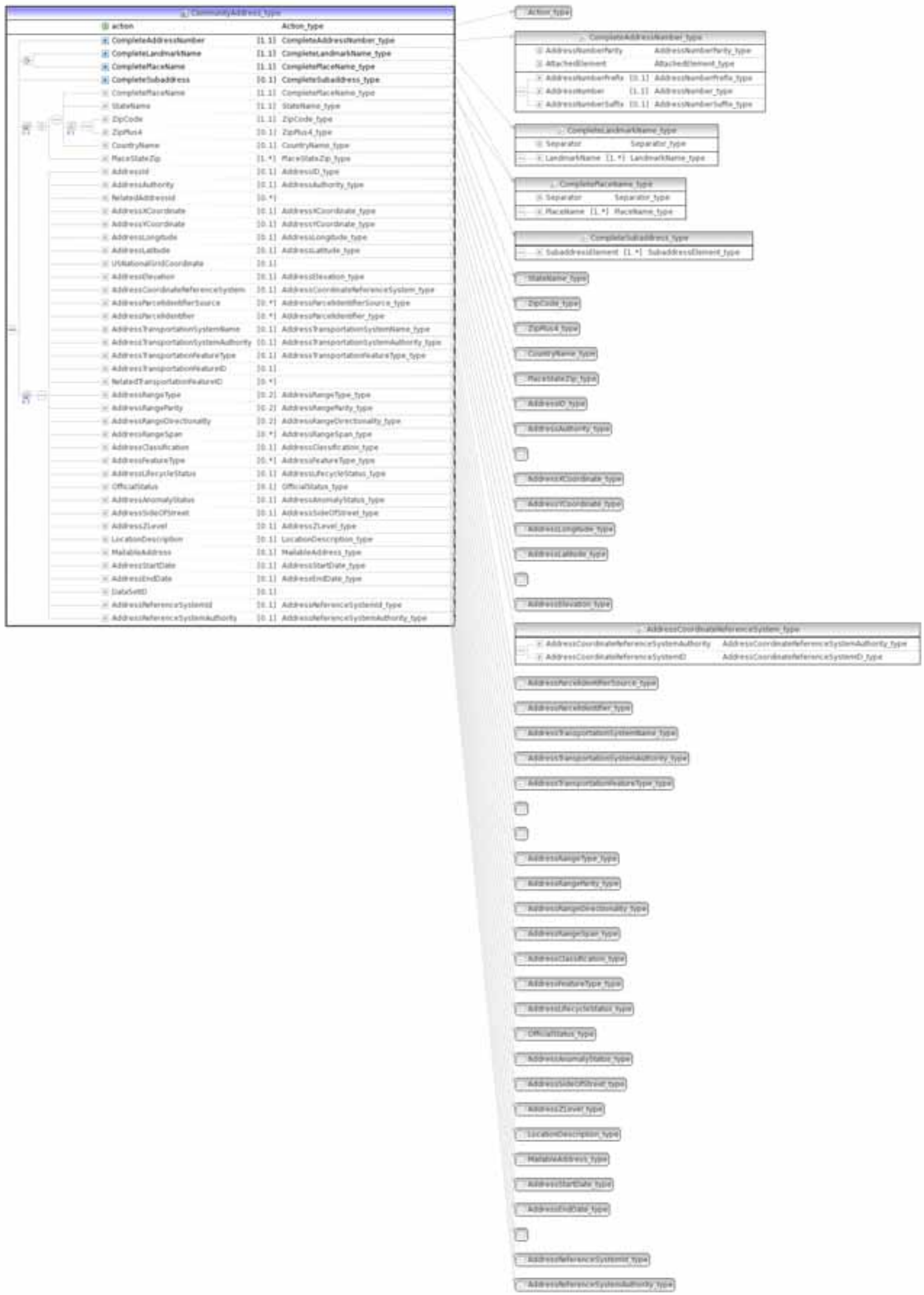
Landmark Address Classes

- Landmark Address v0.4:



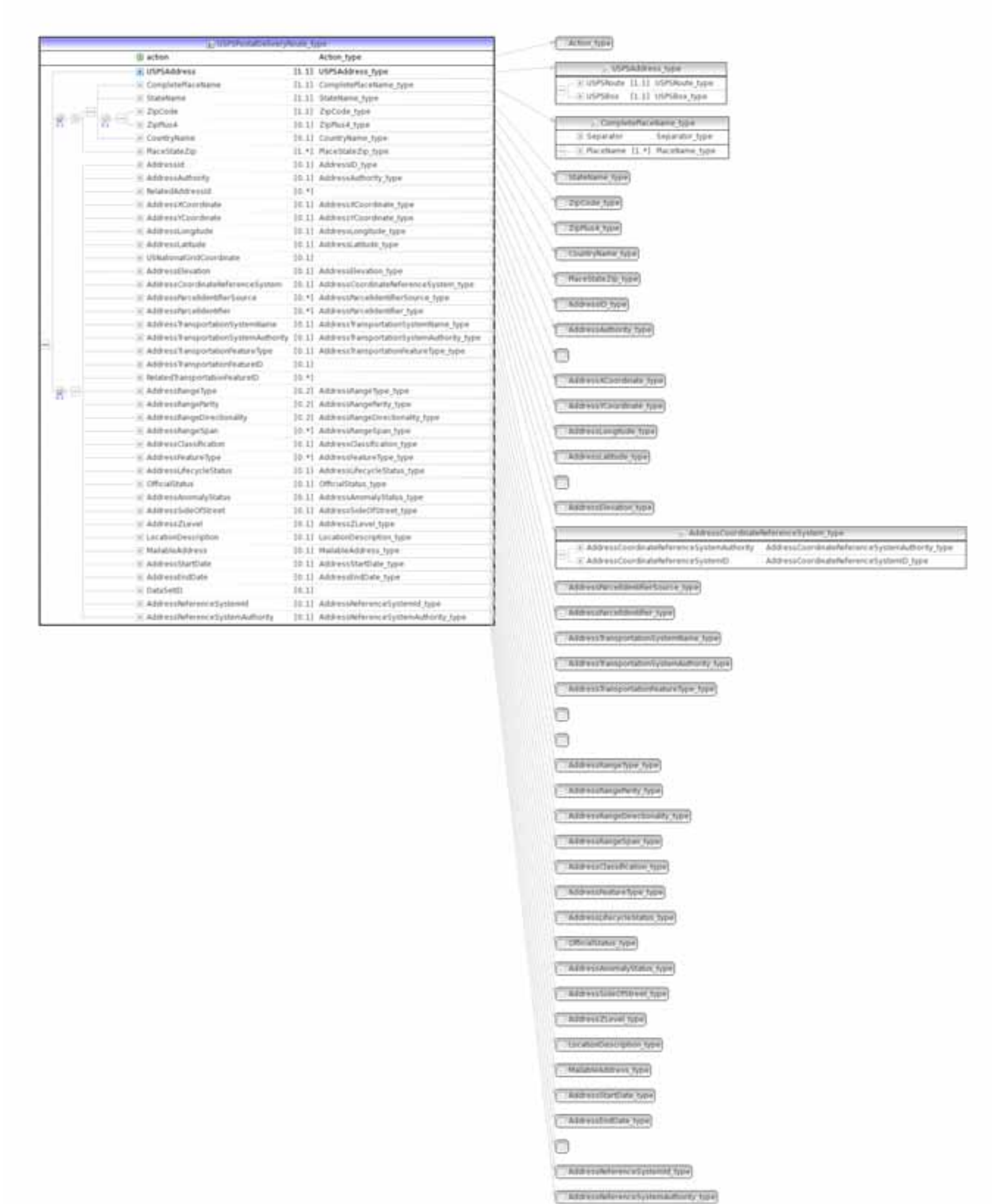
3083

- Community Address v0.4:

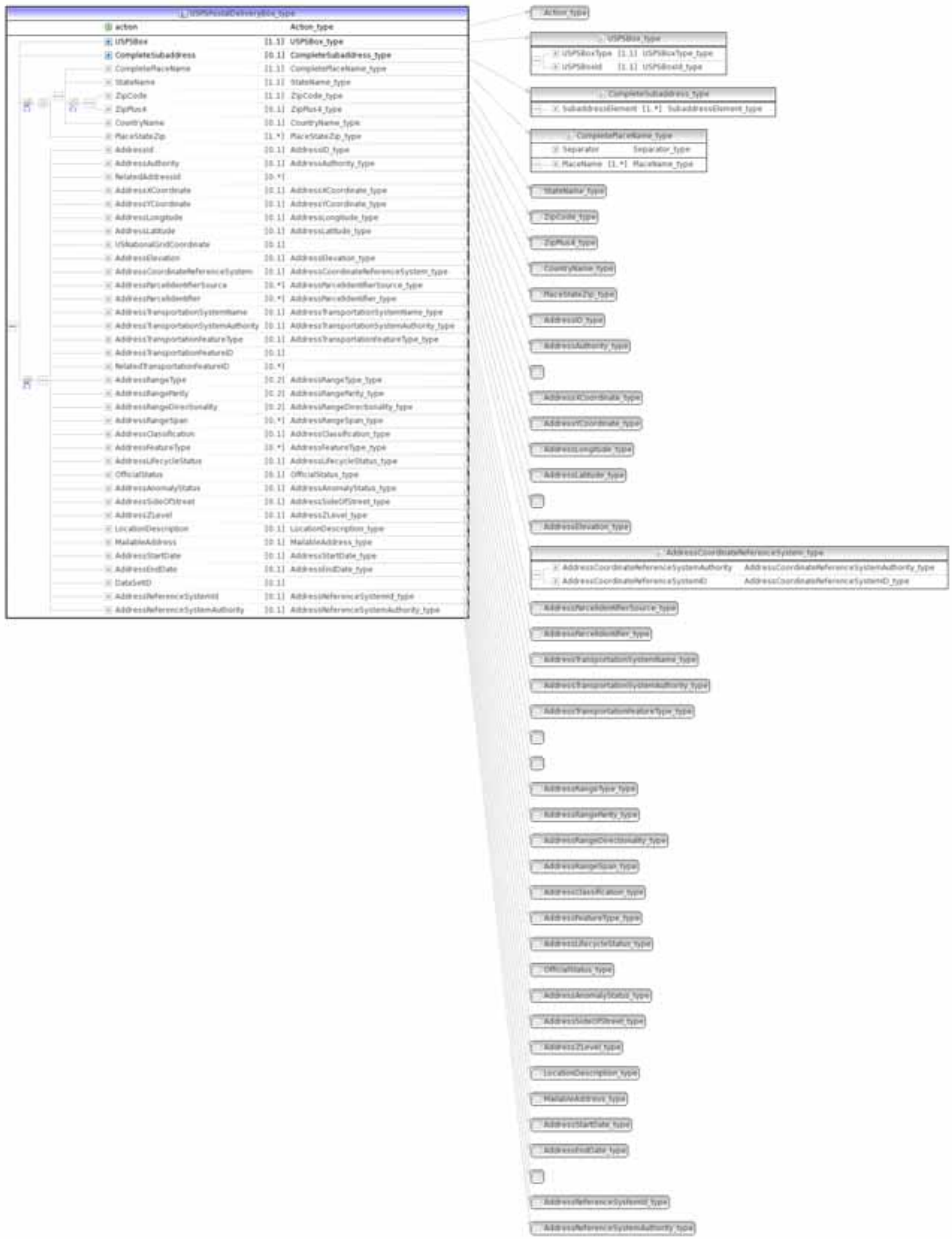


Postal Delivery Address Classes

- USPS Postal Delivery Route v0.4:

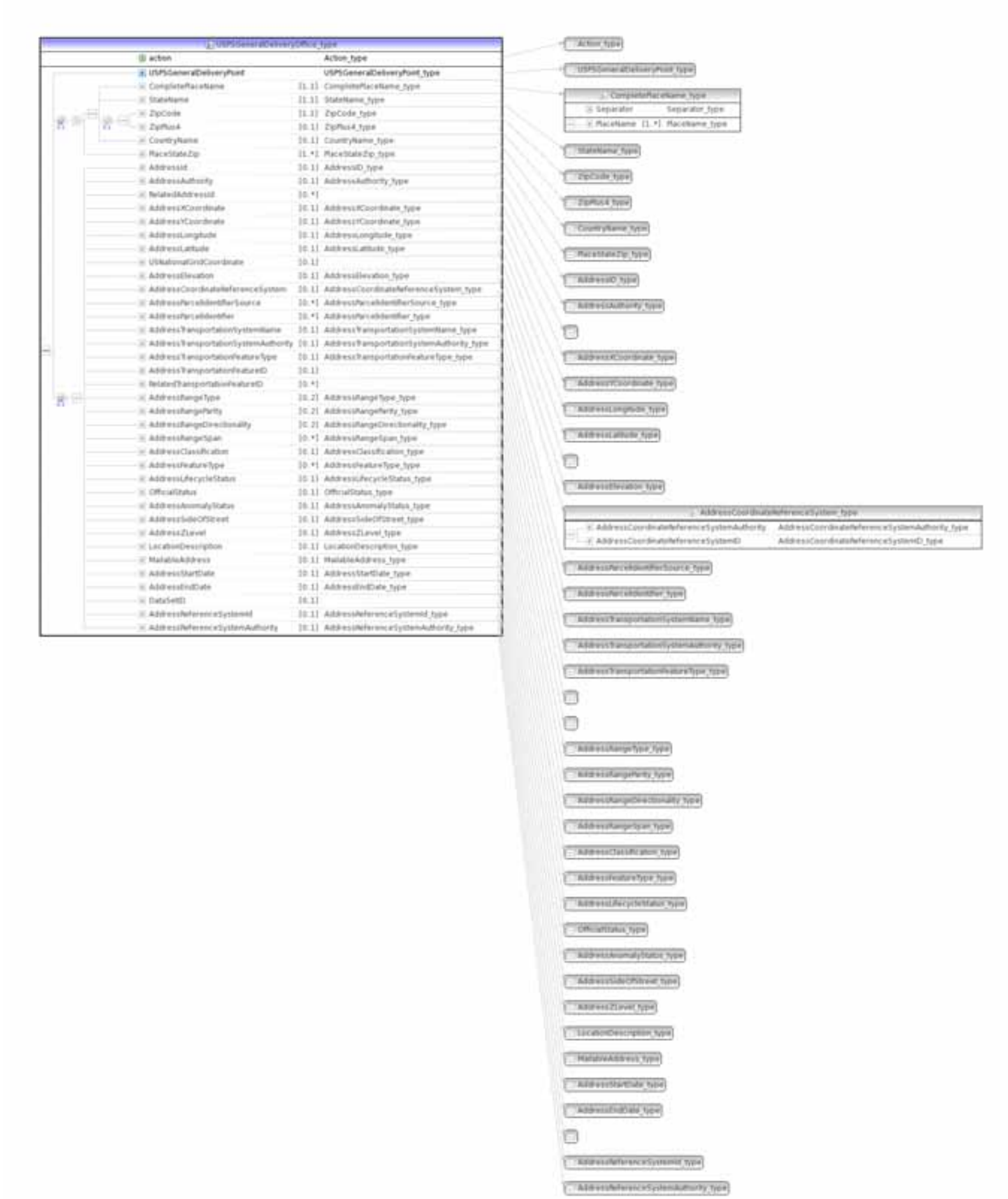


- UPS Postal Delivery Box? v0.4:



3090

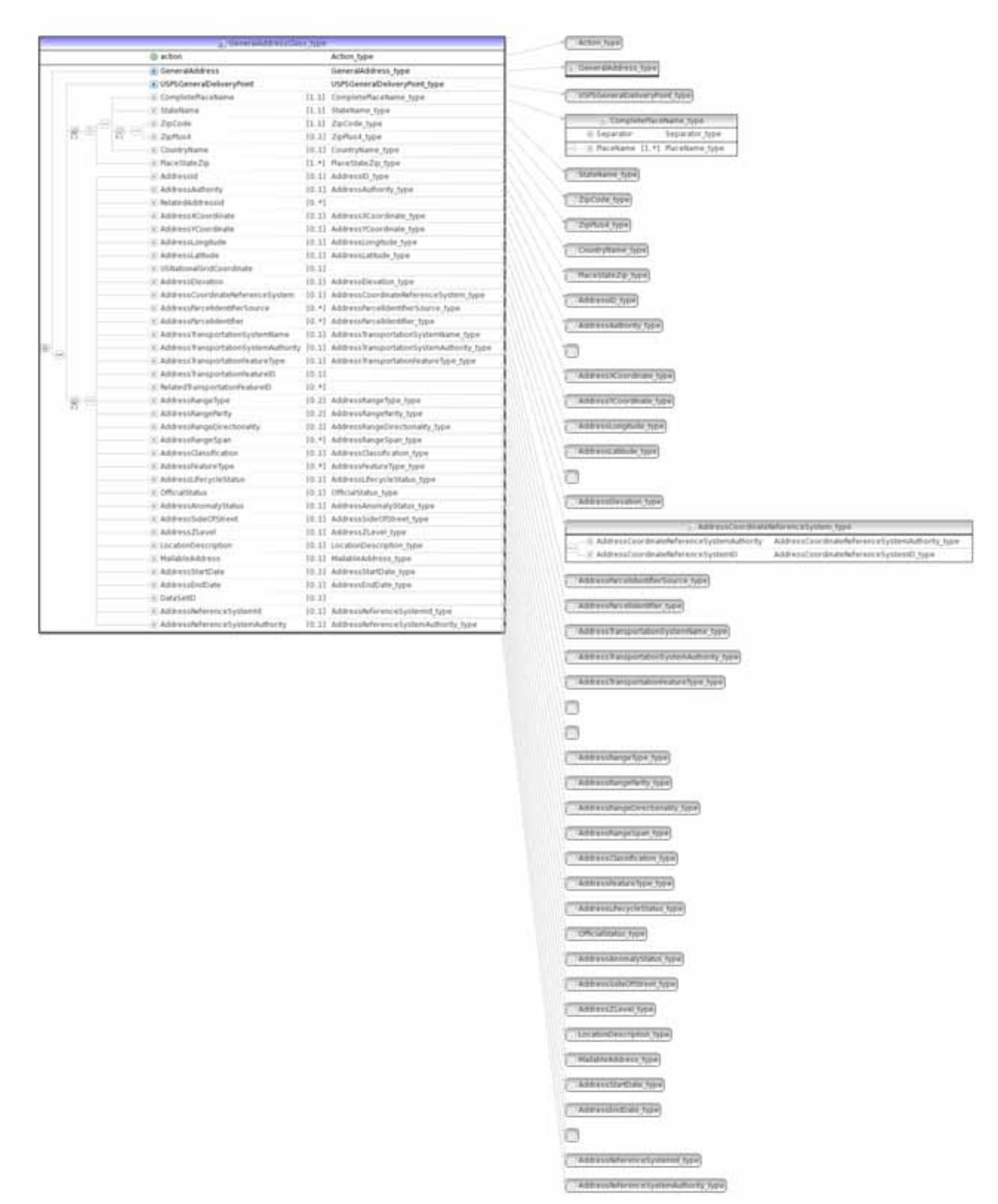
3091 • USPS General Delivery Office v0.4:



3092

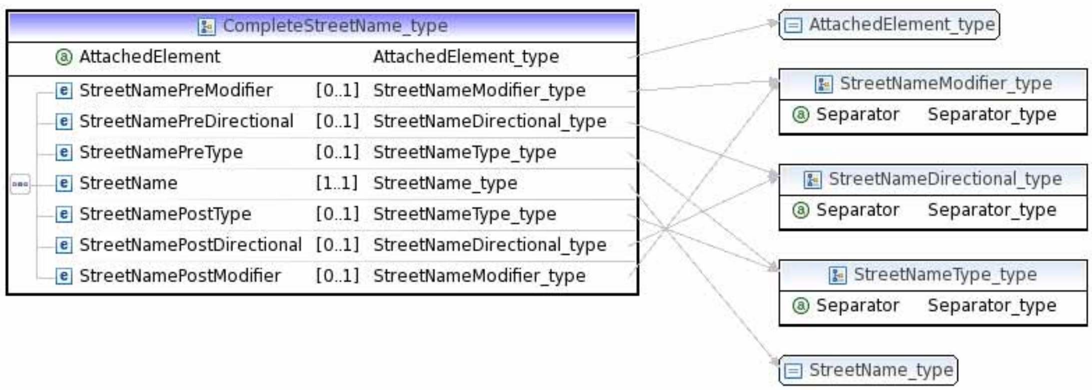
General Address Class

3094 • General Address Class v0.4:

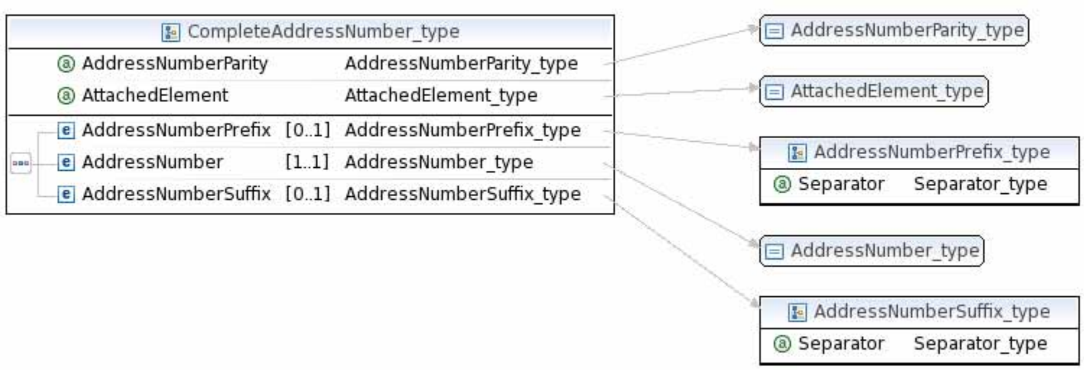


3095

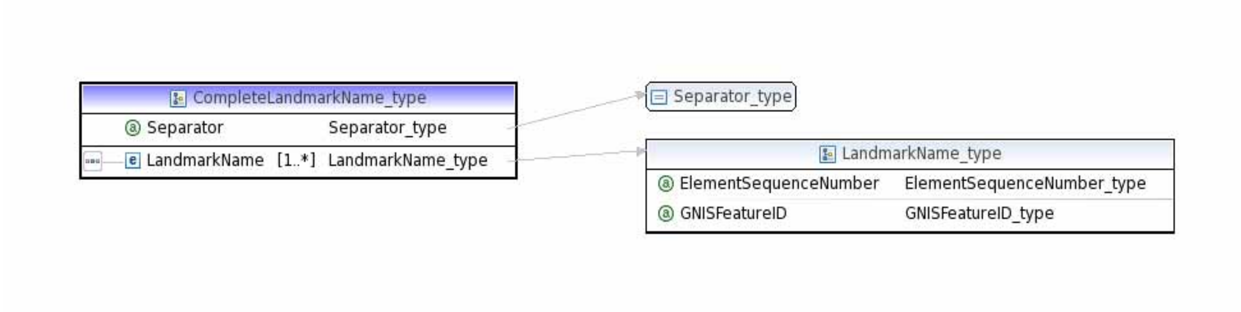
- Complete Street Name v0.4:



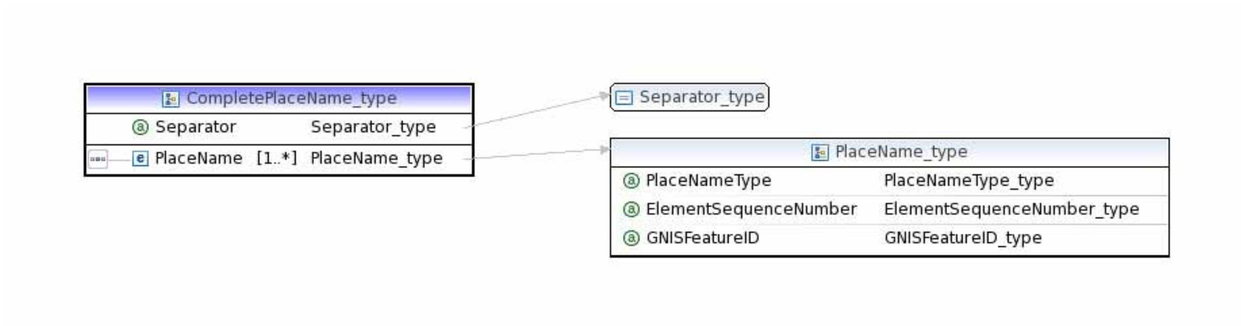
- Complete Address Number v0.4:



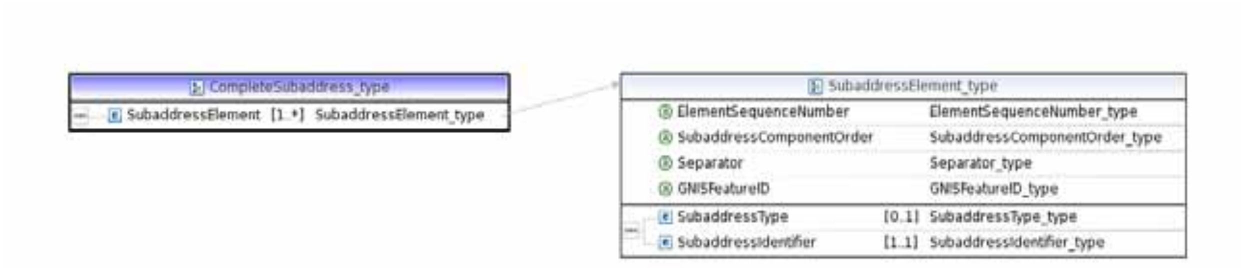
- Complete Landmark Name v0.4:



- Complete Place Name v0.4:



- Complete Subaddress v0.4:



6. REFERENCES

Note: All references and URLs were current as of September 5-8, 2009, with selected updates through January 6, 2010.

6.1 Standards and Specifications Cited

1. American National Standards Institute. "INCITS 38:200x, Codes for the Identification of the States, the District of Columbia, Puerto Rico, and the Insular Areas of the United States." Maintained by the U.S. Census Bureau. (Formerly FIPS Publication 5-2, May 28, 1987). "Last revised January 9, 2009."
   Posted at: http://www.census.gov/geo/www/ansi/statetables.html
   Cited in: Content (State Name)

2. American National Standards Institute. "Spatial Data Transfer Standard" (SDTS). ANSI NCITS 320-1998.
   Draft for review posted at: http://mcmcweb.er.usgs.gov/sdts/standard.html
   Cited in: Data Quality (Introduction and Measuring Address Quality).

3. EPSG - European Petroleum Survey Group (now defunct; superseded by Geodetic Subcommittee of the International Association of Oil and Gas Producers (OGP)).

4. International Association of Oil and Gas Producers (OGP) Geodetic Subcommittee. "EPSG Geodetic Parameter Dataset." (currently at version 7.1), augmented by accompanying guidance notes, especially EPSG Guidance Note Number 7, part 1, "Using the EPSG Geodetic Parameter Dataset." Version 4.0, May 2009.
   Posted at: http//www.epsg.org and at: http://info.ogp.org.uk/geodesy/
   Cited in: Content (Address Coordinate Reference System ID and Address Coordinate

Reference System Authority).

5. International Standards Organization. "ISO 3066-1: English country names and code elements."
   Posted at: http://www.iso.org/iso/en/prods-services/iso3166ma/02iso-3166-code-lists/list-en1.html
   Cited in: Content (Country Name).

6. International Standards Organization. "ISO 8601: Data elements and interchange formats — Information interchange — Representation of dates and times". Third edition 2004-12-01.
   Not in the public domain. Available for sale from ISO.
   Listed (with a public summary) at: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=40874
   ISO has posted a useful FAQ and extended summary at: http://www.iso.org/iso/support/faqs/faqs_widely_used_standards/widely_used_standards_other/date_and_time_format.htm
   Cited in: Content (Address Start Date and Address End Date).

7. International Standards Organization. "ISO/IEC 9075-1:2008 - Information technology -- Database languages -- SQL -- Part 1: Framework (SQL/Framework)" ISO/IEC 9075(1-4,9-11,13,14):2008. Third edition. Not in the public domain. Available for sale from ISO.
   Listed (with a public summary) at:

http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=45498

Cited in: Data Quality

8. International Standards Organization. "ISO 19106:2004 Geographic Information - Profiles."

   Not in the public domain. Available for sale from ISO.

   Listed at: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=26011

   Cited in: Postal Addressing Profile.

9. International Standards Organization. "ISO 19113:2002 Geographic information -- Quality principles."

   Not in the public domain. Available for sale from ISO.

   Listed at: http://www.iso.org/iso/catalogue_detail.htm?csnumber=26018

   Cited in: Data Quality (Introduction and Measuring Address Quality).

10. International Standards Organization. "ISO 19115:2003 Geographic information -- Metadata."

   Not in the public domain. Available for sale from ISO.

   Listed at: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=26020

   Also listed at: http://www.opengeospatial.org/standards/as .

Cited in: Data Quality (Introduction and Measuring Address Quality).

11. International Telecommunications Union, Telecommunication Standardization Sector (ITU-T). "Information technology – Open Systems Interconnection – Procedures for the operation of OSI Registration Authorities: Generation and registration of Universally Unique Identifiers (UUIDs) and their use as ASN.1 object identifier components." ITU-T X.667 (08/2008). SERIES X: DATA NETWORKS, OPEN SYSTEM COMMUNICATIONS AND SECURITY. (Also published as ISO/IEC 9834-8:2008).

    Posted at: http://www.itu.int/rec/T-REC-X.667-200808-I/en

    Cited in: Content (Address ID)

12. Internet Engineering Task Force, Network Working Group. "A Universally Unique Identifier (UUID) URN Namespace." Request for Comments 4122. P. Leach, M. Mealling, R. Salz, July 2005. (RFC 4122 is "...aligned, and ... fully technically compatible..." with ITU-T X.667:2004. (RFC 4122 Sec. 1)).

    Posted at: http://www.ietf.org/rfc/rfc4122.txt

    Cited in: Content (Address ID).

13. Internet Engineering Task Force, Network Working Group. "Revised Civic Location Format for Presence Information Data Format Location Object (PIDF-LO)." Request for Comment 5139. M. Thomson and J. Winterbottom, February 2008.

    Posted at: http://www.ietf.org/rfc/rfc5139.txt

    Cited in: FGDC-NENA Profile

14. Internet Engineering Task Force, Network Working Group. "Location Types

Registry." Request for Comment 4589. H. Schulzrinne and H. Tschofenig, July 2006. (Posted at: http://www.ietf.org/rfc/rfc4589.txt).

Cited in: FGDC-NENA Profile

15. Internet Engineering Task Force, Network Working Group. "A Presence-based GEOPRIV Location Object Format." Request for Comment 4119. J. Peterson, December 2005.

Posted at: http://www.ietf.org/rfc/rfc4119.txt

Cited in: FGDC-NENA Profile

16. National Emergency Number Association. "NENA Next Generation 9-1-1 (NG9-1-1) Civic Location Data Exchange Format (CLDXF) Standard." NENA Joint Data Technical/Next Generation Integration Committees, Next Generation Data Development Working Group (NGDD). Committee working draft, as of January 2010.

Posted at: % [Draft. In NENA approval process. Not yet posted for public review.]

Cited in:FGDC-NENA Profile

17. National Emergency Number Association. “NENA Data Standards For Local Exchange Carriers, ALI Service Providers & 9-1-1 Jurisdictions.” NENA Standard number 02-011 v6. Revised November 21, 2006.

Posted at: http://www.nena.org/standards/technical/data/911-Data-Management

Cited in: Data Quality (Introduction and Measuring Address Quality).

18. OGP - See International Association of Oil and Gas Producers.

19. Open Geospatial Consortium, Inc. "OpenGIS(R) Geography Markup Language

(GML)." Version: 3.1.1. Date: 2004-02-07. OpenGIS(R) project document: OGC 03-105r1. Implementation Specification. Category: OpenGIS(R) Recommendation Paper. Editors: Simon Cox, Paul Daisey, Ron Lake, Clemens Portele, Arliss Whiteside. Also published by the ISO as "ISO 19136: Geographic information – Geography Markup Language."

Posted at: http://www.opengeospatial.org/standards/gml

Cited in: Introduction, Content (Introduction)

20. Open Geospatial Consortium, Inc. "OpenGIS(R) Simple Features Specification For SQL." Revision 1.1. OpenGIS Project Document 99-049. Release Date: May 5, 1999.

Posted at: http://portal.opengeospatial.org/files/?artifact_id=829

Cited in: Data Quality

21. U.S. Board on Geographic Names. "Domestic Names--Principles, Policies, and Procedures." By Donald J. Orth and Roger L. Payne. On-line edition (June, 2003), 52 pp.

Posted at: http://geonames.usgs.gov/docs/pro_pol_pro.pdf

Cited in: Introduction, Content (GNIS Feature ID, Landmark Name, and Place Name), and Classification (Landmark Classes).

22. U.S. Census Bureau. "2008 TIGER/Line® Shapefiles Technical Documentation" Nov. 2008. See especially:

---3.9 County-based Shapefiles: 3.9.1 All Lines Shapefile (pp. 3-71 thru 3-76).

---3.10 County-based Relationship Files:

------3.10.1 Address Range-Feature Name Relationships (pp. 3-86 thru 3-87),

------3.10.2 Address Ranges (pp. 3-87 thru 3-94),

------3.10.3 Feature Names (pp. 3-94 thru 3-95)

---Appendix C Feature Name Directionals

---Appendix D Feature Name Qualifiers

Posted at: http://www.census.gov/geo/www/tiger/tgrshp2008/TGRSHP08.pdf

Cited in: Content (Street Name Pre Modifier, Street Name Post Modifier), and Classification (Four Number Address Range).

Census TIGER terminology is noted in Content (various elements) under "Other common names for this element."

23. U.S. Environmental Protection Agency (EPA). "Contact Information Data Standard." Standard No.: EX000019.2 (Section 3.0, Contact Information Data Standards Table, Part 2.0, Address). Revised January 6, 2006. "This standard has been produced through the Environmental Data Standards Council (EDSC)."

Posted at:

http://www.exchangenetwork.net/standards/Contact_Info_01_06_2006_Final.pdf

Cited in: Content (Delivery Address Type).

24. U.S. Federal Geographic Data Committee. "Cadastral Data Content Standard for the National Spatial Data Infrastructure." Version 1.4 – Fourth Revision. (FGDC-STD-003). Subcommittee on Cadastral Data, Federal Geographic Data Committee. May 2008.

Posted at: http://www.nationalcad.org/data/documents/CADSTAND.v.1.4.pdf

Cited in: Content (Address Parcel Identifier); Appendix J.2.2

25. U.S. Federal Geographic Data Committee. "Content Standard for Digital Geospatial Metadata" (CSDGM). Version 2.0, 1998. FGDC-STD-001-1998.
Posted at: http://www.fgdc.gov/standards/projects/FGDC-standards-projects/metadata/base-metadata/v2_0698.pdf
Cited in: Introduction (Address Data Quality; Related Standards); Content (Data Set ID; Address Start Date; Address End Date); and Data Quality (Introduction; Measuring Address Quality).
For specifics on date format, see "Organization of Standard" (pp. viii - ix); for data set identification, see "Citation Information" (Sec. 8).

26. U.S. Federal Geographic Data Committee. "FGDC Standards Reference Model." March, 1996.
Posted at: http://www.fgdc.gov/standards/process/FGDC-standards-reference-model
Cited in: Introduction.

27. U.S. Federal Geographic Data Committee. "Geographic Information Framework Data Content Standard Part 0: Base document." May 2008. FGDC-STD-014.0-2008
Posted at: http://www.fgdc.gov/standards/standards_publications
Cited in: Content (Introduction); Appendix J.2.1 and J.3.

28. U.S. Federal Geographic Data Committee. "Geographic Information Framework Data Content Standard Part 1: Cadastral." May 2008. FGDC-STD-014.1-2008
Posted at: http://www.fgdc.gov/standards/standards_publications
Cited in: Content (Address Parcel Identifier Source, Address Parcel Identifier); Appendix J.2.2

29. U.S. Federal Geographic Data Committee. "Geographic Information Framework Data Content Standard Part 2: Digital Orthoimagery." May 2008. FGDC-STD-014.2-2008 Posted at: http://www.fgdc.gov/standards/standards_publications Cited in: Appendix J.2.3.

30. U.S. Federal Geographic Data Committee. "Geographic Information Framework Data Content Standard Part 3: Elevation." May 2008. FGDC-STD-014.3-2008 Posted at: http://www.fgdc.gov/standards/standards_publications Cited in: Appendix J.2.4.

31. U.S. Federal Geographic Data Committee. "Geographic Information Framework Data Content Standard Part 4: Geodetic Control." May 2008. FGDC-STD-014.4-2008 Posted at: http://www.fgdc.gov/standards/standards_publications Cited in: Appendix J.2.5.

32. U.S. Federal Geographic Data Committee. "Geographic Information Framework Data Content Standard Part 5: Governmental Unit and Other Geographic Area Boundaries." May 2008. FGDC-STD-014.5-2008 Posted at: http://www.fgdc.gov/standards/standards_publications Cited in: Content (Place Name, State Name, Country Name) and Appendix J.2.6.

33. U.S. Federal Geographic Data Committee. "Geographic Information Framework Data Content Standard Part 6: Hydrography." May 2008. FGDC-STD-014.6-2008 Posted at: http://www.fgdc.gov/standards/standards_publications Cited in: Appendix J.2.7.

34. U.S. Federal Geographic Data Committee. "Geographic Information Framework Data

Content Standard Part 7: Transportation Base." May 2008. FGDC-STD-014.7-2008

Posted at: http://www.fgdc.gov/standards/standards_publications

Cited in: Content (Address Side Of Street), Appendix C, Appendix J.2.8.

35. U.S. Federal Geographic Data Committee. "Geographic Information Framework Data Content Standard Part 7b: Transportation - Rail." May 2008. FGDC-STD-014.7b-2008

Posted at: http://www.fgdc.gov/standards/standards_publications

Cited in: Appendix C, Appendix J.2.8.

36. U.S. Federal Geographic Data Committee. "Geographic Information Framework Data Content Standard Part 7c: Transportation - Roads." May 2008. FGDC-STD-014.7c-2008

Posted at: http://www.fgdc.gov/standards/standards_publications

Cited in: Cited in: Appendix C, Appendix J.2.8.

37. U.S. Federal Geographic Data Committee. "Geographic Information Framework Data Content Standard Part 7d: Transportation - Transit." May 2008. FGDC-STD-014.7d-2008

Posted at: http://www.fgdc.gov/standards/standards_publications

Cited in: Cited in: Appendix C, Appendix J.2.8.

38. U.S. Federal Geographic Data Committee. "Geographic Information Framework Data Content Standard Part 7e: Transportation – Inland Waterways." May 2008. FGDC-STD-014.7e-2008

Posted at: http://www.fgdc.gov/standards/standards_publications

Cited in: Appendix C, Appendix J.2.8.

39. U.S. Federal Geographic Data Committee. "Spatial Data Transfer Standard." FGDC-STD-002.1. (Approved by ANSI as ANSI NCITS 320-1998, June 9, 1998.) (See especially Part 1, Section 3, "Spatial Data Quality").

Posted at: http://mcmcweb.er.usgs.gov/sdts/standard.html

Cited in: Data Quality

40. U.S. Federal Geographic Data Committee. "United States National Grid." FGDC-STD-011-2001.

Posted at: http://www.fgdc.gov/standards/projects/FGDC-standards-projects/usng/fgdc_std_011_2001_usng.pdf

Cited in: Content (US National Grid Coordinate).

41. U.S. Geological Survey, 19810501. "U.S. Geographic Names Information System (GNIS)." U.S. Geological Survey, Reston, VA.

Accessed at: http://geonames.usgs.gov/domestic/index.html

Cited in: Content (GNIS Feature ID, Landmark Name, and Place Name).

42. U.S. Geological Survey, 19810501. "U.S. Geographic Names Information System (GNIS): Feature Class Values and Definitions." Geographic Names Project, USGS.

Accessed at: http://geonames.usgs.gov/pls/gnispublic/f?p=gnispq:8

Cited in: Content (Landmark Name).

43. U.S. Postal Service (USPS). "Postal Addressing Standards." Publication 28, July 2008.

Posted at: http://pe.usps.gov/cpim/ftp/pubs/Pub28/Pub28.pdf

Cited in the Introduction, throughout Content and Classification, in Data Quality

(Introduction), and in the Postal Addressing Profile.

USPS terminology is noted in Content (various elements) under "Other common names for this element."

USPS Publication 28 is a foundational work for the Content and Classification Parts. The Postal Addressing Profile reconciles USPS Publication 28 with the FGDC address standard.

44. U.S. Postal Service (USPS). "Postal Addressing Standards for Puerto Rico and the US Virgin Islands". March 2007.
Posted at:
http://www.ribbs.usps.gov/files/Addressing/PR_AND_USVI_ADDRESSING_-_ENGLISH.PDF
Cited in: Numbered Thoroughfare Address and Community Address.

45. U.S. Postal Service (USPS). "Quick Service Guide 800: Glossary of Postal Terms and Abbreviations in the DMM." September 8, 2009
Posted at: http://pe.usps.com/text/qsg300/Q800.htm
Cited in: Content (Zip Code, Zip Plus 4)

46. Universal Postal Union (UPU). "International postal address components and templates – Part A: Conceptual hierarchy and template languages" and "International postal address components and templates – Part B: Element mapping conventions, template design considerations, address templates and rendition instructions". UPU Data definition and encoding standards Draft S-42a-6 and Draft S-42b-6 (POC C4 SB 2008.4-Doc.4d-Annex 1 and Annex 2). See especially Part A, section 5.3.32

(definition of "thoroughfare"). Part B, Section 22, contains the United States profile of the UPU standard.

Draft S-42a-6 posted at: http://www.upu.int/document/2008/an/cep_c_4_gn-4/src/d004d_ad00_an01_p00_r00.pdf

Draft S-42b-6 posted at: http://www.upu.int/document/2008/an/cep_c_4_gn-4/src/d004d_ad00_an02_p00_r00.pdf

Cited in: Introduction, Classification (Thoroughfare Classes).

47. W3C XML Core Working Group. "Extensible Markup Language (XML) 1.0." (Third Edition), W3C Recommendation 4 February 2004. Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler, François Yergeau, eds.

Posted at: http://www.w3.org/TR/2004/REC-xml-20040204

Cited in: Introduction, Content (introduction), Classification (Introduction), Exchange.

48. Wisconsin State Cartographers Office. "Wisconsin Coordinate Reference Systems." 2nd edition, 2009.

Posted at: http://www.sco.wisc.edu/pubs/WisCoordRefSys_April2009.pdf

Cited in: Content (Address Coordinate Reference System ID, Address Coordinate Reference System Authority)

6.2 Other Works Consulted

1. Australia and New Zealand. "AS/NZS 4819:2003 : Geographic information - Rural and urban addressing." (Amendment 1: 26 October 2006.)

Available for purchase from:

http://www.saiglobal.com/shop/script/Details.asp?docn=AS216071868339

Abstract: "Specifies requirements for a comprehensive urban and rural addressing system for allocation, transfer and management of property addresses. Takes into account data dictionary for transfer of street address systems and interchange of client information and includes geocoding. Guidelines for the use of the addressing system and examples of addressing are included."

2. Britsh Standards Institution. "BS7666: 2006 - Spatial Datasets for Geographical Referencing."

Not in the public domain. Available for purchase from the British Standards Institute. The standard has four parts:

- Part 0:2006 General model for gazetteers and spatial referencing - "Scope: ...defines the essential components of a gazetteer of geographic locations and provides a general model of spatial references based upon named spatial units in the United Kingdom." Listed for sale at: http://www.bsi-global.com/Shop/Publication-Detail/?pid=000000000030127201
- Part 1:2006 Specification for a street gazetter - "Scope: ...specifies the data to be maintained in a gazetteer of streets, consistent with Part 0 of this standard. [A]lso specifies the means of representing the geometry of the street in terms of coordinates." Listed for sale at: http://www.bsi-global.com/en/Shop/Publication-Detail/?pid=000000000030127194
- Part 2:2006 Specification for a land and property gazetteer - "Scope: ...specifies the logical data structure for a gazetteer of land and property." Listed for sale at: http://www.bsi-global.com/en/Shop/Publication-

Detail/?pid=000000000030127196

- Part 5:2006 Specification for a delivery point gazetteer - "Scope: ...specifies the logical data structure and data content for a gazetteer of delivery points, for purposes of identification, access and validation of service requests... It provides a method of referencing delivery points by means of unique references and descriptive delivery addresses." Listed for sale at: http://www.bsi-global.com/en/Shop/Publication-Detail/?pid=000000000030127196

3. Canada Post Corporation. "Canada Postal Guide: Addressing Guidelines." ("Last updated: 2009-06-22").
   Posted at: http://www.canadapost.ca/tools/pg/manual/PGaddress-e.asp
   Purpose: Provides recommended formats for Canadian mailing addresses. "The guidelines in this chapter promote the most technologically efficient formats for addressing. It does not limit mailers to any one format. In some cases, because of individual preference or other considerations, mailers may not be able to follow these formats.

4. Organization for the Advancement of Structured Information Standards (OASIS) Customer Information Quality (CIQ) Technical Committee. "OASIS CIQ V3.0 Final Approved Committee Specifications (xNL, xAL, xNAL, xPIL)." "OASIS CIQ specification that provides a standard format to represent an address/location is "extensible Address Language (xAL)"."
   Posted 1 December 2007 at: http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=ciq

Purpose: "The objective of the OASIS CIQ TC (formed in 2000)is to deliver a set of XML Specifications for defining, representing, interoperating and managing "PARTY (Person or Organisation) INFORMATION" that are truly open, vendor neutral, industry and application independent, and importantly "Global"...The CIQ family of specifications are designed to represent party data (e.g. name and address) independent of any culture, geographical location, application or industry..."

5. South African Bureau of Standards. "Geographic information - Addresses Part 1: Data format of addresses." SANS 1883-1:2009. Approved 2009-10-02.

Available for purchase from the South African Bureau of Standards, https://www.sabs.co.za/index.php?page=standardspurchase (Search on Standard = 1883)

Abstract: "Specifies and defines the data elements, as well as the address types that can be constructed from the data elements for South African addresses. Defines terms and definitions related to addresses in South Africa. Applies to addresses covering the whole of South Africa. Describes the physical location of a point of service delivery, and addresses that could be geo-referenced. Includes definitions for address types that are assigned by the official address issuing body (such as the street address type), as well as address types that are commonly in use (such as the farm and informal address types)."

7. APPENDICES

7.1 Appendix A: Normative XSD

The Address Standard XML Schema Definition is broken into 2 parts. The first part contains element definitions and corresponds to Part One of the Standard. The second part contains the Address Class definitions and corresponds to Part Two of the Stanard.

1. addr_type.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema targetNamespace="addr_type"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:addr_type="addr_type"
 xmlns:gml="http://www.opengis.net/gml">
 <!--
 Draft Address Standard, version 0.4 being prepared and tested by a
 Working Group coordinated by URISA and NENA and the Census Bureau for
 submittal to the FGDC.
 -->
 <!--
 During the initial draft period the rddl can be found at
 http://wfs.co.fulton.ga.us/urisa/addr_std/addr.xsd
 -->
 <xsd:include schemaLocation=""></xsd:include>
 <xsd:include schemaLocation=""></xsd:include>
 <xsd:include schemaLocation=""></xsd:include>
 <xsd:import namespace="http://www.opengis.net/gml"
 schemaLocation="base/gml.xsd">
 <xsd:annotation>
  <xsd:documentation>
  GML 3.1.1 from OpenGeospatial Consortium
  </xsd:documentation>
 </xsd:annotation>
 </xsd:import>
 <xsd:simpleType name="version_type">
 <xsd:annotation>
  <xsd:documentation xml:lang="en">
  The ID for this version
  of the Address Standard.
  </xsd:documentation>
 </xsd:annotation>
 <xsd:restriction base="xsd:token">
```

```
 <xsd:enumeration value='0.4' />
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="Separator_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">
 A symbol or word
 required as a separator between
 components of a complex element or
 class. The separator
 is required for address number ranges and for
 intersection addresses, and it is occasionally needed in
 constructing complete street numbers.
 1. The default separator, an
 empty space, is implicit and is not shown
 in the syntaxes of complex
 elements and classes.
 2. An explicit separator is required for
 address ranges and
 intersection addresses.
 3. Within a given dataset,
 one
 value should be used consistently within
 a given complex
 element.
 4. For complete street numbers, the separator
 is rarely needed and
 its
 use should be minimized. As an
 alternative, the separator symbol
 usually can be
 included with the address number prefix or suffix.
 5.
 The separator is not needed in creating fractions (1/2,
 etc.) for
 address number suffixes.
 6. Some address parsing software permits the
 use of ampersands to
 signify intersection addresses. Be wary, though
 --in many
 programming languages, ampersands are reserved for other
 uses, which
 could complicate data exchange.
 </xsd:documentation>
</xsd:annotation>
```

```
<xsd:restriction base="xsd:string">
 <xsd:pattern value='.*' />
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="ElementSequenceNumber_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">
 The order in which the
 elements of a Complete Subaddress,
 Complete Landmark Name, or
 Complete Place Name should be written.
 </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:integer" />
</xsd:simpleType>
<xsd:simpleType name="GNISFeatureID_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">
 "A permanent, unique
 number assigned to a geographic feature for the
 sole purpose of
 uniquely identifying that feature as a record in any
 information
 system database, dataset, file, or document and for
 distinguishing it
 from all other feature records so identified. The
 number is assigned
 sequentially (highest existing number plus one)
 to
 new records as they
 are created in the Geographic Names Information
 System."
 Definition
 Source Geographic Names Project, USGS, 523
 National Center,
 Reston, VA
 20192-0523, as posted August 25, 2009 at:
 http://geonames.usgs.gov/domestic/metadata.htm "Feature Identifier"
 </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:integer" />
</xsd:simpleType>
<xsd:complexType name="AddressNumberPrefix_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en"> The portion of the
```

```
 Complete Address Number which precedes the Address Number itself.
 </xsd:documentation>
</xsd:annotation>
<xsd:simpleContent>
 <xsd:extension base="xsd:string">
 <xsd:attribute name="Separator" type="addr_type:Separator_type" />
 </xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
<xsd:simpleType name="AddressNumber_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">
 The numeric identifier
 for a land parcel, house,
 building or other feature, as defined by
 the official
 address authority for the given jurisdiction.
 1. The
 address number is defined as an integer to
 support address sorting,
 parity (even/odd) definition,
 and in/out of address range tests.
 2.
 Special care should be taken with records where the
 address number is
 0 (zero). Occasionally zero is issued
 as a valid address number (e.g.
 Zero Prince Street,
 Alexandria, VA 22314) or it can be imputed (1/2
 Fifth
 Avenue, New York, NY 10003 (for which the address number
 would
 be 0 and the suffix would be "1/2")). More often,
 though, the address
 number is either missing or
 non-existent, and null value has been
 converted to zero.
 3. Some addresses may contain letters, fractions,
 hyphens, decimals and other non-integer content within
 the complete
 address number. Those non-integer elements
 should be placed in the
 Address Number Prefix if they
 appear before the site number, or in
 the Address Number
```

```
 Suffix if they follow the number.
 4. The address
 number must be converted to text when it
 is combined with the prefix
 and suffix into a complete
 address number.
 5. If necessary, the
 Separator can be used to separate
 the address number from the prefix
 or suffix elements in
 constructing the complete address number.
 </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string">
 <xsd:pattern value='[0-9]+' />
</xsd:restriction>
</xsd:simpleType>
<xsd:complexType name="AddressNumberSuffix_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">
 The non-integer portion
 of the identifier for the house,
 building or other feature which
 follows the address
 number itself, as defined by the official address
 authority for the given jurisdiction.
 1. This element will not be
 applicable for most
 addresses. When township-range or other
 non-numeric
 geographic index references follow an address number,
 they are placed in this field. The address number can
 then be
 maintained as an integer for sorting and quality
 control tests.
 2. If
 necessary, the Separator can be used to separate
 the address number
 from the prefix or suffix elements in
 constructing the complete
 address number.
 </xsd:documentation>
</xsd:annotation>
<xsd:simpleContent>
 <xsd:extension base="xsd:string">
```

```
 <xsd:attribute name="Separator" type="addr_type:Separator_type" />
 </xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
<!-- StreetName Content -->
<xsd:complexType name="StreetNameModifier_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">
 A word or phrase that
 precedes or follows all other
 elements of the street name and
 modifies it, but is
 separated from the street name by a street name
 pre-directional and/or pre-type.
 The street name pre-modifier is
 rarely needed and its
 use should be minimized. The modifier can be
 included in
 the Street Name element unless it is separated from the
 street name by a pre-type or a pre-directional.
 </xsd:documentation>
</xsd:annotation>
<xsd:simpleContent>
 <xsd:extension base="xsd:string">
 <xsd:attribute name="Separator" type="addr_type:Separator_type"></xsd:attribute>
 </xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
<xsd:complexType name="StreetNameDirectional_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">
 A word preceding or
 following the street name that
 indicates the directional taken by the
 thoroughfare from
 an arbitrary starting point, or the sector where it
 is
 located.
 1. Standard USPS English abbreviations are E, W, S, N,
 NE,
 SE, SW, NW
 2. Standard USPS Spanish abbreviations are E, O, S, N,
 NE,
 SE, SO, NO 3. USPS Publication 28 recommends
 abbreviating
```

```
 post-directionals. The Address Standards
 Working Group differs, and
 recommends storing
 post-directionals fully spelled out, exactly as
 given by
 the local naming authority, to avoid confusion. For
 example:
 "N W Jones St": Is it Northwest Jones Street?
 Ned Walter Jones
 Street? North Walter Jones Street? The
 abbreviations create
 ambiguity. If stored unabbreviated,
 directionals can be exported as
 standard abbreviations
 as needed for mailing and other purposes.
 </xsd:documentation>
</xsd:annotation>
<xsd:simpleContent>
 <xsd:extension base="xsd:string">
 <xsd:attribute name="Separator" type="addr_type:Separator_type"></xsd:attribute>
 </xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
<xsd:complexType name="StreetNameType_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">
 The element of the
 complete street name preceding or
 following the street name element
 that indicates the
 type of street.
 1. A street may have either a
 pre-type or a post-type,
 or neither, but not both. 2. USPS
 Publication 28
 provides the best list of pre-types and post-types,
 but
 it is not exhaustive.
 2. USPS Publication 28 provides a standard
 list of
 street type abbreviations, and recommends their use. The
 Address Standards Working Group differs, and recommends
 storing
 pre-types and post-types fully spelled out,
 exactly as given by the
```

```
 local naming authority, to avoid
 confusion. If stored unabbreviated,
 directionals can be
 exported as standard abbreviations as needed for
 mailing
 and other purposes.
 3. Pre-types are much less common than
 post-types in
 English. They are typically used for numbered highway
 routes and roads. They are much more common in Spanish
 and French
 language street names.
 4. Occasionally a street name may have neither
 a prefix
 type or a suffix type (e.g., "Broadway.")
 5. Names like
 "County Road 28", where the street type
 occurs in the middle of the
 name, should be treated as a
 street name with no pre-type or
 post-type.
 </xsd:documentation>
</xsd:annotation>
<xsd:simpleContent>
 <xsd:extension base="xsd:string">
 <xsd:attribute name="Separator" type="addr_type:Separator_type"></xsd:attribute>
 </xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
<xsd:simpleType name="StreetName_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">
 Official name of a
 street as assigned by a local
 governing authority, or an alternate
 (alias) name that
 is used and recognized, excluding street types,
 directionals, and modifiers.
 1. Each jurisdiction should establish
 its own list of
 street names and use it as a domain of values to
 validate addresses. Alternate and Official names are
 distinguished by
 the address attribute "Alias Status
 Attribute"
```

```
 2. Local addressing
 authorities are urged to follow
 consistent internal street naming
 practices, and to
 resolve internal street name inconsistencies,
 especially
 for numbered streets ("Twentieth" or "20th" ?), internal
 capitalization ("McIntyre" or "Mcintyre" ?), hyphens,
 and
 apostrophes.
 3. If alternate or abbreviated versions of street names
 are needed for a specialized purpose such as mailing or
 emergency
 dispatch, they can be created in views or
 export routines.
 </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string">
 <xsd:pattern value='.*' />
</xsd:restriction>
</xsd:simpleType>
<!-- Occupancy Types -->
<xsd:simpleType name="SubaddressComponentOrder_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">
 The order in which
 SubaddressType and
 SubaddressIdentifier appear within an
 SubaddressElement when expressed as text. "Apartment 7"
 </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:integer">
 <xsd:enumeration value="1">
 <xsd:annotation>
  <xsd:documentation>SubaddressType first, then SubaddressIdentifier
  (or: SubaddressElement does not include an SubaddressType).
  Example: "Floor 7"</xsd:documentation>
 </xsd:annotation>
 </xsd:enumeration>
 <xsd:enumeration value="2">
 <xsd:annotation>
  <xsd:documentation>SubaddressIdentifier first, then SubaddressType.
  Example: "Empire Room"</xsd:documentation>
 </xsd:annotation>
 </xsd:enumeration>
```

```
  <xsd:enumeration value="3">
  <xsd:annotation>
   <xsd:documentation>Order is not known or unstated.
   </xsd:documentation>
  </xsd:annotation>
  </xsd:enumeration>
 </xsd:restriction>
 </xsd:simpleType>
 <xsd:simpleType name="SubaddressType_type">
 <xsd:annotation>
  <xsd:documentation xml:lang="en">
  This is a modifier to
  the SubaddressIdentifier element and this
  cannot exist without it.
  The type of structure (when
  several structures are found at the same
  address), e.g.,
  Apartment, Tower, Block. Used with Building
  Identifier
  to designate one of several structures at a given site.
  Fits within the general USPS definition of a "secondary
  address
  designator" and EPA definitions of "secondary
  address identifier"
  </xsd:documentation>
 </xsd:annotation>
 <xsd:restriction base="xsd:string" />
 </xsd:simpleType>
 <xsd:simpleType name="SubaddressIdentifier_type">
 <xsd:annotation>
  <xsd:documentation xml:lang="en">
  The letters, numbers,
  words or combination thereof used
  to distinguish one structure from
  another when several
  occur at the same address.
  Used with
  SubaddressType to designate one of several
  structures at a given
  site. Fits within the USPS
  defnition of a "secondary address
  designator" and the
  general EPA definition of "secondary address
  identifier"
  </xsd:documentation>
```

```
</xsd:annotation>
<xsd:restriction base="xsd:string" />
</xsd:simpleType>
<xsd:complexType name="SubaddressElement_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">
 A single combination of
 SubaddressType and
 SubaddressIdentifier (or, in some cases, a
 SubaddressIdentifier alone), which, alone or in
 combination with
 other SubaddressElements, distinguishes
 one subaddress within or
 between structures from another
 when several occur within the same
 feature. See
 CompleteSubaddress for a definition of "subaddress."
 </xsd:documentation>
</xsd:annotation>
<xsd:sequence>
 <xsd:element name="SubaddressType" type="addr_type:SubaddressType_type"
 maxOccurs="1" minOccurs="0" />
 <xsd:element name="SubaddressIdentifier" type="addr_type:SubaddressIdentifier_type"
 maxOccurs="1" minOccurs="1" />
</xsd:sequence>
<xsd:attribute name="ElementSequenceNumber"
 type="addr_type:ElementSequenceNumber_type" />
<xsd:attribute name="SubaddressComponentOrder"
 type="addr_type:SubaddressComponentOrder_type" />
<xsd:attribute name="Separator" type="addr_type:Separator_type" />
<xsd:attribute name="GNISFeatureID"
type="addr_type:GNISFeatureID_type"></xsd:attribute>
</xsd:complexType>
<!-- Landmark Name Type -->
<xsd:complexType name="LandmarkName_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">
 The name by which a
 prominent feature is publicly known.
 Landmarks usually have a street
 address. A landmark name
 does not imply official historic landmark
 status, but
 simply a commonly used name that substitutes for an
 address number and street name in identifying the
```

```
 location of a
 specific building or feature. Generally
 the use of a landmark's
 street address is preferable
 because it is unambiguous. All landmark
 names should be
 cross-referenced to a street address or other
 coordinate
 location.
 </xsd:documentation>
</xsd:annotation>
<xsd:simpleContent>
 <xsd:extension base="xsd:string">
 <xsd:attribute name="ElementSequenceNumber"
  type="addr_type:ElementSequenceNumber_type" />
 <xsd:attribute name="GNISFeatureID" type="addr_type:GNISFeatureID_type" />
 </xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
<!-- Area and Zone Elements -->
<xsd:simpleType name="CommunityPlaceName_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">
 A named area, sector,
 or development that is not an
 incorporated municipality or other
 governmental unit,
 such as a neighborhood in a city, or a rural
 settlement
 in unincorporated area. Often called "urbanization" in
 Puerto Rican addressing usage.
 1. "Urbanizacion", commonly used in
 urban areas of
 Puerto Rico, is an important part of the address.
 Street
 names and address ranges are often repeated in a city,
 especially where a city has annexed older towns; they
 are
 distinguished by their urbanizacion or community
 name.
 2. Certain
 other words can be used in place of
 "urbanizacion": extenciones,
 mansiones, reparto, villa,
 parque, jardine, altura, alturas, colinas,
```

```
 estancias,
 extension, quintas, sector, terraza, villa, villas.
 3. For
 more information on Puerto Rican addressing
 conventions, see USPS
 Publication 28 Section 29, and
 USPS "Addressing Standards for Puerto
 Rico and the
 Virgin Islands".
 </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string">
 <xsd:pattern value='.+' />
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="PlaceNameType_type">
<xsd:restriction base="xsd:string">
 <xsd:enumeration value="USPSCommunity">
 <xsd:annotation>
  <xsd:documentation xml:lang="en">
  The name given by the
  U.S. Postal Service to the
  post office from which mail is delivered
  to the
  address. In many places this will be different
  from the name
  of the municipality in which the
  address is physically located.
  The
  name given by the U.S. Postal Service to the
  post office from which
  mail is delivered to the
  address. In many places this will be
  different
  from the name of the incorporated municipality
  in which
  the address is physically located.
  </xsd:documentation>
 </xsd:annotation>
 </xsd:enumeration>
 <xsd:enumeration value="MunicipalJurisdiction">
 <xsd:annotation>
  <xsd:documentation xml:lang="en">
  The name of the
  municipality (city, township, or
```

```
   othe non-county local government)
   in which the
   address is physically located. In many places
   this will
   be different than the city name used
   by the U.S. Postal Service.
   Required by most local governments for tax and
   services
   determinations. This will be null for
   addresses in unincorporated
   portions of
   counties.
   </xsd:documentation>
  </xsd:annotation>
  </xsd:enumeration>
  <xsd:enumeration value="County">
  <xsd:annotation>
   <xsd:documentation xml:lang="en">
   The primary
   administrative subdivision of a
   state in the United States.
   </xsd:documentation>
  </xsd:annotation>
  </xsd:enumeration>
  <xsd:pattern value=".+"></xsd:pattern>
 </xsd:restriction>
 </xsd:simpleType>
 <xsd:simpleType name="StateName_type">
 <xsd:annotation>
  <xsd:documentation xml:lang="en">
  The primary legal
  subdivision of the United States,
  represented by its two letter USPS
  abbreviation.
  This is the only element stored in abbreviated form.
  </xsd:documentation>
 </xsd:annotation>
 <xsd:restriction base="xsd:token">
  <!-- "US State and The District of Columbia" Abbreviations -->
  <xsd:pattern value='.*' />
 </xsd:restriction>
 </xsd:simpleType>
 <xsd:simpleType name="ZipCode_type">
 <xsd:annotation>
  <xsd:documentation xml:lang="en">
```

```
 A five-digit code that
 identifies a specific pseudo
 geographic delivery area. ZIP Codes can
 represent an
 area within a state, an area that crosses state
 boundaries (unusual condition) or a single building or
 company that
 has a very high mail volume. "ZIP" is an
 acronym for Zone Improvement
 Plan. Zero pad from the
 left to fill the range as in 01776.
 </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string">
 <xsd:pattern value='[0-9]{5}' />
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="ZipPlus4_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">
 A four-digit extension
 of the five-digit Zip Code that
 identifies a portion of a carrier
 route for USPS mail
 delivery. Zero pad from the left to fill the
 range as in
 0030.
 </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string">
 <xsd:pattern value='[0-9]{4}' />
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="CountryName_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">
 The name of the country
 in which the address is located.
 1.Although the scope of this
 standard is restricted to
 US addresses, this item is included for two
 reasons: to
 facilitate reconciliation with address standards of
 other
 nations, and to accommodate files which mix
```

```
 addresses from the US and
 other countries. 2. ISO 3166-1
 official short English names are
 specified because they
 a familiar and concise, and because ISO 3166-1
 is
 specified in the UPU address standard. 3. The names can
 be found
 at:
 http://www.iso.org/iso/en/prods-services/iso3166ma/02iso-3166-code-lists/list-
en1.html
 </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string" />
</xsd:simpleType>
<xsd:simpleType name="USPSBoxType_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">
 A box used for receipt
 of USPS mail. The box may be
 located in the post office lobby (e.g PO
 Box), on the
 customer's premises or other USPS authorized place (e.g.
 rural route box).
 </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string">
 <xsd:pattern value='.*' />
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="USPSBoxId_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">
 The numbers or letters
 distinguishing one box from
 another within a post office. May include
 slash, hyphen
 or period.
 </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string">
 <xsd:pattern value='.*' />
</xsd:restriction>
</xsd:simpleType>
<xsd:complexType name="USPSBox_type">
```

```
<xsd:annotation>
 <xsd:documentation>A container for the receipt of USPS mail uniquely
 identified by the combination of a USPS Box Type and a USPS Box ID.
 </xsd:documentation>
</xsd:annotation>
<xsd:sequence>
 <xsd:element name="USPSBoxType" type="addr_type:USPSBoxType_type"
 maxOccurs="1" minOccurs="1" />
 <xsd:element name="USPSBoxId" type="addr_type:USPSBoxId_type"
 maxOccurs="1" minOccurs="1" />
</xsd:sequence>
</xsd:complexType>
<xsd:simpleType name="USPSBoxGroupType_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">
 A collection of postal
 boxes served from a single
 distribution point.
 This group includes
 rural routes, highway contract
 routes, postal service centers,
 overseas military common
 mail rooms and military unit numbers.
 </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string">
 <xsd:pattern value='.*' />
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="USPSBoxGroupId_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">
 The numbers or letters
 distinguishing one group of boxes
 from another within a distribution
 point. May include
 hyphen, slash or period.
 </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string">
 <xsd:pattern value='.*' />
</xsd:restriction>
</xsd:simpleType>
<xsd:complexType name="USPSRoute_type">
<xsd:sequence>
```

```
 <xsd:element name="USPSBoxGroupType"
type="addr_type:USPSBoxGroupType_type"
 maxOccurs="1" minOccurs="1" />
 <xsd:element name="USPSBoxGroupId" type="addr_type:USPSBoxGroupId_type"
 maxOccurs="1" minOccurs="1" />
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="USPSAddress_type">
<xsd:sequence>
 <xsd:element name="USPSRoute" type="addr_type:USPSRoute_type"
 maxOccurs="1" minOccurs="1" />
 <xsd:element name="USPSBox" type="addr_type:USPSBox_type"
 maxOccurs="1" minOccurs="1" />
</xsd:sequence>
</xsd:complexType>
<xsd:simpleType name="USPSGeneralDeliveryPoint_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">
 A central point where
 mail may be picked up by the
 addressee. Two values are permitted:
 "General Delivery"
 (for post offices), and ship's names (for overseas
 military addresses).
 </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string">
 <xsd:pattern value='.*' />
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="AddressAuthorityIdentifiertype_old">
<xsd:annotation>
 <xsd:documentation xml:lang="en">
 A Concatenation of
 codes found in FIPS 5-2, 6-4, and
 55-3 data guides, with a locally
 defined code that MUST
 be defined in the metadata. The general format
 is
 (expressed as regular expressions)
 [0-9]{2}[0-9]{3}[0-9]{5}[0-9]{4}.
 </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string">
 <xsd:pattern value='.*' />
```

```
</xsd:restriction>
</xsd:simpleType>
<!-- Locational type -->
<xsd:simpleType name="AddressXCoordinate_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">The X coordinate of the
 address location.</xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:double" />
</xsd:simpleType>
<xsd:simpleType name="AddressYCoordinate_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">The Y coordinate of the
 address location.</xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:double" />
</xsd:simpleType>
<xsd:simpleType name="AddressLongitude_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">
 The longitude
 coordinate of the address location, noted
 in decimal degrees. For
 point and polygon features,
 coordinate pairs typically locate the
 point of
 assignment: a centroid point, a point locating the entry
 to a
 property, etc.
 </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:double" />
</xsd:simpleType>
<xsd:simpleType name="AddressLatitude_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">
 The latitude coordinate
 of the address location, noted
 in decimal degrees. For point and
 polygon features,
 coordinate pairs typically locate the point of
 assignment: a centroid point, a point locating the entry
 to a
 property, etc.
 </xsd:documentation>
```

```
</xsd:annotation>
<xsd:restriction base="xsd:double" />
</xsd:simpleType>
<xsd:simpleType name="USNationalGridCoordinate_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">
 The USNG or US National
 Grid is an alphanumeric
 reference system that overlays the Universal
 Transverse
 Mercator (UTM) numerical coordinate system. A USNG
 coordinate consists of three parts, the:
 1. Grid Zone Designation
 (GZD) for worldwide unique
 geoaddresses (two digits plus one letter,
 developed from
 the UTM system).
 2. 100,000-meter Square Identification
 for regional
 areas (two letters).
 3. Grid Coordinates for local areas
 (always an even
 number of digits between 2 and 10 depending upon
 precision necesary to uniquely identify the location).
 Look to
 www.fgdc.gov/standards/status/usng.html for a
 normative definition.
 Adapted from US National Grid, FDGC-STD-011-2001, Section 3.3
 </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string">
 <xsd:pattern value='.*' />
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="AddressElevation_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">Distance of the address
 in specified units above or below a vertical datum, as defined by a
 specified coordinate reference system. </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:double" />
</xsd:simpleType>
<xsd:simpleType name="AddressZLevel_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">
```

```
 Floor or level of the
 structure. The lowest level of a building is 1, and ascending
 numbers are assigned in order to each higher level.
 </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string">
 <xsd:pattern value='.*' />
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="AddressCoordinateReferenceSystemID_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">A name or number which,
 along with the Address Coordinate Reference System Authority,
 identifies the coordinate reference system to which Address X
 Coordinate and Address Y Coordinate. Address Latitude and Address
 Longitude, US National Grid Coordinate, or Address Elevation values
 are referenced. </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:integer" />
</xsd:simpleType>
<xsd:simpleType name="AddressCoordinateReferenceSystemAuthority_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">The Authority that
 assigns the unique Address Coordinate Reference System ID (number or
 name) to the Address Coordinate Reference System to which the
 Address X Coordinate and Address Y Coordinate, Address Latitude and
 Address Longitude, US National Grid Coordinate, or Address Elevation
 are referenced. </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string" />
</xsd:simpleType>
<xsd:complexType name="AddressCoordinateReferenceSystem_type">
<xsd:sequence>
 <xsd:element name="AddressCoordinateReferenceSystemAuthority"
 type="addr_type:AddressCoordinateReferenceSystemAuthority_type" />
 <xsd:element name="AddressCoordinateReferenceSystemID"
 type="addr_type:AddressCoordinateReferenceSystemID_type"></xsd:element>
</xsd:sequence>
</xsd:complexType>
<!-- Non Locational Elements -->
<xsd:simpleType name="AddressID_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">The unique
 identification number assigned to an address by the addressing
 authority. The ID number must be unique for each address assigned by
```

```
 an addressing authority.
 </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string">
 <xsd:pattern value='.*' />
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="DataSetID_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">An identifier in each
 record of a transmitted dataset, assigned by the sender or the
 receiver of the dataset, to link each record of the dataset to the
 file-level metadata that accompanies the dataset.
 </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string">
 <xsd:pattern value='.*' />
</xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="AddressReferenceSystemId_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">A unique identifier of
 the Address Reference System for a specified area (Address Reference
 System Extent). </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:integer" />
</xsd:simpleType>
<xsd:simpleType name="AddressReferenceSystemName_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">The name of the address
 system used in a specified area (Address Reference System Extent).
 </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string" />
</xsd:simpleType>
<xsd:simpleType name="AddressReferenceSystemAuthority_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">The Authority that
 assigns the unique Address Coordinate Reference System ID (number or
 name) to
 the Address Coordinate Reference System to which the Address
 X
 Coordinate and Address Y
 Coordinate, Address Latitude and Address
```

```
 Longitude, US National Grid
 Coordinate, or Address Elevation are
 referenced.
 </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string" />
</xsd:simpleType>
<xsd:complexType name="AddressReferenceSystemExtent_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">Boundary of the area(s)
 within which an Address Reference System is used.
 </xsd:documentation>
</xsd:annotation>
<xsd:complexContent>
 <xsd:restriction base="gml:MultiSurfaceType" />
</xsd:complexContent>
</xsd:complexType>
<xsd:simpleType name="AddressReferenceSystemType_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">
 The category of address
 reference system in use. The
 type of reference system determines and
 guides the
 assignment of numbers within the Address Reference
 System
 Extent.
 </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string">
 <xsd:enumeration value="Axial"></xsd:enumeration>
 <xsd:enumeration value="Grid"></xsd:enumeration>
 <xsd:enumeration value="Radial"></xsd:enumeration>
 <xsd:enumeration value="Linear Non-Axial"></xsd:enumeration>
 <xsd:enumeration value="Distance"></xsd:enumeration>
 <xsd:enumeration value="Area Based"></xsd:enumeration>
</xsd:restriction>
</xsd:simpleType>
<xsd:complexType name="AddressReferenceSystemRules_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">
 The rules by which
 address numbers, street names and
 other components of a thoroughfare
 address are
```

```
  determined.
  </xsd:documentation>
 </xsd:annotation>
 <xsd:sequence>
  <xsd:element name="AddressReferenceSystemBlockRules"
  type="addr_type:AddressReferenceSystemBlockRules_type" minOccurs="0"
  maxOccurs="unbounded"></xsd:element>
  <xsd:element name="AddressReferenceSystemNumberingRules"
  type="addr_type:AddressReferenceSystemNumberingRules_type"
  minOccurs="0" maxOccurs="unbounded"></xsd:element>
  <xsd:element name="AddressReferenceSystemStreetNamingRules"
  type="addr_type:AddressReferenceSystemStreetNamingRules_type"
  minOccurs="0" maxOccurs="unbounded"></xsd:element>
  <xsd:element
  name="AddressReferenceSystemStreetTypeDirectionalAndModfierRules"

type="addr_type:AddressReferenceSystemStreetTypeDirectionalAndModifierRules_type
"
  minOccurs="0" maxOccurs="unbounded"></xsd:element>
  <xsd:element
  name="AddressReferenceSystemPlaceNameStateCountyAndZipCodeRules"

type="addr_type:AddressReferenceSystemPlaceNameStateCountryAndZipCodeRules_t
ype"
  minOccurs="0" maxOccurs="unbounded"></xsd:element>
  <xsd:element name="AddressReferenceSystemSubaddressRules"
  type="addr_type:AddressReferenceSystemSubaddressRules_type"
  minOccurs="0" maxOccurs="unbounded"></xsd:element>
 </xsd:sequence>
 </xsd:complexType>
 <xsd:simpleType name="AddressReferenceSystemBlockRules_type">
 <xsd:annotation>
  <xsd:documentation xml:lang="en">This element defines a
  block in an Address Reference System, and sets forth the rules for
  block ranges and block breaks. </xsd:documentation>
 </xsd:annotation>
 <xsd:restriction base="xsd:string" />
 </xsd:simpleType>
 <xsd:simpleType name="AddressReferenceSystemNumberingRules_type">
 <xsd:annotation>
  <xsd:documentation xml:lang="en">The rules for numbering
  along a thoroughfare, including parity (odd/even side definition),
  and numbering increment distance and value.</xsd:documentation>
 </xsd:annotation>
 <xsd:restriction base="xsd:string" />
```

```
</xsd:simpleType>
<xsd:simpleType name="AddressReferenceSystemStreetNamingRules_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">The rules for the
 selection and use of street names within an Address Reference System
 </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string" />
</xsd:simpleType>
<xsd:simpleType
name="AddressReferenceSystemStreetTypeDirectionalAndModifierRules_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">Rules pertaining to the
 use of street types (suffix and prefix), directionals (prefix and
 suffix), and modifiers (prefix and suffix) of street names.
 </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string" />
</xsd:simpleType>
<xsd:simpleType
name="AddressReferenceSystemPlaceNameStateCountryAndZipCodeRules_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">This element contains
 rules for the use of place names, state names, country names, and
 ZIP Codes within the jurisdiction of an Address Authority.
 </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string" />
</xsd:simpleType>
<xsd:simpleType name="AddressReferenceSystemSubaddressRules_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">The rules that are
 applied to the addressing of areas within structures as subaddresses
 (units, suites, apartments, spaces, etc.) within a given Address
 Reference System</xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string" />
</xsd:simpleType>
<xsd:complexType name="AddressReferenceSystemAxis_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">The line that defines
 the points of origin for address numbering along thoroughfares that
 intersect it, or which are numbered in parallel to streets that
 intersect it. It may be a road, another geographic feature, or an
 imaginary line.</xsd:documentation>
```

```
</xsd:annotation>
<xsd:complexContent>
 <xsd:restriction base="gml:MultiCurveType">
 </xsd:restriction>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="AddressReferenceSystemAxisPointOfBeginning_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">Coordinate location of
 the beginning point of address numbering along an Address Reference
 System Axis. </xsd:documentation>
</xsd:annotation>
<xsd:complexContent>
 <xsd:extension base="gml:PointType"></xsd:extension>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="AddressReferenceSystemReferencePolyline_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">A street, geometric
 line, or other line used to measure address number assignment
 intervals and ranges within an Address Reference System. The Address
 Reference System Reference Polyline may consist of a beginning
 point, one or more segments of a street centerline, geographically
 identified line, such as a line of latitude or longitude, a
 land-division based line, such as a township, range, or section
 line, or an imaginary line constructed for the purpose of allocating
 address ranges and address numbers.</xsd:documentation>
</xsd:annotation>
<xsd:complexContent>
 <xsd:restriction base="gml:MultiCurveType"></xsd:restriction>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="AddressReferenceSystemRangeBreakpoint_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">A point along a street
 or other thoroughfare within an Address Reference System where an
 address range beginning and/or endpoint is located.
 </xsd:documentation>
</xsd:annotation>
<xsd:complexContent>
 <xsd:extension base="gml:PointType">
 </xsd:extension>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="AddressReferenceSystemRangeBreakline_type">
```

```
<xsd:annotation>
 <xsd:documentation xml:lang="en">A line connecting the
 Address Reference System Range Breakpoints with the same value
 within an Address Reference System</xsd:documentation>
</xsd:annotation>
<xsd:complexContent>
 <xsd:restriction base="gml:MultiCurveType">
 </xsd:restriction>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="AddressReferenceSystemRangePolygon_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">A line connecting the
 Address Reference System Range Breakpoints with the same value
 within an Address Reference System</xsd:documentation>
</xsd:annotation>
<xsd:complexContent>
 <xsd:restriction base="gml:MultiSurfaceType">
 </xsd:restriction>
</xsd:complexContent>
</xsd:complexType>
<xsd:simpleType name="AddressReferenceSystemReferenceDocumentCitation_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">A bibliographic
 reference to an ordinance, map, manual, or other document in which
 the rules governing an Address Reference System are written.
 </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string" />
</xsd:simpleType>
<xsd:complexType name="AddressReferenceSystem_type">
<xsd:annotation>
 <xsd:documentation>An Address Reference System is a set of rules and
 geometries that define how addresses are
 assigned along thoroughfares
 and/or within a given area (Address Reference
 System Extent).
 At
 minimum, an Address Reference System must specify where Complete
 Address Number sequences
 begin and how Complete Address Numbers are
 assigned along the length of
 thoroughfares governed by
 the Address
 Reference System within the Address Reference System
```

```
 Extent. Address
 Reference Systems
 typically provide rules governing left-right parity
 of Complete Address
 Numbers, assignment of Street Names
 and street
 types, use of directionals and quadrants, and other aspects
 of
 address assignment.An Address
 Reference System that is based on axis
 lines, an Address Reference System
 Axis defined for each axis used
 to
 define address assignment. Each Address Reference System Axis must
 have an identified Address Reference
 System Axis Point Of Beginning.
 An Address Reference System is known by
 its Address Reference System
 Name (required). Additional business rules for an Address Reference
 System are described in the Address Reference
 System Rules.
 </xsd:documentation>
</xsd:annotation>
<xsd:sequence>
 <xsd:element name="AddressReferenceSystemId"
 type="addr_type:AddressReferenceSystemId_type" maxOccurs="1"
 minOccurs="1" />
 <xsd:element name="AddressReferenceSystemName"
 type="addr_type:AddressReferenceSystemName_type" maxOccurs="1"
 minOccurs="1" />
 <xsd:element name="AddressReferenceSystemAuthority"
 type="addr_type:AddressReferenceSystemAuthority_type" maxOccurs="1"
 minOccurs="0" />
 <xsd:element name="AddressReferenceSystemExtent"
 type="addr_type:AddressReferenceSystemExtent_type" maxOccurs="1"
 minOccurs="0" />
 <xsd:element name="AddressReferenceSystemType"
 type="addr_type:AddressReferenceSystemType_type" maxOccurs="1"
 minOccurs="0" />
 <xsd:element name="AddressReferenceSystemRules"
 type="addr_type:AddressReferenceSystemRules_type" maxOccurs="1"
 minOccurs="0" />
 <xsd:element name="AddressReferenceSystemAxis"
 type="addr_type:AddressReferenceSystemAxis_type" maxOccurs="1"
 minOccurs="0" />
```

```
 <xsd:element name="AddressReferenceSystemAxisPointOfBeginning"
 type="addr_type:AddressReferenceSystemAxisPointOfBeginning_type"
 maxOccurs="1" minOccurs="0" />
 <xsd:element name="AddressReferenceSystemReferencePolyline"
 type="addr_type:AddressReferenceSystemReferencePolyline_type"
 maxOccurs="unbounded" minOccurs="0" />
 <xsd:element name="AddressReferenceSystemRangeBreakpoint"
 type="addr_type:AddressReferenceSystemRangeBreakpoint_type"
 maxOccurs="1" minOccurs="0" />
 <xsd:element name="AddressReferenceSystemRangeBreakline"
 type="addr_type:AddressReferenceSystemRangeBreakline_type"
 maxOccurs="unbounded" minOccurs="0" />
 <xsd:element name="AddressReferenceSystemReferenceDocumentCitation"
 type="addr_type:AddressReferenceSystemReferenceDocumentCitation_type"
 maxOccurs="unbounded" minOccurs="0" />
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="RelatedAddressID_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">
 The identifier of an
 address that is related to the
 identifier of another address.
 </xsd:documentation>
</xsd:annotation>
<xsd:simpleContent>
 <xsd:extension base="addr_type:AddressID_type">
 <xsd:attribute name="AddressRelationType"
  type="addr_type:AddressRelationType_type" />
 </xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
<xsd:simpleType name="AddressRelationType_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">The manner in which an
 address identified by a RelatedAddressID is related to an address
 identified by an AddressID.
 </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string">
 <xsd:pattern value='.*' />
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="RelatedTransportationFeatureID_type">
<xsd:annotation>
```

```
 <xsd:documentation xml:lang="en">The unique identifier
 assigned (within the reference transportation base model) to a
 transportation feature to which an address is related. see U.S.
 Federal Geographic Data Committee, "Framework Data Content Standard
 Part 7: Transportation base."
 "Framework Data Content Standard Part
 7c: Roads."
 </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string">
 <xsd:pattern value='.*' />
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="AddressTransportationFeatureID_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">The unique identifier
 assigned to the particular feature that represents an address within
 a transportation base model. see U.S. Federal Geographic Data
 Committee, "Framework Data Content Standard Part 7: Transportation
 base."
 "Framework Data Content Standard Part 7c: Roads."
 </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string">
 <xsd:pattern value='.*' />
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="AddressTransportationFeatureType_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">The type of
 transportation feature (TranFeature) used to represent an address.
 For transportation features generally: U.S. Federal Geographic Data
 Committee, "Framework Data Content Standard Part 7: Transportation
 base."
 For roads features only: U.S. Federal Geographic Data
 Committee,
 "Framework Data Content Standard Part 7: Transportation
 base," as
 extended by "Framework Data Content Standard Part 7c:
 Roads."
 </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string">
 <xsd:pattern value='.*' />
</xsd:restriction>
```

```
</xsd:simpleType>
<xsd:simpleType name="AddressTransportationSystemAuthority_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">The authority that
 maintains the transportation base model specified by the Address
 Transportation System Name, and assigns Address Transportation
 Feature I Ds to the features it represents.
</xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string">
 <xsd:pattern value='.*' />
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="AddressTransportationSystemName_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">The name of the
 transportation base model to which the address is related.
 </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string">
 <xsd:pattern value='.*' />
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="AddressParcelIdentifier_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">The primary permanent
 identifier, as defined by the Address Parcel Identifier Source, for
 a parcel that includes the land or feature identified by an address.
 A parcel is "a single cadastral unit, which is the spatial
 extent of the past, present, and future rights and interests in real
 property."
 Definition source for "parcel identifier":
 Adapted from FGDC, May
 2008. "Geographic Information Framework
 Data Content Standard
 Part 1: Cadastral." Section 4.2.
 Definition source for "parcel": FGDC, May 2008.
 "Cadastral Data
 Content Standard for the National Spatial Data
 Infrastructure."
 Vesion 1.4 - Fourth Revision. p. 45. (Part 3.2
 "Parcel)
 </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string">
```

```
 <xsd:pattern value='.*' />
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="AddressParcelIdentifierSource_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">The permanent
 identifier for the agency, organization, or jurisdiction that
 assigns and maintains the Address Parcel Identifier.
 Definition
 source: FGDC, May 2008. "Geographic Information Framework Data
 Content Standard Part 1: Cadastral." Section 4.7.
 </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string">
 <xsd:pattern value='.*' />
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="AddressUUId_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">
 The unique
 identification number assigned to an address
 by the addressing
 authority. The ID number must be
 unique for each address assigned by
 an addressing
 authority. This, combined with the FIPS number of the
 addressing authority, can provide a unique ID for every
 address in
 the US.
 </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string">
 <xsd:pattern value='.*' />
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="AssociatedAddressId_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">
 The unique
 identification number of and address related to this one.
 </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string">
 <xsd:pattern value='.*' />
```

```
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="MailableAddress_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">
 Identifies whether an
 addresses receives USPS mail
 delivery (that is, the address is
 occupiable, and the
 USPS provides provides on-premises USPS mail
 delivery to
 it).
 </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string">
 <xsd:pattern value='.*' />
 <xsd:enumeration value="Yes">
 <xsd:annotation>
  <xsd:documentation>The USPS delivers mail to this address.
  </xsd:documentation>
 </xsd:annotation>
 </xsd:enumeration>
 <xsd:enumeration value="No">
 <xsd:annotation>
  <xsd:documentation>The USPS does not deliver mail to this address.
  </xsd:documentation>
 </xsd:annotation>
 </xsd:enumeration>
 <xsd:enumeration value="Unknown">
 <xsd:annotation>
  <xsd:documentation>It is unknown whether the USPS delivers mail to
  this address.</xsd:documentation>
 </xsd:annotation>
 </xsd:enumeration>
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="AddressSideOfStreet_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">
 The side of the
 transportation segment (right , left,
 both, none, unknown) on which
 the address is located.
 U.S. Federal Geographic Data Committee,
 "Framework Data
```

```
  Content Standard Part 7: Transportation base,"
  sections
  7.3.2 and B.3.6
  </xsd:documentation>
 </xsd:annotation>
 <xsd:restriction base="xsd:string">
  <xsd:pattern value='.*' />
  <xsd:enumeration value="right">
  <xsd:annotation>
   <xsd:documentation>
   The address is related to the right side of the
   street.
   </xsd:documentation>
  </xsd:annotation>
  </xsd:enumeration>
  <xsd:enumeration value="left">
  <xsd:annotation>
   <xsd:documentation>
   The address is realted to the left side of the
   street.
   </xsd:documentation>
  </xsd:annotation>
  </xsd:enumeration>
  <xsd:enumeration value="both">
  <xsd:annotation>
   <xsd:documentation>
   The address pertains to both sides of the
   street.
   </xsd:documentation>
  </xsd:annotation>
  </xsd:enumeration>
  <xsd:enumeration value="none">
  <xsd:annotation>
   <xsd:documentation>The address is not on either or both sides of
   the street or the concept of side of street does not apply to the
   address.
   For instance an intersection address would have a
   AddressSideOfStreet
   of none.
   </xsd:documentation>
  </xsd:annotation>
  </xsd:enumeration>
  <xsd:enumeration value="unknown"></xsd:enumeration>
 </xsd:restriction>
 </xsd:simpleType>
```

```
<xsd:simpleType name="AddressRangeSide_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">
 The side of the
 transportation segment (right , left,
 both, none, unknown) on which
 the address range applies.
 </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string">
 <xsd:pattern value='.*' />
 <xsd:enumeration value="right">
 <xsd:annotation>
  <xsd:documentation>
  The address is related to the right side of the
  street.
  </xsd:documentation>
 </xsd:annotation>
 </xsd:enumeration>
 <xsd:enumeration value="left">
 <xsd:annotation>
  <xsd:documentation>
  The address is realted to the left side of the
  street.
  </xsd:documentation>
 </xsd:annotation>
 </xsd:enumeration>
 <xsd:enumeration value="both">
 <xsd:annotation>
  <xsd:documentation>
  The address pertains to both sides of the
  street.
  </xsd:documentation>
 </xsd:annotation>
 </xsd:enumeration>
 <xsd:enumeration value="none">
 <xsd:annotation>
  <xsd:documentation>The address is not on either or both sides of
  the street or the concept of side of street does not apply to the
  address.
  For instance an intersection address would have a
  AddressSideOfStreet
  of none.
  </xsd:documentation>
 </xsd:annotation>
```

```
 </xsd:enumeration>
 <xsd:enumeration value="unknown"></xsd:enumeration>
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="AddressRangeParity_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">
 The set of Address
 Number Parity values specified in the Address Reference System
 Numbering Rules for the Address Numbers in an address range.
 </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string">
 <xsd:pattern value='.*' />
 <xsd:enumeration value="even">
 <xsd:annotation>
  <xsd:documentation>
  All Address Numbers in the range have an Address
  Number Parity of "even".
  </xsd:documentation>
 </xsd:annotation>
 </xsd:enumeration>
 <xsd:enumeration value="odd">
 <xsd:annotation>
  <xsd:documentation>
  All Address Numbers in the range have an Address
  Number Parity of "odd".
  </xsd:documentation>
 </xsd:annotation>
 </xsd:enumeration>
 <xsd:enumeration value="both">
 <xsd:annotation>
  <xsd:documentation>
  Both even and odd Address Numbers are found in
  the range.
  </xsd:documentation>
 </xsd:annotation>
 </xsd:enumeration>
 <xsd:enumeration value="none">
 <xsd:annotation>
  <xsd:documentation>
  No Address Number is found within the range.
  </xsd:documentation>
 </xsd:annotation>
 </xsd:enumeration>
```

```
 <xsd:enumeration value="unknown">
 <xsd:annotation>
  <xsd:documentation>The parity of the Address Numbers in the range
  in not known. </xsd:documentation>
 </xsd:annotation>
 </xsd:enumeration>
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="OfficialStatus_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">
 Whether the address,
 street name, landmark name, or
 place name is as given by the official
 addressing
 authority (official), or an alternate or alias (official
 or unofficial), or a verified error.
 </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string">
 <xsd:pattern value='.*' />
 <xsd:enumeration value="Official">
 <xsd:annotation>
  <xsd:documentation>
  The address or name as designated by the Address
  Authority.
  </xsd:documentation>
 </xsd:annotation>
 </xsd:enumeration>
 <xsd:enumeration value="Alternate or Alias">
 <xsd:annotation>
  <xsd:documentation>
  An alternate or alias to the official address or
  name that is also in official or popular use.
  The Related Address
  ID can be used to link an
  alternate or alias to the Address ID of
  the
  official address. There are two types of
  alternate or alias
  names, official and
  unofficial, each of which has subtypes.
  </xsd:documentation>
 </xsd:annotation>
 </xsd:enumeration>
```

```xml
<xsd:enumeration value="Official Alternate or Alias">
<xsd:annotation>
 <xsd:documentation>
 These are alternate names designated by an
 official Address Authority.
 </xsd:documentation>
</xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="Official Renaming Action of the Address Authority">
<xsd:annotation>
 <xsd:documentation>An Address Authority may replace one address or
 name with another, e.g. by renaming or renumbering. The prior,
 older address should be retained as an alias, to provide for
 conversion to the new address.</xsd:documentation>
</xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="Alternates Established by an Address Authority">
<xsd:annotation>
 <xsd:documentation>An Address Authority may establish a name or
 number to be used in addition to the official address or name. For
 example, a state highway designation (State Highway 7) may be
 given to a locally-named road, or a memorial name may be applied
 to an existing street by posting an additional sign, while the
 local or original name and addresses continue to be recognized as
 official.</xsd:documentation>
</xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="Unofficial Alternate or Alias">
<xsd:annotation>
 <xsd:documentation>
 These are addresses or names that are used by
 the public or by an individual, but are not
 recognized as official
 by the Address Authority.
 </xsd:documentation>
</xsd:annotation>
</xsd:enumeration>
<xsd:enumeration
value="Alternate Names Established by Colloquial Use in a Community">
<xsd:annotation>
 <xsd:documentation>An address or name that is in popular use but is
 not the official name or an official alternate or alias.
 </xsd:documentation>
</xsd:annotation>
</xsd:enumeration>
```

```
  <xsd:enumeration value="Unofficial Alternate Names Frequently Encountered">
  <xsd:annotation>
   <xsd:documentation>In data processing, entry errors occur. Such
   errors if frequently encountered may be corrected by a direct
   match of the error and a substitution of a correct name.
   </xsd:documentation>
  </xsd:annotation>
  </xsd:enumeration>
  <xsd:enumeration
  value="Unofficial Alternate Names In Use by an Agency or Entity">
  <xsd:annotation>
   <xsd:documentation>For data processing efficiency, entities often
   create alternate names or abbreviations for internal use. These
   must be changed to the official form for public use and
   transmittal to external users.</xsd:documentation>
  </xsd:annotation>
  </xsd:enumeration>
  <xsd:enumeration value="Posted or Vanity Address">
  <xsd:annotation>
   <xsd:documentation>An address that is posted, but is not recognized
   by the Address Authority (e.g. a vanity address on a building);
   </xsd:documentation>
  </xsd:annotation>
  </xsd:enumeration>
  <xsd:enumeration value="Verified Invalid">
  <xsd:annotation>
   <xsd:documentation>
   An address that has been verified as being
   invalid, but which keeps appearing in address
   lists. Different from
   Unofficial Alternate Names
   in that these addresses are known not to
   exist.
   </xsd:documentation>
  </xsd:annotation>
  </xsd:enumeration>
 </xsd:restriction>
 </xsd:simpleType>
 <xsd:simpleType name="AddressStartDate_type">
 <xsd:annotation>
  <xsd:documentation xml:lang="en">
  The earliest date on
  which the address is known to
  exist.
  </xsd:documentation>
```

```
</xsd:annotation>
<xsd:restriction base="xsd:date" />
</xsd:simpleType>
<xsd:simpleType name="AddressEndDate_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">
 The earliest date on
 which the address is known to no
 longer be valid.
 </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:date" />
</xsd:simpleType>
<xsd:simpleType name="AddressClassification_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">
 The type or
 classification of the address according to
 the classification
 standard.
 </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string">
 <xsd:enumeration value="NumberedThoroughfareAddress"></xsd:enumeration>
 <xsd:enumeration value="IntersectionAddress"></xsd:enumeration>
 <xsd:enumeration value="TwoNumberAddressRange"></xsd:enumeration>
 <xsd:enumeration value="FourNumberAddressRange"></xsd:enumeration>
 <xsd:enumeration value="UnnumberedThoroughfareAddress"></xsd:enumeration>
 <xsd:enumeration value="LandmarkAddress"></xsd:enumeration>
 <xsd:enumeration value="CommunityAddress"></xsd:enumeration>
 <xsd:enumeration value="USPSPostalDeliveryBox"></xsd:enumeration>
 <xsd:enumeration value="USPSPostal DeliveryRoute"></xsd:enumeration>
 <xsd:enumeration value="USPSGeneral DeliveryOffice"></xsd:enumeration>
 <xsd:enumeration value="GeneralAddressClass"></xsd:enumeration>
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="AddressFeatureType_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">
 The type of feature
 identified by the address
 Initial list of feature types: Building
 Utility Cabinet,
 Telephone Pole, Building, Street block, street block
 face, intersection, parcel, building, entrance, unit.
```

```
 The list might
 be expanded indefinitely to include
 infrastructure and other
 features.
 </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string">
 <xsd:pattern value='.+' />
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="AddressLifecycle_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">
 The Lifecycle status of
 the address.
 </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:token">
 <xsd:enumeration value='PROPOSED' />
 <xsd:enumeration value='ACTIVE' />
 <xsd:enumeration value='RETIRED' />
 <xsd:enumeration value='TEMPORARY' />
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="AddressLifecycleStatus_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">
 The life cycle status
 of the address.
 </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:token">
 <xsd:enumeration value="Potential">
 <xsd:annotation>
  <xsd:documentation>
  Address falls within a theoretical range, but
  has never been used.
  </xsd:documentation>
 </xsd:annotation>
 </xsd:enumeration>
 <xsd:enumeration value="Proposed">
 <xsd:annotation>
  <xsd:documentation>
  Application pending for use of this address
  (e.g., address tentatively issued for
```

```
  subdivision plat that is not
  yet fully
  approved).
  </xsd:documentation>
 </xsd:annotation>
 </xsd:enumeration>
 <xsd:enumeration value="Active">
 <xsd:annotation>
  <xsd:documentation>
  Address has been issued and is in use.
  </xsd:documentation>
 </xsd:annotation>
 </xsd:enumeration>
 <xsd:enumeration value="Retired">
 <xsd:annotation>
  <xsd:documentation>
  Address was issued, but is now obsolete (e.g.
  street name has been changed), building was
  demolished, etc.
  </xsd:documentation>
 </xsd:annotation>
 </xsd:enumeration>
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="AddressOfficialStatus_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">
 Whether the address is
 as given by the official
 addressing authority (official), or an
 unofficial
 variant or equivalent of it (alias).
 </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:token">
 <xsd:enumeration value="Official">
 <xsd:annotation>
  <xsd:documentation>
  The address or name as designated by the
  addressing authority.
  </xsd:documentation>
 </xsd:annotation>
 </xsd:enumeration>
 <xsd:enumeration value="Alternate Name">
 <xsd:annotation>
```

```
 <xsd:documentation>
 In any of the address classes described in 2.2,
 the collective name element may have another
 acceptable form. Some
 alternate names may be
 conditional, on attempt, ie if the alias
 resolves the address no further alternate names
 should be
 considered. Other alternate names are
 always applied, such as
 official renamings. All
 alternate names carry a limit of
 applicability
 and a timeframe of applicability. The limit of
 appicability may be a limit to a single zipcode,
 a naming
 authorities boundary, such as city or
 county limits, or a range of
 address numbers
 with such a boundary.
 </xsd:documentation>
</xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="Alternate Renamed">
<xsd:annotation>
 <xsd:documentation>
 Upon official renaming of an address, or
 renumbering of an address, or a series of
 addresses, the prior,
 older address will occur
 in address lists for a period of time and
 a
 conversion to current names or current addresses
 will need to be
 provided. Such an entity may
 match a single address or a range of
 addresses.
 </xsd:documentation>
</xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="Alternate Authority Name">
<xsd:annotation>
 <xsd:documentation>
 The alternate name is established by a separate,
 or the same, naming authority. Such names may
```

```
 apply to any address
 class, including landmarks.
 Such names would be established by
 naming
 authorities with a geographically larger area of
 responsibility, containing all or part of a
 naming authority with a
 smaller region, such as
 a state name overlaying a county name or a
 county name overlaying a city or town name.
 Examples would be a
 state highway designation
 (State Highway 7) overlaid upon locally
 named
 roads or a memorial highway overlaid on local
 road names or
 state highway names.
 </xsd:documentation>
</xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="Alternate Colloquial Name">
<xsd:annotation>
 <xsd:documentation>
 Local communities hold on to address names much
 longer than do regional agencies. A community
 may use a colloquial
 address name as much as 30
 years after that name has either expired
 or is
 no longer salient. This entry provides a
 conversion to a
 current name.
 </xsd:documentation>
</xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="Unoffical Alternate Name">
<xsd:annotation>
 <xsd:documentation>
 In data processing, entry errors occur. Such
 errors if frequently encountered may be
 corrected by a direct match
 of the error and a
 substitution to a current name.
 </xsd:documentation>
</xsd:annotation>
```

```
 </xsd:enumeration>
 <xsd:enumeration value="Unofficial Agencys Name">
 <xsd:annotation>
  <xsd:documentation>
  For data processing efficiency, entities often
  create alternate names for internal use. When
  such alternate names
  are exposed to other
  entities they need to be resolved to a current
  name.
  </xsd:documentation>
 </xsd:annotation>
 </xsd:enumeration>
 <xsd:enumeration value="Posted Address">
 <xsd:annotation>
  <xsd:documentation>
  Address is posted, but not recognized by
  addressing authority (e.g. vanity address on a
  building).
  </xsd:documentation>
 </xsd:annotation>
 </xsd:enumeration>
 <xsd:enumeration value="Verified Invalid">
 <xsd:annotation>
  <xsd:documentation>
  Address is verified as being invalid, but keeps
  appearing in address lists. Different from
  Unoffical Altername
  Names in that these are
  known not to exist;Address has been issued
  and
  is in use.
  </xsd:documentation>
 </xsd:annotation>
 </xsd:enumeration>
 <xsd:pattern value=".+"></xsd:pattern>
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="AddressAnomalyStatus_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">
 A status flag, or an
 explanatory note, for an address
 that is not correct according to the
 address schema in
```

```
 which it is located, but is nonetheless a valid
 address.
 This field may be used to identify the type of anomaly
 (e.g.
 wrong parity, out of sequence, out of range, etc.)
 rather than simply
 whether or not it is anomalous. Local
 jurisdictions may create
 specific categories for
 anomalies.
 </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string"></xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="AddressRangeSpan_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">
 Whether an address
 range covers part of a transportation
 segment, one segment, multiple
 segments, or the entire
 thoroughfare within the Address Reference
 System Extent.
 </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string">
 <xsd:enumeration value="Partial Segment"></xsd:enumeration>
 <xsd:enumeration value="Single Segment"></xsd:enumeration>
 <xsd:enumeration value="Multi Segment"></xsd:enumeration>
 <xsd:enumeration value="Entire Street"></xsd:enumeration>
 <xsd:enumeration value="Unknown"></xsd:enumeration>
 <xsd:pattern value=".+"></xsd:pattern>
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="AddressRangeDirectionality_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">
 Whether the low
 Complete Address Number of an address range is closer to the
 from-node or the to-node of the transportation segment(s) that the
 range is related to.
 </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string">
 <xsd:enumeration value="With">
```

```
<xsd:annotation>
 <xsd:documentation>The low address is nearer the from node; numbers
 ascend toward the to node.
 </xsd:documentation>
</xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="Against">
<xsd:annotation>
 <xsd:documentation>The low address is nearer the to node; numbers
 descend toward the to node.
 </xsd:documentation>
</xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="With-Against">
<xsd:annotation>
 <xsd:documentation>The numbers run in opposite directions on either
 side of the street. The low number on the left side is nearer the
 from node. The low number on the right side is nearer the to node.
 </xsd:documentation>
</xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="Against-With">
<xsd:annotation>
 <xsd:documentation>The numbers run in opposite directions on either
 side of the street. The low number on the left side is nearer the
 to node. The low number on the right side is nearer the from node.
 </xsd:documentation>
</xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="Null">
<xsd:annotation>
 <xsd:documentation>The address range has null values for the high
 and low Complete Address Numbers.
 </xsd:documentation>
</xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="NA">
<xsd:annotation>
 <xsd:documentation>Does not apply (transportation segment
 directionality is inconsistent within the range).
 </xsd:documentation>
</xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="Unknown">
<xsd:annotation>
```

```
   <xsd:documentation>The address range directionality is not known.
   </xsd:documentation>
  </xsd:annotation>
  </xsd:enumeration>
 </xsd:restriction>
 </xsd:simpleType>
 <xsd:simpleType name="AddressRangeType_type">
 <xsd:annotation>
  <xsd:documentation xml:lang="en">This attribute states
  whether an address range (either a Two Number Address Range or a
  Four Number Address Range) is actual or potential.
  </xsd:documentation>
 </xsd:annotation>
 <xsd:restriction base="xsd:string">
  <xsd:enumeration value="Actual">
  <xsd:annotation>
   <xsd:documentation>
   The low and high CompleteAddressNumbers are
   numbers that have been assigned and are in use
   along the addressed
   feature.
   </xsd:documentation>
  </xsd:annotation>
  </xsd:enumeration>
  <xsd:enumeration value="Potential">
  <xsd:annotation>
   <xsd:documentation>
   The low and high CompleteAddressNumbers are
   numbers that would be assigned if all possible
   numbers were in use
   along the addressed feature,
   and there were no gaps between the
   range and its
   preceding and following ranges.
   </xsd:documentation>
  </xsd:annotation>
  </xsd:enumeration>
  <xsd:enumeration value="Unknown">
  <xsd:annotation>
   <xsd:documentation>
   The relationship between the low and high
   CompleteAddressNumbers and the addressed feature
   is unknown.
   </xsd:documentation>
  </xsd:annotation>
```

```
 </xsd:enumeration>
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="LocationDescription_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">
 A text description
 providing more detail on how to
 identify or find the addressed
 feature.
 </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string"></xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="AddressNumberParity_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en"> The property of an
 Address Number with respect to being odd or even.
 "A relation between
 a pair of integers: if both integers
 are odd or
 both are even they
 have the same parity; if
 one is odd and the other
 is even they have
 different
 parity."
 </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:token">
 <xsd:enumeration value="Even" />
 <xsd:enumeration value="Odd" />
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="AttachedElement_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">
 This attribute
 identifies when two or more Complete
 Address Number elements or two
 or more Complete Street
 Name elements have been combined without a
 space
 separating them.
 </xsd:documentation>
```

```
</xsd:annotation>
<xsd:restriction base="xsd:string">
 <xsd:enumeration value="Attached">
 <xsd:annotation>
  <xsd:documentation>
  The elements inside the CompleteAddressNumber or
  CompleteStreetName are attached and need special
  parsing rules.
  </xsd:documentation>
 </xsd:annotation>
 </xsd:enumeration>
 <xsd:enumeration value="Not Attached"></xsd:enumeration>
 <xsd:enumeration value="Unknown"></xsd:enumeration>
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="AddressNumberSide_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">
 "The Concept of Left
 and Right sides of a feature that a Number Range Applies to.
 </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:token">
 <xsd:enumeration value="Left" />
 <xsd:enumeration value="Right" />
 <xsd:enumeration value="Unknown" />
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="AddressNumberBounds_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">
 "The Concept of Low or
 High of numbers participating in a Number Range Applies to.
 </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:token">
 <xsd:enumeration value="Low" />
 <xsd:enumeration value="High" />
 <xsd:enumeration value="Unknown" />
</xsd:restriction>
</xsd:simpleType>
<xsd:complexType name="StreetNameGroup">
<xsd:annotation>
 <xsd:documentation xml:lang="en">
 A geographic area where
```

```
 the street names conform to a
 theme. For example, some neighborhoods
 feature streets
 named for birds, US presidents or trees. A subset of
 the
 complete street name domain applies to this area.
 </xsd:documentation>
</xsd:annotation>
<xsd:complexContent>
 <xsd:extension base="gml:MultiSurfaceType">
 <xsd:attribute name="name" type="xsd:string"></xsd:attribute>
 </xsd:extension>
</xsd:complexContent>
</xsd:complexType>
<xsd:simpleType name="AddressSchemeName_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">
 Name of the address
 scheme that operates over a
 specified area, i.e.: mountain addresses,
 plains
 addresses.
 </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string">
 <xsd:pattern value='.*' />
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="AddressSchemeDescription_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">
 A description of an
 Address Scheme that includes
 business rules about parity, naming
 conventions, and
 other matters concerning the assignment and
 maintenance
 of an addressing scheme.
 This element may refer to an
 address ordinance, Standard
 Operating Procedures manual or other
 external document
 wherein the rules for addresses in a given scheme
 are
 written.
```

```
 </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string">
 <xsd:pattern value='.*' />
</xsd:restriction>
</xsd:simpleType>
<!-- add axes name as an attribute -->
<!-- change data type to GML::Point -->
<xsd:complexType name="AddressSchemeOrigin_type ">
<xsd:annotation>
 <xsd:documentation xml:lang="en">
 Location where the
 address axes meet.
 </xsd:documentation>
</xsd:annotation>
<xsd:complexContent>
 <xsd:extension base="gml:PointType">
 <xsd:attribute name="OriginValue" type="xsd:int"></xsd:attribute>
 <xsd:attribute name="AxisId" type="xsd:string"></xsd:attribute>
 </xsd:extension>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="AddressSchemeAxes_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">
 Address axes define the
 boundaries between adjoining
 zones in address schema. Those zones may
 be quadrants
 (northwest, northeast, southeast, southwest) or other
 geographic divisions. Lowest address numbers occur
 nearest an axis.
 </xsd:documentation>
</xsd:annotation>
<xsd:complexContent>
 <xsd:extension base="gml:MultiCurveType">
 <xsd:attribute name="AxisId" type="xsd:string"></xsd:attribute>
 </xsd:extension>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="AddressSchemeExtent_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">
 Boundary of the area
 over which an address schema
```

```
 is used when assiging addresses.
 </xsd:documentation>
</xsd:annotation>
<xsd:complexContent>
 <xsd:extension base="gml:MultiSurfaceType" />
</xsd:complexContent>
</xsd:complexType>
<xsd:simpleType name="AddressDirectSource_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">
 Source from whom the
 data provider obtained the address,
 or with whom the data provider
 validated the address.
 Important if the data provider did not obtain
 the
 address directly from the local authority.
 </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string">
 <xsd:pattern value='.*' />
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="AddressAuthority_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">
 The authority (e.g.,
 municipality, county) that created
 or has jurisdiction over the
 creation of an address.
 The addressing authority may or may not be
 the same as
 the physical or postal jurisdiction noted for the
 address.
 </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string">
 <xsd:pattern value='.*' />
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="AddressAuthorityIdentifier_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">
 The FIPS or GNIs code
 for the authority (e.g.,
```

```
 municipality, county) that created or has
 jurisdiction
 over the creation of an address.
 The addressing authority
 may or may not be the same as
 the physical or postal jurisdiction
 noted for the
 address.
 </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string">
 <xsd:pattern value='.*' />
</xsd:restriction>
</xsd:simpleType>
<!-- Complex Types -->
<xsd:simpleType name="Action_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">
 An action command for
 incremental datasets. Add
 indicates that the information is new.
 DELETE indicates
 that the information is to be removed.
 </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:token">
 <xsd:enumeration value='ADD' />
 <xsd:enumeration value='DELETE' />
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="DeliveryAddressType_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">
 Whether the Delivery
 Address includes or excludes the Complete Subaddress.
 </xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:token">
 <xsd:enumeration value='SubAddress Included'>
 <xsd:annotation>
  <xsd:documentation>The Delivery Address includes the Complete
  Subaddress (if any) </xsd:documentation>
 </xsd:annotation>
 </xsd:enumeration>
 <xsd:enumeration value='SubAddres Excluded'>
```

```
  <xsd:annotation>
   <xsd:documentation>The Delivery Address includes the Complete
   Subaddress (if any) </xsd:documentation>
  </xsd:annotation>
  </xsd:enumeration>
  <xsd:enumeration value='Unstated'>
  <xsd:annotation>
   <xsd:documentation>Not stated/no information (default value)
   </xsd:documentation>
  </xsd:annotation>
  </xsd:enumeration>
 </xsd:restriction>
 </xsd:simpleType>
 <!-- Complex Elements -->
 <xsd:complexType name="DeliveryAddress_type">
 <xsd:annotation>
  <xsd:documentation xml:lang="en">
  The entire address,
  unparsed, except for the Place Name, State Name,
  Zip Code, Zip Plus
  4, Country Name, and, optionally,
  Complete Subaddress elements.
  </xsd:documentation>
 </xsd:annotation>
 <xsd:simpleContent>
  <xsd:extension base="xsd:string">
  <xsd:attribute name="DeliveryAddressType"
   type="addr_type:DeliveryAddressType_type" />
  </xsd:extension>
 </xsd:simpleContent>
 </xsd:complexType>
 <xsd:simpleType name="PlaceStateZip_type">
 <xsd:annotation>
  <xsd:documentation xml:lang="en">
  The unparsed
  accumulation of Postal City, State, and
  ZIPCode elements.
  </xsd:documentation>
 </xsd:annotation>
 <xsd:restriction base="xsd:string">
  <xsd:pattern value='.*' />
 </xsd:restriction>
 </xsd:simpleType>
 <xsd:complexType name="FeatureOccupancy_type">
 <xsd:annotation>
```

```
 <xsd:documentation xml:lang="en">
 This element is defined
 solely for use with the General
 Address class, which is constructed
 to accommodate and
 mix addresses of all types (e.g., a general postal
 mailing list or contact list). Place Name, State Name,
 Zip Code, and
 Zip Plus 4, which appear in all address
 classes, are kept separate
 from the rest of the address.
 No longer a parsed datatype. Content
 still represents it
 as such.
 </xsd:documentation>
</xsd:annotation>
<xsd:simpleContent>
 <xsd:extension base="xsd:string"></xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
<xsd:complexType name="GeneralAddress_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">
 This element is defined
 solely for use with the General
 Address class, which is constructed
 to accommodate and
 mix addresses of all types (e.g., a general postal
 mailing list or contact list). Place Name, State Name,
 Zip Code, and
 Zip Plus 4, which appear in all address
 classes, are kept separate
 from the rest of the address.
 No longer a parsed datatype. Content
 still represents it
 as such.
 </xsd:documentation>
</xsd:annotation>
<xsd:simpleContent>
 <xsd:extension base="xsd:string" />
</xsd:simpleContent>
</xsd:complexType>
<!--    -->
<xsd:complexType name="LocationXY_type">
<xsd:sequence>
```

```
  <xsd:element name="X" type="addr_type:AddressXCoordinate_type"
  minOccurs="1" maxOccurs="1" />
  <xsd:element name="Y" type="addr_type:AddressYCoordinate_type"
  minOccurs="1" maxOccurs="1" />
 </xsd:sequence>
 </xsd:complexType>
 <xsd:complexType name="CompleteStreetName_type">
 <xsd:sequence>
  <xsd:element name="StreetNamePreModifier"
type="addr_type:StreetNameModifier_type"
  minOccurs="0" maxOccurs="1" />
  <xsd:element name="StreetNamePreDirectional"
type="addr_type:StreetNameDirectional_type"
  minOccurs="0" maxOccurs="1" />
  <xsd:element name="StreetNamePreType" type="addr_type:StreetNameType_type"
  minOccurs="0" maxOccurs="1" />
  <xsd:element name="StreetName" type="addr_type:StreetName_type"
  minOccurs="1" maxOccurs="1" />
  <xsd:element name="StreetNamePostType" type="addr_type:StreetNameType_type"
  minOccurs="0" maxOccurs="1" />
  <xsd:element name="StreetNamePostDirectional"
type="addr_type:StreetNameDirectional_type"
  minOccurs="0" maxOccurs="1" />
  <xsd:element name="StreetNamePostModifier"
type="addr_type:StreetNameModifier_type"
  minOccurs="0" maxOccurs="1" />
 </xsd:sequence>
 <xsd:attribute name="AttachedElement" type="addr_type:AttachedElement_type" />
 </xsd:complexType>
 <xsd:group name="CompleteStreetName_group">
 <xsd:sequence>
  <xsd:element name="StreetNamePreModifier"
type="addr_type:StreetNameModifier_type"
  minOccurs="0" maxOccurs="1" />
  <xsd:element name="StreetNamePreDirectional"
type="addr_type:StreetNameDirectional_type"
  minOccurs="0" maxOccurs="1" />
  <xsd:element name="StreetNamePreType" type="addr_type:StreetNameType_type"
  minOccurs="0" maxOccurs="1" />
  <xsd:element name="StreetName" type="addr_type:StreetName_type"
  minOccurs="1" maxOccurs="1" />
  <xsd:element name="StreetNamePostType" type="addr_type:StreetNameType_type"
  minOccurs="0" maxOccurs="1" />
  <xsd:element name="StreetNamePostDirectional"
type="addr_type:StreetNameDirectional_type"
```

```
  minOccurs="0" maxOccurs="1" />
  <xsd:element name="StreetNamePostModifier"
type="addr_type:StreetNameModifier_type"
  minOccurs="0" maxOccurs="1" />
 </xsd:sequence>
 </xsd:group>
 <xsd:group name="CompleteAddressNumber_group">
 <xsd:sequence>
  <xsd:element name="AddressNumberPrefix"
type="addr_type:AddressNumberPrefix_type"
  minOccurs="0" maxOccurs="1" />
  <xsd:element name="AddressNumber" type="addr_type:AddressNumber_type"
  minOccurs="1" maxOccurs="1" />
  <xsd:element name="AddressNumberSuffix"
type="addr_type:AddressNumberSuffix_type"
  minOccurs="0" maxOccurs="1" />
 </xsd:sequence>
 </xsd:group>
 <xsd:complexType name="CompleteAddressNumber_type">
 <xsd:sequence>
  <xsd:element name="AddressNumberPrefix"
type="addr_type:AddressNumberPrefix_type"
  minOccurs="0" maxOccurs="1" />
  <xsd:element name="AddressNumber" type="addr_type:AddressNumber_type"
  minOccurs="1" maxOccurs="1" />
  <xsd:element name="AddressNumberSuffix"
type="addr_type:AddressNumberSuffix_type"
  minOccurs="0" maxOccurs="1" />
 </xsd:sequence>
 <xsd:attribute name="AddressNumberParity"
type="addr_type:AddressNumberParity_type" />
 <xsd:attribute name="AttachedElement" type="addr_type:AttachedElement_type" />
 </xsd:complexType>
 <xsd:complexType name="AddressNumberRange_type">
 <xsd:annotation>
  <xsd:documentation xml:lang="en">
  { Complete Address
  Number (low)*} + { Separator Element
  *} + { Complete Address Number
  (high)*} A set of two
  address numbers, separated by a "Separator",
  representing the low and high numbers of an address
  range. An address
  number range element should be
  accompanied by an Address Range Type
```

```
 Attribute that
 describes the type of range presented in this element.
 </xsd:documentation>
</xsd:annotation>
<xsd:sequence>
 <xsd:element name="CompleteAddressNumber"
type="addr_type:CompleteAddressNumber_type"
 minOccurs="2" maxOccurs="2" />
</xsd:sequence>
<xsd:attribute name="Separator" type="addr_type:Separator_type" />
<xsd:attribute name="Parity" type="addr_type:AddressNumberParity_type" />
<xsd:attribute name="Side" type="addr_type:AddressNumberSide_type" />
</xsd:complexType>
<xsd:complexType name="PlaceName_type">
<xsd:simpleContent>
 <xsd:extension base="xsd:string">
 <xsd:attribute name="PlaceNameType" type="addr_type:PlaceNameType_type" />
 <xsd:attribute name="ElementSequenceNumber"
  type="addr_type:ElementSequenceNumber_type" />
 <xsd:attribute name="GNISFeatureID" type="addr_type:GNISFeatureID_type" />
 </xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
<xsd:complexType name="CompleteSubaddress_type">
<xsd:sequence>
 <xsd:element name="SubaddressElement" type="addr_type:SubaddressElement_type"
 minOccurs="1" maxOccurs="unbounded" />
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="CompleteLandmarkName_type">
<xsd:sequence>
 <xsd:element name="LandmarkName" type="addr_type:LandmarkName_type"
 minOccurs="1" maxOccurs="unbounded" />
</xsd:sequence>
<xsd:attribute name="Separator" type="addr_type:Separator_type" />
</xsd:complexType>
<xsd:complexType name="CompletePlaceName_type">
<xsd:sequence>
 <xsd:element name="PlaceName" type="addr_type:PlaceName_type"
 minOccurs="1" maxOccurs="unbounded" />
</xsd:sequence>
<xsd:attribute name="Separator" type="addr_type:Separator_type" />
</xsd:complexType>
<!-- Supporting Information -->
<xsd:group name="AddressAttributes_group">
```

```
<xsd:annotation>
 <xsd:documentation xml:lang="en">
 Support information and
 record level metadata for each Address
 </xsd:documentation>
</xsd:annotation>
<xsd:sequence>
 <xsd:element name="AddressId" type="addr_type:AddressID_type"
 minOccurs="0" maxOccurs="1" />
 <xsd:element name="AddressAuthority" type="addr_type:AddressAuthority_type"
 minOccurs="0" maxOccurs="1" />
 <xsd:element name="RelatedAddressId" type="addr_type:RelatedAddressID_type"
 minOccurs="0" maxOccurs="unbounded" />
 <xsd:element name="AddressXCoordinate" type="addr_type:AddressXCoordinate_type"
 minOccurs="0" maxOccurs="1" />
 <xsd:element name="AddressYCoordinate" type="addr_type:AddressYCoordinate_type"
 minOccurs="0" maxOccurs="1" />
 <xsd:element name="AddressLongitude" type="addr_type:AddressLongitude_type"
 minOccurs="0" maxOccurs="1" />
 <xsd:element name="AddressLatitude" type="addr_type:AddressLatitude_type"
 minOccurs="0" maxOccurs="1" />
 <xsd:element name="USNationalGridCoordinate"
 type="addr_type:USNationalGridCoordinate_type" minOccurs="0"
 maxOccurs="1" />
 <xsd:element name="AddressElevation" type="addr_type:AddressElevation_type"
 minOccurs="0" maxOccurs="1" />
 <xsd:element name="AddressCoordinateReferenceSystem"
 type="addr_type:AddressCoordinateReferenceSystem_type" minOccurs="0"
 maxOccurs="1" />
 <xsd:element name="AddressParcelIdentifierSource"
 type="addr_type:AddressParcelIdentifierSource_type" minOccurs="0"
 maxOccurs="unbounded" />
 <xsd:element name="AddressParcelIdentifier"
 type="addr_type:AddressParcelIdentifier_type" minOccurs="0"
 maxOccurs="unbounded" />
 <xsd:element name="AddressTransportationSystemName"
 type="addr_type:AddressTransportationSystemName_type" minOccurs="0"
 maxOccurs="1" />
 <xsd:element name="AddressTransportationSystemAuthority"
 type="addr_type:AddressTransportationSystemAuthority_type"
 minOccurs="0" maxOccurs="1" />
 <xsd:element name="AddressTransportationFeatureType"
 type="addr_type:AddressTransportationFeatureType_type" minOccurs="0"
 maxOccurs="1" />
 <xsd:element name="AddressTransportationFeatureID"
```

```
 type="addr_type:AddressTransportationFeatureID_type" minOccurs="0"
 maxOccurs="1" />
 <xsd:element name="RelatedTransportationFeatureID"
 type="addr_type:RelatedTransportationFeatureID_type" minOccurs="0"
 maxOccurs="unbounded" />
 <xsd:element name="AddressRangeType" type="addr_type:AddressRangeType_type"
 minOccurs="0" maxOccurs="2" />
 <xsd:element name="AddressRangeParity" type="addr_type:AddressRangeParity_type"
 minOccurs="0" maxOccurs="2" />
 <xsd:element name="AddressRangeDirectionality"
 type="addr_type:AddressRangeDirectionality_type" minOccurs="0"
 maxOccurs="2" />
 <xsd:element name="AddressRangeSpan" type="addr_type:AddressRangeSpan_type"
 minOccurs="0" maxOccurs="unbounded" />
 <xsd:element name="AddressClassification"
type="addr_type:AddressClassification_type"
 maxOccurs="1" minOccurs="0" />
 <xsd:element name="AddressFeatureType"
type="addr_type:AddressFeatureType_type"
 minOccurs="0" maxOccurs="unbounded" />
 <xsd:element name="AddressLifecycleStatus"
type="addr_type:AddressLifecycleStatus_type"
 minOccurs="0" maxOccurs="1" />
 <xsd:element name="OfficialStatus" type="addr_type:OfficialStatus_type"
 minOccurs="0" maxOccurs="1" />
 <xsd:element name="AddressAnomalyStatus"
type="addr_type:AddressAnomalyStatus_type"
 minOccurs="0" maxOccurs="1" />
 <xsd:element name="AddressSideOfStreet"
type="addr_type:AddressSideOfStreet_type"
 minOccurs="0" maxOccurs="1" />
 <xsd:element name="AddressZLevel" type="addr_type:AddressZLevel_type"
 minOccurs="0" maxOccurs="1" />
 <xsd:element name="LocationDescription" type="addr_type:LocationDescription_type"
 minOccurs="0" maxOccurs="1" />
 <xsd:element name="MailableAddress" type="addr_type:MailableAddress_type"
 minOccurs="0" maxOccurs="1" />
 <xsd:element name="AddressStartDate" type="addr_type:AddressStartDate_type"
 minOccurs="0" maxOccurs="1" />
 <xsd:element name="AddressEndDate" type="addr_type:AddressEndDate_type"
 minOccurs="0" maxOccurs="1" />
 <xsd:element name="DataSetID" type="addr_type:DataSetID_type"
 minOccurs="0" maxOccurs="1" />
 <xsd:element name="AddressReferenceSystemId"
 type="addr_type:AddressReferenceSystemId_type" minOccurs="0"
```

```
  maxOccurs="1" />
  <xsd:element name="AddressReferenceSystemAuthority"
  type="addr_type:AddressReferenceSystemAuthority_type" minOccurs="0"
  maxOccurs="1" />
 </xsd:sequence>
 </xsd:group>
 <!-- End Content Types -->
 <!-- Begin Utility Groups -->
 <xsd:group name="ZipCode_group">
 <xsd:sequence>
  <xsd:element name="ZipCode" type="addr_type:ZipCode_type"
  minOccurs="1" maxOccurs="1" />
  <xsd:element name="ZipPlus4" type="addr_type:ZipPlus4_type"
  minOccurs="0" maxOccurs="1" />
 </xsd:sequence>
 </xsd:group>
 <xsd:group name="PlaceStateZip_group">
 <xsd:choice>
  <xsd:sequence>
  <xsd:element name="CompletePlaceName"
type="addr_type:CompletePlaceName_type"
   minOccurs="1" maxOccurs="1" />
  <xsd:element name="StateName" type="addr_type:StateName_type"
   minOccurs="1" maxOccurs="1" />
  <xsd:group ref="addr_type:ZipCode_group" minOccurs="0"
   maxOccurs="1" />
  <xsd:element name="CountryName" type="addr_type:CountryName_type"
   maxOccurs="1" minOccurs="0" />
  </xsd:sequence>
  <xsd:element name="PlaceStateZip" type="addr_type:PlaceStateZip_type"
  maxOccurs="unbounded" minOccurs="1" />
 </xsd:choice>
 </xsd:group>
 <!-- End Utility Groups -->
</xsd:schema>
```

2. addr_type.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema targetNamespace="addr"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:addr="addr"
 xmlns:addr_type="addr_type" xmlns:gml="http://opengeospatial.net/gml">
 <!--
 Draft Address Standard, version 0.4 being prepared and tested by a
```

```
Working Group coordinated by URISA and NENA and the Census Bureau for
submittal to the FGDC.
-->
<!--
During the initial draft period the rddl can be found at
http://wfs.co.fulton.ga.us/urisa/addr_std/addr.xsd
-->
<xsd:import namespace="addr_type" schemaLocation="addr_type.xsd">
<xsd:annotation>
 <xsd:documentation>
 Base types form the AddressStandard
 </xsd:documentation>
</xsd:annotation>
</xsd:import>
<!-- Begin Base Class Types -->
<!-- Thoroughfare Addresses -->
<xsd:complexType name="NumberedThoroughfareAddress_type">
<xsd:annotation>
 <xsd:documentation xml:lang="en">
 The Address Class
 associated with singular locations
 referenced off of a linear feature,
 having numeric
 identifiers.
 </xsd:documentation>
</xsd:annotation>
<xsd:sequence>
 <xsd:element name="CompleteAddressNumber"
type="addr_type:CompleteAddressNumber_type"
 minOccurs="1" maxOccurs="1" />
 <xsd:element name="CompleteStreetName"
type="addr_type:CompleteStreetName_type"
 minOccurs="1" maxOccurs="1" />
 <xsd:element name="CompleteSubaddress"
 type="addr_type:CompleteSubaddress_type" minOccurs="0"
 maxOccurs="1" />
 <xsd:group ref="addr_type:PlaceStateZip_group" minOccurs="0"
 maxOccurs="unbounded" />
 <xsd:group ref="addr_type:AddressAttributes_group"
 minOccurs="0" maxOccurs="1" />
</xsd:sequence>
<xsd:attribute name="action" type="addr_type:Action_type"
 use="optional" />
</xsd:complexType>
<xsd:group name="IntersectionAddress_StreetName_group" >
```

```
  <xsd:sequence>
  <xsd:element name="SeparatorElement" type="addr_type:Separator_type"
maxOccurs="1" minOccurs="1"/>
  <xsd:element name="CompleteStreetName"
type="addr_type:CompleteStreetName_type" maxOccurs="1" minOccurs="1"/>
  </xsd:sequence>
 </xsd:group>

 <xsd:complexType name="IntersectionAddress_type">
 <xsd:sequence>
  <xsd:element name="CompleteStreetName"
  type="addr_type:CompleteStreetName_type" minOccurs="1"
  maxOccurs="1" />
  <xsd:group ref="addr:IntersectionAddress_StreetName_group" minOccurs="1"
maxOccurs="unbounded"/>
  <xsd:group ref="addr_type:PlaceStateZip_group" minOccurs="1"
  maxOccurs="1" />
  <xsd:group ref="addr_type:AddressAttributes_group"
  minOccurs="0" maxOccurs="1" />
 </xsd:sequence>
 <xsd:attribute name="action" type="addr_type:Action_type"
  use="optional" />
 </xsd:complexType>
 <xsd:complexType name="TwoNumberAddressRange_type">
 <xsd:sequence>
  <xsd:element name="CompleteAddressNumber"
type="addr_type:CompleteAddressNumber_type"
  minOccurs="1" maxOccurs="1" />
   <xsd:element name="SeparatorElement" type="addr_type:Separator_type"
maxOccurs="1" minOccurs="1"/>
  <xsd:element name="CompleteAddressNumber"
type="addr_type:CompleteAddressNumber_type"
  minOccurs="1" maxOccurs="1" />
  <xsd:element name="CompleteStreetName"
type="addr_type:CompleteStreetName_type"
  minOccurs="1" maxOccurs="1" />
  <xsd:group ref="addr_type:PlaceStateZip_group" minOccurs="1"
  maxOccurs="unbounded" />
  <xsd:group ref="addr_type:AddressAttributes_group"
  minOccurs="0" maxOccurs="1" />
 </xsd:sequence>
 <xsd:attribute name="action" type="addr_type:Action_type"
  use="optional" />
 </xsd:complexType>
 <xsd:complexType name="FourNumberAddressRange_type">
```

```
 <xsd:annotation>
  <xsd:documentation> TIGER file ranges (left low, left high, right
  low, right high, street name) are the most widely-used example of
  block ranges Notes: Although they do not necessarily refer to one
  specific site, block addresses are important for municipal
  operations (such as snow plow dispatch), emergency dispatch, and
  geocoding. A block address range may be expressed by four numbers,
  representing the low and high end of the numeric range for each side
  of a block. By convention, the first number represents the low end
  of the range of addresses for the left side, the second number
  represents the high end of the range of addresses for the left side,
  the third number represents the low end of the range of addresses
  for the right side, and the fourth number represents the high end of
  the range for the right side. A block face is defined as one side of
  a thoroughfare between two intersecting street segments. Generally,
  but not always, a block face has addresses of a single parity, that
  is, either odd or even numbers. However, mixed parities do occur in
  some places. In other cases, where the numeric ranges on opposite
  sides of the same block are not within the same general range, it is
  preferable to express the range in terms of the left low-high, right
  low-high, or to provide individual block face ranges. A block range
  may refer to either a theoretical range (the possible range of
  addresses along that street segment) or to an actual or used range
  of addresses. These types (actual or theoretical) are distinguished
  by the range type attribute.
    </xsd:documentation>
 </xsd:annotation>
 <xsd:sequence>
  <xsd:element name="CompleteAddressNumber"
type="addr_type:CompleteAddressNumber_type"
  minOccurs="1" maxOccurs="1" />
   <xsd:element name="SeparatorElement" type="addr_type:Separator_type"
maxOccurs="1" minOccurs="1"/>
  <xsd:element name="CompleteAddressNumber"
type="addr_type:CompleteAddressNumber_type"
  minOccurs="1" maxOccurs="1" />
  <xsd:element name="CompleteAddressNumber"
type="addr_type:CompleteAddressNumber_type"
  minOccurs="1" maxOccurs="1" />
   <xsd:element name="SeparatorElement" type="addr_type:Separator_type"
maxOccurs="1" minOccurs="1"/>
  <xsd:element name="CompleteAddressNumber"
type="addr_type:CompleteAddressNumber_type"
  minOccurs="1" maxOccurs="1" />
  <xsd:element name="CompleteStreetName"
```

```
type="addr_type:CompleteStreetName_type"
  minOccurs="1" maxOccurs="1" />
  <xsd:group ref="addr_type:PlaceStateZip_group" minOccurs="1"
  maxOccurs="1" />
  <xsd:group ref="addr_type:AddressAttributes_group"
  minOccurs="0" maxOccurs="1" />
 </xsd:sequence>
 <xsd:attribute name="action" type="addr_type:Action_type"
  use="optional" />
 </xsd:complexType>
 <xsd:complexType name="UnnumberedThoroughfareAddress_type">
 <xsd:annotation>
  <xsd:documentation xml:lang="en">
  The Address Class
  associated with singular locations
  referenced off of a linear feature,
  lacking numeric
  identifiers.
  </xsd:documentation>
 </xsd:annotation>
 <xsd:sequence>
  <xsd:element name="CompleteStreetName"
type="addr_type:CompleteStreetName_type"
  minOccurs="1" maxOccurs="1" />
  <xsd:element name="CompleteSubaddress"
type="addr_type:CompleteSubaddress_type"
  minOccurs="0" maxOccurs="1" />
  <xsd:group ref="addr_type:PlaceStateZip_group" minOccurs="1"
  maxOccurs="1" />
  <xsd:group ref="addr_type:AddressAttributes_group"
  minOccurs="0" maxOccurs="1" />
 </xsd:sequence>
 <xsd:attribute name="action" type="addr_type:Action_type"
  use="optional" />
 </xsd:complexType>
 <!-- Landmark Address Classes -->
 <xsd:complexType name="LandmarkAddress_type">
 <xsd:sequence>
  <xsd:element name="CompleteLandmarkName"
type="addr_type:CompleteLandmarkName_type"
  minOccurs="1" maxOccurs="1" />
  <xsd:element name="CompleteSubaddress"
type="addr_type:CompleteSubaddress_type"
  minOccurs="0" maxOccurs="1" />
  <xsd:group ref="addr_type:PlaceStateZip_group" minOccurs="1"
```

```
 maxOccurs="1" />
 <xsd:group ref="addr_type:AddressAttributes_group"
 minOccurs="0" maxOccurs="1" />
</xsd:sequence>
<xsd:attribute name="action" type="addr_type:Action_type"
 use="optional" />
</xsd:complexType>
<xsd:complexType name="CommunityAddress_type">
<xsd:annotation>
 <xsd:documentation> 1. Community Addresses are commonly used for
 housing projects, Puerto Rican urbanizations, trailer courts, and
 similar developments that are built around unnamed interior walkways
 or roadways. Their Complete Address Numbers refer to the community
 name, not to a thoroughfare. 2. A Community Address includes a
 Complete Address Number, a community name, and a Place Name. The
 address does not include a Complete Street Name. The community name
 might be a treated as a Landmark Name or Place Name--the distinction
 is often arbitrary or unclear for community names. 3. If there is no
 Complete Address Number preceding the urbanization name, the address
 fits into the Landmark Address class. 4. If the address includes
 both a Complete Street Name and a community name, it fits in the
 Landmark Site Address class. 5. This class includes Puerto Rican
 urbanization addresses where the urbanization name is preceded by a
 number, and no street name is included. In Puerto Rico, an
 urbanization denotes an area, sector, or residential development
 within a geographic area. For more information on Puerto Rican
 addressing conventions, see USPS Publication 28 Section 29, and USPS
  “Addressing Standards for Puerto Rico and the Virgin Islands” .
 </xsd:documentation>
</xsd:annotation>
<xsd:sequence>
 <xsd:element name="CompleteAddressNumber"
type="addr_type:CompleteAddressNumber_type"
 minOccurs="1" maxOccurs="1" />
 <xsd:choice>
 <xsd:element name="CompleteLandmarkName"
type="addr_type:CompleteLandmarkName_type"
  minOccurs="1" maxOccurs="1" />
 <xsd:element name="CompletePlaceName"
type="addr_type:CompletePlaceName_type"
  minOccurs="1" maxOccurs="1" />
 </xsd:choice>
 <xsd:element name="CompleteSubaddress"
type="addr_type:CompleteSubaddress_type"
 minOccurs="0" maxOccurs="1" />
```

```
 <xsd:group ref="addr_type:PlaceStateZip_group" minOccurs="1"
 maxOccurs="1" />
 <xsd:group ref="addr_type:AddressAttributes_group"
 minOccurs="0" maxOccurs="1" />
</xsd:sequence>
<xsd:attribute name="action" type="addr_type:Action_type"
 use="optional" />
</xsd:complexType>
<!-- Postal Delivery Address Classes -->
<xsd:complexType name="USPSPostalDeliveryBox_type">
<xsd:sequence>
 <xsd:element name="USPSBox" type="addr_type:USPSBox_type"
 minOccurs="1" maxOccurs="1" />
 <xsd:element name="CompleteSubaddress"
type="addr_type:CompleteSubaddress_type"
 minOccurs="0" maxOccurs="1" />
 <xsd:group ref="addr_type:PlaceStateZip_group" minOccurs="1"
 maxOccurs="1" />
 <xsd:group ref="addr_type:AddressAttributes_group"
 minOccurs="0" maxOccurs="1" />
</xsd:sequence>
<xsd:attribute name="action" type="addr_type:Action_type"
 use="optional" />
</xsd:complexType>
<xsd:complexType name="USPSPostalDeliveryRoute_type">
<xsd:sequence>
 <xsd:element name="USPSAddress" type="addr_type:USPSAddress_type"
 minOccurs="1" maxOccurs="1" />
 <xsd:group ref="addr_type:PlaceStateZip_group" minOccurs="1"
 maxOccurs="1" />
 <xsd:group ref="addr_type:AddressAttributes_group"
 minOccurs="0" maxOccurs="1" />
</xsd:sequence>
<xsd:attribute name="action" type="addr_type:Action_type"
 use="optional" />
</xsd:complexType>
<xsd:complexType name="USPSGeneralDeliveryOffice_type">
<xsd:sequence>
 <xsd:element name="USPSGeneralDeliveryPoint"
 type="addr_type:USPSGeneralDeliveryPoint_type" />
 <xsd:group ref="addr_type:PlaceStateZip_group" minOccurs="1"
 maxOccurs="1" />
 <xsd:group ref="addr_type:AddressAttributes_group"
 minOccurs="0" maxOccurs="1" />
</xsd:sequence>
```

```
<xsd:attribute name="action" type="addr_type:Action_type"
 use="optional" />
</xsd:complexType>

<xsd:complexType name="GeneralAddressClass_type">
<xsd:choice>
 <xsd:element name="GeneralAddress"
 type="addr_type:GeneralAddress_type" />
 <xsd:sequence>
 <xsd:element name="USPSGeneralDeliveryPoint"
  type="addr_type:USPSGeneralDeliveryPoint_type" />
 <xsd:group ref="addr_type:PlaceStateZip_group"
  minOccurs="1" maxOccurs="1" />
 <xsd:group ref="addr_type:AddressAttributes_group"
  minOccurs="0" maxOccurs="1" />
 </xsd:sequence>
</xsd:choice>
<xsd:attribute name="action" type="addr_type:Action_type" />
</xsd:complexType>
<xsd:group name="AddressCollection_group">
<xsd:annotation>
 <xsd:documentation>
 The Single Choice Union of all Address Types
 </xsd:documentation>
</xsd:annotation>
<xsd:choice>
 <xsd:element name="NumberedThoroughfareAddress"
type="addr:NumberedThoroughfareAddress_type"
 minOccurs="0" maxOccurs="unbounded" />
 <xsd:element name="IntersectionAddress" type="addr:IntersectionAddress_type"
 minOccurs="0" maxOccurs="unbounded" />
 <xsd:element name="TwoNumberAddressRange"
type="addr:TwoNumberAddressRange_type"
 minOccurs="0" maxOccurs="unbounded" />
 <xsd:element name="FourNumberAddressRange"
type="addr:FourNumberAddressRange_type"
 minOccurs="0" maxOccurs="unbounded" />
 <xsd:element name="UnnumberedThoroughfareAddress"
type="addr:UnnumberedThoroughfareAddress_type"
 minOccurs="0" maxOccurs="unbounded" />
 <xsd:element name="LandmarkAddress" type="addr:LandmarkAddress_type"
 minOccurs="0" maxOccurs="unbounded" />
 <xsd:element name="CommunityAddress" type="addr:CommunityAddress_type"
 minOccurs="0" maxOccurs="unbounded" />
```

```
  <xsd:element name="USPSPostalDeliveryBox"
type="addr:USPSPostalDeliveryBox_type"
  minOccurs="0" maxOccurs="unbounded" />
  <xsd:element name="USPSPostalDeliveryRoute"
type="addr:USPSPostalDeliveryRoute_type"
  minOccurs="0" maxOccurs="unbounded" />
  <xsd:element name="USPSGeneralDeliveryOffice"
type="addr:USPSGeneralDeliveryOffice_type"
  minOccurs="0" maxOccurs="unbounded" />

  <xsd:element name="GeneralAddressClass" type="addr:GeneralAddressClass_type"
  minOccurs="0" maxOccurs="unbounded" />
  <xsd:element name="AddressReferenceSystem"
type="addr_type:AddressReferenceSystem_type"
  minOccurs="0" maxOccurs="unbounded" />
 </xsd:choice>
 </xsd:group>
 <!-- End Complex Types -->
 <!-- Wrapper collecting a set of addresses -->
 <xsd:element name="AddressCollection">
 <xsd:complexType mixed="false">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
  <xsd:group ref="addr:AddressCollection_group" />
  </xsd:choice>
  <xsd:attribute name="version" type="addr_type:version_type"
  use="required" />
 </xsd:complexType>
 </xsd:element>

</xsd:schema>
```

7.2 Appendix B: Address XML Examples

Address exchange packages can be simple, complex and anywhere in between. For clarity each of the Address Classes are shown here in a complete exchange packaged for reference and review.

Thoroughfare Address Classes

Numberered Thoroughfare Address

```
<?xml version="1.0" encoding="UTF-8"?>
<addr:AddressCollection version="0.4" xmlns:addr="addr" xmlns:addr_type="addr_type"
xmlns:gml="http://www.opengis.net/gml"
xmlns:smil20="http://www.w3.org/2001/SMIL20/"
xmlns:smil20lang="http://www.w3.org/2001/SMIL20/Language"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xml="http://www.w3.org/XML/1998/namespace"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="addr
addr.xsd ">
 <NumberedThoroughfareAddress>
 <CompleteAddressNumber>
  <AddressNumber>123</AddressNumber>
 </CompleteAddressNumber>
 <CompleteStreetName>
  <StreetName>Main</StreetName>
  <StreetNamePostType>Street</StreetNamePostType>
 </CompleteStreetName>
 <CompletePlaceName>
  <PlaceName>Buffalo Lake</PlaceName>
 </CompletePlaceName>
 <StateName>MN</StateName>
 <ZipCode>55314</ZipCode>
 </NumberedThoroughfareAddress>
</addr:AddressCollection>
```

Intersection Address

```
<?xml version="1.0" encoding="UTF-8"?>
<addr:AddressCollection version="0.4" xmlns:addr="addr" xmlns:addr_type="addr_type"
xmlns:gml="http://www.opengis.net/gml"
xmlns:smil20="http://www.w3.org/2001/SMIL20/"
```

```
xmlns:smil20lang="http://www.w3.org/2001/SMIL20/Language"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xml="http://www.w3.org/XML/1998/namespace"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="addr
addr.xsd ">
 <IntersectionAddress>
 <CompleteStreetName>
  <StreetName>Boardwalk</StreetName>
 </CompleteStreetName>
 <SeparatorElement>and</SeparatorElement>
 <CompleteStreetName>
  <StreetName>Park</StreetName>
  <StreetNamePostType>Place</StreetNamePostType>
 </CompleteStreetName>
 <CompletePlaceName>
  <PlaceName>Atlantic City</PlaceName>
 </CompletePlaceName>
 <StateName>NJ</StateName>
 </IntersectionAddress>
</addr:AddressCollection>
```


Two Number Address Range


```
<?xml version="1.0" encoding="UTF-8"?>
<addr:AddressCollection version="0.4" xmlns:addr="addr" xmlns:addr_type="addr_type"
xmlns:gml="http://www.opengis.net/gml"
xmlns:smil20="http://www.w3.org/2001/SMIL20/"
xmlns:smil20lang="http://www.w3.org/2001/SMIL20/Language"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xml="http://www.w3.org/XML/1998/namespace"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="addr
addr.xsd ">
 <TwoNumberAddressRange>
 <CompleteAddressNumber>
  <AddressNumber>401</AddressNumber>
 </CompleteAddressNumber>
 <SeparatorElement>-</SeparatorElement>
 <CompleteAddressNumber>
  <AddressNumber>418</AddressNumber>
 </CompleteAddressNumber>
 <CompleteStreetName>
  <StreetName>Green</StreetName>
  <StreetNamePostType>Street</StreetNamePostType>
 </CompleteStreetName>
```

```
 <CompletePlaceName>
  <PlaceName>Flint</PlaceName>
 </CompletePlaceName>
 <StateName>MI</StateName>
 <ZipCode>48503</ZipCode>
 </TwoNumberAddressRange>
</addr:AddressCollection>
```

Four Number Address Range

```
<?xml version="1.0" encoding="UTF-8"?>
<addr:AddressCollection version="0.4" xmlns:addr="addr" xmlns:addr_type="addr_type"
xmlns:gml="http://www.opengis.net/gml"
xmlns:smil20="http://www.w3.org/2001/SMIL20/"
xmlns:smil20lang="http://www.w3.org/2001/SMIL20/Language"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xml="http://www.w3.org/XML/1998/namespace"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="addr
addr.xsd ">
 <FourNumberAddressRange>
 <CompleteAddressNumber>
  <AddressNumber>1900</AddressNumber>
 </CompleteAddressNumber>
 <SeparatorElement>-</SeparatorElement>
 <CompleteAddressNumber>
  <AddressNumber>1908</AddressNumber>
 </CompleteAddressNumber>
 <CompleteAddressNumber>
  <AddressNumber>1901</AddressNumber>
 </CompleteAddressNumber>
 <SeparatorElement>-</SeparatorElement>
 <CompleteAddressNumber>
  <AddressNumber>1909</AddressNumber>
 </CompleteAddressNumber>
 <CompleteStreetName>
  <StreetName>Bear</StreetName>
  <StreetNamePostType>court</StreetNamePostType>
 </CompleteStreetName>
 <CompletePlaceName>
  <PlaceName>Fort Collins</PlaceName>
 </CompletePlaceName>
 <StateName>CO</StateName>
 <ZipCode>80525</ZipCode>
 </FourNumberAddressRange>
```

```
</addr:AddressCollection>
```


Unnumbered Thoroughfare Address


```
<?xml version="1.0" encoding="UTF-8"?>
<addr:AddressCollection version="0.4" xmlns:addr="addr" xmlns:addr_type="addr_type"
xmlns:gml="http://www.opengis.net/gml"
xmlns:smil20="http://www.w3.org/2001/SMIL20/"
xmlns:smil20lang="http://www.w3.org/2001/SMIL20/Language"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xml="http://www.w3.org/XML/1998/namespace"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="addr
addr.xsd ">
 <UnnumberedThoroughfareAddress>
 <CompleteStreetName>
  <StreetName>Fagaima</StreetName>
  <StreetNamePostType>Road</StreetNamePostType>
 </CompleteStreetName>
 <CompletePlaceName>
  <PlaceName>Nu'uli</PlaceName>
 </CompletePlaceName>
 <StateName>AS</StateName>
 <ZipCode>96799</ZipCode>
 </UnnumberedThoroughfareAddress>
</addr:AddressCollection>
```


Landmark Address Classes

Landmark Address


```
<?xml version="1.0" encoding="UTF-8"?>
<addr:AddressCollection version="0.4" xmlns:addr="addr" xmlns:addr_type="addr_type"
xmlns:gml="http://www.opengis.net/gml"
xmlns:smil20="http://www.w3.org/2001/SMIL20/"
xmlns:smil20lang="http://www.w3.org/2001/SMIL20/Language"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xml="http://www.w3.org/XML/1998/namespace"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="addr
addr.xsd ">
 <LandmarkAddress>
 <CompleteLandmarkName>
  <LandmarkName>Condominium Garden Hills Plaza</LandmarkName>
 </CompleteLandmarkName>
```

```
 <CompleteSubaddress>
  <SubaddressElement SubaddressComponentOrder="1">
  <SubaddressType>Torre</SubaddressType>
  <SubaddressIdentifier>2</SubaddressIdentifier>
  </SubaddressElement>
  <SubaddressElement>
  <SubaddressType>Apartamento</SubaddressType>
  <SubaddressIdentifier>905</SubaddressIdentifier>
  </SubaddressElement>
 </CompleteSubaddress>
 <CompletePlaceName>
  <PlaceName>Mayaguez</PlaceName>
 </CompletePlaceName>
 <StateName>PR</StateName>
 <ZipCode>00608</ZipCode>
 <ZipPlus4>1233</ZipPlus4>
 </LandmarkAddress>
</addr:AddressCollection>
```

Community Address

```
<?xml version="1.0" encoding="UTF-8"?>
<addr:AddressCollection version="0.4" xmlns:addr="addr" xmlns:addr_type="addr_type"
xmlns:gml="http://www.opengis.net/gml"
xmlns:smil20="http://www.w3.org/2001/SMIL20/"
xmlns:smil20lang="http://www.w3.org/2001/SMIL20/Language"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xml="http://www.w3.org/XML/1998/namespace"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="addr
addr.xsd ">
 <CommunityAddress>
 <CompleteAddressNumber>
  <AddressNumberPrefix>A</AddressNumberPrefix>
  <AddressNumber>17</AddressNumber>
 </CompleteAddressNumber>
 <CompleteLandmarkName>
  <LandmarkName>Jardine Fagota</LandmarkName>
 </CompleteLandmarkName>
 <CompletePlaceName>
  <PlaceName>Ponce</PlaceName>
 </CompletePlaceName>
 <StateName>PR</StateName>
 <ZipCode>00731</ZipCode>
 </CommunityAddress>
```

```
</addr:AddressCollection>
```

Postal Delivery Address Classes

USPS Postal Delivery Box

```
<?xml version="1.0" encoding="UTF-8"?>
<addr:AddressCollection version="0.4" xmlns:addr="addr" xmlns:addr_type="addr_type"
xmlns:gml="http://www.opengis.net/gml"
xmlns:smil20="http://www.w3.org/2001/SMIL20/"
xmlns:smil20lang="http://www.w3.org/2001/SMIL20/Language"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xml="http://www.w3.org/XML/1998/namespace"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="addr
addr.xsd ">
 <USPSPostalDeliveryBox>
 <USPSBox>
  <USPSBoxType>PO BOX</USPSBoxType>
  <USPSBoxId>159753</USPSBoxId>
 </USPSBox>
 <CompleteSubaddress>
  <SubaddressElement>
  <SubaddressType>PMB</SubaddressType>
  <SubaddressIdentifier>3571</SubaddressIdentifier>
  </SubaddressElement>
 </CompleteSubaddress>
 <CompletePlaceName>
  <PlaceName>Herndon</PlaceName>
 </CompletePlaceName>
 <StateName>VA</StateName>
 <ZipCode>22071</ZipCode>
 </USPSPostalDeliveryBox>
</addr:AddressCollection>
```

USPS Postal Delivery Route

```
<?xml version="1.0" encoding="UTF-8"?>
<addr:AddressCollection version="0.4" xmlns:addr="addr" xmlns:addr_type="addr_type"
xmlns:gml="http://www.opengis.net/gml"
xmlns:smil20="http://www.w3.org/2001/SMIL20/"
xmlns:smil20lang="http://www.w3.org/2001/SMIL20/Language"
xmlns:xlink="http://www.w3.org/1999/xlink"
```

```
xmlns:xml="http://www.w3.org/XML/1998/namespace"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="addr
addr.xsd ">
 <USPSPostalDeliveryRoute>
 <USPSAddress>
  <USPSRoute>
  <USPSBoxGroupType>RR</USPSBoxGroupType>
  <USPSBoxGroupId>2</USPSBoxGroupId>
  </USPSRoute>
  <USPSBox>
  <USPSBoxType>Box</USPSBoxType>
  <USPSBoxId>18</USPSBoxId>
  </USPSBox>
 </USPSAddress>
 <CompletePlaceName>
  <PlaceName>Largo</PlaceName>
 </CompletePlaceName>
 <StateName>FL</StateName>
 <ZipCode>33777</ZipCode>
 </USPSPostalDeliveryRoute>
</addr:AddressCollection>
```


USPS General Delivery Office


```
<?xml version="1.0" encoding="UTF-8"?>
<addr:AddressCollection version="0.4" xmlns:addr="addr" xmlns:addr_type="addr_type"
xmlns:gml="http://www.opengis.net/gml"
xmlns:smil20="http://www.w3.org/2001/SMIL20/"
xmlns:smil20lang="http://www.w3.org/2001/SMIL20/Language"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xml="http://www.w3.org/XML/1998/namespace"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="addr
addr.xsd ">
 <USPSGeneralDeliveryOffice>
 <USPSGeneralDeliveryPoint>General Delivery</USPSGeneralDeliveryPoint>
 <CompletePlaceName>
  <PlaceName>Tampa</PlaceName>
 </CompletePlaceName>
 <StateName>FL</StateName>
 <ZipCode>33602</ZipCode>
 <ZipPlus4>9999</ZipPlus4>
 </USPSGeneralDeliveryOffice>
</addr:AddressCollection>
```

General Address Class

General Address Type 1

```
<?xml version="1.0" encoding="UTF-8"?>
<addr:AddressCollection version="0.4" xmlns:addr="addr" xmlns:addr_type="addr_type"
xmlns:gml="http://www.opengis.net/gml"
xmlns:smil20="http://www.w3.org/2001/SMIL20/"
xmlns:smil20lang="http://www.w3.org/2001/SMIL20/Language"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xml="http://www.w3.org/XML/1998/namespace"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="addr
addr.xsd ">
 <GeneralAddressClass>123 Main Street, Apt 1, Ames, IA 50010</GeneralAddressClass>
</addr:AddressCollection>
```

General Address Type 2

```
<?xml version="1.0" encoding="UTF-8"?>
<addr:AddressCollection version="0.4" xmlns:addr="addr" xmlns:addr_type="addr_type"
xmlns:gml="http://www.opengis.net/gml"
xmlns:smil20="http://www.w3.org/2001/SMIL20/"
xmlns:smil20lang="http://www.w3.org/2001/SMIL20/Language"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xml="http://www.w3.org/XML/1998/namespace"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="addr
addr.xsd ">
 <GeneralAddressClass>
  <DeliveryAddress>123 Main Street, Apt 1</DeliveryAddress>
 </GeneralAddressClass>
</addr:AddressCollection>
```

General Address Type 3

```
<?xml version="1.0" encoding="UTF-8"?>
<addr:AddressCollection version="0.4" xmlns:addr="addr" xmlns:addr_type="addr_type"
xmlns:gml="http://www.opengis.net/gml"
xmlns:smil20="http://www.w3.org/2001/SMIL20/"
xmlns:smil20lang="http://www.w3.org/2001/SMIL20/Language"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xml="http://www.w3.org/XML/1998/namespace"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="addr
addr.xsd ">
```

```
<GeneralAddressClass>
 <DeliveryAddress>123 Main Street, Apt 1</DeliveryAddress>
  <CompletePlaceName>
<PlaceName>Ames</PlaceName>
</CompletePlaceName>
<StateName>IA</StateName>
<ZipCode>50010</ZipCode>
</GeneralAddressClass>
</addr:AddressCollection>
```

7.3 Appendix C (Informative): Table of Element Relationships

Note: The elements listed as part of or contained in other elements may be either mandatory, conditional, or optional. Please see the individual element definitions to determine the parameters for each element.

Element Name	**Element Type**	**Complex Elements This Element is Part Of**	**Element Contains These Simple Elements**
Address Number Prefix	Simple	Complete Address Number	
Address Number	Simple	Complete Address Numbers	
Address Number Suffix	Simple	Complete Address Number	
Separator Element	Simple	Complete Address Number	
Complete Address Number	Complex	Delivery Address	Address Number Prefix Address Number Address Number Suffix Separator Element
Street Name Pre Modifier	Simple	Complete Street Name Delivery Address	
Street Name Pre Directional	Simple	Complete Street Name Delivery Address	
Street Name Pre Type	Simple	Complete Street Name Delivery Address	
Street Name	Simple	Complete Street Name Delivery Address	
Street Name Post Type	Simple	Complete Street Name Delivery Address	
Street Name Post Directional	Simple	Complete Street Name Delivery Address	
Street Name Post Modifier	Simple	Complete Street Name Delivery Address	
Complete Street Name	Complex	Delivery Address	Street Name Pre Modifier Street Name Pre Directional Street Name Pre Type

			Street Name Street Name Post Type Street Name Post Directional Street Name Post Modifier
Subaddress Type	Simple	Subaddress Element	
Subaddress Identifier	Simple	Subaddress Element	
Subaddress Element	Complex	Complete Subaddress	Subaddress Type SubaddressIdentifier
Complete Subaddress	Complex	Delivery Address	Subaddress Element Subaddress Type Subaddress Identifier
Place Name	Simple	Place State ZIP	
State Name	Simple	Place State ZIP	
Zip Code	Simple	Place State ZIP	
Zip Plus 4	Simple	Place State ZIP	
Country Name	Simple		
USPS Box Type	Simple	Delivery Address	
USPS Box ID	Simple	Delivery Address	
USPS Box Group Type	Simple	Delivery Address	
USPS Box Group ID	Simple	Delivery Address	
Delivery Address	Complex		Complete Address Number Complete Street Name Complete Subaddress
Place State ZIP	Complex		Place Name State Name ZIP Code ZIP Plus 4

7.4 Appendix D (Informative): Relationship of Addresses to Transportation Features and Linear Reference Locations

7.4.1 Introduction

Appendix B presents the relationship between the Address Standard and the transportation part of the Framework Standard in three sections:

- Section 2 sets forth the relation between the addresses and transportation networks, and differentiates the scopes of the address standard and the transportation standard.
- Section 3 lists key transportation features defined in the framework standard transportation base part, and states how address classes are related to transportation features.
- Section 4 summarizes (from Annex B of the Framework Standard Transportation Base Part) the definition of linear reference systems and their components, and shows how addresses can be expressed as linear reference locations.

7.4.2 Address Systems and Transportation Networks

Addresses are a means by which people specify the location of travel origins and destinations and relate them to the transportation network. Most addresses specify locations for structures, land parcels, incidents, infrastructure components such as poles or hydrants, None of these features are transportation segments or nodes. By relating non-transportation features to the transportation network, thoroughfare addresses enable people to locate the address using the transportation network and travel to it along network paths. The Address Standard provides the data elements and structures—most of them non-geometric—needed to relate people's

specific travel origins and destinations to the transportation network. The address standard also defines certain elements needed within the transportation standard to describe transportation features, most notably address numbers, address ranges, and street names.

The Transportation Part of the framework standard defines the geometric elements and structures needed to construct transportation networks, and the non-geometric attributes needed to describe them. Transportation networks show the paths of travel from origin to destination.

Transportation networks model the thoroughfares that thoroughfare addresses refer to, the particular thoroughfare segments by which individual addresses may be grouped into address ranges, the nodes that define intersections, and the left/right side by which odd/even parities are located. Numbered Thoroughfare Addresses and some ranges are typically modeled as point events (or occasionally linear events) located along the thoroughfare segments. Intersection Addresses and most ranges correspond to nodes and segments respectively. Thus the Framework Standard Transportation Part provides the geometric elements and structures needed to relate addresses to their corresponding transportation system segments and nodes.

The Address Standard and the Transportation Part are so closely related as to be interdependent. The following principles differentiate their scopes so as to be complementary and mutually exclusive:

1. The Address Standard defines the address classes, elements and attributes, none of which are network elements and almost all of which are non-geometric, and the Transportation Part incorporates them by reference.

2. The Address Standard provides for the description of Address Reference Systems, containing the rules for address assignment, and forming a basis for validation and quality testing of addresses. The elements of an Address Reference system include geometric components including Address Reference System Extent, Address Reference System Axis, Address Reference System Axis Point Of Beginning, Address Reference System Reference Polyline, Address Reference System Range Breakpoint, Address Reference System Range Breakline. and Address Reference System Range Polygon. These geometric elements can be related to the transportation elements as nodes, segments, point events, and linear events.
3. The Transportation Part defines the geometric structures and elements needed to comprise thoroughfare networks, and the address standard incorporates them by reference. They include transportation networks, nodes, and segments; point events and linear events.

7.4.3 Addresses And Transportation Features

7.4.3.1 Key Transportation Feature Definitions

The Transportation Part is Part 7 of the Framework Data Content Standard. It is comprised of five sub-parts: the Transportation Base part (Part 7), and five specialized subparts: Rail, Roads, Transit, and Inland Waterways (Parts 7b through 7e). (Part 7a, Transportation - Air, was drafted but not endorsed.)

The Base part (Part 7, section 5) defines several terms needed to articulate the relationship between addresses and transportation features:

- transportation system - *"set of components that allow movement of goods and people between locations"* (sec. 5.25)
- event - *"mechanism for locating an attribute value or feature along a transportation feature."* (sec. 5.4)
- Point event - *"event that occurs at a single position along a linear feature."* (sec. 5.12)
- Linear event - *"event that occurs for an interval along the length of a feature."* (sec. 5.8)
- transportation point (TranPoint) - *"topological connection between transportation segments."* (sec. 5.22)
- transportation segment (TranSeg) - *"linear section of the physical transportation network."* (sec. 5.23)
- transportation path (TranPath) - *"ordered list of whole or partial...transportation segments."* (sec. 5.21)
- transportation segmentation model - *"set of transportation features (TranPath, TranPoint, and TranSeg) and their topological relationships which together define all possible movements through the transportation system"* (sec. 5.24)
- transportation feature (TranFeature) - *"representation of transportation entities that include transportation segmentation model features, as well as other features relevant to transportation"* (sec. 5.20)

7.4.3.2 Representing Addresses As Transportation Features

An address can be represented within a transportation network (e.g. a road centerline model) in various ways, depending on the class of the address and how it is mapped. This subsection

gives the transportation feature types that can be used to represent each address class. The feature types are defined and explained in the FGDC's "Framework Data Content Standard Part 7: Transportation." See in particular "Transportation Base," Sections 5 (Terms and Definitions) and 7 (Requirements), and "Part 7c: Roads."

7.4.3.2.1 Representation of a Numbered Thoroughfare Address as a Transportation Feature

- (If the address is mapped as a point): Point event, related to one or more transportation segments.
- (If the address is mapped as a line or polygon): Linear event, related to one or more transportation segments.

7.4.3.2.2 Representation of an Intersection Address as a Transportation Feature

- One or more transportation points (TranPoints).
- Note that for complex intersections, or where roads are represented as two or more centerlines, one Intersection Address may be represented by multiple TranPoints.

7.4.3.2.3 Representation of a Two Number Address Range as a Transportation Feature

- (If the range covers part of one transportation segment): Linear event, related to a transportation segment (TranSeg).
- (If the range covers one complete transportation segment): Transportation segment (TranSeg).

- (If the range covers more than one complete transportation segment): Transportation path (TranPath).

7.4.3.2.4 Representation of a Four Number Address Range as a Transportation Feature

- (If the range covers part of one transportation segment): Linear event, related to a transportation segment (TranSeg).
- (If the range covers one complete transportation segment): Transportation segment (TranSeg).
- (If the range covers more than one complete transportation segment): Transportation path (TranPath).

7.4.3.2.5 Representation of an Unnumbered Thoroughfare Address as a Transportation Feature

- (If the thoroughfare has only one segment): Transportation segment (TranSeg)
- (If the thoroughfare has more than one segment): Transportation path (TranPath)

7.4.3.2.6 Representation of a Landmark Address as a Transportation Feature

Cannot be specified within this standard. Addresses of this class have no defined relation to a transportation data model. A Landmark Address might be mapped as a point or a line or a polygon, and if represented as a polygon it might relate to zero or one or many transportation points or segments or paths.

7.4.3.2.7 Representation of a Community Address as a Transportation Feature

Cannot be specified within this standard. Addresses of this class have no defined relation to a transportation data model. A Community Address might be mapped as a point or a line or a polygon, and it might relate to zero or one or many transportation points or segments or paths.

7.4.3.2.8 Representation of a USPS Postal Delivery Box as a Transportation Feature

7.4.3.2.9 Representation of a USPS Postal Delivery Route as a Transportation Feature

USPS Postal Delivery Route addresses have no definite relation to any transportation feature. Within the the US, if the location of the delivery points are known, then Rural route and HC route addresses could be mapped as points, treated as point events, and related to a transportation segment. Overseas military addresses have no relation to any transportation feature.

7.4.3.2.10 Representation of a USPS General Delivery Office as a Transportation Feature

- A USPS General Delivery Office could be mapped to a post office, or it could be said to have no relation to any transportation feature.
- Overseas military addresses have no relation to any transportation feature.

7.4.4. Expressing Address Locations as Linear Reference Positions

7.4.4.1 Linear Reference Systems and Addresses.

Linear reference systems specify locations by reference to distance travelled along a route within a transportation network. Linear reference systems differ fundamentally from address reference systems and coordinate reference systems, and thus offer a third way to specify address locations. Linear reference systems are used primarily in surveying and engineering, but they are also useful in address administration. Linear referencing explicitly ties an address to a specific position along its corresponding street segment. Linear reference systems are useful in visualizing address lists and building address zone information when side of street matters. Linear referencing can therefore be useful in detecting mislocated thoroughfare addresses (out of sequence or wrong parity) and erroneous ranges.

Transportation Base Part (Annex B) provides normative classes and types needed to define linear reference systems and specify positions along curvilinear transportation features. Annex B, Section 5, defines several terms of interest:

- Position expression - "expression used to describe a position using linear referencing and comprised of a measured value (distance expression), the curvilinear element being measured, (linear element), the method of measurement (LRM), and an optional lateral offset (offset expression)."
- Distance expression - "linear distance measured along a linear element (a component of a position expression)."
- Linear element - "underlying curvilinear element along which a linearly referenced measure is taken."

- Offset - "Optional part of a linearly referenced position expression which specifies the lateral distance left or right of the linear element being measured."

7.4.4.2 Linear Referencing Locations and the Address Standard.

Linear reference locations must be specified by reference to a transportation network as defined in the Transportation Part of the Framework Standard. The Transportation Part defines all the elements needed to construct the network and represent addresses within it (typically as point events). The Transportation Part also defines all the elements needed to establish and document linear reference locations, such as route, point of beginning, units of measure, method of measuring along the route, etc. Because linear reference locations can be constructed entirely within the domain of the Transportation Part of the framework standard, no linear reference attributes are provided or needed within the Address Standard.

7.5 Appendix E (Informative): Element Measure Index

Element Name	Component or Subject	Simple or Complex	Measure
Address Number	Address Number	Simple	Data Type Measure
Address Number	Address Number	Simple	Spatial Domain Measure +
Address Number	Address Number	Simple	Tabular Domain Measure
Address Number	Address Number	Simple	Address Number Fishbones Measure +
Address Number Prefix	Address Number	Simple	Range Domain Measure
Address Number Prefix	Address Number	Simple	Spatial Domain Measure +
Address Number Prefix	Address Number	Simple	Tabular Domain Measure
Address Number Prefix	Address Number	Simple	Address Number Fishbones Measure +
Address Number Suffix	Address Number	Simple	Spatial Domain Measure +
Address Number Suffix	Address Number	Simple	Tabular Domain Measure
Address Reference System Id	Address Reference System Elements	Simple	Tabular Domain Measure
Address Reference System Name	Address Reference System Elements	Simple	Tabular Domain Measure
Address Reference System Extent	Address Reference System Elements	Simple	Address Reference System Description
Address Reference System Type	Address Reference System Elements	Simple	Tabular Domain Measure
Address Reference System Rules	Address Reference System Elements	Complex	Address Reference System Rules Measure +
Address Reference System Block Rules	Address Reference System Elements	Simple	See Address Reference System Rules Measure.
Address Reference System Numbering Rules	Address Reference System Elements	Simple	See Address Reference System Rules Measure.
Address Reference System Street Naming Rules	Address Reference System Elements	Simple	See Address Reference System Rules Measure.
Address Reference System Street Type Directional And Modifier Rules	Address Reference System Elements	Simple	See Address Reference System Rules Measure.

Address Reference System Place Name State Country And Zip Code Rules	Address Reference System Elements	Simple	See Address Reference System Rules Measure.
Address Reference System Subaddress Rules	Address Reference System Elements	Simple	See Address Reference System Rules Measure.
Address Reference System Axis	Address Reference System Elements	Simple	Address Reference System Axes Point Of Beginning Measure +
Address Reference System Reference Polyline	Address Reference System Elements	Simple	See Address Reference System Rules Measure.
Address Reference System Range Breakpoint	Address Reference System Elements	Simple	See Address Reference System Rules Measure.
Address Reference System Range Breakline	Address Reference System Elements	Simple	See Address Reference System Rules Measure.
Address Reference System Range Polygon	Address Reference System Elements	Simple	See Address Reference System Rules Measure.
Address Reference System Reference Document Citation	Address Reference System Elements	Simple	See Address Reference System Rules Measure.
Address Reference System	Address Reference System Elements	Complex	Address Reference System Rules Measure +
Complete Address Number	Address Number	Complex	Pattern Sequence Measure
Complete Landmark Name	Landmark Name Elements	Complex	Complex Element Sequence Number Measure
Complete Landmark Name	Landmark Name Elements	Complex	Repeated Element Uniqueness Measure
Complete Place Name	Larger-Area Elements	Complex	Complex Element Sequence Number Measure
Complete Place Name	Larger-Area Elements	Complex	Repeated Element Uniqueness Measure
Complete Street Name	Street Name	Complex	Complete Street Name Tabular Domain Measure
Complete Street Name	Street Name	Complex	Pattern Sequence Measure
Complete Street Name	Street Name	Complex	Duplicate Street Name Measure +
Complete Subaddress	Subaddress Elements	Complex	Complex Element Sequence Number Measure
Complete Subaddress	Subaddress Elements	Complex	Repeated Element

			Uniqueness Measure
Country Name	Larger-Area Elements	Simple	Spatial Domain Measure +
Country Name	Larger-Area Elements	Simple	Tabular Domain Measure
Delivery Address	USPS Address Lines	Simple	Pattern Sequence Measure
Landmark Name	Landmark Name Elements	Simple	Uniqueness Measure
Place Name	Larger-Area Elements	Simple	Spatial Domain Measure +
Place Name	Larger-Area Elements	Simple	Tabular Domain Measure
Place State ZIP	USPS Address Lines	Complex	Pattern Sequence Measure
Private Mail Box	Subaddress Elements	Simple	None
Separator Element	Element	Simple	Tabular Domain Measure
State Name	Larger-Area Elements	Simple	Spatial Domain Measure +
State Name	Larger-Area Elements	Simple	Tabular Domain Measure
Street Name	Street Name	Simple	Spatial Domain Measure +
Street Name	Street Name	Simple	Tabular Domain Measure
Street Name Post Directional	Street Name	Simple	Spatial Domain Measure +
Street Name Post Directional	Street Name	Simple	Tabular Domain Measure
Street Name Post Modifier	Street Name	Simple	Spatial Domain Measure +
Street Name Post Modifier	Street Name	Simple	Tabular Domain Measure
Street Name Post Type	Street Name	Simple	Spatial Domain Measure +
Street Name Post Type	Street Name	Simple	Tabular Domain Measure
Street Name Pre Directional	Street Name	Simple	Spatial Domain Measure +
Street Name Pre Directional	Street Name	Simple	Tabular Domain Measure
Street Name Pre Modifier	Street Name	Simple	Spatial Domain Measure +
Street Name Pre Modifier	Street Name	Simple	Tabular Domain Measure
Street Name Pre Type	Street Name	Simple	Spatial Domain Measure +
Street Name Pre Type	Street Name	Simple	Tabular Domain Measure
Subaddress Element	Subaddress Elements	Complex	Pattern Sequence Measure

Subaddress Element	Subaddress Elements	Complex	Spatial Domain Measure +
Subaddress Identifier	Subaddress Elements	Simple	None
Subaddress Type	Subaddress Elements	Simple	Tabular Domain Measure
USPS Box Group ID	USPS Postal Address Elements	Simple	Tabular Domain Measure
USPS Box Group Type	USPS Postal Address Elements	Simple	Tabular Domain Measure
USPS Box ID	USPS Postal Address Elements	Simple	Tabular Domain Measure
USPS Box Type	USPS Postal Address Elements	Simple	Tabular Domain Measure
USPS General Delivery Point	USPS Postal Address Elements	Simple	Tabular Domain Measure
Usps Address	USPS Postal Address Elements	Complex	Pattern Sequence Measure
Usps Box	USPS Postal Address Elements	Complex	Tabular Domain Measure
Usps Route	USPS Postal Address Elements	Simple	Tabular Domain Measure
Zip Code	Larger-Area Elements	Simple	Spatial Domain Measure +
Zip Code	Larger-Area Elements	Simple	Tabular Domain Measure
Zip Plus 4	Larger-Area Elements	Simple	Spatial Domain Measure + (linear?)
Zip Plus 4	Larger-Area Elements	Simple	Tabular Domain Measure

7103 ## 7.6 Appendix F (Informative): Attribute Measure Index

Attribute Name	Component or Subject	Measure
Address Anomaly Status	Descriptive Attributes	None
Address Authority	Address Lineage Attributes	Tabular Domain Measure
Address Classification	Descriptive Attributes	Pattern Sequence Measure
Address Classification	Descriptive Attributes	Tabular Domain Measure
Address Coordinate Reference System	Address Coordinates	Tabular Domain Measure
Address Coordinate Reference System Authority	Address Coordinates	Tabular Domain Measure
Address Coordinate Reference System ID	Address Coordinates	Tabular Domain Measure
Address Elevation	Address Coordinates	Address Elevation Measure
Address End Date	Address Lineage Attributes	Future Date Measure
Address End Date	Address Lineage Attributes	Start End Date Order Measure
Address Feature Type	Descriptive Attributes	Address Reference System Description
Address Feature Type	Descriptive Attributes	Tabular Domain Measure
Address ID	Address ID	Uniqueness Measure
Address Latitude	Address Coordinates	XY Coordinate Completeness Measure
Address Latitude	Address Coordinates	XY Coordinate Spatial Measure
Address Lifecycle Status	Descriptive Attributes	Address Lifecycle Status Date Consistency Measure
Address Lifecycle Status	Descriptive Attributes	Tabular Domain Measure
Address Longitude	Address Coordinates	XY Coordinate Completeness Measure
Address Longitude	Address Coordinates	XY Coordinate Spatial Measure
Address Number Parity	Attributes Describing Specific Address Elements	Address Number Parity Measure
Address Range Parity	Attributes Describing Specific Address Elements	Address Number Range Parity Consistency Measure
Address Range Side	Attributes Describing	Address Left Right Measure

	Specific Address Elements	
Address Range Side	Attributes Describing Specific Address Elements	Left Right Odd Even Parity Measure
Address Range Type	Attributes Describing Specific Address Elements	None
Address Reference System	Address Reference System Attributes	Refer to component elements
Address Reference System Axis Point Of Beginning	Address Reference System Attributes	Address Reference System Axes Point Of Beginning Measure -
Address Reference System Description	Address Reference System Attributes	None
Address Reference System Extent	Address Reference System Attributes	Compare with the Address Reference System Description
Address Reference System Name	Address Reference System Attributes	Tabular Domain Measure
Address Relation Type	Descriptive Attributes	Tabular Domain Measure
Address Scheme X Axis	Address Reference System Attributes	Address Reference System Axes Point Of Beginning Measure
Address Scheme Y Axis	Address Reference System Attributes	Address Reference System Axes Point Of Beginning Measure
Address Scheme Y Axis Origin	Address Reference System Attributes	Address Reference System Axes Point Of Beginning Measure -
Address Start Date	Address Lineage Attributes	Future Date Measure
Address Start Date	Address Lineage Attributes	Start End Date Order Measure
Address UUID	Address ID	Refer to validation specific to the software used to create the UUID
Address X Coordinate	Address Coordinates	XY Coordinate Completeness Measure
Address X Coordinate	Address Coordinates	XY Coordinate Spatial Measure
Address Y Coordinate	Address Coordinates	XY Coordinate Completeness Measure
Address Y Coordinate	Address Coordinates	XY Coordinate Spatial Measure
Address Z Level	Descriptive Attributes	Subaddress Element Z Level Measure
Attached Element	Attributes Describing Specific Address Elements	Check Attached Pairs Measure
Attached Element	Attributes Describing	Complete Street Name Tabular

	Specific Address Elements	Domain Measure
Data Set ID	Address Lineage Attributes	Related Not Null Measure
Delivery Address Type	Attributes Describing Specific Address Elements	Delivery Address Type Subaddress Measure
Delivery Address Type	Attributes Describing Specific Address Elements	Tabular Domain Measure
Element Sequence Number	Attributes Describing Specific Address Elements	Element Sequence Number Measure
Element Sequence Number	Attributes Describing Specific Address Elements	Uniqueness Measure
GNIS Feature ID	Attributes Describing Specific Address Elements	Related Not Null Measure
Location Description	Descriptive Attributes	None
Official Status	Descriptive Attributes	Official Status Address Authority Consistency Measure
Official Status	Descriptive Attributes	Tabular Domain Measure
Place Name Type	Attributes Describing Specific Address Elements	None
Related Address ID	Descriptive Attributes	Related Element Uniqueness Measure
Related Address ID	Descriptive Attributes	Related Not Null Measure
Related Address ID	Descriptive Attributes	Tabular Domain Measure
Subaddress Component Order	Attributes Describing Specific Address Elements	Subaddress Component Order Measure
Subaddress Component Order	Attributes Describing Specific Address Elements	Tabular Domain Measure
US National Grid Coordinate	Address Coordinates	Usng Coordinate Spatial Measure

7.7 Appendix G (Informative): Classification Measure Index

Classification Name	Subject	Measure
Community Address	Landmark Address Classes	Pattern Sequence Measure
Four Number Address Range	Thoroughfare Address Classes	Address Number Range Completeness Measure
Four Number Address Range	Thoroughfare Address Classes	Address Number Range Parity Consistency Measure
Four Number Address Range	Thoroughfare Address Classes	Address Number Range Sequence Measure
Four Number Address Range	Thoroughfare Address Classes	Left Right Odd Even Parity Measure
Four Number Address Range	Thoroughfare Address Classes	Low High Address Sequence Measure
Four Number Address Range	Thoroughfare Address Classes	Overlapping Ranges Measure
Four Number Address Range	Thoroughfare Address Classes	Pattern Sequence Measure +
Four Number Address Range	Thoroughfare Address Classes	Range Domain Measure
General Address Class	General Address Class	Pattern Sequence Measure
Intersection Address	Thoroughfare Address Classes	Intersection Validity Measure
Intersection Address	Thoroughfare Address Classes	Pattern Sequence Measure
Landmark Address	Landmark Address Classes	Pattern Sequence Measure
Landmark Site Address	Thoroughfare Address Classes	Address Number Fishbones Measure
Landmark Site Address	Thoroughfare Address Classes	Address Number Range Completeness Measure
Landmark Site Address	Thoroughfare Address Classes	Left Right Odd Even Parity Measure
Landmark Site Address	Thoroughfare Address Classes	Pattern Sequence Measure +
Numbered Thoroughfare Address	Thoroughfare Address Classes	Address Completeness Measure

Numbered Thoroughfare Address	Thoroughfare Address Classes	Address Number Fishbones Measure
Numbered Thoroughfare Address	Thoroughfare Address Classes	Left Right Odd Even Parity Measure
Numbered Thoroughfare Address	Thoroughfare Address Classes	Pattern Sequence Measure +
Two Number Address Range	Thoroughfare Address Classes	Address Number Range Completeness Measure
Two Number Address Range	Thoroughfare Address Classes	Address Number Range Parity Consistency Measure
Two Number Address Range	Thoroughfare Address Classes	Address Number Range Sequence Measure
Two Number Address Range	Thoroughfare Address Classes	Low High Address Sequence Measure
Two Number Address Range	Thoroughfare Address Classes	Low High Address Sequence Measure
Two Number Address Range	Thoroughfare Address Classes	Overlapping Ranges Measure
Two Number Address Range	Thoroughfare Address Classes	Pattern Sequence Measure +
Two Number Address Range	Thoroughfare Address Classes	Range Domain Measure
USPS General Delivery Office	Landmark Address Classes	Pattern Sequence Measure
USPS Postal Delivery Box	Postal Delivery Address Classes	Pattern Sequence Measure
Unnumbered Thoroughfare Address	Thoroughfare Address Classes	Pattern Sequence Measure

7.8 Appendix H (Informative): Quality Measures By Data Quality Report

Measure	Attribute (Thematic) Accuracy	Completeness	Lineage	Logical Consistency	Positional Accuracy	Temporal Accuracy
Address Completeness Measure		•				
Address Left Right Measure	•					
Address Lifecycle Status Date Consistency Measure				•		•
Address Number Fishbones Measure				•		
Address Number Parity Measure				•		
Address Number Range Completeness Measure				•		
Address Number Range Parity Consistency Measure				•		
Address Number Range Sequence Measure				•		
Address Range Directionality Measure				•		

Address Reference System Axes Point Of Beginning Measure				•		
Address Reference System Rules Measure				•		
Check Attached Pairs Measure				•		
Complete Street Name Tabular Domain Measure	•					
Complex Element Sequence Number Measure	•					
Data Type Measure				•		
Delivery Address Type Subaddress Measure				•		
Duplicate Street Name Measure				•		
Element Sequence Number Measure	•					
Future Date Measure						•
Intersection Validity Measure				•		
Left Right Odd Even Parity				•		

Measure						
Location Description Field Check Measure			•		•	
Low High Address Sequence Measure				•		
Official Status Address Authority Consistency Measure				•		
Overlapping Ranges Measure				•		
Pattern Sequence Measure				•		
Range Domain Measure	•					
Related Element Uniqueness Measure	•					
Related Not Null Measure				•		
Repeated Element Uniqueness Measure	•					
Segment Directionality Consistency Measure				•		
Spatial Domain Measure					•	
Start End Date Order Measure						•
Subaddress Component	•					

Order Measure						
Subaddress Element Z Level Measure	•					
Tabular Domain Measure	•					
Uniqueness Measure	•					
Usng Coordinate Spatial Measure					•	
XY Coordinate Completeness Measure				•		
XY Coordinate Spatial Measure					•	

7.9 Appendix I (Informative): Compatibility of the Address Standard with the FGDC Geographic Information Framework Data Content Standard for the NDSI

7.9.1 1 Introduction

7.9.1.1. Purpose and Structure.

This appendix assesses the compatibility of the address standard with the FGDC's *Geographic Information Framework Data Content Standard* (hereinafter called the "framework standard"). This appendix is presented in three sections:

- Section 1 states why and how the assessment was done, and summarizes the results.
- Section 2 provides a brief statement of the scope of each part of the framework standard, whether the address standard is consistent with that part, and how the evaluation should be independently confirmed.
- Section 3 shows in detail whether and to what extent the address standard meets the conformance tests set forth in Part Zero (Base Part) of the framework standard.

7.9.1.2. The Framework Standard and the Address Standard.

The framework standard *"provides interrelated thematic standards in seven data areas: cadastral, digital orthoimagery, elevation, geodetic control, governmental unit boundaries and other geographic area boundaries, hydrography, and transportation."* The seven core themes *"are considered framework data of critical importance to the spatial data infrastructure of the Nation... The standard is divided into eight parts, one for each of the seven data themes and a base document containing information common to two or more themes."* (Framework standard Base Part, Introduction and Sec. 1.1)

Address data are used in conjunction with several of the framework themes, most notably cadastral data and transportation data. Addresses and transportation features (especially road networks) are so closely related that their standards are interdependent. Addresses are used by the public to identify cadastral parcels and specify their locations. Street names form an integral part of thoroughfare addresses, and street segments and their network geometry form the basis for Address Reference Systems and their components. In addition, addressed features

have elevations; and place names within addresses are often determined by governmental boundaries.

7.9.1.3. Assessing the Compatibility of the Address Standard with the Framework Standard.

Because address data are closely tied to several framework data themes, the address standard should be compatible with the framework standard. Compatibility assessment requires two types of tests:

- **Consistency tests,** to find whether the address standard is consistent with the standards for the seven data themes, and
- **Conformity tests,** to determine whether the address standard conforms to the requirements set forth in the Base Part of the framework standard, which govern the seven thematic parts of the framework standard.

7.9.1.4. Consistency Tests and Results.

The consistency tests evaluate, for each thematic part, whether the part shares any classes, elements, or defined terms with the address standard, and if so, whether the shared classes, elements, or terms are defined and used consistently. Three outcomes are possible:

- **Unrelated** - The framework part shares no classes, elements, or defined terms with the address standard.
- **Consistent** - The framework part shares classes, elements, or defined terms with the address standard; and they are defined and used consistently; and the two standards are complementary and mutually exclusive in scope.
- **Inconsistent** - The framework part shares classes, elements, or defined terms with the address standard, but they are not defined and used consistently, and/or the two standards overlap in scope.

The address standard relates to the data theme parts as follows:

- **Unrelated** - Digital Orthoimagery, Geodetic Control, and Hydrography.
- **Consistent** - Cadastral, Elevation, and Governmental Unit Boundaries and Other Geographic Area Boundaries.
- **Inconsistent** - Transportation (see 2.8.2 below).

7.9.1.5. Conformity Tests and Results

Section 3 sets forth, section by section, all the conformance requirements given in the framework standard Base Part and analyzes whether and how the address standard conforms

to the requirements. The address standard satisfies all of the requirements. Section 3 below details the specific requirements and shows how the address standard conforms to them.

7.9.1.6. Relating the Address Standard to the Framework Standard

Cadastral and Transportation Parts

The close relation of address data with cadastral data and with transportation data raises the question of how the address standard should be related to the cadastral and transportation parts of the framework standard. If, for example, an address record is to be related to a land parcel record, the address standard should not have to reinvent or repeat the entire cadastral part in order to make use of the data found in a cadastral dataset. This address standard incorporates a framework approach:

- To best serve geographic data users, the address standard should provide explicitly for relationships with other standards.
- This is best done by defining a minimum set of attributes needed to relate features across different themes (e.g. an address to a parcel, or an address to a transportation feature), that is, to provide for the foreign key needed to relate address records to cadastral features or transportation features; and
- Those key attributes should be defined by reference to the other standard.

The Content Part of the address standard includes two elements, Address Parcel Identifier Source and Address Parcel Identifier, that were created to relate addresses with parcels.

The Content Part address standard includes five attributes by which an address feature can be related to a transportation event and a transportation segment or path: Address Transportation System Name, Address Transportation System Authority, Address Transportation Feature Type, Address Transportation Feature ID, and Related Transportation Feature ID. In addition, the Content Part includes five address range attributes, so that address ranges can be properly related to the transportation segments or paths they describe: Address Range Type, Address Range Parity, Address Range Side, Address Range Directionality, and Address Range Span.

7.9.1.7. Format Note

Within this appendix, quotations from the framework standard are italicized and set in quotation marks.

7.9.1.8. Sources

This appendix refers to the May 2008 versions of the *Geographic Information Framework Data Content Standard* as posted on the FDGC website at:

http://www.fgdc.gov/standards/standards_publications/ . Complete citations are given in Appendix A.

7.9.2. Relationship of the Address Standard to Each of the Eight Parts of the Geographic Information Framework Data Content Standard

7.9.2.1 Part 0: Base

7.9.2.1.1 Scope of Part 0: Base.

The Base Part provides *"A high-level view of the seven framework data themes[,] [a]n overall integrating Unified Modeling Language (UML) model that is supplemented by detail in the part for each data theme, [and] [t]erminology and other information common to two or more themes"* (Part 0, Sec 1.2).

The Base Part defines the abstract model that underlies and unifies the seven data themes. It sets forth, for the data themes, specific conformance requirements as to definitions of terms and abbreviations, UML model notation, data dictionary content and formatting, element and attribute naming, incorporation of metadata and record identifiers, and conformance to ISO reference standards and the abstract framework data model.

7.9.2.1.2. Relation of Part 0 to Address Standard.

To be compatible with the framework standard, the address standard must meet the conformance requirements given in the Base Part, or at least not contradict them. As shown in the detailed analysis in Section 3, the address standard conforms to all of the requirements.

7.9.2.1.3. Conclusion

The address standard conforms to the Base Part.

7.9.2.2 Part 1: Cadastral

7.9.2.2.1. Scope of Part 1: Cadastral.

Part 1 *"provides the information necessary to identify the existence of parcel-level cadastral information and the source of that information."* (Part 1, Sec. 1).

Part 1 is a profile of the FGDC's *Cadastral Data Content Standard* (FGDC-STD-003). The *Cadastral Data Content Standard "contains the standardization of the definition of entities*

and objects related to cadastral information including survey measurements, transactions related to interests in land, general property descriptions, and boundary and corner evidence data." (Part 1, Introduction).

7.9.2.2.2. Relation of Part 1 and the Cadastral Data Content Standard to the Address Standard.

The address standard is consistent with both the Cadastral Part of Framework Standard and the Cadastral Data Content Standard. The address standard includes two address attributes, Address Parcel Identifier and Address Parcel Identifier Source, both defined by reference to the *Cadastral Data Content Standard.* They correspond to the Parcel ID and Source Identifier (or Parcel ID Assigner) elements, respectively, in the Cadastral Part and the Cadastral Data Content Standard.

Because addresses and parcels are created and altered independently of each other, no specific address-parcel relationship can be assumed. They should be treated as independent entities, and the relationship between them should be considered, in relational database terms, as a many-to-many relationship--that is, an address can relate to any number of parcels, and a parcel can relate to any number of addresses.

The Address Parcel Identifier and the Address Parcel Identifier Source are both defined by reference to the Cadastral Standard, and they are the only parcel elements included or needed within the address standard. Except for those two attributes, the address and cadastral standards do not share any defined terms, data elements, or data classes. All other parcel elements are defined within the Cadastral Standard and need not be repeated in the address standard. All address elements and classes are defined in the address standard and need not be repeated in the Cadastral Standard. Thus the two standards are consistent in their shared elements, and mutually exclusive and complementary in their scopes.

7.9.2.2.3. Conclusion

The Address Standard is consistent with the Framework Standard Part 1: Cadastral.

7.9.2.3 Part 2: Digital Orthoimagery

7.9.2.3.1 Scope of Part 2: Digital Orthoimagery.

Part 2 *"specifies data content and logical structure for the description and interchange of framework digital orthoimagery. To a certain extent, it also provides guidelines for the acquisition and processing of imagery (leading toward the generation of digital orthoimagery), and specifies the documentation of those acquisition and processing steps."* (Part 2, Sec 1.1)

7.9.2.3.2 Relation of Part 2 to Address Standard.

The address standard does not refer to digital orthoimagery, and it does not share any defined terms, data elements, or data classes with Part 2.

7.9.2.3.3. Conclusion

The address standard is unrelated to the Digital Orthoimagery Part.

7.9.2.4 Part 3: Elevation

7.9.2.4.1. Scope of Part 3: Elevation.

Part 3 *"defines the geospatial data model entities and attributes that permit the exchange of digital elevation data consistent with the National Spatial Data Infrastructure's (NSDI) framework for elevation data."* (Part 3, Sec. 1)

7.9.2.4.2 Relation of Part 3 to Address Standard.

The address standard includes address attributes that define horizontal and vertical coordinates for address points, and the coordinate reference system to which the coordinates are referenced. The attributes are:

Horizontal: Address X Coordinate, Address Y Coordinate, Address Longitude, Address Latitude, US National Grid Coordinate

Vertical: Address Elevation

Coordinate Reference System: Address Coordinate Reference System ID, Address Coordinate Reference System Authority; Complex Element: Address Coordinate Reference System

The address attributes listed above are consistent with Part 3, and otherwise the two standards are independent and unrelated.

7.9.2.4.3 Conclusion

The address standard is consistent with the Elevation Part.

7.9.2.5. Part 4: Geodetic Control

7.9.2.5.1. Scope of Part 4: Geodetic Control.

Part 4 *"provides a common methodology for creating datasets of horizontal coordinate values and vertical coordinate values for geodetic control points represented by survey monuments, such as brass disks and rod marks. It provides a single data structure for relating coordinate values obtained by one geodetic survey method (for example, a classical line-of-sight traverse) with coordinate values obtained by another geodetic survey method (for example, a Global Positioning System geodetic control survey)."* (Part 4, Sec .1.2)

7.9.2.5.2 Relation of Part 4 to Address Standard.

The address standard does not refer to control points, and it does not share any defined terms, data elements, or data classes with Part 4.

7.9.2.5.3. Conclusion

The address standard is unrelated to the Geodetic Control Part.

7.9.2.6. Part 5: Governmental Units and Other Geographic Area Boundaries

7.9.2.6.1. Scope of Part 5: Governmental Units and Other Geographic Area Boundaries.

"The purpose of ...Part 5...is to establish the content requirements for the collection and interchange of governmental units and other geographic area boundary data and to facilitate the maintenance and use of that information." (Part 5, Sec 1).

The part recognizes four types of areas (definitions are quoted from Part 5, Sec.5.5):

- governmental unit - *"geographic area with legally defined boundaries established under Federal, Tribal, State, or local law, and with the authority to elect or appoint officials and raise revenues through taxes"* (Sec. 5.5.12)
- administrative unit - *"area established by rule, treaty, or regulation of a legislative, executive, or judicial governmental authority, a non-profit organization, or private industry for the execution of some function"* (Sec. 5.5.1)
- statistical unit - *"geographic area defined for the collection, tabulation, and/or publication of demographic, and/or other statistical data"* (sec. 5.5.20)
- other unit - *"geographic area that is not a governmental unit, administrative unit, or statistical unit, as defined herein, and that is not an area defined or described in other*

framework parts" (Sec. 5.5.17)

7.9.2.6.2. Relation of Part 5 to Address Standard.

The address standard is related to the Governmental Units and other Geographic Area Boundaries Part in two ways:

- Government unit names and other geographic area names often also serve as address Place Names, State Names, or Country Names.
- Part 5 defines boundaries and spatial relationships. The Data Quality Part of the address standard uses spatial relationships to test whether the address is within the polygon that represents the address Place Name(s), State Name, or Country Name.

To provide for consistency of terminology:

- The address standard definition of Place Name is based in part on the Framework Standard Part 5.
- Tables 11, 13, and 15 of Part 5, which provide an extensive list of terms and definitions for various types of communities and local governments, are cited in the address standard Place Name notes.
- Relevant terms from tables 11, 13, and 15 are listed in the address standard under Place Name as "Other Common Names for the Element."
- The address standard notes for State Name cite the definition of "state" given in framework standard part 5, Table 13.
- The address standard definition of Country Name incorporates the definition of "country" given in framework standard part 5, Table 13.
    - o The data quality tests use boundary polygons and spatial relationships in a manner consistent with the definitions of Part 5.

7.9.2.6.3. Conclusion

The address standard is consistent with the Governmental Units and other Geographic Area Boundaries Part.

7.9.2.7. Part 6: Hydrography

7.9.2.7.1. Scope of Part 6: Hydrography.

"The purpose of ... Part 6 ... is to establish the content requirements for the collection and interchange of hydrography features and to facilitate the maintenance and use of that information by all users of geographic information. The Hydrography part identifies and defines terminology, encoding schema, and the data components required for describing hydrographic features, along with the metadata needed for the hydrography data exchange.... The scope of this part is limited to the information regarding surface water features and

hydrographic networks for the purpose of cartography and network analysis." (Part 6, Sec. 1.1)

7.9.2.7.2. Relation of Part 6 to Address Standard.

The address standard does not refer to hydrography or hydrographic features, and it does not share any defined terms, data elements, or data classes with Part 6.

7.9.2.7.3. Conclusion

The address standard is unrelated to the Hydrography Part.

7.9.2.8 Part 7: Transportation

7.9.2.8.1 Scope of Part 7: Transportation.

Part 7 *"defines the data model for describing transportation systems components of transportation systems for five [sic] modes that compose the Transportation theme of the NSDI."* (Part 7, Sec. 1).

Part 7 is comprised of five sub-parts: the Transportation Base Part (Part 7), and Rail, Roads, Transit, and Inland Waterways (Parts 7b through 7e). (Part 7a, Transportation - Air, was drafted but not endorsed.) The Base, Roads, and Transit subparts are especially germane to the address standard.

7.9.2.8.2 Relation of Part 7 to Address Standard.

Addresses and transportation networks--and the standards that define them--are so closely related as to be interdependent. In particular, the thoroughfare address classes locate addresses by reference to a thoroughfare; thoroughfare networks are defined and described in the Transportation Part of the Framework Standard. Appendix C (informative) describes the interdependence and complementarity of the two standards in detail.

The address standard includes five elements by which an address feature can be related to a transportation event and a transportation segment or path: Address Transportation System Name, Address Transportation System Authority, Address Transportation Feature Type, Address Transportation Feature ID, and Related Transportation Feature ID.

The address standard includes five address range attributes, so that address ranges can be properly related to the transportation segments they describe: Address Range Type, Address Range Parity, Address Range Side, Address Range Directionality, and Address Range Span.

These elements are defined to incorporate by reference the transportation model defined in the Transportation Part, without overlapping it.

Because the Transportation Part was completed before the address standard was started, it overlaps with the address standard in certain respects. Within the Transit subpart, Annex D (Informative) describes an address extension to the transit model. The model is inconsistent with the address standard. In addition, the following classes, attributes, and code list values overlap and in some respects are inconsistent with elements in the address standard:

- Transit, Table 1 (Data Dictionary for TransitStop) attributes: address, street side
- Transit, Table 10(Data Dictionary for Landmark) class and attributes: Landmark, landmarkName, landmarkType, address
- Transit, Table 11 (Data Dictionary for Facility) attributes: address
- Roads, Table 3 (Code List for RoadLinearEventType) code list values: directionalPrefix, directionalSuffix, addressInformation, alternateNameBody, alternateNameText, alternateStreetName, alternateStreetNameBody, alternateStreetNameText, firstHouseNumber, houseNumberRange, houseNumberStructure, intermediateHouseNumber, lastHouseNumber, postalCode

7.9.2.8.3 Conclusion

The address standard and the Transportation Part are inconsistent. They can be made consistent by replacing or redefining Annex D and the class, attributes and values listed above with reference to the address standard.

7.9.3 Conformance Of The Address Standard To Framework Standard

Part Zero Base Part

The framework standard Base Part defines the abstract model that underlies and unifies the framework seven data themes. It sets forth, for the data themes, specific conformance requirements as to definitions of terms and abbreviations, UML model notation, data dictionary content and formatting, element and attribute naming, incorporation of metadata and record identifiers, and conformance to ISO reference standards and the abstract framework data model.

Section 3 sets forth the conformance requirements given in the framework standard Base Part, section by section, and analyzes whether and how the address standard conforms to the requirements. As shown below, the address standard conforms to all of the requirements.

7.9.3.1 Conformance to Base Part Section 1: Scope

Framework Base Part Section 1 states the scope of the Framework Standard, the Base Part and the seven data theme parts. It is descriptive; it imposes no conformance requirements that would apply to the address standard.

7.9.3.2 Conformance to Base Part Section 2: Conformance

Framework Base Part Section 2 states in full: *"2. Conformance. Each thematic part of the Framework Data Content Standard includes a data dictionary based on the conceptual schema presented in that part. To conform to the standard, a thematic dataset shall satisfy the requirements of the data dictionary for that theme. It shall include a value for each mandatory element, and a value for each conditional element for which the condition is true. It may contain values for any optional element. The data type of each value shall be that specified for the element in the data dictionary and the value shall lie within the domain specified for the element."*

Address Standard Conformance to Section 2: The address standard includes a data dictionary (the Content Part) and a conceptual schema (the XSD in the Exchange Part). The Content Part provides data types and (if applicable) domains for each elements and attribute. The Classification Part shows which elements are mandatory for each class. The address standard thus includes all information needed to determine whether a given dataset conforms to the standard.

7.9.3.3 Conformance to Base Part Section 3: Normative References

Framework Base Part Section 3 refers to Annex A, which lists normative references to standards that affect two or more parts of the Framework Data Content Standard. This section imposes no conformance requirements that would apply to the address standard.

7.9.3.4 Conformance to Base Part Section 4: Maintenance Authority

Framework Base Part Section 4 states that the FGDC is the maintenance authority for the Base Part, and it provides a contact point for questions. This section imposes no conformance requirements that would apply to the address standard.

7.9.3.5 Conformance to Base Part Section 5: Terms and Definitions

Framework Base Part Section 5 defines terms used in the Base Part part or common to two or more parts of the standard. Two of the terms are pertinent to the address standard:

"5.12 data content standard – standard that specifies what information is contained within a geospatial dataset and provides an application schema"

Address Standard Conformance to 5.12: The address standard specifies what information is contained within an address dataset and provides an address schema. Thus the address standard fits the definition of a data content standard.

"5.22 feature type – category of real world phenomena with common properties [ISO 19126]"

Address Standard Conformance to 5.22: Addresses are real world phenomena with common properties. The Classification Part of the address standard specifies the common properties of the various classes of addresses. Addresses therefore meet the definition of "feature type."

7.9.3.6 Conformance to Base Part Section 6: Symbols, Abbreviated Terms, and Notations

Framework Base Part Section 6 lists abbreviations used in the Base Part or common to two or more parts of the Framework Standard. Abbreviations used in the address standard are consistent with the abbreviations listed in the Base Part.

7.9.3.7 Conformance to Base Part Section 7: Requirements

7.9.3.7.1 Conformance to Base Part Subsection 7.1: Unified Modeling Language (UML) model

Framework Base Part Section 7.1 reads in full: *"7.1 Unified Modeling Language (UML) model. A data model expressed in UML is provided in each theme part in one of the following ways:*

- *Incorporated in the body text in each section that needs it*
- *Incorporated in the body text in a UML model-only section*
- *Incorporated in a normative annex and referenced in the body text*
- *Incorporated in the body text, but only at a high level or in a general way with detailed data components of the model presented in a normative annex*

"The use of UML class diagrams in the Framework Data Content Standard is an application-neutral approach to depict the inherent description of and relationships among data entities. These diagrams should neither be interpreted as requiring object-oriented implementation – methods or interfaces are not typically shown on these data classes – nor should they be interpreted as representing tables in relational databases. Instead, the UML classes should be used as the basis for translation to and from internal organization data stores and applications. UML modeling environments typically support conversion of logical UML models into implementations in various programming environments through rule-based transforms."

Address Standard Conformance to Base Part 7.1: The Data Exchange Part provides a UML model of the standard, and a complete XSD.

7.9.3.7.2 Conformance to Base Part Subsection 7.2: Dependence on ISO 19100 series of geographic information standards

Framework Base Part Section 7.2 reads in full: *"7.2 Dependence on ISO 19100 series of geographic information standards. The Framework Data Content Standard is dependent on structures and concepts from several standards in the ISO 19100 series of geographic information standards, as shown in Figure 1. Full titles for these standards are found in Annex A. The digital orthoimagery and elevation data parts also are dependent on ISO 19123. Data standards for certain transportation modes are dependent on ISO 19133. All parts have dependencies on ISO 19107, ISO 19108, ISO 19109, ISO 19111, and ISO 19115."*

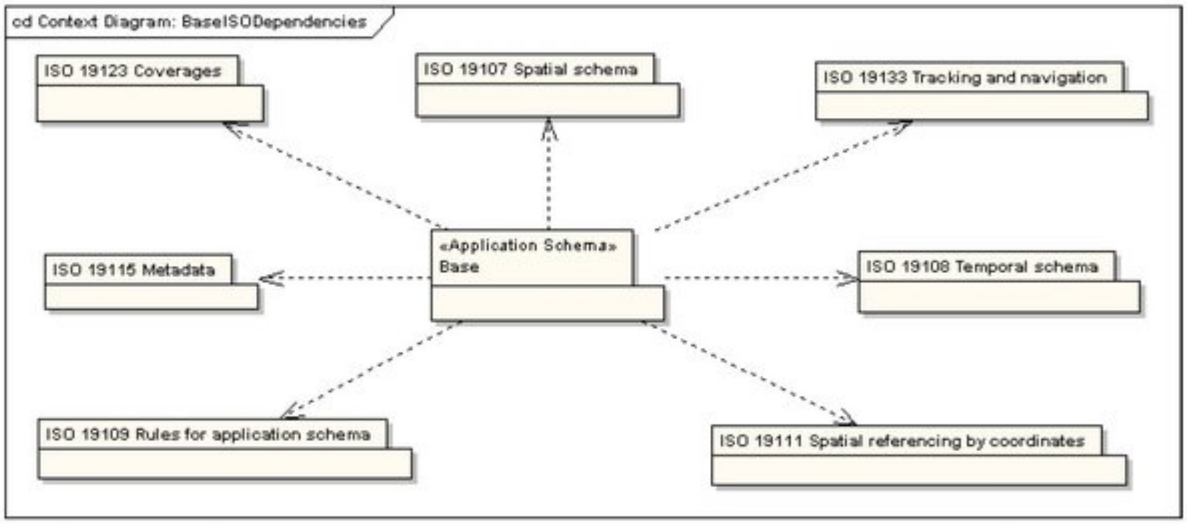


Figure 1 – Framework base dependencies on ISO series of geographic information standards

Address Standard Conformance to Base Part 7.2. The address standard is not directly dependent on any of the ISO 19100 series of geographic information standards, because there is no ISO 19100 standard for addresses. To the extent that the address standard is indirectly dependent on other ISO standards that govern the framework standard, conformance to this section (7.2) is shown by the conformance of the address standard to the Base Part of the Framework Standard.

7.9.3.7.3 Conformance to Base Part Subsection 7.3: Application schema

Framework Base Part Section 7.3 reads in full: *"7.3 Application schema. Each of the thematic Framework Data Content Standard parts includes an integrated application schema expressed in the Unified Modeling Language (UML) according to ISO 19109, Geographic information – Rules for application schema, and its normative references. The application schema specifies, as appropriate, the feature types, attribute types, attribute domain, feature relationships, spatial representation, data organization, and metadata that define the information content of a dataset.*

"The UML models included in the parts of the standard describe the common content and structures that can be exchanged between members of the geospatial community. The use of UML and abstract modeling concepts allows the standard to be technology independent but permits current and future implementation cases to be derived from the UML model.

"Whenever possible, the standard references abstract UML object types from the ISO 19100 series of standards and OGC specifications. Specialization of these classes of objects allows each theme to inherit properties and behaviors and ensure their propagation when transformed into an encoding such as XML.

"UML concepts and notation are described in Annex B." (Base Part subsection 7.3, quoted in full)

Address Standard Conformance to Base Part 7.3. The UML model and XSD provided in the Data Exchange Part express an integrated application schema that define the information content of the standard.

7.9.3.7.4 Conformance to Base Part Subsection 7.4: Data dictionary

7.9.3.7.4.1 Conformance to Base Part Subsection 7.4.1: General requirements

Framework Base Part Section 7.4.1 reads in full: *"7.4.1 General requirements. Each of the thematic Framework Data Content Standard parts contains, as appropriate, documentation of all features, attributes, and relationships and their definitions. A data dictionary table describes the characteristics of the UML model diagrams.*

"The data dictionary (see Table 1) is structured as follows:

- *Each UML model class equates to a data dictionary entity*
- *Each UML model class attribute equates to a data dictionary element*
- *Each UML model role name equates to a data dictionary element*
- *The shaded rows define entities*
- *The entities and elements within the data dictionary are defined by six attributes based on those specified in ISO/IEC 11179-3 for the description of data element concepts, that is, data elements without representation."*

Table 1 – Data dictionary table format

Line	Name/Role Name	Definition	Obligation/Condition	Maximum Occurrence	Data Type	Domain
1						
2						
3						

Address Standard Conformance to Base Part 7.4.1. The address standard Content Part provides a data dictionary of all the elements and attributes specified in the address standard . The dictionary provides the required information about each element and attribute, and extends the base standard by including additional items.

In the address standard each address data element is described by giving its:

1. Element name: The name of the element.
2. Other common names for this element: Common words or phrases having the same or similar meaning as the element name.
3. Definition: The meaning of the element.
4. Definition Source: The source of the definition. ("New" indicates that the definition is original.)
5. Data Type: Whether the element is a characterString, integer, datetime, etc.
6. Existing Standards for this Element: Other standards that govern this element (if any).
7. Domain of Values for this Element: The range or set of values (if any) to which the element is restricted.
8. Source of Values: The source (if any) for the domain of values.
9. How Defined: How the domain of values is defined.
10. Example: Illustrative examples of the element.
11. Notes/Comments: Notes and comments giving further explanation about the element.
12. XML Tag: The XML tag for the element.
13. XML Model: XML model of the element.
14. XML Example: The XML model applied to a specific example of the element.
15. XML Notes: Explanatory notes about the XML model.
16. Quality Measures: Quality tests applied to the class.
17. Quality Notes: Explanatory notes about the quality measures applied to this element.

The list above includes all the information required by the Base Part 7.4.1. Specifically:

1. Name/Role Name is provided under “Element Name”
2. Definition is provided under “Definition”
3. Obligation/Condition is provided in the XML model
4. Maximum Occurrence is provided in the XML model
5. Data Type is provided under “Data Type”
6. Domain is provided under “Domain of Values for this Element”

The address standard data dictionary includes additional information to encourage widespread and consistent use of the standard by providing clear and complete explanatory information, notes, and examples about each element and attribute. The documentation for address data elements in the Address Standard meets the requirements used by the Framework Data Standard, and provides for additional attributes.

7.9.3.7.4.2 Conformance to Base Part Subsection 7.4.2: Name/Role name

Framework Base Part Section 7.4.2 reads in full: *“7.4.2: Name/Role name. The name/role name is a label assigned to a data dictionary entity or to a data dictionary element.*

The class name begins with an upper case letter. Spaces do not appear in an entity name: instead, multiple words are concatenated, with each word starting with a capital letter (example: XnnnYmmm). Entity names are unique within a data theme.

Element names start with a lower case letter. Spaces do not appear in an element name: instead, multiple words are concatenated, with subsequent words starting with a capital letter (example: xnnnYmmm). Element names are unique within an entity. Combinations of the entity and element names (example: Dataset.name) are therefore unique within a data theme.

Role names are used to identify the roles of the classes at the ends of a model association and are preceded by the term "Role name" followed by a colon to distinguish them from other types of data dictionary elements."

Address Standard Conformance to Base Part 7.4.2. The address standard conforms to this section in substance, but not in form:

- The address standard (specifically the content and class parts) provides unique names for every element, attribute, and class.
- Consistent naming conventions are used for for class, element, and attribute names and XML tags.
- The address standard does not define any roles nor specify any role names.

7.9.3.7.4.3 Conformance to Base Part Subsection 7.4.3: Definition

Framework Base Part Section 7.4.3 reads in full: *"7.4.3: Definition. The definition is the entity or element description."*

Address Standard Conformance to Base Part 7.4.3. The address standard (specifically the content and class parts) includes a formal definition for every element, attribute, and class in the standard.

7.9.3.7.4.4 Conformance to Base Part Subsection 7.4.4: Obligation/Condition

Framework Base Part Section 7.4.4 reads in full:

"7.4.4.1 General

"Used only in rows that contain elements, Obligation/Condition is a descriptor indicating whether the element shall always be populated (that is, contain a value or values) or sometimes will be populated for every instance of its owning entity. If the element is a role name, then the obligation/condition shall apply to the element indicated by the Data Type. This descriptor may have the following values: M (mandatory), C (conditional), or O (optional).

"7.4.4.2 Mandatory (M)

"Mandatory (M) indicates that the entity or element shall be populated.

"7.4.4.3 Conditional (C) "Conditional (C) specifies an electronically manageable condition under which at least one entity or element is mandatory. "Conditional" is used for one of the three following possibilities:

- *Expressing a choice between two or more options. At least one option is mandatory and must be populated*
- *Populating an entity or element if another element has been populated*
- *Populating an element if a specific value for another element has been populated.*

"To facilitate reading by humans, the specific value is used in plain text (for example, "C/not defined by encoding?"). However, the code shall be used to verify the condition in electronic user interface,

"If the answer to the condition is positive, then the entity or the element shall be populated.

"7.4.4.4 Optional (O)

"The entity or the element may be populated. Optional (O) entities and optional elements have been defined to provide a guide to those looking to fully document their data. (Use of this common set of defined elements will help promote interoperability among framework data users and producers.) Optional entities may have mandatory elements. If the optional entity is used, the mandatory elements shall be used. If an optional entity is not used, the elements contained within that entity (including mandatory elements) will also not be used. "

Address Standard Conformance to Base Part 7.4.4. Obligation/conditionality is indicated in the XML model of each element and attribute, and in syntax descriptions and XML model of each address class.

7.9.3.7.4.5 Conformance to Base Part Subsection 7.4.5: Maximum occurrence

Framework Base Part Section 7.4.5 reads in full: *"7.4.5: Maximum occurrence Used only in rows that contain elements, maximum occurrence specifies the maximum number of instances the element may have. Single occurrences are shown by "1"; unconstrained number of instances are represented by an asterisk "*". Fixed number occurrences, other than one, are allowed and will be represented by the corresponding number (that is, "2", "3" …and so on). If the element is a role name, then the maximum occurrence shall apply to the element indicated by the Data Type."*

Address Standard Conformance to Base Part 7.4.5. The XML model for each class and complex element shows the maximum occurrence for each of the elements and attributes that may comprise it.

7.9.3.7.4.6 Conformance to Base Part Subsection 7.4.6: Data type

Framework Base Part Section 7.4.6 reads in full: *"7.4.6: Data type. Specifies a set of distinct values for representing the elements (example: integer, real, CharacterString, DateTime, and Boolean). The data type attribute is also used to define stereotypes for entities*

and entity names for elements which are role names. These data types are generic types that do not infer an implementation." (Base Part 7.4.6, quoted in full)

Address Standard Conformance to Base Part 7.4.6. The data type for each element and attribute is specified in its description in the Content Part. Data types are named and defined in accordance with the Code List for Data Type (see Base Part section 7.8.2.2, Table 4), except for certain address reference system elements, which are geometric. All geometric data types are defined in the Open Geospatial Consortium's "OpenGIS(R) Geography Markup Language (GML)" version 3.1.1 (see Appendix A for a complete citation):

7.9.3.7.4.7 Conformance to Subsection Base Part 7.4.7: Domain

Framework Base Part Section 7.4.7 reads in full: *"7.4.7: Domain. For an entity, the domain indicates line numbers covered by the elements of that entity in the table.*

"For an element, the domain specifies the values allowed. "Unrestricted" indicates that no restrictions are placed on the data type of the element. Code lists provide a list of potential values, although additional values can be used. Enumerations provide a non-extensible list of potential values." (Base Part 7.4.7, quoted in full)

Address Standard Conformance to Base Part 7.4.7. Domain information for each element and attribute is provided in its description in the Content Part. For address classes, no domain information is provided, because no address class has any domains.

7.9.3.7.5 Conformance to Subsection Base Part 7.5: Metadata

Framework Base Part Section 7.5 reads in full:

"7.5.1 Requirement for metadata

"All datasets shall have metadata that conforms to at least the minimal set of mandatory elements of either ISO 19115, Geographic Information – Metadata, or FGDC-STD-001-1998, Content Standard for Digital Geospatial Metadata (revised June 1988). However, more extensive metadata should be provided.

"7.5.2 Associating metadata entry with data transfer

"The mechanism used to associate a structured metadata entry with a data transfer is not explicitly declared in the Framework Data Content Standard due to possible complex dependencies on either the structure of FGDC or ISO metadata being used. It is the intention of the standard to logically insert the appropriately structured metadata from either standard wherever the class attribute "metadata" occurs. The implementation of this capability may be specified in the implementation annexes as referenced to external metadata schemas in the appropriate implementation or programming environment."

Address Standard Conformance to Base Part 7.5. The address standard incorporates by reference, for address data files, the FGDC"s *Content Standard for Digital Geospatial*

Metadata (CSDGM)(FGDC 1998). The address standard extends the CSDGM by providing attributes for record-level address metadata.

7.9.3.7.6 Conformance to Subsection Base Part 7.6: Model integration

Framework Base Part Section 7.6 reads in full: *"7.6: Model integration. The dependencies among the models specified in the thematic parts of the standard are shown in Figure 2. In Figure 2, the parenthetical text (from Transportation) means that there is a UML package called "Transportation" in which all transportation constructs reside, including Transportation Base."*

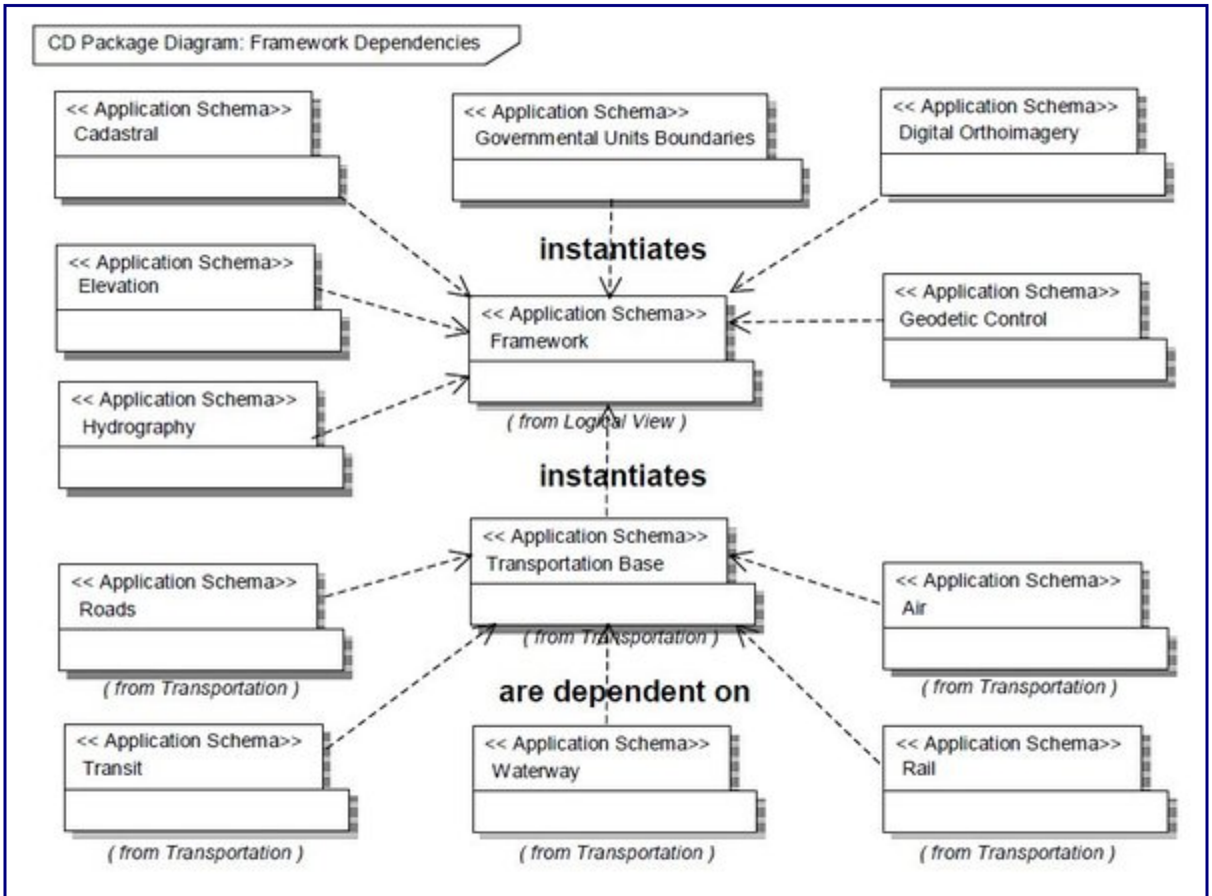


Figure 2 – Dependencies among the models specified in the thematic parts

Address Standard Conformance to Base Part 7.6. If the address standard were to be incorporated into the Framework Data Standard, it would be instantiated by and dependent on the Base Part. The address standard is also be related directly to the Cadastral and Transportation themes, as described in Sections 2.2 and 2.8 this Appendix.

7.9.3.7.7 Conformance to Subsection Base Part 7.7: Establishment of

identifiers

Framework Base Part Section 7.7 reads in full (omitting the footnote): *"7.7: Establishment of identifiers. Every UML class that represents a feature type includes attributes for identifier and an optional identifier authority. This construct can be used to distinguish between similar values in different datasets. Policies may be developed within a community for assigning namespaces and permanent identifiers to features and expressing equivalencies among features that have been assigned different namespaces and, therefore, different identifiers, which may be permanent. If there is no standard way to create and manage identifiers, users may develop their own schema and include its description in the dataset metadata."*

Address Standard Conformance to Base Part 7.7. The address standard defines an address attributes, Address ID, to serve as an address identifier, and another attribute, Address Authority, to serve as an authority identifier. Address ID may be implemented as a local ID or as a UUID.

7.9.3.7.8 Conformance to Base Part Subsection 7.8: Framework feature model and common classes

7.9.3.7.8.1 Conformance to Subsection Base Part 7.8.1: Introduction

Framework Base Part Section 7.8.1 reads in full: *"7.8.1: Introduction. The Framework Data Content Standard organizes information using the ISO General Feature Model [ISO 19109]. Features are abstractions of real-world phenomena or man-made constructs that typically have a persistent or assigned identity, such as a name or code, a location represented by a formalized geometry, and a set of other properties and relationships.*

"Each framework theme, represented by a part in the standard, documents one or more formal feature types using a logical information model (attributes, associations, conditionality) represented as class diagrams in UML. All feature types (see darker shaded classes in Figure 3) are denoted in UML using the stereotype <<Feature>>. All features in every part of the standard are subclasses of this common framework Feature and thus inherit its properties as shown in the diagram. Except for identifier, all properties are optional and most of them are repeatable.

"All classes stereotyped as <<Feature>> implement the Abstract class named "Feature" in the Base and inherit all of its properties. Likewise, any class stereotyped as <<FeatureCollection>> implements the Abstract class of the same name in the Base and inherits its property of "metadata". Inheritance is also shown through an italicized parent classname in the upper right corner of the child class.

"The Framework Data Content Standard supports the transfer of geographic data from one party to another. A group of features, known as a feature collection, would define a transfer. Metadata may be associated with the contents of the transfer, as is done now with FGDC "dataset-level" metadata. This feature collection may include features from one or more thematic parts of the standard, depending on the application and its requirements.

"Table 2 represents the information from Figure 3 in data dictionary format.

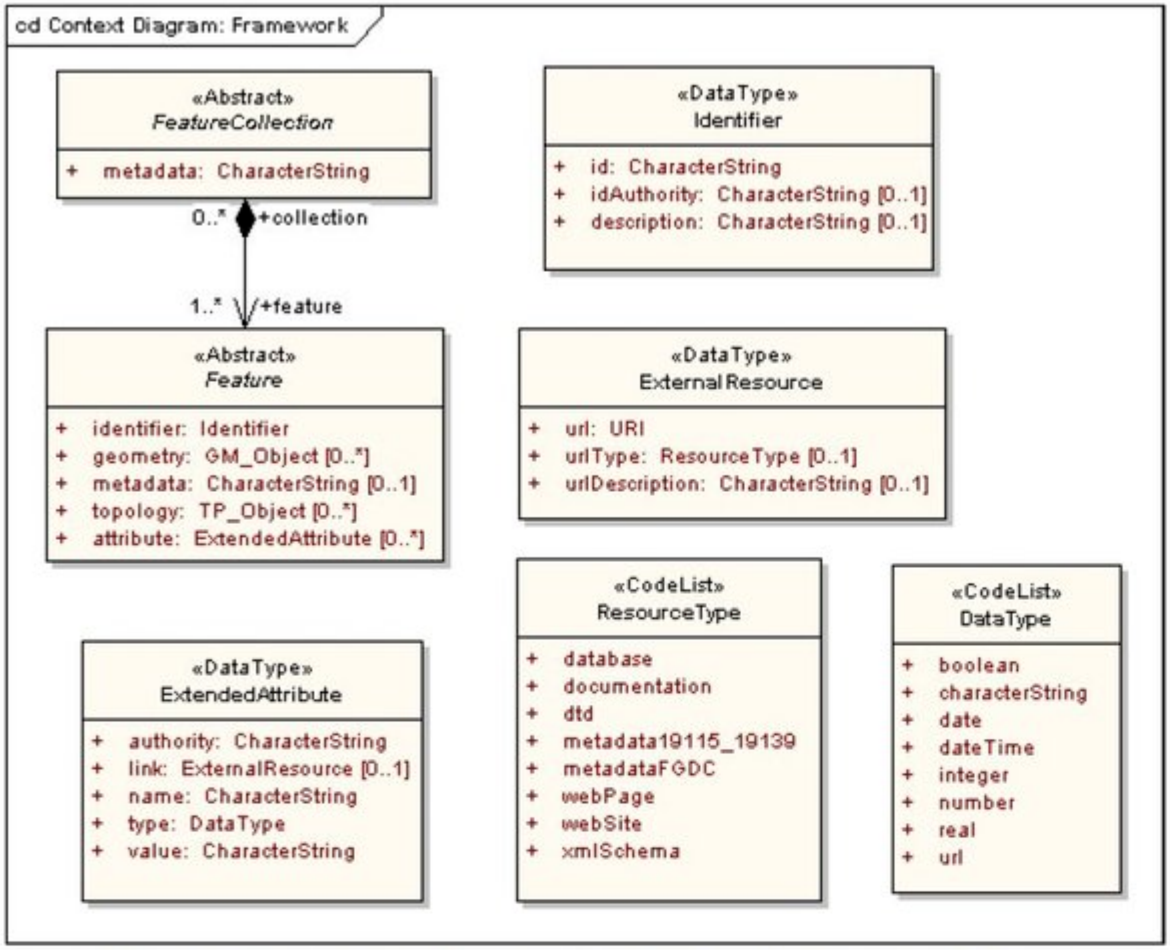


Figure 3 – The conceptual framework feature model and common classes

Table 2 – Description of common UML classes

Federal Geographic Data Committee
Geographic Information Framework Data Content Standard
Part 0: Base document

FGDC-STD-014.0-2008

Table 2 – Description of common UML classes

Line	Name/Role Name	Definition	Obligation/ Condition	Maximum Occurrence	Data Type	Domain
1	FeatureCollection	Aggregation of all features being transferred			<<Abstract>>	Lines 2-3
2	metadata	Structured or unstructured metadata as defined by the community of practice	M	1	CharacterString	May be text or structured metadata fragment or URI
3	Role name: feature	Features in the feature collection	M	*	<<Abstract>> Feature	Unrestricted
4	Feature	Abstraction of a real world phenomenon			<<Abstract>>	Lines 5-10
5	identifier	Label that uniquely identifies a feature, unique within the transfer	M	1	<<DataType>> Identifier	Unrestricted
6	geometry	Geometric representation of the feature	O	*	<<Type>> GM_Object	Defined in ISO 19107
7	metadata	Structured or unstructured metadata as defined by the community of practice	O	1	CharacterString	May be text or structured metadata fragment or URI
8	topology	Connectivity between one feature and another	O	*	<<Type>> TP_Object	Defined in ISO 19107
9	attribute	Producer-defined attribute for inclusion in transfer	O	*	<<DataType>> ExtendedAttribute	Unrestricted
10	Role name: collection	Collection of which this feature is a part	O	*	<<Abstract>> FeatureCollection	Unrestricted
11	Identifier	Construct to group an identifier with an authority and a description			<<DataType>>	Lines 12-14
12	id	Identification value (ID)	M	1	CharacterString	Unrestricted
13	idAuthority	Name of the issuing authority for the identifier	O	1	CharacterString	Unrestricted

Line	Name/Role Name	Definition	Obligation/ Condition	Maximum Occurrence	Data Type	Domain
14	description	Description or qualification of the identification value within the namespace of the authority	O	1	CharacterString	Unrestricted
15	ExtendedAttribute	Property that permits the identification and transport of an unofficial feature attribute			<<DataType>>	Lines 16-20
16	authority	Name of the organization responsible for the naming of this attribute	M	1	CharacterString	Unrestricted
17	link	Identification of an external resource that provides documentation of this attribute	O	1	<<DataType>> ExternalResource	Unrestricted
18	name	Name of the attribute being transferred	M	1	CharacterString	Unrestricted
19	type	Data type of the attribute being transferred	M	1	<<CodeList>> DataType	Unrestricted
20	value	Value of the attribute being transferred	M	1	CharacterString	Constrained by the valid companion data type
21	ExternalResource	Qualified link to a network accessible object			<<DataType>>	Lines 22-24
22	url	Network accessible resource in the form of a Uniform Resource Locator (URL) or valid Uniform Resource Identifier (URI)	M	1	URI	Unrestricted
23	urlType	Classification of the information content referenced by the URL	O	1	<<CodeList>> ResourceType	Unrestricted
24	urlDescription	Additional characteristics of the URL for advice or display	O	1	CharacterString	Unrestricted

"The extensibility mechanism shown in Figure 3 (ExtendedAttribute) allows for the description and transfer of additional ad hoc data content without requiring changes or extensions to the data schema. This repeatable structure may carry one or more additional attributes and their values for use in peer-to-peer transfer of unofficial feature properties. Any feature class may incorporate this reference to the ExtendedAttribute class. The link property of ExtendedAttribute expands to a triplet of elements associated with a Uniform Resource Locator (URL) for external documentation. Some ResourceTypes are shown as a code list to characterize the information content found at the referenced URL. For Transportation parts of this standard, events provide an alternative method of extending attributes when their values are not necessarily constant for the entire length of a feature."

Address Standard Conformance to Base Part 7.8.1. The address standard meets this requirement. Addresses meet the definition of "feature." The Classification Part defines an abstract Address class and defines subclasses of the feature "addresses", using a logical information model. The model is presented as both a UML model and an XSD. The standard supports both record-level and file-level metadata, and the Exchange Part provides a template for both monolithic and transactional exchanges. The data model is extensible.

7.9.3.7.8.2 Conformance to Base Part Subsection 7.8.2: Code lists

Framework Base Part Section 7.8.2 reads in full:

"7.8.2.1 ResourceType code list

" ResourceType is a CodeList of values for the attribute urlType.

"Table 3 – CodeList for ResourceType

Name	Definition
database	Collection of records where each record has the same structure of data elements
documentation	Resource file that describes usage of referenced URL
dtd	Schema expressed via a set of declarations written in Document Type Definition (DTD) language*
metadata 19115_19139	Metadata records formatted using structure from ISO 19115, Geographic information – Metadata, and ISO 19139, Geographic information – Metadata - XML schema implementation
metadataFGDC	Metadata records formatted using structure from a version of the FGDC Content Standard for Digital Geospatial Metadata
webPage	Resource on the World Wide Web usually in Hypertext Markup Language (HTML) format
webSite	Collection of Web pages that common to a particular domain name or subdomain on the World Wide Web
xmlSchema	Schema expressed using a version of the XML Schema World Wide Web Consortium (W3C) Recommendation

"7.8.2.2 DataType code list

" DataType is a CodeList of values for the attribute dataType.

"Table 4 – CodeList for DataType

Name	Definition
boolean	True or False
characterString	A CharacterString is an arbitrary-length sequence of characters including accents and special characters from repertoire of one of the adopted character sets
date	Values for year, month, and day
dateTime	A combination of year, month, and day and hour, minute, and second
integer	Any member of the set of positive whole numbers, negative whole numbers and zero
number	One of a series of symbols of unique meaning in a fixed order which may be

	derived by counting
real	Real numbers are all numbers that can be written as a possibly never repeating decimal fraction
url	Network accessible resource in the form of a Uniform Resource Identifier (URI)

Address Standard Conformance to Base Part 7.8.2. All data types in the data dictionary conform to the code list in Table 4, except for certain address reference system elements, which are geometric features. The address standard includes no resource types, so Table 3 does not apply to the address standard.

7.9.3.8 Conformance to Base Part Section 8: Encoding of framework data content

Framework Base Part Section 8 reads in full: *"8: Encoding of framework data content. To support data exchange, the parts of the Framework Data Content Standard may include informative annexes that provide guidance to implementers on the transformation of the UML information content into a specific encoding environment. These annexes not only document the context and environment of implementation and validation schema for the information content unique to a part of the standard, but also may include encoding or schema representation of heterogeneous collections of features from multiple themes. Because the standard includes a single UML model of all themes that are exposed progressively through a series of limited diagrams in the context of a theme, it represents an integrated set of classes for all framework data."*

Address Standard Conformance to Base Part 8. The address standard provides both a UML model and an XSD. The XSD provides guidance on the transformation of address information into a specific encoding environment. The Content and Classification parts of the address standard provide XML models for each class, element, and attribute defined in the standard. The Exchange part of the standard integrates the XML element, attribute, and class models into a single XSD. The XSD provides complete, open, standard XML data exchange templates for both monolithic and transactional data exchanges. For validation tests, similar guidance is provided by inclusion of complete SQL pseudocode in each test defined in the Data Quality Part of the address standard.