

Postal Address Template Description Language (PATDL): A Contribution To An International Postal Addressing Standard

1.1 In January 2001, the Universal Postal Union (UPU) Standards Board (SB) passed two resolutions pertaining to the development of an international postal addressing standard. These resolutions have Status P, corresponding to a work proposal that must be more fully defined in order to move forward to the next status level.

1.2 The first resolution calls for the UPU POST*Code team to “identify address and other related elements and apply standard definitions to these elements to facilitate the collection and exchange of International Name and Address Information”.

1.3 In the body of the resolution the goal of “automatic formatting of addresses” is described. Some of the benefits of the project are briefly mentioned. The statement of scope calls for identifying and defining name and address elements. It next proposes an “element by element order of presentation for each country”, listing optional and mandatory elements. It then proposes that “address element abbreviations and rendition instructions by country” and “preferred and acceptable character sets” be specified.

1.4 The second resolution was sponsored by the USPS and the Address Management Project Team of the UPU Direct Mail Advisory Board (DMAB). It calls for developing an “EDI and XML message and procedure to facilitate the exchange and final presentation of International Name and Address Information.

1.5 This resolution calls for a “standard order of presentation or template for each country”, using “preferred and acceptable character sets”. This may take the form of “EDI and XML messages to provide standard protocols for data transmission”, but the resolution notes how EDI is evolving in the direction of XML. A goal is to allow “printing rules and rendition instructions to allow formatting of addresses when constraints exist on available space”. It is further proposed to develop a “reference implementation to illustrate the functions of the standard operations”. A technique is sought to evaluate address data sets “in terms of completeness and correctness when measured against external data files”, and to measure the proportion of addresses which meet quality standards.

1.6 The intention of this paper is to discuss criteria that are relevant to achieving the goals laid down in these two resolutions, to examine certain areas of progress in this effort, and to introduce the XML-based Postal Address Template Description Language (PATDL) that demonstrates how many of these goals might be met. The author is industry co-chair of the UPU DMAB Address Management Project Team, and the chair of the IDEAlliance (formerly Graphic Communications Association) Address Data Interchange Specification (ADIS) project. This paper was reviewed by Bob Michelson, Mike Garner and Mabel Grein of the USPS prior to submission. Bob Michelson chairs the UPU Direct Mail Advisory Board (DMAB), and is the postal co-chair of the UPU DMAB Address Management Project Team. Mike Garner chairs the UPU POST*Code Technical Committee and leads the USPS International Address Template Work Group.

Mabel Grein was a leading participant in developing the Data Interchange Standards Association (DISA) ASC X12 EDI Transaction Set 101 format for postal addresses, coordinates the International Address Element Code (IAEC) effort within the Electronic Commerce Code Management Association (ECCMA), and serves as an ADIS Editor. The work presented here is based on ADIS version 01-1, published in April 2001, with certain extensions, and supports both the ECCMA and ADIS codes, as well as direct utilization of address elements defined with descriptive names or tags without codes.

1.7 To define an international postal addressing standard, there are some formal and practical criteria that should be taken into account. The first step is undoubtedly to define a set of elements that is applicable regionally and to the extent possible, globally. This is in itself a major step forward. The efforts of the CEN TC 331, ECCMA, IDEAlliance, and others such as OASIS, coupled with the surveys of the POST*Code team have provided a crucial underpinning for the overall effort.

1.8 It is noteworthy that all of the parties involved in the UPU efforts on international address standardization have adopted the general view that defining elements, and the printing rules or templates that specify the patterns or orderings of these elements, is a prerequisite for a successful outcome. This represents a basic agreement on a common methodology, all the more remarkable for having been arrived at independently in different areas of the world. This common understanding can serve as the foundation for the development of what we refer to generically as address element technology.

1.9 All of the parties have also adopted the view that not only addresses but also names are the appropriate subject matter for a postal addressing standard. This is helpful since the formatting of names, while it can be quite difficult, can affect not just response rates to mailings but also the delivery of the mail piece itself as well as the efforts to keep track of individuals, families, and business entities that have relocated.

1.10 Currently, most of the name and address data used in the global mailing industry is not very well differentiated, and does not consist of address elements that are individually identified and separately stored. Sometimes it is stored in identified constructs that are composites of elements, such as lines, and often it is stored in unidentified rectangular arrays of lines, such as address blocks. The methodological consensus around an element-based approach that has emerged within the UPU efforts is impressive and salutary. But to have a viable international postal addressing standard, the transitional issues for legacy data must be taken into account.

1.11 In trying to define a complete set of elements there are some problems that have already arisen. Naturally different countries have different elements. But even some of the constructs or meaningful subgroups of elements are not the same. Further, what is one element in some countries can be several elements in other countries. When the elements are the same, the ordering can be different. All these complexities take time to sort out.

1.12 It is not easy to determine how many elements need to be separately identified. There is a danger of defining too few elements and simply assigning different types of

element content to the same abstract element definition. This can interfere with the need to validate the element content to ensure that the results will meet quality expectations. In general, if there is a distinct concept with a specific functional purpose and a defined set of values that is used in one country but not in others, it will be helpful to recognize it as a distinct element. This element can then be easily understood in its context, properly validated, and need not appear in the templates of other countries.

1.13 There are ongoing efforts to define address structures for electronic commerce that do not necessarily focus primarily on postal applications. Some of these efforts fall into the category of defining address lines rather than elements. This certainly provides for ease of use. But it may not be understood that the unanalyzed complexity within the address lines will place limits on complete understanding of the data and on practical matters such as elimination of duplicates, obtaining high levels of confirmation of completeness and correctness, and formatting the data under spatial constraints.

1.14 There is also a danger of identifying too many elements. The danger is that it becomes harder to define a set of templates that can organize the element information in order to consistently produce appropriate results. Whether a set of templates reflecting different address types exists for a given country, or whether there is a single template with some capability of choice between multiple address types incorporated within it, the decision procedure when dealing with legacy data can be indeterminate.

1.15 An example of defining elements beyond what is suitable for postal addressing standards is subdividing of secondary address identifiers, such as Apartment, Suite, Unit, Room, and the like, into dozens of separate elements for marketing purposes. This has a valid business purpose, but in cases of multiple elements of this kind being populated, the ordering and sequencing of these elements becomes almost impossible to reconstruct. Another example is defining of multiple elements with only a small number of known instances. This will cause difficulty in cases when the data is ambiguous and can be parsed in many different ways.

1.16 One point of a standard is to get as much subject matter as possible into the public domain rather than depending on proprietary technology for a workable implementation. The more that the standard can encompass, the better the chances that the eventual multiple implementations will be interoperable. There is always enough room left over for competition in the marketplace in the areas of efficiency and collateral applications.

1.17 To define elements and their associated templates would be a great step forward, but even this is not sufficient for an international addressing standard. In our view, the combined scope of both UPU Status P resolutions is necessary to achieve this goal. The first resolution does itself make reference to “abbreviations” and “rendition instructions”, but there is more detail in the second resolution on these topics. It is important that the first resolution specifies a goal of “automatic formatting of addresses”. This sums up in a single phrase the most far-reaching objective of address element technology. But it will require all the innovations described in both resolutions to attain this goal.

1.18 Elements with templates must be supplemented by such items as punctuation and constants in order to be sufficient to the task. The USPS ZIP+4 code, when displayed on a mailpiece, needs a hyphen between the first five digits and the last four digits. However, the last four digits may not be known, in which case the hyphen must be omitted. A fixed hyphen is a constant, but a conditional hyphen calls for rendition instructions.

1.19 The second resolution in particular calls for attention to be paid to the final presentation of the postal address on the mailpiece. This requires a robust set of rendition instructions to allow for deliverability to be preserved under conditions of constraints upon space. In our experience, final presentation of addresses varies from mailpiece to mailpiece in each country. It affects not just response rates and deliverability, but also postal carrier efficiency and the ecological issue of disposal of undeliverable mail. Undoubtedly, the postal services have a direct economic interest in reducing these costs.

1.20 It is fortunate that the same techniques that allow for rendition of the postal name and address components also allow for manipulation of mail production elements that must be imaged along with the name and address for bulk business mail in many countries. It is not proposed that these mailing production variables be included within the scope of an international postal addressing standard, but the ability to define and format them with the same techniques needed for rendition of the address elements is an important part of what will constitute a practical implementation. Without a full set of rendition instructions, not only will the physical address presentation be inconsistent, but also the standard would only support single piece mailings.

1.21 An important criterion for a set of templates is that all of the address elements that may appear on the mailpiece should be mentioned in one or more of the templates. There may be some utility variables defined, such as the date of performance of an address coding process, that do not appear on a mailpiece but are necessary for quality control and in some cases affect processing of mail and the presence or absence of other elements on the mailpiece. These utility variables, of course, would not need to appear in any template. Unless this condition is met, there will be elements defined without any guidance in the standard as to how to use them.

1.22 Eventually it will be necessary to have templates for each country, though in some cases several countries may be able to share the same templates. To develop a testable proposed standard it is not necessary to have developed templates for every country, and since the testing may show a need for revising the template structure, it may not even be advisable to risk doing the work twice. However, the proposed standard might be testable, and indeed be tested successfully for a number of countries, while not being at all adequate for others. This is due not so much to varying addresses as different alphabets, and complexities such as languages that are written from right to left versus left to right. By showing the feasibility of address element technology for solving a fairly wide range of situations, it will be easier to convince more participants to come forward.

1.23 Within a set of rendition instructions, there need to be some that work in the vertical dimension, that is, combining, dividing, and prioritizing the retention or

suppression of logical constructs corresponding to various types of address lines. Other rendition instructions work in the horizontal dimension, that is, abbreviating, shortening, and prioritizing the retention or suppression of elements within a logical construct representing one or more address lines.

1.24 The documentation of at least one major postal service, the USPS, supports favoring horizontal techniques, such as abbreviation, in preference to vertical techniques, such as combining and splitting lines. This can be reflected in the priorities assigned to various rendition instructions. Since in a generalized approach these priorities can be assigned differently, this does not imply a methodological inflexibility.

1.25 What is presented in this paper is a flexible approach allowing the user to determine the rendition priorities, including giving primary consideration to rendition instructions operating in one dimension or the other, as well as an interactive approach which can switch back and forth between dimensions. However, for a given set of differentiated rendition priorities, the solution procedure is straightforwardly sequential and deterministic. This implies that multiple software implementations should produce identical results. It is felt that this is an appropriate starting point that favors efficiency and consistency while supporting potential certification of the results.

1.26 It is interesting to note that the proposed template description language would support an alternate approach in which the requirement for differentiated rendition priorities is intentionally weakened. Specifically, this constraint could be loosened by allowing a subset of the rendition priorities to be set equal to each other. This would permit a more sophisticated solution procedure incorporating backtracking, with evaluation of intermediate results using some internal criteria, which would result in the solution procedure proceeding down a different path. This would, however, sacrifice some of the advantages of the current approach as just discussed.

1.27 Included among the specific rendition instructions defined in IDEAlliance ADIS 01-1 are rendition constants, background characters which can be overwritten, default delimiters and a means to bypass them, wrapping text upward or downward, splitting and combining of lines, substitution of one element for another, abbreviation, substitution by a first initial, shortening by elimination of vowels, repeated consonants, noise words, or other specified patterns, and in the last resort, truncation or elimination of elements. These are described in the ADIS specification document. The proposed PATDL allows for other sets of rendition instructions to be developed, and brought into play separately or together with the ADIS instructions.

1.28 While fitting the address elements into the available space, it is possible that content essential for meeting the criteria set by a postal service for a complete and correct address, or for practical deliverability, will be lost. The proposed template description language provides a means for noting that this has occurred and for an implementation to tabulate the frequency of such occurrences. This is potentially a valuable feature for the postal services that can be linked to postal rate incentives or differentiated treatment during postal operations. However, it depends upon the prior knowledge that the address

was complete and correct before the rendition process has begun. This in turn implies the existence of a database or some other means to validate complete and correct addresses. Perhaps the advent of technology to verify completeness and correctness on the mailpiece will give further impetus to the development of delivery point databases by the postal services. Supporting the development and fostering the use of such databases is one of the goals of the UPU DMAB Address Management Project Team.

1.29 The criterion we postulated for the completeness of a set of templates in relation to a set of address elements states that for those elements that may appear in valid postal there must be at least one template within the set of templates in which the element can be found. This can be most easily satisfied if there are several templates for a given country, with each representing an address type, such as postal box or street address. It is less natural to define a single template that incorporates multiple address types because this requires either conditional logic or some mechanism that accomplishes a similar purpose. However, there is a drawback to having a template for each address type, namely that there will remain a need for specifying which template to use.

1.30 One way to handle this is for the user to bear the burden of deciding which address type to use. In some cases, such a responsibility seems unavoidable. For example, if the database contains both a street address and a postal box, it may be necessary to use the postal box for certain mailing purposes and the street address at other times. Only the user presumably knows the purpose at hand and therefore the user must have a way to express a preference. Given that, it may seem natural for the user to be required always to specify which of a set of templates is to be used. However, that is a burden perhaps of sufficient magnitude to slow down the adoption of address element technology.

1.31 In the proposed template description language, this issue is approached through defining user preferences and trigger conditions. In IDEAlliance ADIS 01-1, there are some user preferences, but not yet any trigger conditions, so this represents an extension of that specification. Trigger conditions intentionally fall short of full conditional logic. As stated in the ADIS 01-1 specification, we ought to be reluctant to implement full conditional logic, with if-then-else constructs, explicit and/or operations, arithmetic operators, and so forth, since these are available through other means such as scripting and programming languages. On the other hand, to define a set of trigger conditions that is just sufficient to manipulate the address elements and rendition instructions provides several advantages. First, it makes it realistic to construct a single template for each country. Second, that template will be primarily data driven, which constitutes significant progress toward the goal of “automatic formatting of addresses” set in the first of the two UPU Standards Board Status P resolutions. Third, the trigger conditions can reference a full range of user preferences to achieve a wide range of outcomes. Fourth, combining these features together makes it less necessary for users to modify the primary template, since by changing the user preferences elements and lines can be made to appear or disappear at will and rendition instructions can be activated and suppressed without changing the underlying template.

1.32 Among the trigger conditions defined in PATDL are the ability to select among a set of line constructs, to test the value of an element, and to test whether an element or a set of elements is populated. This is a relatively small set of trigger conditions, but it appears to be adequate to handle most of the issues with North American addresses. It is noteworthy that it allows for a certain level of if-then-else logic, and/or operations, and arithmetic operators, while doing so implicitly rather than explicitly.

1.33 We intend to demonstrate that an implicit approach to conditional logic using a minimal set of trigger conditions will be sufficient to the task by providing a reference implementation of ADIS that supports PATDL. This will continue to evolve as the work on the UPU international address standard proceeds, so as to support the developing standard, within the limits of available time and resources. The reference implementation will use the ADIS 01-1 feature set as a starting point, and will also from the outset support ECCMA codes. It will add in the CEN address elements, the extensions to ADIS in the form of trigger conditions and additional user preferences, and the POST*Code templates and CEN printing rules as these are codified. Initially this reference implementation will be developed as a prototype using technology not well suited for standardization. The author hopes that by working together with other participants a reference implementation can be developed using technology that is regarded as suitable for incorporation within a proposed standard. This standardized reference implementation would not need to be efficient but would need to be definitive, so as to serve as a benchmark for potential compliance testing, if the POST*Code group and eventually the UPU Standards Board considers this to be warranted.

1.34 There are other features of the proposed template description language that have not yet been mentioned but which will be described in what follows. One such feature is the support of diverse code sets and address element naming conventions or tags on an equal basis. To use PATDL, there must be some definitive way to refer to any particular address element, and there may be other supplementary ways to describe or locate that same element. The author holds the view that a code set is quite helpful because it is a condensed notation which is easily validated and language independent, but the proposal will fully support using natural language element names or tags as identifiers. Of course the languages, as well as the character sets, for both the element names and the element content need to be specified.

1.35 Another essential feature is the support of external data tables. If we want to abbreviate a name or address element, this can be done by providing a capability to carry the abbreviation in XML as an attribute associated with the element. But there is also a need to allow a reference to an external table to accomplish the same purpose. Such an external table must be defined in terms of its data structure and how it is to be found, whether through a URL or URI reference or a file located on an accessible resource. For a production implementation, a fairly large number of such tables may be required.

1.36 What about the issue of optional and mandatory elements? We find it difficult to define mandatory elements for USPS addresses outside the context of a known address type or mailing environment. For example, the name can be omitted for certain types

of mail. The ZIP code is not mandatory for single piece First Class mail. The city or town name could be considered a necessity, but unless validated, it may not match a database entry. No single field in the street address can be considered mandatory since there are postal box addresses as well as other alternative formats. We are concerned that if only a few elements can be considered mandatory, then including all mandatory elements may not provide a guarantee of deliverability. Instead, a more robust approach is to validate the address against a database of delivery points. Since this is only possible in certain industrialized countries, other situations may benefit from designation of certain elements as required and we have incorporated an option to specify this. Potentially, tools could be developed to validate the presence and in some cases the content of mandatory elements for countries where there is no delivery point database. The UPU Universal POST*CODE DataBase product could be utilized in order to do this.

1.37 We would like to note the limits of compatibility between the line-based postal address technology prevalent today and the element-based address technology that we believe will be prevalent in the future. Every current effort to define address element technology that has come to the attention of the author provides a transitional mechanism by allowing for line-based data to be carried within an element-based approach. In this way, legacy data can be brought forward, and hybrid data, in which either the name or the address, but not both, is maintained in a line-based format, can also be supported. However, the rendition instructions, particularly those operating in the horizontal dimension, are designed to operate upon element-based data, in order to enable control over final presentation, preserve deliverability under constraints, and allow for quality measurement. If line-based horizontal rendition instructions were developed, they would have to parse the data into its components, implicitly operating in an element-based manner. By defining the components of an international address standard in the way that the two UPU Standards Board resolutions have done, including elements, templates and rendition instructions, the means are provided for overcoming the inertia of line-based data storage and establishing a new element-based paradigm. This will benefit mailers, the mailing industry, and the postal services.

1.38 The proposed Postal Address Template Description Language (PATDL) is currently expressed as an XML Document Type Definition (DTD). The core features of PATDL, such as the description of templates in terms of elements, the set of rendition instructions, and the trigger conditions, are presented in sufficient detail to support the development of a prototype reference implementation. Other features, such as the specification of character sets and external tables, are indicated but not developed. The XML DTD can be modified to create an XML schema. This would allow for more strict validation, particularly if the schema representation includes supplementary technology to express joint constraints.

1.39 By proposing PATDL for review and discussion, there is no intention to preclude other forms of representation of template structure within an eventual UPU standard. For example, it is well known that some conditions cannot be expressed within an XML DTD, though the XML schema language significantly improves this situation. Within the CEN, there has been some discussion of using ASN1 to describe template structure, and Holger

Wandt has spoken of developing a grammar for templates. In the experience of the author, developing multiple representations of data structures often leads to deeper insights. Further, this activity could reveal an underlying commonality among what may seem initially to be different approaches. However, since the second UPU Standards Board Status P resolution does call for the development of EDI and XML messages for the transmission of international name and address data, it is natural to use the same tools to define the structures used to constrain the data and to determine its final presentation.

1.40 What are the possible further steps to be taken? We have proposed PATDL in the hope that it will be subject to thorough evaluation by all interested parties and eventual determination of its suitability as a starting point for additional work. Work on the prototype reference implementation is already under way. In the USPS environment, when the prototype implementation has been completed, we plan to use it to test addresses that exceed some defined number of characters in order to see how well it does at providing shorter versions for final presentation. We also want to test it against a randomly selected subset of USPS delivery points to see how it handles various address types. Further, with the development of templates for one or more other countries, we would like to test it to determine its adequacy in that context.

1.41 The Postal Address Template Description Language (PATDL) XML DTD is included as Appendix I to this paper. Preceding the main body of the DTD are comments that explain a number of technical matters not documented elsewhere, such as the syntax of the trigger conditions and the rules governing how they are interpreted. Appendix II is a template originally developed by Mabel Grein of USPS to illustrate how street addresses are formatted. Her template has been translated into PATDL format using ECCMA codes and ADIS rendition instructions. Appendix III is an annotated example intended to demonstrate many of the features of PATDL. Through close examination of Appendix III, the reader may be in a better position to form an opinion with respect to the capability of PATDL to represent all address types for one country in a single template.

Joe Lubenow
Industry Co-Chair
UPU DMAB Address Management Project Team
February 20, 2002

APPENDIX I.

<!--DTD generated by XML Spy v3.5 (http://www.xmlspy.com) by Joe Lubenow-->

<!--

This DTD defines version 1.0 of the Postal Address Template Description Language (PATDL).

A template includes identifying information and has content which includes user preferences, trigger conditions, line constructs, lines, segments, elements, and rendition instructions.

User preferences are needed to resolve issues of selection of line constructs, lines, elements, and rendition instructions that are variable depending on the job to be performed.

Trigger conditions determine the inclusion or exclusion of lines and line constructs from the initial rendition, which is the starting point for the rendition instructions to create the final presentation.

Line constructs are logical lines which can become multiple physical lines or parts of physical lines during the rendition process.

Elements are parts of addresses as defined in an address standardization process.

Rendition instructions accomplish the final formatting of the address. There are rendition operators which utilize constants, and rendition commands which manipulate elements, lines, and line constructs.

An element identifier uniquely specifies an element even when multiple sources of elements are being used. If there are multiple sources, a prefix is used with the identifier to guarantee uniqueness.

An element definer documents what an element represents and is not used to access the element.

An element descriptor provides a semantic clue or an alternate description of the meaning of an element and can also serve as an alternate identifier.

A character set is a preferred or allowable set of characters that are eligible to be used in the output of a particular rendition process.

A default delimiter will be used to separate elements and constants unless overridden by a particular type of rendition instruction.

A quality threshold is crossed when, in the process of rendition, essential data has been eliminated, or required data is determined not to be present, that can compromise deliverability of the address as determined by the user or a postal service.

An external entity is a table of elements, data used in the rendition process, or other data relevant to the process.

A template number includes an ISO country code (ALPHA-2) as well as other identifying information specified by the user.

The user preferences include several designed to indicate which of a set of line constructs is preferred when more than one is populated for a given address instance, and also to permit the elimination of particular line constructs from consideration in the current rendition process.

Trigger conditions show whether an element or set of elements is populated, or whether an element, or a user preference, has a certain value. There is also a block construct, lineSelect, to denote the scope of a set of trigger conditions, and a defaultCase trigger condition to guarantee that one of a set of conditions within a block will be satisfied. Within a trigger condition, a semicolon indicates a separation between two arguments, and a comma or hyphen indicates a sequence or set of elements within a single argument. Semicolons and commas are followed by spaces, while a hyphen is not.

Trigger conditions can be followed by one or more line constructs, and if the conditions are satisfied, the line construct is selected into the initial rendition. The line construct with all of its elements and operators must be defined in the lineData section. If no line construct is specified, none will be included. If a line construct is selected but user preferences indicate that it is to be suppressed, it is not brought forward. Whenever one set of trigger conditions within a lineSelect block has been satisfied, none of the others are evaluated.

The isPopulated trigger condition can have multiple arguments and is satisfied only if all arguments, including at least one of a set of elements within an argument, meet the condition of being populated. The hasValue trigger condition can test whether an element has a value, or whether an element has the same value as another element. It has two arguments which are separated by a semicolon. Implicitly, the test is for equality. The hasPreference trigger condition tests whether a user preference has been indicated matching a value or one of a series of values. These values are separated by commas. The test is for equality. The defaultCase trigger condition can be used to ensure that one of a set of trigger conditions is satisfied. It has no arguments. Typically it is used after other trigger conditions, but it can be followed by other trigger conditions if the default case is itself complex.

Implicitly, an or-function can be represented by repeating trigger conditions, each with its own line construct result, and an and-function can be represented by consecutive trigger conditions prior to a line construct result. A test for equality can be accomplished by the hasValue or hasPreference trigger conditions, and a test for inequality can be done by testing for equality, selecting nothing, and then using a defaultCase trigger condition.

Line data includes line names, line numbers, line priorities, rendition instructions for merging lines including specifying priority, order of the line constructs, and location of the merged constructs, optional segmentation of lines, rendition commands, and elements. The line names, line numbers, and line priorities are mandatory. Line names and numbers are both needed since the same line construct may appear in different places, and several constructs may be candidates for the same line number. The line numbers are subject to change during the rendition process. Line priorities are mandatory to establish a decision procedure. Merging lines is part of the rendition process, and segments are used when lines might be split by moving a segment either up or down.

Rendition commands and elements are defined in the appropriate source documents. Multiple sources can be used together as long as they are differentiated.

Element data includes one or more sets of an element identifier, definer, and descriptor, an indicator whether content is required, left or right field justification, field starting position and length, and one or more rendition commands applying to the element. Element substitution is accomplished by indicating multiple elements. If the first is not populated, the next will be tried, and so on, though once one of the set has been included, the remainder of the set is not checked. If content is required and not present, the address is considered to have fallen below the quality threshold.

-->

<!ELEMENT patdl.xml (identifier, contentDefinition+)>

<!ELEMENT identifier (elementIdentifier+, elementDefiner+, elementDescriptor+, renditionInstruction+, characterSet, defaultDelimiter, qualityThreshold, externalEntityData*)>

```

<!ELEMENT elementIdentifier (type, prefix?, language?, system, version, source)>
<!ELEMENT type (#PCDATA)>
<!ELEMENT prefix (#PCDATA)>
<!ELEMENT language (#PCDATA)>
<!ELEMENT system (#PCDATA)>
<!ELEMENT version (#PCDATA)>
<!ELEMENT source (#PCDATA)>
<!ELEMENT elementDefiner (type, prefix?, language, system?, version?, source)>
<!ELEMENT elementDescriptor (type, prefix?, language, system, version, source)>
<!ELEMENT renditionInstruction (type, prefix?, language, system, version, source)>
<!ELEMENT characterSet (#PCDATA)>
<!ELEMENT defaultDelimiter (#PCDATA)>
<!ELEMENT qualityThreshold (#PCDATA)>
<!ELEMENT externalEntityData (entityType, entityId, entitySystem, entityVersion, entityDataStructure, entityPriority, entityLocation)>
<!ELEMENT entityType (#PCDATA)>
<!ELEMENT entityId (#PCDATA)>
<!ELEMENT entitySystem (#PCDATA)>
<!ELEMENT entityVersion (#PCDATA)>
<!ELEMENT entityDataStructure (#PCDATA)>
<!ELEMENT entityPriority (#PCDATA)>
<!ELEMENT entityLocation (#PCDATA)>
<!ELEMENT contentDefinition (templateName?, templateNumber, (userPreferences, triggerConditions)*, lineData+)>
<!ELEMENT templateName (#PCDATA)>
<!ELEMENT templateNumber (countryCode, userId, templateType, templateSeqNum)>
<!ELEMENT countryCode (#PCDATA)>
<!ELEMENT userId (#PCDATA)>
<!ELEMENT templateType (#PCDATA)>
<!ELEMENT templateSeqNum (#PCDATA)>
<!ELEMENT userPreferences (titleVsAddlName, maileeVsAltRecipient, streetVsPostOfcBox, suppressLine*)>
<!ELEMENT titleVsAddlName EMPTY>
<!ATTLIST titleVsAddlName
  userValue (TITLE-PREF | NAME-PREF) #REQUIRED
>
<!ELEMENT maileeVsAltRecipient EMPTY>
<!ATTLIST maileeVsAltRecipient
  userValue (MAILEE-PREF | ALT-PREF | MAILEE-ONLY | ALT-ONLY) #REQUIRED
>
<!ELEMENT streetVsPostOfcBox EMPTY>
<!ATTLIST streetVsPostOfcBox
  userValue (STREET-PREF | PO-BOX-PREF) #REQUIRED
>
<!ELEMENT suppressLine (lineName, lineNumber)?>
<!ELEMENT triggerConditions (lineSelect)*>
<!ELEMENT lineSelect (((isPopulated | hasValue | hasPreference)*, defaultCase?), (lineName, lineNumber)+)>
<!ELEMENT isPopulated (#PCDATA)>
<!ELEMENT hasValue (#PCDATA)>
<!ELEMENT hasPreference (#PCDATA)>
<!ELEMENT defaultCase EMPTY>
<!ELEMENT lineData (lineName, lineNumber, linePriority, lineMerge?, lineMergeOrder?, lineMergeLoc?, (lineSegment?, (renditionOperator | elementData))*>
<!ELEMENT lineName (#PCDATA)>
<!ELEMENT lineNumber (#PCDATA)>
<!ELEMENT linePriority (#PCDATA)>
<!ELEMENT lineMerge (#PCDATA)>
<!ELEMENT lineMergeOrder (#PCDATA)>
<!ELEMENT lineMergeLoc (#PCDATA)>
<!ELEMENT lineSegment (segmentSequence, renditionCommand?)>
<!ELEMENT segmentSequence (#PCDATA)>
<!ELEMENT renditionCommand (cmdId, cmdPriority)>
<!ELEMENT cmdId (#PCDATA)>
<!ELEMENT cmdPriority (#PCDATA)>
<!ELEMENT renditionOperator (operatorId, fldJustify, posStart, posLength, fldText?)>
<!ELEMENT operatorId (#PCDATA)>
<!ELEMENT fldJustify (#PCDATA)>
<!ELEMENT posStart (#PCDATA)>
<!ELEMENT posLength (#PCDATA)>
<!ELEMENT fldText (#PCDATA)>
<!ELEMENT elementData ((elementId, elementDef?, elementDesc?)+, contentRequired?, fldJustify, posStart, posLength, renditionCommand*)>
<!ELEMENT elementId (#PCDATA)>
<!ELEMENT elementDef (#PCDATA)>
<!ELEMENT elementDesc (#PCDATA)>
<!ELEMENT contentRequired (#PCDATA)>

```

APPENDIX II.

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v3.5 (http://www.xmlspy.com) by Joe Lubenow (private) -->
<!--
This is the PATDL version of the template designed by Mabel Grein of USPS to represent
the street address format using ECCMA codes and ADIS rendition instructions
It has been validated using the Postal Address Template Description Language (PATDL) DTD
The file name is straddr.xml
-->
<!DOCTYPE patdl.xml SYSTEM "patdl.dtd">
<patdl.xml>
  <identifier>
    <elementIdentifier>
      <type>code</type>
      <prefix>E</prefix>
      <system>IAEC</system>
      <version/>
      <source>ECCMA</source>
    </elementIdentifier>
    <elementDefiner>
      <type>descriptive</type>
      <language>English</language>
      <source>ECCMA</source>
    </elementDefiner>
    <elementDescriptor>
      <type/>
      <language/>
      <system/>
      <version/>
      <source/>
    </elementDescriptor>
    <renditionInstruction>
      <type>mnemonic</type>
      <language>English</language>
      <system>ADIS</system>
      <version>01-1</version>
      <source>IDEAlliance</source>
    </renditionInstruction>
    <characterSet>ANSI</characterSet>
    <defaultDelimiter>' '</defaultDelimiter>
    <qualityThreshold/>
  </identifier>
  <contentDefinition>
    <templateNumber>
      <countryCode>US</countryCode>
      <userId>USPS</userId>
      <templateType>ADR</templateType>
      <templateSeqNum>001</templateSeqNum>
    </templateNumber>
    <lineData>
      <lineName>mailstop</lineName>
      <lineNumber>001</lineNumber>
      <linePriority>008</linePriority>
      <elementData>
        <elementId>E10.008</elementId>
        <elementDesc>mailstop</elementDesc>
        <fldJustify>L</fldJustify>
        <posStart>001</posStart>
        <posLength>***</posLength>
      </elementData>
    </lineData>
    <lineData>
      <lineName>name</lineName>
      <lineNumber>002</lineNumber>
      <linePriority>003</linePriority>
      <elementData>
        <elementId>E11.002</elementId>
        <elementDesc>primary.honorific</elementDesc>
        <fldJustify>L</fldJustify>
        <posStart>001</posStart>
        <posLength>***</posLength>
      </elementData>
      <elementData>
        <elementId>E11.010</elementId>
        <elementDesc>primary.preferred</elementDesc>
        <elementId>E11.003</elementId>
        <elementDesc>primary.given</elementDesc>
        <fldJustify>L</fldJustify>
        <posStart>***</posStart>
      </elementData>
    </lineData>
  </contentDefinition>
</patdl.xml>

```

```
<posLength>***</posLength>
</elementData>
<elementData>
  <elementId>E11.004</elementId>
  <elementDesc>primary.first.middle.name</elementDesc>
  <fldJustify>L</fldJustify>
  <posStart>***</posStart>
  <posLength>***</posLength>
</elementData>
<elementData>
  <elementId>E11.005</elementId>
  <elementDesc>primary.second.middle.name</elementDesc>
  <fldJustify>L</fldJustify>
  <posStart>***</posStart>
  <posLength>***</posLength>
</elementData>
<elementData>
  <elementId>E11.006</elementId>
  <elementDesc>primary.family</elementDesc>
  <fldJustify>L</fldJustify>
  <posStart>***</posStart>
  <posLength>***</posLength>
</elementData>
<elementData>
  <elementId>E11.007</elementId>
  <elementDesc>primary.suffix</elementDesc>
  <fldJustify>L</fldJustify>
  <posStart>***</posStart>
  <posLength>***</posLength>
</elementData>
</lineData>
<lineData>
  <lineName>title</lineName>
  <lineNumber>003</lineNumber>
  <linePriority>005</linePriority>
  <elementData>
    <elementId>E11.011</elementId>
    <elementDesc>title</elementDesc>
    <fldJustify>L</fldJustify>
    <posStart>001</posStart>
    <posLength>***</posLength>
  </elementData>
</lineData>
<lineData>
  <lineName>deptdiv</lineName>
  <lineNumber>004</lineNumber>
  <linePriority>008</linePriority>
  <elementData>
    <elementId>E10.006</elementId>
    <elementDesc>department</elementDesc>
    <fldJustify>L</fldJustify>
    <posStart>001</posStart>
    <posLength>***</posLength>
  </elementData>
  <elementData>
    <elementId>E10.007</elementId>
    <elementDesc>division</elementDesc>
    <fldJustify>L</fldJustify>
    <posStart>***</posStart>
    <posLength>***</posLength>
  </elementData>
</lineData>
<lineData>
  <lineName>organization</lineName>
  <lineNumber>005</lineNumber>
  <linePriority>006</linePriority>
  <elementData>
    <elementId>E11.001</elementId>
    <elementDesc>organization</elementDesc>
    <fldJustify>L</fldJustify>
    <posStart>001</posStart>
    <posLength>***</posLength>
  </elementData>
</lineData>
<lineData>
  <lineName>urbanization</lineName>
  <lineNumber>006</lineNumber>
  <linePriority>004</linePriority>
```

```

<elementData>
  <elementId>E10.014</elementId>
  <elementDesc>urbanization</elementDesc>
  <fldJustify>L</fldJustify>
  <posStart>001</posStart>
  <posLength>***</posLength>
</elementData>
</lineData>
<lineData>
  <lineName>street1</lineName>
  <lineNumber>007</lineNumber>
  <linePriority>001</linePriority>
  <elementData>
    <elementId>E10.002</elementId>
    <elementDesc>house.number</elementDesc>
    <fldJustify>L</fldJustify>
    <posStart>001</posStart>
    <posLength>***</posLength>
  </elementData>
  <elementData>
    <elementId>E10.045</elementId>
    <elementDesc>prefix.street.direction</elementDesc>
    <fldJustify>L</fldJustify>
    <posStart>***</posStart>
    <posLength>***</posLength>
  </elementData>
  <elementData>
    <elementId>E10.009</elementId>
    <elementDesc>street.name</elementDesc>
    <fldJustify>L</fldJustify>
    <posStart>***</posStart>
    <posLength>***</posLength>
  </elementData>
  <elementData>
    <elementId>E10.010</elementId>
    <elementDesc>street.name.suffix</elementDesc>
    <fldJustify>L</fldJustify>
    <posStart>***</posStart>
    <posLength>***</posLength>
  </elementData>
  <elementData>
    <elementId>E10.011</elementId>
    <elementDesc>suffix.street.direction</elementDesc>
    <fldJustify>L</fldJustify>
    <posStart>***</posStart>
    <posLength>***</posLength>
  </elementData>
  <elementData>
    <elementId>E10.012</elementId>
    <elementDesc>secondary.unit.identifier</elementDesc>
    <fldJustify>L</fldJustify>
    <posStart>***</posStart>
    <posLength>***</posLength>
  </elementData>
  <elementData>
    <elementId>E10.013</elementId>
    <elementDesc>secondary.unit.number</elementDesc>
    <fldJustify>L</fldJustify>
    <posStart>***</posStart>
    <posLength>***</posLength>
  </elementData>
</lineData>
<lineData>
  <lineName>lastline</lineName>
  <lineNumber>008</lineNumber>
  <linePriority>002</linePriority>
  <elementData>
    <elementId>E10.018</elementId>
    <elementDesc>city.name</elementDesc>
    <fldJustify>L</fldJustify>
    <posStart>001</posStart>
    <posLength>***</posLength>
  </elementData>
  <elementData>
    <elementId>E10.020</elementId>
    <elementDesc>state.postal.abbreviation</elementDesc>
    <fldJustify>L</fldJustify>
    <posStart>***</posStart>
  </elementData>

```



```
<posLength>***</posLength>
</elementData>
<elementData>
  <elementId>E10.025</elementId>
  <elementDesc>zip.code</elementDesc>
  <fldJustify>L</fldJustify>
  <posStart>***</posStart>
  <posLength>***</posLength>
</elementData>
<renditionOperator>
  <operatorId>CONCAT</operatorId>
  <fldJustify>L</fldJustify>
  <posStart>***</posStart>
  <posLength>***</posLength>
  <fldText>'-'</fldText>
</renditionOperator>
<elementData>
  <elementId>E10.026</elementId>
  <elementDesc>zip.code.addendum</elementDesc>
  <fldJustify>L</fldJustify>
  <posStart>***</posStart>
  <posLength>***</posLength>
</elementData>
</lineData>
</contentDefinition>
</patdl.xml>
```

APPENDIX III.

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v3.5 (http://www.xmlspy.com) by Joe Lubenow (private) -->
<!--
  This is an annotated example designed to show many of the features of PATDL. It uses ADIS and ECCMA element codes and ADIS rendition instructions. There
  is an example of natural language element names with a CEN TC 331 element definer. It has been validated using the Postal Address Template Description Language
  (PATDL) DTD. The file name is patdl.xml.
-->
<!DOCTYPE patdl.xml SYSTEM "patdl.dtd">
<patdl.xml>
  <identifier>
    <!--
      ADIS codes designated with an A prefix to distinguish them from ECCMA codes.
    -->
    <elementIdentifier>
      <type>code</type>
      <prefix>A</prefix>
      <system>ADIS</system>
      <version>01-1</version>
      <source>IDEAlliance</source>
    </elementIdentifier>
    <!--
      ECCMA codes with an E prefix may be intermixed in the data.
    -->
    <elementIdentifier>
      <type>code</type>
      <prefix>E</prefix>
      <language>English</language>
      <system>IAEC</system>
      <version/>
      <source>ECCMA</source>
    </elementIdentifier>
    <!--
      This example shows that element identifiers need not be codes.
    -->
    <elementIdentifier>
      <type>descriptive</type>
      <prefix>FR</prefix>
      <language>French</language>
      <system/>
      <version/>
      <source/>
    </elementIdentifier>
    <!--
      An element definer is used not to reference the element but to document it.
    -->
    <elementDefiner>
      <type>descriptive</type>
      <language>English</language>
      <source>CEN TC 331</source>
    </elementDefiner>
    <!--
      An element descriptor is used as a synonym for the element identifier which may have meaning to the user, and also can provide a secondary means to
      reference the element.
    -->
    <elementDescriptor>
      <type>mnemonic</type>
      <language>English</language>
      <system>ADIS</system>
      <version>01-1</version>
      <source>IDEAlliance</source>
    </elementDescriptor>
    <!--
      This example uses ADIS rendition instructions without a prefix since no other set of rendition instructions has been declared.
    -->
    <renditionInstruction>
      <type>mnemonic</type>
      <prefix/>
      <language>English</language>
      <system>ADIS</system>
      <version>01-1</version>
      <source>IDEAlliance</source>
    </renditionInstruction>
    <!--
      Although XML incorporates UNICODE, this example is restricted to the ANSI character set.
    -->
    <characterSet>ANSI</characterSet>
    <!--
      A single space is assumed to separate elements unless otherwise specified during rendition.
    -->
  </patdl.xml>

```

```
-->
<defaultDelimiter>' '</defaultDelimiter>
<!--
```

Any rendition priority over 90, or any use of default truncation or default elimination of an element will violate the quality threshold constraint.

```
-->
<qualityThreshold>90</qualityThreshold>
<!--
```

External entities can include tables of personal or firm names, street names, etc.

```
-->
<externalEntityData>
  <entityType/>
  <entityId/>
  <entitySystem/>
  <entityVersion/>
  <entityDataStructure/>
  <entityPriority/>
  <entityLocation/>
</externalEntityData>
</identifier>
```

```
<!--
-->
<contentDefinition>
  <!--
```

A template number includes an ISO country code as its first component.

```
-->
<templateNumber>
  <countryCode>US</countryCode>
  <userId>EXIL</userId>
  <templateType>ADR</templateType>
  <templateSeqNum>001</templateSeqNum>
</templateNumber>
<!--
```

These user preferences help resolve which line constructs should be brought forward during the early stages of rendition.

```
-->
<userPreferences>
  <titleVsAddName userValue="TITLE-PREF"/>
  <maileeVsAltRecipient userValue="MAILEE-ONLY"/>
  <streetVsPostOfcBox userValue="PO-BOX-PREF"/>
<!--
```

The user may suppress any particular line construct or constructs in a given situation without changing the template description by providing its name

here.

```
-->
  <suppressLine/>
</userPreferences>
<!--
```

Following are the trigger conditions that determine which line constructs are brought forward during the early stages of rendition; they are kept separate from the data describing the elements, rendition operators and rendition commands for clarity, and so that the name of the line constructs can be repeated when necessary without having to repeat the associated data.

```
-->
<triggerConditions>
  <lineSelect>
    <!--
      If this field is populated for a given record the line construct is brought forward, otherwise it is bypassed.
    -->
    <isPopulated>A10.400</isPopulated>
    <lineName>ORDER-LINE</lineName>
    <lineNumber>001</lineNumber>
  </lineSelect>
```

```
<!--
  Here no trigger conditions are specified, so the line construct is brought forward, though it could be eliminated later in rendition.
-->
```

```
<lineName>BLANK-LINE</lineName>
<lineNumber>002</lineNumber>
</lineSelect>
```

```
<!--
  This stipulates that if either of two fields is populated, the line construct is brought forward, but if both are populated, only the first reference will take
effect.
```

```
-->
  <isPopulated>A10.540</isPopulated>
  <lineName>ENDORSEMENT-LINE</lineName>
  <lineNumber>003</lineNumber>
  <isPopulated>A70.090</isPopulated>
  <lineName>ENDORSEMENT-LINE</lineName>
  <lineNumber>003</lineNumber>
</lineSelect>
</lineSelect>
```

```
<!--
Here the value in a field is compared to a constant, and if the values are equal, the line construct is brought forward.
```

```
-->
<hasValue>A70.090; "125X"</hasValue>
<lineName>CUST-LINE</lineName>
<lineNumber>004</lineNumber>
</lineSelect>
```

```
<lineSelect>
```

```
<!--
In this case both elements must be populated, and the user preference must be defined as one of those stated, for the line construct to be brought forward.
```

```
-->
<isPopulated>A50.010; A20.220</isPopulated>
<hasPreference>MAILEE-ONLY, MAILEE-PREF</hasPreference>
<lineName>MAILEE-LINE</lineName>
<lineNumber>005</lineNumber>
<isPopulated>A50.010</isPopulated>
<hasPreference>MAILEE-ONLY, MAILEE-PREF, ALT-PREF</hasPreference>
<lineName>MAILEE-LINE</lineName>
<lineNumber>005</lineNumber>
```

```
</lineSelect>
```

```
<lineSelect>
```

```
<lineName>NAME-LINE</lineName>
<lineNumber>006</lineNumber>
```

```
</lineSelect>
```

```
<lineSelect>
```

```
<isPopulated>A50.010; A20.220</isPopulated>
<hasPreference>ALT-ONLY, ALT-PREF</hasPreference>
<lineName>ALT-RECIP-LINE</lineName>
<lineNumber>007</lineNumber>
<isPopulated>A20.220</isPopulated>
<hasPreference>ALT-ONLY, ALT-PREF, MAILEE-PREF</hasPreference>
<lineName>ALT-RECIP-LINE</lineName>
<lineNumber>007</lineNumber>
```

```
</lineSelect>
```

```
<lineSelect>
```

```
<lineName>FIRM-LINE</lineName>
<lineNumber>008</lineNumber>
```

```
</lineSelect>
```

```
<lineSelect>
```

```
<!--
If the French postal box field is populated, it takes precedence, and the French postal box line construct is brought forward; otherwise if either A30.160 or one or more of the set of fields between A30.020 and A30.150 is populated, and the user preference for postal box format has been specified, the US postal box line construct is brought forward; but if the preference is for street address format, this is followed; and then if only the US postal box field has been populated, it is used regardless of preference; and finally in all other cases, the street address format is used.
```

```
-->
<isPopulated>FR-Boîte-Postale</isPopulated>
<lineName>BOÎTE-POSTALE</lineName>
<lineNumber>009</lineNumber>
<isPopulated>A30.160; A30.020-A30.150</isPopulated>
<hasPreference>PO-BOX-PREF</hasPreference>
<lineName>PO-BOX-LINE</lineName>
<lineNumber>009</lineNumber>
<isPopulated>A30.160; A30.020-A30.150</isPopulated>
<hasPreference>STREET-PREF</hasPreference>
<lineName>STREET-ADDR-LINE</lineName>
<lineNumber>009</lineNumber>
<isPopulated>A30.160</isPopulated>
<lineName>PO-BOX-LINE</lineName>
<lineNumber>009</lineNumber>
<defaultCase/>
<lineName>STREET-ADDR-LINE</lineName>
<lineNumber>009</lineNumber>
```

```
</lineSelect>
```

```
<lineSelect>
```

```
<lineName>CITY-STATE-LINE</lineName>
<lineNumber>010</lineNumber>
```

```
</lineSelect>
```

```
<lineSelect>
```

```
<!--
Here if the sending country as defined in ADIS matches the receiving country as defined by ECCMA, there is no need to explicitly include the receiving country in the address; but otherwise the receiving country line construct is brought forward.
```

```
-->
```

```
<!--
```

```
In actual mailings there may be a separate production group defined for these cross-border mail pieces; this allows a different label size or ink jet image size to be used to ensure that all the lines are printed; to do this the same template could be used with a different number of lines and length of lines defined in the address data header for the production group.
```

```
-->
```

```

    <hasValue>A40.070; E10.024</hasValue>
    <lineName>BLANK-LINE</lineName>
    <lineNumber>011</lineNumber>
    <defaultCase/>
    <lineName>COUNTRY-LINE</lineName>
    <lineNumber>011</lineNumber>
  </lineSelect>
</lineSelect>
  <lineSelect>
    <lineName>IN-HOME-LINE</lineName>
    <lineNumber>012</lineNumber>
  </lineSelect>
</triggerConditions>
<lineData>
  <!--

```

This line construct has one rendition operator and one element, and is also eligible for line merging, which would in this case leave its content as the first construct on the first line.

```

-->
  <lineName>ORDER-LINE</lineName>
  <lineNumber>001</lineNumber>
  <linePriority>005</linePriority>
  <lineMerge>1</lineMerge>
  <lineMergeOrder>1</lineMergeOrder>
  <lineMergeLoc>1</lineMergeLoc>
  <renditionOperator>
    <operatorId>LITERAL</operatorId>
    <fldJustify>L</fldJustify>
    <posStart>035</posStart>
    <posLength>018</posLength>
    <fldText>'Your Order Code:'</fldText>
  </renditionOperator>
  <elementData>
    <elementId>A10.400</elementId>
    <elementDesc>PROMOKY</elementDesc>
    <fldJustify>L</fldJustify>
    <posStart>***</posStart>
    <posLength>***</posLength>
  </elementData>
</lineData>
<lineData>
  <!--

```

This line construct is repeated later with a different line number. It has a line priority of 011, so it would be sacrificed before any line with a higher priority. The highest priority line has the lowest priority number.

```

-->
  <lineName>BLANK-LINE</lineName>
  <lineNumber>002</lineNumber>
  <linePriority>011</linePriority>
</lineData>
<lineData>
  <!--

```

This is an Optional Endorsement Line, which is used by the USPS to identify the level of presort, typically the destination, and sometimes other functions. It is constructed here by first laying down a background of asterisks and then overwriting some of them with the variable text.

```

-->
  <lineName>ENDORSEMENT-LINE</lineName>
  <lineNumber>003</lineNumber>
  <linePriority>006</linePriority>
  <renditionOperator>
    <operatorId>OVERLAY</operatorId>
    <fldJustify>L</fldJustify>
    <posStart>040</posStart>
    <posLength>030</posLength>
    <fldText>ALL '*</fldText>
  </renditionOperator>
  <elementData>
    <elementId>A10.540</elementId>
    <elementDesc>OEL</elementDesc>
    <fldJustify>R</fldJustify>
    <posStart>069</posStart>
    <posLength>***</posLength>
  </elementData>
</lineData>
<lineData>
  <lineName>CUST-LINE</lineName>
  <lineNumber>004</lineNumber>
  <linePriority>004</linePriority>
  <lineMerge>1</lineMerge>
  <lineMergeOrder>1</lineMergeOrder>
  <lineMergeLoc>2</lineMergeLoc>
  <elementData>

```

```

    <elementId>A70.090</elementId>
    <elementDesc>LISTCOD</elementDesc>
    <fldJustify>L</fldJustify>
    <posStart>040</posStart>
    <posLength>***</posLength>
  </elementData>
  <renditionOperator>
    <operatorId>LITERAL</operatorId>
    <fldJustify>L</fldJustify>
    <posStart>046</posStart>
    <posLength>017</posLength>
    <fldText>'CUSTOMER NUMBER:'</fldText>
  </renditionOperator>
  <elementData>
    <elementId>A10.500</elementId>
    <elementDesc>KEYLINE</elementDesc>
    <fldJustify>L</fldJustify>
    <posStart>063</posStart>
    <posLength>***</posLength>
  </elementData>
</lineData>
<lineData>
  <lineName>MAILEE-LINE</lineName>
  <lineNumber>005</lineNumber>
  <linePriority>007</linePriority>
  <elementData>
    <elementId>A20.230</elementId>
    <elementDesc>MAILEE</elementDesc>
    <fldJustify>L</fldJustify>
    <posStart>***</posStart>
    <posLength>***</posLength>
  </elementData>
</lineData>
<lineData>
  <!--

```

Here we see multiple rendition commands applying to the same element with different priorities. How many of them are actually executed depends upon the progress of the rendition and whether the address fits into the available space at the time that they would be executed. To allow a rendition command to be executed unconditionally, a zero priority can be assigned to it.

-->

```

  <lineName>NAME-LINE</lineName>
  <lineNumber>006</lineNumber>
  <linePriority>001</linePriority>
  <elementData>
    <elementId>A20.040</elementId>
    <elementDesc>PNCKNME</elementDesc>
    <elementId>A20.030</elementId>
    <elementDesc>PCHSNME</elementDesc>
    <fldJustify>L</fldJustify>
    <posStart>040</posStart>
    <posLength>***</posLength>
    <renditionCommand>
      <cmdId>INIT</cmdId>
      <cmdPriority>1</cmdPriority>
    </renditionCommand>
    <renditionCommand>
      <cmdId>ELIM</cmdId>
      <cmdPriority>2</cmdPriority>
    </renditionCommand>
  </elementData>
  <elementData>
    <elementId>A20.050</elementId>
    <elementDesc>PMD1NME</elementDesc>
    <fldJustify>L</fldJustify>
    <posStart>***</posStart>
    <posLength>***</posLength>
    <renditionCommand>
      <cmdId>INIT</cmdId>
      <cmdPriority>0</cmdPriority>
    </renditionCommand>
  </elementData>
  <elementData>
    <elementId>A20.080</elementId>
    <elementDesc>PLSTNME</elementDesc>
    <fldJustify>L</fldJustify>
    <posStart>***</posStart>
    <posLength>***</posLength>
  </elementData>
  <elementData>

```

```

    <elementId>A10.020</elementId>
    <elementDesc>DISPPAL</elementDesc>
    <fldJustify>L</fldJustify>
    <posStart>065</posStart>
    <posLength>***</posLength>
  </elementData>
</lineData>
<lineData>
  <lineName>ALT-RECIP</lineName>
  <lineNumber>007</lineNumber>
  <linePriority>008</linePriority>
  <renditionOperator>
    <operatorId>LITERAL</operatorId>
    <fldJustify>L</fldJustify>
    <posStart>040</posStart>
    <posLength>***</posLength>
    <fldText>'OR CURRENT RESIDENT:'</fldText>
  </renditionOperator>
</lineData>
<lineData>
  <lineName>FIRM-LINE</lineName>
  <lineNumber>008</lineNumber>
  <linePriority>010</linePriority>
  <elementData>
    <elementId>A50.020</elementId>
    <elementDesc>FIRMNAM</elementDesc>
    <fldJustify>L</fldJustify>
    <posStart>040</posStart>
    <posLength>***</posLength>
    <renditionCommand>
      <cmdId>WRAPDN</cmdId>
      <cmdPriority>1</cmdPriority>
    </renditionCommand>
    <renditionCommand>
      <cmdId>ABBREV</cmdId>
      <cmdPriority>2</cmdPriority>
    </renditionCommand>
  </elementData>
</lineData>
<lineData>
  <lineName>PO-BOX-LINE</lineName>
  <lineNumber>009</lineNumber>
  <linePriority>002</linePriority>
  <renditionOperator>
    <operatorId>LITERAL</operatorId>
    <fldJustify>L</fldJustify>
    <posStart>040</posStart>
    <posLength>***</posLength>
    <fldText>'PO BOX'</fldText>
  </renditionOperator>
  <elementData>
    <elementId>A30.160</elementId>
    <elementDesc>POBOXN</elementDesc>
    <fldJustify>L</fldJustify>
    <posStart>040</posStart>
    <posLength>***</posLength>
  </elementData>
</lineData>
<lineData>
  <!--

```

Here the French postal box number is stored in an ADIS user variable since this element has not otherwise been defined so as to be accessible. Another way to do this would be to define an element or user variable for the language for that particular line construct, store it in the address data, and then use the value of that element or variable as a trigger condition for selecting the French postal boxline format when appropriate. In that case the postal box number could be stored in the same field as the USPS box number.

```

-->
  <lineName>BOÎTE-POSTALE</lineName>
  <lineNumber>009</lineNumber>
  <linePriority>002</linePriority>
  <renditionOperator>
    <operatorId>LITERAL</operatorId>
    <fldJustify>L</fldJustify>
    <posStart>040</posStart>
    <posLength>***</posLength>
    <fldText>'B. P.'</fldText>
  </renditionOperator>
  <elementData>
    <elementId>FR-Boîte-Postale</elementId>
    <elementDef>delivery service qualifier</elementDef>

```



```

    <elementDesc>U01.010</elementDesc>
    <fldJustify>L</fldJustify>
    <posStart>040</posStart>
    <posLength>***</posLength>
  </elementData>
</lineData>
<lineData>
  <!--

```

Here there are two segments on a line, which allows the secondary address information to be wrapped up to a physical line just above the rest of the information, under certain conditions during rendition. If this is done, both the secondary address type and number must stay together. The USPS specifies that this information should either be on the address line or the line directly above, but not directly below, to facilitate OCR readability.

```

-->
  <lineName>STREET-ADDR-LINE</lineName>
  <lineNumber>009</lineNumber>
  <linePriority>002</linePriority>
  <lineSegment>
    <segmentSequence>002</segmentSequence>
  </lineSegment>
  <elementData>
    <elementId>A30.020</elementId>
    <elementDesc>PHSENUM</elementDesc>
    <fldJustify>L</fldJustify>
    <posStart>040</posStart>
    <posLength>***</posLength>
  </elementData>
  <elementData>
    <elementId>A30.030</elementId>
    <elementDesc>PSTPRED</elementDesc>
    <fldJustify>L</fldJustify>
    <posStart>***</posStart>
    <posLength>***</posLength>
    <renditionCommand>
      <cmdId>ABBREV</cmdId>
      <cmdPriority>1</cmdPriority>
    </renditionCommand>
  </elementData>
  <elementData>
    <elementId>A30.040</elementId>
    <elementDesc>PSTPRET</elementDesc>
    <fldJustify>L</fldJustify>
    <posStart>***</posStart>
    <posLength>***</posLength>
  </elementData>
  <elementData>
    <elementId>A30.050</elementId>
    <elementDesc>PSTNME</elementDesc>
    <fldJustify>L</fldJustify>
    <posStart>***</posStart>
    <posLength>***</posLength>
    <renditionCommand>
      <cmdId>REDUCE</cmdId>
      <cmdPriority>3</cmdPriority>
    </renditionCommand>
  </elementData>
  <elementData>
    <elementId>A30.060</elementId>
    <elementDesc>PSTTYPE</elementDesc>
    <fldJustify>L</fldJustify>
    <posStart>***</posStart>
    <posLength>***</posLength>
    <renditionCommand>
      <cmdId>ABBREV</cmdId>
      <cmdPriority>2</cmdPriority>
    </renditionCommand>
  </elementData>
  <elementData>
    <elementId>A30.070</elementId>
    <elementDesc>PSTPOST</elementDesc>
    <fldJustify>L</fldJustify>
    <posStart>***</posStart>
    <posLength>***</posLength>
  </elementData>
  <lineSegment>
    <segmentSequence>001</segmentSequence>
    <renditionCommand>
      <cmdId>WRAPUP</cmdId>
      <cmdPriority>4</cmdPriority>
    </renditionCommand>

```

```

</lineSegment>
<elementData>
  <elementId>A30.080</elementId>
  <elementDesc>PLSECD1</elementDesc>
  <fldJustify>L</fldJustify>
  <posStart>***</posStart>
  <posLength>***</posLength>
  <renditionCommand>
    <cmdId>ABBREV</cmdId>
    <cmdPriority>5</cmdPriority>
  </renditionCommand>
</elementData>
<elementData>
  <elementId>A30.090</elementId>
  <elementDesc>PLSECV1</elementDesc>
  <fldJustify>L</fldJustify>
  <posStart>***</posStart>
  <posLength>***</posLength>
</elementData>
</lineData>
<lineData>
  <!--

```

Here the CONCAT rendition command will place the hyphen between the two parts of the ZIP code, but only if both parts are present. It also suppresses the default delimiter, so there are no spaces preceding or following the hyphen.

```

-->
<lineName>CITY-STATE-LINE</lineName>
<lineNumber>010</lineNumber>
<linePriority>003</linePriority>
<elementData>
  <elementId>A40.040</elementId>
  <elementDesc>CITYNAM</elementDesc>
  <fldJustify>L</fldJustify>
  <posStart>040</posStart>
  <posLength>***</posLength>
</elementData>
<elementData>
  <elementId>A40.060</elementId>
  <elementDesc>STPRVAB</elementDesc>
  <fldJustify>L</fldJustify>
  <posStart>***</posStart>
  <posLength>***</posLength>
</elementData>
<elementData>
  <elementId>A10.170</elementId>
  <elementDesc>ZIPCODE</elementDesc>
  <contentRequired>Y</contentRequired>
  <fldJustify>L</fldJustify>
  <posStart>***</posStart>
  <posLength>***</posLength>
</elementData>
<renditionOperator>
  <operatorId>CONCAT</operatorId>
  <fldJustify>L</fldJustify>
  <posStart>***</posStart>
  <posLength>***</posLength>
  <fldText>'.' </fldText>
</renditionOperator>
<elementData>
  <elementId>A10.180</elementId>
  <elementDesc>PLUS4</elementDesc>
  <fldJustify>L</fldJustify>
  <posStart>***</posStart>
  <posLength>***</posLength>
</elementData>
</lineData>
<lineData>
  <lineName>BLANK-LINE</lineName>
  <lineNumber>011</lineNumber>
  <linePriority>012</linePriority>
</lineData>
<lineData>
  <!--
  This line construct will only be selected if the sending country differs from the receiving country, which implies cross border mail.
  -->
  <lineName>COUNTRY-LINE</lineName>
  <lineNumber>011</lineNumber>
  <linePriority>012</linePriority>
  <elementData>

```

```
<elementId>E10.024</elementId>
<elementDesc>CTRYREC</elementDesc>
<fldJustify>L</fldJustify>
<posStart>040</posStart>
<posLength>***</posLength>
</elementData>
</lineData>
<lineData>
<lineName>IN-HOME-LINE</lineName>
<lineNumber>012</lineNumber>
<linePriority>009</linePriority>
<renditionOperator>
<operatorId>LITERAL</operatorId>
<fldJustify>L</fldJustify>
<posStart>050</posStart>
<posLength>022</posLength>
<fldText>'IN HOME: AUG 29-AUG 31'</fldText>
</renditionOperator>
</lineData>
</contentDefinition>
</patdl.xml>
```