NAACCR
Data Exchange Standard

XML Specifications for Cancer Registry Records, Version 1.4

July 2020
Revised February 2021

Edited by:
Fabian Depry, Information Management Services
Isaac Hands, Kentucky Cancer Registry
Rich Pinder, Los Angeles Cancer Surveillance Program
Valerie Yoder, Utah Cancer Registry
Lori Havener, NAACCR

Acknowledgement:
We would like to thank the members of the NAACCR XML Data Exchange Work Group for their dedication and contributions.

Suggested citation:
Depry F, Hands I, Pinder R, Yoder V, Havener L, (eds). NAACCR Data Exchange Standard. Springfield, IL: North American Association of Central Cancer Registries, January 2021.

Funding for the NAACCR Data Exchange Standard was made possible in part by a cooperative agreement with Federal funds from the Centers for Disease Control and Prevention Cooperative Agreement number 5NU58DP006458. Its contents are solely the responsibility of the authors and do not necessarily represent the official view of the CDC.

Table of Contents

1 Purpose and Use of the XML Data Exchange Standard 1
2 Standard Overview 2
2.1 Dictionary Specifications 3
2.1.1 XML Elements 3
2.1.2 XML Samples 12
2.2 Data Exchange Specifications 13
2.2.1 XML Elements 13
2.2.2 XML Samples 16
2.3 Record Types 17
2.3.1 Special Considerations for the Modification Record Type 18
2.3.2 Record Types and Empty Data Items 19
2.3.3 Record Types and Custom User Dictionary Items 19
2.3.4 Record Types and Extensions to the NAACCR XML Standard 19
2.4 Other Considerations 19
2.4.1 Blank values 19
2.4.2 Consolidated Data and Nested Elements 20
2.4.3 Placement of Items at the Patient or Tumor Level 21
2.4.4 XSD Files 21
2.4.5 Validation 21
2.4.6 Extending the NAACCR XML Standard 22
2.4.7 Guidelines for Creating a Delimited Data File from a NAACCR XML File 23
2.4.8 Guidelines for Creating New XML NAACCR IDs 24
2.4.9 Compression 25
3 Appendix A: Changes to the NAACCR XML Specifications 26
3.1 Changes introduced in version 1.1 26
3.2 Changes introduced in version 1.2 26
3.3 Changes introduced in version 1.3 26
3.4 Changes introduced in version 1.4 26
4 Appendix B: Document History 27

NAACCR BOARD OF DIRECTORS

President:
Randi K. Rycroft, MSPH, CTR
Cancer Data Registry of Idaho
E-mail: rrycroft@teamiha.org

President-Elect:
Winny Roshala, VA, CTR
Cancer Registry of Greater California
E-mail: wroshala@crgc-cancer.org

Treasurer:
Maria Schymura, PhD
New York State Cancer Registry
E-mail: maria.schymura@health.ny.gov

Executive Director *ex officio*:
Betsy A. Kohler, MPH, CTR
NAACCR
E-mail: bkohler@naaccr.org

Advisory Board Member:
Lori Swain
National Cancer Registrars Association
E-mail: lswain@ncra-usa.org

Members at Large:
Angela Meisner, MPH
New Mexico Tumor Registry
E-mail: awmeisner@salud.unm.edu

Isaac Hands, MPH
Kentucky Cancer Registry
E-mail: isaac.hands@uky.edu

Kevin Ward, PhD, MPH, CTR
Metropolitan Atlanta SEER Registry
E-mail: kward@emory.edu

Monique Hernandez, PhD
Florida Cancer Data System
E-mail: mhernandez5@med.miami.edu

Lorraine Shack
Alberta Cancer Registry
E-mail: Lorraine.shack@ahs.ca

Mary Jane King
Ontario Cancer Registry, Ontario Health
E-mail: Maryjane.king@ontariohealth.ca

XML Data Exchange Work Group

Isaac Hands (co-chair)
Kentucky Cancer Registry
E-mail: Isaac.hands@uky.edu

Valerie Yoder (co-chair)
Utah Cancer Registry
E-mail: valerie.yoder@hsc.utah.edu

Sanjeev Baral
CDC Contractor
E-mail: sbaral@cdc.gov

Oliver Bucher
Cancer Care Ontario
E-mail: obucher@cancercare.mb.ca

Todd Carter
E-mail: todd@ers-can.com

Dan Curran, MS, CTR
C/NET Solutions
E-mail: danc@askcnet.org

Fabian Depry
IMS, Inc.
E-mail: depryf@imsweb.com

Michelle Esterly, RHIA, CTR
CDC
E-mail: Hjf7@cdc.gov

Cathleen Geiger
New Hampshire State Cancer Registry
E-mail: Cathleen.a.geiger@dartmouth.edu

Monica Guistwite, MPH, CTR
Electronic Registry Systems, Inc.
E-mail: mguistwite@mycrstar.com

Lori Havener, CTR
NAACCR
E-mail: lhavener@naaccr.org

Bert Heuer
C/NET Solutions
E-mail: berth@askcnet.org

Annette Hurlbut, RHIT, CTR
Elekta
E-mail: Annette.hurlbut@elekta.com

Sandy Jones
CDC
E-mail: Sft1@cdc.gov

Michael Koluder
CDC
E-mail: Kvt8@cdc.gov

Gary Levin
Florida Cancer Data System
E-mail: glevin@med.miami.edu

Rich Pinder
Los Angeles Cancer Surveillance Program
E-mail: rpinder@usc.edu

Jeff Reed
American College of Surgeons Cancer Programs
E-mail: jreed@facs.org

Joseph Rogers
CDC
E-mail: jrogers@cdc.gov

Landon Switzer
Saskatchewan Cancer Agency
E-mail: Landon.switzer@saskcancer.ca

Tuyet Thieu
Alberta Cancer Registry
E-mail: Tuyet.thier@ahs.ca

Barb Weatherby
CDC
E-mail: Wwj8@cdc.gov

Lin Xue
Manitoba Cancer Registry
E-mail: Lin.xue@cancercare.mb.ca

1 Purpose and Use of the XML Data Exchange Standard

The North American Association of Central Cancer Registries (NAACCR) fixed-width format (sometimes called the NAACCR flat file format) has been broadly accepted and widely used by the cancer surveillance community since before 1993. The simplicity of the format made it easy to implement in different types of software, but it also had some limitations. For example, it lacked metadata and the ability to handle large amounts of text. Also, it was difficult for organizations to define their own data items and ensure that they were used in a consistent way. And finally, it was difficult to add new data items and retire existing ones when their position and length in the file affected the position of every other data item (For a complete list of standard data items, refer to the Standards for Cancer Registries Volume II: Data Standards and Data Dictionary). These limitations led the NAACCR community to look into alternative data exchange file formats, eventually deciding on XML. The XML syntax provides extensibility that encourages experimentation with new data types and structures while supporting the development of registry software where the details of an individual registry does not need to be known to the entire NAACCR community.

In 2014, a NAACCR XML Task Force was created to define and demonstrate a custom, NAACCR-defined XML data exchange standard for the NAACCR community. The NAACCR Board approved the initial version (XML Specifications v1.0) of the standard in 2015. This started a transition period where NAACCR XML was fully compatible with the corresponding fixed-width format, providing an easy way to convert from XML to fixed-width and back again. This backward compatibility constrained the new XML standard to the same limitations as the fixed-width format, but at the same time provided several years for the NAACCR community to gradually adjust to XML and update its software dependencies. The transition period ended with the retirement of the fixed-width format alongside the publication of the NAACCR Standards for Cancer Registries, Data Standards and Data Dictionary, Version 21.

With the retirement of the fixed-width format, the following happened:

- The NAACCR Data Dictionary no longer specifies start columns for any data items, whether they are carried over from a previous version or newly introduced data items. This change is reflected in the searchable online NAACCR Data Dictionary and the published NAACCR Data Standards and Data Dictionary.
- New NAACCR XML dictionaries, both base dictionaries and user dictionaries, no longer specify start columns. Dictionaries for NAACCR versions 18 and earlier are not affected and will retain start columns, allowing these earlier versions to still be converted to the fixed-width format.
- Software that requires the fixed-width format and has not been updated to process XML will not be able to process data files using NAACCR version 21 or later.

The NAACCR XML data exchange standard was designed to facilitate electronic transmission of cancer registry data among registries for multiple purposes. It can be used

to provide standardized data from reporting sources to central registries, to share tumor reports on residents of other states or provinces from one central registry to another, or to report data from diverse data sources contributing to a combined study. The standard also makes the addition or retirement of standard data items simpler without affecting other data items.

Although NAACCR XML files in Version 21 and later do not have a defined conversion to the fixed-width format, they can be converted into a delimited file format that can be used in SAS or other software that relies on a tabular data model. Section 2.4.6 of this document provides guidelines for creating delimited data files from NAACCR XML. In addition, several software applications and libraries have been developed by the NAACCR community to create a delimited file from a given NAACCR XML file.

For more information about the NAACCR XML standard or to download related documentation and supplementary files, visit:

https://www.naaccr.org/xml-data-exchange-standard/.

2 Standard Overview

Detailed coding instructions for many data items in the data exchange record have been provided by national standard setting agencies such as American College of Surgeons (ACoS) Commission on Cancer (CoC), American Joint Committee on Cancer (AJCC), National Cancer Institute (NCI) Surveillance Epidemiology and End Results (SEER) Program, Centers for Disease Control and Prevention (CDC) National Program of Cancer Registries (NPCR), and the Canadian Council of Cancer Registries (CCCR). To determine which group has proposed specific standards for an item, refer to the "Source of Standard" column located in NAACCR Data Standards and Data Dictionary.

The NAACCR XML standard is based on XML 1.0 (https://www.w3.org/TR/REC-xml/) and defines two sets of specifications:

1. **Dictionary specifications**: defines the NAACCR Data Standards and Data Dictionary data items and their location in the NAACCR XML hierarchy.
2. **Data-exchange specifications**: defines the overall structure of a NAACCR XML data exchange document and some basic syntax conventions.

Line terminations are optional in a NAACCR XML data exchange file, but since the data files and dictionaries might be opened in text editors, it is recommended to use them along with proper indentation to make the XML file more human-readable for debugging purposes. New lines can be indicated with any combination of carriage-return and line-feed characters, however, be aware that XML processing software conforming to the XML 1.0 specification will convert all line terminators in a file to a single line-feed character (see section 2.11 End-of-Line Handling in the XML 1.0 specification, https://www.w3.org/TR/REC-xml/#sec-line-ends). This parsing behavior can sometimes be disabled in the XML processing software.

The standard does not enforce a specific character encoding, but recommends using the default for XML, UTF-8, which can be optionally specified in the XML header:

```
<?xml version="1.0" encoding="UTF-8"?>
```

or, alternately:

```
<?xml version="1.0"?>
```

Since version 1.0 is also the current default for XML, a file not defining a header is also valid and should be assumed to be XML version 1.0 with the UTF-8 character encoding.
Some cancer registry software systems and tools have not been written to properly handle characters outside the US-ASCII printable character range, so it is safest for text nodes in a NAACCR XML data exchange document to only contain ASCII characters in the decimal range 23 through 126.

All XML tags and attributes defined by this standard exist under the following namespace:

```
http://naaccr.org/naaccrxml
```

Dictionaries and data files should define this default namespace as a root attribute. More information about namespaces is provided in section 2.1.1.1.

NAACCR XML files must always have an extension of .xml unless they are compressed (see section 2.4.8).

2.1 Dictionary Specifications

NAACCR XML Dictionaries are XML files that define key metadata about NAACCR data items as they relate to a NAACCR XML data exchange file, their valid XML parent elements, and processing rules for dealing with text nodes containing coded values.

There are two types of dictionaries:

1. **Base dictionaries** define standard NAACCR items that are defined in the NAACCR Data Standards and Data Dictionary.
2. **User dictionaries** define non-standard (state, province, or organization-specific) items, these are maintained by the organizations defining them.

There is a base dictionary for each NAACCR Data Standards and Data Dictionary version. The XML standard also defines a default user dictionary for each version for reading and writing NAACCR XML data files when no other user dictionary is provided.

2.1.1 XML Elements

The XML structure for a given dictionary is defined by the following elements:

```
<NaaccrDictionary>
    <ItemDefs>
        <ItemDef/>
```

```
        </ItemDefs>
        <GroupedItemDefs>
            <GroupedItemDef/>
        </GroupedItemDefs>
    </NaaccrDictionary>
```

The <NaaccrDictionary>, <ItemDefs> and <GroupedItemDefs> elements occur once per document, the <ItemDef> and <GroupedItemDef> elements are repeated for each data item definition.

2.1.1.1 NaaccrDictionary

The <NaaccrDictionary> root element allows the following attributes:

- **dictionaryUri (required)**: a unique string that defines the dictionary. It acts as an identifier and is referenced by the NAACCR XML data files. The base and default user-defined dictionaries maintained by the NAACCR organization use a strong naming convention on their ID:
    - Base dictionaries use the format `http://naaccr.org/naaccrxml/naaccr-dictionary-xxx.xml` where xxx is the corresponding NAACCR version.
    - Default user dictionaries use the format `http://naaccr.org/naaccrxml/user-defined-naaccr-dictionary-xxx.xml` where xxx is the corresponding NAACCR version.
    - Custom user dictionaries use the format `[custom URI]/[organization]-naaccr-dictionary[-xxx][-v#.#].xml` where xxx is the corresponding NAACCR version, which is only required if the custom dictionary depends on a specific NAACCR version. v#.# is your organization's version of the dictionary (semantic versioning recommended) . The suggested custom user dictionary's file name is the same as the end of this URI.
      For example: https://mystate.gov/state-cancer-registry/mystate-naaccr-dictionary-180-v1.xml

  Using those conventions, a simple regular expression can be used to determine if a given dictionary is a base one or a user-defined one.

- **naaccrVersion (required for base dictionaries, optional for user dictionaries)**: NAACCR version that this dictionary defines. This field corresponds to the NAACCR Version data item 50 and allows the same values. Since version 1.1 of the NAACCR XML specification, this attribute is optional in user dictionaries; if not provided, it is assumed to be the same as the corresponding base dictionary.
- **specificationVersion (optional)**: the version of the specifications that the dictionary uses. If not provided, "1.0" is assumed.
- **description (optional)**: a short description of the dictionary.

In addition to those attributes, a dictionary should also define the default namespace: "http://naaccr.org/naaccrxml". Note that this value may not be a valid Internet address, it is simply a unique name for a specific XML namespace that defines all of the allowed elements and attributes. In summary, a dictionary must define the following attribute on the root XML element:

```
xmlns="http://naaccr.org/naaccrxml"
```

2.1.1.2 ItemDefs

The `<ItemDefs>` element acts as a container for repeated `<ItemDef>` elements and does not allow any attributes.

2.1.1.3 ItemDef

The base dictionary for a particular NAACCR version contains an `<ItemDef>` element for every NAACCR Data Standards and Data Dictionary data item defined in that version, except for the following items, which are defined in the default user dictionary instead:

- State/Requestor Items
- NPCR Specific Field

The `<ItemDef>` element allows the following attributes and associated values with the attribute name indicated in bold followed by a description of the value structure.

- **naaccrId (required)**: a unique value that identifies a data item, sometimes called a NAACCR ID or XML NAACCR ID. This value needs to be retained from one NAACCR version to the next, or the data item will not be recognized as being the same in both versions. The value should only contain letters and digits; it should start with a lowercase letter and use an uppercase letter to separate the words. The value must be 32 characters or less. Here are a few examples of valid values:
    - recordType
    - myItem
    - myOtherVariable2
    - old1970Variable

    The values for all NAACCR IDs in the base and default user dictionaries were derived from the NAACCR data item name in NAACCR Data Dictionary with the following rules applied:

    - Spaces, dashes, slashes, ampersands, periods and underscores are considered as word separators and replaced by a single space.
    - Anything in parenthesis is removed (including the parenthesis punctuation).
    - Any non-digit and non-letter character is removed.
    - The result is split by spaces (called words in the rest of this logic).

- Roman numeral words are converted to the corresponding numbers (so I becomes 1, II becomes 2, etc...); this applies only to full words, and only for numbers up to IX (9).
- If the two last words were converted roman numerals, a "to" word is inserted between them.
- The first word is uncapitalized, the other words are capitalized. All abbreviations are considered words (so EOD becomes Eod).
- All the words are concatenated back together.
- The resulting ID is manually reviewed to ensure it is no more than 32 characters.

The NAACCR organization will maintain the XML NAACCR ID of all the standard items it defines. Organizations defining their own non-standard items are responsible for providing the XML NAACCR ID of those items in their user-defined dictionaries.

- **naaccrNum (required):** the NAACCR data item number from the NAACCR Data Dictionary. User dictionaries can use numbers falling in any range as long as they follow these rules:
  - The number is not already defined in the base dictionary referenced by the data file.
  - The number is not defined in another user dictionary when several dictionaries are provided for the data file.

  Historically, ranges of item numbers have been assigned to different uses and for different organizations to manage. In practice, item numbers have been assigned outside their intended use or by organizations that were not assigned within a particular range. Since version 1.3 of the NAACCR XML specifications, item number ranges are no longer assigned to organizations or for specific uses.

  While this is not enforced by the standard, it is recommended to only use numbers in the 9500 to 99999 range in user-defined dictionaries to avoid conflicts with standard items (retired, current, or future items). There are two exceptions to this recommendation:

  - If an item has been retired by NAACCR but your organization wants to keep collecting it, you can define that item in your user-defined dictionary with the same number (and ID and name) as the retired item.
  - Some numbers are assigned to official data items that fall outside of the cancer record exchange format (Pathology records for example); if your organization wishes to collect those data items using the NAACCR XML data exchange format, you can use the same number (and name) as those other items.

  It is important to understand that number conflicts only happen in the context of reading or writing a given data file. It is perfectly safe to have the same number used

in two different user-defined dictionaries, as long as those dictionaries are never referenced together in a given data file.

- **naaccrName (optional):** the NAACCR data item name from the NAACCR Data Dictionary. While this attribute is optional because it is not strictly necessary for exchanging data, in practice a name should always be provided for all data items defined in a dictionary.

  Unlike the naaccrID, naaccrName can contain spaces, parentheses and other special characters, as long as they are properly escaped when written as an XML attribute value in the dictionary file. Names must be 50 characters or less (prior to 2018 the limit was 25 characters). Standardized abbreviations are used when necessary. Standardized punctuation and spacing are also used. Related fields are sometimes created with an identical stem and changing suffix. For example, names of all modalities of treatment in the first course of therapy have the identical stem "RX Summ", for Treatment Summary, followed by an indicator of the type of treatment (for example: "Chemo" is RX Summ—Chemo). NAACCR data item names are relatively stable across versions, but on rare occasions they have been updated between versions to reflect a better understanding of the data item that was not known at the time the item was defined. In contrast, naaccrNums, are unchanged during the life of the data item. And NAACCR IDs almost never change. There are rare exceptions; for example, if keeping an old NAACCR ID with a new updated naaccrName would introduce a lot of confusion. The NAACCR organization might decide it is better to also change the NAACCR ID in that case to avoid that confusion.

  A small set of characters have a special meaning in XML and must be escaped if they appear as the naaccrName attribute value or anywhere else in an XML document. The following table shows the characters that must be escaped:

Original Character	Escaped Value
<	`<`
>	`>`
"	`"`
&	`&`
'	`'`

- **startColumn (required for NAACCR Version 18 and prior, ignored for all other versions)**: the starting column where this data item will be read or written from a fixed-width file. The startColumn attribute was originally part of NAACCR XML to allow conversion to a fixed-width file during a transitional period of the data exchange standard. See section 1 of this document for a discussion of the fixed-width transition.
- **length (required):** the maximum length of the value in a NAACCR XML data file, can be overridden by the allowUnlimitedText attribute.

- **allowUnlimitedText (optional):** boolean indicating that the value of the data may be longer than the specified length attribute. This attribute may be needed for data items that contain unstructured text of an unpredictable length. The XML standard does not put a restriction on the actual length of text values in data files, therefore, applications may create data items that contain data outside the bounds of the length attribute with the understanding that some data consumers may truncate the data.
- **recordTypes (optional):** a comma-separated list of the record types from the NAACCR Data Dictionary where this data item can appear. For example, the value for an item defined in a NAACCR Incidence record would be "A,M,C,I" while an item defined only in a NAACCR Abstract/Modified record would be "A,M". Defaults to "A,M,C,I". Section 2.3 of this document defines these record types.
- **parentXmlElement (required):** expected parent tag in the XML data files; either "NaaccrData", "Patient", or "Tumor". This parent element is what defines the nested structure of a NAACCR XML data exchange document.
- **dataType (optional)**: a name for the type of data contained in a data item, maps directly to a regular expression that can be used to validate the value. If not provided, "text" is assumed.

dataType	Usage	Regular Expression
digits	Codes composed of digits only, which always occupy the full width of the field (n in the regular expression). Spaces are not allowed. The following examples assume a data item of length 2. A data item not appearing in the data file (not transmitted) is always valid. Valid Example: • <Item naaccrId="code">01</Item> Invalid Examples (assuming length 2): • <Item naaccrId="code">1</Item> • <Item naaccrId="code"> 1</Item> • <Item naaccrId="code"> </Item> • <Item naaccrId="code">001</Item>	^\d{n}$
alpha	Codes composed of uppercase characters only, which always occupy the full width of the field (n in the regular expression). Spaces are not allowed. The following examples assume a data item	^[A-Z]{n}$

	of length 2. A data item not appearing in the data file (not transmitted) is always valid. Valid Example: • <Item naaccrId=”code”>AK</Item> Invalid Examples: • <Item naaccrId=”code”>X</Item> • <Item naaccrId=”code”> X</Item> • <Item naaccrId=”code”> </Item> • <Item naaccrId=”code”>ak</Item> • <Item naaccrId=”code”>12</Item> • <Item naaccrId=”code”>ABC</Item>	
mixed	Codes composed of digits and/or uppercase characters, which always occupy the full width of the field (n in the regular expression). Spaces are not allowed. The following examples assume a data item of length 2. A data item not appearing in the data file (not transmitted) is always valid. Valid Examples: • <Item naaccrId=”code”>AA</Item> • <Item naaccrId=”code”>12</Item> • <Item naaccrId=”code”>A1</Item> Invalid Examples: • <Item naaccrId=”code”>A</Item> • <Item naaccrId=”code”>1</Item> • <Item naaccrId=”code”> A</Item> • <Item naaccrId=”code”> 1</Item> • <Item naaccrId=”code”> </Item> • <Item naaccrId=”code”>aa</Item> • <Item naaccrId=”code”>a1</Item>	^[A-Z\d]{n}$
numeric	Variable length digits with an optional period. Spaces are not allowed The following examples assume a data item of length 3. A data item not appearing in the data file (not transmitted) is always valid. Valid Examples:	^\d+(\.d+)?$

	• <Item naaccrId=”code”>123</Item> • <Item naaccrId=”code”>1</Item> • <Item naaccrId=”code”>1.2</Item> Invalid Examples: • <Item naaccrId=”code”>ABC</Item> • <Item naaccrId=”code”>1 </Item> • <Item naaccrId=”code”> 1</Item> • <Item naaccrId=”code”> </Item> • <Item naaccrId=”code”>12.3</Item>	
date	A NAACCR-style full or partial date (yyyy, yyyymm or yyyymmdd). Spaces are not allowed. Dates data item should always be defined as length 8. A data item not appearing in the data file (not transmitted) is always valid. Valid Examples: • <Item naaccrId=”code”>20200615</Item> • <Item naaccrId=”code”>202006</Item> • <Item naaccrId=”code”>2020</Item> Invalid Examples: • <Item naaccrId=”code”>20201632</Item> • <Item naaccrId=”code”>20200600</Item> • <Item naaccrId=”code”>20200699</Item> • <Item naaccrId=”code”>20201315</Item> • <Item naaccrId=”code”>20200015</Item> • <Item naaccrId=”code”>20209915</Item> • <Item naaccrId=”code”>00000000</Item> • <Item naaccrId=”code”>99999999</Item> • <Item naaccrId=”code”>	^(18|19|20)\d\d((0[1-9]|1[012])(0[1-9]|[12]\d|3[01])?)?$

	202006 </Item> • <Item naaccrId=”code”> 2020 </Item> • <Item naaccrId=”code”></Item>	
text	Variable length text that can contain any characters, up to the specified with of the field (n in the regular expression). A data item not appearing in the data file (not transmitted) is always valid. Note that this is the only type that allows spaces to be transmitted in the value; those spaces can be anywhere in the value, including at the very beginning or end. It is also valid for this data type to have a value composed of spaces only.	^.{1,n}$

Note that data types are tied to data validation. Historically, that validation has been performed by running edits and the data types can be viewed as very basic edits. However, a data file that contains data items not agreeing with their data type is not necessarily an invalid file. What is considered “invalid” is left to the discretion of the software implementing the specifications. Some software might decide to be very strict and reject any data files that has a type error; others might prefer to only use edits for data validation and ignore any type errors. Libraries that provide generic NAACCR XML read/write features should report data errors but let the user of the library decide how to consume those errors.

- **padding (optional, only relevant for NAACCR Version 18 and prior):** a name for the text padding rules when writing a fixed-width file; can be set to “rightBlank”, “leftBlank”, “rightZero”, or “leftZero”; defaults to “rightBlank”. The padding attribute was added to NAACCR XML to allow conversion of a NAACCR XML file to a fixed-width file during a transitional period of the data exchange standard. From a pure formatting point-of-view, padding rules are not necessary for NAACCR XML and can cause problems when they are not applied consistently across software products. These padding rules should not be used to enforce data coding standards, such as converting a “1” to an “01”. At some point in the future, this optional padding attribute will be retired, therefore, software should rely on edits to identify invalid values instead of these padding attributes.
- **trim (optional, only relevant for NAACCR Version 18 and prior)**: a name for the text trimming rules when converting a fixed-width file to XML; can be set to “none” or “all”, defaults to “all”. The trim attribute was added to NAACCR XML to handle converting a NAACCR XML file to a fixed-width file. Specifically, it is needed to read special data items such as the State Requestor Items from fixed-width data files without losing any information in the process. These special data items rely on start

columns in a fixed-width file along with the trim value to provide a final value for multiple fields. When applied incorrectly to a fixed-width file, trimming rules can cause data loss. From a pure formatting point-of-view, trimming rules don't make sense for NAACCR XML. Values can appear with or without leading/trailing spaces in an XML data file; which has no impact on whether the file is valid or not. It might impact whether edits deem the values valid or not, but that is outside the scope of the NAACCR XML exchange format. At some point in the future, this optional trim attribute will be retired, therefore, it is recommended to use edits to identify invalid values instead of relying on these trim attributes.

2.1.1.4 GroupedItemDefs

The `<GroupedItemDefs>` element acts as a container for the repeated grouped item definitions. It does not allow any attribute. It can only appear in a base dictionary, which means organizations cannot define their own grouped items.

2.1.1.5 GroupedItemDef

Grouped item definitions follow the same rules as the regular `<ItemDef>` definitions, allowing the same attributes and having the same requirements. In addition, every grouped item definition must supply the following attribute:

- **contains (required)**: a comma-separated list of naaccrIds referencing the items making up the grouped item, in the order they are specified. The list should not contain any spaces and must reference naaccrIds that exist in the current base dictionary.

2.1.2 XML Samples

Sample base dictionary:

```
<NaaccrDictionary
  dictionaryUri="http://naaccr.org/naaccrxml/naaccr-dictionary-180.xml"
  naaccrVersion="180"
  specificationVersion="1.4"
  description="NAACCR 18 base dictionary"
  xmlns="http://naaccr.org/naaccrxml">
    <ItemDefs>
        <ItemDef naaccrId="recordType"
            naaccrNum="10"
            naaccrName="Record Type"
            startColumn="1"
            length="1"
            recordTypes="A,M,C,I"
            parentXmlElement="NaaccrData"
        <ItemDef naaccrId="registryType"
            naaccrNum="30"
            naaccrName="Registry Type"
            startColumn="2"
            length="1"
            recordTypes="A,M,C,I"
            parentXmlElement="NaaccrData"
            dataType="digits"/>
        …
```

```
    </ItemDefs>
</NaaccrDictionary>
```

The current NAACCR dictionary is available at https://www.naaccr.org/xml-data-exchange-standard/.

Sample user dictionary:

```
<NaaccrDictionary
  dictionaryUri="http://myregistry.org/myregistry-naaccr-dictionary-001.xml"
  specificationVersion="1.4"
  xmlns="http://naaccr.org/naaccrxml">
    <ItemDefs>
        <ItemDef naaccrId="myRegistryVariable1"
            naaccrNum="10001"
            length="1"
            parentXmlElement="NaaccrData"/>
        <ItemDef naaccrId="myRegistryVariable2"
            naaccrNum="10002"
            length="1"
            parentXmlElement="Patient"/>
        <ItemDef naaccrId="myRegistryVariable3"
            naaccrNum="10003"
            length="1"
            parentXmlElement="Tumor"/>
    </ItemDefs>
</NaaccrDictionary>
```

2.2 Data Exchange Specifications

A NAACCR XML data file must define a base dictionary and may define one or several user dictionaries. It can only contain NAACCR data items that are defined in one of those dictionaries.

2.2.1 XML Elements

The structure of the data files is defined by the following elements:

```
<NaaccrData>
    <Item/></Item>
    <Patient>
        <Item></Item>
        <Tumor>
            <Item></Item>
        </Tumor>
    </Patient>
</NaaccrData>
```

The <NaaccrData> element occurs once per document, the <Patient> elements occur once per patient, and the <Tumor> elements occur once per tumor within a given patient. Each of those levels can contain any number of <Item> elements, defined by the dictionaries specified in the file.

The XML syntax is clearly separated from the data item definitions by using a single <Item> XML tag with an attribute providing the data item ID, as opposed to having a specific tag per data item like <PrimarySite>. The advantage of this approach is that new standard data items can be added on a regular basis without affecting the XML syntax itself, making the maintenance of the format much simpler for NAACCR and cancer surveillance software vendors.

Note that grouped data items from the current NAACCR Data Standards and Data Dictionary, Appendix E are not supported as <Item> elements in data files, even though their <GroupedItemDef> elements are included in the base dictionary for the convenience of parsing logic.

2.2.1.1 NaaccrData

The <NaaccrData> element is the root of NAACCR XML data files. It can have one or more <Item> children as long as they are defined in the base or user dictionary with a parentXmlElement attribute set to "NaaccrData". The <NaaccrData> element can also contain any number of <Patient> elements.

It allows the following attributes**:**

- **baseDictionaryUri (required)**: the base dictionary URI that defines the data items used in the data file.
- **userDictionaryUri (optional)**: the user dictionary URI that defines the data items used in the data file. If not provided, a default user dictionary will be used. Several URIs can be provided in this attribute, each separated by a space.
- **recordType (required)**: the record types contained in the data file (i.e., A, M, C or I). Dictionaries will typically contain the definitions of all data items for any record type, but this attribute restricts which items can actually appear in the data exchange file.
- **specificationVersion (optional)**: the version of the specifications that the data file uses. If not provided, "1.0" is assumed.
- **timeGenerated (optional)**: a timestamp in ISO 8601 format indicating when the XML data file was created, for example: 2020-04-24T09:32:35.604-04:00

In addition to those attributes, a data file should also define a default namespace of: "http://naaccr.org/naaccrxml". Note that this value may not be a valid Internet address (i.e., a URL), it is simply a unique name for a specific XML namespace that defines all of the allowed elements and attributes (i.e., a URI). In summary, a data file must define the following attribute on the root XML element:

```
xmlns="http://naaccr.org/naaccrxml"
```

As part of defining "extensions" (see corresponding section), other namespaces can be defined. For example, the following attributes can be added to define an "ext" tag prefix for a custom namespace, "http://my.company.org/naaccrxml":

```
xmlns:ext="http://my.company.org/naaccrxml"
```

This is an advanced feature and in most situations, only the default namespace needs to be specified.

Finally, the NAACCR XML standard allows for user-defined root attributes in their own custom namespaces. The standard makes no assumptions on what those are and what they mean. A library implementing the standard needs to allow all those user-defined attributes to be retrieved when a data file is parsed. As an example, the following attribute can appear on the root:

```
ext:internalFileId="XYZ-1"
```

2.2.1.2 Patient

A <NaaccrData> element can contain any number of <Patient> elements as children. Each <Patient> element can have one or more <Item> children as long as they are defined in the base or user dictionary with a parentXmlElement attribute set to "Patient". <Patient> elements can also contain any number of <Tumor> elements.

<Item> elements that are direct children of a <Patient> element apply to all <Tumor> elements underneath that <Patient>. For example, the social security number of a patient defined as <Item naaccrId="socialSecurityNumber"> will apply to every <Tumor> underneath that <Patient>. For a complete list of <Item> elements that are valid children of <Patient>, see the base dictionary description and file.

This element has no attributes.

2.2.1.3 Tumor

A <Patient> element can have any number of <Tumor> elements as children. Each <Tumor> element can have one or more <Item> children as long as they are defined in the base or user dictionary with a parentXmlElement attribute set to "Tumor". Each <Tumor> element corresponds to a separate tumor record for its parent <Patient> element. For a complete list of all <Item> elements that are valid children of <Tumor>, see the base dictionary description and file.

This element has no attributes.

2.2.1.4 Item

Items define the value for specific data items. The <Item> element specifies the following attributes:

- **naaccrId (required)**: the unique identifier of the data item.
- **naaccrNum (optional)**: the NAACCR data item number of the data item. While the NAACCR community would typically use the NAACCR data item number to uniquely identify items, those are not a good fit for identifying them in XML data files. The

main reason is that different organizations tend to re-use the same numbers for their own custom items; which can cause confusion when multiple organizations need to be referenced in one data file. A second reason to require naaccrIds instead of item numbers is to make the XML data files more readable when opened in a text editor. Therefore, if you are relying on naaccrNum as a tracking resource, it is advisable to cross check the number with the naaccrId.

The order of <Item> elements under their parent does not matter, and their item value can contain newline characters without disrupting the syntax of the XML document. If an <Item> element does not have a value or consists entirely of whitespace characters, it should be omitted entirely from the data file.

A small set of characters have a special meaning in XML and must be escaped if they appear in the value of an item. The following table shows the characters that must be escaped:

Original Character	Escaped Value
<	<
>	>
"	"
&	&
'	'

The only alternative to escaping these characters is to use a CDATA block around the entire value. Software that creates NAACCR XML data with any of these characters must either escape them as shown in the table above, or surround them in a CDATA block. Software that consumes NAACCR XML data should be able to support both approaches. Since dealing with special characters is a generic XML issue, it is expected that most software will use standardized libraries to read and write XML files, and that those libraries will handle the escaping automatically.

2.2.2 XML Samples

Sample XML data file with two patient entities, second one has no tumor:

```
<NaaccrData
  baseDictionaryUri="http://naaccr.org/naaccrxml/naaccr-dictionary-180.xml"
  recordType="I"
  timeGenerated="2019-08-16T08:09:19-04:00"
  specificationVersion="1.4"
  xmlns="http://naaccr.org/naaccrxml">
    <Item naaccrId="registryId">0000000001</Item>
    <Patient>
        <Item naaccrId="patientIdNumber">00000001</Item>
        <Tumor>
            <Item naaccrId="primarySite">C123</Item>
        </Tumor>
    </Patient>
    <Patient>
        <Item naaccrId="patientIdNumber">00000002</Item>
    </Patient>
</NaaccrData>
```

Simple XML data file using a user-dictionary and specifying the NAACCR Item Numbers:

```
<NaaccrData
  baseDictionaryUri="http://naaccr.org/naaccrxml/naaccr-dictionary-180.xml"
  userDictionaryUri="http://mycompany.org/mycompany-naaccr-dictionary.xml"
  recordType="I"
  timeGenerated="2019-08-16T08:09:19-04:00"
  specificationVersion="1.4"
  xmlns="http://naaccr.org/naaccrxml">
    <Item naaccrId="registryId" naaccrNum="40">0000000001</Item>
    <Item naaccrId="myCompanyVariable1" naaccrNum="10001">X</Item>
    <Patient>
        <Item naaccrId="patientIdNumber" naaccrNum="20">00000001</Item>
        <Item naaccrId="myCompanyVariable2" naaccrNum="10002">Y</Item>
        <Tumor>
            <Item naaccrId="primarySite" naaccrNum="400">C123</Item>
            <Item naaccrId="myCompanyVariable3" naaccrNum="10003">Z</Item>
        </Tumor>
    </Patient>
</NaaccrData>
```

2.3 Record Types

The record type of an XML file is a single character code that indicates which set of data items are included in the XML file. Only one record type can be defined for each XML data exchange file, specified by the "recordType" attribute on the <NaaccrData> element. The following record types are defined:

Incidence Record

Record Type "I": Coded data without direct patient identifiers
Item Types Include: Demographic, Tumor Identifiers, Staging, Treatment, and Follow-up
Examples of Use: Studies where direct patient identifiers should not be exchanged or to transmit data for multi-registry research projects or surveillance.

Confidential Record

Record Type "C": Incidence record with the addition of patient identifiers
Item Types Include: Demographic, Tumor Identifiers, Staging, Treatment, Follow-up, Patient Identifiers, and Physician Identifiers
Examples of Use: Exchange cases between registries, whether central-based or hospital-based.

Full Case Abstract

Record Type "A": Confidential record with the addition of text
Item Types Include: Demographic, Tumor Identifiers, Staging, Treatment, Follow-up, Patient Identifiers, Physician Identifiers, and Text
Examples of Use: Allows the receiving registry to perform a higher degree of quality control with each case report, or for use in research studies where narrative text is important such as natural language processing.

Modification Record

Record Type "M": Identical to Full Case Abstract but only modified items are sent
Item Types Include: Demographic, Tumor Identifiers, Staging, Treatment, Follow-up, Patient Identifiers, Physician Identifiers, and Text
Examples of Use: The Modification Record is a special record type that can contain any data item from the Full Case Abstract and is meant to communicate a change in a data item value for a previously received cancer abstract. See section 2.3.1, Special Considerations for the Modification Record Type, for more details.

With the Incidence, Confidential, and Full Case Abstract, each record type contains progressively more data items. For a complete list of all data items included in each record type, refer to the NAACCR Data Dictionary or the Base Dictionary file corresponding to a specific version of the NAACCR Data Dictionary.

2.3.1 Special Considerations for the Modification Record Type

The Modified record type is not supported at all central registries and some registry software does not support it. When supported, it should be used to communicate changes to previously submitted data records. The "M" record is identical in format to the "A" record type and uses the same base dictionary. The "M" record may be used to transmit corrections or follow-up, and for any change to any item, including abstracting text. The vendor software, or producer of the M record, should write out the new, corrected values, in addition to writing all data items that would normally be transmitted with an "A" record. That will allow central registries to perform a full field-by-field comparison of the data they currently have vs the data they are receiving, and to have full control on which items they want to update in their system.

Tumor records that have not been reported to the central registry should be written in the "A" format, and tumor records that have already been transmitted but that have had an update to any field, should be written in the "M" format. The Date Case Report Exported field [2110] can be used to identify tumor records, which have already been transmitted, and a comparison of item #2110 to the Date Case Last Changed field [2100] can be used to identify records that have been modified since the last time they were exported. Also, it is assumed that the Date Case Report Exported field will be updated when an "M" record is generated.

There is no standard frequency for transmitting files of accumulated modified records. Frequency will vary with caseload and frequency of transmission of new cases. The most common approach is to send accumulated modified or corrected records each time a transmittal of new cases is generated. It might also be useful to allow ad hoc submissions of modified or corrected records for those times when numerous corrections are made at once.

As a historical note, when the NAACCR data exchange standard used fixed-width records, an Update/Correction record (type "U") was defined as distinct from a Modified record. In the current XML data exchange standard, those "U" record types would have the same

structure as a Modification record, therefore only the Modified (M) record type is defined now.

2.3.2 Record Types and Empty Data Items

Sometimes a data item will not have a value, such as when a patient does not have a middle name, in which case an XML data exchange file should not include those empty items, even though they are specified in the record type definition.

2.3.3 Record Types and Custom User Dictionary Items

Record types do not indicate which custom data items are included in a NAACCR XML file (otherwise known as user dictionary items). The only way to know which user dictionary items are included in an XML file is to read the *userDictionaryUri* attribute of the <NaaccrData> XML Item and find the corresponding User Dictionary file that lists the custom items.

2.3.4 Record Types and Extensions to the NAACCR XML Standard

Record types do not indicate whether custom extensions to the NAACCR XML Standard have been used in an XML file, sometimes referred to as XML namespace extensions. Consumers of any NAACCR XML file must be able to ignore custom XML extensions if they are not supported by the software that is reading the XML.

2.4 Other Considerations

2.4.1 Blank values

A Blank value for a given data item means that no value is being transmitted for that item. For a NAACCR XML data file, it means the Item tag should not be provided in the file or that data item.

Blank values should not be associated with spaces, but historically that has been the case for the retired NAACCR fixed-columns format. The nature of that format forced blank values to be provided as spaces to fill the required gap in the data line and not disrupt the location of the following values. Because of that, there was no possible distinction between a value that was not transmitted vs a value that was transmitted with spaces-only. For the NAACCR XML format, that distinction is now possible, and if a value is provided with leading spaces, trailing spaces, or spaces only, those spaces make up the actual value and are not an artifact of the transmission itself.

Note that the NAACCR XML standard also defines data types for every item, and most of those data types don't allow spaces. In fact, the "text" data is the only one that allows spaces.

The following examples are valid for the Date of Diagnosis item (defined as type "date"):

- <Item naaccrId="dateOfDiagnosis">20200615</Item>
- <Item naaccrId="dateOfDiagnosis">202006</Item>
- <Item naaccrId="dateOfDiagnosis">2020</Item>
- Item not appearing in the data file (so not transmitted)

The following examples are invalid for the Date of Diagnosis item (defined as type “date”):

- <Item naaccrId=”dateOfDiagnosis”>202006 </Item> (does not follow the regular expression used to validate date values)
- <Item naaccrId=”dateOfDiagnosis”>2020 </Item> (does not follow the regular expression used to validate date values)
- <Item naaccrId=”dateOfDiagnosis”>20200699</Item> (does not follow the regular expression used to validate date values)
- <Item naaccrId=”dateOfDiagnosis”>20209999</Item> (does not follow the regular expression used to validate date values)
- <Item naaccrId=”dateOfDiagnosis”>99999999</Item> (does not follow the regular expression used to validate date values)
- <Item naaccrId=”dateOfDiagnosis”>00000000</Item> (does not follow the regular expression used to validate date values)
- <Item naaccrId=”dateOfDiagnosis”></Item> (empty values are not allowed)
- <Item naaccrId=”dateOfDiagnosis”/> (empty values are not allowed)

The following examples are valid for the Date of Diagnosis Flag item (defined as type “digits”):

- <Item naaccrId=”dateOfDiagnosisFlag”>12</Item>
- Item not appearing in the data file (so not transmitted)

The following examples are invalid for the Date of Diagnosis Flag item (defined as type “digits):

- <Item naaccrId=”dateOfDiagnosisFlag”> </Item> (does not follow the regular expression used to validate digits values)
- <Item naaccrId=”dateOfDiagnosisFlag”></Item> (empty values are not allowed)

<Item naaccrId=”dateOfDiagnosisFlag”/> (empty values are not allowed)
The following examples are valid for the Text Remarks item (defined as type “text”):

- <Item naaccrId=”textRemarks”> Some text after some leading spaces<Item>
- <Item naaccrId=”textRemarks”>Some text before some trailing spaces <Item>
- <Item naaccrId=”textRemarks”> <Item>
- Item not appearing in the data file (so not transmitted)

That example containing only spaces is valid because the “text” data type doesn’t impose any restriction on the values and allows spaces (unlike the other data types). In practice, proving a text value that only contains spaces is not useful and should be avoided.

The following examples are invalid for the Text Remarks data item (defined as type “text”):

- <Item naaccrId=”textRemarks”></Item> (empty values are not allowed)
- <Item naaccrId=”textRemarks”/> (empty values are not allowed)

2.4.2 Consolidated Data and Nested Elements

Since the standard defines a nested structure for patients and tumors, registries with consolidated patient information can send a single set of patient data in a <Patient> element when multiple tumors are included in <Tumor> elements. If a registry does not

have consolidated patient information, then a separate <Patient> and <Tumor> element will be sent for every tumor record. The standard does not enforce <Patient> or <Tumor> element uniqueness, for example, multiple <Patient> elements can be sent with the same Patient ID and multiple <Tumor> elements can be sent for the same tumor record to represent a patient visit history. The standard has been designed to be flexible enough to accommodate many data collection scenarios, while also supporting consolidated, heavily curated cancer abstracts. In any scenario, <Item> elements must be nested under the appropriate parent, <Patient> or <Tumor>, according to the Base Dictionary.

2.4.3 Placement of Items at the Patient or Tumor Level

One of the most commonly discussed aspects of the standard during its development was how to place data items at the correct level in the Patient/Tumor hierarchy. The NAACCR Data Standards and Data Dictionary alluded to some patient and tumor delineations, but could be ambiguous and confusing about whether some data items should appear once per patient or once per tumor record. The NAACCR organization worked with the community to define the most appropriate data level for all existing data items.

As of NAACCR version 21, the data level is a required piece of information on the new data item request forms that organizations submit to NAACCR.

2.4.4 XSD Files

The standard includes a W3C-compliant XML Schema Definition (XSD) file called "naaccr_data_VERSION.xsd" (where VERSION is the specifications version) that defines the valid elements and attributes in a NAACCR XML data exchange file. Some XML parsers can use the XSD file to validate the basic syntax of a NAACCR XML data exchange file.

The XSD was specifically designed to avoid both data type validation and <Item> parent-child validation because of two limitations inherent in most XSD validation software. First, many XSD validators read an entire XML file before reporting its validity, making them inappropriate for large registry data exchange files. Second, most XSD validators will reject an entire XML file as soon as they encounter any portion that is non-compliant with the XSD. This behavior is problematic for most NAACCR data exchange use cases that need an overall picture of the portions of a file that are invalid rather than a Boolean valid-invalid result, as well as the ability to ignore certain validation errors based on special circumstances. For these reasons, sophisticated validation of NAACCR XML data exchange files is left to a custom software tool instead of the XSD.

The standard also includes an XSD file for the dictionary called "naaccr_dictionary_VERSION.xsd" (where VERSION is the specifications version).

2.4.5 Validation

The NAACCR XML standard defines a stepwise approach to XML data validation where each step provides an increasing level of validation complexity:

Step 1. Element and Attribute Validation

XSDs provide basic validation of the correct element and attribute naming conventions in a NAACCR XML data exchange file. This step relies on W3C standards-compliant XML parsing software.

Step 2. Data Type and Nesting Structure

The Base Dictionary and User Dictionary files contain all of the information necessary to validate the data types of data items and their nested structure within <Patient> and <Tumor> elements. This step can be accomplished with an XML software tool provided by NAACCR, or custom NAACCR XML processing software.

Step 3. Coding and Context

Standard Edits metafiles can be used the same way they are currently used to validate a NAACCR XML data exchange file.

2.4.6 Extending the NAACCR XML Standard

The NAACCR XML standard allows custom, user-defined data and metadata at multiple insertion points in a data exchange document. This built-in extensibility is a forward-looking feature of the standard that encourages experimentation with new data types and structures while supporting the development of registry software where the details of an individual registry does not need to be known to the entire NAACCR community. All of these extensibility features can be used without any change to the XSDs.

One means of extending a NAACCR XML data exchange document is to add custom attributes to the root elements in both data files and dictionaries.

Another extensibility feature allows the inclusion of arbitrary XML data in a NAACCR XML data exchange document at the <NaaccrData>, <Patient>, or <Tumor> levels. This extension method allows sophisticated data exchange scenarios where biomarker data, synoptic pathology reports, or anything else that can be encoded as valid XML can be included in a NAACCR XML file. Including data in this manner is different from defining a new <Item> in a custom User Dictionary because it is not bound by the same space limitations. These new data inclusions can be any size and do not have to be defined as <Item> elements with a NAACCR data item number, but they do have some important limitations. First, the extra data cannot be easily converted into a delimited or fixed-width file. And second, these data extensions require some kind of external data dictionary outside of the NAACCR XML specifications to define their semantics and syntax for communication to other registries.

While these extensions are powerful, custom data that can be contained in an <Item> element and defined in a custom User Dictionary is preferable because it is easier to preserve in a conversion to and from delimited or fixed-width files. However, when the custom data is too large to fit within an <Item> element, or it has a sophisticated structure that needs to be retained, the extensibility features of the standard permit advanced users to satisfy those needs.

2.4.7 Guidelines for Creating a Delimited Data File from a NAACCR XML File

Some use cases for NAACCR XML data require a standardized method for creating a flattened version of a limited set of NAACCR XML data items. For example, some statistical software applications cannot input or output XML without difficulty, moreover, loading XML into a relational database often requires an intermediate flattening of the data to fit into tables. In these scenarios, when a software application needs to consume a flattened version of data items from a NAACCR XML document, delimited files are the best intermediate data format.

This recommendation should not be confused with the previous fixed-width NAACCR format, since a fixed-width file and a delimited file have different characteristics, namely, a delimited file does not require start positions or a specific order of data items. Since XML does not prescribe an order of elements and the NAACCR XML standard is designed to promote deep, hierarchical models of data where the length and presence of large sections of the XML cannot be predicted, a delimited file is better suited for a flattened representation of NAACCR XML data. The added flexibility of the XML data model fits more closely with the flexible nature of delimited files where the presence and order of each item can be changed easily. The complexity of an XML data model will never translate efficiently and completely into a flattened format, but in scenarios where a limited number of data items or a limited level of the data hierarchy needs to be flattened, delimited files are the best choice.

In order to prevent the reintroduction of limitations and problems associated with the previous fixed-width data exchange format, while still addressing the need to flatten data in certain scenarios, the following guidelines should be followed:

- The delimited file should use a pipe character (|, ASCII 124) as the delimiter.
- Delimiter characters that are not meant to be used as delimiters must be escaped with a backslash (\, ASCII 92).
- Backslash characters that are not used for escaping a delimiter should be escaped into a double backslash (\\, ASCII 92 + ASCII 92).
- Line endings that denote the end of a record must contain a CR (ASCII 13) followed by a LF (ASCII 10).
- All CR and LF characters that are not part of a line ending must be removed or re-encoded in an application-specific manner.
- All text must be UTF-8 encoded.
- The delimited file must have a header line as the first line in the file.
- The header line must use naaccrIds as header names.
- The naaccrId header names can be in any order.
- There is no minimum or maximum number of naaccrIds that can be included in a delimited file.

The NAACCR XML Data Exchange standard was initially created with two levels of hierarchy in the data model, a Patient and a Tumor with the intention and extensibility to support additional levels in the future through the NAACCR Data Standards and Data Dictionary change management process and any custom user-defined expansions. Creating a delimited file from an XML file can neither represent the complexity of a hierarchical data model nor retain the integrity of text values that contain newline characters, but with those limitations in mind, a delimited version of a NAACCR XML file can be helpful where a small number of data items needs to be analyzed by software that does not natively support XML or when text values are not needed.

2.4.8 Guidelines for Creating New XML NAACCR IDs

Every time a new data item is created, a corresponding XML NAACCR ID needs to be generated. For data items that need to be included in a new NAACCR version (standard data items included in a base dictionary), NAACCR will generate an ID based on the name. But for non-standard items that need to be created as part of a new user-defined dictionary, it is the responsibility of the software or person creating the dictionary to provide the new IDs.

Several software applications already have the ability to create a user-defined dictionary via a Graphical User Interface, and some of them also provide a way to generate an XML NAACCR ID based on a data item name. In general, the following guidelines should be used to create an XML NAACCR ID from an item name:

- Start with the data item name, then remove spaces and other word separation characters.
- Identify the different words or parts of the name and upper-case them, then start the ID with a lower case character. For example, “Primary Site” would become “primarySite”.
- Consider adding a prefix that reflects the ownership of the data item. This step is considered good practice to avoid XML NAACCR ID conflicts. For example, the data items own by a state “XX” could all start with “xx” or “xxState”.
- Remove special characters. XML NAACCR ID should only contain letters and digits, while item names can contain other characters. For example, parentheses and ampersands should be removed. When removing special characters, consider if the words separated by the special characters should be retained in the XML NAACCR ID or not.
- Ensure the length of the XML NAACCR ID does not exceed 32 characters. Item names can be 50 characters but the XML NAACCR ID for an item can only be 32 characters. If the final XML NAACCR ID is too long, consider replacing some of the words by an abbreviation.

These steps are only guidelines, the XML NAACCR ID needs to be valid and should be close enough to the item name so that there is no confusion with other data items.

2.4.9 Compression

Because the NAACCR XML format tends to generate large data files, compression is an important part of the standard. Large XML files should be compressed with either the ZIP (https://en.wikipedia.org/wiki/Zip_(file_format))or GZip (https://en.wikipedia.org/wiki/Gzip) compression methods. Also, software that is written to consume NAACCR XML files should be able to handle compressed or uncompressed files internally without requiring a user to first decompress the file. Files compressed with GZip should have the extension .xml.gz, and files compressed with ZIP should have the extension .zip.

3 Appendix A: Changes to the NAACCR XML Specifications

3.1 Changes introduced in version 1.1

- A new "allowUnlimitedText" attribute was introduced in the dictionary to allow some text items to not be bound by their length in the XML data files.
- Items defined in user-defined dictionaries do not have to provide a start column anymore.
- User-defined dictionaries do not have to provide a version anymore; if missing, the version will be assumed to be the same as the provided base dictionary.
- A new "specificationVersion" attribute was added to the dictionaries and data files, it defaults to 1.0 if not provided.

3.2 Changes introduced in version 1.2

- The "userDictionaryUri" appearing in the data file can now provide more than one dictionary URI by separating them with a space.
- The data type of several data items was relaxed.
- The "regexValidation" attribute was deprecated and removed from the dictionaries.
- A new "GroupedItemDefs" section was added to the base dictionaries, it defines the "GroupedItemDef" for every grouped item that NAACCR supports. Note that grouped items cannot appear in data files.

3.3 Changes introduced in version 1.3

- The range restriction (9500-99999) imposed on non-standard items defined in user dictionaries has been removed. Non-standard items can use any number as long as
    - The number is not already defined in the corresponding base dictionary.
    - The number is not already defined in another user dictionary when several dictionaries are provided for a given data file.

3.4 Changes introduced in version 1.4

- The maximum length for NAACCR IDs was changed from 50 to 32 characters. As a result, several NAACCR 18 IDs had to be renamed to follow this new requirement (the full list is provided in Appendix A). For software that create XML data files, it is recommended to start writing the new IDs as soon as possible. For software that consume XML data files, one of the following approaches can be taken:
    - Switch the software to only accept the new IDs and reject any data files that still use the old IDs (and if possible, contact the organization who created the file and request them to provide it again with the new IDs).
    - Switch the software to accept both IDs for a certain period of time (until it is clear that no organization will create data files with the old IDs).

4 Appendix B: Document History

Revision Date	Description
January 2021	• Updated table of dataTypes to include examples of valid and invalid values • Updated section 2.3.1 on modified records • Added new section 2.4.1 to describe blank values
February 2021	• Added clarity in 2.2.1.4 and updated examples in 2.4.1 for dealing with empty or blank item values • Added paragraph in 2.2.1.4 for escaping XML entity characters (&, <, >, ‘, “) as another reminder of what to escape