RTP index file validation metadata is represented as an XML document.

<indexFile>

This is the top-level element describing a single index file. It consists of a single <columns> element.

<columns>

The <columns> element consists of one or more <column> elements. Each <column> element contains the following:

- <index> - The index of this column. i.e. the position of this column relative to other columns.
- <type> - The data type of this column.
- <parsing> - Additional parsing information for this column.
- <required> - Whether or not this column is required to have a value. Defaults to false.
- <validations> - Consists of one or more <validation> elements (described below).

<type> / <parsing>

The column <type> can be one of the following:

- boolean – A Boolean value.
- integer – A 32-bit signed integer.
- string – An arbitrary string of characters.
- date – a local date with no time or time zone information.

Depending on the type, the `<parsing>` element can have different values, which describe how to parse the column data into the appropriate data type. Below is a list of parsing elements for each data type.

- boolean
  - <trueValue> - The string representing a “true” value.
    - e.g. “true”, “yes”, “Y”.
  - <falseValue> - The string representing a “false” value.
    - e.g. “false”, “no”, “N”.
- integer – No parsing information. Assumed to be a valid integer.
- string – No parsing information. Value is used as-is.
- date
  - <format> - A string describing the format of the date, using Java’s DateTimeFormatter.ofPattern(String) syntax.
    - e.g. “yyyy-MM-dd” for dates formatted like “2020-10-26”
    - e.g. “MMMM dd, yyyy” for dates formatted like “October 26, 2020”

<validations>

The <validations> element consists of one or more <validation> elements. Each <validation> contains the following:

- <type> - The validation type.

- <configuration> - The validation configuration.
- <condition> - The condition under which validation will occur. Optional; if no condition is specified, validation will always occur.

<type> / <configuration>

The validation <type> can be one of the following:

- length – The column value must be between a specified minimum and maximum length.
- regex – The column value must match a specified regular expression.
- integer-range – The column value must be an integer between a specified minimum and maximum.
- date-range The column value must be a date between a specified minimum and maximum.
- restricted-value – The column value must be one of a specified set of allowed values.

Depending on the type, the <configuration> element can have different values, which describe the exact validation which should be applied. Below is a list of parsing elements for each data type.

- length
  - <min> – The minimum allowed length. Optional.
  - <max> – The maximum allowed length. Optional.
- regex
  - pattern – The regular expression pattern which the value must match.
- integer-range
  - min – The minimum allowed value. Optional if <max> is specified.
  - max – The maximum allowed value. Optional if <min> is specified.
- date-range
  - min – The minimum allowed date in ISO 8601 extended date format (i.e. YYYY-MM-DD). Optional if <max> is specified.
  - min – The maximum allowed date in ISO 8601 extended date format (i.e. YYYY-MM-DD). Optional if <min> is specified.
- restricted-value
  - allowedValues – Contains one or more <value> elements.
    - value – An allowed value.

<condition>

There are two “base” condition types, <equals> and <pattern>, which can be combined using Boolean logic.

The <equals> condition checks that a column value is exactly equal to a specified value. The <pattern> condition checks that a column value matches a specified regular expression pattern. Both conditions consist of a <column> element and a <value> element. The <column> represents the index of the column value being checked. The value represents either the specific value being compared against (for <equals> conditions) or the pattern being compared against (for <pattern> conditions).

These two conditions can be combined using Boolean logic by wrapping them in <and>, <or>, and <not> elements. For example, the following represents the condition that the value of column 1 must be exactly equal to “Example” *and* the value of column 2 must be an upper or lower case “F” followed by two or more lower case “O”s.

Example

Below is example which partially demonstrates each of the features.

```xml
<indexFile>
  <columns>
    <!-- Integer which must be 0 or greater. -->
    <column>
      <index>0</index>
      <type>integer</type>
      <required>true</required>
      <validations>
        <validation>
          <type>integer-range</type>
          <configuration>
            <min>0</min>
          </configuration>
        </validation>
      </validations>
    </column>
    <!-- Optional string which can have any value. -->
    <column>
      <index>1</index>
      <type>string</type>
    </column>
    <!-- Required string which must have the value "Foo", "Bar", or "Baz". -->
    <column>
      <index>2</index>
      <type>string</type>
      <required>true</required>
      <validations>
        <validation>
          <type>restricted-value</type>
          <configuration>
            <allowedValues>
              <value>Foo</value>
              <value>Bar</value>
              <value>Baz</value>
            </allowedValues>
          </configuration>
        </validation>
      </validations>
    </column>
    <!-- A required date which must be formatted like "January 01, 1970".
         Must be between 2000-01-01 and 2010-12-31, if column 1 matches a
         particular regex and column 2 is "Foo". -->
    <column>
      <index>3</index>
      <type>date</type>
      <parsing>
        <format>MMMM dd, yyyy</format>
      </parsing>
      <required>true</required>
      <validations>
        <type>date-range</type>
        <configuration>
          <min>2000-01-01</min>
          <max>2010-12-31</max>
        </configuration>
        <condition>
          <and>
            <pattern>
              <column>1</column>
              <value>[Ff]oo+</value>
            </pattern>
            <equals>
              <column>2</column>
              <value>Foo</value>
            </equals>
          </and>
        </condition>
      </validations>
    </column>
  </columns>
</indexFile>
```