

Conventions used when converting XML to JSON and JSON to XML

PRODUCT: qRules
LAST UPDATED: Sept 22, 2015

JsonToXml	1
XmlToJson	3

JsonToXml

The JsonToXml command takes a JSON string as input, converts it to XML, and inserts the converted XML into one of the form's data sources. The converted XML has a single element at its root, and the children of this element are inserted in the location indicated by the command parameters. In other words, the root element itself is discarded.

This command is essentially the reverse of the process that XmlToJson uses, but it employs additional conventions to ensure a deterministic conversion from JSON to XML. The main difference is that while the **type=** attribute is sometimes optional and element names are not always meaningful, JsonToXml uses a **type=** attribute on all elements that have a type, and employs specific naming conventions.

Converting JSON types to their target xml

null – Null values are converted to an element with a **null=** attribute that has the value **true**.

numbers, strings, Booleans – These types are converted to an element with a **type=** attribute that has the value **number**, **string**, or **boolean**, respectively. Boolean values are converted to the value **true** or **false**.

objects – Objects are converted to an element containing a series of XML elements, one for each property on the object. The elements are sequenced in the alphabetical order of the original property names.

If any of the object's property names contains characters that would be invalid in an element's name, these characters are escaped to the form **_xNNNN_** where NNNN is the character's numeric code in hexadecimal. If the property name is an empty string, this is represented with the element name **_x0000_**.

If an element's name requires escaping, a **name=** attribute with the property's actual name is included on the element. For example the property

"It's good": true



would be represented as:

```
<lt_x0027_s_x0020_good name="It's good">true</ lt_x0027_s_x0020_good>
```

arrays – Arrays are converted to an element containing a series of XML elements, one for each item in the array. The name used for the item's elements is the name of the array's main element, plus **Item**.

Example

Source JSON:

```
{
  "Details": {
    "Name": "Fred Astaire",
    "Phone": "212-555-1212",
    "Fax": null,
    "RemainingBalance": 222.11,
    "Active": true
  },
  "Orders": [
    {
      "Id": "122",
      "Amount": 999.11
    },
    {
      "Id": "221",
      "Amount": 333.11
    }
  ]
}
```

Result XML:

In this example, the <Customer> example would be an existing element in the data source where the converted JSON is inserted. Its name is not based on the source JSON.

```
<Customer type="object">
  <Details type="object">
    <Name type="string">Fred Astaire</Name>
    <Phone type="string">212-555-1212</Phone>
    <Fax null="true" />
    <RemainingBalance type="number">222.11</RemainingBalance>
    <Active type="boolean">true</Active>
  </Details>
  <Orders type="array">
    <OrdersItem type="object">
      <Id type="string">122</Id>
      <Amount type="number">999.11</Amount>
    </Order>
    <Order type="object">
```



```

    <Id type="string">221</Id>
    <Amount type="number">333.11</Amount>
  </Order>
</Orders>
</Customer>

```

XmlToJson

The XmlToJson command starts at the indicated XML node and recursively traverses its contents, converting them into JSON. It determines the type to which each node should be converted and converts each accordingly.

Determining Type

If the node has an attribute with a local-name of **null** and a value of **true** or **1**, the node is converted to a JSON **null** value. This takes precedence over any other criteria for determining type.

If the node has an attribute with a local-name of **type**, the indicated type is used. Allowed values are **array**, **object**, **number**, **boolean**, and **string**. Any value other than the first four will be treated as **string** if the **type** attribute is present.

If no **type** attribute is present, the type is inferred as follows.

If the node contains any child elements, the type is assumed to be **object**.

If not, the type is assumed to be **string**.

Converting Nodes to their Target Type

The following describes the process used in converting nodes to their determined type.

string – The string value of the node is used as a JSON string. E.g. the element `<MyValue type="string">a<Inner>b</Inner>c</MyValue>` would result in the JSON "abc".

boolean – The node's string value is converted to a Boolean value and an error is thrown if it is not valid (allowed values are **true**, **false**, **1**, or **0**). A **null** value is used if the string value is blank.

number – The node's string value is converted to a number and an error is thrown if it is not valid. A **null** value is used if the string value is blank.

object – All of the node's child elements are converted to JSON values according to their respective type, and included as a property in a JSON object. The property name for each value is determined as follows:

If the element for the property has a **name=** attribute, that value is used as the property name.

Otherwise, the element's local-name is used as the property name.



The command will fail with an error if the object would result in having duplicate property names.

array – All of the node's child elements is converted to a JSON value according to their respective type and included in a JSON array. The child elements' names have no effect on this process.

Example

Source XML:

```
<Customer>
  <Details>
    <Name>Fred Astaire</Name>
    <Phone>212-555-1212</Phone>
    <Fax null="true" />
    <RemainingBalance type="number">222.11</RemainingBalance>
    <Active type="boolean">true</Active>
  </Details>
  <Orders type="array">
    <Order>
      <Id>122</Id>
      <Amount type="number">999.11</Amount>
    </Order>
    <Order>
      <Id>221</Id>
      <Amount type="number">333.11</Amount>
    </Order>
  </Orders>
</Customer>
```

Result JSON:

```
{
  "Details": {
    "Name": "Fred Astaire",
    "Phone": "212-555-1212",
    "Fax": null,
    "RemainingBalance": 222.11,
    "Active": true
  },
  "Orders": [
    {
      "Id": "122",
      "Amount": 999.11
    },
    {
      "Id": "221",
      "Amount": 333.11
    }
  ]
}
```



] }
}

