NAMELIST-XML Translator Usage Guide

O. Naito

November 18, 2016

1 Introduction

NAMELIST-XML Translator (nml2xml) is a browser-based tool for conversion between Fortran NAMELIST and XML/HTML. It should run on any modern HTML5 compliant browser with JavaScript enabled.

The tool consists of an HTML file (nml2xml.html) and a JavaScript source file (nml2xml.js), both of which must be in the same directory.

2 Window Layout

After opening nml2xml.html with the browser of the user's choice, the following window appears.

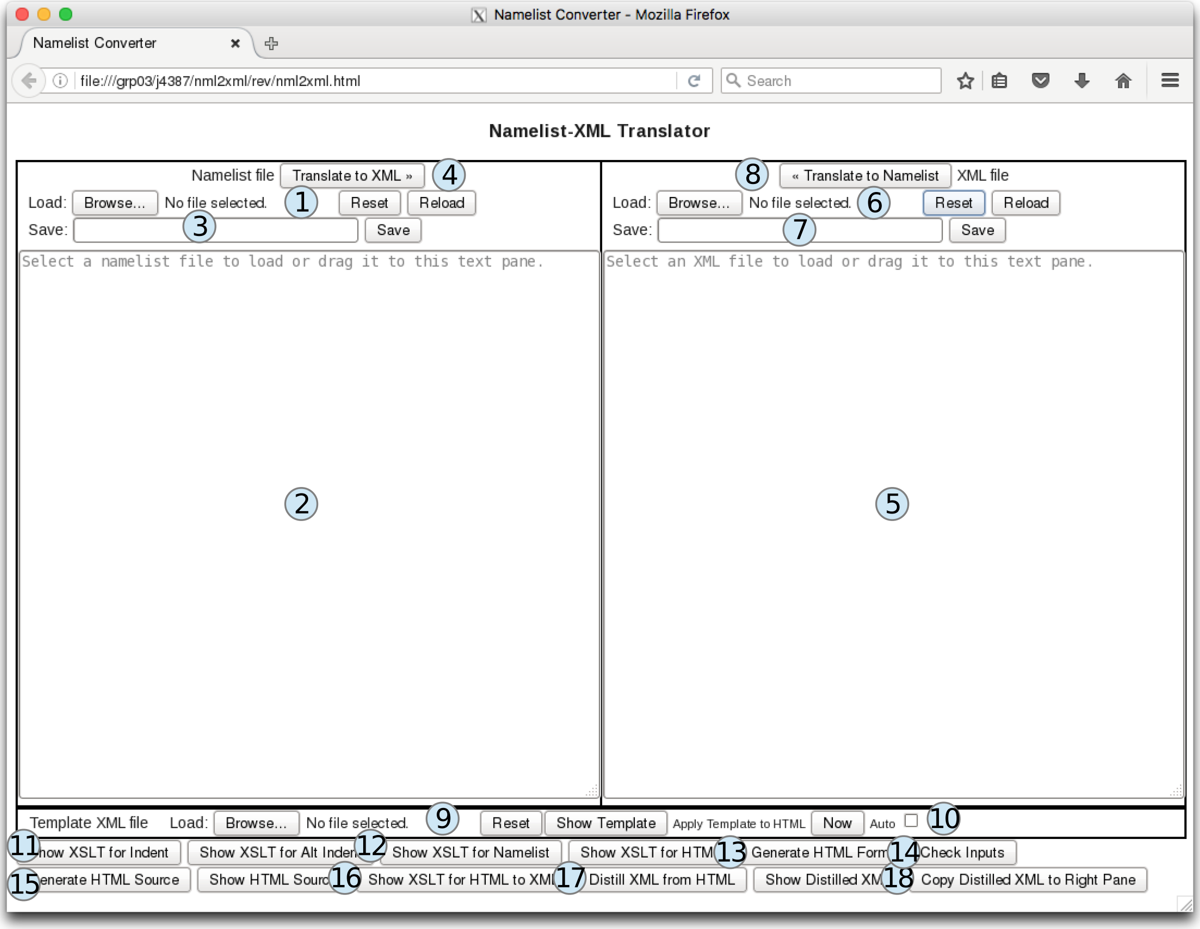


Figure 1: Window layout of this tool.

The function of each button is as follows:

① Use this file selector to load a NAMELIST file. The file will be loaded to the text pane ②. In some browsers, the selected file may be cached, and even if the NAMELIST file is modified, the old contents may persist. In that case, click the Reset button, and the cached file and the text in ② are cleared. When the Reload button is clicked, the contents are reloaded from the `NAMELIST` file.

② The contents of the `NAMELIST` file appear in this text pane. If the browser is equipped with drag-and-drop functionality, the `NAMELIST` file can be dragged over to this pane to be loaded.

③ To save the contents of text pane ② to a file, enter the file name to be saved as and click the Save button. For some browsers that do not allow writing to local files, just copy the contents and paste to a plain text file using some text editors.

④ Click this Translate to XML » button to convert the loaded `NAMELIST` to an XML document. The result will appear in the right text pane ⑤.

⑤ The contents of the XML document appear in this text pane. You can drag an existing XML file over to this pane to be loaded if that functionality is available.

⑥ Use this file selector to load an existing XML file. The contents will appear in the text pane ⑤.

⑦ To save the contents of text pane ⑤ to a file, enter the file name to be saved as and click the Save button.

⑧ Click this «Translate to Namelist button to convert the XML document back to a plain `NAMELIST` document. The result will appear in the left text pane ②.

⑨ Use this file selector to select a template XML file that is to be applied to the currently displayed HTML data editing form. The template may contain attributes of *group*s and *object*s that are not easy to store in a plain `NAMELIST` file, such as the descriptions of the *group*s and *object*s, data types and units of *object*s. Those auxiliary attributes are copied to the HTML data editing form when the Now button in ⑩ is clicked or the Auto checkbox is checked.

⑩ Click the Now button to apply the contents of the template to the current HTML data editing form. If Auto checkbox is checked, the template is applied automatically every time the HTML data editing form is generated.

⑪ Click this Show XSLT for Indent or Show XSLT for Alt Indent to see the XSLT sources for pretty printing the XML document with an XSLT processor that is or isn't equipped with an indentation functionality, respectively.

⑫ Click this Show XSLT for Namelist to see the XSLT source for converting an XML document to a plain `NAMELIST` document. Click the Show XSLT for HTML to see the XSLT source for converting an XML document to an HTML data editing form.

⑬ Click this Generate HTML Form button to generate an HTML data editing form from the XML document.

⑭ Click this Check Inputs button to check the data types of loaded values in the HTML data editing form against the currently selected data types in the drop-down lists described later in ㉖.

⑮ Click this Generate HTML Source button to generate an HTML text source for the currently displayed HTML data editing form and click the Show HTML Source button to see that source.

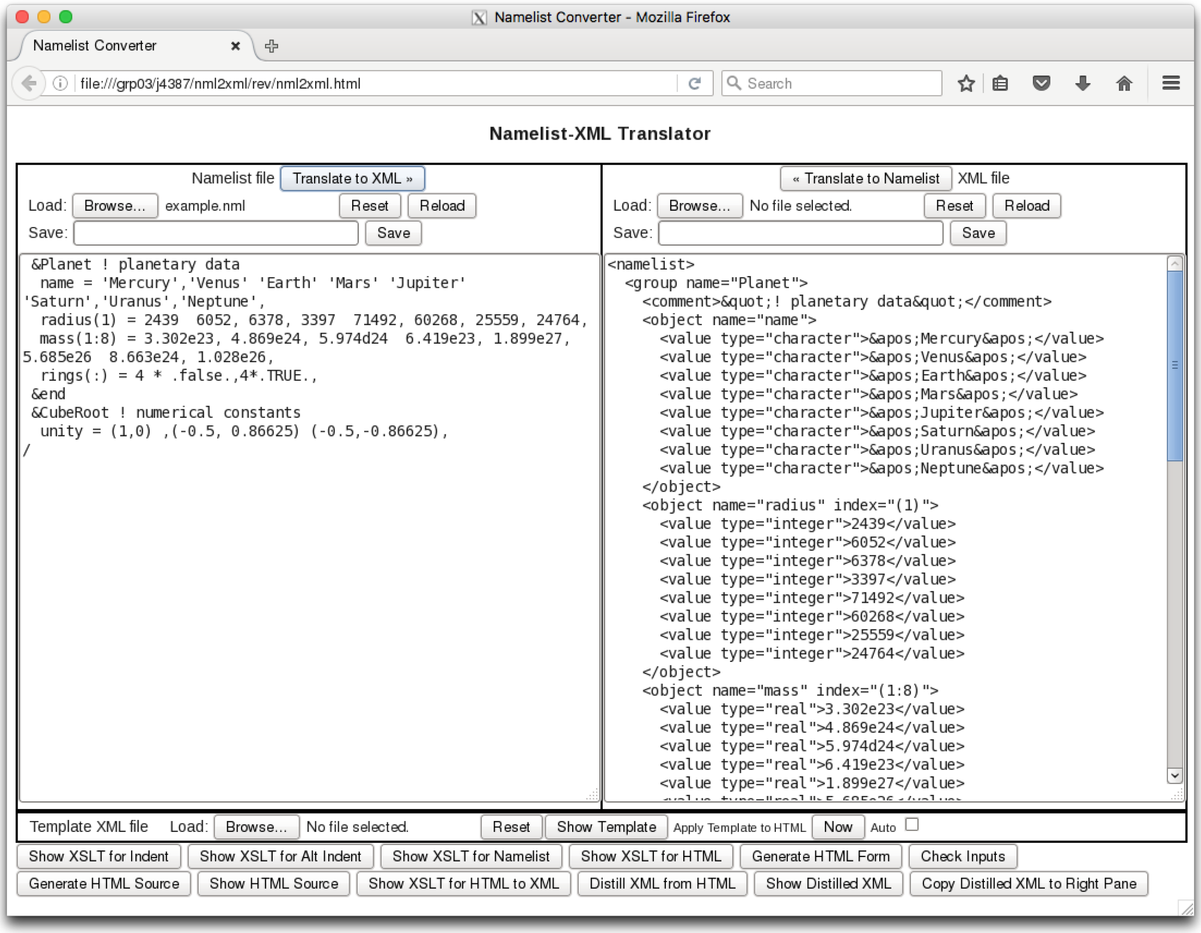
⑯ Click this Show XSLT for HTML to XML button to see the XSLT source for distilling the contents of the HTML data editing form into an XML document.

⑰ Click this Distill XML from HTML button to extract the information from the currently displayed HTML data editing form as an XML document. Click the Show Distilled XML button to see the resultant XML contents.

⑱ Click this Copy Distilled XML to Right Pane button to copy the extracted XML contents to the upper-right text pane ⑤. Once copied, this XML document can be saved to an external file using the upper-right Save button.

3 Example Session

When the accompanying example NAMELIST file (example.nml) is loaded with the upper-left file selector and the Translate to XML » button is clicked, the window will look like this:



Load: example.nml

Save:

```
 &Planet ! planetary data
  name = 'Mercury','Venus' 'Earth' 'Mars' 'Jupiter'
'Saturn','Uranus','Neptune',
  radius(1) = 2439  6052, 6378, 3397  71492, 60268, 25559, 24764,
  mass(1:8) = 3.302e23, 4.869e24, 5.974d24  6.419e23, 1.899e27,
5.685e26  8.663e24, 1.028e26,
  rings(:) = 4 * .false.,4*.TRUE.,
 &end
 &CubeRoot ! numerical constants
  unity = (1,0) ,(-0.5, 0.86625) (-0.5,-0.86625),
/
```

Load: No file selected.

Save:

```
<namelist>
  <group name="Planet">
    <comment>&quot;! planetary data&quot;</comment>
    <object name="name">
      <value type="character">&apos;Mercury&apos;</value>
      <value type="character">&apos;Venus&apos;</value>
      <value type="character">&apos;Earth&apos;</value>
      <value type="character">&apos;Mars&apos;</value>
      <value type="character">&apos;Jupiter&apos;</value>
      <value type="character">&apos;Saturn&apos;</value>
      <value type="character">&apos;Uranus&apos;</value>
      <value type="character">&apos;Neptune&apos;</value>
    </object>
    <object name="radius" index="(1)">
      <value type="integer">2439</value>
      <value type="integer">6052</value>
      <value type="integer">6378</value>
      <value type="integer">3397</value>
      <value type="integer">71492</value>
      <value type="integer">60268</value>
      <value type="integer">25559</value>
      <value type="integer">24764</value>
    </object>
    <object name="mass" index="(1:8)">
      <value type="real">3.302e23</value>
      <value type="real">4.869e24</value>
      <value type="real">5.974d24</value>
      <value type="real">6.419e23</value>
      <value type="real">1.899e27</value>
```

Template XML file Load: No file selected.

Figure 2: Window after NAMELIST is loaded and converted.

When the Generate HTML Form button is clicked, the HTML data editing form as shown in Fig. 3 appears. The description of each part is as follows:

⑲ Index to *groups* and *objects*. By clicking any link, the user can jump to the corresponding *group* or *object*.

⑳ The name of this *group*.

㉑ An editable text area for describing the summary of this *group*.

㉒ An editable text area for describing the detail of this *group*. Initially this area is hidden. The editable area appears by clicking the right-pointing triangle toggle button ▷. To hide it, click the down-pointing triangle ▽.

㉓ An editable text area for describing the summary of this *object*.

㉔ An editable text area for describing the detail of this *object*.

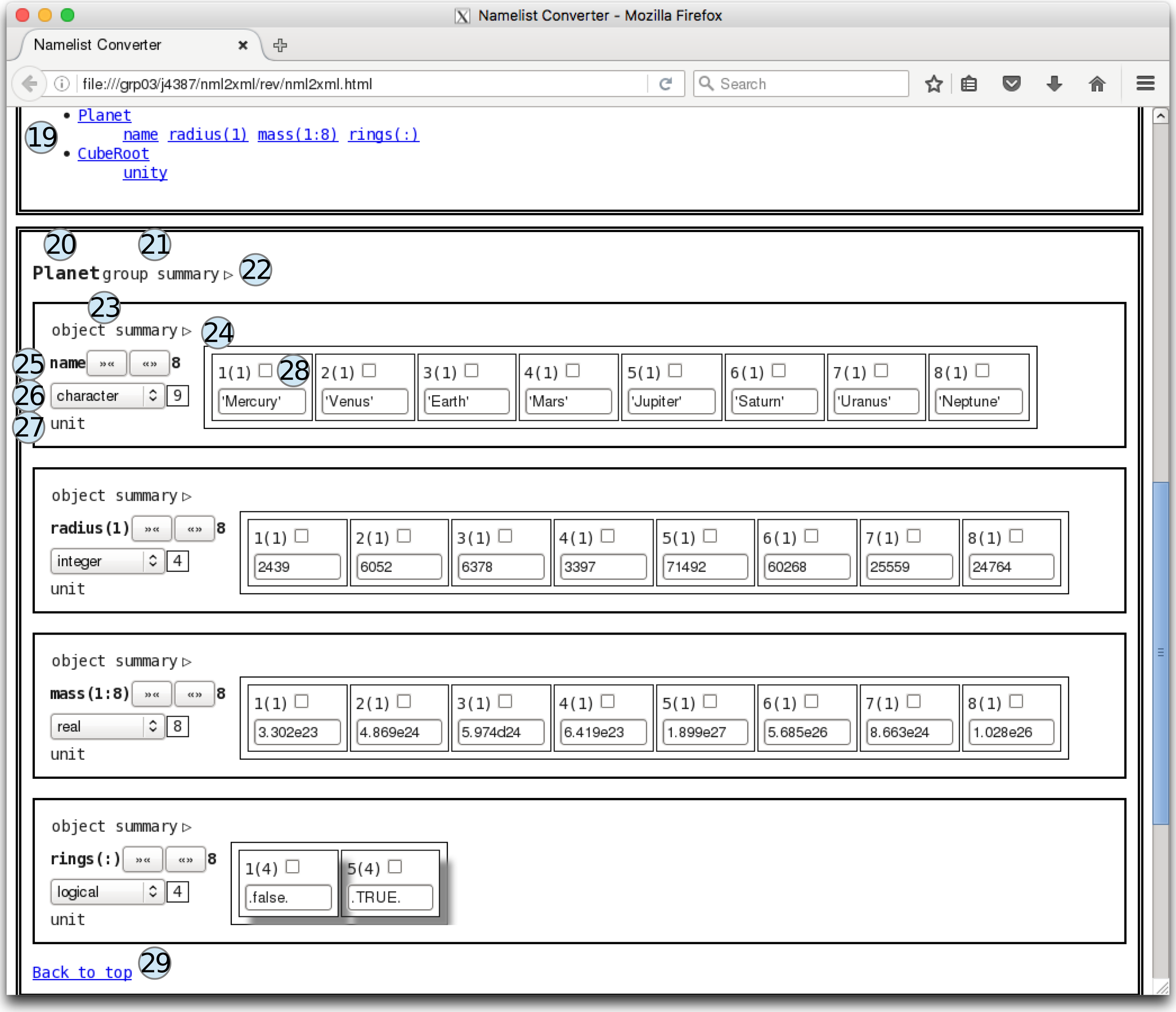
㉕ The name of this *object*. If the *object* is an array, its index is also shown. To the right are the join »« button and the disjoin «» button to bundle or unbundle the selected value cells (the detail will be shown later in this section). The rightmost number indicated the total number of *value* elements in this *object*.

㉖ The data type of this *object*. Initially the most plausible data type is shown here. If the data type is not what it should, the user can choose the correct type from this drop-down list. The data types of the *value* cells to the right are validated against the data type chosen in this drop-down list. Note that once the data type of the *object* is set as complex, its real and imaginary parts are assumed to have the data type "real". The number in the framed box is the data size of each *value* element in bytes.

㉗ An editable area for specifying the measurement unit of this *object*.

㉘ Data input cell for *value* element. The upper-left number is the sequential serial number in this *object*. The number in parentheses shows the multiplicity of this *value* cell. Raised patterns are used when the *value* cells are bundled. The checkbox is used to select which *value* cells are to be joined or disjoined. The bottom input box is for editing the data values.

㉙ By clicking this Back to top link, the user can jump back to the top index.


Namelist Converter - Mozilla Firefox
Namelist Converter
file:///grp03/j4387/nml2xml/rev/nml2xml.html
Search
19
20
21
22
23
24
25
26
27
28
29

- Planet
  name radius(1) mass(1:8) rings(:)
- CubeRoot
  unity

Planet group summary ▷

object summary ▷

name »« «» 8
character 9
unit

1(1) □	2(1) □	3(1) □	4(1) □	5(1) □	6(1) □	7(1) □	8(1) □
'Mercury'	'Venus'	'Earth'	'Mars'	'Jupiter'	'Saturn'	'Uranus'	'Neptune'

object summary ▷

radius(1) »« «» 8
integer 4
unit

1(1) □	2(1) □	3(1) □	4(1) □	5(1) □	6(1) □	7(1) □	8(1) □
2439	6052	6378	3397	71492	60268	25559	24764

object summary ▷

mass(1:8) »« «» 8
real 8
unit

1(1) □	2(1) □	3(1) □	4(1) □	5(1) □	6(1) □	7(1) □	8(1) □
3.302e23	4.869e24	5.974d24	6.419e23	1.899e27	5.685e26	8.663e24	1.028e26

object summary ▷

rings(:) »« «» 8
logical 4
unit

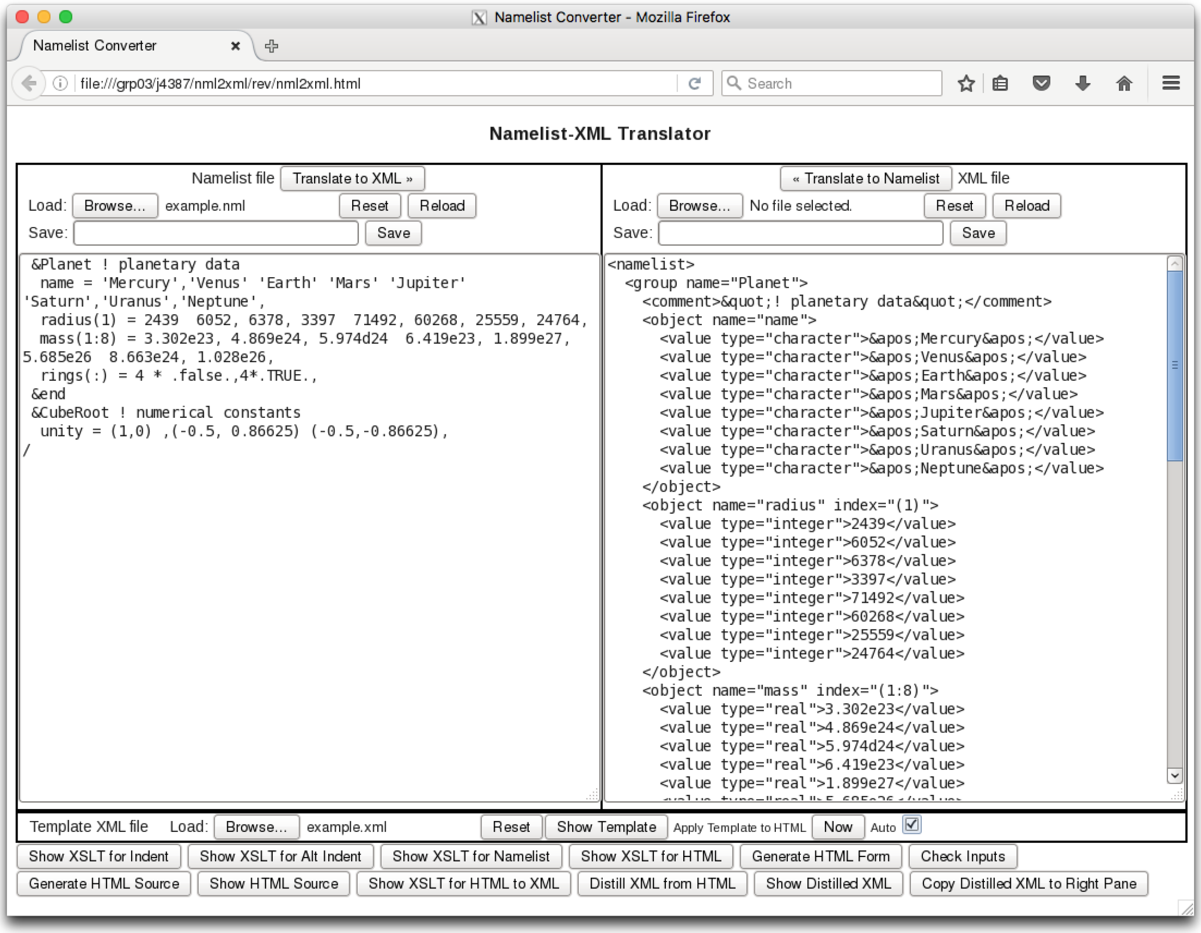
1(4) □	5(4) □
.false.	.TRUE.

Back to top

Figure 3: HTML data editing form generated from XML.

After editing the form, click the Distill XML from HTML button (and if the user wants to check the result, click the Show Distilled XML button), then click the Copy Distilled XML to Right Pane button to copy the result to the top-right text pane. Click the « Translate to Namelist button to convert the result back to a plain NAMELIST text. Save the NAMELIST file or XML file as necessary. If the XML file is saved, the user can start the session later directly from that XML file by loading it, or if the XML file is populated with auxiliary information, it can be used as a template.

For example, in the case as shown in Fig. 4, the original plain NAMELIST file is loaded and converted to an XML document with this tool. By specifying the template file (example.xml) with the Auto checkbox checked and clicking the Generate HTML Form button, the HTML form as shown in Fig. 5 will appear. Note that pieces of auxiliary information like summary and detail descriptions of *groups* and *objects* are copied from the template XML file.



Load: example.nml

Save:

```
 &Planet ! planetary data
  name = 'Mercury','Venus' 'Earth' 'Mars' 'Jupiter'
'Saturn','Uranus','Neptune',
  radius(1) = 2439  6052, 6378, 3397  71492, 60268, 25559, 24764,
  mass(1:8) = 3.302e23, 4.869e24, 5.974d24  6.419e23, 1.899e27,
5.685e26  8.663e24, 1.028e26,
  rings(:) = 4 * .false.,4*.TRUE.,
 &end
 &CubeRoot ! numerical constants
  unity = (1,0) ,(-0.5, 0.86625) (-0.5,-0.86625),
/
```

Load: No file selected.

Save:

```
<namelist>
  <group name="Planet">
    <comment>&quot;! planetary data&quot;</comment>
    <object name="name">
      <value type="character">&apos;Mercury&apos;</value>
      <value type="character">&apos;Venus&apos;</value>
      <value type="character">&apos;Earth&apos;</value>
      <value type="character">&apos;Mars&apos;</value>
      <value type="character">&apos;Jupiter&apos;</value>
      <value type="character">&apos;Saturn&apos;</value>
      <value type="character">&apos;Uranus&apos;</value>
      <value type="character">&apos;Neptune&apos;</value>
    </object>
    <object name="radius" index="(1)">
      <value type="integer">2439</value>
      <value type="integer">6052</value>
      <value type="integer">6378</value>
      <value type="integer">3397</value>
      <value type="integer">71492</value>
      <value type="integer">60268</value>
      <value type="integer">25559</value>
      <value type="integer">24764</value>
    </object>
    <object name="mass" index="(1:8)">
      <value type="real">3.302e23</value>
      <value type="real">4.869e24</value>
      <value type="real">5.974d24</value>
      <value type="real">6.419e23</value>
      <value type="real">1.899e27</value>
```

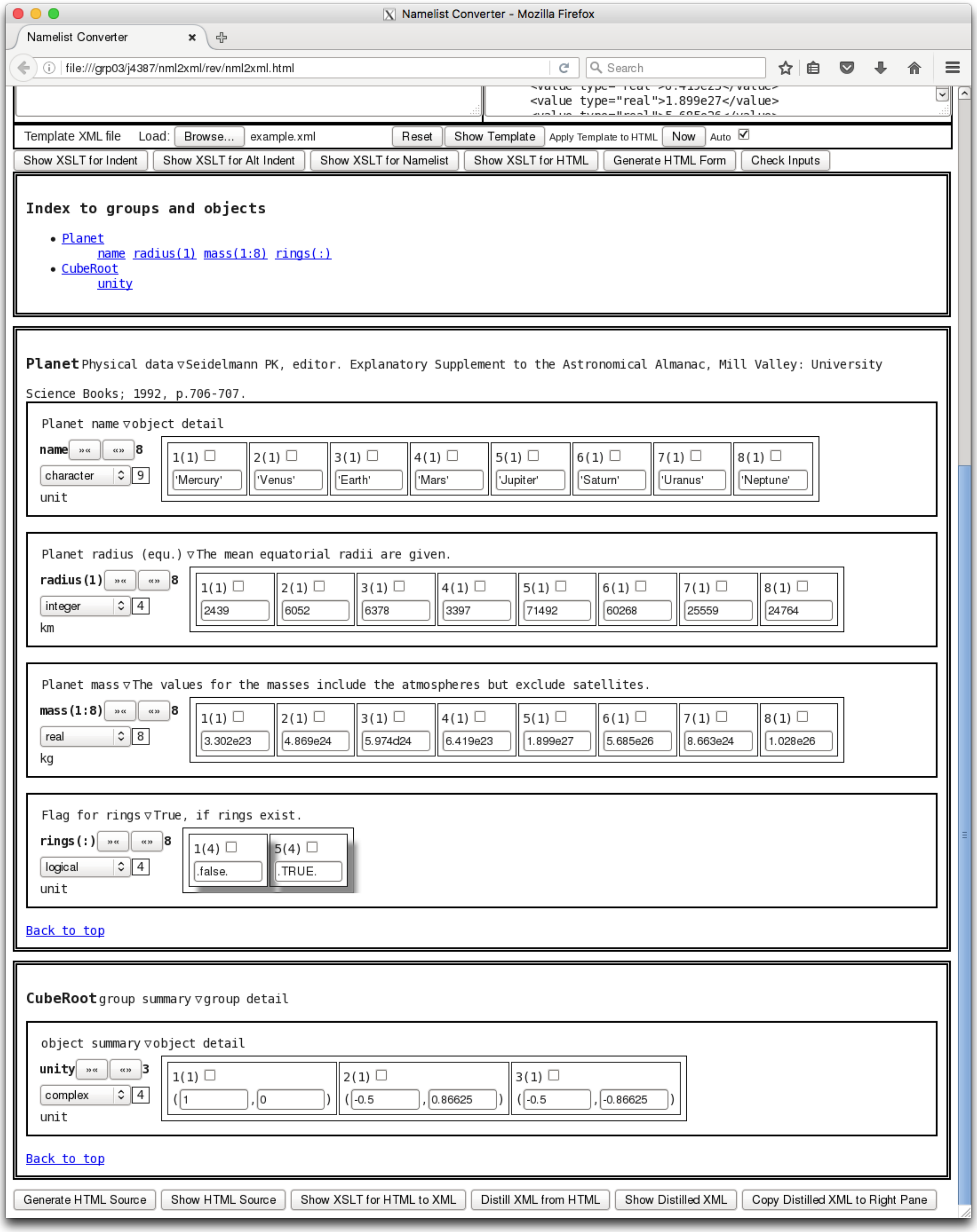
Template XML file Load: example.xml

Auto ☑

Figure 4: Window after NAMELIST is loaded and converted with the functionality of automatic application of template enabled.

The procedures to join or disjoin the *value* cells are shown in Fig. 6: 1) To disjoin the bundled *value* cell, check the checkbox of that cell and click the disjoin «» button. Then *value* cells with the same values are copied as many times as in the parentheses and the raised pattern is changed to a flat pattern. 2) To combine the cells to a bundled cell, check the cells to be combined and click the the join »« button. 3) Then a bundled cell with the value of leftmost checked cell is produced. The flat pattern is changed to a raised pattern and the multiplicity of that cell is shown in the parentheses. The corresponding Fortran NAMELIST statements are shown to the right of the figure.

When values of invalid data types are entered into the *value* cells, or detected while the validation is executed after clicking the Check Inputs button, an alert box pops up as shown in Fig. 7 and the values are reverted to the original ones.



```
<value type="real">1.899e27</value>
```

Template XML file: example.xml

Index to groups and objects

- Planet
  name radius(1) mass(1:8) rings(:)
- CubeRoot
  unity

Planet Physical data ▿Seidelmann PK, editor. Explanatory Supplement to the Astronomical Almanac, Mill Valley: University Science Books; 1992, p.706-707.

Planet name ▿object detail

name »« «» 8
character 9
unit

1(1) □	2(1) □	3(1) □	4(1) □	5(1) □	6(1) □	7(1) □	8(1) □
'Mercury'	'Venus'	'Earth'	'Mars'	'Jupiter'	'Saturn'	'Uranus'	'Neptune'

Planet radius (equ.) ▿The mean equatorial radii are given.

radius(1) »« «» 8
integer 4
km

1(1) □	2(1) □	3(1) □	4(1) □	5(1) □	6(1) □	7(1) □	8(1) □
2439	6052	6378	3397	71492	60268	25559	24764

Planet mass ▿The values for the masses include the atmospheres but exclude satellites.

mass(1:8) »« «» 8
real 8
kg

1(1) □	2(1) □	3(1) □	4(1) □	5(1) □	6(1) □	7(1) □	8(1) □
3.302e23	4.869e24	5.974d24	6.419e23	1.899e27	5.685e26	8.663e24	1.028e26

Flag for rings ▿True, if rings exist.

rings(:) »« «» 8
logical 4
unit

1(4) □	5(4) □
.false.	.TRUE.

Back to top

CubeRoot group summary ▿group detail

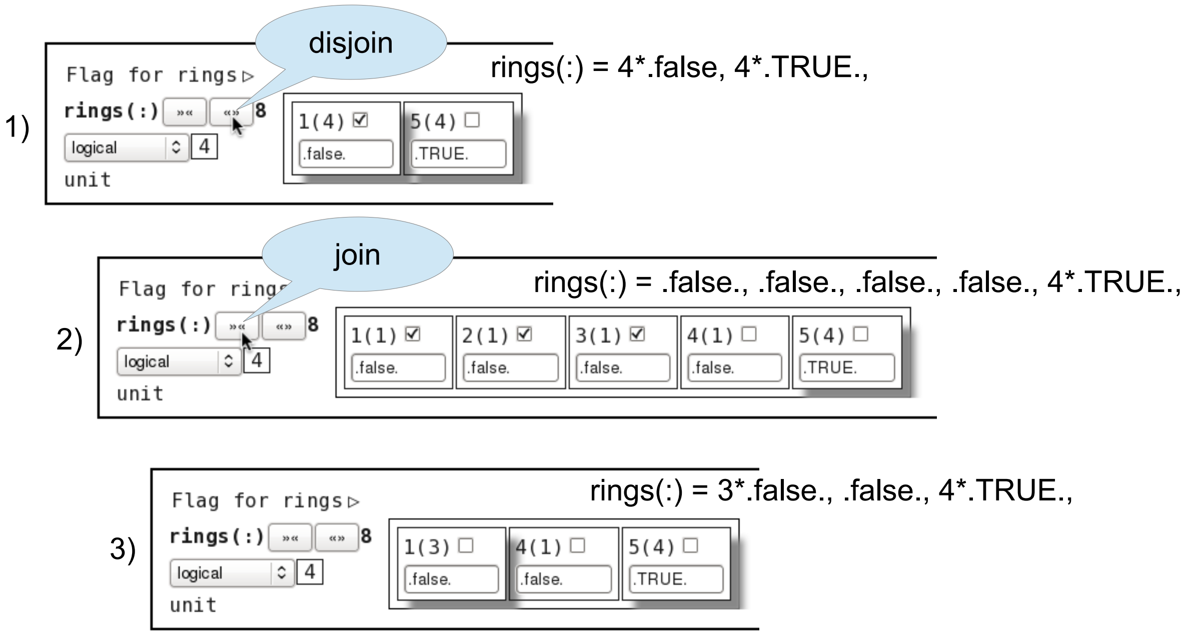
object summary ▿object detail

unity »« «» 3
complex 4
unit

1(1) □	2(1) □	3(1) □
(1 , 0)	(-0.5 , 0.86625)	(-0.5 , -0.86625)

Back to top

Figure 5: HTML data editing form generated from XML and populated with auxiliary information from the template XML file.



1) Flag for rings ▷
rings(:) »« «» 8
logical 4
unit
1(4) ☑ .false.
5(4) □ .TRUE.

rings(:) = 4*.false, 4*.TRUE.,

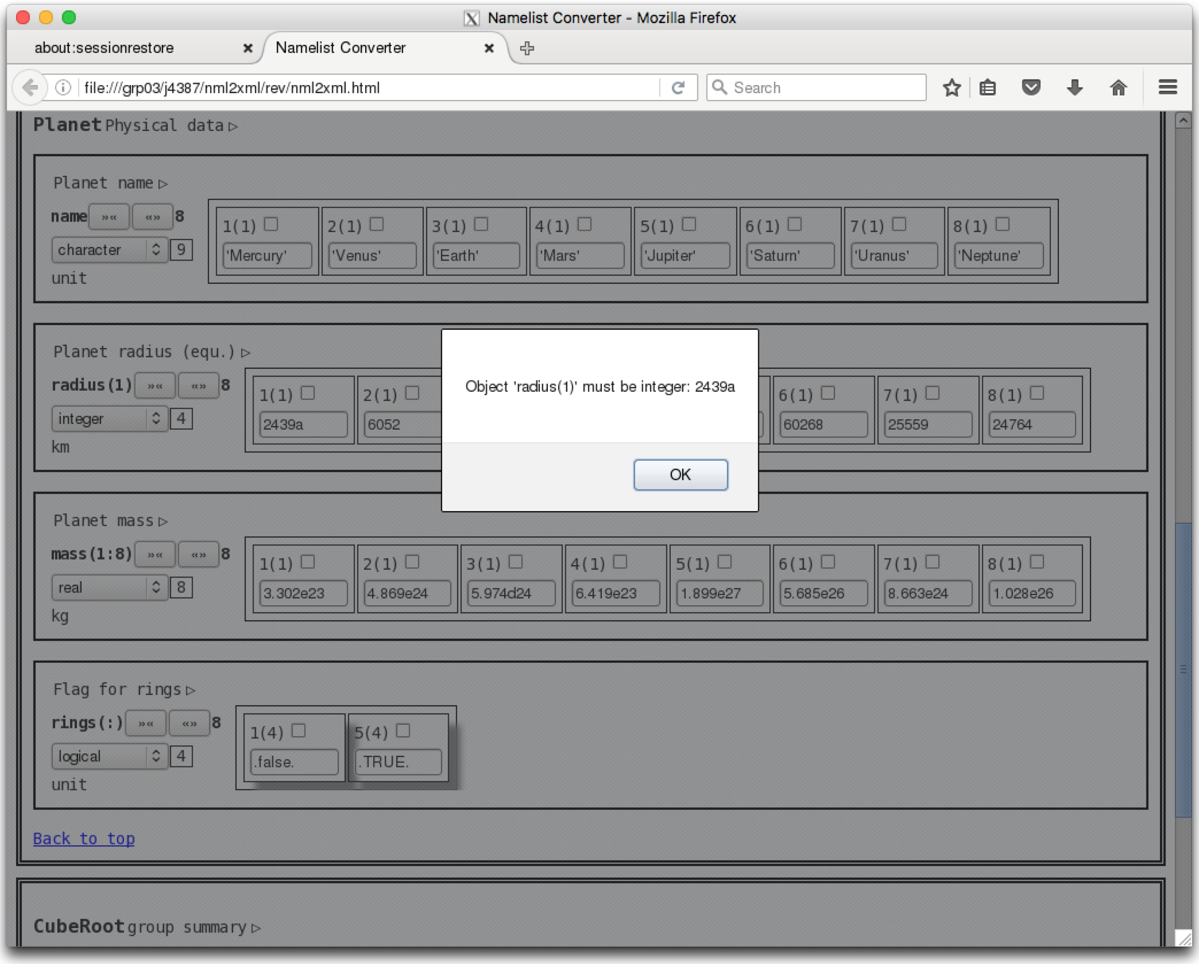
2) Flag for rings
rings(:) »« «» 8
logical 4
unit
1(1) ☑ .false.
2(1) ☑ .false.
3(1) ☑ .false.
4(1) □ .false.
5(4) □ .TRUE.

rings(:) = .false., .false., .false., .false., 4*.TRUE.,

3) Flag for rings ▷
rings(:) »« «» 8
logical 4
unit
1(3) □ .false.
4(1) □ .false.
5(4) □ .TRUE.

rings(:) = 3*.false., .false., 4*.TRUE.,

Figure 6: Procedures to join or disjoin the *value* cells. Equivalent Fortran `NAMELIST` statements are shown to the right.



file:///grp03/j4387/nml2xml/rev/nml2xml.html

Planet Physical data ▷

Planet name ▷
name »« «» 8
character 9
unit
1(1) □ 'Mercury'
2(1) □ 'Venus'
3(1) □ 'Earth'
4(1) □ 'Mars'
5(1) □ 'Jupiter'
6(1) □ 'Saturn'
7(1) □ 'Uranus'
8(1) □ 'Neptune'

Planet radius (equ.) ▷
radius(1) »« «» 8
integer 4
km
1(1) □ 2439a
2(1) □ 6052
6(1) □ 60268
7(1) □ 25559
8(1) □ 24764

Object 'radius(1)' must be integer: 2439a

OK

Planet mass ▷
mass(1:8) »« «» 8
real 8
kg
1(1) □ 3.302e23
2(1) □ 4.869e24
3(1) □ 5.974d24
4(1) □ 6.419e23
5(1) □ 1.899e27
6(1) □ 5.685e26
7(1) □ 8.663e24
8(1) □ 1.028e26

Flag for rings ▷
rings(:) »« «» 8
logical 4
unit
1(4) □ .false.
5(4) □ .TRUE.

Back to top

CubeRoot group summary ▷

Figure 7: An alert box pops up when invalid data type is detected.