

Atom Indonesia

Journal homepage: <http://atomindonesia.brin.go.id>

Supplementary Materials

Annexure – I

A Source Code written in Visual Basic is developed to convert the ST Data set from Excel Spreadsheet to a JRODOS-compatible XML formatted file.

	A	B	C	D	E	F	G
1	Reactor Type	VVER					
2	Input Type	6					
3	End of Reaction	0					
4	Volume Flux	-1	m ³ /s				
5	Vent Area	-1	m ²				
6	Thermal Energy	3200	MW				
7	Element Iodine	100	Total				
8	OrgBo Iodine	0	100%				
9	Aerosol Iodine	0					
10							
11	Release Rate	Bq/h					
12	Time	h					
13							
14	XML File Path	C:\Users\Admin\Desktop\SourceTerm.xml					
15							
16							
17							
18							
19							
20							

*Note: The Button “Generate XML File” is referred to as **CommandButton1** and the button “Plot ST” is referred to as **CommandButton2** in the source code given below. The XML file path must be a proper directory which should include the filename.xml*

```

-----This code was written by-----
-----S. M. Tabirul Hassan-----
-----Senior Engineer-----
-----Institute of Energy Science-----
-----Bangladesh Atomic Energy Commission-----
-----Date: May 2023-----
-----This code is open for all to use. Appropriate citation is requested before use-----

```

Option Explicit

```
Private Sub CommandButton1_Click()
```

```
Dim ReactorType As String, Height As Single, Thermal_Energy As Single, VolumeFlux As Single, VentArea As Single
```

```
Dim ElemIod As Single, OrgBoIod As Single, AeroIod As Single
```

```
ReactorType = Sheets("CreateXML").Range("B1")
VolumeFlux = Sheets("CreateXML").Range("B4")
VentArea = Sheets("CreateXML").Range("B5")
Thermal_Energy = Sheets("CreateXML").Range("B6")
ElemIod = Sheets("CreateXML").Range("B7")
OrgBoIod = Sheets("CreateXML").Range("B8")
AeroIod = Sheets("CreateXML").Range("B9")
```

```
Dim Total_TimeSteps As Single, Total_Nuclides As Single
```

```
Total_Nuclides = Sheets("SourceTerm").Cells(1, Columns.Count).End(xlToLeft).Column - 3
```

Total_TimeSteps = Sheets("SourceTerm").Cells(Rows.Count, 1).End(xlUp).Row - 2

Dim ModNuclide() As String, j As Integer

For j = 0 To Total_Nuclides

ModNuclide = Split(Replace(Sheets("SourceTerm").Range("C1").Offset(0, j), " ", ""), "-")

If Len(ModNuclide(0)) = 1 Then

ModNuclide(0) = ModNuclide(0) & " "

End If

If Len(ModNuclide(1)) = 2 Then

ModNuclide(1) = " " & ModNuclide(1)

End If

If Len(ModNuclide(1)) = 1 Then

ModNuclide(1) = " " & ModNuclide(1)

End If

Sheets("SourceTerm").Range("C1").Offset(0, j) = ModNuclide(0) & "-" & ModNuclide(1)

Next j

Dim XMLContent As String

XMLContent = "<?xml version=" & """"1.0"""" & " encoding=" & """"UTF-8"""" & "?>" & vbNewLine

XMLContent = XMLContent & "<root>" & vbNewLine & " <sourcetermData inputType="" &

Sheets("CreateXML").Range("B2") & """" & ">" & vbNewLine

XMLContent = XMLContent & " <label>UserDefined</label>" & vbNewLine & " <reactorType>" & ReactorType

& "</reactorType>" & vbNewLine

XMLContent = XMLContent & " <associatedSites />" & vbNewLine

XMLContent = XMLContent & " <comment />" & vbNewLine

XMLContent = XMLContent & " <endOfReaction>" & Sheets("CreateXML").Range("B3") & "</endOfReaction>" & vbNewLine

Dim Activity As Double, Nuclide As String, Start_Time As Single, End_Time As Single, i As Integer

Start_Time = 0

For i = 0 To Total_TimeSteps

End_Time = Sheets("SourceTerm").Range("A2").Offset(i, 0)

XMLContent = XMLContent & " <releaseTimeInterval start="" & Start_Time & """"

XMLContent = XMLContent & " end="" & End_Time & """" & ">" & vbNewLine

XMLContent = XMLContent & " <releaseAttribute height="" & Sheets("SourceTerm").Range("B1").Offset(i + 1, 0) & """"

XMLContent = XMLContent & " thermalEnergy="" & Thermal_Energy & """"

XMLContent = XMLContent & " volumeFlux="" & VolumeFlux & """"

XMLContent = XMLContent & " ventArea="" & VentArea & """" & ">" & vbNewLine

XMLContent = XMLContent & " <relativeIodineFraction elemIod="" & ElemIod & """"

XMLContent = XMLContent & " orgBoIod="" & OrgBoIod & """" & " aerosolIod="" & AeroIod & """" & ">" & vbNewLine

Start_Time = End_Time

For j = 0 To Total_Nuclides

XMLContent = XMLContent & " <nuclide name=" & """"

Nuclide = Sheets("SourceTerm").Range("C1").Offset(0, j)

XMLContent = XMLContent & Nuclide & """" & " value=""

```
Activity = Sheets("SourceTerm").Range("C1").Offset(i + 1, j)
XMLContent = XMLContent & Activity & " " & " />" & vbNewLine

Next j

XMLContent = XMLContent & " </releaseTimeInterval>" & vbNewLine

Next i

XMLContent = XMLContent & " </sourcetermData>" & vbNewLine
XMLContent = XMLContent & "</root>"

Dim FSO As Object, FileToCreate As Object, FileDirectory As String

FileDirectory = Sheets("CreateXML").Range("B14")

Set FSO = CreateObject("Scripting.FileSystemObject")
Set FileToCreate = FSO.CreateTextFile(FileDirectory, True)

FileToCreate.Write XMLContent
FileToCreate.Close

End Sub

Private Sub CommandButton2_Click()

Dim sht As Worksheet

For Each sht In Worksheets
    If sht.Name = "SourceTerm Plot" Then
        Application.DisplayAlerts = False
        Sheets("SourceTerm Plot").Delete
        Application.DisplayAlerts = True
    End If
Next sht

Sheets("SourceTerm").Copy After:=Sheets("SourceTerm")
ActiveSheet.Name = "SourceTerm Plot"

Worksheets("SourceTerm Plot").Columns("B").Delete
Sheets("SourceTerm Plot").Visible = False
Worksheets("SourceTerm").Activate

Dim LastCellAdrs As String, rw As Integer, cl As Integer

rw = Sheets("SourceTerm Plot").Cells(Rows.Count, 1).End(xlUp).Row
cl = Sheets("SourceTerm Plot").Cells(1, Columns.Count).End(xlToLeft).Column

LastCellAdrs = Sheets("SourceTerm Plot").Range("A1").Offset(rw - 1, cl - 1).Address
LastCellAdrs = Replace(LastCellAdrs, "$", "")

Sheets("SourceTerm Plot").Range("A1") = ""

Sheets("SourceTerm Plot").Range("A2:" & LastCellAdrs).Copy _
(Sheets("SourceTerm Plot").Range("A1").Offset(rw, 0))

LastCellAdrs = Sheets("SourceTerm Plot").Range("A" & rw).Offset(rw - 1, cl - 1).Address
LastCellAdrs = Replace(LastCellAdrs, "$", "")

Sheets("SourceTerm Plot").Range("A" & (rw + 1) & ":" & "A" & (2 * rw - 1)).Cut _
(Sheets("SourceTerm Plot").Range("A" & (rw + 2)))
```

```
Sheets("SourceTerm Plot").Range("A" & (rw + 1)).Value = 0  
Sheets("SourceTerm Plot").Range("A" & (2 * rw)).Value = ""
```

```
Sheets("SourceTerm Plot") _  
.Range("A1:" & LastCellAdrs).Sort Key1:=Sheets("SourceTerm Plot").Range("A2"), _  
Order1:=xlAscending, _  
Header:=xlYes
```

Dim Chrt As Object

```
Set Chrt = ActiveSheet.Shapes.AddChart2
```

```
Chrt.Chart.Parent.Height = 250  
Chrt.Chart.Parent.Width = 700  
Chrt.Chart.Parent.Left = 200  
Chrt.Chart.Parent.Top = 20
```

```
Chrt.Chart.SetSourceData Source:=Sheets("SourceTerm Plot").Range("A1:" & LastCellAdrs)
```

```
Chrt.Chart.PlotBy = xlColumns
```

```
Chrt.Chart.SetElement (msoElementPrimaryValueGridLinesMinorMajor)  
Chrt.Chart.SetElement (msoElementPrimaryCategoryGridLinesMinorMajor)
```

```
Chrt.Chart.SetElement (msoElementPrimaryValueAxisShow)  
Chrt.Chart.SetElement (msoElementPrimaryValueAxisTitleHorizontal)  
Chrt.Chart.SetElement (msoElementPrimaryCategoryAxisShow)  
Chrt.Chart.SetElement (msoElementPrimaryCategoryAxisTitleHorizontal)  
Chrt.Chart.SetElement (msoElementLegendRight)
```

```
Chrt.Chart.Axes(xlValue).AxisTitle.Text = Sheets("CreateXML").Range("A12") & " (" &  
Sheets("CreateXML").Range("B12") & ")"
```

```
Chrt.Chart.Axes(xlValue).AxisTitle.Orientation = xlUpward
```

```
Chrt.Chart.Axes(xlCategory).AxisTitle.Text = Sheets("CreateXML").Range("A13") & " (" &  
Sheets("CreateXML").Range("B13") & ")"
```

```
Chrt.Chart.ChartType = xlXYScatterLinesNoMarkers  
Chrt.Chart.Axes(xlValue).ScaleType = xlScaleLogarithmic  
Chrt.Chart.Axes(xlValue).MinimumScale = WorksheetFunction.Min(Sheets("SourceTerm Plot").Range("B2:" &  
LastCellAdrs))  
Chrt.Chart.Axes(xlValue).MaximumScale = WorksheetFunction.Max(Sheets("SourceTerm Plot").Range("B2:" &  
LastCellAdrs))  
Chrt.Chart.Axes(xlCategory).MinimumScale = WorksheetFunction.Min(Sheets("SourceTerm Plot").Range("A2:" &  
"A" & (2 * rw)))  
Chrt.Chart.Axes(xlCategory).MaximumScale = WorksheetFunction.Max(Sheets("SourceTerm Plot").Range("A2:" &  
"A" & (2 * rw)))  
Chrt.Chart.HasTitle = False  
Chrt.Chart.legend.Font.ColorIndex = 0
```

Dim srs As Series

```
For Each srs In Chrt.Chart.SeriesCollection
```

```
    srs.Format.Line.Weight = 0.75
```

```
Next srs
```

End Sub

A typical JRODOS compatible XML file is given below for a simple ST data set.

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <sourcetermData inputType="6">
    <label>UserDefined</label>
    <reactorType>VVER</reactorType>
    <associatedSites />
    <comment />
    <endOfReaction>0</endOfReaction>
    <releaseTimeInterval start="0" end="4.5">
      <releaseAttribute height="20" thermalEnergy="3200" volumeFlux="-1" ventArea="-1" />
      <relativeIodineFraction elemIod="100" orgBoIod="0" aerosolIod="0" />
      <nuclide name="Cs-137" value="3700000000000" />
      <nuclide name="I -131" value="3700000000000" />
      <nuclide name="Xe-133" value="1500000000000" />
    </releaseTimeInterval>
    <releaseTimeInterval start="4.5" end="10.5">
      <releaseAttribute height="20" thermalEnergy="3200" volumeFlux="-1" ventArea="-1" />
      <relativeIodineFraction elemIod="100" orgBoIod="0" aerosolIod="0" />
      <nuclide name="Cs-137" value="2700000000000" />
      <nuclide name="I -131" value="1700000000000" />
      <nuclide name="Xe-133" value="1100000000000" />
    </releaseTimeInterval>
    <releaseTimeInterval start="10.5" end="11">
      <releaseAttribute height="20" thermalEnergy="3200" volumeFlux="-1" ventArea="-1" />
      <relativeIodineFraction elemIod="100" orgBoIod="0" aerosolIod="0" />
      <nuclide name="Cs-137" value="1100000000000" />
      <nuclide name="I -131" value="3000000000000" />
      <nuclide name="Xe-133" value="2200000000000" />
    </releaseTimeInterval>
    <releaseTimeInterval start="11" end="21">
      <releaseAttribute height="20" thermalEnergy="3200" volumeFlux="-1" ventArea="-1" />
      <relativeIodineFraction elemIod="100" orgBoIod="0" aerosolIod="0" />
      <nuclide name="Cs-137" value="8400000000000" />
      <nuclide name="I -131" value="8400000000000" />
      <nuclide name="Xe-133" value="9000000000000" />
    </releaseTimeInterval>
  </sourcetermData>
</root>
```
