Saving/Retrieving Unicode Data in APL64

Contents

Overview

Application-specific text data may contain Unicode code points which are not in ⎕AV. As of APL64 version 2024.0.2.19263, the native and component file system functions and the wrapl algorithm only support text data with elements in ⎕AV. This limitation is present to preserve compatibility with APL+Win.

The APL64 ⎕NFE (Native Files with Encoding) system function fully supports Unicode-encoded text data.

The Unicode-encoded text data can include:

- Arbitrary Unicode text
- ⎕VR of an APL64 programmer-defined function
- XML- or json-format representation of an APL64 variable

An ⎕AV-created Unicode-encoded file may be compressed using the APL64 ⎕ZIP system function.

Read this document to see a simple example illustrating these APL64 features. These APL64 techniques may be incorporated into application-specific APL64 programmer-developed functions.

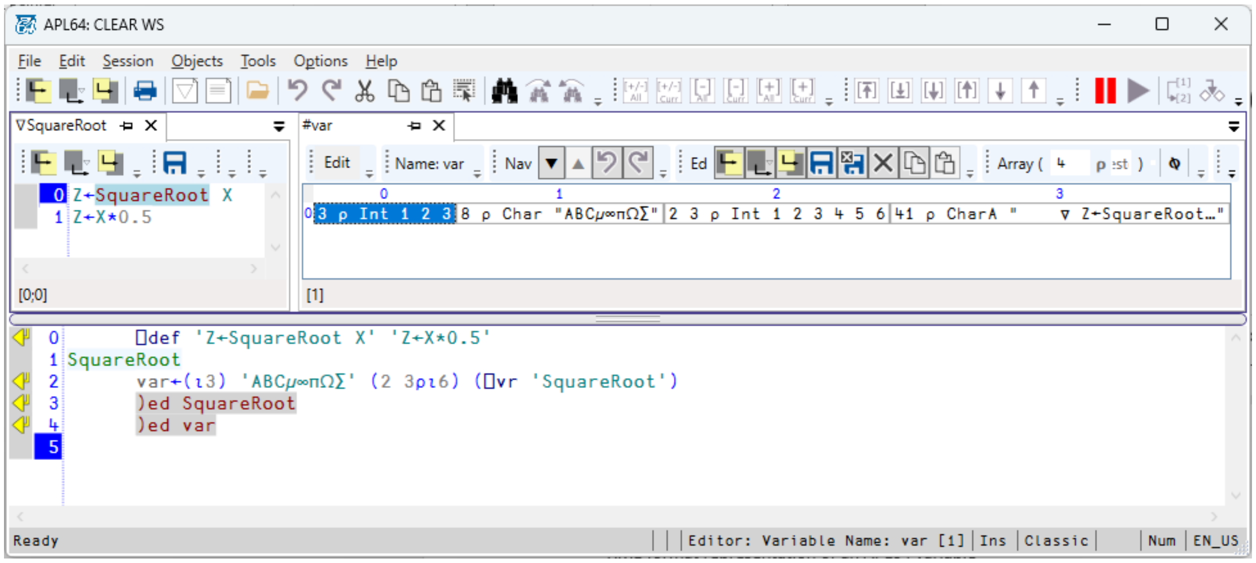
Since this example uses physical files, be sure to delete these files, if they exist, before following the example:

- c:\nfeFile.txt
- c:\zipNfeFile.zip
- c:\unZippedNfeFile.txt

Sample Unicode Data

In an APL64 developer version instance, create some sample Unicode data which contains text data with elements not in ⎕AV:

```
⎕def 'Z←SquareRoot X' 'Z←X*0.5'
var←(ι3) 'ABCμ∞πΩ∑' (2 3ρι6) (⎕vr 'SquareRoot')
```



```
Z←SquareRoot X
Z←X*0.5
```

```
       0                 1                       2                        3
0 3 ρ Int 1 2 3 8 ρ Char "ABCμ∞πΩ∑" 2 3 ρ Int 1 2 3 4 5 6 41 ρ CharA "     ∇ Z←SquareRoot…"
```

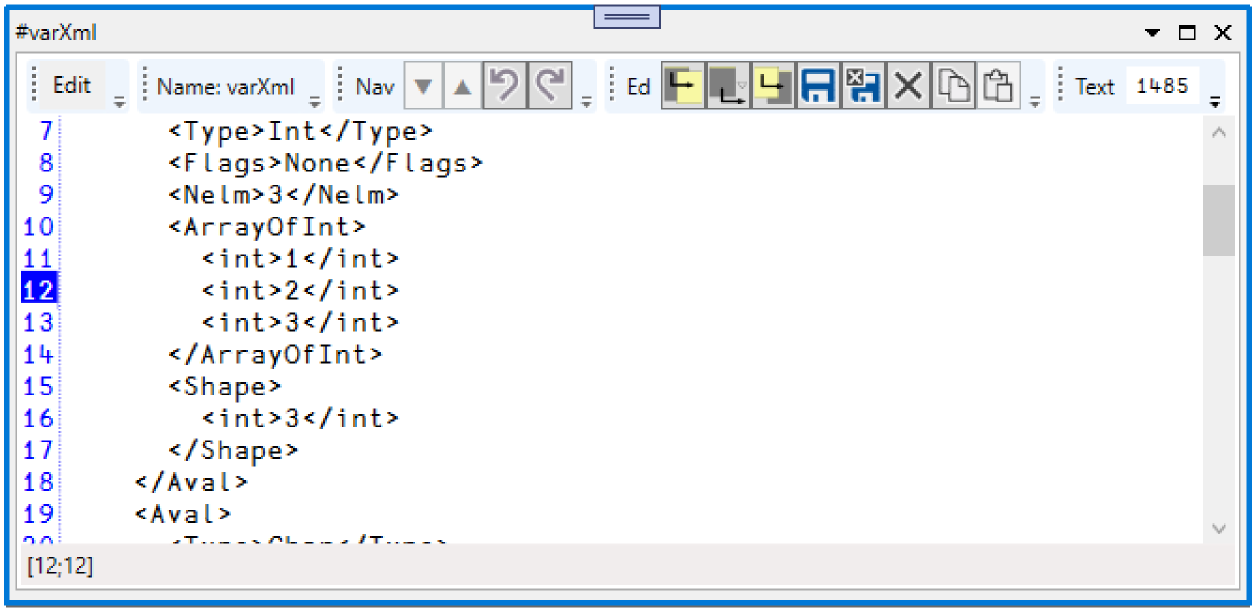
```
        ⎕def 'Z←SquareRoot X' 'Z←X*0.5'
SquareRoot
        var←(ι3) 'ABCμ∞πΩ∑' (2 3ρι6) (⎕vr 'SquareRoot')
        )ed SquareRoot
        )ed var
```

```
Ready        Editor: Variable Name: var [1] | Ins | Classic | Num | EN_US
```

XML or Json Representation of An APL64 Variable

The APL64 ⎕dr system function supports left arguments: 'xml', 'unxml', 'json' and 'unjson'. Both representations are a APL64 character vector. The json representation of an APL64 variable is slightly more compact. In this example the xml-format representation of an APL64 variable is used because it is more human-readable.

```
varXml←'xml' ⎕dr var
```



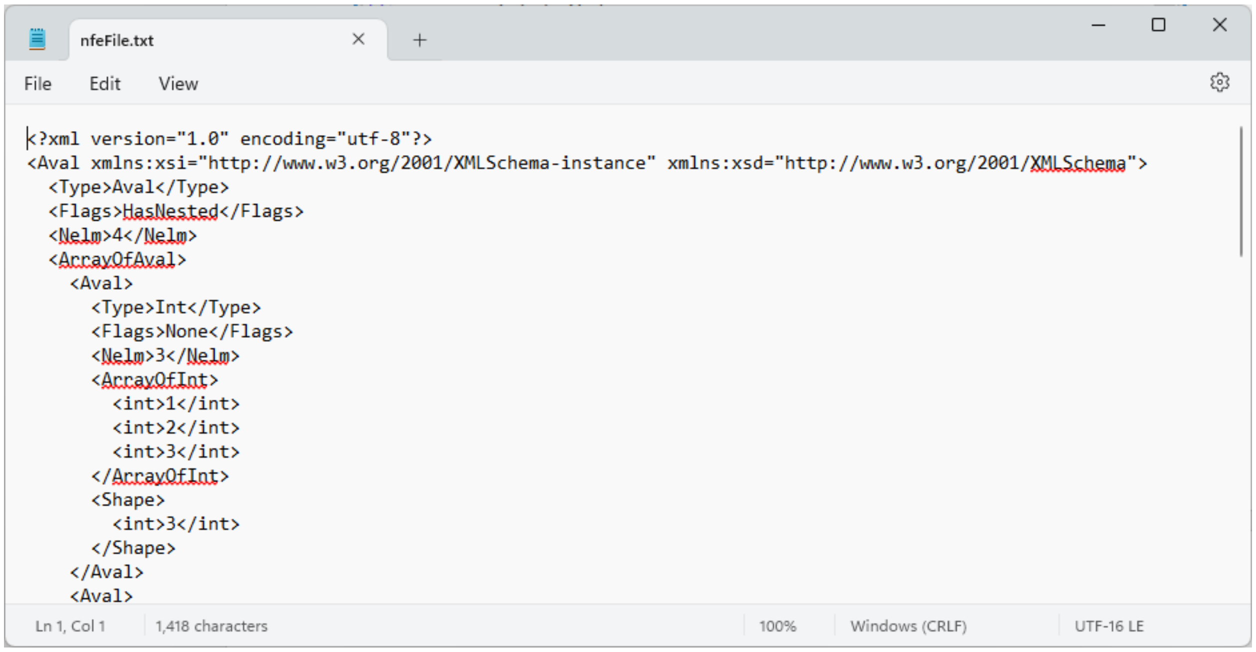
```
      <Type>Int</Type>
      <Flags>None</Flags>
      <Nelm>3</Nelm>
      <ArrayOfInt>
        <int>1</int>
        <int>2</int>
        <int>3</int>
      </ArrayOfInt>
      <Shape>
        <int>3</int>
      </Shape>
    </Aval>
    <Aval>
```

Create Unicode Text File

Use the APL64 ⎕nfe system function to specify the desired encoding, create the file and append the XML-format representation of the APL64 variable:

```
⎕nfe 'Encoding' 'Unicode'
filePath←'c:\nfeFile.txt'
filePath⎕nfe 'Create' 'ReadWrite' 'ReadWrite'
varXml ⎕nfe 'Append' filePath
```

Use Notepad to view the resulting file:



```
<?xml version="1.0" encoding="utf-8"?>
<Aval xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Type>Aval</Type>
  <Flags>HasNested</Flags>
  <Nelm>4</Nelm>
  <ArrayOfAval>
    <Aval>
      <Type>Int</Type>
      <Flags>None</Flags>
      <Nelm>3</Nelm>
      <ArrayOfInt>
        <int>1</int>
        <int>2</int>
        <int>3</int>
      </ArrayOfInt>
      <Shape>
        <int>3</int>
      </Shape>
    </Aval>
    <Aval>
```

Zip-compress the Unicode File

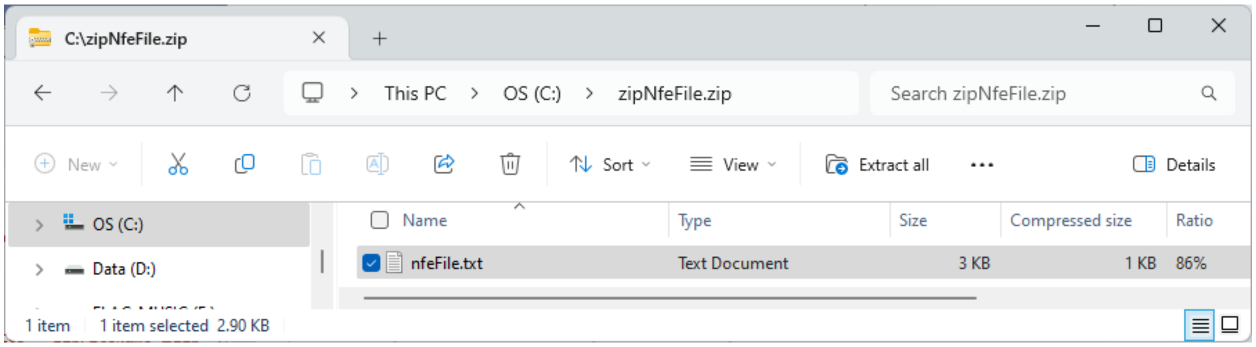
Use the APL64 ⎕Zip system function to create a zip-format file and insert the Unicode file into the zip-format file at a specified zip-path within the zip-format file.

In this example, the zip-path of the inserted Unicode file puts it in the topmost path within the zip-format file. The APL64 ⎕Path system function 'GetFileName' method is used to obtain the filename from the filePath variable.

More complex zip-paths may be specified. Multiple source files may be inserted into the same zip-format file.

```
zipFilePath←'c:\zipNfeFile.zip' ⍝Specify the zip-format file name
zipPath←'GetFileName' ⎕Path filePath ⍝Specify the zip-path within the zip-format file
zipFilePath ⎕Zip 'InsertFile' zipPath filePath 'SmallestSize' ⍝Insert Unicode file into the zip-format file
```

Use Windows Explorer to view the content of the zip-format file:



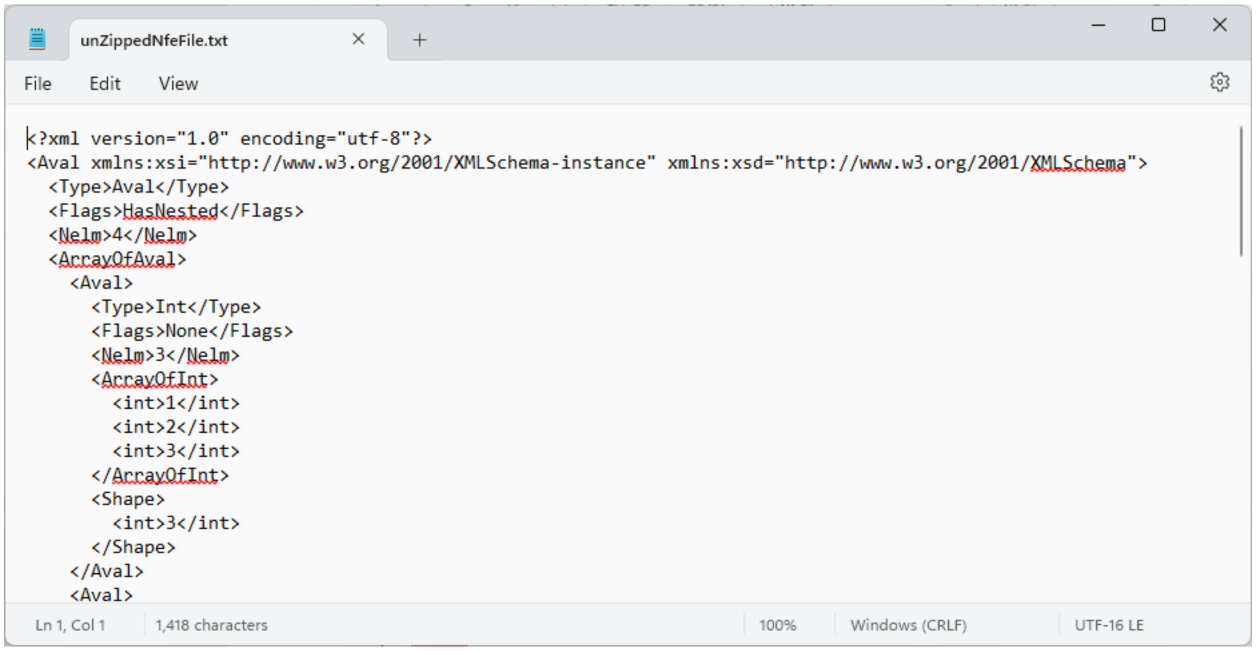
Name	Type	Size	Compressed size	Ratio
nfeFile.txt	Text Document	3 KB	1 KB	86%

Extract the Unicode File from the Zip-format File

The APL64 ⎕Zip system function 'GetAllFilePaths' action is used to obtain the zip-paths of all the files in the zip-format file. In this case there is only the Unicode file in the zip-format file. In this example, the zip-format file is extracted to a different Unicode file at the unZippedFilePath so its content can be compared with that of the original Unicode file.

```
zipFilePath←'c:\zipNfeFile.zip'
zipPath← 1⊃ zipFilePath ⎕Zip 'GetAllFilePaths'
unZippedFilePath←'c:\unZippedNfeFile.txt'
zipFilePath ⎕Zip 'ExtractFile' zipPath unZippedFilePath 1
```

Use Notepad to view the content of the unZippedFilePath file, unZippedNfeFile.txt:



```xml
<?xml version="1.0" encoding="utf-8"?>
<Aval xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Type>Aval</Type>
  <Flags>HasNested</Flags>
  <Nelm>4</Nelm>
  <ArrayOfAval>
    <Aval>
      <Type>Int</Type>
      <Flags>None</Flags>
      <Nelm>3</Nelm>
      <ArrayOfInt>
        <int>1</int>
        <int>2</int>
        <int>3</int>
      </ArrayOfInt>
      <Shape>
        <int>3</int>
      </Shape>
    </Aval>
    <Aval>
```

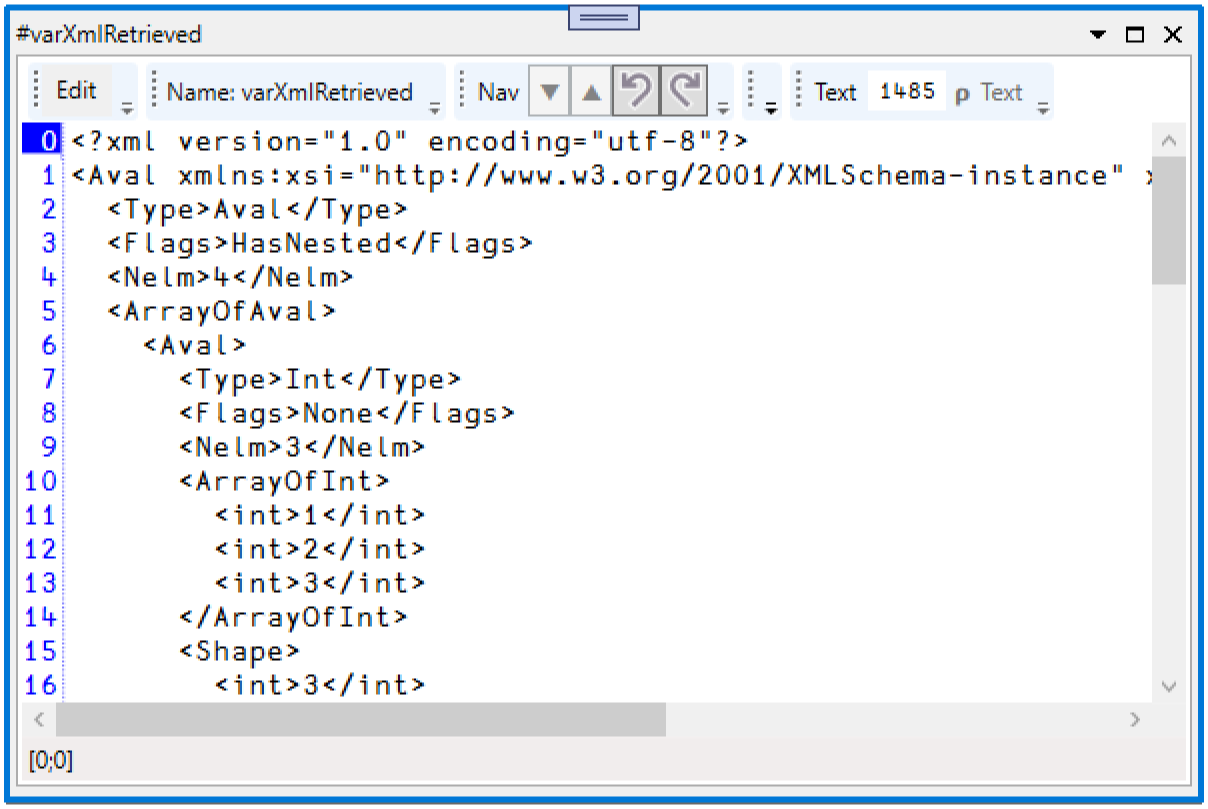
```
Ln 1, Col 1    1,418 characters    100%    Windows (CRLF)    UTF-16 LE
```

Read the Content of the Un-zipped Unicode File

The APL64 ⎕NFE system function is used to specify the encoding and read the content of the Unicode file:

```
⎕nfe 'Encoding' 'Unicode'
unZippedFilePath←'c:\unZippedNfeFile.txt'
varXmlRetrieved←⎕nfe 'ReadAllText' unZippedFilePath
```

Use an APL64 variable editor to display the APL64 varXmlRetrieved data from the Unicode file to observe its XML-format content:



```
<?xml version="1.0" encoding="utf-8"?>
<Aval xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" x
  <Type>Aval</Type>
  <Flags>HasNested</Flags>
  <Nelm>4</Nelm>
  <ArrayOfAval>
    <Aval>
      <Type>Int</Type>
      <Flags>None</Flags>
      <Nelm>3</Nelm>
      <ArrayOfInt>
        <int>1</int>
        <int>2</int>
        <int>3</int>
      </ArrayOfInt>
      <Shape>
        <int>3</int>
```

Materialize the XML-format Unicode Data as an APL64 Variable

The APL64 DR 'unXml' action is used to create an APL64 variable, 'varRetrieved' and compare it with the original variable 'var':

```
varRetrieved←'unxml' ⎕dr varXmlRetrieved
```

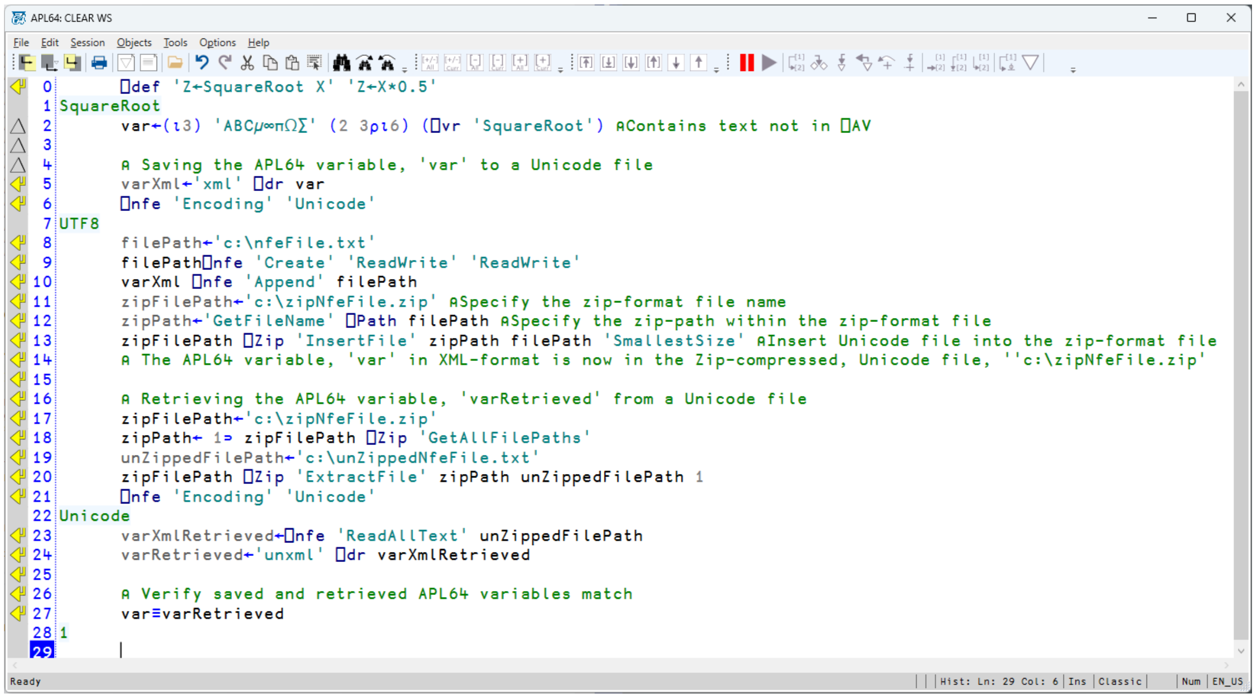
Check that the Retrieved Variable Matches the Original Variable

```
⎕def 'Z←SquareRoot X' 'Z←X*0.5'
var←(ι3) 'ABCμ∞πΩ∑' (2 3ρι6) (⎕vr 'SquareRoot') ⍝Contains text not in ⎕AV

⍝ Saving the APL64 variable, 'var' to a Unicode file and ZIP-compressing it
varXml←'xml' ⎕dr var
⎕nfe 'Encoding' 'Unicode'
filePath←'c:\nfeFile.txt'
filePath⎕nfe 'Create' 'ReadWrite' 'ReadWrite'
varXml ⎕nfe 'Append' filePath
zipFilePath←'c:\zipNfeFile.zip' ⍝Specify the zip-format file name
zipPath←'GetFileName' ⎕Path filePath ⍝Specify the zip-path within the zip-format file
zipFilePath ⎕Zip 'InsertFile' zipPath filePath 'SmallestSize' ⍝Insert Unicode file into the zip-format
file ⍝ The APL64 variable, 'var' in XML-format is now in the Zip-compressed, Unicode file,
''c:\zipNfeFile.zip'
```

```
⍝ Retrieving the APL64 variable, 'varRetrieved' from a Unicode file
zipFilePath←'c:\zipNfeFile.zip'
zipPath← 1⊃ zipFilePath ⎕Zip 'GetAllFilePaths'
unZippedFilePath←'c:\unZippedNfeFile.txt'
zipFilePath ⎕Zip 'ExtractFile' zipPath unZippedFilePath 1
⎕nfe 'Encoding' 'Unicode'
varXmlRetrieved←⎕nfe 'ReadAllText' unZippedFilePath
varRetrieved←'unxml' ⎕dr varXmlRetrieved

⍝ Verify saved and retrieved APL64 variables match
var≡varRetrieved
```


APL64: CLEAR WS

```
      ⎕def 'Z←SquareRoot X' 'Z←X*0.5'
SquareRoot
      var←(⍳3) 'ABCμ∞πΩ∑' (2 3⍴⍳6) (⎕vr 'SquareRoot') ⍝Contains text not in ⎕AV

      ⍝ Saving the APL64 variable, 'var' to a Unicode file
      varXml←'xml' ⎕dr var
      ⎕nfe 'Encoding' 'Unicode'
UTF8
      filePath←'c:\nfeFile.txt'
      filePath⎕nfe 'Create' 'ReadWrite' 'ReadWrite'
      varXml ⎕nfe 'Append' filePath
      zipFilePath←'c:\zipNfeFile.zip' ⍝Specify the zip-format file name
      zipPath←'GetFileName' ⎕Path filePath ⍝Specify the zip-path within the zip-format file
      zipFilePath ⎕Zip 'InsertFile' zipPath filePath 'SmallestSize' ⍝Insert Unicode file into the zip-format file
      ⍝ The APL64 variable, 'var' in XML-format is now in the Zip-compressed, Unicode file, ''c:\zipNfeFile.zip'


      ⍝ Retrieving the APL64 variable, 'varRetrieved' from a Unicode file
      zipFilePath←'c:\zipNfeFile.zip'
      zipPath← 1⊃ zipFilePath ⎕Zip 'GetAllFilePaths'
      unZippedFilePath←'c:\unZippedNfeFile.txt'
      zipFilePath ⎕Zip 'ExtractFile' zipPath unZippedFilePath 1
      ⎕nfe 'Encoding' 'Unicode'
Unicode
      varXmlRetrieved←⎕nfe 'ReadAllText' unZippedFilePath
      varRetrieved←'unxml' ⎕dr varXmlRetrieved

      ⍝ Verify saved and retrieved APL64 variables match
      var≡varRetrieved
1
```

Ready | Hist: Ln: 29 Col: 6 | Ins | Classic | Num | EN_US