Using ⎕XML

Table of Contents

Overview

XML can be used to Create, Add and Delete .Net XML objects.

Syntax: result ← [LeftArg] ⎕XML Action [Arg1]… [ArgN]

⎕XML exposes the features of Microsoft Linq for XML: Overview - LINQ to XML | Microsoft Docs

The action argument of ⎕XML is a case-insensitive character scalar, character vector or string. The selected action will determine the additional arguments and the result of an APL64 executable statement containing ⎕XML.

What is XML?

XML is a hierarchical text representation of data using specified glyphs to identify a data element by name and represent its value as text. Learn more about XML here.

Because XML-format data is text-based, easy for humans to read and its structure is considered self-descriptive, it is often used to transmit information between two entities such as web clients and web servers, possibly using different operating systems.

XML XObjects in .Net

Each type of XML XObject is an instance of the .Net class defining the XObject. XObject types include XDocument, XElement, XAttribute, XNamespace, XComment, XCData, XmlDeclaration, etc. Some properties of an XML XObject have representations in APL64, such as its text name and text value, but other properties of an XML XObject have no direct representation in APL64.

Working with XObjects using ⎕XML

To facilitate the manipulation of XML XObjects in APL64, the APL64 programmer can assign a unique, case-sensitive text name for the XML XObject. This text name can be assigned to an APL64 variable. The text name of an XML XObject selected by the APL64 programmer can be different than the XObject's name in its .Net class instance.

Using the programmer-assigned name, an XObject can be manipulated during an APL64 instance even though the XObject cannot be directly represented in APL64.

Selecting and obtaining a subordinate XObject from a parent XObject requires *a priori* knowledge of the structure of the XObjects composing the parent. A specific subordinate XObject in a parent XObject may be obtained by, possibly multiple, use of the ⎕XML actions such as AttGet, EltGet or XPathSelectElt.

⎕XML Representation of XML Objects in APL64

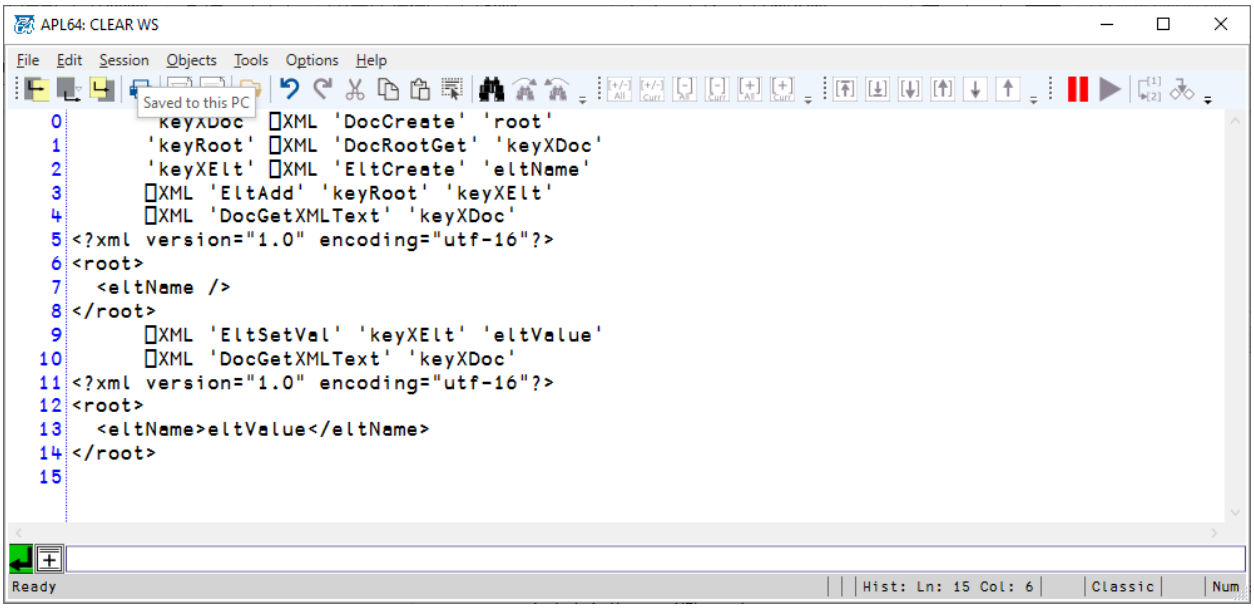
In an instance of APL64, a .Net dictionary is used to contain ⎕XML-based XObjects. Elements of the ⎕XML dictionary are <key, value> pairs. The XObject's key is an APL-programmer-selected text name for the XML XObject which is generally a required ⎕XML argument. The XObject's value is the XObject's .Net class instance including all of its properties and methods.

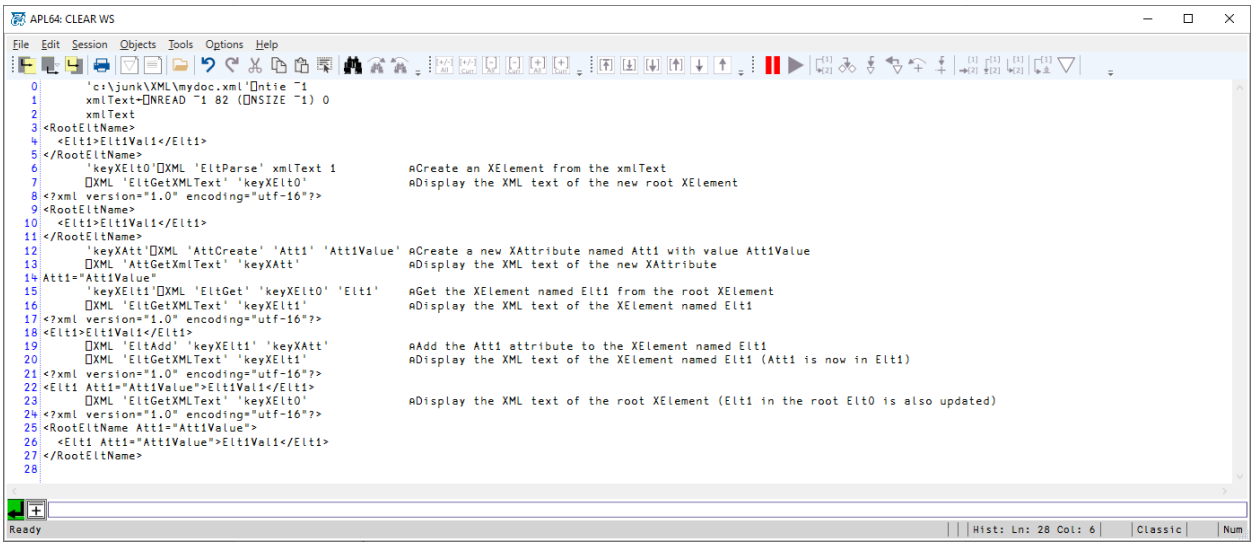
The XML XObjects dictionary is persisted during an APL64 instance but is not persisted between APL64 instances.

Most ⎕XML actions apply to XObjects in the ⎕XML dictionary. ⎕XML provides some actions which operate directly on the ⎕XML dictionary.

⎕XML Uses .Net Assignment by Reference

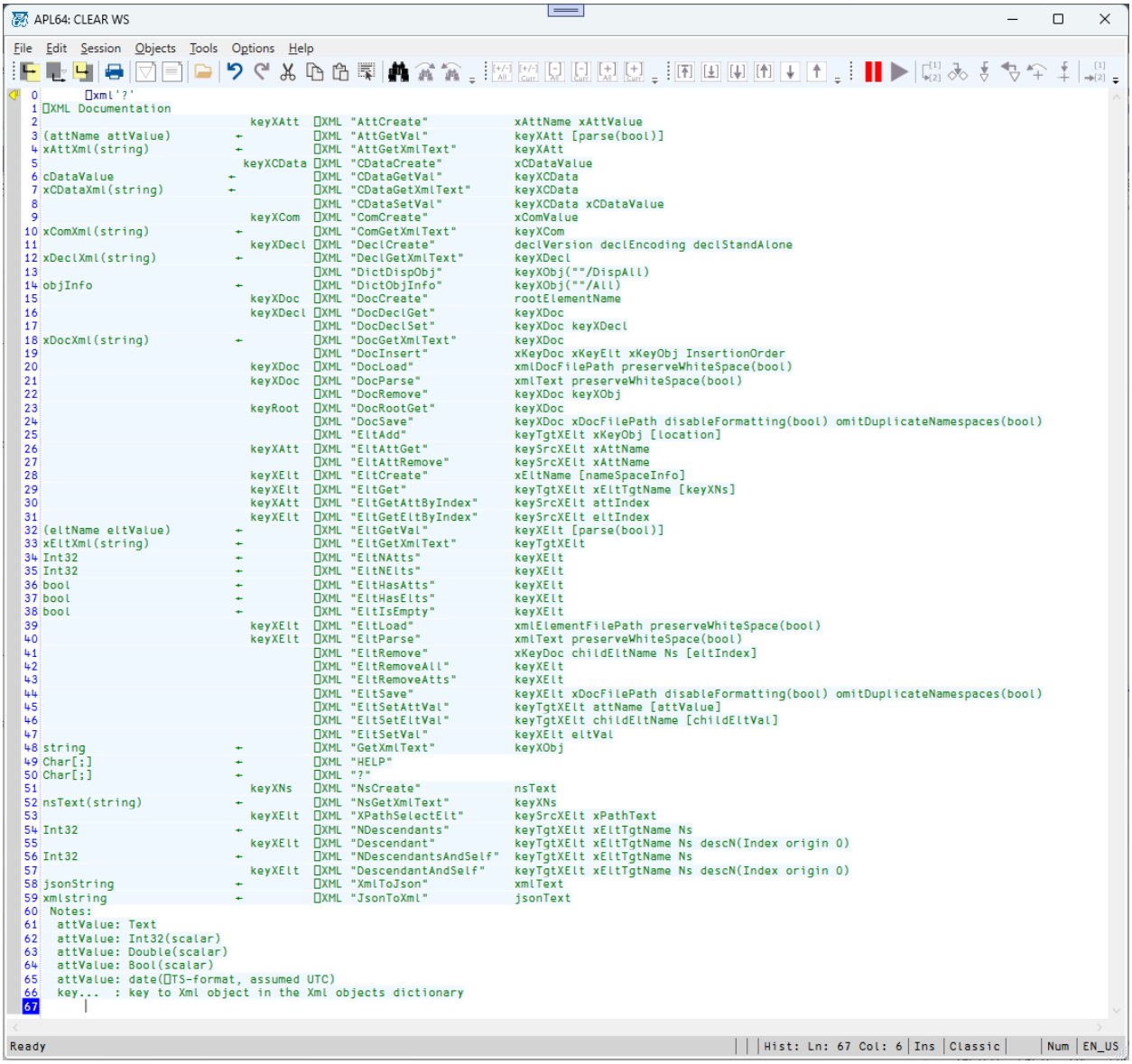
When ⎕XML is used to obtain an XObject (XElement, XAttribute, etc.) from a parent XObject and ⎕XML is used to modify the child XObject, the parent object is also modified. This behavior is different than classic APL variable assignment by value.





Summary of ⎕XML Actions

To obtain the summary, execute ⎕XML "?" or ⎕XML "Help",



⎕XML Actions

AttCreate

Syntax: keyXAtt ⎕XML "AttCreate" xAttName xAttValue

Arguments & Result:

Action Synonym: CreateATT

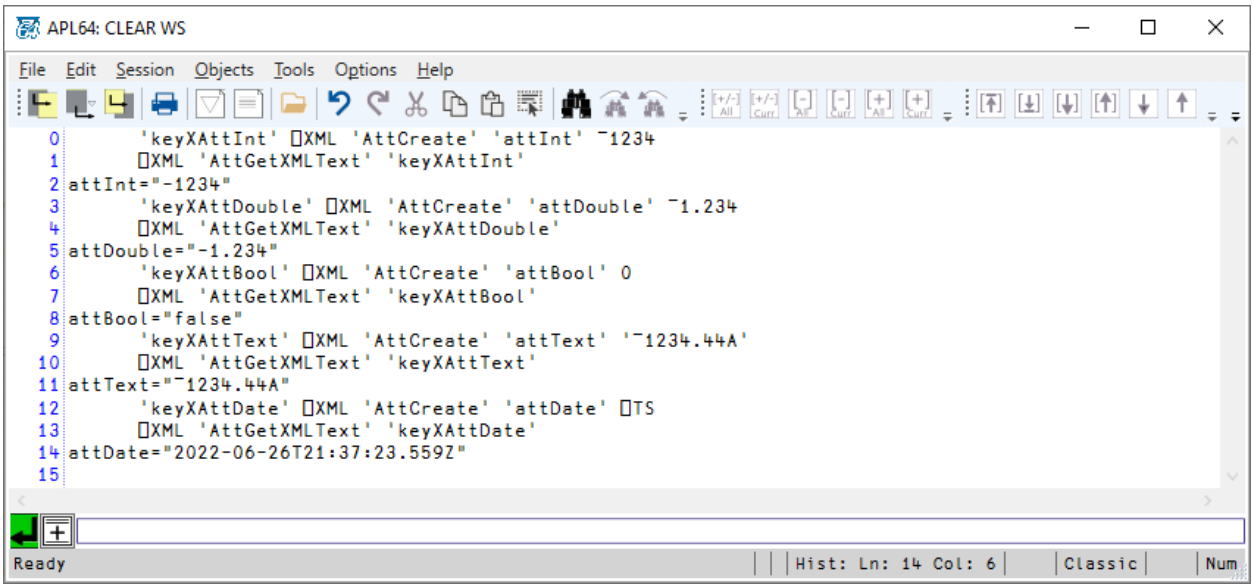
keyXElt is the APL programmer-selected key in the XML XObjects dictionary to be associated with the XAttribute created by the AttCreate action.

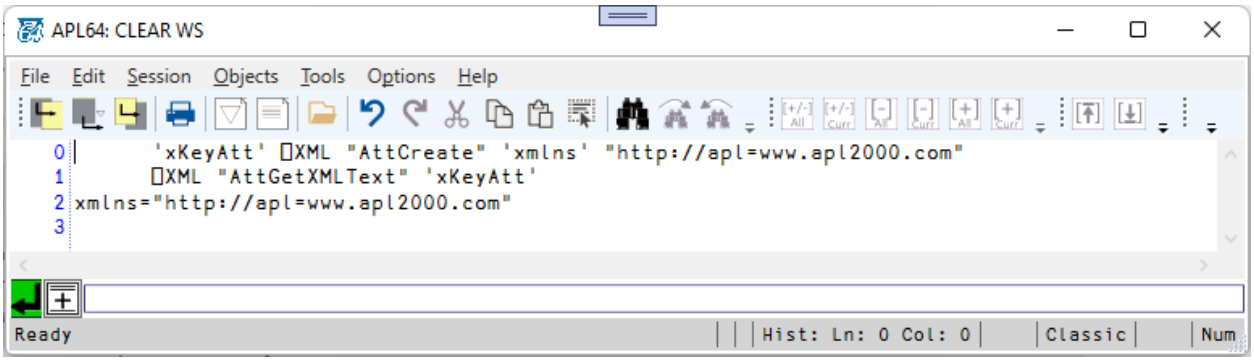
xAttName is the APL programmer-selected attribute name which will become part of the new XAttribute.

xAttValue is the APL programmer-selected value which will become the XAttribute value.

The XAttribute value will be set according to the APL data type of xAttValue: APL64 scalar integer, scalar double, scalar Boolean, text (string) or an integer vector date (in ⎕TS-format and assumed to be UTC time zone).

If the APL data type of xAttValue is not to be considered, provide xAttValue as a scalar string or character or as a character vector.





GetAttVal

Syntax: (attName attValue(parsed)) ←⎕XML "GetAttVal" keyXAtt [parse(bool)]

Arguments & Result:

Action Synonyms: AttValGet, AttGetVal

The AttGetVal action will parse the value of the specified XAttribute into APL64 values, e.g. scalar Boolean, Integer, Double, String or Date (in ⎕TS-format assuming UTC).

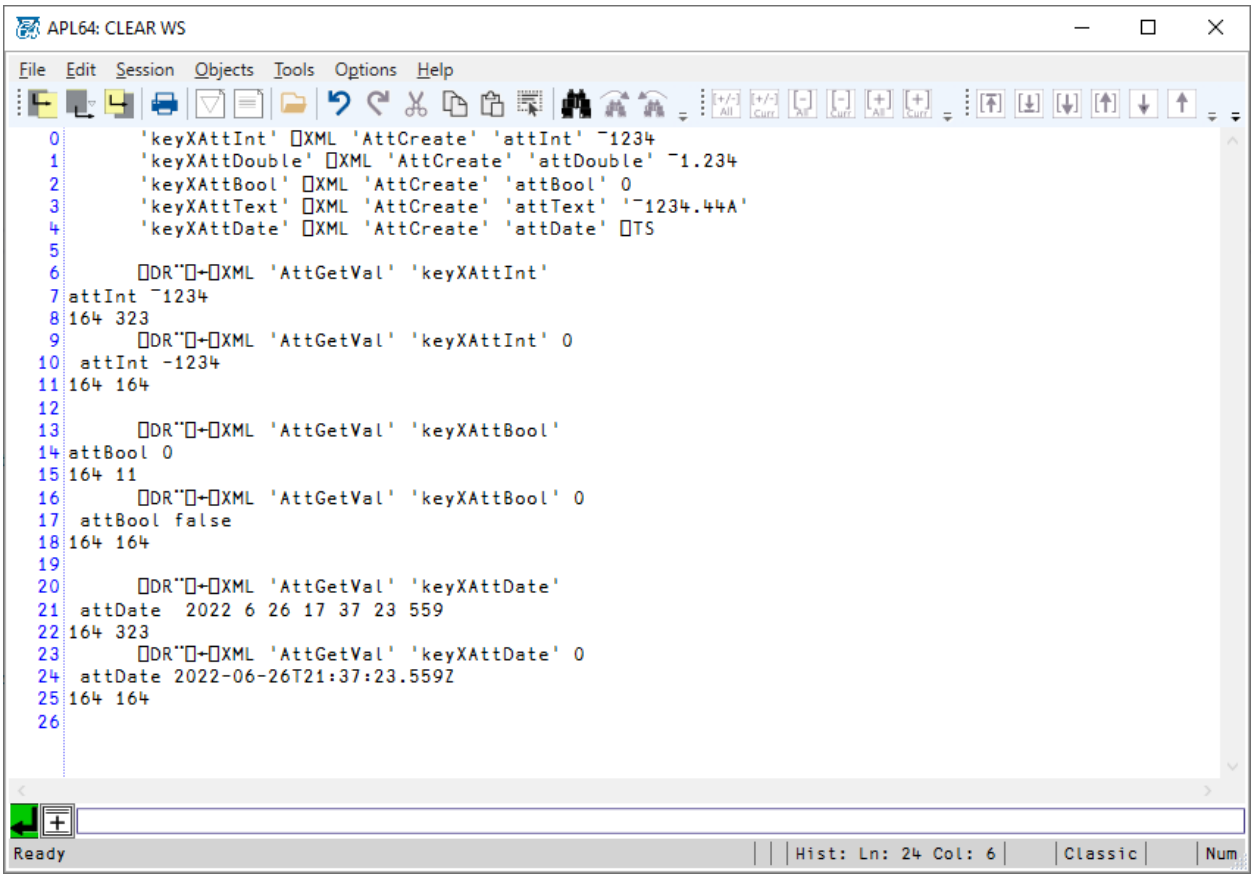
keyXAtt is the APL programmer-provided key in the ⎕XML XObject Dictionary associated with the existing XAttribute.

attName(string) is the XAttribute name property value.

parse is an optional Boolean scalar with default value 1/True.

- If parse is true, the XAttribute value will be parsed into the applicable APL values: APL64 scalar integer, double, Boolean, string or a integer vector date (in ⎕TS-format and assumed to be UTC time zone).
- If parse is false, the attValue returned will be an APL scalar string.

attValue is the value of the specified XAttribute.



GetAttXmlText

Syntax: xAttXml(string)←⎕XML "GetAttXmlText" keyXAtt

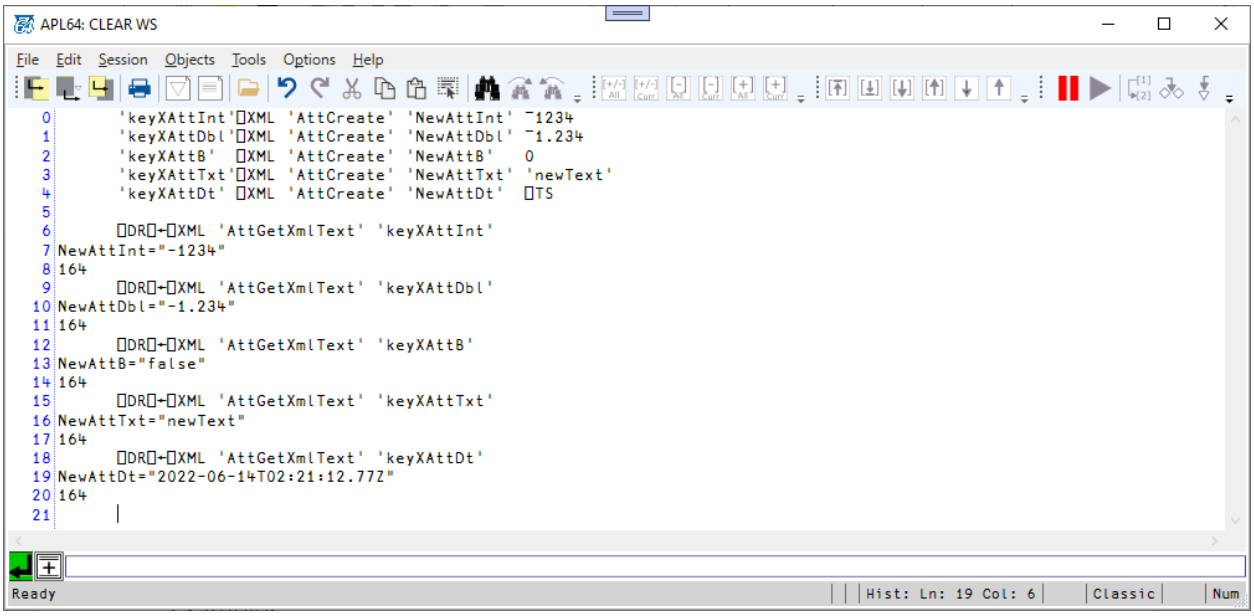
Arguments & Result:

Action Synonyms: AttXmlTextGet, AttGetXmlText

keyXAtt is the APL programmer-provided key in the ⎕XML XObject Dictionary associated with the existing XAttribute.

xAttXml is the text string representation of the existing XAttribute associated with keyXAtt.

If the specified XAttribute is null, an empty string will be returned.



CDataCreate

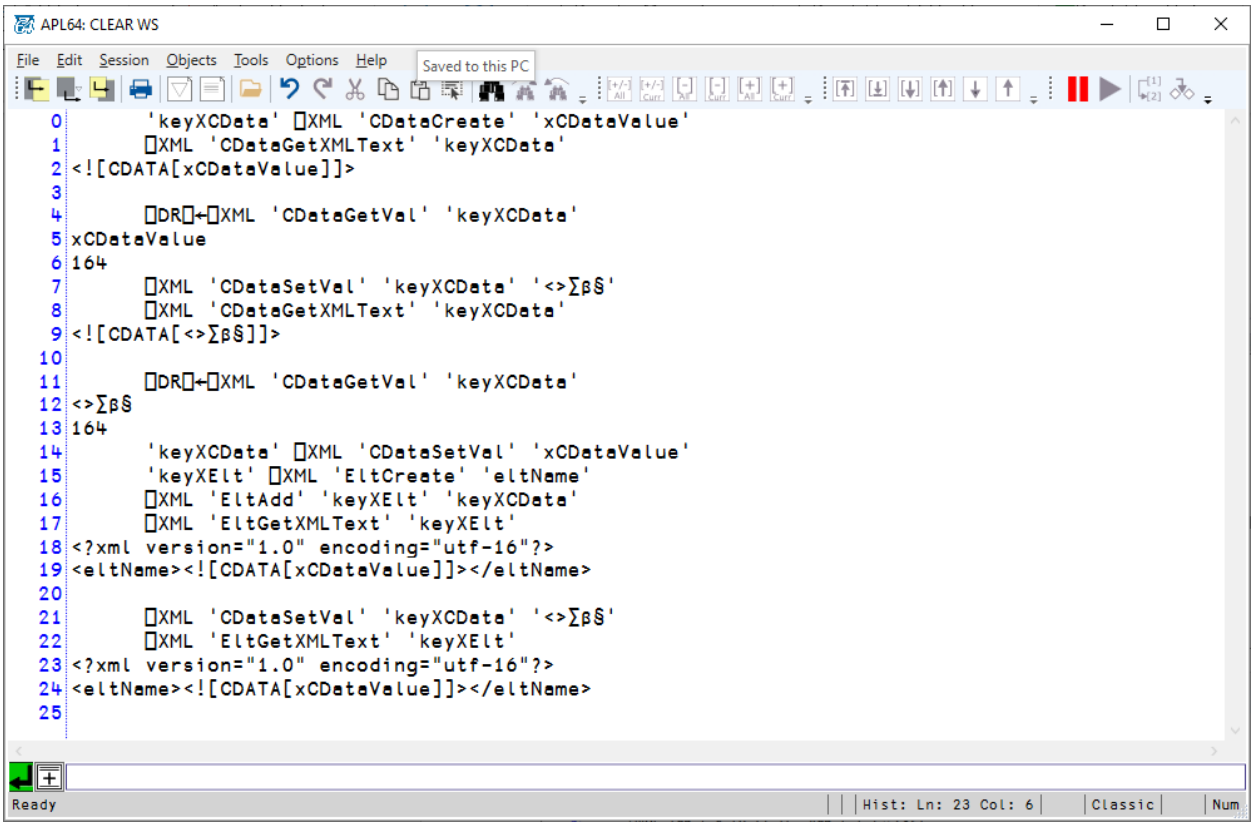
Syntax: keyXCData ⎕XML "CDataCreate" xCDataValue

Arguments & Result:

Action Synonym: CreateCData

keyXCData is the APL programmer-selected key in the XML XObjects dictionary to be associated with the XCData created by the CDatareate action.

xCDataValue is the APL programmer-selected text value which will become the XCData value.



CDataGetVal

Syntax: cDataValue ←⎕XML "CDataGetVal" keyXCData

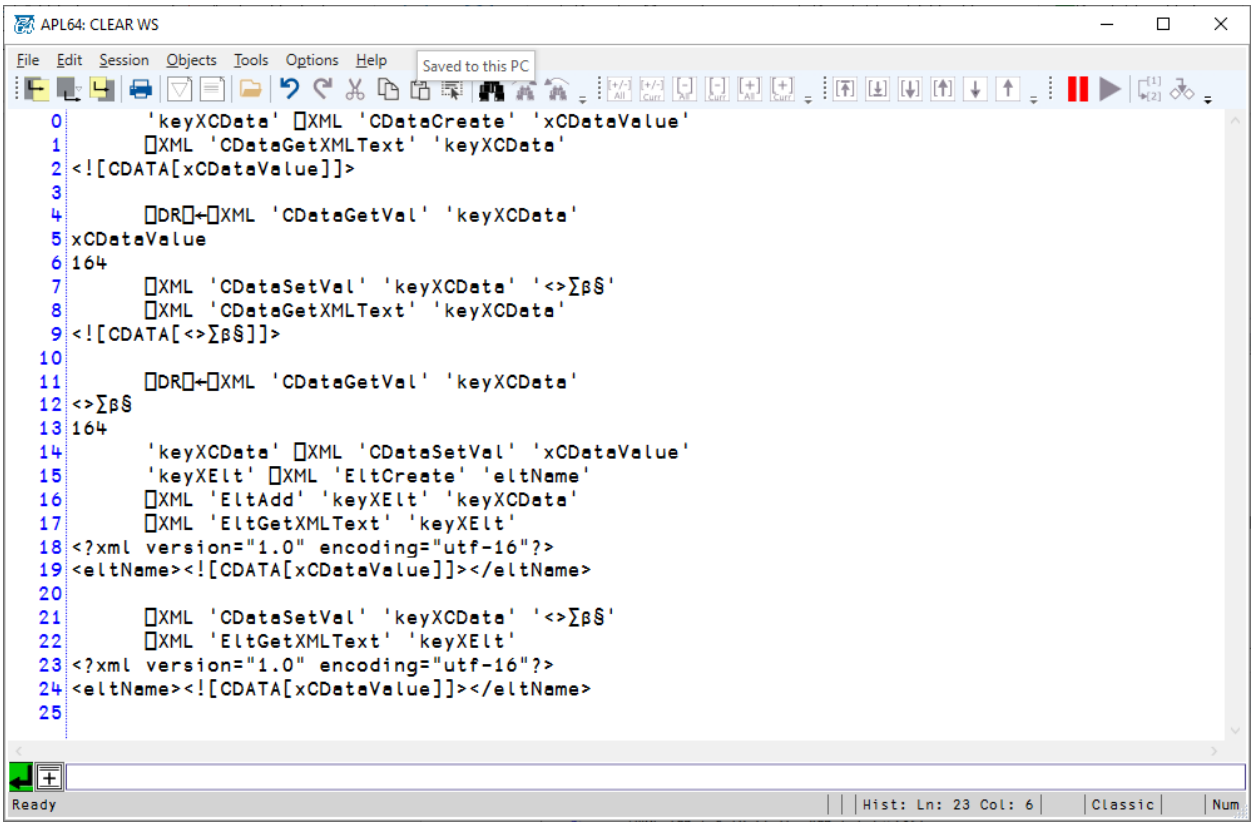
Arguments & Result:

Action Synonym: CDataValGet

The CDataGetVal action will return the value of the specified XXData into an APL64 scalar string.

keyXCData is the APL programmer-provided key in the ⎕XML XObject Dictionary associated with the existing XCData.

cDataValue is the value of the specified XCData.



CDataSetVal

Syntax:

⎕XML "CDataSetVal" keyXCData xCDataValue

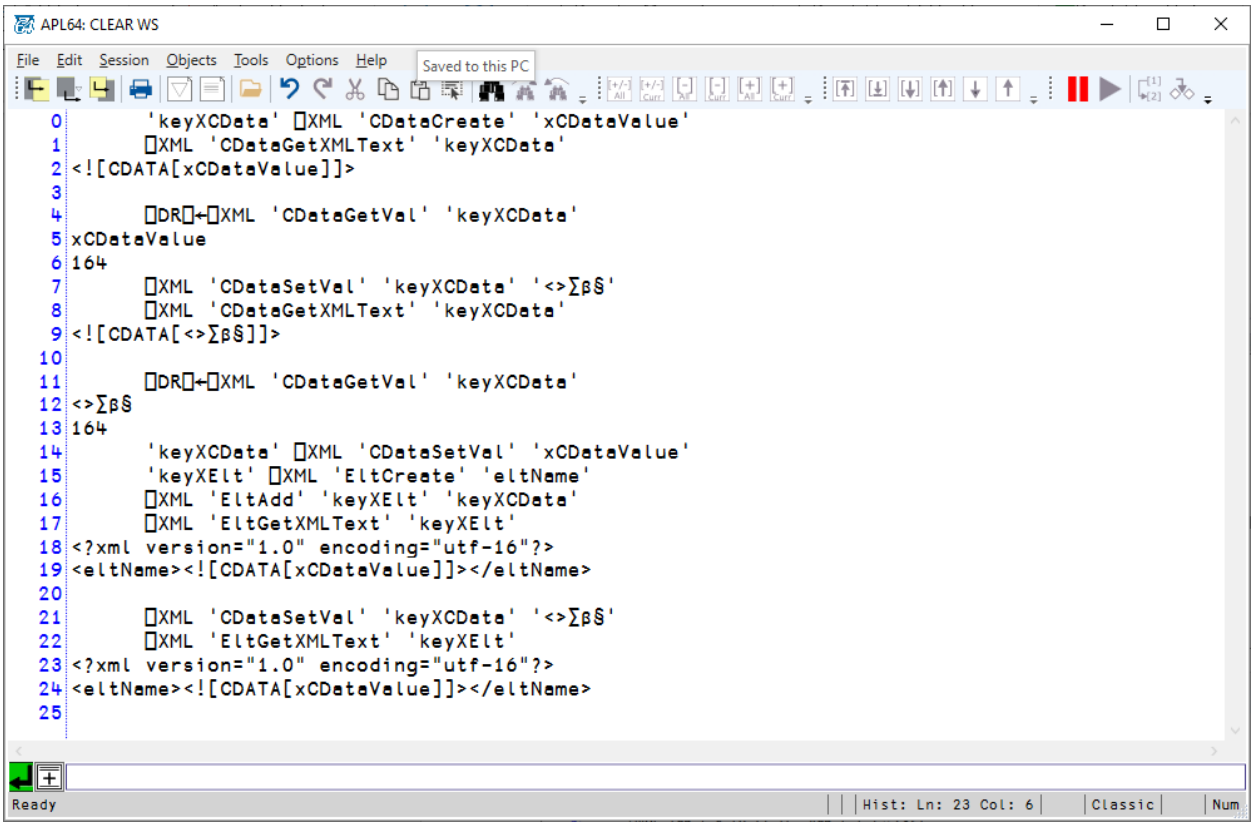
Arguments & Result:

Action Synonyma: CDataValSet, SetCDataVal

keyXCData is the APL programmer-provided key in the ⎕XML XObject Dictionary associated with the existing XCData.

xCDataValue is the new value of the specified XCData.

Assignment by reference does not apply to XCData objects .



ComCreate

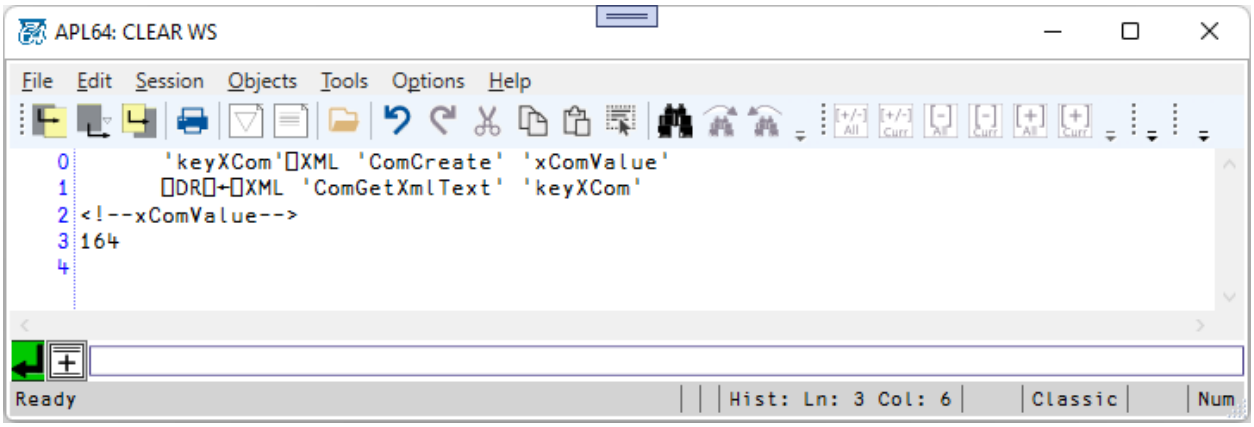
Syntax: keyXCom ⎕XML "ComCreate" xComValue

Arguments & Result:

Action Synonym: CreateCom

keyXCom is the APL programmer-selected key in the XML XObjects dictionary to be associated with the XComment created by the ComCreate action.

xComValue is the APL programmer-selected text value which will become the XComment value.



DeclCreate

Syntax: keyXDecl ⎕XML "DeclCreate" declVersion declEncoding declStandAlone

Arguments & Result:

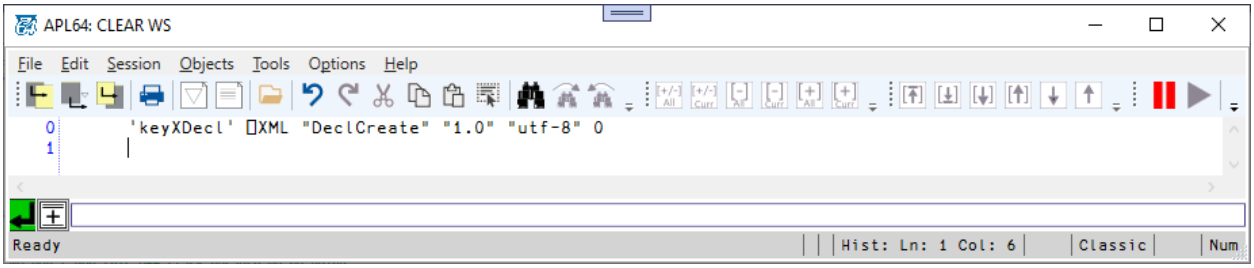
Action Synonym: CreateDecl

keyXDecl is the APL programmer-selected key in the XML XObjects dictionary to be associated with the XDeclaration created by the DeclCreate action.

declVersion is text specifying the XML-format version, e.g. “1.0”.

declEncoding is text specifying the encoding used, e.g. “utf-8”.

declStandAlone is an APL64 Boolean scalar indicating if the XDeclaration is ‘standalone’.



NDescendants

Syntax: Int32←⎕XML "NDescendants" keyTgtXElt xEltTgtName Ns

Arguments & Result:

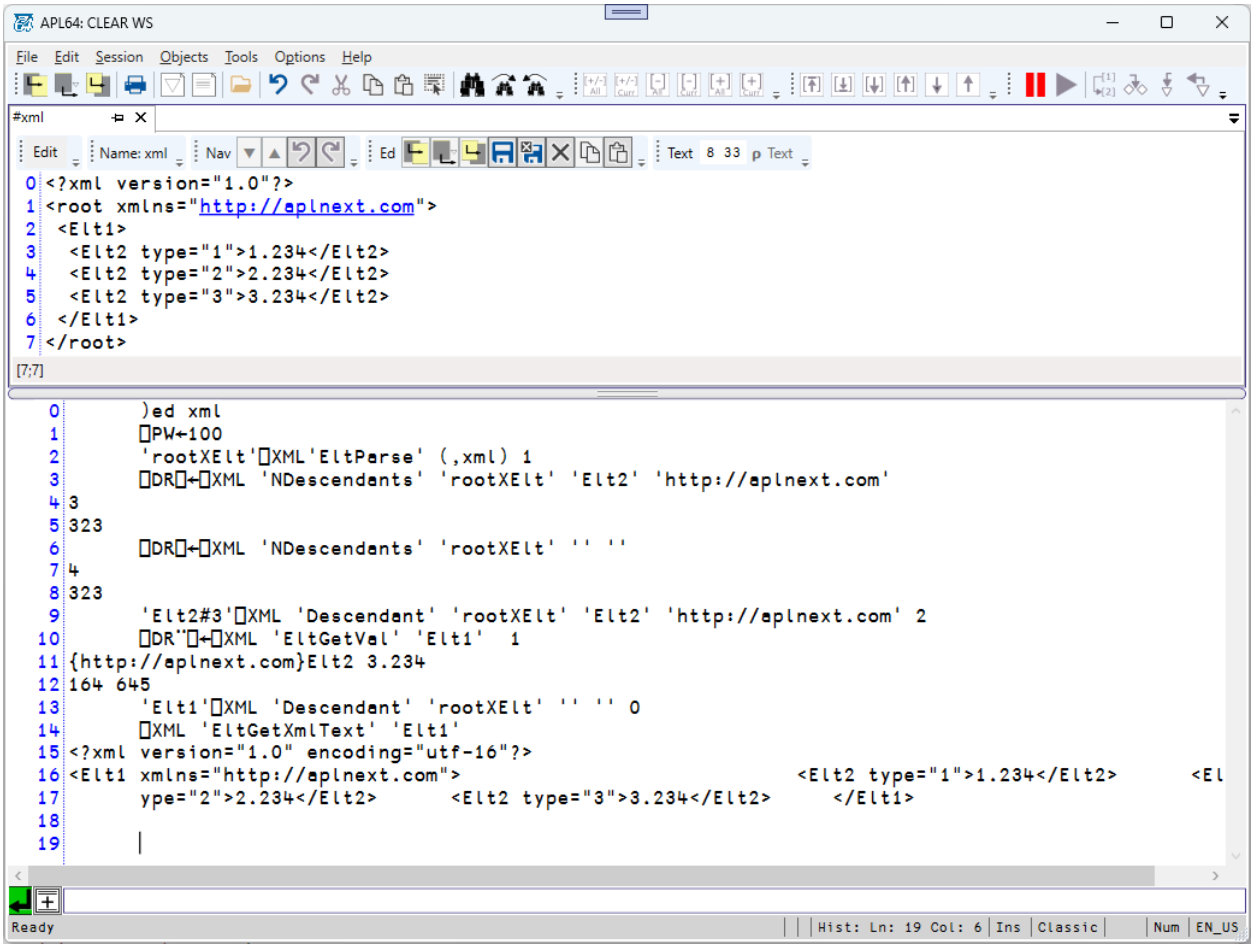
Action Synonym: NDesc

keyTgtXElt is the APL programmer- provided key in the ⎕XML XObject Dictionary associated with the XElement for which the number of specified XElement descendants is to be determined.

xEltTgtName is the APL programmer-provided text which is the name of specified XElement descendants. If xEltTgtName is "", all XElement descendants will be counted in the result.

Ns is the APL programmer-provided text for the XNamespace associated with the XElement descendants, or "" if none applies.

The result is an Int32 scalar which specifies the number of specified XElement descendants, or zero if none.



Descendant

Syntax: keyXElt ⎕XML "NDescendants" keyTgtXElt xEltTgtName Ns DescN(Index origin 0)

Arguments & Result:

Action Synonym: Desc

keyTgtXElt is the APL programmer- provided key in the ⎕XML XObject Dictionary associated with the XElement for which the selected XElement descendant is to be obtained.

xEltTgtName is the APL programmer-provided text which is the name of specified XElement descendants. If xEltTgtName is "", all XElement descendants will be considered in the collection from which the DescN element will be selected.

Ns is the APL programmer-provided text for the XNamespace associated with the XElement descendants, or "" if none applies.

DescN is the non-negative integer (Index origin 0) which indicates the XElement descendant to obtain.

keyXElt is the APL programmer-provided key in the ⎕XML XObject Dictionary which will be associated with the specified descendant XElement.

For an example of the Descentant acton, see the NDescendants action example.

NDescendantsAndSelf

Syntax: Int32←⎕XML "NDescendantsAndSelf" keyTgtXElt xEltTgtName Ns

Arguments & Result:

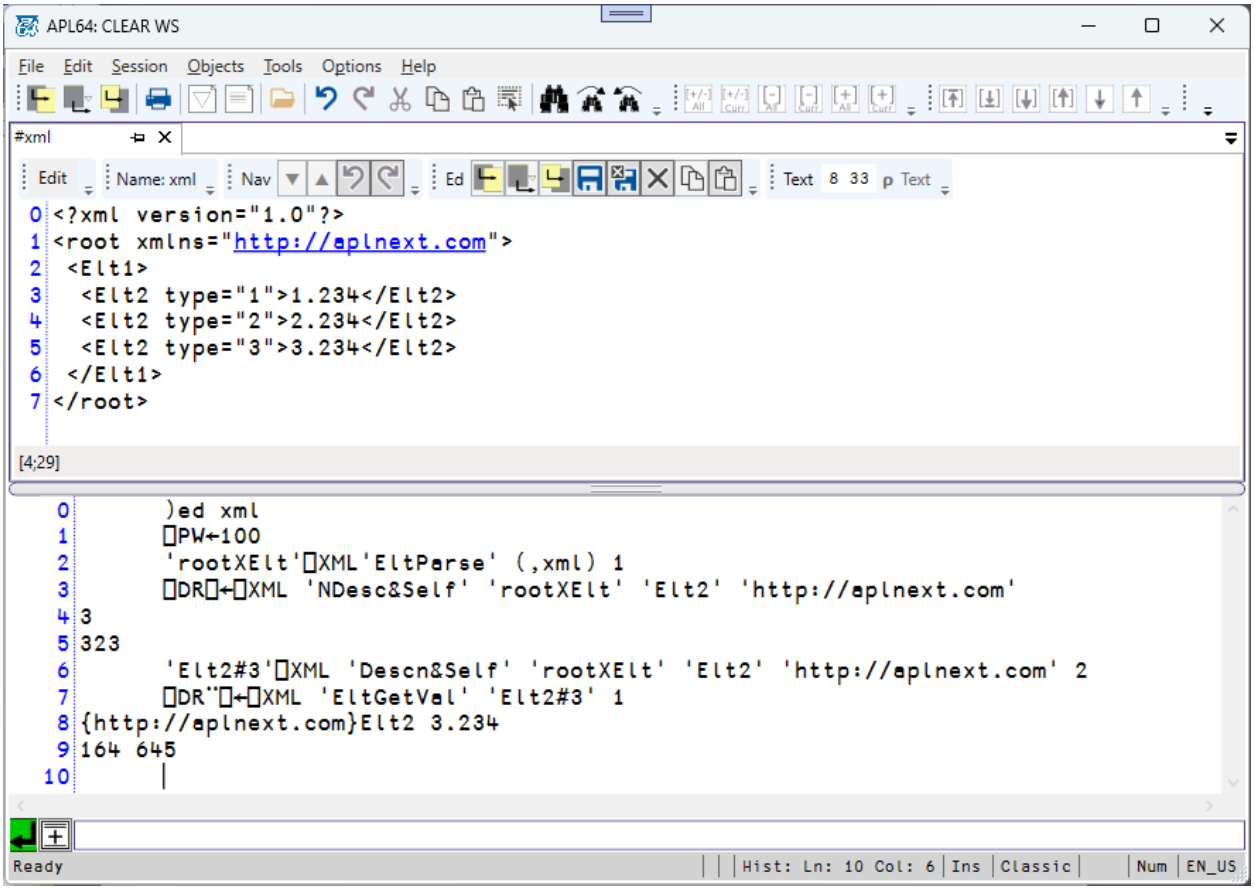
Action Synonym: NDesc&Self

keyTgtXElt is the APL programmer- provided key in the ⎕XML XObject Dictionary associated with the XElement for which the number of specified XElement descendants, including this XElement, is to be determined.

xEltTgtName is the APL programmer-provided text which is the name of specified XElement descendants. If xEltTgtName is "", all XElement descendants and the containing XElement will be counted in the result.

Ns is the APL programmer-provided text for the XNamespace associated with the XElement descendants, or "" if none applies.

The result is an Int32 scalar which specifies the number of specified XElement descendants including this XElement, or zero if none.



DescendantAndSelf

Syntax: keyXElt ⎕XML "NDescendants" keyTgtXElt xEltTgtName Ns DescN(Index origin 0)

Arguments & Result:

Action Synonym: Descn&Self

keyTgtXElt is the APL programmer- provided key in the ⎕XML XObject Dictionary associated with the XElement for which the selected XElement descendant or containing XElement is to be obtained.

xEltTgtName is the APL programmer-provided text which is the name of specified XElement descendants. If xEltTgtName is "", all XElement descendants and the containing XElement will be considered in the collection from which the DescN element will be selected.

Ns is the APL programmer-provided text for the XNamespace associated with the XElement descendants, or "" if none applies.

DescN is the non-negative integer (Index origin 0) which indicates the XElement descendant to obtain.

keyXElt is the APL programmer-provided key in the ⎕XML XObject Dictionary which will be associated with the specified descendant XElement.

For an example of the DescendantAndSelf action, see the NDescendantsAndSelf action example.

DictDispObj

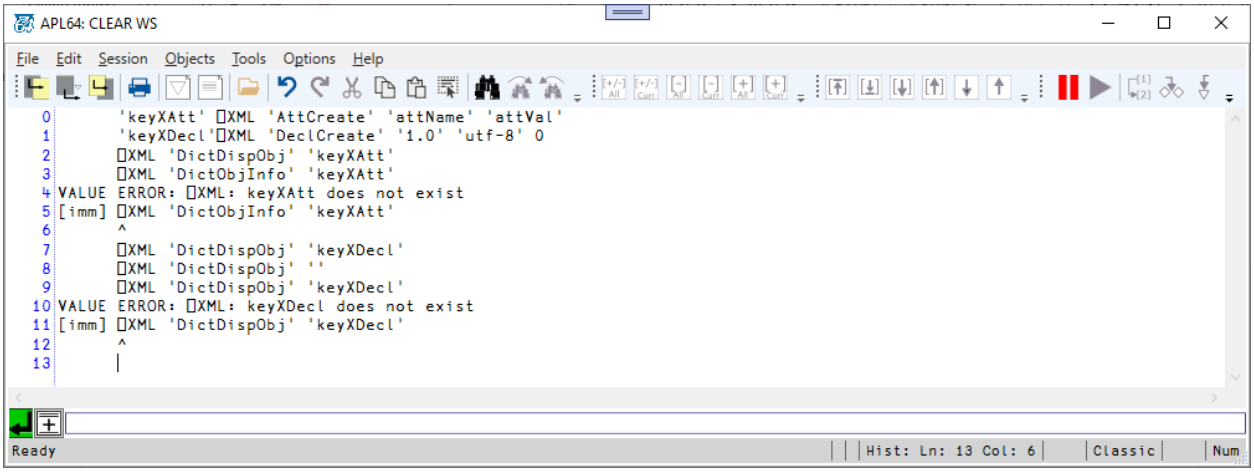
Syntax: ⎕XML "DictDispObj" keyXObj(""/Dispose All)"

Arguments & Result:

Action Synonym: DictObjDisp

keyXObj is the APL programmer-provided key in the ⎕XML XObject Dictionary associated with the XObject to be disposed. If keyXObj is "", all existing XML objects in the ⎕XML XObject Dictionarywill be disposed. When an XObject is disposed, the memory associated with that object is released and the dictionary <key, value> pair associated with the XObject is eliminated.

No exception will occur if the specified keyXObj is not contained in the ⎕XML XObject Dictionarykeys.



DictObjInfo

Syntax: xObjInfo ⎕XML "DictObjInfo" keyXObj(""/All Objects)

Arguments & Result:

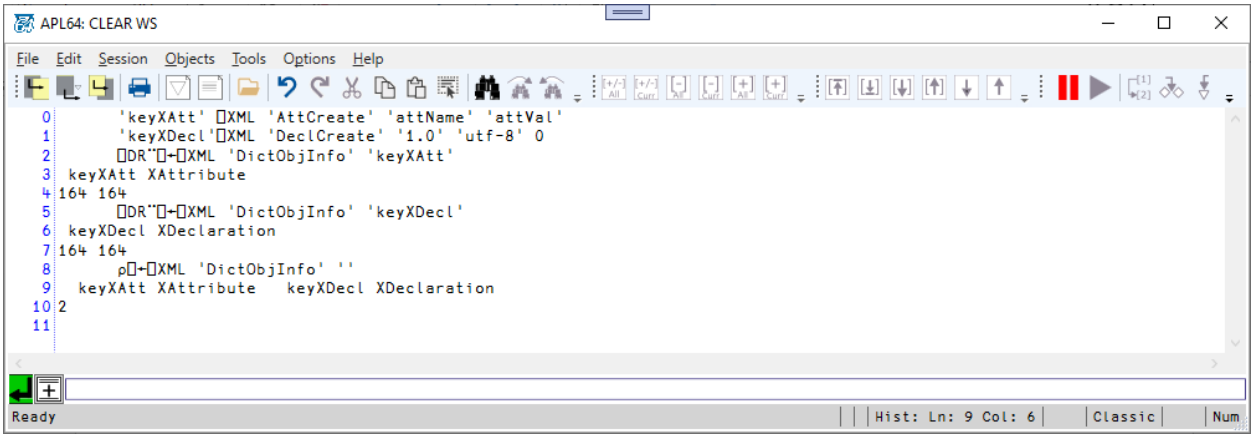
Action Synonym: DictInfoObj

keyXObj is the APL programmer-provided key in the ⎕XML XObject Dictionary associated with the XObject for which the xObjInfo is requested.

xObjInfo for a specific XObject is a two-element vector of strings containing the keyXObj and the XObject type.

xObjInfo for all objects is a vector of two-element vectors of strings, each containing the xKeyObj and the XObject type.

A value error exception will be thrown if keyXObj is not contained in the ⎕XML XObject Dictionarykeys.



DocCreate

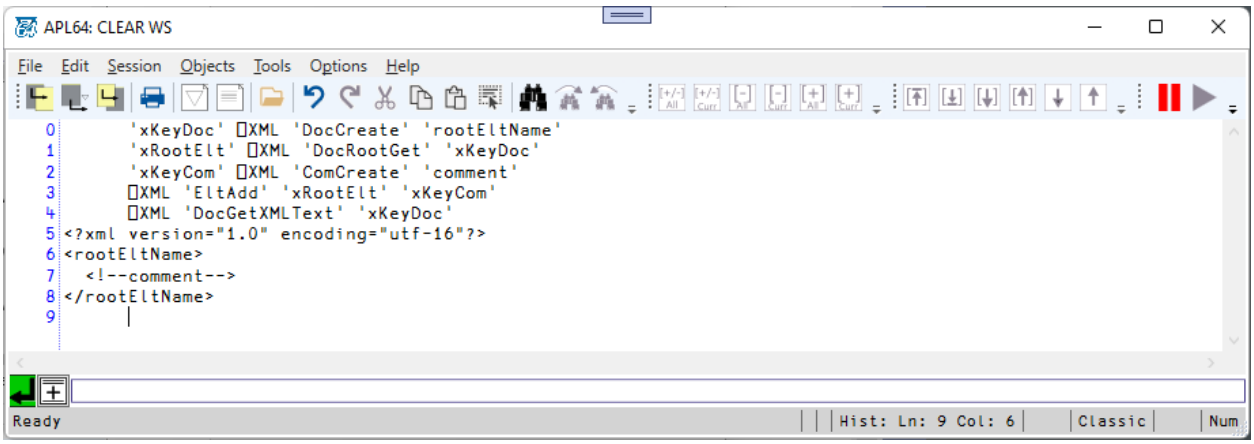
Syntax: keyXDoc ⎕XML "DocCreate" rootElementName

Arguments & Result:

Action Synonym: CreateDoc

keyXDoc is the APL programmer-selected key in the XML XObjects dictionary to be associated with the XDocument created by the DocCreate action.

rootElementName is the APL programmer-selected name for the root XElement in the new XDocument.



DocGetDecl

Syntax: keyXDecl ⎕XML "DocGetDecl" keyXDoc

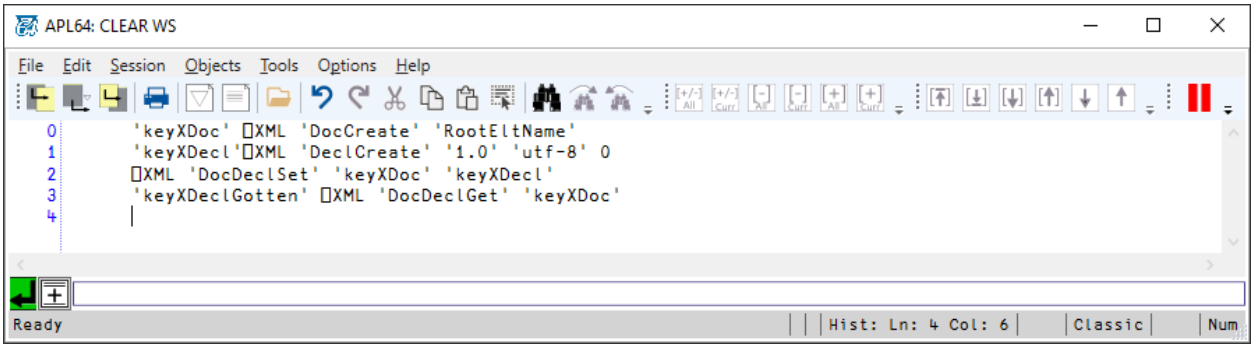
Arguments & Result:

Action Synonyms: DocDeclGet, GetDocDecl

keyXDecl is the APL programmer-selected key in the XML XObjects to be associated with the XDeclaration in the XDocument.

keyXDoc is the APL programmer-provided key in the XML XObjects dictionary of the existing XDocument containing the desired XDeclaration.

If the XDocument contains no XDeclaration, the result XDeclaration will be null.



DocGetRoot

Syntax: keyRoot ⎕XML "DocRootGet" keyXDoc

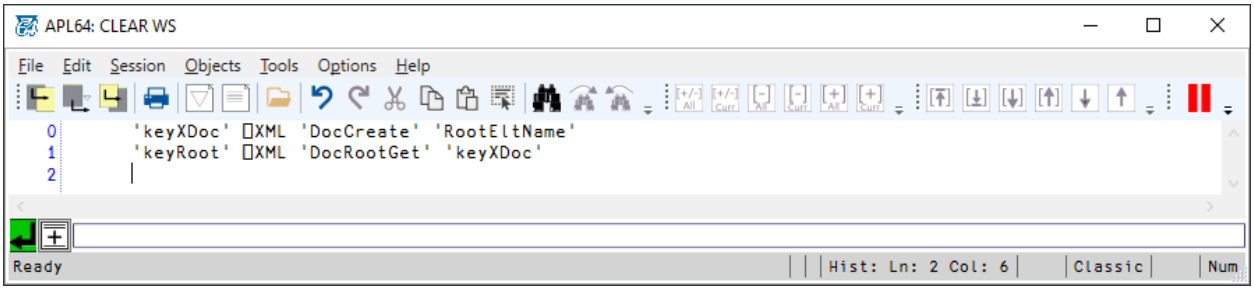
Arguments & Result:

Action Synonyms: DocRootGet, GetDocRoot

keyXDoc is the APL programmer-provided key in the ⎕XML XObject Dictionary associated with the existing XDocument.

keyRoot is the APL programmer-selected key in the ⎕XML XObject Dictionary which will be associated with the root element obtained from the existing XDocument.

If the XDocument contains no root element, the resulting root element will be null.



DocInsert

Syntax: ⍬⎕XML "DocInsert" xKeyDoc xKeyElt xKeyObj InsertionOrder

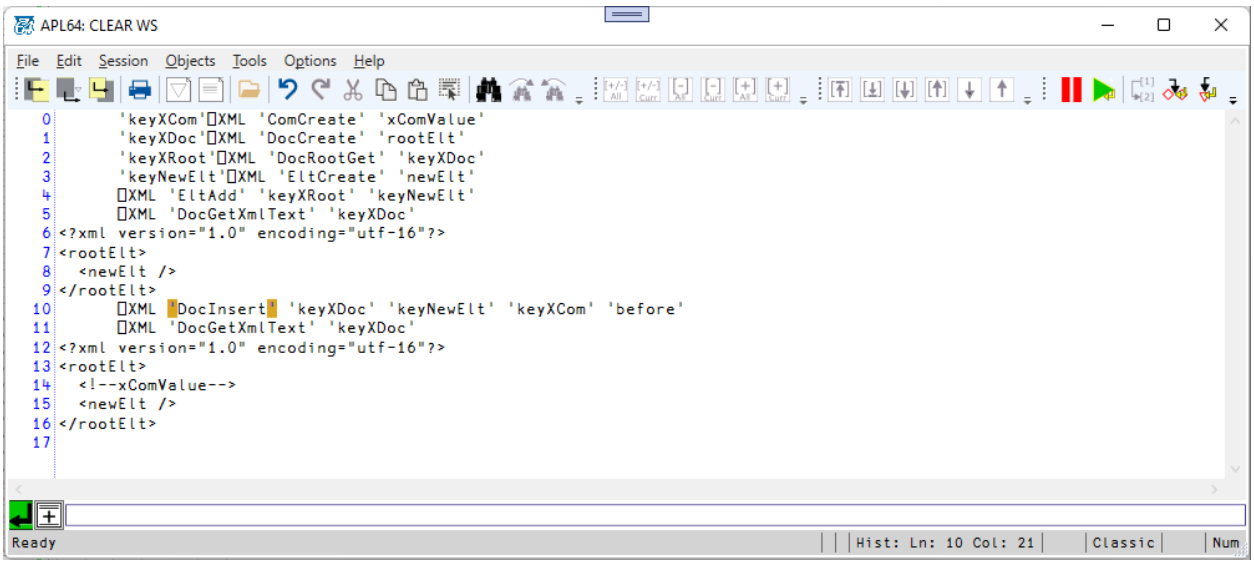
Arguments & Result:

xKeyDoc is the APL programmer-provided key in the XML XObjects dictionary associated with the existing XDocument into which the XObject will be inserted.

xKeyElt is the APL programmer-provided key in the XML XObjects dictionary associated with the existing XElement in the XDocument.

keyXObj is the APL programmer-provided key in the XML XObjects dictionary associated with the existing XObject to be inserted.

InsertionOrder is case-insensitive text which indicates if the XComment will be inserted before or after the specified XElement in the XDocument. Supported value of InsertionOrder are ‘before’ and ‘after’.



DocLoad

Syntax: keyXDoc ⍬⎕XML "DocLoad" xmlDocFilePath preserveWhiteSpace

Arguments & Result:

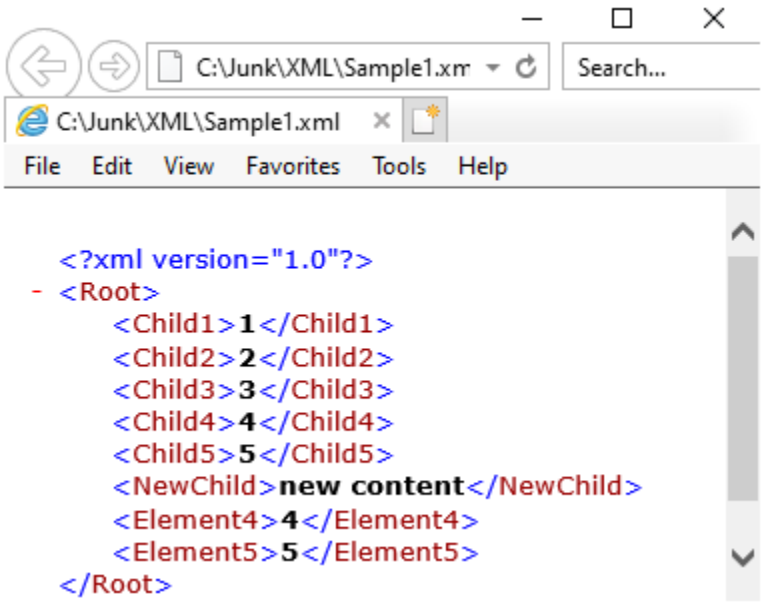
Action Synonym: LoadDoc

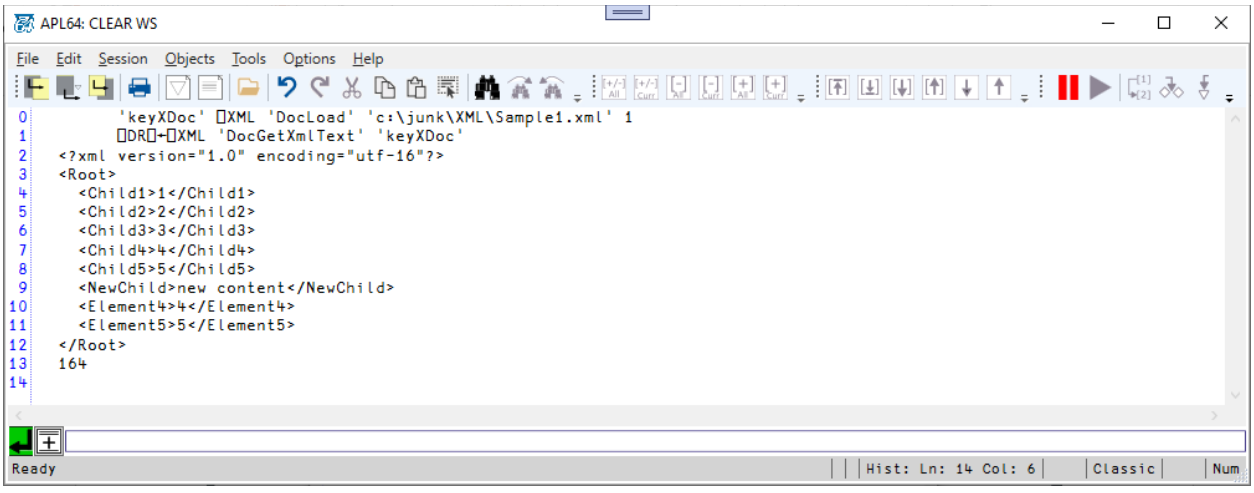
keyXElt is the APL programmer-selected key in the XML XObjects dictionary to be associated with the XDocument created by the DocLoad action.

xmlDocFilePath is the native file which is assumed to contain the text representation of an XDocument.

preserveWhiteSpace is an APL Boolean. If true, white space, if any, in the xmlText will be preserved in the XDocument created by this action. White space preservation may be preserved merely for the convenience of a human observer of the xml-format text.

Content of the Sample1.xml file:





DocParse

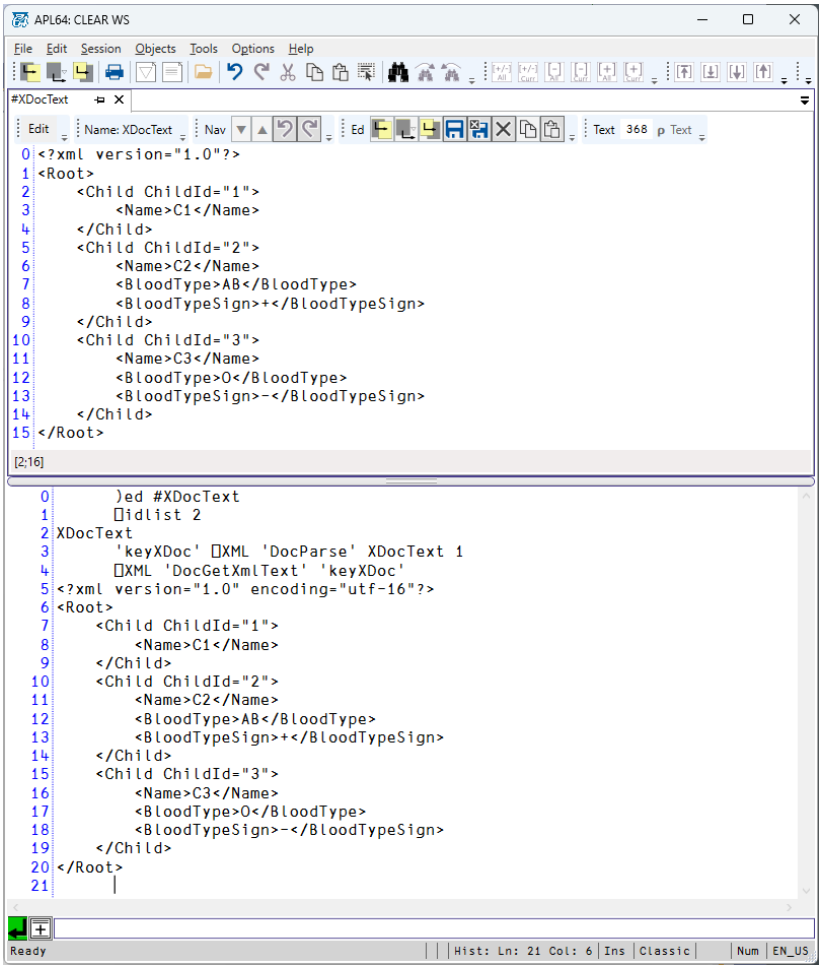
Syntax: keyXDoc ⎕XML "DocParse" xmlText preserveWhiteSpace

Arguments & Result:

keyXDoc is the APL programmer-selected key in the XML XObjects dictionary to be associated with the XDocument created by the DocParse action.

xmlText is APL programmer-provided text representation of an XDocument. Generally, xmlText is read, using ⎕NREAD or ⎕NFE, from an application-specific native-format file or received as a response from a web server.

preserveWhiteSpace is an APL Boolean. If true, white space, if any, in the xmlText will be preserved in the XDocument created by this action. White space preservation is merely for the convenience of a human observer of the xml-format text.



DocRemove

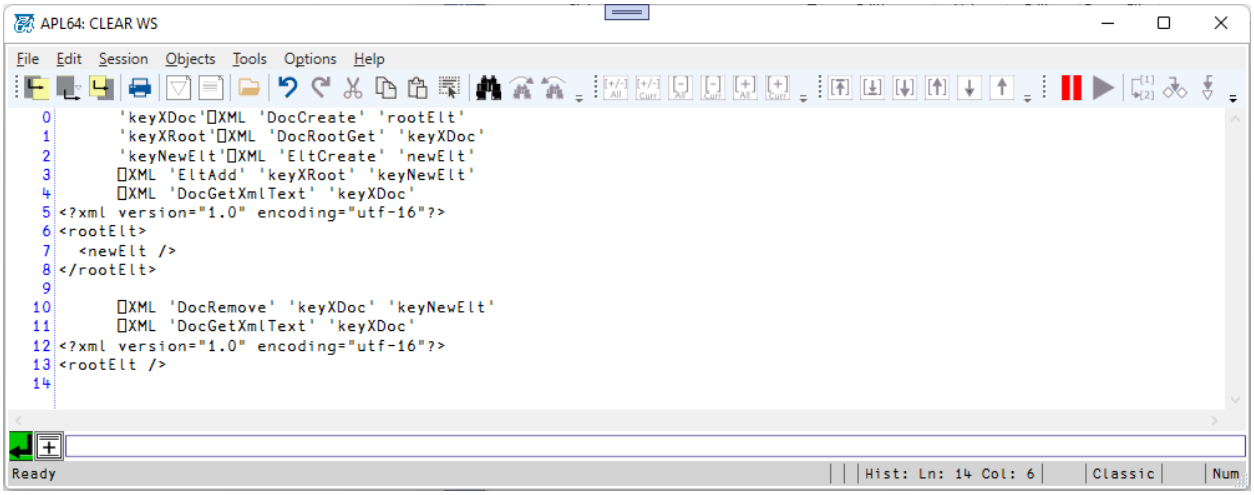
Syntax: ⎕XML 'DocRemove' 'keyXDoc' 'keyXObj'

Arguments & Result:

Action Synonyms: DocRemoveXObj, DocXObjRemove

keyXDoc is the APL programmer-provided key in the ⎕XML XObject Dictionary associated with the existing XDocument.

keyXObj is the APL programmer-provided key in the ⎕XML XObject Dictionary which is associated with the existing XObject to be removed from the XDocument.



DocSave

Syntax: ⎕XML "DocSave" keyXDoc xDocFilePath disableFormatting omitDuplicateNamespaces

Arguments & Result:

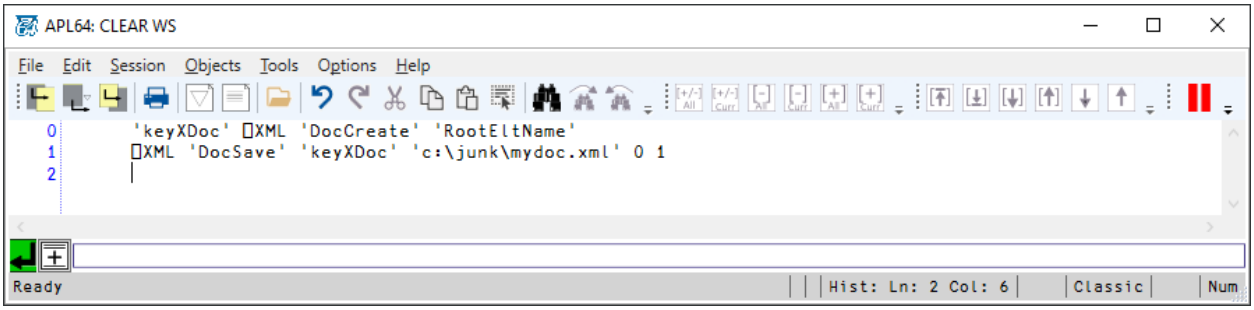
Action Synonym: SaveDoc

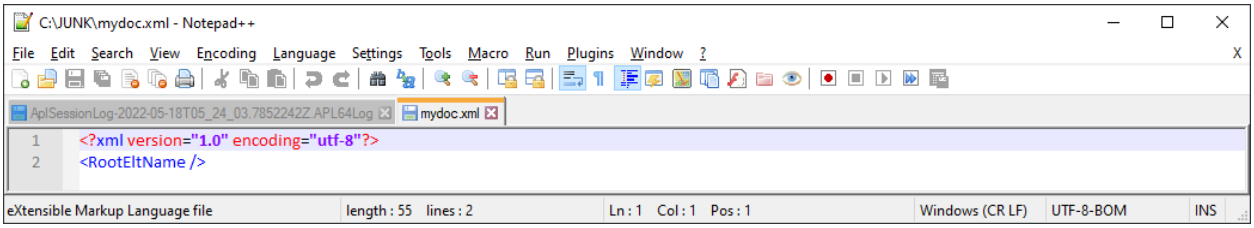
keyXDoc is the APL programmer-provided key in the ⎕XML XObject Dictionary associated with the existing XDocument to be saved.

xDocFilePath is the APL programmer-provided target file path for the saved XDocument. If the target file exists, it will be overwritten.

disableFormatting is a Boolean scalar.

omitDuplicateNamespaces is a Boolean scalar.





DocSetDecl

Syntax: ⎕XML "DocSetDecl" keyXDoc keyXDecl

Arguments & Result:

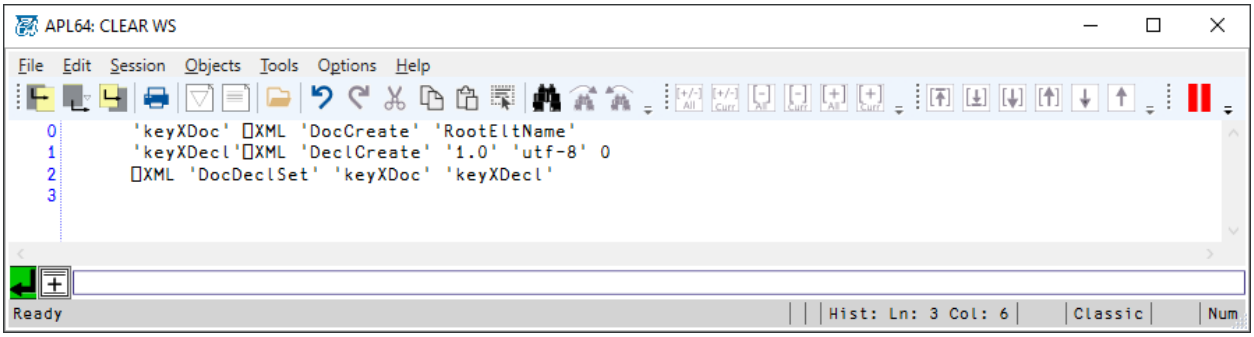
Action Synonyms: DocDeclSet, SetDocDecl

keyXDoc is the APL programmer-provided key in the XML XObjects dictionary of the existing XDocument to receive the XDeclaration.

keyXDecl is the APL programmer-provided key in the XML XObjects dictionary of the existing XDeclaration to be set in the XDocument.

An exception will be thrown if the source XDeclaration is null.

A pre-existing XDeclaration in the XDocument will be replaced by the DocDeclSet action.



EltAdd

Syntax: ⎕XML "EltAdd" keyTgtXElt xKeyObj [location]

Arguments & Result:

Action Synonyms: EltAddXObj, EltXObjAdd

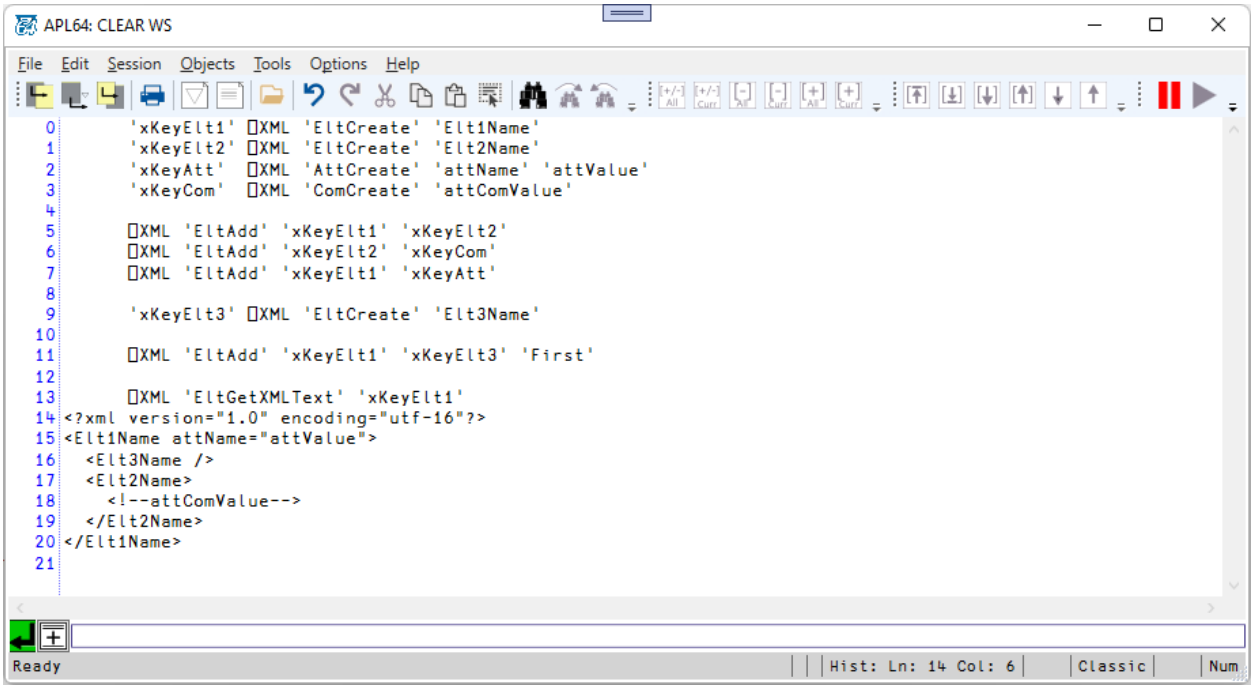
keyTgtXElt is the APL programmer-provided key in the ⎕XML XObject Dictionary associated with the existing target XElement.

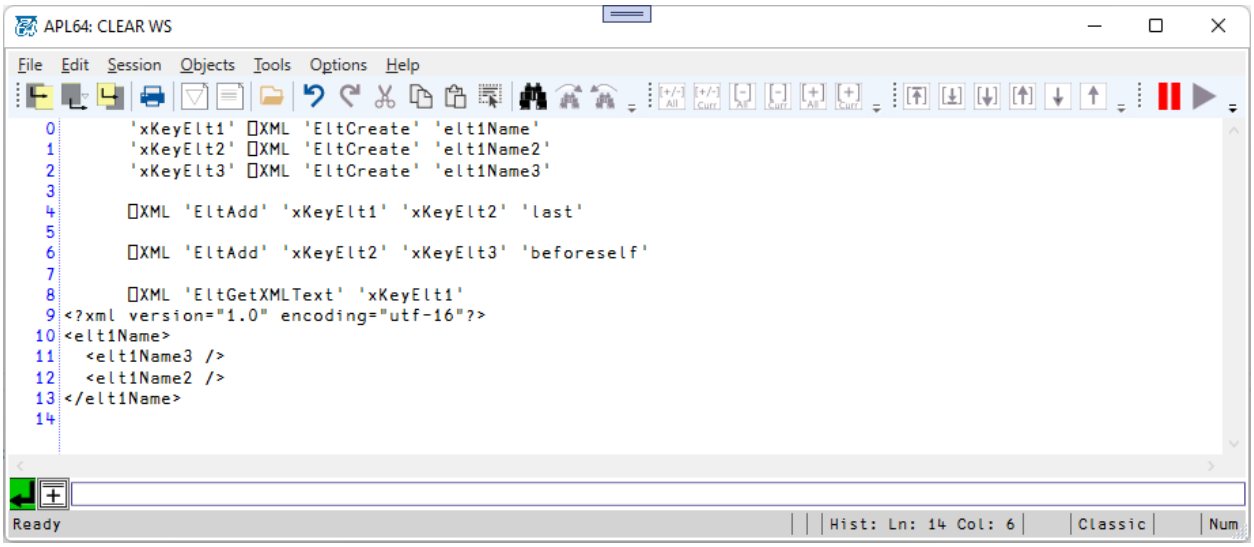
xKeyObj is the APL programmer-provided key in the ⎕XML XObject Dictionary associated with the existing source XObject, which can be an XAttribute, XElement, XComment or XCData type.

The 'EltAdd' ⎕XML action will insert the XObject into the XElement associated with keyTgtXElt. If the specified XObject exists, it will be replaced.

Location is an optional, context-insensitive, APL-programmer provided text which indicates the location for the added XObject:

Location	XObject location in target XElement
'last' (or if location is not provided)	After all other content of XElement
'first'	Before all other content of XElement





EltAttRemove

Syntax: ⎕XML "EltAttRemove" keySrcXElt xAttName

Arguments & Result:

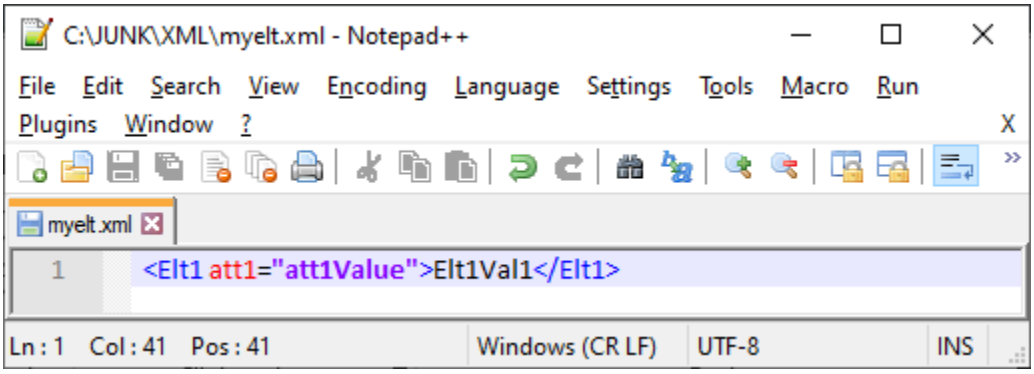
Action Synonym: EltRemoveAtt

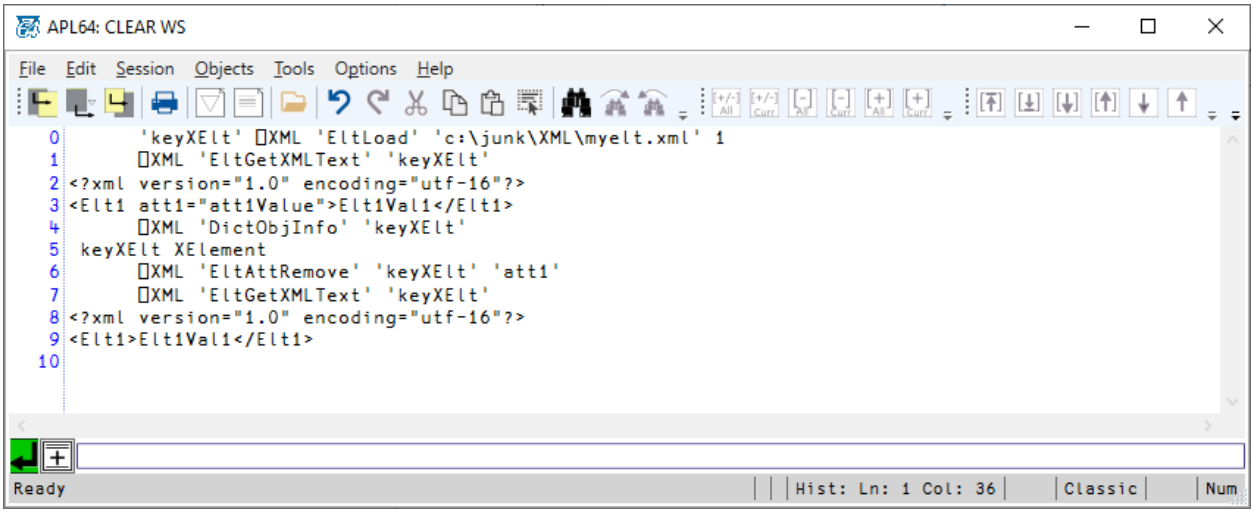
keySrcXElt is the APL programmer-provided key in the ⎕XML XObject Dictionary associated with the existing XElement which contains the XAttribute to be removed.

xAttName is the name of the XAttribute in the XElement associated with keySrcXElt.

No exception will be thrown if the specified XAttribute does not exist in the target XElement.

Source XElement in the c:\junk\XML\myelt.xml:





EltCreate

Syntax: keyXElt ⎕XML "EltCreate" XEltName [nameSpaceInfo]

Arguments & Result:

Action Synonym: CreateElt

keyXElt is the APL programmer-selected key in the XML XObjects dictionary to be associated with the XElement created by the EltCreate action.

XEltName is the name of the XElement to be created.

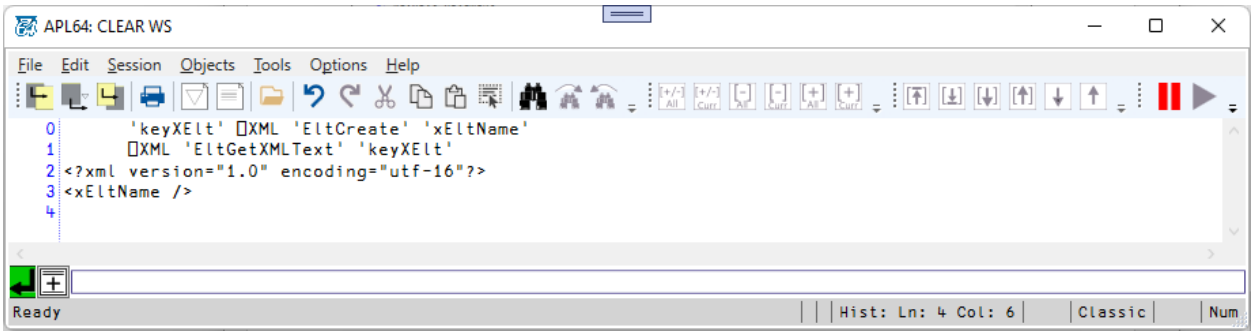
nameSpaceInfo is an optional, APL programmer-provided, rank-2 array containing the XNamespaces to include in the new XElement.

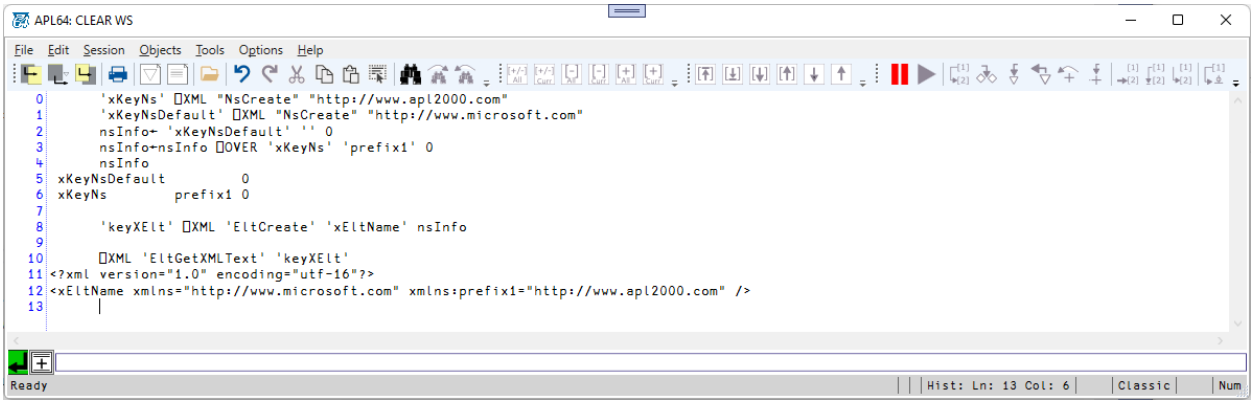
nameSpaceInfo Column#	**nameSpaceInfo Column Description**
1	xKeyNs
2	nsPrefix
3	NameSpaceAppliesToNewXElt: 0/No 1/Yes

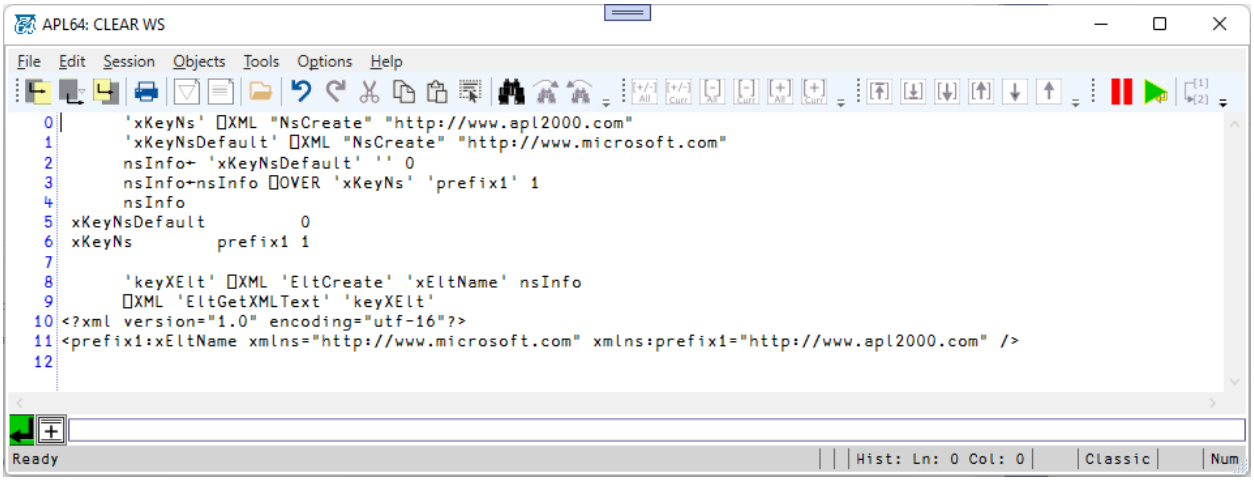
xKeyNs is the APL programmer-provided key in the XML XObjects dictionary which is associated with an existing XNamespace to be included in the new XElement.

nsPrefix is an APL programmer-provided namespace prefix text. nsPrefix is "" for the default namespace, if any, of the document. There can be only one default namespace in an XDocument.

NameSpaceAppliesToNewXElt indicates if the selected namespacewill apply to the new XElement. At most one namespace in the namespaceInfo array can apply to the new XElement.







EltGet

Syntax: keyXElt ⎕XML "EltGet" keySrcXElt xEltTgtName

Arguments & Result:

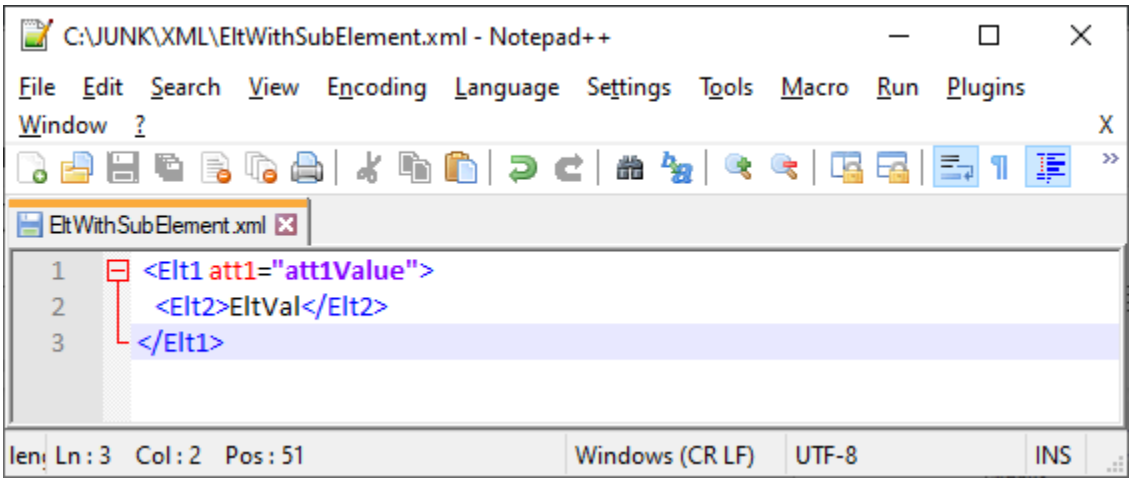
Action Synonyms: GetElt, GetFirstElt, EltGetElt, EltGetEltByName, EltGetFirstElt, EltGetFirstEltByName

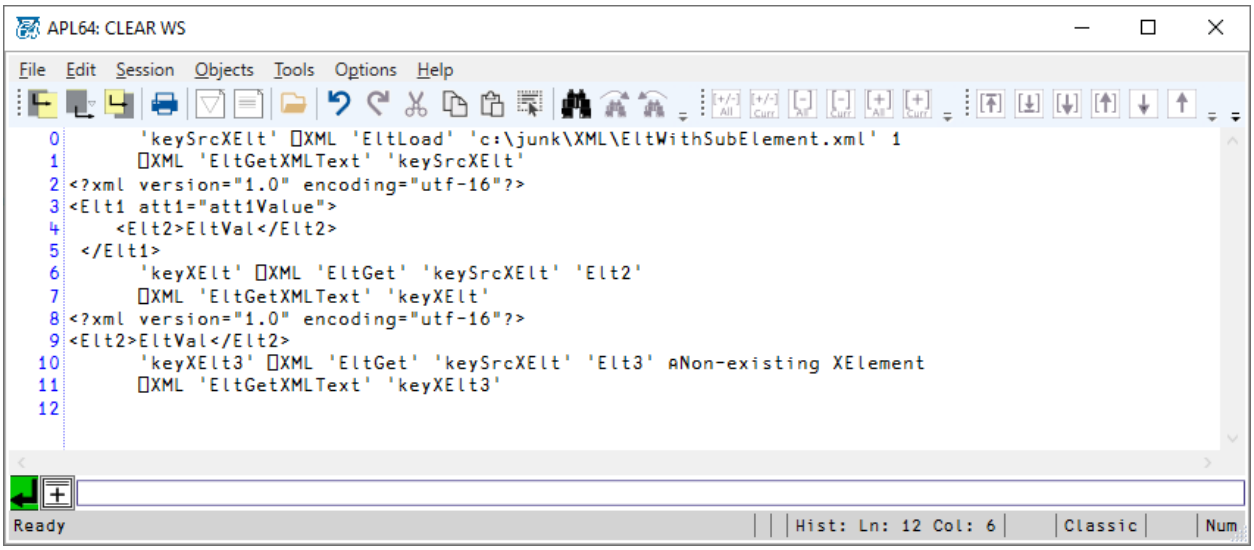
The EltGet action obtains the first element in the source XElement with the specified target name.

keySrcXElt is the APL programmer-provided key in the ⎕XML XObject Dictionary associated with the existing XElement which contains the XElement with the specified target name.

xEltTgtName is the name of the XElement to be obtained.

File containing XElement with a subordinate XElement





EltGetAtt

Syntax: keyXAtt ⎕XML "EltAttGet" keySrcXElt xAttName

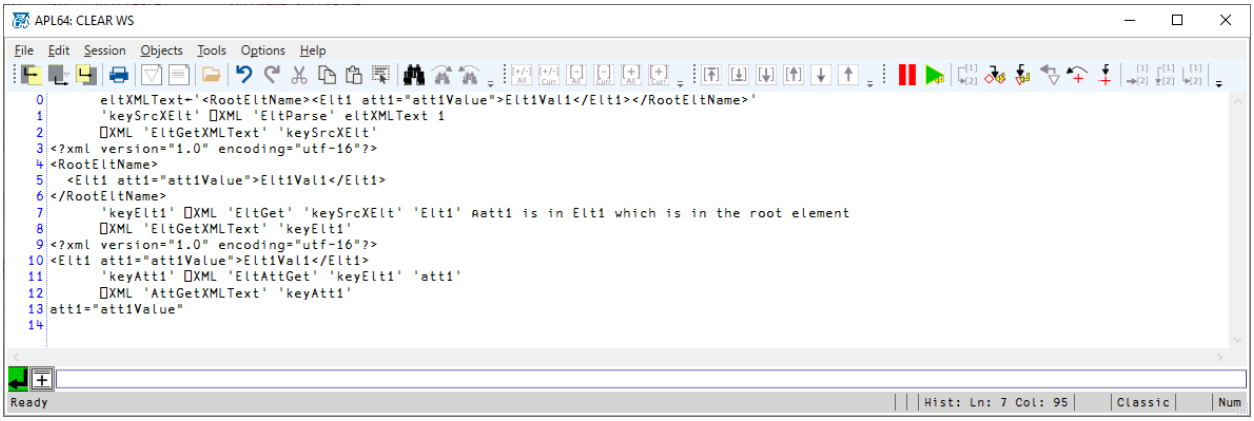
Arguments & Result:

Action Synonym: EltAttGet

keyXAtt is the APL programmer-provided key in the ⎕XML XObject Dictionary which will be associated with the obtained XAttribute.

keySrcXElt is the APL programmer-provided key in the ⎕XML XObject Dictionary associated with the existing XElement which contains the desired XAttribute.

xAttName is the name of the XAttribute.



EltGetAttByIndex

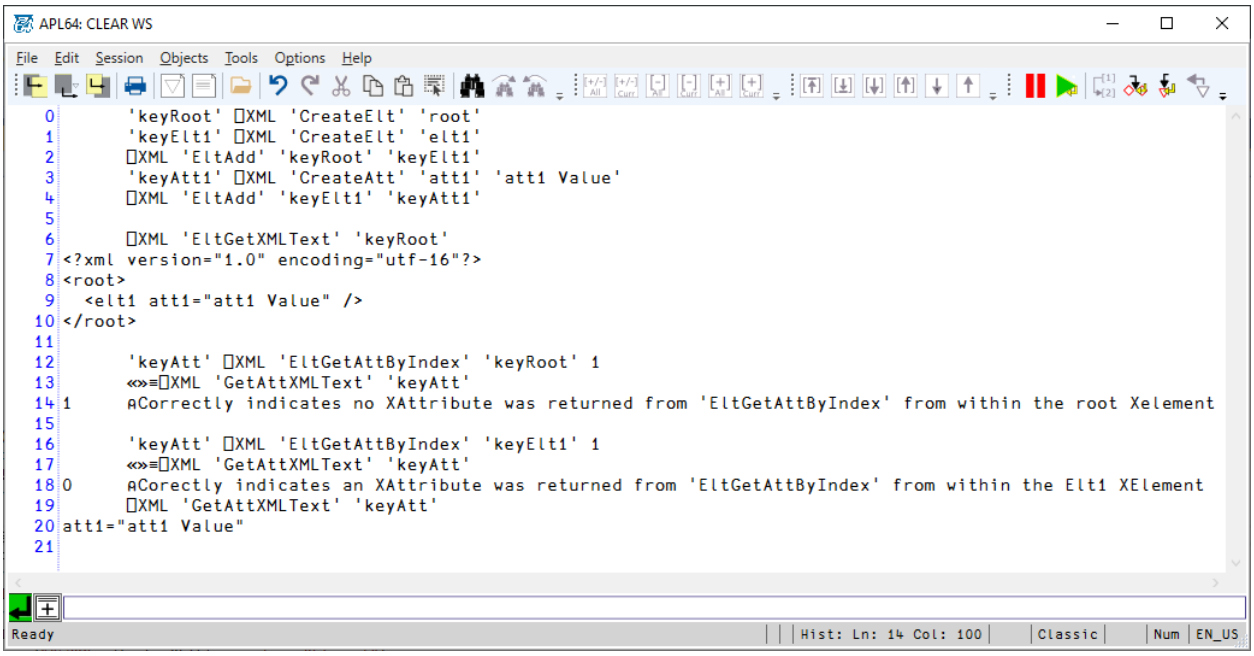
Syntax: keyXAtt ⎕XML "EltGetByIndex" keySrcXElt attIndex

Arguments & Result:

The EltGetAttByIndex action obtains the attIndex -th XAttribute, if any, in the source XElement. This action does not consider XAttributes which may exist in XElements which are contained in the source XElement.

keySrcXElt is the APL programmer-provided key in the ⎕XML XObject Dictionary associated with the existing XElement which contains the desired XAttribute.

attIndex is the APL programmer-specified, ⎕IO significant, integer index of the XElement to obtain. No exception is thrown if the index is out of range. Use the 'AttGetXMLText' action to determine if the XAttribute returned is valid.



EltGetEltByIndex

Syntax: keyXElt ⎕XML "EltGetByIndex" keySrcXElt eltIndex

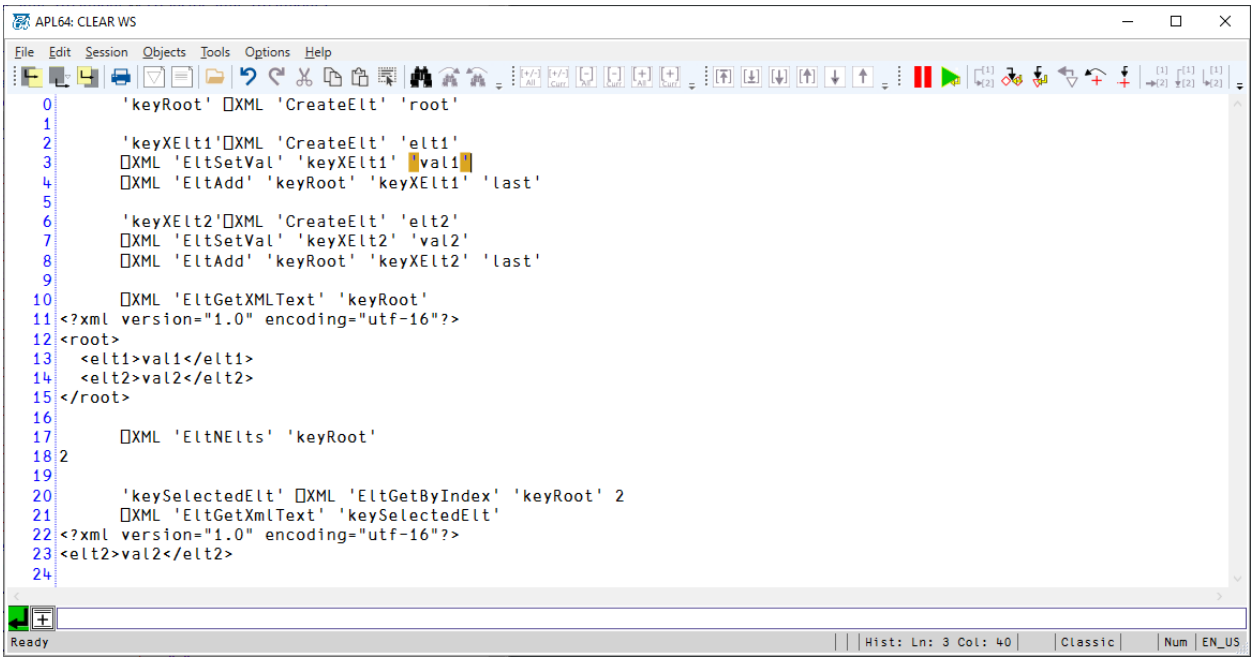
Arguments & Result:

Action Synonym: EltGetEltByIndex

The EltGetEltByIndex action obtains the eltIndex-th XElement, if any, in the source XElement. . This action does not consider XElements which may exist in XElements which are contained in the source XElement.

keySrcXElt is the APL programmer-provided key in the ⎕XML XObject Dictionary associated with the existing XElement which contains the desired XElement.

eltIndex is the APL programmer-specified, ⎕IO significant, integer index of the XElement to obtain. No exception is thrown if the index is out of range. Use the 'EltGetXMLText' action to determine if the XElement returned is valid.



EltGetVal

Syntax: (eltName(string) eltValue)←⎕XML "EltGetVal" keyXElt [parse(bool)]

Action Synonyms: EltGetValue, EltValGet, EltValueGet, GetEltVal, GetEltValue

Arguments & Result:

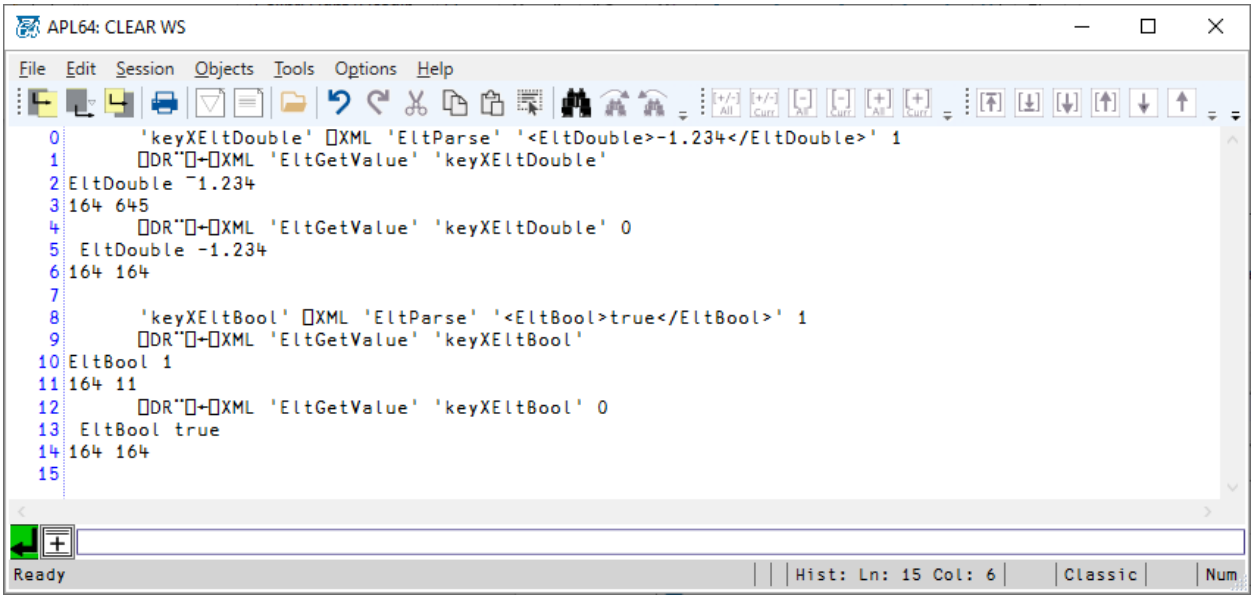
eltName is the value of the Name property of the specified XElement.

eltValue is the value of the specified XElement.

parse is an optional Boolean scalar argument with default value 1/True.

- If parse is true, the eltValue returned is Value property of the specified XElement parsed into the applicable APL data type, which can be a scalar integer, double, string or Boolean or a ⎕TS-format integer vector assuming UTC timezone.
- If parse if false, the Value property of the specified XElement as a scalar string.

keyXElt is the APL programmer-provided key in the XML XObjects dictionary associated with the existing XElement.



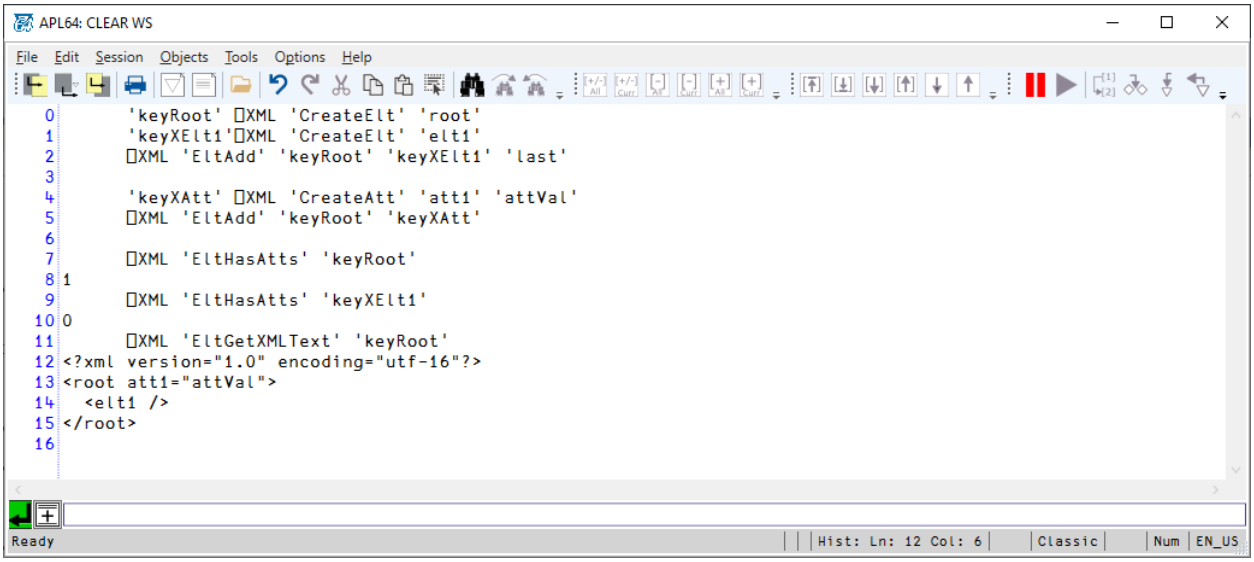
EltHasAtts

Syntax: Int32←⎕XML "EltNAtt" keyXElt

Arguments & Result:

Returns: 0/False 1/True

keyXElt is the APL programmer-provided key in the XML XObjects dictionary associated with the existing XElement.



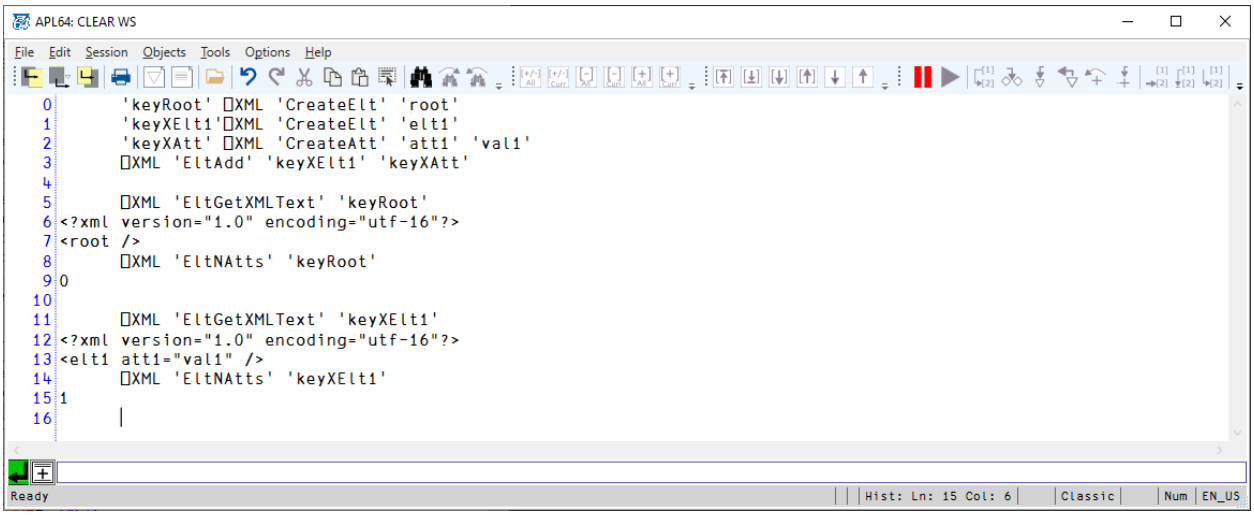
EltNAtts

Syntax: Int32←⎕XML "EltNAtt" keyXElt

Arguments & Result:

Returns the number of XAttributes in the specified XElement.

keyXElt is the APL programmer-provided key in the XML XObjects dictionary associated with the existing XElement.



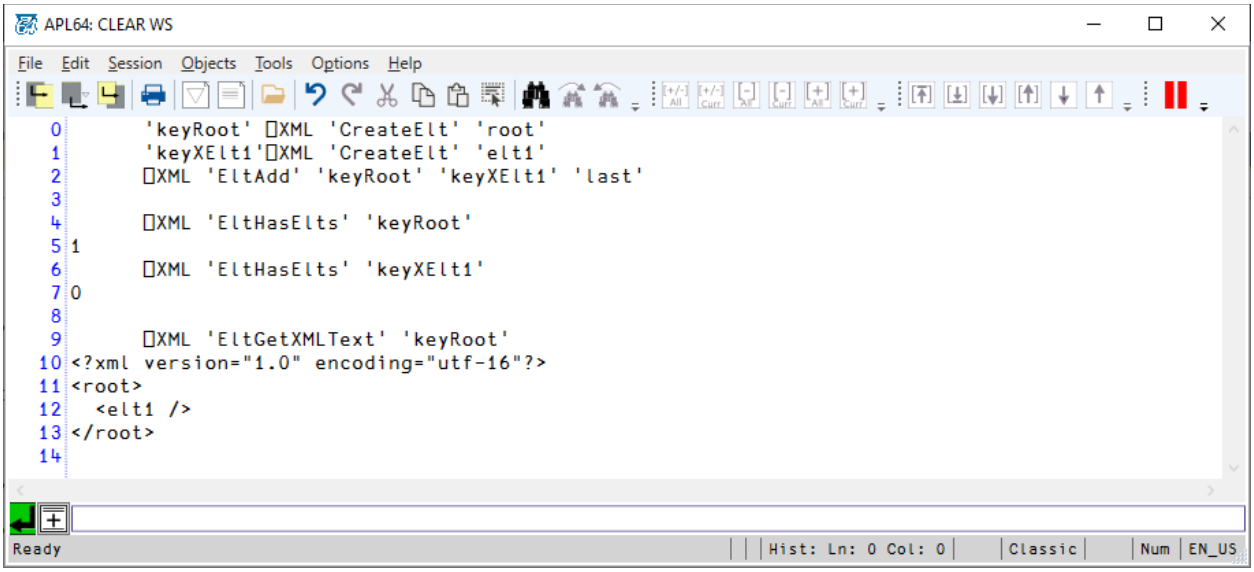
EltHasElts

Syntax: Int32←⎕XML "EltHasElts" keyXElt

Arguments & Result:

Returns: 0/False 1/True

keyXElt is the APL programmer-provided key in the XML XObjects dictionary associated with the existing XElement.



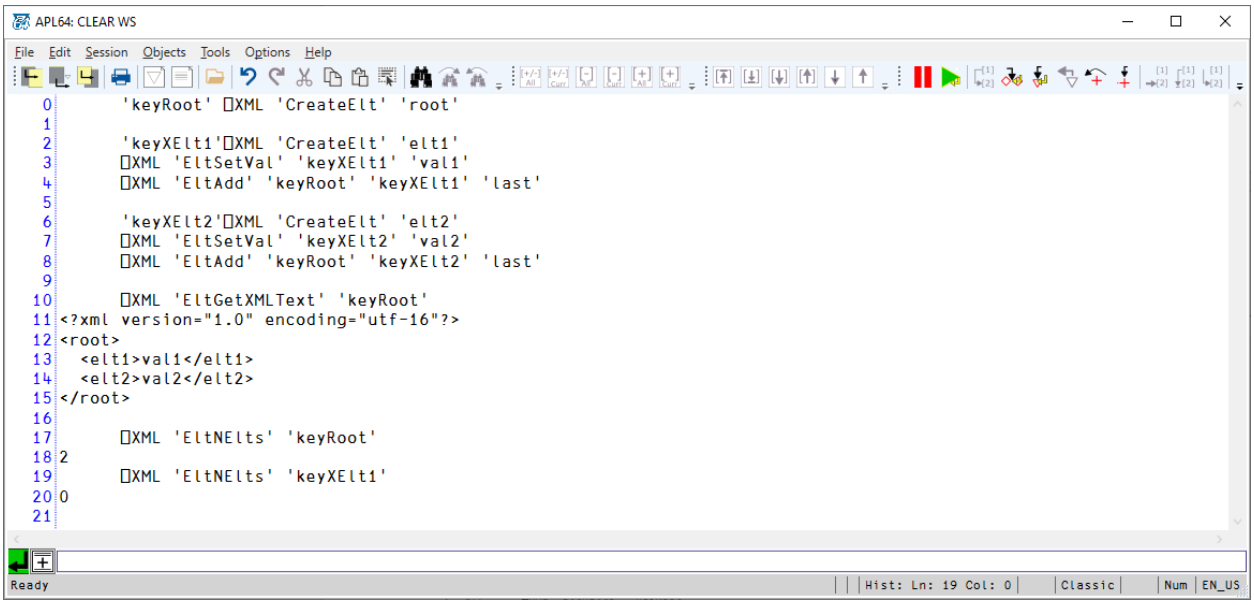
EltNElts

Syntax: Int32←⎕XML "EltHasElts" keyXElt

Arguments & Result:

Returns the number of XElements in the specified XElement.

keyXElt is the APL programmer-provided key in the XML XObjects dictionary associated with the existing XElement.



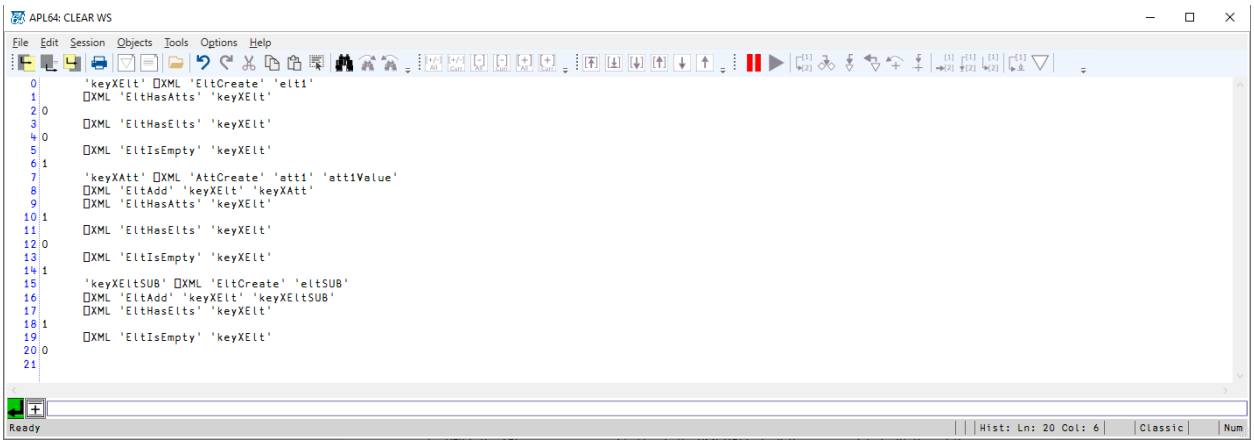
EltIsEmpty

Syntax: bool←⎕XML "EltIsEmpty" keyXElt

Arguments & Result:

Returns an APL scalar Boolean

keyXElt is the APL programmer-provided key in the XML XObjects dictionary associated with the existing XElement.



EltLoad

Syntax: keyXElt ⎕XML "EltLoad" xmlElementFilePathpreserveWhiteSpace

Arguments & Result:

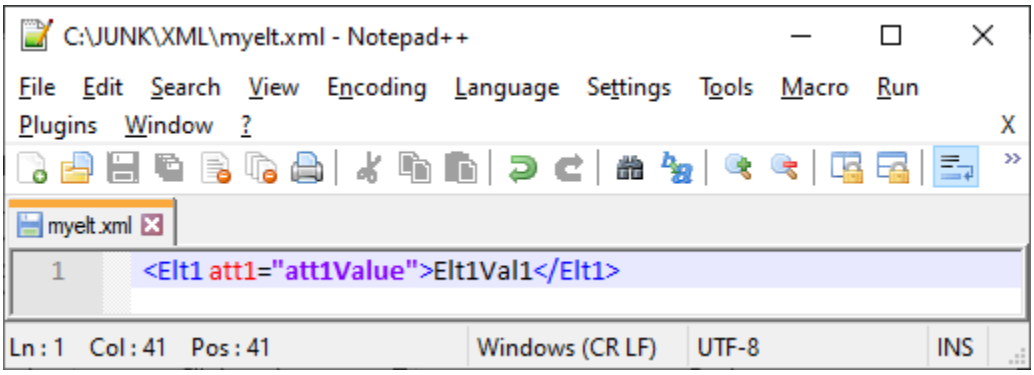
Action Synonym: LoadElt

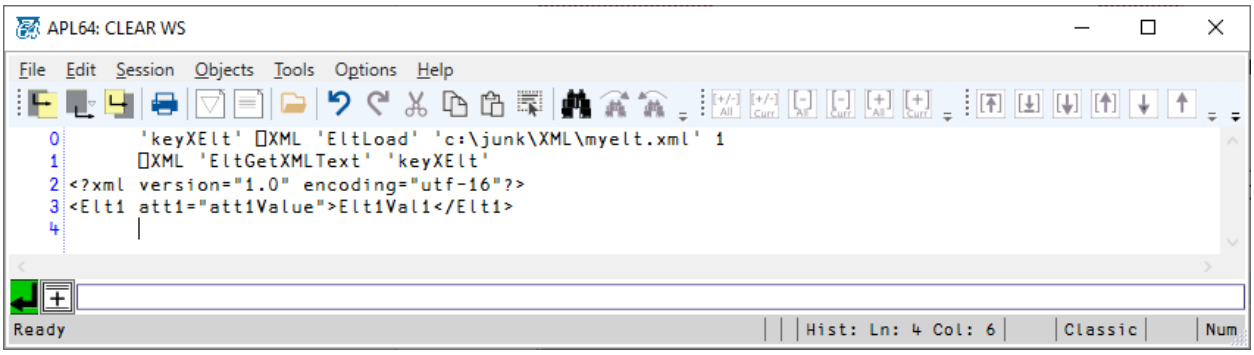
keyXElt is the APL programmer-selected key in the XML XObjects dictionary to be associated with the XElement loaded by the EltLoad action.

xmlElementFilePath is path of the file containing the text of the XElement to load.

preserveWhiteSpace is an APL Boolean. If true, white space, if any, in the xmlText will be preserved in the XElement loaded. White space preservation is merely for the convenience of a human observer of the xml-format text.

Content of the myelt.xml file:





EltParse

Syntax: keyXElt ⎕XML "EltParse" xmlText preserveWhiteSpace

Arguments & Result:

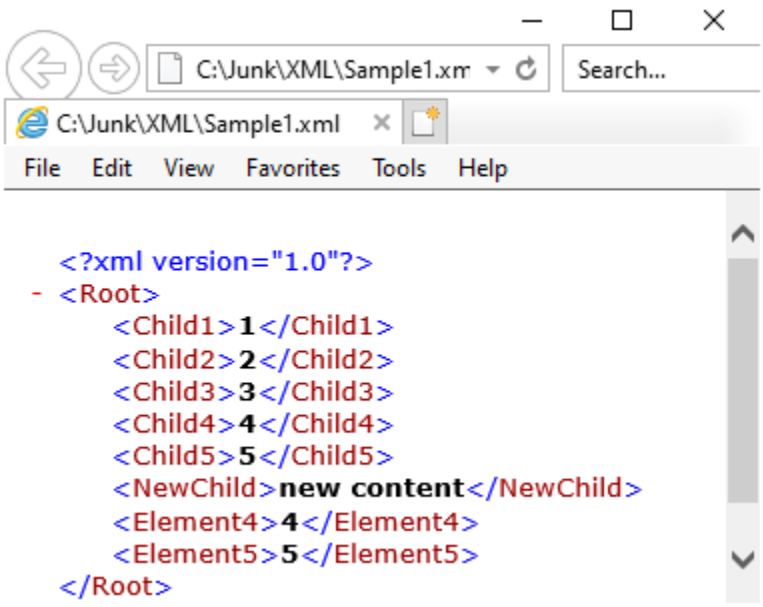
Action Synonym: ParseElt

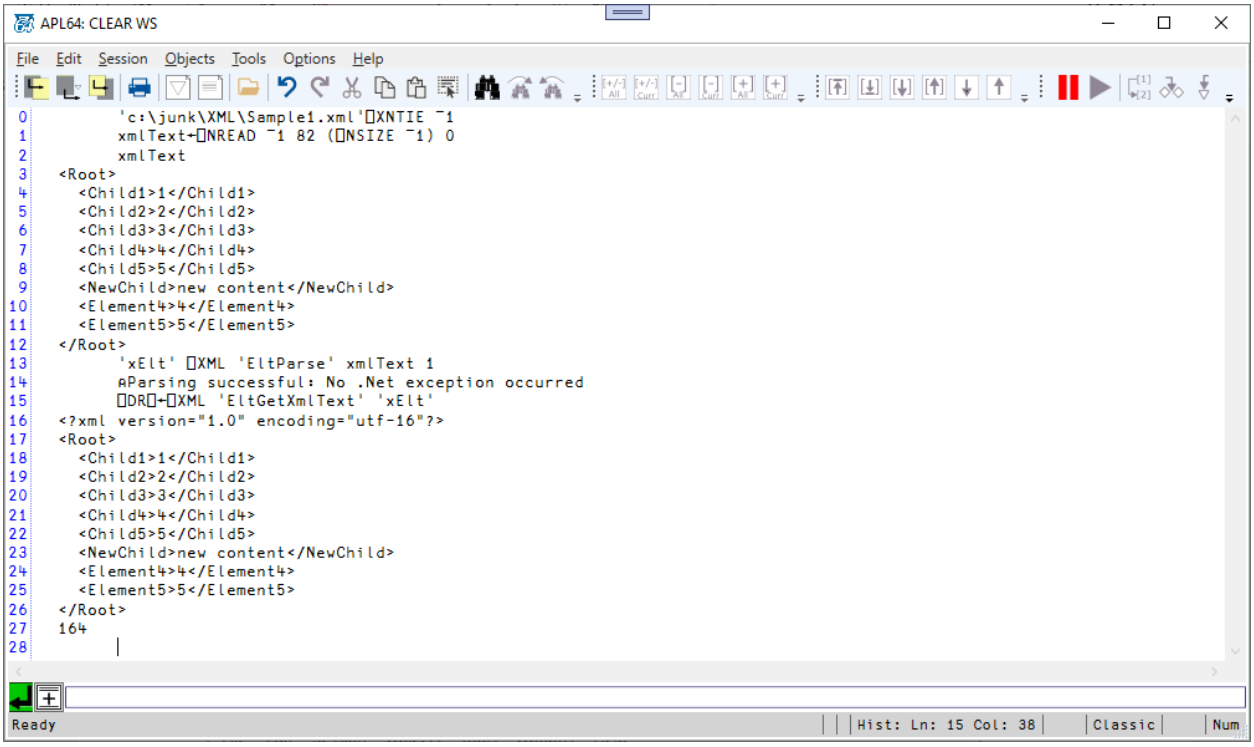
keyXElt is the APL programmer-selected key in the XML XObjects dictionary to be associated with the XElement created by the EltParse action.

xmlText is APL programmer-provided text representation of an XElement. Generally, xmlText is read, using □NREAD or □NFE, from an application-specific native-format file or received as a response from a web server.

preserveWhiteSpace is an APL Boolean. If true, white space, if any, in the xmlText will be preserved in the XElement created by this action. White space preservation is merely for the convenience of a human observer of the xml-format text.

Content of the Sample1.xml file:





EltRemove

Syntax: ⎕XML "EltRemove" Elt keyTgtXElt childEltName Ns [eltIndex]

Arguments & Result:

keyTgtXElt is the APL programmer-provided key in the ⎕XML XObject dictionary that is associated with the target XElement.

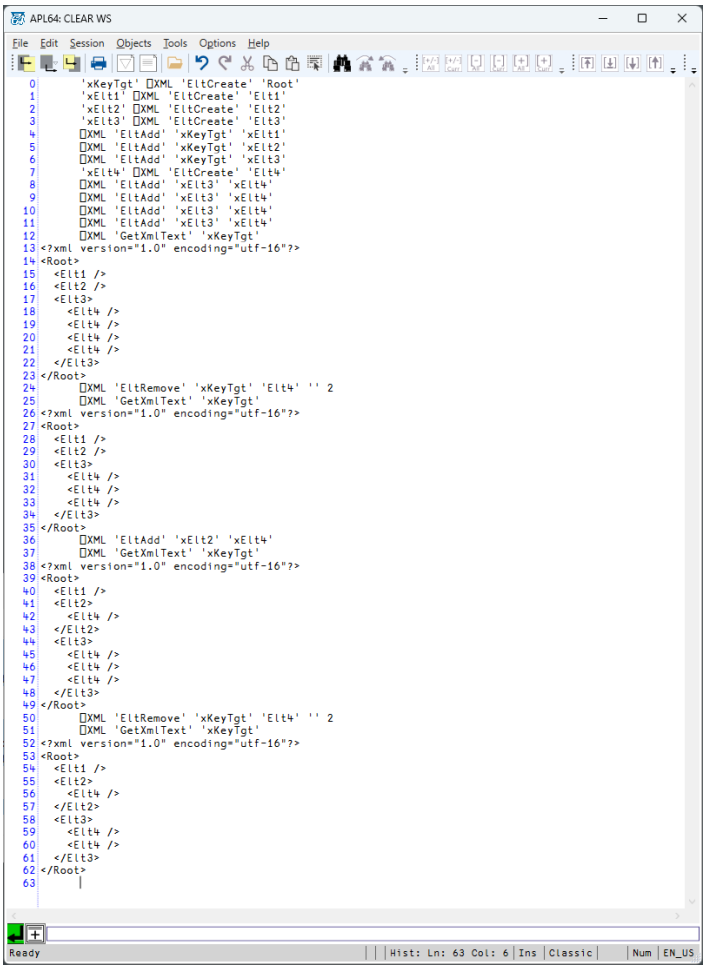
childEltName is the name of a child XElement to be removed from the target XElement.

If there are multiple child XElements with the childEltName in the target XElement, the optional eltIndex argument will be used to select the child element to delete.

Ns is the APL programmer-provided text for the XNamespace associated with the XElement descendants, or "" if none applies.

eltIndex is the optional index origin zero index of the XElement to remove. Its default value is 0 if the argument is not present.

No action will be taken, and no exception thrown if the child element does not exist.



EltRemoveAll

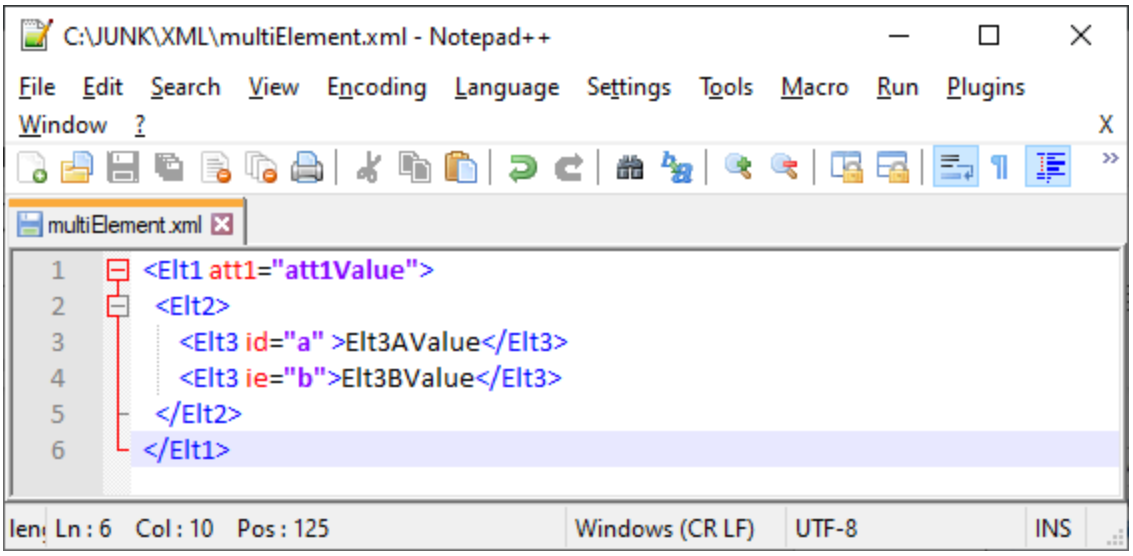
Syntax: ⎕XML "EltRemoveAll" keyXElt

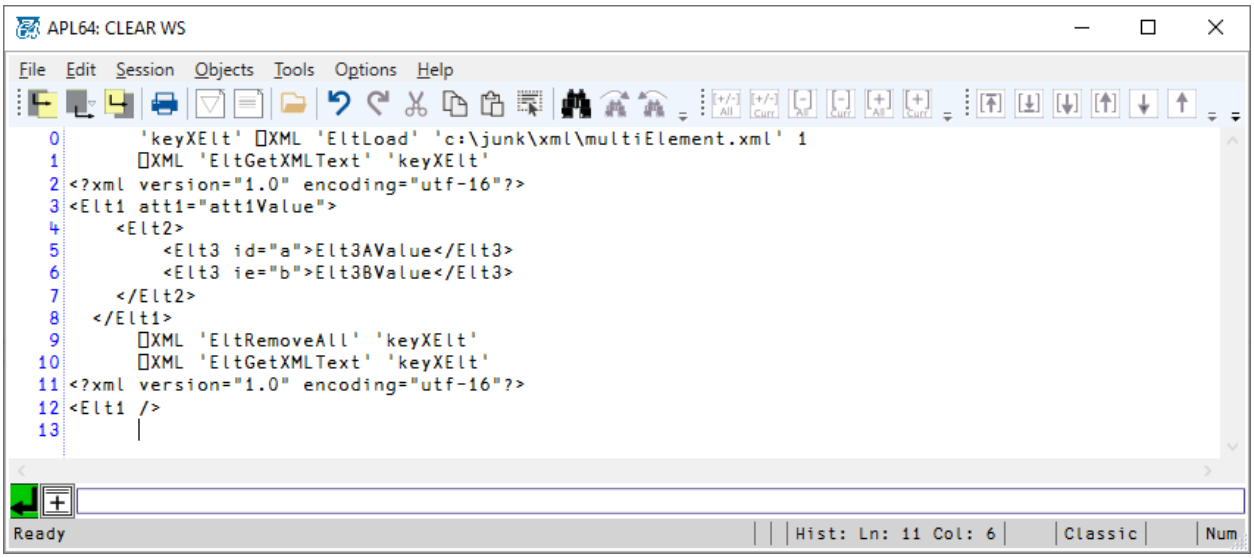
Arguments & Result:

The EltRemoveAll action will remove all subordinate elements from the specified XElement.

keyXElt is the APL programmer-provided key in the XML XObjects dictionary which is associated with the existing XElement to be modified by the EltRemoveAll action.

Source XElement:





EltRemoveAtts

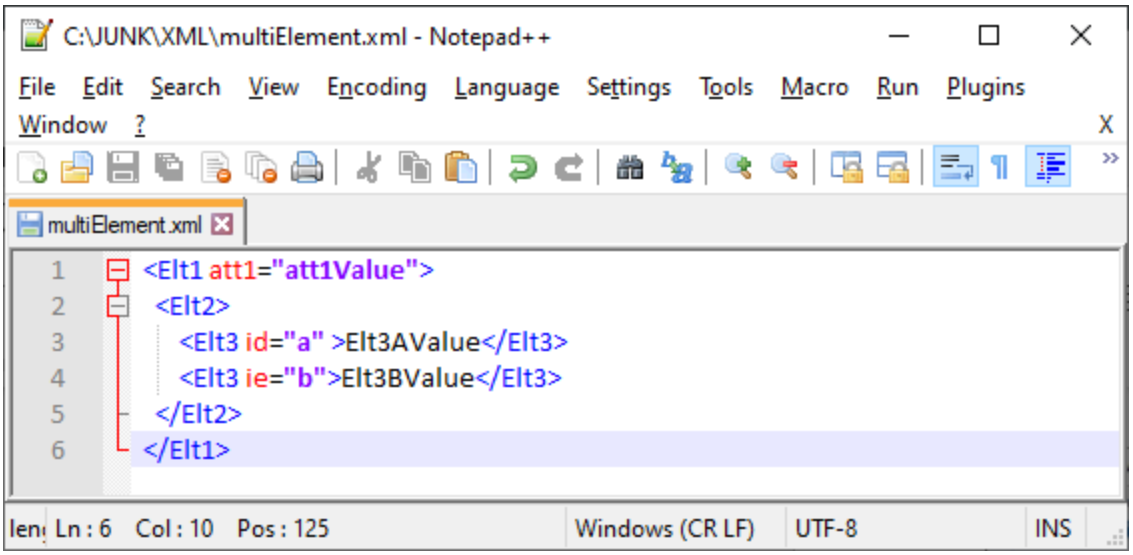
Syntax: ⎕XML "EltRemoveAtts" keyXElt

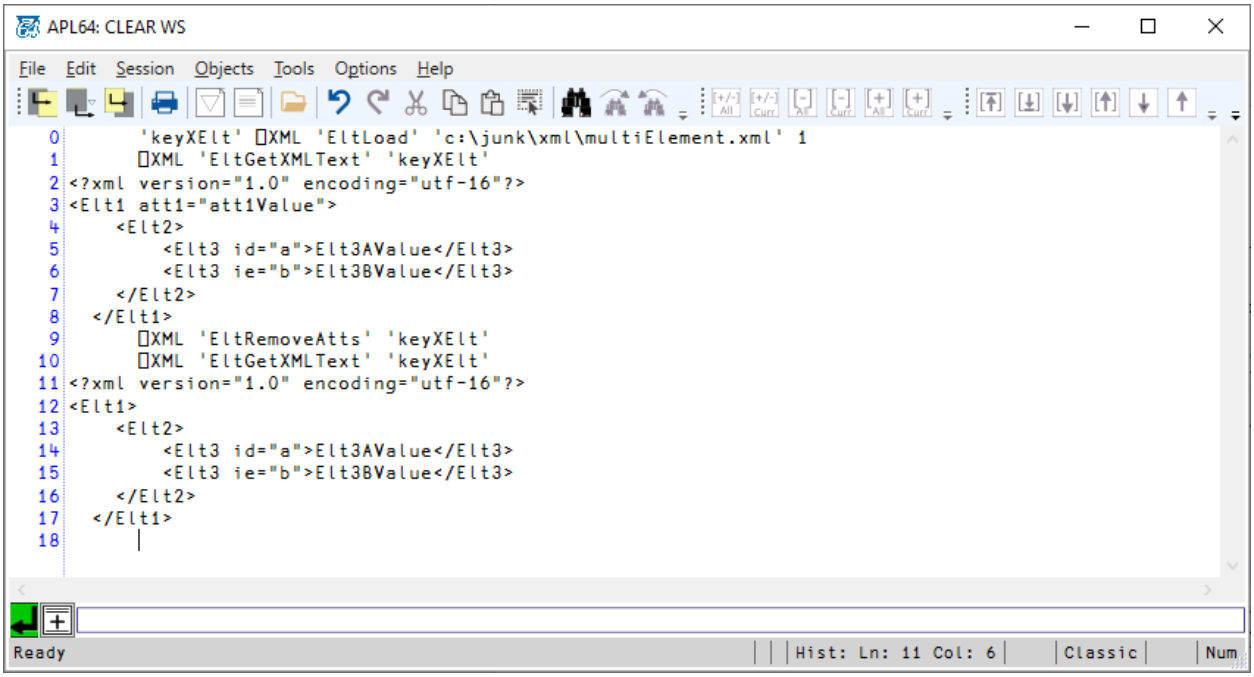
Arguments & Result:

The EltRemoveAtts action will remove all XAttributes ubordinate elements from the specified XElement.

keyXElt is the APL programmer-provided key in the XML XObjects dictionary which is associated with the existing XElement to be modified by the EltRemoveAtts action.

Source XElement:





EltSave

Syntax: ⎕XML "EltSave" keyXElt xDocFilePath disableFormatting omitDuplicateNamespaces

Arguments & Result

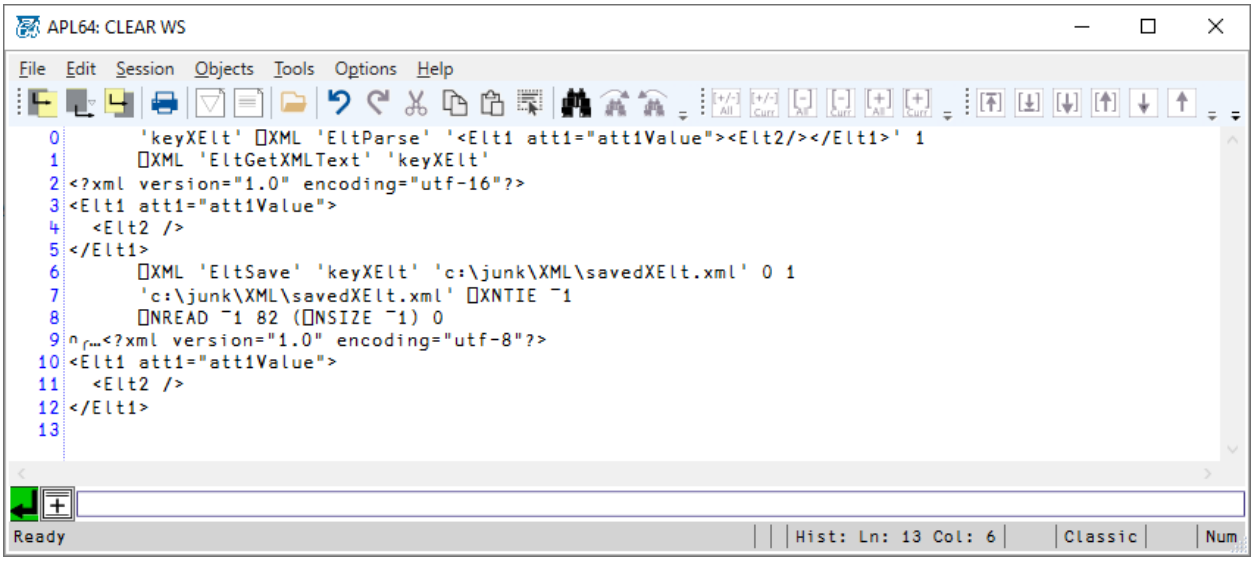
Action Synonym: SaveElt

keyXElt is the APL programmer-provided key in the ⎕XML XObjects dictionary associated with the existing XElement to be saved

xDocFilePath is the target file path where the XML-format text representation of the XElement will be saved. If the target file exists, it will be overwritten.

disableFormatting is a Boolean scalar.

omitDuplicateNamespaces is a Boolean scalar.



EltSetAttVal

Syntax: ⎕XML "EltSetAttVal" keyTgtXElt attName [attValue]

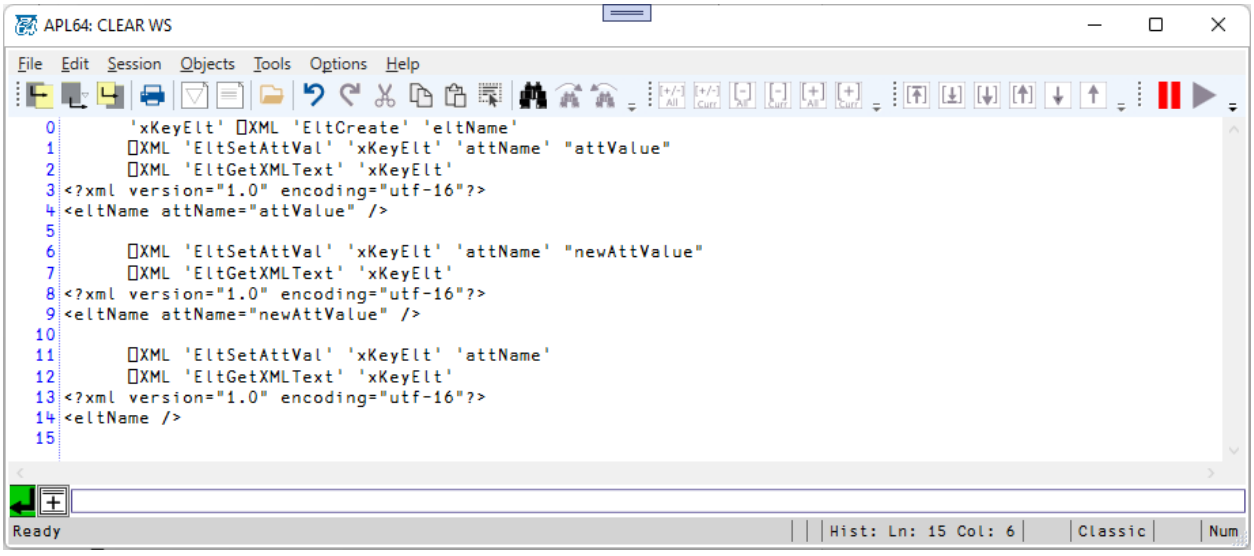
Arguments & Results:

Action Synonyms: EltAttValSet, SetEltAttVal

keyTgtXElt is the APL programmer-provided key in the ⎕XML XObject dictionary that is associated with the target XElement.

attName is the APL programmer-provided name of the XAttribute to add, delete or modify in the XDocument.

attValue is the APL programmer-provided value of the XAttribute to add or modify. If attValue is not provided, the XAttribute named attName in the xElement, if present, will be deleted.



EltSetEltVal

Syntax: □XML "EltSetEltVal" keyTgtXElt childEltName [childEltVal]

Arguments & Result:

The EltSetEltVal action will set the value of the child XElement named childEltName in the source XElement, to the childEltVal.

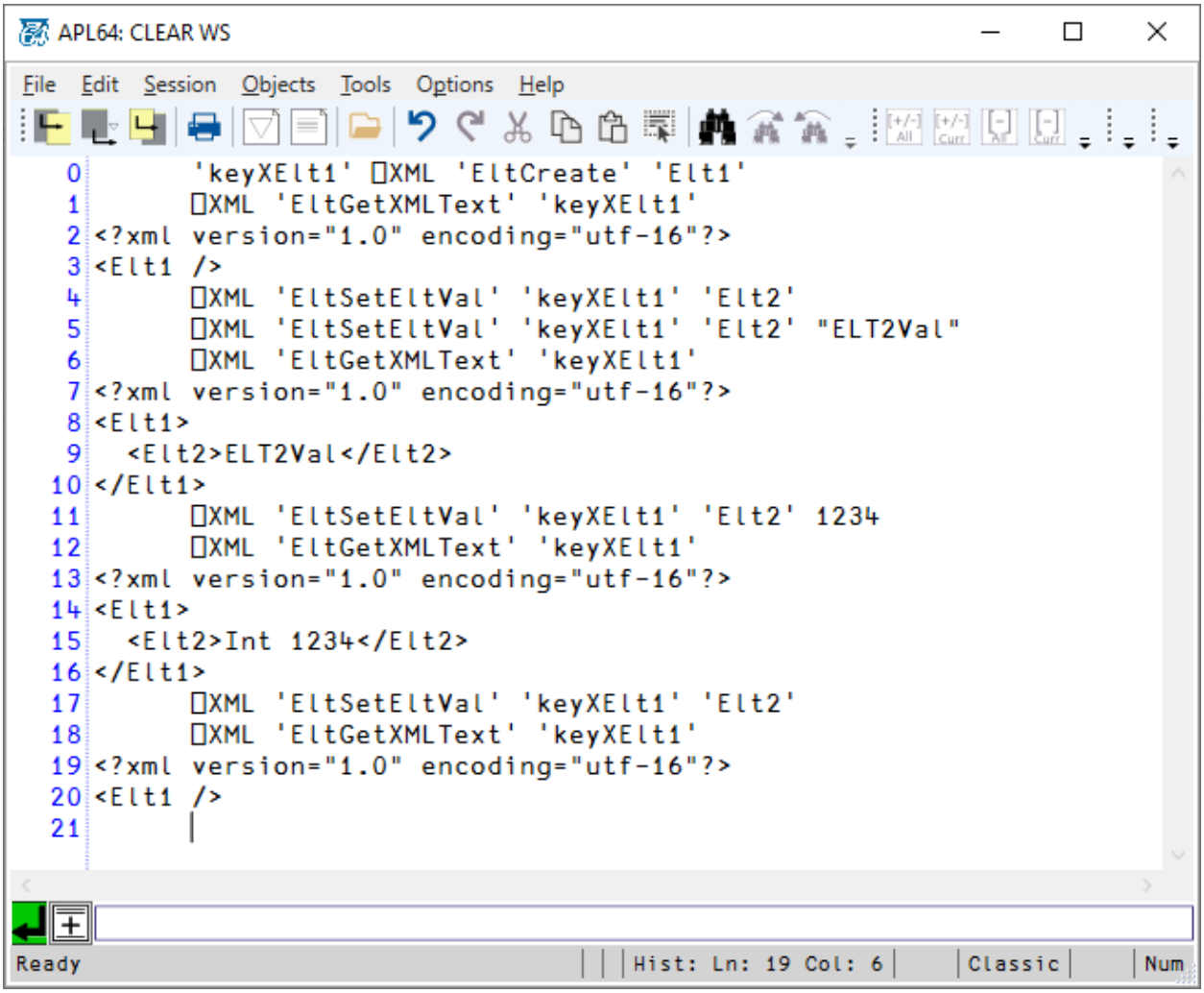
If the child XElement does not exist in the target XElement, it will be created.

If the childEltVal argument is not present, the child XElement, if it exists in the source XElement, will be removed.

keyTgtXElt is the APL programmer-provided key in the □XML XObject dictionary that is associated with the target XElement.

childEltName is the name of the child XElement in the target XElement whose value will be set.

childEltVal is the APL programmer-provided value to be set as the value of the child XElement. This value can be an APL scalar character, string, double, integer or Boolean, a character vector or a □TS-format integer vector assumed to be UTC.



EltSetVal

Syntax: ⎕XML "EltSetVal" keyXElt valXElt

Arguments & Result:

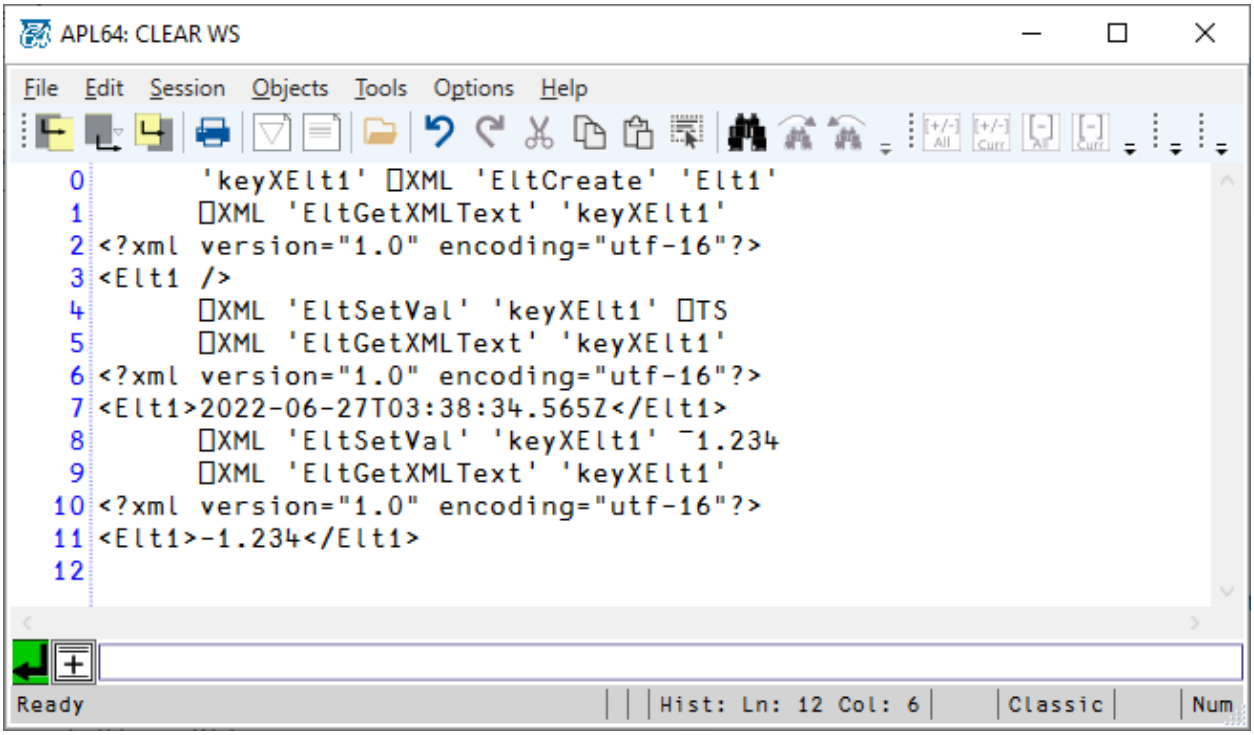
Action Synonyms: EltValSet, SetEltVal

The EltSetVal action will set the value of an existing XElement.

Arguments & Result

keyXElt is the APL programmer-provided key in the ⎕XML XObjects dictionary which is associated with the XElement whose value will be set by the EltSetVal action.

valXElt is the value to be set. This value can be an APL scalar character, string, double, integer or Boolean, a character vector or a ⎕TS-format integer vector assumed to be UTC.



GetXmlText

Syntax: xAttXml(string)←⎕XML "GetAttXmlText" keyXAtt

Arguments & Result:

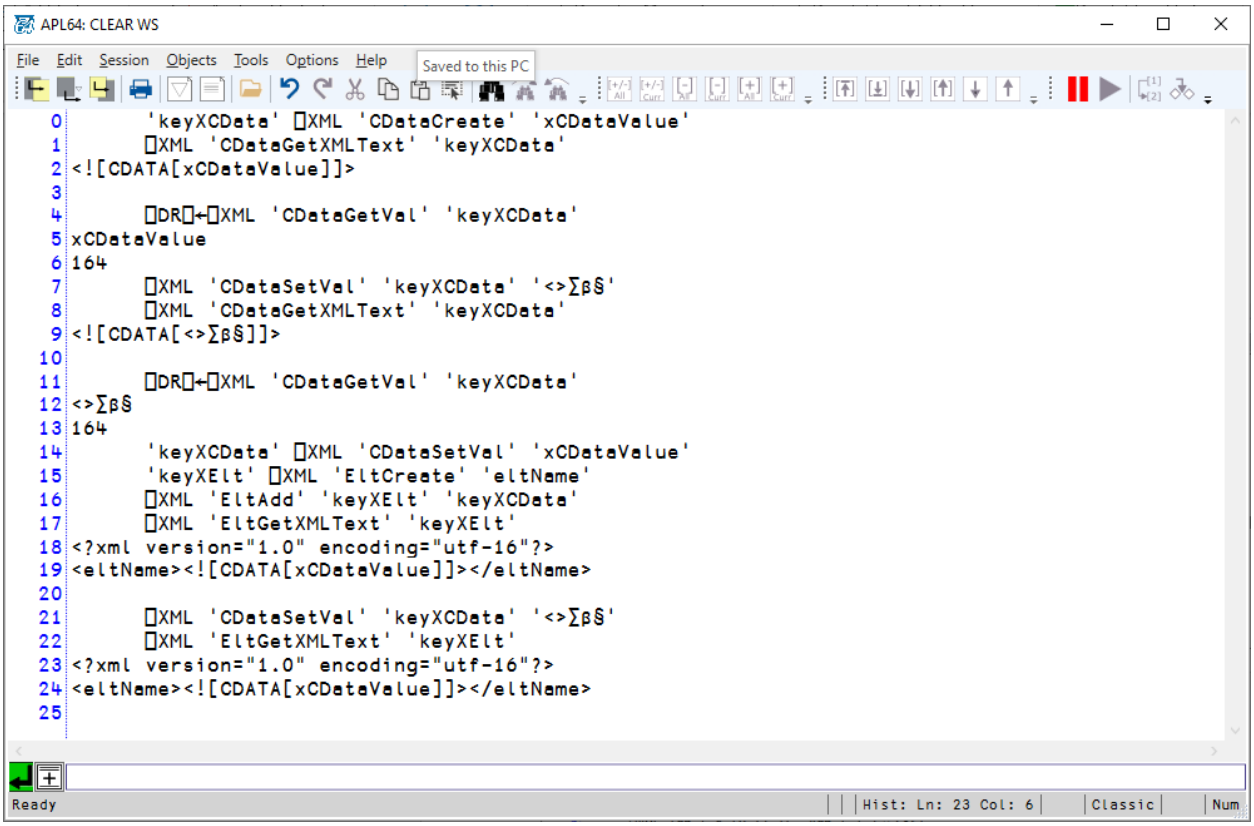
Action Synonyms: AttXmlTextGet, AttGetXmlText

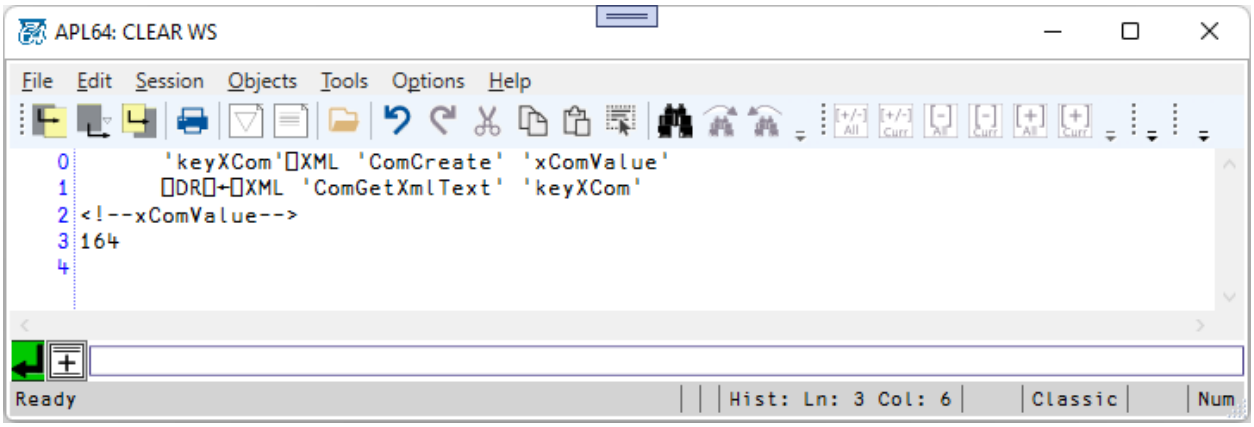
keyXAtt is the APL programmer-provided key in the ⎕XML XObject Dictionary associated with the existing XAttribute.

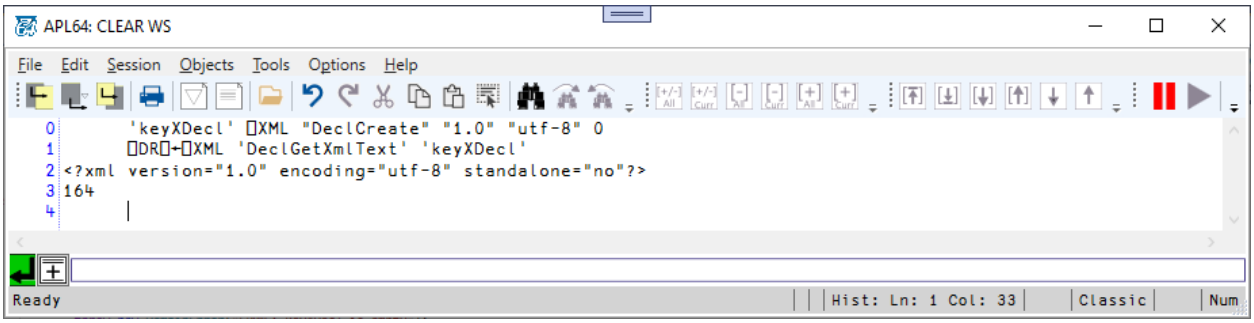
xAttXml is the text string representation of the existing XAttribute associated with keyXAtt.

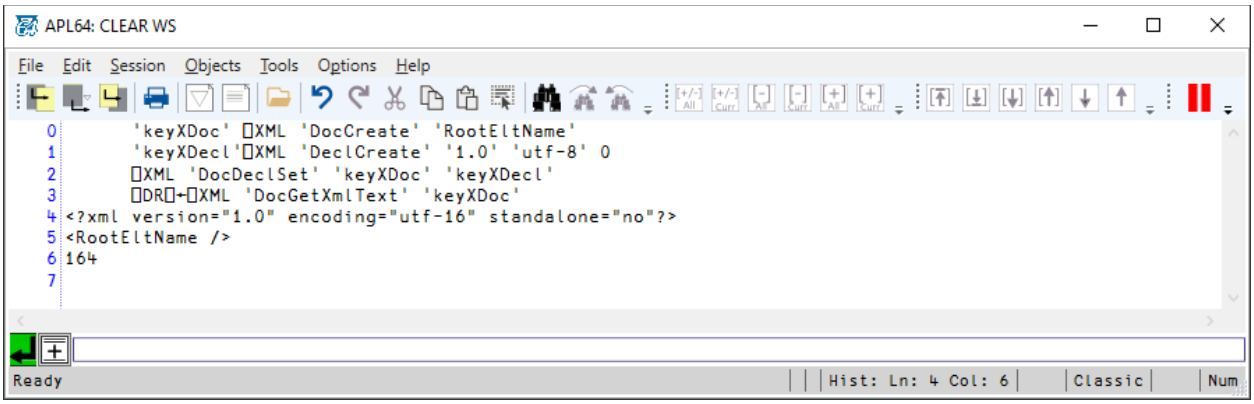
If the specified XAttribute is null, an empty string will be returned.

Synonyms for the GetXmlText action are: GETDOCXMLTEXT, DOCGETXMLTEXT, DOCXMLTEXTGET, GETELTXMLTEXT, ELTGETXMLTEXT, ELTXMLTEXTGET, GETATTXMLTEXT, ATTGETXMLTEXT, ATTXMLTEXTGET, GETCDATAXMLTEXT, CDATAGETXMLTEXT, CDATAXMLTEXTGET, GETCOMXMLTEXT, COMGETXMLTEXT, COMXMLTEXTGET, GETDECLXMLTEXT, DECLGETXMLTEXT, DECLXMLTEXTGET, GETNSXMLTEXT, NSGETXMLTEXT, NSXMLTEXTGET

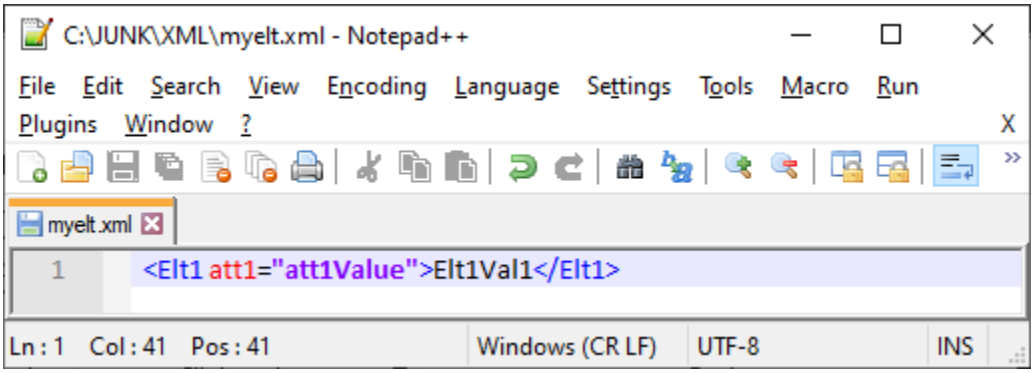


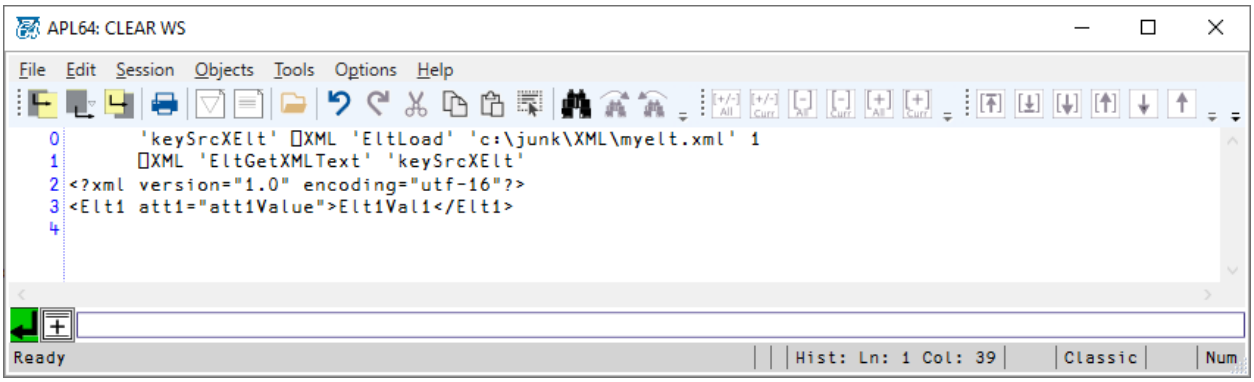


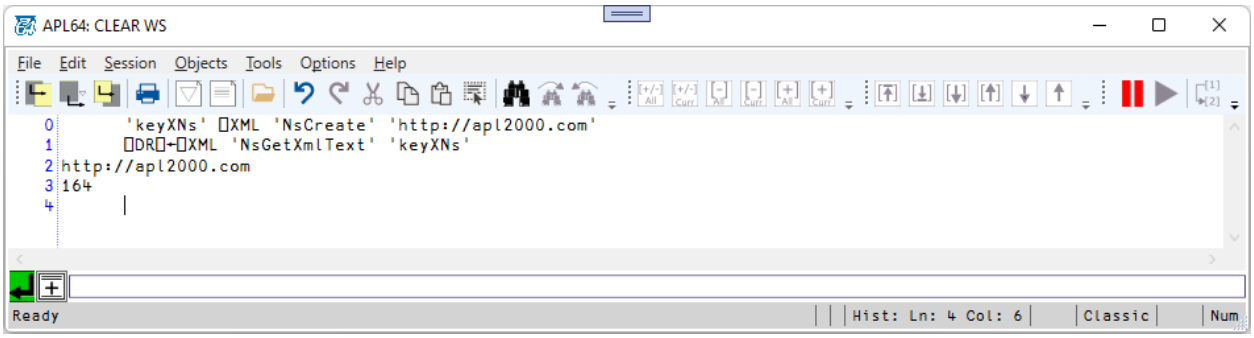




Content of the myelt.xml file:







NsCreate

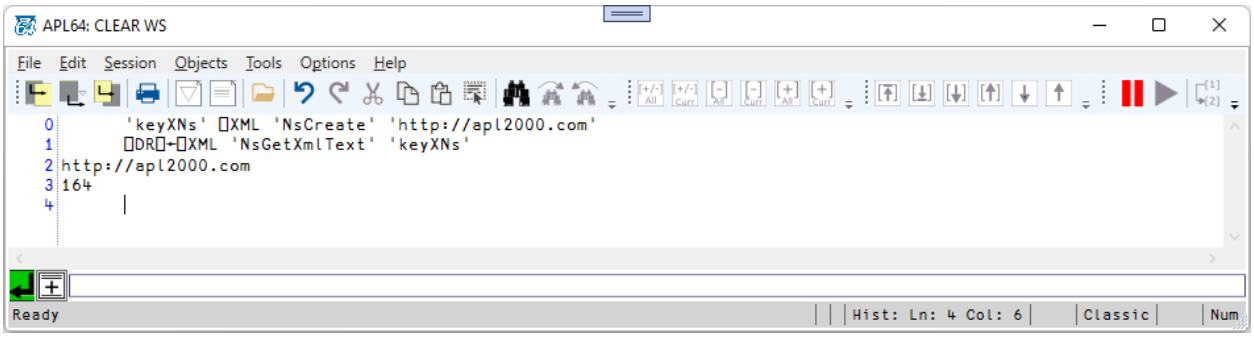
Syntax: xkeyXNs ⎕XML 'NsCreate' nsText

Arguments & Result:

Action Synonym: CreateNs

xKeyNs is the APL programmer-provided key in the ⎕XML XObjects dictionary which will be associated with the new XNamespace object.

nsText is the desired text of the new XNamespace object.



XPathSelectElt

Syntax: keyXElt ⎕XML "XPathSelectElt" keySrcXElt xPathText

Arguments & Result:

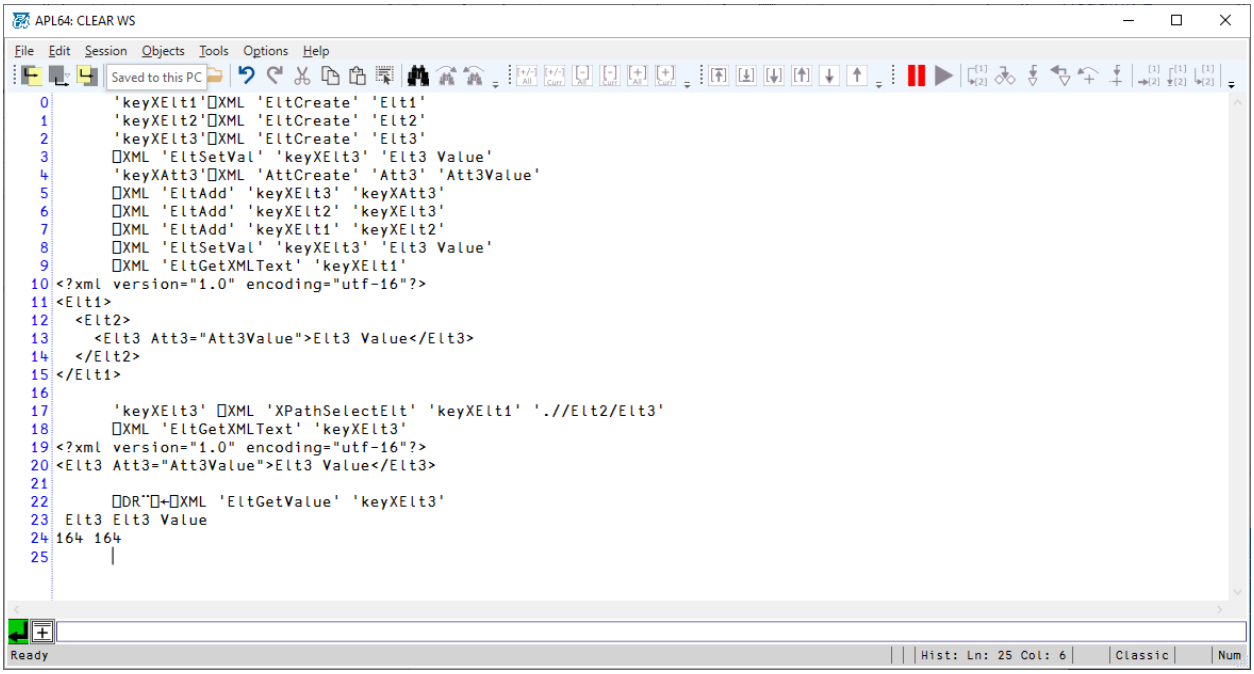
keySrcXElt is the APL programmer-provided key in the ⎕XML XObjects dictionary associated with the existing source XElement.

xPathText is a .Net LINQ XPath scalar string, character or character vector.

keyXElt is the APL programmer-provided key in the ⎕XML XObjects dictionary which will be associated with the XElt result, if any.

An exception will be thrown if the XPath returns no XElement.

Click for .Net XPath information



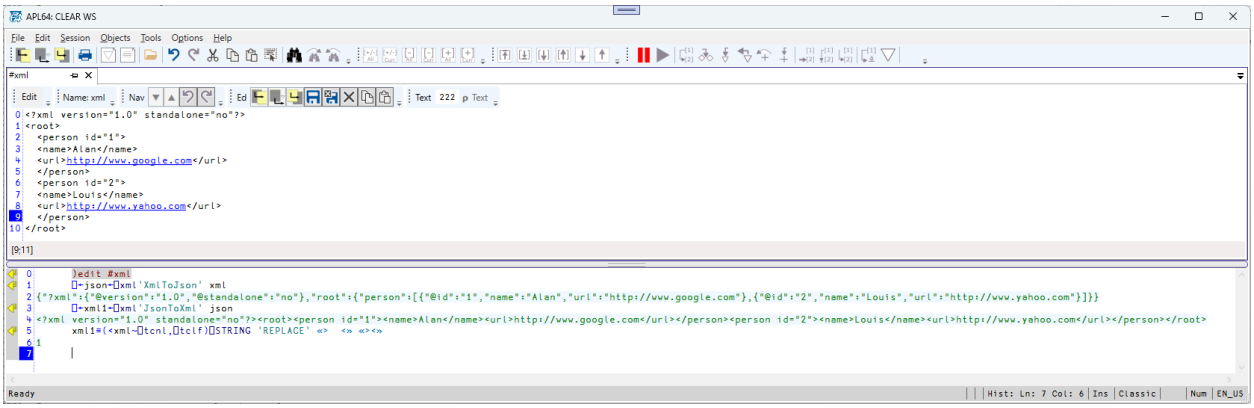
XmlToJson

Syntax: jsonString ← ⎕XML "XmlToJson" xmlText

Arguments and Result:

xmlText is a scalar string, character scalar or character vector in xml-format containing data

jsonString is a string containing the same data in json-format



JsonToXml

Syntax: xmlString ← ⎕XML "JsonToXml" jsonText

Arguments and Result:

jsonlText is a scalar string, character scalar or character vector in json-format containing data

xmlString is a string containing the same data in xml-format without white space.