

New Features in the Janus SOAP XML API

Dave Evans



Agenda

- Doc: Method index
 - Not Janus SOAP [XML] related, but... be sure to see Sirius Mods Command and Parameter Reference Manual (by itself or part of master index)
- Sirius Mods 6.9 & 7.0/7.1 (this summer?)
- Steal some time to finish Sunday's intro
 - Null URI for SelectionNamespace
 - (start demo: `_TEST*` procs)
- Other new features (demo)
 - XPathNodeID
 - InvalidCharacterPosition
 - ExclCanonical
- Validate so new it's ...



Agenda *(continued)*

- Common mistakes
 - Add... vs. LoadXml
 - XPath & Namespaces
 - Speaking of XPath: be careful of //





Doc: Xml API

Index (*excerpts*)

Xml API method prototypes

AddElement: [%nod =] nr:AddElement(name, [URI]) ... 42

AddNamespace: Call nod:AddNamespace(prefix, URI) ... 46

Count: %count = nlis:Count ... 62



Sirius Software,
Inc.

Steal some time to finish Sunday...

- Methods include parsing, Element & Attribute selection, updating, and generating serial form
- Three Xml* classes define the Xml methods
- XmlDocument - contains an XML document
- XmlNode - points to a node in an XmlDocument
- XmlNodeList - set of pointers to nodes in an XmlDocument
- Many methods (e.g., Print) operate starting from a node; defined for both XmlDocument (start from Root) and XmlNode (start from that node); in manual: nr
- XPath used to locate node(s) in an XmlDocument: pervasive



Text To: create "static" document



- "Hand-coded" XmlDocument: very useful, at least to play/test

```

%sl Object Stringlist
%sl = New
%d Object XmlDocument
%d = New
Text to %sl
  <msg>
    <hello>World!</hello>
    <goodNight>And good luck.</goodNight>
  </msg>
End Text
%d:LoadXml(%sl)
%d:Print

```



Sirius Software,
Inc.

Recap: Two important Janus SOAP Xml API retrieval methods

```
%nod = nr:SelectSingleNode([XPath])
```

```
* For debugging (see serialization ...):
```

```
nr:Print([XPath], [options])
```

```
* also Audit and Trace
```



Sirius Software,
Inc.

Janus SOAP Xml API "node info" methods

- Methods to get information "at" single node

```
%str = nr:Value([XPath])
```

```
%str = nr:ValueDefault([XPath],  
                        [default])
```

```
%name = nr:QName([XPath])
```

```
* LocalName, Prefix, URI, DefaultURI
```

```
Print %x:Type([XPath]):ToString
```

```
%strLis = nr:XPathNodeID([XPath])
```



Sirius Software,
Inc.

Send (serialize) and receive (deserialize) XML document requests/responses

```
[%pos =] doc:WebReceive([formFldNam], -  
                        [formFldOcc], [options])  
doc:WebSend([options])
```

```
* Note: In HttpRequest class, not Xml*:  
httpReq:AddXml(doc, [options])  
[%errPos =] httpResp:ParseXml(doc,  
                               [options])
```



Send (serialize) and receive (deserialize) XML document requests/responses (*continued*)

```
* [%errPos =] nr:LoadXml(str | strList,  
                        [options])
```

* **Serial & XML** among "less commonly" ...



Sirius Software,
Inc.

AddSubtree DefaultURI= arg etc.

* Were presented in SUG 06, had not shipped:

```
[%nod =] nr:AddSubtree(sourceSubtreeNod,  
                        [DefaultURI=string])
```

```
%uri = nr:DefaultURI([XPath])
```

```
%uri = nr:PrefixURI(prefix, [XPath])
```



Sirius Software,
Inc.

Namespaces and XPath



- Prefixes in XPath expression are completely independent of those in an XML document
- XPath does not have any notion of default namespace

* ("Optional") save URI of prefix:
`%currURI = doc:SelectionNamespace(prefix)`

* Associate prefix with URI for XPath:
`%doc:SelectionNamespace(prefix) = newURI`



Sirius Software,
Inc.

SelectionNamespace property examples

```
%doc Object XmlDoc Auto New
%sl Object Stringlist Auto New
Text To %sl
<x:outer xmlns:x="urn:Toto" xmlns="urn:Kansas">
  <inner x:attrib="contrived example"/>
</x:outer>
End Text
%doc:LoadXml(%sl)
%doc:SelectionNamespace('myx', 'urn:Toto')
%doc:SelectionNamespace('d', 'urn:Kansas')
Print %doc:QName('myx:outer')
Print %doc:Serial('* /d:inner')
Print %doc:Value('* /* /@myx:attrib')
```

- * Prefix in XPath expression has nothing to do with
- * prefixes in document



Sirius Software,
Inc.

SelectionNamespace XPath extension

```
* (start demo: _TEST* procs)
* XPath standard requires prefix bound to (non-null) URI; in 7.0 mods
* null URI = "select name in no namespace"; better parameterization
%doc1:LoadXml('<a/>')
Call prNamElt(%doc1, 'a', '')
%doc2:LoadXml('<a xmlns="urn:foo"/>')
Call prNamElt(%doc2, 'a', 'urn:foo')
*-----*
```



```
Subroutine prNamElt(%d Object XmlDoc, %lcl Longstring, %u Longstring)
%d:SelectionNamespace('junk') = %u
**** Speaking of XPath ****
* "/" should usually be avoided - where called for, following
* case is not so bad: one step with just element name:
%d:Print('//junk:' With %lcl)
```



Sirius Software,
Inc.



XPathNodeID

* Provide "feedback" about location in XML document

```
%sl = nr:XPathNodeID([selection_XPath])
```

```
%sl(1): Result XPath expression
```

```
%sl(2): Prefix used in XPath
```

```
%sl(3): URI for prefix
```

```
%sl(4/5) etc.
```



Sirius Software,
Inc.



InvalidCharacterPosition

* Only shared method (so also can use XmlDocument obj):
%pos = %(XmlDoc):InvalidCharacterPosition(string)



Sirius Software,
Inc.



ExclCanonical option of Serial

- * There are many equivalent serializations of
- * any given XmlDocument

`%utfOrEbc = %nr:([XPath], ExclCanonical)`





Next and Previous methods

- * Works for all nodes, but needed to navigate
- * attributes, which following-sibling and
- * preceding-sibling don't handle:

```
%nod = nod:Next
```

```
%nod = nod:Previous
```





Validate so new it's ...

```
%rc = nr:Validate(schemaInput [, options])
```

- * schemaInput is string or Stringlist containing the
- * serialized XML Schema document

- * One little thing to note:

- * 'C' option of Regex methods/functions can be used

- * to check operation of pattern as defined in XML Schema

```
%sl Object Stringlist Auto New
```



Sirius Software,
Inc.

Validate so new it's ... *(continued)*

```

%d:LoadXml ( '<x><t>.234</t></x>' )
Text To %s1
  <xs:schema
    xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="t" type="xs:decimal"/>
  </xs:schema>
End Text
%d:Print
%n = %d:SelectSingleNode('* /t')
%rc = %n:Validate(%s1, 'ErrRet Mod')
Print 'Validate rc=' %rc ' - XmlDocument after Validate:'
%d:Print
%n>DeleteSubtree
%rc = %n:Validate(%s1, 'Keep ErrRet Mod')

```



Validate so new it's ... (*continued*)

```
<x>  
  <t>.234</t>  
</x>
```

Validate rc=0 - XmlDocument after Validate:

```
<x>  
  <t>0.234</t>  
</x>
```

```
*** 1 CANCELLING REQUEST: MSIR.0750: Class XMLNODE,  
function VALIDATE: reference to null object in line 20,
```



Sirius Software,
Inc.