

# Janus XML API New Features

**SUG 06: Dave Evans**  
**Tuesday 15:45**



# Sirius Mods 6.8 (i.e., available now)

- InvalidChar = Disallow | Allow property of XmlDocument
  - Support XmlDocs as general data container
  - Some LDAP servers seem to return binary data
- IsValidString shared function
  - Already better idea: InvalidCharPosition; 6.9?
- LoadXml accepts Stringlist arg
  - So now LoadFromStringlist obsolete
- LoadXml now also in XmlNode class
  - Able to parse "subdocument"
    - ▶ Previously could parse into another XmlDocument & copy with AddSubtree
  - Also can parse "fragment": **time to <b>quit</b>**



Sirius Software, Inc.

# Build WSDL document all in one XmlDocument

```
%nod = %doc:AddElement('definitions', , -  
'http://wsdl')  
%nod:AddElement('types')  
* etc  
%doc:Print  
Print 'types is in namespace:' And %doc:URI('/*/*')
```

Result is:

```
<definitions xmlns="http://wsdl">  
  <types/>  
</definitions>  
types is in namespace: http://wsdl
```



# Enhance AddSubtree, document building workhorse

\* Build WSDL document in two XmlDocument; AddSubtree merges:

```
%nod1 = %doc1:AddElement('definitions', , 'http://wsdl')
%nod2 = %doc2:AddElement('types')
%nod1:AddSubtree(%nod2)
%doc1:Print
Print 'types is in namespace:' And %doc1:URI('/*//*')
```

Result is not what we want:

```
<definitions xmlns="http://wsdl">
  <types xmlns="" />
</definitions>
types is in namespace:
```



# AddSubtree option to specify URI of default namespace for inserted subtree

```
%nod1 = %doc1:AddElement('definitions', , -  
'http://wsdl')  
%nod2 = %doc2:AddElement('types')  
%nod1:AddSubtree(%nod2, DefaultURI=%nod1:URI)  
%doc1:Print  
Print 'types is in namespace:' And %doc1:URI('/**/*')
```

Result is as in first example:

```
<definitions xmlns="http://wsdl">  
  <types/>  
</definitions>  
types is in namespace: http://wsdl
```



# Also DefaultURI *method* to get URI of default namespace at a node

```
* Unrealistic example? 1st, use single XmlDocument:
%nod1 = %doc1:AddElement('o:outer', , 'http://outer')
%nod1:AddNamespace('', 'http://default')
%nod1:AddElement('types')
%doc1:Print
Print 'types is in namespace:' And %doc1:URI('/**/*')
```

Result is:

```
<o:outer xmlns:o="http://outer" xmlns="http://default">
  <types/>
</o:outer>
types is in namespace: http://default
```



# Show it using AddSubtree, 6.8 Sirius Mods

```
%nod1 = %doc1:AddElement('o:outer', , 'http://outer')
%nod1:AddNamespace('', 'http://default')
%nod2 = %doc2:AddElement('types')
%nod1:AddSubtree(%nod2)
%doc1:Print
Print 'types is in namespace:' And %doc1:URI('/*/*')
```

Result is (not what we'd like):

```
<o:outer xmlns:o="http://outer" xmlns="http://default">
  <types xmlns="" />
</o:outer>
types is in namespace:
```



# Now use new **DefaultURI= *argument*** and **DefaultURI *method*** to get URI of default namespace at "target"

```
%nod1 = %doc1:AddElement('o:outer', , 'http://outer')
%nod1:AddNamespace('', 'http://default')
%nod2 = %doc2:AddElement('types')
%nod1:AddSubtree(%nod2, DefaultUri = %nod1:DefaultUri)
%doc1:Print
Print 'types is in namespace:' And %doc1:URI('/*/*')
```

Result is (this is what we want):

```
<o:outer xmlns:o="http://outer" xmlns="http://default">
  <types/>
</o:outer>
types is in namespace: http://default
```



Sirius Software, Inc.

## Last AddSubtree notes (6.9)

- DefaultURI= may create needless declaration
- Any declaration *within* source subtree is copied
- OK from Namespace=None XmlDocument to Namespace=Ignore XmlDocument
- OK from Namespace=None XmlDocument to Namespace=On XmlDocument *if* DefaultURI= arg used
- Note MSXML (and probably .Net) normal subtree copy acts like DefaultURI=%target:DefaultURI
  - Maybe inherit prefix declarations too



## Other possible 6.9 features

- InvalidCharacterPosition: better than IsValidString
- ToXml for Object
- Generate XPath expression to identify node
- Again improve AddSubtree namespace handling
- Support more of XPath:
  - not() function in predicates
  - Allow predicate to contain another predicate
  - Support parentheses in predicate
- Schema?

