YANG Next Analysis

NETMOD WG
IETF 104 (Prague)
Meetings

3 Virtual Interims in two months:
- Feb 6
- Feb 20
- Mar 20

Focus on scoring the YANG Next issues
- 70 issues entered over the course of 3 years
  - repo created on Mar 11, 2016

This presentation reviews the results of those meetings.

Deep dive meeting on Wednesday, Mar 27, 15:00-17:00 in Karlin 3 (seats 60)
3-D Views

Issues: 48 Open  (22 Closed, not displayed)
3-D Views Showing Complexity

**Complexity:** Mostly low, with fair amounts medium and unknown

Most of the important issues have low complexity.

Most of the backwards compatible issues have low complexity.
Most issues are highly backwards compatible.
Backwards Compatibility x Importance
(with Unknowns distributed equally into other values)

Effect of Unknowns is approximated...
Backwards Compatibility x Importance
(with Complexity factored in as well)

Most time spent on the desirable green quadrant.
Where to Focus?

(This is what Wednesday's meeting is about)

Focus on **Importance**

Focus on **Backwards Compatibility**
Comments?
# The 8 Issues Behind a Possible 2.0

<table>
<thead>
<tr>
<th>Importance</th>
<th>Backwards Compatibility</th>
<th>Backwards Compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>High</td>
<td>Low</td>
<td>Unknown</td>
</tr>
<tr>
<td>1.</td>
<td>Context-independent encoding of instance-identifiers and identityrefs</td>
<td>3. Introduce critical extensions</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4. Refine YANG versioning</td>
</tr>
<tr>
<td>Medium</td>
<td>2. Consider removing support for sub modules from YANG</td>
<td>5. Allow some references to from config-true to config-false</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6. Add an “inactive” metadata annotation</td>
</tr>
<tr>
<td>Unknown</td>
<td>N/A</td>
<td>7. Introduce critical annotations</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8. Clarify 'deviation' substatements to match ABNF grammar</td>
</tr>
</tbody>
</table>
The 8 Issues with Importance == Low
(Unlikely to be supported under any circumstance)

1. Add if-feature on "must" statement
2. Introduce XPath function datastore()
3. Create a way for a statement to tie-in with augment/deviation
4. add 'conformance-type' leaf to 'import' statement
5. Restrict usage to a subset of XPATH
6. Restrict regex to a subset of XML regex specification
7. Replace 'encoding' with 'representation'?
8. Default to namespace urn:yang:<module-name> ?