Silicon Release

OpenDayLight Silicon Release: Inputs to Marketing Communications

Overall

Key	Summary	T	Updated	Status	Resolution
TSC-285	Netvirt Silicon Release Plan	📝	Sept 22, 2020	OPEN	Unresolved
TSC-286	DAEXIM Silicon Release Plan	📝	Sept 21, 2020	OPEN	Unresolved
TSC-287	JSONRPC Silicon Release Plan	📝	Sept 21, 2020	OPEN	Unresolved
TSC-288	Ovsdb Silicon Release Plan	📝	Oct 04, 2020	OPEN	Unresolved
TSC-292	Distribution Silicon Release Plan	📝	Oct 02, 2020	OPEN	Unresolved
TSC-293	TransportPCE Silicon release plan	📝	Mar 25, 2021	CLOSED	Unresolved
TSC-294	ODL-Guice Silicon Plan	📝	Oct 06, 2020	OPEN	Unresolved
TSC-295	ODL-Micro Silicon Plan	📝	Oct 11, 2020	OPEN	Unresolved
TSC-297	AAA Silicon Release Plan	📝	Oct 22, 2020	OPEN	Unresolved
TSC-298	BGPCEP Silicon Plan	📝	Oct 22, 2020	OPEN	Unresolved

10 issues

BGPCEP

Status	Key	Summary	Description
RESOLVED	BGPCEP-932	Remove use of SchemaPath from config.loader.spi.ConfigFileProcessor	SchemaPath is going to be deprecated. Replace it with SchemaNodeIdentifier.Absolute, which seems to be the appropriate thing.
RESOLVED	BGPCEP-905	Remove use of blueprint in config-loader-impl	config-loader-impl does not use anything blueprint-specific, in fact it actively battles blueprint by using odl:static-reference. Reformulate it in terms of OSGi DS, binding it to proper codec services, so that we can discover the associated EffectiveModelContext. Also clean up the FileWatcher lifecycle, as it seems it should not really be a separate component.
RESOLVED	BGPCEP-923	Remove blueprint from rsvp-spi	rsvp-spi is using blueprint to wire itself. Convert it to OSGi DS/inject annotations instead. This might be problematic, as it is using the ODL BP extension.
RESOLVED	BGPCEP-924	Remove blueprint from rsvp-impl	rsvp-impl is using blueprint to wire itself. Convert it to OSGi DS/inject annotations instead.
RESOLVED	BGPCEP-926	Remove blueprint from bmp-monitors-config-loader	bmp-monitors-config-loader is using blueprint to wire itself. Convert it to OSGi DS/inject annotations instead.
RESOLVED	BGPCEP-927	Remove blueprint from protocols-config-loader	protocols-config-loader is using blueprint to wire itself. Convert it to OSGi DS/inject annotations instead.
RESOLVED	BGPCEP-928	Remove blueprint from routing-policy-config-loader	routing-policy-config-loader is using blueprint to wire itself. Convert it to OSGi DS/inject annotations instead.

RESOLVED	BGPCEP-929	Remove blueprint from topology-config-loader	topology-config-loader is using blueprint to wire itself. Convert it to OSGi DS/inject annotations instead.
RESOLVED	BGPCEP-930	Remove blueprint from data-change-counter	data-change-counter is using blueprint to wire itself. Convert it to OSGi DS/inject annotations instead.
RESOLVED	BGPCEP-935	Remove blueprint from pcep-base-parser	This is is simplistic service, convert it to OSGi DS
RESOLVED	BGPCEP-936	Remove blueprint from bgp-parser-impl	This has a very small component, convert it to OSGi DS.
RESOLVED	BGPCEP-937	Convert ServiceLoader definitions to @MetaInfServices	We are using ServiceLoader injection, having hand-written META-INF/services entries. This makes is hard to reason about them. Generate these entries through @MetaInfServices, so that we have a clear code-visible control of what services we provide.
RESOLVED	BGPCEP-938	Remove blueprint from bgp-openconfig-rp-statement	This component is rather simple, clean it up and convert it to OSGi DS.
RESOLVED	BGPCEP-939	Remove blueprint from bgp-openconfig-rp-spi	The blueprint here seems to be involved, but is in reality a simple whiteboard. Let's convert it to an OSGi DS component instead. This shows we could actually directly retain registrations, eliminating the need for an abstract class.
RESOLVED	BGPCEP-940	Remove blueprint from bgp-route-target	This component is really simple, convert it to OSGi DS.
RESOLVED	BGPCEP-942	Remove blueprint from bgp-openconfig-spi	bgp-openconfig-spi wiring is using a combination of Blueprint and ServiceLoader, while performing a rather straightforward whiteboard activation. Refactor the classes to provide properly-injected immutable environment, so that table types work seamlessly in any environment. While we are refactoring, also remove use of Optional from BGPTableTypeRegistryConsumer, as users benefit from nullable.
RESOLVED	BGPCEP-943	Remove blueprint from bgp-openconfig-rp-impl	With bgp-openconfig-rp-spi refactored and properly working we can now convert bgp-openconfig-rp-impl, which ends up being a trivial conversion.
RESOLVED	BGPCEP-944	Remove blueprint from bgp-rib-spi	bgp-rib-spi ends up being a very simple whiteboard, convert it to OSGi DS instead of blueprint.
RESOLVED	BGPCEP-933	Switch config loaders to DOM API	All four config loaders end up translating the parsed structure to Binding form only to feed it into DataBroker. This is completely unnecessary, as the data is coming from XML files and is therefore in DOM form – hence we end up encoding it to and from Binding for no particularly good reason. We are also using inefficient datastore operations in individual implementations, which really should be using the bare minimum required. Switching to DOM interfaces is a bit of juggling of paths, but will improve performance and most notably will make config loaders independent of codec services.
RESOLVED	BGPCEP-853	Move RIB to be independent	EffectiveRibInWriter and LocRibWriter should be binding-independent. This means they need to interface with DOMDataBroker only and handle attribute encoding when interfacing with import/export policies.
RESOLVED	BGPCEP-861	Audit and fix use of read-only transactions in openconfig-rp-statement	Cursory look during migration to MD-SAL APIs reveals this component is not properly closing read-only transactions. Audit all callers and make sure transactions are used in a proper try-with-resources block. While we are in the area, it seems the results of the call could use Optional.map() handling rather than the explicit isPresent()/get() combo.
RESOLVED	BGPCEP-906	Propagate only negotiated afi/safi routes to adj-rib-out	When BGP session comes, empty entries for negotiated afi/safi are created in peer's adj-rib-out. If routes belonging to afi/safi that were not negotiated for the peer are present in loc-rib, those entries are being propagated to the peer, resulting in ModifiedNodeDoesNotExistException. Patch prevents this by allowing routes for only the negotiated afi/safi to be propagated from loc-rib to peer's adj-rib-out. If routes are written to non-existent afi/safi table in adj-rib-out, it results in transaction chain failure which leads to a scenario where multiple threads block to get a lock on BGPPeer but the thread handling transaction failure is waiting for the submitted futures to exit. Patch fixes this scenario by preventing wait on the submtited futures when handling transaction failures.

22 issues

Controller

Status	Assignee	Summary	Description
RESOLVED	Tadey Bilan	Update cluster documentation re. multi-dc deployment	Akka 2.6 brings about a great deal of change we are pulling in. Most notably there is no auto-downing, but rather SBR or something else coming in.
RESOLVED	Tadey Bilan	Upgrade Akka to 2.6.x	Now that we have control over Akka packaging, we can safely upgrade to 2.6.x release train.

2 issues

NETCONF

Status	Assignee	Summary	Description
RESOLVED	Robert Varga	Deprecate messagebus-netconf	messagebus-netconf is a proof-of-concept component, which is not used by anyone and has some serious deficiencies. Deprecate it for removal.
RESOLVED	Iaroslav Kholiavko	Support for Root Resource Discovery is missing	Both RFC8040 [0] and draft 18 [1] define Root Resource Discovery mechanism. Neither Draft18 in Nitrogen snapshots, not rfc8040 in oxygen snapshots seem to have that implemented.

```
$ curl -v -u admin:admin -H "Accept:application/xrd+xml" http://127.0.0.1:8181/.well-known/host-meta

*   Trying 127.0.0.1...
* TCP_NODELAY set
* Connected to 127.0.0.1 (127.0.0.1) port 8181 (#0)
* Server auth using Basic with user 'admin'
> GET /.well-known/host-meta HTTP/1.1
> Host: 127.0.0.1:8181
> Authorization: Basic YWRtaW46YWRtaW4=
> User-Agent: curl/7.52.1
> Accept:application/xrd+xml
>
< HTTP/1.1 404 Not Found
< Content-Type: text/html; charset=ISO-8859-1
< Cache-Control: must-revalidate,no-cache,no-store
< Content-Length: 305
<
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
<title>Error 404 Not Found</title>
</head>
<body><h2>HTTP ERROR 404</h2>
<p>Problem accessing /.well-known/host-meta. Reason:
<pre>    Not Found</pre></p><hr><i><small>Powered by Jetty://</small></i><hr/>

</body>
</html>
* Curl_http_done: called premature == 0
* Connection #0 to host 127.0.0.1 left intact
```

[0] https://tools.ietf.org/html/rfc8040#section-3.1
[1] https://tools.ietf.org/html/draft-ietf-netconf-restconf-18#section-3.1

RESOLVED	Oleksii Mozghovyi	Update call home to RFC8071	http://tools.ietf.org/html/draft-ietf-netconf-reverse-ssh is a very useful feature for controller deployments, where we do not need to pre-configure the device locations. Implement it such that a sal-netconf-connector instance can be configured to accept Call Home connections.
RESOLVED	Oleksii Mozghovyi	Get rid of GzipFilter	GzipFilter is deprecated and the functionality would much better we served through registering a GzipHandler instance with the HTTP service. Examine how we can do this in a seamless way, so that we do not have a hard dependency on jetty-servlets.
RESOLVED	Nikhil Soni	Extend Websocket streams for data-less notifications	In one of our customers, the use case was to listen to changes in a particular yangtree. however the data underneath was quite huge, thereby causing the heap running over and the controller crashing in response to lot of changes in the subtree. Hence, we had to fallback to a mechanism to detect just a change in the yangtree from a northbound application and on detection of the change, do a bulk get of the data. that was more effective.

In supporting this usecase, the current implementation needed an extension. Currently, WebSocket stream notification contains notification metadata with data. However, there is no way to avoid collecting data during notification processing in the websocket server. Hence, we decided to add a new filter named "odl-skip-notification-data", similar to "odl-leaf-nodes-only" in the subscription API.

1. Create Stream:

```
POST http://\controller-ip:8181/restconf/operations/sal-remote:create-data-change-event-subscription
{
  "input": {
    "path": "/network-topology:network-topology",
    "sal-remote-augment:datastore": "CONFIGURATION",
    "sal-remote-augment:scope": "SUBTREE",
    "sal-remote-augment:notification-output-type": "JSON"
  }
}

Response:
{
  "output": {
    "stream-name": "data-change-event-subscription/network-topology:
network-topology/datastore=CONFIGURATION/scope=SUBTREE"
  }
}
```

2. Subscribe:

```
GET http://{{controller-ip}}:8181/restconf/streams/stream/data-change-event-subscription/network-topology:network-topology
/datastore=CONFIGURATION/scope=SUBTREE?&odl-skip-notification-data=true

Response:
{
    "location": "ws://localhost:8185/data-change-event-subscription
/network-topology:network-topology/datastore=CONFIGURATION/scope=SUBTREE"
}
```

3. Get Notification:

```
GET  ws://localhost:8185/data-change-event-subscription/network-topology:
network-topology/datastore=CONFIGURATION/scope=SUBTREE
```

4. Write to topology or any respective data tree:

5. Websocket JSON Response -

```
{
   "notification":{
      "xmlns":"urn:ietf:params:xml:ns:netconf:notification:1.0",
      "data-changed-notification":{
         "xmlns":"urn:opendaylight:params:xml:ns:yang:controller:md:sal:
remote",
         "data-change-event":{
            "path":"/network-topology:network-topology",
            "operation":"updated"
         }
      },
      "eventTime":"2020-05-05T20:22:23.293+05:30"
   }
}
```

RESOLVED	Vladyslav Marchenko	Rework KeepaliveSalFacade time tracking	Current implementation ends up rescheduling timer tasks all over the place. This should be reworked to keep only a single task and maintain the time of last activity.
RESOLVED	Robert Varga	Create NetconfTopologyRPCProvider using OSGi DS	NetconfTopologyRPCProvider are being created and they are very simple registrations. Convert them to a separate component provided from netconf-topology automatically.
RESOLVED	Iaroslav Kholiavko	Upgrade sshd to 2.6.0	Upgrade to sshd-2.6.0, removing workaround introduced in ~~NETCONF-736~~.
RESOLVED	Oleksii Mozghovyi	Unstable test behavior on Windows (NetconfDeviceCommunicatorTest)	There is NetconfDeviceCommunicatorTest.testNetconfDeviceReconnectInCommunicator test that works fine on Unix-type OS but is not stable on Windows environments.
RESOLVED	Iaroslav Kholiavko	Return status code 503 instead of 404 when a mountpoint is not mounted	Currently the returned status code is 404, which callers might understand as a confirmation that the resource is not present in the mounted datastore. But the resource might be in fact present in the datastore, just ODL cannot access the datastore this time. This distinction is important for CSIT to distinguish transient instability (for example in cluster High Availability testing) from really missing data.
RESOLVED	Vladyslav Marchenko	Eliminate blocking checks in loops, so the checks are done in parallel	Do not mix read and write operations in one loop. This change is needed for old restconf (BrokerFacade.java) and also for new restconf (RestconfDataServiceImpl and its utility helper classes).
RESOLVED	Tomas Cere	ListenerAdapter needs to serialize JSON directly	Currently, when producing JSON, ListenerAdapter serialises to XML, and then converts that to JSON using org.json. It should instead serialise directly to JSON using the YANG codecs.

RESOLVED	Jamo Luhrsen	Extra superfluous edit-config RPC sent just prior to correct edit-config RPC for top-level lists	Symptom: POST & PUT to certain top-level lists in a mounted device's model returns a 500 error via Restconf, although in some cases (device-dependent) the config appears to have committed. This problem first appeared in Boron-SR3, and continues in Carbon SR-1. What's actually happening: Two edit-config RPCs appear on the wire / in TRACE logs. The first edit-config's body includes a single empty parent node, such as <configuration> or <interfaces>, but then no actual contents within that. Different vendors' devices handle this in different ways, sometimes with a ~graceful rpc-error-reply on that incorrect edit-config but accepting the correct one that follows. Others enter a half-configured state and then fail the correct edit-config state (such as a POST where the resource was created without useful payload). Solution: If the target is a mounted network element, don't create the empty subtree at all. Regardless of whether or not the standard allows it, most vendor netconf implementations reject this behavior. See MakePut in BrokerFacade for the root cause, and also see corresponding post method.. One example of the failure: 2017-10-05 16:26:40,416 | WARN | oupCloseable-3-2 | NetconfDeviceCommunicator | 299 - org.opendaylight.netconf.sal-netconf-connector - 1.5.1.Carbon | RemoteDevice {dfwm2k02-ie1ri} : Error reply from remote device, request: <rpc message-id="m-7538" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"> <edit-config> <target> <candidate/> </target> <config> <configuration xmlns="http://yang.juniper.net/junos/conf/root"/> </config> </edit-config> </rpc> , response: <nc:rpc-reply xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns:junos="http://xml.juniper.net/junos/17.4X48/junos" message-id="m-7538"> <nc:rpc-error> <nc:error-type>protocol</nc:error-type> <nc:error-tag>operation-failed</nc:error-tag> <nc:error-severity>error</nc:error-severity> <nc:error-message>syntax error, expecting <configuration></nc:error-message> <nc:error-info> <nc:bad-element>configuration</nc:bad-element> </nc:error-info> </nc:rpc-error> </nc:rpc-reply>
RESOLVED	Peter Suna	Non-clustered NETCONF topology leaks schema registrations	This issue is correctly flagged by SpotBugs and has escaped attention for quite some time. When we are creating the context for a device, we add yanglib if it is configured for that particular device. If there are multiple devices pointing to the same url, we end up registering for each of them, and if they point to the same schema directory we end up with quadratic number of registrations. At the very least we should be tearing down the registrations when the device goes away.
VERIFIED	Jamo Luhrsen	regression in netconf-scale CSIT betweeen Mg Sr1 and Sr2	This job appears to have a regression between SR1 and SR2. The job will pass from time to time, but for the most part it will fail in the Getmulti and Getsingle suites which mount 500 instances of the netconf-testtool and then attempt to "issue requests" on each device in order (essentially a GET to yang-ext:mount for each device). When the python tool doing those requests has trouble the test case fails. In the sandbox, I ran a job with the Sr1 release which saw two failures in 35 iterations, whereas using a distro built recently (so Sr2 bits) the results were 8 passes in 36 iterations. Jenkins job history in the web ui only goes back 30 builds or so, but I pulled all the console logs since job 225 (May 1st – just before SR1 was released) until the most recent build #327. Some of those jobs were infra aborts, but it's aprox 90 data points. here is a hacky spreadsheet to illustrate results. There seems to be two points of interest. The job was mostly passing for quite some time, then had 10 straight failures before becoming stable again. That happened at job #265 (June 6th). Finally job #301 failed again (July 11th) and it seems to be unstable since that point on. Nothing in the netconf project stands out to me in the June 6th timeframe, but there was an MRI bump on July 10th Here is a karaf.log from the most recent failed job. There is some ugliness that may be a place to start looking. Search for "Getmulti.Issue_Requests_On_Devices" and scroll down. Seems that one mount session went down, then a lot of NETCONF operation failed Exceptions.

RESOLVED	Vladyslav Marchenko	Yang model file without "revision" statement	According to RFC6020 https://tools.ietf.org/html/rfc6020#section-7.1.9 the "revision" statement is not mandatory in yang model file. But currently if we are connecting device simulator (NetConf TestTool) to ODL and this simulator has yang model without "revision" - we have NoSuchElementException.

RESOLVED	Robert Varga	RFC8040 ignores unknown with-defaults values	RFC8040 states:
			`   If the server does not support the requested value of the`
			`   "with-defaults" parameter, the server MUST return a response with a`
			`   "400 Bad Request" status-line.  The error-tag value "invalid-value"`
			`   is used in this case.`
			our implementation ignores the value, letting the request succeed.

RESOLVED	Robert Varga	SSH session does not close properly after device disconnects	This issue is related to https://issues.apache.org/jira/browse/SSHD-1028. sshd/common/io/nio2/**Nio2Session.handleReadCycleCompletion()** no longer closes the session. The close (boolean) statement invocation was removed in version 2.5.0. Upstream have fixed the issue, but there is no sshd-2.6.0 yet.

RESOLVED	Michal Cmarada	ApiDoc generator fails to parse string types with pattern containing {isBasicLatin}	example typedef with failing pattern:
			`    typedef DisplayString {`
			`        type string {`
			`            length "0..255";`
			`            pattern "((\p{IsBasicLatin}{0,255}))";`
			`        }`
			`    }`
			parsing this type fails with IllegalArgumentException and stops further processing of api docs.

RESOLVED	Robert Varga	restconf /operations: XML not acceptable	> curl -u admin:admin -H "Accept: application/yang-data+json" http://localhost:8181/rests/operations < OK > curl -v -u admin:admin -H "Accept: application/yang-data+xml" http://localhost:8181/rests/operations < 406 Not Acceptable
RESOLVED	Tomas Cere	Regression in SSE Notification test	SSE notification test started failing in master: https://jenkins.opendaylight.org/releng/view/netconf/job/netconf-csit-1node-userfeatures-rfc8040-all-silicon/ After this patch was merged: https://git.opendaylight.org/gerrit/c/netconf/+/92446 Some of the issues detected: • Get DCN streams API fail 500 Server Error: GET /rests/data/ietf-restconf-monitoring:restconf-state/streams • Missing information in DS add event. • No notification sent for DS delete event.
RESOLVED	Kostiantyn Nosach	RFC8040 RESTCONF Auth mechanism is broken	The integration of AAA and RFC8040 RESTconf is broken - auth is not required to access RESTconf API anymore(expectation - at least basic auth should be on by default). It seems like the root cause was the following change-set: https://git.opendaylight.org/gerrit/c/netconf/+/89419

RESOLVED

Marek Hurban

Commit to non-candidate device

Please find attached karaf log (TRACE)
The interesting part is around line 27392

```
2021-01-28T11:35:55,158 | TRACE | qtp1820169956-75 |
NetconfDeviceCommunicator        | 279 - org.opendaylight.netconf.sal-
netconf-connector - 1.12.1 | RemoteDevice{NO-CANDIDATE}: Sending message
<rpc message-id="m-24" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <edit-config>
        <target>
            <running/>
        </target>
        <error-option>rollback-on-error</error-option>
        <config>
            <control-construct xmlns="urn:onf:yang:core-model-1-4">
                <logical-termination-point>
                    <uuid>cdcd6b3f-1543-0d47-48c5-6132ff3f493a</uuid>
                    <label>
                        <value-name>my-label</value-name>
                        <value xmlns:op="urn:ietf:params:xml:ns:netconf:
base:1.0" op:operation="create">works</value>
                    </label>
                </logical-termination-point>
            </control-construct>
        </config>
    </edit-config>
</rpc>

2021-01-28T11:35:55,158 | TRACE | nioEventLoopGroupCloseable-3-1 |
NetconfClientSessionImpl         | 281 - org.opendaylight.netconf.shaded-
sshd - 1.9.1 | encode(NetconfClientSessionImpl[netconf@/127.0.0.1:18300])
packet #89 [chunk #11](704/785) 20 20 20 20 20 20 20 20 20 20 3c 2f 6c 61
62 65 6c 3e 0a 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 3c 2f 6c 6f
67 69 63 61 6c 2d 74 65 72 6d 69 6e 61 74 69 6f 6e 2d 70 6f 69 6e 74 3e
0a    ..........</label>................</logical-termination-point>.
2021-01-28T11:35:55,158 | TRACE | qtp1820169956-75 |
KeepaliveSalFacade               | 279 - org.opendaylight.netconf.sal-
netconf-connector - 1.12.1 | Start to waiting for result.
2021-01-28T11:35:55,159 | TRACE | qtp1820169956-75 |
KeepaliveSalFacade               | 279 - org.opendaylight.netconf.sal-
netconf-connector - 1.12.1 | RemoteDevice{NO-CANDIDATE}: Resetting netconf
keepalive timer
2021-01-28T11:35:55,159 | TRACE | qtp1820169956-75 |
KeepaliveSalFacade               | 279 - org.opendaylight.netconf.sal-
netconf-connector - 1.12.1 | RemoteDevice{NO-CANDIDATE}: Scheduling
keepalives every  120 SECONDS
2021-01-28T11:35:55,159 | TRACE | qtp1820169956-75 |
NetconfDeviceCommunicator        | 279 - org.opendaylight.netconf.sal-
netconf-connector - 1.12.1 | RemoteDevice{NO-CANDIDATE}: Sending message
<rpc message-id="m-25" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <commit/>
</rpc>
```

The NetconfDeviceCommunicator correctly sends the <edit-config> to <running/> - but later also the <commit/>.
In case of non candidate devices the <commit/> must not be send.

This functions correctly in Magnesium.

karaf.log

RESOLVED

Iaroslav Kholiavko

Netconf Testtool fails on Windows if --schemas-dir parameter set

While --schemas-dir parameter set Netconf Testtool fails with exception. Bug reproduces only on Windows.

Expected behavior: Test tool should work the same way as on Linux.

Actual behavior: Test tool fails with exception:

Exception in thread "main" java.lang.IllegalStateException: Failed to rename '%s'.D:\workspace\netconf\schemas\augment-main-a.yangException in thread "main" java.lang.IllegalStateException: Failed to rename '%s'.D:\workspace\netconf\schemas\augment-main-a.yang at org.opendaylight.netconf.test.tool.TesttoolParameters.validate(TesttoolParameters.java:329) at org.opendaylight.netconf.test.tool.Main.main(Main.java:38)

RESOLVED	Michal Banik	`/.well-known /*` endpoint responses only once	/.well-know/* REST endpoint introduced in ~~NETCONF-499~~ response only once. For every other request it is stucked. javax.ws.rs.core.Response.status(Status.OK) creates new OutboundMessageContext with CommittingOutputStream instance, which is closed during the process of sending the response to client. Closed output-stream causes `java.lang.IllegalStateException: The output stream has already been closed.` when `/.well-known/*` endpoint is requesting multiple times, which makes the endpoint working only once.

25 issues

AAA

Status	Assignee	Summary	Description

RESOLVED

Robert Varga

Authentication Manager requires OSGi to function

This issue happens when we are running AAA outside of karaf environment. We try to create instance of org.opendaylight.aaa.shiro.tokenauthrealm.auth.AuthenticationManager, which implements also org.osgi.service.cm.ManagedService. This dependency is stated as provided in poms, but for non karaf environment this is causing runtime errors since it doesnt need to use osgi at all.

stack trace from our app:

```
Exception in thread "pool-14-thread-1" java.lang.RuntimeException: java.
lang.NoClassDefFoundError: org/osgi/service/cm/ManagedServiceException in
thread "pool-14-thread-1"
java.lang.RuntimeException:
java.lang.NoClassDefFoundError: org/osgi/service/cm/ManagedService at
 io.lighty.kit.examples.community.aaa.restconf.Main$1.onFailure(Main.java:
136) at
 com.google.common.util.concurrent.Futures$CallbackListener.run(Futures.
java:1064) at
 java.base/java.util.concurrent.ThreadPoolExecutor.runWorker
(ThreadPoolExecutor.java:1128) at
 java.base/java.util.concurrent.ThreadPoolExecutor$Worker.run
(ThreadPoolExecutor.java:628) at
 java.base/java.lang.Thread.run(Thread.java:834)
Caused by: java.lang.NoClassDefFoundError: org/osgi/service/cm
/ManagedService at
 java.base/java.lang.ClassLoader.defineClass1(Native Method) at
 java.base/java.lang.ClassLoader.defineClass(ClassLoader.java:1017) at
 java.base/java.security.SecureClassLoader.defineClass(SecureClassLoader.
java:174) at
 java.base/jdk.internal.loader.BuiltinClassLoader.defineClass
(BuiltinClassLoader.java:800) at
 java.base/jdk.internal.loader.BuiltinClassLoader.
findClassOnClassPathOrNull(BuiltinClassLoader.java:698) at
 java.base/jdk.internal.loader.BuiltinClassLoader.loadClassOrNull
(BuiltinClassLoader.java:621) at
 java.base/jdk.internal.loader.BuiltinClassLoader.loadClass
(BuiltinClassLoader.java:579) at
 java.base/jdk.internal.loader.ClassLoaders$AppClassLoader.loadClass
(ClassLoaders.java:178) at
 java.base/java.lang.ClassLoader.loadClass(ClassLoader.java:522)
 at io.lighty.aaa.AAALightyShiroProvider.<init>(AAALightyShiroProvider.
java:104) at
 io.lighty.aaa.AAALightyShiroProvider.newInstance(AAALightyShiroProvider.
java:184) at
 io.lighty.aaa.AAALighty.initProcedure(AAALighty.java:63) at
 io.lighty.core.controller.api.AbstractLightyModule.lambda$start$0
(AbstractLightyModule.java:117) at
 com.google.common.util.concurrent.
TrustedListenableFutureTask$TrustedFutureInterruptibleTask.runInterruptibly
(TrustedListenableFutureTask.java:125) at
 com.google.common.util.concurrent.InterruptibleTask.run(InterruptibleTask.
java:69) at
 com.google.common.util.concurrent.TrustedListenableFutureTask.run
(TrustedListenableFutureTask.java:78)
 ... 3 more
Caused by: java.lang.ClassNotFoundException: org.osgi.service.cm.
ManagedService at
 java.base/jdk.internal.loader.BuiltinClassLoader.loadClass
(BuiltinClassLoader.java:581) at
 java.base/jdk.internal.loader.ClassLoaders$AppClassLoader.loadClass
(ClassLoaders.java:178) at
 java.base/java.lang.ClassLoader.loadClass(ClassLoader.java:522) ... 19
more
```

<table>
<tr><td>RESOLVED</td><td>Ilya Igushev</td><td>Migrate aaa-cert use of BC APIs</td><td>

Current build reports:

```
[INFO] --- maven-compiler-plugin:3.8.0:compile (default-compile) @ aaa-cert ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 64 source files to /home/nite/odl/aaa/aaa-cert/target/classes
[WARNING] /home/nite/odl/aaa/aaa-cert/src/main/java/org/opendaylight/aaa/cert/impl/ODLKeyTool.java:[37,34] org.bouncycastle.asn1.x509.X509Name in org.bouncycastle.asn1.x509 has been deprecated
[WARNING] /home/nite/odl/aaa/aaa-cert/src/main/java/org/opendaylight/aaa/cert/impl/ODLKeyTool.java:[38,28] org.bouncycastle.jce.PKCS10CertificationRequest in org.bouncycastle.jce has been deprecated
[WARNING] /home/nite/odl/aaa/aaa-cert/src/main/java/org/opendaylight/aaa/cert/impl/ODLKeyTool.java:[39,28] org.bouncycastle.jce.X509Principal in org.bouncycastle.jce has been deprecated
[WARNING] /home/nite/odl/aaa/aaa-cert/src/main/java/org/opendaylight/aaa/cert/impl/ODLKeyTool.java:[40,29] org.bouncycastle.x509.X509V3CertificateGenerator in org.bouncycastle.x509 has been deprecated
[WARNING] /home/nite/odl/aaa/aaa-cert/src/main/java/org/opendaylight/aaa/cert/impl/ODLKeyTool.java:[173,19] org.bouncycastle.x509.X509V3CertificateGenerator in org.bouncycastle.x509 has been deprecated
[WARNING] /home/nite/odl/aaa/aaa-cert/src/main/java/org/opendaylight/aaa/cert/impl/ODLKeyTool.java:[173,66] org.bouncycastle.x509.X509V3CertificateGenerator in org.bouncycastle.x509 has been deprecated
[WARNING] /home/nite/odl/aaa/aaa-cert/src/main/java/org/opendaylight/aaa/cert/impl/ODLKeyTool.java:[175,43] org.bouncycastle.jce.X509Principal in org.bouncycastle.jce has been deprecated
[WARNING] /home/nite/odl/aaa/aaa-cert/src/main/java/org/opendaylight/aaa/cert/impl/ODLKeyTool.java:[178,44] org.bouncycastle.jce.X509Principal in org.bouncycastle.jce has been deprecated
[WARNING] /home/nite/odl/aaa/aaa-cert/src/main/java/org/opendaylight/aaa/cert/impl/ODLKeyTool.java:[181,59] generateX509Certificate(java.security.PrivateKey) in org.bouncycastle.x509.X509V3CertificateGenerator has been deprecated
[WARNING] /home/nite/odl/aaa/aaa-cert/src/main/java/org/opendaylight/aaa/cert/impl/ODLKeyTool.java:[262,23] org.bouncycastle.asn1.x509.X509Name in org.bouncycastle.asn1.x509 has been deprecated
[WARNING] /home/nite/odl/aaa/aaa-cert/src/main/java/org/opendaylight/aaa/cert/impl/ODLKeyTool.java:[262,44] org.bouncycastle.asn1.x509.X509Name in org.bouncycastle.asn1.x509 has been deprecated
[WARNING] /home/nite/odl/aaa/aaa-cert/src/main/java/org/opendaylight/aaa/cert/impl/ODLKeyTool.java:[262,40] X509Name(java.lang.String) in org.bouncycastle.asn1.x509.X509Name has been deprecated
[WARNING] /home/nite/odl/aaa/aaa-cert/src/main/java/org/opendaylight/aaa/cert/impl/ODLKeyTool.java:[264,23] org.bouncycastle.jce.PKCS10CertificationRequest in org.bouncycastle.jce has been deprecated
[WARNING] /home/nite/odl/aaa/aaa-cert/src/main/java/org/opendaylight/aaa/cert/impl/ODLKeyTool.java:[264,60] org.bouncycastle.jce.PKCS10CertificationRequest in org.bouncycastle.jce has been deprecated
```

This should be addressed, but doing so is not completely straightforward.

</td></tr>
<tr><td>RESOLVED</td><td>Robert Varga</td><td>Eliminate bleuprint from aaa-password-service</td><td>We are using aaa-password-service here, but other than that this is a straightforward conversion. It is on critical path to netconf, hence we should take extra effort to convert it.</td></tr>
<tr><td>RESOLVED</td><td>Robert Varga</td><td>Convert H2 datastore to a OSGi DS component</td><td>The datastore is really standalone here, let's split it into a component first.</td></tr>
</table>

RESOLVED	Tomas Cere	Convert IdmLight to OSGi DS	This is a simple service on top of IDMStore and aaa-password-service, which provides a number of critical services:
			`org.opendaylight.aaa.api.IdMService`
			`org.opendaylight.aaa.api.ClaimCache`
			`org.opendaylight.aaa.api.CredentialAuth`
			`org.opendaylight.aaa.api.PasswordCredentialAuth`
			It is also used by downstreams, hence we want to make it OSGi DS for easier consumability.

RESOLVED	Tomas Cere	Convert ODLAuthenticator into a component	This is a simple standalone component. Split it out of aaa-shiro's impl-blueprint as an OSGi DS component with immediate activation.
RESOLVED	Robert Varga	Convert web-servlet-jersey to OSGi DS	We are using OSGi activator, we should just use OSGi DS and provide a javax.inject singleton.

7 issues

Infrautils

Status	Assignee	Summary	Description

No issues found

JSONRPC

Status	Assignee	Summary	Description
OPEN	Unassigned	Rework Netty pipeline	Current Netty pipeline doesn't model output as jsonrpc message only - it's just a utf-8 text. Also, there is no possibility to do interception, which is required for JSONRPC-35
RESOLVED	Richard Kosegi	Remove async RPC dispatch	This feature never seen wide usage and has various problems: • it promotes usage of bad pattern (application should use PUB/SUB to deliver asynchronous event about task completion) • Separate thread break opentracing propagation as it rely on ThreadLocal to establish span boundary
RESOLVED	Richard Kosegi	Use proxy API in RPC	Currently, JsonRPCtoRPCBridge is using lower-level API (RequesterSession) to perform communication to remote ends. This is because exact interface it's not known at compile-time. Generate interface on-the-fly (using ByteBuddy) based on resolved SchemaContext and replace communication with proxy API.
RESOLVED	Unassigned	Move JSON/DOM codec into separate module	Currently most of conversion is implemented in JsonConverter class with lot of code duplicated /redundant. Create separate maven module for codecs and make their usage more clear to downstream consumers.
RESOLVED	Richard Kosegi	Read operation report HTTP200 via restconf, even for non-existent data	JSONRPC return empty JSON object to restconf, even when that does not actually exists. This behavior prevent simple things such as verify success of delete operation.

RESOLVED	Richard Kosegi	Inband schema provider require YangXPathParserFactory	Error is thrown when peer is mounted using inband schema provider, as it now require YangXPathParserFactory

```
java.lang.ExceptionInInitializerError: No YangXPathParserFactory
found
        at org.opendaylight.yangtools.yang.parser.rfc7950.
reactor.ServiceLoaderState$XPath.lambda$static$0
(ServiceLoaderState.java:32) ~[bundleFile:?]
        at java.util.Optional.orElseThrow(Optional.java:408) ~
[?:?]
        at org.opendaylight.yangtools.yang.parser.rfc7950.
reactor.ServiceLoaderState$XPath.<clinit>(ServiceLoaderState.
java:32) ~[bundleFile:?]
        at org.opendaylight.yangtools.yang.parser.rfc7950.
reactor.RFC7950Reactors.vanillaReactorBuilder(RFC7950Reactors.
java:327) ~[bundleFile:?]
        at org.opendaylight.yangtools.yang.parser.rfc7950.
reactor.RFC7950Reactors.defaultReactorBuilder(RFC7950Reactors.
java:290) ~[bundleFile:?]
        at org.opendaylight.yangtools.yang.parser.rfc7950.
reactor.ServiceLoaderState$DefaultReactor.<clinit>
(ServiceLoaderState.java:23) ~[bundleFile:?]
        at org.opendaylight.yangtools.yang.parser.rfc7950.
reactor.RFC7950Reactors.defaultReactor(RFC7950Reactors.java:280)
~[bundleFile:?]
        at org.opendaylight.jsonrpc.provider.common.
InbandModelsSchemaContextProvider.createSchemaContext
(InbandModelsSchemaContextProvider.java:57) ~[bundleFile:?]
        at org.opendaylight.jsonrpc.model.
CombinedSchemaContextProvider.createSchemaContext
(CombinedSchemaContextProvider.java:44) ~[bundleFile:?]
        at org.opendaylight.jsonrpc.provider.common.
MappedPeerContext.<init>(MappedPeerContext.java:76) ~
[bundleFile:?]
        at org.opendaylight.jsonrpc.provider.single.
JsonRPCProvider.doMountDevice(JsonRPCProvider.java:181) ~
[bundleFile:?]
        at org.opendaylight.jsonrpc.provider.single.
JsonRPCProvider.mountPeers(JsonRPCProvider.java:131) ~
[bundleFile:?]
        at org.opendaylight.jsonrpc.provider.single.
JsonRPCProvider.processNotificationInternal(JsonRPCProvider.java:
106) ~[bundleFile:?]
        at org.opendaylight.jsonrpc.provider.single.
JsonRPCProvider.processNotification(JsonRPCProvider.java:149) ~
[bundleFile:?]
        at org.opendaylight.jsonrpc.provider.single.
JsonRPCProvider.lambda$init$1(JsonRPCProvider.java:158) ~
[bundleFile:?]
```

RESOLVED	Richard Kosegi	RPC implementation not available in cluster	Cluster-aware JSONRPC provider doesn't implement and register JsonRpcService. This leads to

```
{
    "errors": {
        "error": [
            {
                "error-type": "application",
                "error-tag": "operation-not-supported",
                "error-message": "No implementation of RPC
AbsoluteSchemaPath{path=[(urn:opendaylight:jsonrpc?revision=2016-
12-01)force-reload]} available"
            }
        ]
    }
}
```

7 issues

ODL parent

Status	Assignee	Summary	Description
RESOLVED	Robert Varga	Remove sshd declaration	NETCONF is moving to shading mina-sshd itself, to not conflict with Karaf. Once that is done, we should remove the feature and declaration, so that we do not interfere with downstreams.
RESOLVED	Robert Varga	Remove org.eclipse.persistence dependency declarations	These dependencies are only used by Neutron, which has to re-specify versions and wiring to make things working again and provide overrides-and-whatnot to get around ~~NEUTRON-197~~. This is in scope of NEUTRON-200 for which declarations in odlparent would only hurt. Remove all declarations that are used only in Neutron, so that it can a completely local solution based on whatever technology it chooses to deal with that mess.
RESOLVED	Robert Varga	Remove javax.json declaration	This declaration has interplay with org.eclipse.persistence and seems to only be used in neutron. aaa has an equivalent declaration, which seems to be unreferenced. Remove it in our next release.
RESOLVED	Robert Varga	Remove org.codehaus.jettison declaration	This declaration is only used in lispflowmapping, where it is only used in integration tests. Remove its declaration and let lispflowmapping manage the version used.
RESOLVED	Robert Varga	Remove Scala/Akka packaging	As a follow-up from ~~ODLPARENT-223~~, remove Akka/Scala packaging, so we do not interfere with whatever downstreams decide to do. https://github.com/akka/akka/issues/28304 tracks upstream's effort to eradicate existing OSGi support.
RESOLVED	Robert Varga	Add com.guicedee.services:javax.inject declaration	This artifact has the same contents, but modularized for JPMS, allowing downstreams to link against a proper module.
RESOLVED	Oleksii Mozghovyi	Add support for Karaf static distribution	Dynamic distro works ouf of the box, but we provide no tooling to make static distro. Add a proper parent.
RESOLVED	Robert Varga	Add support for @apiNote, @implSpec, @implNote javadoc tags	As noted in https://nipafx.dev/javadoc-tags-apiNote-implSpec-implNote/ there are new apidoc tags used throughout the JDK. While these are not standardized, but were introducied with https://bugs.openjdk.java.net/browse/JDK-8008632 and are documented in https://bugs.openjdk.java.net/browse/JDK-8068562 and are very useful for documentation. Add required javadoc configuration to recognize these tags, essentially saying we are endorsing them – but the spec is really owned by odlparent.
RESOLVED	Robert Varga	Exclude Guava's transitive dependencies from features	Testing with static distribution it is clear we do not want to pull in 'wrap' feature – and the only reason we do is Guava's transitive dependencies. Two of them are annotations, which we do not need and one is a safety against listenablefuture.jar duplicates. Filter these dependencies out, so we do not package them anymore.

RESOLVED	Robert Varga	Service Component Runtime does not conform to R6 spec w.r.t Factory Components	Felix SCR 4.1.20 (and before), as bundled with karaf-4.2.9, fails to enforce correct lifecycle on factory-defined component configurations. In particular, SCR will deactivate such a component configuration when its services become unused, such as going through a getService/ungetService cycle. This mirrors the lifecycle of a Delayed Component, not a Factory component – see sections 112.5.4 and 112.5.5 of OSGi Compendium spec. This issue tracks two things: 1. the upstream fix as tracked in https://issues.apache.org/jira/browse/FELIX-6315 and its adoption in odlparent 2. a stop-gap mitigation The mitigation comes in form of https://issues.apache.org/jira/browse/FELIX-3039, i.e. preventing Felix from deactivating unused delayed components.

10 issues

YANG Tools

Status	Assignee	Summary	Description
RESOLVED	Robert Varga	Cleanup yang-model-util	yang-model-util contains a number of classes relevant to implentations of yang-model-api as well as a number of cl The problem here is that the latter cannot be really tested with real models, as yang-model-util cannot depend on y Split out the implementation classes to yang-model-spi, which is used by yang-parser-rfc7950 (and others) and ma
RESOLVED	Robert Varga	Sort out yang.model.repo.{api,spi,util} placement	During the course of transitioning to JPMS modules the repo.api package placement came up. Resolve it somehow This needs to be examined with yang-parser-api in mind.
RESOLVED	Robert Varga	Restructure git repository	Our current layout is a bit redundant, in that most of our artifacts are in yang/ directory and most start with yang- pr As our scope is pretty much nailed down and we know the general structure of our deliverables, we should have a • model/, hosting yang-model-api and all related API/SPI/util artifacts • data/, hosting yang-data-api and related artifacts (including yang-data-impl) • codec/, hosting yang-data-codec-* • parser/, hosting yang-parser* and *parser-support artifacts • plugin/, hosting maven plugin and plugin-generator-api • tools/, hosting yang-validation-tool Also, yang-common-* should be moved to common/ .
RESOLVED	Robert Varga	Graduate OrderedByStatement. Ordering as yang.common	The concept of ordering is shared between yang-model-api and yang-data-api – similar as QName. Graduate Orde
RESOLVED	Robert Varga	Remove YangModeledAnyxmlSchemaNode	YangModeledAnyxmlSchemaNode is a relic of a design, where this kind of node was treated specially for various X These days we have ForeignDataNode to mediate data exchange. The semantics of this node are readily available There is actually a notable gap in support for this node, which cannot really be serialized by the datastore – hence i Remove this construct and all special handling it receives.
RESOLVED	Robert Varga	Remove yang-data-jaxen	yang-data-jaxen has been deprecated for some time, remove its experimental feature and the corresponding artifac
RESOLVED	Robert Varga	Split out yang-data-spi	yang-data-impl and yang-data-api host a number of elements which should really be part of yang-data-spi. Split the
RESOLVED	Robert Varga	Remove AnyxmlSchemaLocation	AnyxmlSchemaLocationStatement is not defined in the MD-SAL version yang-ext, which everyone is using – hence

RESOLVED	Robert Varga	Move data.impl.schema.builder.api to yang-data-api	Builders exposed from yang-data-impl conform to a well-defined API. Quite a few components depend on this API t Factor this API surface into yang-data-api, so it can be implementated without having ties to yang-data-impl itself.
RESOLVED	Robert Varga	Refine statements are not SchemaNodes	RefineEffectiveStatementImpl happens to implement SchemaNode contract, which is not actually used anywhere. I Drop the implementation of SchemaNode, reducit overhead and the mismatched contract we are attempting to fulfi
RESOLVED	Robert Varga	Add RootStmtContext	Global namespace behavior might play well with StmtContexts which are known to be root. Add such a marker inte
RESOLVED	Miroslav Kovac	Improve InferredStatementContext on-demand instantiation	When we are faced for a query for a particular child, we really do not want to instantiate all children. Given we can use logic from ~~YANGTOOLS-1157~~, we can now precisely know which statement from the prototype w Initialize only that child and remember that action in a Map. When it comes time to materialize children be mindful c
RESOLVED	Miroslav Kovac	Reformulate StatementContextFactory.createEffective()	Rather than passing down StmtContext and its parent via StmtContext.getParent() we want to be in control of what The core contract is StatementContextFactory, which will need two replacement interfaces, and the primary implem We then need to audit all subclasses of StatementContextBase and migrate their accesses to the new interfaces. InferredStatement context plays a role here in that will plugin into these interfaces with methods which do not force
RESOLVED	Michal Banik	Eliminate parser.rfc7950.stmt.AbstractDeclaredStatement's use of StmtContext	As the FIXME in the class hints, we are about to promote this class. We also are moving away from using StmtCon
RESOLVED	Robert Varga	Clarify StatementSupport.applyCopyPolicy()'s targetModule	Audit all callers of this method to see how they are specifying targetModule. Also audit all implementations to see w Based on the audit determine what the contract is and specify it.
RESOLVED	Miroslav Kovac	Audit all StatementSupport classes for correct CopyPolicy	Having a correct CopyPolicy is critical in minimizing the amount of copying we do. Audit all StatementSupports and
RESOLVED	Robert Varga	Release InferredStatementContext's effective statements after build	Performing lazy copy operations means we can end up copying a statement after it has been built and hence shoul Add an explicit ModelProcessingPhase.EFFECTIVE_VIEW, which will run after EFFECTIVE_MODEL has been co It's execution is a straightforward children-first walk from root, just as any other processing phase. All copy operations need to hook a prerequisite for EFFECTIVE_VIEW on their source context – which prevents it f
RESOLVED	Miroslav Kovac	Integrate EffectiveStmtCtx and StmtContext	EffectiveStmtCtx is always implemented as a proxy on top of a StatementContextBase – hence it is a superfluous c Make sure StmtContext and EffectiveStmtCtx.Current can be implemented by StatementContextBase at the same StmtContext should then extend CommonStmtCtx and provide its current method as default forwarders.
RESOLVED	Miroslav Kovac	Shortcut InferredStatement.buildEffective() when no inference was done	With all the work done in ~~YANGTOOLS-784~~ we are now in a place where we have rather good command of substa Unfortunately we still end up materializing substatements even when there was no inference and we may actually e Devise a reactor-internal shortcut where InferredStatementContext can ask its prototype to provide an EffectiveSta InferredStatementContext has readily available.

<table>
<tr><th></th><th></th><th></th><th></th></tr>
<tr><td>RESOLVED</td><td>Robert Varga</td><td>Reuse effective substatement lists when possible</td><td>This is the first step in fly-weight statements. As described in ~~YANGTOOLS-1067~~, we want to reuse List<EffectiveS

This is mostly yang-parser-reactor work, but also requires exposing access to AbstractStatementSupport.createEff

This should prove effective in this test case:
<pre>module foo {
  namespace foo;
  prefix foo;

  grouping grp {
    container foo {
      description "desc";
      reference "ref";
    }
  }

  uses foo;
}</pre>
while /grp/foo and /foo are distinct objects, their EffectiveStatement.effectiveSubstatements() need to return the sar</td></tr>
<tr><td>RESOLVED</td><td>Robert Varga</td><td>Allow substatement-derived state to be inherited</td><td>{Data,Schema}
TreeAwareEffectiveStatements are deriving QName->EffectiveStatement maps from substatements of a particular

With ~~YANGTOOLS-1212~~ we get to reuse the substatement file, but that is not enough – we need to pass the previ

Also update all implementations to pass this derived state around.</td></tr>
<tr><td>RESOLVED</td><td>Robert Varga</td><td>Expand DataTreeAwareEffectiveStatements</td><td>Wider adoption of SchemaInferenceStack shows that we are lacking the ability to follow data tree from choice/case

While we are reconsidering things, it might well be suitable for groupings as well.</td></tr>
<tr><td>RESOLVED</td><td>Robert Varga</td><td>Add SchemaInferenceStack.enterChoice(QName)</td><td>yang-data-impl's SchemaTracker performs a rather weird dance, where (due to NormalizedNode) hierarchy it perfo

While it can enter a ChoiceEffectiveStatement via schema tree, it needs to deal with choice/case/choice traversal,

SchemaInferenceStack is well equipped to handle this, as it already needs to deal with choice/case crossing during

Introduce an alternative enterChoice(QName), which will examine parent and based on that either turns into a ente</td></tr>
<tr><td>RESOLVED</td><td>Robert Varga</td><td>Do not use DataSchemaNode in Anydata normalization</td><td>NormalizedAnydata and NormalizableAnydata rely on DataSchemaNode to provide enough context to decode the

We have a better replacement in EffectiveStatementInference, which combines an EffectiveModelContext with an</td></tr>
<tr><td>RESOLVED</td><td>Robert Varga</td><td>Promote yang.data.impl.SchemaTracker</td><td>yang-data-impl's SchemaTracker provides a common interface used by yang-data-codec-{gson,xml} when interacti

SchemaTracker really follows NormalizedNodeStreamWriter contract's view on NormalizedNode, hence it is fair to
is probably a good-enough name.</td></tr>
<tr><td>RESOLVED</td><td>Robert Varga</td><td>Remove SchemaUtils methods</td><td>There is a ton of now-unused methods in SchemaUtils. Remove them to prevent confusion.</td></tr>
<tr><td>RESOLVED</td><td>Robert Varga</td><td>Define ModelProcessingPhase.ExecutionOrder</td><td>ModelProcessingPhase enumeration really is a strongly-ordered, i.e. it amounts to what steps need to be taken, in

Define ExecutionOrder, which is actually a set of positive byte values reflecting current implications of ModelProces

This allows us to store the equivalent of ModelProcessingPhase in StatementContextBase for the cost of a single b</td></tr>
<tr><td>RESOLVED</td><td>Robert Varga</td><td>Inline CopyHistory storage</td><td>CopyHistory is a rather simple construct, recording 4 bits of information, for which are expending 4/8 bytes in every

Store all history information in a single byte, providing CopyHistory operations as a bit masking exercise.</td></tr>
</table>

RESOLVED	Robert Varga	Add org.opendaylight.yang.parser.api.YangParserFactory et al.	We have slight confusion in yang-parser-api naming, where the name of the artifact does not actually match the ex As a first step retrofit the API and implementations to export both packages.
RESOLVED	Robert Varga	Add ItemOrder.{Ordered, Unordered} concept	YANG defines distinct equality classes based on whether order of items is significant. This is typically expressed vi Our NormalizedNode model does not correctly capture this property, leading to ill-defined equality contract where N Define a general ItemOrder construct with two subclasses, Ordered and Unordered, which capture this contract. Th
RESOLVED	Robert Varga	Enforce unique constraints when opted-in	DataTreeConfiguration.uniqueIndexes is a configuration option we are currently ignoring. It should result in unique As a first step add UniqueValidation, which will perform external unindexed validation – i.e. will not keep additional
RESOLVED		Add the option to retain declaration reference in DeclaredStatement	This feature allows to create a navigation link to anchor in source file for "main" statements in YANGs Visualization This feature already implemented in https://github.com/anton-tregubov/yangtools/tree/6.0.x But I can't find any way
RESOLVED	Robert Varga	Do not generate EffectiveStatements for unbound extensions	Our current behavior regarding extensions is that we try to fudge them, making a number of guesses around what t This involves quite some complexity, recursive statement builds and the like – which seems counter-productive giv
RESOLVED	Robert Varga	Add yang.model.api.EffectiveStatementInference	A number of places need to exchange a stack of EffectiveStatements as a means to maintain cohesiveness within • associated EffectiveModelContext • an ordered sequalve of EffectiveStatements Introduce EffectiveStatementInference as such a capture and integrate it with SchemaInferenceStack, providing a s
RESOLVED	Robert Varga	Add yang.model.api.SchemaTreeInference	As a special case of EffectiveStatementInference, we have a class which guarantees that all constituent statements Introduce SchemaTreeInference, which will capture this capability.
RESOLVED	Robert Varga	Move RFC7950 effective statements to yang-model-ri	yang-model-spi is a common place to keep implementation-shared bits and pieces. Just as we moved declared sta
RESOLVED	Robert Varga	Convert yang-common-netty to a JPMS module	yang-common-netty is a very simple library, convert it to a JPMS module

RESOLVED	Miroslav Kovac	Eliminate duplicate DescriptionEffectiveStatementImpl objects	Analysis of a memory snapshot with vendor models shows we end up with a SchemaContext which weighs about 2 grouping.
			This hints at us not having the understanding that a statement has been copied and unmodified from a grouping. A
			This theory is reinforced by the fact that the same dump shows ~64K WhenEffectiveStatementImpl objects (23x gro
			`grouping foo {`
			`  leaf bar {`
			`    when "../x";`
			`  }`
			`}`
			A similar situation happens with UnitsEffectiveStatementImpl – 15K instances wasting 450kB.

RESOLVED	Miroslav Kovac	Defer statement initialization in InferredStatementContext	In order to implement efficient grouping inlining with working augment/deviate statements, we need to have the con Current design assumption is that namespaces are populated eagerly with key/value pairs and inference actions ca Efficient inlining (~~YANGTOOLS-694~~) requires that not all nodes are instantiated, which would lead to under-populat Currently the only use of listeners is to perform mutation, which simplifies the contract, as we really need to have a - understand that a node represents a tree (and is otherwise a StmtContext.Mutable) - understand that a listener is interested in some of its children - request such a node to give us a StmtContext.Mutable representation of its specific child (potentially again a Mut Further simplification comes from the fact that we need this to work only in one specific namespace, which is ChildS
RESOLVED	Robert Varga	Make DataSchemaNode. isConfiguration() tri-state	The idea that isConfiguration() boils down to a simple boolean is flawed, as DataSchemaNodes defined under a gr

```
module foo {

  grouping foo {
    container indeterminate {
       // ancestors is neither true or false, hence we do not know
    }
  }

  container cfg {
    // No config, defaults to true
    grouping foo {
      container indeterminate {
        // ancestor is config=true, but this may be used in config=false
      }
    }
  }

  container oper {
    // Explicitly false
    config false;

    grouping foo {
      container false {
        // ancestor is config=false and hence this can only be referenced
        // by other config=false nodes.
      }
    }
  }
}
```

			For the purposes of modeling, though, DataSchemaNodes defined within a grouping hierarchy should have a isCor Semantically the question is what happens to nodes in notifications/actions/RPCs – they are not inherently tied to a /notifications on any datastore, these should end up being undefined, too.
RESOLVED	Robert Varga	Redefine TypeStatement in terms of QName	TypeStatement currently takes String argument, which is not entirely accurate, as it does not provide type expansic This leads to SourceExceptions during building of effective statement rather than a failure to parse the argument wi Fix the definition of TypeStatement and TypeEffectiveStatement so that these take a QName argument. This leads to the question of which QNameModule should built-in types resolve – and that should be "urn:ietf:paran
RESOLVED	Robert Varga	Ditch URI as namespace identifier	URI is a bit baroque for our needs as namespace representation. It is not as clunky as java.util.Date was for revisio Most notably we routinely need to convert between String-based Java namespace and URI, hence we end up addi Bite the bullet and define a yang.common.Namespace, which will just wrap a String and validate it.
RESOLVED	Robert Varga	Rename concepts. Variant	Variant is a general name, where Scala has a Try (for exceptions) and Either – the latter been a terminus technicus

RESOLVED	Robert Varga	Introduce parser.spi.SchemaTreeNamespace	As it turns out ChildSchemaNodeNamespace is an addressing contract missing from reactor, but intended to be the Promote ChildSchemaNodeNamespace to a yang-parser-spi construct and properly tie it with other pieces of that A
RESOLVED	Robert Varga	Remove parser.spi.SchemaNodeIdentifierNamespace	SchemaNodeIdentifierNamespace was never implemented and reactor-wards we have SchemaTreeNamespace, w
RESOLVED	Robert Varga	Do not depend on javax.xml.bind	We are using DatatypeConverter just for Base64 parsing. Replace this dependency with Base64's MIME decoder.
RESOLVED	Robert Varga	Convert yang-data-impl into a JPMS module	yang-data-impl is our primary implementation, make sure it is a proper JPMS module.
RESOLVED	Robert Varga	Define ModificationApplyOperation.toString()	Most of the implementations already have a toString() contract, but it is not carried in the base class. Improve debu

RESOLVED	Robert Varga	Update StoreTreeNode methods	yang.data.api.schema.tree.StoreTreeNode.getChild() is forcing users to deal with optionals, where a lot of the time
			Update StoreTreeNode so that it has three methods:
			`@Nullable TreeNode childByArg(PathArgument);`
			`@NonNull TreeNode getChildByArg(PathArgument); // throws VerifyException`
			`Optional<TreeNode> findChildByArg(PathArgument);`

RESOLVED	Robert Varga	Update MutableTreeNode methods	yang.data.api.schema.tree.spi.MutableTreeNode acts as a lazy intermediary between ModificationApplyOperation a Update these methods (renaming addChild() to putChild()) to match map semantics.
RESOLVED	Robert Varga	Update DataNodeContainer methods	DataNodeContainer.getDataChildByName() is a @Nullable getter, which is used in that capacity all over the place. Fix this by centering the implementation around dataChildByName(), which is nullable and then spinning Optional a
RESOLVED	Robert Varga	Remove use of StatementSourceReference from AbstractDeclaredEffectiveStatement	AbstractDeclaredEffectiveStatement uses StatementSourceReference to throw SourceException when schema tre Define two IllegalArgumentExceptions and throws those instead, adjusting callers to wrap these exceptions with Sc
RESOLVED	Robert Varga	Eliminate duplicate indexing in AbstractEffectiveModule	AbstractEffectiveModule performs schema tree/data tree indexing, which is duplicating work already done by its su Switch to using pre-built index, allowing us to speed up statement creation and lower memory footprint of {Module,S

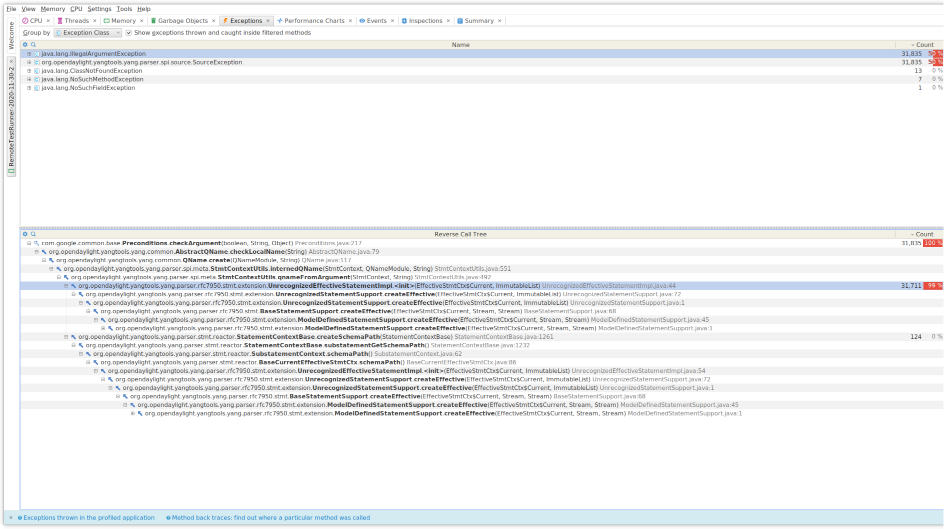
RESOLVED

Robert Varga

Eliminate exception throws from UnrecognizedEffectiveStatementImpl

Unrecognized statements get some very weird handling in an attempt to create a SchemaPath, including attempting

This leads to a lot of exceptions being thrown:


File View Memory CPU Settings Tools Help
CPU × Threads × Memory × Garbage Objects × Exceptions × Performance Charts × Events × Inspections × Summary ×
Welcome
RemoteTestRunner-2020-11-30-2

Group by Exception Class ☑ Show exceptions thrown and caught inside filtered methods

```
Name                                                               Count
⊞ java.lang.IllegalArgumentException                              31,835  50 %
⊞ org.opendaylight.yangtools.yang.parser.spi.source.SourceException 31,835  50 %
⊞ java.lang.ClassNotFoundException                                    13   0 %
⊞ java.lang.NoSuchMethodException                                      7   0 %
⊞ java.lang.NoSuchFieldException                                       1   0 %
```

```
Reverse Call Tree                                                                                                                                                     Count
⊟ com.google.common.base.Preconditions.checkArgument(boolean, String, Object) Preconditions.java:217                                                                31,835 100 %
  ⊟ org.opendaylight.yangtools.yang.common.AbstractQName.checkLocalName(String) AbstractQName.java:79
    ⊟ org.opendaylight.yangtools.yang.common.QName.create(QNameModule, String) QName.java:117
      ⊟ org.opendaylight.yangtools.yang.parser.spi.meta.StmtContextUtils.internedQName(StmtContext, QNameModule, String) StmtContextUtils.java:551
        ⊟ org.opendaylight.yangtools.yang.parser.spi.meta.StmtContextUtils.qnameFromArgument(StmtContext, String) StmtContextUtils.java:492
          ⊟ org.opendaylight.yangtools.yang.parser.rfc7950.stmt.extension.UnrecognizedEffectiveStatementImpl.<init>(EffectiveStmtCtx$Current, ImmutableList) UnrecognizedEffectiveStatementImpl.java:44   31,711 99 %
            ⊟ org.opendaylight.yangtools.yang.parser.rfc7950.stmt.extension.UnrecognizedStatementSupport.createEffective(EffectiveStmtCtx$Current, ImmutableList) UnrecognizedStatementSupport.java:72
              ⊟ org.opendaylight.yangtools.yang.parser.rfc7950.stmt.extension.UnrecognizedStatementSupport.createEffective(EffectiveStmtCtx$Current, ImmutableList) UnrecognizedStatementSupport.java:1
                ⊟ org.opendaylight.yangtools.yang.parser.rfc7950.stmt.BaseStatementSupport.createEffective(EffectiveStmtCtx$Current, Stream, Stream) BaseStatementSupport.java:68
                  ⊟ org.opendaylight.yangtools.yang.parser.rfc7950.stmt.extension.ModelDefinedStatementSupport.createEffective(EffectiveStmtCtx$Current, Stream, Stream) ModelDefinedStatementSupport.java:45
                    ⊞ org.opendaylight.yangtools.yang.parser.rfc7950.stmt.extension.ModelDefinedStatementSupport.createEffective(EffectiveStmtCtx$Current, Stream, Stream) ModelDefinedStatementSupport.java:1
          ⊟ org.opendaylight.yangtools.yang.parser.stmt.reactor.StatementContextBase.createSchemaPath(StatementContextBase) StatementContextBase.java:1261   124 0 %
            ⊟ org.opendaylight.yangtools.yang.parser.stmt.reactor.StatementContextBase.substatementGetSchemaPath() StatementContextBase.java:1232
              ⊟ org.opendaylight.yangtools.yang.parser.stmt.reactor.SubstatementContext.schemaPath() SubstatementContext.java:62
                ⊟ org.opendaylight.yangtools.yang.parser.stmt.reactor.BaseCurrentEffectiveStmtCtx.schemaPath() BaseCurrentEffectiveStmtCtx.java:86
                  ⊟ org.opendaylight.yangtools.yang.parser.rfc7950.stmt.extension.UnrecognizedEffectiveStatementImpl.<init>(EffectiveStmtCtx$Current, ImmutableList) UnrecognizedEffectiveStatementImpl.java:54
                    ⊟ org.opendaylight.yangtools.yang.parser.rfc7950.stmt.extension.UnrecognizedStatementSupport.createEffective(EffectiveStmtCtx$Current, ImmutableList) UnrecognizedStatementSupport.java:72
                      ⊟ org.opendaylight.yangtools.yang.parser.rfc7950.stmt.extension.UnrecognizedStatementSupport.createEffective(EffectiveStmtCtx$Current, ImmutableList) UnrecognizedStatementSupport.java:1
                        ⊟ org.opendaylight.yangtools.yang.parser.rfc7950.stmt.BaseStatementSupport.createEffective(EffectiveStmtCtx$Current, Stream, Stream) BaseStatementSupport.java:68
                          ⊟ org.opendaylight.yangtools.yang.parser.rfc7950.stmt.extension.ModelDefinedStatementSupport.createEffective(EffectiveStmtCtx$Current, Stream, Stream) ModelDefinedStatementSupport.java:45
                            ⊞ org.opendaylight.yangtools.yang.parser.rfc7950.stmt.extension.ModelDefinedStatementSupport.createEffective(EffectiveStmtCtx$Current, Stream, Stream) ModelDefinedStatementSupport.java:1
```

× Exceptions thrown in the profiled application
Method back traces: find out where a particular method was called

We should be able to do better if we expose validity checking mechanics from AbstractQName and rely on that inst

RESOLVED

Robert Varga

Reduce StatementSourceReference proliferation

We have a large number of users who assume acquiring CommonStmtContext.sourceReference() is cheap, which

Refactor SourceException and its subclasses to offer instantiation with CommonStmtCtx, calling sourceReference(

This should help with code verbosity as well as improve performance.

RESOLVED

Robert Varga

Release GroupingNamespace when module/submodule is done

We currently do not wipe GroupingNamespace because it is used after inference by UsesStatementSupport.getSo

Logic in UsesStatementSupport needs to be updated to store a StmtContext.replicaAsChildOf() for its context and a

This way we do not look into GroupingNamespace after inference, have the grouping tied down with a proper refere

Once this is done, add GroupingNamespace to RootStatementContext.SWEPT_NAMESPACES.

RESOLVED

Robert Varga

Release TypeNamespace when module/submodule is done

We currently do not wipe TypeNamespace because it is used by AbstractTypeSupport.resolveType() after inferenc

Similar to ~~YANGTOOLS-1197~~, we need to pick the source type during inference and store a replica of it in the type'

Once that is done, add TypeNamespace to RootStatementContext.SWEPT_NAMESPACES.

RESOLVED

Ivan Hrasko

Improve NormalizedNode formatting capabilities

In case of data tree operation errors we end up dumping the NormalizedNode involved in the operation this way:

```
[pool-3-thread-5] ERROR org.opendaylight.mdsal.dom.spi.store.SnapshotBackedWri
opendaylight:params:xml:ns:yang:bgp-rib?revision=2018-03-29)id=test-rib}]/loc-
ndaylight:params:xml:ns:yang:bgp-rib?revision=2018-03-29)afi=(urn:opendaylight
29)safi=(urn:opendaylight:params:xml:ns:yang:bgp-types?revision=2020-01-20)uni
ily}]/(urn:opendaylight:params:xml:ns:yang:bgp-inet?revision=2018-03-29)ipv4-r
xml:ns:yang:bgp-inet?revision=2018-03-29)route-key=1.1.1.1/32}]:ImmutableMapEn
aylight:params:xml:ns:yang:bgp-inet?revision=2018-03-29)ipv4-route[{(urn:opend
route-key=1.1.1.1/32}], value=[ImmutableLeafNode{identifier=(urn:opendaylight:
?revision=2018-03-29)prefix, value=1.1.1.1/32}, ImmutableContainerNode{identif
opendaylight:params:xml:ns:yang:bgp-inet?revision=2018-03-29)origin, value=[Im
opendaylight:params:xml:ns:yang:bgp-inet?revision=2018-03-29)value, value=igp}
[ImmutableLeafNode{identifier=(urn:opendaylight:params:xml:ns:yang:bgp-inet?re
, ImmutableChoiceNode{identifier=(urn:opendaylight:params:xml:ns:yang:bgp-inet
revision=2018-03-29)ipv4-next-hop, value=[ImmutableLeafNode{identifier=(urn:op
evision=2018-03-29)global, value=2.2.2.2}]}]}, ImmutableContainerNode{identifi
opendaylight:params:xml:ns:yang:bgp-inet?revision=2018-03-29)pref, value=100}]
urn:opendaylight:params:xml:ns:yang:bgp-inet?revision=2018-03-29)as-path, valu
ImmutableLeafNode{identifier=(urn:opendaylight:params:xml:ns:yang:bgp-inet?rev
.1.1/32}]} in MutableDataTree [modification=ModifiedNode{identifier=(urn:ietf:
03-29)bgp-rib=ModifiedNode{identifier=(urn:opendaylight:params:xml:ns:yang:bgp
-rib, operation=TOUCH, childModification={(urn:opendaylight:params:xml:ns:yang
operation=TOUCH, childModification={(urn:opendaylight:params:xml:ns:yang:bgp-r
opendaylight:params:xml:ns:yang:bgp-rib?revision=2018-03-29)id=test-rib}]=Modi
rib?revision=2018-03-29)id=test-rib}], operation=TOUCH, childModification={(ur
yang:bgp-rib?revision=2018-03-29)loc-rib=ModifiedNode{identifier=(urn:opendayl
yang:bgp-rib?revision=2018-03-29)tables=ModifiedNode{identifier=(urn:opendayli
revision=2018-03-29)tables, operation=TOUCH, childModification={(urn:opendayli
opendaylight:params:xml:ns:yang:bgp-types?revision=2020-01-20)ipv4-address-fam
ms:xml:ns:yang:bgp-rib?revision=2018-03-29)safi=(urn:opendaylight:params:xml:n
yang:bgp-rib?revision=2018-03-29)tables[{(urn:opendaylight:params:xml:ns:yang:
=(urn:opendaylight:params:xml:ns:yang:bgp-types?revision=2020-01-20)ipv4-addre
revision=2020-01-20)unicast-subsequent-address-family}], operation=NONE}}}}}}}

org.opendaylight.yangtools.yang.data.impl.schema.tree.SchemaValidationFailedEx
yang:bgp-rib?revision=2018-03-29)id=test-rib}]/loc-rib/tables/tables[{(urn:ope
gp-rib?revision=2018-03-29)afi=(urn:opendaylight:params:xml:ns:yang:bgp-types?
params:xml:ns:yang:bgp-types?revision=2020-01-20)unicast-subsequent-address-fa
:xml:ns:yang:bgp-inet?revision=2018-03-29)ipv4-routes is not present in schema

    at org.opendaylight.yangtools.yang.data.impl.schema.tree.InMemoryDataTreeM

    at org.opendaylight.yangtools.yang.data.impl.schema.tree.InMemoryDataTreeM

    at org.opendaylight.mdsal.dom.spi.store.SnapshotBackedWriteTransaction.wri

    at org.opendaylight.mdsal.dom.broker.DOMForwardedWriteTransaction.put(DOMF

    at org.opendaylight.mdsal.dom.spi.ForwardingDOMDataReadWriteTransaction.pu

    at org.opendaylight.protocol.bgp.rib.impl.LocRibWriter.updateLocRib(LocRib

    at org.opendaylight.protocol.bgp.rib.impl.LocRibWriter.walkThrough(LocRibW

    at org.opendaylight.protocol.bgp.rib.impl.LocRibWriter.onDataTreeChanged(L

    at org.opendaylight.mdsal.dom.store.inmemory.InMemoryDOMStoreTreeChangePub

    at org.opendaylight.yangtools.util.concurrent.AbstractQueuedNotificationMa

    at org.opendaylight.yangtools.util.concurrent.AbstractBatchingExecutor$Dis

    at org.opendaylight.yangtools.util.concurrent.AbstractBatchingExecutor$Dis

    at java.base/java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolE

    at java.base/java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPool

    at java.base/java.lang.Thread.run(Thread.java:834)
```

which does contain a lot of information, but it is in a very unfriendly format.

Improve NormalizedNode with:

```
interface NormalizedNode {
   Supplier<String> prettyTree();
}
```

which will return a memoizing supplier:

```
final class NormalizedNodePrettyTree implements Supplier<String> {
   private @Nullable String str;

    // Lazy init, etc.

    @Override
    public String toString() {
        return get();
    }
}
```

The resulting string should be very similar to what NormalizedNodes.toStringTree() does, except it needs to print a

RESOLVED	Robert Varga	Refactor IdentifierNamespace	org.opendaylight.yangtools.yang.model.api.meta.IdentifierNamespace is an interface specifying a get() method, wh The reality is that this interface is not really meant to be instantiated – there are no yang-model-api-level means to Since the parser internals will be disconnected from this interface with ~~YANGTOOLS-1204~~, reconsider the overall c
RESOLVED	Robert Varga	Improve model.api.ValueRange$Range usage	ValueRange is a simple construct which works on any java.lang.Number. Examining a heap dump of Junos 19.3 te Each BigInteger costs us 64 bytes of memory, even when we could get away with much smaller storage – Integer/L Track down the user and try to persuade it to use smaller Number classes.
RESOLVED	Robert Varga	Migrate users of StatementPolicy.legacyDeclaredCopy()	~~YANGTOOLS-1195~~ left out a few complex statements out of the full-dedup conversion. This includes schema tree Eliminate these users by providing the appropriate policy and culminating in removal of legacyDeclaredCopy().
RESOLVED	Robert Varga	Track substatement independence in StatementContextBase	When we are considering a statement for reuse of EffectiveStatement, we need to consult both the statement itself The substatement consultation is very rudimentary: we only deal with the case on empty statements and fall back t There is a middle step we can take: if all substatements are CONTEXT_INDEPENDENT or IGNORE, we can reuse While this is very similar to the aims of ~~YANGTOOLS-1067~~, this mechanism should kick in sooner, i.e. when the sta ~~YANGTOOLS-1067~~ then deals with the case when all substatements (including nominally non-reusable DECLARE
RESOLVED	Robert Varga	Make CopyHistory implement CopyableNode	It has become obvious most users want the same thing and we are ending up with duplicated code. Furthermore th Make CopyHistory implement CopyableNode and migrate all callers mucking with contains(CopyType) to use those
RESOLVED	Robert Varga	Require QName instead of SchemaPath for TypeDefinition implementions	TypeDefinition has no semantics dependent on SchemaPath, just QName – ever since ~~YANGTOOLS-1098~~. Since https://git.opendaylight.org/gerrit/q/I3490d6d706113ea14ee3f4181a7ba827380c8181 the default SchemaPat This makes TypeDefinition.getPath() a pure implementation detail. This ends up being costly in terms of memory o
RESOLVED	Robert Varga	Add QNameAwareData{Input,Output}	QName implements WritableObject and we also have Data{input,Output} in yang-data-codec-binfmt. Currently QName.writeTo() has no way to take advantage of facilities provided by NormalizedNodeDataOutput, lea Add common meeting points for the two components, so that they can efficiently interoperate.

RESOLVED	Robert Varga	Do not synchronize in ReactorStmtContext. schemaPath()	Our parser is inherently single-threaded, there is no point in having volatiles/synchronized guarding just this field. M
RESOLVED	Robert Varga	Switch non-existent arguments to Empty instead of Void	We currently allow ModelStatement.argument() to return null for the purposes on supporting argumentless stateme We really want to make sure nulls do no occur – and for that yang.common.Empty is a much better fit. Convert all Void use to Empty, making sure ModelStatement.argument() always returns non-null.
RESOLVED	Robert Varga	Defer substatement search to effective instace	BoundStmtCtx.findSubstatementArgument() and hasSubstatement() are search methods very similar to what Effec Take advantage of this to perform a faster search without the need to look at isSupportedToBuildEffective().
RESOLVED	Robert Varga	Do not retain reused substatements	In case we end up not needing to copy substatements and there is no existing reference towards us, we can skip ir This reduces the amount of garbage for the cost of a modest check against parent reference axis.
RESOLVED	Robert Varga	Move AbstractDeclaredStatement et al. to yang-model-spi	AbstractDeclaredStatement is a useful baseline class for implementations. Since we now have have a proper artifa
RESOLVED	Robert Varga	Move RFC7950 declared statements to yang-model-ri	RFC7950 declared statements are covered by the yang.model.api.stmt set of contracts. Make sure yang.model.spi
RESOLVED	Robert Varga	Remove SchemaNode .getPath() from yang-data-codec-xml	SchemaNode.getPath() is only used for diagnostic purposes, other logic should not be affected. Find a suitable rep
RESOLVED	Miroslav Kovac	Rework SchemaContextUtil. getBaseTypeForLeafRef()	SchemaContextUtil.getBaseTypeForLeafRef() is the most significant user of SchemaNode.getPath(). Create a suitable replacement to work with data.util.codec.AbstractCodecFactory and SchemaAwareXMLStreamW
RESOLVED	Robert Varga	Remove SchemaUtils. findParentSchemaNodesOnPath()	This method is overly-general in an attempt to deal with uninstantiated contexts. As it turns out, though, these are r Use SchemaInference stack to sort out addressing and initialization, removing users of this method and then the m
RESOLVED	Robert Varga	A SchemaInferenceStack. enterDataTree()	So far SchemaInferenceStack only dealt with simple steps like grouping/schema tree. In order to make it more usef While the enter/exit logic is very much similar to what schema tree does, the result is not exactly the same: toScher
RESOLVED	Robert Varga	Rework AbstractCodecFactory. codecFor()	AbstractCodecFactory performs leafref resolution based on an 'S extends TypeAware & SchemaNode' – leading to ~~YANGTOOLS-1229~~ deals with properly moving that method to SchemaInferenceStack, effectively resulting in code Since SchemaInferenceStack is a mutable structure, we do not want to deal with its lifecycle, but rather have caller
RESOLVED	Robert Varga	Maintain a SchemaInferenceStack in SchemaTracker	SchemaTracker acts as a common tracker for yang.model.api equivalence, tracking arguments for things like Sche Maintain a SchemaInferenceStack inside SchemaTracker, which should improve performance and remove some c

RESOLVED	Robert Varga	Clean up {Schema, Data}TreeAwareStatement lookups	Current set of lookup methods uses an unsafe type-variance:
			`<E extends SchemaTreeEffectiveStatement<?>> Optional<E> findSchemaTreeNode(Lis`
			This needs to be split up into multiple methods, so we support proper Class-based filters:
			`<E> Optional<E> findSchemaTreeNode(Class<E> type, List<QName> qnames);`
			while also providing a simplified
			`Optional<SchemaTreeEffectiveStatement<?>> findSchemaTreeNode(List<QName> qname`

RESOLVED	Robert Varga	Improve SharedSchemaContextFactory	SharedSchemaContextFactory is widely used for caching context. Unfortunately it has a major deficiency around fi Refactor the implementation to remember futures, so concurrent accesses end up waiting for the initial computatior not being referenced.

RESOLVED	Robert Varga	Optimize StatementContextBase layout	We currently store CopyHistory as a reference to an object, costing us 4b/8b of storage in StatementContextBase.
			StatementContextBase is the bread&butter of memory/CPU trade-offs and it does not have a single bit of storage le
			At the end of the day, CopyHistory encodes the following information:
			• last CopyType
			• CopyableNode.isAugmenting()
			• AddedByUsesAware.isAddedByUses()
			This currently boils down to 2+log2(4)=4 significant bits, for which we are storing 32/64 bits.
			Refactor StatementContextBase.copyHistory to be
			`private final byte copyHistory;`
			This frees up 3-7 bytes of storage, which provides 24bits of additional free storage.
			Another candidate is StatementContextBase.completedPhase, which again is 4/8 bytes of storage, but it only store
			We could use only 3 bits for storing that state, but that would incur masking operations for sub-byte access. Since t
			`private byte completedPhase;`
			Which again frees up 3-7 bytes of storage.
			Combining these two, though, we get real gains – we trim down 8-16 bytes of storage for everything 64bit, while als

RESOLVED	Robert Varga	Convert yang-data-codec-xml into a JPMS module	The XML codec is one of the key bits of yangtools, make sure it is a JPMS module
RESOLVED	Robert Varga	Propagate SchemaSourceRepresentation symbolic names	SchemaSourceRepresentation has the notion of a symbolic name, which aids debugging. Unfortunately a number Audit all participants and ensure they are doing the right thing. For yangtools-7, also make sure symbolic name is s

<table>
<tr><td>RESOLVED</td><td>Miroslav Kovac</td><td>Fix SchemaContextUtil. findDataSchemaNodeForRelativeXPath()</td><td>

On resolving relative paths for key_leaf1 and leaf2 leafrefs(with findDataSchemaNodeForRelativeXPath) SchemaC

module

```
module a {
  yang-version 1;
  namespace "a";
  prefix "a";

  grouping foo_grp {
    leaf key_leaf1 {
      type leafref {
        path "../../../../../../a:foo_cont/a:name";
      }
    }
  }

  container bar_cont {
    leaf leaf2 {
       type leafref {
        path "../../../../../../a:foo_cont/a:name";
      }
    }
  }
}
```

Incorrect paths:
for key_leaf1: [(a)foo_grp, (a)key_leaf1, (a)foo_cont, (a)name]

for leaf2: [(a)bar_cont, (a)leaf2, (a)foo_cont, (a)name]

On call findDataSchemaNodeForRelativeXPath with leaf2 fails with

```
java.lang.IllegalArgumentException: Path [(a)bar_cont, (a)leaf2, (a)foo_cont,
```

On call findDataSchemaNodeForRelativeXPath with key_leaf1 returns null.

Let's check whether the path goes behind grouping root node or root node of a different type. With this improvemen

</td></tr>
<tr><td>RESOLVED</td><td>Robert Varga</td><td>ListModificationStrategy cannot handle child modifications</td><td>Ever since at least I70c9745fc9ed8751c643d9bd3e9d58be10aa4a0e our handling of nested operations is broken, a</td></tr>
<tr><td>RESOLVED</td><td>Robert Varga</td><td>Parser does not reject use of mis-matching augment argument</td><td>

Augment statement support gets quite confused by itself and how it is supposed to work its argument. For starters i

Fix the implementation to enforce augment statement validity.

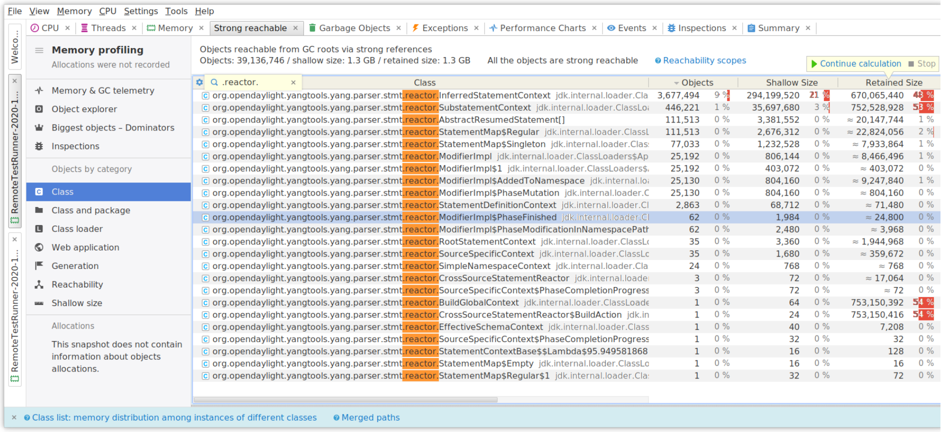
</td></tr>
</table>

RESOLVED

Robert Varga

ModifierImpl leaks post buildEffective ()

Examining a reference benchmark's heap dump, the following is retained between when buildEffective() finishes ar



File View Memory CPU Settings Tools Help
CPU × | Threads × | Memory × | Strong reachable × | Garbage Objects × | Exceptions × | Performance Charts × | Events × | Inspections × | Summary ×
Welco...
RemoteTestRunner-2020-1...
RemoteTestRunner-2020-1...
Memory profiling
Allocations were not recorded
Memory & GC telemetry
Object explorer
Biggest objects – Dominators
Inspections
Objects by category
Class
Class and package
Class loader
Web application
Generation
Reachability
Shallow size
Allocations
Continue calculation
Stop
Reachability scopes
.reactor.
Class list: memory distribution among instances of different classes
Merged paths

This snapshot does not contain information about objects allocations.

Objects reachable from GC roots via strong references
Objects: 39,136,746 / shallow size: 1.3 GB / retained size: 1.3 GB   All the objects are strong reachable

Class	Objects		Shallow Size		Retained Size	
org.opendaylight.yangtools.yang.parser.stmt.reactor.InferredStatementContext jdk.internal.loader.Cla	3,677,494	9 %	294,199,520	21 %	670,065,440	48 %
org.opendaylight.yangtools.yang.parser.stmt.reactor.SubstatementContext jdk.internal.loader.ClassLo	446,221	1 %	35,697,680	3 %	752,528,928	53 %
org.opendaylight.yangtools.yang.parser.stmt.reactor.AbstractResumedStatement[]	111,513	0 %	3,381,552	0 %	≈ 20,147,744	1 %
org.opendaylight.yangtools.yang.parser.stmt.reactor.StatementMap$Regular jdk.internal.loader.ClassL	111,513	0 %	2,676,312	0 %	≈ 22,824,056	2 %
org.opendaylight.yangtools.yang.parser.stmt.reactor.StatementMap$Singleton jdk.internal.loader.Clas	77,033	0 %	1,232,528	0 %	≈ 7,933,864	1 %
org.opendaylight.yangtools.yang.parser.stmt.reactor.ModifierImpl jdk.internal.loader.ClassLoaders$Ap	25,192	0 %	806,144	0 %	≈ 8,466,496	1 %
org.opendaylight.yangtools.yang.parser.stmt.reactor.ModifierImpl$1 jdk.internal.loader.ClassLoaders$	25,192	0 %	403,072	0 %	≈ 403,072	0 %
org.opendaylight.yangtools.yang.parser.stmt.reactor.ModifierImpl$AddedToNamespace jdk.internal.lo	25,130	0 %	804,160	0 %	≈ 9,247,840	1 %
org.opendaylight.yangtools.yang.parser.stmt.reactor.ModifierImpl$PhaseMutation jdk.internal.loader.C	25,130	0 %	804,160	0 %	≈ 804,160	0 %
org.opendaylight.yangtools.yang.parser.stmt.reactor.StatementDefinitionContext jdk.internal.loader.Cl	2,863	0 %	68,712	0 %	≈ 71,480	0 %
org.opendaylight.yangtools.yang.parser.stmt.reactor.ModifierImpl$PhaseFinished jdk.internal.loader.Cl	62	0 %	1,984	0 %	≈ 24,800	0 %
org.opendaylight.yangtools.yang.parser.stmt.reactor.ModifierImpl$PhaseModificationInNamespacePath	62	0 %	2,480	0 %	≈ 3,968	0 %
org.opendaylight.yangtools.yang.parser.stmt.reactor.RootStatementContext jdk.internal.loader.ClassLo	35	0 %	3,360	0 %	≈ 1,944,968	0 %
org.opendaylight.yangtools.yang.parser.stmt.reactor.SourceSpecificContext jdk.internal.loader.ClassLo	35	0 %	1,680	0 %	≈ 359,672	0 %
org.opendaylight.yangtools.yang.parser.stmt.reactor.SimpleNamespaceContext jdk.internal.loader.Cla	24	0 %	768	0 %	≈ 768	0 %
org.opendaylight.yangtools.yang.parser.stmt.reactor.CrossSourceStatementReactor jdk.internal.loade	3	0 %	72	0 %	≈ 17,064	0 %
org.opendaylight.yangtools.yang.parser.stmt.reactor.SourceSpecificContext$PhaseCompletionProgress	3	0 %	72	0 %	≈ 72	0 %
org.opendaylight.yangtools.yang.parser.stmt.reactor.BuildGlobalContext jdk.internal.loader.ClassLoade	1	0 %	64	0 %	753,150,392	54 %
org.opendaylight.yangtools.yang.parser.stmt.reactor.CrossSourceStatementReactor$BuildAction jdk.in	1	0 %	24	0 %	753,150,416	54 %
org.opendaylight.yangtools.yang.parser.stmt.reactor.EffectiveSchemaContext jdk.internal.loader.Class	1	0 %	40	0 %	7,208	0 %
org.opendaylight.yangtools.yang.parser.stmt.reactor.SourceSpecificContext$PhaseCompletionProgress	1	0 %	32	0 %	32	0 %
org.opendaylight.yangtools.yang.parser.stmt.reactor.StatementContextBase$$Lambda$95.949581868	1	0 %	16	0 %	128	0 %
org.opendaylight.yangtools.yang.parser.stmt.reactor.StatementMap$Empty jdk.internal.loader.ClassLo	1	0 %	16	0 %	16	0 %
org.opendaylight.yangtools.yang.parser.stmt.reactor.StatementMap$Regular$1 jdk.internal.loader.Cla	1	0 %	32	0 %	72	0 %

Since inference has completed, there should be no outstanding actions: here we have some listeners still hanging i

RESOLVED

Robert Varga

Unable to validate new line separated key-arg nodes

Problem with parsing list's key statement argument where nodes are separted by new line character, like in this exa

list gen-test-session-allocation {

key "entity-name
test-instance-name
untagged
pcp
color";

Error message:

```
Caused by: org.opendaylight.yangtools.yang.parser.spi.source.SourceException:
test-instance-name
untagged
pcp
color' [at vendor/ciena/ciena-sat.yang:1854:6]
```

By YANG ABNF Grammar [2]:

key-arg = node-identifier *(sep node-identifier)
sep = 1*(WSP / line-break)

By RFC6020 list's key Statement documentation [3]:

The "key" statement, which MUST be present if the list represents
configuration, and MAY be present otherwise, takes as an argument a
string that specifies a space-separated list of leaf identifiers of
this list.

[1] - https://github.com/YangModels/yang/blob/master/vendor/ciena/ciena-sat.yang#L1854

[2] - https://tools.ietf.org/html/rfc6020#section-12

[3] - https://tools.ietf.org/html/rfc6020#section-7.8.2

<table>
<tr><th></th><th></th><th></th><th></th></tr>
<tr><td>RESOLVED</td><td>Robert Varga</td><td>Unable to use prefix in augmention when argument inside submodule</td><td>Problem with using prefix inside a when argument. This happens only inside a submodule trying to augment nodes

Example [1]:<pre>
submodule ietf-snmp-community {
  belongs-to ietf-snmp {
    prefix snmp;
  }
 ...
  augment /snmp:snmp/snmp:target {
    when "snmp:v1 or snmp:v2c";
    ...
  }
}
</pre>Error message:<pre>
Caused by: org.opendaylight.yangtools.yang.parser.spi.source.SourceException:
yang:192:4]
...
Caused by: javax.xml.xpath.XPathExpressionException: java.lang.IllegalArgument
</pre>[1]- https://github.com/YangModels/yang/blob/ac6e52f0574d8db59d3ca61f52b339f4a49ea09b/experimental/ietf-ex</td></tr>
<tr><td>RESOLVED</td><td>Robert Varga</td><td>Data codecs may mis-enforce String length constraints</td><td>RFC7950 states that:<pre>
 A "length" statement restricts the number of Unicode characters in
 the string.
</pre>This is distinct diffrent what java.lang.String.length() provides:<pre>
Returns the length of this string. The length is equal to the number of Unicod
</pre>with the following (from java.lang.Character):<pre>
In the Java SE API documentation, Unicode code point is used for character val
and Unicode code unit is used for 16-bit char values that are code units of th
</pre>We should not be checking the String.length(), but rather String.codePointCount(). Audit all enforcement points for</td></tr>
<tr><td>RESOLVED</td><td>Ivan Hrasko</td><td>QName translation fails for identity leaf value</td><td><b>The background:</b>
Identity leafs are special because their value is QName that needs to be translated too.

<b>The problem:</b>
When we try to translate identity leaf with QNameTransformingStreamWriter the node identifier for identity leaf is tr

<b>Expected behavior:</b>
We expect the value of identity leaf to be translated, too.</td></tr>
</table>

90 issues

MD-SAL

Status	Assignee	Summary	Description
RESOLVED	Robert Varga	Binding Query - Interface MatchBuilderPath doesn't support matching leafs with boolean value	Current interface MatchBuilderPath not support matching leaf with Boolean type.
RESOLVED	Robert Varga	Binding Query - Result from method lessThan and lessThanOrEquals are switched	Result from method lessThan and lessThanOrEquals are switched. Same for method greaterThan and greaterThanOrEqual.

RESOLVED	Robert Varga	Binding Query - Failed to execute object() method in QueryResult.Item	Binding Query result data failed after executing object() method at QueryResult.Item instance.

3 issues