

# Table Type Patterns: Lithium: Release Notes

## Contents

- [Major Features](#)
- [Target Environment](#)
- [Known Issues and Limitations](#)
- [Bugs Fixed in this Release](#)
- [Compatibility and Migration from Previous Releases](#)
- [Deprecated, End of Lived, and/or Retired Features/API](#)

## Major Features

- **TTP Command Line Tools (new in Lithium):** Allow for reading, parsing, verifying, printing and analyzing TTPs as read from JSON via an executable CLI jar file.
- **TTP Model:** A YANG model for TTPs.
- **TTP Database:** A database of available TTPs using the YANG model.
- **TTP Switch Augmentation:** An augmentation for nodes in OpenDaylight to allow them to have zero-or-more supported TTPs and zero-or-one active TTP.

## Target Environment

As per the general Lithium release, i.e., JDK7 for development and JRE7 for execution.

## Known Issues and Limitations

- The TTP YANG model does not match the ONF TTP JSON precisely. Exact details are documented in the [TTP model YANG file](#).

## Bugs Fixed in this Release

- None. No reported bugs.

## Compatibility and Migration from Previous Releases

- The TTP model is the same as in the Helium release so any migration and/or tools should continue to work.
- To migrate actual data, it is recommended that users back up their current TTP-related data, upgrade and then restore it like so:
  1. Do an HTTP GET to: <http://localhost:8181/restconf/config/onf-ttp:opendaylight-ttps/onf-ttp:table-type-patterns/>
  2. Store the result
  3. After upgrading do an HTTP PUT with the stored result and the same URL
- You can use the same process to backup and restore any data associated with a switch, but the exact URL will vary depending on the switch and it will have to be done on a switch-by-switch basis.

## Deprecated, End of Lived, and/or Retired Features/API

- None.