

OpenLDAP Architecture

Refactoring ABIs

Howard Chu March 21, 2003

hyc@symas.com

Objectives: Why Change?

- Reduce Redundant Functions
- Streamline Calling Sequences
- Enhance Callback Functionality

Reduce Redundancy

- `bi_op_attribute` and `bi_op_group`
 - Both share a lot of code in common
 - Frequently require the same bug fixes
 - `backend_group` is just `backend_attribute` with one additional check
 - Replace with `bi_op_entry`, get more functionality at the same time

Streamline Calling Sequences

- ~ 20% of functions in slapd take Backend, Connection, Operation arguments
 - 58 of 305 declarations in proto-slap.h
- All of the backend operations take <ditto>
- They are needed together so frequently, they are essentially one parameter
- Consolidating into Operation ensures they are all always available when needed

Enhance Callbacks

- Callbacks don't have direct access to request parameters, must re-parse ber
- Again, multiple related arguments for results
 - Conn, Op, err, matched, text, Controls
- Callback nesting eats up stack for arguments
- Consolidating parameters into Operation provides more functionality while improving execution efficiency

Other Considerations

- Backend layering
 - Reimplement back-meta as an engine that can be overlaid over any other backend – suffixalias done right, among other things
 - Overlay back-ldap over any other backend – handle chaining instead of returning references to client

Conclusions

- Improve maintainability
 - Eliminate redundant code, only need to fix bugs once
- Improve efficiency and scalability
- Enables additional functionality with minimal overhead