# Namespace synchronization with the Common Name Library

**Jeff Thompson (**jefft0@remap.ucla.edu**)**, Jeff Burke (jburke@remap.ucla.edu)

## Motivation

The Common Name Library (CNL) is an experimental new API for NDN applications. Built on top of the lower-level Interest/Data exchange primitives of the Common Client Libraries (CCL), the CNL maintains an in-memory abstraction of the application's namespace. The application can attach specialized handlers to nodes of the CNL namespace object, for example to treat part of the name tree as segmented content, or to do data encryption/decryption. The CNL can also alert the application when new names are added to the namespace or when content is attached to a namespace node, whether by receiving a Data packet from the network, retrieving from a repo, or assembling the result of segmented content.

Dataset synchronization (SYNC) is an efficient approach to sharing and updating namespaces in NDN and an alternative to using Interest selectors for name discovery. It enables applications to exchange and update local copies of the names in a namespace, and then perform namespace searches locally. This functionality is a good fit for the Common Name Library where a background synchronize process could discover new names and add them to the CNL namespace object. The application can be notified of new names, or it can directly search the namespace as a local in-memory operation independent of network queries. We propose a Hackathon project to add a new CNL handler to synchronize announcements of new names in a namespace among participants.

## Contribution to NDN

This project would contribute an API for applications to automatically synchronize names and demonstrate name discovery without needing Interest selectors. We will demonstrate automatically retrieving the latest version of a document. The hackathon project is in Python but the result, based on the Common Client Libraries, can easily be ported to C++, JavaScript and Java.

## Tasks

1. Familiarize participants with the Python implementation of the CNL, [PyCNL](https://github.com/named-data/PyCNL).
2. Parallel tasks:
   1. Make a stub client application to gets CNL notifications of a new name, fetch and display it.
   2. Make a ChronoSync-based protocol to synchronize a names and update the CNL namespace.
3. Merge tasks 2a and 2b into the final demo to display the latest version of a document from a producer.
4. Time permitting: Use the CNL namespace to "time travel" to earlier versions.

## Required Knowledge for Participants

Required: Python, basic NDN networking

Useful: ChronoSync

## Expected Outcome

* The ChronoSync-based name synchronization protocol
* A handler for the Common Name Library (Python) that implements it
* A demo application to sync the latest version of a document