Disobedience Redesign

disorder_eclient_ API changes

The separate error callback will be eliminated.

Instead, separate error behavior for each operation will be defined. Usually this will be calling the standard callback with some error indicator. Therefore for any operation, both success and failure will be reported to the same place.

This will eliminate Disobedience's complicated and ad-hoc logic for redistributing error information back out to the right places.

Internal Event Notification

There will be a generalized event reporting system. This will be fed by from the server log but we will also synthesize additional events to report e.g. switching to a different notebook tab.

```
/** @brief Signature for event handlers
 * @param event Event type
 * @param eventdata Event-specific data
 * @param callbackdata Handler-specific data (as passed to event_register())
 */
typedef void event_handler(const char *event,
                           void *eventdata,
                           void *callbackdata);

/** @brief Handle identifying an event monitor */
typedef struct event_data *event_handle;

/** @brief Register an event handler
 * @param event Event type to handle
 * @param callback Function to call when event occurs
 * @param callbackdata Passed to @p callback
 * @return Handle for this registration (for use with event_cancel())
 */
event_handle event_register(const char *event,
                            event_handler *callback,
                            void *callbackdata);

/** @brief Stop handling an event
 * @param handle Registration to cancel (as returned from event_register())
 */
void event_cancel(event_handle handle);

/** @brief Raise an event
 * @param event Event type to raise
```

```
 * @param eventdata Event-specific data
 */
void event_raise(const char *event,
                 void *eventdata);
```

Events are identified by name (not by the value of the pointer).

We'll have a convenience function to associate an event handle with a Gtk+ widget, to make it easy to cancel them all when the widget is deleted.

This code will be in `lib`, to allow for the server to use it in the future.

Specific Bugs

- ~~The UI should more easily allow AF_UNIX connection to be used.~~
- Keyboard events should do something worthwhile in more places; e.g.:
  - ~~enter should work in most popups~~
  - ~~typing in the Choose tab should adjust the search string~~
- Controls you can't use should more reliably be insensitive
  - e.g. `remote_userman no` should make Users menu item insensitive
  - ~~controls you don't have rights for should be insensitive~~