C API for PAPI SDE

```
typedef long long int (*papi_sde_fptr_t)( void * );
typedef int (*sde_cmpr_fptr_t)( const void *p1, const void *p2 );
typedef void * papi_handle_t;

papi_handle_t papi_sde_init( const char *library_name );
int papi_sde_register_counter( papi_handle_t handle, const char *ev_name, int mode, int type, void *cntr );
int papi_sde_register_fp_counter( papi_handle_t handle, const char *ev_name, int mode, int type,
                                  papi_sde_fptr_t func_ptr, const void *param );
int papi_sde_unregister_counter( void *handle, const char *ev_name );
int papi_sde_describe_counter( papi_handle_t handle, const char *ev_name, const char *ev_descr );
int papi_sde_add_counter_to_group( papi_handle_t handle, const char *ev_name, const char *grp_name,
                                   uint32_t grp_flags );
int papi_sde_create_counter( papi_handle_t handle, const char *ev_name, int mode, void **cntr_handle );
int papi_sde_inc_counter( void *cntr_handle, long long int increment );
int papi_sde_create_recorder( papi_handle_t handle, const char *ev_name, size_t typesize,
                              sde_cmpr_fptr_t cmpr_func, void **record_handle );
int papi_sde_record( void *record_handle, size_t typesize, const void *value );
int papi_sde_reset_recorder(void *record_handle );
int papi_sde_reset_counter( void *cntr_handle );
void *papi_sde_get_counter_handle( papi_handle_t handle, const char *ev_name);
papi_handle_t papi_sde_hook_list_events( papi_sde_fptr_struct_t *fptr_struct);
```

C++ API for PAPI SDE

```
namespace papi_sde {
    class PapiSde {
        PapiSde(const char *name_of_library);
        template <typename T> int register_counter(const char *event_name, int cntr_mode, T &counter );
        template <typename T, typename P>
        int register_fp_counter(const char *event_name, int cntr_mode, T (*func_ptr)(P*), P const &param);
        int unregister_counter(const char *event_name );
        int describe_counter(const char *event_name, const char *event_description );
        int add_counter_to_group(const char *event_name, const char *group_name, uint32_t group_flags );
        CreatedCounter *create_counter(const char *event_name, int cntr_mode);
        Recorder *create_recorder(const char *event_name, size_t typesize,
                                  int (*cmpr_func_ptr)(const void *p1, const void *p2));
    }
    class PapiSde::Recorder {
        Recorder(papi_handle_t sde_handle, const char *event_name, size_t typesize,
                 int (*cmpr_func_ptr)(const void *p1, const void *p2));
        template <typename T> int record(T const &value);
        int reset(void);
    };
    class PapiSde::CreatedCounter {
        CreatedCounter(papi_handle_t sde_handle, const char *event_name, int cntr_mode);
        template <typename T> int increment(T const &increment);
        int reset(void);
    };
    PapiSde::CreatedCounter &operator+=(PapiSde::CreatedCounter &X, const long long incr);
    PapiSde::CreatedCounter &operator++(PapiSde::CreatedCounter &X);
    PapiSde::CreatedCounter &operator++(PapiSde::CreatedCounter &X, int);
    PapiSde::CreatedCounter &operator--(PapiSde::CreatedCounter &X);
    PapiSde::CreatedCounter &operator--(PapiSde::CreatedCounter &X, int );
}
```