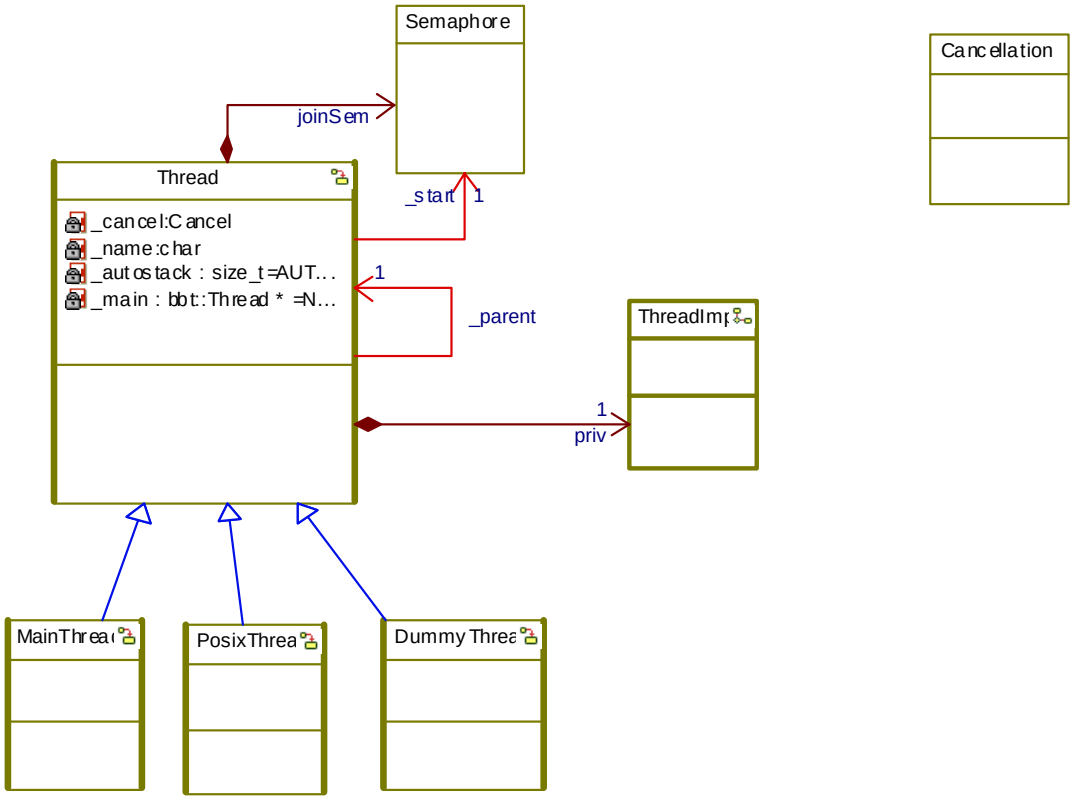
bbt

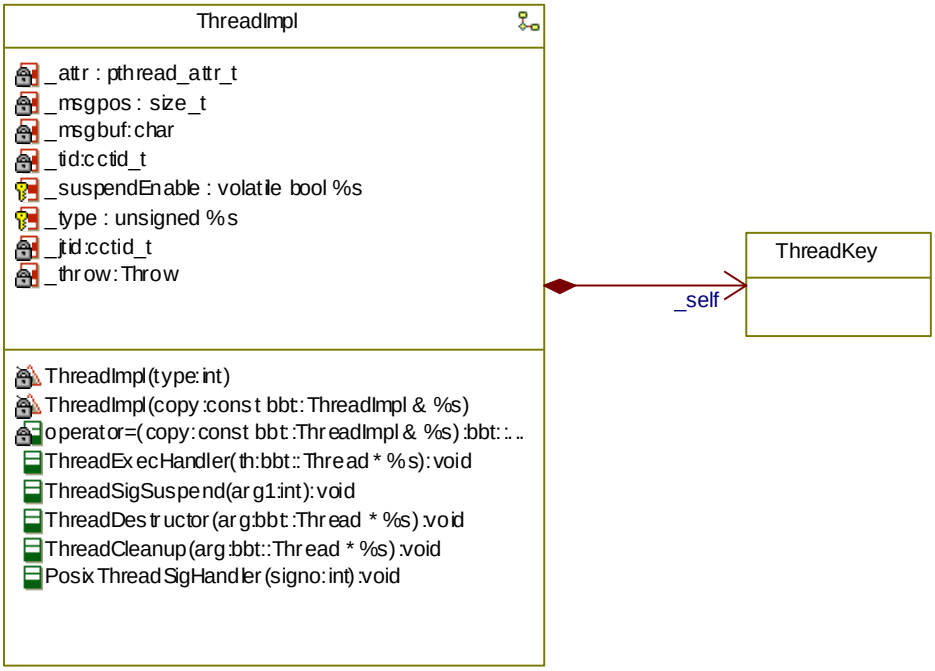
OBJECT MODEL DIAGRAMS:

releation

thread_obj



ThreadImpl



Thread

detach

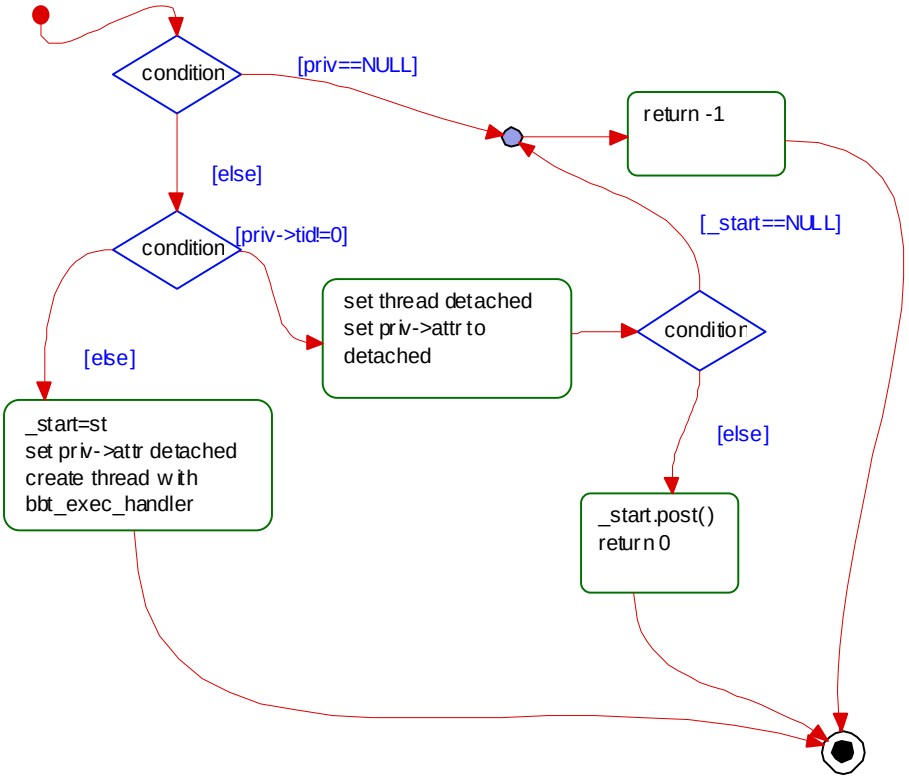
```
*
* Start a new thread as "detached".  This is an alternative
* start() method that resolves some issues with later glibc
* implimentations which incorrectly impliment self-detach.
*
* @return error code if execution fails.
* @param start optional starting semaphore to alternately use.
```

Primitive-operation , Public, Return type is int

Args:

'bbt::Semaphore * st = 0

Activity Diagram



resume

```
*
* Resumes execution of the selected thread.
```

Triggered-operation , Public, Return type is void

start

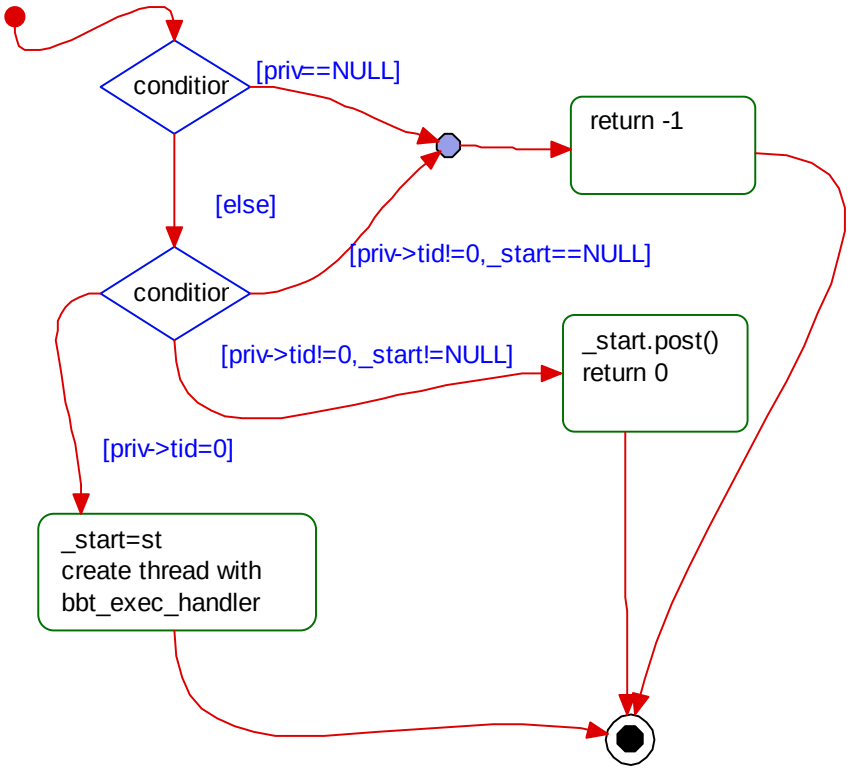
```
*
* When a new thread is created, it does not begin immediate
* execution.  This is because the derived class virtual tables
* are not properly loaded at the time the C++ object is created
* within the constructor itself, at least in some compiler/system
* combinations.  The thread can either be told to wait for an
* external semaphore, or it can be started directly after the
* constructor completes by calling the start() method.
*
* @return error code if execution fails.
* @param start optional starting semaphore to alternately use.
```

Primitive-operation , Public, Return type is int

Args:

'bbt::Semaphore * st = 0

Activity Diagram



suspend

```
*
* Suspends execution of the selected thread.  Pthreads do not
* normally support suspendable threads, so the behavior is
* simulated with signals.  On systems such as Linux that
* define threads as processes, SIGSTOP and SIGCONT may be used.
```

Triggered-operation , Public, Return type is void

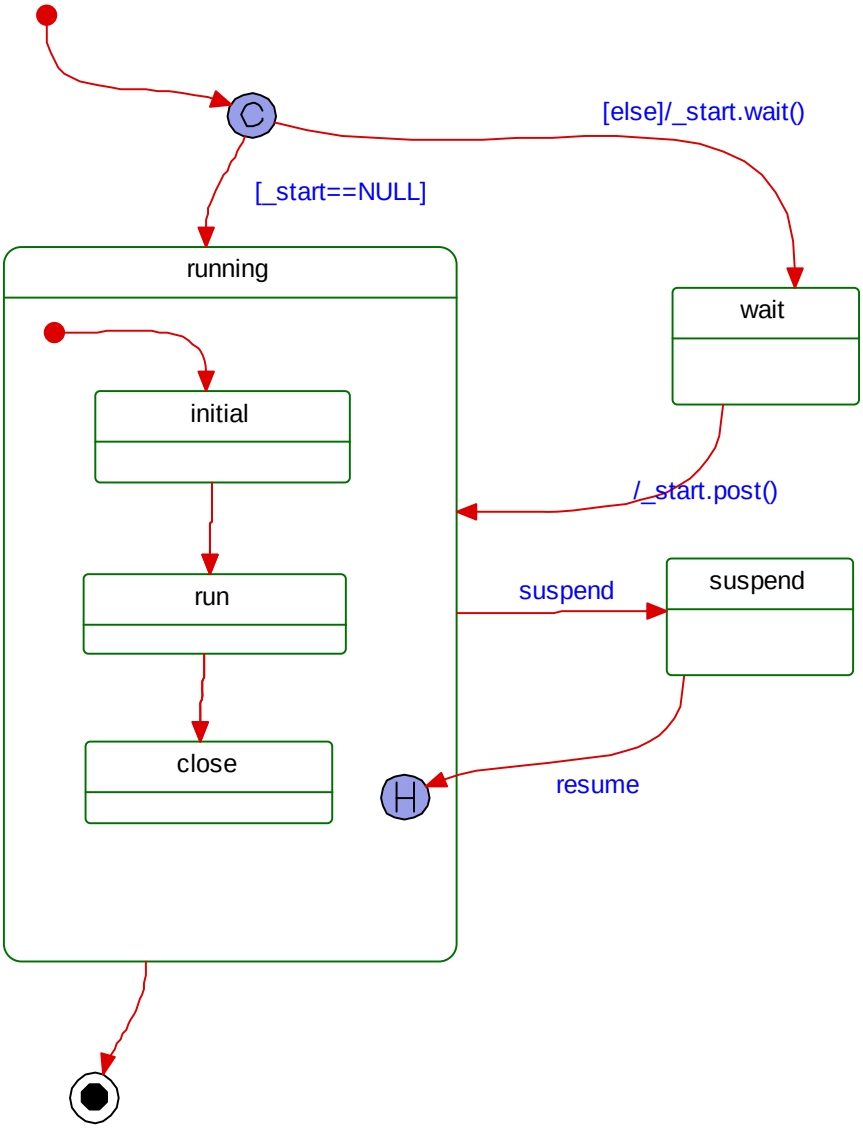
```
* thread.  The existing thread's properties (cancel mode, priority,
* etc), are also duplicated.
*
* @param th currently executing thread object to clone.
* @todo implement in win32
```

Constructor , Public

Args:

'const bbt::Thread & %s' th

Statechart



ThreadImpl

ThreadExecHandler

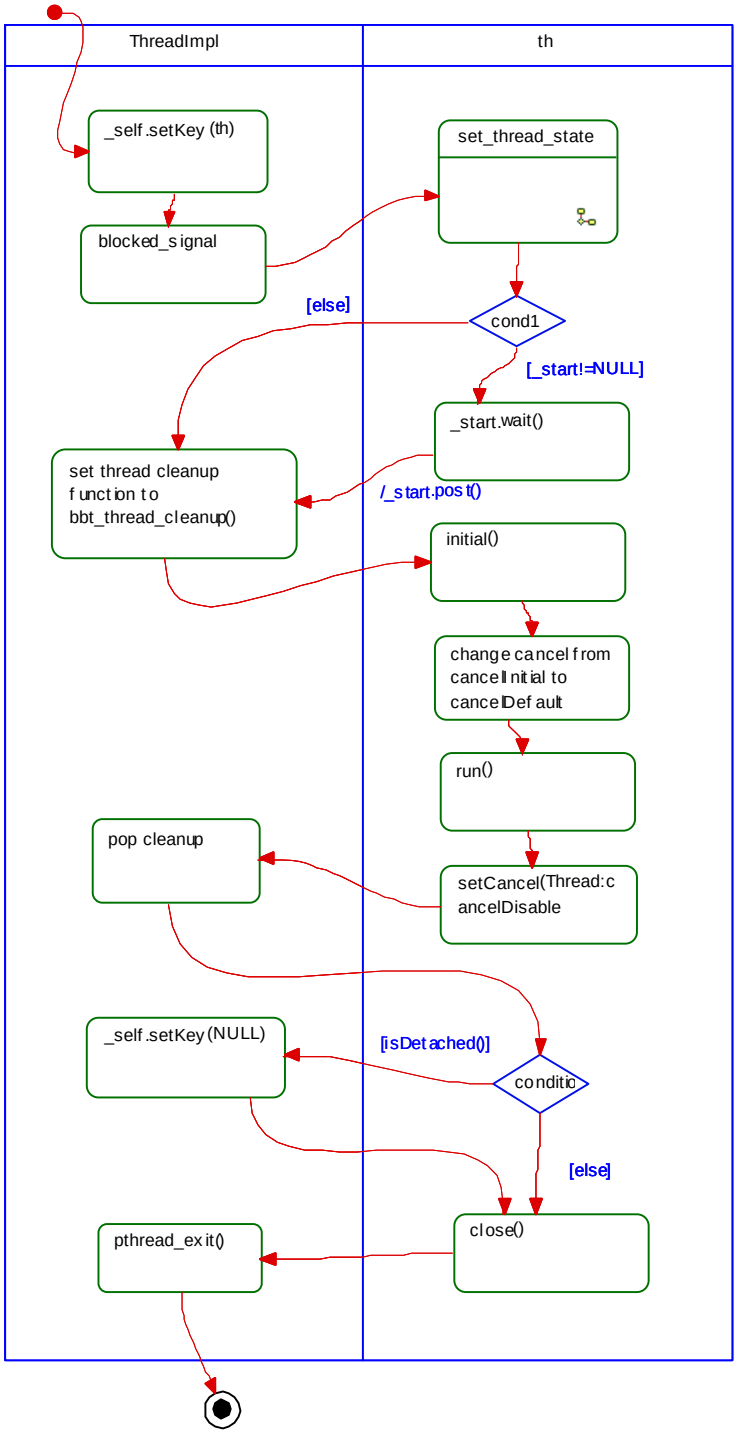
C binding functions

Primitive-operation , Public, Static, Return type is void

Args:

'bbt::Thread * th

Activity Diagram



Activity Diagram of set_thread_state