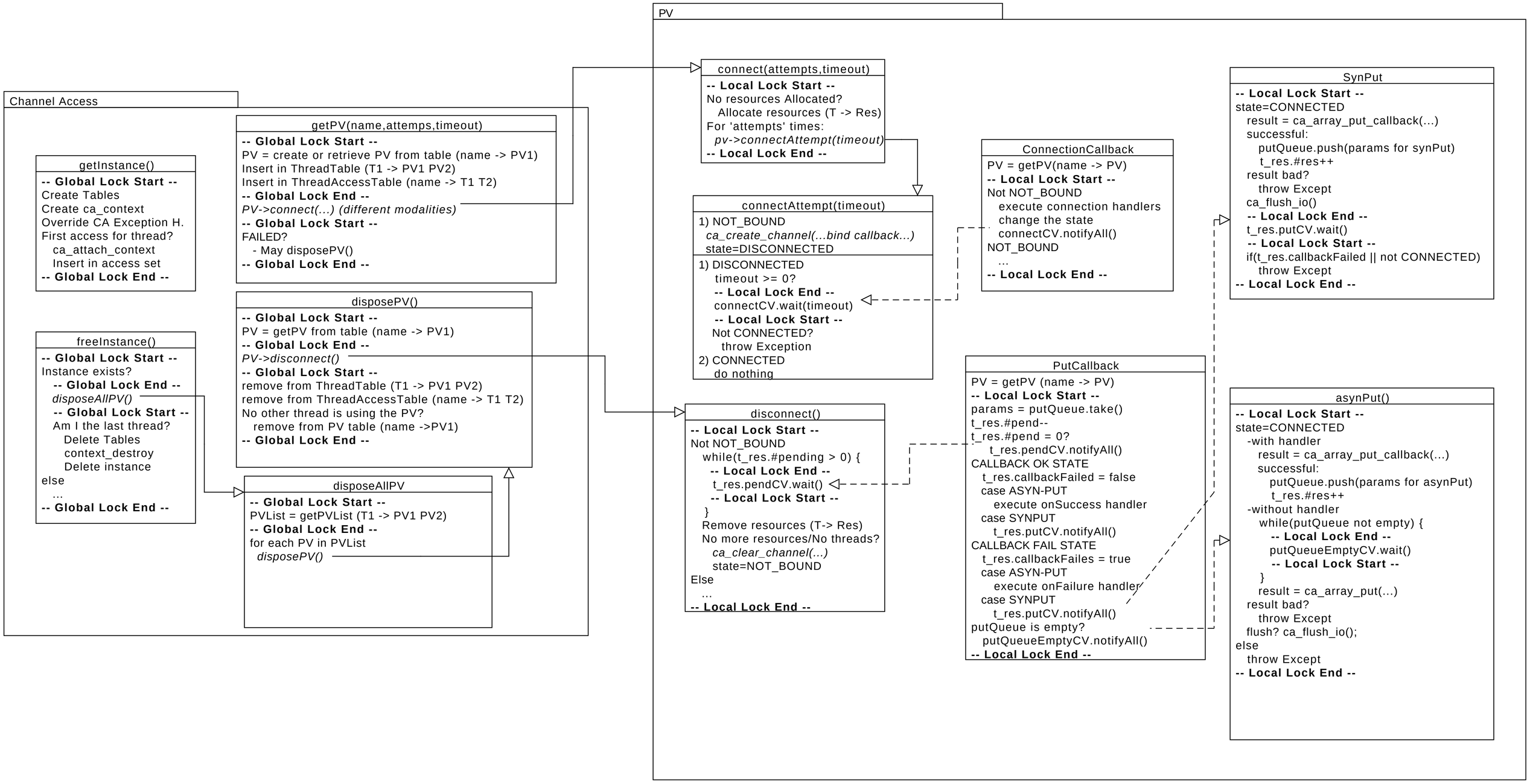
Channel Access
getInstance()
-- Global Lock Start --
Create Tables
Create ca_context
Override CA Exception H.
First access for thread?
ca_attach_context
Insert in access set
-- Global Lock End --
freeInstance()
-- Global Lock Start --
Instance exists?
-- Global Lock End --
disposeAllPV()
-- Global Lock Start --
Am I the last thread?
Delete Tables
context_destroy
Delete instance
else
...
-- Global Lock End --
getPV(name,attemps,timeout)
-- Global Lock Start --
PV = create or retrieve PV from table (name -> PV1)
Insert in ThreadTable (T1 -> PV1 PV2)
Insert in ThreadAccessTable (name -> T1 T2)
-- Global Lock End --
PV->connect(...) (different modalities)
-- Global Lock Start --
FAILED?
- May disposePV()
-- Global Lock End --
disposePV()
-- Global Lock Start --
PV = getPV from table (name -> PV1)
-- Global Lock End --
PV->disconnect()
-- Global Lock Start --
remove from ThreadTable (T1 -> PV1 PV2)
remove from ThreadAccessTable (name -> T1 T2)
No other thread is using the PV?
remove from PV table (name ->PV1)
-- Global Lock End --
disposeAllPV
-- Global Lock Start --
PVList = getPVList (T1 -> PV1 PV2)
-- Global Lock End --
for each PV in PVList
disposePV()
PV
connect(attempts,timeout)
-- Local Lock Start --
No resources Allocated?
Allocate resources (T -> Res)
For 'attempts' times:
pv->connectAttempt(timeout)
-- Local Lock End --
connectAttempt(timeout)
1) NOT_BOUND
ca_create_channel(...bind callback...)
state=DISCONNECTED
1) DISCONNECTED
timeout >= 0?
-- Local Lock End --
connectCV.wait(timeout)
-- Local Lock Start --
Not CONNECTED?
throw Exception
2) CONNECTED
do nothing
disconnect()
-- Local Lock Start --
Not NOT_BOUND
while(t_res.#pending > 0) {
-- Local Lock End --
t_res.pendCV.wait()
-- Local Lock Start --
}
Remove resources (T-> Res)
No more resources/No threads?
ca_clear_channel(...)
state=NOT_BOUND
Else
...
-- Local Lock End --
ConnectionCallback
PV = getPV(name -> PV)
-- Local Lock Start --
Not NOT_BOUND
execute connection handlers
change the state
connectCV.notifyAll()
NOT_BOUND
...
-- Local Lock End --
PutCallback
PV = getPV (name -> PV)
-- Local Lock Start --
params = putQueue.take()
t_res.#pend--
t_res.#pend = 0?
t_res.pendCV.notifyAll()
CALLBACK OK STATE
t_res.callbackFailed = false
case ASYN-PUT
execute onSuccess handler
case SYNPUT
t_res.putCV.notifyAll()
CALLBACK FAIL STATE
t_res.callbackFailes = true
case ASYN-PUT
execute onFailure handler
case SYNPUT
t_res.putCV.notifyAll()
putQueue is empty?
putQueueEmptyCV.notifyAll()
-- Local Lock End --
SynPut
-- Local Lock Start --
state=CONNECTED
result = ca_array_put_callback(...)
successful:
putQueue.push(params for synPut)
t_res.#res++
result bad?
throw Except
ca_flush_io()
-- Local Lock End --
t_res.putCV.wait()
-- Local Lock Start --
if(t_res.callbackFailed || not CONNECTED)
throw Except
-- Local Lock End --
asynPut()
-- Local Lock Start --
state=CONNECTED
-with handler
result = ca_array_put_callback(...)
successful:
putQueue.push(params for asynPut)
t_res.#res++
-without handler
while(putQueue not empty) {
-- Local Lock End --
putQueueEmptyCV.wait()
-- Local Lock Start --
}
result = ca_array_put(...)
result bad?
throw Except
flush? ca_flush_io();
else
throw Except
-- Local Lock End --