

common_library

目次 [Table of contents]

- 目次 [Table of contents]
- 図表目次 [Table of figures]
- common_library [common_library]
 - 機能概要 [Functional overview]
 - 機能詳細 [Function detail]
 - ソフトウェア構成図 [software block]
 - ユースケースとAPI一覧 [Use case and API lists]
 - 外部要因 ユースケース一覧 [External factor use-case lists]
 - 内部処理 エラーユースケース一覧 [Internal processing error use-case lists]
 - プロセス属性設定[set process property]
 - 概要 [Overview]
 - シーケンス [Sequence]
 - プロセス生成[create process]
 - 概要 [Overview]
 - シーケンス [Sequence]
 - Cgroup[Cgroup]
 - 概要 [Overview]
 - シーケンス [Sequence]
 - Lock[Lock]
 - 概要 [Overview]
 - シーケンス [Sequence]
 - スレッド[thread]
 - 概要 [Overview]
 - シーケンス [Sequence]
 - セマフォ[semaphore]
 - 概要 [Overview]
 - シーケンス [Sequence]
 - Monitor[Monitor]
 - 概要 [Overview]
 - シーケンス [Sequence]
 - Tls[Tls]
 - 概要 [Overview]
 - シーケンス [Sequence]
 - Region[Region]
 - 概要[Overview]
 - シーケンス[Sequence]

図表目次 [Table of figures]

- 表.ユニット概要
- 図.ソフトウェア構成図 [Figure. Software structure diagram]
 - 表.外部要因ユースケース一覧 [Table. External factor use case list]
 - 表.内部処理エラーユースケース一覧 [Table. Internal processing error use case list]

common_library [common_library]

機能概要 [Functional overview]

本機能は、プロセスの管理機能の共通処理を提供する。

This unit provides function of process management

表.ユニット概要

ユニット名[Unit Name]	コンポーネント名[Component Name]	ユニット概要[Description]	オーナーディレクトリ[Owner Dir]
------------------	--------------------------	---------------------	-----------------------

common_library	NativeService	This function provides common processing for process management functions.	common_library
----------------	---------------	--	----------------

機能詳細 *[Function detail]*

本機能のプロセス管理処理について、主たる利用手順を以下に示す。

The main functions of process management are as follows.

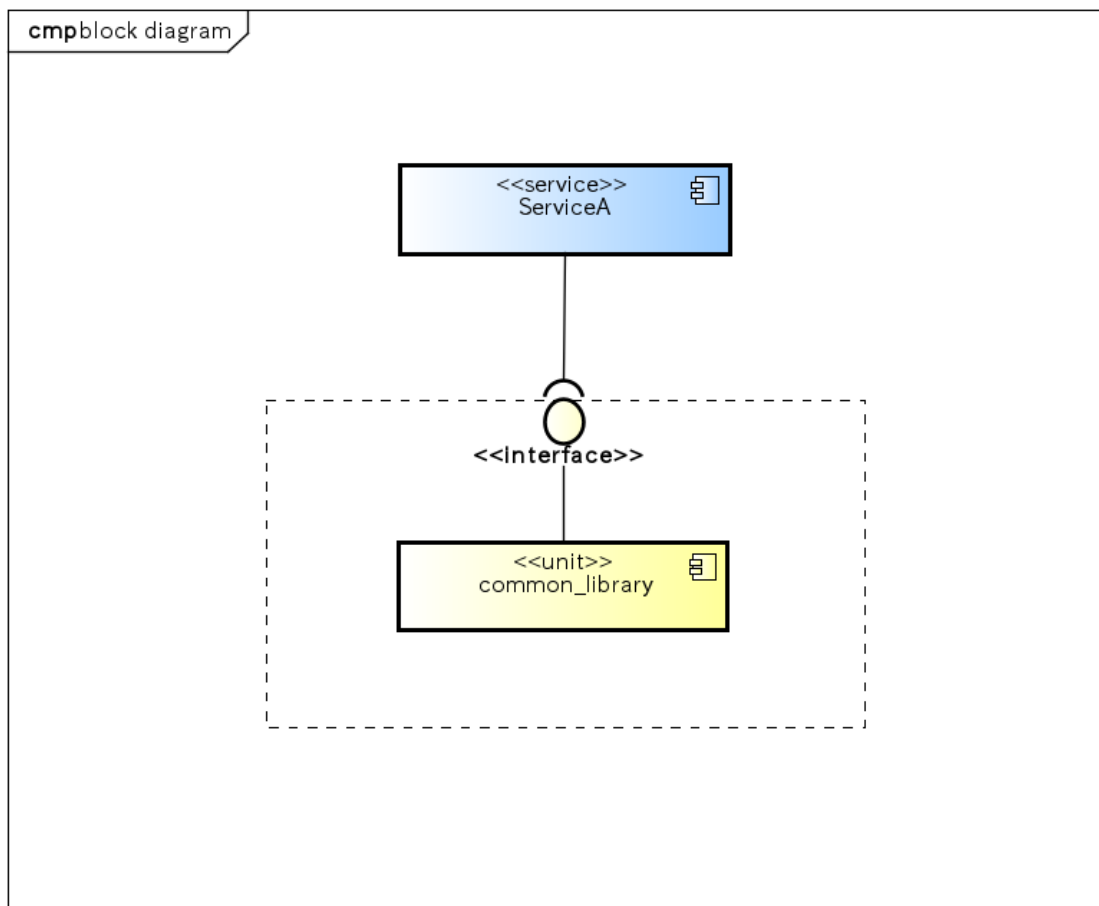
1. プロセス属性を設定する。
2. プロセス属性からプロセスを生成する。
3. TIsにプロセスNo.を登録する。
4. Monitorにプロセス情報を登録する。
5. プロセスグループを生成する。
6. 生成したプロセスをグループに登録する。
7. プロセス間の排他にミューテックス、セマフォを使用する。
8. スレッドを生成する。

1. *Set process property*
2. *Create process based on process property*
3. *Register process number to TIs*
4. *Register process information to Monitor*
5. *Create process loop*
6. *Register process to process group*
7. *Use mutex and semaphore way to realize exclusive process*
8. *Create thread*

ソフトウェア構成図 *[software block]*

common_libraryのブロック図を以下に示す。

block diagram of common_library is as follows.



powered by Astah

図.ソフトウェア構成図 [Figure. Software structure diagram]

ユースケースとAPI一覧 [Use case and API lists]

Usecases and API lists of common_library are as follows.

外部要因 ユースケース一覧 [External factor use-case lists]

表.外部要因ユースケース一覧 [Table. External factor use case list]

ユースケース番号 [use-case number]	機能カテゴリ [function category]	ユースケース名 [usecase name]	ユースケースを実現する API [APIs that realize the usecase]	備考 [note]

common_library_Process_001	Process	プロセス属性設定 <i>set process property</i>	CL_ProcessCreateAttrInit CL_ProcessCreateAttrSetCgroup CL_ProcessCreateAttrSetCpuAssign CL_ProcessCreateAttrSetDisableCloseFds CL_ProcessCreateAttrSetGid CL_ProcessCreateAttrSetGroup CL_ProcessCreateAttrSetHoldFds CL_ProcessCreateAttrSetName CL_ProcessCreateAttrSetSchedule CL_ProcessCreateAttrSetStackSize CL_ProcessCreateAttrSetUid	
common_library_Process_002		プロセス生成 <i>create process</i>	CL_ProcessCleanup CL_ProcessCreate CL_ProcessInit CL_ProcessTerminate	
common_library_Process_003		Cgroup <i>Cgroup</i>	CL_ProcessCreate CL_ProcessCreateCgroupAttrInit CL_ProcessCreateCgroupAttrSetCfsBandwidthControl CL_ProcessCreateCgroupAttrSetCpuShares CL_ProcessCreateCgroupAttrSetMemoryLimit CL_ProcessCreateCgroupAttrSetMemoryUsageNotification CL_ProcessCreateCgroupAttrSetRtThrottling CL_ProcessCreateCgroupClassify CL_ProcessCreateCgroupCreate CL_ProcessCreateCgroupDelete CL_ProcessInit ProcessTerminateGroup	
common_library_Lock_001	Lock	Lock <i>Lock</i>	CL_LockGet CL_LockMap CL_LockProcessInit CL_LockRelease CL_LockSystemInit CL_LockUnmap	
common_library_Thread_001	Thread	スレッド <i>thread</i>	CL_ThreadCreate CL_ThreadCreateAttrInit CL_ThreadCreateAttrSetName	
common_library_Sem_001	Sem	セマフォ <i>semaphore</i>	CL_SemWait	
common_library_Monitor_001	Monitor	Monitor <i>monitor</i>	CL_MonitorGetEntry CL_MonitorInit CL_MonitorSearchDestroy CL_MonitorSearchInit CL_MonitorSearchTimeout CL_MonitorSetEntry	
common_library_Tls_001	Tls	Tls <i>Tls</i>	CL_TlsGet CL_TlsInit	
common_library_Region_001	Region	Region <i>Region</i>	CL_RegionCreate CL_RegionDestroy CL_RegionAlloc CL_RegionFree CL_RegionCleanupAdd	

内部処理 エラーユースケース一覧 *[Internal processing error use-case lists]*

表.内部処理エラーユースケース一覧 [Table. Internal processing error use case list]

エラー番号 [error number]	機能カテゴリ [function category]	ユースケース名 [use-case name]	戻り値 [return value]	備考 [notes]
common_library_Process_Error_001	Process	プロセス生成	-1	
common_library_Lock_Error_001	Lock	Lock	EINVAL	

プロセス属性設定 [set process property]

概要 [Overview]

Client側で確保した領域にプロセスの属性の設定値を設定する。

※ CL_ProcessCreateAttrInit()をコールすることで領域の初期化を実施できる。

(注意) この時点では、Client側で確保した領域に設定されるだけなのでプロセスの属性が更新されるわけではない。CL_ProcessCreate ()でClient側で確保した領域をポインタで指定しプロセスの属性が反映される。

Set process property to the area which is secured by Client.

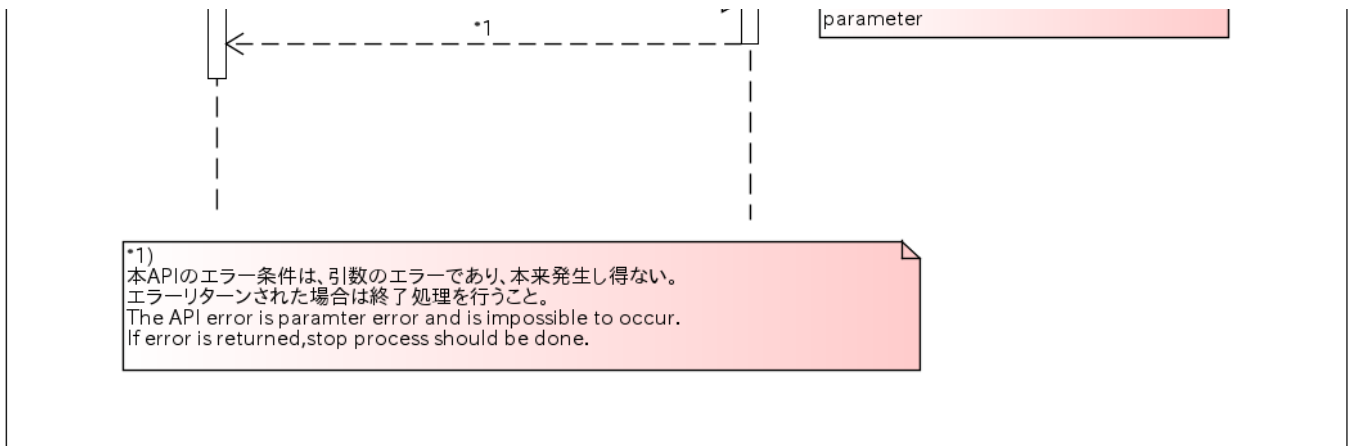
※ Initialize the area by calling CL_ProcessCreateAttrInit() function.

(note)At this time,process property is only set to Client area, but hasn't been updated.

Call CL_ProcessCreate () to specify the area which is secured by Client as a pointer and then process property is reflected.

シーケンス [Sequence]





powered by Astah

プロセス生成[create process]

概要 [Overview]

プロセスを生成する。

create process.

引数のプロセスの属性には、プロセス属性設定APIを使用して設定した。

Client側で確保した領域のポインタを指定する。

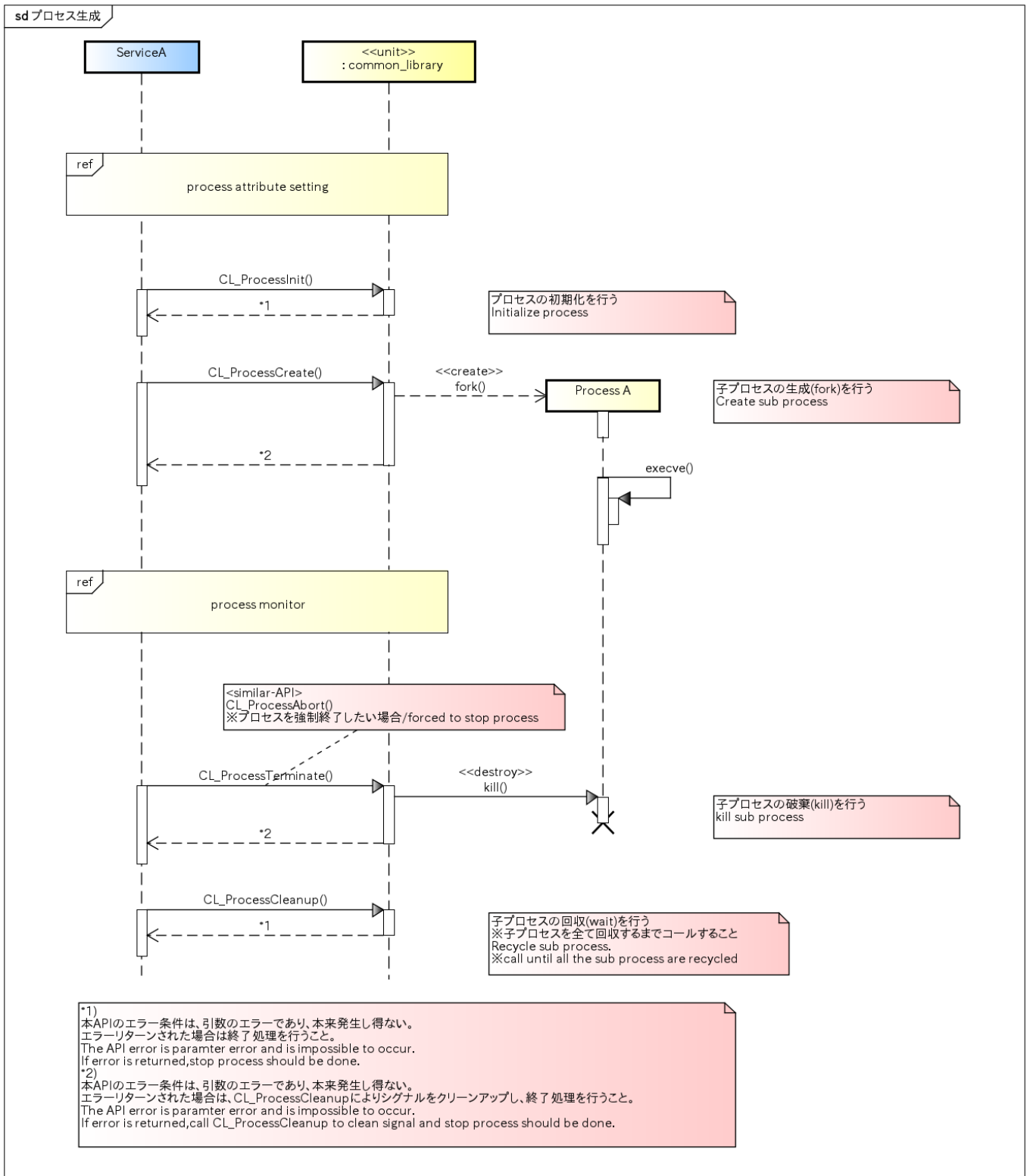
生成した子プロセスを開放する場合はCL_ProcessCleanup()をコールする必要がある。CL_ProcessCleanup()の戻り値が1（成功：子プロセスあり）の間、繰り返しコールする必要がある。

Use process property set API to set process property.

Specify pointer of data area which is secured by Client.

Call CL_ProcessCleanup() If users want to open process which is successfully created.

シーケンス [Sequence]



powered by Astah

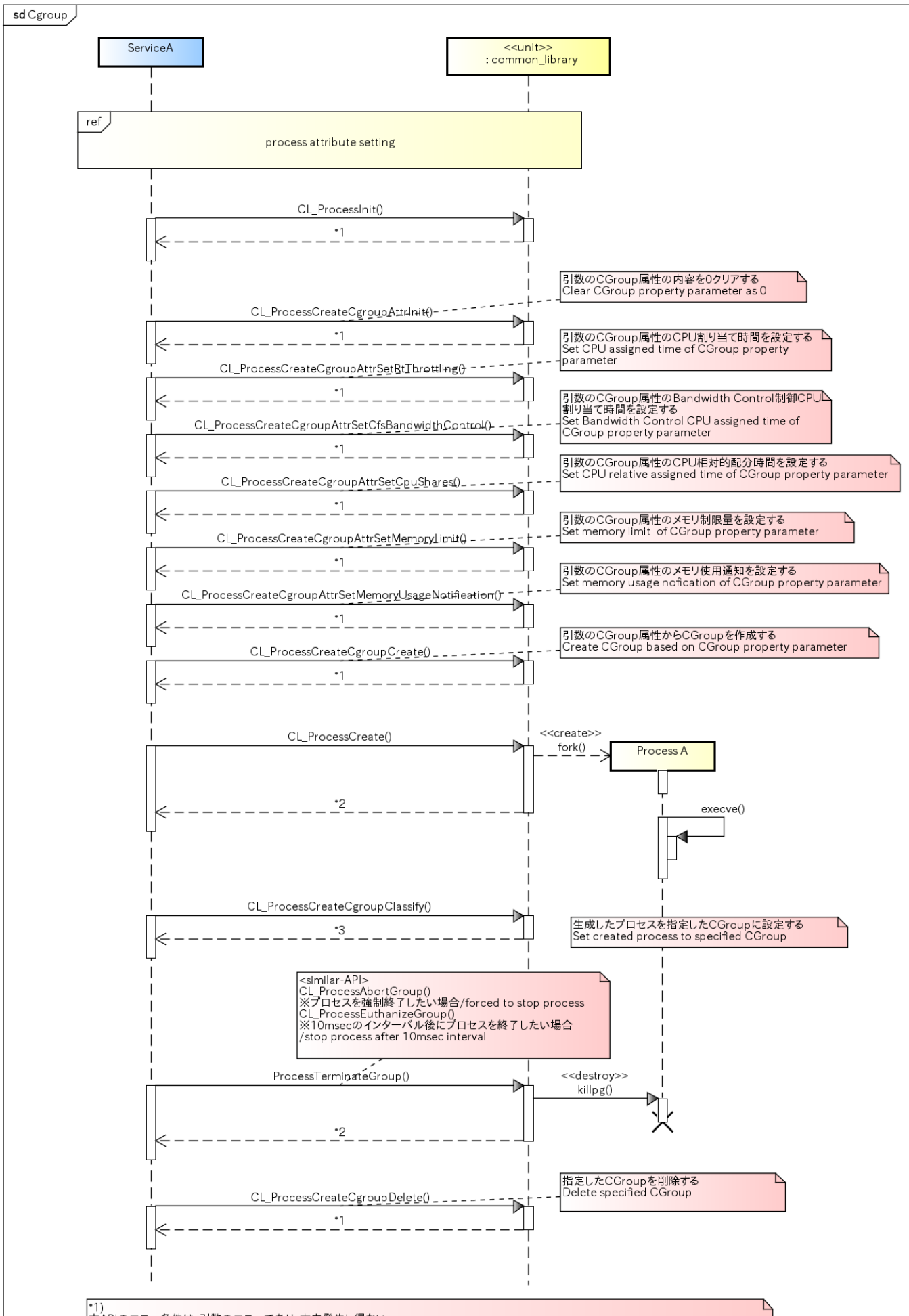
Cgroup[Cgroup]

概要 [Overview]

本処理は、生成したプロセスをグループ化する機能を提供する。

This function can make the created processes as a group.

シーケンス [Sequence]



本APIのエラー条件は、引数のエラーであり、本来発生し得ない。
エラーリターンされた場合は終了処理を行うこと。
The API error is paramter error and is impossible to occur.
If error is returned, stop process should be done.

*2)
本APIのエラー条件は、引数のエラーであり、本来発生し得ない。
エラーリターンされた場合は、CL_ProcessCreateCgroup Delete()によりグループファイルを削除し、終了処理を行うこと。
The API error is paramter error and is impossible to occur.
If error is returned, call CL_ProcessCreateCgroup Delete() to delete group file and stop process should be done.

*3)
本APIのエラー条件は、引数のエラーであり、本来発生し得ない。
エラーリターンされた場合は、ProcessTerminateGroup()により子プロセスを終了、CL_ProcessCreateCgroup Delete()によりグループファイルを削除し、終了処理を行うこと。
The API error is paramter error and is impossible to occur.
If error is returned, call ProcessTerminateGroup() to stop sub process and call CL_ProcessCreateCgroup Delete() to delete group file then stop process.

powered by Astah

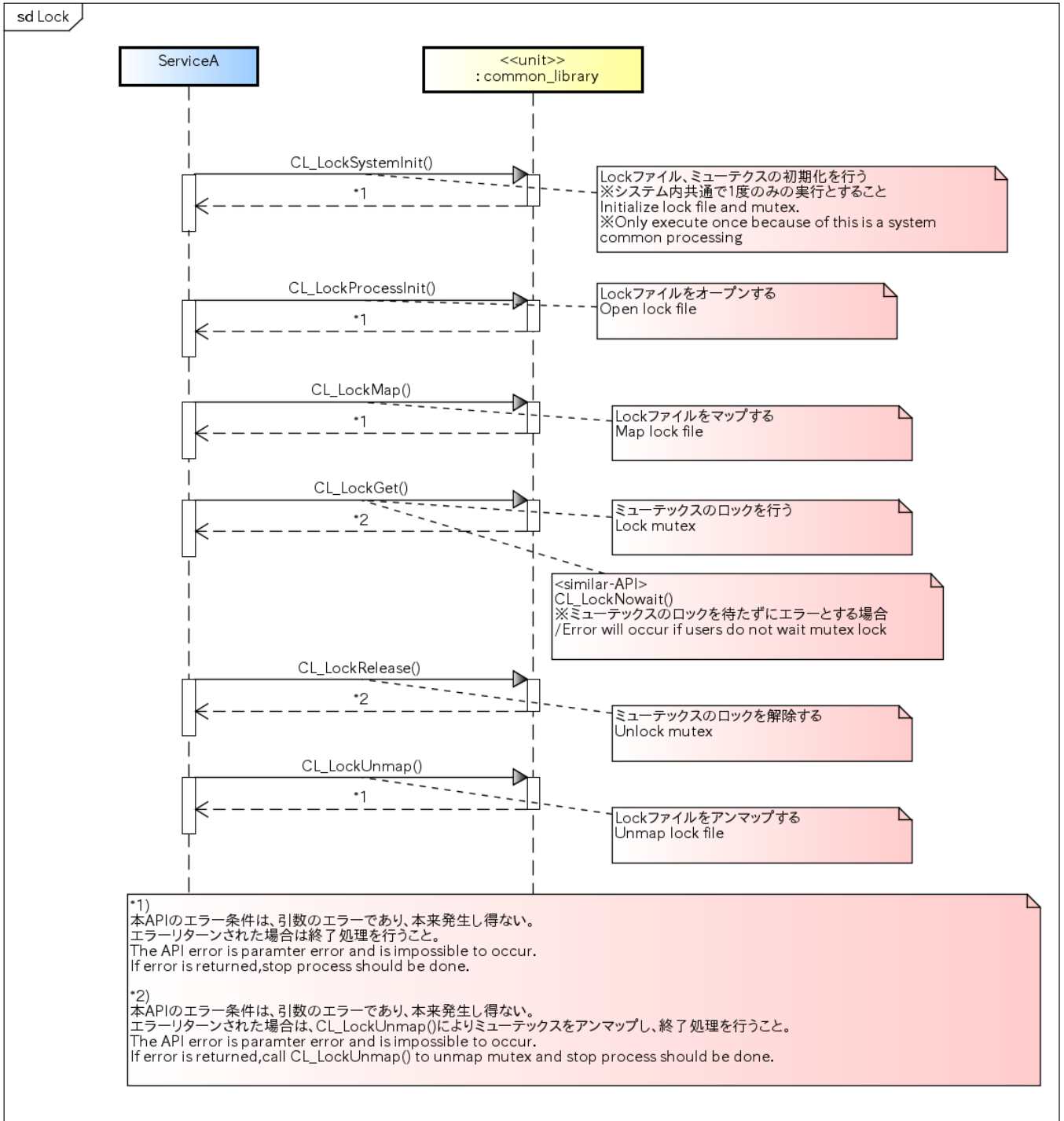
Lock[Lock]

概要 [Overview]

本処理は、ミューテックスによる排他制御機能を提供する。

Realize exclusive process function by mutex way

シーケンス [Sequence]



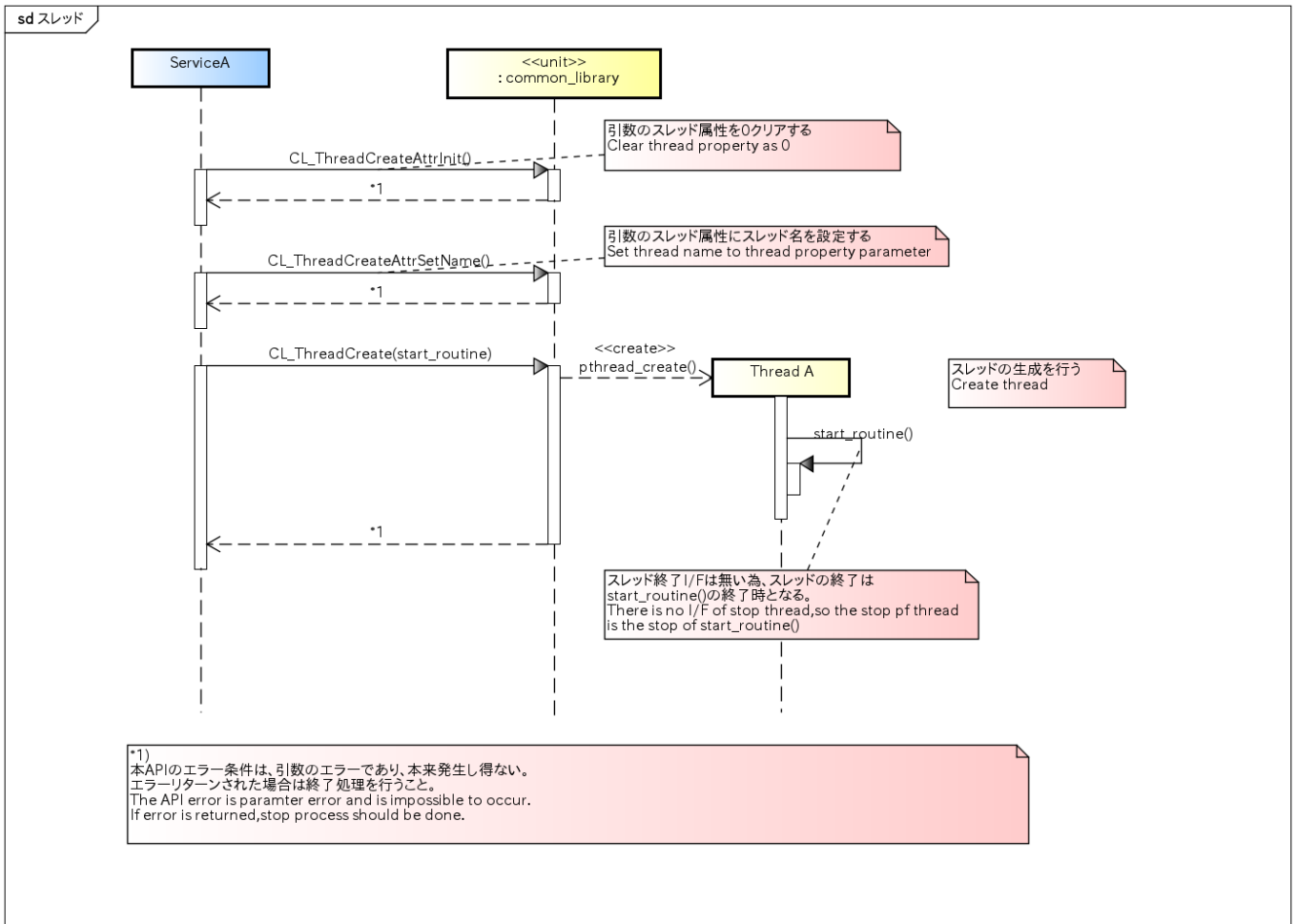
スレッド[thread]

概要 [Overview]

本処理は、pthread_create()によるスレッド生成機能を提供する。

Create thread with pthread_create().

シーケンス [Sequence]



powered by Astah

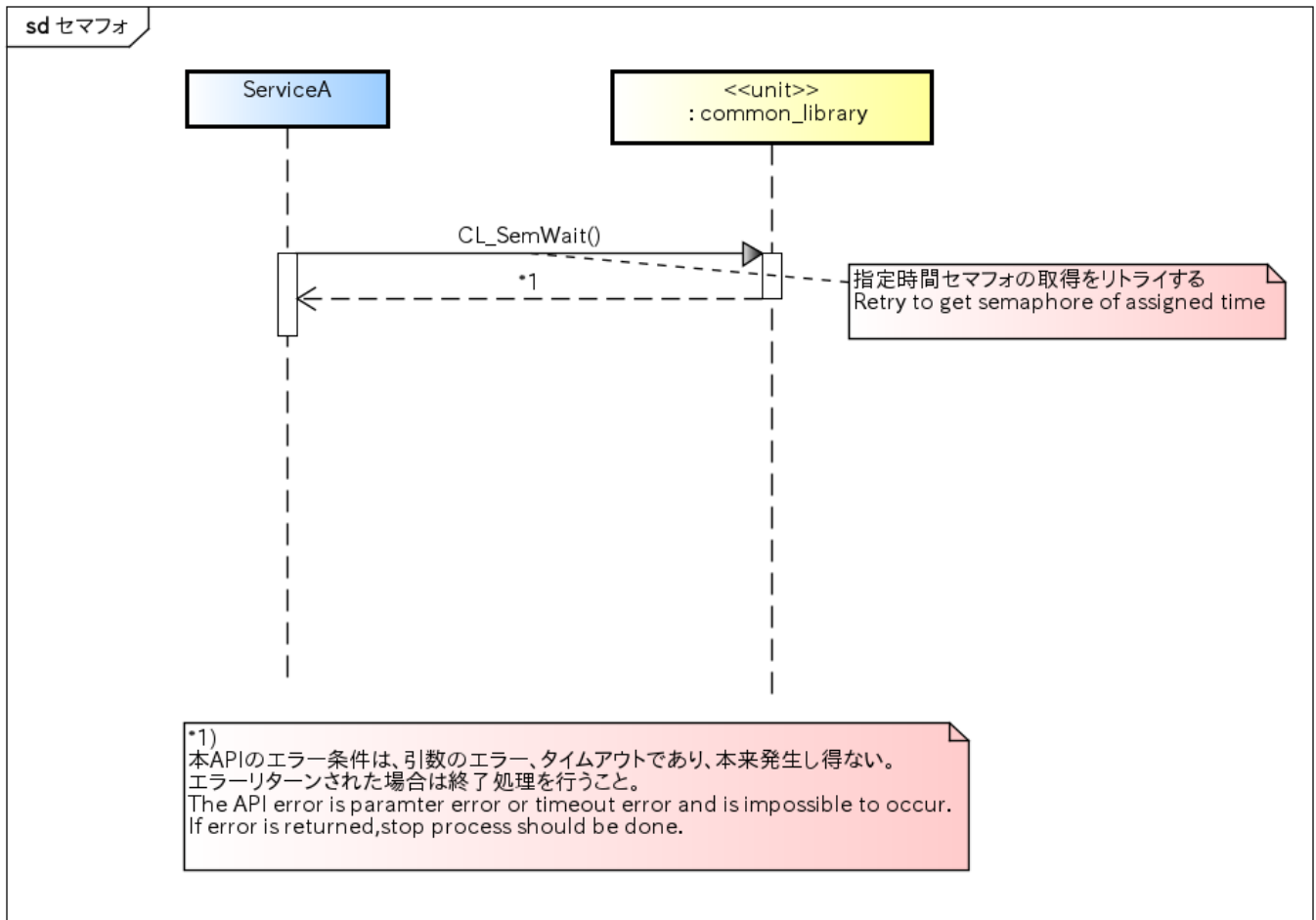
セマフォ [semaphore]

概要 [Overview]

本処理は、セマフォによる排他制御機能を提供する。

Realize exclusive process function by semaphore way

シーケンス [Sequence]



powered by Astah

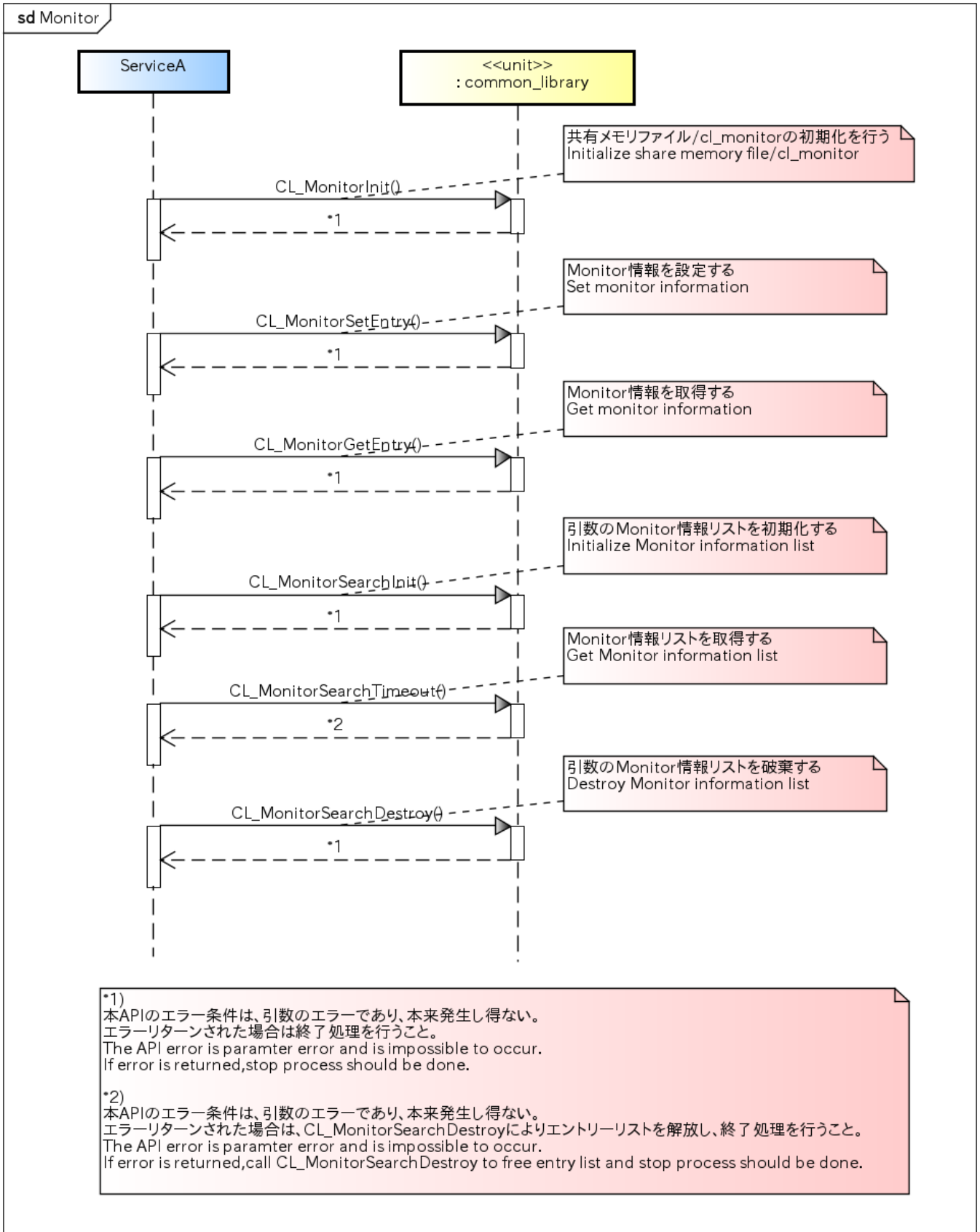
Monitor[Monitor]

概要 [Overview]

本処理は、各プロセスの状態を保持する機能を提供する。

Monitor provides function to maintain process status.

シーケンス [Sequence]

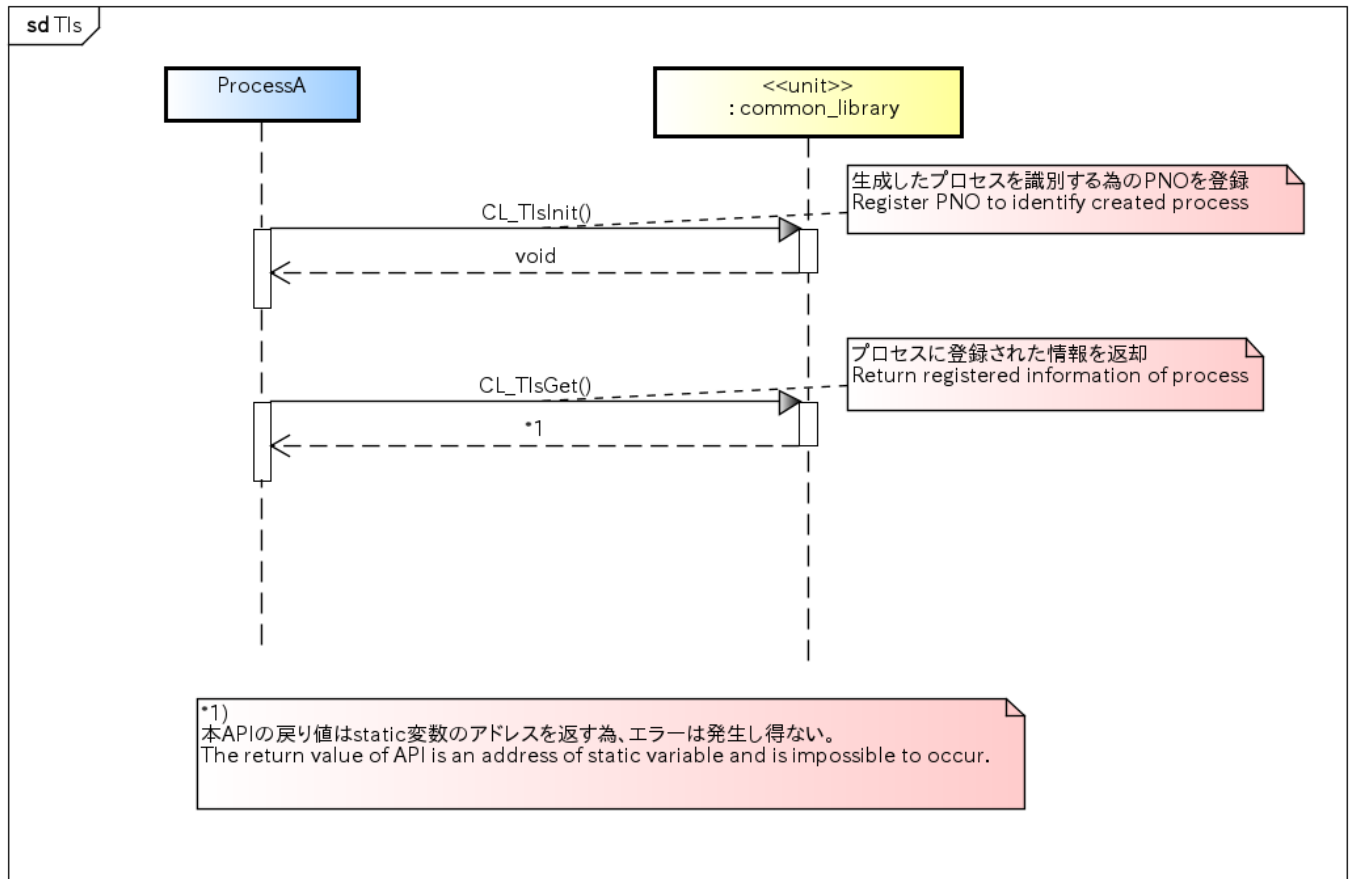


概要 [Overview]

本処理は、生成したプロセスの識別情報を制御する機能を提供する。

TIs provides function to control identification information of process that has been created.

シーケンス [Sequence]



powered by Astah

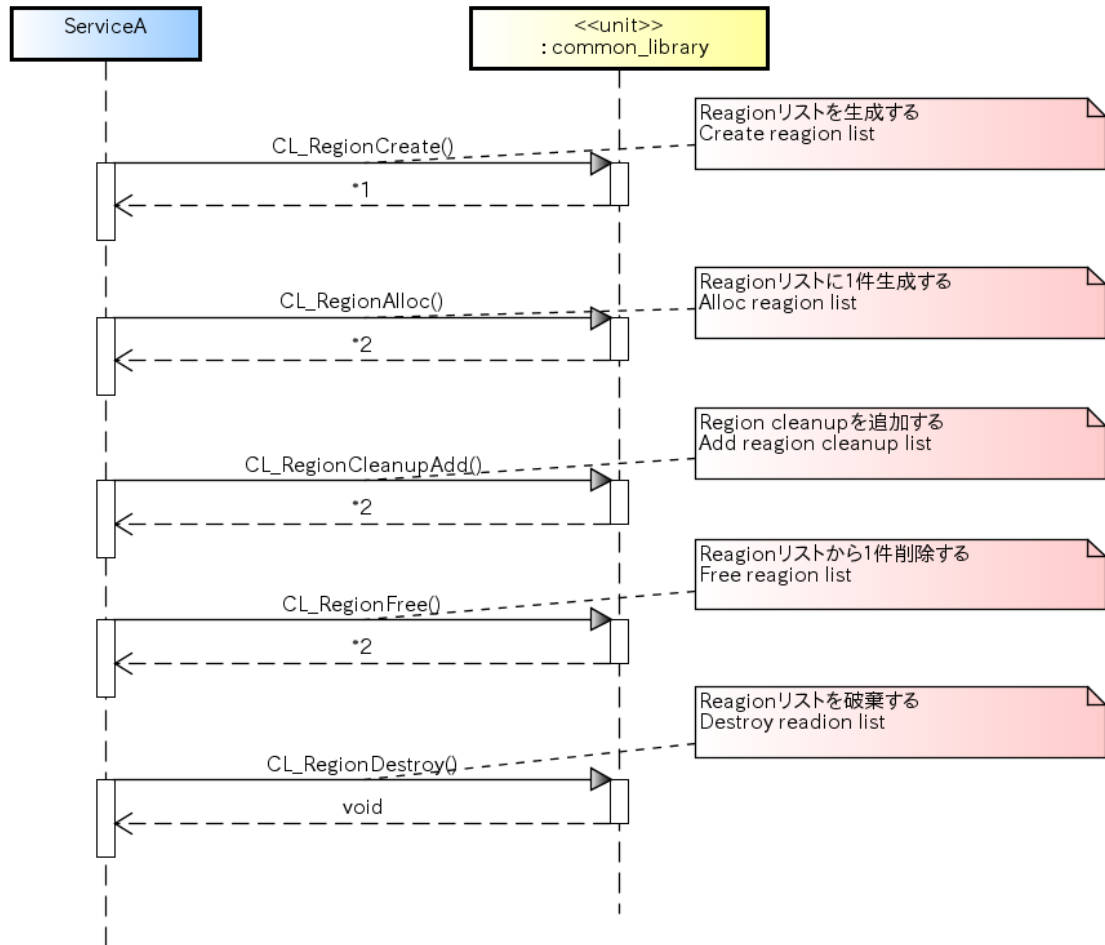
Region [Region]

概要 [Overview]

Regionの生成、割当、開放、削除機能を提供する。

Region provides function to maintain memory mapping create, allocate, free, destroy.

シーケンス [Sequence]



*1)
本APIのエラー条件は引数のエラー、もしくはヒープの枯渇であり、本来発生しない。
エラーリターンされた場合は終了処理を行うこと。
The API error is paramter error or heap full error and is impossible to occur.
If error is returned, stop process should be done.

*2)
本APIのエラー条件は引数のエラー、もしくはヒープの枯渇であり、本来発生しない。
エラーリターンされた場合は、CL_RegionDestroy()によりRegionオブジェクトを解放し、終了処理を行うこと。
The API error is paramter error or heap full error and is impossible to occur.
If error is returned, call CL_RegionDestroy() to free reigion object and stop process should be done