

event_library

目次 [\[Table of contents\]](#)

- 目次 [Table of contents]
- 図表目次 [Table of figures]
- event_library [event_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[outside factor use-case list]
 - 1, イベント機能概要 / Event function information.
 - 2, メッセージ機能概要 / Message function information.
 - 内部処理 エラーユースケース一覧 [internal processing error use-case list]
 - ユースケースevent_library_EventInitial_001 [use-case event_library_EventInitial_001]
 - 概要 [Overview]
 - シーケンス [Sequence]
 - ユースケースevent_library_EventInitial_002 [use-case event_library_EventInitial_002]
 - 概要 [Overview]
 - シーケンス [Sequence]
 - ユースケースevent_library_EventInitial_003 [use-case event_library_EventInitial_003]
 - 概要 [Overview]
 - シーケンス [Sequence]
 - ユースケースevent_library_EventInitial_004 [use-case event_library_EventInitial_004]
 - 概要 [Overview]
 - シーケンス [Sequence]
 - ユースケースevent_library_EventInitial_005 [use-case event_library_EventInitial_005]
 - 概要 [Overview]
 - シーケンス [Sequence]
 - ユースケースevent_library_EventInitial_006 [use-case event_library_EventInitial_006]
 - 概要 [Overview]
 - シーケンス [Sequence]
 - ユースケースevent_library_EventInitial_007 [use-case event_library_EventInitial_007]
 - 概要 [Overview]
 - シーケンス [Sequence]
 - ユースケースevent_library_EventInitial_008 [use-case event_library_EventInitial_008]
 - 概要 [Overview]
 - シーケンス [Sequence]
 - ユースケースevent_library_SendEvent_001 [use-case event_library_SendEvent_001]
 - 概要 [Overview]
 - シーケンス [Sequence]
 - ユースケースevent_library_SendEvent_002 [use-case event_library_SendEvent_002]
 - 概要 [Overview]
 - シーケンス [Sequence]
 - ユースケースevent_library_SendEvent_003 [use-case event_library_SendEvent_003]
 - 概要 [Overview]
 - シーケンス [Sequence]
 - ユースケースevent_library_SendEvent_004 [use-case event_library_SendEvent_004]
 - 概要 [Overview]
 - シーケンス [Sequence]
 - ユースケースevent_library_ReceiveEvent_001 [use-case event_library_ReceiveEvent_001]
 - 概要 [Overview]
 - シーケンス [Sequence]
 - ユースケースevent_library_ReceiveEvent_002 [use-case event_library_ReceiveEvent_002]
 - 概要 [Overview]
 - シーケンス [Sequence]
 - ユースケースevent_library_ReceiveEvent_003 [use-case event_library_ReceiveEvent_003]
 - 概要 [Overview]
 - シーケンス [Sequence]
 - ユースケースevent_library_ReceiveEvent_004 [use-case event_library_ReceiveEvent_004]
 - 概要 [Overview]
 - シーケンス [Sequence]
 - ユースケースevent_library_ReceiveEvent_005 [use-case event_library_ReceiveEvent_005]
 - 概要 [Overview]
 - シーケンス [Sequence]
 - ユースケースevent_library_ReceiveEvent_006 [use-case event_library_ReceiveEvent_006]
 - 概要 [Overview]
 - シーケンス [Sequence]
 - ユースケースevent_library_ReceiveEvent_007 [use-case event_library_ReceiveEvent_007]

- 概要 [Overview]
 - シーケンス [Sequence]
- ユースケースevent_library_ReceiveEvent_008 [use-case event_library_ReceiveEvent_008]
 - 概要 [Overview]
 - シーケンス [Sequence]
- ユースケースevent_library_ReceiveEvent_009 [use-case event_library_ReceiveEvent_009]
 - 概要 [Overview]
 - シーケンス [Sequence]
- ユースケースevent_library_DeleteEvent_001 [use-case event_library_DeleteEvent_001]
 - 概要 [Overview]
 - シーケンス [Sequence]
- ユースケースevent_library_DeleteEvent_002 [use-case event_library_DeleteEvent_002]
 - 概要 [Overview]
 - シーケンス [Sequence]
- ユースケースevent_library_MessageInitial_001 [use-case event_library_MessageInitial_001]
 - 概要 [Overview]
 - シーケンス [Sequence]
- ユースケースevent_library_MessageInitial_002 [use-case event_library_MessageInitial_002]
 - 概要 [Overview]
 - シーケンス [Sequence]
- ユースケースevent_library_MessageInitial_003 [use-case event_library_MessageInitial_003]
 - 概要 [Overview]
 - シーケンス [Sequence]
- ユースケースevent_library_MessageInitial_004 [use-case event_library_MessageInitial_004]
 - 概要 [Overview]
 - シーケンス [Sequence]
- ユースケースevent_library_SendMessage_001 [use-case event_library_SendMessage_001]
 - 概要 [Overview]
 - シーケンス [Sequence]
- ユースケースevent_library_SendMessage_002 [use-case event_library_SendMessage_002]
 - 概要 [Overview]
 - シーケンス [Sequence]
- ユースケースevent_library_ReceiveMessage_001 [use-case event_library_ReceiveMessage_001]
 - 概要 [Overview]
 - シーケンス [Sequence]
- ユースケースevent_library_ReceiveMessage_002 [use-case event_library_ReceiveMessage_002]
 - 概要 [Overview]
 - シーケンス [Sequence]
- ユースケースevent_library_ReceiveMessage_003 [use-case event_library_ReceiveMessage_003]
 - 概要 [Overview]
 - シーケンス [Sequence]
- ユースケースevent_library_ReceiveMessage_004 [use-case event_library_ReceiveMessage_004]
 - 概要 [Overview]
 - シーケンス [Sequence]
- ユースケースevent_library_ReceiveMessage_005 [use-case event_library_ReceiveMessage_005]
 - 概要 [Overview]
 - シーケンス [Sequence]
- ユースケースevent_library_ReceiveMessage_006 [use-case event_library_ReceiveMessage_006]
 - 概要 [Overview]
 - シーケンス [Sequence]
- ユースケースevent_library_ReceiveMessage_007 [use-case event_library_ReceiveMessage_007]
 - 概要 [Overview]
 - シーケンス [Sequence]
- ユースケースevent_library_ReceiveMessage_008 [use-case event_library_ReceiveMessage_008]
 - 概要 [Overview]
 - シーケンス [Sequence]
- ユースケースevent_library_DeleteMessage_001 [use-case event_library_DeleteMessage_001]
 - 概要 [Overview]
 - シーケンス [Sequence]
- ユースケースevent_library_DeleteMessage_002 [use-case event_library_DeleteMessage_002]
 - 概要 [Overview]
 - シーケンス [Sequence]

図表目次 [Table of figures]

- 表.ユニット概要
- 表.機能一覧
- 図. イベント機能詳細シーケンス
- 図. メッセージ機能詳細シーケンス
- 図.ソフトウェア構成図
 - 表.外部要因ユースケース一覧（イベント機能）
 - 表.外部要因ユースケース一覧（メッセージ機能）

event_library [event_library]

機能概要 [Functional overview]

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]
event_library	OtherService	<p>プロセス間（またはスレッド間）のメッセージの送信と受信を提供する。以下2種類の機能を提供する：</p> <p><i>Provides the sending and receiving of message between process, thread, according to the transfer data, there are two kind function as below:</i></p> <p>1, イベントの送信、受信（固定長ビット列(フラグ)を使用して通信を行う）</p> <p><i>Event's data 's send and receive (Communicate using fixed length bit string (flag)) .</i></p> <p>2, メッセージの送信、受信（可変長バイト列(メッセージ)を使用して通信を行う）</p> <p><i>Message's data send and receive. (Communicate using a variable-length record byte string (message)) .</i></p>	ev

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。

Please check below information about function process manager.

表. 機能一覧

#	種類 [Category]	機能 [Function]
1	イベント / <i>Event</i>	イベントの初期化 / <i>Event initial</i>
2		イベント送信 / <i>Send event</i>
3		イベント受信 / <i>Receive events</i>
4		イベントの削除 / <i>Delete event</i>
5	メッセージ / <i>Message</i>	メッセージの初期化 / <i>Message initial</i>
6		メッセージ送信 / <i>Send message</i>
7		メッセージ受信 / <i>Receive message</i>
8		メッセージの削除 / <i>Delete message</i>

イベントフラグ・メッセージイベント共に、同様の機能が存在する。

イベントフラグはメモリ枯渇通知など、メモリが制限状態である場合に使用する。

各ユースケースの詳細は、それぞれの機能項目に記載する。

.There are similar functions in Event flag and Message events.

.When memory is a limit state, use event flag. For example, notification of memory exhaustion.

The details of each use case is discribed in each function item.

①イベントフラグ

32ビットの固定長ビット列(フラグ)を使用して通信を行う。

フラグはシステム全体でユニークな「フラグID」で識別される。

フラグを取得するプログラムは、通信を行うためにフラグを作成する必要がある。

また、あらかじめフラグをセットする可能性のある他プログラムに対して、作成するフラグのフラグIDを知らせる必要がある。

知らせる方法として、決められた固定値を使用する、RPC-APIで通知する等の方法がある。

以下に、イベントフラグの使用例を記載する。

1. 利用者はフラグIDを使用してフラグを作成する。
2. 利用者はフラグの通知を受信するために、1で作成したフラグのFDを取得する。
3. 利用者は取得したFDに対し、poll/select等で通知を待つ。
4. フラグ送信側は、1で作成したフラグにビットパターンをセットする。

※1つのフラグに複数のビットパターンをセットする場合、"全てのビットパターンのOR演算結果のみ"をフラグに保持する。

5. 利用者はフラグIDを使用して、該当するフラグのビットパターンを取得する。

取得方法はノンブロッキング、ブロッキングの2通りあり、

ブロッキングの場合、フラグのビットパターンがセットされていなければ、フラグのビットパターンをセットするまでタスクをスリープする。

また、どちらの取得方法も、取得時にフラグに保持しているビットパターンをクリアする。

6. 利用者はフラグが未使用になった場合にフラグを削除する

①Event flag

Communicate using fixed length bit string of 32bit(flag).

Distinguish a flag in unique "flag ID" in the whole system.

The program to get a flag has to make a flag to communicate.

In addition, it is necessary to inform the flag ID of the flag to make beforehand, for other programs that may set a flag.

There are methods such as notifying in RPC-API or using fixed value as how to inform the flag ID.

Below, usage examples of Event flag are described.

1. *The user creates flag using flag ID.*
2. *In order to receive notification of flag, the user gets FD of flag created in 1.*
3. *The user waits for notification with poll/select etc, for gotten FD.*
4. *The sender sets bit pattern to flag created in 1.*
5. *The user gets bit pattern of flag corresponding to flag ID.*

There are methods "block" and "non-block" to gets bit pattern.

In the case of "block", if the bit pattern of the flag is not set, perform sleep of a task until bit pattern of the flag is set.

In addition, both methods clear bit pattern held in the flag.

6. When the flag becomes unused, the user delete the flag.

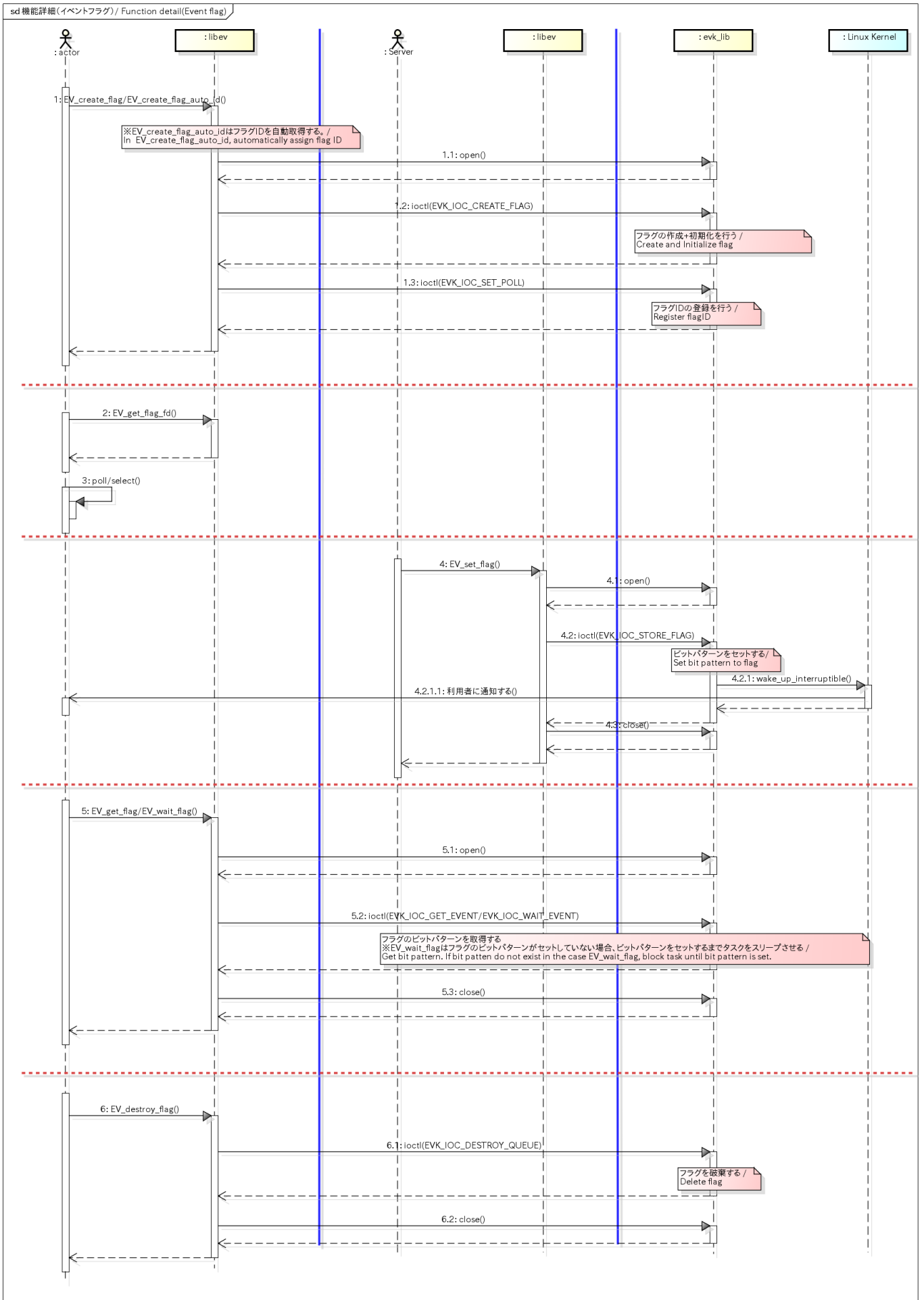


図. イベント機能詳細シーケンス

②メッセージイベント

可変長のバイト列(メッセージ) を使用して通信を行う。

メッセージはキューに対して送信し、キュー作成時に、メッセージキューの長さ（蓄積できるメッセージ数）、1メッセージあたりの最大バイト数、キューの種別を指定する。

キューの種別とは、キューがフルの状態ですらにイベントを受信した場合の処理の種別であり、次の3種類を規定する。

- メッセージ送信元にエラー (EV_ERR_Busy) を返す
- キューの先頭のメッセージを削除してシフトする(最も古いメッセージを削除する)
- キューの最後のメッセージを上書きする(最も新しいメッセージを上書きする)

キューはシステム全体でユニークな「キューID」で識別される。

メッセージを取得するプログラムは、通信を行うためにメッセージキューを作成する必要がある。

また、あらかじめメッセージを送信する可能性のある他プログラムに対して、作成するメッセージキューのキューIDを知らせる必要がある。

知らせる方法として、決められた固定値を使用する、RPC-APIで通知する等の方法がある。

以下に、メッセージイベントの使用例を記載する。

1. 利用者はキューIDを使用してメッセージキューを作成する。
2. 利用者はメッセージを受信するために、1で作成したメッセージキューのFDを取得する。
3. 利用者は取得したFDに対し、poll/select等で通知を待つ。
4. メッセージ送信側は、1で作成したメッセージキューにメッセージを送信する。
5. 利用者は、キューIDを使用してメッセージを取得する。
取得方法はノンブロッキング、ブロッキングの2通りあり、
ブロッキングの場合、メッセージが存在しなければ、メッセージが発生するまでタスクをスリープする。
また、どちらの取得方法も、取得したメッセージをキューから削除する。
6. 利用者は、メッセージキューが未使用になった場合に、メッセージキューを削除する

②Message event

Communicate using a variable-length record byte string (message).

The message is transmitted to a queue and, when queue is created, appoints the length of the message queue (Maximum accumulated number of the messages), the maximum number of bytes per 1 message, the classification of the queue.

The classification of the queue is classification of the processing when the queue received more event in the queue full state, and defines the following three settings.

- *Return error (EV_ERR_Busy) to message source.*
- *Delete the top message of the queue and shift (Delete the oldest message).*
- *Overwrite the last message of the queue (Overwrite the newest message).*

Distinguish a message queue in unique "queue ID" in the whole system.

The program to get a message has to make a message queue to communicate.

In addition, it is necessary to inform the queue ID of the message queue to make beforehand, for other programs that may send a message.

There are methods such as notifying in RPC-API or using fixed value as how to inform the queue ID.

Below, usage examples of Message event are described.

1. The user creates flag using flag ID.

In order to receive notification of message, the user gets FD of message queue created in 1.

2. The user waits for notification with poll/select etc, for gotten FD.

3. The sender sends message to message queue created in 1.

4. The user gets message of message queue corresponding to queue ID.

5. There are methods "block" and "non-block" to gets one message.

In the case of "block", if the message of the message queue is not sent, perform sleep of a task until message of the message queue is sent.

In addition, both methods delete message held in the message queue.

6. When the message queue becomes unused, the user delete the message queue.

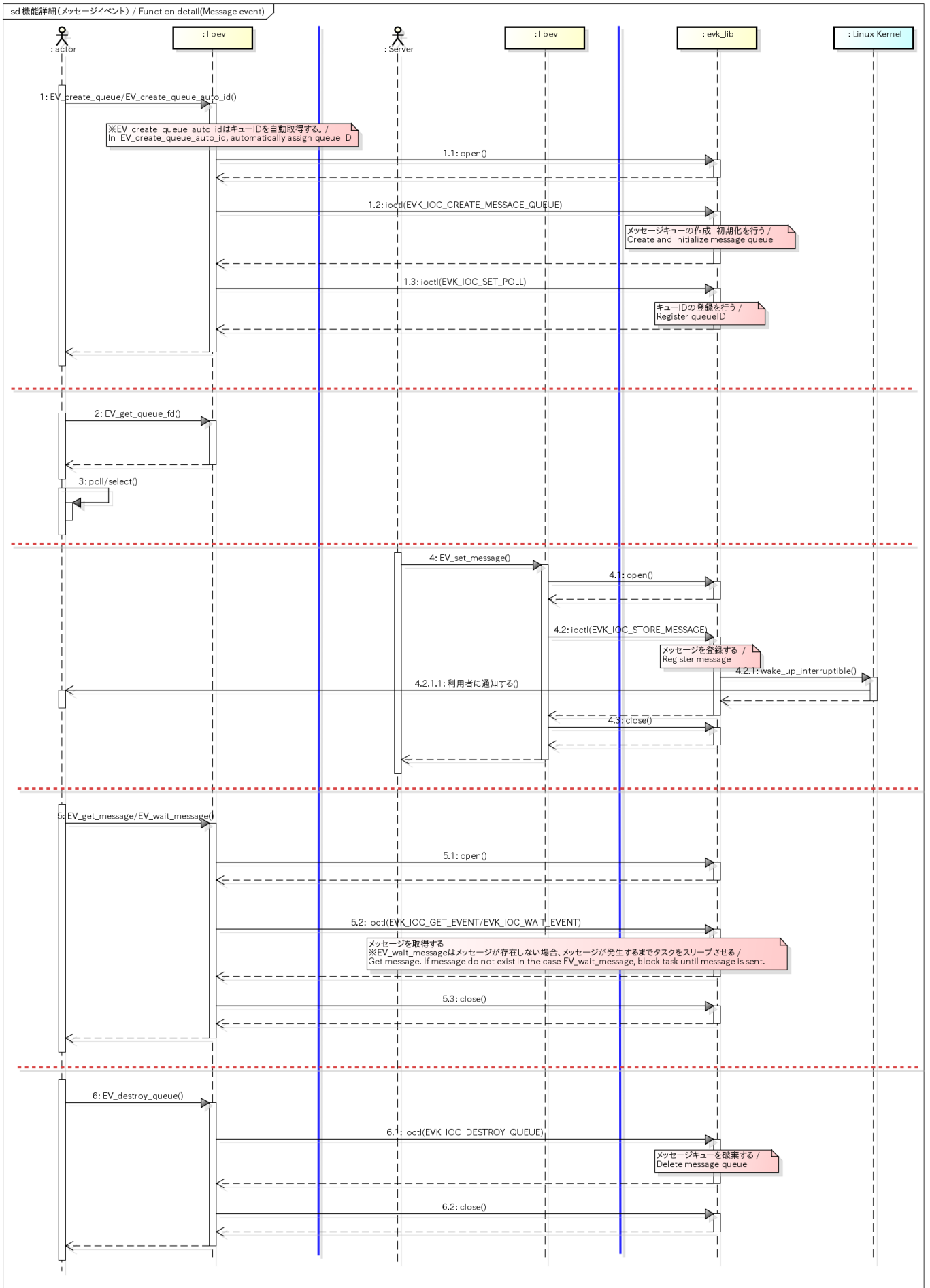


図. メッセージ機能詳細シーケンス

ソフトウェア構成図 [software block]

event_libraryのブロック図を以下に示す。

event_library 's block as below:

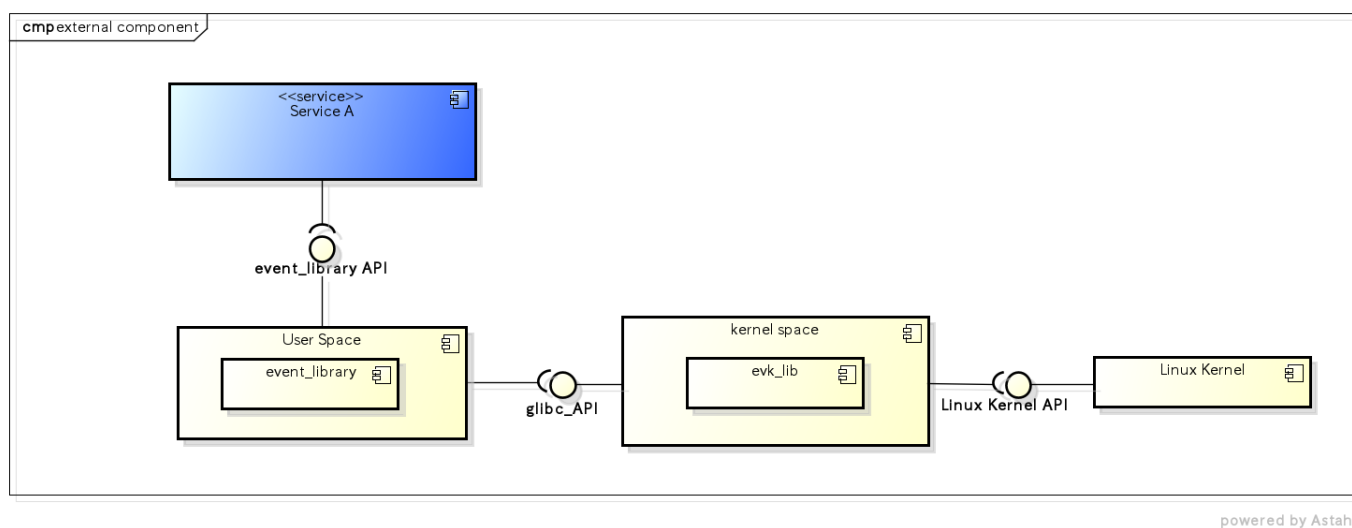


図.ソフトウェア構成図

上図の凡例) 黄色:ターゲットユニット 青色:外部ユニット

legend of above) Yellow:target units Blue:outside units

ユースケースとAPI一覧 [use-case and API lists]

Event libraryを実行する, Eventのユースケース一覧を記載する。

Use-cases of Event library, which are used by other services as follows.

外部要因 ユースケース一覧[outside factor use-case list]

1, イベント機能概要 / Event function information.

表.外部要因ユースケース一覧 (イベント機能)

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 [use-case name]	ユースケースを実現するAPI [use-case implement API]
event_library_EventInitial_001	イベントの初期化 <i>Event initial</i>	mIDからフラグIDへ変換 <i>Change from mID to flagID</i>	EV_moduleID_to_flagID
event_library_EventInitial_002		フラグ作成 <i>Create flag.</i>	EV_moduleID_to_flagID EV_create_flag
event_library_EventInitial_003		フラグ作成(mID指定) <i>Create flag(mID specified)</i>	EV_create_flag_by_mID

event_library_EventInitial_004		フラグIDを自動割当し、フラグを作成する。 <i>Create a flag(ID is allocated automatically)</i>	EV_create_flag_auto_id
event_library_EventInitial_005		mIDから64bitフラグIDへ変換 <i>Change from mID to 64bit's flagID</i>	EV_moduleID_to_flag64ID
event_library_EventInitial_006		64bit's フラグ作成 <i>Create 64bit's flag.</i>	EV_moduleID_to_flag64ID EV_create_flag64
event_library_EventInitial_007		64bit's フラグ作成(mID指定) <i>Create 64bit's flag(mID specified)</i>	EV_create_flag64_by_mID
event_library_EventInitial_008		64bit's フラグIDを自動割当し、フラグを作成する。 <i>Create 64bit's flag(ID is allocated automatically)</i>	EV_create_flag64_auto_id
event_library_SendEvent_001	イベント送信 <i>Send event</i>	フラグ送信 <i>Send flag.</i>	EV_moduleID_to_flagID EV_create_flag EV_create_flag_by_mID EV_create_flag_auto_id EV_set_flag
event_library_SendEvent_002		フラグ送信(mID指定) <i>Send flag(mID specified)</i>	EV_moduleID_to_flagID EV_create_flag EV_create_flag_by_mID EV_set_flag_by_mID
event_library_SendEvent_003		64bitフラグ送信 <i>Send 64bit's flag</i>	EV_moduleID_to_flag64ID EV_create_flag64 EV_create_flag64_by_mID EV_create_flag64_auto_id EV_set_flag64
event_library_SendEvent_004		64bitフラグ送信(モジュールID指定) <i>Send 64bit's flag(mID specified)</i>	EV_moduleID_to_flag64ID EV_create_flag64 EV_create_flag64_by_mID EV_set_flag64_by_mID
event_library_ReceiveEvent_001	イベント受信 <i>Receive events</i>	最初に到着したイベントの取得(non-block) <i>Get the first reached event(non-block)</i>	EV_moduleID_to_flagID EV_create_flag EV_create_flag_by_mID EV_create_flag_auto_id EV_set_flag EV_set_flag_by_mID EV_moduleID_to_flag64ID EV_create_flag64 EV_create_flag64_by_mID EV_create_flag64_auto_id EV_set_flag64 EV_set_flag64_by_mID EV_get_next_event

event_library_ReceiveEvent_002	<p>最初に到着したイベントの取得(非破壊的)</p> <p>非破壊的：リードオンリー的方式でメッセージキューからメッセージ内容を取得するので、メッセージキュー中のメッセージ内容をクリアしない。</p> <p><i>Get the first reached event(non-destructive)</i></p> <p><i>non-destructive: Because we get message contents from message queue by read only method, so it will not clear the message contents.</i></p>	<p>EV_moduleID_to_flagID</p> <p>EV_create_flag</p> <p>EV_create_flag_by_mID</p> <p>EV_create_flag_auto_id</p> <p>EV_set_flag</p> <p>EV_set_flag_by_mID</p> <p>EV_moduleID_to_flag64ID</p> <p>EV_create_flag64</p> <p>EV_create_flag64_by_mID</p> <p>EV_create_flag64_auto_id</p> <p>EV_set_flag64</p> <p>EV_set_flag64_by_mID</p> <p>EV_peek_next_event</p>
event_library_ReceiveEvent_003	<p>フラグイベントの取得(non-block)</p> <p><i>Get flag event(non-block)</i></p>	<p>EV_moduleID_to_flag64ID</p> <p>EV_create_flag</p> <p>EV_create_flag_by_mID</p> <p>EV_create_flag_auto_id</p> <p>EV_set_flag</p> <p>EV_set_flag_by_mID</p> <p>EV_get_flag</p>
event_library_ReceiveEvent_004	<p>フラグイベントの取得(block)</p> <p><i>Get flag event(block)</i></p>	<p>EV_moduleID_to_flagID</p> <p>EV_create_flag</p> <p>EV_create_flag_by_mID</p> <p>EV_create_flag_auto_id</p> <p>EV_set_flag</p> <p>EV_set_flag_by_mID</p> <p>EV_wait_flag</p>
event_library_ReceiveEvent_005	<p>フラグイベントの取得(非破壊的)</p> <p>非破壊的：リードオンリー的方式でメッセージキューからメッセージ内容を取得するので、メッセージキュー中のメッセージ内容をクリアしない。</p> <p><i>Get flag event(non-destructive)</i></p> <p><i>non-destructive: Because we get message contents from message queue by read only method, so it will not clear the message contents.</i></p>	<p>EV_moduleID_to_flagID</p> <p>EV_create_flag</p> <p>EV_create_flag_by_mID</p> <p>EV_create_flag_auto_id</p> <p>EV_set_flag</p> <p>EV_set_flag_by_mID</p> <p>EV_peek_flag</p>

event_library_ReceiveEvent_006		<p>フラグイベントのpoll用fdの取得</p> <p><i>Get fd which is used to poll flag event</i></p>	<p>EV_moduleID_to_flagID</p> <p>EV_create_flag</p> <p>EV_create_flag_by_mID</p> <p>EV_create_flag_auto_id</p> <p>EV_moduleID_to_flag64ID</p> <p>EV_create_flag64</p> <p>EV_create_flag64_by_mID</p> <p>EV_create_flag64_auto_id</p> <p>EV_get_flag_fd</p>
event_library_ReceiveEvent_007		<p>64bitフラグイベントの取得(non-block)</p> <p><i>Get 64bit's flag event(non-block)</i></p>	<p>EV_moduleID_to_flag64ID</p> <p>EV_create_flag64</p> <p>EV_create_flag64_by_mID</p> <p>EV_create_flag64_auto_id</p> <p>EV_set_flag64</p> <p>EV_set_flag64_by_mID</p> <p>EV_peek_flag64</p> <p>EV_get_flag64</p>
event_library_ReceiveEvent_008		<p>64bitフラグイベントの取得(block)</p> <p><i>Get 64bit's flag event(block)</i></p>	<p>EV_moduleID_to_flag64ID</p> <p>EV_create_flag64</p> <p>EV_create_flag64_by_mID</p> <p>EV_create_flag64_auto_id</p> <p>EV_set_flag64</p> <p>EV_set_flag64_by_mID</p> <p>EV_peek_flag64</p> <p>EV_wait_flag64</p>
event_library_ReceiveEvent_009		<p>64bitフラグイベントの取得(非破壊的)</p> <p>非破壊的：リードオンリーの方式でメッセージキューからメッセージ内容を取得するので、メッセージキュー中のメッセージ内容をクリアしない。</p> <p><i>Get 64bit's flag event(non-destructive)</i></p> <p><i>non-destructive: Because we get message contents from message queue by read only method, so it will not clear the message contents.</i></p>	<p>EV_moduleID_to_flag64ID</p> <p>EV_create_flag64</p> <p>EV_create_flag64_by_mID</p> <p>EV_create_flag64_auto_id</p> <p>EV_set_flag64</p> <p>EV_set_flag64_by_mID</p> <p>EV_peek_flag64</p>
event_library_DeleteEvent_001	<p>イベントの削除.</p> <p><i>Delete event.</i></p>	<p>フラグイベントの削除</p> <p><i>Delete a flag event</i></p>	<p>EV_moduleID_to_flagID</p> <p>EV_create_flag</p> <p>EV_create_flag_by_mID</p> <p>EV_create_flag_auto_id</p> <p>EV_moduleID_to_flag64ID</p> <p>EV_create_flag64</p> <p>EV_create_flag64_by_mID</p> <p>EV_create_flag64_auto_id</p> <p>EV_destroy_flag</p>

event_library_DeleteEvent_002	フラグ削除(mID指定) <i>Delete flag(mID specified)</i>	EV_moduleID_to_flagID EV_create_flag EV_create_flag_by_mID EV_create_flag_auto_id EV_moduleID_to_flag64ID EV_create_flag64 EV_create_flag64_by_mID EV_create_flag64_auto_id EV_destroy_flag EV_destroy_flag_by_mID
-------------------------------	---	---

2, メッセージ機能概要 / Message function information.

表.外部要因ユースケース一覧 (メッセージ機能)

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 [use-case name]	ユースケースを実現するAPI [use-case implement API]
event_library_MessageInitial_001	メッセージの初期化 <i>Message initial</i>	mIDからキューIDへ変換 <i>Change from mID to queueID</i>	EV_moduleID_to_queueID
event_library_MessageInitial_002		指定されたIDのメッセージイベントのキューを作成する <i>Create a message event queue whose ID is specified</i>	EV_moduleID_to_queueID EV_create_queue
event_library_MessageInitial_003		キュー作成(mID指定) <i>Create queue(mID specified)</i>	EV_moduleID_to_queueID EV_create_queue EV_create_queue_by_mID
event_library_MessageInitial_004		メッセージイベントキュー作成(ID自動割り当て) <i>Create a message event queue(ID is allocated automatically)</i>	EV_create_queue_auto_id
event_library_SendMessage_001	メッセージ送信 <i>Send message</i>	メッセージイベントの送信 <i>Send message event</i>	EV_moduleID_to_queueID EV_create_queue EV_create_queue_by_mID EV_create_queue_auto_id EV_send_message
event_library_SendMessage_002		メッセージ送信(mID指定) <i>Send message(mID specified)</i>	EV_moduleID_to_queueID EV_create_queue EV_create_queue_by_mID EV_create_queue_auto_id EV_send_message EV_send_message_by_mID

event_library_ReceiveMessage_001	メッセージ受信 <i>Receive message</i>	最初に到着したイベントの取得(non-block) <i>Get the first reached event(non-block)</i>	EV_moduleID_to_queueID EV_create_queue EV_create_queue_by_mID EV_create_queue_auto_id EV_send_message EV_send_message_by_mID EV_get_next_event
event_library_ReceiveMessage_002		最初に到着したイベントの取得(非破壊的) 非破壊的：リードオンリー的方式でメッセージキューからメッセージ内容を取得するので、メッセージキュー中のメッセージ内容をクリアしない。 <i>Get the first reached event(non-destructive)</i> <i>non-destructive: Because we get message contents from message queue by read only method, so it will not clear the message contents.</i>	EV_moduleID_to_queueID EV_create_queue EV_create_queue_by_mID EV_create_queue_auto_id EV_send_message EV_send_message_by_mID EV_peek_next_event
event_library_ReceiveMessage_003		メッセージイベントの取得(non-block) <i>Get message event queue(ID specified)</i>	EV_moduleID_to_queueID EV_create_queue EV_create_queue_by_mID EV_create_queue_auto_id EV_send_message EV_send_message_by_mID EV_get_message
event_library_ReceiveMessage_004		メッセージイベントの取得(block) <i>Get message event queue(block)</i>	EV_moduleID_to_queueID EV_create_queue EV_create_queue_by_mID EV_create_queue_auto_id EV_send_message EV_send_message_by_mID EV_wait_message
event_library_ReceiveMessage_005		メッセージイベントの取得(非破壊的) 非破壊的：リードオンリー的方式でメッセージキューからメッセージ内容を取得するので、メッセージキュー中のメッセージ内容をクリアしない。 <i>Get message event(non-destructive)</i> <i>non-destructive: Because we get message contents from message queue by read only method, so it will not clear the message contents.</i>	EV_moduleID_to_queueID EV_create_queue EV_create_queue_by_mID EV_create_queue_auto_id EV_send_message EV_send_message_by_mID EV_peek_message

event_library_ReceiveMessage_006		メッセージイベントの取得(送信元情報で検索) <i>Get message event(search with sender information)</i>	EV_moduleID_to_queueID EV_create_queue EV_create_queue_by_mID EV_create_queue_auto_id EV_send_message EV_send_message_by_mID EV_find_message_by_sender
event_library_ReceiveMessage_007		メッセージイベントの取得(内容で検索) <i>Get message event(search with contents)</i>	EV_moduleID_to_queueID EV_create_queue EV_create_queue_by_mID EV_create_queue_auto_id EV_send_message EV_send_message_by_mID EV_find_message_by_content
event_library_ReceiveMessage_008		メッセージイベントキューのpoll用fdの取得 <i>Get fd which is used to poll message event queue</i>	EV_moduleID_to_queueID EV_create_queue EV_create_queue_by_mID EV_create_queue_auto_id EV_get_queue_fd
event_library_DeleteMessage_001	メッセージの削除. <i>Delete message.</i>	メッセージイベントキューの削除 <i>Delete a message event queue</i>	EV_moduleID_to_queueID EV_create_queue EV_create_queue_by_mID EV_create_queue_auto_id EV_destroy_queue
event_library_DeleteMessage_002		キュー削除(mID指定) <i>Delete queue(mID specified)</i>	EV_moduleID_to_queueID EV_create_queue EV_create_queue_by_mID EV_create_queue_auto_id EV_destroy_queue_by_mID

内部処理 エラーユースケース一覧 [\[internal processing error use-case list\]](#)

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]
event_library_Initial_Error_001	イベントの初期化 <i>Event panel</i>	イベント作成 (open) <i>Create event (open)</i>	EV_ERR_Fatal	
event_library_Initial_Error_002	メッセージの初期化 <i>Message initial</i>	キュー作成 <i>Create queue (open)</i>	EV_ERR_Fatal	

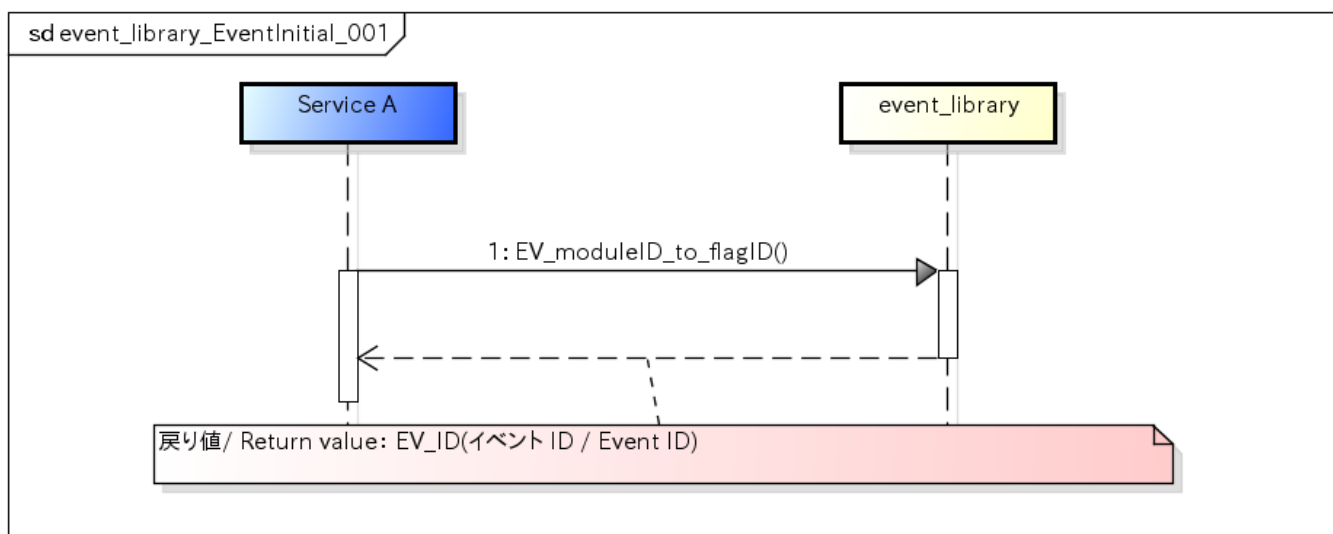
ユースケースevent_library_EventInitial_001 [\[use-case event_library_EventInitial_001\]](#)

概要 [\[Overview\]](#)

mIDからフラグIDへ変換する

Change from mID to flagID.

シーケンス [Sequence]



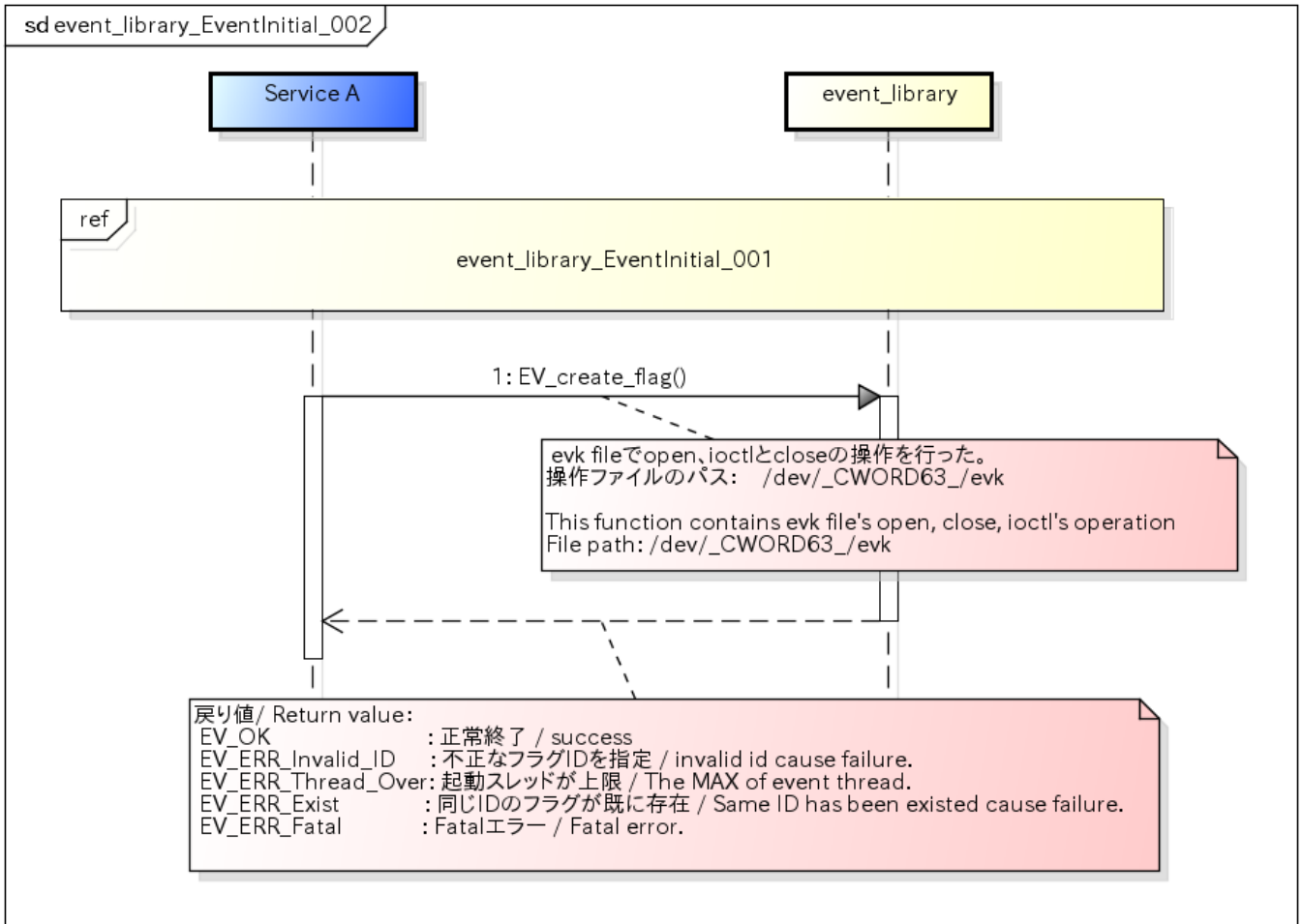
ユースケースevent_library_EventInitial_002 [use-case event_library_EventInitial_002]

概要 [Overview]

フラグ作成

Create flag.

シーケンス [Sequence]



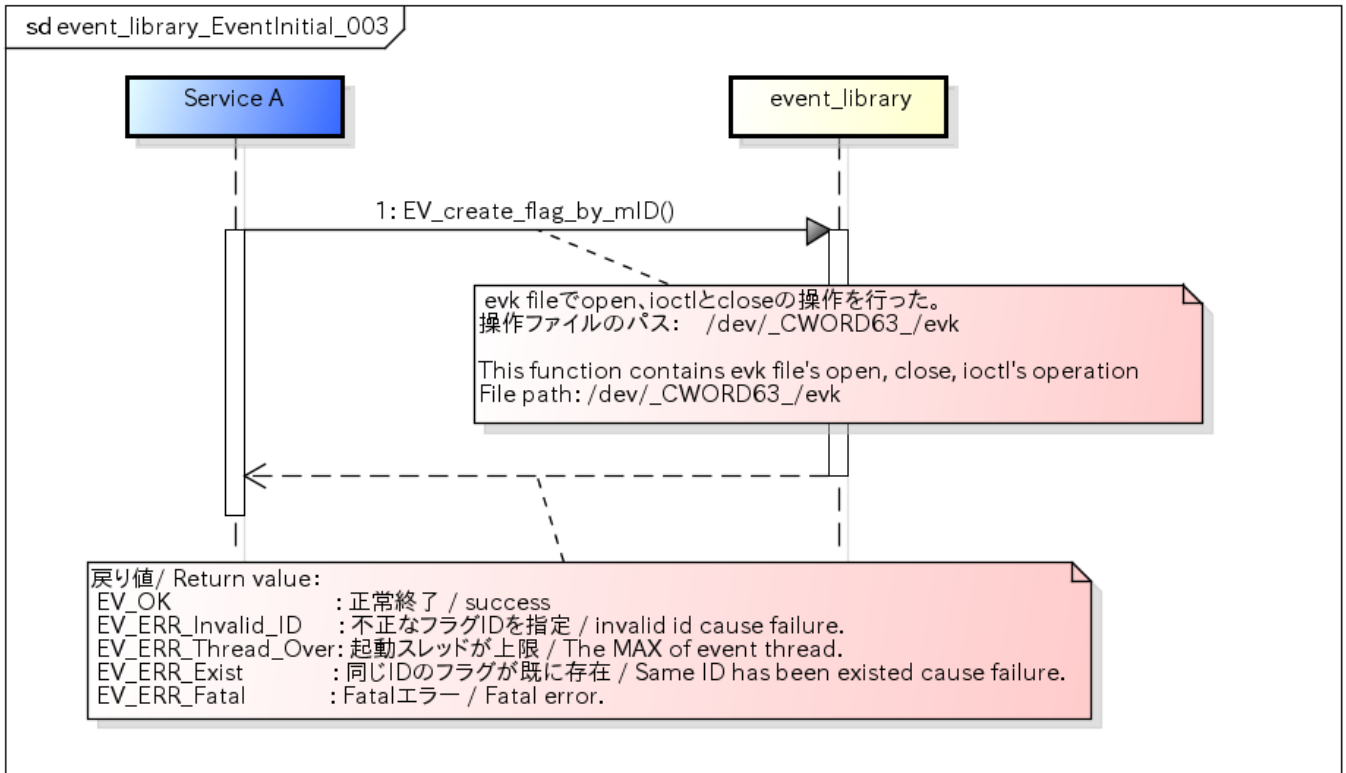
ユースケースevent_library_EventInitial_003 [\[use-case event_library_EventInitial_003\]](#)

概要 [\[Overview\]](#)

フラグ作成(mID 指定)

Create flag (mID specified)

シーケンス [\[Sequence\]](#)



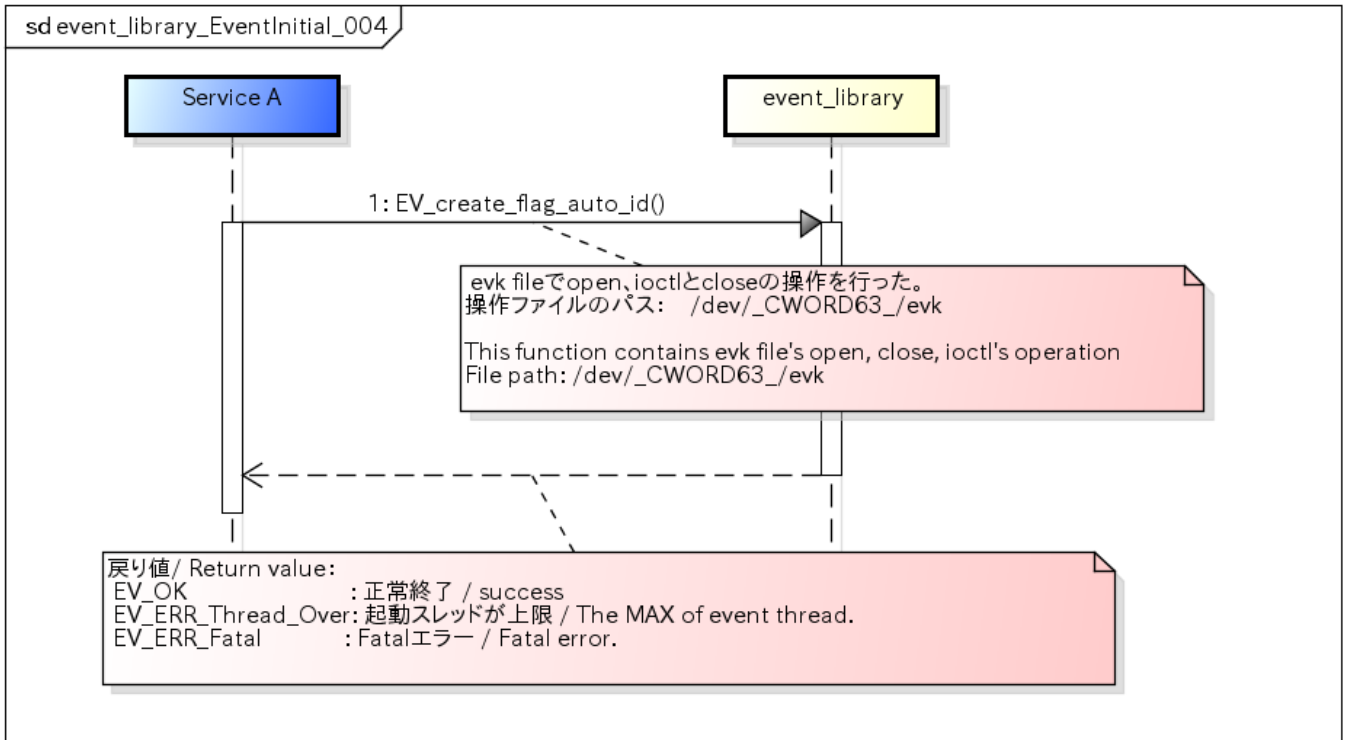
ユースケースevent_library_EventInitial_004 [[use-case event_library_EventInitial_004](#)]

概要 [[Overview](#)]

フラグIDを自動割当し、フラグを作成する。

Create a flag ID is allocated automatically.

シーケンス [[Sequence](#)]



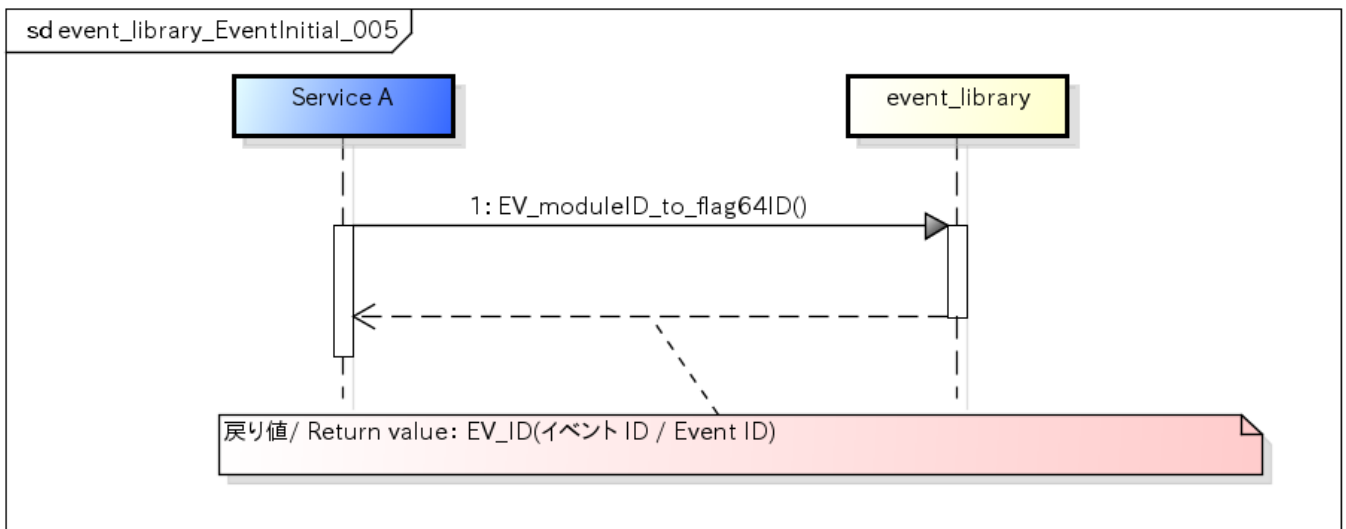
ユースケースevent_library_EventInitial_005 [[use-case event_library_EventInitial_005](#)]

概要 [[Overview](#)]

mIDから64 ビットフラグIDへ変換する

Change from mID to 64bit's flagID.

シーケンス [[Sequence](#)]



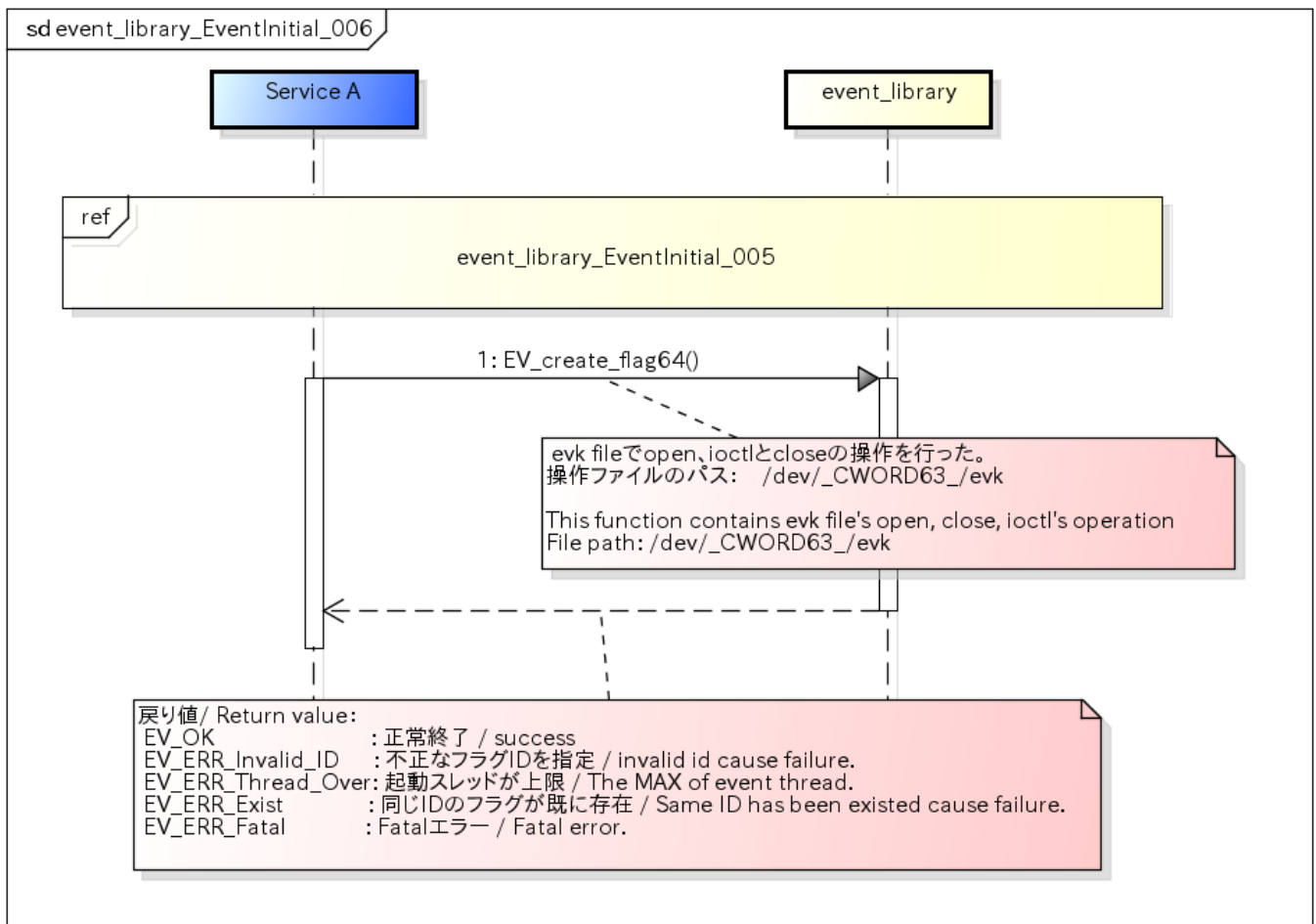
ユースケースevent_library_EventInitial_006 [[use-case event_library_EventInitial_006](#)]

概要 [[Overview](#)]

64 ビットフラグ作成

Create 64bit's flag.

シーケンス [Sequence]



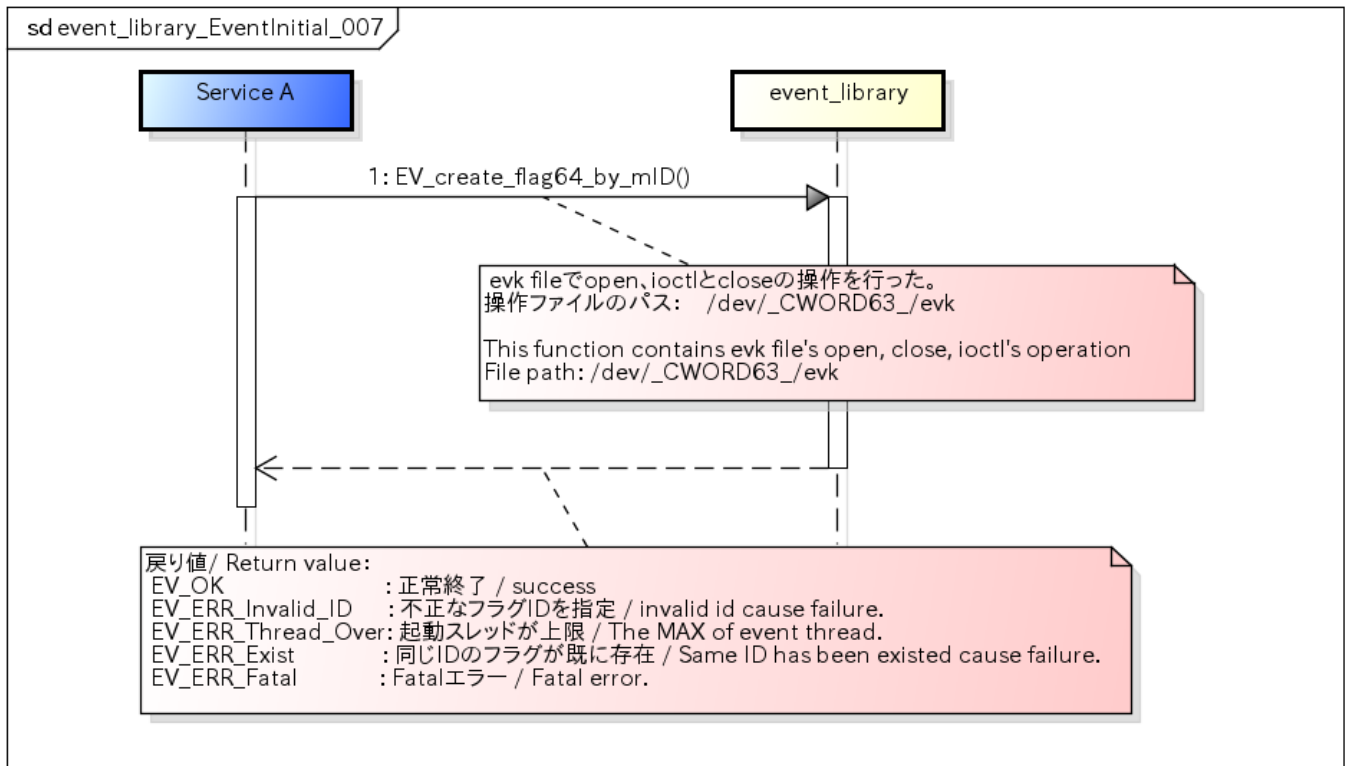
ユースケースevent_library_EventInitial_007 [use-case event_library_EventInitial_007]

概要 [Overview]

64ビットフラグ作成(mID 指定)

Create 64bit's flag(mID specified)

シーケンス [Sequence]



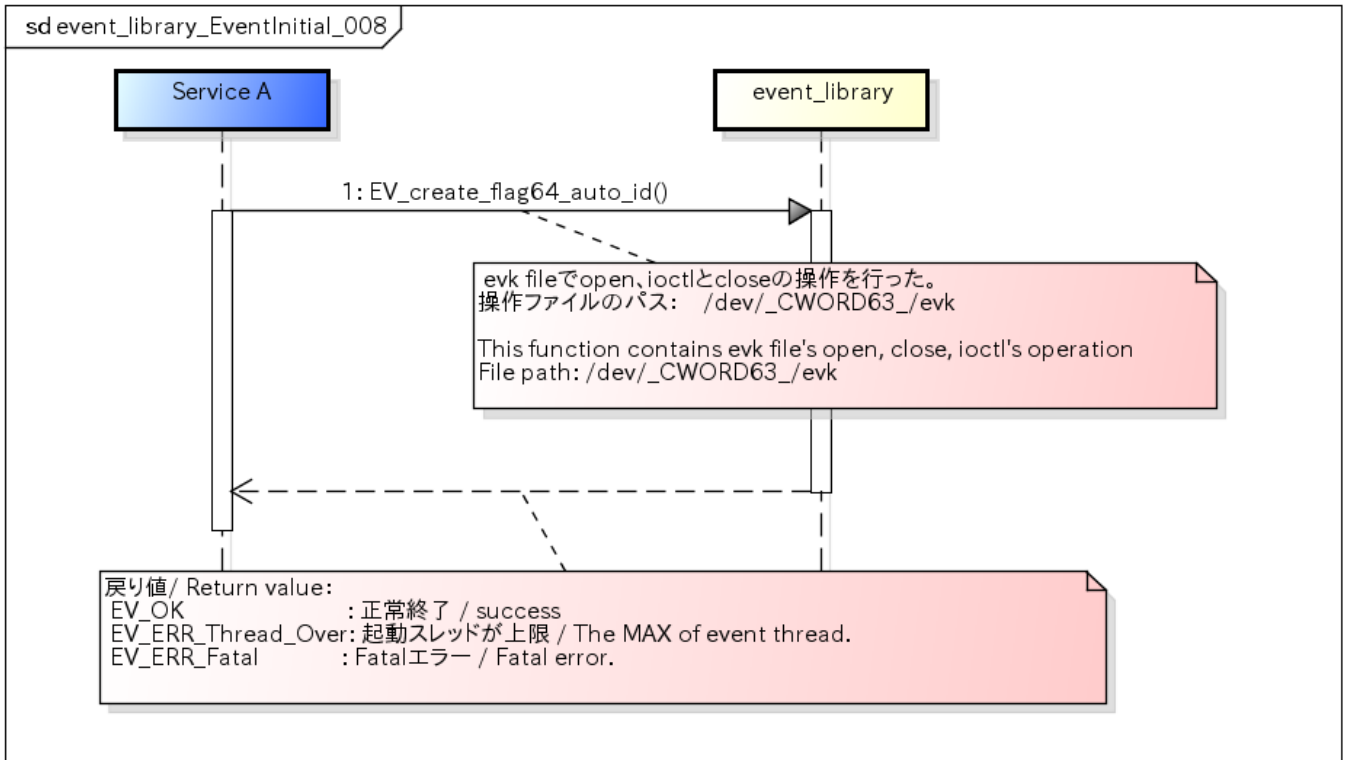
ユースケースevent_library_EventInitial_008 [\[use-case event_library_EventInitial_008\]](#)

概要 [\[Overview\]](#)

64 ビットフラグIDを自動割当し、フラグを作成する。

Create 64bit's flag ID is allocated automatically.

シーケンス [\[Sequence\]](#)



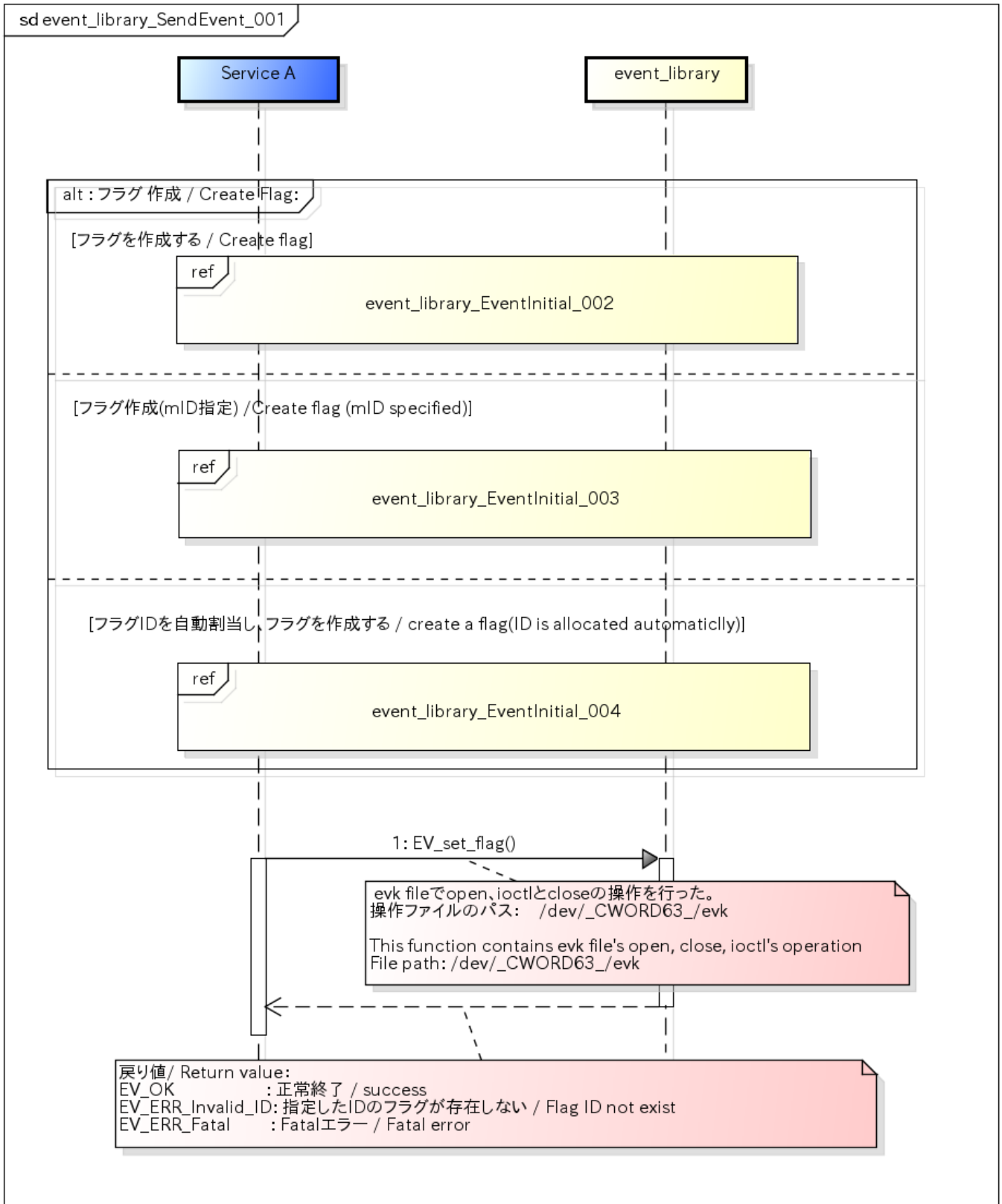
ユースケースevent_library_SendEvent_001 [[use-case event_library_SendEvent_001](#)]

概要 [[Overview](#)]

フラグ送信.

Send flag.

シーケンス [[Sequence](#)]



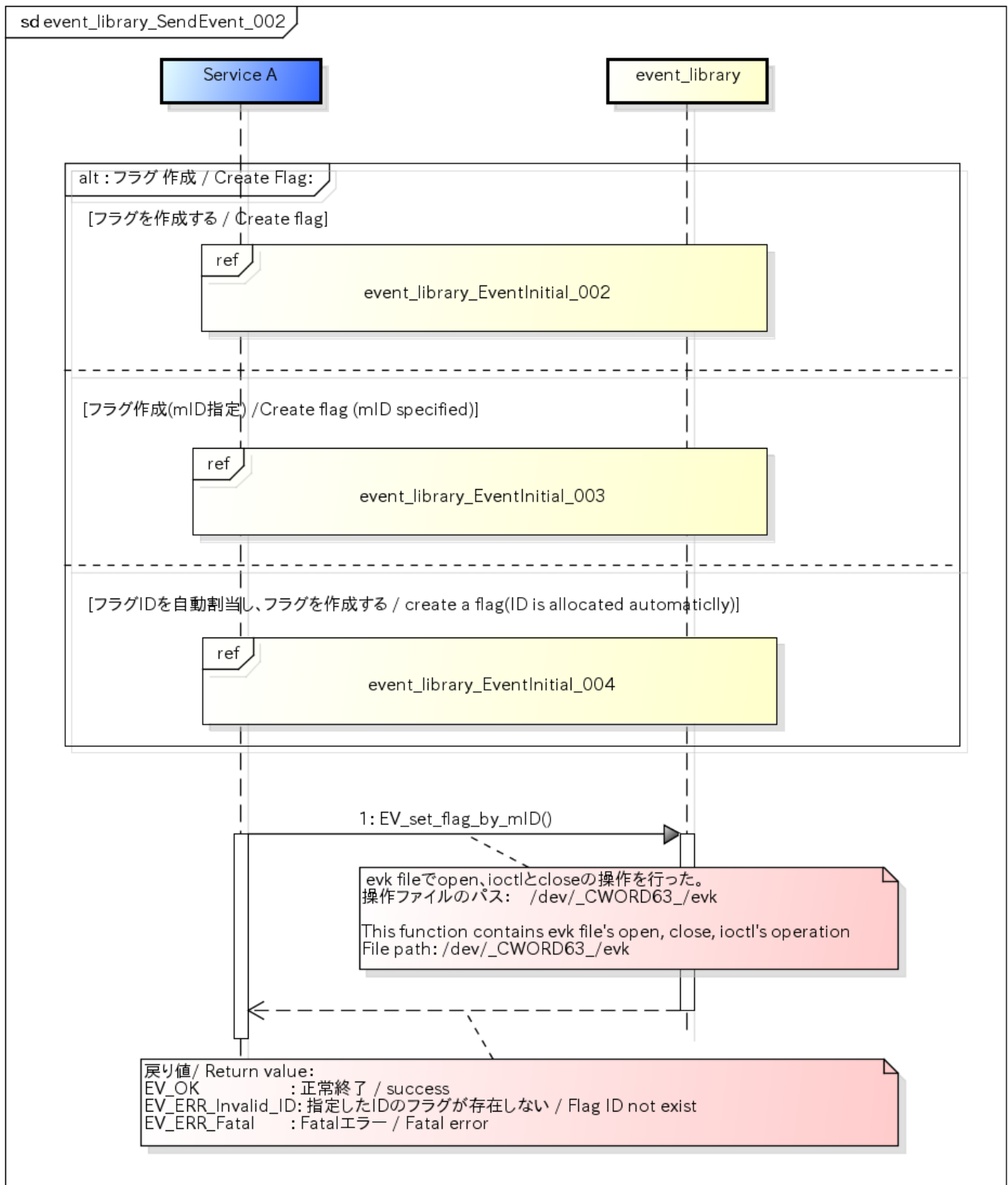
ユースケースevent_library_SendEvent_002 [\[use-case event_library_SendEvent_002\]](#)

概要 [\[Overview\]](#)

フラグ送信(mID指定).

[Send flag flag\(mID specified\).](#)

シーケンス [Sequence]

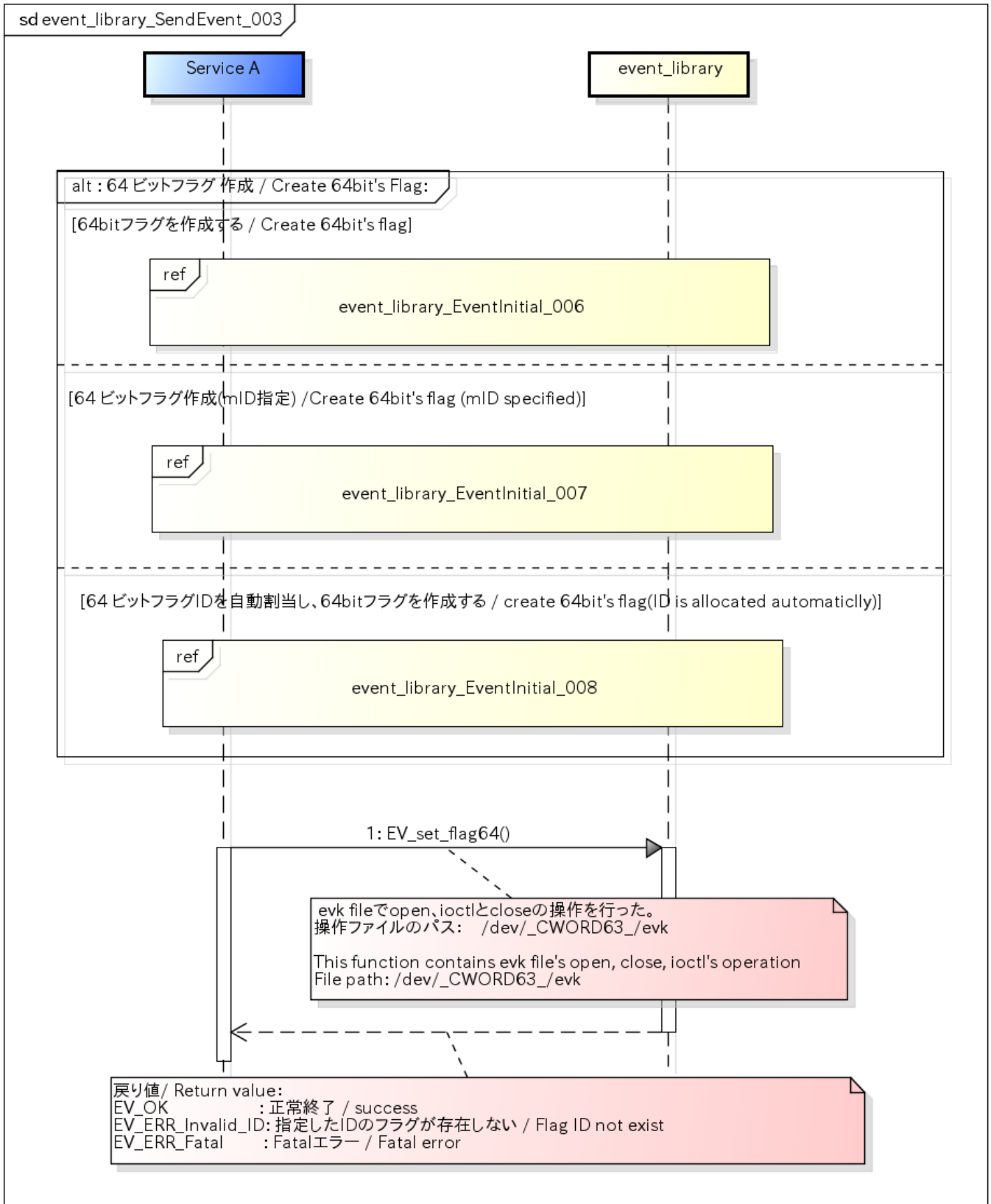


ユースケースevent_library_SendEvent_003 [use-case event_library_SendEvent_003]

概要 [Overview]

64 ビットフラグ送信.

シーケンス [Sequence]



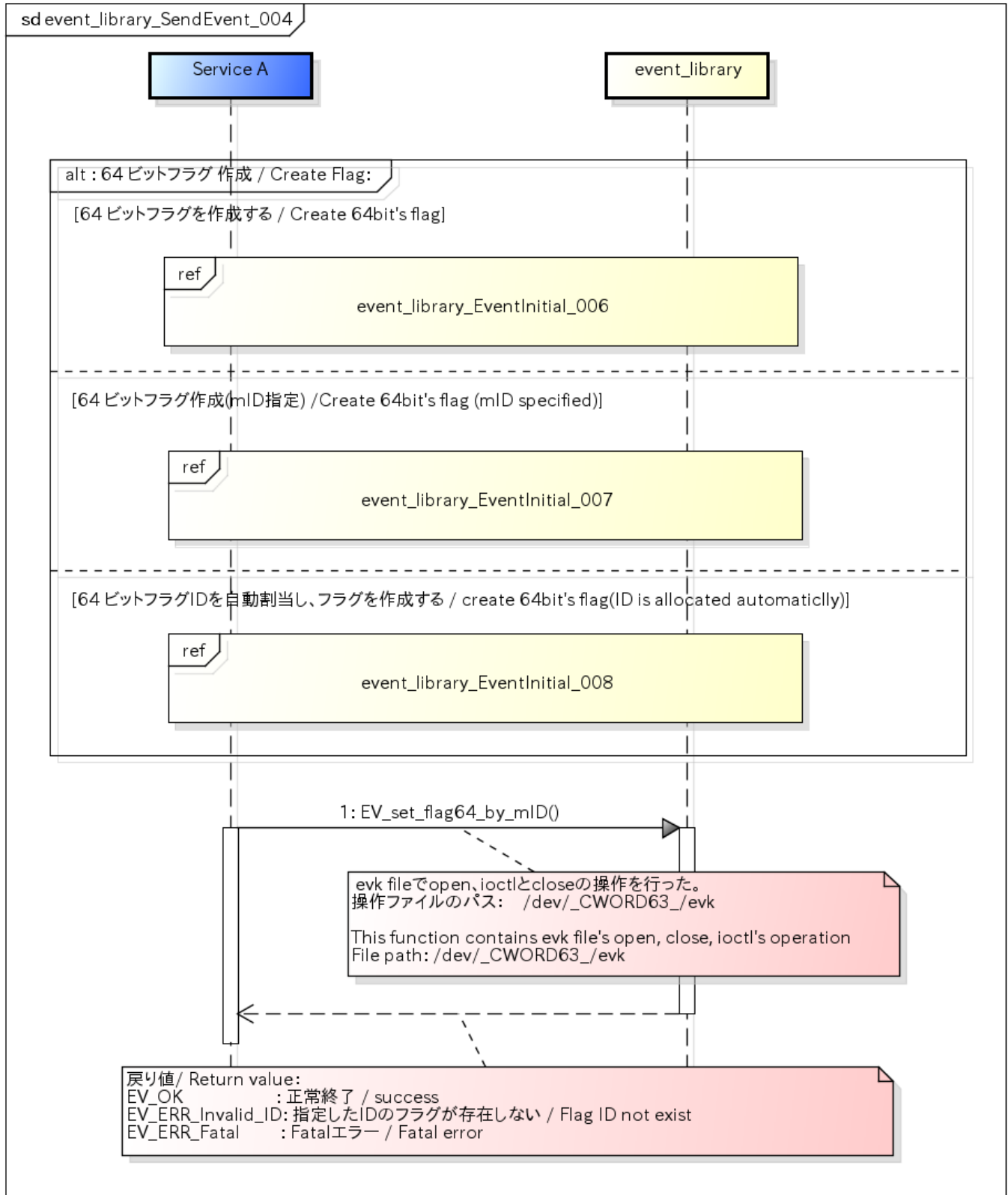
ユースケース event_library_SendEvent_004 [use-case event_library_SendEvent_004]

概要 [Overview]

64 ビットフラグ送信(mID指定).

Send 64bit's flag(mID specified).

シーケンス [Sequence]



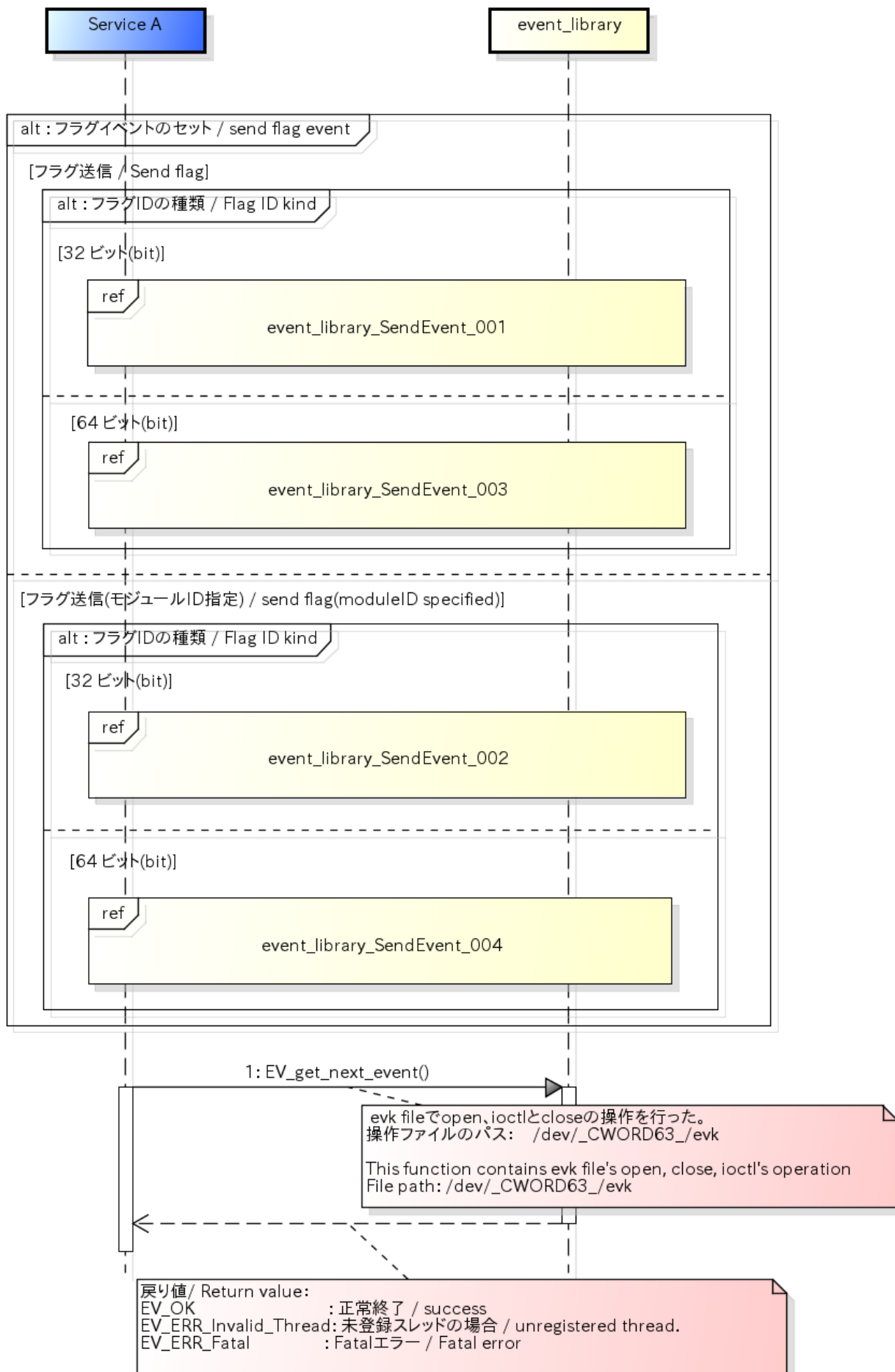
ユースケースevent_library_ReceiveEvent_001 [\[use-case event_library_ReceiveEvent_001\]](#)

概要 [\[Overview\]](#)

最初に到着したイベントの取得(non-block)

Get the first reached event(non-block)

シーケンス [\[Sequence\]](#)



ユースケースevent_library_ReceiveEvent_002 [*use-case event_library_ReceiveEvent_002*]

概要 [*Overview*]

最初に到着したイベントの取得(非破壊的).

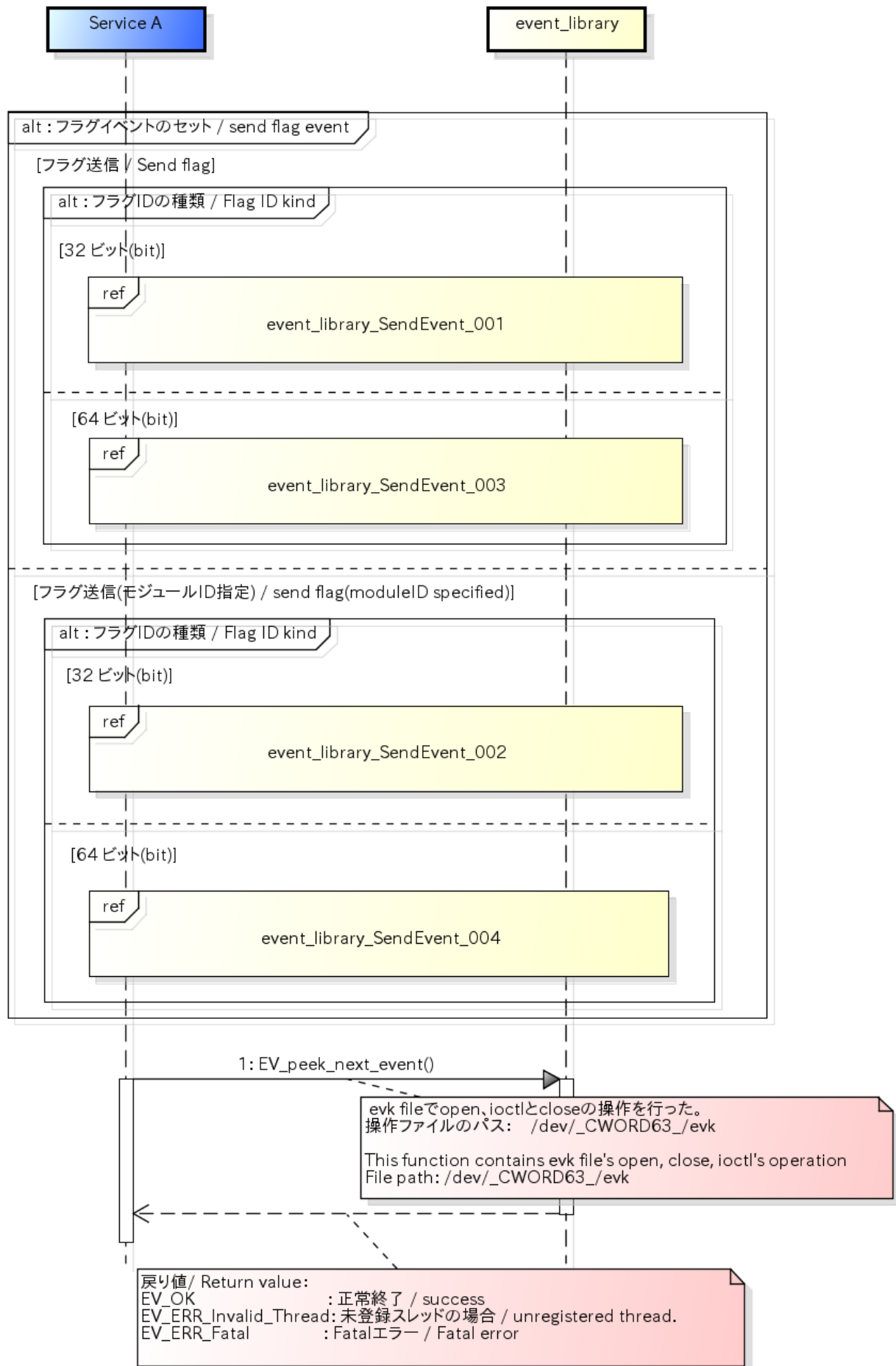
非破壊的：リードオンリー的方式でメッセージキューからメッセージ内容を取得するので、メッセージキュー中のメッセージ内容をクリアしない。

Get the first reached event(non-destructive).

non-destructive: Because we get message contents from message queue by read only method, so it will not clear the message contents.

シーケンス [*Sequence*]

sd event_library_ReceiveEvent_002



ユースケースevent_library_ReceiveEvent_003 [[use-case event_library_ReceiveEvent_003](#)]

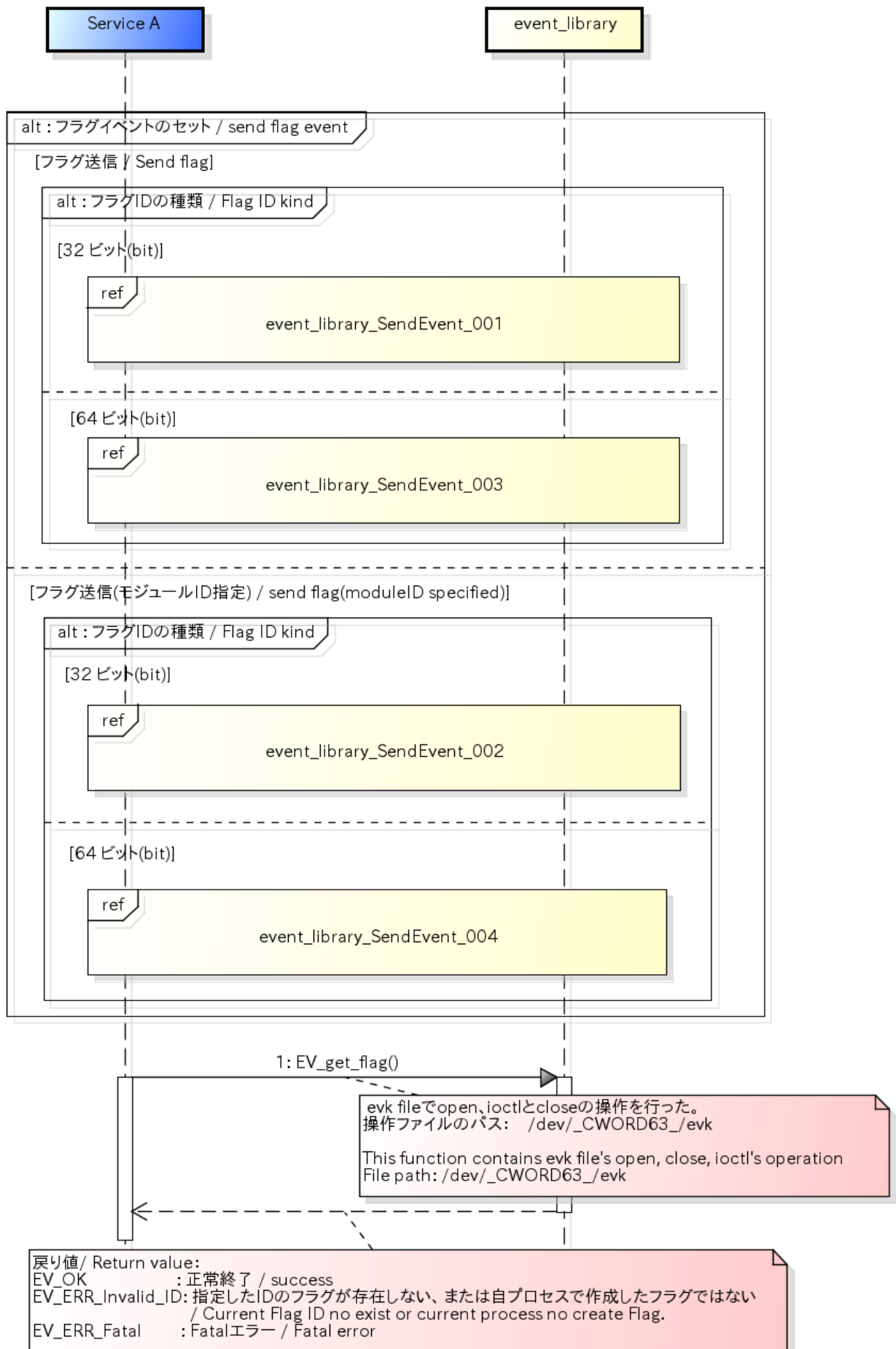
概要 [[Overview](#)]

フラグイベントの取得(non-block)

[Get flag event\(non-block\)](#)

シーケンス [[Sequence](#)]

sd event_library_ReceiveEvent_003



ユースケースevent_library_ReceiveEvent_004 [[use-case event_library_ReceiveEvent_004](#)]

概要 [[Overview](#)]

フラグイベントの取得(block)

[Get flag event\(block\)](#)

シーケンス [[Sequence](#)]

sd event_library_ReceiveEvent_004

Service A

event_library

alt : フラグイベントのセット / send flag event

[フラグ送信 / Send flag]

alt : フラグIDの種類 / Flag ID kind

[32 ビット(bit)]

ref

event_library_SendEvent_001

[64 ビット(bit)]

ref

event_library_SendEvent_003

[フラグ送信(モジュールID指定) / send flag(moduleID specified)]

alt : フラグIDの種類 / Flag ID kind

[32 ビット(bit)]

ref

event_library_SendEvent_002

[64 ビット(bit)]

ref

event_library_SendEvent_004

1: EV_wait_flag()

evk fileでopen、ioctlとcloseの操作を行った。
操作ファイルのパス: /dev/_CWORD63_/evk

This function contains evk file's open, close, ioctl's operation
File path: /dev/_CWORD63_/evk

戻り値 / Return value:

EV_OK : 正常終了 / success

EV_ERR_Invalid_ID: 指定したIDのフラグが存在しない、または自プロセスで作成したフラグではない
/ Current Flag ID no exist or current process no create Flag.

EV_ERR_Interrupted: シグナル割り込みによりタスクブロッキングが解除された場合
/ Task blocking was canceled by signal interrupt.

EV_ERR_Fatal / Fatalエラー / Fatal error

ユースケースevent_library_ReceiveEvent_005 [[use-case event_library_ReceiveEvent_005](#)]

概要 [[Overview](#)]

フラグイベントの取得(非破壊的).

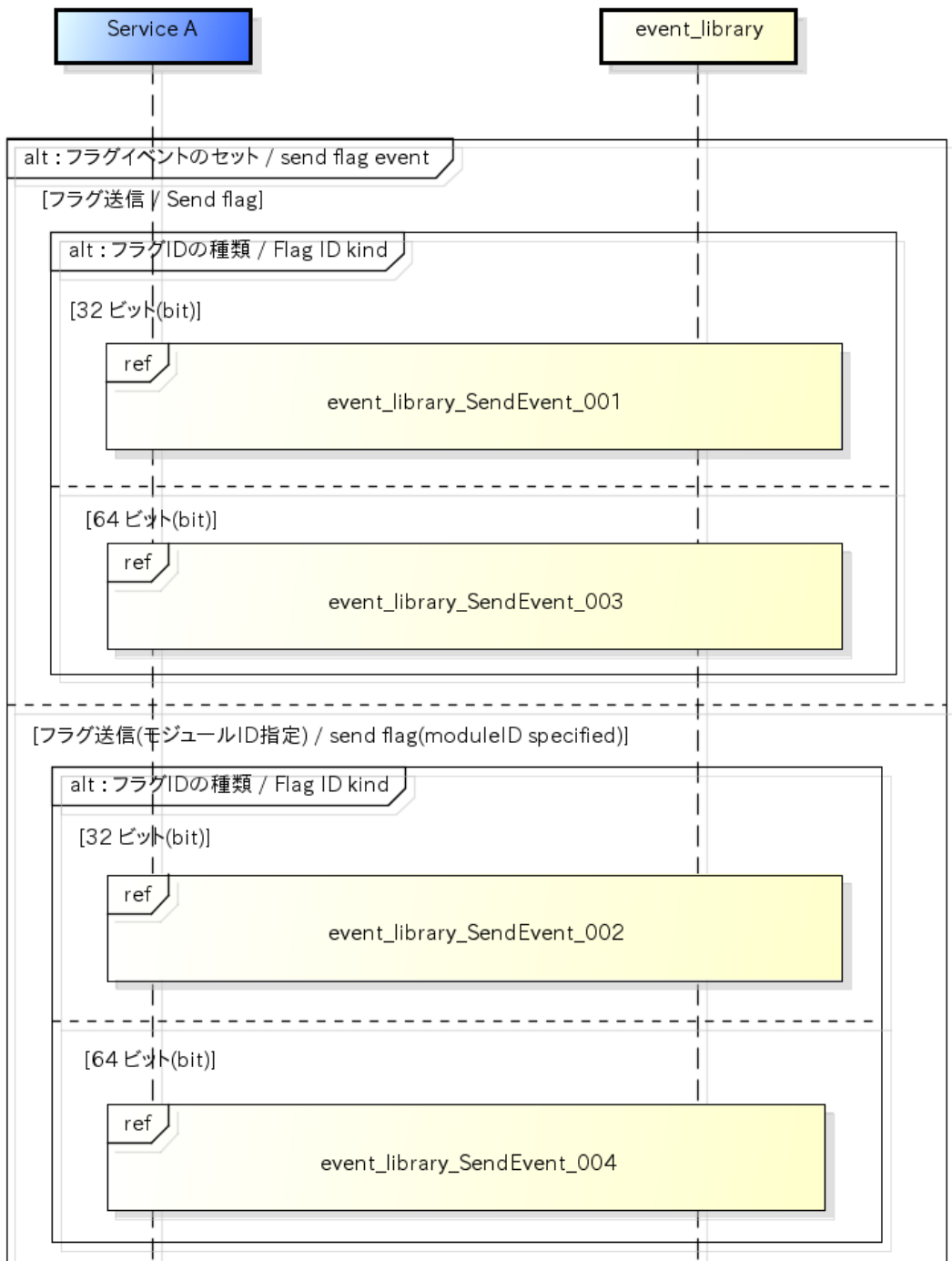
非破壊的：リードオンリー的方式でメッセージキューからメッセージ内容を取得するので、メッセージキュー中のメッセージ内容をクリアしない。

Get flag event(non-destructive).

non-destructive: Because we get message contents from message queue by read only method, so it will not clear the message contents.

シーケンス [[Sequence](#)]

sd event_library_ReceiveEvent_005



1: EV_peek_flag()

evk fileでopen, ioctlとcloseの操作を行った。
 操作ファイルのパス: /dev/_CWORD63_/evk
 This function contains evk file's open, close, ioctl's operation
 File path: /dev/_CWORD63_/evk

戻り値 / Return value:
 EV_OK : 正常終了 / success
 EV_ERR_Invalid_ID: 指定したIDのフラグが存在しない、または自プロセスで作成したフラグではない
 / Current Flag ID no exist or current process no create Flag.
 EV_ERR_Fatal : Fatalエラー / Fatal error

ユースケースevent_library_ReceiveEvent_006 [[use-case event_library_ReceiveEvent_006](#)]

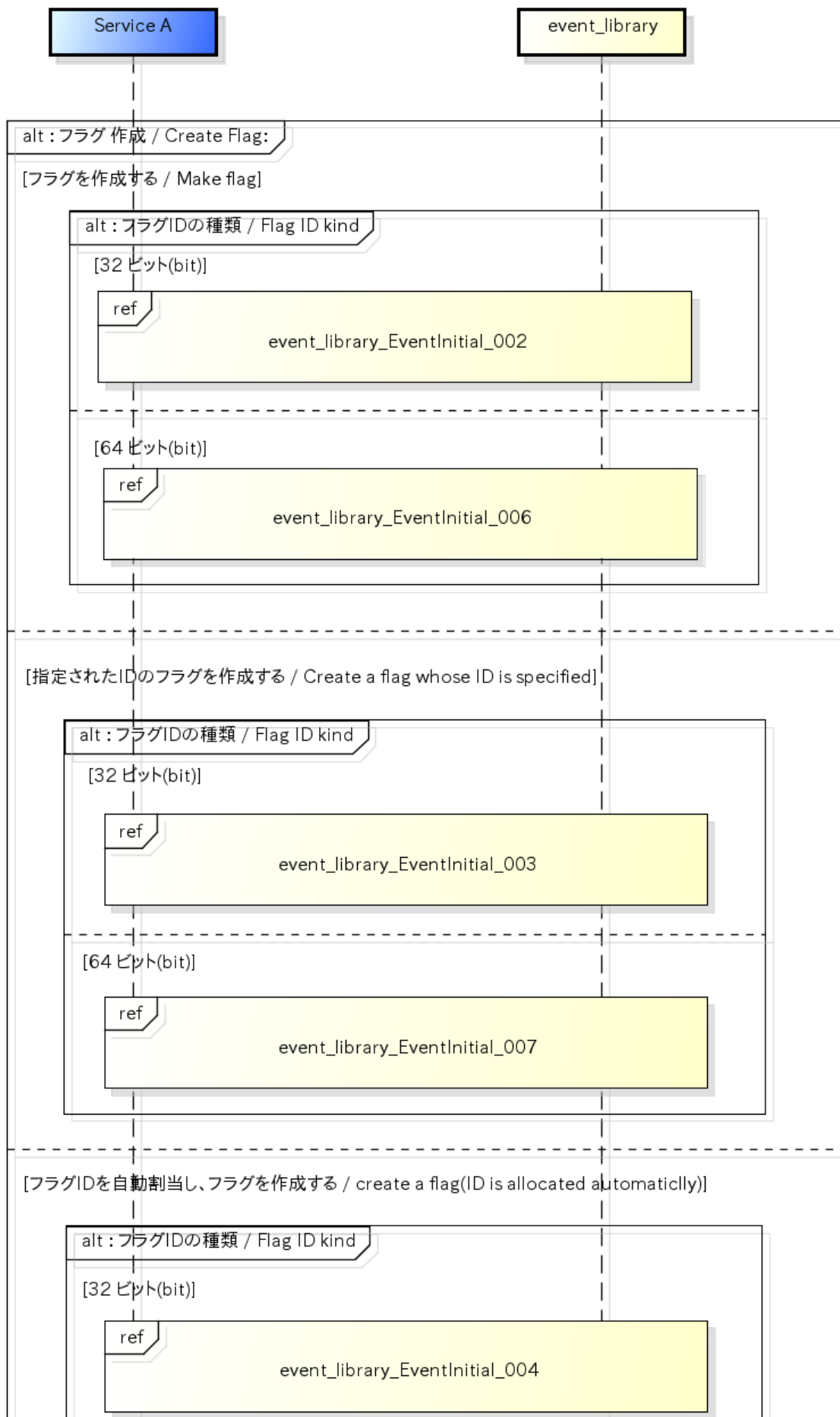
概要 [[Overview](#)]

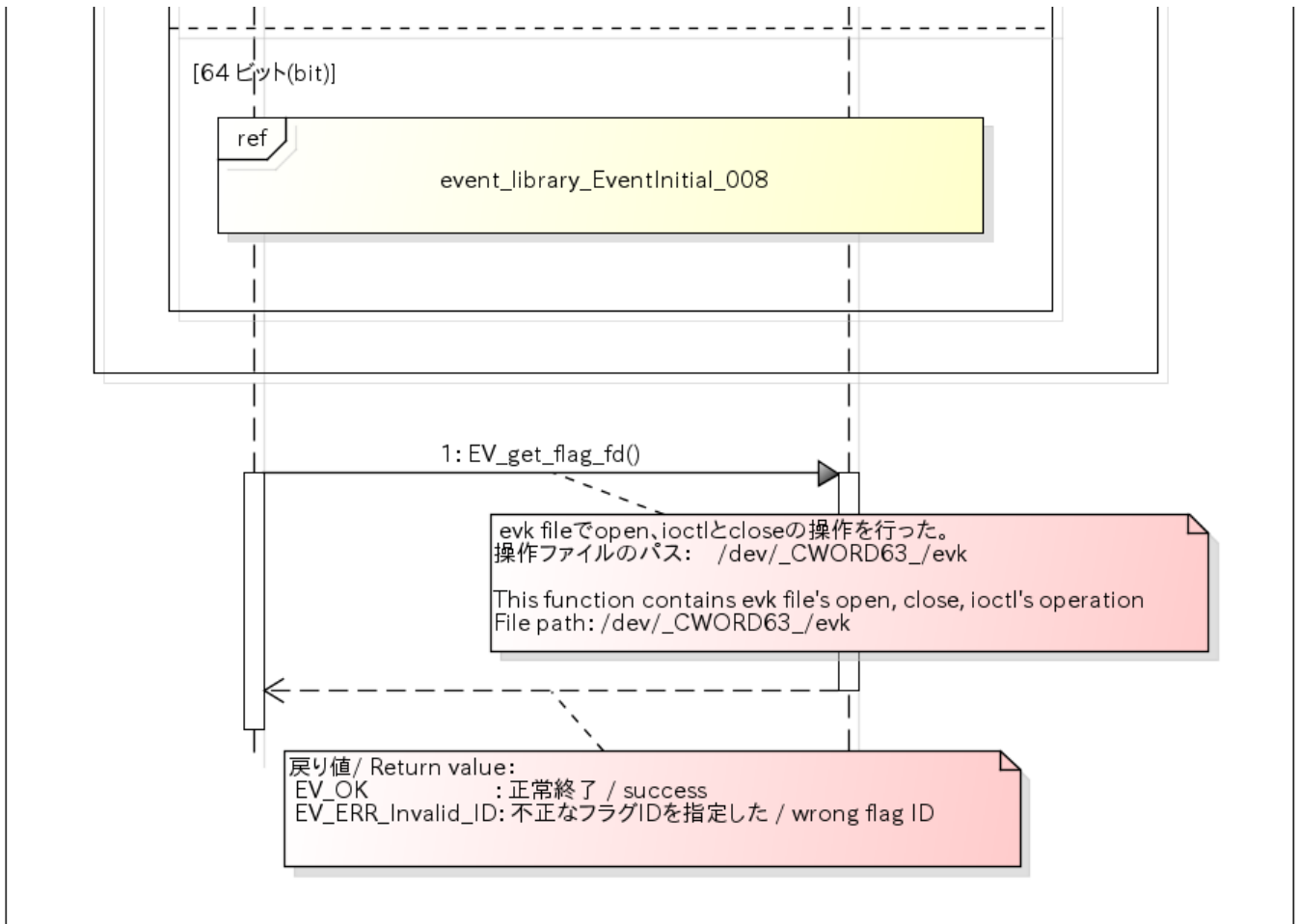
フラグイベントのpoll用fdの取得

Get fd which is used to poll flag event

シーケンス [[Sequence](#)]

sd event_library_ReceiveEvent_006





ユースケースevent_library_ReceiveEvent_007 [[use-case event_library_ReceiveEvent_007](#)]

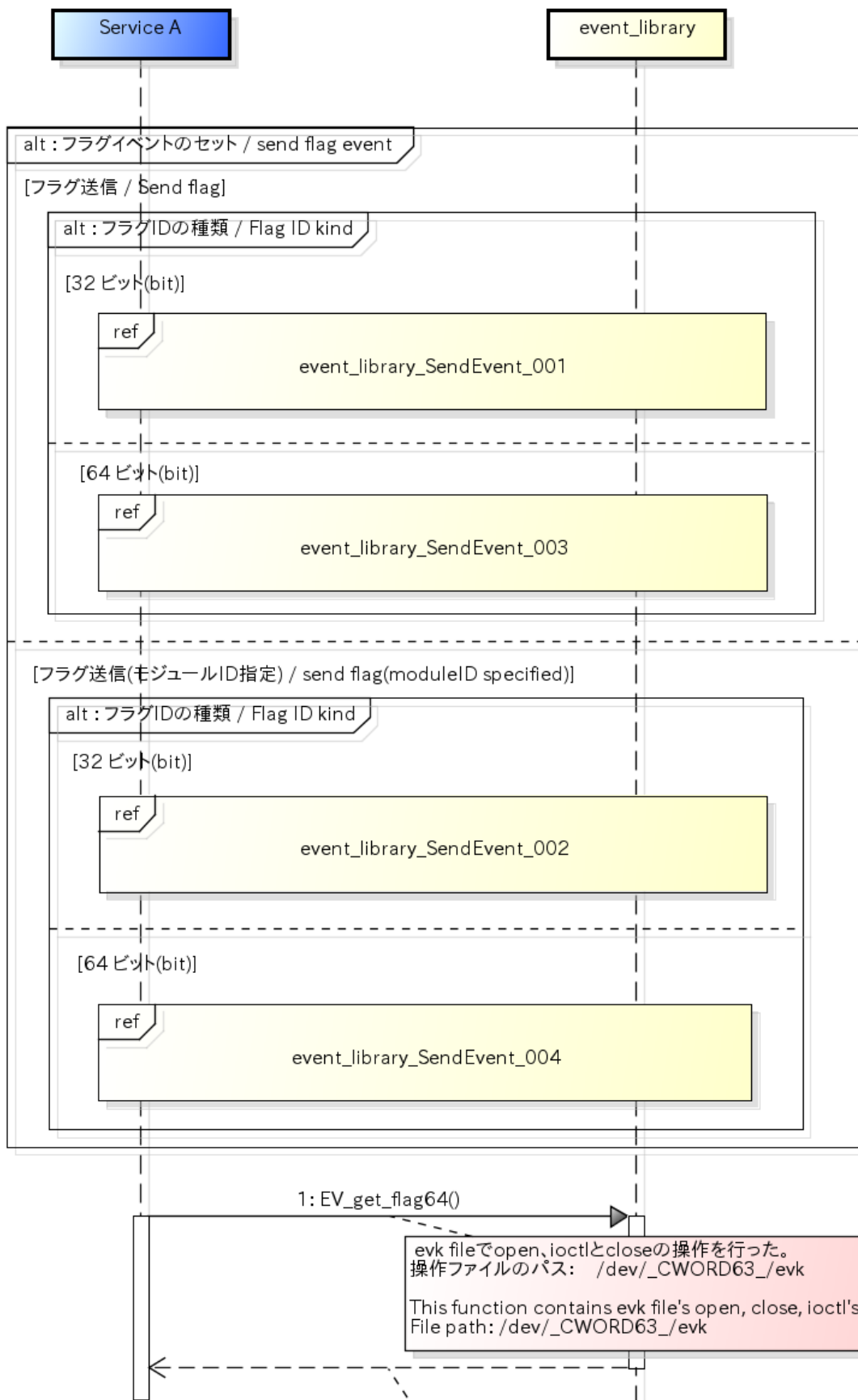
概要 [[Overview](#)]

64 ビットフラグイベントの取得(non-block)

[Get 64bit's flag event\(non-block\)](#)

シーケンス [[Sequence](#)]

sd event_library_ReceiveEvent_007



evk fileでopen, ioctlとcloseの操作を行った。
 操作ファイルのパス: /dev/_CWORD63_/evk
 This function contains evk file's open, close, ioctl's operation
 File path: /dev/_CWORD63_/evk

*1)
 戻り値/ Return value:
 EV_OK : 正常終了 / success
 EV_ERR_Invalid_ID: 指定したIDのフラグが存在しない、または自プロセスで作成したフラグではない
 / Current Flag ID no exist or current process no create Flag.
 EV_ERR_Exist : Existエラー / Exist error

EV_ERR_fatal : fatalエラー / fatal error

ユースケースevent_library_ReceiveEvent_008 [[use-case event_library_ReceiveEvent_008](#)]

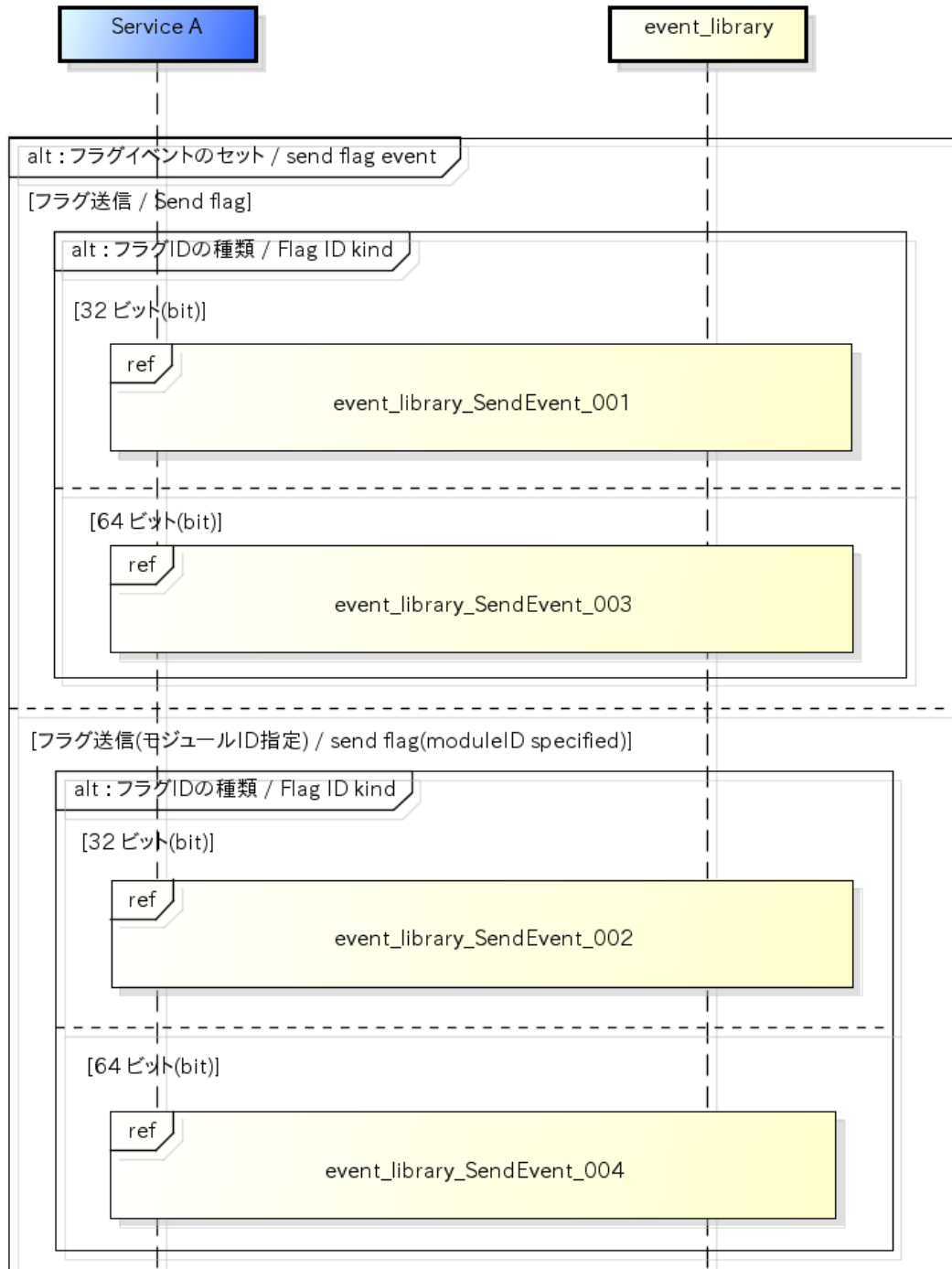
概要 [[Overview](#)]

64 ビットフラグイベントの取得(block)

Get 64bit's flag event(block)

シーケンス [[Sequence](#)]

sd event_library_ReceiveEvent_008



1: EV_wait_flag64()

evk fileでopen、ioctlとcloseの操作を行った。
 操作ファイルのパス: /dev/_CWORD63_/evk
 This function contains evk file's open, close, ioctl's operation
 File path: /dev/_CWORD63_/evk

戻り値/ Return value:
 EV_OK : 正常終了 / success
 EV_ERR_Invalid_ID: 指定したIDのフラグが存在しない、または自プロセスで作成したフラグではない
 / Current Flag ID no exist or current process no create Flag.
 EV_ERR_Interrupted: シグナル割り込みによりタスクブロッキングが解除された場合
 / Task blocking was canceled by signal interrupt.
 EV_ERR_Exit : 終了 / Exit

EV_ERR_fatal : fatalエラー / fatal error

ユースケースevent_library_ReceiveEvent_009 [[use-case event_library_ReceiveEvent_009](#)]

概要 [[Overview](#)]

64 ビットフラグイベントの取得(非破壊的).

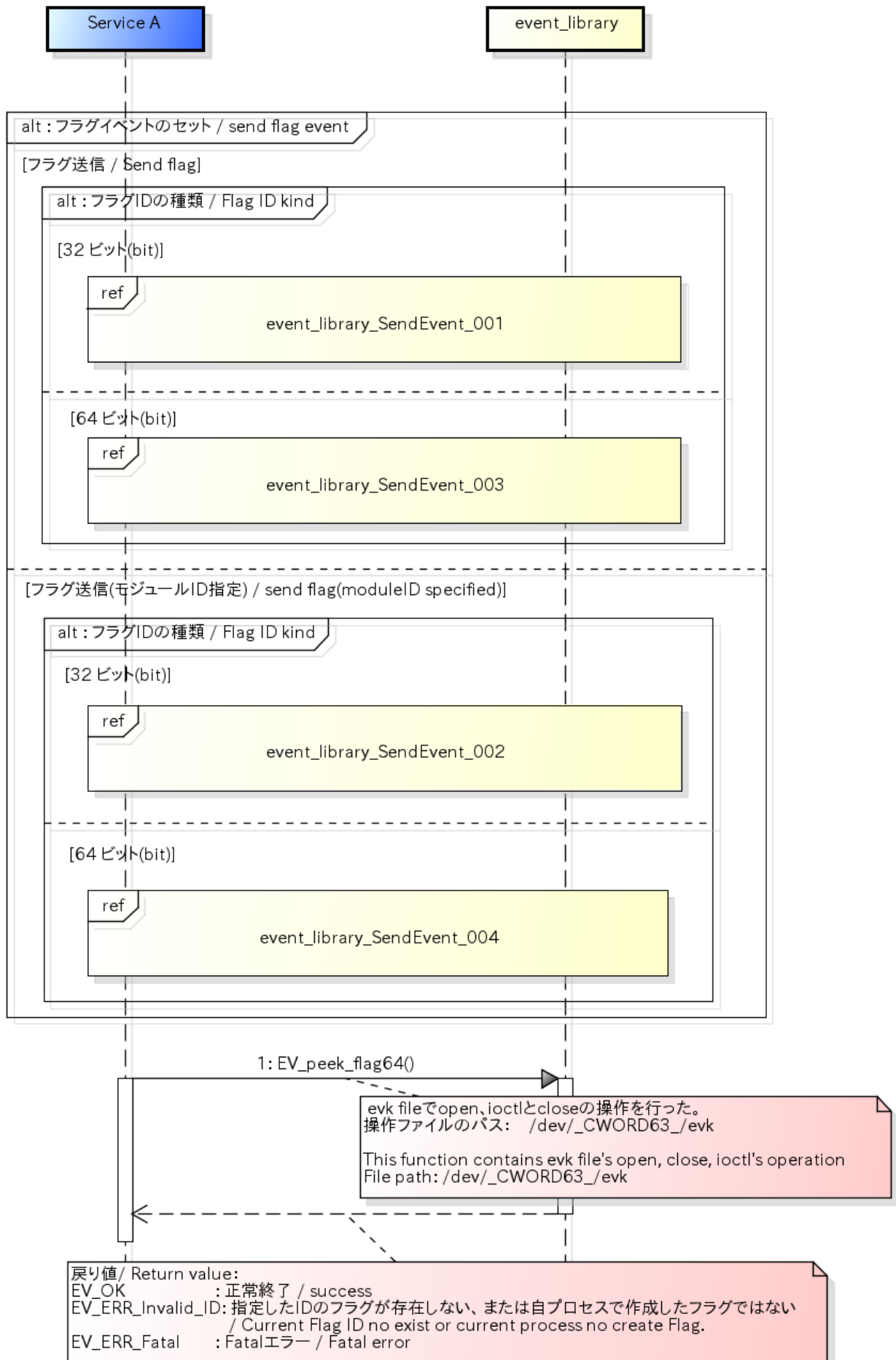
非破壊的：リードオンリー的方式でメッセージキューからメッセージ内容を取得するので、メッセージキュー中のメッセージ内容をクリアしない。

Get 64bit's flag event(non-destructive)

non-destructive: Because we get message contents from message queue by read only method, so it will not clear the message contents.

シーケンス [[Sequence](#)]

sd event_library_ReceiveEvent_009



ユースケースevent_library_DeleteEvent_001 [*use-case event_library_DeleteEvent_001*]

概要 [*Overview*]

フラグイベントの削除

Delete a flag event

シーケンス [*Sequence*]

sd event_library_DeleteEvent_001

Service A

event_library

alt : フラグ作成 / Create Flag:

[フラグを作成する / Make flag]

alt : フラグIDの種類 / Flag ID kind

[32ビット(bit)]

ref

event_library_EventInitial_002

[64ビット(bit)]

ref

event_library_EventInitial_006

[指定されたIDのフラグを作成する / Create a flag whose ID is specified]

alt : フラグIDの種類 / Flag ID kind

[32ビット(bit)]

ref

event_library_EventInitial_003

[64ビット(bit)]

ref

event_library_EventInitial_007

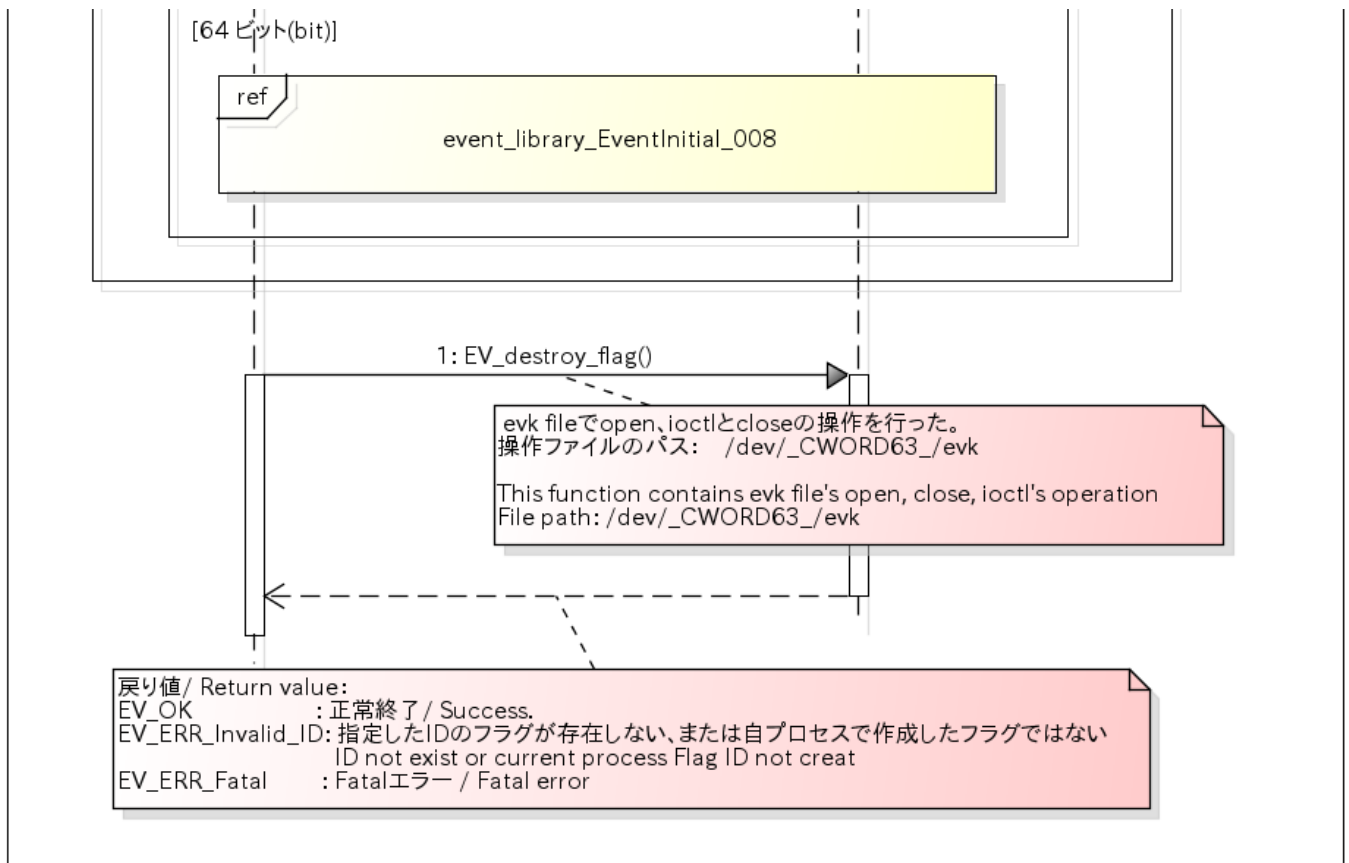
[フラグIDを自動割当し、フラグを作成する / create a flag(ID is allocated automaticlly)]

alt : フラグIDの種類 / Flag ID kind

[32ビット(bit)]

ref

event_library_EventInitial_004



ユースケースevent_library_DeleteEvent_002 [[use-case event_library_DeleteEvent_002](#)]

概要 [[Overview](#)]

フラグ削除(mID指定)

Delete flag(mID specified)

シーケンス [[Sequence](#)]

Service A

event_library

alt : フラグ作成 / Create Flag:

[フラグを作成する / Make flag]

alt : フラグIDの種類 / Flag ID kind

[32ビット(bit)]

ref

event_library_EventInitial_002

[64ビット(bit)]

ref

event_library_EventInitial_006

[指定されたIDのフラグを作成する / Create a flag whose ID is specified]

alt : フラグIDの種類 / Flag ID kind

[32ビット(bit)]

ref

event_library_EventInitial_003

[64ビット(bit)]

ref

event_library_EventInitial_007

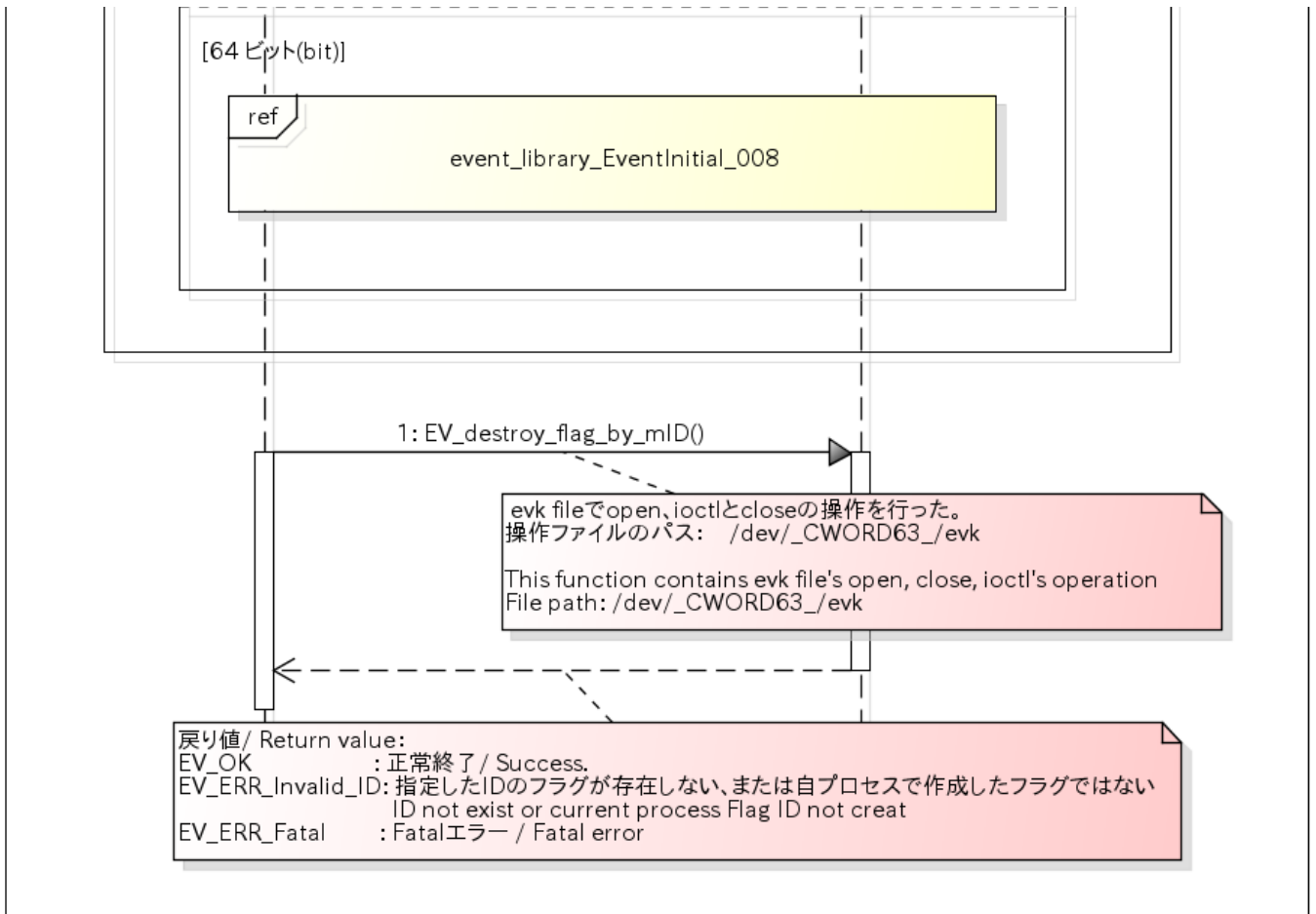
[フラグIDを自動割当し、フラグを作成する / create a flag(ID is allocated automatically)]

alt : フラグIDの種類 / Flag ID kind

[32ビット(bit)]

ref

event_library_EventInitial_004



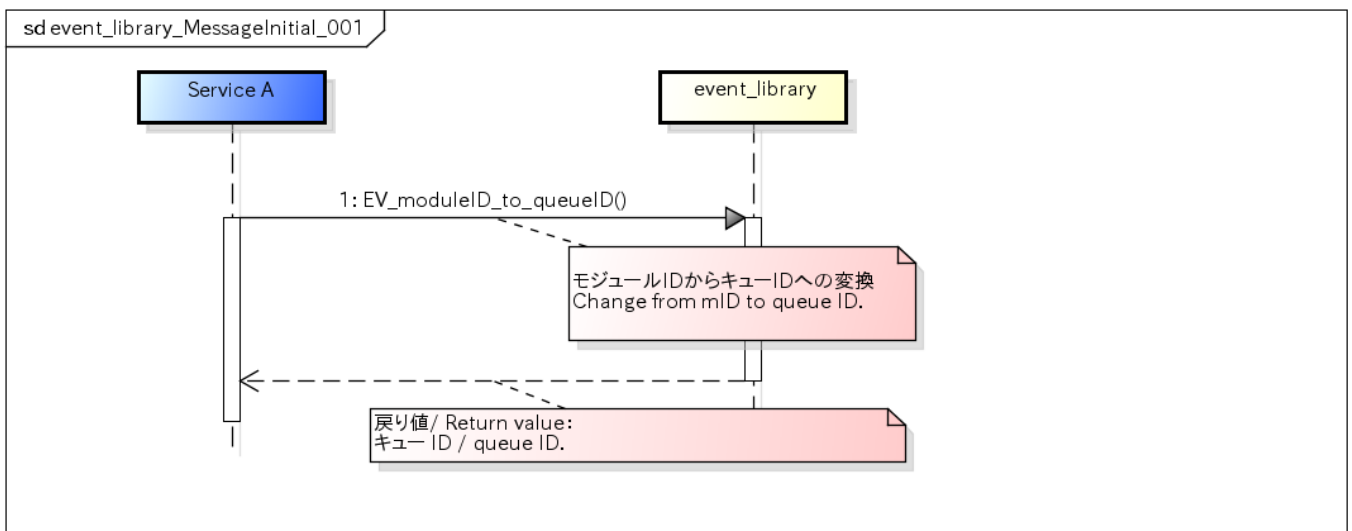
ユースケースevent_library_MessageInitial_001 [[use-case event_library_MessageInitial_001](#)]

概要 [[Overview](#)]

mIDからキューIDへ変換する

Change from mID to queue ID.

シーケンス [[Sequence](#)]



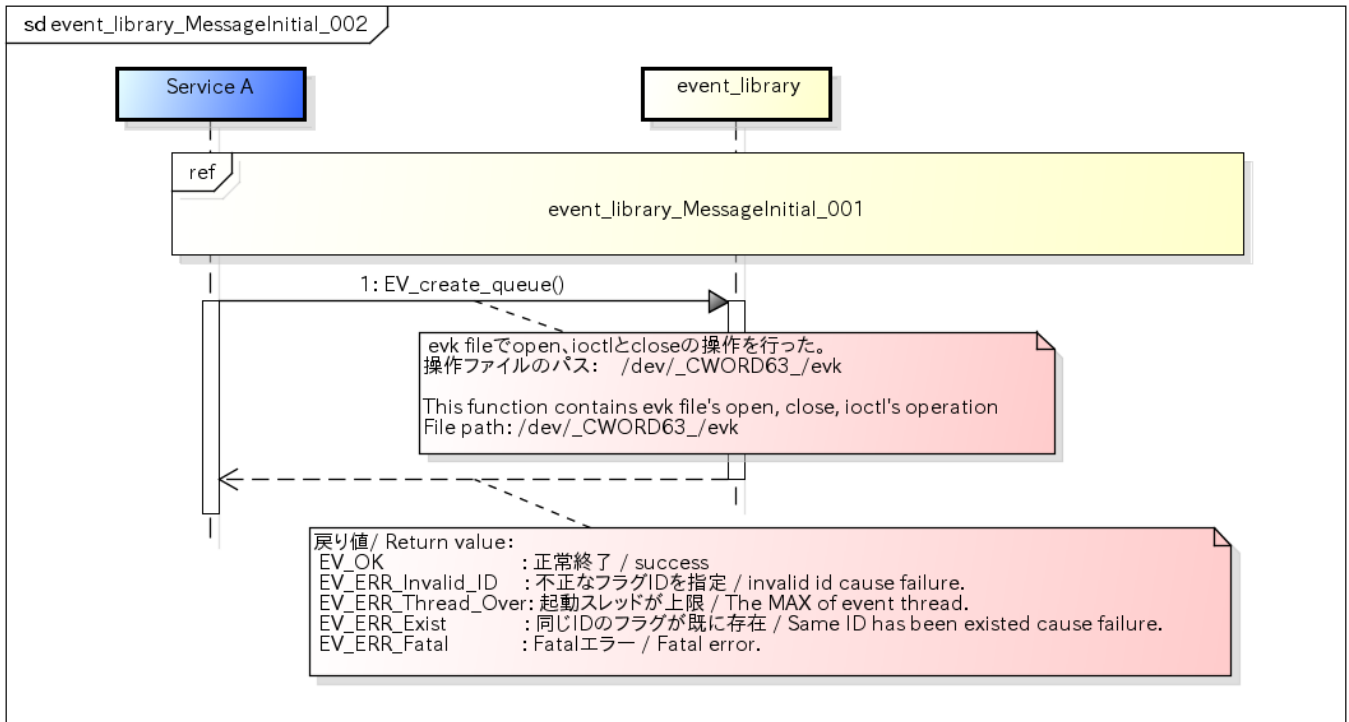
ユースケースevent_library_MessagelInitial_002 [[use-case event_library_MessagelInitial_002](#)]

概要 [[Overview](#)]

指定されたIDのメッセージイベントのキューを作成する

Create a message event queue whose ID is specified

シーケンス [[Sequence](#)]



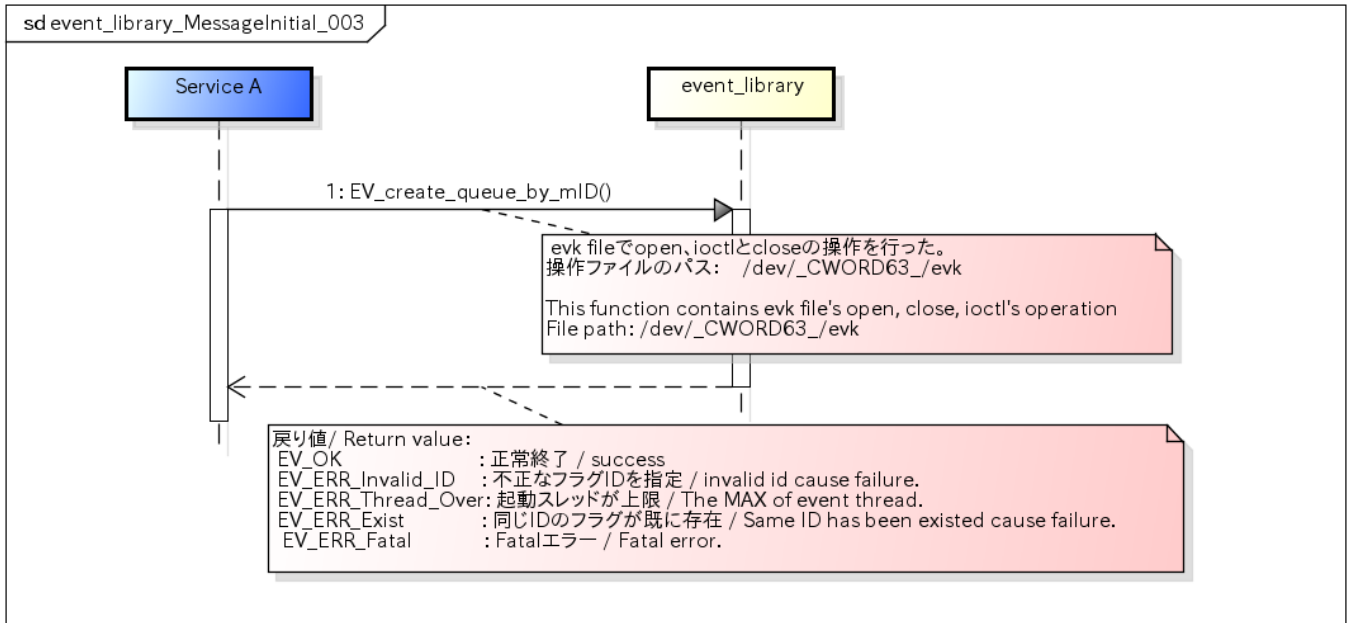
ユースケースevent_library_MessagelInitial_003 [[use-case event_library_MessagelInitial_003](#)]

概要 [[Overview](#)]

キュー作成(mID指定)

Create queue(mID specified)

シーケンス [[Sequence](#)]



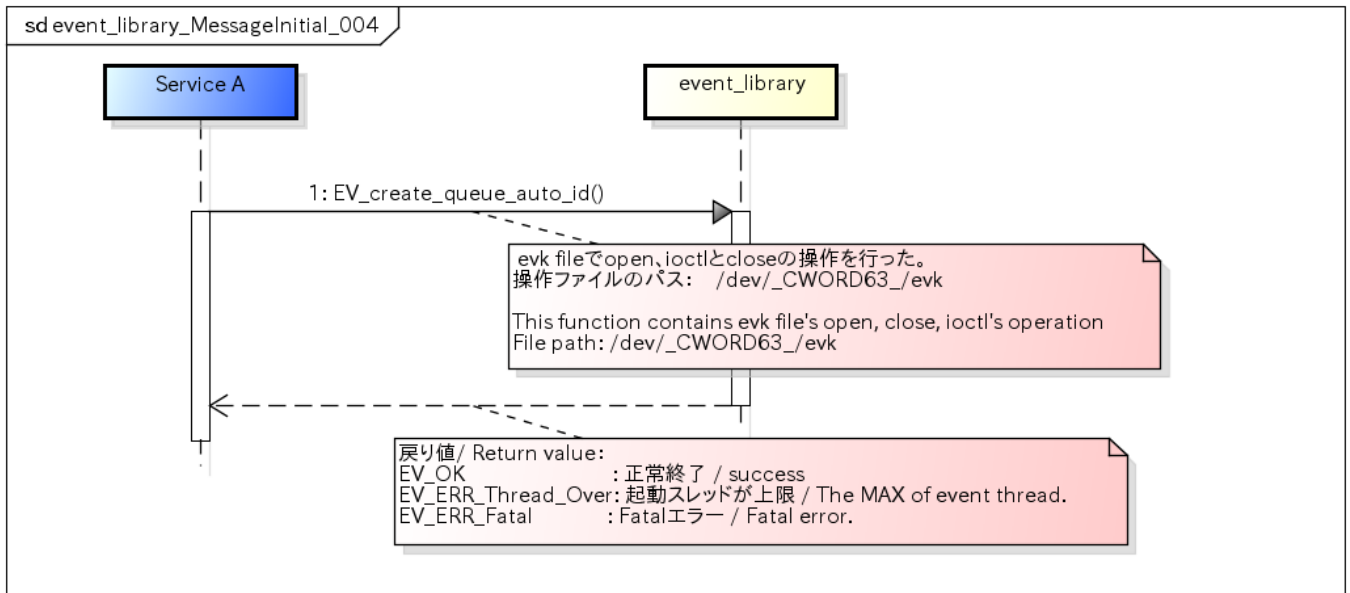
ユースケースevent_library_MessageInitial_004 [[use-case event_library_MessageInitial_004](#)]

概要 [[Overview](#)]

メッセージイベントキュー作成(ID自動割り当て)

[Create a message event queue\(ID is allocated automatically\)](#)

シーケンス [[Sequence](#)]



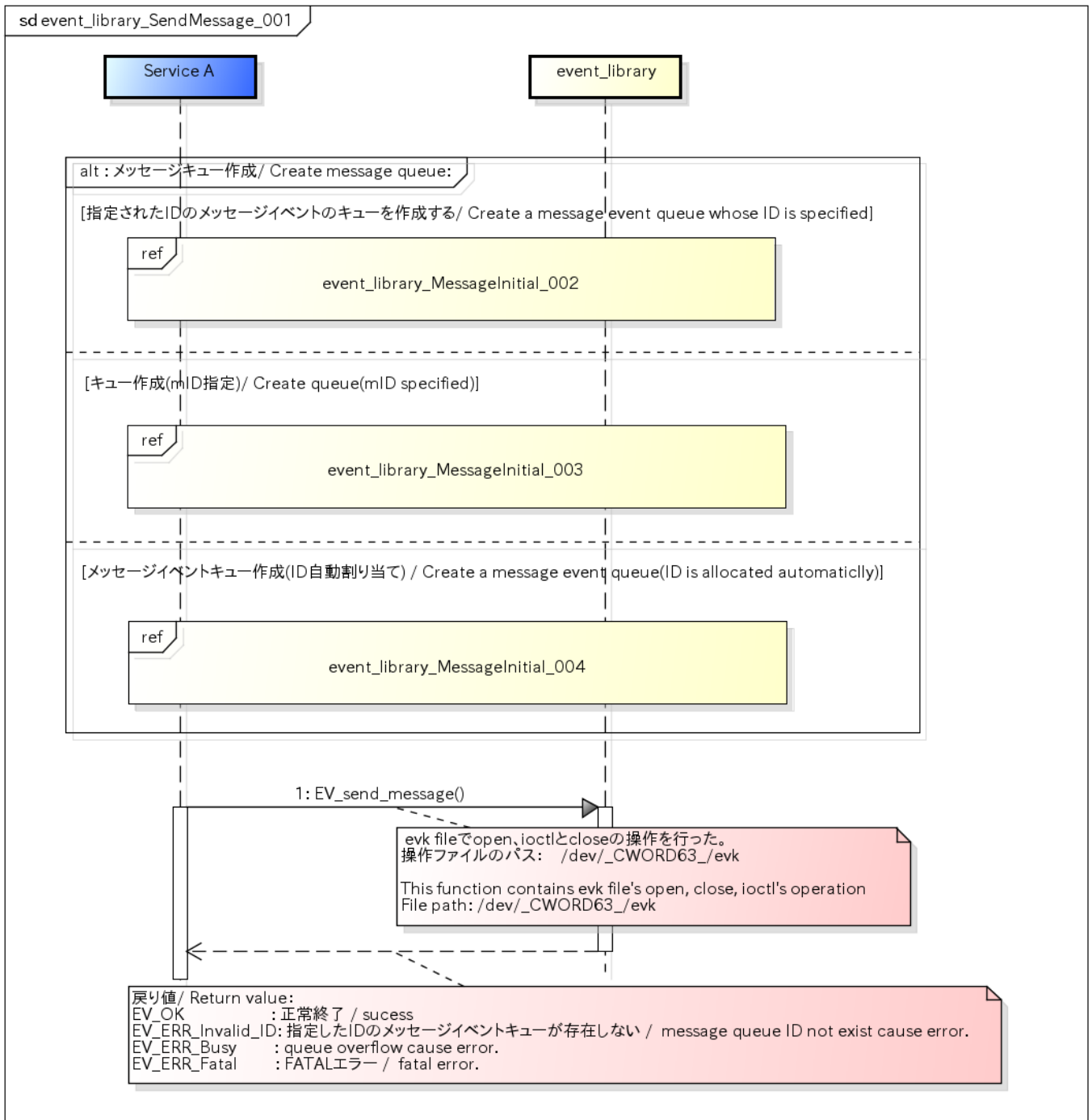
ユースケースevent_library_SendMessage_001 [[use-case event_library_SendMessage_001](#)]

概要 [[Overview](#)]

メッセージイベントの送信

[Send message event](#)

シーケンス [Sequence]



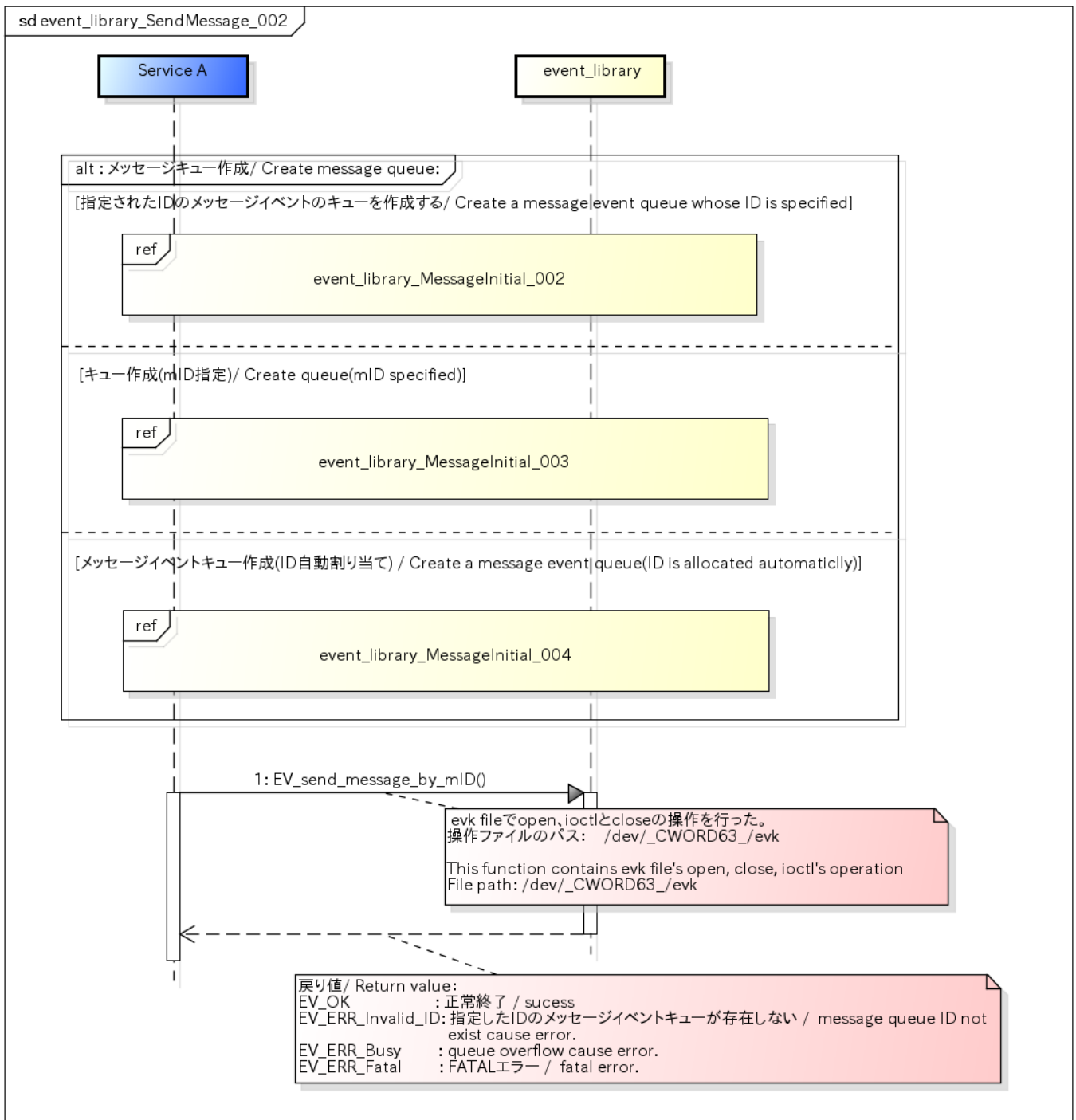
ユースケースevent_library_SendMessage_002 [use-case event_library_SendMessage_002]

概要 [Overview]

メッセージ送信(mID指定)

Send message(mID specified)

シーケンス [Sequence]



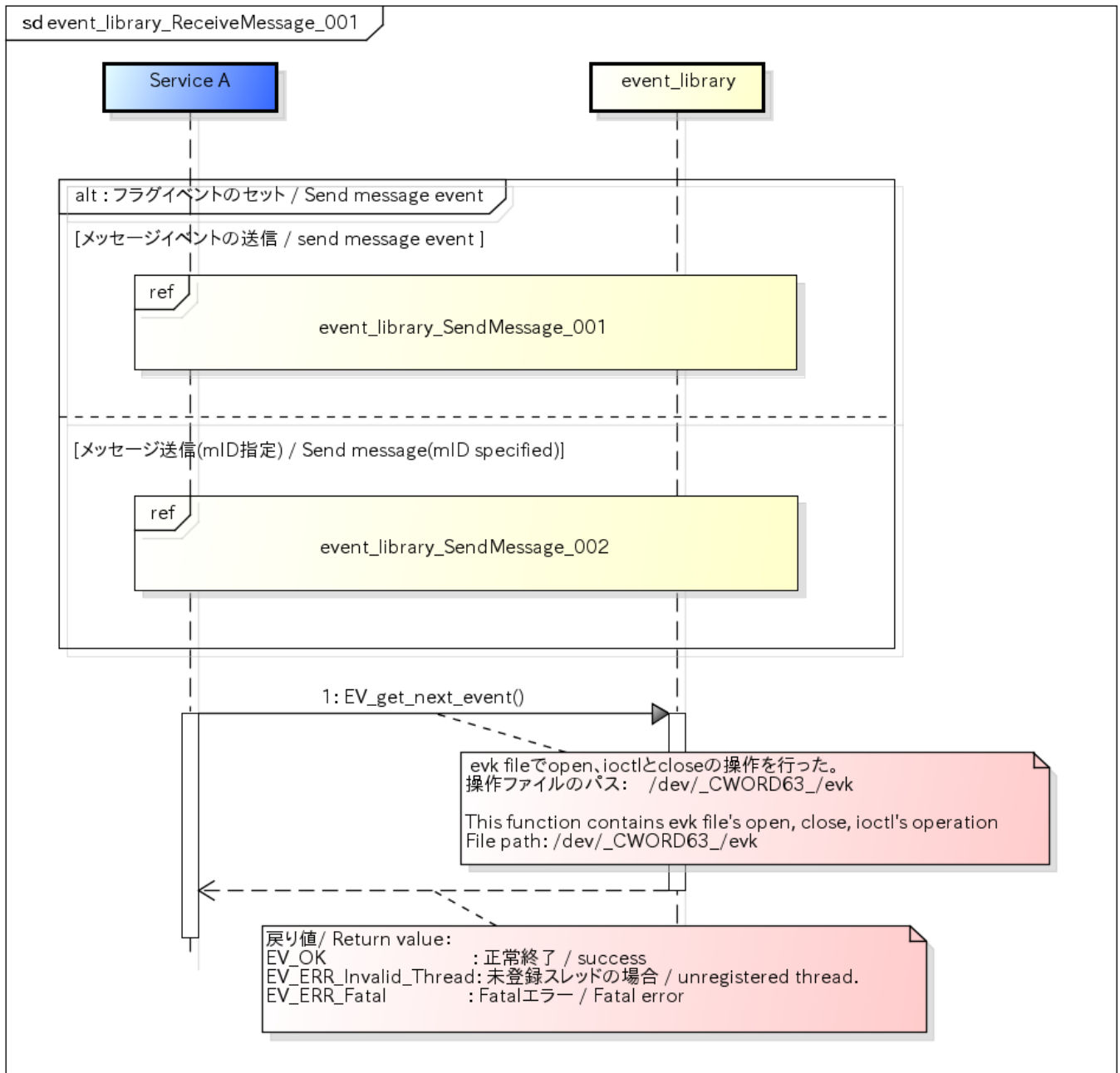
ユースケース event_library_ReceiveMessage_001 [use-case event_library_ReceiveMessage_001]

概要 [Overview]

最初に到着したイベントの取得(non-block)

Get the first reached event(non-block)

シーケンス [Sequence]



ユースケースevent_library_ReceiveMessage_002 [\[use-case event_library_ReceiveMessage_002\]](#)

概要 [\[Overview\]](#)

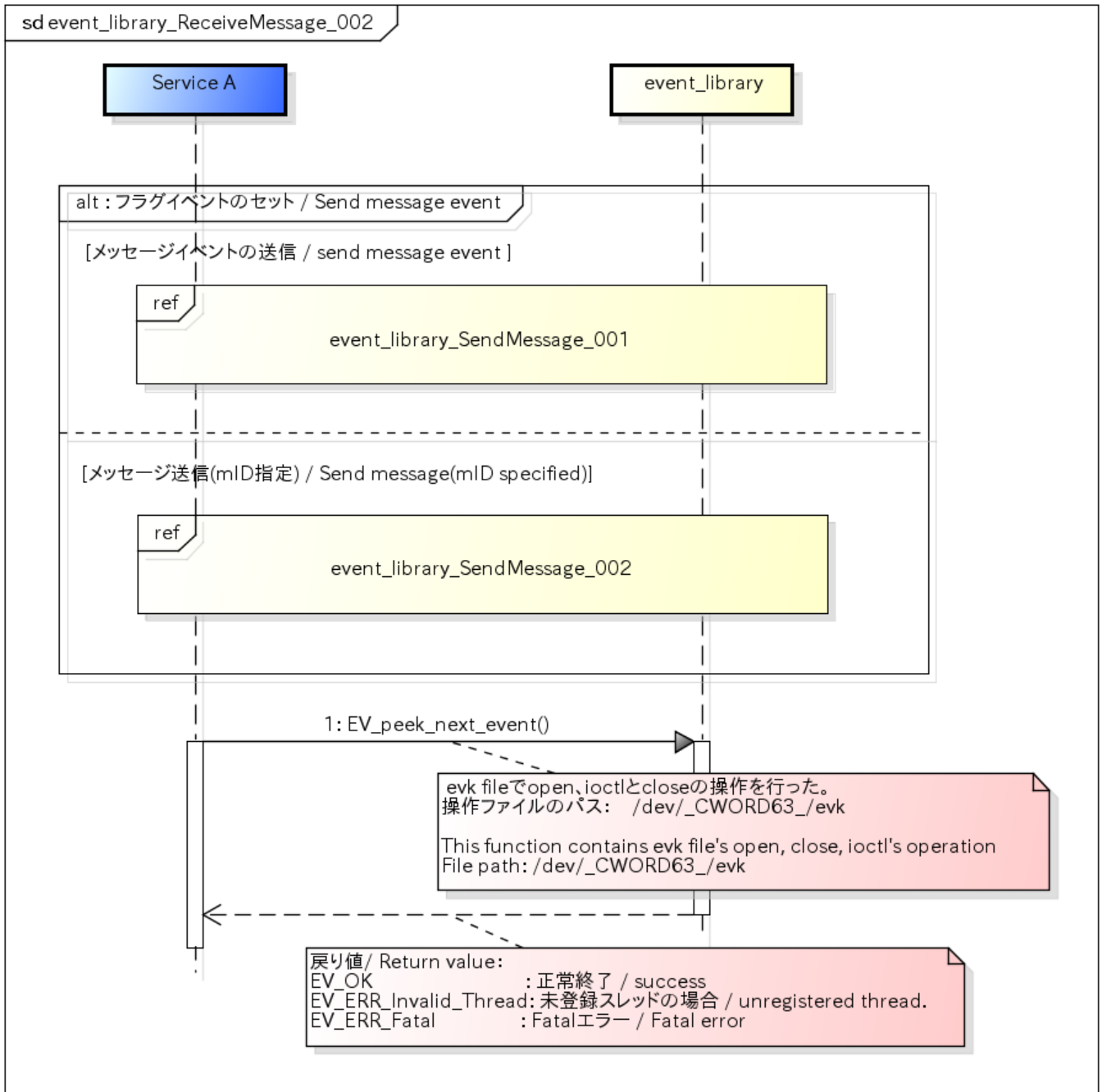
最初に到着したイベントの取得(非破壊的).

非破壊的: リードオンリー的方式でメッセージキューからメッセージ内容を取得するので、メッセージキュー中のメッセージ内容をクリアしない。

Get the first reached event(non-destructive).

non-destructive: Because we get message contents from message queue by read only method, so it will not clear the message contents.

シーケンス [\[Sequence\]](#)



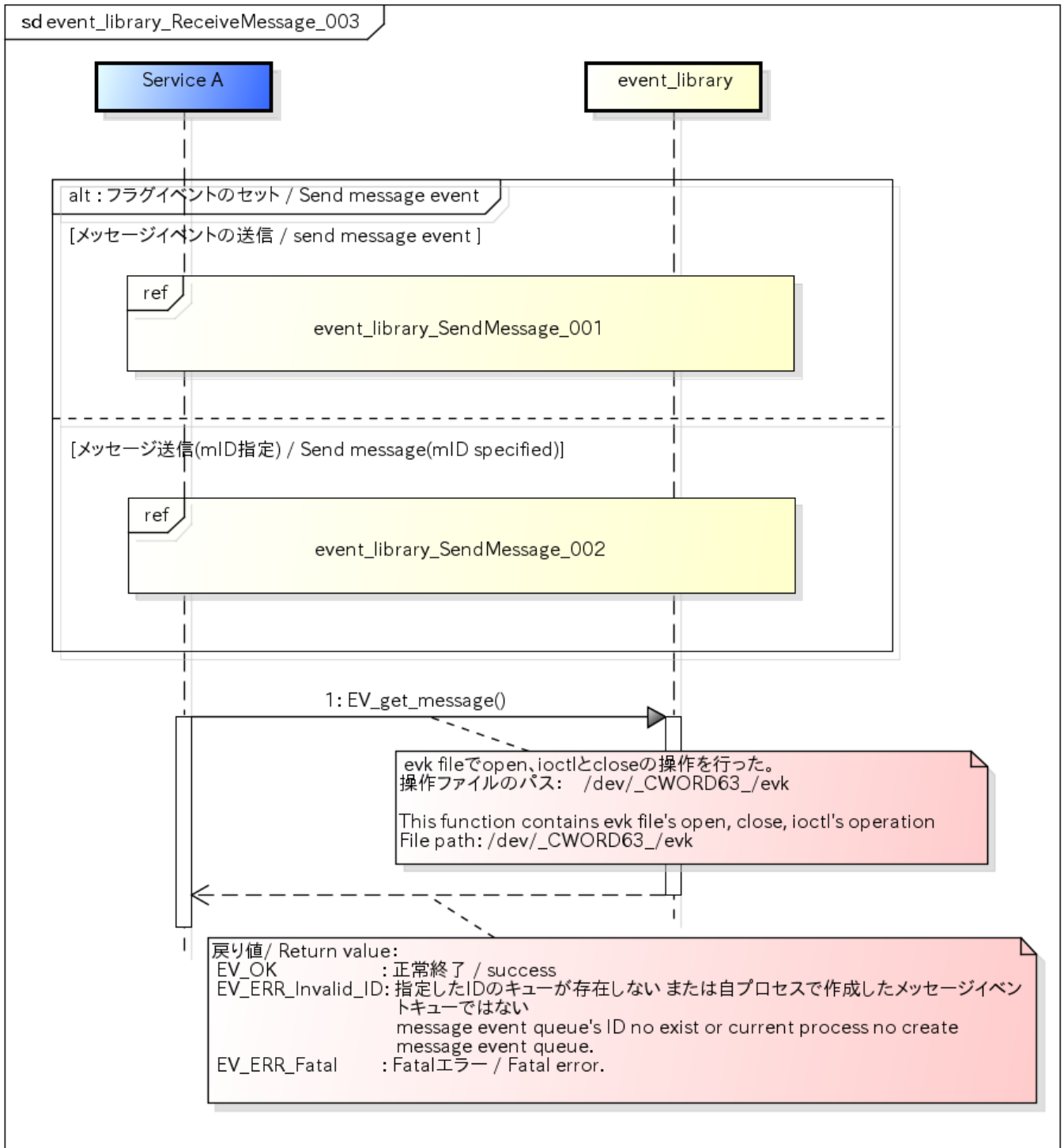
ユースケースevent_library_ReceiveMessage_003 [\[use-case event_library_ReceiveMessage_003\]](#)

概要 [\[Overview\]](#)

メッセージイベントの取得(non-block)

[Get message event queue\(ID specified\)](#)

シーケンス [\[Sequence\]](#)



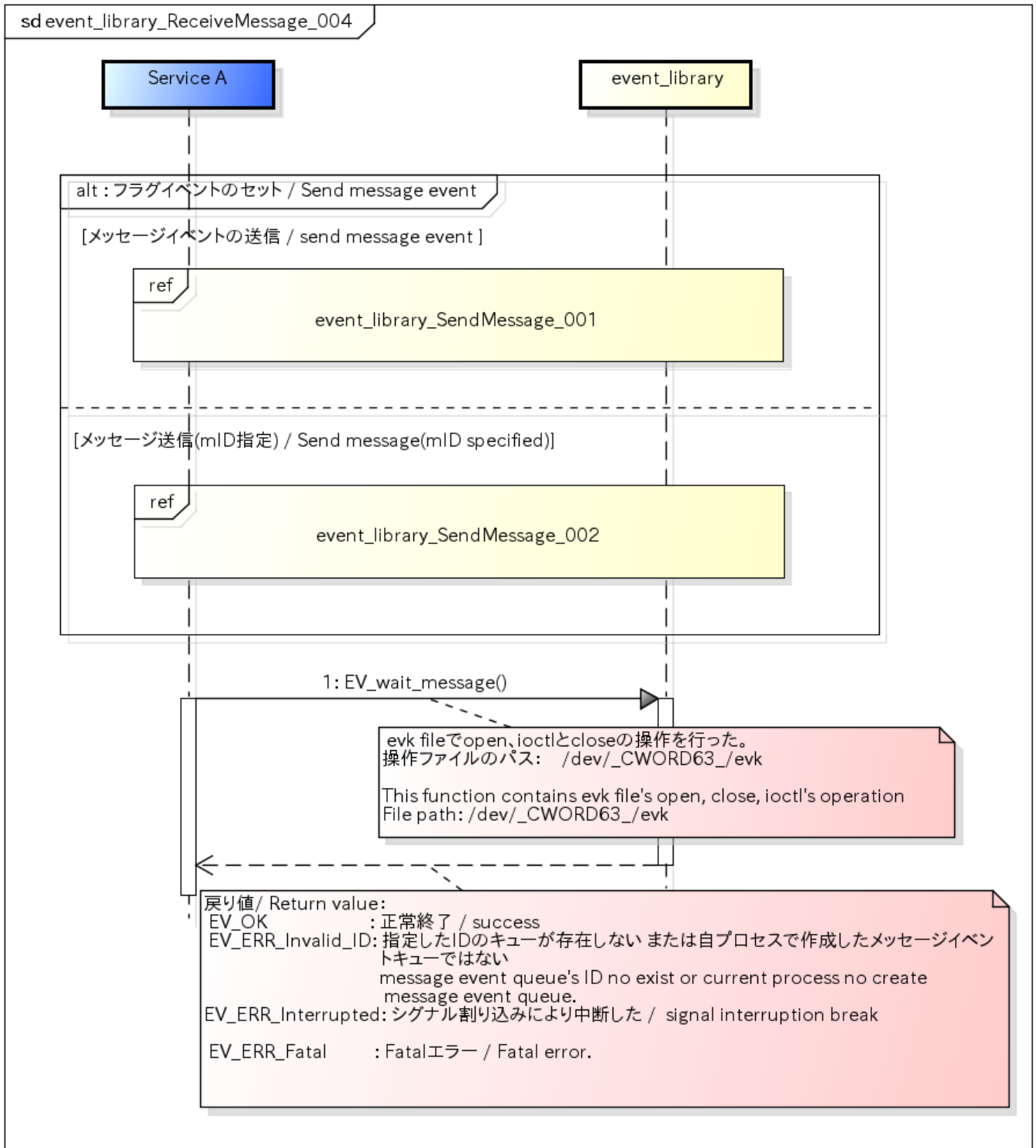
ユースケースevent_library_ReceiveMessage_004 [\[use-case event_library_ReceiveMessage_004\]](#)

概要 [\[Overview\]](#)

メッセージイベントの取得(block)

[Get message event queue\(block\)](#)

シーケンス [\[Sequence\]](#)



ユースケースevent_library_ReceiveMessage_005 [[use-case event_library_ReceiveMessage_005](#)]

概要 [[Overview](#)]

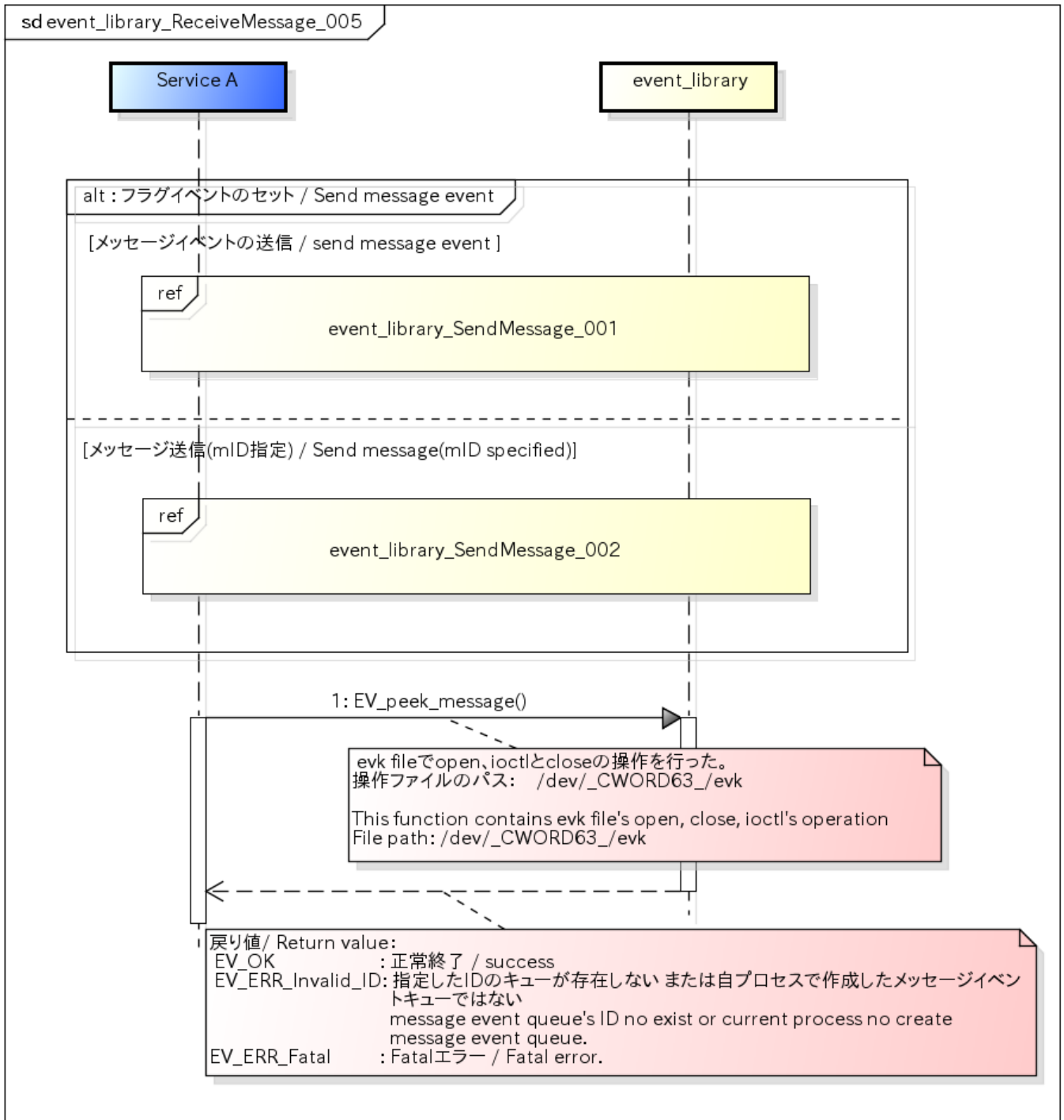
メッセージイベントの取得(非破壊的).

非破壊的: リードオンリーの方式でメッセージキューからメッセージ内容を取得するので、メッセージキュー中のメッセージ内容をクリアしない。

[Get message event\(non-destructive\)](#).

non-destructive: Because we get message contents from message queue by read only method, so it will not clear the message contents.

シーケンス [Sequence]



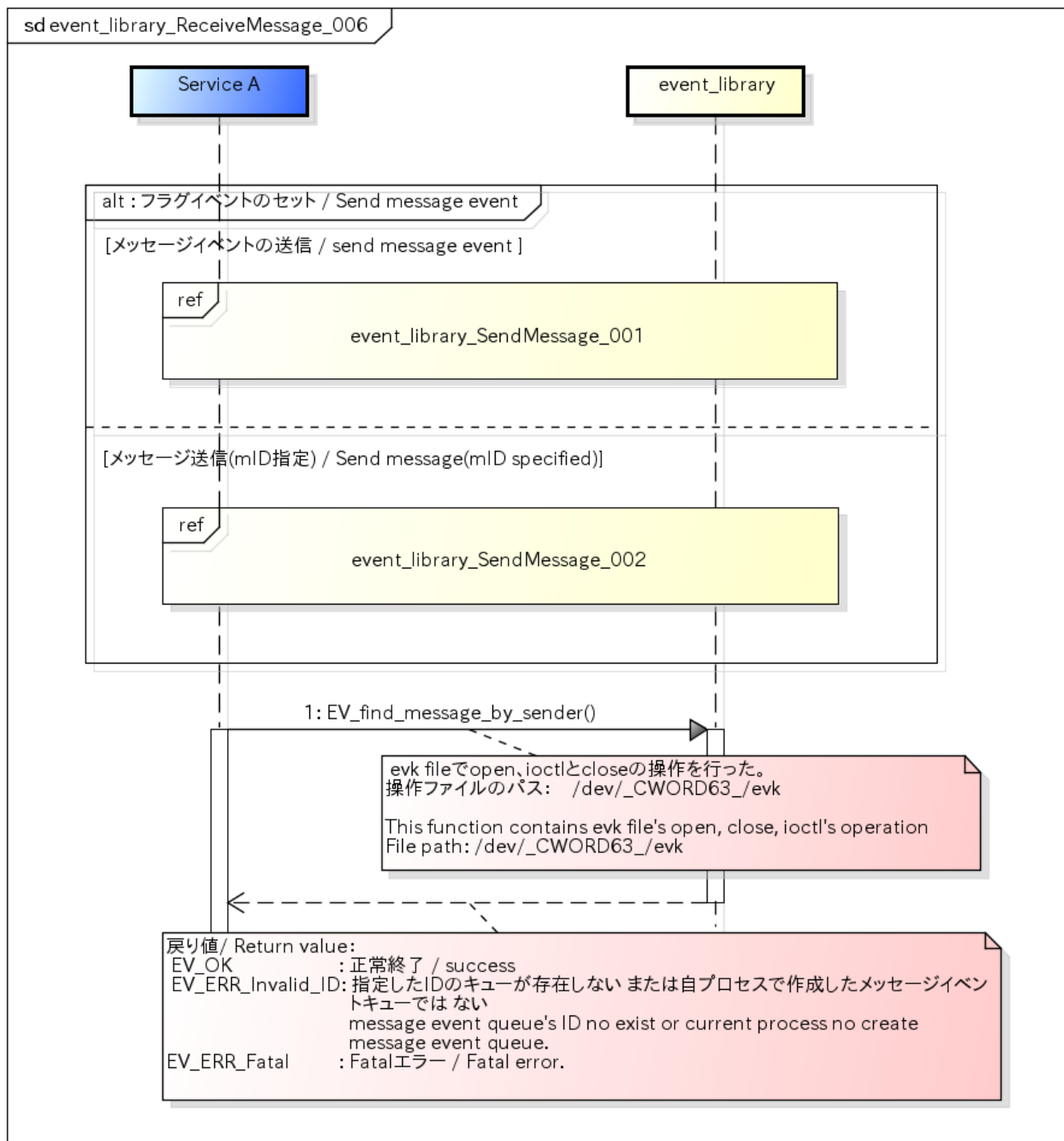
ユースケース event_library_ReceiveMessage_006 [use-case event_library_ReceiveMessage_006]

概要 [Overview]

メッセージイベントの取得(送信元情報で検索)

[Get message event\(search with sender information\)](#)

シーケンス [Sequence]



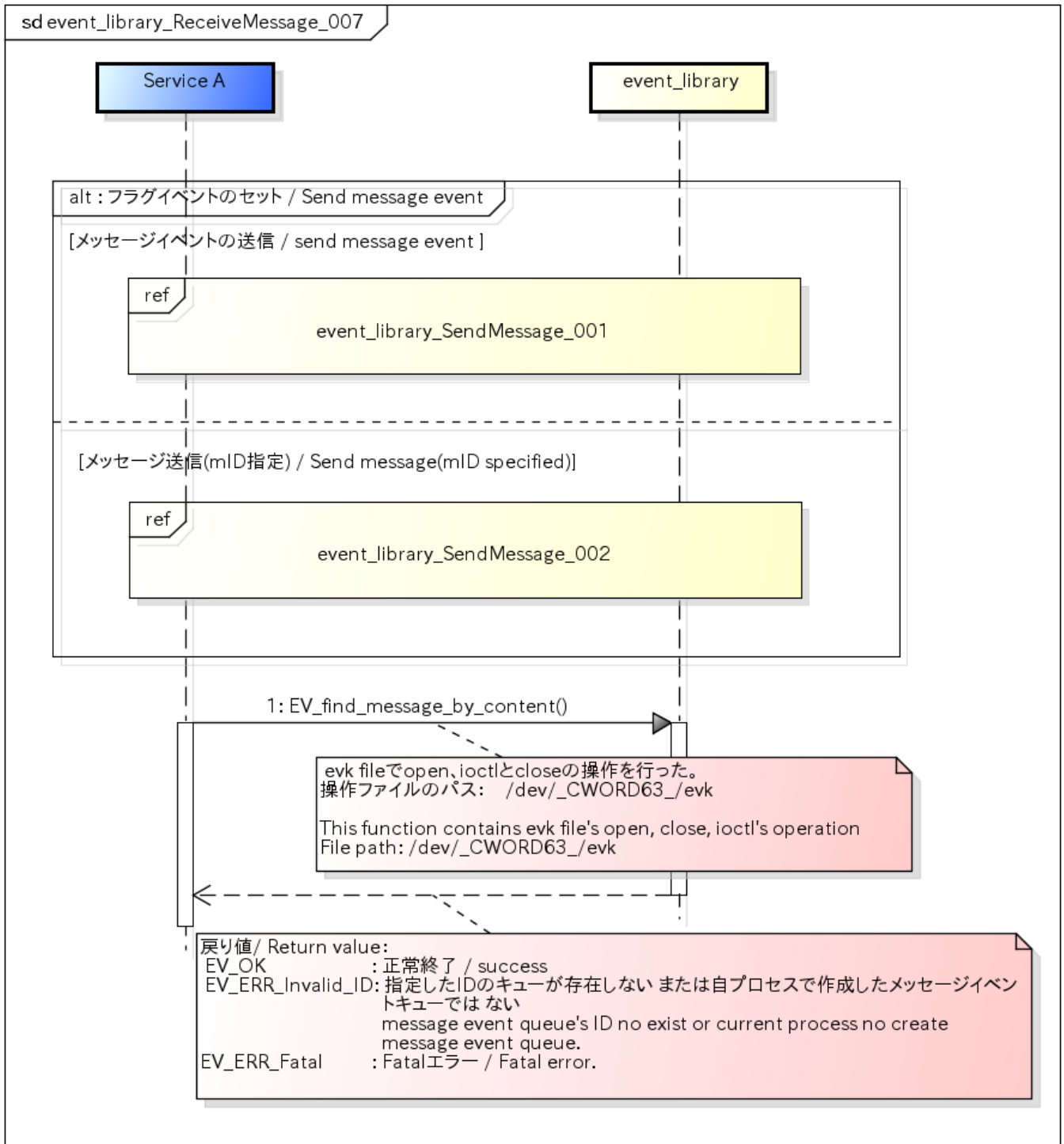
ユースケースevent_library_ReceiveMessage_007 [use-case event_library_ReceiveMessage_007]

概要 [Overview]

メッセージイベントの取得(内容で検索)

[Get message event\(search with contents\)](#)

シーケンス [Sequence]



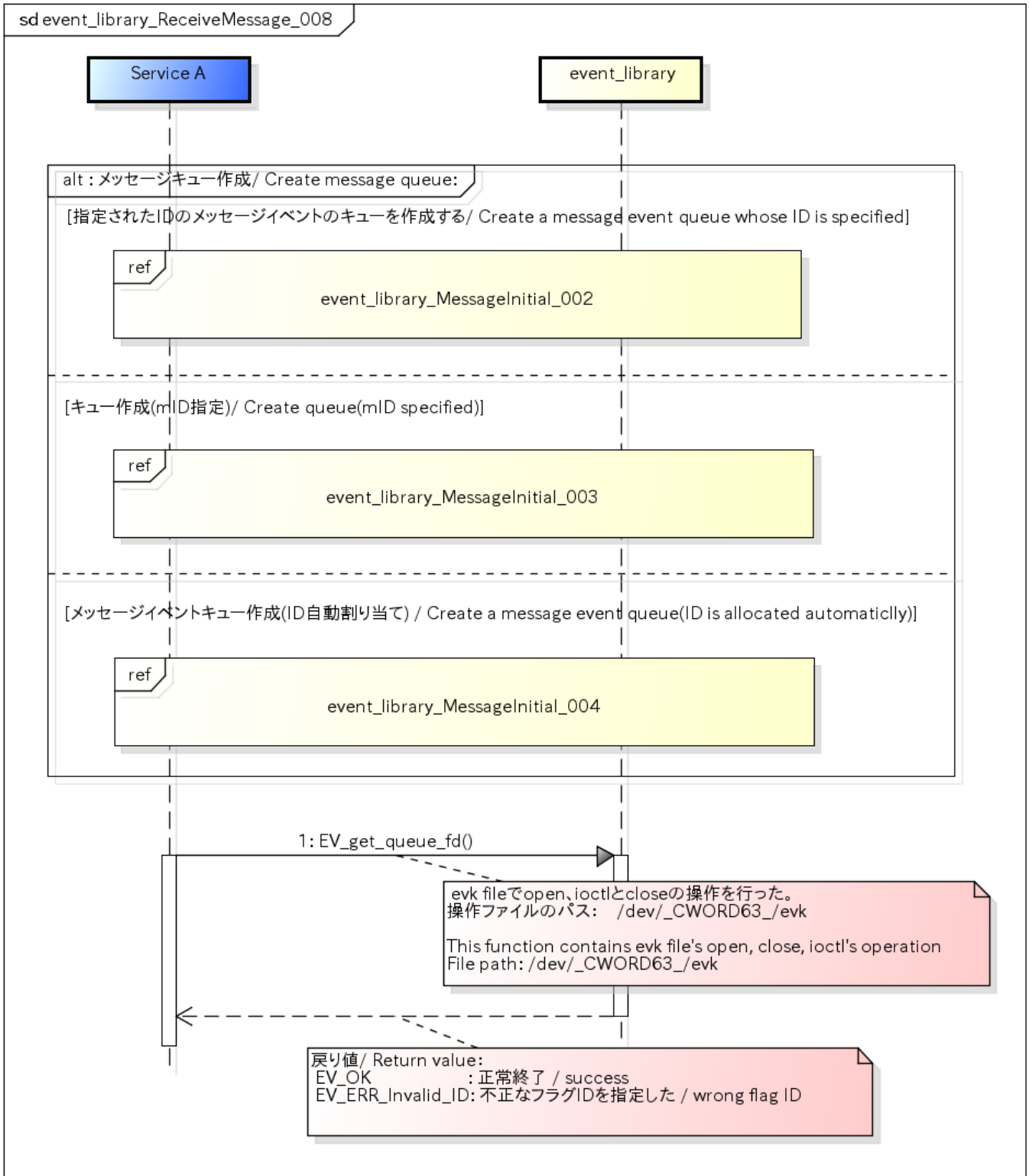
ユースケースevent_library_ReceiveMessage_008 [\[use-case event_library_ReceiveMessage_008\]](#)

概要 [\[Overview\]](#)

メッセージイベントキューのpoll用fdの取得

Get fd which is used to poll message event queue

シーケンス [\[Sequence\]](#)



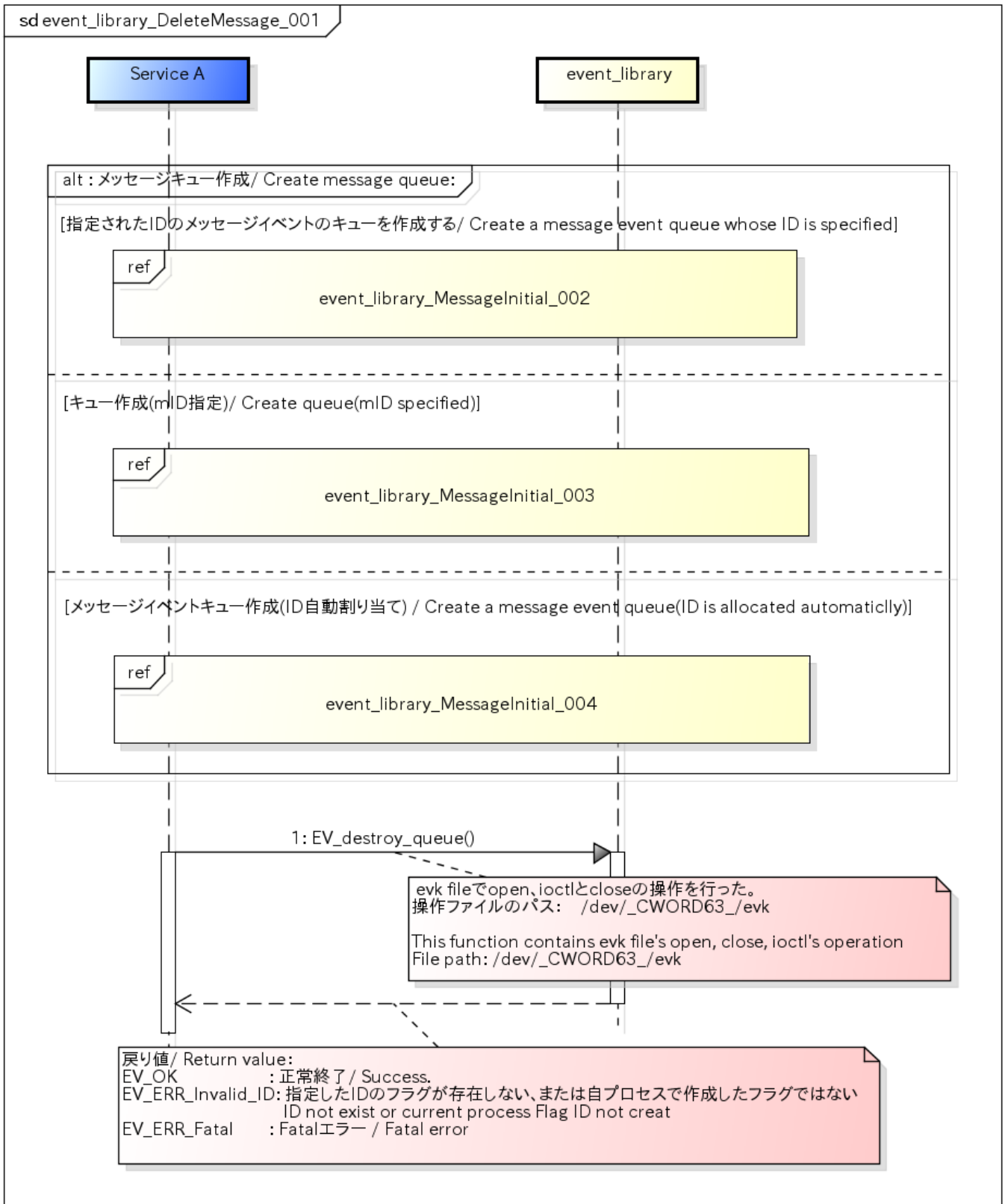
ユースケースevent_library_DeleteMessage_001 [\[use-case event_library_DeleteMessage_001\]](#)

概要 [\[Overview\]](#)

メッセージイベントキューの削除

Delete a message event queue

シーケンス [\[Sequence\]](#)



ユースケースevent_library_DeleteMessage_002 [use-case event_library_DeleteMessage_002]

概要 [Overview]

キュー削除(mID指定)

シーケンス [Sequence]

