

Vehicle.Cabin.Window API解説書

バージョン: 202509α

発行: 2025年9月30日

更新履歴

2025/03/31

- ・新規作成

2025/06/30

- ・ウインドウAPIの解説更新(202503α内部Ver)

2025/09/10

- ・最新検討状況へ情報更新

2025/09/28

- ・最新検討状況へ情報更新

202509α API策定方針

1. 実装可能なレベルのAPI定義を目標とする。

- ▶ 202503αは、ビークルAPIとは何かを世に問うことを優先した。
- ▶ この版では、実装が十分に検討できることが可能なレベルまでAPIを定義することを目標にした。

2. 重点検討対象のオブジェクトはウィンドウに絞った。

- ▶ 当初は、202503αで公開した全オブジェクトのAPI更新を目標とした。
- ▶ ウィンドウAPIの検討に想定より時間がかかり、すべてのオブジェクトのAPIを更新することより、ウィンドウAPI定義の記述レベルを上げることを優先した。

- API定義は、YAMLで記述した。
- 本ドキュメントの記法とAPIコンセプトWG発行情書の記法が異なっている場合は、APIコンセプトWG発行情書の記法を正とする。
- 抽象的(型名がTypeで終わる型)に対して、型に対する要件を明確にした。
 - 列挙型は、列挙定数まで定義した。
 - 列挙定数は、大文字で表現し、スネークケースで記述した。
- 論理定義の記述に注力しているが、一部物理定義(実装)について補足した。

ウィンドウAPI一覧

情報取得

getConfigAll

getStatus

getErrorString?

制御操作

startMove

startMove2?

stopMove

lock

unlock

イベントの制御

notify

unnotify

getEvent?

API名の末尾が?となっているAPIは、オプション扱いとする。
型名の末尾が?となっている属性(変数)は、オプション扱いとする。
このドキュメントでは、見やすくするために、?を赤字にした。

ReturnType 型定義

OSDVIで定めるAPIの実行結果を表現する型

列挙型とする。以下の列挙定数を持つ。

```
E_OK, // 正常終了
E_INVALID_ID, // IDが不正
E_INVALID_HANDLE, // ハンドルが不正
E_INVALID_PARAMETER, // その他のパラメータが不正
E_RISK_APPLICATION, // アプリケーションが対応していないリスクがある
E_RISK_USER, // ユーザがリスクを受け入れていない
E_DENIED_PRIORITY, // 使用できない操作優先度
E_DENIED_SERVICE, // サービスコールの呼び出しが認められていない
E_OBJECT_FAULT, // 対象オブジェクトが故障中
E_OBJECT_LOCKED, // 対象オブジェクトがロックされている
E_OBJECT_STATUS, // 対象オブジェクトが指定された操作をできない状態に
// ある
E_NO_DATA, // データがない
E_NOT_SUPPORTED, // サポートされていない機能
E_NOT_OK // その他のエラー
```

補足)この列挙定数が多く該当する場合は、必要に応じて列挙定数を増やす。

WindowConfigType メンバの型定義(1)

WindowConfigType のメンバが持つ型の定義を行う。

IdType

- インスタンスが持つIdの型
- 1以上の整数とするが連続である必要はない。
- 0に特別な意味を持たせる(APIにより解釈は異なる)。0は特定のインスタンスとならない。
- オブジェクトのすべてのインスタンスに異なるIdを割り当て可能であること。(1~65535)
 - 補足)物理的なオブジェクト(ウインドウなど)は100で十分である。

PlacementType

- オブジェクトの場所を表す型
- 物理定義の暫定案:何列目(uint8:row)と左右中上下(enum:zone)を組みにした構造体とする。

```
- TypeName: PlacementType
  type: struct
  member:
    - type: UInt8
      name: row
      description: "何列目 (uint8) "
    - type: enum
      name: ZoneType
      enum:
        LEFT: "左"
        RIGHT: "右"
        CENTER: "中"
        TOP: "上"
        BOTTOM: "下"
      description: "左右中上下 (enum) "
```

WindowConfigType メンバの型定義(2)

WindowConfigType のメンバが持つ型の定義を行う。

WindowCapabilityType

ウィンドウが持つ機能的な機能を表す機能クラス

- 以下の3つの列挙定数を持つ列挙型とする。
 - MOVABLE
APIにより開閉できる場合 True
 - CLOSED_DETECTABLE
全閉が検出できる場合 True
 - HAS_MOVE2
Move2(二次元作動)を持つ場合 True

WindowRiskClassType

ウィンドウの安全性クラスを表す型

- 以下の2つの列挙定数を持つ列挙型とする。
 - RISK_PINCH, // 挟み込み
 - RISK_OPEN // 開動作に伴うリスク全般(落下・盗難)

WindowConfigType メンバの型定義(3)

WindowConfigType のメンバが持つ型の定義を行う。

RelatedObjectType

- 関連するオブジェクトを指定する型
 - オブジェクトにより厳密な定義は異なる。
 - ウィンドウの場合、どのドア(もしくはキャビン)に付いているかを表す。
- 以下の2つのメンバを持つ構造体
 - 上位のオブジェクト種類 objectKindType objectKind
 - objectKindTypeについては、次スライドで定義する。
 - ウィンドウの場合は、DOORもしくはCABINになる。
 - オブジェクト名 String instanceName
 - 補足)物理定義(実装)では、instanceNameをinstanceIdに変更して返してもよい。

ObjectKindType 型定義

OSDVIのすべてのオブジェクトを表す型

列挙型とする。

202509 α では、以下の列挙定数について定義する(性質上、順次追加されていく)。

- VEHICLE
車両全体を表す。
- BODY
ボディーを表す。
- CABIN
キャビン(乗員スペース)を表す。
- DOOR
乗員が乗降するドアを表す。
- WINDOW
ウインドウを表す。
OSDVIの定義では、サンルーフはウインドウの一種とした。

WindowConfigType 型定義

WindowConfigType は、以下のメンバを持つ構造体とする。

String instanceName

インスタンスの名前(識別子)

IdType instanceId

インスタンスが持つId

PlacementType placement

インスタンスがある場所

enumSet(WindowCapabilityType) capabilities

そのウインドウが持つ機能

enumSet(WindowRiskType) riskClasses

そのウインドウのリスククラスの集合

RelatedObjectType mountedOn

関連するオブジェクト

ウインドウの場合は、どのドアについているか、またはキャビンについているかを表す。

String displayName

HMIに表示する文字列

アプリがもつロケールに対応した文字列を返すため、以下の前提がある。

ビークルサービスは、サービスコールを呼び出したアプリケーションのロケール(言語および地域設定)を取得できるものとする。

getConfigAll API 定義

複数インスタンス全てのウィンドウ全体の静的コンフィグレーションを返す。
WindowConfigType型の可変長配列を返す。

ServiceCalls:

- name: getConfigAll
summary: ウィンドウ構成情報取得 (マルチインスタンス)
returnParameters:
 - name: returnValue
type: Returntype
 - name: config
type: WindowConfigType[]

車両config変更時のビークルOSへの要求

- オブジェクトが持つ静的なコンフィグ情報(getConfig の出力値)が変更された場合、システムイベントを設定する。
- システムイベントの詳細は、**ウィンドウオブジェクトを超えた車両全体に影響するものであり 202509αとしてはTBD項目とする。**

WindowStateType メンバの型定義(1)

WindowStateType のメンバが持つ型の定義を行う。

WindoMoveStatusType

- ウィンドウ特定次元方向の作動状態を表す型
- 以下の列挙定数をもつ列挙型とする。
 - STOPPED
停止中
 - OPENING
開方向へ移動中(開作動、チルトアップなど)
 - CLOSING
閉方向へ移動中(閉作動、チルトダウンなど)
 - FAULT
故障中

WindowStateType メンバの型定義(2)

WindowStateType のメンバが持つ型の定義を行う。

LockStatusType

- ロック状態と関連属性を保持する型
- 以下のメンバを持つ構造体とする。
 - Bool locked
 - ロック状態を表す。
 - ApplicationIdType? lockApplication
 - ロックしているアプリケーションを表す。
 - ApplicationIdType の物理定義は採用OSに大きく依存する。
 - PriorityType? lockPriority
 - ロック操作を行った操作優先度
 - PriorityTypeの定義は次スライドで後述する。

WindowStateType メンバの型定義(3)

WindowStateType のメンバが持つ型の定義を行う。

PositionType

- Movable object の開閉度を表す型
- 0は全閉、100は全開を表し、その中間の値は、開閉度の%とする。
- 0-100、不明、0以外(全閉ではないこと以外わからない場合)が表現できること。

PriorityType

- 操作優先度を表す型
- 1を最高優先度とする。1-100 が表現できること
- 1-100に加えて、そのアプリに許される最高優先度というPRIORITY_SELFが表現できること。

WindowStateType 型定義

WindowStateType は、以下のメンバを持つ構造体とする。

WindowStateType moveStatus

WindowStateType? moveStatus2

LockStatusType lockStatus

PositionType position;

PositionType targetPosition;

PositionType? position2

PositionType? targetPosition2

mainStatus2、position2とtargetPosition2を持つかは、

WindowStateType型変数configの

config.capabilitiesがHAS_MOVE2を含むかどうかで判定する。

getStatus API 定義

指定した instanceId を持つウィンドウの状態を返す。

ServiceCalls:

- name: getStatus
summary: ウィンドウ状態取得
parameters:
 - name: instanceId
type: IdType
- returnParameters:
 - name: returnValue
type: ReturnType
 - name: status
type: WindowStatusType

getErrorString API 定義

指定した instanceId に対して、故障の詳細を表す文字列を返す。
この文字列は、HMIがユーザに対して故障状態を通知するために使うことを想定している。
アプリがもつロケールに対応した文字列を返すため、以下の前提がある。
ビークルサービスは、サービスコールを呼び出したアプリケーションのロケール
(言語および地域設定)を取得できるものとする。

ServiceCalls:

- name: getErrorString
- summary: ウィンドウエラー情報の文字列取得
- parameters:
 - name: instanceId
 - type: IdType
- returnParameters:
 - name: returnValue
 - type: ReturnType
 - name: errorString
 - type: String

ウィンドウ制御操作API(準備)

2つのクラスを導入して制御操作APIを統一した。

1. Movable Object

- ▶ 開閉動作をするオブジェクトの抽象化
- ▶ 2次元までの動作をサポートする。
- ▶ 状態として、mainStatus/position/targetPosition(前述)を持つ。
- ▶ 操作APIとして、startMove/startMove2/stopMove を持つ。

2. Lockable Object

- ▶ 操作優先度によりロックする機構を提供する。
- ▶ 動的な情報として、lockStatus(前述)を持つ。
- ▶ 操作APIとして、lock/unlockを持つ。

startMove API 定義

指定した instanceId を持つウィンドウの目標開閉度を targetPosition に設定して開閉開始する。
このAPIは、開閉指示をした時点で完了する。
操作優先度 priority はオプションとする。
この priority を指定しない場合は、そのアプリに許される最高の操作優先度を指定したとする。

ServiceCalls:

- name: startMove
- summary: ウィンドウ開閉開始
- parameters:
 - name: instanceId
 - type: IdType
 - name: targetPosition
 - type: PositionType
 - name: priority
 - type: PriorityType?
- returnParameters:
 - name: returnValue
 - type: Returntype

startMove2 API 定義

指定した instanceId を持つウィンドウの目標開閉度2を targetPosition2 に設定して開閉開始する。
このAPIは、開閉指示をした時点で完了する。
操作優先度 priority はオプションとする。
この priority を指定しない場合は、そのアプリに許される最高の操作優先度を指定したとする。
startMoveと異なり、このAPIはオプションとなる。

ServiceCalls:

- name: startMove2
summary: ウィンドウ開閉開始 (2次元目の動作)
parameters:
 - name: instanceId
type: IdType
 - name: targetPosition2
type: PositionType
 - name: priority
type: PriorityType?
- returnParameters:
 - name: returnValue
type: Returntype

stopMove API 定義

指定した instanceId を持つウィンドウを停止する。

このAPIは、停止指示をした時点で完了する。

注意)ウィンドウおよびモータに慣性があるため、停止指示と停止することは異なる。

操作優先度 priority はオプションとする。

この priority を指定しない場合は、そのアプリに許される最高の操作優先度を指定したとする。

ServiceCalls:

- name: stopMove

 - summary: ウィンドウ停止

 - parameters:

 - name: instanceId

 - type: IdType

 - name: priority

 - type: PriorityType?

 - returnParameters:

 - name: returnValue

 - type: ReturnType

lock API 定義

指定した instanceId を持つウィンドウをロックする。
OSDVIの制御操作は、後勝ちであるため、その制御操作を後勝ちで打ち消されたくない場合
およびウィンドウをロックしたい場合に、このAPIを使用する。

ServiceCalls:

- name: lock
summary: ウィンドウをロックする
parameters:
 - name: instanceId
type: IdType
 - name: priority
type: PriorityType?

returnParameters:

- name: returnValue
type: ReturnType

unlock API 定義

指定した instanceId を持つウィンドウをロック解除する。
APIを使用するアプリがオブジェクトを lock によってロックしていない場合は、エラーとなる。

ServiceCalls:

- name: unlock
- summary: ウィンドウをロック解除する
- parameters:
 - name: instanceId
 - type: IdType
- returnParameters:
 - name: returnValue
 - type: ReturnType

ウィンドウで扱うイベント

ウィンドウの動作について、以下のイベントを持つ。

なお、自分自身のAPI発行によって状態が変わる場合はイベントと扱わない。

- 指定された `targetPosition` に到達して停止する。
- 他のアプリが制御API(`StartMove/StartMove2?/StopMove`)を呼び出す。
- 故障検出
- 故障復帰
- 挟み込み検出

ロック操作について、以下のイベントを持つ。

- 自分がロックしているインスタンスが他のアプリの操作によってロック解除された。
- 自分がロックしていないが、他のアプリがロック解除した。

補足)

`position(position2)`も状態として変化するが、その変化を継続して値を取得する場合(ウィンドウ開閉をアニメーションで見せる場合を想定)については、イベントを設定するのではなく、表示間隔に合わせて `getStatus` を呼び出すこととした。このため `position` 変化については、イベントと扱わない。

WindowEventKindType型定義

ウインドウイベントを表す型

以下の7つのBool変数を持つ構造体とする。

イベントのフィルタをして使う場合(notify)では、true でそのイベントを対象とする。

イベントの結果として使う場合(getEvent)では、true でそのイベントが発生したことを表す。

- TARGET_REACHED
指定された targetPosition に到達して停止する。
- CONTROLLED_BY_OTHER
他のアプリが制御API(StartMove/StartMove2?/StopMove)を呼び出す。
- FAULT_DETECTED
故障検出
- FAULT_RECOVERED
故障復帰
- PINCH_DETECTED
挟み込み検出
- OWN_LOCK_REVOKED
自分がロックしているインスタンスが他のアプリの操作によってロック解除された。
- OTHER_LOCK_RELEASED
自分がロックしていないが、他のアプリがロック解除した。

notify API 定義

イベント通知を開始する。

指定するフィルタは以下の2つ

- 対象とするウィンドウを `instanceId` で指定する。

このAPIでは、`instanceId`として `ID_INVALID` を指定するとすべてのウィンドウが対象となる。

- 対象とするイベントを `WindowEventKindType` 型 `eventType` で指定する。

このAPIを使用することによりイベントハンドラーを生成する。

このイベントキューを表す `NotifyHandleType` 型 `notifyHandle` を返す。

ServiceCalls:

- name: `notify`
summary: イベント通知を開始する
parameters:
 - name: `instanceId`
type: `IdType?`
 - name: `eventType`
type: `enumSet(WindowEventKindType)`

returnParameters:

- name: `returnValue`
type: `ReturnType`
- name: `notifyHandle`
type: `NotifyHandleType`

unnotify API 定義

イベント通知を停止する

引数として、notify で生成した notifyHandle を指定する。

このAPIを使用することによりイベント通知を停止する。対象のイベントキューを削除する。

ServiceCalls:

- name: unnotify

 - summary: イベント通知を停止する

 - parameters:

 - name: notifyHandle

 - type: NotifyHandleType

returnParameters:

- name: returnValue

 - type: ReturnType

WindowEventType 型定義(1)

ウィンドウに関するイベントが発生時にイベントキューに保存する情報を表す型
以下のメンバを持つ構造体とする。
補足) イベントキューに、OS依存でその他の情報を追加してよい。

instanceId

- IdType型の変数、対象オブジェクトを格納する。(マルチインスタンスオブジェクトの場合のみ)

eventInfo

- union(WindowEventKindType) 型の変数、イベントの原因を格納する。

sourceApplication

- ApplicationIdType型の変数、イベントの原因となったアプリを格納する。
- 参考) この型は、lockStatus.lockApplication にも使われている。

sourceService

- WindowServiceType 型の変数、イベントの原因となったAPIを格納する。

WindowEventType 型定義(2)

```
type WindowEventKindType = enum {
    TARGET_REACHED,
    CONTROLLED_BY_OTHER,
    FAULT_DETECTED,
    FAULT_RECOVERED,
    PINCH_DETECTED,
    OWN_LOCK_REVOKED,
    OTHER_LOCK_RELEASED
};

type WindowServiceType = enum {
    START_MOVE,
    START_MOVE2,
    STOP_MOVE
};

type WindowEventType = struct {
    instanceId : IdType;
    eventInfo : union(WindowEventKindType) {
        CONTROLLED_BY_OTHER : struct {
            sourceApplication : ApplicationIdType,
            sourceService : WindowServiceType
        }
    }
};
```

WindowServiceType 型定義

OSDVIのウィンドウAPIを表す。

列挙型とする。

202509αでは、イベントに必要な以下の列挙定数について定義する。

- `START_MOVE`
startMoveを表す。
- `START_MOVE2`
startMove2を表す。
- `STOP_MOVE`
stopMoveを表す。

getEvent API 定義

イベント詳細を取得する。

イベント詳細情報が必要な場合、本APIを実行することで、イベント詳細情報を取得できる。イベントがパラメータ等で渡されてくるプログラミング環境向けの物理APIでは、本APIは不要なためオプションとなる。

ServiceCalls:

- name: getEvent
summary: イベント詳細情報を取得する
parameters:
 - name: notifyHandle
type: NotifyHandleType

returnParameters:

- name: returnValue
type: ReturnType
- name: WindowEventType
type: windowEvent

主な未定義事項

1. システムイベントについて

- 車両全体の構成が変わるため、影響がウィンドウを超えるため、TBDとした。

2. 物理定義(実装)について

- 当初は、POSIXで動作するC実装を与えることを検討したが、API検討に注力することにした。
- 物理実装は、他WGと連携して検討を継続していく。

EOF