

UIEPlayer 移植用 STB API 仕様 4

-視聴・録画予約データ編-

Revision 2.2.8

11/16/2009

UIEvolution, Inc.

1. はじめに	4
2. 用語について	5
3. ひかり TV 版 UIEPlayer と受信機レジデントの構成	6
4. 機能分担の想定	8
5 呼び出しシーケンスサンプル	9
5.1. 予約一覧	9
5.2. 予約登録	9
5.3. 予約視聴開始	10
6. API 概要	11
6.1. 視聴・録画予約データ関連 API 概要	11
視聴・録画予約データ API 共通エラーコード一覧	11
視聴・録画予約データ関連 API 関数一覧	11
視聴・録画予約データ関連 API イベント一覧	17
7. 視聴・録画予約データ関連 API リファレンス	18
7.1. 視聴・録画予約データ関連 API 関数	18
int uie_Reserve_init(int *subErrNo)	18
int uie_Reserve_exit(int *subErrNo)	19
int uie_Reserve_getCount(int *subErrNo)	20
int uie_Reserve_getProperty(int property , int *subErrNo)	21
int uie_Reserve_getReserveld(int index , char reserveld[], int *subErrNo)	23
int uie_Reserve_get(const char reserve_id[], int property , int *subErrNo)	24
int uie_Reserve_getString(const char reserve_id[], int property, char resultStr[], int size , int *subErrNo)	28
int uie_Reserve_add(int type, int network_id, int service_id, int start_date, int duration, int cycle, int reserve_type, int trigger_type, int rating, const char* event_title, int event_id, int tracking, char* tracking_keyword, int lastmodified_time, void* reserve , int *subErrNo)	29
int uie_Reserve_edit(char reserve_id[], int type, int network_id, int service_id, int start_date, int duration, int cycle, int reserve_type, int trigger_type, int rating, const char* event_title, int event_id, int tracking, char* tracking_keyword, int lastmodified_time, void* reserve , int *subErrNo)	32
int uie_Reserve_delete(const char reserve_id[] , int delete_flag, int *subErrNo)	35
int uie_Reserve_cancel(const char reserve_id[] , int *subErrNo)	36
int uie_Reserve_findDuplicate(char reserve_id[], int type, int network_id, int service_id, int start_date, int duration, int cycle, int reserve_type, char* dup_id[], int dup_state[], int size, void* reserve , int *subErrNo)	37
int uie_Reserve_getCurrent(int index , char reserve_id[], int *subErrNo)	39
int uie_Reserve_getNext(char reserve_id[], int *subErrNo)	40
int uie_Reserve_findByServiceId(int type, int network_id, int service_id, int index, void* reserve)	41
int uie_Reserve_getReserveldByEvent(int type, int network_id, int service_id, int event_id, char	

reserve_id[], void* reserve , int *subErrNo).....	42
int uie_Reserve_findByTime(int type, int network_id, int service_id, int start_date, int duration, void* reserve , int *subErrNo).....	44
int uie_Reserve_getFindResult(int index, int property , int *subErrNo)	46
int uie_Reserve_getStringFindResult(int index, int property, char resultStr[], int size , int *subErrNo)	50
int uie_Reserve_initialize(int *subErrNo)	51
int uie_Reserve_getLastModifiedTime(int *subErrNo).....	52
7.2. 視聴・録画予約データ関連 API イベント	53
8. 変更履歴	56

1. はじめに

本資料では、UIEPlayer 版ひかり TV 対応 STB を開発するに当たり、受信機のファームウェアでどのような API セットを用意する必要があるかについて概要を述べたものです (UIEPlayer を除いた受信機の構成を、本資料では受信機レジデントと呼びます)。これらの API セットはライブラリとして提供され、UIEPlayer からリンクされることを想定しています。

本資料では、IPTV で必要となる機能を実現するための、サーバーとのインタフェースや動画の再生等、UIEPlayer(UJML)の機能を拡張するために必要となる拡張 STB API について述べます。その他の関連資料として、以下が用意されています。

関連資料:

- UIEPlayer 移植用標準 STB API 資料1 (概要および標準 STB API 編)
- UIEPlayer 移植用拡張 STB API 資料2 (システム・メディアプレイヤー編)
- UIEPlayer 移植用拡張 STB API 資料3 (番組表データ編)
- UIEPlayer 移植用拡張 STB API 資料4 (視聴・録画予約データ編・本資料)
- UIEPlayer 移植用拡張 STB API 資料5 (録画データ編)
- ひかり TV UIEPlayer UJML プログラミングノート
- UJML Language Reference 2.1 (<http://developer.uievolution.com/docs/en/LangRef/index.html>)

また、本資料は以下のひかり TV 関連資料の知識を前提とします。

- ・NTT ぷらら 録画機能仕様書 共通
- ・NTT ぷらら テレビ録画予約要求定義書
- ・NTT ぷらら 1. 予約基本シーケンス

2. 用語について

用語は「IPTV 規定」にて使用されているものを原則として使用します。

UIEvolution および本資料独自の用語を以下に説明します：

- UIEPlayer：UJBC ファイルを解釈・実行する GUI ミドルウェア。UIEngine と呼ばれるが本資料では UIEPlayer に統一
- UJML, UJBC：UJML は XML 形式で記述される UIEPlayer 用言語。UJML は中間コード形式の UJBC にコンパイルされ、UIEPlayer 上で実行される。詳細は <http://www.uievolution.com/support/> 参照
- 受信機レジデント(受信機システム)：UIEPlayer 担当部分を除く受信機の機能(ファームウェア/ソフトウェア)を表す
- GUI：UIEPlayer 上で実行されるひかり TV アプリケーション。複数の UJBC ファイル、イメージファイルから構成される。これらのファイルは受信機レジデント内のストレージまたはネットワーク経由で読み込まれる。
- レジデント UJML：受信機のファイルシステムに保存される UJML(UJBC)コード
- メディアプレイヤー：受信機レジデント中のコンテンツ処理機能(主に、映像・音声複合処理及び出力)を表す
- STB API：本資料で述べる、UIEPlayer – 受信機レジデント間の API を表す
- Native component：UJML 言語のコンポーネントの実装を、ネイティブコードで記述したもの。UJML からネイティブコード(拡張 STB API)を呼び出すために必要となる。

予約に関する用語を以下に説明します：

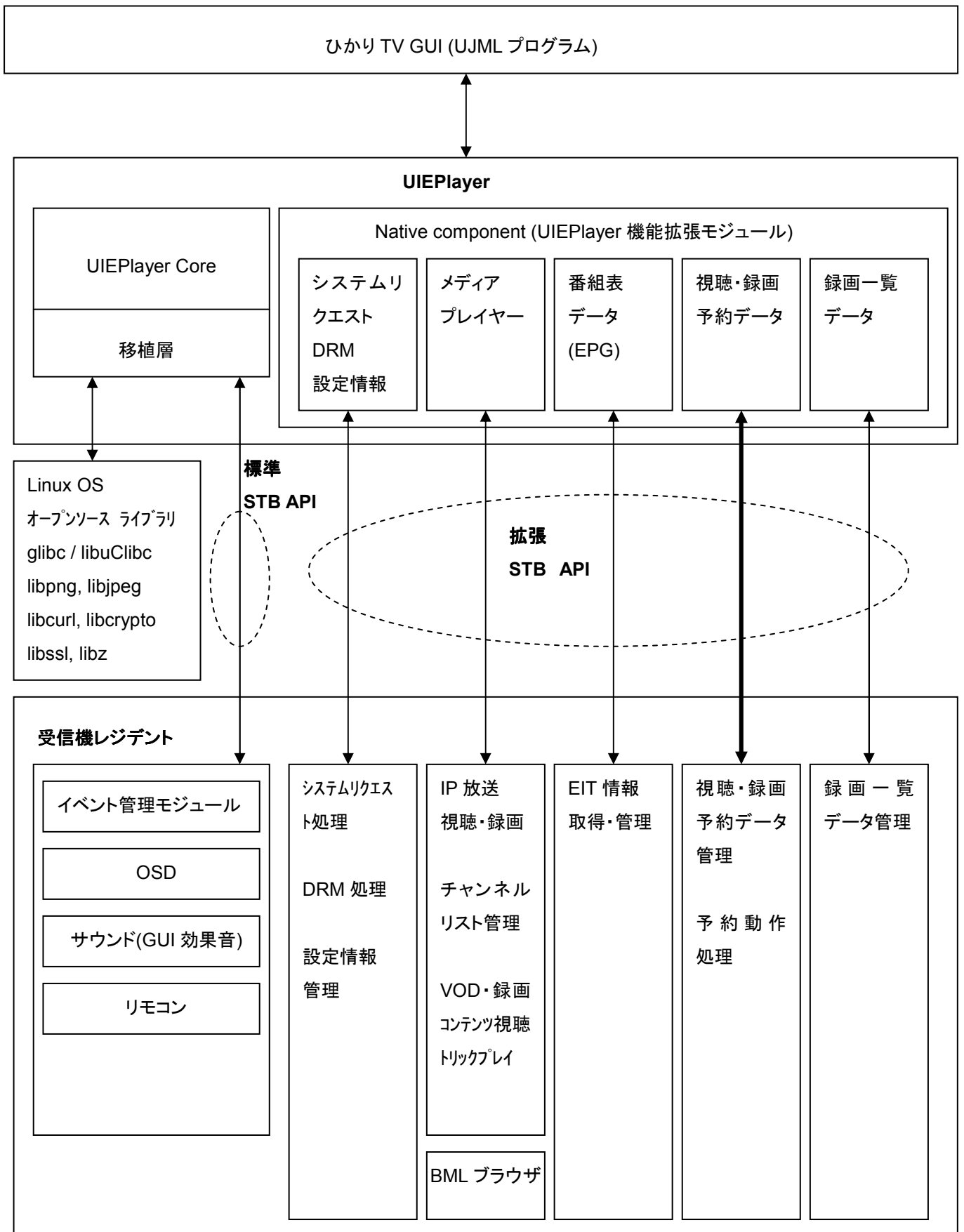
- 時間指定予約：開始日時・時間を指定された予約
- 番組指定予約：番組イベント ID ベースで指定された予約。追従などの対象となる。
- 予約発火データ：予約がいつ実行されるかの視点で見た予約データ。繰り返し予約は同一予約 ID に複数の予約発火データが存在する。
- 予約登録データ：予約一覧に出てくる予約データに相当する。予約 ID と一対一対応する。

3. ひかり TV 版 UIEPlayer と受信機レジデントの構成

受信機レジデントと UIEPlayer のブロック図を以下に示します。UIEPlayer はライブラリとして受信機レジデントに組み込まれ、専用のスレッドまたはプロセスとして受信機レジデントから起動されます。受信機全体の動作や設定情報の管理などは、受信機レジデント主体で行われます。

以下に、UIEPlayer の視点から見た全体構成図を示します。本資料では、太線で記載された拡張 API の仕様について記述されています。

UIEPlayer から受信機レジデントへのリクエストは、STB API 関数コールで行われます。一方、受信機レジデントから UIEPlayer へのリクエストやステータス通知は STB API イベントベースで行われ、UIEPlayer 側で非同期に処理されます。



4. 機能分担の想定

視聴・録画予約における受信機レジデント側と UIEPlayer 側の機能分担は、現状以下を想定しています。

受信機レジデント側:

- 視聴・録画予約 API の実装(本ドキュメント)
- 録画予約の実行開始処理 (選局含む)
録画用の選局を行ったために、視聴中の放送を強制切替するケースも含まれる
- 視聴予約の実行開始処理(選局及び映像表示を含む)
- 電源 OFF 状態における、視聴予約データの参照と視聴・録画開始処理 (UIEPlayer 起動も含む)

UIEPlayer(UJML)側:

- 予約登録 GUI・予約一覧 GUI の実行。
- 予約視聴・予約録画開始イベントに基づき、画面の切替や、視聴・録画を開始する旨の通知を GUI で行う。

5 呼び出しシーケンスサンプル

5.1. 予約一覧

```
for(index = 0; ; i++){
    const char* reserve_id;
    uie_Reserve_getReserveld(index, reserve_id);
    if(NULL == reserve_id){
        break;    // 予約データ終了
    }

    // uie_Reserve_get(reserve_id,,,), uie_Reserve_getString(reserve_id,,, )をコールして、予約内容表示
}
```

5.2. 予約登録

```
const char* reserve_id = uie_Reserve_getCount();
if (NULL == count){
    // Error return
}
if (count >= uie_Reserve_getProperty(UIE_RESERVE_GLOBAL_MAX)){
    // 予約最大数超過
} else {
    if (0 < uie_Reserve_findDuplicate()){
        // 重複の警告
    }
    uie_Reserve_add();
}
```

5.3. 予約視聴開始

```

stbEventHandler(api_event_type, api_event_param)    // 受信機レジデント側からのイベントハンドラ
{
    if(RESERVE_EVENT_START == api_event_type) {    // 視聴開始
        char* reserve_id = api_event_param;

        視聴開始確認ダイアログの表示();
        if (視聴開始で「いいえ」を選択){
            uie_Reserve_cancel(reserve_id);
        } else {
            int type = uie_Reserve_get(reserve_id, UIE_RESERVE_PROPERTY_TYPE);
            int network_id = uie_Reserve_get(reserve_id, UIE_RESERVE_PROPERTY_NETWORK_ID);
            int service_id = uie_Reserve_get(reserve_id, UIE_RESERVE_PROPERTY_SERVICE_ID);

            if(UIE_MEDIA_ERR_OK == uie_Media_checkService(type, network_id, service_id)){
                // すべて視聴可能な設定にパレンタルレベルをセット。シークレットモード ON
                uie_System_setProperty("secret_parentallevel", "17");
                uie_System_setProperty("secret_mode", "1");

                uie_Media_openService(type, network_id, service_id);
            }
        }
    } else if(RESERVE_EVENT_FINISH == api_event_type) {    // 視聴終了
        uie_Media_stop();

        シークレットモード、パレンタルレベルを復元();
        予約視聴開始前のページへ遷移();
    }
}

```

6. API 概要

6.1. 視聴・録画予約データ関連 API 概要

視聴・録画予約データ API 共通エラーコード一覧

UIE_RESERVE_ERR_OK	0	正常終了
UIE_RESERVE_ERR_NG	-1	その他のエラー
UIE_RESERVE_ERR_ENDOFDATA	-2	データの終了を表す
UIE_RESERVE_ERR_UNAVAILABLE	-3	操作無効
UIE_RESERVE_ERR_INVALID_PARAMETER	-4	引数が不正
UIE_RESERVE_ERR_NOTREADY	-5	予約モジュール準備中
UIE_RESERVE_ERR_FULL	-6	容量不足のため録画予約不可

受信機レジデント側固有の詳細エラーコード:

各 API 関数には、受信機レジデント側の詳細エラーコード格納用のポインタ引数を伴います。UIEPlayer は初期値として 0 を設定し、受信機レジデントが 0 以外の値を設定すると、それを詳細エラーコードとして認識します。設定される値は本 API では規定せず、受信機依存とします。返される詳細エラーコードは、デバッグログへの記録や、GUI でのエラーダイアログでの詳細エラーコードとしての表示を想定しています。

視聴・録画予約データ関連 API 関数一覧

API Name	Synopsys	Parameters	Note
uie_Reserve_init()	予約データAPIの初期化处理 (データの初期化ではない)。 UIEPlayer 起動時に呼び出される。	なし	
uie_Reserve_exit()	予約データAPIの終了処理。 UIEPlayer 終了時に呼び出される。	なし	
uie_Reserve_getCount()	予約データの現在の登録件 数を取得する。	なし	
uie_Reserve_getProperty()	予約データAPIの各種属性値 を取得する。	property	
uie_Reserve_getReserveld()	予約データIDを順に取得する	インデックス値	

API Name	Synopsys	Parameters	Note
uie_Reserve_get (UIE_RESERVE_PROPERTY_TYPE)	放送種別取得。	reserve_id	
uie_Reserve_get (UIE_RESERVE_PROPERTY_SERVICE_ID)	サービス ID 取得。	reserve_id	
uie_Reserve_get (UIE_RESERVE_PROPERTY_START_DATE)	予約開始日時取得。	reserve_id	
uie_Reserve_get (UIE_RESERVE_PROPERTY_DURATION)	予約時間長取得。	reserve_id	
uie_Reserve_get (UIE_RESERVE_PROPERTY_CYCLE)	予約周期取得。	reserve_id	
uie_Reserve_get (UIE_RESERVE_PROPERTY_RATING)	年齢制限情報値取得。	reserve_id	廃止 Revision2.2.5 で 復活
uie_Reserve_get (UIE_RESERVE_PROPERTY_STATE)	予約の重複状態取得。	reserve_id	
uie_Reserve_get (UIE_RESERVE_PROPERTY_EVENT_ID)	イベント ID 取得。	reserve_id	
uie_Reserve_get (UIE_RESERVE_PROPERTY_NETWORK_ID)	ネットワーク ID 取得。	reserve_id	
uie_Reserve_get (UIE_RESERVE_PROPERTY_RESERVE_TYPE)	予約種別取得。	reserve_id	
uie_Reserve_get (UIE_RESERVE_PROPERTY_TRACKING)	放送時間変更の追従設定取得。	reserve_id	
uie_Reserve_get	削除フラグの状態	reserve_id	

API Name	Synopsys	Parameters	Note
(UIE_RESERVE_PROPERTY_DELETE_FLAG)			
uie_Reserve_getString (UIE_RESERVE_PROPERTY_EVENT_TITLE)	イベント(番組)タイトル取得。	reserve_id size	
uie_Reserve_getString (UIE_RESERVE_PROPERTY_TRACKING_KEYWORD)	連ドラ追従キーワード取得。	reserve_id size	
uie_Reserve_add()	予約情報の追加を行う。	type, network_id service_id start_date duration cycle reserve_type trigger_type rating event_title event_id tracking tracking_keyword lastmodified_time reserve	

API Name	Synopsys	Parameters	Note
uie_Reserve_edit()	予約情報の編集を行う。	reserve_id, type, network_id service_id start_date duration cycle reserve_type trigger_type rating event_title event_id tracking tracking_keyword lastmodified_time, reserve	
uie_Reserve_delete()	予約情報の削除を行う。	reserve_id delete_flag	
uie_Reserve_cancel()	特定の予約情報を取り消す。	reserve_id	録画を停止する際にもこの関数を利用。
uie_Reserve_findDuplicate()	新規登録時 (uie_Reserve_add())、および 予約情報更新時 (uie_Reserve_edit())の重複 チェックに使用する。	reserve_id, type, network_id service_id start_date duration cycle reserve_type size reserve	
uie_Reserve_getCurrent()	現在実行中の予約が存在すればその予約情報の ID を取得する。	index	
uie_Reserve_getNext()	次に実行される視聴予約情報の予約IDを取得する。	なし	視聴予約が対象。 録画予約は対象外。

API Name	Synopsys	Parameters	Note
			未使用につき廃止
uie_Reserve_findByServiceId()	ネットワーク ID, サービス ID に関連する予約 ID を取得する。	network_id service_id index reserve	未使用につき廃止
uie_Reserve_getReserveldByEvent()	特定のイベント ID (番組) をも つ予約 ID を取得する。	type network_id service_id event_id reserve	番組指定予約が 対象になる。
uie_Reserve_findByTime()	指定されたサービス ID, 時間 帯に該当する予約データを検 索する。 番組表で予約時間帯の表示 を行うために使用する。	type network_id service_id start_date duration reserve	予約データの内容 取得は、uie_Res erve_getFindRes ult(), uie_Reserv e_getStringFindR esult()で行う。
uie_Reserve_getFindResult()	予約データの数値情報の取得	uie_Reserve_findBy Time()の index 値	uie_Reserve_find ByTime() の 結 果 取得用。取得対象 は uie_Reserve_get ()と同等。
uie_Reserve_getStringFindResult()	予約データの文字列情報の 取得	uie_Reserve_findBy Time()の index 値	uie_Reserve_find ByTime() の 結 果 取得用。取得対象 は uie_Reserve_get String()をベースに 予約 ID の取得を 追加。
uie_Reserve_initialize()	予約に関連するデータの初期 化を行う。	なし	廃止 uie_System_requ est (SYSTEM_REQ UEST_ NITIALIZE_PRO

API Name	Synopsys	Parameters	Note
			PERTY)で予約データも初期化する
uie_Reserve_getLastModifiedTime()	端末に登録されている予約データが最後に変更された日時を取得する。	なし	

視聴・録画予約データ関連 API イベント一覧

Event Name	Synopsys	Parameters	Note
RESERVE_EVENT_START	予約視聴・予約録画の開始を通知。	予約種別・ 予約開始イベント詳細種別 裏録画フラグ・ 開始される視聴予約 ID	予約視聴、予約録画の開始を通知するイベント。実際に予約視聴、録画が開始されるまで、複数の詳細種別イベントが順番に発生する。
RESERVE_EVENT_FINISH	予約視聴・予約録画の終了を通知。	予約種別 終了する視聴予約 ID	終了時間ちょうどに通知される。 なお、続けて別の視聴予約が開始される場合は、本イベントの通知は行わない。
RESERVE_EVENT_DATACHANGED	視聴・録画予約データの変更を通知。	予約 ID	具体的には終了した予約が削除された場合や、放送時間変更への追従等を想定。

7. 視聴・録画予約データ関連 API リファレンス

7.1. 視聴・録画予約データ関連 API 関数

```
int uie_Reserve_init(int *subErrNo)
```

予約データ API の初期化処理(データの初期化ではない)。UIEPlayer 起動時に呼び出される。

Parameters:

[OUTPUT] int *subErrNo

受信機レジデント側が返す詳細エラーコード。初期値として 0 をセットして呼び出される。

Returns:

成功時: UIE_RESERVE_ERR_OK (0)

失敗時: UIE_RESERVE_ERR_NG (-1)

```
int uie_Reserve_exit(int *subErrNo)
```

予約データ API の終了処理。UIEPlayer 終了時に呼び出される。

Parameters:

[OUTPUT] int *subErrNo

受信機レジデント側が返す詳細エラーコード。初期値として 0 をセットして呼び出される。

Returns:

成功時: UIE_RESERVE_ERR_OK (0)

失敗時: UIE_RESERVE_ERR_NG (-1)

int uie_Reserve_getCount(int *subErrNo)

予約データの現在の登録件数を取得する。

Parameters:

[OUTPUT] int *subErrNo

受信機レジデント側が返す詳細エラーコード。初期値として 0 をセットして呼び出される。

Returns:

成功時: 正の値が返る。(0 は有効)

失敗時:

UIE_RESERVE_ERR_NG	-1	
UIE_RESERVE_ERR_NOTREADY	-5	予約モジュール準備中

```
int uie_Reserve_getProperty(int property , int *subErrNo)
```

予約データ API の各種属性値を取得する。

Parameters:

(property) - 取得したい属性値の種類。指定可能なものは以下:

数	UIE_RESERVE_GLOBAL_PROPERTY_MAX	(0)	登録可能最大件数
	UIE_RESERVE_GLOBAL_PROPERTY_DIRECTREC_MAX	(1)	ダイレクト録画予約最大件数
	UIE_RESERVE_GLOBAL_PROPERTY_RECORDABLE	(2)	録画予約可能
	UIE_RESERVE_GLOBAL_PROPERTY_TRACKING_KEYWORD	(3)	連ドラ追従対応
	UIE_RESERVE_GLOBAL_PROPERTY_DELETE_FLAG	(4)	削除フラグ対応
	UIE_RESERVE_GLOBAL_PROPERTY_REMOTE_RESERVE	(5)	遠隔予約対応
	UIE_RESERVE_GLOBAL_PROPERTY_ID_LENGTH	(6)	予約 ID の文字列最大長

[OUTPUT] int *subErrNo

受信機レジデント側が返す詳細エラーコード。初期値として 0 をセットして呼び出される。

Returns:

成功時: int 型で属性値を返す。

property = UIE_RESERVE_GLOBAL_PROPERTY_MAX

登録可能最大件数

property = UIE_RESERVE_GLOBAL_PROPERTY_DIRECTREC_MAX

ダイレクト録画予約最大件数 ※同時録画可能なストリーム数と同じとなる

property = UIE_RESERVE_GLOBAL_PROPERTY_RECORDABLE

0 で録画予約不可能。1 で可能

property = UIE_RESERVE_GLOBAL_PROPERTY_TRACKING_KEYWORD

0 で連ドラ追従不可能。1 で可能

property = UIE_RESERVE_GLOBAL_PROPERTY_DELETE_FLAG

0 で削除フラグ付与不可能。1 で可能

property = UIE_RESERVE_GLOBAL_REMOTE_RESERVE

受信機レジデントが遠隔予約への対応機能をもつかどうかを表す。0 で非対応、1 で対応

property = UIE_RESERVE_GLOBAL_ID_LENGTH

受信機レジデントが返す予約 ID の文字列最大長文字列の最大長を表す。文字列終端の'\0'を除いた長さ。

失敗時: 負の値

UIE_RESERVE_ERR_INVALID_PARAMETER 引数が不正

UIE_RESERVE_ERR_NG その他のエラー

UIE_RESERVE_ERR_NOTREADY 予約モジュール準備中

```
int uie_Reserve_getReserveld(int index , char reserveld[], int *subErrNo)
```

受信機レジデント側に登録されている予約データの予約 ID を順に取得する。

予約 ID は予約発火時間順でかえられることとします。

Parameters:

(index) – 取得したい予約のインデックス値。

インデックスの初期値は 0 を指定し、予約開始時刻がもっとも若い予約情報の ID を取得する。取得後は、次の予約 ID を取得するためにインデックスに 1 を加えたものを設定する。

最後の予約に達した後にこの関数を実行すると UIE_RESERVE_ERR_ENDOFDATA エラーが返り、これにより取得終了を判断します。

[OUTPUT] char reserveld[]

受信機レジデント側が返す予約 ID 文字列の格納領域。UIEPlayer があらかじめ

uie_Reserve_getProperty(UIE_RESERVE_GLOBAL_PROPERTY_ID_LENGTH) + 1 の領域を確保済み。

[OUTPUT] int *subErrNo

受信機レジデント側が返す詳細エラーコード。初期値として 0 をセットして呼び出される。

Returns:

成功時: UIE_RESERVE_OK

reserve_id[]に予約 ID 文字列がセットされる。

失敗時: 負の値

UIE_RESERVE_ERR_ENDOFDATA

データが存在しない

UIE_RESERVE_ERR_NG

その他のエラー

UIE_RESERVE_ERR_NOTREADY

予約モジュール準備中

Note:

本関数は、予約の一覧 GUI で使用されることを想定しています。

```
int uie_Reserve_get(const char reserve_id[], int property , int *subErrNo)
```

予約 ID で指定された予約情報より、property で指定された int 型の属性値を取得する。

繰り返し予約の場合、直近に実行される予約の情報を返す(実行中の場合も含む)。

Parameters:

(reserve_id) - 予約 ID 文字列

(property) - 取得したい属性値の種類。指定可能なものは以下:

UIE_RESERVE_PROPERTY_SERVICE_ID	(0)	サービス ID
UIE_RESERVE_PROPERTY_START_DATE	(1)	予約開始日時(重複無視)
UIE_RESERVE_PROPERTY_DURATION	(2)	予約時間長(重複無視)
UIE_RESERVE_PROPERTY_ACTUAL_START_DATE	(15)	予約開始日時(重複考慮済)
UIE_RESERVE_PROPERTY_ACTUAL_DURATION	(16)	予約時間長(重複考慮済)
UIE_RESERVE_PROPERTY_CYCLE	(3)	予約周期
UIE_RESERVE_PROPERTY_RATING	(4)	年齢制限情報値
UIE_RESERVE_PROPERTY_STATE	(5)	予約の重複状態
UIE_RESERVE_PROPERTY_EVENT_ID	(7)	イベント ID
UIE_RESERVE_PROPERTY_NETWORK_ID	(9)	ネットワーク ID
UIE_RESERVE_PROPERTY_RESERVE_TYPE	(10)	予約種別
UIE_RESERVE_PROPERTY_TRIGGER_TYPE	(17)	発火種別
(時間指定予約・番組指定予約)		
UIE_RESERVE_PROPERTY_TRACKING	(11)	放送時間変更の追従設定
UIE_RESERVE_PROPERTY_TYPE	(13)	放送種別
UIE_RESERVE_PROPERTY_DELETE_FLAG	(14)	遠隔予約用削除フラグ
UIE_RESERVE_PROPERTY_LASTMODIFIED_TIME	(18)	最終更新日時

[OUTPUT] int *subErrNo

受信機レジデント側が返す詳細エラーコード。初期値として 0 をセットして呼び出される。

Returns:

成功時: int 型で属性値を返す。

property = UIE_RESERVE_PROPERTY_SERVICE_ID
サービス ID(チャンネル)

property = UIE_RESERVE_PROPERTY_START_DATE

予約重複を考慮しない場合の予約開始日時。繰り返し予約の場合は、直近の時刻となる。

POSIX time 形式 (1970 年 1 月 1 日 00:00:00 UTC からの経過秒数。非負値。2038 年以降の値

は想定しない)

property = UIE_RESERVE_PROPERTY_DURATION

予約重複を考慮しない場合の予約時間長(秒)

property = UIE_RESERVE_PROPERTY_CYCLE

UIE_RESERVE_CYCLE_ONCE (0) 予約周期: 1 回

UIE_RESERVE_CYCLE Everyday (1) 予約周期: 毎日

UIE_RESERVE_CYCLE WEEKLY (2) 予約周期: 予約開始日を基準に 1 週間毎

UIE_RESERVE_CYCLE MON2SAT (3) 予約周期: 月曜から土曜まで

UIE_RESERVE_CYCLE MON2FRI (4) 予約周期: 月曜から金曜まで

property = UIE_RESERVE_PROPERTY_RATING

ARIB STD-B10 のコンテンツ記述子の内容に準じる

値 + 3 歳が推奨最低年齢となる。// TBD: 削除を検討中 (EPG API から取得)

property = UIE_RESERVE_PROPERTY_STATE

UIE_RESERVE_STATE_NO_DUP (0) 重複なし

UIE_RESERVE_STATE_DUP_OK (1) 重複するが全て視聴・録画可能

UIE_RESERVE_STATE_DUP_NG (2) 重複し、一部が視聴・録画できない

~~UIE_RESERVE_STATE_CANCELED (3) 実行中止状態 (uie_reserve_cancel() が
行われたイベント)~~

UIE_RESERVE_STATE_DISABLED (4) 重複し、全部が視聴・録画できない

property = UIE_RESERVE_PROPERTY_EVENT_ID

特定の予約に対するイベント ID (番組)

この値は EPG API に event_id として渡される。

時間指定予約で、event_id が無効な場合 -1 を返す

property = UIE_RESERVE_PROPERTY_NETWORK_ID

特定の予約に対するネットワーク ID。

property = UIE_RESERVE_PROPERTY_RESERVE_TYPE

UIE_RESERVE_RESERVE_TYPE_PLAY (0) 視聴

UIE_RESERVE_RESERVE_TYPE_RECORD (1) 録画

UIE_RESERVE_RESERVE_TYPE_DIRECT_REC (2) ダイレクト録画

property = UIE_RESERVE_PROPERTY_TRIGGER_TYPE

UIE_RESERVE_TRIGGER_TYPE_TIME	(0)	時間指定予約
UIE_RESERVE_TRIGGER_TYPE_EVENT	(1)	番組指定予約

property = UIE_RESERVE_PROPERTY_TRACKING

UIE_RESERVE_TRACKING_OFF	(0)	追従しない
UIE_RESERVE_TRACKING_ON	(1)	追従する
UIE_RESERVE_TRACKING_KEYWORD	(2)	連ドラ追従する。オプション

property = UIE_RESERVE_PROPERTY_ACTUAL_START_DATE

予約重複を考慮した場合の予約開始日時。繰り返し予約の場合は、直近の時刻となる。

重複のため予約が実行されない場合は 0 を返す。

POSIX time 形式 (1970 年 1 月 1 日 00:00:00 UTC からの経過秒数。非負値。2038 年以降の値は想定しない))

property = UIE_RESERVE_PROPERTY_ACTUAL_DURATION

予約重複を考慮した場合の予約時間長(秒)

重複のため予約が実行されない場合は 0 を返す。

property = UIE_RESERVE_PROPERTY_TYPE

UIE_RESERVE_TYPE_IPTV	(0)	多チャンネル
UIE_RESERVE_TYPE_TB	(1)	地上デジタル IP 再送信
UIE_RESERVE_TYPE_BS	(2)	BS 放送 IP 再送信

property = UIE_RESERVE_PROPERTY_DELETE_FLAG

UIE_RESERVE_DELETE_FLAG_ON	(1)	削除フラグが付与されている
UIE_RESERVE_DELETE_FLAG_OFF	(0)	削除フラグが付与されていない

削除フラグが付与されている予約情報は端末内部には存在するが予約情報は実行されない。遠隔予約機能に対応する場合に考慮する必要がある。

property = UIE_RESERVE_PROPERTY_LASTMODIFIED_TIME 最終更新日時

POSIX time 形式 (1970 年 1 月 1 日 00:00:00 UTC からの経過秒数。非負値。2038 年以降の値は想定しない)

失敗時: 負の値

UIE_RESERVE_ERR_INVALID_PARAMETER	引数が不正
UIE_RESERVE_ERR_NG	その他のエラー
UIE_RESERVE_ERR_NOTREADY	予約モジュール準備中

Note: 番組指定予約の場合でも、予約開始時間や予約時間などの時間情報は有効です。

```
int uie_Reserve_getString(const char reserve_id[], int property, char resultStr[], int size , int *subErrNo)
```

予約 ID で指定された予約情報より、property で指定された char 型の属性値を取得する。

繰り返し予約の場合、直近に実行される予約の情報を返す(実行中の場合も含む)。

呼び出し元(UIE Player)が文字列領域を用意する。

Parameters:

(reserve_id) - 予約 ID 文字列

(property) - 取得したい属性値の種類。指定可能なものは以下:

UIE_RESERVE_PROPERTY_EVENT_TITLE (6) //TBD: EPG API から取得を検討中

UIE_RESERVE_PROPERTY_TRACKING_KEYWORD (12) 連ドラ追従キーワード

uie_Reserve_get(UIE_RESERVE_PROPERTY_TRACKING)が

UIE_RESERVE_TRACKING_KEYWORD の場合有効

(resultStr[]) [OUTPUT] - 指定したプロパティの内容。UTF8 フォーマット。

情報が存在しない場合には、空文字列を返す。

文字列の受け渡しに必要な領域は、あらかじめ UIEPlayer 側で確保される。

(size) - 前述の resultStr[] のサイズを表す(文字終端分を含む)

[OUTPUT] int *subErrNo

受信機レジデント側が返す詳細エラーコード。初期値として 0 をセットして呼び出される。

Returns:

成功時:

UIE_RESERVE_ERR_OK。resultStr[] がセットされる

失敗時: 負の値

UIE_RESERVE_ERR_INVALID_PARAMETER 引数が不正

UIE_RESERVE_ERR_NG その他のエラー

UIE_RESERVE_ERR_NOTREADY 予約モジュール準備中

```
int uie_Reserve_add(int type, int network_id, int service_id, int start_date, int duration, int cycle, int reserve_type,
int trigger_type, int rating, const char* event_title, int event_id, int tracking, char* tracking_keyword, int
lastmodified_time, void* reserve , int *subErrNo)
```

予約情報の追加を行う。

Parameters:

(type)	- 予約情報の放送種別
(network_id)	- 予約情報のネットワーク ID
(service_id)	- 予約情報のサービス ID
(start_date)	- 予約開始日時 POSIX time 形式 (1970 年 1 月 1 日 00:00:00 UTC からの経過秒数。非負値。2038 年以降の値は想定しない))
(duration)	- 予約時間長 (秒)
(cycle)	- 予約周期
(reserve_type)	- 予約種別
(trigger_type)	- 発火条件(番組指定予約・時間指定予約)
(rating)	- Rating of the reservation ARIB STD-B10 のコンテンツ記述子の内容に準じる 値 +3 歳が推奨最低年齢となる。 // TBD: 削除を検討中 (EPG API から取得)
(event_title)	- 予約情報のイベント(番組) タイトル文字列 。UTF8 format. // TBD: 削除を検討中 (EPG API から取得)
(event_id)	- 予約情報のイベント(番組)ID <u>番組指定予約の際は必須。時間指定予約で event_id が無効な場合は-1を指定する。</u>
(tracking)	- 番組追従設定。 時間指定予約では UIE_RESERVE_TRACKING_OFF (追従しない) が設定される。
(tracking_keyword)	- 番組追従設定が連ドラ追従時のキーワード文字列。UTF8 format。 空文字列を指定した場合は、受信機レジデント側のデフォルト設定が適用される
(lastmodified_time)	- 最終更新日時。遠隔予約での同期に利用される。 POSIX time 形式 (1970 年 1 月 1 日 00:00:00 UTC からの経過秒数。非負値。2038 年以降の値は想定しない))
(reserve)	- 将来拡張用。未使用
[OUTPUT] int *subErrNo	

受信機レジデント側が返す詳細エラーコード。初期値として 0 をセットして呼び出される。

type に指定可能なものは以下

UIE_RESERVE_TYPE_IPTV	(0)	多チャンネル
UIE_RESERVE_TYPE_TB	(1)	地上デジタル IP 再送信
UIE_RESERVE_TYPE_BS	(2)	BS 放送 IP 再送信

cycle に指定可能なものは以下:

UIE_RESERVE_CYCLE_ONCE	(0)	予約周期: 1 回
UIE_RESERVE_CYCLE Everyday	(1)	予約周期: 毎日
UIE_RESERVE_CYCLE_WEEKLY	(2)	予約周期: 予約開始日を基準に 1 週間毎
UIE_RESERVE_CYCLE_MON2SAT	(3)	予約周期: 月曜から土曜まで
UIE_RESERVE_CYCLE_MON2FRI	(4)	予約周期: 月曜から金曜まで

reserve_type に指定可能なものは以下:

UIE_RESERVE_RESERVE_TYPE_PLAY	(0)	視聴
UIE_RESERVE_RESERVE_TYPE_RECORD	(1)	録画
UIE_RESERVE_RESERVE_TYPE_DIRECT_REC	(2)	ダイレクト録画

trigger_type に指定可能なものは以下:

UIE_RESERVE_TRIGGER_TYPE_TIME	(0)	時間指定予約
UIE_RESERVE_TRIGGER_TYPE_EVENT	(1)	番組指定予約

tracking に指定可能なものは以下:

UIE_RESERVE_TRACKING_OFF	(0)	追従しない
UIE_RESERVE_TRACKING_ON	(1)	追従する
UIE_RESERVE_TRACKING_KEYWORD	(2)	連ドラ追従する。オプション

Returns:

成功時 UIE_RESERVE_ERR_OK (0)

失敗時: 負の値

UIE_RESERVE_ERR_UNAVAILABLE	登録不可能(最大件数超過など)
UIE_RESERVE_ERR_INVALID_PARAMETER	引数が不正
UIE_RESERVE_ERR_NG	その他のエラー
UIE_RESERVE_ERR_NOTREADY	予約モジュール準備中
UIE_RESERVE_ERR_FULL	容量不足のため録画予約不可。本エラーを返す 判断は受信機レジデントに依存する。

Note:

時間指定予約か番組指定予約かどうかの区別は trigger_type によって区別されます。

ダイレクト録画の際は trigger_type「時間指定」、tracking「追従しない」が設定されることを想定します。

受信機レジデント側には、GUI 側から渡されたサービス ID、ネットワーク ID に対する有効チェックを行うことを期待します。

ダイレクト録画が選択されてかつ現在時刻より前の開始時間が設定された場合には、即録画開始となる想定です。

```
int uie_Reserve_edit(char reserve_id[], int type, int network_id, int service_id, int start_date, int duration, int cycle,
int reserve_type, int trigger_type, int rating, const char* event_title, int event_id, int tracking, char*
tracking_keyword, int lastmodified_time, void* reserve, int *subErrNo)
```

予約情報の編集を行う。

Parameters:

(reserve_id)	- 変更対象の予約 ID 文字列
(type)	- 予約情報の放送種別
(network_id)	- 予約情報のネットワーク ID
(service_id)	- 予約情報のサービス ID
(start_date)	- 予約開始日時 POSIX time 形式 (1970 年 1 月 1 日 00:00:00 UTC からの経過秒数。非負値。2038 年以降の値は想定しない)
(duration)	- 予約時間長 (秒)
(cycle)	- 予約周期
(reserve_type)	- 予約種別
(trigger_type)	- 発火条件(番組指定予約・時間指定予約)
(rating)	- Rating of the reservation ARIB STD-B10 のコンテンツ記述子の内容に準じる 値+3 歳が推奨最低年齢となる。 // TBD: 削除を検討中 (EPG API から取得)
(event_title)	- 予約情報のイベント(番組) タイトル文字列。UTF8 format.
(event_id)	- 予約情報のイベント(番組)ID <u>番組指定予約の際は必須。時間指定予約で event_id が無効な場合は-1を指定する。</u>
(tracking)	- 番組追従設定。 時間指定予約では UIE_RESERVE_TRACKING_OFF (追従しない) が設定される。
(tracking_keyword)	- 番組追従設定が連ドラ追従時のキーワード文字列。UTF8 format。 空文字列を指定した場合は、受信機レジデント側のデフォルト設定が適用される
(lastmodified_time)	- 最終更新日時。遠隔予約での同期に利用される。 POSIX time 形式 (1970 年 1 月 1 日 00:00:00 UTC からの経過秒数。非負値。2038 年以降の値は想定しない)
(reserve)	- 将来拡張用。未使用
[OUTPUT] int *subErrNo	
受信機レジデント側が返す詳細エラーコード。初期値として 0 をセットして呼び出される。	

type に指定可能なものは以下

UIE_RESERVE_TYPE_IPTV	(0)	多チャンネル
UIE_RESERVE_TYPE_TB	(1)	地上デジタル IP 再送信
UIE_RESERVE_TYPE_BS	(2)	BS 放送 IP 再送信

cycle に指定可能なものは以下:

UIE_RESERVE_CYCLE_ONCE	(0)	予約周期: 1 回
UIE_RESERVE_CYCLE Everyday	(1)	予約周期: 毎日
UIE_RESERVE_CYCLE_WEEKLY	(2)	予約周期: 予約開始日を基準に 1 週間毎
UIE_RESERVE_CYCLE_MON2SAT	(3)	予約周期: 月曜から土曜まで
UIE_RESERVE_CYCLE_MON2FRI	(4)	予約周期: 月曜から金曜まで

reserve_type に指定可能なものは以下:

UIE_RESERVE_RESERVE_TYPE_PLAY	(0)	視聴
UIE_RESERVE_RESERVE_TYPE_RECORD	(1)	録画
UIE_RESERVE_RESERVE_TYPE_DIRECT_REC	(2)	ダイレクト録画

trigger_type に指定可能なものは以下:

UIE_RESERVE_TRIGGER_TYPE_TIME	(0)	時間指定予約
UIE_RESERVE_TRIGGER_TYPE_EVENT	(1)	番組指定予約

tracking に指定可能なものは以下:

UIE_RESERVE_TRACKING_OFF	(0)	追従しない
UIE_RESERVE_TRACKING_ON	(1)	追従する
UIE_RESERVE_TRACKING_KEYWORD	(2)	連ドラ追従する。オプション

Returns:

成功時 UIE_RESERVE_ERR_OK (0)

失敗時: 負の値

UIE_RESERVE_ERR_UNAVAILABLE	変更不可能(予約実行厨など)
UIE_RESERVE_ERR_INVALID_PARAMETER	引数が不正(予約 ID 不正も含む)
UIE_RESERVE_ERR_NG	その他のエラー
UIE_RESERVE_ERR_NOTREADY	予約モジュール準備中
UIE_RESERVE_ERR_FULL	容量不足のため録画予約不可。本エラーを返す判断は受信機レジデントに依存する。

Note:

実行中の録画については、原則として変更不可能。ダイレクト録画予約の終了時間変更が可能かについては TBD。

時間指定予約か番組指定予約かどうかの区別は trigger_type によって区別されます。

ダイレクト録画の際は trigger_type「時間指定」、tracking「追従しない」が設定されることを想定します。

受信機レジデント側には、GUI 側から渡されたサービス ID、ネットワーク ID に対する有効チェックを行うことを期待します。

ダイレクト録画が選択されてかつ現在時刻より前の開始時間が設定された場合には、即録画開始となる想定です。

```
int uie_Reserve_delete(const char reserve_id[] , int delete_flag, int *subErrNo)
```

delete_flag に UIE_RESERVE_DELETE_FLAG_OFF が設定された場合には、予約情報の削除を行う。add_delete_flag に UIE_RESERVE_DELETE_FLAG_ON が設定された場合には、削除の変わりに当該予約に対して削除フラグを付与するのみとし、関数を呼んだ直後の削除は行わない。

削除フラグを付与する機能に対応していない端末レジデントに対して UIE_RESERVE_DELETE_FLAG_ON を設定した場合にはエラー UIE_RESERVE_ERR_NG が返る。

Parameters:

char reserve_id[] 予約 ID 文字列

int delete_flag 削除フラグの有無

UIE_RESERVE_DELETE_FLAG_ON (1) 当該予約を削除せずに削除フラグを付与する

UIE_RESERVE_DELETE_FLAG_OFF (0) 当該予約を削除する

[OUTPUT] int *subErrNo

受信機レジデント側が返す詳細エラーコード。初期値として 0 をセットして呼び出される。

Returns:

成功時 UIE_RESERVE_ERR_OK (0)

失敗時: 負の値

UIE_RESERVE_ERR_UNAVAILABLE 削除不可能

UIE_RESERVE_ERR_INVALID_PARAMETER 引数が不正(予約 ID 不正も含む)

UIE_RESERVE_ERR_NG その他のエラー

UIE_RESERVE_ERR_NOTREADY 予約モジュール準備中

Note:

実行中の予約を削除する場合には、uie_Reserve_delete()を呼ぶ前に uie_Reserve_cancel()を呼ぶ必要がある。

```
int uie_Reserve_cancel(const char reserve_id[], int *subErrNo)
```

特定の視聴予約・録画予約情報を取り消す。

指定された予約情報はキャンセルされたものとして扱われる。削除されたことは意味しない。

当該の予約情報は予約時間内である限り、有効なものとして扱われる。

Parameters:

(reserve_id) - 予約 ID

[OUTPUT] int *subErrNo

受信機レジデント側が返す詳細エラーコード。初期値として 0 をセットして呼び出される。

Returns:

成功時 UIE_RESERVE_ERR_OK (0)

失敗時: 負の値

UIE_RESERVE_ERR_UNAVAILABLE

削除不可能

UIE_RESERVE_ERR_INVALID_PARAMETER 引数が不正(予約 ID 不正も含む)

UIE_RESERVE_ERR_NG

その他のエラー

UIE_RESERVE_ERR_NOTREADY

予約モジュール準備中

Note:

本 API は、UIE_RESERVE_EVENT_START イベントを受けて UIEPlayer(GUI)側で予約視聴・録画が開始され、その後ユーザー操作で予約実行がキャンセルされた場合に呼び出されます。

キャンセルした場合、当該予約の UIE_RESERVE_EVENT_FINISH が発行されます。

繰り返し予約の場合は、次回(翌日以降)の予約実行は通常通り行われます。

実行中の視聴予約が取り消された場合にも、当該予約に対する予約終了イベント(UIE_RESERVE_EVENT_FINISH) は発行されます。実行中の録画予約が取り消された場合、録画は停止され、この場合にも UIE_RESERVE_EVENT_FINISH が発行されます。

予約視聴(録画)中予約キャンセル(uie_Reserve_cancel)を実施した場合に予約キャンセル対象となるのは、以下のいずれかの条件を満たす予約情報とします。

1. 指定した予約 ID をもつ
- 2.すでに予約時間(前後のりしろこみ)になっているが予約が実行(EVENT_RESERVE_START を発行)されていない

※ つまり同時録画されている場合に一方の録画予約を停止しても、裏録画されている予約はこれらの条件を満たさない為対象となりません。

```
int uie_Reserve_findDuplicate(char reserve_id[], int type, int network_id, int service_id, int start_date, int duration, int cycle, int reserve_type, char* dup_id[], int dup_state[], int size, void* reserve , int *subErrNo)
```

新規登録時(uie_Reserve_add())、および予約情報更新時(uie_Reserve_edit())の重複チェックに使用する

Parameters:

- (reserve_id) - 予約 ID 文字列。予約情報更新時(uie_Reserve_edit())を呼ぶ前)に設定する。
新規登録時は NULL
- (type) - 放送種別
- (network_id) - 予約情報のネットワーク ID
- (service_id) - 予約情報のサービス ID
- (start_date) - 予約の開始日時 POSIX time 形式 (1970 年 1 月 1 日 00:00:00 UTC からの経過秒数。
非負値。2038 年以降の値は想定しない)
- (duration) - 予約時間長(秒)
- (cycle) - 予約周期
- (reserve_type) - 予約種別
- (char *dup_id[]) - [OUTPUT] 重複する予約 ID 格納用の領域。領域は UIEPlayer が確保。
uie_Reserve_getProperty(UIE_RESERVE_GLOBAL_PROPERTY_ID_LENGTH)
文字分の領域を後述の<size>個確保し、各領域へのポインタの配列が渡される。
- (dup_state[]) - [OUTPUT] 重複状態
(本 API 引数の予約を仮に登録した場合の、既登録予約の重複状態となる)。
dup_id[]と対になり、後述の<size>個分の領域を UIEPlayer が確保済み。
- (size) - dup_id[], dup_state[]の確保済み配列要素数
- (reserve) - 将来拡張用。未使用
- [OUTPUT] int *subErrNo
受信機レジデント側が返す詳細エラーコード。初期値として 0 をセットして呼び出される。

type に指定可能なものは以下

UIE_RESERVE_TYPE_IPTV	(0)	多チャンネル
UIE_RESERVE_TYPE_TB	(1)	地上デジタル IP 再送信
UIE_RESERVE_TYPE_BS	(2)	BS 放送 IP 再送信

cycle に指定可能なものは以下:

UIE_RESERVE_CYCLE_ONCE	(0)予約周期: 1 回
UIE_RESERVE_CYCLE Everyday	(1)予約周期: 毎日
UIE_RESERVE_CYCLE WEEKLY	(2)予約周期: 予約開始日を基準に 1 週間毎

UIE_RESERVE_CYCLE_MON2SAT (3)予約周期:月曜から土曜まで

UIE_RESERVE_CYCLE_MON2FRI (4)予約周期:月曜から金曜まで

reserve_type に指定可能なものは以下:

UIE_RESERVE_RESERVE_TYPE_PLAY	(0)	視聴
UIE_RESERVE_RESERVE_TYPE_RECORD	(1)	録画
UIE_RESERVE_RESERVE_TYPE_DIRECT_REC	(2)	ダイレクト録画

dup_state[]に設定されるものは以下:

UIE_RESERVE_STATE_DUP_OK	(1)	重複するが全て視聴・録画可能
UIE_RESERVE_STATE_DUP_NG	(2)	重複し、一部が視聴・録画できない
UIE_RESERVE_STATE_CANCELED	(3)	実行中止状態 (uie_Reserve_cancel()が行われたイベント)
UIE_RESERVE_STATE_DISABLED	(4)	重複し、全部が視聴・録画できない(予約失敗)

Returns:

成功時: 重複する予約の個数 (0 有効)

失敗時: 負の値

UIE_RESERVE_ERR_INVALID_PARAMETER	引数が不正
UIE_RESERVE_ERR_NG	その他のエラー
UIE_RESERVE_ERR_NOTREADY	予約モジュール準備中

Note:

本関数は、予約の登録(uie_Reserve_add())や、予約の更新(uie_Reserve_edit())を行う前に呼び出し、これから行う登録や更新の結果、予約の重複が発生するかどうかをあらかじめ調べるために使われます。本関数での比較対象は、新規に登録しようとする予約と発火リストに含まれる予約です。ダブル録画可能な端末にて重複する録画予約を設定する際に findDuplicate()を実行すると、重複する予約ID(dup_id[])が取得でき、かつ予約可能のステータス(dup_state[])が取得できるものとします。また、番組追従等にて延長した予約も重複判定対象です。

```
int uie_Reserve_getCurrent(int index , char reserve_id[], int *subErrNo)
```

現在実行中の予約が存在していればその予約情報の ID を取得する

Parameters:

(index) - 複数の予約が実行中の場合を想定し、index=0,1,2 と順番に呼ばれる

[OUTPUT] char reserveld[]

受信機レジデント側が返す予約 ID 文字列の格納領域。UIEPlayer があらかじめ
 uie_Reserve_getProperty(UIE_RESERVE_GLOBAL_PROPERTY_ID_LENGTH) + 1 の領域を
 を確保済み。

[OUTPUT] int *subErrNo

受信機レジデント側が返す詳細エラーコード。初期値として 0 をセットして呼び出される。

Returns:

成功時: UIE_RESERVE_ERR_OK

reserve_id[] に予約 ID 文字列がセットされる。

失敗時:

UIE_RESERVE_ERR_ENDOFDATA	これ以上予約が存在しない場合 (1 つもない場合を含む)
UIE_RESERVE_ERR_NG	その他のエラー
UIE_RESERVE_ERR_NOTREADY	予約モジュール準備中

Note:

本関数で返る予約情報は、視聴予約・録画予約(ダイレクト録画含む)の双方を含みます。

```
int uie_Reserve_getNext(char reserve_id[], int *subErrNo)
```

次に実行される視聴予約情報の予約 ID を取得する

予約視聴の実行開始前に、どの視聴予約が実行されるのかを知るために使われる。

Parameters:

[OUTPUT] char reserveId[]

——— 受信機レジデント側が返す予約 ID 文字列の格納領域。UIEPlayer があらかじめ
uie_Reserve_getProperty(UIE_RESERVE_GLOBAL_PROPERTY_ID_LENGTH) + 1 の領域を
を確保済み。

[OUTPUT] int *subErrNo

——— 受信機レジデント側が返す詳細エラーコード。初期値として 0 をセットして呼び出される。

——— Returns:

——— 成功時: UIE_RESERVE_ERR_OK

——— reserve_id[] に予約 ID 文字列がセットされる。

——— 失敗時:

——— UIE_RESERVE_ERR_UNAVAILABLE ———— 該当予約無し

——— UIE_RESERVE_ERR_NG ———— その他のエラー

——— Note:

——— 本関数は、視聴予約のみを対象とする。録画予約は対象外。


```
int uie_Reserve_findByServiceId(int type, int network_id, int service_id, int index, void* reserve)
```

ネットワーク ID, サービス ID に関連する予約 ID を取得する

パラメータで指定された要求に対応する予約 ID の取得を行う。

インデックスの初期値は 0 を指定し、適切な予約情報に対応する予約 ID を取得する。取得後は、次の予約 ID を取得するためにインデックスに 1 を加えたものを設定する。予約情報は開始時間順で返ってくるものとします。

最後の予約に達した後にこの関数を実行するとエラーが返り、これにより検索終了を判断することが可能である。

Parameters:

_____ (type) _____	放送種別
_____ (network_id) _____	予約情報のネットワーク ID
_____ (service_id) _____	予約情報のサービス ID
_____ (index) _____	検索を開始する予約情報のインデックス
_____ (reserve) _____	将来拡張用。未使用

_____ type に指定可能なものは以下

_____ UIE_RESERVE_TYPE_IPTV _____	(0)	多チャンネル
_____ UIE_RESERVE_TYPE_TB _____	(1)	地上デジタル IP 再送信
_____ UIE_RESERVE_TYPE_BS _____	(2)	BS 放送 IP 再送信

Returns:

_____ 成功時: 予約 ID を表す負でない値
_____ 失敗時: UIE_RESERVE_ERR_NG (-1)

Note:

_____ 返される予約 ID は、実行時間順にソートされていることを想定しています。

使用例:

_____ 特定のサービス ID (チャンネル) に関連するすべての予約情報へのインデックスを取得する場合

```
_____
_____ index = 0;
_____ while(!_ne(index, -1)){
_____     reserve_id = uie_Reserve_findByChannel(0x7780, 701, index);
_____     if(!_ne(reserve_id, -1)){
_____         do_something(reserve_id);
_____     }
_____     index++;
_____ }
```

```
int uie_Reserve_getReserveldByEvent(int type, int network_id, int service_id, int event_id, char reserve_id[], void*
reserve , int *subErrNo)
```

特定のイベント ID(番組)をもつ予約 ID を取得する。番組指定予約が対象となり、時間指定予約は対象外。
関数実行の場合にはネットワーク ID、サービス ID が指定されていることが必要になる。

Parameters:

- (type) - 検索対象となる番組の放送種別
- (network_id) - 検索対象となる番組のネットワーク ID
- (service_id) - 検索対象となる番組のサービス ID
- (event_id) - 検索対象となる番組のイベント(番組)ID
- [OUTPUT] char reserveld[]
受信機レジデント側が返す予約 ID 文字列の格納領域。UIEPlayer があらかじめ
uie_Reserve_getProperty(UIE_RESERVE_GLOBAL_PROPERTY_ID_LENGTH)
+ 1 の領域をを確保済み。
- (reserve) - 将来拡張用。未使用
- [OUTPUT] int *subErrNo
受信機レジデント側が返す詳細エラーコード。初期値として 0 をセットして呼び出される。

type に指定可能なものは以下

UIE_RESERVE_TYPE_IPTV	(0)	多チャンネル
UIE_RESERVE_TYPE_TB	(1)	地上デジタル IP 再送信
UIE_RESERVE_TYPE_BS	(2)	BS 放送 IP 再送信

Returns:

- 成功時: UIE_RESERVE_ERR_OK
reserve_id[]に予約 ID 文字列がセットされる。
- 失敗時:
 - UIE_RESERVE_ERR_INVALID_PARAMETER 引数が不正
 - UIE_RESERVE_ERR_NG その他のエラー
 - UIE_RESERVE_ERR_NOTREADY 予約モジュール準備中

使用例:

特定のサービス ID(チャンネル)およびイベント ID(番組)に関連するすべての予約情報へのインデックスを取得する場合

```
index = 0;
while(!_ne(index, -1)) {
    uie_Reserve_findByEventId(0x7780, 701, 12345678, index, reserve_id[]);
    if(!_ne(index, -1)) {
        do_something(reserve_id);
        index++;
    }
}
```

Notes:

本 API は、番組表から番組詳細ダイアログを表示する際、「視聴予約」「予約削除」のどちらのボタンを出すか判定するために使用されます。

従って、時間指定予約は検索の対象外となります。

```
int uie_Reserve_findByTime(int type, int network_id, int service_id, int start_date, int duration, void* reserve , int
*subErrNo)
```

本 API は、番組表で予約時間帯の表示を行うために使用することを想定しており、パラメータで指定された要求に対応する予約発火データの検索を内部的に行い、該当する予約の件数を返す。個々の予約データの詳細情報は、後述の uie_Reserve_getFindResult(), uie_Reserve_getStringFindResult()で行う。

関数実行の場合にはサービス ID が設定されていることが必要になる。

start_date は int 型の日付データであり、1970/01/01 00:00 UTC からの経過時間(秒)で表現される。たとえば 2008/04/14 00:00 は 4357440 として表現される。duration は秒単位で表現された 24 時間以下の値とする。この条件を満たさない場合にはエラーとなる。

Parameters:

(type)	-検索対象となる放送種別
(network_id)	-検索対象となるネットワーク ID
(service_id)	-検索対象となるサービス ID
(start_date)	-検索対象となる開始日時 POSIX time 形式 (1970 年 1 月 1 日 00:00:00 UTC からの経過秒数。非負値。2038 年以降の値は想定しない))
(duration)	-検索対象となる時間長 (秒)。
(reserve)	-将来拡張用。未使用
[OUTPUT] int *subErrNo	

受信機レジデント側が返す詳細エラーコード。初期値として 0 をセットして呼び出される。

type に指定可能なものは以下

UIE_RESERVE_TYPE_IPTV	(0)	多チャンネル
UIE_RESERVE_TYPE_TB	(1)	地上デジタル IP 再送信
UIE_RESERVE_TYPE_BS	(2)	BS 放送 IP 再送信

Returns:

成功時: 該当する予約の件数 (0 以上)

失敗時:

UIE_RESERVE_ERR_INVALID_PARAMETER 引数が不正

UIE_RESERVE_ERR_NG その他のエラー

UIE_RESERVE_ERR_NOTREADY 予約モジュール準備中

Note:

返される予約 ID は、実行時間順にソートされていることを想定しています。

使用例:

2008/4/14 に重なるすべての予約情報を取得する

```
index = 0;
starttime = 4357440*60; // seconds from 1970/0/0 to 2008/4/14
count = uie_Reserve_findByTime(0x7780, 701, starttime, 1440*60);
for(index = 0; _lt(index, count); index++) {
    uie_getFindResult(index, ...)
    uie_getStringFindResult(index, ...)
}
```

```
int uie_Reserve_getFindResult(int index, int property , int *subErrNo)
```

uie_Reserve_findByTime()で検索された予約発火データの index 番目について、property で指定された int 型の属性値を取得する。取得する内容は、uie_Reserve_get()と同等。

Parameters:

(index) - インデックス番号

uie_Reserve_findByTime()のあと、最初は 0 で呼ばれ、その後1つずつ値を増やして繰り返し呼ばれる。

(property) - 取得したい属性値の種類。指定可能なものは以下:

UIE_RESERVE_PROPERTY_SERVICE_ID	(0)	サービス ID
UIE_RESERVE_PROPERTY_START_DATE	(1)	予約開始日時(重複無視)
UIE_RESERVE_PROPERTY_DURATION	(2)	予約時間長(重複無視)
UIE_RESERVE_PROPERTY_ACTUAL_START_DATE	(15)	予約開始日時(重複考慮済)
UIE_RESERVE_PROPERTY_ACTUAL_DURATION	(16)	予約時間長(重複考慮済)
UIE_RESERVE_PROPERTY_CYCLE	(3)	予約周期
UIE_RESERVE_PROPERTY_RATING	(4)	年齢制限情報値
UIE_RESERVE_PROPERTY_STATE	(5)	予約の重複状態
UIE_RESERVE_PROPERTY_EVENT_ID	(7)	イベント ID
UIE_RESERVE_PROPERTY_NETWORK_ID	(9)	ネットワーク ID
UIE_RESERVE_PROPERTY_RESERVE_TYPE	(10)	予約種別
UIE_RESERVE_PROPERTY_TRIGGER_TYPE	(17)	発火種別
(時間指定予約・番組指定予約)		
UIE_RESERVE_PROPERTY_TRACKING	(11)	放送時間変更の追従設定
UIE_RESERVE_PROPERTY_TYPE	(13)	放送種別
UIE_RESERVE_PROPERTY_DELETE_FLAG	(14)	遠隔予約用削除フラグ
UIE_RESERVE_PROPERTY_LASTMODIFIED_TIME	(18)	最終更新日時

[OUTPUT] int *subErrNo

受信機レジデント側が返す詳細エラーコード。初期値として 0 をセットして呼び出される。

Returns:

成功時: int 型で属性値を返す。

property = UIE_RESERVE_PROPERTY_SERVICE_ID
サービス ID(チャンネル)

property = UIE_RESERVE_PROPERTY_START_DATE
予約重複を考慮しない場合の予約開始日時。

POSIX time 形式 (1970 年 1 月 1 日 00:00:00 UTC からの経過秒数。非負値。2038 年以降の値は想定しない))

property = UIE_RESERVE_PROPERTY_DURATION

予約重複を考慮しない場合の予約時間長(秒)

property = UIE_RESERVE_PROPERTY_ACTUAL_START_DATE

予約重複を考慮した場合の予約開始日時。繰り返し予約の場合は、直近の時刻となる。

重複のため予約が実行されない場合は 0 を返す。

POSIX time 形式 (1970 年 1 月 1 日 00:00:00 UTC からの経過秒数。非負値。2038 年以降の値は想定しない))

property = UIE_RESERVE_PROPERTY_ACTUAL_DURATION

予約重複を考慮した場合の予約時間長(秒)

重複のため予約が実行されない場合は 0 を返す。

property = UIE_RESERVE_PROPERTY_CYCLE

UIE_RESERVE_CYCLE_ONCE (0) 予約周期: 1 回

UIE_RESERVE_CYCLE Everyday (1) 予約周期: 毎日

UIE_RESERVE_CYCLE_WEEKLY (2) 予約周期: 予約開始日を基準に 1 週間毎

UIE_RESERVE_CYCLE_MON2SAT (3) 予約周期: 月曜から土曜まで

UIE_RESERVE_CYCLE_MON2FRI (4) 予約周期: 月曜から金曜まで

property = UIE_RESERVE_PROPERTY_RATING

ARIB STD-B10 のコンテンツ記述子の内容に準じる

値 +3 歳が推奨最低年齢となる。 // TBD: 削除を検討中 (EPG API から取得)

property = UIE_RESERVE_PROPERTY_STATE

UIE_RESERVE_STATE_NO_DUP (0) 重複なし

UIE_RESERVE_STATE_DUP_OK (1) 重複するが全て視聴・録画可能

UIE_RESERVE_STATE_DUP_NG (2) 重複し、一部が視聴・録画できない

~~UIE_RESERVE_STATE_CANCELED (3) 実行中止状態 (uie_reserve_cancel()が
行われたイベント)~~

UIE_RESERVE_STATE_DISABLED (4) 重複し、全部が視聴・録画できない

property = UIE_RESERVE_PROPERTY_EVENT_ID

特定の予約に対するイベント ID(番組)

この値は EPG API に event_id として渡される。

時間指定予約で、event_id が無効な場合-1 を返す

property = UIE_RESERVE_PROPERTY_NETWORK_ID

特定の予約に対するネットワーク ID。

property = UIE_RESERVE_PROPERTY_RESERVE_TYPE

UIE_RESERVE_RESERVE_TYPE_PLAY	(0)	視聴
UIE_RESERVE_RESERVE_TYPE_RECORD	(1)	録画
UIE_RESERVE_RESERVE_TYPE_DIRECT_REC	(2)	ダイレクト録画

property = UIE_RESERVE_PROPERTY_TRIGGER_TYPE

UIE_RESERVE_TRIGGER_TYPE_TIME	(0)	時間指定予約
UIE_RESERVE_TRIGGER_TYPE_EVENT	(1)	番組指定予約

property = UIE_RESERVE_PROPERTY_TRACKING

UIE_RESERVE_TRACKING_OFF	(0)	追従しない
UIE_RESERVE_TRACKING_ON	(1)	追従する
UIE_RESERVE_TRACKING_KEYWORD	(2)	連ドラ追従する。オプション

property = UIE_RESERVE_PROPERTY_TYPE

UIE_RESERVE_TYPE_IPTV	(0)	多チャンネル
UIE_RESERVE_TYPE_TB	(1)	地上デジタル IP 再送信
UIE_RESERVE_TYPE_BS	(2)	BS 放送 IP 再送信

property = UIE_RESERVE_PROPERTY_DELETE_FLAG

UIE_RESERVE_DELETE_FLAG_ON	(1)	削除フラグが付与されている
UIE_RESERVE_DELETE_FLAG_OFF	(0)	削除フラグが付与されていない

削除フラグが付与されている予約情報は端末内部には存在するが予約情報は実行されない。遠隔予約機能に対応する場合に考慮する必要がある。

失敗時: 負の値

UIE_RESERVE_ERR_INVALID_PARAMETER 引数が不正

UIE_RESERVE_ERR_ENDOFDATA これ以上データが存在しない

UIE_RESERVE_ERR_NG	その他のエラー
UIE_RESERVE_ERR_NOTREADY	予約モジュール準備中

Note: 番組指定予約の場合でも、予約開始時間や予約時間などの時間情報は有効です。

```
int uie_Reserve_getStringFindResult(int index, int property, char resultStr[], int size , int *subErrNo)
```

uie_Reserve_findByTime()で検索された予約発火データの index 番目について、property で指定された char 型の属性値を取得する。取得する内容は、uie_Reserve_getString()と同等だが、予約 ID を返す property が追加されている。
呼び出し元(UIE Player)が文字列領域を用意する。

Parameters:

(index) - インデックス番号

uie_Reserve_findByTime()のあと、最初は 0 で呼ばれ、その後1つずつ値を増やして繰り返し呼ばれる。

(property) - 取得したい属性値の種類。指定可能なものは以下:

UIE_RESERVE_PROPERTY_EVENT_TITLE	(6)	// TBD: EPG API から取得を検討中
UIE_RESERVE_PROPERTY_TRACKING_KEYWORD	(12)	連ドラ追従キーワード
uie_Reserve_get(UIE_RESERVE_PROPERTY_TRACKING)が		
UIE_RESERVE_TRACKING_KEYWORD の場合有効		
UIE_RESERVE_PROPERTY_RESERVE_ID	(19)	予約 ID 文字列

(resultStr[]) [OUTPUT] - 指定したプロパティの内容。UTF8 フォーマット。

情報が存在しない場合には、空文字列を返す。

文字列の受け渡しに必要な領域は、あらかじめ UIEPlayer 側で確保される。

(size) - 前述の resultStr[]のサイズを表す(文字終端分を含む)

[OUTPUT] int *subErrNo

受信機レジデント側が返す詳細エラーコード。初期値として 0 をセットして呼び出される。

Returns:

成功時:

UIE_RESERVE_ERR_OK。resultStr[]がセットされる

失敗時: 負の値

UIE_RESERVE_ERR_INVALID_PARAMETER 引数が不正

UIE_RESERVE_ERR_ENDOFDATA これ以上データが存在しない

UIE_RESERVE_ERR_NG その他のエラー

UIE_RESERVE_ERR_NOTREADY 予約モジュール準備中

```
int uie_Reserve_initialize(int *subErrNo)
```

予約に関連するデータの初期化を行う。

この関数は予約に関連するすべてのデータを初期化する。

Parameters:

[OUTPUT] int *subErrNo

受信機レジデント側が返す詳細エラーコード。初期値として 0 をセットして呼び出される。

Returns:

成功時: UIE_RESERVE_ERR_OK (0)

失敗時: UIE_RESERVE_ERR_NG (-1)

Note:

この API 呼び出しの際に予約録画を実行中だった場合、

受信機レジデント側によって、録画が停止されることを想定しています。

廃止。予約データの初期化は、`uie_System_request(SYSTEM_REQUEST_INITIALIZE_PROPERTY)`で行う。

```
int uie_Reserve_getLastModifiedTime(int *subErrNo)
```

最後に予約情報の変更(追加・修正・削除)があった時刻を取得する。

この値は遠隔予約の同期処理に利用される。一度も操作がない場合には 0 を返す。

受信機レジデントが遠隔予約機能に対応していない場合にはこの関数の返り値がエラーとなる。

Parameters:

[OUTPUT] int *subErrNo

受信機レジデント側が返す詳細エラーコード。初期値として 0 をセットして呼び出される。

Returns:

成功時: 最後に予約情報の変更(追加・修正・削除)があった時刻

POSIX time 形式 (1970 年 1 月 1 日 00:00:00 UTC からの経過秒数。非負値。2038 年以降の値は想定しない))

失敗時:

UIE_RESERVE_ERR_UNAVAILABLE	端末機能なし
UIE_RESERVE_ERR_NG	その他のエラー
UIE_RESERVE_ERR_NOTREADY	予約モジュール準備中

7.2. 視聴・録画予約データ関連 API イベント

UIEPlayer will receive the following system events by calling uie_getEvent().

event type:

```
#define RESERVE_EVENT_START (0x700)
```

予約視聴、予約録画の開始を通知するイベント。視聴予約、録画予約において共通のイベントを利用する。実際に予約視聴、録画が開始されるまで、複数の詳細種別イベントが順番に発生します。詳細は NTT ぷらら資料「1. 予約基本シーケンス」参照。なお、以下に示す秒数はあくまでも例であり該当の録画予約の前に重複する予約が存在せず、のりしろが 30 秒、MCjoin の分散が 60 秒の範囲内である場合のものになります。詳細なタイミングは TBD。

event parameter :

param : 予約種別

UIE_RESERVE_RESERVE_TYPE_PLAY	(0)	視聴
UIE_RESERVE_RESERVE_TYPE_RECORD	(1)	録画
UIE_RESERVE_RESERVE_TYPE_DIRECT_REC	(2)	ダイレクト録画

param2 : 予約開始イベント詳細種別

0 : 予約発火通知 : 指定開始時刻 105 秒前。

GUI 側では予約情報ステータスを「予約実行中」にする

1: 開始予告通知 : 指定開始時刻 45+XX 秒前

視聴予約の場合か、録画予約で後述の裏録画フラグが 0 の場合はイベント受信から YY 秒後に GUI 側で以下の処理を行い視聴画面の強制切替に備える

- VOD, 録画コンテンツの再生中の場合はそれを停止

- MC 全画面表示に切り替え、「まもなく録画開始します」「まもなく予約視聴開始します」のダイアログ表示を行う。

2: 開始準備通知 : 指定開始時刻 30+XX 秒前

GUI 側では「録画準備中です」「視聴予約準備中です」のダイアログ表示

受信機レジデントはこのイベント発火後に対象チャンネルへの選局処理を行う。

GUI 側からの uie_Media_openService()の呼び出しは必要としない。

3: 開始処理 : 実際に録画または視聴が開始された時点で発生する。

通常の視聴・録画予約の場合は、指定開始時刻 30 秒前に発生。ダイレクト録画の場合は、録画開始時に発生する。

録画の場合、GUI 側では簡易情報ダイアログに「録画中」を表示し、

録画情報ステータスを「録画中」にする。

※ダイレクト録画の場合は、録画開始時に「3:録画開始処理」だけが通知される

param3: 裏録画フラグ

0: 録画開始と共に、受信機レジデントが視聴中画面の強制切替を行う。

1: 裏録画

2: ダイレクト録画

※ 視聴予約の際には裏録画フラグは常に 0 に設定されます。

(const char *)eventData: reserve_id (開始される視聴予約 ID/録画予約 ID 文字列)

Note:

上記にある XX 秒および YY 秒は MC Join のタイミングをずらすために設定される値であり、本 API では定義されません。

GUI 側で全画面表示に切り替える際、

uie_System_request(SYSTEM_REQUEST_MODE_AVAILABILITY) および
uie_System_request(SYSTEM_REQUEST_MODE_TRANSITION) は不要。

#define RESERVE_EVENT_FINISH (0x701)

予約の終了を通知するイベント。終了時間から 30 秒後に通知される。(時間幅 30 秒は T.B.D)視聴予約では OPTIONAL とする。このイベントを受けて、録画の場合 GUI 側では「録画停止しました」のダイアログを表示する。
 なお、続けて別の視聴予約・録画予約が開始される場合は、本イベントの通知は行いません。その場合、最初の視聴予約・録画予約の終了 30 秒前に、次の視聴予約・録画予約に関する RESERVE_EVENT_START イベントが発行されます。

event parameter :

param : 予約種別

UIE_RESERVE_RESERVE_TYPE_PLAY	(0)	視聴
UIE_RESERVE_RESERVE_TYPE_RECORD	(1)	録画
UIE_RESERVE_RESERVE_TYPE_DIRECT_REC	(2)	ダイレクト録画

(const char *)eventData : reserve_id (終了する予約 ID 文字列)

#define RESERVE_EVENT_DATACHANGED (0x704)

event parameter:

(const char *)eventData : reserve_id (変更対象となる予約 ID (全体変更の場合 NULL))

視聴・録画予約データを受信機側で変更した場合に通知される。具体的には終了した予約が削除された場合や、放送時間変更への追従等を想定しています。

なお、本 API 関数の実行の結果データの変更があった場合は、このイベントでの通知対象外となります。

8. 変更履歴

07/13/2009

- ・時間の単位を分ではなく秒に変更 (EPG の時間フォーマットと統一)
- ・uie_Reserve_findDuplicate()を修正。戻り値で重複の種類(重複無し、重複登録不可、重複登録可)を表す。
- ・uie_Reserve_cancel
- ・uie_Reserve_findByProgramId()の使用目的を追記
- ・将来拡張用の引数を一部の API に追加

07/22/2009

- ・パレンタルレート の値を ARIB STB-B10 に従う旨変更
- ・uie_Reserve_getString()で文字列用領域を UIEPlayer 側が確保する形に変更
- ・時間指定予約の場合、uie_Reserve_get(UIE_STBVR_PROPERTY_EVENT_ID)は-1 を返すことを明記
- ・関数名・引数・定数も含め、用語を ARIB ベースに統一:チャンネル → サービス ID、プログラム(番組) → イベント
 - uie_Reserve_findByChannel() → uie_Reserve_findByChannel()に名称変更
 - uie_Reserve_findByProgramId() → uie_Reserve_findByEventId()に名称変更
- ・uie_Reserve_checkEvent() → uie_Reserve_checkNextEvent()に名称変更
- ・uie_Reserve_getEvent() → uie_Reserve_getNextEvent()に名称変更
- ・時間の単位(開始時間・期間)を秒に変更対応で漏れがあったので修正
 - uie_Reserve_add(), uie_Reserve_findDuplicate(), uie_Reserve_findByTime()

07/27/2009 Revision2.2

- ・日本語化
- ・uie_Reserve_load(), uie_Reserve_refresh()を削除
- ・uie_Reserve_init(), uie_Reserve_exit()を追加
- ・uie_Reserve_get(UIE_STBVR_PROPERTY_ID)で渡される ID の記述を修正
- ・番組追従設定を追加
 - uie_Reserve_get(UIE_STBVR_PROPERTY_TRACKING)
 - uie_Reserve_getString(UIE_STBVR_PROPERTY_TRACKING_KEYWORD)
 - uie_Reserve_add(), uie_Reserve_edit()
- ・uie_Reserve_initialize()に記述追加
- ・予約を指し示す値を、インデックススペースから予約 ID ベースに変更
 - uie_Reserve_get(), uie_Reserve_getString(), uie_Reserve_delete(), uie_Reserve_cancel()
 - uie_Reserve_findByServiceId(), uie_Reserve_findByEventId(), uie_Reserve_findByTime()
- ・uie_Reserve_cancel()で実行中の予約が取り消された場合の動作を追加
- ・機種依存情報を返す uie_Reserve_getProperty()追加
- ・uie_Reserve_findDuplicate()で複数の重複予約データを返す形に変更

- ・ uie_Reserve_getCurrent() を変更。同時に複数の予約が実行中な場合に対応。
- ・ uie_Reserve_getNext() が仕様予約のみを対象とする旨記述。
- ・ uie_Reserve_getIndexFromId() を削除
- ・ 予約の開始・終了を通知するイベント RESERVE_EVENT_PLAY_START/FINISH を追加
 - uie_Reserve_checkNextEvent(), uie_Reserve_getNextEvent() を削除
- ・ 視聴・録画予約データの変更を通知するイベント RESERVE_EVENT_DATACHANGED を追加
- ・ uie_Reserve_findByServiceId(), uie_Reserve_FindByTime() で返される予約データは
予約開始順にソートされていることを想定している旨追加。

07/31/2009 Revision2.2.1

- ・ 「機能分担」が Revision2.2 の時点において更新されていなかったのを修正
- ・ uie_Reserve_getString() の返り値が int になっていなかったのを修正
- ・ uie_Reserve_edit() の第一引数が Revision2.2 の際の変更漏れで index のままとなっていたため、reserve_id に修正
- ・ サービスを指定する際は、放送種別 (type) ・ network_id ・ service_id の 3 つを指定するように全体を統一。
uie_Reserve_get(), Reserve_add(), uie_Reserve_edit(), uie_Reserve_findDuplicate(),
uie_Reserve_findByServiceId(), uie_Reserve_findByEventId(), uie_Reserve_findByTime() を修正。また、これに関連して録画種別を表す名称を "type" → "reserve_type" に変更。
- ・ RESERVE_EVENT_CHANED イベントのパラメータで、変更があった予約 ID を通知するように追記
- ・ 全予約データを取得するために、uie_Reserve_getReserveld() を追加
- ・ uie_Reserve_findByServiceId() は未使用に付き廃止
- ・ UIE_STBVR_xxx を UIE_RESERVE_xxx にリネーム
- ・ uie_Reserve_add(), uie_Reserve_edit() の引数から rating (パレンタルレベル) を削除。
- ・ 呼び出しシーケンスサンプルを追加
- ・ インデックス値を指定する関数の戻り値に、データが存在しない旨を表す UIE_RESERVE_ERR_ENDOFDATA を追加し、その他のエラーと区別

07/31/2009 Revision2.2.2

- ・ uie_Reserve_add(), uie_Reserve_edit() の連ドラ予約時に、キーワードが空文字列を与えられた場合、受信機レジデント側のデフォルトキーワードが使用されることを追記
- ・ 不正な引数を表す UIE_RESERVE_ERR_INVALID_PARAMETER と、無効エラーを通知する UIE_RESERVE_ERR_UNAVAILABLE を追加し、全 API のエラーコードを見直し。
- ・ 全 API に、詳細エラーコードを格納用の引数を追加
- ・ 予約録画の開始と終了を通知する API イベント、RESERVE_EVENT_RECORD_START, RESERVE_EVENT_RECORD_FINISH を追加。

08/24/2009 Revision2.2.3

- ・ RESERVE_EVENT_RECORD_START, RESERVE_EVENT_RECORD_FINISH を視聴・録画予約データ関連イベント

一覧に追加。

- ・ 予約種別を UIE_RESERVE_PROPERTY_CATEGORY から、UIE_RESERVE_PROPERTY_RESERVE_TYPE へ名称変更。
- ・ 遠隔予約機能対応として、uie_Reserve_getLastModifiedTime() 関数を追加。uie_Reserve_get() に property=UIE_RESERVE_PROPERTY_DELETE_FLAG を追加。uie_Reserve_delete() の引数に delete_flag を追加。uie_Reserve_getProperty() に property=UIE_RESERVE_GLOBAL_PROPERTY_DELETE_FLAG および property=UIE_RESERVE_GLOBAL_PROPERTY_REMOTE_RESERVE を追加。

9/14/2009 Revision2.2.4

- ・ 時間指定予約・番組指定予約、予約発火データ・予約登録データについての説明を「用語について」に追加
- ・ イベントの値の定義を 0x500 台から 0x700 台に変更。
- ・ 全 API について、予約 ID の型を int から char* に変更。
- ・ 「手動録画」→「ダイレクト録画」に用語統一。uie_Reserve_getProperty() の UIE_RESERVE_RESERVE_TYPE_MANUAL_REC を UIE_RESERVE_RESERVE_TYPE_DIRECT_REC に修正。
- ・ uie_Reserve_getReserveld() で返される予約情報は、発火時間順でソートされていることを明記。
- ・ uie_Reserve_add(), uie_Reserve_edit() で、番組指定予約の場合は時間情報に 0 を設定する旨を記述。
- ・ uie_Reserve_edit() で実行中の予約は原則として変更不可能な旨を記述(手動録画予約の終了時間が変更可能かどうかについては TBD)。
- ・ uie_Reserve_findDuplicate() の引数に、予約 ID 文字列(reserve_id)を追加。また、どのようなタイミングで呼び出されるかを追記。
- ・ uie_Reserve_get() で得られる予約開始時間・予約時間長を、重複予約を考慮しない場合と、考慮した場合の2種類の値を返せるように変更。
- ・ uie_Reserve_get(), uie_Reserve_getString() は、繰り返し予約の場合、直近に実行される予約の情報(開始時間など)を返す旨を追記。
- ・ uie_Reserve_get() の UIE_RESERVE_PROPERTY_TYPE で返される予約種別に、番組指定予約・時間指定予約を明示するビットフィールドを追加
- ・ uie_Reserve_findByTime() で得られた予約データの情報を順番に取得する uie_Reserve_getFindResult(), uie_Reserve_getStringFindResult() を新設(従来の uie_Reserve_get() は直近に実行される予約が対象で、毎日繰り返し予約などの情報を番組表に反映させることが出来なため)。これに伴い、uie_Reserve_findByTime() は検索された予約データの個数を返す仕様に変更。
- ・ uie_Reserve_findByEventId() を uie_Reserve_getReserveldByEvent() に名前変更。また、番組指定予約が対象で、時間指定予約は対象外になる旨を追記。
- ・ API イベントで返す予約 ID を文字列形式にしたため、イベントパラメータの変数名を変更。また、録画開始イベントをさらにネットワークス「録画基本シーケンス」に従ってイベント種別の詳細化を行った(予約発火通知・録画開始予告通知・録画開始準備通知・録画開始処理 の4つ)
- ・ uie_Reserve_getNext() は不要と思われるので廃止。

10/2/2009 Revision2.2.5

- ・ ダイレクト録画時の RESERVE_EVENT_RECORD_START イベントに関する記述で、誤解を招く点を修正
- ・ パレンタルレベルの指定と取得を復活。uie_Reserve_add(), uie_Reserve_edit(), uie_Reserve_get(), uie_Reserve_getFindResult() で削除されていた当該箇所を元に戻した。
- ・ uie_Reserve_add(), uie_Reserve_edit() に番組指定予約・時間指定予約の区別を指定する trigger_type 引数を追加。これに伴い、uie_Reserve_get(), uie_Reserve_getFindResult() の UIE_RESERVE_PROPERTY_TYPE で番組指定・時間指定の区別を返すようにした 2.2.4 の変更をやめ、UIE_RESERVE_TRIGGER_TYPE で番組指定・時間指定の区別を返すように修正。
- ・ 予約データの初期化を行う uie_Reserve_initialize() を廃止。uie_System_request(SYSTEM_REQUEST_INITIALIZE_PROPERTY) で予約データの初期化を行うこととする。
- ・ 視聴予約と録画予約の動作シーケンスの共通化に伴い、RESERVE_EVENT_RECORD/PLAY_START/FINISH イベントを、RESERVE_EVENT_START/FINISH に一本化し、パラメータで録画種別を区別するようにした。
- ・ Rev2.2.4 で予約 ID を文字列としたことに伴い、uie_Reserve_findDuplicate() の引数を変更。
- ・ Rev2.2.4 で予約 ID を文字列としたことに伴い、RESERVE_EVENT_DATACHANGED イベントで全件更新を表す値を-1 から NULL に変更。
- ・ uie_Reserve_add(), uie_Reserve_edit() に遠隔予約用の各レコードの最終更新日時をセットするパラメータ lastmodified_time を追加。また、get(UIE_RESERVE_PROPERTY_LASTMODIFIED_TIME) を追加。

10/16/2009 Revision2.2.6

- ・ 2.2.5 での修正漏れ: 「4. 機能分担の想定」をぶら下仕様「1. 予約基本シーケンス」の内容に合わせて変更(視聴・録画共に受信機レジデント側が選局を行う)。
- ・ 予約開始イベントを受けて GUI 側で全画面表示に切り替える際、uie_System_request(SYSTEM_REQUEST_MODE_AVAILABILITY) および uie_System_request(SYSTEM_REQUEST_MODE_TRANSITION) は不要となる旨を追記。
- ・ uie_Reserve_getString(), uie_Reserve_add() の TBD コメントを削除
- ・ uie_Reserve_findByTime() の説明欄より index に関する説明を削除
- ・ 2.2.5 において、UIE_RESERVE_PROPERTY_RESERVE_ID の値を変更(15 =>19)
- ・ UIE_RESERVE_STATE_CANCELED を廃止
- ・ uie_Reserve_cancel() を実行した際には、必ず UIE_RESERVE_EVENT_FINISH が上がる旨を明記。また、録画予約の停止も本関数を用いて行うことを明記。
- ・ 予約モジュールの起動が完了していない状況を示すエラーとして UIE_RESERVE_ERR_NOTREADY(-5) を uie_Reserve_init/exit 以外の各関数に追加
- ・ UIE_RESERVE_EVENT_START の説明文に追記。
- ・ uie_Reserve_findDuplicate() では、予約しようとしている登録と発火リストを比較する旨を追記。

10/30/2009 Revision2.2.7

- ・ uie_Reserve_findByTime() の引数の説明を詳細化。duration は 24 時間以下であることを明示。
- ・ 時間指定予約では uie_Reserve_edit(), uie_Reserve_add() の Tracking は「追従しない」が設定されることを明記。
- ・ uie_Reserve_add() における Note の記述が古かった部分を更新。あわせて受信機側でサービス ID/ネットワーク ID チェッ

クを期待することを明示。

- ・ uie_Reserve_getReserveldByEvent()のパラメータの記述を修正。
- ・ uie_Reserve_add()に、ダイレクト録画を実行した場合の各パラメータの既定の値を明記。
- ・ uie_Reserve_cancel()が、視聴予約・録画予約の両方に使われる旨を明示。
- ・ 実行中の予約を削除する場合には、uie_Reserve_delete()を呼ぶ前に uie_Reserve_cancel()を呼ぶ必要があることを明記。
- ・ uie_Reserve_add(),uie_Reserve_Edit()において、ダイレクト録画が選択されてかつ現在時刻より前の開始時間が設定された場合には即録画開始となる旨を明記。

11/16/2009 Revision2.2.8

- ・ uie_Reserve_add(), uie_Reserve_edit()にエラーコード UIE_RESERVE_ERR_FULL を追加。
- ・ POSIX time で指定される時刻は非負値であり 2038 年以降の日付は設定されないことを明記。
- ・ uie_Reserve_findDuplicate()にダブル録画可能な端末の挙動を明記。
- ・ uie_Reserve_findDuplicate()は番組追従等で延長した予約も重複判定対象となることを明記。