



# EZ アプリ (BREW<sup>TM</sup>) サービス仕様書

**EZ アプリ(BREW<sup>TM</sup>)サービス概要**

**Version1.0**



2004 年 5 月 25 日

KDDI 株式会社



# 目次

<b>1. EZ アプリ (BREW™) とは</b>	<b>4</b>
1.1. EZ アプリ (BREW™) とは	4
<b>2. KDDI BREW サービス概要</b>	<b>5</b>
2.1. 実行環境概要	5
2.2. アプリケーションの種類	6
2.3. ネットワーク概要	7
2.4. 課金体系	8
2.5. BREW2.0 日本拡張機能概要	9
2.6. BREW2.1 標準機能概要	12
2.7. BREW2.1 日本拡張機能概要	14
<b>3. サービス提供体系</b>	<b>16</b>
3.1. EZweb 月額連携課金	17
3.2. BREW 回収代行課金	18
3.3. まとめて au 支払い課金	19
3.4. アップグレード(バージョンアップ)	20
<b>4. BREW サービス対応端末仕様</b>	<b>21</b>
4.1. 対応 API	21
4.2. メモリ構成	23
4.3. 画面表示	27
4.4. キー入力	29
4.5. 文字仕様	36
4.6. 音源仕様	37
4.7. バイブレータ	42
4.8. ビープ	42
4.9. バックライト	43
4.10. 対応画像フォーマット	46
4.11. 日時	48
4.12. BREW アプリのダウンロード	48
4.13. BREW アプリの起動	51
4.14. BREW アプリの制御	64
4.15. アプリ連携	71
4.16. BREW アプリからのデータ通信	78
4.17. 位置情報の取得	85
4.18. その他の端末機能	89
4.19. QVGA 対応端末におけるアプリアイコンサイズ	90
<b>5. ネットワーク仕様</b>	<b>91</b>
5.1. BREW ダウンロードネットワーク	91
5.2. BREW アプリ通信ネットワーク (BREW. NET)	91
<b>6. BREW アプリ開発から提供までの流れ</b>	<b>92</b>
6.1. BREW アプリ開発概要	92
6.2. 開発前の準備	93
6.3. BREW アプリ検証	93
6.4. BREW™ ディベロッパー ラボ	93
<b>7. BREW アプリのユーザインターフェースガイドライン</b>	<b>95</b>
7.1. 端末キーの割り当て	95
7.2. ユーザビリティに対する基本ポリシー	96

<b>8. セキュリティについて.....</b>	<b>98</b>
8.1. セキュリティガイドライン.....	98
8.2. BREW 標準のセキュリティ機能.....	99
8.3. KDDI が独自に用意するセキュリティ機能.....	101
<b>9. ユーザ意思確認画面表示ガイドライン.....</b>	<b>102</b>
9.1. 保護すべき端末内情報.....	102
9.2. ユーザ意思確認画面の表示（必須）.....	103
9.3. ユーザ意思確認画面の表示例.....	104
9.4. ユーザ意思確認画面のお好み設定機能の設置（必須）.....	105
9.5. 機能別画面遷移例と注意事項.....	107
9.6. BREW アプリ側で配慮が必要なエラー画面.....	124
<b>10. (付録 1) BREW アプリ全般における文字制限.....</b>	<b>131</b>
<b>11. (付録 2) アプリケーション名における文字制限.....</b>	<b>133</b>
<b>12. (付録 3) SmallFont について.....</b>	<b>135</b>
<b>13. (巻末 1) HTMLViewer サポートタグ一覧.....</b>	<b>136</b>
<b>14. (巻末 2) 端末が表示するエラーメッセージ一覧.....</b>	<b>142</b>
14.1. BREW 側で表示する確認画面とエラー画面.....	142
<b>15. サンプルコード集.....</b>	<b>144</b>
15.1. BREW アプリが使用予定の最大ワークエリアの確保.....	144
<b>16. (巻末 3) 用語集.....</b>	<b>146</b>
<b>17. 更新履歴.....</b>	<b>153</b>

## 1. EZ アプリ (BREW™) とは

### 1.1. EZ アプリ (BREW™) とは

「EZアプリ (BREW™)」(以下、BREWアプリ)とは、KDDI-BREW環境上で動作するBREWアプリケーションサービスです。

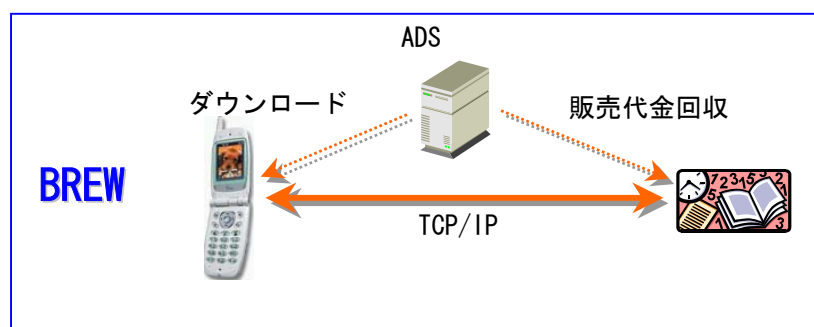
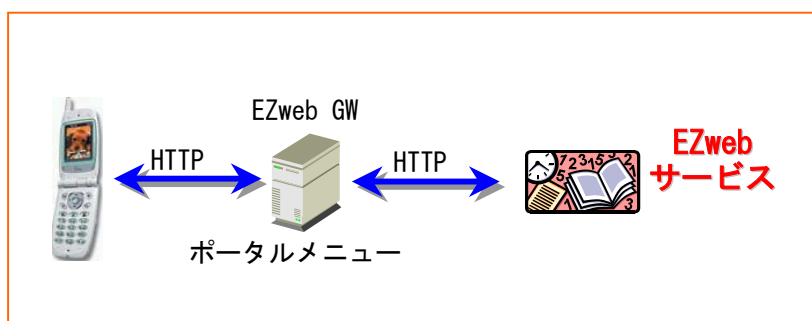
BREWはBinary Runtime Environment for Wirelessの略称で、cdmaOneの開発元である米国Qualcomm社が開発した、端末のアプリケーションプラットフォームです。共通APIを装備したアプリケーションプラットフォームをcdmaOne端末上に搭載しコンテンツプロバイダにそのAPIを公開することで端末上のアプリケーションを自由に開発することが可能となります。C/C++で開発したアプリケーションはBREW対応端末から専用アプリケーションダウンロードサーバ(ADS)へアクセスしBREWアプリカタログよりBREWアプリを選択することでユーザが自由にダウンロード、インストールし実行することが可能となります。

#### ワールドワイドなアプリケーション展開

BREWは世界市場を目指したオープンな共通アプリケーションプラットフォームであることから、端末メーカーの開発サイクルを効率化できると同時に、ワールドワイドな流通アプリケーションを実現することができコンテンツプロバイダの収益改善が期待できます。

#### キャリアネットワークの解放

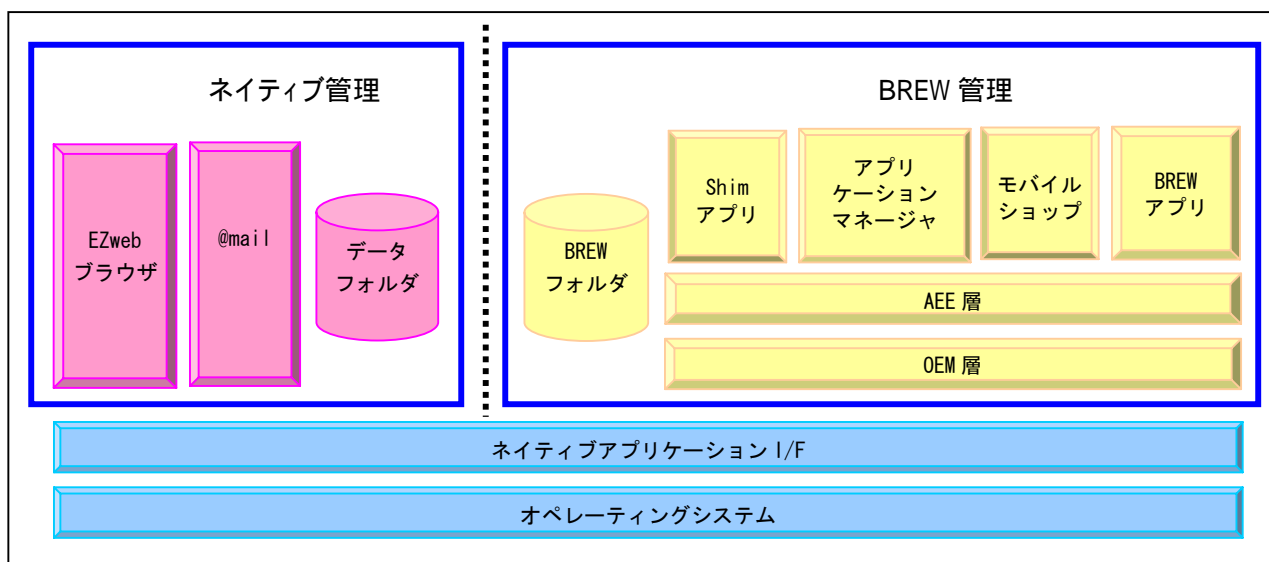
BREWはHTTP以外のTCP/IP上の任意プロトコルに対応させることが可能です。従来からのEZwebやezアプリのようなキャリアのポータルメニュー経由でのHTTPのみといった制約はなく、端末上のアプリケーションとサービス提供会社との間をダイレクトに結ぶことが可能となり、End-to-Endのダイレクトマーケティングサービス提供が可能となります。



## 2. KDDI BREW サービス概要

### 2.1. 実行環境概要

端末内に実装される BREW アプリの実行環境を下図に示します。



#### (1) BREW フォルダ

ADS (Application Download Server) からのダウンロードや、BREW App-Loader によるシリアル転送により端末内部に取り込まれた BREW アプリを保存する領域です。詳細は「[4.2.4. BREW フォルダ \(アプリケーション保存領域\)](#)」を参照して下さい。

#### (2) アプリケーションマネージャ

端末内の BREW アプリを一括して管理 (バージョンアップ、削除等) する BREW アプリです。詳細は、「[2.2.1. アプリケーションマネージャ](#)」を参照して下さい。

#### (3) Shim アプリケーション

BREW アプリからネイティブアドレス帳のデータを取得したり、BREW アプリから@メールアドレスを渡して起動させたり、WAP ブラウザへ URL を渡して起動させたりすることを可能にするアプリケーションです。詳細は「[2.2.2. Shim アプリケーション \(アドレス帳連携/URL 連携/メール連携\)](#)」を参照して下さい。

#### (4) モバイルショップ

ADS への接続、BREW アプリカタログの表示や BREW アプリのダウンロード等一連のダウンロード機能を担うアプリです。

#### (5) AEE (Application Execution Environment) 層

BREW アプリ実行環境です。BREW アプレットを読み込んで実行します。

#### (6) OEM (Original Equipment Manufacture) 層

BREW 実装時に AEE 層と接続する為に端末メーカーが提供するアプリケーションです。

## 2.2. アプリケーションの種類

アプリケーションには、デフォルトで端末に搭載されているアプリケーションと、ADS からダウンロードするアプリケーションとがあります。前者をプリインストールアプリケーション、後者をダウンロードアプリケーションと呼び、前者と後者を総称したものを BREW アプリと呼びます。また、端末メーカーにより出荷状態から端末内部に搭載された WAP ブラウザや@メーラなどの BREW 以外のアプリケーションを Native アプリケーションと呼びます。このうち、コンテンツプロバイダが提供するアプリケーションはダウンロードアプリケーションになります。なお、このダウンロードアプリケーションは運用上 200kB 以下（QVGA 機は 300kB 以下）で作成しなければなりません。

BREW アプリ	プリインストールアプリケーション	アプリケーションマネージャ	端末内の BREW アプリの管理を行うアプリケーションで BREW メニューの表示も担う
		モバイルショップ	BREW アプリのダウンロード機能を担う
		Shim(アドレス帳連携)	BREW アプリが Native アプリケーションとの連携で使用するアプリケーション
		Shim(URL 連携)	
		Shim(メーラ連携)	
	ダウンロードアプリケーション (200kB 以下。ただし QVGA 機は 300kB 以下。)	コンテンツプロバイダにより提供され、ユーザによって ADS よりダウンロードされるアプリケーション	
Native アプリケーション		WAP ブラウザや@メーラなど	

ダウンロードアプリケーション以外はすべて端末出荷時に端末内に存在することになります。

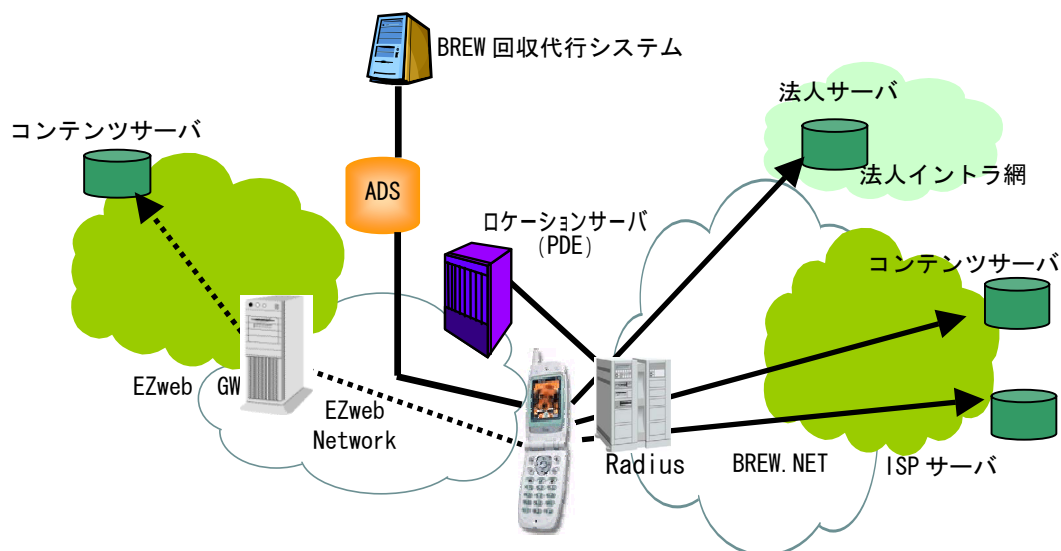
### 2.2.1. アプリケーションマネージャ

端末内の BREW アプリを一括して管理する BREW アプリです。このアプリケーションは端末メーカーにより用意され出荷時にデフォルトで搭載されます。主な機能として、端末内に登録されている BREW アプリの起動（個別表示、アイコン表示、リスト表示より選択）、BREW メニューの表示、一発起動への設定、待受 BREW アプリへの設定、BREW アプリの削除、メモリ使用状況の表示、バージョンアップなどになります。

### 2.2.2. Shim アプリケーション(アドレス帳連携/URL 連携/メーラ連携)

Native アドレス帳からのデータ取得、BREW アプリから@メーラ (Native メーラ) へアドレスを引き渡しながらの起動や BREW アプリから WAP ブラウザ (Native ブラウザ) へ URL を引き渡しながらの起動を可能にするアプリケーションです。これも端末メーカーにより用意され出荷時にデフォルトで搭載されます。

### 2.3. ネットワーク概要 ネットワーク構成イメージ



#### 通信種別

ADS からの BREW アプリのダウンロード通信の他、BREW アプリを利用してインターネット上のコンテンツを閲覧したり、BREW アプリ対応コンテンツをダウンロードする通信があります。

通信種別	内容	通信プロトコル
BREW アプリのダウンロード通信	ADS から BREW アプリを端末にダウンロードする。アプリケーションダウンロード通信については、EZweb 網を経由する。	独自プロトコル (HTTP ベース)
BREW アプリとコンテンツサーバ間通信	端末上の BREW アプリがインターネット上のコンテンツサーバと通信して、BREW アプリとコンテンツサーバが連動したサービスを提供する。BREW アプリの通信については、TCP/IP 上の任意プロトコルに対応し、BREW アプリに PPP 接続情報を持たせる BREW.NET を経由することでインターネットへ接続できる。	TCP/IP 上の任意プロトコル (SSL 含む)

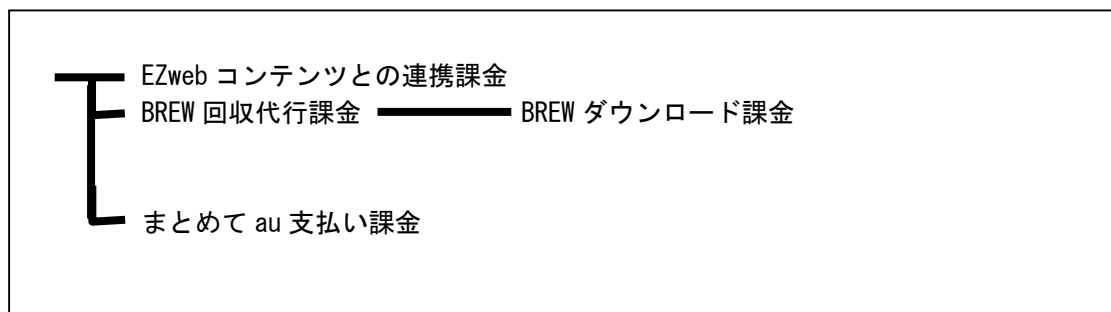
#### 通信速度

cdma2000 1x における通信種別毎の通信速度は下表の通りです。

通信種別		
アプリケーションのダウンロード通信	下り	144kbps
	上り	64kbps
BREW アプリとコンテンツサーバ間通信	下り	144kbps
	上り	64kbps

## 2.4. 課金体系

BREW アプリ提供にあたって、利用可能な課金システム体系は以下の通りであり、これらを複数組み合わせた課金体系も可能です。なお、これら以外の課金体系についても別途検討しています。詳細は、「[3. サービス提供体系](#)」をご参照下さい。





## 2.5. BREW2.0 日本拡張機能概要

BREW2.0 の標準機能に加え、日本市場により合った機能ならびに API を Qualcomm 社と共に検討し BREW2.0 日本拡張機能として以下を追加しています。

### 2.5.1. 日本拡張機能概要

#### ①個人情報/イベントの取得 [IKDDIPHONESYSTEM インターフェース]

BREW2.0 で取得可能な個人情報/イベントの他に、BREW2.0 日本拡張機能として「IKDDIPHONESYSTEM」インターフェースを追加することで、通話履歴/着信履歴、充電中などが取得可能となっています。

拡張 API	機能
履歴情報用 API	
IKDDIPHONESYSTEM_GetNumberOfHistories()	端末内の情報の通話履歴もしくは着信履歴の件数を取得
IKDDIPHONESYSTEM_GetHistory()	端末内の情報の通話履歴、着信履歴を取得
IKDDIPHONESYSTEM_GetChargeInfo()	端末内の情報の直前の通話料金、累計通話料金を取得
IKDDIPHONESYSTEM_GetCallInfo()	端末内の情報の通話時間、累計通話時間を取得
通信環境用 API	
IKDDIPHONESYSTEM_GetIntensity()	現在の電界強度情報を取得
IKDDIPHONESYSTEM_GetPowerSupplyLevel()	現在の電池残量および充電中の情報を取得
その他	
IKDDIPHONESYSTEM_GetMaxVolumeLevel()	着信音量の最大レベルを取得
IKDDIPHONESYSTEM_GetVolumeLevel()	現在設定されている着信音の音量レベルを取得
IKDDIPHONESYSTEM_GetVibrationStatus()	端末のバイブレーションデバイスの現在の設定状態を取得
IKDDIPHONESYSTEM_GetMannerMode()	マナーモード設定の有効・無効の情報を取得
IKDDIPHONESYSTEM_AddRef()	インターフェースオブジェクトの参照カウント増やす
IKDDIPHONESYSTEM_Release()	インターフェースオブジェクトの参照カウント減らす

#### ②WAP ブラウザからの BREW アプリ起動

WEB サイト上で起動したい BREW アプリのリンク用 Item\_ID をタグに記述し、起動時に引き渡ししたいパラメータ列(URL など)を合わせて記述することで EZweb サイト閲覧状態からブラウザを終了しパラメータを引き渡ししながら BREW アプリを起動することが可能になります。ここで、指定したリンク用 Item\_ID に相当する BREW アプリが端末内に存在しない場合や期限が切れてしまって起動できない場合は、そのまま ADS に接続に行き BREW アプリカタログ上の指定した BREW アプリのダウンロード確認画面へと遷移します。詳細は、「[4.13.4. WAP ブラウザからの BREW アプリ起動](#)」を参照して下さい。

#### ③ブラウザとの連携(URL\_to)

BREW アプリから Shim アプリを経由して Native 実装ブラウザである WAP ブラウザに対して URL を引き渡ししながら起動することができます。詳細は、「[4.15.2. ブラウザとの連携\(URL\\_to\)](#)」を参照して下さい。

#### ④メーラとの連携(mail\_to)

BREW アプリから Shim アプリを経由して Native 実装メーラである@メーラに対して E メールアドレス(To/Cc/Bcc)、件名、本文を引き渡ししながら起動することができます。詳細は、「[4.15.3. メーラとの連携\(mail\\_to\)](#)」を参照して下さい。

### ⑤アドレス帳との連携

BREW アプリと Shim アプリを経由した Native アドレス帳の連携によりユーザ操作を契機にアドレス帳内データを取得する機能を実現しています。詳細は、「[4. 15. 4. アドレス帳との連携](#)」を参照して下さい。

### ⑥アプリキー(対応端末のみ)

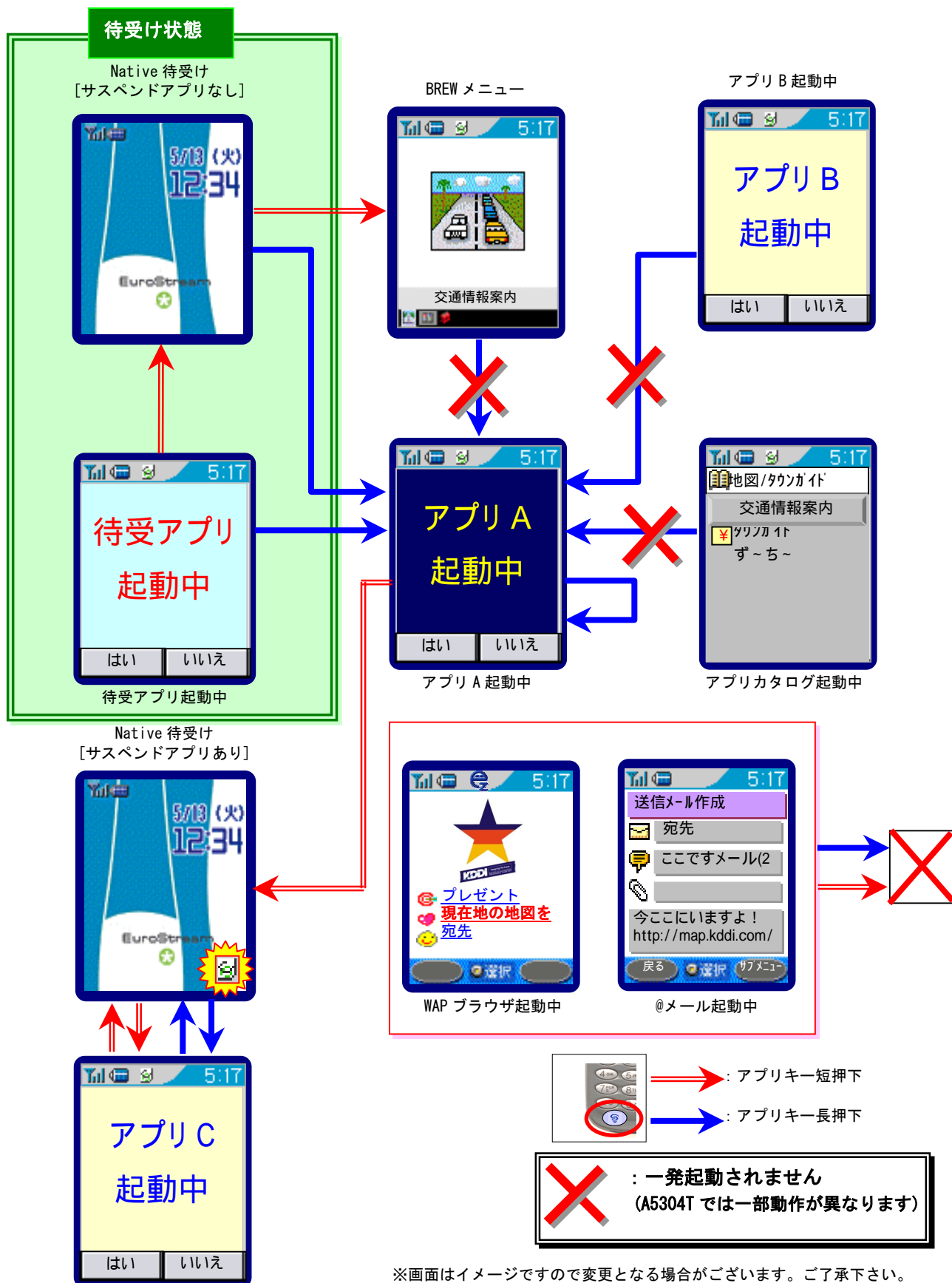
「アプリキー」(「HotKey」、「BREW 専用キー」と呼ぶ場合もあります)を搭載する端末では以下の機能が具備されます。なお、アプリケーションマネージャ、モバイルショップ起動中のアプリキー押下は全て無効になります。

#### (1) アプリキー短押下

- (i) Native 待受け状態(サスペンドアプリなし)  
アプリケーションマネージャが起動され BREW メニューが表示されます。
- (ii) Native 待受け状態(サスペンドアプリあり)  
サスペンドされていた BREW アプリをレジュームします。  
※アプリキーを長押下しても同動作です。
- (iii) 待受けアプリ起動中  
Native 待受け状態に戻ります。

#### (2) アプリキー長押下

- (i) 待受け状態[Native 待受け(サスペンドアプリなし)・待受けアプリ起動中]  
「BREW 設定」内の「一発起動設定」に登録された BREW アプリが起動します。
- (ii) BREW アプリ・BREW メニュー・BREW カタログ・Native アプリ起動中  
「一発起動設定」されていてもアプリは起動されません。また、「一発起動設定」された BREW アプリが起動している場合は、そのアプリにレジューム要求が来るだけとなります。  
※次項イメージ図参照。



※画面はイメージですので変更となる場合がございます。ご了承下さい。

## 2. 6. BREW2.1 標準機能概要

BREW2.0 から BREW2.1 へバージョンアップし、新たな機能を追加します。

### ①カメラ制御機能 [ICamera インターフェース]

BREW アプリにてカメラ固有の機能情報の取得・設定が可能となることにより、プレビュー画像の操作が可能となります。(カメラ撮影(JPEG 保存)、プレビュー(BMP)が可能)

また、撮影時を含め全ての UI を、BREW アプリ側で作成が可能です。

ICamera 連携で、アプリから LCD へ表示出来るサイズ及び表示位置は、機種依存となります。

インターフェースの詳細は、「BREW2.1 新インターフェース (Qualcomm SDK 付属)」を参照して下さい。

※プレビューモード移行時及びシャッター音は Native 側で自動鳴音されます。また、プレビュー中は LED が自動点灯します。

### <例> フレーム付きでViewしながらの画像撮影



### <注意>

アプリの作成方法によっては、プレビューモードで表示される BMP 画像を無音で保存する事が可能です。しかし、プレビューモードで取得した画像を取得する場合は、アプリにて必ず鳴音する仕様にして下さい。

ICamera インターフェースの動作シーケンスには、以下の 2 種類があります。

- ①Qualcomm 標準の ICamera インターフェース動作シーケンス準拠
- ②Qualcomm 標準の ICamera インターフェース動作シーケンスの一部カスタマイズ動作シーケンス

動作シーケンスは移動機毎に異なります。

上記②は、カメラ処理、LCD 表示処理を Coprocessor が担っていることに起因します。

ICamera インターフェースによるカメラ連携を行うアプリは上記①②を考慮する必要があります。

Qualcomm 標準シーケンスでは、ICamera\_GetFrame にて撮影画像データを取得し、IDISPLAY\_BitBlt() で LCD 描画することによって、プレビューを実現することになっています。

しかし、Coprocessor 搭載機種ではプレビュー動作を Coprocessor で行ってしまう為、BREW アプリでの上記動作は不要になります。

### ②簡易位置情報取得機能 [IPOSDET\_GetSectorInfo]

BREW アプリで簡易位置情報機能を使用する事により、情報が早く取得可能となります。

EZ ナビ未対応機種においても、基地局から送られてくる位置情報を取得が可能となります。詳細は、「[4.17. 位置情報の取得](#)」を参照して下さい。

### ③その他

BREW2.1 では、上記以外にも次のインターフェース/API が追加されます。詳細は、「BREW2.1 新インターフェース (Qualcomm SDK 付属)」を参照して下さい。

#### 新インターフェース

##### (1) ILogger

ILoggerインターフェースは、AEE レイヤーの一般的なロギングインターフェースを提供します。

ILoggerインターフェースは、ログ情報をファイルへ出力する方法と、エミュレータの出力ウインドウへの表示との2通りの方法があり、これによりデバッグが容易になります。

##### (2) IRecordStore

簡単なキーと値のペアによって作成されたレコードストアにアクセスするためのインターフェースです。

#### 新関数追加 (既存インターフェース)

##### (1) IBitmap

Procパラメータで渡された長方形を含めるように、ビットマップのダーティ長方形(書き換え領域)を拡張するAPIです。 [IBITMAP\_Invalidate]

##### (2) IShell

指定されたClassIDの所有モジュールに関連付けられた32ビットの一意の識別子を返すAPI [ISHELL\_GetClassitemID]と、指定されたアプレットに対して非同期イベントをポストするAPI [ISHELL\_PostEventEx]です。

##### (3) IPosDet

デバイスの水平面上の向きを返す API です。 [IPOSDET\_GetOrientation]

## 2.7. BREW2.1 日本拡張機能概要

BREW2.1 の標準機能に加え、日本市場により合った機能ならびに API を Qualcomm 社と共に検討し、BREW2.1 日本拡張機能として以下の機能を追加しています。

### ①Native データフォルダ連携 [ IFileCp インターフェース ]

BREW アプリからのデータフォルダ連携、または NativeUI へ遷移することにより、BREW アプリ内データのデータフォルダへの保存や、データフォルダ内データの読み込み機能を実現しています。詳細は、「[4.15.5. データフォルダとの連携](#)」を参照して下さい。

拡張 API	機能
<b>IFileCp</b>	
IFileCp_AddRef()	インターフェースオブジェクトの参照カウントを増やす。
IFileCp_EnumInit()	インターフェースオブジェクトを、データフォルダ上で初期化する。
IFileCp_EnumNext()	データフォルダ内の、次のファイルの情報を取得する。
IFileCp_Get()	データフォルダUI に遷移し、ユーザ操作により選択されたデータを読み込む。
IFileCp_GetEx()	データフォルダから、IFileCp_FILEINFO で指定されたファイルを読み込む。
IFileCp_GetFreeSpace()	データフォルダで現在、利用可能な容量をバイト数で取得する。
IFileCp_Put()	データフォルダにファイルを書き込む。
IFileCp_QueryInterface()	指定されたオブジェクトから、別の API コントラクトのためにオブジェクト情報を取得する。
IFileCp_Release()	インターフェースオブジェクトの参照カウントを減らす。

### ②Native アドレス帳連携強化 [ IADDR インターフェース ]

Shim 連携によるアドレス帳連携では項目 READ のみでしたが、IAddr インターフェースにより BREW アプリから Native アドレス帳の Read/Write を行う事を実現しました。また、アドレス帳の U/I を BREW アプリで作成可能です。詳細は、「[4.15.4. アドレス帳との連携](#)」を参照して下さい。

拡張 API	機能
<b>IADDR</b>	
IADDR_AddRef()	インターフェースオブジェクトの参照カウンタを増やす。
IADDR_Get()	アドレス帳から指定したメモリ番号のレコードを取得する。
IADDR_GetInfo()	アドレス帳情報を取得する。
IADDR_GetMemNoStatus()	メモリ番号の状態を取得する。
IADDR_Put()	指定したレコードをアドレス帳に書き込む。
IADDR_Release()	インターフェースオブジェクトの参照カウンタを減らす。

### ③電子コンパス機能 [ IOrDet インターフェース ]

BREW アプリにて、Native で持つ電子コンパスから方位角情報の取得が可能です。詳細は、「[4.17. 位置情報の取得](#)」を参照して下さい。

拡張 API	機能
<b>IOrDet</b>	
IORDET_AddRef()	インターフェースオブジェクトの参照カウントを増やす。
IORDET_GetDirectionXYZ()	3 軸の磁界の強さを取得する。
IORDET_GetInclination()	携帯電話の傾きを x と y 軸方向を取得する。
IORDET_Release()	インターフェースオブジェクトの参照カウントを減らす。

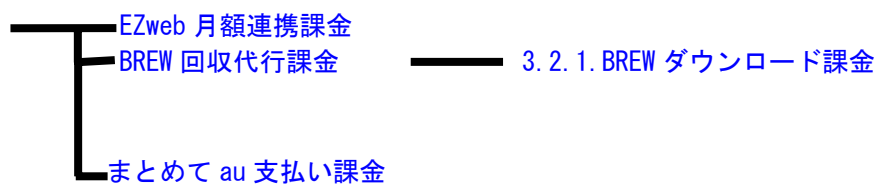
### ④SJIS Unicode 変換機能 [ IUnicodeSjis インターフェース ]

BREW アプリにて、キャラクターセット (Unicode (USC2) と SJIS) の双方向変換が可能です。

拡張 API	機能
<b>IUnicodeSjis</b>	
IUNICODESJIS_AddRef()	インターフェースオブジェクトの参照カウントを増やす。
IUNICODESJIS_Release()	インターフェースオブジェクトの参照カウントを減らす。
IUNICODESJIS_QueryInterface()	指定されたオブジェクトから、別の API コントラクトのためにオブジェクト情報を取得する。
IUNICODESJIS_Unicode16toSjis()	指定された Unicode (USC2) スtring を SJIS フォーマットに変換する。
IUNICODESJIS_SjistoUnicode16()	指定された SJIS スtring を Unicode (USC2) フォーマットに変換する。


### 3. サービス提供体系

BREW サービスを提供するにあたって一般的となっている利用制限のない有料アプリの他に以下のサービス体系を提供することが可能です。また、これらを複数組み合わせることで多彩なサービス体系を提供することが可能です。これら以外にも様々な課金体系を検討していきます。なお、これらとは別にソフトウェアのバージョンアップを実現するアップグレード機能も提供することが可能です。





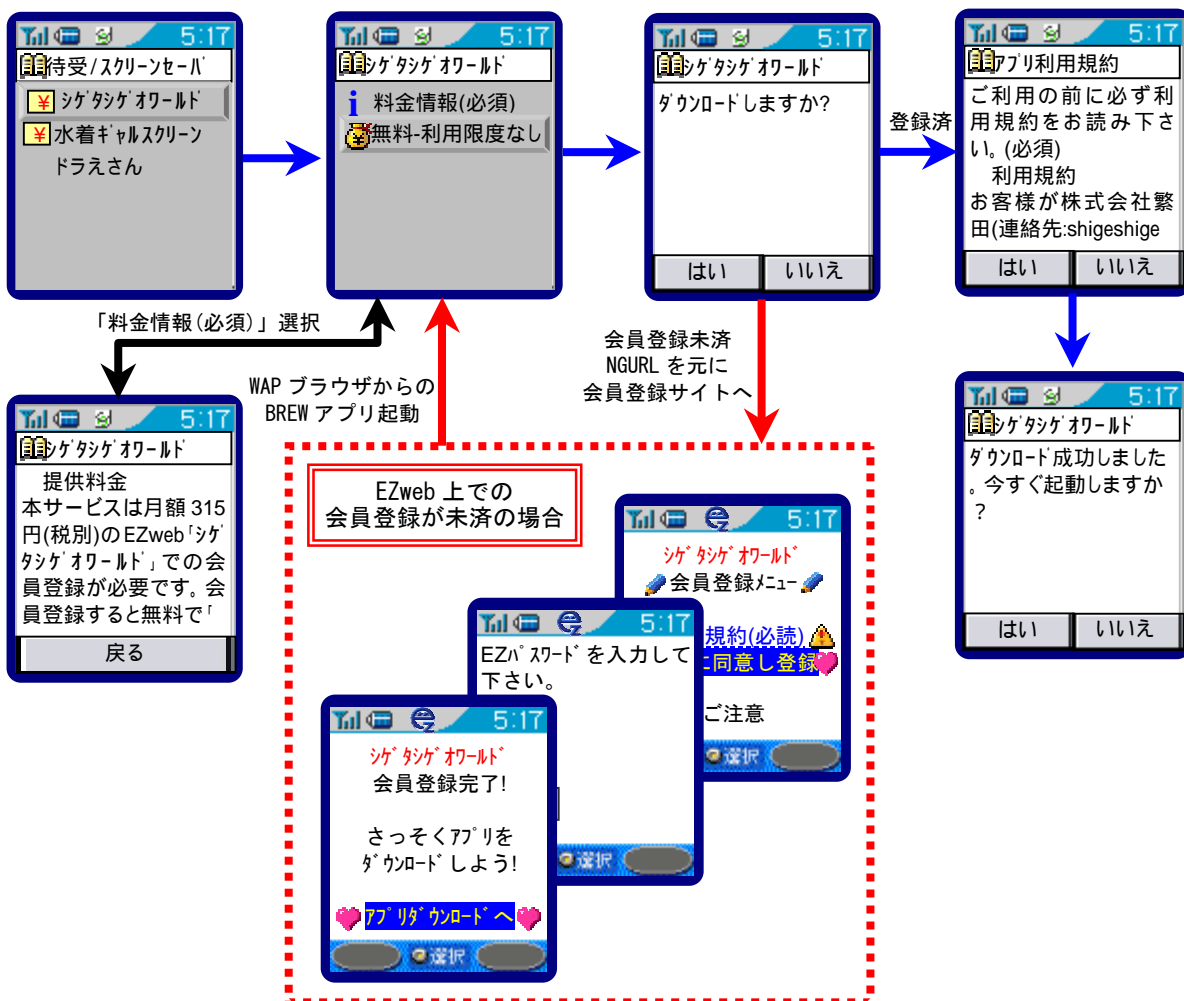
### 3.1. EZweb 月額連携課金

EZweb の回収代行サービスにおける月額固定課金を利用し、サイトの会員のみ BREW アプリがダウンロード可能な画面へ進むことができ、ADS からダウンロードを可能とするモデルです。BREW アプリカタログ上では””アイコンによりそのステータスが確認できます。この時、EZweb 網から利用できる KCGS の会員管理システムが BREW 網からは利用できない為、BREW 網からのアクセスに対してコンテンツプロバイダが独自にデータベースを構築する必要があります。

入退会は WAP ブラウザに切り替えて行い、コンテンツプロバイダの管理するデータベースと整合を取る必要があります。下記のフローのように EZweb での会員登録が済んでいない場合はダウンロードには移行せず一旦 NGURL を元に EZweb 上の会員登録サイトに接続し、会員登録終了後、BREW アプリカタログ上でのダウンロードが行われます。

例：ABC コンテンツ/月額 300 円の加入者のみゲーム BREW アプリがダウンロード可能。

(1)EZweb 月額連携課金用サイト作成上の注意  
検討中です。



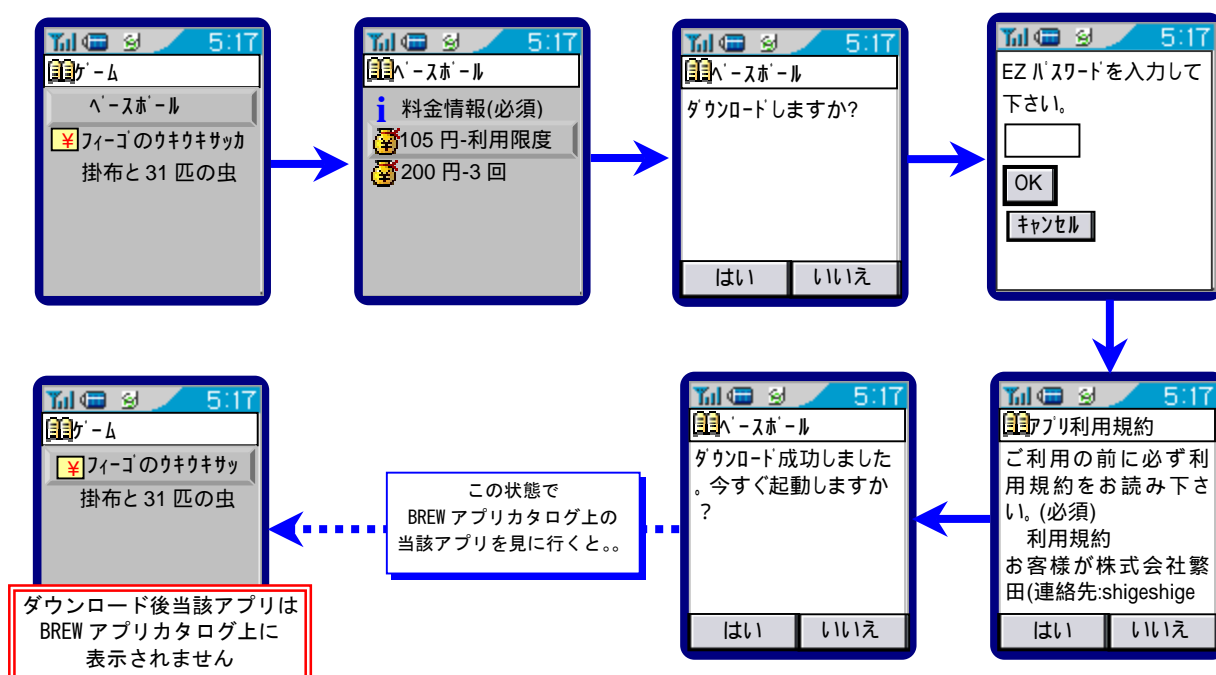
### 3.2. BREW 回収代行課金

BREW 回収代行システムを利用した課金システムにより、以下の課金が可能です。これらは「BREW アプリ新規登録シート」上にて申し込みます。

#### 3.2.1. BREW ダウンロード課金

”利用制限なし”のごく一般的なサービスモデルで BREW アプリをダウンロード後、端末から ADS に送信されるダウンロード完了 ACK 毎に課金されます。なお、ダウンロードされた BREW アプリはその後 BREW フォルダ上から消去するまで利用制限付きのライセンスも含め BREW アプリカタログ上には一切表示されなくなります。

例：1 ダウンロードにつき 300 円



### 3.3. まとめて au 支払い課金

まとめてau支払い課金は、エンドユーザとサービス提供者との個別契約における決済を、KDDI がサービス提供者に代わって回収し、サービス提供者に支払うサービスです。

### 3.4. アップグレード(バージョンアップ)

既にダウンロード済みの BREW アプリを新たにアップグレードすることで、BREW アプリ内の必要なファイルのみを更新しアップグレード前の情報を引継ぎつつサービスすることが可能なモデルです。

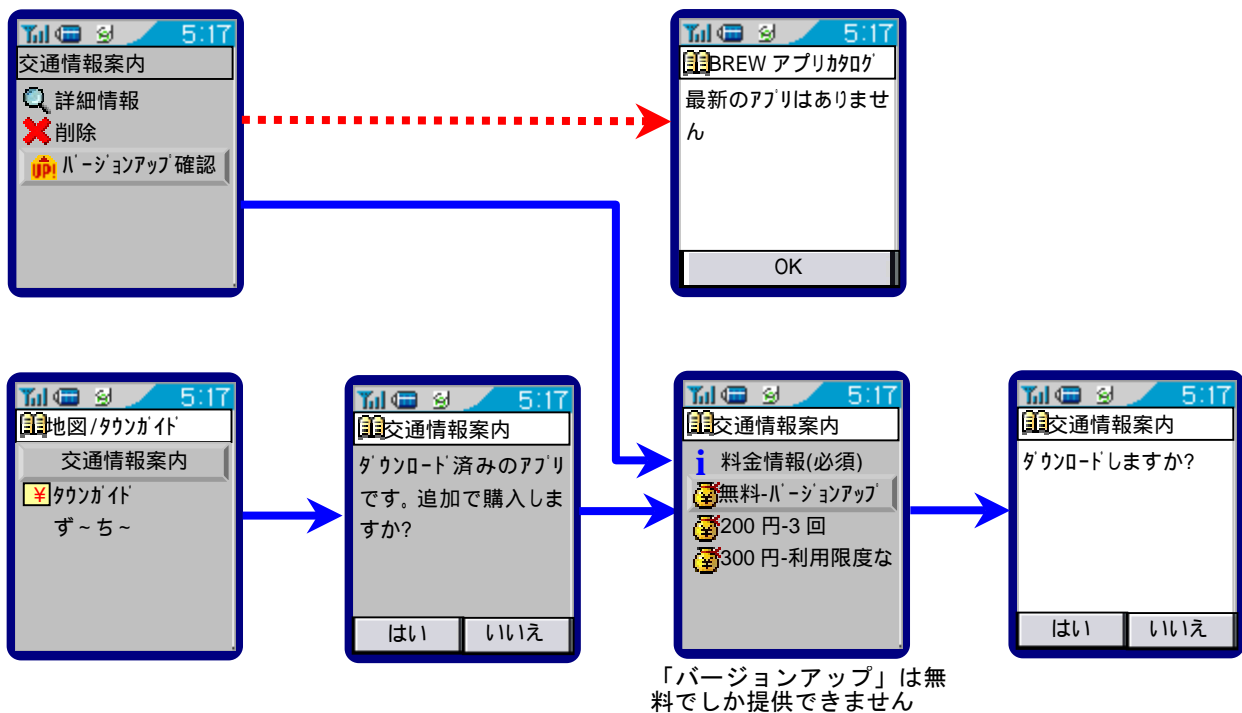
#### (1) 対象ユーザ毎の動作

##### ①既に BREW アプリをダウンロードしているユーザ

BREW アプリを既にダウンロードしているユーザは、「BREW 設定」内の「アプリマネージャー」で当該 BREW アプリを選択し、「バージョンアップ確認」で、アップグレード版の BREW アプリをダウンロードします。端末内にある BREW アプリと BREW アプリカタログ上にある BREW アプリが同一のバージョンである場合は BREW アプリカタログ上には表示されず、バージョンが異なる場合は BREW アプリカタログ上に表示されます。その表示された BREW アプリカタログ上で当該アプリを選択することでもアップグレード版の BREW アプリをダウンロードすることが可能です。なお、この時アップグレード料金単体として徴収することは出来ません。

##### ②未だ BREW アプリをダウンロードしていないユーザ

既存 BREW アプリをまだダウンロードしていないユーザは、BREW アプリカタログ上で当該アプリを選択することでアップグレード版の BREW アプリが新規に提供されます。この時、既存 BREW アプリとアップグレード版 BREW アプリとを並行して提供することは出来ません。なお、料金体系は新規アプリを提供する場合と同じ形となります。



→ : VersionUp がある場合

→ : VersionUp がない場合

※アップグレードがないアプリはアプリカタログ上には表示されません

## 4. BREW サービス対応端末仕様

### 4.1. 対応 API

KDDI より発売される BREW 対応端末は下記表に示す API が標準的に実装されます。

(1) BREW2.0

・標準仕様

番号	インターフェース名	番号	インターフェース名
1	IApplet	31	IMenuCtl
2	IStream	32	IModule
3	IBase	33	INetMgr
4	IBitmap	34	INotifier
5	IClipboard	35	IPeek
6	ICipher	36	IPosDet
7	IControl	37	IRAMCACHE
8	IDatabase	38	IQuery
9	IDateCtl	39	IRSA
10	IDBMgr	40	IShell
11	IDBRecord	41	ISocket
12	IDialog	42	ISound
13	IDIB	43	ISoundPlayer
14	IDisplay	44	ISource
15	IDNS	45	ISourceUtil
16	IFile	46	ISprite
17	IFileMgr	47	ISSL
18	IFont	48	IStatic
19	IGetLine	49	ITAPI
20	IGraphics	50	ITextCtl
21	IHash	51	ITimeCtl
22	IHASHCTX	52	ITransform
23	IHeap	53	IUnzipAStream
24	IHtmlViewer	54	IWeb
25	IImage	55	IWebOpts
26	IImageCtl	56	IWebResp
27	ILicense	57	IWebUtil
28	IMedia	58	IViewer
29	IMediaQCP	59	IX509Chain
30	IMemAStream		

・日本拡張

60	IKDDIPHONESYSTEM	64	
----	------------------	----	--

## (2) BREW2.1 追加インターフェース

## ・標準仕様

62	ICamera	64	IRecordStore
63	ILogger		

## ・日本拡張

65	IADDR	67	IOrDet
66	IFileCp	68	IUnicodeSjis

## 4.1.1. 各 API における注意事項

## (1) ITAPI\_GetStatus() における“szMobileID”

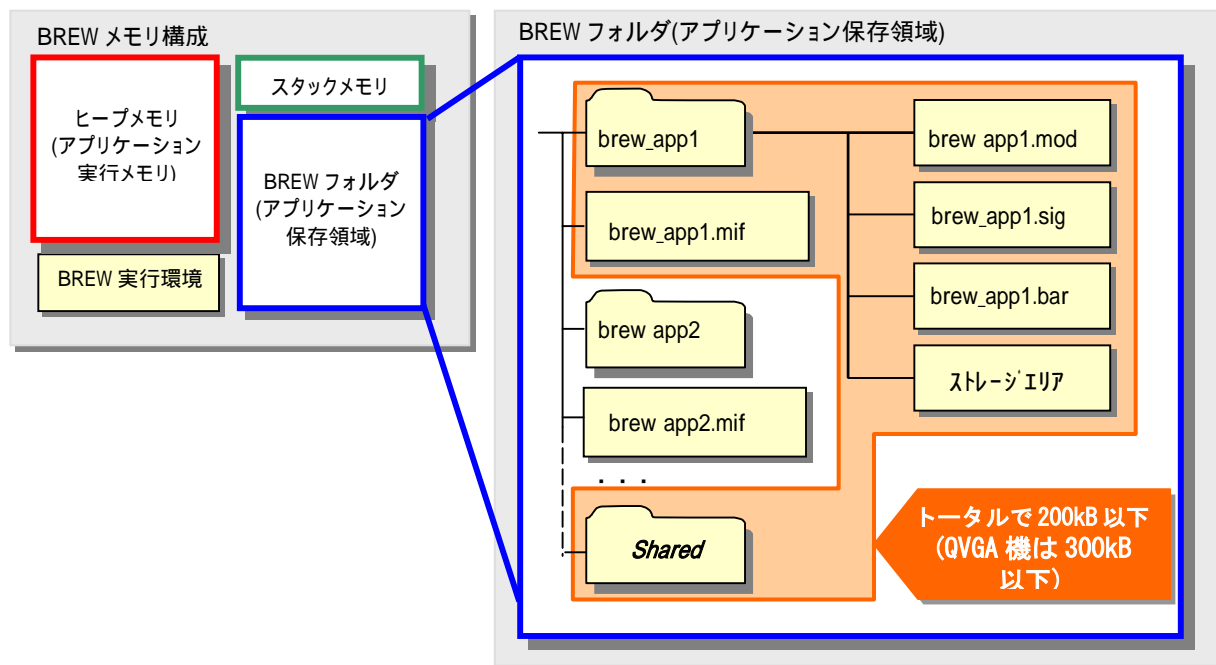
当該指定をした場合の取得できる値は、電話番号となります。

## (2) IHtmlViewer() におけるタグの深さ制限

IHtmlViewer() において、階層化されたタグの深さが 15 ネストまでに制限されています。

#### 4.2. メモリ構成

BREW 実行環境ならびに BREW アプリが使用するメモリの構成は次図の通りとなります。



#### 4.2.1 アプリケーションサイズ

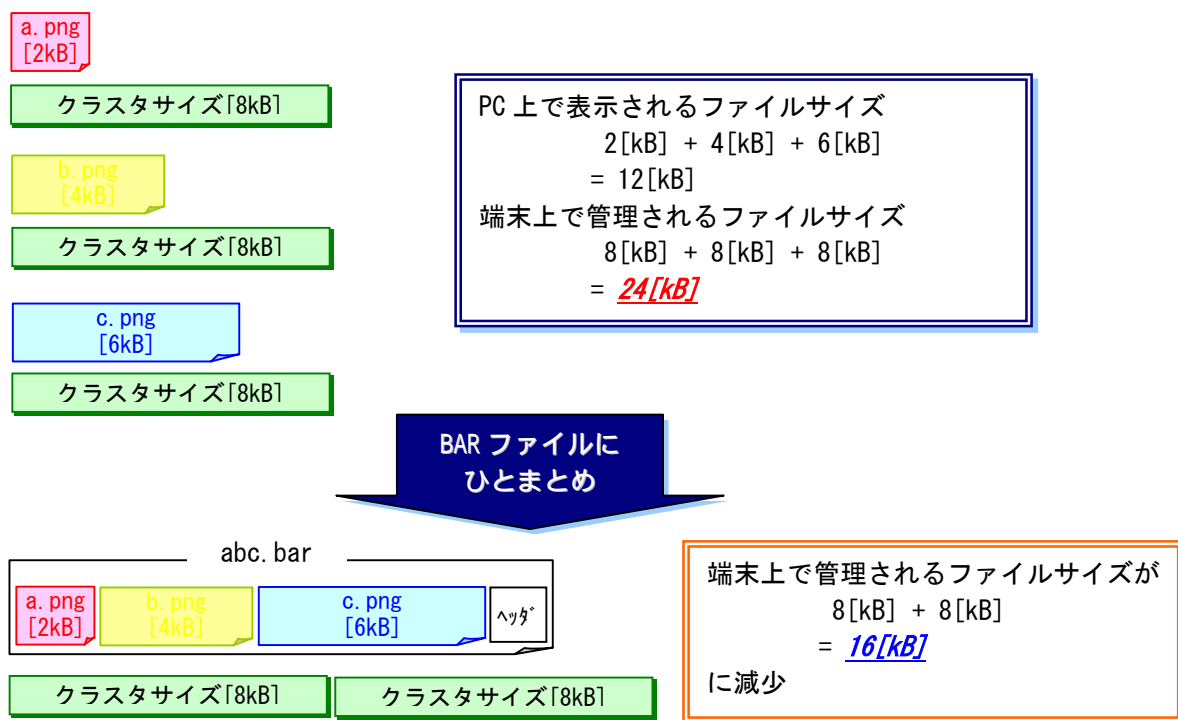
運用規則として BREW アプリは、ストレージエリア含めたモジュールディレクトリ内の全てのファイル (mod, sig, bar 等) と、BREW アプリによって共有ディレクトリ (Shared Directory) に作成されるファイルの合計で” 200kB 以下 (QVGA 機は 300kB 以下) ” の BREW アプリを作成しなければなりません。

##### (1) クラスタへの配慮

この時、BREW フォルダに使用するファイルシステムにより数 Byte の小さいファイルでもファイルシステム管理上ではクラスタサイズ分のメモリが必要で、端末の BREW フォルダへ格納した際にサイズが大きくなってしまい PC 上の合計サイズと比較して非常に大きなサイズになってしまう場合があります。リソースエディタを使用して画像ファイルやテキストファイル等を BAR ファイルに一まとめにしたり、HTML ファイルに区切り文字を入れて一まとめに連結し、読み出すときは IFILE\_SEEK をする等ファイル数を少なくする等の工夫により” 200kB 以下 (QVGA 機は 300kB 以下) ” に収まるようにして下さい。なお、IHTMLViewer 上では BAR ファイルは使用できません。

端末への転送後のアプリサイズは端末内の「BREW 設定」→「システム情報」にて確認することができます。転送前に「システム情報」で使用容量を確認し、転送後に再度「システム情報」を確認して転送前後の使用容量の差分により端末上でのアプリサイズを確認して下さい。

A5304T の場合 (クラスタサイズ” 8[kB]” )



##### (2) ” 0” Byte ファイル

“0” Byte のファイルがあるとクラスタサイズ分の余計なメモリを消費してしまうだけでなく UAM 登録後ダウンロードできなくなるという症状がありますので、作成しないようにご注意ください。

##### (3) データフォルダからのファイル取得などによるサイズの拡大

データフォルダからファイルを取得やカメラ連携による画像の取りこみ等で、起動中に一時的にサイズが大きくなってしまう場合は、以下の制限を守ってください。



- ① ファイルの取得等で一時的にアプリケーションのファイルサイズが拡大しても、アプリ起動前および終了後のサイズは規定サイズ以内(200kB 以下、QVGA 機は 300kB 以下)としてください。
- ② ファイルなどを取得して BREW フォルダに書き込む際に、空き領域不足のため失敗した場合はアプリよりエラーメッセージを表示してください。

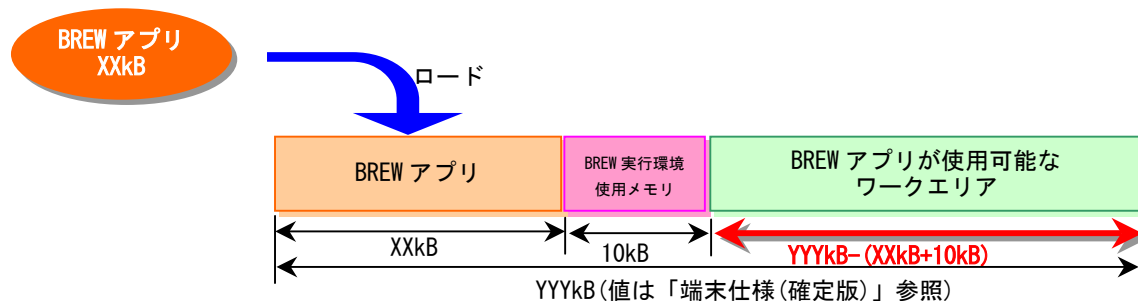
※一部機種にて、空き容量不足でもアプリにエラーが返されずに 0 バイトファイルが作成される端末がありますが、その場合でもアプリでエラーのポップアップを表示するようにしてください。

#### 4.2.2. ヒープメモリ(アプリケーション実行メモリ)

BREW アプリを起動する際には、BREW フォルダ(アプリケーション保存領域)に保存された BREW アプリをヒープメモリ(アプリケーション実行メモリ)にロードし実行します。

他の BREW アプリも共有しながら使用する領域となる為、BREW アプリが使用するヒープを必要以上に使用すると、既に起動されサスペンド中にある他の BREW アプリが終了される(退避起動)等の状態が起きてしまい BREW サービスの提供品質を著しく落としてしまう可能性があります。必要以上のヒープを使用しないよう細心の注意を払うようにして下さい。なお、BREW 実行環境からも約 10kB 程使用します。また、「4.13.8. BREW アプリ起動における留意点」にある BREW アプリ起動時のヒープ残容量のチェックや以下の「(1)「ISHELL\_LoadXX」系 API 使用時のオブジェクト生成チェック」を必ず入れるようにして下さい。起動時には MAX ヒープサイズ以上の空き領域があるかを確認してください。

BREW アプリ終了時にはメモリリークを起こさないよう取得したヒープメモリ(MALLOC 等)は必ず解放するようにし、取得したリソース(IShell\_LoadXX 等)も必ず解放(IShell\_FreeResData 等)しなければなりません。この時、継続して記憶しておく必要のあるデータがある場合は終了と同時にファイルにて保存する等の処理を行って下さい。



## (1) "ISHELL\_LoadXX"系 API 使用時のオブジェクト生成チェック

空きヒープ容量が不足している状況で下記"ISHELL\_LoadXX"系 API を使用した場合、失敗した旨を表す為"NULL"が返る事が正しい動作となりますが、その他の異なる値が返るようになってしまっています。これにより、以降の動作が正しく行われなくなります。

"IIMAGE\_GetInfo()"にて"AEEImageInfo"を得て、そのメンバ変数 AEEImageInfo.cx, AEEImageInfo.cy の中身が"0"か否かにより正しく生成されたかをチェックし、"ISOUNDPLAYER\_GetInfo()"では"AEESoundPlayerInfo"を得て、そのメンバ変数 AEESoundPlayerInfo.dwSize の中身が"0"か否かにより正しく生成されたかをチェックするようにして下さい。

この時、void IIMAGE\_GetInfo(IImage \* pImage, AEEImageInfo\* pi) の第一引数 pImage の値として、"NULL"を代入してしまうと端末はリセットを起こしてしまいます。ISHELL\_LoadResImage()に対して、存在しないリソース・ファイル名や存在しない画像 ID を引数として指定すると、返り値が"NULL"となりますので、直接 ISHELL\_LoadResImage()の返り値を IIMAGE\_GetInfo()の引数に使用しないようにして下さい。なお、IIMAGE を例としてますが、ISOUNDPLAYER も同様に注意が必要です。

- ISHELL\_LoadImage()
- ISHELL\_LoadResImage()
- ISHELL\_LoadBusyImage()
- ISHELL\_LoadResObject() [HTYPE\_VIEWER/HTYPE\_SOUND 利用時]
- ISHELL\_LoadSound()
- ISHELL\_LoadResSound()

#### 4.2.3. スタックメモリ (FILO メモリ)

BREW アプリが実行中にスタック領域として使用します。一時的なデータ格納用領域となることから BREW アプリを高速に動作させる為にもスタックメモリサイズ以内で使用するよう注意して下さい。大きな配列をスタック変数ではなくヒープから確保したり、再帰処理などの関数を作成した場合スタックメモリを極端に消費する恐れがある為深く同一関数が呼ばれないよう極力使用を避けたり注意して下さい。

#### 4.2.4. BREW フォルダ (アプリケーション保存領域)

ADS からのダウンロードや、BREW AppLoader によるシリアル転送により端末内部に取り込まれた BREW アプリを保存する場所です。EFS と呼ばれる場合もあります。ルートディレクトリには BREW アプリの MIF ファイルを格納し、ルートディレクトリ下にモジュールディレクトリとして BREW アプリ関連ファイル (mod, sig, bar 等) が格納され、これらのディレクトリは BREW アプリ単位に作成されます。

なお、モジュールディレクトリ配下にサブディレクトリを作成した場合、UAM に登録する際に、これらのツリー構造は引き継がれずに削除されてしまいます。また、この場合には、検証不合格となりますのでご注意ください。ただし、DL 後に BREW アプリが自フォルダ内にサブフォルダを作成することは可能です。

#### 4.2.5. BREW アプリからの各種データアクセス

ルートディレクトリ下のモジュールディレクトリに置かれる BREW アプリ関連ファイル (mod, sig, bar 等) 及びストレージエリアは同一のディレクトリであり自アプリ配下であることから、これらファイルへのアクセス (Read/Write) を自由に行うことができます。ただし、異なるアプリ (brew app2) 配下のフォルダにアクセスすることはできません。

#### 4.2.6. 共有ディレクトリ (Shared Directory) へのアクセス

共有ディレクトリは複数のアプリから任意にアクセス (Read/Write) 出来る共有メモリ空間です。よって、同フォルダ内に個人情報に関連するファイルや、その他セキュリティに関わるファイルを格納すると、他アプリへの情報漏洩の恐れがありますので、共有ディレクトリをアプリから利用する際には十分な配慮が必要です。

#### 4.2.7. パスとファイル名

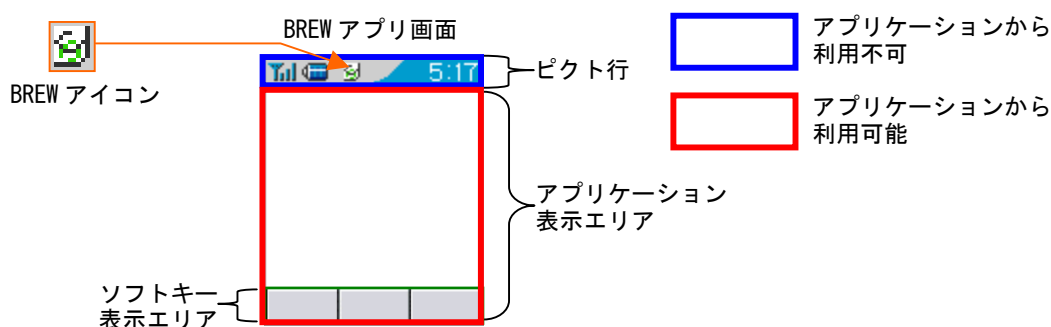
端末上でパスやファイル名に使用する文字は全て小文字でなければなりません。また使用できるも端末により異なります。

#### 4.2.8. 最大同時ファイルオープン数

BREW 上で同時に展開できるファイル数の上限値がデバイス依存となることから端末毎に存在します。なお、MIF エディタ上で設定可能な「EFS 制約 最大ファイル数」はモジュールが作成可能なファイルの最大数を指定するもので意味合いが異なります。

### 4.3. 画面表示

BREW アプリの実行画面 (レイアウト) は以下ようになります。



#### (1) ピクト行表示エリア

電池残量や電波状況などネイティブにより端末の状態を表示する領域であり、BREW アプリから制御することはできません。なお、BREW アプリによるネットワークへの接続中は通信中である旨をユーザに知らせるため、ネイティブにより BREW アイコンを点滅表示させます。点滅は最初の Socket open 制御を契機にピクトの点滅を開始し、接続中の Socket 数が「0」になった時点でピクトの点滅を終了します。通信中は、BREW アプリからの位置情報測位も含まれます。

#### (2) アプリケーション表示エリア

BREW アプリが自由に使用できる表示エリアです。

#### (3) ソフトキー表示エリア

ソフトキー表示エリアは ITextCtl インターフェースと IMenuCtl インターフェースとによりソフトキーを描画した場合に使われる領域です。BREW アプリからの表示コントロールが可能であり、BREW メニューのソフトキーは表示色等 BREW アプリから指定できず固定色にて表示され、フォントは Normal Font、文字数は最大 6 バイトになります。これに対しソフトキーを使用しない BREW アプリではソフトキー表示エリアもアプリケーション表示エリアとして使用することができます。また、ソフトキーが必要な時点でアプリケーションにより表示することも可能です。詳細は「4. 4. 4.5. ソフトキー」を参照して下さい。

#### (4) その他の表示

##### ① スクロール表示

ピクト行にスクロールナビゲーション(▲▼)は表示しません。スクロールバーのみの表示となります。

##### ② メニュー文字列の横スクロール表示

メニュー文字列の横スクロール表示は Native 機能での表示と同様の動作を行います。

#### 4.4. キー入力

##### 4.4.1 基本仕様

###### (1) キープレスイベント

BREW にて扱われるキーはそれぞれ以下のように BREW アプリに通知されます。「決定/確定」動作は” EVT\_KEY” により判断されます。なお、Hold(EVT\_KEY\_HELD)にて、押下中の状態を通知する仕様はありません。よって PRESS と RELEASE にてキー押し下げ、キー長押し、キー離しを判定する必要があります。

動作	イベント
PRESS	EVT_KEY_PRESS
確定	EVT_KEY
RELEASE	EVT_KEY_RELEASE

※ BREW2.1 対応機種では、50ms 程度の短いタイマ間隔でコールバックを繰り返すとキープレスイベントが破棄される場合があります。  
タイマ間隔を長くするなどのアプリ側でのチューニングが必要となりますのでご注意ください。

#### 4.4.2. 対応キーコード

BREW アプリからはキーイベントと共に下記のキーコードの取得が可能となっています。なお、長押しによるキーイベントのリピータ動作におけるリピートの時間間隔は機種依存となります。実機上にて確認して下さい。

また、ソフトキーは「ソフトキー1」、「ソフトキー2」までの2つが”必須”で、「ソフトキー3」は”オプション”となっています。

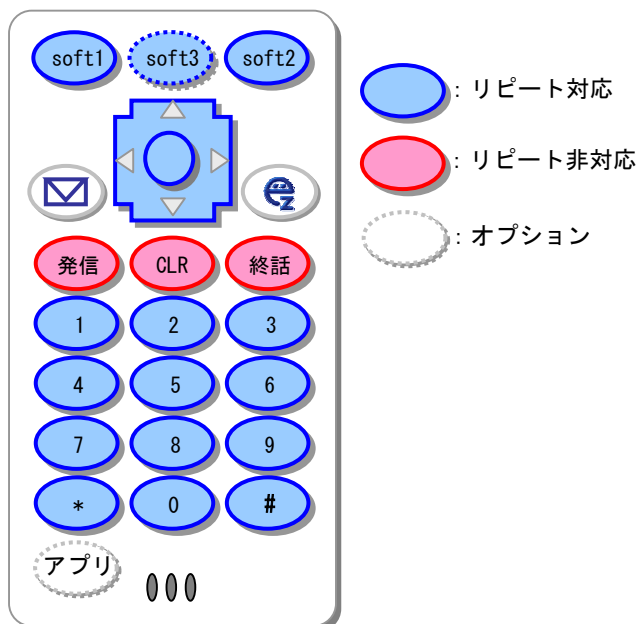
対応キー	キーコード	リピート
0	AVK_0	○
1	AVK_1	○
2	AVK_2	○
3	AVK_3	○
4	AVK_4	○
5	AVK_5	○
6	AVK_6	○
7	AVK_7	○
8	AVK_8	○
9	AVK_9	○
*	AVK_STAR	○
#	AVK_POUND	○
確定	AVK_SELECT	○
クリア	AVK_CLEAR	×
通話	AVK_SEND	×
終話	AVK_END※1	×

対応キー	キーコード	リピート
上	AVK_UP	○
下	AVK_DOWN	○
右	AVK_RIGHT	○
左	AVK_LEFT	○
ソフトキー1(左)	AVK_SOFT1	○
ソフトキー2(右)	AVK_SOFT2	○
ソフトキー3(中)※2	AVK_SOFT3	○
アプリキー※3	—	—
EZ キー※3	—	—
メールキー※3	—	—

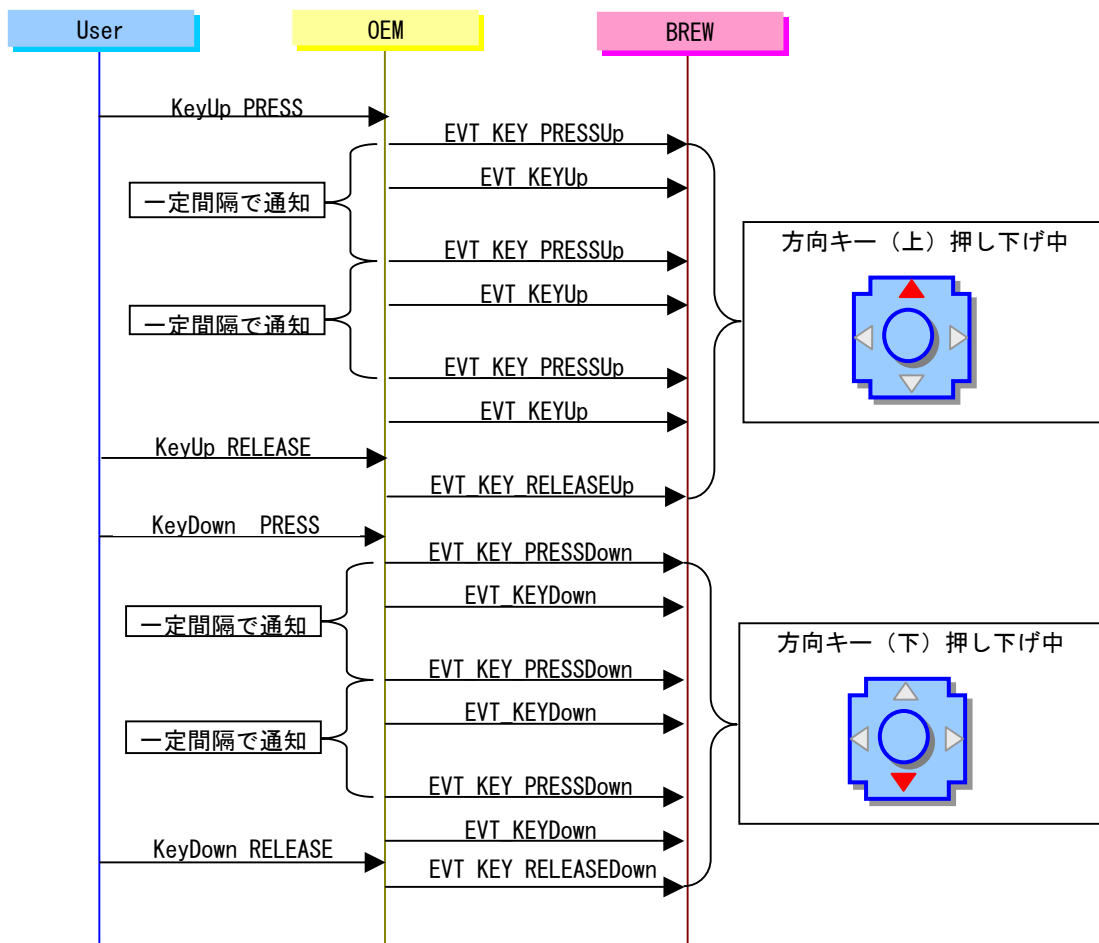
※1 AVK\_END は AEE に渡され、BREW アプリには最終的に STOP\_EVENT として渡されます。

※2 「ソフトキー3」は推奨です。

※3 BREW には通知されません。



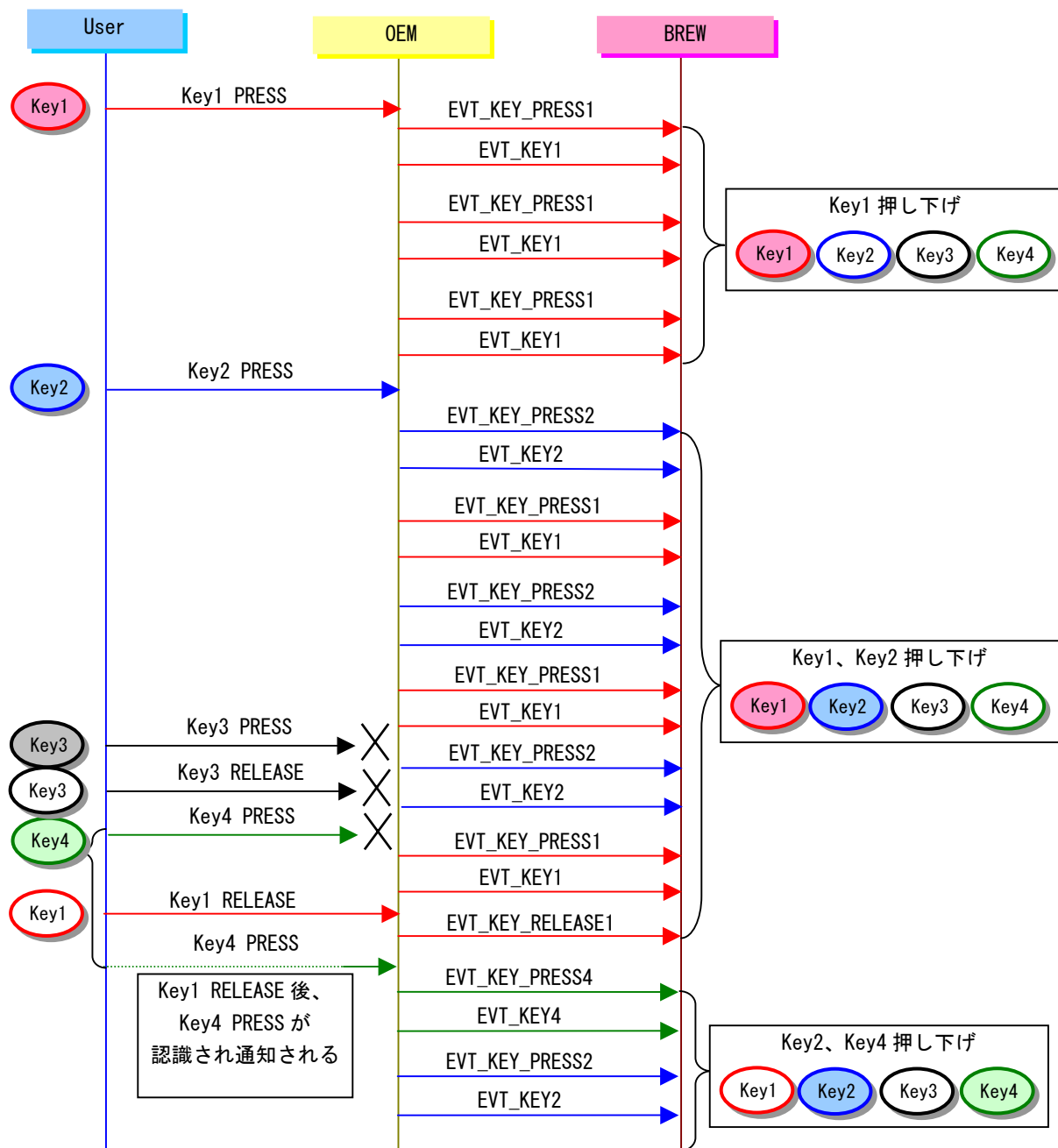
①リピート動作のキーイベント通知



#### 4.4.3. キーの同時押下

BREW アプリでは Key On で確定となり、2 つまでのキー同時押下の検出を可能としています。3 つ目のキーに関しては無視され、1 つ目のキーが離されたタイミングで3 つ目のキーが有効となります。この仕様を利用し、“右上”等の斜め方向検出は上キーと右キーを同時押下しこれらを認識することで実現します。

##### ①同時押下のキーイベント通知

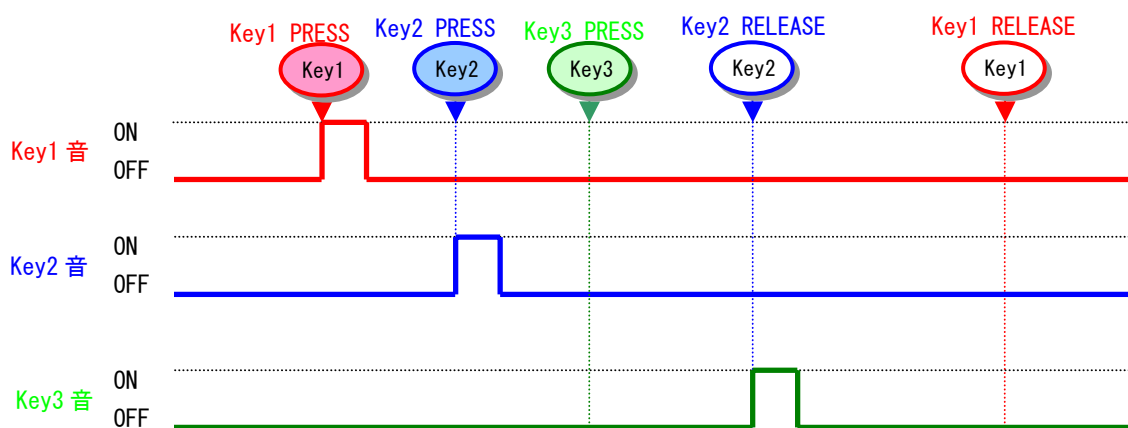




#### 4.4.4. キー操作音

BREW アプリ実行中のキー操作音は端末の Native 設定 (M24 : キー操作) に従い鳴音します。キーの押下順序を Key1, Key2, Key3 として、Key1 押下中に、Key2 の押し下げの鳴音は行いますが、Key1、Key2 押下中に Key3 押下してもキー操作音は鳴音しません。Key3 音は Key1 もしくは Key2 を RELEASE した瞬間に鳴音します。なお、マナーモード設定時は、マナーモードに従った動作音とします。

また、ISOUND\_PLAYTONE でキー操作音を鳴音させる場合、「ms」単位での鳴音時間指定が可能です。実際には端末毎の差異が生じてしまい忠実な鳴音時間の再現ができません。実機にてご確認ください。



#### 4.4.5. ソフトキー

ソフトキーを表示する場合は IDisplay 等のグラフィック系インターフェースにて BREW アプリ自身で描画してください。

なお、ITextCtl インターフェースと IMenuCtl インターフェースを利用して作成したソフトキーにつきましては、ソフトキーを 1 つのみ使用した場合に横画面いっぱいに引き伸ばされたソフトキーデザインになってしまう事や以下の理由により使用を禁止させていただきます。

##### ① ITextCtl インターフェース使用時の 6 ピクセルの空白について

ITextCtl インターフェースにより描画された入力ボックスの下 6 ピクセルが空白で表示されてしまいます。これは BREW2.0 の仕様である為、ITEXTCTL\_GetRect() を使用する場合は入力ボックスの下 6 ピクセルには文字列等を配置できません。

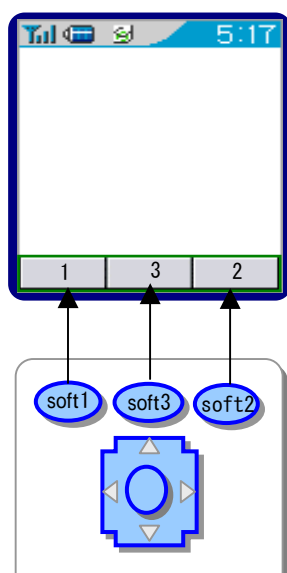
##### ② ソフトキー上のフォーカスならびにローテーション動作について

Emulator 上でのソフトキーの動作は、真中のソフトキーにカーソルがあたりフォーカスされている状態で左右キー押下によりローテーションさせセンターキー押下で選択するものとなっておりますが、KDDI の提供する BREW 搭載端末では日本市場に合わせ、端末に搭載されたソフトキーと連動したキー操作ができるよう仕様変更しています。よって、ITextCtl インターフェースと IMenuCtl インターフェースによるソフトキーを使用する場合は必ず各ソフトキーのキーイベントを割り当てる必要があります。

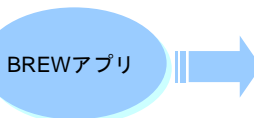
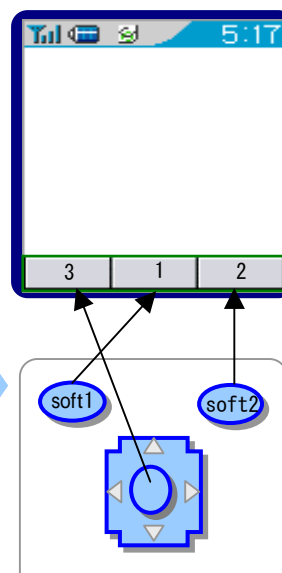
##### ③ 端末毎のキー配置仕様の違いについて

ITextCtl インターフェースと IMenuCtl インターフェースを使用して下図左のようなソフトキーを表示させた場合、それをソフトキーが 2 つの端末に対応させると下図右のようなキー配置となります。このように端末によってキー配置に差異が生じてしまいます。

ソフトキーを 3 つ搭載している端末の場合



ソフトキーを 2 つ搭載している端末の場合



IDisplay 等のグラフィック系インターフェースにてソフトキーを表示する際の各規定については以下の通りです。

描画方法	IDisplay 等グラフィック API
表示位置	自由に描画可能とするが推奨は最下行
形状	自由に描画可能
色	自由に描画可能
数	自由に描画可能とするが KeyEvent は AVK_SOFT1、AVK_SOFT2、AVK_SOFT3
文字数	自由に描画可能
サイズ	自由に描画可能
備考	

## 4.5. 文字仕様

### 4.5.1. 表示文字

BREW アプリが使用可能な文字は、各端末にて用意されている文字のみとなります。なお、同一フォントサイズ (Normal font, Large font 等) でも端末によって表示されるサイズが異なりますのでご注意ください。

### 4.5.2. 文字入力

#### (1) 指定可能入力モード

BREW アプリより文字入力を行う場合は端末メーカーにより実装されている外部エディタ利用し、以下の入力モードの指定が可能です。なお、パスワード入力時は全て伏せ字にて入力/表示されます。

入力モード	BREW からの指定	入力可能モード	入力切替可能な種別(※)						
			漢あ	か	かな	A a (*1)	Abcd (*2)	12 (*1)	1234 (*2)
かな漢	AEE_TM_NONE AEE_TM_SYMBOLS AEE_TM_RAPID AEE_TM_KANJI	全/半角	◎	○	○	○	○	○	○
半角英字	AEE_TM_LETTERS	半角	×	×	×	×	◎	×	○
半角数字	AEE_TM_NUMBERS	半角	×	×	×	×	×	×	◎

※「入力可能な種別」◎：可能+初期値 ○：可能 ×：不可

※半角記号/全角記号/絵文字については現在検討中

\*1：全角英字、または全角数字

\*2：半角英字、または半角数字

#### (2) 入力可能文字

文字入力 Shift-JIS (JIS-X0208)にて規定される第一、第二水準の文字コードで、記号は Shift-JIS (JIS-X0208)にて規定されるもの、及び Shift-JIS (8740~8775、877E、8780~879C) のもの、絵文字は KDDI 絵文字用 Shift-JIS コードにて規定されるものとなります。なお、絵文字のアニメーションには対応していません。また、「半角カタカナ」の範囲は、0xa0~0xdf までとします。

#### (3) 最大入力文字

入力エディタ上の入力文字最大数以下の通りです。

- ① かな漢モード：500 文字 (500~1000byte)
- ② 半角英字モード：500 文字 (500byte)
- ③ 半角数字モード：500 文字 (500byte)

#### (4) 留意点

BREW 内では BREW 用の 2 バイト文字コードにて処理を行う為、アプレットにて Shift-JIS コードから BREW 用の 2 バイト文字コード (AECHAR 文字列、Pseudo-UNICODE と呼ぶことがあります) に変換して渡す必要があります。BREW 用の 2 バイト文字コード変換にはヘルパー関数の STREXPAND() を使用して下さい。また、その逆は WSTRCOMPRESS() 関数となります。

なお、STREXPAND() は改行コードには対応していません。また、STRTOWSTR()、WSTRTOWSTR() は使用しないで下さい。

### 4.5.3. 絵文字

SDK 上で表示するには外字登録を行うことで可能となります。

## 4.6. 音源仕様

### 4.6.1. サポート音源フォーマット

BREW 対応端末で対応する音源フォーマットは以下の通りです。

フォーマット	拡張子	対応	備考
QCELP (PureVoice)	.qcp	標準	Qualcomm 社が提唱する音声圧縮に優れたフォーマット
SMAF MA3/MA2/MA1	.mmf	標準	YAMAHA 社が提唱するアニメーションにも対応した MIDI ベースのマルチメディアフォーマット。ADPCM 等音声コーデックにも対応し最大 32 音のメロディと 8 音の PCM が再生可能。
SMAF MA3 Phrase	.spf	標準	SMAF をベースとしたマルチシーケンサ。4 種類の音を同時に制御可能。
C-MIDI3.0/2.0/1.0 (CMX)	.pmd	端末依存	Qualcomm 社が提唱するアニメーションにも対応した MIDI ベースのマルチメディアフォーマット。QCELP の音声コーデックにも対応し最大 16 音のメロディと 1 音の QCELP が再生可能。PMD (WebAudio) と呼ぶ場合もあります。
MP3	.mp3	端末依存	『MPEG Audio Layer 3』の略。音質をほとんど劣化させることなく、比較的小容量のファイルに圧縮(エンコード)できる。圧縮率は 1/10~1/12。

### 4.6.2. IMedia インターフェースと ISoundPlayer インターフェース

IMedia インターフェースでは SMAF コンテンツの同時複数制御 (Phrase シーケンサ) を使用することができのに対し、ISoundPlayer インターフェースでは同時複数制御は使用することができません。ISoundPlayer インターフェースは既存アプリとの互換性の為に存在していることもあり、ISoundPlayer インターフェースを使用するよりも IMedia インターフェースを使用することを推奨します。

なお、IMedia インターフェースと ISoundPlayer インターフェースの API レベルの対応状況は端末依存となります。

#### 4.6.3. 音源使用上の注意

音源を使用するにあたり以下の点について注意しなければなりません。

①表示系を含んだ SMAF ファイルは演奏系自体の再生もできません。

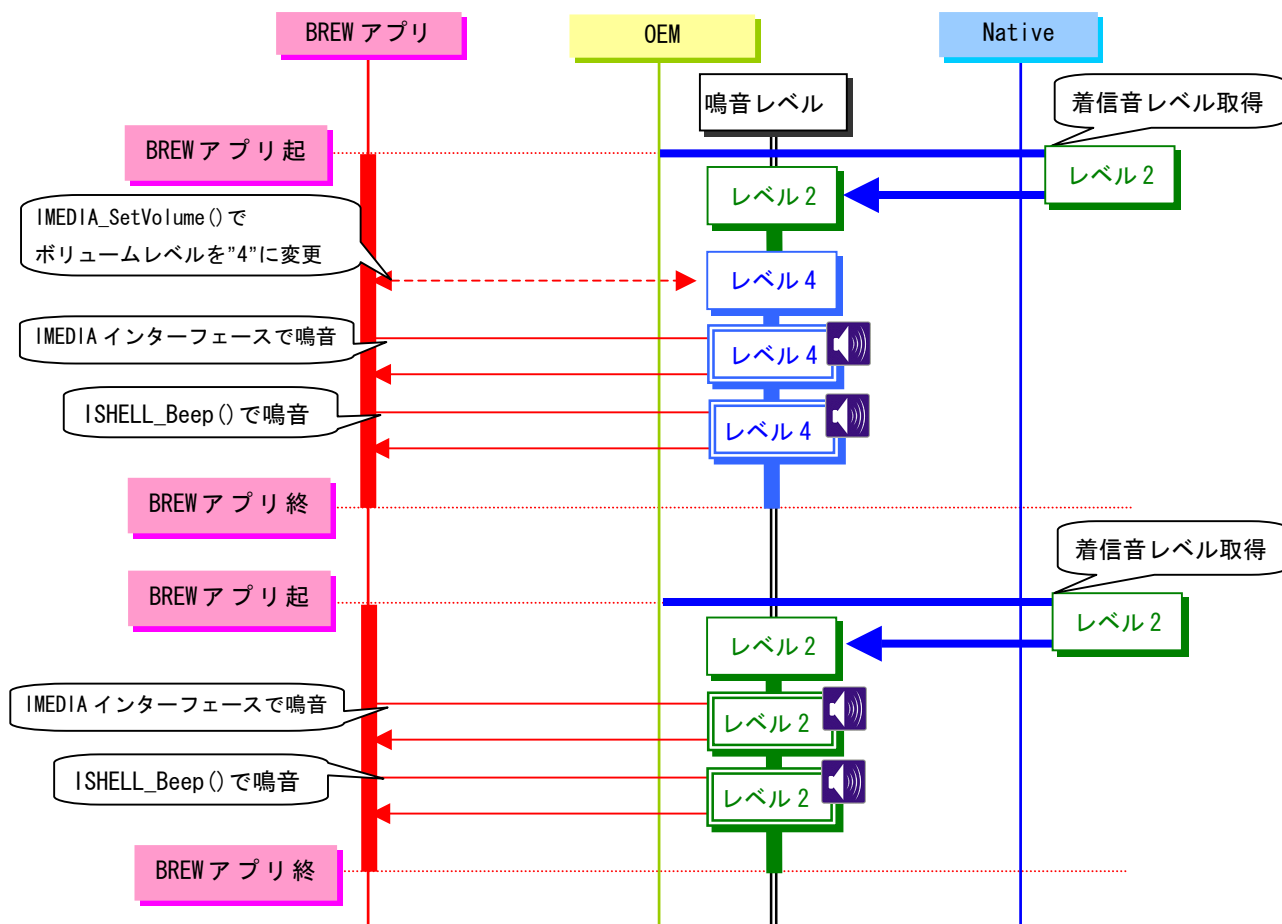
SMAF 種別	再生可否
和音メロディ SMAF	○
アニメ&メロディ SMAF	×
ボイス SMAF	○

②SMAF ファイルと SMAF/Phrase 用ファイルの同時再生はできません。

## 4.6.4. ボリューム(音量)

BREW にて使用可能な音源のボリュームは 100 段階で指定可能ですが、実際には Native の着信音量調節(通常音量 M211)の設定に合わせレベル 0/1/2/3/4/5 の 6 種類のボリュームに丸め込まれます。音を鳴音させる場合に BREW アプリからのボリューム設定がない時には端末にて設定されている着信音量設定(M211)に従い鳴音を開始します。BREW アプリからボリュームの変更要求があった場合にはボリュームを変更することができますが、BREW アプリから Native のボリューム設定(M21X)自体を変更することはできません。

	BREW アプリの状態	
	起動直後	ボリュームレベル変更後
鳴音する ボリュームレベル	端末内で、設定されている着信音量レベルで鳴音する	変更されたボリュームレベルに対応する端末内の着信音レベル(0~5)で鳴音する



## (1) BREW アプリから指定/取得設定可能なボリュームレベル

BREW アプリからの指定は 0～100 までの可変で設定可能ですが、AEE 層で丸め処理が実行され丸められたボリュームとなり、丸められたボリューム値が取得されます。これに対し、IMEDIA (SMAF、SMAF/Phrase) は、AEE 層での設定ボリュームを丸める処理がない為、設定したボリュームがそのまま取得されます。インターフェース毎の指定時/取得時のボリュームレベルの違いを以下に示します。

BREW API (ファイル種別)	BREW アプリから 指定可能な ボリュームレベル	BREW アプリが 取得可能な ボリュームレベル	端末内での 鳴音ボリュームレベル
ISOUND インターフェース IMEDIA インターフェース (QCELP/CMX/MP3) ※CMX/MP3 は検討中	0～19	0	レベル 0
	20～39	20	レベル 1
	40～59	40	レベル 2
	60～79	60	レベル 3
	80～99	80	レベル 4
	100	100	レベル 5
IMEDIA インターフェース (SMAF、SMAF/Phrase)	0～19	0～19	レベル 0
	20～39	20～39	レベル 1
	40～59	40～59	レベル 2
	60～79	60～79	レベル 3
	80～99	80～99	レベル 4
	100	100	レベル 5

## (2) マナーモードとの競合

マナーモード設定がされている場合は、ボリュームの変更動作が可能な以下の API を使用し指示をしても端末はマナーモードで設定されているボリュームレベルにて鳴音します。また、常に成功のステータスが通知されるのみで、エラーは通知されません。ただし、IKDDIPHONESYSTEM\_GetMannerMode() を利用することでマナーモード設定の有効/無効情報を取得し、マナーモード中になっている為音がならない等の注意をユーザに促すことができます。

- IMEDIA\_SetVolume
- ISOUND\_SetVolume
- ISOUNDPLAYER\_SetVolume



#### 4.6.5. トーン

BREW アプリからのトーン(Tone/Alert/Key)の鳴音が可能です。端末でサポートされるトーンの種別は以下の通りです。その他を指定した場合は鳴音しませんのでご注意ください。

BREW AEE Tone 定義	端末内定義音	音種別	鳴音デバイス	
			イヤホンなし	イヤホンあり
AEE_TONE_RBACK	リングバックトーン	Tone	レシーバ	イヤホン
AEE_TONE_BUSY	ビジートーン			
AEE_TONE_DIAL_TONE_TONE	ダイヤルトーン			
AEE_TONE_PWRUP	電源投入音	Alert	スピーカ	
AEE_TONE_0	0key 押し下げ音	Key		
AEE_TONE_1	1key 押し下げ音			
AEE_TONE_2	2key 押し下げ音			
AEE_TONE_3	3key 押し下げ音			
AEE_TONE_4	4key 押し下げ音			
AEE_TONE_5	5key 押し下げ音			
AEE_TONE_6	6key 押し下げ音			
AEE_TONE_7	7key 押し下げ音			
AEE_TONE_8	8key 押し下げ音			
AEE_TONE_9	9key 押し下げ音			
AEE_TONE_POUND	#key 押し下げ音			
AEE_TONE_STAR	*key 押し下げ音			
AEE_TONE_WARN	キー後入力等の警告音			
AEE_TONE_ERR	キー後入力等の警告音			

※レシーバ：受話口

#### 4.6.6. 鳴音デバイスの指定

QCELP に限り、BREW アプリから特定の鳴音デバイス（例：スピーカのみ）を指定することが可能ですが、イヤホンの接続状態により端末スピーカもしくはイヤホンからの鳴音を決定することが一般的でありユーザビリティを大きく損ねる可能性があることから、鳴音デバイスの指定を禁止します。必ず、「AEE\_SOUND\_DEVICE\_CURRENT」を選択するようにして下さい。

#### 4.6.7. テンポ設定

IMEDIA インターフェースにて、SMAF/C-MIDI (CMX) ファイルのみ BREW アプリからのテンポレベルの設定が可能です。BREW アプリから指定するテンポレベルと端末内のテンポレベルの対応は以下の通りです。なお、下記設定範囲以外が設定された場合はエラーが返され、以降はデフォルト値「100」で動作します。※C-MIDI については現在検討中 (0-255、デフォルト 128)

BREW アプリ設定範囲	端末内テンポレベル
70~79	70
80~89	80
90~99	90
100~109	100 (Default)
110~119	110
120~129	120
130	130

#### 4.7. バイブレータ

ISOUND\_Vibrate()により BREW アプリからのバイブレート要求があった場合、端末の設定には依存せず BREW アプリの要求にしたがってバイブレーションします。バイブレーションは BREW アプリからの START/STOP にて制御されます。Native 側でのイベントが発生した場合にも BREW アプリにより制御されますので、必ず STOP するようにして下さい。なお、バイブレータ振動パターンは固定パターンとなります。

#### 4.8. ビープ

エラー発生等をユーザに通知するシステムビープを BREW アプリからでも使用することができます。

##### (1) ビープ音

BREW アプリではビープ音の鳴音が可能ですが、通常の音源再生とは異なりビープ音には BREW アプリからボリュームを設定する API は用意されていません。ビープ音は端末で使われている音を使うのでその音に関連するボリュームレベルによって鳴音させることになります。但し、BREW アプリにより自信がアクティブ中の時の音量を変更している場合にはその音量に従い鳴音します。詳細は「[4.6.4. ボリューム\(音量\)](#)」を参照して下さい。

また、ERROR 音は現在鳴音しているキー操作音量に従い鳴音し、マナーモード設定時はマナーモードに従った動作音量に従い鳴動します。

Beep Type	従う音量設定	ビープ音の種類
BEEP_OFF	—	ビープ音、ビープバイブレータの停止
BEEP_ALERT	通常音量 [M211]	通常着信音量でのビープ音
BEEP_REMINDER		通常着信音量でのビープ音を 1 分ごとに繰り返す
BEEP_MSG		BREW SMS 受信時のビープ音
BEEP_ERROR	キー操作音 [M24]	Text 入力などで使われる Error のビープ音

※BEEP\_REMINDER は BREW アプリ終了時には停止すること。

※ビープ音を higher volume で鳴音させる場合は各 Volume の最大値で鳴音すること。

※BREW SMS 受信の場合は、Native にて SMS 受信の着信音(バイブレートも含む)の動作は行わない。

##### (2) ビープバイブレータ

BREW アプリではビープ音同様に端末をバイブレーションさせることが可能です。

Beep Type	ビープバイブレータの種類
BEEP_OFF	ビープ音、ビープバイブレーションの停止
BEEP_VIBRATE_ALERT	固定時間バイブレーション
BEEP_VIBRATE_REMIND	一定時間ごとに固定時間バイブレーション

#### 4.9. バックライト

バックライトは IDISPLAY\_Backlight() により BREW アプリから点灯/消灯の制御を行うことが可能で、端末の「通常照明 (M321)」の設定が「ON」の場合は、BREW アプリの点灯/消灯要求を優先して動作します。これに対し、端末の「通常照明 (M321)」の設定が「OFF」の場合は、BREW アプリの点灯要求にかかわらず、常に消灯となります。なお、「車載照明 (M33)」が「ON」に設定され、かつ電源アダプタが接続されている場合は、BREW アプリの点灯要求に係らず常時点灯となりますので注意が必要です。

「通常照明 (M321)」が「OFF」、もしくは「ON」状態において電源アダプタが接続されていない場合での「照明設定 (M321)」、「節電画面設定 (M34)」との対応関係は以下の通り。

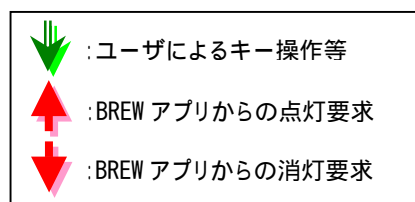
照明設定 (M321)			節電画面設定 (M34)		
	BREW アプリ 点灯要求	BREW アプリ 消灯要求		BREW アプリ 点灯要求	BREW アプリ 消灯要求
照明 ON	点灯	消灯	設定 ON (節電画面移行前)	点灯	消灯
			設定 ON (節電画面移行後)	節電画面	節電画面
			設定 OFF	点灯	消灯
照明 OFF	常時消灯	常時消灯	設定 ON (節電画面移行前)	消灯	消灯
			設定 ON (節電画面移行後)	節電画面	節電画面
			設定 OFF	常時消灯	常時消灯

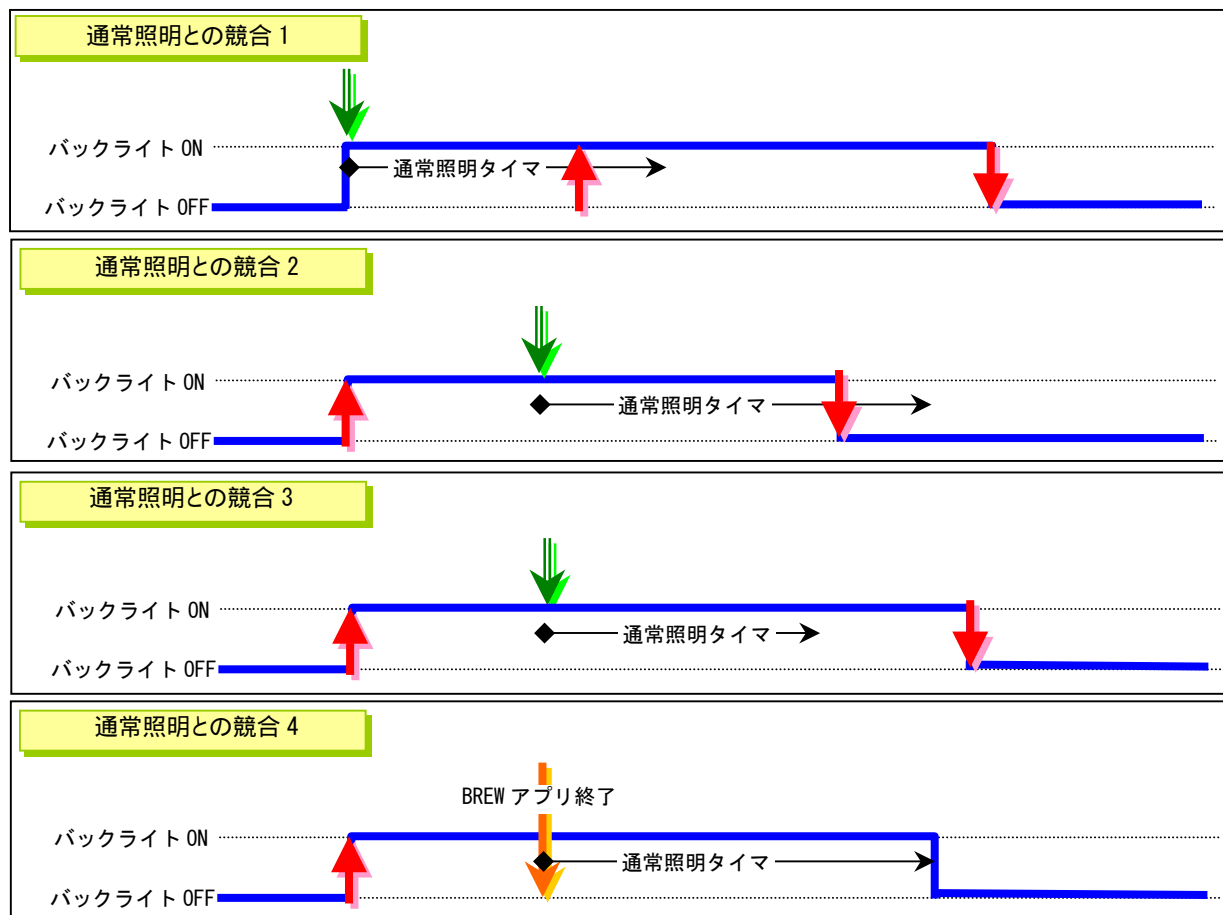
##### (1) 「節電画面設定 (M34)」との競合上の注意事項

節電画面 (パーシャルとも言います) へ移行する前に BREW アプリからの点灯要求があった場合、「節電画面設定 (M34)」にて設定した時間が経過しても節電画面へは移行しません。

なお、一度節電画面へ移行するとユーザによるキー操作以外に節電画面から抜ける術はなく、BREW アプリからの点灯要求を行っても要求は無視され節電画面を継続し、点灯要求は保持されることなく破棄されてしまいます。

これらをまとめた具体的な動作を次頁に示します。各々の矢印は以下を表しています。



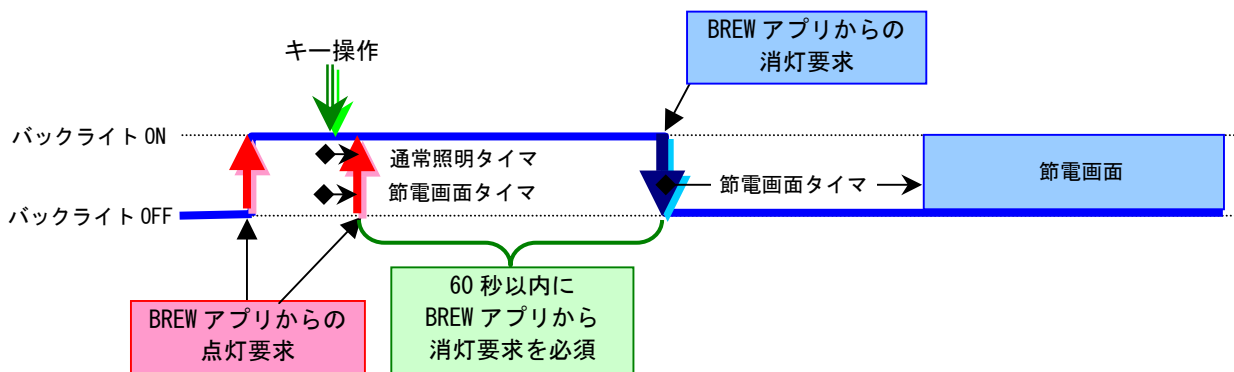


## (2) バックライト制御における注意事項

## ① BREW アプリからの点灯/消灯要求ルール

点灯要求を出す場合は必ず”60秒以内”（※）の消灯要求を併せて持つことを前提に、BREW アプリからバックライトの点灯制御を自由に行っても構いません。これにより節電画面設定の移行時間がデフォルトの”30秒”としていた場合でも無操作状態において”90秒”（=60+30）の画面表示が可能であり、併せて常時バックライトが点灯され続ける状態を回避できる為バッテリー消費を低減することもできるようになります。なお、ユーザのキー操作を契機に節電画面用のタイマーがリスタートしてしまいタイマー満了後節電画面へ移行してしまうことから、極力長い時間画面表示を見せたい BREW アプリはキーイベント発生後に続けてバックライトの点灯要求を出すことで画面表示を継続させることができます。ただし、上記と同様に 60 秒以内の消灯要求を条件に許容することとします。

※”60秒”の根拠は、各端末における照明設定の最大値が”60秒”となっており、これ以上を許容することは各端末におけるバッテリー消費に大きく影響を与える可能性がある為。



## ② Active でない BREW アプリからのバックライト点灯制御

Active でない BREW アプリからバックライト点灯をされると端末側は点灯制御をされたことが検出出来ずにバックライトが点灯しっぱなしになります。よって、Active でない BREW アプリからは、バックライトの点灯制御は決して行わないでください。

## ③ サスペンドならびに BREW アプリ終了時のバックライト消灯

バックライト点灯要求をしたアプリは、サスペンドならびに BREW アプリ終了時には必ず自らバックライトの消灯を行って下さい。A アプリがバックライト点灯を要求したまま次の B アプリが起動すると、B アプリからはバックライト点灯の要求をしなくても、バックライトが継続点灯してしまう端末が存在します。

#### 4.10. 対応画像フォーマット

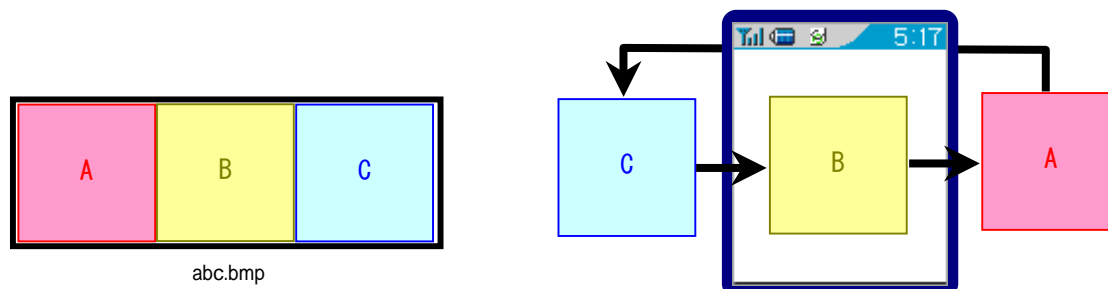
BREW 対応端末にて実装される画像デコーダは以下の通りです。なお、ファイル数低減を考慮しこれら画像ファイルをリソースエディタにて BAR ファイルに纏めた上で使用するようになっています。

実装デコーダ	拡張子	備考
BMP	.bmp	16bit カラー以下に対応
BCI (BREW Compressed Image)	.bci	Qualcomm 社により準備された BREW 上の画像フォーマット。 SDK にある BCI 作成ツールで作成可能。
PNG	.png	透過はインデックスカラーの tRNS チャンクのみ対応。 $\alpha$ チャンネルによる非対応。
JPEG	.jpg .jpeg	JFIF のみ対応。プログレッシブ JPEG 等は非対応。

##### 4.10.1. 画像ファイル使用上の注意事項

###### (1) Image インターフェースを利用したアニメーション表示

Image インターフェースでは IMAGE\_SetParm() や IMAGE\_Start() 等複数の API により、ビットマップの幅方向に複数のフレームを配置することでアニメーション表示が可能です。アニメーションを開始すると、Image により指定した速度で画像のフレームが順次表示されます。最終フレームの表示が終了すると、最初のフレームからアニメーションが再開され、アニメーションを停止するか Image インターフェースのインスタンスを解放するまで、このサイクルが繰り返されます。ただし、これらアニメーション可能なフォーマットは”BMP”に限られ、PNG や JPEG は非対応となりますのでご注意ください。詳細は「BREW2.0 API リファレンス」を参照して下さい。



###### (2) 透過指定

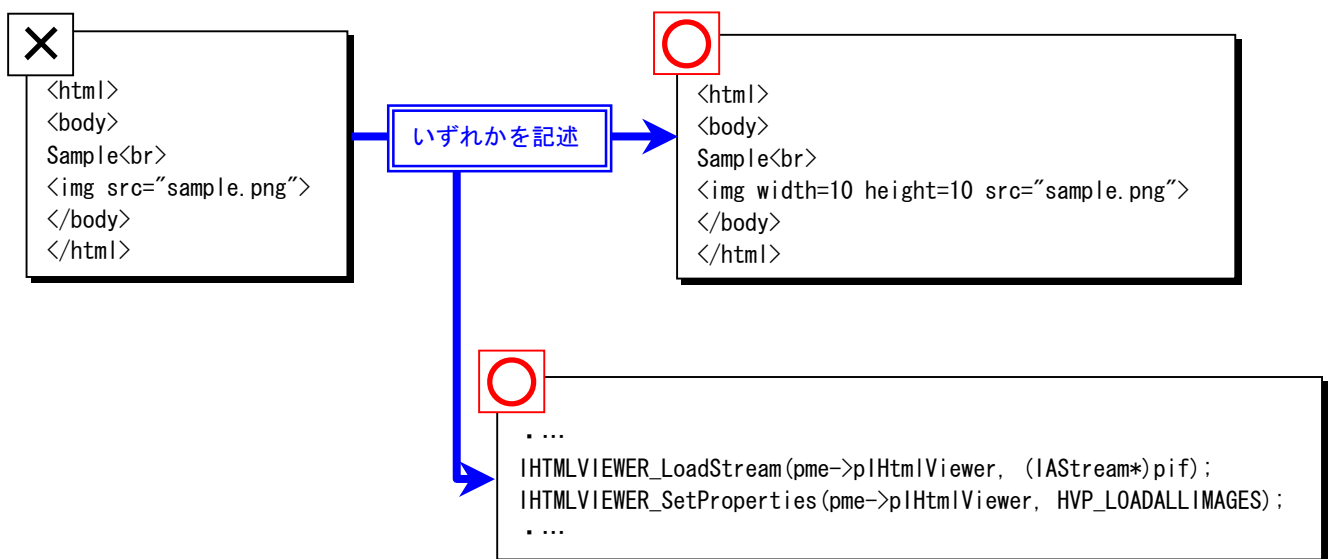
透過指定には、「PNG 内の tRNS チャンクにより透過指定する方法」と「BREW アプリから明示的に AEE\_RO\_TRANSPARENT を IPARM\_ROP に指定することにより実現する方法」の 2 種類が存在します。2 種類の透過が競合することを想定し、「PNG 内の tRNS チャンクにより透過指定する方法」を用いるようにして下さい。なお、アプリケーションマネージャ上では後者の方法を応用しパレットインデックスの”0x00”が強制的に透過されるよう作られていますので注意が必要です。また、 $\alpha$  チャンネルを用いた透過は非対応となります。

## (3) JPEG データのデコード

JPEG の画像データを表示するには ISHELL\_LoadResImage() 関数で読み込み、直後に IIMAGE\_Notify() 関数を使ってコールバック関数を登録しこのコールバック関数内で IIMAGE\_SetDrawSize() と IIMAGE\_Draw() を使う必要があります。JPEG の画像データを ISHELL\_LoadResImage() 関数で読み込み、直後に IIMAGE\_SetDrawSize() と IIMAGE\_Draw() 関数を使って表示させようとしても表示されませんのでご注意ください。なお、読み込んだデータが BMP であれば表示されます。

## (4) IHTMLViewer 上での画像表示

IHTMLViewer で IMG タグを使って画像を表示する際に、以下のような単純な<img>タグ指定だけでは画像が表示されません。本現象を回避できる方法として「<img>タグ内で画像サイズ(Width/Height)を明示的に記述する方法」と「IHtmlViewer\_SetProperty()を使用する方法」の2つが存在しますのでいずれかを必ず記述するようにして下さい。



## (5) MIF エディタ内にて使用不可能な画像フォーマット

MIF エディタ内で指定された JPEG は動作保障されていませんので使用しないようにして下さい。

また、BREW2.1 対応端末用アプリにおいてはアイコンの対応画像形式に BCI を使用しないでください。

## (6) デコーダの仕様差異

エミュレータを含む BREW2.0 SDK で実装されている各種デコーダと端末上で使用する各種デコーダとの間に仕様の差異が生じる可能性があります。このため、対応 bit 数や透過処理など扱える画像の種類やデコーダの処理速度が異なることが想定されます。

## (7) BCI、PNG、JPEG データのメモリ展開

BCI、PNG、JPEG データは瞬間的に 40KB 程度のヒープを消費します。このためメモリに余裕があることを確認してからメモリに展開するようにして下さい。

## (8) JPEG 画像の変換について

24 ビットの JPEG 画像を処理するために、IImage から IDIB に変換を行うと、24 ビットから 16 ビットに減色される機種があります。

#### **4.11. 日時**

BREW アプリは端末から日時情報を取得することが可能で、この時利用する時間は System Time となります。圏外においても SystemTime を取得できます。

#### **4.12. BREW アプリのダウンロード**

##### **4.12.1. BREW アプリダウンロード基本仕様**

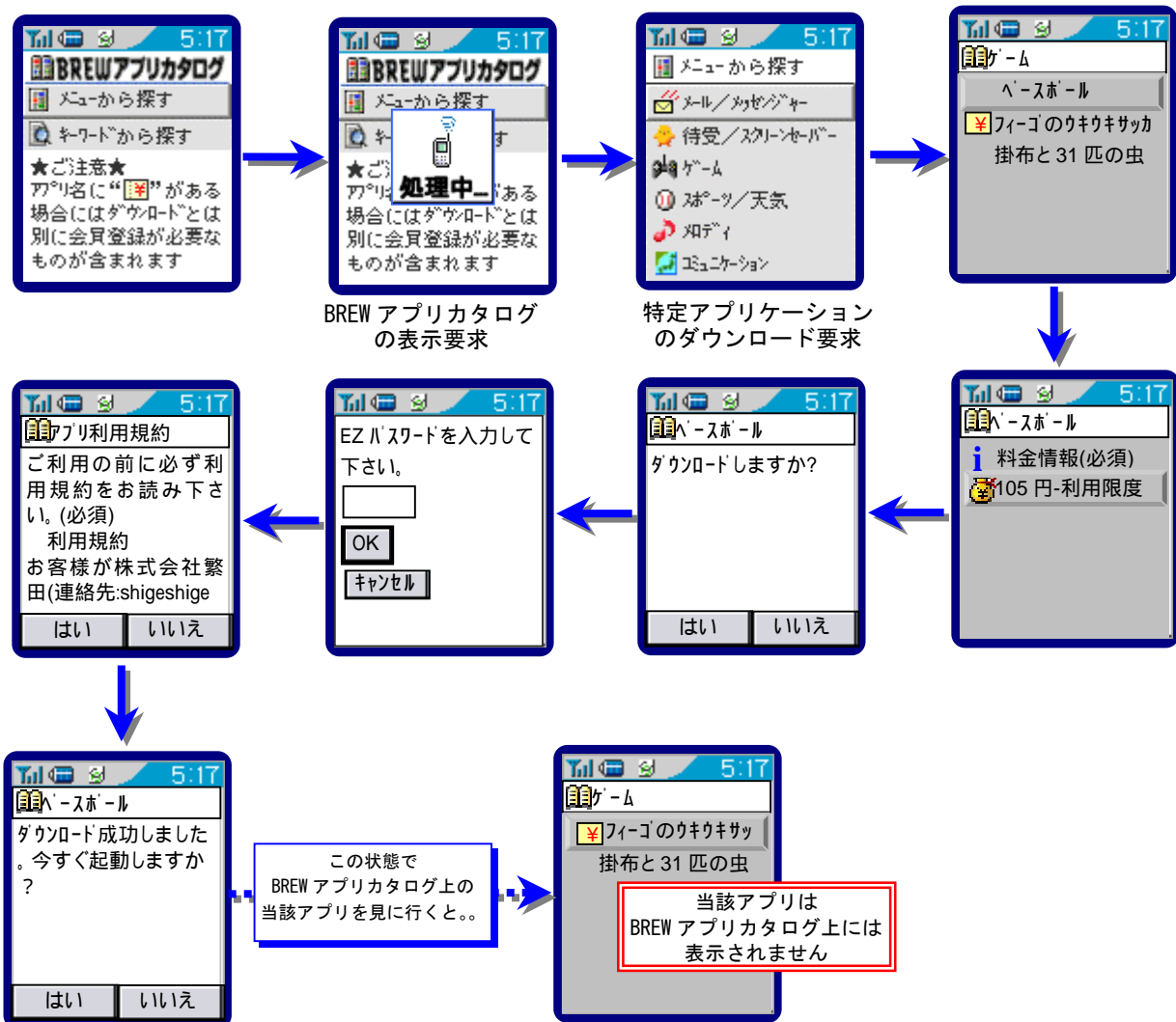
検証等動作確認が完了となり UAM に登録された BREW アプリを BREW 対応端末にプリインストールされたモバイルショップにより ADS からダウンロードされ端末内部に保存します。ADS へのアクセスは初期設定 (BREW サーバ、EZ サーバ両方の初期設定) 完了後アクセス可能となります。

なお、ダウンロード以外に BREW アプリを端末内部に保存するには PC 上に BREW アプリ転送用ツールをインストールし、BREW 端末を BREW アプリ転送可能なモードに切り替える必要があります。この転送の為の方法は一般のユーザへは開示していません。実際に動かすためには、BREW アプリの企画をご提案頂き、これが採用され、BREW アプリの提供 CP (コンテンツプロバイダ) として KDDI に認定される必要があります。

##### **4.12.2. BREW アプリカタログからのダウンロード**

ユーザ操作により「BREW メニュー」内の「BREW アプリカタログ」を選択し要求があった場合、ADS に接続しに行き BREW アプリカタログを表示します。カタログのメニューの中から BREW アプリを選択することで、任意の BREW アプリをダウンロードする事が出来ます。ダウンロードされた BREW アプリはその後 BREW フォルダ上から消去するまで BREW アプリカタログ上には表示されなくなります。なお、ダウンロードされる BREW アプリが、端末上の保存するエリア、つまり BREW フォルダの残容量よりも大きい場合には、対象 BREW アプリはダウンロードすることができません。





※画面はイメージですので変更となる場合がございます。ご了承下さい。

#### 4. 12. 3. WAP ブラウザからのダウンロード

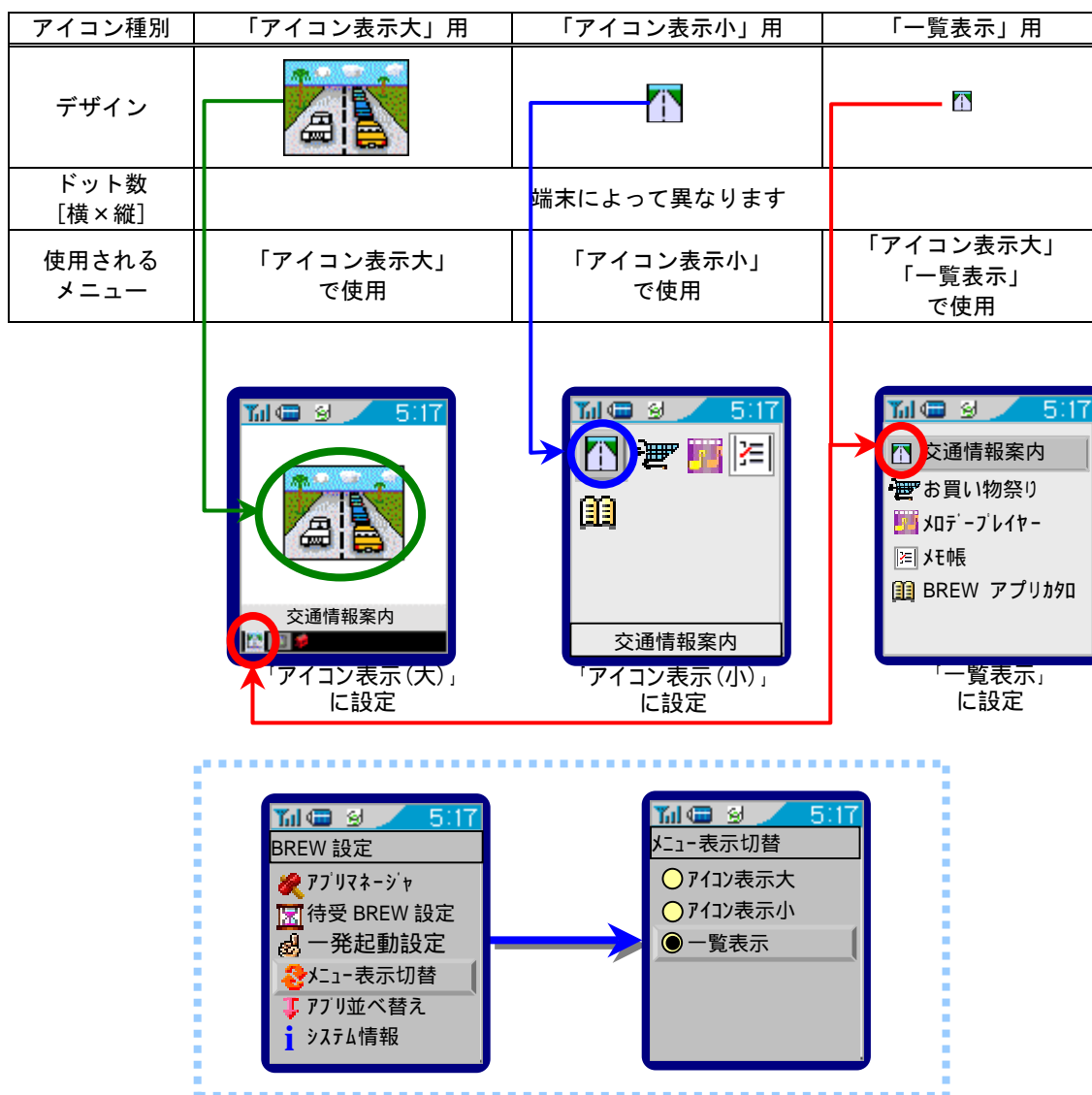
EZweb サイト上に書かれた特定のタグ記述により特定の BREW アプリを指定することで、WAP ブラウザからモバイルショップを呼び出し、特定の BREW アプリが BREW フォルダ上に保存されていない場合はその指定したアプリケーションを ADS から連続してダウンロードさせることができます。この時 BREW アプリの指定にはリンク用 Item\_ID が用いられます。EZweb サイトからアプリケーションを選択した場合の遷移先はダウンロード直前のライセンス情報の表示画面となります。詳細は、「4. 13. 4. WAP ブラウザからの起動」を参照下さい。

なお、ダウンロードされる BREW アプリが、端末上の保存するエリア、つまり BREW フォルダの残容量よりも大きい場合には、対象 BREW アプリはダウンロードすることができません。

#### 4.12.4. BREW フォルダへの保存

ADS からダウンロードされた BREW アプリは、モバイルショップによりダウンロードした BREW アプリの MIF ファイルを端末のルートディレクトリ配下に、その他 MOD ファイル等 BREW アプリ関連ファイルをモジュールディレクトリに保存します。格納方法の詳細については、「[4.2.4. BREW フォルダ\(アプリケーション保存領域\)](#)」をご覧ください。なお、ダウンロードする BREW アプリが保存するエリアよりも大きい場合には、領域不足を表示し対象 BREW アプリをダウンロードすることができません。

ADS からダウンロードされた BREW アプリは、BREW メニュー上に表示されますが、このときの表示は「BREW 設定」内の「メニュー表示切替」に設定されている内容により異なります。アイコンファイルは、下記のように大きさに対応したサイズの規定があります。間違ったサイズでアイコンを作成しますと、表示されない場合があります。なお、デフォルトは「アイコン表示大」で表示されます。



#### 4.13. BREW アプリの起動

BREW アプリの起動には大きく分けて以下の 2 種類が存在します。一つは「通常の BREW アプリとして起動」するパターンと「待受け BREW アプリとして起動」するパターンです。後者は BREW アプリに待受 BREW アプリとしての属性を持たせることで実現できます（詳細は「[4.13.7. 待受 BREW アプリ起動](#)」参照）。

なお、BREW アプリの重複起動に特に制限はありませんが、BREW アプリサスペンドにより起動に必要なメモリが取得不可能な場合は、サスペンド中の BREW アプリが UnLoad されることになります。（回避起動）

##### (1) 「通常の BREW アプリとして起動」

起動パターンは以下の通りとなります。詳細は次項以降を参照して下さい。

通常 BREW アプリの起動パターン
BREW メニューからの起動(通常起動)
アプリキー長押下による”一発起動” (※)
ダウンロード直後の起動
WAP ブラウザからの起動
アラーム起動
SMS (C メール) 受信による BREW アプリの起動
BREW アプリからの起動 (BREW アプリ連携起動)

※アプリキー搭載端末のみ

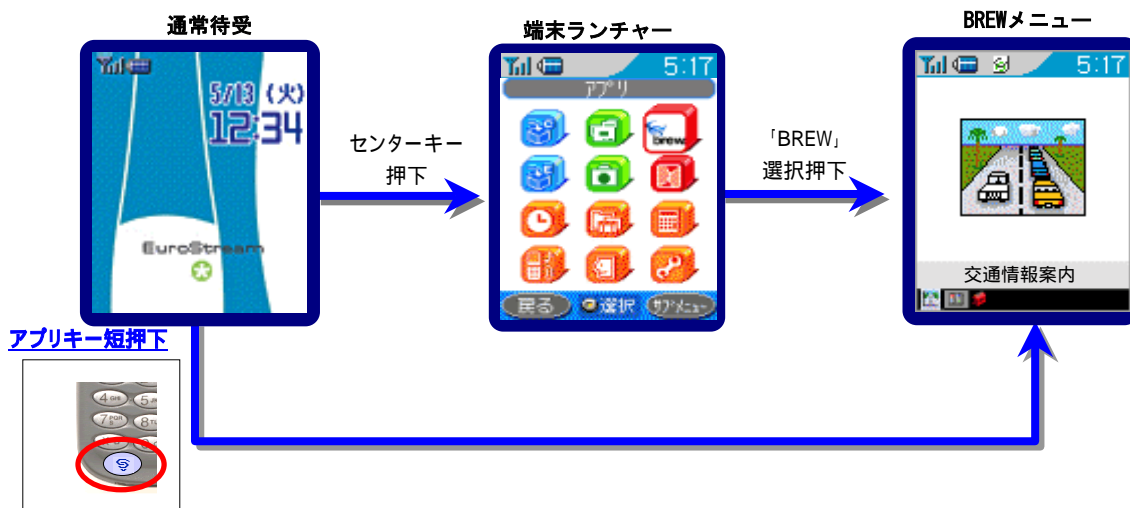
※アラーム起動は現在禁止させていただいております。

##### (2) 「待受 BREW アプリとして起動」

起動条件は、「BREW 設定」内の「待受 BREW 設定」により”待受 BREW アプリ”に設定された BREW アプリが対象で、Native 待受け状態で無操作状態が一定時間経過すると自動的に起動されます。なお、待受 BREW 設定できる BREW アプリは、1 アプリケーションのみとなります。

#### 4.13.1. BREW メニューからの起動(通常起動)

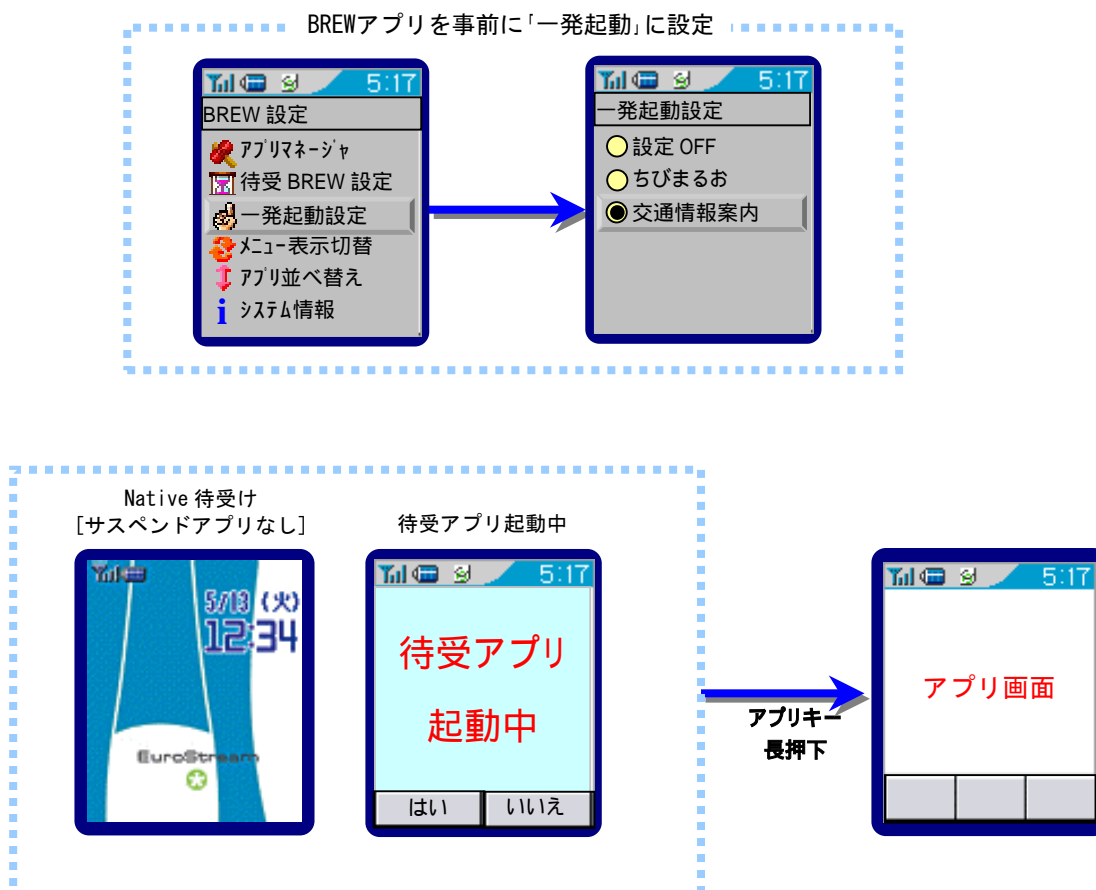
Native ランチャーで「BREW」を選択、若しくはアプリキーを短押下すると、アプリケーションマネージャが起動し BREW メニューが表示されます。ユーザはカーソルで任意の BREW アプリ選択し起動する事が出来ます。



※画面はイメージですので変更となる場合がございます。ご了承下さい。

#### 4.13.2. アプリキー長押下による「一発起動」(対応端末のみ)

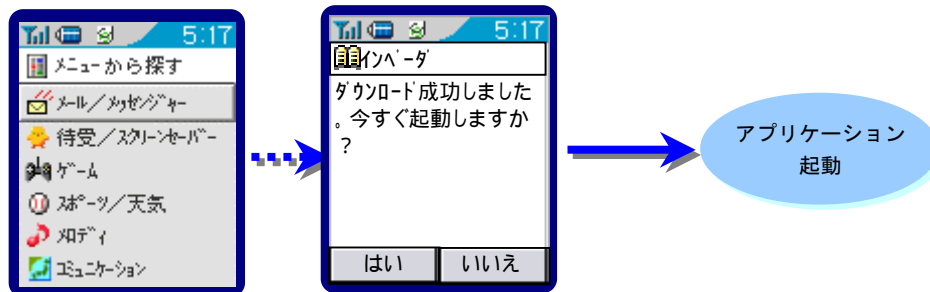
本仕様はアプリキー搭載端末のみ有効です。Native 待受け中、もしくは任意の待受け BREW アプリ起動中に、アプリキーを長押下することでユーザが事前に「一発起動」に設定した BREW アプリを起動させる事が出来ます。この設定は、「BREW 設定」内の「一発起動設定」からのみ設定可能で、BREW アプリ自身からの設定は不可能であり、必ずユーザによる事前の設定操作が必要となります。BREW アプリ起動の後の動作は BREW メニューより通常起動した時と同じ動作条件となります。なお、待受け BREW アプリ以外の任意の BREW アプリ起動中には「一発起動」はできませんので、ご注意ください。



※画面はイメージですので変更となる場合がございます。ご了承下さい。

#### 4.13.3. ダウンロード直後の起動

BREW アプリを ADS からダウンロードした直後にも起動することが可能です。BREW アプリ起動の後の動作は BREW メニューより通常起動した時と同じ動作条件となります。ダウンロード動作の詳細については「4.12.2. BREW アプリカタログからのダウンロード」を参照して下さい



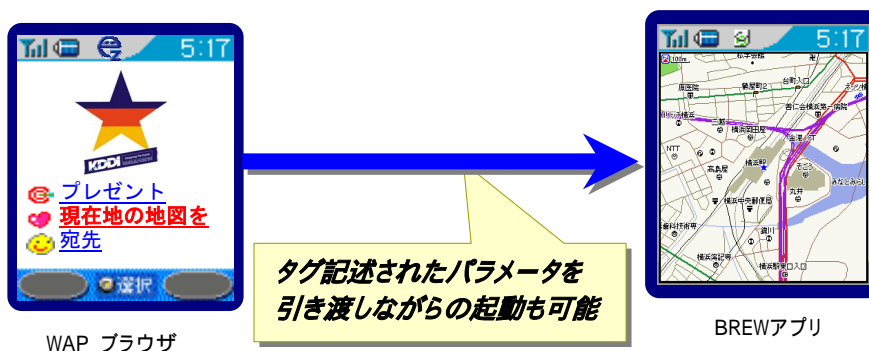
#### 4.13.4. WAP ブラウザからの起動

##### (1) ターゲット BREW アプリを指定しての起動

WAP ブラウザで EZweb ページを閲覧中、xHTML 上のタグに指定の記述がある場合、パラメータとして起動タグに記載されるリンク用 Item\_ID を元にブラウザから ISHELL\_StartAppletArgs() によるリンク用 Item\_ID に対応する BREW アプリの起動が可能となります。なお、HDML では利用できません。BREW アプリ起動の後の動作は BREW メニューより通常起動した時と同じ動作条件となります。

リンク用 Item\_ID により指定した BREW アプリが BREW フォルダに存在すると WAP ブラウザから特定の BREW アプリの起動が行えます。また、起動にあたっては起動タグにパラメータを併せて記載することで BREW アプリに対しパラメータを引き渡しながら起動することが可能となり、起動時の BREW アプリ状態を指定することが出来ます。BREW フォルダに指定した BREW アプリが存在しない場合は自動で ADS に接続に行き BREW アプリのダウンロード画面へと遷移します。

なお、WAP ブラウザの設定で「添付データ再生設定」を”OFF”にしていると上記のブラウザからの BREW アプリ起動ができません。ページ中にてユーザに注意を促すよう記述をお願いします。



## ①タグ記述方法

ML 内の記述は以下のようになり、各 Attribute(属性)に対する Value(値)は票の通りです。

```
<object data="mshop1" type="brew" standby="現在地の地図">
  <param name="itemid" value="itemid" valuetype="data">
    <param name="param" value="param" valuetype="data">
  </object>
```

## タグ内容詳細

Tag	Attribute	Value	MaxLength	Description
<object>	data	mshop1	6byte(固定)	ADS 接続
	type	brew	4byte(固定)	BREW アプリのダウンロードであることが分るように「brew」を指定する
<param>	name	itemid	6byte(固定)	リンク用 Item_ID を指定
	value	itemid	10byte 以下	32bit-Number
<param>	name	param	5byte(固定)	アプリケーションが使用するパラメータ情報
	value	param	512byte	ダウンロード後アプリケーション起動する際に指定されたパラメータをもって起動する。パラメータは ASCII のみ。

※「mshop1」は小文字のみ端末で許容し、それ以外はパラメータエラーとなります。

※「name」の「itemid」「param」は小文字のみ許容し、Null であった場合やそれ以外の文字列であった場合はパラメータエラーとなります。

※また、対応パラメータ以外はすべて無視して動作します。

## ②「EVT\_APP\_START」の次に続くイベント

ISHELL\_StartAppletArgs() を利用してアプリ間で引き渡すパラメータによって「EVT\_APP\_START」の次に続くイベントが変化します。“:”(コロン)が含まれたパラメータの場合は「EVT\_APP\_BROWSE\_URL」、“(コロン)が含まれないパラメータの場合は「EVT\_APP\_BROWSE\_FILE」になります。

## (2) 「リンク用 ItemID」 (ブラウザ連携用)

Qualcomm 社→KDDI 経由で払い出される Item\_ID は 10 進で付与され「0~4294967295」数値の範囲となります。これら Item\_ID をサイト上に記述する場合は運用上、下位 8bit をオール” 0” に差し替えた上で記述しなければなりません。この下位 8bit をオール” 0” に差し替えた Item\_ID をブラウザ連携用として「リンク用 ItemID」と呼びます。ブラウザ上で Item\_ID を指定する場合はリンク用 Item\_ID を記述すよう十分ご注意ください。リンク用 ItemID は BREW アプリのバージョンアップ時や BREW アプリを動かす端末が変わった時でも同じものを使用できます。なお、KDDI からは、Item\_ID、リンク用 ItemID の両方が払い出されます。

参考までに下位 8bit を” 0” にするという例を以下に示します。

KDDI から払い出された Item\_ID が「Item\_ID=177666」であった場合、

- ( i ) まず 16 進に変換します  
177666 (Dec) → 2B602 (Hex)
- ( ii ) 次に下位 8bit (下 2 桁) を” 0” に変更します  
2B602 (Hex) → 2B600 (Hex)
- ( iii ) 再度 10 進に変換します  
2B600 (Hex) → 177664 (Dec)
- ( iv ) 177664 (Dec) をリンク用 Item\_ID としてサイトに記述します  

```
<object data="mshop1" type="brew" standby="EZ アプリを起動!">
  <param name="itemid" value="177664" valuetype="data">
  <param name="param" value="lat=+35.51.14&lon=136.39.08" valuetype="data">
</object>
```



#### 4.13.5. アラーム通知/起動 ※アラーム起動に関しては、使用を禁止させていただいています。

#### 4.13.6. SMS (C メール) 受信による起動

KDDI より提供しているセンタースッシュサービス(網起動)の SMS を受信することにより、特定の Class\_ID を指定しながらターゲット BREW アプリを起動することができます。また、任意領域 (85byte 以下) を利用し SMS から BREW アプリに対しパラメータを引き渡すことも可能です。

##### (1) SMS フォーマット

BREW 用 SMS のフォーマットは以下の通り。

項目名	記述内容	長さ	文字
メール種別	コントロールコード	3bytes	0x00 0x03 0x03[固定]
デリミタ	区切り	1byte	LF (0x0A) [固定]
App_Class_ID	BREWCclassID	8bytes	ASCII 形式
デリミタ	区切り	1byte	LF (0x0A) [固定]
任意領域	任意領域	0~85bytes	
終了コード	終了コード	2bytes	CR LF (0x0D 0x0A) [固定]

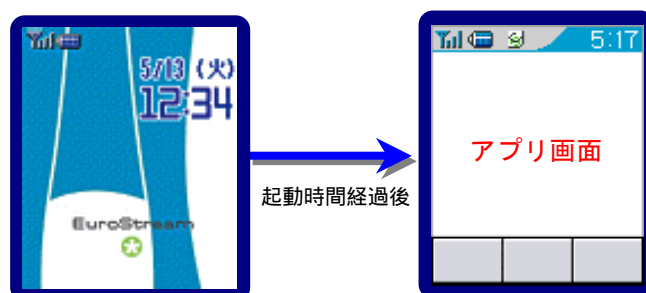
#### 4.13.7. 待受 BREW アプリ起動

MIF エディタ上で「待受 BREW アプリ」としての属性が与えられた BREW アプリは、ユーザが事前に端末上の「BREW 設定」内の「待受 BREW 設定」(1 アプリケーションまで設定可能)に設定することで、Native 待受け状態において一定時間キー操作等がない場合に自動的に起動される機能「待受 BREW アプリ」として動作することができます。BREW アプリに「待受 BREW アプリ」としての属性を付加するには、開発ツールのひとつである”MIF エディタ”上の「アプレットの詳細情報」にある「設定」にて「スクリーンセーバ」をチェックすることで可能となります。

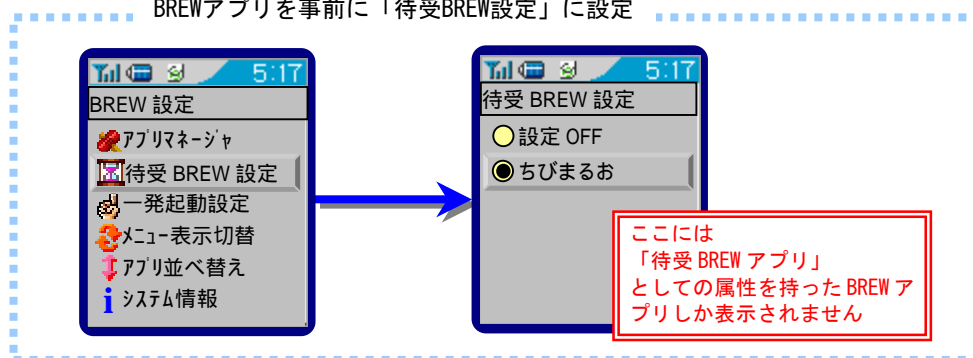
なお、待受 BREW アプリにキーイベントを全く受け付けられないような全てのキーイベントに対し FALSE を返すよう作りこんでおくことでキー操作を契機に BREW アプリは終了となり PC のスクリーンセーバと同様のイメージの「スクリーンセーバアプリ」として見せることができます。

(1) BREW アプリカタログ(ADS)からのダウンロードから登録まで

他の BREW アプリと同様にダウンロードされた「待受 BREW アプリ」は「BREW メニュー」の一覧に表示されます。これに対し、「BREW 設定」内の「待受 BREW 設定」では対象となる BREW アプリ、つまり「待受 BREW アプリ」のみが一覧として表示されこれを選択することで待受起動の対象となる「待受 BREW アプリ」を決定します。この時、「待受 BREW 設定」された「待受 BREW アプリ」は、「BREW メニュー」上の一覧に表示されなくなり、設定解除後、再度表示されるようになります。なお、ユーザ操作により「BREW メニュー」から通常起動される場合がありますので、BREW アプリを終了する為にソフトキー等に終了動作を割り当てるようにして下さい。また、サスペンド/レジューム処理も必要に応じて入れて下さい。



BREWアプリを事前に「待受BREW設定」に設定



(2) 待受 BREW アプリ実行中の割り込み

割り込み等の動作諸条件が通常の BREW アプリとは異なりサスペンドされずすべての場合アプリケーションは終了となりますので注意が必要です。「 [4. 14. 2. サスペンド\(一時停止\)](#) 」をご参照下さい。

(3) 起動時間変更メニュー

※現在、待受け BREW アプリの起動時間変更は出来ない仕様となっております。

(4) 待受 BREW アプリと Shim アプリを利用した Native 機能との連携

待受 BREW アプリの Shim アプリを利用した Native 機能(アドレス帳、URL\_to、mail\_to)との連携は、待受 BREW アプリが他の BREW アプリと連携した場合、サスペンドではなく終了動作となり戻り先の BREW アプリが終了し、なくなってしまうことから端末リセットを引き起こす可能性がありますので使用しないで下さい。

## (5) 待受 BREW アプリ実行中の C/E メール受信

以下のように E メール受信時に限り通常の BREW アプリ起動時と動作が異なりますのでご注意ください。

## (i) 通常 BREW アプリ

BREW アプリ実行中に Eメールのシグナル受信した場合は、BREW アプリはサスペンドせずに動作を継続し、BREW アプリ終了後に Native により Eメール自動受信処理動作に移行します。この時、ピクト行に Eメールアイコンを表示するのみとなりメール着信音、メール着信バイブレーションの動作は行われません。なお、Eメール受信設定にて、自動受信を設定した場合も同様の動作となります。Cメールの場合は、BREW アプリ実行中にも Cメール受信を完了し、ピクト行に Cメールアイコンを表示しますが、メール着信音、メール着信バイブレーション動作は行いません。

## (ii) 待受 BREW アプリ

待受 BREW アプリ実行中に Eメールのシグナル受信した場合は、即座に待受 BREW アプリを終了させ、Native により Eメール自動受信処理動作に移行します。受信完了後、待受に移行し一定時間キー操作が無い場合には再び待受 BREW アプリが起動します。Cメール受信時には通常 BREW アプリと同様に、待受 BREW アプリ実行中に Cメール受信を完了し、ピクト行に Cメールアイコンを表示するのみとなります。尚、メール着信音、メール着信バイブレーション動作は行いません。

	Eメール受信時		Cメール受信	
	アプリ動作	受信タイミング	アプリ動作	受信タイミング
(i) 通常 BREW アプリ	継続	BREW アプリ終了後受信処理動作に移行	継続	BREW アプリ実行中にバックグラウンドで受信
(ii) 待受 BREW アプリ	終了	BREW アプリを終了し即座に受信処理動作に移行		

## (6) 待受アプリの強制終了について

端末がパーシャルモードに遷移するタイミングで待受けアプリが強制終了する機種があります。待受アプリを提供の際はご注意ください。

#### 4.13.8. BREW アプリ起動における留意点

BREW アプリの起動について、下記の制限やガイドラインがありますのでご留意下さい。

##### (1) 起動時間

BREW サービスでは Java との差別化を意識し BREW アプリの起動における高速性を訴求しています。BREW アプリを作成するにあたり起動時間の目安として 1 秒内にアプリケーション起動できるよう起動直後の画面を簡素化する等含め方法をご留意下さい。

##### (2) 複数 BREW アプリの重複起動

BREW 実行環境では複数の BREW アプリを同時に起動させておくことが可能です。ところが、新たに BREW アプリを起動するのに必要なヒープ領域が足りない場合、BREW アプリが起動できなかったり動作が不安定となる恐れがあります。これを回避するために、なるべくヒープ領域を無駄に利用しないようにしないようにし、以下に記す②の起動時のチェック機構を必ず入れるようにしてください。

##### ①モジュール自体のロードエリア確保

端末側でチェック機構が働きます。確保できない場合には `ISELL_StartApplet()` に対し `ENOMEMORY` が返され、この時端末側で

「エラーが発生しました(002) ※クリアキーを押した後、一度電源を切り、電源を入れなおしてください」

との表示が出され終了になり BREW アプリの起動ができません。

##### ②BREW アプリが使用予定の最大ワークエリアの確保

アプリ側でチェックをする必要があります。BREW アプリを起動、もしくはレジュームを行う際に以下の手順にて事前に残メモリについてチェックし起動を制限します。このチェックがうまく機能しない場合、残容量以上のメモリを使用しようとしてしまう可能性があり不足分を補う為にサスペンドされている BREW アプリが強制的に終了されたり、動作が不安定になる恐れがありますので必ず対応するようにして下さい。

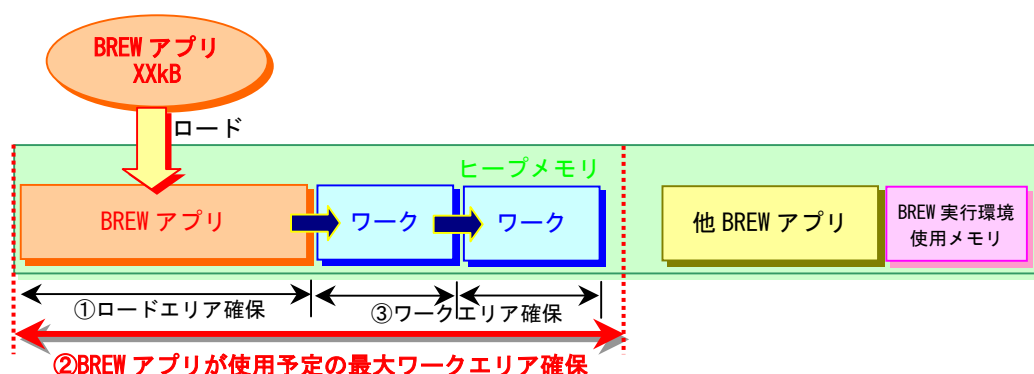
##### <手順 1>malloc 出来る最大長を調査する

`IHEAP_CheckAvail()` を利用し、アプリが `MALLOC` する最大メモリが残されているかどうかを調査します。この関数は連続した空き容量、すなわち `MALLOC` 可能なメモリを返します。特に一度に大きなメモリを取得する場合は注意して制限値を設けてください。

##### <手順 2>利用できる残メモリを調査する

`IHEAP_GetMemStats()` を利用し、アプリが利用する最大メモリが残されているかどうかを調査します。アプリが利用する最大メモリよりも大きい値が残されていることを確認してください。

本チェック機構により BREW アプリに起動を制限する場合には下記のように起動できない理由を添えたメッセージを表示し、そのまま ANY キーにてアプリを終了して下さい。なお、詳細は「プログラミングガイド」ならびに「[15.1. BREW アプリが使用予定の最大ワークエリアの確保](#)」を参照して下さい。確認画面の表示内容等の詳細については「[9.6.3. IWEB 通信時のエラー画面表示 \[エラー画面表示：必須\]](#)」を参照して下さい。



### ③モジュールのロード成功後のワークエリア確保

Shaker 使用時に限り、インスタンスが生成の為のワークエリアを確保できなかった場合、端末側でチェック機構が働き

「アプリを起動できませんでした ※クリアキーを押してください」

または

「エラーが発生しました(001) ※クリアキーを押してください」

との表示が出され終了になります。これは通常ヒープメモリが一杯になった時にサスペンドされている BREW アプリを終了させて確保するという仕様があるのに対し、サスペンドされているアプリが Shaker アプリであった場合はヒープメモリが一杯になっても Shaker アプリは終了されず残り続けるという Shaker アプリ特有の仕様が起因している為起こる現象です。

### (3) 非アクティブ状態ではいけないこと

アラーム起動で使われる "EVT\_ALARM"、SMS 受信起動で使われる "EVT\_APP\_MESSAGE" 更に "EVT\_NOTIFY" を受けられるように設定された BREW アプリは、"EVT\_NOTIFY"、"EVT\_ALARM"、"EVT\_APP\_MESSAGE" を契機にアクティブ以外の状態にある BREW アプリも ISHELL\_StartApplet() を行うことでアクティブにして画面描画や鳴音、さらには通信を行うことが可能です。ところが完全に起動させるのではなく ISHELL\_StartApplet() をせずに非アクティブ状態のまま処理をさせようと思った場合、非アクティブな BREW アプリが描画すると、既にアクティブになっている BREW アプリが描画している画面に割り込みの画面が入ることになってしまいます。また、鳴音に関しても表示されている内容と全く関係の無い音が鳴り始めることになりユーザビリティを損ねてしまいますので、"EVT\_NOTIFY"、"EVT\_ALARM"、"EVT\_APPMESSAGE" 受取時には以下に示す動作は行わないようにして下さい。

- (i) 通信処理
- (ii) 音声発呼
- (iii) UI 操作
- (iv) 音源操作
- (v) バックライト操作
- (vi) バイブレーション
- (vii) 更にサスペンドが発生するような API コール

非アクティブ状態ではいけない動作の対象となる API は以下の通りです。

インターフェース/関数名	インターフェース/関数名	インターフェース/関数名
IAclockCtl	IMediaMP3 Interface	ISoundPlayer Interface
IBTAG Interface	IMediaQCP Interface	ISOUNDPLAYER_AddRef()
IBTSDP Interface	IMediaPMD Interface	ISOUNDPLAYER_FastForward()

IBTSIOPORT Interface	IMemAStream Interface	ISOUNDPLAYER_Pause()
IControl Interface	IMenuCtl Interface	ISOUNDPLAYER_Play()
CONTROL_HandleEvent()	INetMgr Interface	ISOUNDPLAYER_RegisterNotify()
CONTROL_Redraw()	INETMGR_GetHostByName()	ISOUNDPLAYER_Release()
CONTROL_Reset()	IPosDet Interface	ISOUNDPLAYER_Resume()
CONTROL_SetActive()	IRingerMgr Interface	ISOUNDPLAYER_Rewind()
CONTROL_SetProperties()	IRINGERMGR_Play()	ISOUNDPLAYER_Set()
CONTROL_SetRect()	IRINGERMGR_PlayEx()	ISOUNDPLAYER_SetInfo()
IDateCtl Interface	IRINGERMGR_PlayFile()	ISOUNDPLAYER_SetSoundDevice()
IDATECTL_AddRef()	IRINGERMGR_PlayStream()	ISOUNDPLAYER_SetStream()
IDATECTL_EnableCommand()	IRINGERMGR_Stop()	ISOUNDPLAYER_SetTempo()
IDATECTL_HandleEvent()	IShell Interface	ISOUNDPLAYER_SetTune()
IDATECTL_Redraw()	ISHELL_Beep()	ISOUNDPLAYER_SetVolume()
IDATECTL_Release()	ISHELL_BrowseFile()	ISOUNDPLAYER_Stop()
IDATECTL_Reset()	ISHELL_BrowseURL()	ISource Interface
IDATECTL_SetActive()	ISHELL_CloseApplet()	ISprite Interface
IDATECTL_SetActiveDayMask()	ISHELL_CreateDialog()	IStatic Interface
IDATECTL_SetDate()	ISHELL_EndDialog()	ITAPI Interface
IDATECTL_SetFont()	ISHELL_ForceExit()	ITAPI_AddRef()
IDATECTL_SetJulianDay()	ISHELL_GetPosition()	ITAPI_GetStatus()
IDATECTL_SetProperties()	ISHELL_HandleEvent()	ITAPI_MakeVoiceCall()
IDATECTL_SetRect()	ISHELL_MessageBox()	ITAPI_OnCallStatus()
IDATECTL_SetTitle()	ISHELL_MessageBoxText()	ITAPI_OnCallEnd()
IDATECTL_SizeToFit()	ISHELL_Prompt()	ITAPI_Release()
IDialog Interface	ISHELL_ShowCopyright()	ITAPI_SendSMS()
IDisplay Interface	ISocket Interface	ITextCtl Interface
IDNS Interface	ISound Interface	ITimeCtl Interface
IFont Interface	ISOUND_AddRef()	IViewer Interface
IFONT_DrawText()	ISOUND_GetVolume()	IVocoder Interface
IGraphics Interface	ISOUND_PlayFreqTone()	IWeb Interface
IHtmlViewer Interface	ISOUND_PlayTone()	IWebOpts Interface
IImage Interface	ISOUND_PlayToneList()	IWebResp Interface
IImageCtl Interface	ISOUND_Release()	IWebUtil Interface
IMedia Interface	ISOUND_SetVolume()	
IMediaMIDI Interface	ISOUND_StopTone()	
IMediaMIDIOutMsg Interface	ISOUND_StopVibrate()	
IMediaMIDIOutQCP Interface	ISOUND_Vibrate()	
		IKDDI_PhoneSYSTEM

なお、EVT\_NOTIFY について ISHELL\_RegisterNotify() で設定した場合は BREW アプリが実行中(アクティブ中/サスペンド中)に限り通知可能であるのに対し、MIF エディタ上で設定した場合は BREW アプリが実行中でない場合でも起動させることが可能です(ただし、Native アプリ起動中は除く)。

#### 4. 14. BREW アプリの制御

##### 4. 14. 1. 終了動作

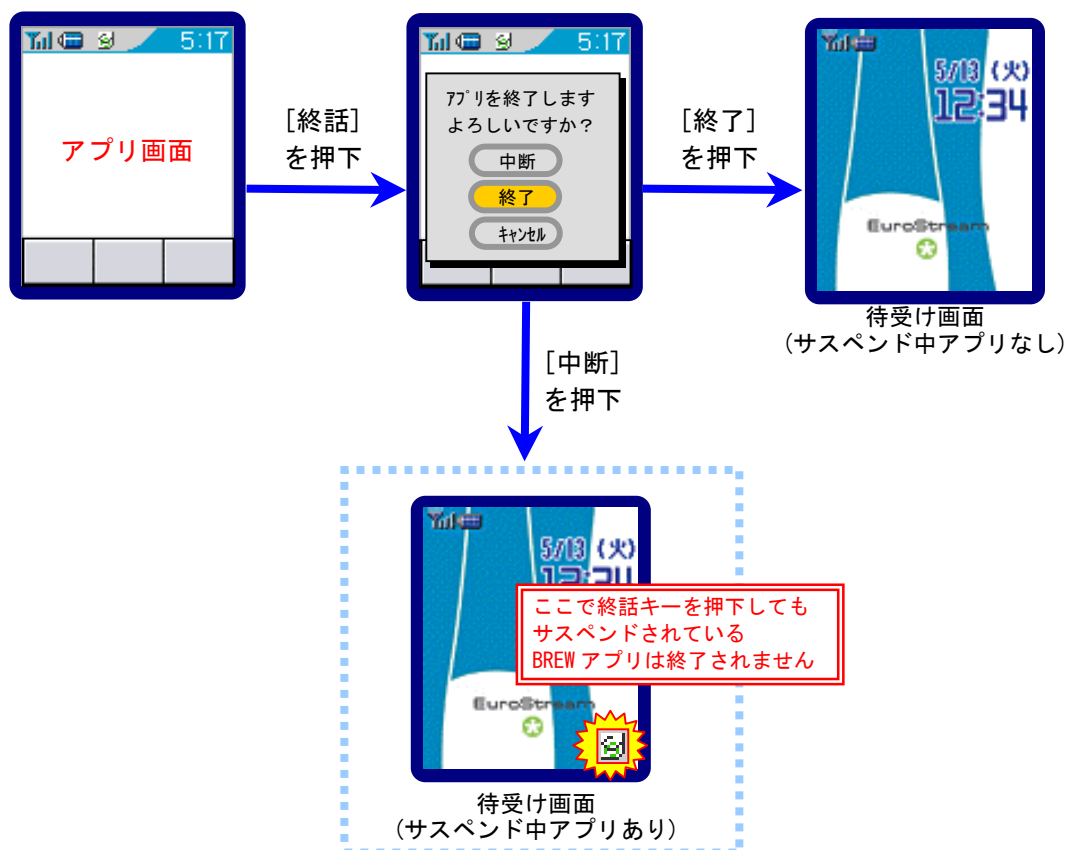
実行中の BREW アプリは以下の場合終了となります。アプリ終了時 (EVT\_APP\_STOP) には、必ず、ISHELL\_SetTimer () したコールバック関数は、キャンセル [ISHELL\_CancelTimer ()] しなければなりません。

- ① 終了動作が割り当てられたキー押下による終了
- ② 終話キー押下による終了 (強制終了)

##### (1) 終話キー押下による終了 (強制終了)

BREW アプリ実行中に終話キーを押下した場合は、ポップアップ画面が表示され、そこで [終了] を選択するとアクティブ中の BREW アプリはもちろん、サスペンド中の BREW アプリが存在する場合であっても全ての BREW アプリは終了され Native 待受画面になります。また、ポップアップ画面にて [終了] を押下するまでは、BREW アプリはアクティブなままとなります。一方、待受け画面状態においてサスペンドされている BREW アプリが存在している場合は、そのサスペンド中の BREW アプリは終了されず残ります。

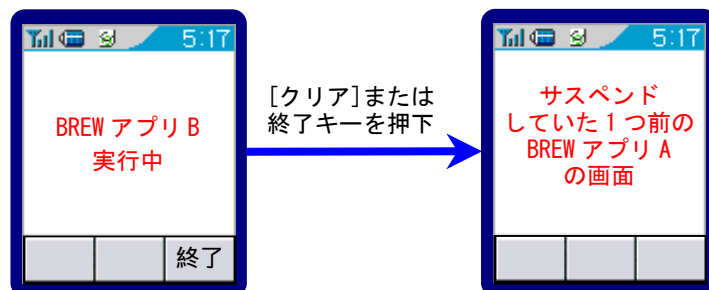
ポップアップ画面にて [中断] を選んだ場合、BREW アプリはサスペンドされます。



##### (2) 二重起動下での終了動作が割り当てられたキーまたはクリアキー押下による終了



後から起動された BREW アプリ B の終了動作が割り当てられたソフトキー等もしくはクリアキーを押下した場合、BREW アプリ B は終了となりサスペンド中の BREW アプリ A がレジュームされます。ただし、Text 入力中のクリアキーの扱いは文字の削除動作として扱われます。なお、ソフトキー等による終了動作を行う場合、ISHELL\_CloseApplet には必ず FALSE をセットして下さい。TRUE をセットすると、レジューム中のアプリも終了してしまいます。



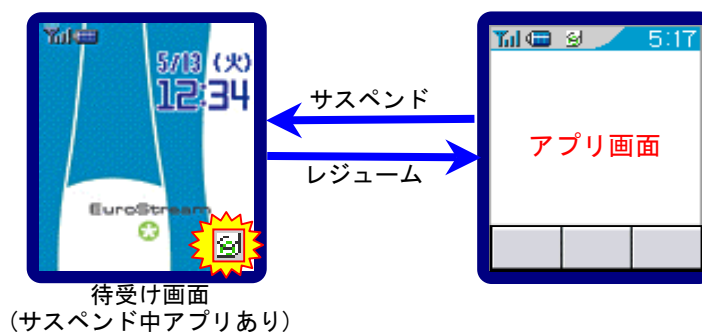
#### 4. 14. 2. サスペンド(一時停止)

BREW では、BREW アプリを終了する以外に、BREW アプリをサスペンド(一時停止)する機能を有します。ただし、この機能は待受 BREW アプリには適応されず、イベントを受け付けけないキーを押下した場合(アプリキーを含む)や他アプリケーション起動でサスペンドすることなく終了動作となります。なお、BREW アプリのサスペンドは、以下のパターンで実行されます。

サスペンド移行パターン	備考
ユーザからのキー操作	
アプリキー短押下によるサスペンド	「 2. 5. ⑥アプリキー (対応端末のみ) 」参照
アプリキー長押下からの他 BREW アプリ起動によるサスペンド	「 2. 5. ⑥アプリキー (対応端末のみ) 」参照
Native からの割り込み	
音声着信によるサスペンド	
Native アラームによるサスペンド	「M72 時刻アラーム設定」のアラーム
スケジュールアラームによるサスペンド	「M73 カレンダー/スケジュール」のアラーム
ローバッテリーアラームによるサスペンド	電池残量が残りに少なくなってきた場合のアラーム
BREW アプリからの Native アプリ連携	
BREW アプリからのブラウザ連携実行 (Shim アプリ連携)	「 4. 15. 1. ブラウザとの連携 (URL_to) 」参照
BREW アプリからのメール連携実行 (Shim アプリ連携)	「 4. 15. 2. メールとの連携 (mail_to) 」参照
BREW アプリからのアドレス帳連携実行 (Shim アプリ連携)	「 4. 15. 4. アドレス帳との連携 」参照
BREW アプリからの発呼実行	「 4. 15. 3. 発信動作 (Phone_to) 」参照
BREW 内	
SMS 受信 (BREW アプリを起動する SMS 受信)によるサスペンド	「 4. 13. 5. アラーム通知/起動」参照
BREW アプリからの他 BREW アプリ起動によるサスペンド	「 4. 15. 6. BREW アプリとの連携 」参照

##### (1) BREW アプリサスペンド中の BREW アイコン表示

待受け状態において、BREW アプリがサスペンドされている場合には待受け画面上にて BREW アイコンが何らかの方法によって表示されます。表示方法は端末依存となりますので実機にてご確認ください。



## (2) 待受 BREW アプリと Shim アプリを利用した Native 機能との連携

待受 BREW アプリからの Shim アプリを利用した Native 機能(アドレス帳、URL\_to、mail\_to)との連携の一切を禁止します。詳細は「[4.13.7. 待受 BREW アプリ起動](#)」を参照して下さい。

## (3) 割り込み競合

BREW アプリは音声着信等のすべての割り込みに備え、サスペンドをハンドリング出来るよう作成されていなければなりません。サスペンドをハンドリング出来ないと音声着信等の割り込み時に BREW アプリは終了されてしまいレジュームできません。なお、サスペンドイベントには以下の 2 つが存在し BREW アプリからはそれが判別可能ですので必要に応じて使い分け、処理を行って下さい。詳細は「BREW2.0 API リファレンス」を参照して下さい。

サスペンドイベント	説明
AEE_SUSPEND_NORMAL	他アプリの起動時などにより BREW (AEE) からサスペンド要求された場合のパラメータ
AEE_SUSPEND_EXTERNAL	音声着信等外部イベントにより OEM からサスペンド要求された場合のパラメータ

## 4.14.3. レジューム(再開)

BREW アプリは、サスペンド中からレジューム(再開)する機能を有します。サスペンド(一時停止)となったアプリケーションは、連携先のアプリケーションが終了した時点で自動的にレジューム(再開)されます。但し、サスペンド中に何らかの理由で BREW 実行環境からサスペンドされていた BREW アプリが終了された場合は、初期状態で再起動されます。また、「[4.14.1. 終了動作](#)」で述べた終話キーによる強制終了を実行した場合は、Active であった BREW アプリの他にサスペンドしていた BREW アプリも同時に終了され待受け画面へ遷移します。なお、BREW アプリのレジュームは、以下のパターンで実行されます。

サスペンド移行パターン	備考
ユーザからのキー操作	
アプリキー短押下によるレジューム	「 <a href="#">2.5. ⑥アプリキー(対応端末のみ)</a> 」参照
Native アプリの割り込みからの復帰	
音声通話終了後の自動レジューム	
Native アラーム終了処理による自動レジューム	「M72 時刻アラーム設定」のアラーム
スケジュールアラーム終了処理による自動レジューム	「M73 カレンダー/スケジュール」のアラーム
ローバッテリーアラーム終了処理による自動レジューム	電池残量が残りに少なくなってきた場合のアラーム
BREW アプリからの Native アプリケーション連携起動によるサスペンド	
BREW アプリからのブラウザ連携実行(Shim アプリ連携)	「 <a href="#">4.15.1. ブラウザとの連携(URL_to)</a> 」参照
BREW アプリからのメール連携実行(Shim アプリ連携)	「 <a href="#">4.15.2. メールとの連携(mail_to)</a> 」参照
BREW アプリからのアドレス帳連携実行(Shim アプリ連携)	「 <a href="#">4.15.4. アドレス帳との連携</a> 」参照
BREW アプリからの発呼実行	「 <a href="#">4.15.3. 発信動作(Phone_to)</a> 」参照
BREW 内	
他 BREW アプリ終了によるレジューム	「 <a href="#">4.14.1. 終了動作</a> 」参照
他 BREW アプリから起動要求によるレジューム	「 <a href="#">4.15.6. BREW アプリとの連携</a> 」参照

#### 4. 14. 4. サスペンドとレジュームにおける留意点

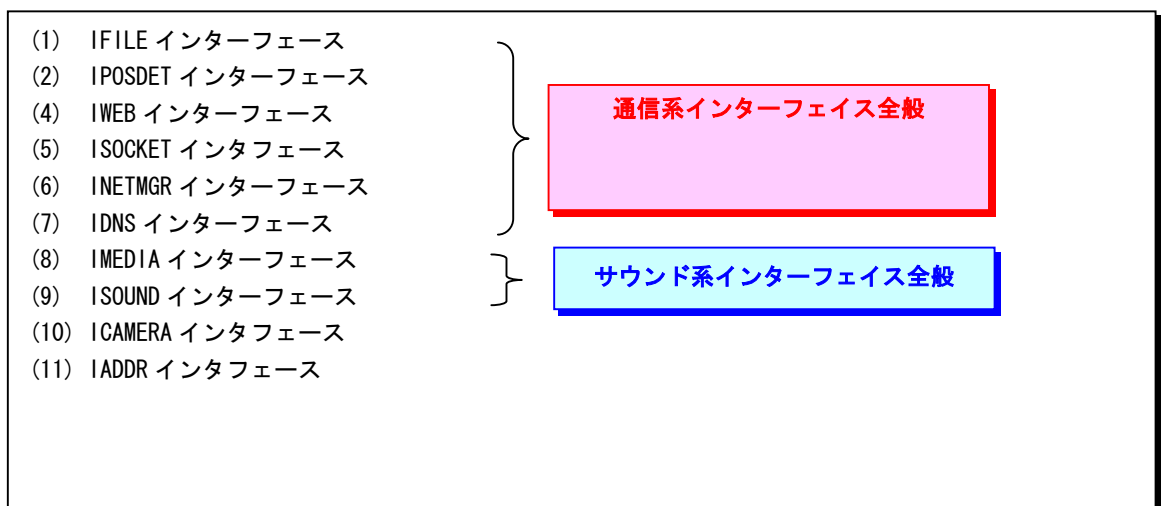
BREW アプリは実行中に何らかの割り込みにより、現在実行中の BREW アプリにサスペンドイベント (EVT\_APP\_SUSPEND) が送出されてくることがあります。この時の留意点について以下に記します。

##### (1) サスペンドイベント前にしなければならない処理

アプリへ通知される"EVT\_BUSY"への返却値として"FALSE"を返さなければいけません。何等かの BREW アプリ A が起動している状態で他の BREW アプリ B が起動する場合には、先に起動していた BREW アプリ A にまず"EVT\_BUSY"が通知されます。"EVT\_BUSY"に"FALSE"を返すと"EVT\_APP\_SUSPEND"が通知され正しくサスペンド動作に移行できますが、"EVT\_BUSY"に"TRUE"を返すと同アプリへは"EVT\_APP\_SUSPEND"が通知されず延々と動作継続できてしまう為、SMS 契機起動など他 BREW アプリが起動出来なくなってしまいます。

##### (2) サスペンドイベント中にしなければならない処理

①以下のインスタンスを使用している場合は、必ず解放して下さい。



②以下の処理をしている場合は、必ずその処理を終了(停止)して下さい。

- (i) サウンド(メロディ、ビープなど)
- (ii) アニメーション
- (iii) バイブレート
- (iv) バックライト(※)
- (v) ISHELL\_SetTimer() したコールバック関数(キャンセルは ISHELL\_CancelTimer())
- (vii) ファイル処理中[読み込み/書き込み時](オープンされているファイルはクローズする)

※時間のかかる処理は極力避け、すみやかにサスペンドするようにして下さい。

※ファイル処理等においてはこれら処理と共に必要に応じて保存等を施すようにして下さい

※バックライトは長時間点灯しつづけると電池消耗が激しく消費される為「4. 9. バックライト」の規定に従うようにして下さい。

③以下のインスタンスを使用している場合は必ず保持してください。(解放しないで下さい。)

- (1) IFILECP インターフェース
- (2) ITEXTCTL インターフェース

※IFILECP を解放すると、Native からの連携エラーで端末がリセットする可能性があります。  
 ※ITEXTCTL を解放すると、機種によっては入力された文字の戻り先がなくなってしまう、不具合が生じることがあります。  
 ※機種によっては ITEXTCTL を解放する必要があります。

(3) レジュームイベント中にしなければならない処理

- ①レジューム要求” EVT\_APP\_RESUME” に対し、返却値として” 必ず” TRUE” を返して下さい。” FALSE” を返却すると” EVT\_APP\_STOP” が送出され BREW アプリは終了してしまい、BREW アプリは再起動されてしまいます。
- ②以下の動作/処理を入れるようにして下さい。

- ( i ) 画面再描画
- ( ii ) その他復帰処理

(4) アクティブ/サスペンド状態での起動要求に対する送出イベント

BREW アプリがアクティブもしくはサスペンド状態において当該 BREW アプリを各種起動方法にて起動しようとした場合に送出されるイベントを下記に纏めます。これらを元に正しい動作を行うようにして下さい。

BREW アプリの状態	送出イベント	備考
アプリキー長押下による一発起動(「 2. 5. ⑥アプリキー(対応端末のみ) 」参照)		
アクティブ	通知なし	
サスペンド	“EVT_APP_RESUME “	
SMS 受信による起動(「 4. 13. 5. アラーム通知/起動 」参照)		
アクティブ	“EVT_APP_MESSAGE “	
サスペンド	“EVT_APP_RESUME “+ “EVT_APP_MESSAGE “	
WAP ブラウザ連携による起動(「 4. 13. 4. WAP ブラウザからの起動 」参照)		
アクティブ	———	
サスペンド	“EVT_APP_RESUME” +” EVT_APP_BROWSE_URL” “EVT_APP_RESUME” +” EVT_APP_BROWSE_FILE”	パラメータ内に一文字でも’ : ’ (コロン) の文字が含まれている場合は “EVT_APP_BROWSE_URL”、’ : ’ の文字が全く含まれない場合は“EVT_APP_BROWSE_FILE”が発生します。

#### 4.14.5. スリープ

BREW アプリがスリープとなるには、まず、端末がスリープとなる条件を満たしている必要があります。端末がスリープとなる条件を満たした上で、BREW アプリが AEE 層より通知される EVT\_APP\_NO\_SLEEP イベントに対して、FALSE を返却した場合スリープ動作に移行します。

スリープモードは電池消耗等を考慮した動作ですので、基本的には BREW アプリは端末に従い、スリープモードに入る作りとしてください。

##### BREW アプリがスリープになる条件

- ①端末がスリープになる条件を満たしている
- ②EVT\_APP\_NO\_SLEEP に対して FALSE を返却した場合
- ①、②を共に満たしている場合のみスリープに移行

##### (1) 端末がスリープとなる条件

①端末 LCD の照明が ON の状態では、端末のスリープには移行しません。スリープに移行するには端末 LCD の照明がオフである必要があります。従って、パーシャル中も照明が OFF となる為パーシャル中もスリープの条件を満たすことになります。

②また、アプリが通信を行っている場合もスリープには移行しません。通信を行うインスタンス全てが解放されている必要があります。

③IMEDIA 系のインスタンスが作成されている場合はスリープに移行しません。端末がスリープに移行するためには IMEDIA 系のインスタンス全てがリリースされている必要があります。

##### (2) スリープからアクティブへの復帰契機

実行中の BREW アプリがスリープ状態となった場合、スリープ状態からアクティブ状態に復帰する契機は以下のパターンのいずれかとなります。

##### BREW アプリがスリープから復帰する契機

- ①ユーザによるキーイベント通知（但しフリップ開状態のみ）
- ②フリップ開閉時
- ③SMS 受信時
- ④BREW アプリによる復帰  
(EVT\_ALARM でアラーム起動し ISHELL\_StartApplet で自らをアクティブ)
- ⑤通信開始時
- ⑥鳴音開始時

#### (4) スリープ中のタイマ誤差

BREW アプリがスリープ状態になった場合、通常状態でのタイマ動作とスリープ状態でのタイマ動作に誤差が生じてしまいます。この為、BREW アプリは、スリープ状態でのタイマ動作に誤差が発生することを十分考慮しなければなりません。

#### 4. 14. 6. フリップ開閉時の BREW アプリ制御

BREW アプリ実行中にフリップを閉じても BREW アプリは終了することなく動作し、フリップ開閉時の状態は EVT\_FLIP(TRUE/FALSE)にて BREW アプリに通知されます。この開閉 Event をうまく利用して BREW アプリに変化を持たせることが可能です。また、フリップを閉じた状態での BREW アプリの SMS 起動、待受 BREW アプリ起動、BREW アプリによるデータ通信や音の鳴音も可能です。

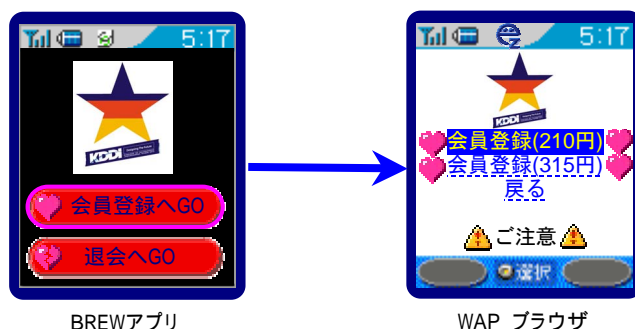
なお、フリップが閉じた状態で BREW アプリが起動した場合は BREW アプリが起動したことをサブディスプレイ上に表示されます。

## 4. 15. アプリ連携

BREW アプリから Native アプリと連携することが可能です。

### 4. 15. 1. ブラウザとの連携(URL\_to)

ISHELL\_BrowseURL で ezurlto を指定すると、接続先 URL を指定して Native 実装ブラウザである WAP ブラウザが起動されます。指定できる URL 長は NULL 文字を含まずに最大 1024byte となり上限サイズ以降を切り捨てになります。EZweb 上での会員管理等に有効です。



#### (1) WAP ブラウザから BREW アプリへの復帰

WAP ブラウザ遷移後起動元の BREW アプリに戻る場合はコンテンツ側のタグで

```
device:kddibrowser-end(ブラウザから直前の Native 機能に戻るコマンド)
```

を記述しユーザに選択させることで確実に戻ることができます。ブラウザ上での「クリアキー」押下では WAP ブラウザの仕様上、直前のキャッシュを表示することになります。なお、複数 Native/BREW アプリ連携状態での終話キー押下やキャッシュがない場合のクリア押下等ある特定のシチュエーションにより起動前の BREW アプリ画面に戻る場合がありますが、条件によって異なるものであり確実性の薄いものになりますので BREW アプリに戻る場合は必ず本仕様を使用するようにして下さい。

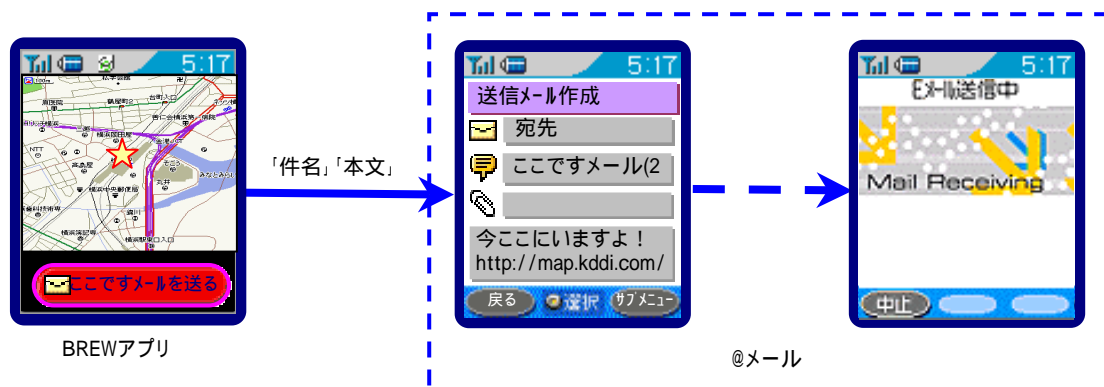
なお、WAP ブラウザからの BREW アプリ連動については、「[4. 12. 3. WAP ブラウザからのダウンロード](#)」「[4. 13. 4. WAP ブラウザからの起動](#)」を参照して下さい。

#### 4. 15. 2. メーラとの連携(mail\_to)

ISHELL\_BrowseURL で atmailto を指定すると、送信先アドレス (To/Cc/Bcc)、題名、本文を指定して Native 実装メーラである@メーラが起動されます。それぞれ引き渡し可能な文字列の最大バイト数は下記の通りとなります。指定された送信先、題名、本文が最大バイト数を越えた場合、上限サイズ以降を切り捨て、送信先では切り捨てた結果アドレスが不正の場合にはそのアドレスは切り捨てます。

項目		最大バイト数
送信先アドレス	To	合計で 640 バイト
	Cc	
	Bcc	
件名 (Subject)		100 バイト
本文		10,000 バイト

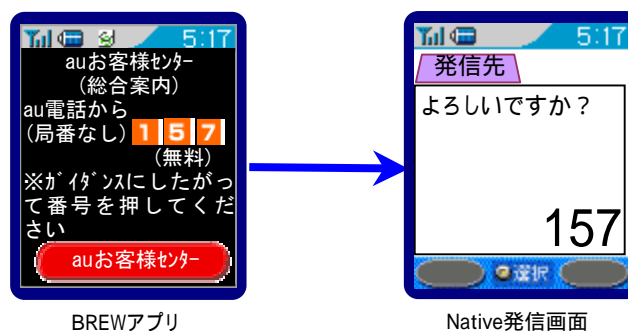
※複数アドレスの区切りは、カンマ “,” ごととします。なお、カンマも 1 バイトとして計算します。



#### 4. 15. 3. 発信動作 (Phone\_to)

Native アプリケーションとの連携とは異なり Shim アプリを介さず ITAPI\_MakeVoiceCall() を利用することで最大 36 桁の発信動作とも連携が可能です。発呼処理は「184」、「186」や「\*、#、-(半角)、P」(P: 半角大文字のみ) が付いていれば発信動作はそれに従い、Native で発信番号通知の設定がされていればそれに従い発信します。

なお、ユーザへの発信動作確認は Native にて表示されます。





## 4. 15. 4. アドレス帳との連携

## (1) Shim 連携によるアドレス帳連携

ISHELL\_BrowseURL で Addrbook を指定すると Native のアドレス帳が起動し、ユーザ操作を挟んで選択した 1 レコード分の電話番号など値を取得することができます。なお、BREW アプリからのアドレス帳連携は参照のみで、登録、削除はできません。

## 検索モード

検索モードが指定可能で各々の属性は以下の通りとなります。また、これらの属性を纏めて全てを取得することも可能です。なお、Native アドレス帳検索時のシークレット設定されたデータは、M 機能のシークレットモードが ON の場合にはデータ検索表示しません。

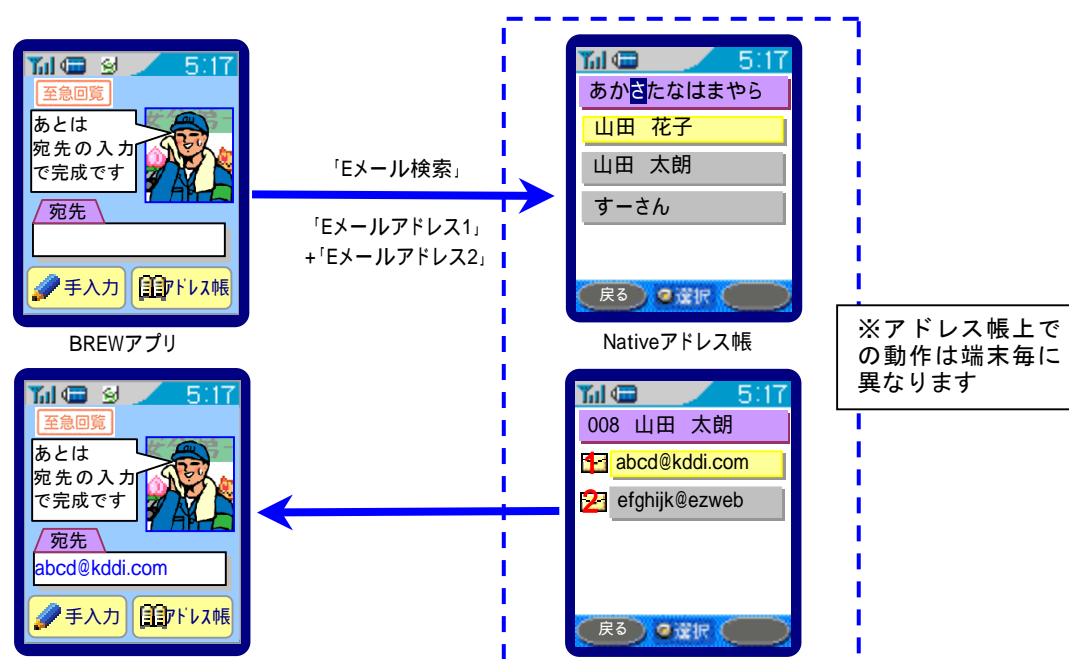
	検索モード	内容
0	通常起動	通常のアドレス帳を起動するモード
1	電話番号検索	電話番号で検索するモード
2	E メール検索	E メールアドレスで検索するモード
3	URL 検索	URL で検索するモード
4	GPS 検索	GPS 情報で検索するモード

## ・取得可能項目

検索の結果、Native のアドレス帳から取得する項目を指定することができます。複数の項目を指定する場合には項目を OR (|) で繋ぎます。指定できる項目は以下の通りです。

Hex	項目
0x0001	名前
0x0002	読み
0x0004	電話番号 1
0x0008	電話番号 2
0x0010	E メールアドレス 1
0x0020	E メールアドレス 2
0x0040	グループ No
0x0080	メモリ No

Hex	項目
0x0100	誕生日
0x0200	趣味
0x0400	住所
0x0800	URL
0x1000	メモ
0x2000	位置情報



## (2) IAddr インターフェースによるアドレス帳連携

BREW2.1 から追加となる IAddr インターフェースにより、BREW アプリからの Native アドレス帳操作が以下の通り Shim アプリ連携によるアドレス帳操作より機能が拡大されています。

- ① アドレス帳各パラメータへの READ/WRITE。(Shim 連携では READ のみ)
- ② アドレス帳の U/I を BREW アプリで作成可能。
- ③ アドレス帳の画像および第3アドレス（電話番号、メールアドレス）までの取得。



Shim 連携画面遷移イメージ



IAddr インターフェースでの画面遷移イメージ

## (3) ユーザ意思確認画面の表示

BREW アプリが IAddr インターフェースによるアドレス帳連携を利用する場合は（Shim 連携および NativeUI 連携のみを利用する場合を除く。）、ユーザにその実行に対する意思を確認するための画面をアプリケーション側で表示して下さい。詳細は、9章を参照して下さい。

## 4.15.5. データフォルダとの連携

BREW アプリ内データのデータフォルダへの保存や、データフォルダ内データのアプリ内への読み込みが可能です (IFileCp インターフェース)

## (1) データフォルダ連携によるファイル読み込み

読み込みには、「BREW アプリ内で行う方法」と「NativeUI との連携で行う」2通りの方法が可能です。

## 読み込み連携イメージ



## (2) データフォルダ連携によるファイル書き込み (保存)

NativeUI と連携することなく、データフォルダへのファイル書き込みが可能です。

## データフォルダへの保存イメージ



## &lt;注意事項&gt;

保存ファイルと同名ファイルが既にデータフォルダに存在していた場合は、アプリ側から指定したファイル名がNativeにて変更(変更ロジックについては移動機依存。例: “指定ファイル名+001”)されて保存されますのでご注意ください。

BREW アプリの動作に必須となるデータ以外の保存は行わないで下さい。

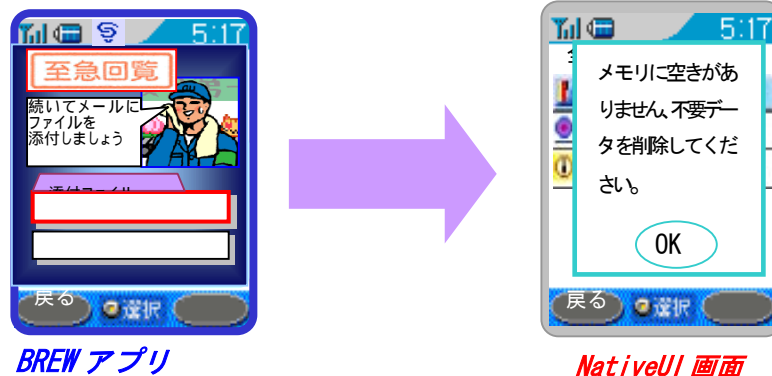
読み込み／書き込みが可能なファイルあたりのサイズ上限は移動機依存となります。

## (3) データフォルダ連携によるファイル削除

BREW アプリからファイル名指定等により直接ファイル削除を行うことは出来ません。このためファイルの上書きも出来ません。

ただし、ファイル保存時にメモリ容量が不足していた場合は、Native 側で UI へ遷移しユーザが不要データを削除することで容量が確保されることになります。

## メモリ容量不足時の連携イメージ



## (4) ユーザ意思確認画面の表示

BREW アプリがデータフォルダ連携を利用する場合は(NativeUI 連携のみを利用する場合を除く。)、ユーザにその実行に対する意思を確認するための画面をアプリケーション側で表示して下さい。詳細は、9章を参照して下さい。

## (5) 著作権

著作権のあり／なしに係わらずすべてのデータの書き込み（保存）が可能です。  
著作権ありのデータは読み込み不可能です。

## 4. 15. 6. BREW アプリとの連携 (BREW アプリ連携)

BREW アプリ起動中に他の BREW アプリを呼び出すことが可能です。

**※ただし、BREW2.0 では待受け BREW アプリにて ISHELL\_StartApplet を利用したアプリ間連携を行うことはできません。ご注意ください。**

## (1) 連携起動時の ClassID の取り扱いについて

連携起動で呼び出す BREW アプリは ClassID で指定を行いますが、呼び出す BREW アプリの ClassID は、CP にて設置したサーバから通信により取得するようにし、固定の ClassID を BREW アプリ内にもたせないようにしてください。もし固定で ClassID を持たせた場合、呼び出し先の ClassID が変更になったり、呼び出し先アプリが追加されたりすると、呼び出し元の BREW アプリも修正が必要となるためです。

## (2) 単独起動時の動作について

連携起動によって呼び出される BREW アプリは単独でも起動し、サービス提供ができるようにして下さい。

## (3) 連携起動時の呼び出し元アプリについて

呼び出し元 BREW アプリが他の BREW アプリを正常に呼び出した場合、呼び出し元の BREW アプリは必ず終了させてください。

(4) 連携先アプリが存在しない場合の動作について

呼び出し先の BREW アプリが存在しない場合は、ブラウザ連携を行い、ダウンロードページを表示するか、ダウンロードの方法について表示を行う等ユーザを誘導してください。アプリカタログからダウンロードした場合、ユーザはそのアプリが連携を行うことについての認識がないことが想定されるため、連携するアプリをダウンロードできるように誘導する必要があるからです。

#### 4.15.7. アプリからのテキスト入力

BREW2.0 では待受け BREW アプリより Native の文字入力画面へ遷移させる動作は行わないでください。待受 BREW アプリに於いて、アプリより Native 文字入力画面に遷移した状態で着信を受けると端末動作が不安定になる場合があります。待受け BREW アプリにて文字入力を行う場合は、アプリ内で文字入力画面を設けていただきますよう、お願い致します。なお、BREW2.1 以降では上記現象は発生しません。

#### 4.16. BREW アプリからのデータ通信

BREW アプリから、INETMgr や ISocket 等を利用することで様々なデータ通信を自由に行うことが可能です。HTTP 通信実行は BREW アプリよりユーザへの HTTP 通信実行確認を行いユーザからの実行許可により実行し、HTTP 通信実行確認は全て BREW アプリ側でコントロールします。HTTP Request Header については特に規定しておりませんが、実行環境により必ず以下の 2 つの Header は付加されます。

- ・ Connection :keep-alive
- ・ User-Agent :BREW/2.0.0.39 (built Nov 18 2002)

※User-Agent は「BREW+Version+BuildDate」

※エミュレータでの Version は” 2.0.0.38”

##### (1) BREW 実行環境に実装されている SSL 機能

BREW 実行環境に実装されている SSL 機能は以下の通りです。

- ①公開鍵暗号方式 : RSA
- ②共通鍵暗号方式 : RC-4 128bit のみ
- ③ハッシュ : SSL 証明書においては、MD2 のみ (SSL の通信アルゴリズム内では、MD5、SHA-1 も使用可)
- ④搭載するルート証明書 :
  - Thawte Server CA
  - Thawte Premium Server CA
  - Verisign Class3 Primary CA
  - Verisign Class2 Primary CA
  - RSA Secure Server Certification Authority
  - Equifax Secure Certificate Authority (但し、A5306ST 以降)
  - Equifax Secure eBusiness CA-1 (但し、A5306ST 以降)

##### (2) SSL 通信時の「鍵マーク」アイコン表示

SSL 通信時の「鍵マーク」アイコンに相当する表示は BREW アプリの表示内で行わなければなりませんのでご注意ください (Native による SSL での HTTP 通信実行中のアイコン表示はされません)。

※BREW アプリからのピクト表示操作は出来ません。

##### (3) BREW アプリ通信制限

データ通信実行は、「BREW アプリ通信制限」(M44X)で制限 (OFF : 送信可能、ON : 送信不可)され、BREW アプリからの「BREW アプリ通信制限」の一時解除はできません。

##### (4) サスペンド時のインスタンス解放

通信を行っている BREW アプリをサスペンドする際には、次に起動される BREW アプリや Native アプリケーションが通信を行おうとした場合競合が発生してしまう為、必ずインスタンスを解放した上でサスペンドするようにしなければなりません (アドレス帳連携時は除く)。

#### 4. 16. 1. 通信エラー処理の実装

##### (1) IWeb インターフェースにおけるエラーの通知とその処理

IWeb インターフェースにおいて、通信開始時および通信中に発生するエラー情報は、IWEB\_GetResponse の引数で指定した IWebResp オブジェクトに格納されます。IWeb インターフェースを使用する BREW アプリは、IWEB\_GetResponse で設定されたコールバック関数の呼び出し時に IWebResp オブジェクトに格納されたエラー情報を元にユーザに対してエラー通知をしなくてはなりません。この IWebResp オブジェクトからエラー情報を取得するには、IWEBRESP\_GetInfo 関数を使用します。BREW アプリがこれらの状態に関わるエラーを受領した場合にはユーザに通信が正常に行えない状態である旨の確認画面の表示を行って下さい。確認画面に関するガイドラインは「[9. 6. 3. IWEB 通信時のエラー画面表示 \[エラー画面表示：必須\]](#)」を参照して下さい。

##### (2) コネクトタイムアウトの設定

コネクトタイムアウトは、通信の接続が確立されるまでに BREW アプリが許容する最大待ち時間です。BREW AEE は、通信の接続が確立していない状態で BREW アプリから通信が要求されると、このタイムアウトの期間が満了するまで接続の確立を継続して試みます。この値のデフォルトは 30 秒です。この値は、IWEB\_AddOpt 関数にて WEBOPT\_CONNECTTIMEOUT キーを使用することで変更できますが、一般的な BREW アプリでは、特別な理由がない限りこの値を変更しないでください。また、接続の確立中にエラーが発生した場合も、IWebResp オブジェクトにそのエラーが格納されます。BREW アプリは IWEB\_GetResponse 関数で指定したコールバック関数内でこのエラーを参照することができます。BREW アプリがこれらの状態に関わるエラーを受領した場合にはユーザに通信が正常に行えない状態である旨の確認画面の表示を行って下さい。

##### (3) IWEB におけるエラーメッセージ

上記(1)(2)において得られるエラーは以下の表が全てとなりますが「BREW アプリ通信制限 (M444)」時に得られるものと「圏外/弱電界下」に得られるものとは事実上「DNS キャッシュがない場合」のみしか 2 者の見分けが付かずその他の状態においてはどちらが要因で通信が出来なかったかを知る術がありません。よってこの時に表示する確認画面は両者を含むような内容として下さい。確認画面に関するガイドラインの詳細は「[9. 6. 3. IWEB 通信時のエラー画面表示 \[エラー画面表示：必須\]](#)」を参照して下さい。

	BREW アプリ通信制限中	圏外/弱電界下
IP アドレスを指定した場合	"WEB_ERROR_CONNECT"	"WEB_ERROR_CONNECT"
DNS のキャッシュがある場合	"WEB_ERROR_CONNECT"	"WEB_ERROR_CONNECT"
DNS のキャッシュがない場合	"AEE_NET_GENERAL_FAILURE"	"AEE_NET_ENETDOWN"



#### (4) IWEB\_GetResponse のタイムアウト処理

IWEB を利用して通信する BREW アプリにおいて、弱電界下やネットワークでの輻輳による通信不成立状態が発生した場合に IWEB\_GetResponse のコールバック関数が呼ばれないことがあり、通信状態を表す BREW アイコンの点滅状態に陥り終話キー押下等するまでこの状態が継続されてしまいます。よって、この状態から脱する為、BREW アプリが独自にタイマを用意して、コールバック関数のタイムアウト処理を行う必要があります。

コールバック関数のタイムアウト処理を設定するには、ISHELL\_SetTimer 関数を使用します。ここで、設定するタイマ値は、必ずコネクトタイムアウトの設定値よりも大きくします。タイマ値がコネクトタイムアウトの設定値より小さいと、コネクトタイムアウトより前に IWEB のタイムアウトが発生し、内部的な不整合が発生する恐れがあります。IWEB\_GetResponse のコールバックタイマの値をコネクトタイムアウトのデフォルト”30 秒”のまま変更しない場合には、31~60 秒を推奨します。最後にコールバック関数のタイムアウトが発生した場合、ユーザーにその旨を伝えるエラー表示を行います。以下に本動作を実現するためのアプリケーション実装例を示します。

なお、一般的に、このタイムアウト処理は、ひとつの HTTP リクエスト（IWEB\_GetResponse の呼び出しとそれに対するコールバックの組み合わせ一つ）ごとに行うべきものですが、複数の HTTP リクエストに対してタイムアウト処理をひとまとめにすることも可とします。いずれの場合でも、IWEB\_GetResponse のコールバックが呼ばれない可能性があることを認識し、アプリケーションが不正な挙動をしないように実装を行ってください。

##### ① コールバック関数のタイムアウト用タイマーの設定

IWeb インターフェースを使用するアプリケーションは、IWEB\_GetResponse を呼び出した直後に、コールバック関数のタイムアウト用タイマーをセットします。このエラー表示用タイマの設定は、ISHELL\_SetTimer 関数を使用して行います。

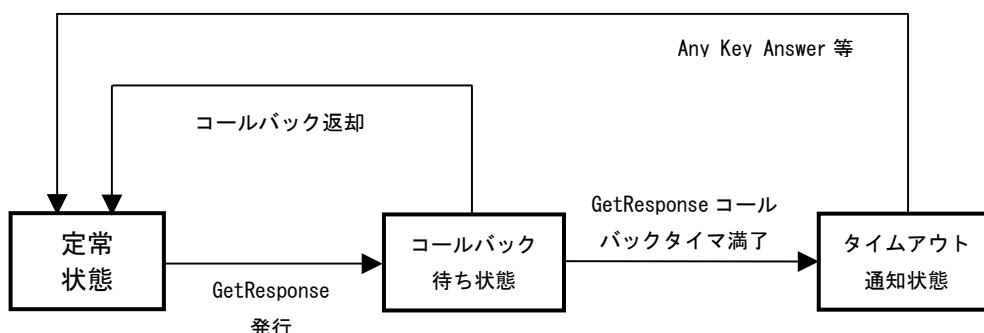
##### ② IWEB\_GetResponse 関数のコールバックタイムアウト時の処理

ISHELL\_SetTimer 関数で指定したタイマーが満了した場合、通信のタイムアウト処理を行います。アプリケーションは、このタイムアウト用コールバック関数内で IWEB\_GetResponse 関数のコールバックをキャンセルし、ネットワークリソースの解放を行います。その後ユーザーにリクエストがタイムアウトしたことを示すエラー表示を行います。

##### ③ 正常通信終了時のタイマのキャンセル処理

IWEB\_GetResponse のコールバックがタイムアウト満了前に呼び出されたとき、上記のコールバック待ちタイマをキャンセルする必要があります。ISHELL\_SetTimer 関数で設定したタイマをキャンセルするには、ISHELL\_CancelTimer 関数を使用します。

以下に、この常時接続アプリケーションの状態遷移の例を示します。



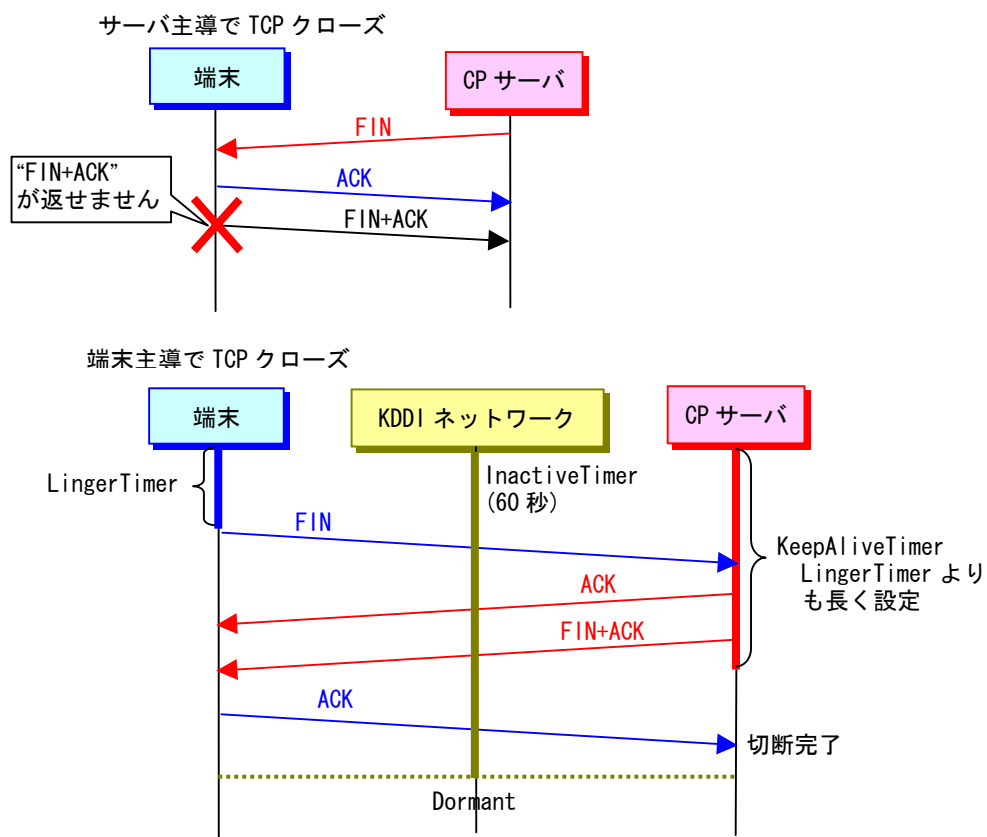


#### 4.16.2. 通信 (TCP コネクション) の終了動作

BREW 対応端末では、自ら FIN を送信して TCP をクローズに移行することは可能ですが、サーバからの FIN に対し正常に応答することができずサーバに対し FIN+ACK が返せません。よって準正常動作をシビアに考慮しなくてはならなくなる為 TCP のクローズは常に端末主導で行うようにして下さい。

端末主導で TCP をクローズするには LingerTimer 満了時間を CP サーバの KeepAliveTimer より短く設定することで実現できます。LingerTimer は SocketWrite もしくは SocketRead からカウントし始め、次の SocketWrite もしくは SocketRead でリセットし再度カウントし始めます。ただし、"60 秒"で網側の InactiveTimer により強制的に Dormant に落とされてしまい、TCP が残ったまま Dormant 移行してしまいますので LingerTimer 値をデフォルトの"30 秒"として、変更を行わないで下さい。

なお、IWEB インターフェース利用時では LingerTimer 満了時に端末により Socket をクローズしますが、ISOCKET インターフェース利用時では LingerTimer 満了時まで ISOCKET インターフェースを Release しておく必要がありますのでご注意ください。

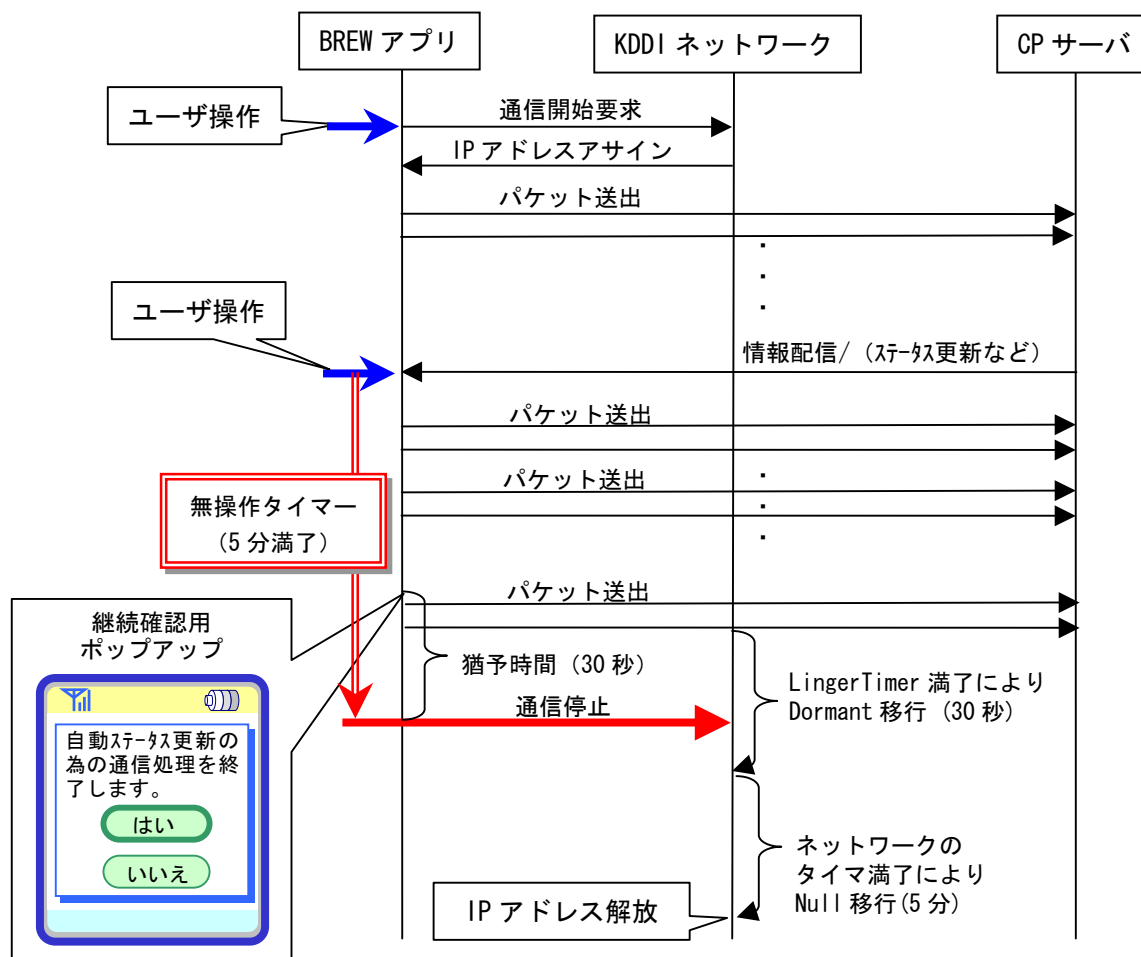


#### 4.16.3. 「常時接続型アプリ」に関するガイドライン

“インスタントメッセージアプリ”や“待ち受け型アプリ”などにおいては、ユーザの操作が無くてもステータスの更新やメッセージの受信が出来る事が望まれます。その為には、ポーリング等定期的に CP サーバへの接続を行う、もしくは BREW 端末にアサインされている IP アドレスを識別子として、サーバからの Push 型情報配信を行う事によりこれら「常時接続型アプリ」を実現する事が可能です。

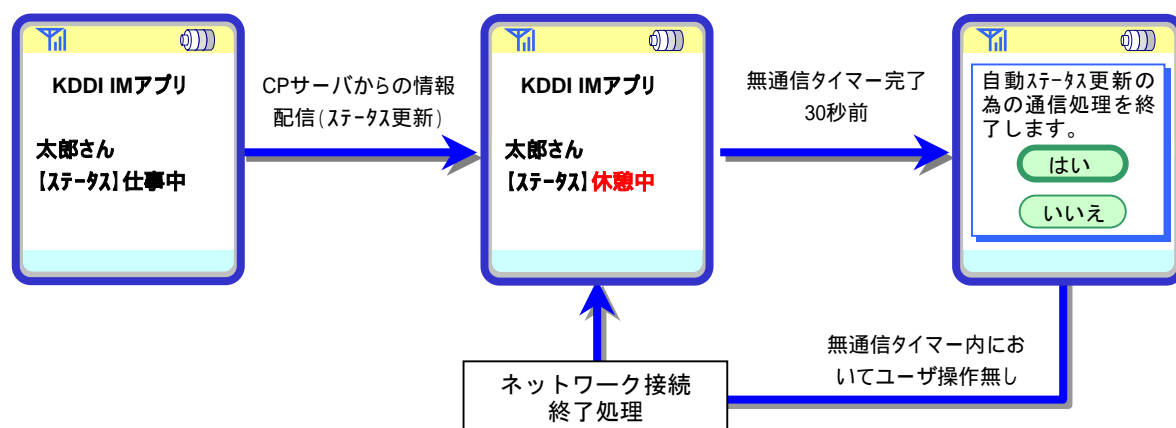
しかし、上記のような「常時接続型アプリ」は、ユーザに対する不要な通信料の負担を強いる事や、不必要な無線リソースの確保等 KDDI ネットワークインフラへの影響が懸念されます。

よって、一定時間内に無操作状態が続いた場合はユーザへネットワーク接続を継続するか否かの「継続確認用ポップアップ」画面を表示し、ユーザの許可が得られない場合にはその後の接続保持の為の packets 送出等ユーザ操作を契機とした通信の再開以外を一切禁止します。なお、この packets 送出を停止するまでの無操作タイマー時間を「5 分間」と規定します。また、「継続確認用ポップアップ」の表示はタイマ満了前の 30 秒間、つまり無操作タイマ 4 分 30 秒移行の猶予時間に必ず表示しなければなりません。



## (1) 「継続確認用ポップアップ」表示

通信継続を確認する為の「継続確認用ポップアップ」が無操作タイマ満了前の猶予時間 30 秒間に表示されたにも関わらず、タイマー完了時までユーザ操作が無い場合は、ネットワーク接続終了処理を行います。ポップアップの表示内容については、アプリの種類によって利用者へのメッセージが異なる事が想定される為、コンテンツプロバイダにより自由に作成いただく事が出来ます。



## (2) 無操作タイマの実装例

「常時接続型アプリ」における無操作状態からの 5 分経過後の通信タイムアウト処理について、以下に実装例を示します。なおこれらは本規定における実装の一例であり、コンテンツプロバイダの自由な実装を妨げるものではありません。

## ① 無操作タイマの設定

キー操作が発生するタイミングで「継続確認用ポップアップ」の表示タイマ (4 分 30 秒=270000ms) をセットします。タイマの設定は、ISHELL\_SetTimer() を使用して行います。次に、ポップアップ表示用のコールバック関数を実装します。アプリは、このコールバック関数内でユーザに注意を促すポップアップを画面上に表示し、「継続確認用ポップアップ」表示タイマと同じ手順で強制切断タイマ (30 秒=3000ms) をセットします。また、強制切断用のコールバックを実装する必要があります。

## ② 「継続確認用ポップアップ」上での通信継続確認処理

「継続確認用ポップアップ」では、ユーザが通信を継続するか、切断するかを選択をできるようにします。このポップアップの表示中 (強制切断用タイマが満了する前) に、ユーザの意思による通信の継続が選択されたときには、元の画面を再表示し強制切断タイマをキャンセルし再度「継続確認用ポップアップ」表示用タイマをセットします。

逆に通信の切断が選択された場合には、現在表示されている「継続確認用ポップアップ」を破棄し、アプリが使用していた、ISocket や INetMgr などのリソースを解放した上で元の画面に復帰しなくてはなりません。なお、この場合でも、通信の継続処理のときと同様に強制切断タイマのキャンセルが必要です。

また、どちらの処理も選択されずに強制切断タイマ (30 秒) が満了した場合は、現在表示されている「継続確認用ポップアップ」を破棄し、アプリが使用していた、ISocket や INetMgr などのリソースを解放した上で元の画面に復帰しなくてはなりません。

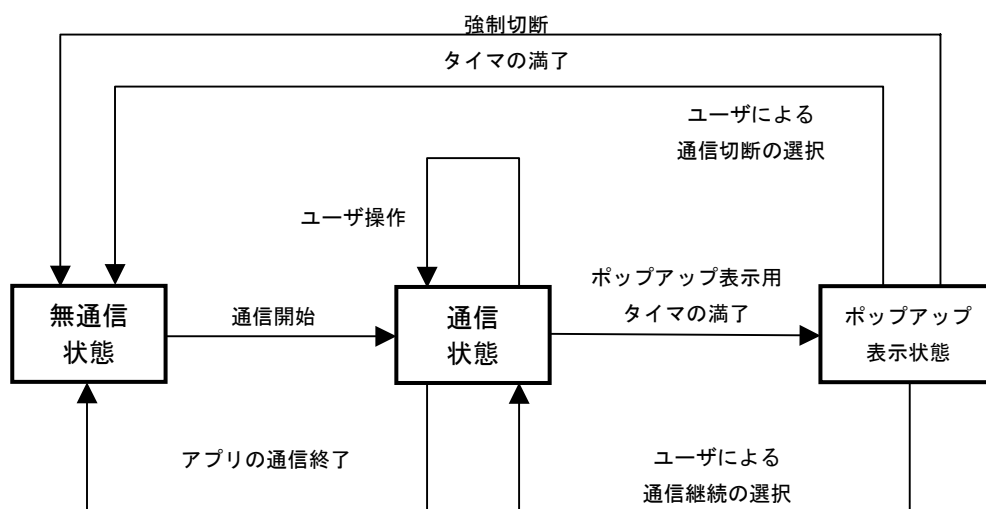
## ③キーイベント受信時のタイマーのクリア

「継続確認用ポップアップ」が表示されていない通信状態でユーザによるキー入力があった場合には、アプリは「継続確認用ポップアップ」表示タイマーをリセットする必要があります。タイマーのリセットは、ISHELL\_SetTimer() を初めにセットしたときと同じパラメータで呼び出すことにより実現できます。詳しくは Qualcomm 社提供の API リファレンスをご参照ください。

## ④アプリによる通信の終了処理

アプリがユーザによる通信中断などの一般的な理由で通信を終了した場合には、上記のいずれのタイマーもキャンセルする必要があります。ISHELL\_SetTimer() で設定したタイマーをキャンセルするには、ISHELL\_CancelTimer() を使用します。

以下に、この「常時接続型アプリ」の状態遷移の例を示します。



#### 4.17. 位置情報の取得

##### (1) IPosDet インターフェース利用の推奨

BREW アプリから位置情報測位が可能で、eznavigation と同等の位置情報パラメータを取得することが可能です。位置情報取得にあたって、IPosDet インターフェースと ISHELL\_GetPosition の 2 つの方法が存在しますが、ISHELL\_GetPosition では一部のパラメータしか取得できない仕様となっている為、ISHELL\_GetPosition は使用せず、IPosDet インターフェースを使用して下さい。

##### (2) IPosDet インターフェース利用前に解放する必要があるインスタンス

IPosDet インターフェース利用前には、必ず次のインスタンスを解放しなければなりません。

- (i) IWEB インターフェース
- (ii) ISOCKET インターフェース
- (iii) INETMGR インターフェース
- (iv) IDNS インターフェース

##### (3) BREW GPS 通信制限

GPS 取得の為に通信を行う場合、BREW アイコンをピクト行に点滅表示させながら測位しますがこの時、「eznavigation 設定」(M93)ではなく BREW 用の「BREW GPS 通信制限」(M445) [OFF: 送信可能、ON: 送信不可]の設定状態で制限がかかります。BREW アプリからの位置情報取得制限の解除/一時解除はできませんのでご注意ください。

##### (4) BREW で取得した位置情報パラメータのマッピング

これらにより取得した位置情報パラメータは次頁表のようにマッピングされ BREW アプリで取得します。なお、BREW での測地系は「WGS84 のみ」であり、フォーマットは「degree のみ」となります。なお、BREW アプリから測位された位置情報は端末の「GPS 履歴」には残りません。

IPOSEDET パラメータ	eznavigation パラメータ	パラメータの概要
dwTimeStamp	time	位置情報を位置測位サーバ(PDE)が算出した時刻
status	—	ステータス 正常時: 0 (BREW2. 0) : AEEGPS_ERR_NO_ERR (BREW2. 1) 異常時: AEEGPS_GENERAL_FAILURE
dwLat	lat	算出した位置情報の緯度情報(WGS84/degree のみ)
dwLon	lon	算出した位置情報の経度情報(WGS84/degree のみ)
wAltitude	alt	算出した位置情報の海拔高度(m)
wHeading	—	eznavigation に対応パラメータなし (常に 0 を返却)
wVelocityHor	—	eznavigation に対応パラメータなし (常に 0 を返却)
wVelocityVer	—	eznavigation に対応パラメータなし (常に 0 を返却)
accuracy	fm	実際に測位した測位方法
fValid	—	有効なフィールドを示すフラグ
bHorUnc	smaj	算出した位置情報の長軸成分誤差(m)
bHorUncAngle	majaa	算出した位置情報の長軸・短軸の傾き
bHorUncPerp	smin	算出した位置情報の短軸成分誤差(m)
wVerUnc	vert	算出した位置情報の高度誤差(m)

※ IPOSEDET パラメータの詳細については API Reference を参照して下さい

※ 不正規位置測位サーバ (PDE) 接続時には dwTimeStamp==0 が返ってきます。BREW2. 1 では上記 IPOSEDET のパラメータが全て「0」が返ってきます。

## (5) fm 値

IPosDet パラメータの AEEGPS\_ACCURACY\_LEVEL と fm 値の対応について以下表に記します

精度	AEEGPS_ACCURACY_LEVEL	fm 値
低	AEEGPS_ACCURACY_LEVEL1	3
↓	AEEGPS_ACCURACY_LEVEL2	4
↓	AEEGPS_ACCURACY_LEVEL3	5
↓	AEEGPS_ACCURACY_LEVEL4	2
↓	AEEGPS_ACCURACY_LEVEL5	1
高	AEEGPS_ACCURACY_LEVEL6	0

## (6) IPOSDet\_GetGPSInfo() の req 指定と fValid の設定値

IPOSDet パラメータの req に指定した値と fValid に設定されるフラグについて次の表に示します。なお、表中の「○」が有効なフィールドとして fValid の対応するビットとして設定されます。

IPOSDet パラメータ	req パラメータの指定			
	AEEGPS_GETINFO_LOCATION	AEEGPS_GETINFO_VELOCITY	AEEGPS_GETINFO_ALTITUDE	左記の全てを 要求
dwTimeStamp	○	○	○	○
status	○	○	○	○
dwLat	○	×	×	○
dwLon	○	×	×	○
wAltitude	(○)	×	○	○
wHeading	(○)	—	—	—
wVelocityHor	(○)	(○)	—	—
wVelocityVer	(○)	(○)	—	—
accuracy	○	×	○	○
bHorUnc	○	×	×	○
bHorUncAngle	○	×	×	○
bHorUncPerp	○	×	×	○
wVerUnc	×	×	×	×

\* (○) : BREW2.1 以降から有効フィールド。

## (7) 接続先位置測位サーバ (PDE) の不正規判断

IPosDet インターフェース仕様では、PDE サーバを切り替える仕組みを有しています。同機能で切り替えられた接続先サーバが KDDI eznavigation 仕様に準拠していない測位結果を返却した場合に、「dwTimeStamp == 0」が BREW アプリに返却することで接続先 PDE が「不正規」であり測位が正常に処理されなかったことを示します。※BREW2.1 では IPOSDet のパラメータが全て「0」が返ってきます。

(8) AEEGPSConfig の使用可能 Value

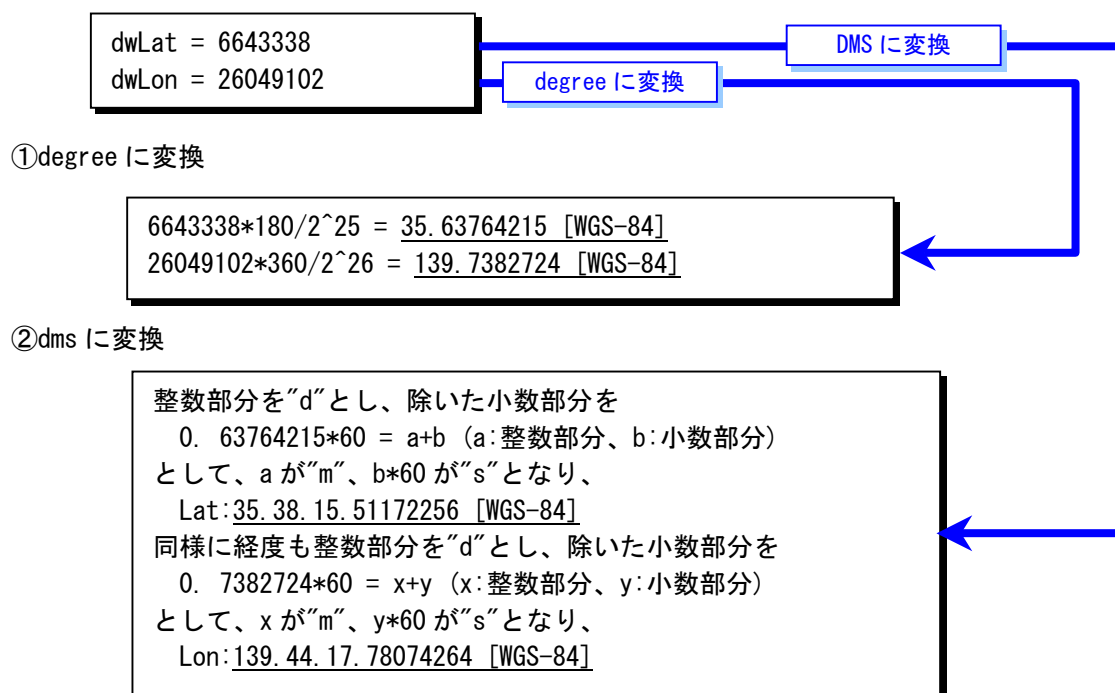
AEEGPSConfig を使用するにあたって BREW アプリからは以下の Value しか使用することができません。  
使用した場合はエラーが帰りますのでご注意ください。

AEEGPSConfig

Parameter	Value	memo
AEEGPSMode	AEEGPS_MODE_DLOAD_FIRST	左記設定を Default とする。他設定が指定された場合には EBADPARM を返却する。
AEEGPSOpt	AEEGPS_OPT_SPEED	【KDDI PDE の場合】 左記設定を Default とする。KDDI の PDE 使用時は左記以外の設定が指定された場合には EBADPARM を返却する。
		【KDDI PDE 以外の場合】 左記設定を Default とする。左記以外の設定が指定された場合には EBADPARM を返却する。
AEEGPSSvrType	AEEGPS_SERVER_DEFAULT	KDDI PDE へ接続する。
	AEEGPS_SERVER_IP	AEEGPSIPServer で設定された server へ接続する。
	AEEGPS_SERVER_DBURST	EBADPARM を返却する。
AEEGPSIPServer	addr⇒任意 port⇒任意	左記設定値で指定された Server へ接続する。
nFixes	1	先数値以外が設定された場合、EBADPARM を返却する。
nInterval	0	先数値以外が設定された場合、EBADPARM を返却する。

(9) “dwLat”、“dwLon” からの “degree” ならびに “DMS” 変換

AEEGPSInfo の dwLat, dwLon は以下の手順で degree(\*\*\*.\*\*\*), DMS(\*\*\*.\*\*\*.\*\*\*.\*\*)の形式に変換することができます。



- (10) 電子コンパス（地磁気センサー）による方位角の取得（対応機種のみ）  
IPosDet、IOrDet インターフェースにより現在地の方角取得が可能です。



#### 4.17.2 基地局からの位置情報取得（簡易位置情報）

簡易位置情報は、IPOSET\_GetSectorInfo により簡易位置情報の取得が可能です。



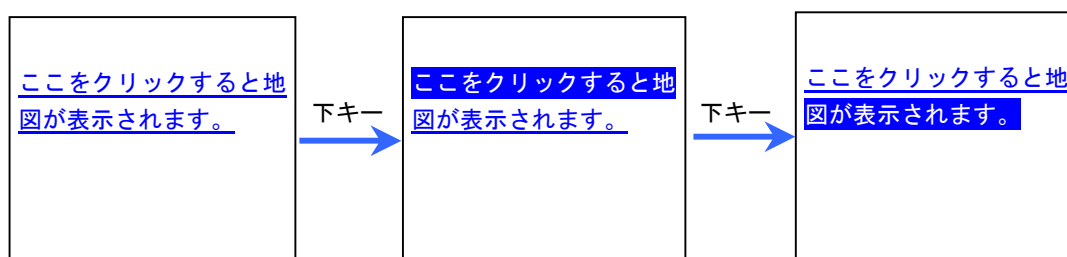
## 4. 18. その他の端末機能

### 4. 18. 1. HTML 表示

IHTMLViewer を使用することで BREW アプリ上での HTML 表示が可能となります (HTML version 4.0 サブセット)。巻末に「[\(巻末 1\) IHTMLViewer サポートタグ一覧](#)」を記します。

(1) 2 行にまたがった<a>タグ

<a>タグが 2 行にまたがった記述をした場合、以下のようにカーソルがあたってしまいます。<a>タグを使用する場合は極力 1 行に収まるようにして下さい。



### 4. 18. 2. WAP ブラウザの HTTP リクエストヘッダを用いた対応端末の判別

WAP ブラウザの HTTP リクエストヘッダに書かれる「x-up-devcap-multimedia」フィールドを用いることでその端末が BREW 対応端末か否かを判定することができます。これらは、16byte の数字列で表現され、BREW 対応に関する情報は先頭(左)より 10byte 目に位置します (No9 の「絵文字」の次)。

0: BREW 非対応
1: BREW Phase1 対応
2: Reserved
3: Reserved

#### 4.19. QVGA 対応端末におけるアプリアイコンサイズ

QVGA 対応端末では、全端末共通のアプリ・アイコンを作成したい場合、以下の標準サイズにて作成する必要があります。

(1) BREW アプリ描画可能領域：240×288 ピクセル

QVGA 対応移動機では、この画面サイズより小さくならないことが保証されます。

(2) アイコン表示（大）：192×192 ピクセル

QVGA 対応 BREW 端末にて保証される大アイコンのサイズです。

今後の QVGA 対応移動機では、このサイズより小さくならないことが保証されます。

QVGA 対応 BREW 端末にてすべてに共通するアプリを作成したい場合、大アイコンサイズを上記サイズにて作成することをお勧めします。

(3) アイコン表示（小）：48×48 ピクセル

こちらは QVGA 対応 BREW 端末での固定値となります。

移動機毎の差異はありません。

(4) 一覧表示用アイコン：32×32 ピクセル

こちらは QVGA 対応 BREW 端末での固定値となります。

移動機毎の差異はありません。

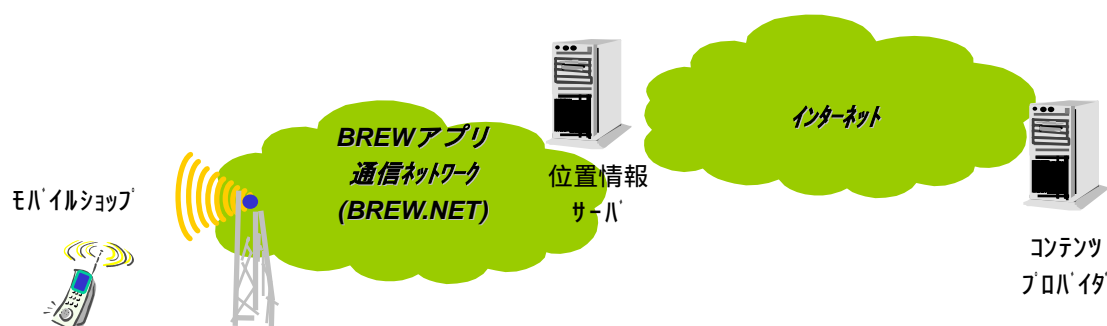
## 5. ネットワーク仕様

### 5.1. BREW ダウンロードネットワーク

BREW のダウンロードネットワークは端末から BREW アプリをダウンロードする際に利用するプライベートなネットワークとなっています。BREW アプリがアップロードされているアプリケーションダウンロードサーバ(ADS), や認証サーバ等が存在します。

### 5.2. BREW アプリ通信ネットワーク (BREW.NET)

BREW.NET とは BREW アプリが通信を必要とする時に利用できるインターネット接続環境で、BREW アプリから位置情報が利用できるように位置情報サーバも設置されています。このネットワークにおける通信は HTTP に限定されず、オープンな通信環境を提供しています。



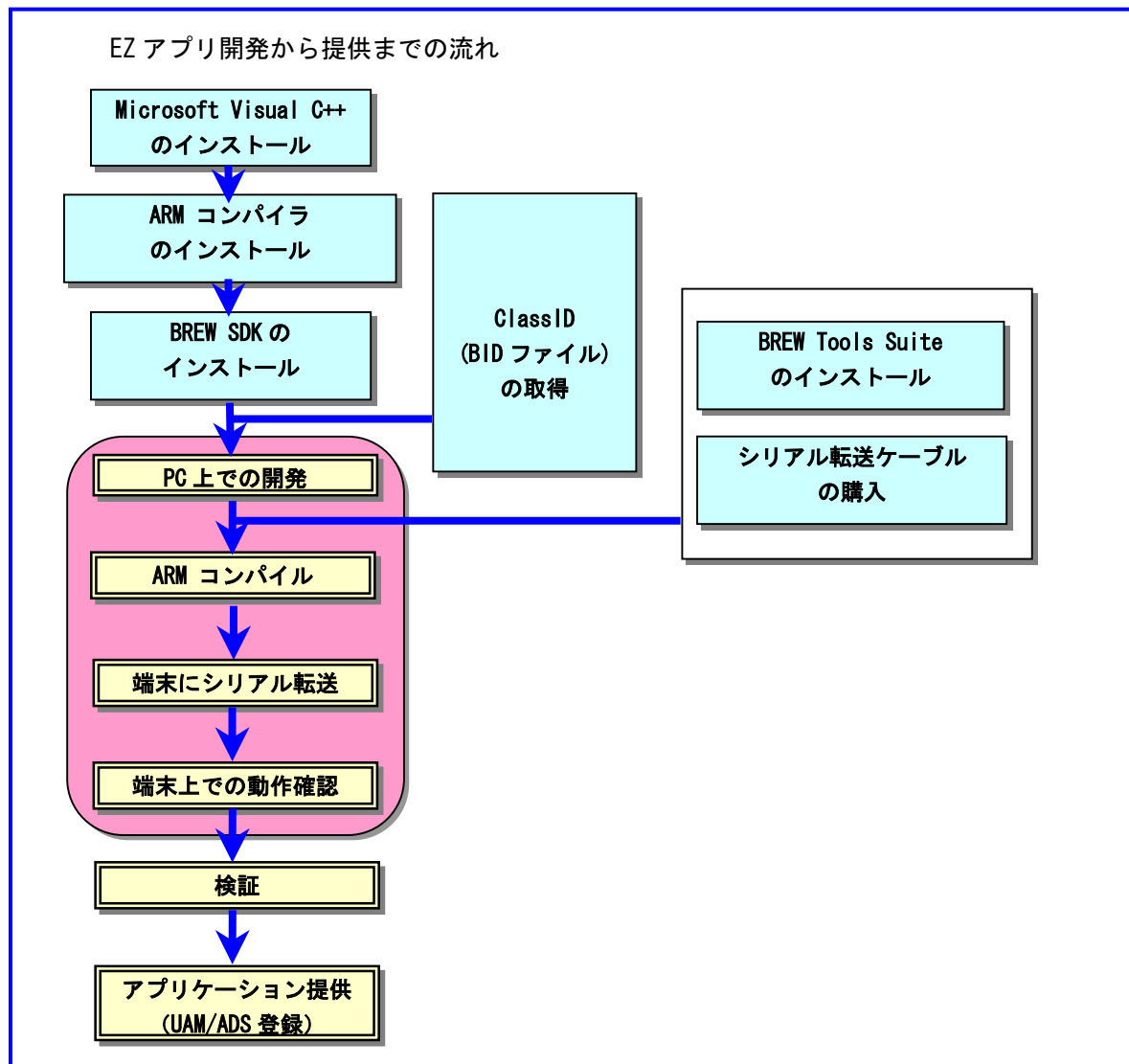
#### 5.2.1. 位置情報サーバ

BREW 位置情報 API を利用した BREW アプリからの位置情報要求に対し正確な現在位置の測定できるサーバです。

## 6. BREW アプリ開発から提供までの流れ

### 6.1. BREW アプリ開発概要

BREW アプリ開発の一連の流れを下図に示します。



## 6.2. 開発前の準備

BREW アプリを作成し、PC 上で動作させる為には以下のツール類の準備が必要です。

(1) Microsoft Visual C++

Microsoft 社が提供する Visual Studio (Visual C++) 6.0 (またはそれ以降)

(2) BREW 2.0/2.1 SDK

EZ アプリ (BREW™) を PC 上で開発するための標準の開発環境で、(1) エミュレータ、リソースエディタ、MIF エディタ、開発関連ドキュメント等が収められています。Qualcomm 社ホームページ (<http://www.qualcomm.com/brew/jp/>) にて無償で配布されています。

(3) 日本拡張パッケージファイル

KDDI により拡張された仕様 (日本拡張パッケージ) を動作させるに必要なヘッダーファイルや DLL ファイルです。「Ezweb ホームページをつくろう (<http://www.au.kddi.com/ezfactory/>)」のホームページの「技術情報」→「EZ アプリ (BREW™)」からダウンロードできます。

## 6.3. BREW アプリ検証

BREW 実行環境は、Java 実行環境に比べて、より自由度の高いアプリケーションを提供することができるため、アプリケーションのバグ等により端末にメモリ破壊等の致命的問題を引き起こす場合やセキュリティの問題を引き起こす場合があるため、すべてのアプリケーションを ADS 登録前に検証することになります。検証には、以下の 3 つの検証があります。3 つの検証を順に実施し、BREW サービス開始判定テストに合格すると、アプリケーションが商用 ADS に登録されてサービスの提供を開始することができます。詳細は「[7. BREW アプリのユーザインタフェースガイドライン](#)」、「[8. セキュリティについて](#)」を参照して下さい。

### ① 事前検証

KDDI にアプリケーションを提出する前に、コンテンツプロバイダの責任において検証していただきます。この検証は、BREW アプリの品質向上と KDDI 検証の合格率をあげることを目的としています。事前検証は「基本検証」と「U/I 検証」とで構成され、BREW アプリの正常動作 (例：状態遷移、画面遷移、及び KDDI 仕様に準拠しているか) を確認していただきます。

### ② KDDI 検証

事前検証終了後、KDDI にて実施する検証です。この検証は、基本検証、U/I 検証、セキュリティ検証で構成されます。KDDI が脅威と考えるセキュリティ問題 (特にメモリ/ファイル操作関連) や、KDDI (au) の通信設備への影響を考慮した検証の他、MIF ファイルの設定事項や基本動作の検証を行いません。

### ③ BREW アプリ導入テスト

KDDI 検証に合格した BREW アプリは、最終検証と Ezweb のサービス開始判定テスト (Ezweb) を実施します。BREW アプリを ADS に登録し、ADS からのダウンロード試験、ブラウザ連携試験 (Ezweb 連携を行う場合) 等を行います。また、Ezweb 回収代行の課金モジュールの組み込み等、試験実施にあたり、コンテンツサーバ側の作り込みが必要になる場合は、試験日程を調整の上、コンテンツプロバイダと KDDI (外部委託会社含む) が対向で試験を実施する場合があります。

## 6.4. BREW™ ディベロッパー ラボ

開発中アプリケーションの端末上での動作確認につきましては、Qualcomm 社の「BREW™ ディベロッパ

ー ラボ」にて、KDDI の BREW テスト用端末を使用して実施することが可能です。  
詳細は「BREW JAPAN.COM サイト」(<http://www.brewjapan.com/>) をご参照ください。

## 7. BREW アプリのユーザインターフェースガイドライン

本章では、BREW アプリ間で統一的な操作感を実現し、ユーザに混乱を与えないようにするため、ユーザインターフェースに対する KDDI としてのガイドラインを規定します。BREW アプリ仕様策定の際は、原則として、本ガイドラインに従ってください。

### 7.1. 端末キーの割り当て

以下に端末の基本キーを押下した場合に、それらに割り当てられる基本機能の一般的なパターンを規定します。端末 Native アプリケーション（メーラ、ブラウザ、アドレス帳等）との統一的な操作感を実現するため、BREW アプリの基本動作は、本パターンに従ってください。

ただし、BREW アプリの企画上、端末キーの割り当て変更が必要である場合は、その限りではありません。

キー名称	割り当てられる主な機能
終話（電源）キー	・ Active/Suspend を問わず全ての BREW アプリを終了し待受画面に遷移します。
発信キー	・ 発信/着信の動作を行ないます。
センターキー	・ 各種画面において選択、確定を行ないます。
方向キー	・ カーソルを上下左右に移動します。
クリアキー	・ ダイヤル、文字等の消去、及び一つ前の画面に戻ります。
ダイヤルキー	・ 0～9、*、#、および文字等の入力を行ないます。
ソフト1キー	・ 各種画面においてソフト1キーに割り当てられた動作を行ないます。
ソフト2キー	・ 各種画面においてソフト2キーに割り当てられた動作を行ないます。

## 7.2. ユーザビリティに対する基本ポリシー

以下に BREW アプリのユーザビリティに対する基本的ポリシーを規定します。ユーザフレンドリなユーザインターフェースを実現するため、極力、本ポリシーに従って下さい。

### (1) わかりやすい操作法

- 操作の手順、現在の状況をユーザが把握しやすい作りとしてください。
- 操作は、コマンド入力方式でなくメニューからの選択方式が望ましい。

### (2) わかりやすい識別性

- 必要な情報を見やすく配置、表示してください。
- (例)
  - 類似の機能は同じ色、同じ形のアイコンやボタンでまとめる。
  - 類似の機能のアイコンやボタンを近くに並べ、他のものから遠ざける。
  - 類似の機能のアイコンやボタンを閉じた矩形などで囲む。
- 重要な情報を強調してください。
- (例)
  - 目立たせるべき表示には高いコントラストを使う。
  - 目立たせるべき表示を他の部分より大きくする。
  - ブリンク表示する。
  - 動きを与える。
- 操作可能な部位と不可能な部位は視覚的に区別してください。
- 現在、どのような状態にあるかがわかりやすい作りとしてください。
  - ステータスをユーザへ通知する手段として、画面表示、音、バイブレーション等複数の手段を効果的に活用してください。

### (3) わかりやすいメニュー

- メニューの階層構造は適切に機能を分類したものにしてください。
  - メニューの標題からは予想しにくい機能を含めないこと。階層メニューの場合、下位メニューを見なくてもわかるように、上位メニューの項目名をわかりやすくしてください。
  - メニュー階層は、4 階層以下とするのが望ましい。
- メニューは、使える機能の量や種類、操作の全体量、表示内容の全体を把握できるように、できるだけ一覧で表示してください。
- 画面に表示された項目以上の選択肢がある場合、スクロール表示は、分かりやすい位置に表示してください。
- 一度に提示する要素はできるだけマジカルナンバー（ $7 \pm 2$ ）の範囲内におさめてください。
- 特定メニューへの直接ジャンプなどの近道法も用意されているのが望ましい。

### (4) わかりやすい用語・メッセージ

- 分かりにくい用語は使わないでください。
  - ユーザの知らない用語、なじみのない用語を使わないでください。
  - むやみに造語・隠語を用いないでください。
  - ユーザにとってなじみのない記号列、英単語や略称は使わないでください。
  - 初めての用語でも、日常生活から容易にその内容が類推できる表現としてください。
- 一つの用語に複数の操作上の意味を持たせないでください。
- 簡潔な表現を用いないでください。
  - 二重否定は使わないでください。
- ユーザに不快感与える用語（暴力的、公序良俗に反する等）を用いないでください。



(5) 操作の一貫性

- 異なる画面でも、同じ操作は同じ名称、色、形にしてください。
- 「はい」、「いいえ」、「戻る」、「決定」など選択は、原則として同じ操作で可能としてください。  
また、メニュー内に選択肢として置く場合は、相対的位置が変わらない位置に置いてください。
- 操作方式を統一してください。
- 画面遷移方式を統一してください。

(6) わかりやすい誘導性

- 適切な操作誘導を行ってください。
  - 操作誘導表示と操作キーの対応関係をわかりやすく表示してください。
- 言葉だけでなく視覚的な表現も利用してください。
  - 視覚的な表現は内容が直感的に理解しやすいものとしてください。
  - 視覚的な表現だけでなく、言語的なラベルもつけてください。
  - アイコン表現は必要以上に細かくしないでください。
- 具体例を示すことによって操作性のわかりやすくしてください。
- 操作手順のガイダンスを行ってください。
  - 値を入力する必要がある場合、デフォルトとして推奨値を入れてください。
  - 項目を選択する必要がある場合、デフォルトとして推奨する項目を選択してください。
- ヘルプ機能を適切に提供してください。

(7) 操作結果の確認しやすさ

- ユーザ操作に対する結果のフィードバックがあり、その内容の確認が容易にできるようにしてください。
- ユーザ操作に対するフィードバックは、2 秒以内であることが望ましい。2 秒以上かかる場合には、メッセージを表示するなどして、ユーザに通知してください。

(8) 誤操作対応

- 誤操作をしてしまっても容易に直前の状態に復帰することができるようになしてください。
- 誤操作をしてしまっても容易に初期状態に復帰することができるようになしてください。

(9) 名称” BREW” の表記方法

- アルファベット大文字にて『BREW』と表記してください。
- アルファベット小文字 (brew)、大文字と小文字の混在 (Brew) 等は使用しないでください。

## 8. セキュリティについて

### 8.1. セキュリティガイドライン

コンテンツプロバイダが BREW アプリを提供する際は、以下の全セキュリティ要求条件を満足するように BREW アプリを提供して頂きます。

#### (1) 法律の遵守

- 通信の秘密を侵さないでください。

#### (2) プライバシー保護

- 個人情報の漏洩、破損を行わないでください。
- 社会的差別の原因となりうる個人情報を収集しないでください。  
社会的差別の原因となりうる情報の例：戸籍、人種、民族、本籍地（所在都道府県に関する情報を除く）、出自、信教、病歴、犯罪歴、身体的障害、性癖、支持政党、所属団体等
- その他の個人情報についても、収集する情報の利用目的を明確に定め、その目的の達成のために必要最小限の範囲にて収集してください。

#### (3) 課金に関する不具合

- ユーザに対して不必要な課金を発生させないでください。
- BREW アプリの種別によりユーザが想定する適正な通信回数、通信データ量を超えないようにしてください。
- ユーザが意図しない通信を防ぐため、BREW アプリ仕様策定時は、以下のことに留意してください。
  - 通信発生契機と通信中の状態をユーザが認識できるようにしてください。
  - 通信先の相手をユーザが認識できるようにしてください。
  - 必要に応じてユーザの意思で通信の中断が行えるようにしてください。

#### (4) 端末動作品質の確保

- 端末基本動作不良・品質劣化を引き起こさないでください。
- メモリ破壊・メモリーリーク等のメモリ系問題を引き起こさないでください。
- バイブレータ、音源、バックライト等の端末リソースを過度に利用することで、著しい電池消耗を引き起こさないでください。

#### (5) ネットワークサービス品質の確保

- au のネットワークサービスに影響を与える過剰なトラヒック及び不正なトラヒックを発生させないでください。

#### (6) コンテンツ著作権の保護

- 自分の著作物は、自己責任にて管理してください。
- 著作物を安易に共有ディレクトリ (Shared Directory) に置かないでください。

## 8.2. BREW 標準のセキュリティ機能

BREW 標準のセキュリティ機能は以下の通りです。

### (1) 専用サーバの使用

アプリケーションは、KDDI の管理下にある専用サーバ（ADS）からのみダウンロードできる仕組みとなっています。いわゆる「勝手サイト」からアプリケーションを入手することはできません。

### (2) 実行許可署名の付与

検証に合格したアプリケーションに、実行許可のための電子署名（実行許可署名）を付与します。BREW は実行許可署名が付与されたアプリケーション以外は実行しない仕組みになっています。万一、「勝手サイト」等から非公式なアプリケーションを端末内に取り込んだとしても、実行許可署名が付与されていない場合は実行することはできません。

### (3) プログラム内容の改竄（かいざん）対策

BREW は、アプリケーション内容の改竄を検知して実行しない仕組みを備えています。BREW アプリをターゲットとしたコンピュータウイルス感染に対しても効果があります。

### (4) 特権レベル（Privilege Level）API の使用

BREW で用意されている API には、「一般レベル API」と「特権レベル API」の 2 つがあります。一般レベル API は、特に制限なく利用できるのに対して、特権レベル API は MIF において事前にその使用を宣言しておく必要があります。特権レベル API には表 1、表 2 に示すものがあり、「利用に際してセキュリティに注意しなければならない API」ということができます。これらは、アプリケーション開発者にて 1 つまたは複数個宣言できますが、アプリケーションの企画上、本当に必要な特権レベル API のみを宣言して使用するようして下さい。なお、事前に宣言されていない特権レベル API は、アプリケーションでコールされても BREW の実行環境でエラーを返して実行しない仕組みとなっています。

表 1 特権レベル API 一覧

No	特権レベル名	概要	備考
1	ファイル [PL_FILE]	ファイルを扱う	IFile IDBMgr IDatabase IDBRecord
2	ネットワーク [PL_NETWORK]	ネットワークを扱う TCP および UDP ソケットを確立する  PL_NETWORK は PL_WEB のスーパー権限。	INetMgr ISocket IWeb
3	位置情報取得 [PL_POS_LOCATION]	測位情報を扱う	IPOSDet (推奨) ISHELL_GetPosition では一部のパラメータしか取得できないため KDDI では使用を <b>非推奨</b> 。
4	TAPI [PL_TAPI]	音声通話および G メール送信を扱う	ITAPI
5	Web アクセス [PL_WEB]	Web 接続を扱う	IWeb
6	共有ディレクトリへの書き込みアクセス [PL_SHARED_WRITE] (使用を <b>非推奨</b> )	Shared Directory に書き込みを行う	出力先が Shared Directory となる API (KDDI では使用を <b>非推奨</b> )

No	特権レベル名	概要	備考
7	ベル音ディレクトリへの書き込みアクセス [PL_RINGER_WRITE] (使用禁止)	Ringer Directory に着信音ファイルを作成する	IRingerMgr は KDDI の端末には実装しないため <b>使用禁止</b>
8	アドレス帳へのアクセス [PL_ADDRBOOK]	アドレス帳の参照を行なう	IKDDIPHONESYSTEM (IAddrBook は KDDI の端末には実装しないため <b>使用禁止</b> )
9	セクタ情報へのアクセス [PL_SECTORINFO] (使用を非推奨)	測位のためにセクタ情報を扱う	IPOSEDET_GetSectorInfo() は一部のパラメータしか取得できないため KDDI では使用を <b>非推奨</b> 。
10	ダウンロード [PL_DOWNLOAD] (使用禁止)	通信事業者 (KDDI)、端末メーカー専用の特権レベル	通信事業者 (KDDI)、端末メーカー専用の特権レベルのためコンテンツプロバイダにおいては <b>使用禁止</b>
11	すべて [PL_SYSTEM] (使用禁止)		

※使用禁止の特権レベルが設定されている場合は、KDDI 検証に不合格となります。

また、KDDI 拡張の各 API については、下表のようにそれぞれの特権レベルにマッピングしてあります。

表 2 KDDI 拡張 API の特権レベルマッピング

No	API	特権レベル
1	<i>IKDDIPHONESYSTEM_***</i>	(各関数毎に PL 制限有無を規定)
2	IKDDIPHONESYSTEM_GetNumberOfHistories	アドレス帳へのアクセス [PL_ADDRBOOK]
3	IKDDIPHONESYSTEM_GetHistory	アドレス帳へのアクセス [PL_ADDRBOOK]
4	IKDDIPHONESYSTEM_GetChargeInfo	アドレス帳へのアクセス [PL_ADDRBOOK]
5	IKDDIPHONESYSTEM_GetCallInfo	アドレス帳へのアクセス [PL_ADDRBOOK]
6	IKDDIPHONESYSTEM_GetIntensity	<制限なし>
7	IKDDIPHONESYSTEM_GetPowerSupplyLevel	<制限なし>
8	IKDDIPHONESYSTEM_GetMaxVolumeLevel	<制限なし>
9	IKDDIPHONESYSTEM_GetVolumeLevel	<制限なし>
10	IKDDIPHONESYSTEM_GetVibrationStatus	<制限なし>
11	IKDDIPHONESYSTEM_GetMannerMode	<制限なし>

なお、表 1、表 2に記載のない API は「一般レベル API」となります。

### 8.3. KDDI が独自に用意するセキュリティ機能

高機能な BREW サービスをユーザに提供するためには、BREW 標準セキュリティ機能をベースとしながらも、KDDI 独自のセキュリティ機能を拡張する必要があると考えています。KDDI が提供する独自のセキュリティ機能は次の通りです。

#### (1) 通信制限

端末の M（メニュー）機能に、各種通信制限メニューを設けています。制限される通信は、BREW アプリによる BREW アプリ通信、BREW GPS 通信及び Bluetooth 通信です。ユーザは、通信を発生させる BREW アプリであっても、この M（メニュー）機能を利用することによって、それらの通信を端末側で個別に禁止（例えば、BREW アプリ通信だけを禁止し、BREW GPS 通信及び Bluetooth 通信を許可）することができます。このように、ユーザの同意のもとで通信を行い、端末内の情報がユーザの意図しない状況下で外部に漏洩することを防止しています。

表 3 通信制限メニュー

M 機能番号	端末に表示される機能名称	機能内容	ユーザの選択肢
M44x	BREW アプリ通信	UDP/TCP/Web 接続の制限	・通信を許可する ・通信を許可しない
M44x	BREW GPS 通信	測位情報送信の制限	・通信を許可する ・通信を許可しない
現在非実装	Bluetooth 通信	Bluetooth 通信の制限	・通信を許可する ・通信を許可しない

※M 機能番号は端末により異なります。

#### (2) ユーザ意思確認画面の表示

BREW アプリが通信を発生させる場合は、ユーザにその実行に対する意思を確認するための画面をアプリケーション側で表示して下さい。詳細は、9章を参照して下さい。

## 9. ユーザ意思確認画面表示ガイドライン

### 9.1. 保護すべき端末内情報

BREW が端末内から取得することができる情報はすべてセキュリティ上保護する必要があります。

## 9.2. ユーザ意思確認画面の表示（必須）

BREWアプリが表 4に記載されている機能を利用しようとする場合には、表 4に記載されたタイミングにてユーザに対してユーザ意思確認画面の表示を行い、ユーザの意思を確認し、その意思に基づいた処理を行って下さい。なお、ユーザ意思確認画面の表示は「ポップアップ」や「企画画面を流用」などその形式を問いませんが、ユーザの意思をキー押下にて確実に確認できるようにして下さい。

表 4 BREW アプリ側で確認画面表示の対象となる機能

参照先章番号	機能	確認画面表示 タイミング	確認画面表示	表示対応	全 BREW アプリ 統一表示文言
9.5.1	BREW アプリ通信時	BREW アプリ通信開始直前まで	〔確認画面表示：必須〕 ユーザへの確認画面を表示して下さい	BREW アプリ	このアプリはネットワーク通信を行いません。よろしいですか？
9.5.2	HTTP 通信時 ※BREW アプリ通信に包含	—	BREW アプリ通信に包含	BREW アプリ	
9.5.3	BREW GPS 通信時 (ユーザ操作による位置情報通知)	測位開始直前	〔確認画面表示：必須〕 位置情報送信の都度、ユーザへの確認画面を表示して下さい。また、位置情報送信中は、アニメーション等の表示を行って下さい。	BREW アプリ	位置情報を通知します。よろしいですか？
9.5.4	BREW GPS 通信時 (位置情報定期取得アプリケーションによる位置情報通知)	BREW アプリ起動時	〔確認画面表示：必須〕 起動時に1度、ユーザへの確認画面を表示して下さい。	BREW アプリ	このアプリは位置情報を定期的に通知します。よろしいですか？
9.5.5	SSL 通信時	表 6のパターンからコンテンツプロバイダが任意に選択して下さい。	〔確認画面表示：推奨〕 ユーザへの確認画面の表示を推奨します。	BREW アプリ	企画内容に応じてコンテンツプロバイダが決定して下さい。
9.5.6	ストレージエリア(Application Directory) 内ファイル／ディレクトリ操作	企画内容に応じてコンテンツプロバイダが任意に決定して下さい。	〔確認画面：コンテンツプロバイダ判断〕 企画内容に応じてコンテンツプロバイダが決定して下さい。	BREW アプリ	企画内容に応じてコンテンツプロバイダが決定して下さい。
9.5.7	共有保存領域 (Shared Directory) 内ファイルの操作	企画内容に応じてコンテンツプロバイダが任意に決定して下さい。	〔確認画面表示：必須〕 企画内容に応じてコンテンツプロバイダが決定して下さい。	BREW アプリ	企画内容に応じてコンテンツプロバイダが決定して下さい。
9.5.8	MIME タイプの登録／上書登録(本機能の使用は原則禁止)	MIME タイプの登録／上書登録直前	〔確認画面表示：必須〕 登録の都度、ユーザへの確認画面を表示して下さい。	BREW アプリ	【新規登録】 XXX を通常起動するアプリに登録しますか？ 【上書登録】 XXX を通常起動するアプリに変更しますか？ 【登録完了】 XXX を通常起動するアプリに変更しました。 ※XXX はアプリケーション名が入ります。
9.5.9	ユーザプロフィール情報の参照	ユーザプロフィール情報の参照直前	〔確認画面表示：必須〕 参照の都度、ユーザへの確認画面を表示して下さい。	BREW アプリ	XXX を通知します。よろしいですか？ XXX は取得するプロフィール情報項目名が入ります。
9.5.10	アドレス帳情報への操作 (Shim 連携およびNativeUI 連携のみ利用する場合を除く。)	アドレス帳情報の操作直前まで	〔確認画面表示：必須〕 ユーザへの確認画面を表示して下さい。	BREW アプリ	このアプリはアドレス帳の情報への読み込み／書き込みを行います。よろしいですか？

参照先章番号	機能	確認画面表示タイミング	確認画面表示	表示対応	全 BREW アプリ統一表示文言
9.5.11	データフォルダ内ファイル操作 (NativeUI 連携のみ利用する場合を除く。)	ファイル読み・書き込みの直前まで	[確認画面表示：必須] ユーザへの確認画面を表示して下さい。	BREW アプリ	このアプリはデータフォルダ内のファイル読み込み／書き込みを行います。よろしいですか？
9.5.12	データファイル転送の通信発生時	通信発生直前	[確認画面表示：必須] ユーザへの確認画面を表示して下さい。	BREW アプリ	xxxxB のデータ通信を行います。よろしいですか？
9.5.13	通常終了時 (終話キーによる強制終了時を除く)	ユーザによる通常終了操作直後	[通常終了手段設置：必須] [確認画面：コンテンツのハッシュ判断] ユーザへの確認画面を表示して下さい。	BREW アプリ	このアプリを終了します。よろしいですか？

### 9.3. ユーザ意思確認画面の表示例

ユーザ意思確認画面の表示例を示します。この例の他にもいろいろな表示方法が考えられます。

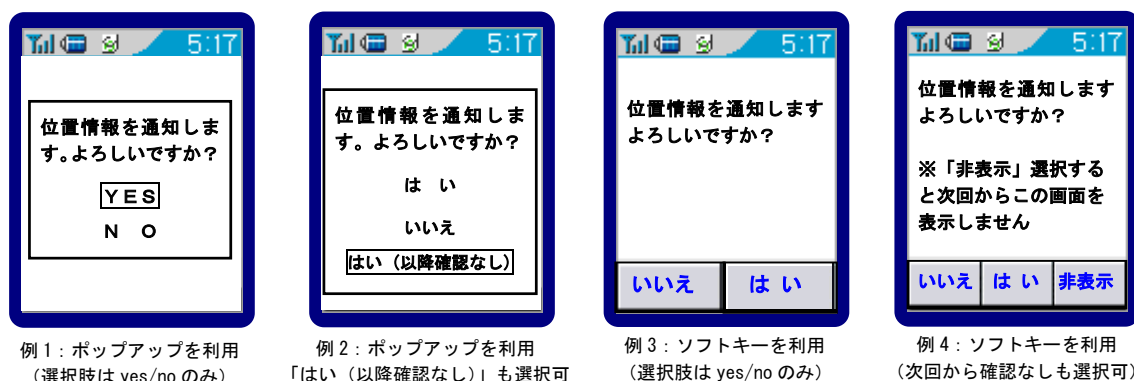


図 1 ユーザ意思確認画面の表示例

#### ✓ チェックリスト

No.	種別	内容	備考
1	必須	図 1 の例 1～例 4 において、フォーカスを当てることのできる場合は、初期フォーカスを「はい」(または Yes 等の“accept”を表す文言) 上として下さい。	
2	推奨	ユーザの利便性を考慮して、図 1 の例 2 や例 4 のような「次回から非表示」を表す選択肢を入れることを推奨します。	
3	必須	「次回から非表示」を表す選択肢を入れる場合は、文言を「はい (以降確認なし)」として下さい。	
4	注意	文字の大きさ、全半角、句点の有無、配置 (改行処理)、ソフトキーの配置は規定しませんが、わかりやすく工夫して表示して下さい。	
5	必須	例 2 や例 4 のように「はい (以降確認なし)」を選択肢に入れる場合、後述の図 2 画面 4 の「BREW 確認画面設定機能」と連携を図り、ユーザ設定直後にその設定情報を設定情報データファイルとして自らのストレージエリア (Application Directory) 内に出力する等の方法により、ユーザ設定内容をアプリケーション終了後も保持できるようにして下さい。	「はい (以降確認なし)」が利用できるのは表 5 に示す機能のみが対象です。
6	注意	図 1 の例 1～例 4 はあくまでも一例です。この他の表示方法でもユーザが問題なく認識できる方法があれば自由に選択し利用して下さい。	



#### 9.4. ユーザ意思確認画面のお好み設定機能の設置（必須）

BREW アプリには、表 5に示す **BREW 通信系機能、アドレス帳およびデータフォルダ連携、データフォルダ内のファイル転送通信を対象**に、ユーザ意思確認画面のお好み設定機能（以下、確認画面設定機能という）を設け、ユーザの好みに応じて確認画面の「全表示」「一切非表示」を切り替えられるようにして下さい。ユーザが BREW アプリを初めて起動した時（出荷時）の初期値は、表 5の通りとして下さい。ユーザが好みに応じて変更できるようにすることで、ユーザの利便性を向上させ、BREW アプリの価値を高める狙いがあります。

表 5 BREW 確認画面設定機能の対象項目と初期値

No	BREW 機能	ユーザ選択肢	初期値	備考
1	BREW アプリ通信	表示／非表示	<b>表示</b>	
2	BREW GPS 通信	表示／非表示	<b>表示</b>	BREW と GPS の間にはスペースが入ります
3	アドレス帳連携	表示／非表示	<b>表示</b>	Shim 連携および NativeUI 連携のみ利用時を除く
4	データフォルダ 連携	表示／非表示	<b>表示</b>	NativeUI 連携のみ利用時を除く

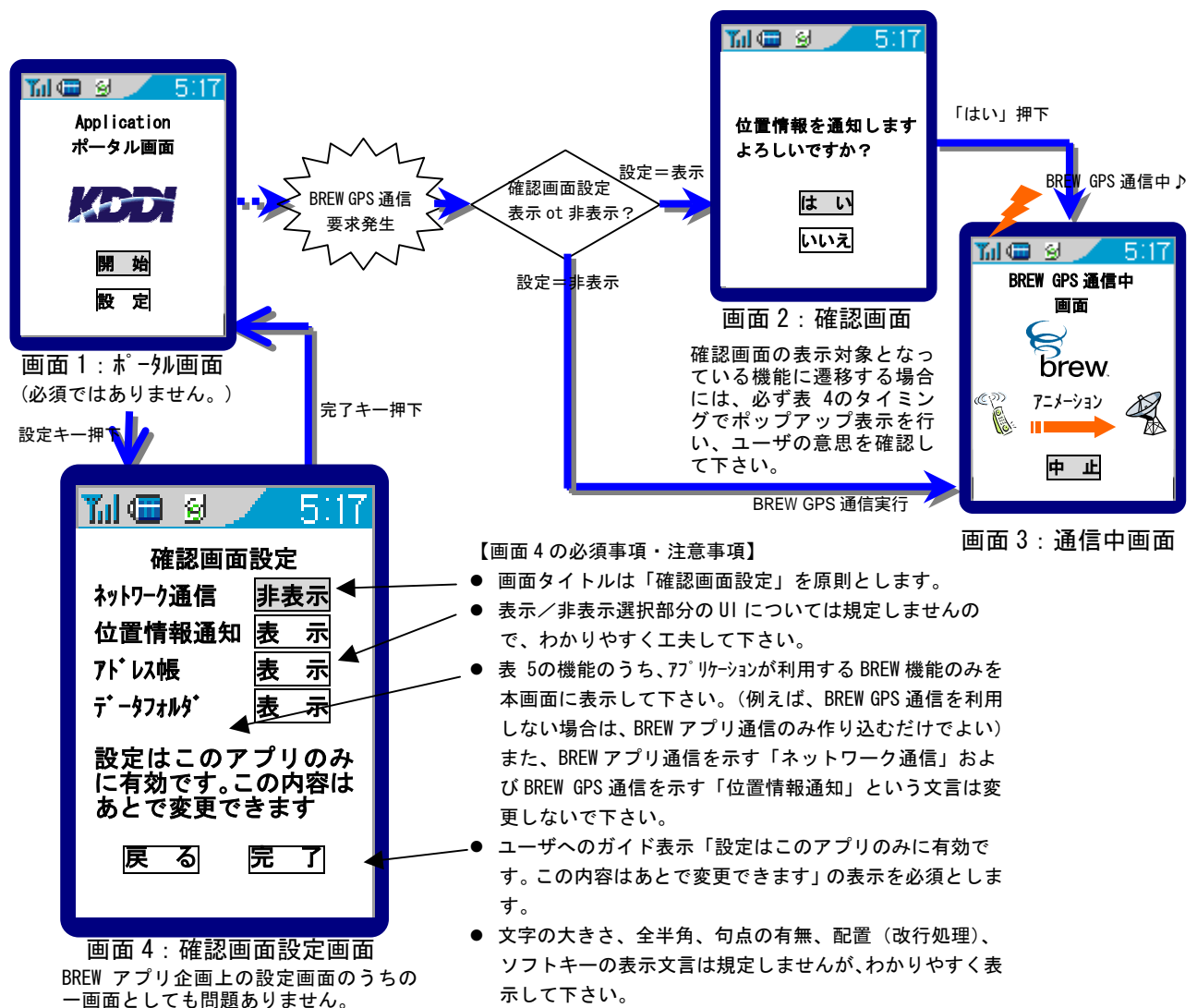


図 2 基本的な画面遷移例

✓ チェックリスト

No.	種別	内容	備考
1	必須	ユーザ設定完了直後にその設定情報を設定情報データファイルとして自らのストレージエリア (Application Directory) 内に出力する等の方法により、ユーザ設定内容をアプリケーション終了後も保持できるようにして下さい。	
2	必須	ユーザが設定を完了したあとでも設定内容を変更できるようにして下さい。	
3	必須	ユーザにわかりやすい適切な場所に設定画面への入口を設けて下さい。	
4	必須	確認画面設定機能でユーザが「表示」を選択している BREW 機能に遷移しようとする場合には、表 4 に指定されているタイミングで確認画面を表示したあとに当該 BREW 機能に遷移し、「表示」を選択している機能に遷移しようとする場合は、確認画面を表示せずに当該 BREW 機能に遷移して下さい。	
5	必須	図 1 の例 2 や例 4 のように「はい (以降確認なし)」を選択肢に入れる場合、前述の図 2 画面 4 の「確認画面設定機能」と連携を図り、ユーザ設定直後にその設定情報を設定情報データファイルとして自らのストレージエリア (Application Directory) 内に出力し、ユーザ設定内容をアプリケーション終了後も保持できるようにして下さい。	
6	注意	端末のメニューの中にも「BREW アプリ通信制限」および「BREW GPS 通信制限」の ON/OFF 設定画面がありますが、そちらは「全ての BREW アプリ」に関する通信行為の許可／拒否を設定するものです。それに対して、本項の「確認画面設定機能」は、当該アプリのみに有効となる「通信を行なう前の確認画面の表示／非表示」を設定するものです。	「確認画面設定機能」が利用できるのは表 5 に示す内、BREW 通信系機能のみが対象です。
7	注意	端末のメニューの中にも「アドレス帳制限」の ON/OFF 設定画面がありますが、そちらは「全ての BREW アプリ」に関するアドレス帳への行為の許可／拒否を設定するものです。それに対して、本項の「確認画面設定機能」は、当該アプリのみに有効となる「アドレス帳への行為を行なう前の確認画面の表示／非表示」を設定するものです。	「確認画面設定機能」が利用できるのは表 5 に示す内、アドレス帳連携のみが対象です。

## 9. 5. 機能別画面遷移例と注意事項

本項では、各機能別の画面遷移例、表示文言、注意事項を記載します。本項において、画面遷移図を記入してありますがあくまでも一例と考えて下さい。また、通信中に表示されるBREWロゴの点滅は省略してあります。

### 9. 5. 1. BREW アプリ通信時【確認画面表示：必須】

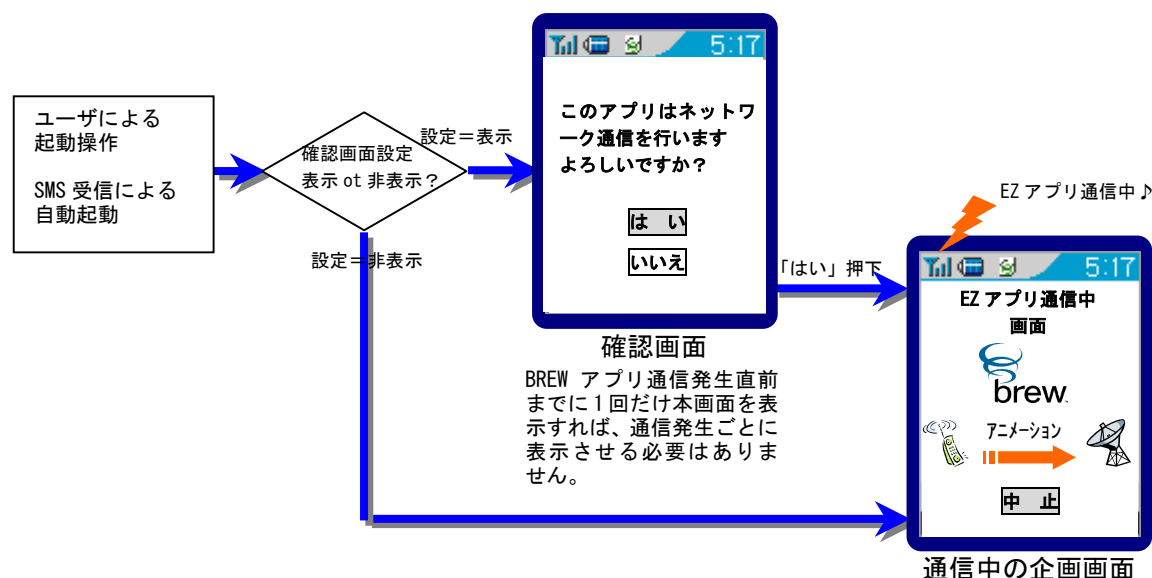


図 3 BREW アプリ通信時の画面遷移例

#### ✓ チェックリスト

No.	種別	内容	備考
1	必須	BREW アプリ通信を利用する場合には、BREW アプリの起動ごとかつ BREW アプリ通信発生直前までに確認画面を表示して下さい。文言は「このアプリはネットワーク通信を行ないます。よろしいですか?」として下さい。	
2	必須	確認画面において、「いいえ」が押下された場合は、通信が実行されないことが原因で企画画面と通信エラー画面を行き来する（無限）ループが発生しないように遷移先を考慮して下さい。	
3	必須	文字の大きさ、全半角、句点の有無、配置（改行処理）、ソフトキーの表示文言は規定しませんが、わかりやすく表示して下さい。	
4	必須	BREW アプリの企画内容により語尾を変化させることは許容しますが、容易にその内容が類推できる範囲で変化させて下さい。（例：「このアプリはネットワーク通信を行なうであります。よろしいでありますか?」）	
5	必須	企画画面においては、通信中は通信中であることがわかるように表示を工夫して下さい。（上記遷移図では例としてアニメーションを利用しています。）	
6	必須	ユーザ設定完了直後にその設定情報を設定情報データファイルとして自らのストレージエリア（Application Directory）内に出力する等の方法により、ユーザ設定内容をアプリケーション終了後も保持できるようにして下さい。	
7	必須	通信中にユーザがハードキー「クリア」を押下した場合は、通信を中断できるようにして下さい。	
8	推奨	通信中の企画画面において、通信を中断できることをユーザに視認させるために、「中止」ソフトキー（またはこれに相当するソフトキー）をつくりこむことを推奨します。	
9	推奨	なるべくユーザに不必要な確認画面を見せないようにするために、確認画面を表示するタイミングを次図 4 の遷移例のように工夫することを推奨します。	

[参考：推奨画面遷移例]

ここでは、ネットワーク接続型ゲームでありながら、ネットワーク非接続でも楽しめるゲームアプリケーションを想定した推奨遷移例を記載します。

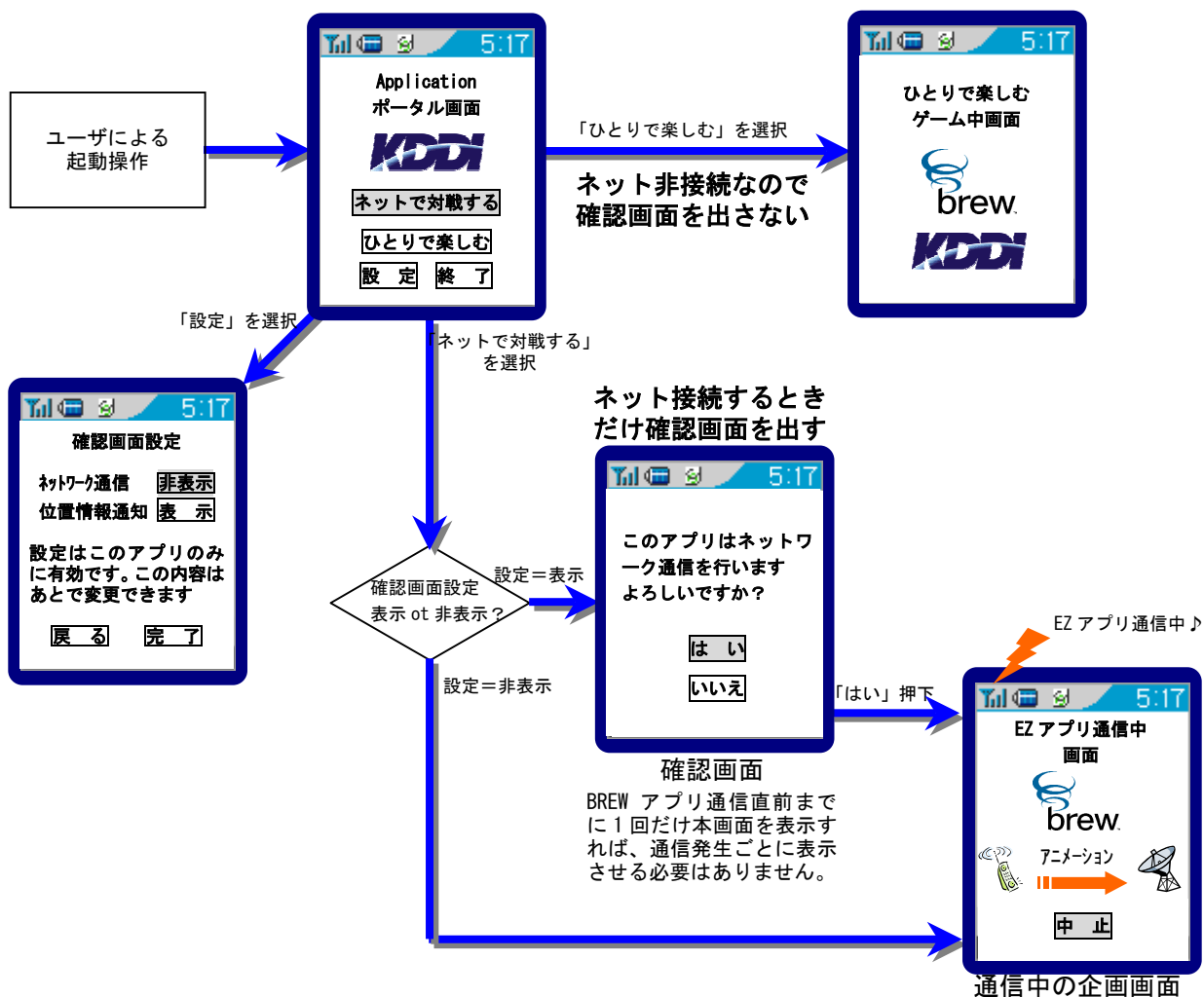


図 4 許容画面遷移例

図 4では、BREWアプリ通信を行なう場合の例となっていますが、BREW GPS通信の場合や、BREWアプリ通信とBREW GPS通信を併用する場合には、確認画面の部分に相当する確認画面(図 6、図 7、図 14、図 15等)に置き換えてお考え下さい。

## 9.5.2. BREW アプリ通信時（定期的通信型または常時接続型アプリの場合）【確認画面表示：必須】

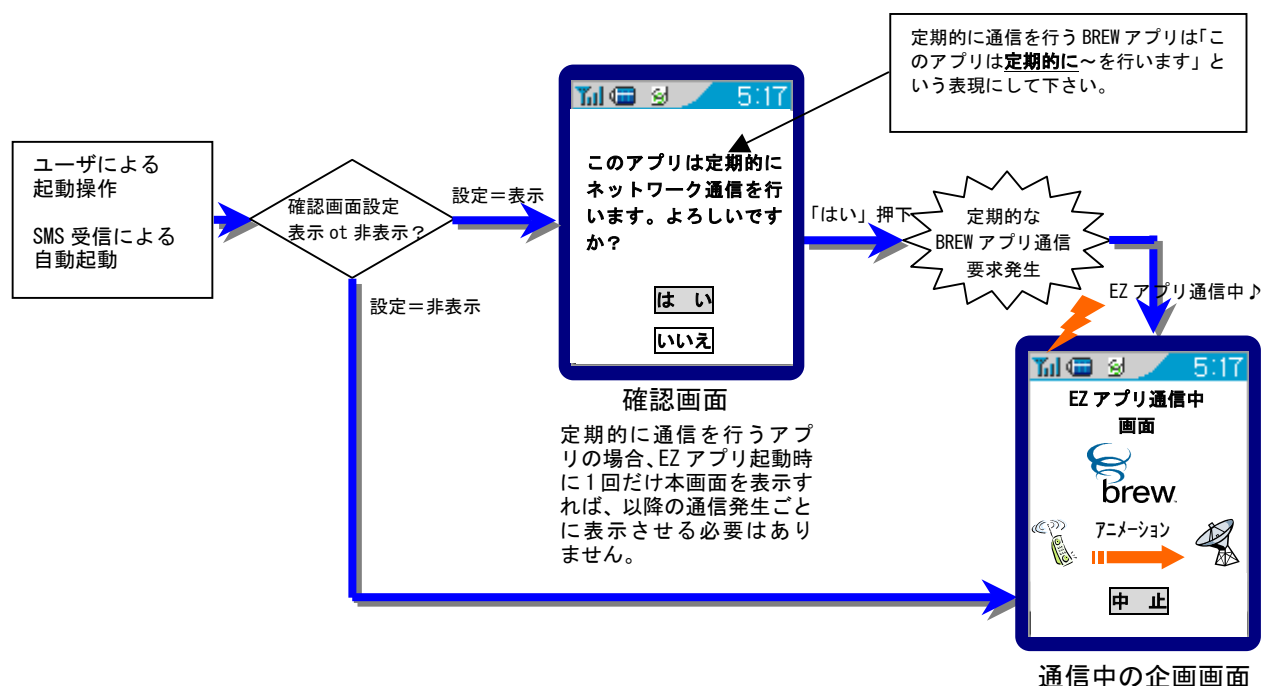


図 5 BREW アプリ通信時（定期的通信）の画面遷移例

## ✓ チェックリスト

No.	種別	内容	備考
1	必須	定期的に（定期的接続型または常時接続型）通信を行う BREW アプリは、BREW アプリ起動ごとかつ BREW アプリ起動時に確認画面を表示して下さい。文言は「このアプリは定期的にネットワーク通信を行ないます。よろしいですか？」として下さい。	
2	必須	文字の大きさ、全半角、句点の有無、配置（改行処理）、ソフトキーの表示文言は規定しませんが、わかりやすく表示して下さい。	
3	必須	BREW アプリの企画内容により語尾を変化させることは許容しますが、容易にその内容が類推できる範囲で変化させて下さい。（例：「このアプリは定期的にネットワーク通信を行います。よろしいでちゅか？」）	
4	必須	企画画面においては、通信中は通信中であることがわかるように表示を工夫して下さい。（上記遷移図では例としてアニメーションを利用しています。）	
5	必須	通信中にユーザがハードキー「クリア」を押下した場合は、通信を中断できるようにして下さい。	
6	必須	ユーザ設定完了直後にその設定情報を設定情報データファイルとして自らのストレージエリア（Application Directory）内に出力する等の方法により、ユーザ設定内容をアプリケーション終了後も保持できるようにして下さい。	
7	必須	定期的に（定期的接続型または常時接続型）通信を行う BREW アプリは、一定時間ユーザ操作がない場合には 9.5.12 に規定する確認画面を表示して下さい。	
8	推奨	通信中の企画画面において、通信を中断できることをユーザに視認させるために、「中止」ソフトキー（またはこれに相当するソフトキー）をつくりこむことを推奨します。	
9	推奨	なるべくユーザに不必要な確認画面を見せないようにするために、確認画面を表示するタイミングを図 4 の遷移例のように工夫することを推奨します。	

## 9.5.3. BREW GPS 通信時（ユーザ操作による場合）〔確認画面表示：必須〕

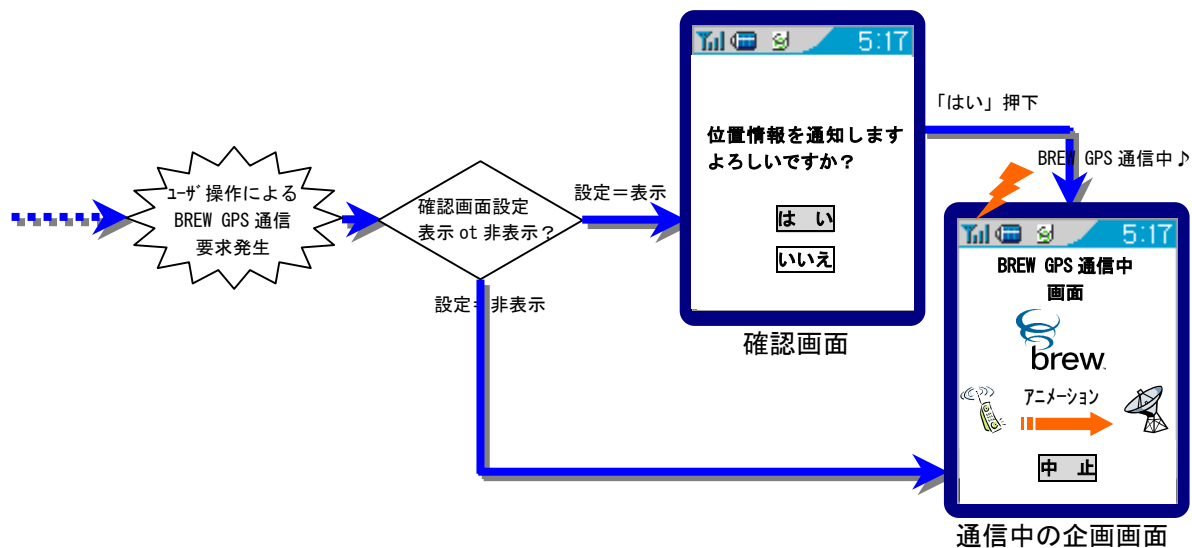


図 6 BREW GPS 通信時（ユーザ操作）の画面遷移例

## ✓ チェックリスト

No.	種別	内容	備考
1	必須	BREW GPS 通信（ユーザ操作による）を利用する場合には、位置情報通知ごとに毎回確認画面を表示して下さい。	
2	必須	確認画面において、文言は「位置情報を通知します。よろしいですか?」を原則とします。	
3	必須	確認画面において、「いいえ」が押下された場合は、通信が実行されないことが原因で企画画面と通信エラー画面を行き来する（無限）ループが発生しないように遷移先を考慮して下さい。	
4	必須	文字の大きさ、全半角、句点の有無、配置（改行処理）、ソフトキーの表示文言は規定しませんが、わかりやすく表示して下さい。	
5	必須	BREW アプリの企画内容により語尾を変化させることは許容しますが、容易にその内容が類推できる範囲で変化させて下さい。（例：「このアプリは位置情報を通知しまちゅ。よろしいでちゅか?」）	
6	必須	企画画面においては、通信中は通信中であることがわかるように表示を工夫して下さい。（上記遷移図では例としてアニメーションを利用しています。）	
7	必須	ユーザ設定完了直後にその設定情報を設定情報データファイルとして自らのストレージエリア（Application Directory）内に出力する等の方法により、ユーザ設定内容をアプリケーション終了後も保持できるようにして下さい。	
8	必須	通信中にユーザがハードキー「クリア」を押下した場合は、通信を中断できるようにして下さい。	
9	推奨	通信中の企画画面において、通信を中断できることをユーザに視認させるために、「中止」ソフトキー（またはこれに相当するソフトキー）をつくりこむことを推奨します。	

## 9.5.4. BREW GPS 通信時（位置情報を定期的に通知するアプリの場合）【確認画面表示：必須】

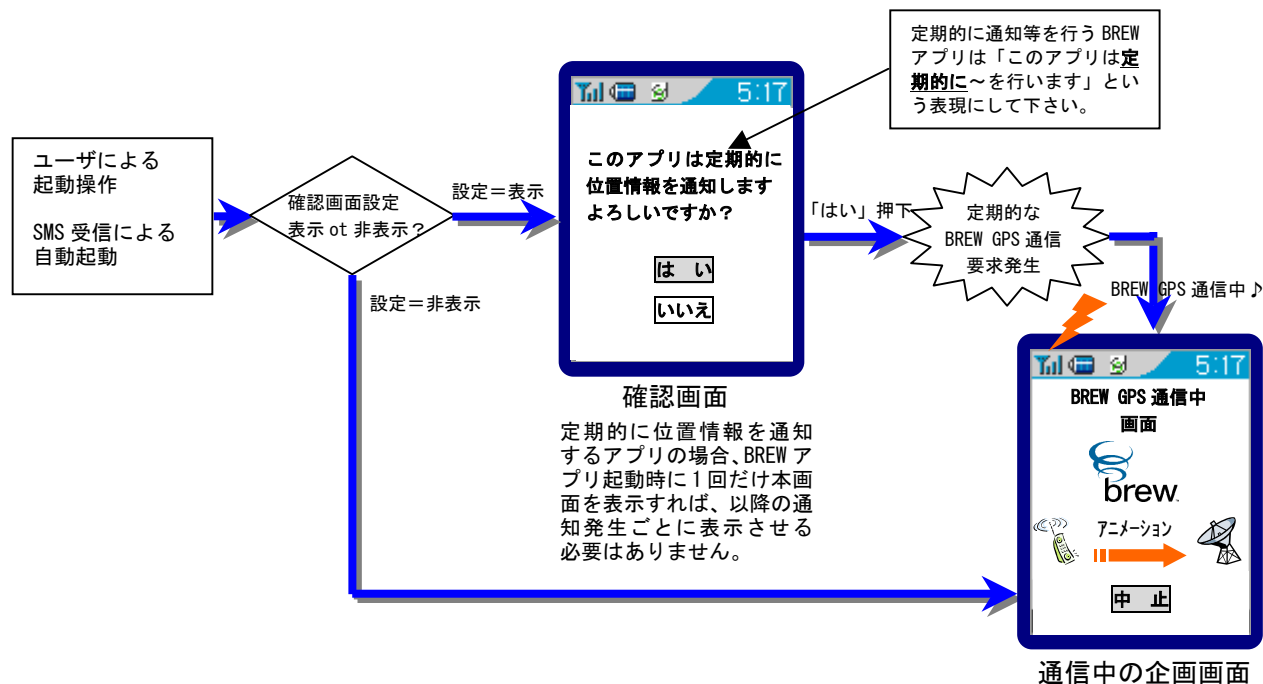


図 7 BREW GPS 通信時（定期的取得）の画面遷移例

## ✓ チェックリスト

No.	種別	内容	備考
1	必須	位置情報を定期的に通知する BREW アプリは、BREW アプリ起動ごとかつ BREW アプリ起動時に確認画面を表示して下さい。	
2	必須	確認画面において、文言は「このアプリは定期的に位置情報を通知します。よろしいですか？」を原則とします。	文言は「通知」をお願いします。
3	必須	文字の大きさ、全半角、句点の有無、配置（改行処理）、ソフトキーの表示文言は規定しませんが、わかりやすく表示して下さい。	
4	必須	BREW アプリの企画内容により語尾を変化させることは許容しますが、容易にその内容が類推できる範囲で変化させて下さい。（例：「このアプリは定期的に位置情報を通知しましゅ。よろしいでちゅか？」）	
5	必須	企画画面においては、通信中は通信中であることがわかるように表示を工夫して下さい。（上記遷移図では例としてアニメーションを利用しています。）	
6	必須	通信中にユーザがハードキー「クリア」を押下した場合は、通信を中断できるようにして下さい。	
7	推奨	通信中の企画画面において、通信を中断できることをユーザに視認させるために、「中止」ソフトキー（またはこれに相当するソフトキー）をつくりこむことを推奨します。	



## 9.5.5. SSL 通信時 [確認画面表示 : 推奨]

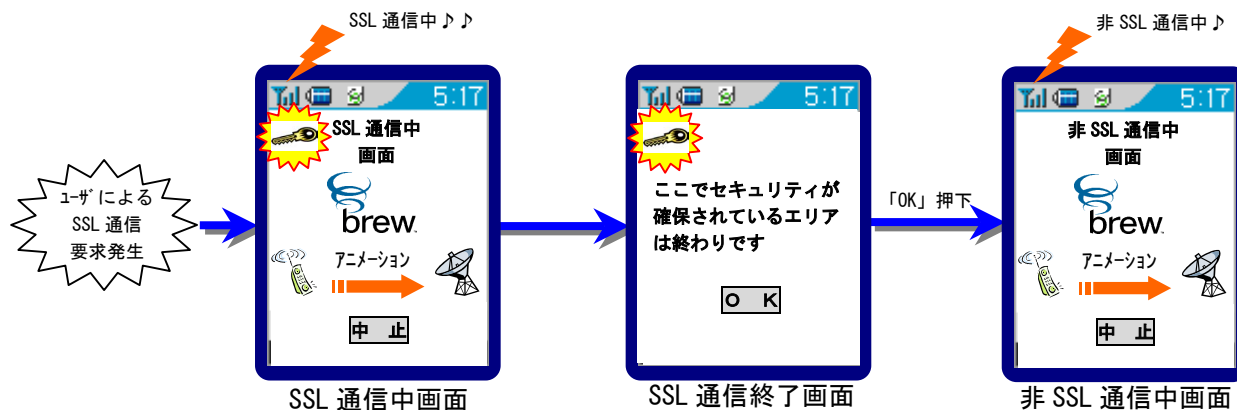


図 8 SSL 通信時の画面遷移の例 (参考記載)

表 6 SSL 通信時確認画面の表示パターン

No.	SSL 通信開始時の確認画面	SSL 通信中画面	SSL 通信終了時の確認画面	推奨/許容の別
1	表示あり	表示あり	表示あり	推奨
2	表示なし	表示あり	表示あり	許容
3	表示なし	表示あり	表示なし	許容

## ✓ チェックリスト

No.	種別	内容	備考
1	注意	BREW プラットフォームでは、端末側でピクト行に SSL 通信中を示す「鍵」アイコンを表示する仕組みはありません。必要に応じて BREW アプリにて表示して下さい。	
2	推奨	SSL 通信時の表示は必須ではありませんが、画面を作りこむことを推奨します。	
3	推奨	SSL 通信中は、継続して SSL 通信中であることを表すアイコンや文字列等をアプリケーション表示エリア内の視認しやすい場所に表示することを推奨します。アイコンは SSL 通信中を表すことがわかれば「鍵」でなくてもよいこととします。アイコンのかわりに文字列で表現することを許容します。その場合の文言については規定しませんので、SSL 通信中であることをユーザが容易に類推することができる文言として下さい。	
4	推奨	SSL 通信開始時および SSL 通信終了時に確認画面を表示することを推奨します。SSL 通信開始時の表示については、ユーザにキー押下を求めることなく、一定時間表示後に SSL 通信に遷移しても問題ありません。SSL 通信開始時および SSL 通信終了時とも文言については規定しませんので、当該動作へ移行することをユーザが容易に類推することができる文言として下さい。	SSL 通信から非 SSL 通信への遷移はユーザの同意後に行って下さい。
5	必須	画面を作りこむ場合、確認画面表示を行うパターンは表 6No. 1~3 の中から選択して下さい。表 6No. 1~3 以外のパターンは許容できません。	
6	必須	SSL 通信中を示すアイコンや文字列を表示している場合には SSL 通信終了時に必ず消灯して下さい。	
7	必須	SSL 通信中であることを表す文字列や確認画面等を表示する場合の文字の大きさ、全半角、句点の有無、配置 (改行処理)、ソフトキーの表示文言は規定しませんが、わかりやすく表示して下さい。	
8	必須	文言は BREW アプリの企画を考慮した遊び心あるものでも許容することとします。	
9	推奨	通信中の企画画面において、通信を中断できることをユーザに視認させるために、「中止」ソフトキー (またはこれに相当するソフトキー) をつくりこむことを推奨します。	

## 9.5.6. ストレージエリア内ファイル操作時 [確認画面表示 : コンテンツの判定]



ここでは、ストレージエリア（Application Directory）内のファイル／ディレクトリをユーザが意識する状態で操作する場合の確認画面について記載します。

表 7 ストレージエリア内のファイル操作時の確認画面

No.	ユーザが意識するファイル操作	確認画面表示	確認画面表示 タイミング	表示文言	表示対応
1	ファイル新規保存	確認画面表示の有無、表示タイミング、表示文言はコンテンツプロバイダ任意とします。			BREWアプリ
2	ファイルリネーム	確認画面表示の有無、表示タイミング、表示文言はコンテンツプロバイダ任意とします。			BREWアプリ
3	ファイル上書保存	確認画面表示の有無、表示タイミング、表示文言はコンテンツプロバイダ任意とします。			BREWアプリ
4	ファイル削除	確認画面表示の有無、表示タイミング、表示文言はコンテンツプロバイダ任意とします。			BREWアプリ
5	ディレクトリ新規作成	確認画面表示の有無、表示タイミング、表示文言はコンテンツプロバイダ任意とします。			BREWアプリ
6	ディレクトリリネーム	確認画面表示の有無、表示タイミング、表示文言はコンテンツプロバイダ任意とします。			BREWアプリ
7-1	ディレクトリ削除 （ディレクトリ内にファイルなし）	確認画面表示の有無、表示タイミング、表示文言はコンテンツプロバイダ任意とします。			BREWアプリ
7-2	ディレクトリ削除 （ディレクトリ内にファイルあり）	確認画面表示の有無、表示タイミング、表示文言はコンテンツプロバイダ任意とします。			BREWアプリ

#### ✓ チェックリスト

No.	種別	内容	備考
1	注意	ストレージエリア（Application Directory）内のファイルをユーザが意識して取り扱う場合は、アプリケーションの企画内容を考慮して、コンテンツプロバイダにて確認画面を表示するかどうかを判断して下さい。表 7のように特別な規定は設けていませんので、アプリケーションの企画内容を考慮して、コンテンツプロバイダにて確認画面の表示タイミングや表示文言を決定して下さい。例えば、新規保存確認画面と上書保存確認画面を共通で利用するような画面の作りかたも可能です。ただし、わかりやすく表示して下さい。BREWアプリの企画内容により表示文言に遊び心を加えることを許容しますが、容易にその内容が類推できる範囲で変化させて下さい。（例：「削除しまちゅか？」）	

## 9.5.7. 共有保存領域内ファイルの操作時【確認画面表示：必須】

KDDIとしては共有保存領域（Shared Directory）の使用は推奨しません。BREWアプリの企画上やむを得ず共有保存領域（Shared Directory）を使用し、ユーザが意識する状態でファイル操作機能を利用する場合は、下記の規定に従って確認画面を表示して下さい。

表 8 共有保存領域内のファイル操作時の確認画面

No.	ユーザが意識するファイル操作	確認画面を表示する／しない	確認画面表示 タイミング	表示文言	表示対応
1	ファイル新規保存	表示必須	コンテンツロパイト任意		BREWアプリ
2	ファイルリネーム	表示必須	コンテンツロパイト任意		BREWアプリ
3	ファイル上書保存	表示必須	コンテンツロパイト任意		BREWアプリ
4	ファイル削除	表示必須	コンテンツロパイト任意		BREWアプリ
5	ディレクトリ新規作成	共有保存領域（Shared Directory）内にディレクトリを作成し、それに関連する操作を行なうことを禁止します。			
6	ディレクトリリネーム				
7-1	ディレクトリ削除 （ディレクトリ内にファイルなし）				
7-2	ディレクトリ削除 （ディレクトリ内にファイルあり）				

## ✓ チェックリスト

No.	種別	内容	備考
1	禁止	ユーザが意識しない状況下で共有保存領域（Shared Directory）内のファイル进行操作することは一切禁止します。	
2	禁止	ユーザが意識できる状況下であっても共有保存領域（Shared Directory）内にサブディレクトリを作成し、それに関連する操作を行なうことを禁止します。	
3	禁止	個人情報が含まれるファイル、著作権が主張されているファイル等を共有保存領域（Shared Directory）に保存することを禁止します。	
4	必須	共有保存領域（Shared Directory）内ファイルをユーザが意識して取り扱う場合は、表 8に従って確認画面を表示して下さい。この確認画面の文言はアプリケーションの企画内容を考慮に入れて表示できるように特別な規定は設けていません。例えば、新規保存確認画面と上書保存確認画面を共通で利用するような画面の作りかたも可能です。ただし、わかりやすく表示して下さい。BREWアプリの企画内容により表示文言に遊び心を加えることを許容しますが、容易にその内容が類推できる範囲で変化させて下さい。（例：「削除しまちゅか？」）	
5	注意	ファイルを共有保存領域（Shared Directory）内に保存する場合、他のBREWアプリを用いて内容を改変されたり、端末外に転送されたりする可能性が考えられるため、共有保存領域（Shared Directory）の利用にあたっては、個人情報のセキュリティ管理に加えて、著作権の管理に十分注意して下さい。	

**9.5.8. MIME タイプの登録／上書登録 [機能使用：原則禁止、確認画面表示：必須]**

ユーザがMIMEタイプを登録・削除する方法がなく、MIMEタイプがBREWアプリにより登録・更新されると別のBREWアプリが動作しなくなる場合があるため、現在MIMEタイプの取り扱いについて検討中です。当面、BREWアプリがMIMEタイプを登録することを原則として禁止します。

また、Extension等をBREWアプリに組み込んで利用する場合、ClassIDを直接指定するようにして下さい。

## 9.5.9. ユーザプロフィール情報の参照 [確認画面表示：必須]

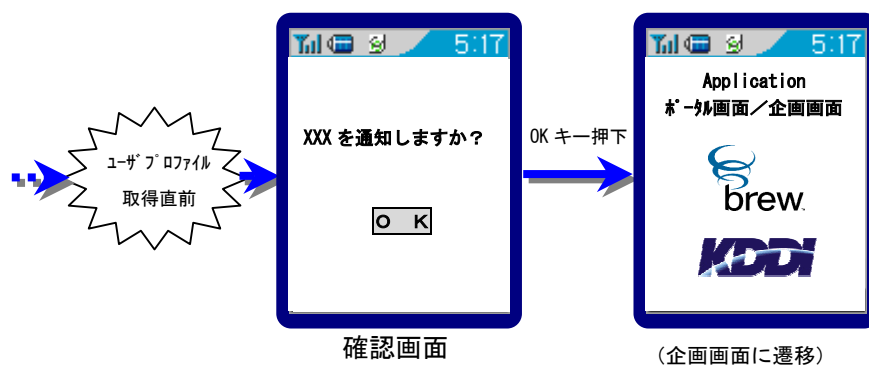


図 9 ユーザプロフィール情報参照時の画面遷移例

## ✓ チェックリスト

No.	種別	内容	備考
1	必須	BREWアプリが端末から <b>エラー! 参照元が見つかりません。</b> に示すユーザプロフィール情報のうち、自局電話番号、名前、読み仮名、E-mailアドレスのいずれかを参照する（参照のみで端末外へ送出不含む場合も含む）場合は、取得直前に毎回、BREWアプリにて確認画面を表示して下さい。	BREW 確認画面設定機能で「非表示」にすることは許容しません。
2	必須	ユーザE-mailアドレス参照時の確認画面について、E-mailアプリケーション（メール）などにおいて毎回表示させることで操作性が著しく損なわれる場合は、E-mailアドレス初回参照時までに確認画面を表示すればよいものとします。この際、他の確認画面と組み合わせることも許容します。（図 15参照）	
3	必須	確認画面において文言は具体的に「XXXを通知しますか?」とすることを原則とします。	
4	必須	上記3のXXXの部分には、「電話番号」「名前」「名前読み仮名」「メールアドレス」が入ります。	文言の定義は表 9の通りです。
5	必須	文字の大きさ、全半角、句点の有無、配置（改行処理）、ソフトキーの表示文言は規定しませんが、わかりやすく表示して下さい。	
6	必須	BREWアプリの企画内容により語尾を変化させることは許容しますが、容易にその内容が類推できる範囲で変化させて下さい。（例：「XXXを通知するよ?」）	

表 9 ユーザプロフィール情報名称変換表

No.	ユーザプロフィール情報名称 (エラー! 参照元が見つかりません。での定義名称)	確認画面表示上の名称
1	自局電話番号	電話番号
2	名前	名前
3	読み仮名	名前読み仮名
4	E-mail アドレス	メールアドレス
5	最終改訂日時	(確認画面必要なし)

# 9.5.10. アドレス帳情報操作 [確認画面表示：必須]

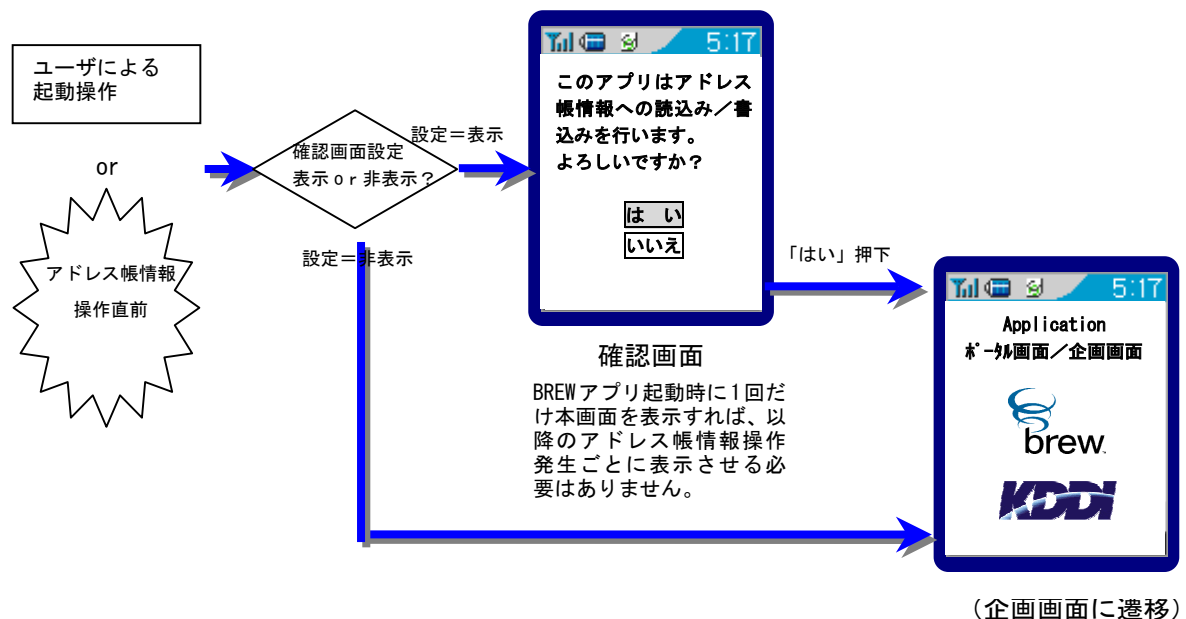


図 10 アドレス帳情報操作時の画面遷移例

## ✓ チェックリスト

No.	種別	内容	備考
1	必須	アドレス帳連携 (Shim連携およびNativeUI連携のみを利用した場合を除く。) により アドレス帳情報の操作を利用する場合には、BREWアプリの起動ごとかつアドレス帳 情報操作直前までに確認画面を表示して下さい。文言は「このアプリはアドレス帳 情報への読み込み／書き込みを行います。よろしいですか?」として下さい。	
2	必須	文字の大きさ、全半角、句点の有無、配置 (改行処理)、ソフトキーの表示文言は規 定しませんが、わかりやすく表示して下さい。	
3	必須	BREW アプリの企画内容により語尾を変化させることは許容しますが、容易にその内 容が類推できる範囲で変化させて下さい。 (例: 「このアプリはアドレス帳情報への 読み込み／書き込みを行うであります。よろしいでありますか?」)	
4	必須	図 1の例 2 や例 4 のように「はい (以降確認なし)」を選択肢に入れる場合、前述の 図 2画面 4 の「BREW 確認画面設定機能」と連携を図り、ユーザ設定直後にその設定 情報を設定情報データファイルとして自らのストレージエリア (Application Directory) 内に出力し、ユーザ設定内容をアプリケーション終了後も保持できるよ うにして下さい。	

## 9.5.11. データフォルダ内ファイル操作 [確認画面表示：必須]

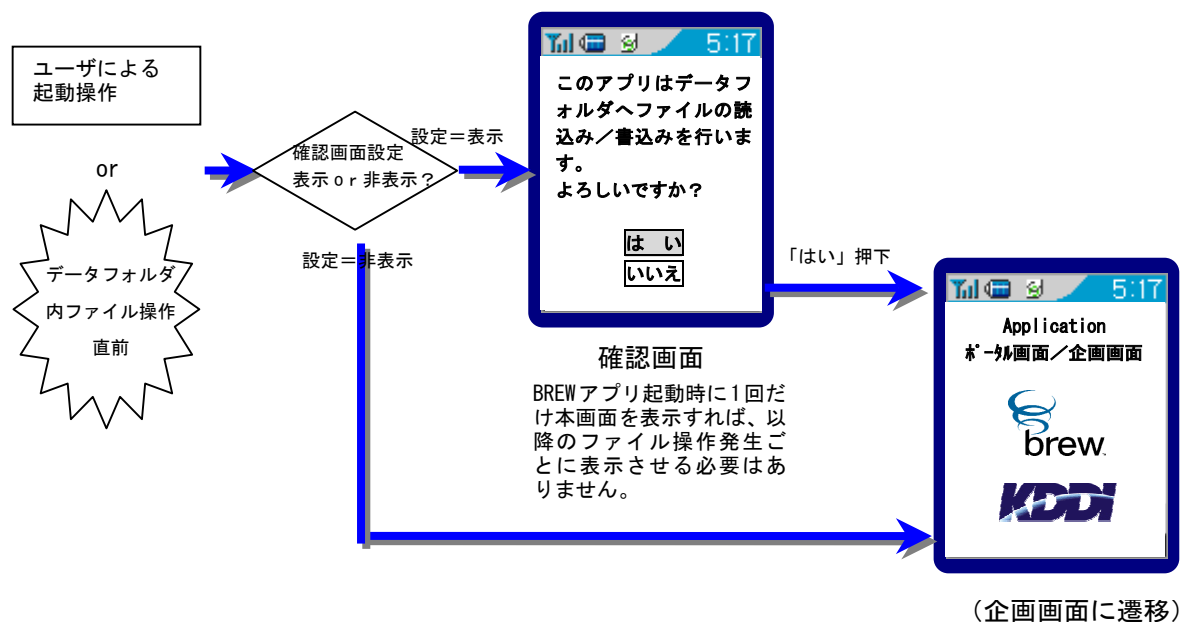


図 11 データフォルダ内ファイル操作時の画面遷移例

## ✓ チェックリスト

No.	種別	内容	備考
1	必須	データフォルダ連携 (NativeUI 連携のみを利用した場合を除く。)によりデータフォルダ内ファイルの操作を利用する場合には、BREW アプリの起動ごとかつファイル操作直前までに確認画面を表示して下さい。文言は「このアプリはデータフォルダへファイルの読み込み／書き込みを行います。よろしいですか?」として下さい。	
2	必須	文字の大きさ、全半角、句点の有無、配置 (改行処理)、ソフトキーの表示文言は規定しませんが、わかりやすく表示して下さい。	
3	必須	BREW アプリの企画内容により語尾を変化させることは許容しますが、容易にその内容が類推できる範囲で変化させて下さい。(例:「このアプリはデータフォルダへファイルの読み込み／書き込みを行うであります。よろしいでありますか?」) また、読み込み、あるいは書き込みのどちらか一方しか操作を行わない場合は、操作を行う方だけの文言を明記してください。(例:「このアプリはデータフォルダのファイルの読み込みを行います。よろしいですか?」)	
4	必須	図 1 の例 2 や例 4 のように「はい (以降確認なし)」を選択肢に入れる場合、前述の図 2 画面 4 の「BREW 確認画面設定機能」と連携を図り、ユーザ設定直後にその設定情報を設定情報データファイルとして自らのストレージエリア (Application Directory) 内に出力し、ユーザ設定内容をアプリケーション終了後も保持できるようにして下さい。	

### 9.5.12. データファイル転送の通信発生時〔確認画面表示：必須〕

10KB 以上のデータファイルを転送する通信を行う場合は、ユーザが意識することなく大量のデータ通信を行ってしまい、高額な通信料を課金されることを防ぐために、必ず確認画面を表示してください。

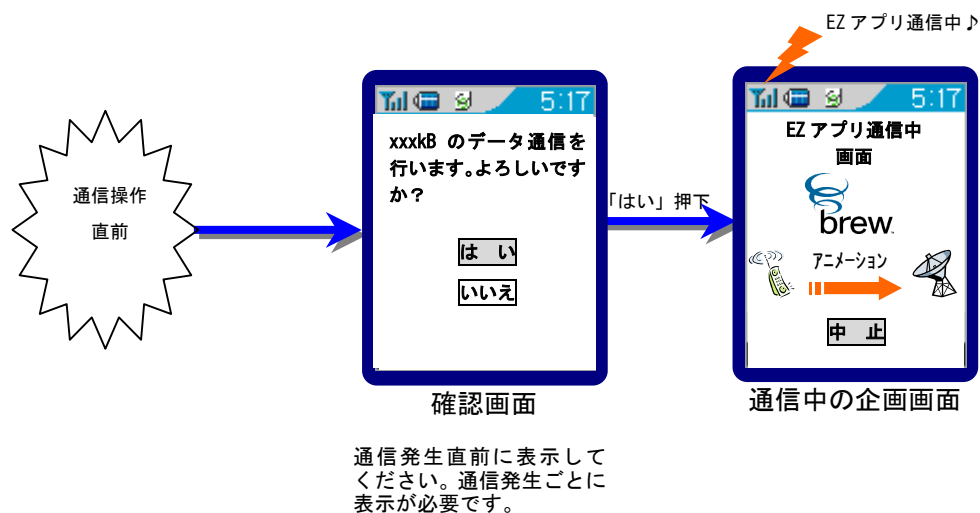


図 12 BREW アプリ通信時の画面遷移例

#### ✓ チェックリスト

No.	種別	内容	備考
1	必須	データフォルダ内のファイルをサーバにアップする等の通信が発生する場合には、BREW アプリ通信発生直前に確認画面を表示して下さい。文言は「xxxkB のデータ通信を行います。よろしいですか?」として下さい。	転送するファイルサイズを表示してください。
2	必須	確認画面において、「いいえ」が押下された場合は、通信が実行されないことが原因で企画画面と通信エラー画面を行き来する（無限）ループが発生しないように遷移先を考慮して下さい。	
3	必須	文字の大きさ、全半角、句点の有無、配置（改行処理）、ソフトキーの表示文言は規定しませんが、わかりやすく表示して下さい。	
4	必須	BREW アプリの企画内容により語尾を変化させることは許容しますが、容易にその内容が類推できる範囲で変化させて下さい。（例：「このアプリはネットワーク通信を行なうであります。よろしいでありますか?」）	
5	必須	企画画面においては、通信中は通信中であることがわかるように表示を工夫して下さい。（上記遷移図では例としてアニメーションを利用しています。）	
6	必須	通信中にユーザがハードキー「クリア」を押下した場合は、通信を中断できるようにして下さい。	
7	推奨	通信中の企画画面において、通信を中断できることをユーザに視認させるために、「中止」ソフトキー（またはこれに相当するソフトキー）をつくりこむことを推奨します。	

9.5.13. BREW アプリの通常終了 [通常終了手段設置：必須] [確認画面表示：推奨]

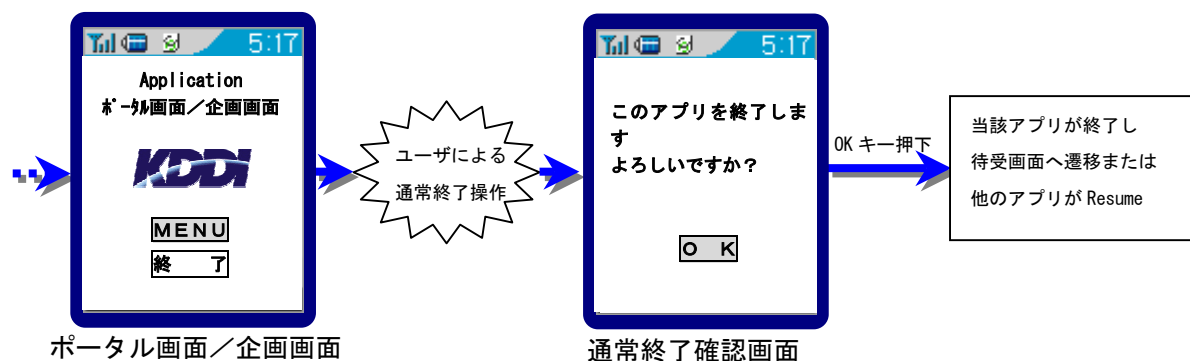


図 13 アプリ通常終了時の画面遷移例

✓ チェックリスト

No.	種別	内容	備考
1	必須	全てのBREWアプリは、ユーザが少なくとも次の2通りの手段でアプリを終了できるようにして下さい。 ・ 終話キーによる終了（強制終了）……………端末側にて強制制御 ・ 終話キー以外の手段による終了（通常終了）……BREWアプリにて用意が必要 通常終了のための手段については特に規定しませんので、BREWアプリの企画内容を考慮してつくりこんで下さい。終了手段としては、クリアキーの利用、ソフトキーの利用、IHTMLViewerタグの利用、アプリケーションメニューへ終了メニューを編入させる方法等が考えられます。	
2	推奨	ユーザによる通常終了操作があった場合、BREWアプリは通常終了確認画面を表示して下さい。	
3	推奨	通常終了確認画面において文言は「このアプリを終了します。よろしいですか？」とすることを原則とします。	
4	推奨	文字の大きさ、全半角、句点の有無、配置（改行処理）、ソフトキーの表示文言は規定しませんが、わかりやすく表示して下さい。	
5	推奨	BREWアプリの企画内容により語尾を変化させることは許容しますが、容易にその内容が類推できる範囲で変化させて下さい。（例：「このアプリを終了しますよ？よろしいですかあ？？」）	



#### 9.5.14. 各種確認画面を統合する場合の画面遷移例 [下記例のような画面遷移でも許容]

一般的に、BREWアプリは、BREW GPS通信、BREWアプリ通信、アドレス帳の参照等を組み合わせて行うことが多いと考えられ、ここまでに記載したような画面遷移例をそのまま利用した場合、確認画面が頻繁に表示されることになります。そこで、ユーザの利便性を考慮し、表示回数を削減するために、図14に記載のような統合遷移パターンの利用も許容することとします。

また、図 15ではBREWメールアプリ（メーラー）を想定した具体的な画面作成例を記載します。

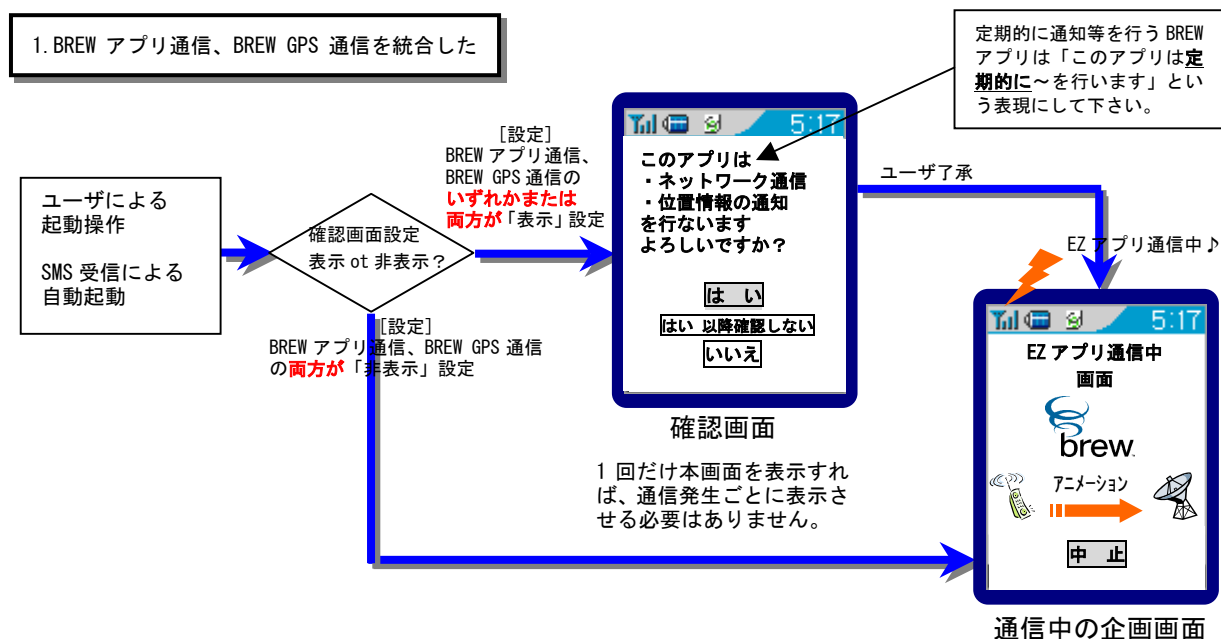


図 14 確認画面を統合した場合の画面遷移例(1)

✓ チェックリスト (図 14のような遷移パターンを利用する場合)

No.	種別	内容	備考
1	必須	BREWアプリの起動ごとかつBREWアプリ起動時に確認画面を表示して下さい。	
2	必須	確認画面において、文言は「このアプリはネットワーク通信、位置情報の通知を行いません。よろしいですか?」を原則とします。定期的にBREW GPS通信を行なうBREWアプリについては「このアプリは定期的にネットワーク通信、位置情報の通知を行いません。よろしいですか?」を原則とします。	
3	必須	ユーザが「はい 以降確認しない」を押下した場合は、BREW確認画面設定機能のネットワーク通信および位置情報通知とともに非表示設定に変更し、その設定情報を設定情報データファイルとして自らのストレージエリア (Application Directory) 内に出力する等の方法により、ユーザ設定内容をアプリケーション終了後も保持できるようにして下さい。	
4	必須	文字の大きさ、全半角、句点の有無、配置 (改行処理)、ソフトキーの表示文言は規定しません。ただし、図 14の例では、①ネットワーク接続、②位置情報通知、という2つの行為を行なっていることをユーザに意識させるような工夫して下さい。	
5	必須	BREWアプリの企画内容により語尾を変化させることは許容しますが、容易にその内容が類推できる範囲で変化させて下さい。(例:「このアプリは定期的にネットワーク通信、位置情報の通知を行ないまちゅ。よろしいでちゅか?」)	
6	必須	企画画面においては、通信中は通信中であることがわかるように表示を工夫して下さい。(上記遷移図では例としてアニメーションを利用しています。)	
7	必須	通信中にユーザがハードキー「クリア」を押下した場合は、通信を中断できるようにして下さい。	
8	推奨	通信中の企画画面において、通信を中断できることをユーザに視認させるために、「中止」ソフトキー (またはこれに相当するソフトキー) をつくりこむことを推奨します。	

## 2. BREW メールアプリ（メーラー）等の画面作成例

BREWメールアプリのように「BREWアプリ通信」と「ユーザのE-mailアドレスの取得」を行うアプリケーションを想定し、2つの確認画面を統合した画面の一例を下図に記載します。

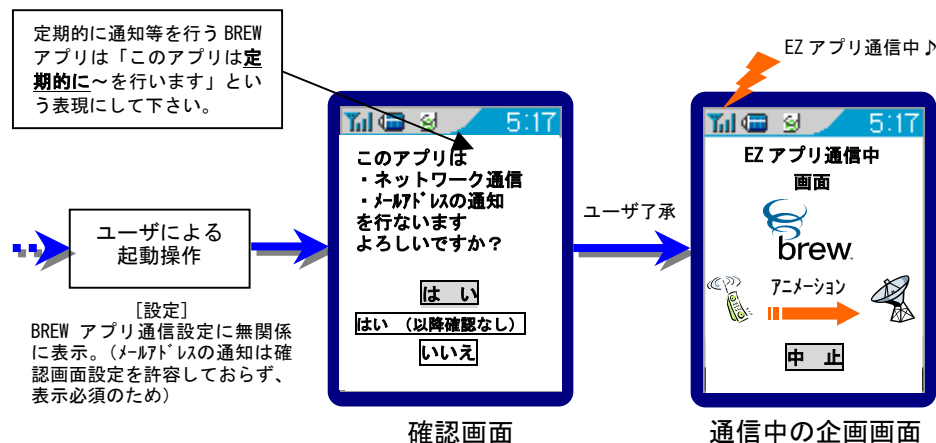


図 15 確認画面を統合した場合の画面遷移例(2)

✓ チェックリスト（図 15のような遷移パターンを利用する場合）

No.	種別	内容	備考
1	必須	BREWアプリの起動ごとかつBREWアプリ起動時に確認画面を表示して下さい。	
2	必須	確認画面において、文言は「このアプリはネットワーク通信、メールアドレスの通知を行ないます。よろしいですか？」を原則とします。定期的にネットワーク通信を行うBREWアプリ（例：定期的にメールボックスを確認するメーラー）については「このアプリは定期的にネットワーク通信、メールアドレスの通知を行ないます。よろしいですか？」を原則とします。	
3	必須	ユーザが「はい 以降確認しない」を押下した場合は、BREW確認画面設定機能のネットワーク通信を非表示設定に変更し、その設定情報を設定情報データファイルとして自らのストレージエリア（Application Directory）内に出力する等の方法により、ユーザ設定内容をアプリケーション終了後も保持できるようにして下さい。	メールアドレスの通知については BREW 確認画面設定の利用を許容していません。
4	必須	ユーザプロフィール情報各項目の確認画面表示上の名称については、表 9に準じます。	
5	必須	その他の必須事項、推奨事項は（図 14のような遷移パターンを利用する場合）のチェックリストに準じます。	

### 9.5.15. 「常時接続型アプリ」における通信継続確認画面

「常時接続型アプリ」は、ユーザに対する不要な通信料の負担を強いる事や、不必要な無線リソースの確保等 KDDI ネットワークインフラへの影響が懸念されます。したがって、一定時間内に無操作状態が続いた場合はユーザへネットワーク接続を継続するか否かの「通信継続確認画面」を表示し、ユーザの許可が得られない場合にはその後の接続保持のためのパケット送出等、ユーザ操作を契機とした通信の再開以外を一切禁止し、このパケット送出を停止するまでの無操作タイマ時間を「5 分間」と規定します。また、「通信継続確認画面」の表示はタイマ満了前の 30 秒間、つまり無操作タイマ 4 分 30 秒移行の猶予時間に必ず表示しなければなりません。

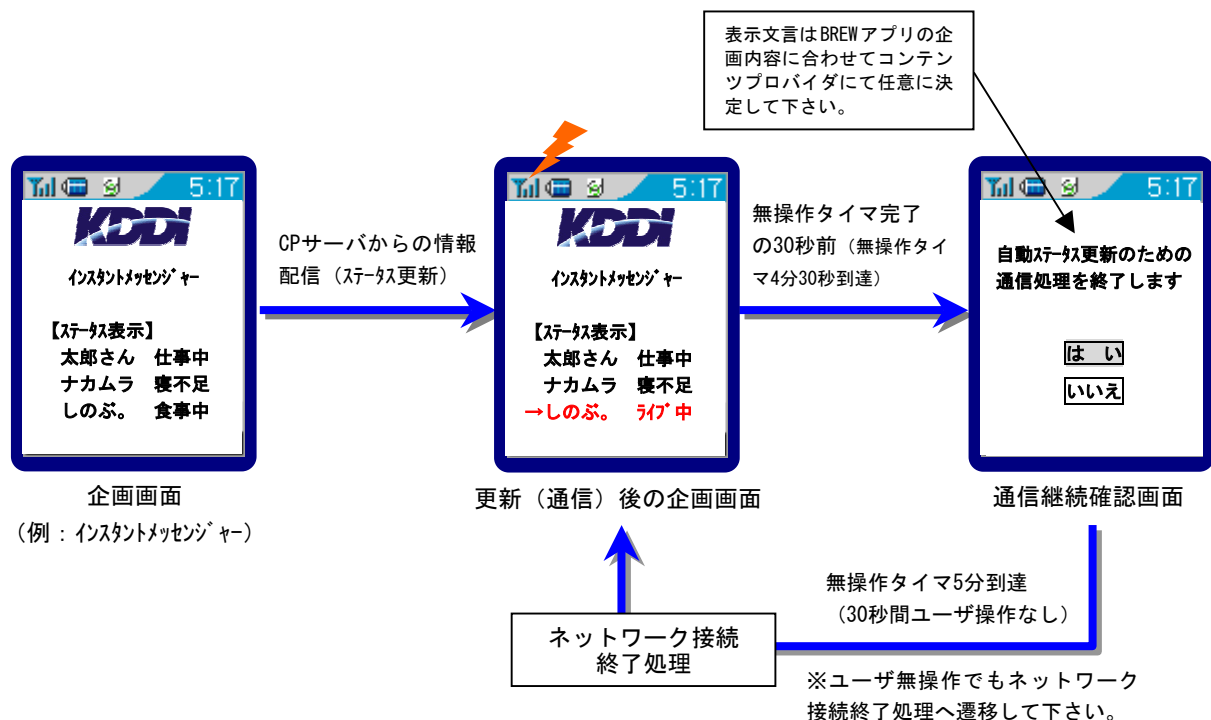


図 16 通信確認画面表示遷移例

### ✓ チェックリスト

No.	種別	内容	備考
1	必須	ユーザの最後の操作時に「無操作タイマ」を起動し、その4分30秒後に通信継続確認画面を表示して下さい。通信継続確認後、30秒間ユーザ操作がない（無操作タイマが5分に達した）場合は、ネットワーク接続終了処理を行って下さい。	
2	必須	図 16の通信継続確認画面において、ユーザが「はい／いいえ」のいずれも押下しなかった場合、無操作タイマが5分に到達した時点でネットワーク接続終了処理を行って下さい。	
3	注意	通信終了確認画面の表示文言はBREWアプリの企画内容に合わせてコンテンツプロバイダにて任意に決定して下さい。	

## 9. 6. BREW アプリ側で配慮が必要なエラー画面

### 9. 6. 1. BREW アプリ側で表示が必要なエラー画面

BREW プラットフォームは、ez アプリ（JAVA）とは異なり、端末側や BREW 側でエラーメッセージを表示する仕組みがなく、BREW からアプリケーションにエラーコードが通知されるのみです。（表 12 および表 13 のケースは除く。）したがって、BREW アプリはエラーコードを通知された場合には、ユーザに混乱を与えないよう適切なエラーメッセージ表示し、ユーザが希望するアプリケーション動作が継続できるようユーザを誘導して下さい。

以下に、想定されるエラーケースと各ケースにおけるエラー表示例を示しますので、BREW アプリ側で企画に添ったエラーメッセージを用意して下さい。

表 10 エラーケースとエラー表示例

No.	分類	エラーケース	想定されるシーン	エラー表示例
1	メモリ	メモリ不足	各種API使用時にヒープ領域不足によるエラーが発生	「メモリが足りません。他のアプリを終了してから起動して下さい。」
2	ファイル	EFS不足 (新規作成)	ユーザ契機、BREWアプリ契機によりファイルを新規に作成する場合にEFS不足によるエラーが発生	「メモリが足りません。いくつかのファイルを削除して下さい」
3	ファイル	EFS不足 (リネーム)	ユーザ契機、BREWアプリ契機によりファイルのリネームを行う場合にEFS不足によるエラーが発生	「メモリが足りません。いくつかのファイルを削除して下さい」
4	ファイル	EFS不足 (ディレクトリ作成)	ユーザ契機、BREWアプリ契機によりディレクトリの作成を行う場合にEFS不足によるエラーが発生	「メモリが足りません。いくつかのファイルを削除して下さい」
5	ファイル	EFS不足 (書込み)	ユーザ契機、BREWアプリ契機によりファイルの上書き保存を行う場合にEFS不足によるエラーが発生	「メモリが足りません。いくつかのファイルを削除して下さい」
6	ファイル	ネットワーク系エラー	ユーザ契機、BREWアプリ契機により通信を行い、ユーザ回避不可能なエラーが発生	「設定時間内に接続できませんでした」
7	その他	ユーザ設定	BREWアプリ仕様によりユーザへ設定や入力を指示する場合で、その設定項目によりエラーが発生した場合	「URLが不正です」「そのファイルは未サポートです」「宛先を確認して下さい」などエラーの理由通知し、ユーザに次の操作へ誘導する
8	その他	圏外時通信（ユーザ操作時）	ユーザ操作時に圏外であった場合	「電波状態の良いところでリトライして下さい」

### ✓ チェックリスト

No.	種別	内容	備考
1	注意	本表には、各BREWアプリで共通的に遭遇すると想定されるケースのみを参考記載しています。	
2	注意	エラー画面表示文言は一例であり、BREWアプリの企画内容に沿ったかたちで適切な文言を表示して下さい。	

### 9.6.2. 最大ワークエリアの確保ができない場合のエラー画面【エラー画面表示：必須】

BREW アプリを起動、もしくはレジュームを行う際に以下の手順にて事前に残メモリについてチェックし起動を制限します。このチェックがうまく機能しない場合、残容量以上のメモリを使用しようとしてしまう可能性があり不足を補うためにサスペンドされているBREWアプリが強制的に終了されたり、動作が不安定になる恐れがありますので必ず対応するようにして下さい。

＜手順 1＞malloc 出来る最大長を調査する

＜手順 2＞利用できる残メモリを調査する

本チェック機構により BREW アプリに起動を制限する場合には下記のように起動できない理由を添えたメッセージを表示し、そのまま ANY キーにてアプリを終了して下さい。

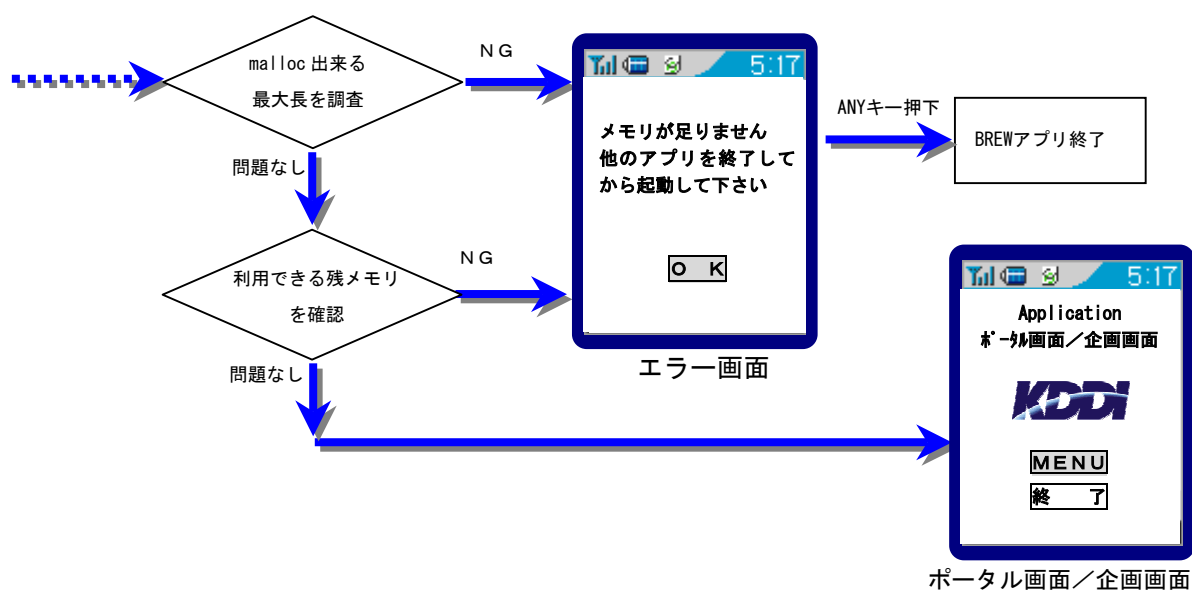


図 17 ワークエリアの確保に関する画面遷移例

#### ✓ チェックリスト

No.	種別	内容	備考
1	必須	通信フローの中で上図に記載の戻り値があった場合にBREWアプリはエラー画面を表示して下さい。	
2	必須	エラー画面において文言は「メモリが足りません。他のアプリを終了してから起動して下さい」とすることを原則とします。	
3	必須	BREWアプリの企画内容により語尾を変化させることは許容しますが、容易にその内容が類推できる範囲で変化させて下さい。(例:「メモリが足りないの。他のアプリを終了してから起動してね。お願い!」)	
4	必須	エラー画面表示後は、ユーザのANYキー押下にてアプリを終了して下さい。	

## 9.6.3. IWEB 通信時のエラー画面表示 [エラー画面表示：必須]

## (1) コネクトタイムアウト時のエラー画面

コネクトタイムアウトは、通信の接続が確立されるまでにBREWアプリが許容する最大待ち時間です。BREW AEEは、通信の接続が確立していない状態でBREWアプリから通信が要求されると、このタイムアウトの期間が満了するまで接続の確立を継続して試みます。この値のデフォルトは30秒です。BREWアプリがこの通信の接続に失敗し、下図に記載のエラーを受領した場合には、ユーザに対して通信ができないことを示す確認画面表示を行って下さい。

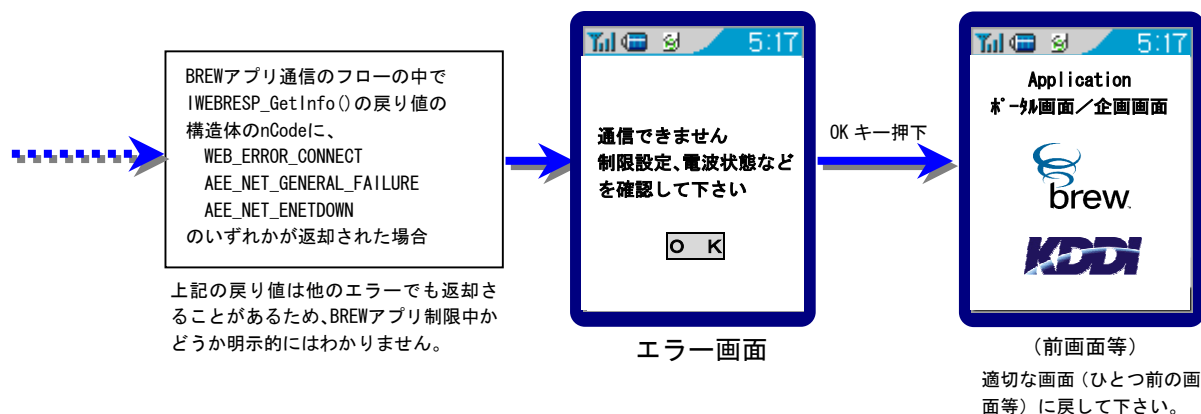


図 18 IWEB 通信画面遷移例（コネクトタイムアウトの場合）

## ✓ チェックリスト

No.	種別	内容	備考
1	必須	通信フローの中で上記に記載の戻り値があった場合にBREWアプリはエラー画面を表示して下さい。	
2	必須	エラー画面において文言は「通信できません。制限設定、電波状態などを確認して下さい」とすることを原則とします。	
3	必須	文字の大きさ、全半角、句点の有無、配置（改行処理）、ソフトキーの表示文言は規定しませんが、わかりやすく表示して下さい。	
4	必須	BREWアプリの企画内容により語尾を変化させることは許容しますが、容易にその内容が類推できる範囲で変化させて下さい。（例：「通信できないよーだ！制限設定とか電波状態などを確認してねー！」）	
5	注意	エラー画面表示後、OKキー（または同等のキー）が入力された場合は、ユーザにとって適切な画面（ひとつ前の画面等）に戻して下さい。	
6	参考	BREWには、端末側の「BREWアプリ通信制限設定」がONかOFFかを取得するAPIがないため、アプリケーションは通信を試みた後にErrorが返ってきて初めて端末側が通信制限状態であることを知ることになります。上記のように戻り値が一般的なErrorであることから、BREWアプリ側は通信制限による通信失敗なのか、他の原因での通信失敗なのかは明示的にはわかりません。	

## (2) ISHELL\_SetTimer() が満了した場合

コネクトタイムアウト（デフォルト 30 秒）までにエラーが返らず、ISHELL\_SetTimer()によりセットされたタイマが満了した場合は、Connection 確立後の通信エラーであり「通信中の圏外移行」や「ハンドオーバー」等の状態になっています。この時も、必ず確認画面を表示して下さい。この時、IWeb、IWebResp 等、必要に応じてリソースを解放が必要です。

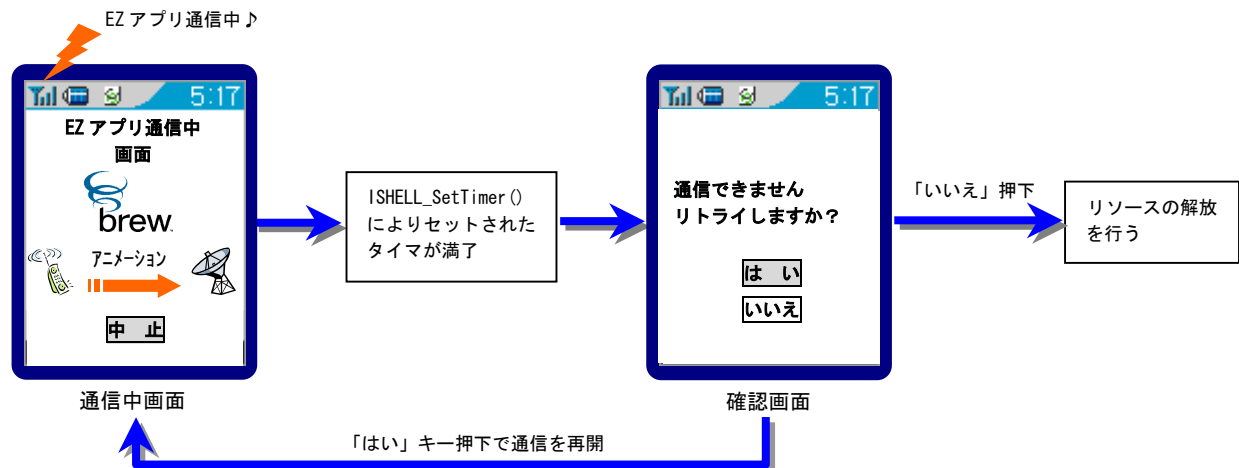


図 19 IWEB 通信画面遷移例 (ISHELL\_SetTimer() が満了した場合)

## ✓ チェックリスト

No.	種別	内容	備考
1	必須	エラー画面において文言は「通信できません。リトライしますか?」とすることを原則とします。	
2	必須	BREWアプリの企画内容により語尾を変化させることは許容しますが、容易にその内容が類推できる範囲で変化させて下さい。(例:「接続できないから、リトライしてみる?」)	
3	注意	エラー画面表示後、ユーザが通信継続を望まない場合は、適切にリソースの解放を行って下さい。	

## 【参考】正常にデータが受領できた場合

IWEB\_GetResponse のコールバックがタイムアウト満了前に呼び出されたとき、上記のコールバック待ちタイマをキャンセルする必要があります。ISHELL\_SetTimer 関数で設定したタイマをキャンセルするには、ISHELL\_CancelTimer 関数を使用します。

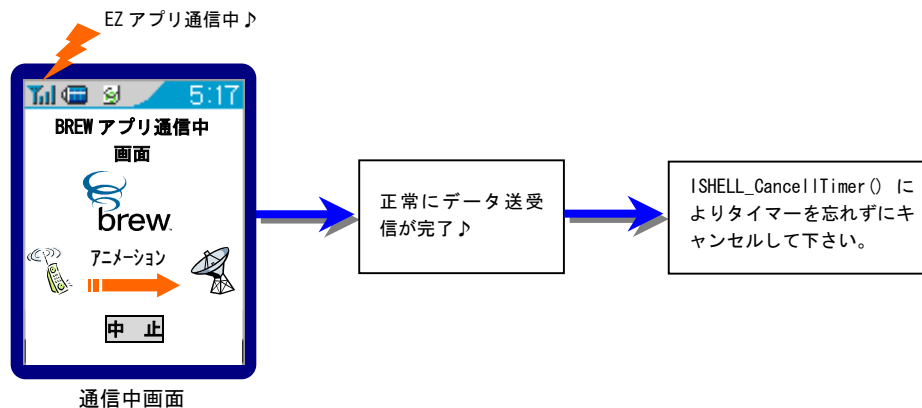


図 20 IWEB 通信画面遷移例 (正常にデータが受領できた場合)



## 9.6.4. ISOCKET インターフェースを利用した通信におけるエラー画面 [エラー画面表示：必須]

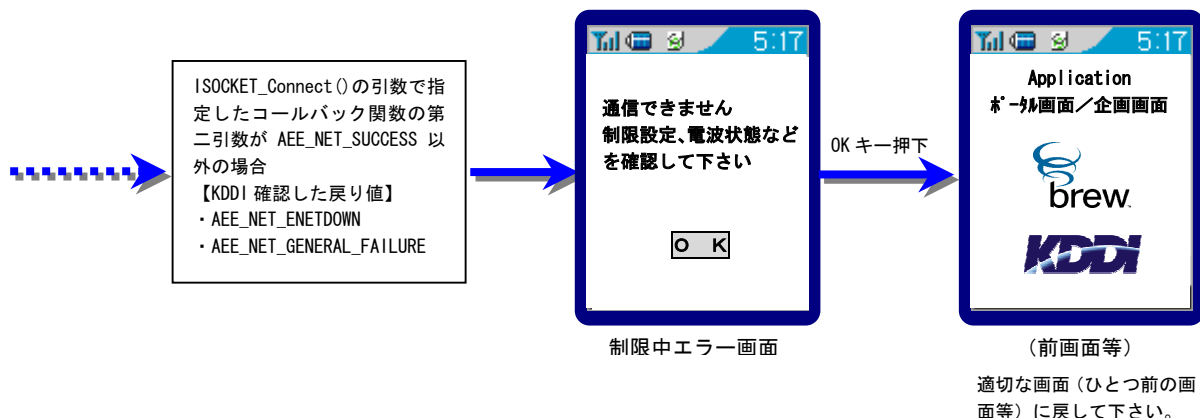


図 21 ISOCKET 通信時の画面遷移例

## ✓ チェックリスト

No.	種別	内容	備考
1	必須	通信フローの中で図 21に記載の戻り値があった場合にBREWアプリはエラー画面を表示して下さい。	
2	必須	エラー画面において文言は「通信できません。制限設定、電波状態などを確認して下さい」とすることを原則とします。	
3	必須	文字の大きさ、全半角、句点の有無、配置（改行処理）、ソフトキーの表示文言は規定しませんが、わかりやすく表示して下さい。	
4	必須	BREWアプリの企画内容により語尾を変化させることは許容しますが、容易にその内容が類推できる範囲で変化させて下さい。（例：「通信できないよーだ！制限設定とか電波状態などを確認してねー！」）	
5	注意	エラー画面表示後、OKキー（または同等のキー）が入力された場合は、ユーザにとって適切な画面（ひとつ前の画面等）に戻して下さい。	
6	参考	BREWには、端末側の「BREWアプリ通信制限設定」がONかOFFかを取得するAPIがないため、アプリケーションは通信を試みた後にErrorが返ってきて初めて端末側が通信制限状態であることを知ることになります。図 21に記載したように戻り値が一般的なErrorであることから、BREWアプリ側は通信制限による通信失敗なのか、他の原因での通信失敗なのかは明示的にはわかりません。	



9.6.5. BREW GPS 通信時のエラー画面表示 [エラー画面表示：必須]

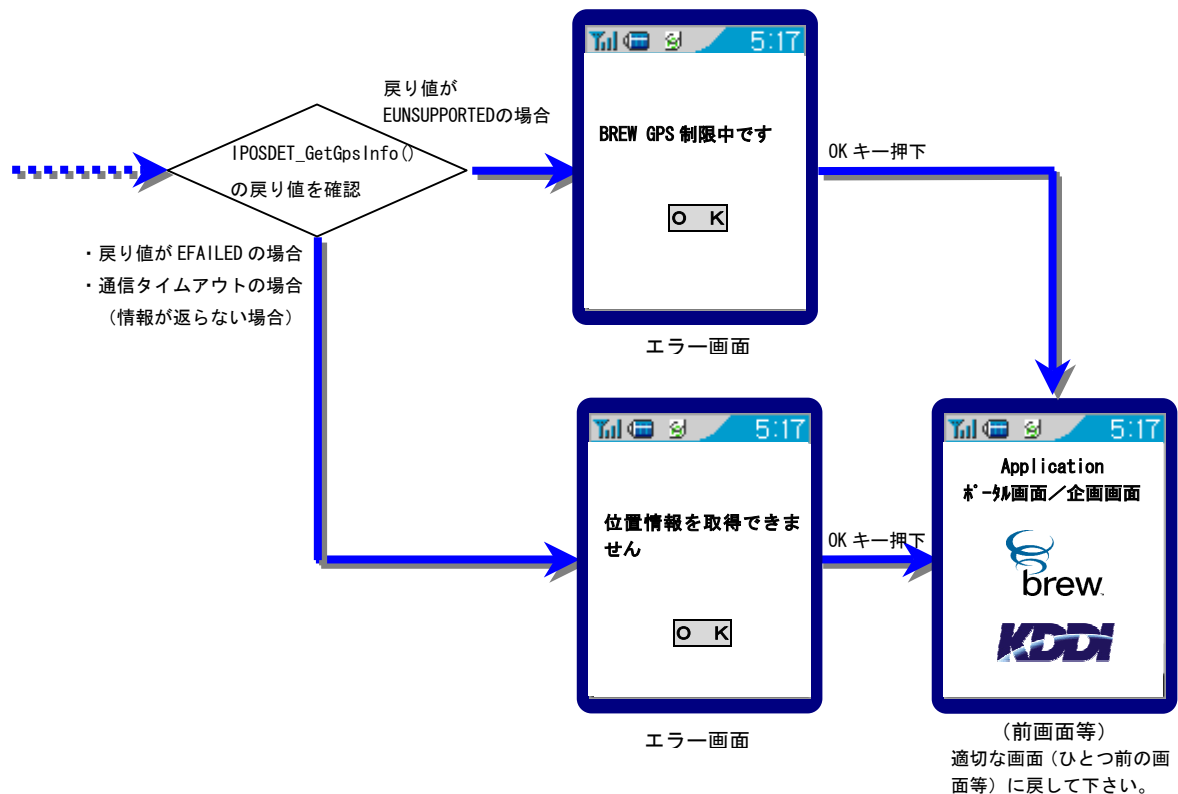


図 22 BREW GPS 通信制限時の画面遷移例

✓ チェックリスト

No.	種別	内容	備考
1	必須	図 22に記載の戻り値があった場合にBREWアプリはエラー画面を表示して下さい。	
2	必須	エラー画面において文言は「BREW GPS制限中です。」「位置情報を取得できません。」とすることを原則とします。	
3	必須	文字の大きさ、全半角、句点の有無、配置（改行処理）、ソフトキーの表示文言は規定しませんが、わかりやすく表示して下さい。	
4	必須	Applicationの企画内容により語尾を変化させることは許容しますが、容易にその内容が類推できる範囲で変化させて下さい。（例：「BREW GPS制限中なのだ」）	
5	注意	エラー画面表示後、OKキー（または同等のキー）が入力された場合は、ユーザにとって適切な画面（ひとつ前の画面等）に戻して下さい。	
6	参考	BREW GPS制限が設定されていた場合のIPOSDET_GetGPSInfoの戻り値は表 11の通りとなります、	

表 11 IPOSDET\_GetGPSInfo 戻り値

ピクト表示	制限なし	BREW GPS 通信制限中	BREW アプリ 通信制限中	BREW GPS 通信制限 +77° 通信制限中	EZweb 制限
強電界 3 本	測位値が返る	EUNSUPPORTED	測位値が返る	EUNSUPPORTED	測位値が返る
弱電界 0 本	通信タイムアウト	EUNSUPPORTED	通信タイムアウト	EUNSUPPORTED	通信タイムアウト
圏外表示中	EFAILED	EUNSUPPORTED	EFAILED	EUNSUPPORTED	EFAILED

#### 9.6.6. 【重要】EULA への明記が必要な事項（必須）

BREW アプリが端末内情報を取得したり、端末外に送出する場合には、EULA (End User License Agreement) の中に次の事項を明示して下さい。なお、EULA にこの事項が明示されていない場合、当該 BREW アプリを ADS (Application Download Server) に登録できない場合があります。

- 端末内情報を取得する（端末外に送出する）BREW アプリである旨
- 端末内情報を取得する（端末外に送出する）目的
- 取得する（端末外に送出する）端末内情報の内容または範囲

## 10. (付録 1) BREW アプリ全般における文字制限

BREWアプリ全般における文字制限は下記の通りです。ただし、アプリケーション名については、付録2の通り、別途制限がありますので、そちらに従ってください。

## (1) BREWアプリにおいて文字数制限があるフィールド

フィールド名	文字数	文字数の計算方法
利用規約	2000[文字]	半角・全角とも1文字として計算
料金情報	350[文字]	半角・全角とも1文字として計算

## (2) BREWアプリにおいて使用できる文字

全半各別	文字種類
半角	英数字、記号、カタカナ
全角	英数字、記号（下記(3)の通り一部使用できない記号あり） カタカナ、ひらがな、JIS第一水準漢字、KDDI絵文字

上表記載の全角・半角文字を混在させることができますが、文字数の計算方法にはご注意ください。  
JIS第二水準、JIS第三水準、第四水準漢字、ギリシャ文字、ロシア文字は原則として使用しないで下さい。

## (3) BREWアプリにおいて使用できない記号一覧表（2版）

記号	使用できない記号															
S-JIS	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0x8140		、	。	，	．	・	：	；	？	！	“	”	‘	’	…	^
0x8150	—	—	、	、	、	、	、	、	、	、	○	—	—	—	／	＼
0x8160	～			…	…	‘	’	“	”	( )	[ ]	[ ]	[ ]	[ ]	{ }	
0x8170	}	<	>	《 》	「 」	『 』	【 】	+	—	±	×					
0x8180	÷	=	≠	<	>	≤	≥	∞	∴	♂	♀	°	′	″	℃	¥
0x8190	\$	¢	£	%	#	&	*	@	§	☆	★	○	●	◎	◇	◆
0x81A0	□	■	△	▲	▽	▼	※	〒	→	←	↑	↓	=	・	・	・
0x81B0	・	・	・	・	・	・	・	・	∈	≡	≤	≥	⊂	⊃	⊆	⊇
0x81C0	・	・	・	・	・	・	・	・	∧	∨	¬	⇒	⇔	∀	∃	・
0x81D0	・	・	・	・	・	・	・	・	・	・	∠	⊥	⌒	∅	▽	≡
0x81E0	≡	≪	≫	√	∞	∞	∴	∫	∫	・	・	・	・	・	・	・
0x81F0	À	%	#	b	♪	†	‡	¶	・	・	・	・	○			

## 罫線素片

S-JIS	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0x8490																—
0x84A0		┌	┐	└	┘	├	┤	┴	┴	┴	┴	┴	┴	┴	┴	┴
0x84B0	┐	└	├	┤	┴	┴	┴	┴	┴	┴	┴	┴	┴	┴	┴	┴

囲み英数字／アラビア数字／単位記号等

S-JIS	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0x8740	①	②	③	④	⑤	⑥	⑦	⑧	⑨	⑩	⑪	⑫	⑬	⑭	⑮	⑯
0x8750	⑰	⑱	⑲	⑳	I	II	III	IV	V	VI	VII	VIII	IX	X	・	ミリ
0x8760	キロ	セン	メートル	グラム	トン	アル	ヘクタール	リットル	ワット	カロリー	ドル	セント	パーセント	ミリ	ベー	mm
0x8770	cm	km	mg	kg	cc	m <sup>2</sup>	・	・	・	・	・	・	・	・	平成	
0x8780	”	”	No.	KK.	TEL	Ⓐ	Ⓑ	Ⓒ	Ⓓ	Ⓔ	(株)	(有)	(代)	明治	大正	昭和
0x8790	≡	≡	∫	∮	Σ	√	⊥	∠	└	└	∴	∩	∪	・	・	・

## 11. (付録2)アプリケーション名における文字制限

BREWアプリのアプリケーション名における文字制限は次の通りです。

## (1) アプリケーション名の文字数制限

フィールド名	文字数	文字数の計算方法
アプリケーション名	20[byte]	半角=1byte、全角=2byteとして計算

## (2) アプリケーション名に使用できる文字

全半各別	文字種類
半角	英数字、記号、カタカナ
全角	英数字、記号（下記(3)の通り一部使用できない記号あり） カタカナ、ひらがな、JIS第一水準漢字

上表記載の全角・半角文字を混在させることができますが、文字数の計算方法にはご注意ください。  
JIS第二水準、JIS第三水準、第四水準漢字、ギリシャ文字、ロシア文字は原則として使用しないで下さい。

## (3) アプリケーション名に使用できない記号一覧表（2版）

記号	使用できない記号															
S-JIS	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0x8140		、	。	、	、	、	、	、	、	、	、	、	、	、	、	、
0x8150	—	—	、	、	、	、	、	、	、	、	、	、	、	、	、	、
0x8160	～			...	..	‘	’	“	”	(	)	[	]	[	]	{
0x8170	}	<	>	《	》	「	」	『	』	【	】	+	—	±	×	
0x8180	÷	=	≠	<	>	≤	≥	∞	∴	♂	♀	°	′	″	℃	¥
0x8190	\$	¢	£	%	#	&	*	@	§	☆	★	○	●	◎	◇	◆
0x81A0	□	■	△	▲	▽	▼	※	〒	→	←	↑	↓	=	・	・	・
0x81B0	・	・	・	・	・	・	・	・	∈	≡	≤	≥	⊂	⊃	⊆	⊇
0x81C0	・	・	・	・	・	・	・	・	∧	∨	¬	⇒	⇔	∇	∃	・
0x81D0	・	・	・	・	・	・	・	・	・	・	・	・	・	・	・	・
0x81E0	≡	≪	≫	√	∞	∞	∴	∫	∫	・	・	・	・	・	・	・
0x81F0	Å	‰	#	b	♪	†	‡	¶	・	・	・	・	○			

## 罫線素片

S-JIS	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0x8490																—
0x84A0		┌	┐	└	┘	┙	┚	┛	├	┤	┥	┦	┧	┨	┩	┪
0x84B0	┫	┬	┴	┴	┴	┴	┴	┴	┴	┴	┴	┴	┴	┴	┴	┴

囲み英数字／アラビア数字／単位記号等

S-JIS	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0x8740	①	②	③	④	⑤	⑥	⑦	⑧	⑨	⑩	⑪	⑫	⑬	⑭	⑮	⑯
0x8750	⑰	⑱	⑲	⑳	I	II	III	IV	V	VI	VII	VIII	IX	X	・	ミリ
0x8760	キロ	セン	メートル	グラム	トン	アル	ヘクタール	リットル	ワット	カロ	ドル	セント	パーセント	ミリ	ベー	mm
0x8770	cm	km	mg	kg	cc	m <sup>2</sup>	・	・	・	・	・	・	・	・	平成	
0x8780	”	”	No.	KK.	TEL	上	中	下	左	右	(株)	(有)	(代)	明治	大正	昭和
0x8790	≡	≡	∫	∫	Σ	√	⊥	∠	∟	∠	∴	∩	∪	・	・	・

## 12. (付録 3) SmallFont について

SmallFontを使用する為には、以下の事を行う必要があります。

また、SmallFontは移動機が対応していないと使用することが出来ません。**各端末仕様書にてSmallFontに対応していることをご確認ください。**

削除

### (1) KDDI\_SMALL\_FONT.BID のインストール

SmallFont を BREW SDK 上で使用するためには、KDDI\_SMALL\_FONT.BID のインストールを行います。

インストール方法などの詳細は BREW2.1 日本拡張パッケージ内の readme.txt をご参照ください。

### (2) SmallFont の描画

SmallFont は、IDisplay インスタンスを作成した後、IDisplay\_Setfont() にて文字フォントを設定し、IDisplay\_DrawText() で描画します。

IDisplay\_SetFont() の第 2 引数 AEEFont には変換したいフォントを記述してください。

例えばノーマルフォントを SmallFont に置換したい場合は、「AEE\_FONT\_NORMAL」を第 2 引数に記述します。

そうすることによって IDisplay\_DrawText(, AEE\_FONT\_NORMAL, ...) で描画されるフォントが SmallFont になります。

Ex)

```
ISHELL_CreateInstance(pIShell, AEECLSID_KDDI_SMALL_FONT, &pSmallFont)
```

```
pOldFont = IDisplay_SetFont(pIDisplay, AEE_FONT_NORMAL, pSmallFont);
```

```
IDisplay_DrawText(... AEE_FONT_NORMAL, ...);
```

```
IDisplay_SetFont( pIDisplay, AEE_FONT_NORMAL, pOldFont );
```

**※IFONT\_Release() でインスタンスの解放を必ず行ってください。**

## 13. (巻末 1) IHTMLViewer サポートタグ一覧

## Structured Module

Start tag	Attribute	Values	BREW	note
<html>			○	
	version		×	
<head>			○	
	profile		×	
<style>			×	
	media	handheld   all	×	
	type	text/css or nothing	×	
<body>			○	
	background	<i>href or color</i>	×	
	bgcolor		×	
	text		×	
	vlink	color visisted link	×	
	link	color nonvisted link	×	
<title>			×	
<meta>			×	
	name		×	
	content	<i>text</i>	×	
	scheme	<i>text</i>	×	
	http-equiv		×	
<!-->			×	



Basic Text Module

Start tag	Attribute	Values	BREW	
<blink>			x	
<blockquote>			○	
	bgcolor	%color	x	
 			○	
	clear	all	x	
		left	x	
		right	x	
		none	x	
<center>			○	
	bgcolor	%color	x	
<div>			○	
	bgcolor	%color	x	
	align	right	○	
		center	○	
		left	○	
<font>			○	
	color	%Color	○	
<h1>to<h6>			○	Large font になる
	bgcolor	%color	x	
	align	left	○	
		center	○	
		right	○	
<hr>			○	
	size		x	
	width		x	
	align	left	x	
		center	x	
<marquee>			x	
	behavior	scroll	x	
		slide	x	
		alternate	x	
	direction	left	x	
		right	x	
	loop	1 through 20	x	
	scrollamount	pixels	x	
	scrolldelay	milliseconds	x	
	bgcolor	%Color	x	
<p>			○	
	bgcolor	%color	x	
	align	right	○	
		center	○	
		left	○	
<pre>			○	
	bgcolor	%color	x	
<plaintext>			△	<pre>扱い
	bgcolor	%color	x	
<span>			x	
	bgcolor	%color	x	

List Module

Start tag	Attribute	Values	BREW	
<dl>			○	
	bgcolor	%color	x	
<dt>			○	
	bgcolor	%color	x	
<dd>			○	
	bgcolor	%color	x	
<ol>			△	<ul>扱い
	bgcolor	%color	x	
	start		x	
	type	l	x	
		a	x	
		A	x	
<ul>			○	
	bgcolor	%color	x	
	type	disc	x	
		square	x	
		circle	○	
<li>			○	
	bgcolor	%color	x	
	type	l	x	
		a	x	
		A	x	
		disc	x	
		square	x	
		circle	○	
	value	number	x	
<menu>			△	<ul>扱い
	bgcolor	%color	x	
<dir>			△	<ul>扱い
	bgcolor	%color	x	
<u>			○	

Basic Table Module

Start tag	Attribute	Values	BREW	
<table>			△	<p>扱い
	border	%Size, thin, medium, thick	x	
	cellpadding	%Size	x	
	cellspacing	%Size	x	
	width	%Size	x	
	summary	text	x	
	bgcolor	%color	x	
<caption>			△	<p>扱い
	align	top	x	
		bottom	x	
		left	x	
		right	x	
<tr>			△	<p>扱い
	bgcolor	%color	x	
	align	right	x	
		left	x	
		center	x	
	valign	top	x	
		middle	x	
		bottom	x	
<th>			△	<p>扱い
	bgcolor	%color	x	
	align	left	x	
		right	x	
		center	x	
	colspan	number	x	
	rowspan	number	x	
	width	pixels	x	
	valign	top	x	
		middle	x	
		bottom	x	
<td>			△	<p>扱い
	bgcolor	%color	x	
	align	left	x	
		right	x	
		center	x	
	colspan	number	x	
	rowspan	number	x	
	valign	top	x	
		middle	x	
		bottom	x	
	width	pixels	x	

Link Module

Start tag	Attribute	Values	BREW	
<a>			○	
	accesskey	0to9, #, *	x	
	charset	character_encoding	x	
	href	url	○	
		mailto:/tel:	△	URL として処理、アプリケーションにリンク先情報が返る
	hreflang	language_code	x	
	name	section_name	x	
<base>			x	
	href	url	x	
<link>			x	
	href	url	x	
	type	CDATA	x	

Image Module

Start tag	Attribute	Values	BREW	
<object>			x	
	codebase	<i>url</i>	x	
	data	<i>url</i>	x	
	standby	<i>text</i>	x	
	type	<i>mimetype</i>	x	
	copyright		x	
<img>			○	
	src	<i>url</i>	○	
	alt	<i>text</i>	x	
	vspace	<i>pixels</i>	x	
	hspace	<i>pixels</i>	x	
	align	bottom	x	
		middle	x	
		top	x	
		left	x	
		right	x	
	width	<i>%Size</i>	○	
	height	<i>%Size</i>	○	
	localsrc	<i>CDATA</i>	x	
	border	<i>%Size</i>	x	
	copyright		x	
<param>			x	
	name	<i>size</i>	x	
		<i>disposition</i>	x	
		<i>title</i>	x	
	value	<i>value</i>	x	
	valuetype	<i>data</i>	x	
		<i>ref</i>	x	
		<i>object</i>	x	

# Forms

Start tag	Attribute	Values	BREW	
<form>			○	
	action	url	○	
	method	get	○	
		post	○	
<input>			○	
	type	text	○	
		checkbox	○	
		radio	○	
		password	○	
		hidden	○	
		submit	○	
		reset	○	
	name	<i>field_name</i>	○	
	value	<i>value</i>	○	
	size	<i>number_of_char</i>	○	
	maxlength	<i>number</i>	○	
	checked	checked	○	
	readonly	readonly	○	
	src	<i>url</i>	○	
	disabled	disabled	×	
	localsrc		×	
	accesskey	0to9, #, *	×	
	format	A, a, N, X, x, M, m	×	別途検討中
	liststyle	1, 2, 3, 4	○	
	emptyok		○	
<label>			×	
	for	<i>IDREF</i>	×	
<select>			○	
	disabled	disabled	×	
	multiple	multiple	×	Multiple-Selection lists are not Supported
	name	<i>text</i>	○	
	size	<i>number</i>	×	
<option>			○	
	disabled	disabled	×	
	value	<i>text</i>	○	
	selected	selected	○	
<optgroup>			×	
	disabled	disabled	×	
	label	<i>text</i>	×	
<textarea>			○	Multiple-line data is not Supported. (not contain CRLF)
	cols	<i>number</i>	×	
	rows	<i>number</i>	×	
	disabled	disabled	×	
	name	<i>textarea_name</i>	○	
	readonly	readonly	×	
	accesskey	0to9, #, *	×	
	maxlength		○	
	format	A, a, N, X, x, M, m	×	

※ <textarea></textarea> で囲まれた値をデフォルトとしてセットすることは出来ません。<textarea value="default"></textarea>とすることで"default"をデフォルトとして表示する仕様になっています。

## Other supported Elements

Start tag	Attribute	Values	BREW	
<bgsound>			×	
	src	URL	×	
	loop	n, infinite	×	

## 14. (巻末 2) 端末が表示するエラーメッセージ一覧

## 14. 1. BREW 側で表示する確認画面とエラー画面

## 14. 1. 1. BREW 実行環境が表示するエラー画面

表 124 BREW 実行環境が表示するエラー一覧

ID	エラー表示内容	原因	備考
1	「メモリが解放できません ※クリアキーを押してください」	アプリケーションが、実行中に割り当てたメモリをすべて解放していません。	BREW実行環境がメモリリークを検知した時（インスタンス未解放など）発生します。そもそもこれが発生するようではアプリに欠陥があることは明らかであり、商用サービス上起こってはいけないエラーです。
2	「アプリを起動できませんでした ※クリアキーを押してください」  「エラーが発生しました(001) ※クリアキーを押してください」	初期化エラーのため、アプリケーションの読み込みに失敗しました。これは、AEECreateInstance() 関数の使用中またはAPP_START イベント処理中に1 つ以上のオペレーションに失敗した場合に発生する可能性があります。  (001) モジュールのロードに成功したが、その後、ワークエリアの確保に失敗した場合。具体的には、AEECreateInstance もしくは EVT_APP_START にFALSEを返した場合。	ISHELL_StartAppletを行った際、このエラーに遭遇し、実行環境がエラーをポップアップする可能性があります。
3	「エラーが発生しました(002) ※クリアキーを押した後、一度電源を切り、電源を入れなおしてください」	アプリケーションを読み込んで実行するのに十分なメモリがありません。  (002) モジュールのロード自体に失敗した場合。具体的には、AEEModGen.c 内で ENOMEMORYを返した場合。	ISHELL_StartAppletを行った際、このエラーに遭遇し、実行環境がエラーをポップアップする可能性があります。
4	「エラーが発生しました(004) ※クリアキーを押してください」	シグネチャ検証プロセスが失敗しました。アプリケーションがデバイスにダウンロードされた後、アプリケーションが実行される前に、このプロセスが実行されます。	ISHELL_StartAppletを行った際、このエラーに遭遇し、実行環境がエラーをポップアップする可能性があります。これは、署名の検証失敗であり、 <u>アプリが不正に書き換えられている可能性があります</u> 。 もしくは、署名付加時に「変更予定なし」で署名したファイルをアプリが書き換えてしまった場合に発生します。
5	「このアプリは有効期限が切れています ※クリアキーを押してください」	アプリケーションの使用許可期限が切れています。	ISHELL_StartAppletを行った際、このエラーに遭遇し、実行環境がエラーをポップアップする可能性があります。

#### 14. 1. 2. BREW-OEM 層（端末）が表示する確認画面

端末側（BREW-OEM 層）にて表示する確認画面は表 13の通りです。BREW アプリがこの表に記載されている機能を利用する場合は、BREW アプリ側で確認画面表示を行う必要はありません。ただし、端末側で確認画面が表示される場合には BREW アプリが強制的にサスペンドされることを考慮に入れてアプリケーションを製作して下さい。

表 13 端末側（BREW-OEM 層）にて確認画面を表示する機能

No.	機能	確認画面表示 タイミング	表示概要（端末実装により内容が異なる場合があります）	表示対応
1	音声発信時	音声発信動作 直前	発信先電話番号を表示し、音声発信に対する同意を求める画面を表示します。（例：0312345678に発信します。よろしいですか？）	BREW-OEM層 （端末）
2	ダイヤル発信 制限時の発信 操作	ダイヤル発信 操作時	発信操作時に一時解除のためのロックコード入力画面を表示します。（例：ダイヤル発信制限中。ロックコード_ _ _ _）	BREW-OEM層 （端末）
3	アドレス帳制 限時の利用操 作	アドレス帳操 作時	アドレス帳操作時に一時解除のためのロックコード入力画面を表示します。（例：アドレス帳制限中。ロックコード_ _ _ _）	BREW-OEM層 （端末）

## 15. サンプルコード集

### 15.1. BREW アプリが使用予定の最大ワークエリアの確保

```
// -----
// 残容量をチェックする関数例
// -----
boolean CheckMemory(MyApp* pApp)
{
    IHeap* pIHeap;
    uint32 un32Max;
    AEEDeviceInfo di;

    pIHeap = NULL;
    un32Max = 0;

    if (SUCCESS != ISHELL_CreateInstance(pApp->a.m_pIShell,
                                         AEECLSID_HEAP, (void**)&pIHeap))
    {
        return FALSE;
    }

    if ( !IHEAP_CheckAvail(pIHeap, MAX_NEED_MALLOC_SIZE) )
    {
        IHEAP_Release(pIHeap);
        return FALSE;
    }

    ISHELL_GetDeviceInfo(pApp->a.m_pIShell, &di);
    un32Max = IHEAP_GetMemStats(pIHeap);

    if ( (di.dwRAM - un32Max) < MAX_NEED_HEAP_SIZE )
    {
        IHEAP_Release(pIHeap);
        return FALSE;
    }

    IHEAP_Release(pIHeap);

    return TRUE;
}

// -----
// 起動時のインスタンス作成の処理例
// -----
int AEEClsCreateInstance(AEECLSID Clsid, IShell * pIShell,
                        IModule * po, void ** ppObj)
{
    *ppObj = NULL;

    // Rewrite class id here.
    if(Clsid == AEECLSID_KD03CHECKMEM)
    {
        if(AEEApplet_New(sizeof(MyApp), Clsid, pIShell, po, (IApplet**)ppObj,
                        (AEEHANDLER)my_HandleEvent, (PFNFREEAPPDATA)myAppCleanup ) == TRUE)
        {
            // Add your code here .....
            MyApp* pApp = (MyApp*)*ppObj;

            if( myInitMyApp(pApp) == TRUE )
            {
                if( CheckMemory(pApp) )
                    return (AEE_SUCCESS);
            }
        }
    }
    return (EFAILED);
}
```

(次項に続く)



```

// -----
// イベントハンドラ設定
// -----
boolean my_HandleEvent(IApplet * pi, AEEEvent eCode,
                      uint16 wParam, uint32 dwParam)

{
    switch (eCode)
    {
    case EVT_APP_START:
        if( !CheckMemory(pApp) )
        {
            Error(pApp);
            // *** Go to terminate mode... ***
            return TRUE;    // return true for error message loop
        }

        // Add your code here .....
        return TRUE;

    case EVT_APP_RESUME:
        if( !CheckMemory(pApp) )
        {
            Error(pApp);
            // *** Go to terminate mode... ***
            return TRUE;    // return true for error message loop
        }
        // Add your code here .....
        return TRUE;

    case EVT_APP_SUSPEND:
        // Add your code here .....
        return TRUE;

    case EVT_APP_STOP:
        // Add your code here .....
        return TRUE;

    case EVT_...
        // Add your code here .....

    default...

    return FALSE;
}

```

## 16. (巻末 3) 用語集

### A

AEE (Application Execution Environment)  
アプレットを読み込んで実行できる環境のことを言います。

ADS (Application Download Server)  
BREW アプリを登録するサーバです。端末は本サーバから BREW アプリをダウンロードします。

アプリケーションマネージャ  
端末内の BREW アプリを一括して管理 (バージョンアップ、削除) する BREW アプリです。

### B

BAR ファイル [. bar]  
BMP、PNG などの画像ファイルやテキスト等、アプレットが使用するリソースファイルを一つにまとめたファイルです。

BCI (BREW Compressed Image) ファイル [. bci]  
Qualcomm 社により開発された圧縮ファイル形式です。このファイル形式により、復号および端末上での画像の表示がすばやく容易なものになります。BCI では、アニメーションもサポートしています。

Bluetooth  
モバイル PC または通常の固定 PC、携帯電話、及びその他の周辺機器との間を無線でつなぐ通信技術です。

BMP ファイル [. bmp]  
画像データフォーマットの一つです。白黒 (2 値) の画像からフルカラー (1677 万 7216 色) までの色数を指定できます。BREW 2.0 では 16bit (65536 色) までの色数を指定することができます。

BREW.net (BREW dot net)  
UDP または TCP プロトコルを使用してインターネット通信を行なう KDDI のネットワークです。BREW アプリに対応したコンテンツのダウンロードはこのネットワークを介して行なわれます。

BREW AppLoader  
PC 上で開発した BREW アプリを端末に転送するためのソフトウェアツールです。Tools Suite に含まれ、[Qualcomm 社 Carrier Extranet](#) からダウンロードすることができます。

BREW アプリカタログ  
BREW アプリを端末にダウンロードするための BREW アプリです。メニューから検索する機能やキーワードから検索する機能を有しています。

BREW 設定  
BREW メニューの表示設定や待受 BREW アプリ、一発起動の BREW アプリを設定するメニューです。

## BREW メニュー

起動したい BREW アプリや「[BREW アプリカタログ](#)」、「[BREW 設定](#)」を選択するメニューです。

## C

### C-MIDI (Compact- Musical Instrument Digital Interface) ファイル[.pmd]

携帯電話向けの音楽データフォーマットです。Qualcomm 社が提唱するアニメーションにも対応した MIDI ベースのマルチメディアフォーマットです。QCELP の音声コーデックにも対応し、最大 16 音のメロディと 1 音の QCELP が再生可能です。PMD と呼ぶ場合もあります。

### ClassID

BREW アプレットまたは BREW [Extension](#) ごとに付与される一意な 32bit の ID です。BREW では、ある BREW アプレットが別のアプレットまたは BREW Extension を呼び出す場合は ClassID を利用します。

## D

### Dormant

PPP セッションが確立されているが、無線チャネルは解放されている状態のと言い、接続呼が一定時間（60 秒程度）ない状態が続いた場合に、この状態になります。また、この状態が 5 分間継続した場合は、PPP セッションも解放されます。

### デバイス

携帯電話または端末のことを言います。

## E

### EFS (Embedded File System)

端末内のファイルシステムです。BREW フォルダとも言います。

### EULA (End User License Agreement)

一般に、製品を販売する企業とそれを購入して使用するユーザとの間で交わされる使用許諾契約書のことを言います。ユーザの権利や制限、万一の障害発生時の免責事項などが記述されます。BREW アプリをダウンロードする際に必ず表示し、ユーザにその内容に承諾してもらわなければなりません。

### Extension

BREW Applet の機能を拡張するために作られた外付けのプログラム、または、様々な BREW Applet において共通して利用する機能をパッケージ化した外付けのプログラムです。Extension 単体では、BREW Applet のようにユーザに見える形では動作しません。例えば 3D 機能を提供するものなどがあります。

### EZ ナビ

au (KDDI) が提供する位置情報サービスの名称です。ezweb 上のコンテンツや、EZ アプリ (JAVA) によって位置情報サービスを利用します。

#### EZweb Network

HTTP プロトコルを使用して通信を行なう KDDI のネットワークです。BREW アプリのダウンロードはこのネットワークを介して行われます。

## G

#### GUI (Graphical User Interface)

操作しやすいようにアイコンなどを使い、コンピュータ等をわかりやすく使用できるようにしたユーザインターフェースです。ユーザがアイコンをクリックするだけで、簡単にファイルの操作やアプリケーションソフトの起動などを実行できるようになっています。

#### GW (ゲートウェイ)

ネットワーク上で、媒体やプロトコルが異なるデータを相互に変換して通信を可能にする機器です。OSI 参照モデルの全階層を認識し、通信媒体や伝送方式の違いを吸収して異機種間の接続を可能にします。

## H

#### ヒープメモリ

BREW フォルダに保存された BREW アプリを読み込み実行するためのメモリです。

## I

#### ISP (Internet Service Provider)

一般に、インターネットプロバイダまたはプロバイダと呼ばれるインターネットへの接続サービスを提供する事業者のことを言います。

#### ItemID

BREW アプリ毎に付与される ID です。端末は ItemID 単位でダウンロードを行います。同一 BREW アプリでも、対応機種毎にアプリを作りかえる場合やアップグレード版を作成した場合、BREW アプリ毎に ItemID が付与されます。

## J

#### JPEG ファイル[. jpeg または . jpg]

自然画の圧縮に向けた圧縮方式を用いて圧縮された画像ファイルです。圧縮の際に、若干の画質劣化を許容する(一部のデータを切り捨てる)方式と、まったく劣化のない方式を選ぶことができ、許容する場合はどの程度劣化させるかを指定することができる。方式によりばらつきはありますが、圧縮率はおおむね 1/10~1/100 程度です。

## K

### 共有ディレクトリ (Shared Directory)

通常、BREW アプリは、自分自身（プログラム）が格納されているディレクトリ（サブディレクトリを含む）にしかアクセスできませんが、共有ディレクトリには複数の BREW アプリが共通してアクセスすることができます。例えば、ここに電話帳データを格納しておく、複数の BREW アプリからその電話帳にアクセスして、レコードを取得したりできるようになります。

## M

### MD5 (Message Digest #5)

広く一般に使われているメッセージ要約関数アルゴリズムのうちの 1 つで、RFC1321 で定義されています。任意長メッセージから 128bit の一方向ハッシュ値を生成します。RSA 暗号系の開発者の一人、Rivest 氏らによって開発され、32bit コンピュータ上で効率よく計算できるようにアルゴリズムが決められています。ほかに MD2 や MD4 という、類似のアルゴリズムも存在します。

### MIME Type

一般的には、ファイルのコンテンツ種別の定義情報を指します。BREW では、コンテンツ種別（e.g., "Text/html"）、ファイル拡張子（e.g., ".doc"）、スキーム（e.g., "http://"）を総称して使用されます。

### MIME Type Handler

一般的には、拡張子からファイルの種類を判別し、そのファイルに応じたアプリケーションを起動する管理機構のことを言います。BREW では、MIME Type に関連付けて BREW Class ID を管理する機構のことを言います。BREW アプリは MIME Handler に MIME Type を提示することで、同 MIME Type で定義された種別のファイルや処理を実行できる Applet やインターフェースの Class ID を知ることができます。MIME Handler という言い方をする場合もあります。

### mod ファイル (モジュールファイル) [.mod]

アプリケーション実行ファイルです。ARM BREW Builder で ARM コンパイルした際に、生成されます。

### MP3 ファイル [.mp3]

高圧縮率で高品質なサウンドデータ圧縮技術を使って圧縮されたサウンドデータファイルのことを言います。

### MSM (Mobile Station Modem)

通信および BREW アプリなどの実行を行なっている Qualcomm 社のチップセットです。

### モバイルショップ

ADS への接続、BREW アプリカタログの表示や BREW アプリのダウンロード等一連のダウンロード機能を担うアプリです。

## N

### Native (もしくは Native アプリケーション)

メーカーにより準備され、端末出荷時に搭載される UI (ユーザインターフェース) に相当するアプリケー

ションです。

## O

OEM(Original Equipment Manufacturer)

端末製造メーカーが BREW 実装時に BREW AEE 層接続する為に作成するレイヤー層です。

## P

PDE(Position Determination Entity)

ロケーションサーバとも言い、位置測位サーバのことを言います。

PNG(Portable Network Graphics) ファイル[.png]

アニメ画像などのベタ塗りの絵を圧縮するのに適した画像圧縮方式で圧縮された画像ファイルです。  
可逆圧縮(データが失われない)でありながら、圧縮率もきわめて高い圧縮方式で圧縮されます。

## Q

QCELP ファイル[.qcp]

Qualcomm 社が提唱する音声圧縮に優れたサウンドファイルです。

QSC ファイル[.qsc]

デバイス構成ファイルです。

## R

Radius

Livingston Enterprise 社が開発したダイヤルアップユーザの認証システムです。IETF によって RFC 2138 として標準化されています。電話回線などを通じてアクセスサーバにダイヤルアップしたユーザを認証し、割り当てるべき IP アドレスをアクセスサーバに伝えたり、課金情報を収集したりします。アクセスサーバとの間で認証用のユーザ ID やパスワードをやり取りする際には[MD5](#)を使った独自の方式を用いて、セキュリティを高めています。

Ringer Directory

着信メロディを格納するディレクトリです。このディレクトリは複数の BREW アプリが共通してアクセスすることができます。ただし、外国で使用されているものであり、現在、KDDI では使用を禁止しています。

リンク用 ItemID

WEB サイトから BREW アプリを呼び出す際のキーとなる ID です。BREW アプリのバージョンアップなどの際に、ItemID は変更されますが、リンク用 ItemID は変更されませんので、WEB サイトの修正は不要です。

## レジューム (Resume)

中断中の BREW アプリの復帰処理 (BREW が管理している BREW アプリについてのみのみ) のことを言います。

## ロケーションサーバ

→[PDE](#)

## S

## SDK (Software Development Kit)

ソフトウェア開発者向けに Qualcomm 社が無償で提供する開発キットです。開発ツールやサンプルプログラム、ドキュメントなどをパッケージしたものです。

## Shared Directory

→[共有ディレクトリ](#)

## Shim アプリケーション

BREW アプリと Native アプリケーション間をイベントにて中継するアプリケーションです。アドレス帳連携、ブラウザ連携、メーラ連携の shim アプリケーションが端末にデフォルトで搭載されています。

## sig ファイル[. sig]

ダウンロードアプリケーションなどダイナミックアプリケーションの起動に必要な署名ファイルです。

## SMAF ファイル[. mmf]

YAMAHA 社が提唱するアニメーションにも対応した MIDI ベースのマルチメディアフォーマットのファイルです。ADPCM 等音声コーデックにも対応した最大 32 音のメロディと 8 音の PCM が再生可能です。

## SMAF/Phrase ファイル[. spf]

SMAF をベースとしたマルチシーケンサファイルです。4 種類の音を同時に制御することが可能です。

## SSL (Secure Socket Layer)

Web ブラウザと Web サーバ間で安全な通信を行なうために Netscape Communications 社が開発したセキュリティ機能です。IETF による標準化も行なわれており、TLS (Transport Layer Security) として RFC2246 になっています。認証局の署名の入った証明書を使ったサーバの認証と Web ブラウザと Web サーバ間での通信内容の暗号化という 2 つの機能を持っています。

## サスペンド (Suspend)

BREW アプリの中断処理 (BREW が管理している BREW アプリについてのみのみ) のことを言います。

## スタックメモリ

プログラムで関数を呼び出した場合に、引数やリターンアドレスなどを一時的に保存するメモリです。

## ストレージエリア

BREW アプリは、自分自身が格納されているディレクトリ内に、ファイルのダウンロードや情報の保存をすることができます。このための保存領域をストレージエリアと言います。

## T

TAPI (Telephone Application Programming Interface)

BREW で用意されている API のうちのひとつです。BREW アプリから音声発信などを可能にする API のことを言います。

TRUE BREW

米国 Qualcomm 社が規定した検証に合格した BREW アプリに付与される称号です。KDDI の端末で動作させるためには、現在の運用では KDDI が規定した検証に合格する必要があります。

## U

UAM

Qualcomm 社が運用する BREW アプリの登録サーバです。UAM から世界の ADS へ BREW アプリが配信されます。

## W

WAP

携帯電話や腕時計などの携帯端末用の通信プロトコルです。Ericsson 社、Motorola 社、Nokia 社、Unwired Planet 社 (現 Openwave Systems 社) によって設立された WAP Forum によって策定されました。無線区間ではデータを圧縮して転送するなど、少ないリソースや遅い転送速度でも効率よく通信が行えるように工夫されています。WAP Forum では、WAP で送受信されるコンテンツを記述するためのタグ付け言語 WML も規定している。

WAP ブラウザ

携帯電話向けのデータ通信規格である WAP や WML に対応した情報閲覧用のアプリケーションソフトです。

WGS84

地球上の位置を緯度・経度で表すための基準に測地系があります。WGS84 は、この測地系の一つです。他に日本 (東京) 測地系がありますが、BREW で取得できるのは WGS84 による位置情報となります。



## 17. 更新履歴

更新日付	更新内容	項目番号
Ver1.0 作成 2003/5/25	初版作成	—