

Biz Collections

Biz Browser AI

Biz の宝箱

Tips集



## はじめに

- このドキュメントは、Biz-Collections –Bizの宝箱-サイトへ掲載されているBiz-Collections製品のTips集をネットワーク環境を問わずに参照していただくことが出来るようにPDFファイルとして編集したものです。  
このドキュメントが設計者、開発者の皆様のお役に立つことを心より願っております。
- 本ドキュメントの所有権、知的財産権、その他全ての権利および権限は、株式会社オープンストリームが所有しています。
- 本ドキュメントを株式会社オープンストリームの承諾を得ず、無断で複写、複製することを禁止します。
- 本ドキュメントに記載されている内容は、将来予告なく変更されることがあります。
- Windowsは米国マイクロソフト社の登録商標です。
- その他製品名は各社の商標または登録商標です。

NOTE !

**B** 1 Biz/Browser AIでの留意点

**A** 1 Biz/Browser AIをご利用の際は、以下の点をご留意ください。

**Biz/Browser AI特有の挙動/制約**

- 内部文字コードはUnicodeで、エンコード形式はUTF-8です。Stringオブジェクトにおける文字数の数え方や、外部データを取り込む際の標準形式がPC版、Mobile版とは異なります。詳細は、  
Biz/Browser AI CRSマニュアル > Biz/Browserの概要  
- Android版の特徴と制約  
- iOS版の特徴と制約  
をご参照ください。
- Biz/Browser AIが前面に表示されていない状態では、しばらく操作が行われないと、OSが自動的にBiz/Browserのプロセスを終了する場合があります。
- PC版と比べて、動作中のアプリケーションが使用可能なメモリサイズが小さいため、大きなデータや画像の取り扱いには注意する必要があります。
- IMEModeプロパティで、IMEの入力モードを制御することができません。SIP(Software Input Panel)の表示、非表示のみが適用されます。

**Android OS特有の挙動/制約**

- Biz/Browser Vと同等のセキュリティレベルの概念を搭載しています。規則については、Biz/Browser Vと同等となっていますので、詳細に関してはBiz/Browser CRSマニュアルの「Biz/Browserの概要-セキュリティレベルについて」をご参照ください。
- KeyDownイベントは、下記のキーで発生します。その他のキーでは保障されません。

キーの種類	格納される値
Enter	“RETURN”
Tab	“TAB”
音量+	“VOLUME_UP”
音量-	“VOLUME_DOWN”
Backキー	“BACK_BTN”

※Biz/Browser AI Ver2.1.0以降ではKeyDownイベントは発生しません

- FileSystemクラスでは、以下のディレクトリを基点に端末内のディレクトリにアクセスできます。パブリックルートには、Biz/Browser AI以外のアプリもアクセスできます。
  - ◆プライベートルート：
    - Biz/Browser AI Ver2.2.0以降  
/data/data/jp.co.opst.bizai.v2/files/FileSystem/private\_root/(サーバ名)
    - Biz/Browser AI Ver1.0.2.1迄  
/data/data/jp.co.axisoft.biz/files/filesystem/private\_root/(サーバ名)
  - ◆パブリックルート：
    - Biz/Browser AI Ver2.2.0以降  
/sdcard/Android/data/jp.co.opst.bizai.v2/files/FileSystem/root
    - Biz/Browser AI Ver1.0.2.1迄  
/sdcard/Android/data/jp.co.axisoft.biz/files/filesystem/root
- 現時点では、XMLDocumentクラスで日本語のタグ名を扱うことができません。
- Intentクラスを使用し、他のアプリと連携が可能です。

**iOS特有の挙動/制約**

- セキュリティレベルの概念がありません。セキュリティレベルにより動作が違うメソッド等では、レベルが指定されていない場合と同様になります。
- KeyDownイベントは発生しません。
- FileSystemクラスでは、下記のルートディレクトリ内にものみアクセスできます。また、他のアプリが下記ディレクトリにアクセスすることはできません。
  - ◆プライベートルート：  
(アプリのホームディレクトリ)/Library/Caches/FileSystem/private\_root/(サーバ名)
  - ◆パブリックルート：  
(アプリのホームディレクトリ)/Library/Caches/FileSystem/root/
- カスタムURLスキームを使用し、他のアプリと連携が可能です。

管理番号 : PntAi\_001

## NOTE !

**Q 2** 通信のタイムアウト時間について**A 2** Biz/Browser AIでは、通信のタイムアウト時間を以下のプロパティで設定できます。

Root.HttpTimeout プロパティ

Biz/Browserから行われる通信全体にタイムアウト時間を設定できます。

HttpRequest.Timeout プロパティ

個別の通信に対し、タイムアウト時間を設定できます。

設定されない場合、Root.HttpTimeout プロパティの設定に合わせます。

どちらのプロパティも設定されていない場合は、Android OS、iOSの標準の動作となり、サーバから応答が返るまで待ち続けます。

この場合、Biz/Browserはサーバの応答を待つ間に他の処理を行う事ができないため、何らかの不具合でサーバから応答が返らない場合、サーバの応答待ち状態から復帰することができなくなる可能性があります。

Root.HttpTimeout プロパティは未設定のままにせず、アプリケーションに合った値を設定することをお勧めします。

管理番号: PntAi\_002

起動

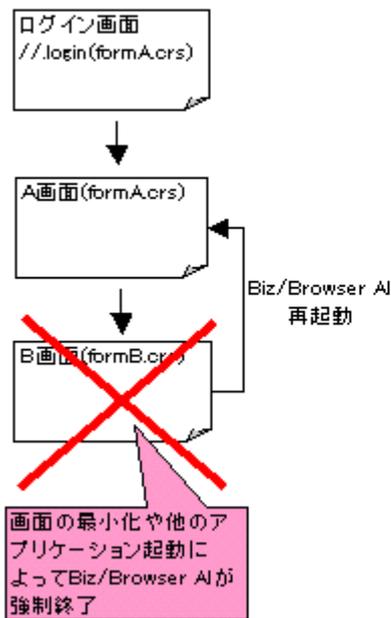
**Q 1** RootクラスのSetReLoginURLメソッドのログイン(遷移)先制御機能

**A 1** Biz/Browser AIでは、画面の最小化や他のアプリケーション起動によってBiz/Browser AIの画面が背面に回った場合、Biz/Browser AIが強制終了されてしまう場合があります。これは、Android OSによる挙動です。上記の場合、Biz/Browser AIを再起動すると最初にログインした(RootクラスのLoginメソッドを実行して遷移した)画面を表示します。ですが、RootクラスのSetReLoginURLメソッドを利用すると、Biz/Browser AIが強制終了された後に再起動した際の遷移先を任意の画面へ指定する事が可能です。

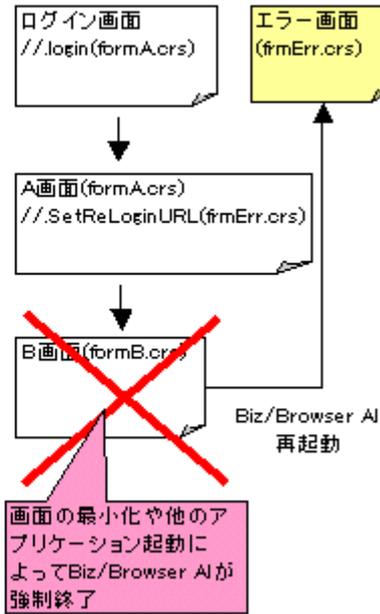
SetReLoginURLメソッドを利用する事によって、Biz/Browser AIが強制終了された場合にエラー画面や再ログイン画面へ遷移させるといった制御が可能となります。

[ 処理イメージ ]

**SetReLoginURLメソッドの記述がない場合**



**SetReLoginURLメソッドの記述がある場合**



管理番号: BotAi\_001

**Q** 1 Dialogを閉じる操作の方法によって処理を制御する **UP**

**A** 1 Biz/Browser AIでは、DialogクラスのCloseイベントオブジェクトに以下のような独自のプロパティが存在します。

**[追加されたプロパティ]**

```
Event{
    Number CauseClose; /* 閉じようとしている原因*/
}
```

CauseCloseプロパティにはどのように閉じられようとしているかを示す定数が格納されています。定数はAndroid OSで操作している場合と、iOSで動作している場合で違いがあります。

< Android OSの場合 >

Dialog.CLOSE\_ON\_BACK\_KEY : Android のBack キーで閉じようとした場合  
Dialog.CLOSE\_ON\_TOUCH\_OUTSIDE : ダイアログ外をタップして閉じようとした場合

< iOSの場合 >

Dialog.CLOSE\_ON\_CLOSE\_BUTTON : ダイアログの閉じるボタンで閉じようとした場合  
Dialog.CLOSE\_ON\_TOUCH\_OUTSIDE : ダイアログ外をタップして閉じようとした場合

CauseCloseプロパティを利用すると、Dialogを閉じる操作の方法によって処理を分ける事ができます。

**[サンプルコード]**

```
Dialog Dialog1 {
    Title = "Dialog1";
    Form Form1 {
        Width = 400;
        Height = 300;
    }

    Function OnClose( e ) {
        /* ダイアログ外をタップした場合を除き、ダイアログを閉じる */
        if(e.CauseClose != Dialog.CLOSE_ON_TOUCH_OUTSIDE){
            Dialog1.Delete();
        }
    }
}
```

**B** 1 KeyDownイベントが発生するキーについて

- A** 1 Biz/Browser AI Ver1.0xでは、以下のキーでKeyDownイベントが発生します。  
(※Biz/Browser AI Ver2.1.0以降では、KeyDownイベントは発生しません)

キーの種類	格納される値
Enter	“RETURN”
Tab	“TAB”
音量+	“VOLUME_UP”
音量-	“VOLUME_DOWN”
Backキー	“BACK_BTN”

OnKeyDownイベントハンドラをForm上に記述すると、上記のキーが押下された際のKeyDownイベントを捕捉し、イベントハンドラに記述した任意の処理を行う事が出来ます。

例えば、Backキーは、2回押下する事でBiz/Browser AIを終了します(※端末やOSに依存する動作です)が、1回目のBackキー押下時には、入力内容の保存の処理を行い、2回目のBackキー押下でBiz/Browser AIを終了といった動作をさせたい場合、以下のサンプルコードのように記述する事で実装する事が可能です。

**[サンプルコード]**

```
Form Form1 {
  X = 0;
  Y = 0;
  Width = 310;
  Height = 204;

  Number s_flg = 0;

  . . . . 省略

Function OnKeyDown( e ) {
  if(e.key == "BACK_BTN"){
    if(s_flg == 0){
      /* 画面入力内容を保存する処理等を記載 */
      s_flg = 1;
    }
    /* PostEventメソッドを実行し、イベントを上位へ伝播させる */
    PostEvent();
  }
}
```

**B** 2 端末の画面サイズに合わせてスケーリングしたい **NEW**

- A** 2 FormクラスのHorizontalScale、VerticalScale、FontScaleプロパティに、倍率を設定し、オブジェクトをスケーリングできます。

Biz/Browser AIでは、画面全体からステータスバー、メニューバー等を引いた描画可能な部分のサイズを、RootクラスのHeightプロパティ、Widthプロパティで取得できます。

この値を基準にRoot直下のFormのサイズを拡大し、元々のサイズと比較して、各オブジェクトを画面サイズに合わせるための倍率を求めます。

**[サンプルコード]**

```
Form Form1 {
    X = 0;
    Y = 0;
    Width = 220;
    Height = 267;

    Number w = Width;
    Number h = Height;

    /* Biz/Designerのデザインビューでは実行しない
    (エラーとなるため) */
    if ( !$DESIGNTIME ) {
        /* Formを画面描画サイズいっぱい広げる */
        Width &= //.Width;
        Height &= //.Height;
    }

    /* 縦横の倍率を算出 */
    HorizontalScale &= Width/w;
    VerticalScale &= height/h > 0.1 ? height/h : 0.1;
    /* フォントの倍率を算出 */
    FontScale &= VerticalScale > HorizontalScale
        ? HorizontalScale : VerticalScale;

    ...
}
```

なお、端末上でメニューから表示できる「スケーリング設定」画面からスケーリングの倍率を設定することもできます。  
この時、CRS上でもスケーリングの設定がされていると、倍率は重ねて適用されます。  
例えばCRSで1.5、端末の設定画面で2という倍率が設定されていれば、画面は1.5×2で、元の画面から3倍に拡大されて表示されることになります。

## CRSファイル形式のサンプル

「画面サイズに合わせたスケーリング(FormAi\_002.zip)」  
で実際の動作をご確認いただけます。

管理番号: FormAi\_002

**B** 1 FileSystemクラスでファイル操作を行う際の初期ディレクトリ(ルートディレクトリ)

**A** 1 OSごとのルートディレクトリは下記の通りです。  
また、FileSystemオブジェクトの生成時にコンストラクタで指定する属性によっても、ルートディレクトリにマップするディレクトリは変わります。

▼Android OSの場合

・パブリック属性

[初期化例]

```
var fs = new FileSystem(FileSystem.PUBLIC_ROOT);
```

[ルートディレクトリ]

- ・Biz/Browser AI Ver 2.2.0以降  
/sdcard/Android/data/jp.co.opst.bizai.v2/files/FileSystem/root
- ・Biz/Browser AI Ver 1.0.2.1迄  
/sdcard/Android/data/jp.co.axisoft.biz/files/filesystem/root

・プライベート属性

[初期化例]

```
var fs = new FileSystem(FileSystem.PRIVATE_ROOT);
```

[ルートディレクトリ]

- ・Biz/Browser AI Ver 2.2.0以降  
/data/data/jp.co.opst.bizai.v2/files/FileSystem/private\_root/(サーバ名)
- ・Biz/Browser AI Ver 1.0.2.1迄  
/data/data/jp.co.axisoft.biz/files/filesystem/private\_root/(サーバ名)

▼iOSの場合

※アプリのホームディレクトリは、端末ごとに違います。

・パブリック属性

[初期化例]

```
var fs = new FileSystem(FileSystem.PUBLIC_ROOT);
```

[ルートディレクトリ]

(アプリのホームディレクトリ)/Library/Caches/FileSystem/root

・プライベート属性

[初期化例]

```
var fs = new FileSystem(FileSystem.PRIVATE_ROOT);
```

[ルートディレクトリ]

(アプリのホームディレクトリ)/Library/Caches/FileSystem/private\_root/(サーバ名)

iOSの場合、Biz/Browser AIはルートディレクトリ以外の場所にあるファイルを操作することはできません。また、Biz/Browser以外のアプリがルートディレクトリのファイルを操作することもできません。  
これはOSの仕様上、アプリが自身に割り当てられた領域以外へアクセスすることを禁止しているためです。

管理番号:FileAi\_001

**B** 1 GPSやネットワークを利用して位置情報の取得を行う(Android) **UP****A** 1 Biz/Browser AIでは、GPSやネットワークを利用した位置測定が可能です。

Android、iOSのいずれのOSでも位置測定機能が利用できますが、指定する引数等、記述方法に違いがあります。

iOS上で使用する場合は以下のTIPSをご参照下さい。

・GPSやネットワークを利用して位置情報の取得を行う(iOS)

本TIPSでは、Android上で位置情報を取得するサンプルコードをご紹介します。

**[ 利用するプロパティ・メソッド ]****Runtimeクラス**

- ・ StartLocationMeasureメソッド  
引数で位置測位の方法( )を指定し、位置測位を開始します。  
位置測位中は、位置情報が更新される度にLocationChangedイベントが発生します。
- ・ LocationChangedイベント  
位置測位を開始した状態で、位置情報が更新されたときに発生するイベントです。  
イベントハンドラ[OnLocationChanged(e){...}]で、各種の位置情報を取得できます。
- ・ StopLocationMeasureメソッド  
位置測位を停止します。
- ・ IsProviderEnabledメソッド  
引数で位置測位の方法( )を指定し、測位に必要な機能が有効になっているかを取得します。
- ・ ShowLocationSettingsメソッド  
端末の位置情報サービス設定画面を表示します。

位置測位の方法(ロケーションプロバイダー)は次の2種類が使用できます。

- ・ Runtime.LOCATION\_PROV\_GPS :GPSを用いて測位します。
- ・ Runtime.LOCATION\_PROV\_NETWORK :ネットワークやWiFiを用いて測位します。

**[ サンプルコード ]**

```
TextBox latitude {
    ...
}
TextBox longitude {
    ...
}
Runtime rt1{

    /* LocationChangedイベントのイベントハンドラ */
    Function OnLocationChanged(e){

        if(e == null){
            /* 測定情報が取得できない場合 */
            //.MessageBox("位置情報が取得できませんでした");
        }

        /* 取得した緯度、経度をTextBoxへ表示 */
        ^.latitude.value = e.latitude;
        ^.longitude.value = e.longitude;

        /* 位置測位を停止 */
        StopLocationMeasure();
    }
}
Button Button1 {
    X = 11;
    Y = 129;
```

```
Width = 232;
Height = 27;
Title = "位置情報を取得";

Function OnTouch( e ) {

    if (Runtime.isProviderEnabled(Runtime.LOCATION_PROV_NETWORK)){
        /* ネットワークでの位置計測が有効になっている場合 */
        /* 位置計測を開始 */
        ^.rt1.startLocationMeasure(Runtime.LOCATION_PROV_NETWORK);

    }else{
        /* 有効になっていない場合 */
        /* メッセージを表示 */
        //.MessageBox("位置計測が無効になっています。有効にしてください。");
        /* 端末の位置情報サービス設定画面を表示 */
        Runtime.showLocationSettings();
    }
}
}
```

CRSファイル形式のサンプル

「GPSやネットワークを利用して位置情報の取得を行う(Android)(RtAi\_001.zip)」  
で実際の動作をご確認いただけます。

---

管理番号: RtAi\_001

[■ カテゴリメニューへ](#)

[次の TIPS へ >>](#)

[Biz-Collections Bizの宝箱 トップへ](#)

[Biz/Browser AI TIPS集 トップへ](#)

**Q 2** 他のアプリケーションと連携する(Android) **◀UP**

- A 2** Android OSは、アプリケーション(以下、アプリ)間で連携を行うために「Intent(インテント)」という機能を用意しています。  
これは、アクション(action=実行したい動作)、データ(data=連携したいデータ)を付加したインテントをアプリが発行すると、対応するアプリが起動する仕組みです。  
発行するインテントに付加された情報で、対象となるアプリが決定します。

例えばアクションに「ダイアルをする」、データに「電話番号」を付加したインテントのあるアプリから発行します。  
すると、端末にインストールされたアプリの中から、「ダイアルをする」動作と、「電話番号」を受け取ることが可能な電話アプリをOSが起動します。  
電話アプリが端末に複数インストールされていれば、アプリの選択画面を表示します。

Biz/Browser AIでは、Intentクラスを使用してインテントを扱い、他のアプリへインテントを発行することができます。  
例として、Biz/Browser AIから電話アプリを起動する場合のサンプルコードを紹介します。

本サンプルで発行するインテントの形式は、以下の通りです。

	値	意味
action	android.intent.action.DIAL	dataの値を元にダイアルを行う
data	tel:(電話番号)	値が電話番号であることを示す

**[サンプルコード]**

```
Form Form1 {
    (省略)

    Button Button1 {
        X = 16;
        Y = 84;
        Width = 232;
        Height = 27;
        Title = "電話番号を指定して電話アプリを起動";

        Function OnTouch( e ) {
            try {
                var action_dial = "android.intent.action.DIAL";
                var tel = "tel:00-0000-0000";

                /* インテントを生成 */
                var i = new Intent(action_dial, tel);
                /* インテントを発行 */
                i.startActivity();

            } catch(e) {
                //.MessageBox(e.message);
            }
        }
    }
}
```

管理番号: IntAi\_001

Biz/Browser AI 独自機能

**Q 3 Biz/Browser AI から他のアプリケーションを起動したい(Android) ◀UP**

- A 3** Biz/Browser AI から他のアプリケーション(以下、アプリ)を起動するには、Android OS のインテント機能を利用します。  
 (参考)  
 ・他のアプリケーションと連携する(Android)

サンプルでは、リストより選択した顧客データを元に、メールアプリと連携して顧客のアドレスにメール送信、及び地図アプリと連携して顧客の住所を地図で表示します。

Intentクラスを使用して、メールアプリや地図アプリと連携します。  
 本サンプルでは、インテントに以下の形式を使用します。

<メール送信ができるアプリを起動する>

	値	意味
action	Intent.ACTION_SENDTO	dataの値を元にデータ送信を行う
data	mailto:(メールアドレス)	値がメールの宛先であることを示す

<地図表示ができるアプリを起動する>

	値	意味
action	Intent.ACTION_VIEW	dataの値を元にデータを閲覧する
data	geo:0,0?q=(住所)	値が住所であることを示す

また、このサンプルではメール送信アプリの起動をするインテントへ、アクションとデータの他に「エクストラ情報(extra)」を付加しています。  
 エクストラ情報とは、起動先のアプリに渡すことができる追加情報です。  
 URLに付加するパラメータと似た仕組みで、キーと値の組み合わせで複数設定できます。

ここでは、メールの件名と本文をエクストラ情報に設定しています。

キー	値
android.intent.extra.SUBJECT	件名
android.intent.extra.TEXT	メール本文

アプリによって受け取れるエクストラ情報の形式は決まっているので、起動したいアプリの仕様を確認して下さい。

**[サンプルコード]**

```

FlexView FlexView1 {
    ...
    FlexRecord FlexRecord1{
    FlexLabel FlexLabel1{
        Title = "名前";
        ...
    }
    FlexLabel FlexLabel2{
        Title = "メールアドレス";
        ...
    }
    FlexLabel FlexLabel3{
        Title = "住所";
        ...
    }
    FlexButton FlexButton1{
        Value = "選択";
        ...
        Function OnPushed( e ) {
            var row = Form3.FlexView1.GetRow(e.row);
            ^.^.^Label1 = row.Flexlabel1.Value;;
            ^.^.^Label2 = row.Flexlabel2.Value;
            ^.^.^Label3 = row.Flexlabel3.Value;
        }
    }
}
    
```

```

    }
}
Label Label1 {
    ...
}
Label Label2 {
    ...
}
Label Label3 {
    ...
}
Button Button1 {
    ...
    Title = "メールを作成する";

    Function onTouch(e){
        try {
            /*インテントを作成します*/
            var i = new Intent("android.intent.action.SENDTO",
                "mailto:" + ^.Label2);

            /*インテントへExtra情報を格納します*/
            i.putExtra(Intent.TYPE_STRING, "android.intent.extra.SUBJECT",
                ^.Label1 + "様へ【sample】");
            i.putExtra(Intent.TYPE_STRING, "android.intent.extra.TEXT",
                ^.Label1 + "様\nサンプルメールです。 \nサンプルメールです。");

            /*アプリケーション選択画面を表示します*/
            i.createChooser("メーラーを選択してください").startActivity();

        } catch(e) {
            //.MessageBox(e.message);
        }
    }
}

Button Button2 {
    ...
    Title = "地図で見る";

    Function onTouch(e){
        try {
            var i2 = new Intent("android.intent.action.VIEW",
                "geo:0,0?q=" + ^.Label3);
            i2.createChooser("アプリケーションを選択してください")
                .startActivity();
        } catch(e) {
            //.MessageBox(e.message);
        }
    }
}

```

#### CRSファイル形式のサンプル

「顧客一覧から外部アプリケーションを起動するサンプル(IntAi\_002.zip)」  
で実際の動作をご確認いただけます。

---

管理番号: IntAi\_002

[◀◀ 前の TIPS へ](#)

[■ カテゴリメニューへ](#)

[次の TIPS へ▶▶](#)

[Biz-Collections Bizの宝箱 トップへ](#)

[Biz/Browser AI TIPS集 トップへ](#)

**Q 4** アプリケーションのメモリ使用量が一定量を超えたとき警告メッセージを表示したい **◀UP****A 4** Biz/Browser AIにはメモリ使用量が一定値を超えたことを検知するための機能があります。

- ・ Root.SetMemoryLimitThresholdメソッド  
...メモリ使用量警告イベントを発生させる値を設定する
- ・ Root.MemoryLimitExceededイベント  
...メモリ使用量がSetMemoryLimitThresholdメソッドで指定した値を超えた時に1度だけ発生するイベント

この機能を利用すると、現在の大きなメモリ使用量のデバッグや、メモリ不足の危険性を警告する等の処理ができます。

特にAndroid OSの場合、1つのアプリが使用できるメモリ領域について上限が決まっています。端末によってはこの上限が低く、メモリ不足による強制終了が発生しやすい場合があります。この上限値はRootクラスのGetMaxMemoryメソッドで取得することができますので、メモリ不足の危険性を警告する基準にご使用ください。

**[サンプルコード]**

```
if ( !$DESIGNTIME ) {  
    /* 閾値24.0MB*/  
    //.SetMemoryLimitThreshold( 24.0 );  
}  
Form Form1 {  
    . . . . . (略)  
}  
  
Function OnMemoryLimitExceeded( e ) {  
    //.MessageBox("警告：使用メモリサイズが：" + str(e.UsedMemory)+" [MB]を超えました");  
}
```

iOSの場合、1つのアプリが使用できるメモリ領域について上限は決まっておらず、OSが自動的に他のアプリを終了する等で調整します。そのため、iOS上でGetMaxMemoryメソッドを実行しても「-1」が返され、上限を取得することはできません。

管理番号: OrgAi\_004

Biz/Browser AI 独自機能

**B** 5 Android 端末の「戻る」ボタン (Back キー) 押下時の制御

- A** 5 Biz/Browser AI ver 1.0.2.0 より追加された、Root クラスの BackKeyMode プロパティを利用することで、Android 端末の「戻る」ボタン (Back キー) 押下時の制御が可能です。  
 (Biz/Browser AI Ver2.1.0 以降では KeyDown イベントが発生しないため、利用できません。)

BackKeyMode プロパティには 4 パターンの定数を指定でき、それぞれ以下のような動作を行います。

定数	動作※2
0	「戻る」ボタン押下で、ウィンドウを閉じます。
Root.ConfirmClose	1 度目の「戻る」ボタン押下で終了確認メッセージを表示し、2 度目の「戻る」ボタン押下でウィンドウを閉じます。
Root.RaiseKeyDown ※1	「戻る」ボタン押下で、KeyDown イベントを発行します。 OnKeyDown ハンドラの記述がない場合は、ウィンドウを閉じます。
Root.ConfirmClose + Root.RaiseKeyDown ※1	1 度目の「戻る」ボタン押下で終了確認メッセージの表示と KeyDown イベントの発行を行います。 OnKeyDown ハンドラの記述がない場合は、2 度目の「戻る」ボタン押下でウィンドウを閉じます。

- ※1 KeyDown イベントを OnKeyDown イベントハンドラで捕捉すると、ウィンドウを閉じる動作は行われません。OnKeyDown イベントハンドラの処理後ウィンドウを閉じる場合は、イベントハンドラ内で [PostEvent メソッド](#) を実行し、イベントを上位へ伝播させてください。
- ※2 「戻る」ボタン押下で、ウィンドウを閉じる際には、Close イベントが発生します。このイベントを OnClose イベントハンドラで捕捉すると閉じる動作はキャンセルされます。

A 画面では「データの初期化」、B 画面では「終了」のように設定するのであれば、以下のよう  
 に、A 画面には「Root.RaiseKeyDown」B 画面には、「0」または「Root.ConfirmClose」で  
 BackKeyMode プロパティを設定します。

[サンプルコード]

```

Opkg.crs
Package pkg {
    class myForm extends Form{
        /* KeyDown イベント発生時の共通処理 */
        Function OnKeyDown( e ) {
            if(e.key == "BACK_BTN"){

                /* myForm オブジェクト配下に存在する FlexView を検索 */
                var flexview_obj = FindChild(FlexView);

                if(flexview_obj != null){
                    /* 「戻る」ボタン押下時、FlexView の全行を削除*/
                    flexview_obj.ClearRows();
                }
            }
        }
    }
}
    
```

```

OMyFormA.crs
import "pkg.crs";
//. BackKeyMode = Root.RaiseKeyDown;
myForm MyFormA {
    Width = 334;
    Height = 351;
    FlexView FlexView1 {
        X = 3;
        Y = 153;
        Width = 251;
        Height = 134;

        CursorColor = $FFCCFF;
        CursorLineOpacity = 0;

        FlexRecord FlexRecord1 {
    
```

```

        FlexLabel FlexLabel1 {
            Title = "商品コード";
        }
        FlexLabel FlexLabel2 {
            Width = 150;
            Title = "商品名";
        }
    }
}
Function init(){
    var data = new CSVDocument();
    data << CSV{
        A0001, リンゴ
        A0002, みかん
        C0101, スイカ
        C0102, メロン
        D0201, もも
    };
    FlexView1 << data;
}

if ( !$DESIGNTIME ) {
    init();
}
}

```

#### ○MyFormB.crs

```

import "pkg.crs";
//.BackKeyMode = Root.ConfirmClose;
myForm MyFormB {
    Width = 334;
    Height = 351;
    FlexView FlexView1 {
        X = 5;
        Y = 163;
        Width = 251;
        Height = 134;

        CursorColor = $CCFFFF;
        CursorLineOpacity = 0;

        FlexRecord FlexRecord1 {
            FlexLabel FlexLabel1 {
                Title = "商品コード";
            }
            FlexLabel FlexLabel2 {
                Width = 150;
                Title = "商品名";
            }
        }
    }
}
Function init(){
    var data = new CSVDocument();
    data << CSV{
        E1001, 目薬
        E1002, 風邪薬
        E1003, 胃腸薬
        F2002, 湿布
        W2001, 包帯
    };
    FlexView1 << data;
}

if ( !$DESIGNTIME ) {
    init();
}
}

```

```
}  
}
```

「MyFormA」・「MyFormB」ともにユーザ定義クラス「myForm」のオブジェクトのため、KeyDownイベントが発生すると、OnKeyDownイベントハンドラ内の処理が実行されます。

ですが、BackKeyModeプロパティに「Root.ConfirmClose」を設定すると、「戻る」ボタンを押下してもKeyDownイベントは発生しません。  
よって、A画面では「戻る」ボタンを押下した際にFlexViewのデータ削除(データの初期化)を行いB画面ではウィンドウを閉じるといったように、「戻る」ボタンを押下した際の挙動を分ける事が可能となります。

また、B画面のBackKeyModeプロパティを「Root.RaiseKeyDown」へ変更すれば「戻る」ボタンの挙動をA画面と同様に設定する事も可能となりますので、例えば、同じ画面であってもシステムのログインユーザによってBackKeyModeプロパティの設定を変更すれば、「戻る」ボタンの動作をユーザごとに変更するといった実装も可能です。

CRSファイル形式のサンプル

「Android端末の「戻る」ボタン(Backキー)押下時の制御(OrgAi\_005.zip)」  
で実際の動作をご確認いただけます。

---

管理番号: OrgAi\_005

[◀◀ 前の TIPS へ](#)

[■ カテゴリメニューへ](#)

[次の TIPS へ ▶▶](#)

[Biz-Collections Bizの宝箱 トップへ](#)

[Biz/Browser AI TIPS集 トップへ](#)

**B** 6 スクロールバー幅を指定したい

- A** 6 Biz/Browser AI Ver1.0.xでは、RootクラスのScrollBarSizeプロパティで、アプリケーション全体に適用されるデフォルトのスクロールバー幅を指定できます。  
※Biz/Browser AI Ver2.xではスクロールバーが表示されません。

設定した値は、Biz/Browserを終了する、もしくはLoginメソッド、Logoutメソッドが実行されるまで有効です。

**[サンプルコード]**

```
// ScrollBarSize = 30;
```

なお、端末上でメニューから表示できる「スケーリング設定」画面からスクロールバー幅を指定することもできます。  
CRS上と、「スケーリング設定」画面の両方でスクロールバー幅が設定されている場合は、「スケーリング設定」画面の指定が優先されます。

---

管理番号: OrgAi\_006

**Q 7** 他のアプリケーションと連携する(iOS) **◀NEW**

- A 7** iOSは、アプリケーション(以下、アプリ)間での連携のために「カスタムURLスキーム」という機能を用意しています。  
これは、アプリから別のアプリを起動するとき、起動したいアプリが定めた「URLスキーム」でURLを開くという仕組みです。

例えばあるアプリから「itms-apps://itunes.com/apps/bizbrowserai」というURLを開くと、App Storeが起動して「Biz/Browser AI」のページが表示されます。  
この場合、「itms-apps」がApp StoreのURLスキームです。  
URLスキームの後には、起動するアプリに渡したい情報を設定します。この情報の書式はURLスキームによって違います。

Biz/Browserでは、RuntimeクラスのShellLinkメソッドで、カスタムURLスキームを利用することができます。

例として、App Storeを起動してBiz/Browser AIのページを表示する場合の、サンプルコードを紹介します。

**[サンプルコード]**

```
Form Form1 {
    (省略)

    Button Button1 {
        X = 16;
        Y = 84;
        Width = 232;
        Height = 27;
        Title = "AppStoreでBiz/Browser AIを見る";

        Function OnTouch( e ) {
            try {
                var rt = new Runtime;
                rt.ShellLink("itms-apps://itunes.com/apps/bizbrowserai");
            } catch(e) {
                //.MessageBox(e.message);
            }
        }
    }
}
```

---

管理番号: OrgAi\_007

**Q 8 Biz/Browser AI から他のアプリケーションを起動したい(iOS) ◀NEW**

- A 8** Biz/Browser AI から他のアプリケーション(以下、アプリ)を起動するには、iOS のカスタム URL スキームを利用します。  
 (参考)  
 ・他のアプリケーションと連携する(iOS)

サンプルでは、リストより選択した顧客データを元に、メールアプリと連携して顧客のアドレスにメール送信、及び地図アプリと連携して顧客の住所を地図で表示します。

Runtime.ShellLinkメソッド実行時、地図ソフトやメーカーなど、連携したいアプリケーションのURLスキームを指定することにより、Biz/Browser AI から他のアプリケーションを起動することが可能です。

本サンプルでは、以下のURLスキームを使用します。

アプリ	URLスキーム	書式
メール	mailto	mailto://(メールアドレス) 指定された情報を宛先として扱う
マップ	maps	maps://?q=(検索キー) パラメータ「q」で指定された情報を検索キーとして、検索結果を表示

**[サンプルコード]**

```

FlexView FlexView1 {
    ...

    FlexRecord FlexRecord1{
        Height = 40;
        FlexLabel FlexLabel1{
            Title = "名前";
            ...
        }
        FlexLabel FlexLabel2{
            Title = "メールアドレス";
            ...
        }
        FlexLabel FlexLabel3{
            Title = "住所";
            ...
        }
        FlexButton FlexButton1{
            Value = "選択";
            ...

            Function OnPushed( e ) {
                /* 「選択」ボタンが押下された行の各セルの値を
                取得してLabelへ表示 */
                ^.^.^Label1 = e.row.Flexlabel1.Value; /* 名前 */
                ^.^.^Label2 = e.row.Flexlabel2.Value; /* メールアドレス */
                ^.^.^Label3 = e.row.Flexlabel3.Value; /* 住所 */
            }
        }
    }
}

Label Label1 {
    ...
}

Label Label2 {
    ...
}

Label Label3 {
    ...
}

Button Button1 {
    ...
}
    
```

```

...
Title = "メールを作成する";
Function onTouch(e){
    try {
        if(^.Label2.Value != ""){
            var mailaddress = ^.Label2.Value;
            var rt = new Runtime;

            rt.ShellLink("mailto://" + mailaddress);
        }else{
            //.MessageBox("メールアドレスが指定されていません");
        }
    } catch(e) {
        //.MessageBox(e.message);
    }
}
}

Button Button2 {
    ...
    Title = "地図で見る";
    Function onTouch(e){
        try {
            if(^.Label3.Value != ""){
                var address = ^.Label3.Value;
                var rt = new Runtime;

                rt.ShellLink("maps://?q=" + address);
            }else{
                //.MessageBox("住所が指定されていません");
            }

        } catch(e) {
            //.MessageBox(e.message);
        }
    }
}
}

```

#### CRSファイル形式のサンプル

「顧客一覧から外部アプリケーションを起動するサンプル(OrgAi\_008.zip)」  
で実際の動作をご確認いただけます。

---

管理番号 : OrgAi\_008

[◀◀ 前の TIPS へ](#)

[■ カテゴリメニューへ](#)

[Biz-Collections Bizの宝箱 トップへ](#)

[Biz/Browser AI TIPS集 トップへ](#)

**B** 9 GPSやネットワークを利用して位置情報の取得を行う(iOS) **NEW**

**A** 9 Biz/Browser AIでは、GPSやネットワークを利用した位置測定が可能です。

Android、iOSのいずれのOSでも位置測定機能が利用できますが、指定する引数等、記述方法に違いがあります。

Android上で使用する場合は以下のTIPSをご参照下さい。

・[GPSやネットワークを利用して位置情報の取得を行う\(Android\)](#)

本TIPSでは、iOS上で位置情報を取得するサンプルコードをご紹介します。

**[ 利用するプロパティ・メソッド ]****Runtimeクラス**

- ・ StartLocationMeasureメソッド  
引数で位置測位の精度( )を指定し、位置測位を開始します。  
位置測位中は、位置情報が更新される度にLocationChangedイベントが発生します
- ・ LocationChangedイベント  
位置測位を開始した状態で、位置情報が更新されたときに発生するイベントです。  
イベントハンドラ[OnLocationChanged(e){...}]で、各種の位置情報を取得できます。
- ・ StopLocationMeasureメソッド  
位置測位を停止します。
- ・ IsProviderEnabledメソッド  
位置情報サービスが有効になっているかを取得します。
  - ロケーションプロバイダは指定できません。
  - アプリ個別の有効/無効設定は判定できません。

位置測位の精度は次の3種類が使用できます。

- ・ Runtime.LOCATION\_ACCURACY\_HIGH : 最高精度で測位します。
- ・ Runtime.LOCATION\_ACCURACY\_MEDIUM : 中精度(誤差数百m前後)で測位します。
- ・ Runtime.LOCATION\_ACCURACY\_LOW : 低精度(誤差数km前後)で測位します。

**[ サンプルコード ]**

```
TextBox latitude {
    ...
}
TextBox longitude {
    ...
}
Runtime rt1{

    /* LocationChangedイベントハンドラ */
    Function OnLocationChanged(e){

        if(e.isMeasuringFailed == 1){
            /* 測定情報が取得できない場合 */
            ///.MessageBox("位置情報が取得できませんでした");
        }

        /* 取得した緯度、経度をTextBoxへ表示 */
        ^.latitude.value = e.latitude;
        ^.longitude.value = e.longitude;

        /* 位置測位を停止 */
        StopLocationMeasure();
    }
}
Button Button1 {
    X = 11;
    Y = 129;
```

```
Width = 232;
Height = 27;
Title = "現在位置の測定";

Function OnTouch( e ) {

    if(Runtime.isProviderEnabled()){
        /* 位置情報サービスが有効になっている場合 */
        /* 位置計測を開始 */
        ^.rt1.startLocationMeasure(Runtime.LOCATION_ACCURACY_MEDIUM);

    }else{
        /* 有効になっていない場合 */
        /* メッセージを表示 */
        //.MessageBox("位置計測が無効になっています。有効にしてください。");
    }
}
}
```

#### CRSファイル形式のサンプル

「GPSやネットワークを利用して位置情報の取得を行う(iOS)(OrgAi\_009.zip)」  
で実際の動作をご確認いただけます。

---

管理番号: OrgAi\_009

[◀◀ 前の TIPS へ](#)

[■ カテゴリメニューへ](#)

[Biz-Collections Bizの宝箱 トップへ](#)

[Biz/Browser AI TIPS集 トップへ](#)

**B** 10 カメラで撮影した写真を画像として取得したい

**A** 10 iOS上でBiz/Browser AIをご利用の場合、下記のRuntimeクラスのShowImagePickerメソッドを利用してカメラで撮影した写真へアクセスする事が可能です。

- ShowImagePickerメソッド  
端末に保存された写真を画像として取得することができます。  
引数で、アルバムから選択するか、カメラで新しく撮影するか等を指定できます。

引数は次の4種類が使用できます。

- Runtime.SourceTypePhotoAlbum … フォトアルバムから選択します
- Runtime.SourceTypePhotoLibrary … フォトライブラリから選択します
- Runtime.SourceTypeRearCamera … カメラを起動し、撮影した写真を取得します
- Runtime.SourceTypeFrontCamera … フロントカメラ(液晶側)を起動し、撮影した写真を取得します

**[サンプルコード]**

```
Form OrgAi_010 {
    ...

    ImageLabel ImageLabel1 {
        ...
    }

    Button Button1 {
        ...
        Title = "フォトアルバムから¥r¥n写真を選択";

        Function OnTouch( e ) {
            ar rt = New Runtime;
            ar img = rt.ShowImagePicker( Runtime.SourceTypePhotoAlbum );
            .ImageLabel1.SetImage( img );
        }
    }

    Button Button2 {
        ...
        Title = "カメラで撮影した写真を¥r¥n取得";

        Function OnTouch( e ) {
            var rt = New Runtime;
            var img = rt.ShowImagePicker( Runtime.SourceTypeRearCamera );
            ^.ImageLabel1.SetImage( img );
        }
    }
}
```

**CRSファイル形式のサンプル**

「カメラで撮影した写真を画像として取得したい(OrgAi\_010.zip)」  
で実際の動作をご確認いただけます。

Androidをご利用の場合は、Intentを利用してカメラアプリと連携します。  
詳細は、以下のページをご参照ください。

[参考]

[他のアプリケーションと連携する\(Android\)](#)

管理番号:OrgAi\_010

**G** 11 AsReaderでスキャンしたバーコードデータの取得方法 ◀ NEW

- A** 11 Biz/Browser AI Ver2.2.1より、株式会社アスタリスクのiPhone/iPod Touch用バーコードスキャナー「[AsReader](#)」との連携が可能となりました。  
AsReaderでスキャンしたデータを、Biz/Browser AIで取得する事ができます。

バーコードスキャン時に発生する「ExternalMessageイベント」を捕捉し、スキャンしたデータを取得します。その際、以下のイベント及びイベントオブジェクトの子オブジェクトを参照します。

- ExternalMessageイベント  
AsReaderでバーコードをスキャンした際に発生します。  
Eventオブジェクトから、スキャンしたデータを取得することができます。
- what  
ExternalMessageイベントの子オブジェクトで、メッセージ送信元を特定する値が格納されます。メッセージ送信元がAsReaderである場合は、「1001」が格納されます。
- data  
ExternalMessageイベントの子オブジェクトです。  
AsReaderでスキャンしたバーコードのデータが格納されます。

**[サンプルコード]**

```
Form OrgAi_011 {
    ...
}

/* ExternalMessageイベントハンドラ */

/* Asreaderで読み取ったデータを受け取る
ExternalMessageイベントの子オブジェクト”what”を取得し
メッセージ送信元を特定します。
AsReaderでは”1001”が取得されます */
Function OnExternalMessage(e) {
    if(e.what == 1001) {
        /* ExternalMessageイベントの子オブジェクト”data”を
        参照し、読み取ったバーコードをTextBoxに表示します */
        OrgAi_011.TextBox1.value = e.data.toString();
    } else {
        //. MessageBox(“AsReaderを使用してください”);
    }
}
```

CRSファイル形式のサンプル

「[AsReaderでスキャンしたバーコードデータの取得方法\(OrgAi\\_011.zip\)](#)」  
で実際の動作をご確認いただけます。

---

管理番号: OrgAi\_011

## TabFrame/TabForm

## G 1 タブへ背景色の設定や、文字色の設定を行いたい

- A 1 Biz/Browser AI Ver1.0xをご利用の場合、Biz/Browser Vと同様に、TabFrameのタブの耳の色や文字色をプロパティの設定にて変更する事が可能です。  
(※iOS上で動作する場合は、タブは表示されません。)

TabFrameのタブの耳の色や文字色の設定は以下のプロパティで行います。

## [ TabFormクラスのプロパティ ]

TabFgColorプロパティ : タブ部分の文字の色を設定します。  
TabBgColorプロパティ : タブ部分の背景の色を設定します。

## [ サンプルコード ]

```
TabForm TabForm1 {
    Title = "文字赤";
    TabFgColor = $RED;
    TabBgColor = $WHITE;
}
```

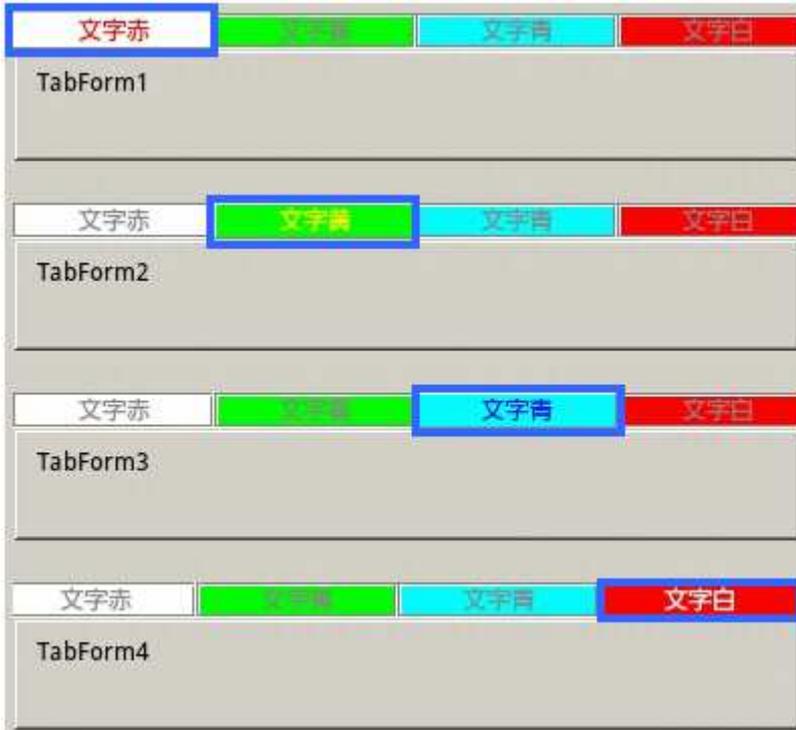
## [ 画面表示例 ]

青い四角で囲まれた部分が、タブ選択時の表示です。

・通常のタブ表示



・背景色、文字色を設定したタブ表示



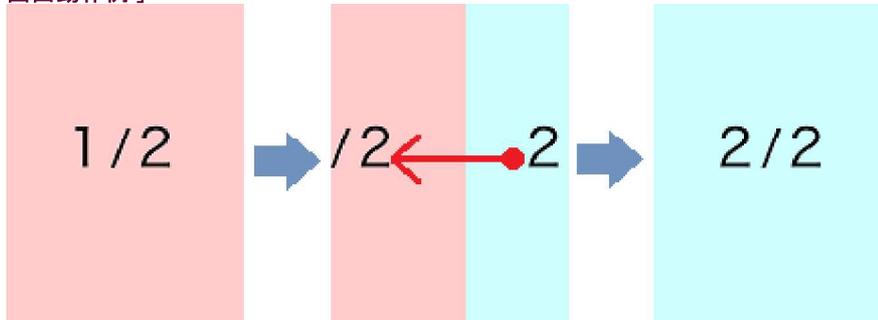
※Biz/Browser AI Ver2.1.0以降では、タブは表示されません。

管理番号: TbFAi\_001

- B** 2 スワイプで切り替える画面を作成したい  
[内容]  
iOSのホーム画面のような、スワイプでスライドさせて切り替える画面を作成したい

- A** 2 Biz/Browser AI Ver2.1.0以降では、TabFrameオブジェクトのUsePageTurningプロパティを設定すると、スワイプで切り替える画面を簡単に作成することができます。

## [画面動作例]



## [サンプルコード]

```
TabFrame TabFrame1 {
    ...

    /* 横のスワイプで画面を切り替える */
    UsePageTurning = TabFrame.TurningHorz;

    TabForm TabForm1[2];

    TabForm1[0].Get("frm_menu.crs");
    TabForm1[1].Get("frm_menu.crs");
}

TabFrame1.TabForm1[0].frm_menu{
    BgColor = $FFCCCC;
    Label1.Value = "1 / 2";
}

TabFrame1.TabForm1[1].frm_menu{
    BgColor = $CCFFFF;
    Label1.Value = "2 / 2";
}

<frm_menu.crs>
Form frm_menu {
    ...

    Label Label1 {
        X = 64;
        Y = 165;
        Width = 163;
        Height = 93;
        ...
    }
}
```

CRSファイル形式のサンプル  
「スワイプで切り替える画面を作成したい(TbFAi\_002.zip)」  
で実際の動作をご確認いただけます。

## Q 1 Biz/Browser AIの例外発生箇所の調査方法(Android)

### [内容]

Biz/Browser AIで発生した例外をBiz/Designer Mobileのデバッグで調査できない場合どうすればよいか

## A 1 [解決方法]

Biz/Browser AIでは、例外エラーが発生すると自動でエラーの詳細を出力した

「BizBrowser.core」を端末内に出力します。

Biz/Designer Mobileでのデバッグや独自のアプリケーションログから例外箇所や要因が不明な場合は、例外発生時に以下のディレクトリより「BizBrowser.core」を取得し、サポート係へお問合せください。

※「BizBrowser.core」はユーザには解析できない形式で出力されています。

[BizBrowser.coreの出力場所(Androidの場合)]

/sdcard/Android/data/jp.co.axissoft.biz/files/

「BizBrowser.core」は、例外が出力される度に上書き保存され、過去の履歴は残りません。

「BizBrowser.core」を送付する場合は必ず、例外発生直後に取得したファイルを送付してください。

また、CRSスクリプト内でtry … catch文やグローバル例外ハンドラを記述している場合は、「BizBrowser.core」は、出力されませんのでご注意ください。

### [ 補足 ]

Biz/Browser AI ver1.0.2.0から、Android上でショートカットを作成する際に、「BizBrowser.core」の出力モードを設定できるようになりました。

値	説明
0	デバッグ情報(コアダンプ)を出力しません。
1	ハンドリングされていない例外が発生した場合、デバッグ情報(コアダンプ)を出力します。コアダンプがすでに存在する場合、上書きされます。(従来のBiz/Browserの動作と同等です)
2	ハンドリングの有無にかかわらず、例外が発生するたびにデバッグ情報(コアダンプ)を出力します。コアダンプファイル名にはプロセスIDと4桁の連番が付与され、コアダンプが出力される度に新たに作成されます。

### [ 参考 ]

Biz/Browser CRSマニュアル > Biz/Browserの概要

> Android版の特徴と制約

ショートカット作成画面について

管理番号 : CrsAi\_001

- Q** 2 例外「CRS-105」や「CRS-900」のメッセージについて  
[内容]  
例外「CRS-105」や「CRS-900」が発生したが、こういったエラーが分からない

- A** 2 **[解決方法]**  
Biz/Browser AIでは、何らかの理由でアプリケーションエラーが発生した場合、例外「CRS-105」(ver1.0.1.0以前では「CRS-900」)の例外が発生します。

この例外が発生した場合、内部的な齟齬が発生している可能性が高く、アプリケーションの実行を続けたとしても正常な動作は期待できません。例外「CRS-105」(「CRS-900」)をハンドリングしてエラーメッセージの表示などを行い、速やかにアプリケーションの終了処理を行うことを推奨いたします。

< エラーメッセージ例 : ver1.0.2.0以降 >

**×** Biz/Browser AI

CRS 105

実行エラー (java.lang.StringIndexOutOfBoundsException: length=27; index=27) がメソッドLoginで発生しました :

- jp.co.axisoft.biz.util.Encoder.decodeURL(Encoder.java:103)
- jp.co.axisoft.biz.util.Encoder.decodeURL(Encoder.java:84)
- jp.co.axisoft.biz.net.URL.setURL(URL.java:90)
- jp.co.axisoft.biz.net.URL.<init>(URL.java:20)
- jp.co.axisoft.biz.packages.standard.CRSRootWnd.fncLogin(CRSRootWnd.java:618)
- jp.co.axisoft.biz.packages.standard.CRSRootWnd.callMethod(CRSRootWnd.java:196)
- jp.co.axisoft.biz.crs.CRSvm.callFunction(CRSvm.java:2723)
- jp.co.axisoft.biz.crs.CRSvm.op\_cal\_fnc(CRSvm.java:3947)
- jp.co.axisoft.biz.crs.CRSvm.interpret(CRSvm.java:1310)
- jp.co.axisoft.biz.crs.CRSvm.execSingleContext(CRSvm.java:925)
- jp.co.axisoft.biz.crs.CRSvm.callFunction(CRSvm.java:2672)
- jp.co.axisoft.biz.crs.CRSvm.op\_cal\_fnc(CRSvm.java:3947)
- jp.co.axisoft.biz.crs.CRSvm.interpret(CRSvm.java:1310)
- jp.co.axisoft.biz.crs.CRSvm.execSingleContext(CRSvm.java:925)

< エラーメッセージ例 : ver1.0.1.0以前 >

**×** Biz/Browser AI

CRS 900

メソッド'\_get\_nc'の呼び出しでアプリケーションエラーが発生

しました : \* java.net.URI.create(URI.java:727)  
\* org.apache.http.client.methods.HttpGet.<init>(HttpGet.java:75)  
\* jp.co.axisoft.biz.net.HttpRequest.sendRequest(HttpRequest.java:100)  
\* jp.co.axisoft.biz.net.HttpSession.send(HttpSession.java:164)  
\* jp.co.axisoft.biz.browser.CRequestTarget.httpSendRequest(CRequestTarget.java:227)  
\* jp.co.axisoft.biz.browser.CRequestTarget.send(CRequestTarget.java:40)  
\* jp.co.axisoft.biz.crs.CRSResourceLoader.load(CRSResourceLoader.java:118)  
\* jp.co.axisoft.biz.crs.CRSContext.fncGetNC(CRSContext.java:659)  
\* jp.co.axisoft.biz.packages.system.CRSObject.callMethod(CRSObject.java:212)  
\* jp.co.axisoft.biz.packages.standard.CRSDisplayObjectMethod.

上記のメッセージは、Android OS上での表示例です。

---

管理番号: CrsAi\_002

[◀◀ 前の TIPS へ](#)

[■ カテゴリメニューへ](#)

[Biz-Collections Bizの宝箱 トップへ](#)

[Biz/Browser AI TIPS集 トップへ](#)

**Q 3** [接続ライセンス証明書を配布したい](#) **NEW****[内容]**

接続ライセンス証明書をどのように各ユーザへ配布すればよいか

**A 3** **[解決方法]**

接続ライセンス証明書(XMLファイル)の配布は、接続ライセンス証明書をサーバに設置し、各端末からサーバへアクセスして、接続ライセンス証明書をインポートする方法が一般的です。

上記、接続ライセンス証明書のインポート処理は、グローバル関数「ImportConnectionLicense」を利用して実装することが可能です。  
この関数では、対象ファイルが正常な接続ライセンス証明書であるか、有効期限が切れていないかもチェックされます。

**[サンプルコード]**

```
ImportConnectionLicense("http://server/license/license.xml");
```

実際の動作は、以下のサンプルプログラムをご確認頂けます。

「接続ライセンス証明書のインポート(CrsAi\_003.zip)」

アプリケーションで一番初めに実行されるCRSへ、接続ライセンス証明書のインポート処理を組み込むと、必要な接続ライセンス証明書を自動的に各ユーザの端末へインポートすることも可能です。詳細は、以下のページをご参照ください。

**[参考]**

TIPS集 > Biz/Browser & Biz/Designer > [接続ライセンス証明書の配布を自動化したい](#)

管理番号 : CrsAi\_003

**B** 4 Biz/Browser AIの例外発生箇所の調査方法(iOS)**[内容]**

Biz/Browser AIで発生した例外をBiz/Designer Mobileのデバッグで調査できない場合  
どうすればよいか

**R** 4 **[解決方法]**

Biz/Browser AIでは、例外エラーが発生すると自動でエラーの詳細を出力した  
「BizBrowser.core」を端末内に出力します。  
Biz/Designer Mobileでのデバッグや独自のアプリケーションログから例外箇所や要因が  
不明な場合は、例外発生時に「BizBrowser.core」を取得し、サポート係へお問合せください。  
※「BizBrowser.core」はユーザには解析できない形式で出力されています。

iOSでは、OSのセキュリティ制約上、出力された「BizBrowser.core」へ、GetCoreDump関数  
以外ではアクセスできないようになっています。  
GetCoreDump関数でfileオブジェクトとして取得し、下記のような方法で端末から取り出します。

(例1) 取得したファイルを他のアプリで開く(ファイル共有アプリを選択し、PCと共有する等)

```
/* 一番新しいコアダンプを取得 */
var cdfile = getCoreDump();

/* コアダンプをRootディレクトリに保存
※iOS版のShellOpenメソッドはRootディレクトリのファイルのみ開ける */
var fs = new FileSystem();
var f = fs.Open("/sample.core", FileSystem.OPEN_WRITE);
f.Write(cdfile.readBinary(-1));
f.Close();

/* ShellOpenで開く */
var rt = new Runtime();
rt.ShellOpen("/sample.core");
```

(例2) サーバにアップロードする

```
/* 一番新しいコアダンプを取得 */
var cdfile = getCoreDump();

/* サーバーにアップロード
※サーバでPUTメソッドが許可されている必要がある */
var session = getHttpSession();
var res = session.Put("/upload/coredump/sample.core", cdfile);
if(res == 200 | res == 201) {
    MessageBox("アップロードに成功しました");
}
```

「BizBrowser.core」は、例外が出力される度に上書き保存され、過去の履歴は残りません。  
「BizBrowser.core」を送付する場合は必ず、例外発生直後に取得したファイルを送付して  
ください。  
また、CRSスクリプト内でtry ... catch文やグローバル例外ハンドラを記述している場合は、  
「BizBrowser.core」は、出力されませんのでご注意ください。

CRSファイル形式のサンプル

「コアダンプを取得する(CrsAi\_004.zip)」  
で実際の動作をご確認いただけます。

管理番号: CrsAi\_004

- Q** 1 OSごとのSYSオブジェクトのOSプロパティ、OS\_VERSIONプロパティで取得できる値について [↑UP](#)

- A** 1 **[解決方法]**  
SYSオブジェクトのOSプロパティでBiz/Browser AIが動作している端末OS名、OS\_VERSIONプロパティでOSのバージョン(Androidの場合APIレベル)を取得できます。

以下に代表的なOSでの値を例示します。

	OS	OS_VERSION
Android 3.2	android	13
Android 4.0 - 4.0.2		14
Android 4.0.3 - 4.0.4		15
iOS 7	iPhone OS	7.xxxx

なお、Biz/Browser AI for Windows上で動作している場合は、OS名はWindows名、バージョンはWindowsのバージョンが取得されます。  
具体的な値はPC版Biz/Browserと同様ですので、以下をご参照下さい。

(参考)

Biz/Browser > クライアント情報

・OSごとのSYSオブジェクトのOSプロパティ、OS\_VERSIONプロパティで取得できる値について

これを利用して端末上で動作している時と、Biz/Browser AI for Windowsで動作している時の処理を分岐する事もできます。

#### **[記述例]**

< Android端末以外で動作している場合に処理を分岐する >

```
if(//.SYS.OS == "android"){  
    /* Android端末上でしか動作しない処理 */  
    ...  
}else{  
    MessageBox("この処理はAndroid端末以外では動作しません");  
}
```

管理番号: CliAi\_001

---

Biz-Collections -Bizの宝箱-Tips集

2023年 10月 版

発行：株式会社オープンストリーム

〒163-0709 東京都新宿区西新宿2丁目7番1号

新宿第一生命ビルディング9階

問い合わせ先：サポート係 [biz-qa@opst.co.jp](mailto:biz-qa@opst.co.jp)

---