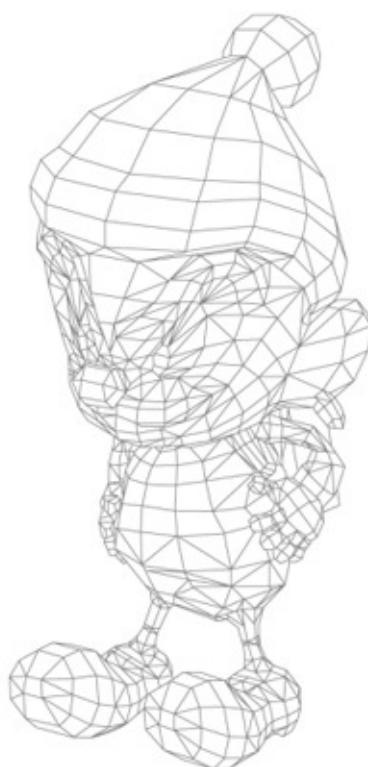

TRA File Ver.4.0 Data Format Specification

MascotCapsuleV3

日本語版



Ver.2.0

—更新履歴—

2003 年 10 月 8 日

- ・ Ver. 1.0 新規作成

2005 年 6 月 15 日

- ・ Ver. 1.0

更新内容

- 文章の一部およびレイアウトが修正

2007 年 3 月 20 日

- ・ Ver. 2.0

更新内容

- 文章およびレイアウトが修正
- サンプルコードを追加

目 次

1. はじめに.....	1
1.1. TRA4 とは.....	1
2. ファイル構造.....	2
2.1. マクロ定義	2
2.2. 基本コード規約.....	2
3. 各チャンクについて	5
3.1. (Head)チャンク	5
3.2. (Figure)チャンク	5
3.2.1. (name)チャンク	5
3.2.2. (totalFrame)チャンク	6
3.2.3. (bone)チャンク	6
3.2.3.1 (name [STRING])チャンク	7
3.2.3.2 (kf [INT] [FLOAT])チャンク	7
3.2.3.3 (translate)チャンク.....	8
3.2.3.4 (scale)チャンク.....	9
3.2.3.5 (rotate)チャンク.....	9
3.2.3.6 (roll)チャンク.....	10
3.2.4. (DynamicPolygons)チャンク	11
3.2.4.1 (kgf [INT] [INT] [true false])チャンク	11
4. サンプルコード	13
4.1. TRA4 サンプル 1	13
4.2. TRA4 サンプル 2	14
4.3. TRA4 サンプル 3	16

1. はじめに

本稿は **MascotCapsuleV2/V3**（以下、V2/V3 と記す）の中間ファイルである **TRA ファイル**（以下、TRA と記す）について解説しています。

TRA ファイルとは、V2/V3 で利用するモデル情報を持つデータに適用されたアニメーション情報を記述していきます。バージョン 3.0/4.0（以下、TRA3/TRA4 と記す）があります。TRA3 とは V2 で利用するモデルのアニメーション情報を持つデータファイルです。それに対して TRA4 は V3 で利用するモデルのアニメーション情報を持つデータファイルになります。本稿ではこの TRA4 のデータフォーマットについて詳細を説明します。

1.1. TRA4 とは

TRA4 とはテキストフォーマット形式（TRA3 はバイナリフォーマット形式のファイルです）のファイルで、V3 で利用するモデルデータに適用されているアニメーション情報などを文字列で記述していきます。モデルデータ情報に関しては BAC6 ファイルに定義します。別紙「BAC File Ver.6.0 Data Format Specification」を参照して下さい。TRA4 と BAC6 は V3 で利用する対となる中間ファイルです。TRA4 のファイル拡張子は「.tra」です。テキスト形式で扱う文字規約は「ASCII コード」の範囲でサポートしています。

2. ファイル構造

ここではTRA4の基本となる構造と規約について説明します。TRA4の基本構造を理解する上で「2.1 マクロ定義」と「2.2 基本コード規約」を参照して下さい。

2.1. マクロ定義

ここではTRA4に記述されるパラメータのマクロ定義について説明します。「2.2 基本コード規約」を参照する上で、ベーシックタイプとしてパラメータ部を以下のマクロで定義しています。

1) [STRING]

ヌル終端処理された文字列を意味しています。

文字列はダブルクォーテーション(“”)で囲い、255文字(byte)以内に収めてください。

2) [A|B]

AあるいはBを意味しています。

A又はBはヌル終端文字列で255文字(byte)以内に収めてください。

3) [INT]

整数値を表す文字列を意味します。

文字列(数値)の範囲は int 型のビット範囲に収めて下さい。

4) [FLOAT]

浮動小数値を表す文字列を意味します。

文字列(数値)の範囲は float 型のビット範囲に収めて下さい。

5) ...

任意の繰り返しを意味します。

6) ;(セミコロン)

セミコロンから行末まではコメント文字列を意味します。

2.2. 基本コード規約

TRA4の基本コード規約として、ファイルの先頭に「;」(セミコロン)から始まる文字列を記述してはいけません。これは、「【図】基本フォーマット」の先頭に位置する「; TRA」に相当します。これより下に位置(改行)する「;」から始まる文字列は、バイナリファイルへ変換する際にコメント文字列として破棄されます。また、TRAは大きく分けて2つのチャンクから構成されています。それは(Head)チャンクと(Figure)チャンクです。これら2つのチャンクの記述順序は変更してはいけません。(Head)チャンクはファイルのヘッダーが記述され、(Figure)チャンクにはモデルデータに適用されているアニメーション情報が記述されるからです。必ずヘッダーの役割をする(Head)チャンクの後に(Figure)チャンクの記述をして下さい。

【図】基本フォーマット

```

;TRA

( Head
    ( traVersion [FLOAT] )
)
( Figure
    ( name [STRING] )

    ( totalFrame [INT] )

    ( bone
        ( name [STRING] )
        ( translate.x
            ( kf [INT] [FLOAT] )
            ( kf [INT] [FLOAT] )
            ...
        )
        ( translate.y
            ( kf [INT] [FLOAT] )
            ( kf [INT] [FLOAT] )
            ...
        )
        ( translate.z
            ( kf [INT] [FLOAT] )
            ( kf [INT] [FLOAT] )
            ...
        )
        ...
        ( scale.x
            ( kf [INT] [FLOAT] )
            ( kf [INT] [FLOAT] )
            ...
        )
        ( scale.y
            ( kf [INT] [FLOAT] )
            ( kf [INT] [FLOAT] )
            ...
        )
        ( scale.z
            ( kf [INT] [FLOAT] )
            ( kf [INT] [FLOAT] )
            ...
        )
    )
)

```

```

        ( rotate.x
          ( kf    [INT] [FLOAT] )
          ( kf    [INT] [FLOAT] )
          ...
        )
        ( rotate.y
          ( kf    [INT] [FLOAT] )
          ( kf    [INT] [FLOAT] )
          ...
        )
        ( rotate.z
          ( kf    [INT] [FLOAT] )
          ( kf    [INT] [FLOAT] )
          ...
        )
        ( roll
          ( kf    [INT] [FLOAT] )
          ( kf    [INT] [FLOAT] )
          ...
        )
    )

    ( DynamicPolygons
      ( kgf    [INT] [INT] [A | B] )
      ( kgf    [INT] [INT] [A | B] )
      ...
    )
)

```


3. 各チャンクについて

ここでは、TRA4 に記述する各チャンクの詳細を説明します。

3.1. (Head)チャンク

(traVersion [FLOAT]) チャンクには TRA のバージョンを指定し、TRA4 のファイルヘッダーを定義します。TRA4 はバージョン 4.0 なので[FLOAT]には **4.0** と文字列で定義します。

```
( Head
    ( traVersion 4.0 )
)
```

3.2. (Figure)チャンク

モデルデータ(ポリゴンメッシュ)に適用されているボーン・アニメーション情報を記述します。(Figure)チャンク内の各チャンクにアニメーションパラメータの詳細が記述されます。それらの各チャンクに関する詳細は 3.2.1～3.2.4 を参照して下さい。

```
( Figure
    ( name ... )
    ( totalFrame ... )
    ( bone ... )
    ( DynamicPolygons ... )
)
```

(name)チャンクにはアニメーション対象のボーンの名前を定義します。

(totalFrame)チャンクにはアニメーションのフレーム数を定義します。

(bone)チャンクには対象となるボーンアニメーション情報を定義します。

(DynamicPolygons)チャンクには対象となるポリゴンに設定されている動的ポリゴンのアニメーション情報を任意の数だけ定義します。

3.2.1. (name)チャンク

(Figure)チャンクの中の(name [STRING])チャンクにはアニメーション対象のボーン名を定義します。文字列は 255 文字(byte)以内で収めてください。ボーンの名前を必ずしも定義する必要はありませんが、定義する場合は(name "")のようにダブルクォーテーションで囲む必要があります。

```
( name "bone" )
```

3.2.2. (totalFrame)チャンク

(totalFrame)チャンクには、対象となるモデルデータに適用されているアニメーションのキーフレームの長さを定義します。アニメーション対象となるボーンや、ダイナミックポリゴンで定義するキーフレーム値は、ここで定義するフレーム数未満である必要があります。

キーフレームの有効範囲は符号無し整数値 1～32767 です。

```
( totalFrame 100 )
```

3.2.3. (bone)チャンク

(bone)チャンクには、BAC6 ファイル(別紙「BAC6 File Ver.6.0 Data Format Specification」を参照)で登録されているボーンのアニメーション・キーフレーム情報を定義します。

登録順序は BAC6 ファイルに記述した(bone)チャンクの順序と一致させる必要があります、登録順序が BAC6 ファイルの(bone)チャンクと異なる場合、正しいアニメーションを行いません。

ボーンのアニメーションには移動、拡大縮小、回転を定義することが出来ます。

アニメーションの対象となるボーンのアニメーション値は、間に挟まれたキーフレーム値によって線形補間されます。

```
( bone
  ( name "bone" )
  ( translate.x
    ( kf 0 0.000 )
    ...
  )
  ( translate.y
    ( kf 0 0.000 )
    ...
  )
  ( translate.z
    ( kf 0 0.000 )
    ...
  )
  ( scale.x
    ( kf 0 0.000 )
    ...
  )
  ( scale.y
    ( kf 0 0.000 )
    ...
  )
  ( scale.z
    ( kf 0 0.000 )
    ...
  )
)
```

```

...
( rotate.x
    ( kf 0 0.000 )
    ...
)
( rotate.y
    ( kf 0 0.000 )
    ...
)
( rotate.z
    ( kf 0 0.000 )
    ...
)
( roll
    ( kf 0 0.000 )
    ...
)
)
...

```

3.2.3.1 (name [STRING])チャンク

(bone)チャンクに属する(name)チャンクには、名前を定義することができます。

名前を必ずしも記述する必要はありませんが、記述する場合はダブルクォーテーション(name "")で囲まなければいけません。

```

( bone
    ( name "bone" )
    ...
)

```

3.2.3.2 (kf [INT] [FLOAT])チャンク

このチャンクには対象となるボーンのキーフレーム値と、そのフレーム値に対応するアニメーション値を定義します。(kf)チャンクは(bone)チャンクに属する(translate)、(rotate)、(scale)、(roll)チャンクのそれぞれに属して定義します。

1 番目のパラメータには任意のキーフレーム値を定義し、2 番目のパラメータには 1 番目のパラメータで定義するキーフレームに対応する各種アニメーション値(平行移動、回転、拡大縮小)を定義します。

1 番目のパラメータに指定する開始キーフレーム値は、必ず 0 フレームとして定義しなければいけません。その為、仮にアニメーションが 5 フレームから開始していた場合でも、最初に定義するフレーム値を 0 に指定する必要があります。

これに伴い次のキーフレーム値の定義には下記の式で表すことができます。

$$\text{【次のフレーム値】} = \text{【次のフレーム値】} - \text{【前フレーム値】}$$

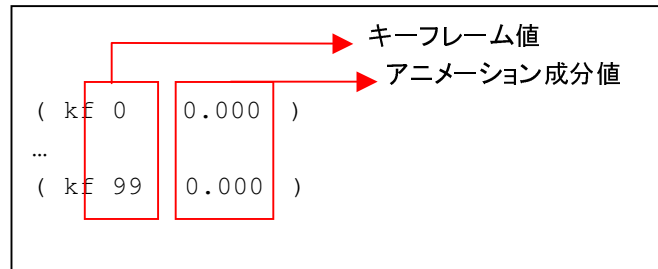
このことから、終了フレーム値の定義は…

【終了フレーム値】 = 【全フレーム数】 - 1

という式で表すことができます。

また、キーフレームに設定する値は(totalFrames)チャンクで定義するキーフレーム数未満でなければいけません。

下記の例はキーフレーム範囲が 1 ～ 100 フレームの指定の場合です。



3.2.3.3 (translate)チャンク

このチャンクには対象となるボーンの三次元平行移動ベクトル成分を定義します。

(translate)に定義するベクトル値は、BAC6 で定義されたボーン座標系において相対的な平行移動距離を表します。

デフォルトの標準姿勢を保つ場合は XYZ(0, 0, 0)で定義します。

(translate.x)	対象ボーンの X 軸方向の移動成分値を定義します
(translate.y)	対象ボーンの Y 軸方向の移動成分値を定義します
(translate.z)	対象ボーンの Z 軸方向の移動成分値を定義します

下記の例では平行移動アニメーションのデフォルトの値を定義しています。

(kf)チャンクの 1 番目のパラメータには任意のキーフレーム値を定義し、2 番目のパラメータには 1 番目のパラメータに対する各方向の移動成分値をそれぞれ定義します。

```
( translate.x
  ( kf 0 0.000 )
  ...
  ( kf 99 0.000 )
)
( translate.y
  ( kf 0 0.000 )
  ...
  ( kf 99 0.000 )
)
( translate.z
  ( kf 0 0.000 )
  ...
  ( kf 99 0.000 )
)
```

3.2.3.4 (scale)チャンク

このチャンクには対象となるボーンの三次元拡大縮小ベクトル成分を定義します。

(scale)で定義されるベクトル値は、BAC6 で定義されたボーン座標系を基準に拡大縮小値を表します。
単位は倍率を表し、等倍を XYZ(100 100 100)として定義します。

デフォルトの標準姿勢を保つ場合は XYZ(100 100 100)となります。

(scale.x)	対象ボーンの X 軸方向への拡大縮小成分値を定義します
(scale.y)	対象ボーンの Y 軸方向への拡大縮小成分値を定義します
(scale.z)	対象ボーンの Z 軸方向への拡大縮小成分値を定義します

下記の例では拡大縮小アニメーションのデフォルトの値を定義しています。

(kf)チャンクの 1 番目のパラメータには任意のキーフレーム値を定義し、2 番目のパラメータには 1 番目のパラメータに対する各方向の拡大縮小成分値をそれぞれ定義します。

```
( scale.x
  ( kf 0 100.000 )
  ...
  ( kf 99 100.000 )
)
( scale.y
  ( kf 0 100.000 )
  ...
  ( kf 99 100.000 )
)
( scale.z
  ( kf 0 100.000 )
  ...
  ( kf 99 100.000 )
)
```

3.2.3.5 (rotate)チャンク

このチャンクには対象となるボーンの三次元回転ベクトル成分(方位)を定義します。

(rotate)で設定するベクトル成分は、BAC6 で定義されたボーン座標系を基準に、対象となるボーンの先端方向(+Z)を単位ベクトルとして表しています。

デフォルトの標準姿勢を保つには XYZ(0, 0, 1)となります。

(rotate)のベクトル値の結果 XYZ(0, 0, -1)はエラー特異点(姿勢)になり、特定のキーフレーム値(この値もしくは隣り合うキーフレーム値)によってこの点を通過する場合に正常な姿勢を保てません。

(rotate.x)	対象ボーンの X 軸方向に対する回転ベクトルを定義します
(rotate.y)	対象ボーンの Y 軸方向に対する回転ベクトルを定義します
(rotate.z)	対象ボーンの Z 軸方向に対する回転ベクトルを定義します

下記の例では(rotate)における回転アニメーションのデフォルトの値を定義しています。

(kf)チャンクの 1 番目のパラメータには任意のキーフレーム値を定義し、2 番目のパラメータには 1 番目のパラメータに対する回転度数(スカラー)値を定義します。

```
( rotate.x
  ( kf 0 0.000 )
  ...
  ( kf 99 0.000 )
)
( rotate.y
  ( kf 0 0.000 )
  ...
  ( kf 99 0.000 )
)
( rotate.z
  ( kf 0 1.000 )
  ...
  ( kf 99 1.000 )
)
```

3.2.3.6 (roll)チャンク

このチャンクには BAC6 ファイルで登録した対象となるボーン座標系を基準に、キーフレーム毎にオイラー変換を行った後の Z 軸の度数(スカラー)を定義します。

(roll)で設定するスカラーは BAC6 ファイルで定義したボーン座標系を基準に、対象ボーンの先端方向(Z 軸)を軸にして表すスカラーを定義します。

デフォルトの標準姿勢を保つには 0 となります。

(roll)	対象ボーンを中心に対する回転角度(スカラー[度数])
----------	------------------------------

下記の例では(roll)におけるデフォルトの回転アニメーションを定義しています。

```
( roll
  ( kf 0 0.000 )
  ( kf 50 0.000 )
  ...
  ( kf 99 0.000 )
)
```

3.2.4. (DynamicPolygons)チャンク

(DynamicPolygons)チャンクでは BAC6 で定義されたダイナミックポリゴン情報を定義します。BAC6 はダイナミックポリゴンの任意のパターンをこのチャンクで定義します。これに対して TRA4 のこのチャンクでは、BAC6 で定義したダイナミックポリゴンのアニメーション情報を定義します。BAC6 にダイナミックポリゴンが定義されていない場合は、このチャンクを TRA4 に定義する必要はありません。

3.2.4.1 (kgf [INT][INT][true | false])チャンク

(kgf)チャンクにはダイナミックポリゴンのパターン・アニメーションの表示/非表示の切り替えを任意のキーフレーム単位に定義します。

1 番目のパラメータには任意のキーフレームを定義します。これはダイナミックポリゴンのパターン表示/非表示を切り替える為のキーフレーム値を指定するのと同じです。

TRA4 に定義するパターンの開始フレーム値を必ず 0 フレームに定義して下さい。

ただし、明記するキーフレームの順番は必ずキーフレームの若い番号からというわけではなく、順不同でも問題はありません。下記の図、A)、B)を参照して下さい。これらは不正フォーマットではありませんが、キーフレーム値が若い順に定義している図 A)を推奨します。

また、ここで定義するキーフレーム値は(totalFrame)チャンクで定義されているフレーム値未満でなければいけません。

A) 【推奨順序】

```
( DynamicPolygons
  ( kgf 0 0 true )
  ( kgf 10 1 true )
  ( kgf 10 0 false )
  ( kgf 15 1 false )
)
```

B) 【非推奨順序】

```
( DynamicPolygons
  ( kgf 10 1 true )
  ( kgf 10 0 false )
  ( kgf 0 0 true )
  ( kgf 15 1 false )
)
```

2 番目のパラメータには任意のダイナミックポリゴンのパターン ID を定義します。

このパターン ID とは BAC6 で定義する (DynamicPolygons)チャンクに属する(group)チャンクのインデックス番号です。

下記の例では、開始フレーム目にインデックス 0 番のパターンを定義しています。続いて 10 フレーム目には、インデックス 1 番のパターンを定義しています。

```
( DynamicPolygons
  ( kgf 0 0 true )
  ( kgf 10 1 true )
  ( kgf 10 0 false )
  ( kgf 15 1 false )
)
```

3 番目のパラメータには任意のフラグメント処理[true | false]を定義します。

パラメータに true を定義すると 2 番目のパラメータで定義したグループポリゴンが表示されます。false を指定するとグループポリゴンは非表示になります。

```
( DynamicPolygons
  ( kgf  0  0  true )
  ( kgf 10  1  true )
  ( kgf 10  0  false )
  ( kgf 15  1  false )
)
```


4. サンプルコード

4.1. TRA4 サンプル 1

ここではZ軸方向に回転するアニメーションを出力したコードを記述しています。
(キーフレーム範囲は開始1フレームから終了11フレームです。)

【Sample01.tra】

```
;TRA

( Head
  ( traVersion 4.0 )
)
( Figure
  ( totalFrame 11 )

  ( bone
    ( name "sample01" )
    ( translate.x
      ( kf 0 0.000000 )
    )
    ( translate.y
      ( kf 0 0.000000 )
    )
    ( translate.z
      ( kf 0 0.000000 )
    )
    ( scale.x
      ( kf 0 100.000000 )
    )
    ( scale.y
      ( kf 0 100.000000 )
    )
    ( scale.z
      ( kf 0 100.000000 )
    )
    ( rotate.x
      ( kf 0 0.000000 )
    )
    ( rotate.y
      ( kf 0 0.000000 )
    )
    ( rotate.z
      ( kf 0 1.000000 )
    )
    ...
```

```

        ( roll
          ( kf 0 0.000000 )
          ( kf 1 6.138396 )
          ( kf 2 22.589470 )
          ( kf 3 46.406628 )
          ( kf 6 132.590363 )
          ( kf 7 156.407547 )
          ( kf 8 172.858566 )
          ( kf 9 178.997116 )
          ( kf 10 0.000000 )
        )
      )
    )
  )

```

4.2. TRA4 サンプル 2

ここでは YZ 軸方向に 90 度回転するアニメーションを出力したコードを記述しています。
(キーフレーム範囲は開始 5 フレームから終了 15 フレームです)

【Sample02.tra】

```

;TRA

( Head
  ( traVersion 4.0 )
) ; Head

( Figure
  ( totalFrame 11 )

  ( bone
    ( name "sample02" )
    ( translate.x
      ( kf 0 0.000000 )
    )
    ( translate.y
      ( kf 0 0.000000 )
    )
    ( translate.z
      ( kf 0 0.000000 )
    )
    ...
  )
)

```

```

        ( scale.x
          ( kf 0 100.000000 )
        )
        ( scale.y
          ( kf 0 100.000000 )
        )
        ( scale.z
          ( kf 0 100.000000 )
        )
        ( rotate.x
          ( kf 0 0.000000 )
          ( kf 4 0.000000 )
          ( kf 5 0.043925 )
          ( kf 6 0.160469 )
          ( kf 7 0.313841 )
          ( kf 8 0.446917 )
          ( kf 9 0.500000 )
          ( kf 10 0.446928 )
        )
        ( rotate.y
          ( kf 0 0.000000 )
          ( kf 4 0.000000 )
          ( kf 5 0.001933 )
          ( kf 6 0.026450 )
          ( kf 7 0.110765 )
          ( kf 8 0.275800 )
          ( kf 10 0.724176 )
        )
        ( rotate.z
          ( kf 0 1.000000 )
          ( kf 5 0.999033 )
          ( kf 6 0.986686 )
          ( kf 7 0.942992 )
          ( kf 8 0.850999 )
          ( kf 9 0.707116 )
          ( kf 10 0.525189 )
        )
        ( roll
          ( kf 0 0.000000 )
          ( kf 4 0.000000 )
          ( kf 5 2.519793 )
          ( kf 6 9.359774 )
          ( kf 7 19.439661 )
          ( kf 8 31.679462 )
          ( kf 10 58.319023 )
        )
      )
    )
  )

```

4.3. TRA4 サンプル 3

ここではダイナミックポリゴンが適用されているアニメーションを出力したコードを記述しています。(キーフレーム範囲は開始 1 フレームから終了 45 フレームです)

【Sample03.tra】

```
;TRA

( Head
  ( traVersion 4.0 )
)
( Figure
  ( totalFrame 45 )

  ( bone
    ( name "sample03" )
    ( translate.x
      ( kf 0 0.000000 )
    )
    ( translate.y
      ( kf 0 0.000000 )
    )
    ( translate.z
      ( kf 0 0.000000 )
    )
    ( scale.x
      ( kf 0 100.000000 )
    )
    ( scale.y
      ( kf 0 100.000000 )
    )
    ( scale.z
      ( kf 0 100.000000 )
    )
    ( rotate.x
      ( kf 0 0.000000 )
    )
    ( rotate.y
      ( kf 0 0.000000 )
    )
    ( rotate.z
      ( kf 0 1.000000 )
    )
    ( roll
      ( kf 0 0.000000 )
    )
  )
  ...
)
```

```
( DynamicPolygons
    ( kgf 0 0 true )
    ( kgf 0 1 false )
    ( kgf 0 2 false )
    ( kgf 0 3 false )
    ( kgf 9 0 false )
    ( kgf 9 1 true )
    ( kgf 9 2 false )
    ( kgf 9 3 false )
    ( kgf 19 1 false )
    ( kgf 19 2 true )
    ( kgf 19 3 false )
    ( kgf 29 2 false )
    ( kgf 29 3 true )
    ( kgf 39 3 false )
)
)
```


TRA File Ver.4.0 Data Format Specification MascotCapsuleV3

日本語版

バージョン 2.0
発行日 2003 年 10 月 8 日
発行者 株式会社 エイチアイ
〒153-0043 東京都目黒区東山 1-4-4 目黒東山ビル 5F
TEL.03-3710-2843 FAX.03-5773-8660
<http://www.hicorp.co.jp/>

－著作権・商標・免責事項について－

・本書は著作権法上の保護を受けています。本書の一部または全部について(ソフトウェアおよびプログラムを含む)、株式会社エイチアイから書面による承諾を得ずに、いかなる方法においても無断で複写、複製、転載することは禁じられています。

・MascotCapsule(R)は、株式会社エイチアイの日本国における登録商標です。本書に記載されているその他の製品名は、各社の商標、または登録商標です。

・本書に掲載されている情報を利用することで発生するトラブルや損失・損害に対して、当社は一切責任を負いません。

・本書の内容に関しては訂正・改善のため、将来予告なしに変更することがあります。