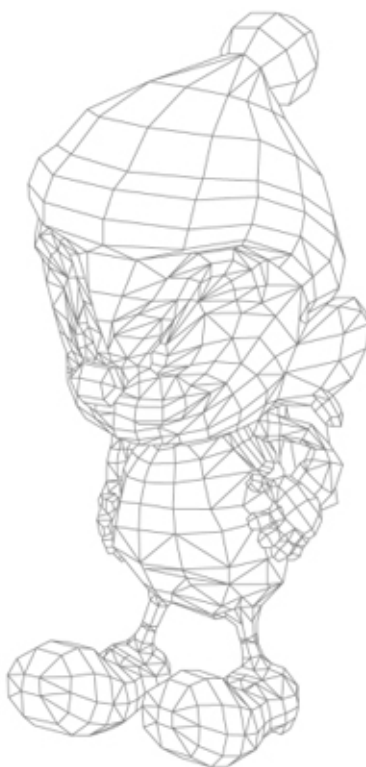

TRA File Ver.4.0

Data Format Specification

MascotCapsule V3

English version



Ver.2.1

— Revision History —

October 8, 2003

- Ver.1.0 Created

June 15, 2005

- Ver.1.0

Updated

- Corrected descriptions and page layout.

March 20, 2007

- Ver.2.0

Updated

- Corrected descriptions and page layout.
- Added sample code.

August 24, 2007

- Ver.2.1

Updated

- Corrected descriptions and page layout.
- Added descriptions of the chunks that define bone animation.

— Table of Contents —

1.. Introduction	1
1.1. What is TRA4?	1
2.. About macros	2
2.1. Macro definition	2
2.1.1. [STRING].....	2
2.1.2. [A B].....	2
2.1.3. [INT]	2
2.1.4. [FLOAT].....	2
2.1.5.	2
2.1.6. ; (Semicolon)	2
3.. File structure	3
3.1. TRA4 file structure	3
4.. About each chunk	6
4.1. (Head) chunk.....	6
4.2. (Figure) chunk.....	6
4.2.1. (name) chunk	7
4.2.2. (totalFrame) chunk	7
4.2.3. (bone) chunk	8
4.2.3.1. (name [STRING]) chunk	9
4.2.3.2. (translate) chunk.....	9
4.2.3.3. (rotate) chunk.....	10
4.2.3.4. (roll) chunk.....	10
4.2.3.5. (scale) chunk	10
4.2.3.6. (kf) chunk.....	11
4.2.3.7. Additional notes on animation definition	11
4.2.3.8. Additional notes on bone animation in TRA.....	13
4.2.4. (DynamicPolygons) chunk	14
4.2.4.1. (kgf [INT] [INT] [true false]) chunk.....	14
5.. Sample code	15
5.1. TRA4 sample 1	15
5.2. TRA4 sample 2.....	17
5.3. TRA4 sample 3.....	19

1. Introduction

This document provides explanation of the **TRA file** (we call it **TRA** from now on), which is an intermediate file of MascotCapsule V2/V3 (**V2/V3** from now on).

TRA describes the animation information applied to the data that contains model definitions to be used in **V2/V3**. There are file versions 3.0 and 4.0 (**TRA3/TRA4** from now on).

TRA3 is a data file that contains animation information for the model to be used in **V2**. On the other hand, **TRA4** is a data file that contains animation information for the model to be used in **V3**.

This document provides detailed description of the **TRA4** data format.

1.1. What is TRA4?

TRA4 is a text format file (**TRA3** is a binary format file); using text strings, **TRA4** describes animation information applied to the model data to be used in **V3**.

TRA4 specifies definitions for animations that are translation (lateral movement), rotation, and enlargement/reduction for the bones defined in the **BAC6** file; as well as pattern animation for dynamic polygons. Details of model data are defined in the **BAC6** file; please refer to the separate document, "**BAC File Ver.6.0 Data Format Specification.**"

TRA4 and **BAC6** are a pair of corresponding intermediate files to be used in **V3**. **TRA4** has ".tra" file name extension. Text format characters are supported within "**ASCII code**" range.

2. About macros

In order to simplify the explanation and description of **TRA4**, this document uses macro definitions. The following section provides explanation of each macro.

2.1. Macro definition

Parameter portion of **TRA4** is defined by the macros described in following subsections:

2.1.1. [STRING]

The **[STRING]** macro signifies a null-terminated text string.

Text string must be enclosed in double quotation marks (""), and must not exceed 255 characters (byte).

2.1.2. [A | B]

The **[A | B]** macro signifies **A** or **B**.

A or **B** is a null-terminated text string; it must not exceed 255 characters (byte).

2.1.3. [INT]

The **[INT]** macro signifies a text string that represents integer values.

Text string (numeric value) must fit within the bit range of **int** type.

2.1.4. [FLOAT]

The **[FLOAT]** macro signifies a text string that represents floating point values.

Text string (numeric value) must fit within the bit range of **float** type.

2.1.5. ...

The ... macro signifies arbitrary number of repetition.

2.1.6. ; (Semicolon)

From the semicolon to the end of line, it is a comment text string.

3. File structure

This chapter provides details of basic structure and coding convention for **TRA4**.

3.1. TRA4 file structure

TRA4 consists of ";TRA" at the top as a file identifier, and two chunks: the (**Head**) and (**Figure**) chunks. After defining the (**Head**) chunk signifying the header, you must define the (**Figure**) chunk that specifies bone animation information applied to the model data defined in **BAC6**. The order of these definitions should not be altered.

The (**Head**) chunk and (**Figure**) chunk respectively contain a series of subchunks; and each of subchunk includes dependent subchunks.

Please keep these in mind when you look at **Figure 1** "TRA4 file structure."

Figure 1 “TRA4 file structure”

```

;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] )
      ...
    )
  )

```

4. About each chunk

This chapter provides details of each chunk to be defined in **TRA4**.

4.1. (Head) chunk

Specify the version number of **TRA** in the (**traVersion** [FLOAT]) chunk, in order to define the file header.

TRA4 is version 4.0; therefore, fill in the text string **4.0** in [FLOAT] as definition.

[See also] **Figure 2** “(**Head**) chunk definition”

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

Figure 2 “(**Head**) chunk definition”

4.2. (Figure) chunk

In the (**Figure**) chunk, using the bone coordinate system of each bone defined in **BAC6**, specify the definition of bone animation per keyframe, as well as pattern animation for dynamic polygons.

Also, use the subchunks within the (**Figure**) chunk in order to define information for bone animation and dynamic polygons' animation.

[See also] **Figure 3** “(**Figure**) chunk”

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

Figure 3 “(**Figure**) chunk”

The following subsections provide details on each subchunk to be defined in the (**Figure**) chunk. Please refer to the descriptions in **4.2.1** to **4.2.4**.

4.2.1. (name) chunk

In the (**name**) chunk within the (**Figure**) chunk, define the name of target bone to be animated, using a text string not exceeding 255 characters (byte).

You do not necessarily have to define the bone name, but when you do, you must enclose it in double quotation marks as in (**name** " ").

[See also] **Figure 4** "(**name**) chunk definition"

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

Figure 4 "(**name**) chunk definition"

[Related links] [4.2.3.1. \(name \[STRING\] \) chunk](#)

4.2.2. (totalFrame) chunk

In the (**totalFrame**) chunk, define the keyframe length of the animation applied to the target model data.

The keyframe value for the bones to be animated, as well as for dynamic polygons must be less than the frame number defined in this chunk.

[See also] **Figure 5** "(**totalFrame**) chunk definition"

```
( totalFrame 100 )
```

Figure 5 "(**totalFrame**) chunk definition"

Valid range of keyframe value is 1 to 32767 unsigned integers.

4.2.3. (bone) chunk

In the **(bone)** chunk, define the animation information for the bones that are defined in **BAC6**.

The same number of **(bone)** chunks must be defined as the number of target **(bone)** chunks defined in **BAC6**. In other words, all the **(bone)** chunks must be defined regardless of being animated or not. The order of definitions must always match the order of bones defined in **BAC6**. When the definition order is different from the **(bone)** chunks in **BAC6**, animation will not be rendered correctly.

In the **(bone)** chunk, define target bone's animation of translation, enlargement/reduction, and rotation. The order of transformation should be as follows:

(scale) → (roll) → (rotate) → (translate)

Linear interpolation will be applied to the target bone's animation values, using the keyframe values inserted in this chunk.

[See also] **Figure 6 “(bone) chunk definition”**

```
( 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 )
            ...
        )
    )
    ...

```

Figure 6 “(bone) chunk definition”

Each subchunk to be defined within the (bone) chunk is as follows:

4.2.3.1. (name [STRING]) chunk

In the (name) chunk within (bone) chunk, define the name of the bone to be animated.

[See also] **Figure 7 “(name) chunk definition”**

```

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

```

Figure 7 “(name) chunk definition”

[Related links] [4.2.1. \(name \) chunk](#)

4.2.3.2. (translate) chunk

In the (translate) chunk, define the translation element value (difference) of the bone per keyframe, using the target bone coordinate system defined in **BAC6** as the origin position.

Use the (kf) chunk within the (translate) chunk to specify the animation value at the keyframe.

(translate.x)	X-element of origin position for the target bone coordinate system in BAC6
(translate.y)	Y-element of origin position for the target bone coordinate system in BAC6
(translate.z)	Z-element of origin position for the target bone coordinate system in BAC6

[Related links] [4.2.3.6 \(kf \) chunk](#)

[4.2.3.7.1. Maintaining BAC6 bone coordinate system in the \(translate \) chunk](#)

[4.2.3.8. Additional notes on bone animation in TRA](#)

4.2.3.3. (rotate) chunk

In the **(rotate)** chunk, define the direction vector of the +Z-axis using the target bone coordinate system defined in **BAC6**.

Use the **(kf)** chunk within the **(rotate)** chunk to specify the animation value at the keyframe.

(rotate.x)	X-element of +Z-axis direction vector for the target bone coordinate system in BAC6
(rotate.y)	Y-element of +Z-axis direction vector for the target bone coordinate system in BAC6
(rotate.z)	Z-element of +Z-axis direction vector for the target bone coordinate system in BAC6

The **(rotate)** chunk specifies the rotation within a plane, based on the Z-axis of the bone coordinate system defined in **BAC6**. The rotation axis within this plane goes through the origin of the target bone's coordinate system, and is perpendicular to the plane.

[Related links] [4.2.3.6 \(kf \) chunk](#)

[4.2.3.7.2. Maintaining BAC6 bone coordinate system in the \(rotate \) chunk](#)

[4.2.3.8. Additional notes on bone animation in TRA](#)

4.2.3.4. (roll) chunk

In the **(roll)** chunk, define the rotation angle (scalar value) around the Z-axis of the target bone coordinate system defined in **BAC6**.

Use the **(kf)** chunk within the **(roll)** chunk to specify the animation value at the keyframe.

(roll)	Rotation angle of Z-axis for the target bone coordinate system in BAC6
----------	---

[Related links] [4.2.3.6 \(kf \) chunk](#)

[4.2.3.7.3. Maintaining BAC6 bone coordinate system in the \(roll \) chunk](#)

[4.2.3.8. Additional notes on bone animation in TRA](#)

4.2.3.5. (scale) chunk

In the **(scale)** chunk, specify the enlargement/reduction element value for each axis, based on the origin position of the target bone coordinate system defined in **BAC6**.

Use the **(kf)** chunk within the **(scale)** chunk to specify the animation value at the keyframe.

(scale.x)	Enlargement/reduction value along the X-axis for the target bone coordinate system in BAC6
(scale.y)	Enlargement/reduction value along the Y-axis for the target bone coordinate system in BAC6
(scale.z)	Enlargement/reduction value along the Z-axis for the target bone coordinate system in BAC6

[Related links] [4.2.3.6 \(kf \) chunk](#)

[4.2.3.7.4. Maintaining BAC6 bone coordinate system in the \(scale \) chunk](#)

[4.2.3.8. Additional notes on bone animation in TRA](#)

4.2.3.6. (kf) chunk

In the (**kf**) chunk, define the keyframe value of the target bone animation, and the animation value that corresponds to that frame value.

In the first [**INT**] parameter, define the arbitrary keyframe value. The starting keyframe value must always be **0** frame. As a result, the keyframe value definition can be represented as the following expression:

$$\text{Frame value definition} = \text{Current frame value} - \text{Starting frame value}$$

Also, the keyframe value must be less than the number of keyframes to be defined in the (**totalFrame**) chunk.

In the second [**FLOAT**] parameter, define various animation values (translation, rotation, and enlargement/reduction) that correspond to the keyframe defined in the first [**INT**] parameter.

[See also] **Figure 8 “(kf) chunk”**

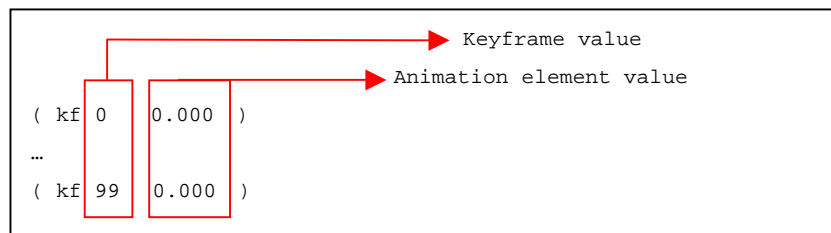


Figure 8 “(kf) chunk”

[Related links] [4.2.3.2. \(translate \) chunk](#) [4.2.3.3. \(rotate \) chunk](#)
[4.2.3.4. \(roll \) chunk](#) [4.2.3.5. \(scale \) chunk](#)

4.2.3.7. Additional notes on animation definition

The (**translate**), (**rotate**), (**roll**), and (**scale**) chunks within the (**bone**) chunk must always be defined, even when a static condition is maintained from the bone coordinate system defined in the **BAC6** file.

In order to maintain the posture of bone defined in the **BAC6** file, each chunk must be defined as follows:

4.2.3.7.1. Maintaining BAC6 bone coordinate system in the (translate) chunk

If the origin position does not laterally move (translate) from the bone coordinate system defined in **BAC6**, definition should be as follows:

```
( 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 )
)
```

[Related links] [4.2.3.2. \(translate \) chunk](#)

4.2.3.7.2. Maintaining BAC6 bone coordinate system in the (rotate) chunk

If there is no rotation by the (**rotate**) chunk from the bone coordinate system defined in **BAC6**, definition should be as follows:

```
( 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 )
)
```

[Related links] [4.2.3.3. \(rotate \) chunk](#)

4.2.3.7.3. Maintaining BAC6 bone coordinate system in the (roll) chunk

If there is no rotation around the Z-axis from the bone coordinate system defined in **BAC6**, definition should be as follows:

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

[Related links] [4.2.3.4. \(roll \) chunk](#)

4.2.3.7.4. Maintaining BAC6 bone coordinate system in the (scale) chunk

If there is no enlargement/reduction from the bone coordinate system defined in **BAC6**, definition should be as shown below. The same size should be **XYZ (100,100,100)**.

```
( 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 )
)
```

[Related links] [4.2.3.5. \(scale \) chunk](#)

4.2.3.8. Additional notes on bone animation in TRA

The following **Figure 9** "Correlation of bone coordinate system" illustrates the correlation of bone coordination systems defined in **BAC6**, as well as the correlation of bone coordination systems for each keyframe defined in **TRA4**. As shown in **Figure 9** "Correlation of bone coordinate system," define the bone position after the rotation, using the (**rotate**) chunk and the (**roll**) chunk in **TRA4**. Also, the bone animation in **TRA4** should be defined as the difference per keyframe, from the bone coordinate system defined in **BAC6**.

Figure 10 "Order of calculation" shows the calculation order for the matrices that define bone coordinate system.

Multiplication order of the (**rotate**) chunk matrix, and the (**roll**) chunk matrix is interchangeable.

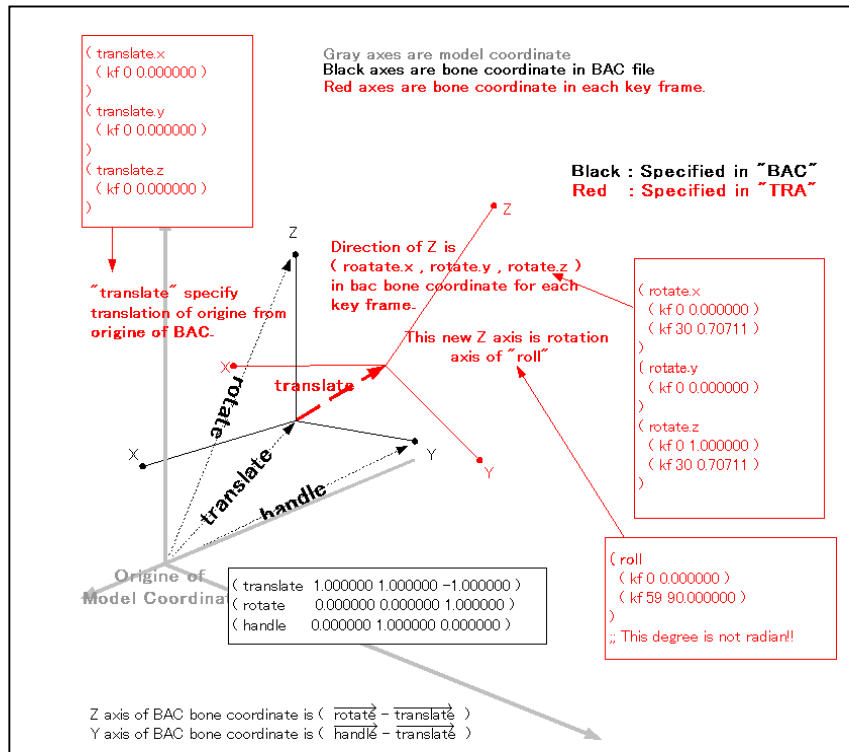


Figure 9 "Correlation of bone coordinate system"

$$\vec{T}(\text{rotate}) \cdot \vec{T}(\text{roll}) \cdot \vec{T}(\text{scale}) \cdot \vec{P}$$

Figure 10 "Order of calculation"

4.2.4. (DynamicPolygons) chunk

In the (**DynamicPolygons**) chunk, specify the animation for dynamic polygons' pattern defined in **BAC6**.

Use the (**kfg**) chunk within the (**DynamicPolygons**) chunk to define the keyframe value for the animation.

When dynamic polygons' pattern is not defined in **BAC6**, you don't have to define this chunk within the (**Figure**) chunk.

4.2.4.1. (kgf [INT] [INT] [true|false]) chunk

The (**kgf**) chunk switches between enabling/disabling the display of dynamic polygons' pattern animation per arbitrary keyframe.

In the first [INT] parameter, define keyframe value for switching between enabling/disabling the display of dynamic polygons. Animation's starting keyframe value must always be **0** frame. Valid range for the keyframe definitions must be less than the keyframe value defined in the (**totalFrame**) chunk.

In the second [INT] parameter, define the index ID that is automatically assigned to the (**group**) chunk within the (**DynamicPolygons**) chunk. It is a sequential ID starting from **0**.

Define the third [true|false] parameter as a fragment process to enable/disable the display of dynamic polygons' pattern. The definition **true** means to display, and the definition **false** means not to display.

[See also] **Figure 11 “(DynamicPolygons) chunk”**

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

Figure 11 **“(DynamicPolygons) chunk”**

[Related links] Document: “**BAC6 Data Format Specification**,” 4.2.9.1. (**group**) chunk

5. Sample code

5.1. TRA4 sample 1

The following is code output for the animation that rotates around the Z-axis.

[Sample01.tra]

```
;TRA

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

  ( bone
    ( name "sample01" )
    ( translate.x
      ( kf 0 0.000000 )
      ( kf 10 0.000000 )
    )
    ( translate.y
      ( kf 0 0.000000 )
      ( kf 10 0.000000 )
    )
    ( translate.z
      ( kf 0 0.000000 )
      ( kf 10 0.000000 )
    )
    ( scale.x
      ( kf 0 100.000000 )
      ( kf 10 100.000000 )
    )
    ( scale.y
      ( kf 0 100.000000 )
      ( kf 10 100.000000 )
    )
    ( scale.z
      ( kf 0 100.000000 )
      ( kf 10 100.000000 )
    )
    ( rotate.x
      ( kf 0 0.000000 )
      ( kf 10 0.000000 )
    )
  )
)
```

```

        ( rotate.y
          ( kf 0 0.000000 )
          ( kf 10 0.000000 )
        )
        ( rotate.z
          ( kf 0 1.000000 )
          ( kf 10 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 )
        )
      )
    )
  )

```

5.2. TRA4 sample 2

The following is code output for the animation that combines the 90 degree rotation around Y-axis, and the 90 degree rotation around Z-axis.

[Sample02.tra]

```
;TRA

( Head
  ( traVersion 4.0 )
) ; Head

( Figure
  ( totalFrame 11 )

  ( bone
    ( name "sample02" )
    ( translate.x
      ( kf 0 0.000000 )
      ( kf 10 0.000000 )
    )
    ( translate.y
      ( kf 0 0.000000 )
      ( kf 10 0.000000 )
    )
    ( translate.z
      ( kf 0 0.000000 )
      ( kf 10 0.000000 )
    )
    ( scale.x
      ( kf 0 100.000000 )
      ( kf 10 100.000000 )
    )
    ( scale.y
      ( kf 0 100.000000 )
      ( kf 10 100.000000 )
    )
    ( scale.z
      ( kf 0 100.000000 )
      ( kf 10 100.000000 )
    )
    ( rotate.x
      ( kf 0 0.000000 )
      ( kf 1 0.043925 )
      ( kf 2 0.160469 )
      ( kf 3 0.313841 )
      ...
    )
  )
)
```

```

        ( kf 4 0.446917 )
        ( kf 5 0.500000 )
        ( kf 6 0.446928 )
        ( kf 7 0.313862 )
        ( kf 8 0.160494 )
        ( kf 9 0.043951 )
        ( kf 10 0.000026 )
    )
    ( rotate.y
        ( kf 0 0.000000 )
        ( kf 1 0.001933 )
        ( kf 2 0.026450 )
        ( kf 3 0.110765 )
        ( kf 4 0.275800 )
        ( kf 6 0.724176 )
        ( kf 7 0.889218 )
        ( kf 8 0.973542 )
        ( kf 9 0.998065 )
        ( kf 10 1.000000 )
    )
    ( rotate.z
        ( kf 0 1.000000 )
        ( kf 1 0.999033 )
        ( kf 2 0.986686 )
        ( kf 3 0.942992 )
        ( kf 4 0.850999 )
        ( kf 5 0.707116 )
        ( kf 6 0.525189 )
        ( kf 7 0.332839 )
        ( kf 8 0.162661 )
        ( kf 9 0.043994 )
        ( kf 10 0.000026 )
    )
    ( roll
        ( kf 0 0.000000 )
        ( kf 1 2.519793 )
        ( kf 2 9.359774 )
        ( kf 3 19.439661 )
        ( kf 4 31.679462 )
        ( kf 6 58.319023 )
        ( kf 7 70.558815 )
        ( kf 8 80.638649 )
        ( kf 9 87.478539 )
        ( kf 10 89.998497 )
    )
    )
    )
    )

```

5.3. TRA4 sample 3

The following is code output for the animation where dynamic polygons are applied.

[Sample03.tra]

```
;TRA

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

  ( bone
    ( name "sample03" )
    ( translate.x
      ( kf 0 0.000000 )
      ( kf 44 0.000000 )
    )
    ( translate.y
      ( kf 0 0.000000 )
      ( kf 44 0.000000 )
    )
    ( translate.z
      ( kf 0 0.000000 )
      ( kf 44 0.000000 )
    )
    ( scale.x
      ( kf 0 100.000000 )
      ( kf 44 100.000000 )
    )
    ( scale.y
      ( kf 0 100.000000 )
      ( kf 44 100.000000 )
    )
    ( scale.z
      ( kf 0 100.000000 )
      ( kf 44 100.000000 )
    )
    ( rotate.x
      ( kf 0 0.000000 )
      ( kf 44 0.000000 )
    )
    ( rotate.y
      ( kf 0 0.000000 )
      ( kf 44 0.000000 )
    )
    ...
  )
)
```

```
        ( rotate.z
          ( kf 0 1.000000 )
          ( kf 44 1.000000 )
        )
        ( roll
          ( kf 0 0.000000 )
          ( kf 44 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

MascotCapsule V3

English version

Version: 2.1

Publication date: August 24, 2007

Publisher: HI CORPORATION
Meguro Higashiyama Bldg. 5F
1-4-4 Higashiyama
Meguro-ku, Tokyo 153-0043 Japan
<http://www.hicorp.co.jp/>

—Trademarks, copyrights, and disclaimer—

- This document is protected by copyright law. Neither the whole nor any part of the information (including software and program) contained in this document may be copied, reproduced, or reprinted in any material form without the written permission from HI CORPORATION.
- MascotCapsule® is the registered trademark of HI CORPORATION in Japan. Other brand names and product names in this document are trademarks or registered trademarks of their respective owners.
- HI CORPORATION is not responsible for any problem, loss, or damages that may be caused by utilizing the materials or information contained in this document.
- The contents of this document are subject to change for revision without prior notice.

© 2007 HI CORPORATION. All Rights Reserved.