

# N2NE コマンドリファレンス

ver 1.20

変更履歴

作成日	作成者	ver	変更内容
2010/11/16	西村 祥俊	1.20	BACK_TITLE コマンドを追加
2010/10/30	西村 祥俊	1.10	IF コマンドに複数条件 SET_FLAG コマンドに演算機能を追加
2010/10/24	西村 祥俊	1.00	新規作成

1.目的.....	4
2.システム.....	4
3.コマンドリファレンス.....	4
3-1.MSG コマンド.....	5
3-2.IMG_LOAD コマンド.....	6
3-3.IMG_BLEND コマンド.....	7
3-4.IMG_UPDATE コマンド.....	8
3-5.IMG_VIEW コマンド.....	9
3-6.SCR_LOAD コマンド.....	10
3-7.LABEL コマンド.....	11
3-8.LABEL_JUMP コマンド.....	12
3-9.WAIT コマンド.....	13
3-10.SET_FLAG コマンド.....	14
3-11.IF コマンド.....	15
3-12.CHOICE コマンド.....	17
3-13.SYS_MENU コマンド.....	18
3-14.SND_LOAD コマンド.....	19
3-15.SND_PLAY コマンド.....	20
3-16.SND_STOP コマンド.....	21
3-17.SWF_LOAD コマンド.....	22
3-18.SWF_PLAY コマンド.....	23
3-19.LAYER_DEL コマンド.....	24
3-20.SWF_MOVIE コマンド.....	25
3-21.FADE_IN_ALPHA コマンド.....	26
3-22.FADE_OUT_ALPHA コマンド.....	27
3-23.BACK_TITLE コマンド.....	28

## 1.目的

本ドキュメントは、「N2NovelEngine」のスク립ト作成時に使用するコマンドのリファレンスである。

## 2.システム

スク립トのソースはテキストファイルで本ドキュメントの仕様に従ったコマンドを記述し、「N2ScriptConverter」でコンパイルすることにより「N2NovelEngine」にて使用できるようになる。

## 3.コマンドリファレンス

次ページより、コマンドの仕様を記載する。

【役割】

【フォーマット】

【解説】

【使用例】

上記のような書式で記載する。

また共通のルールを以下に示す。

- ・スク립トソースファイルの最初の行には、[N2SCRIPT]という文字列を記載する。
- ・スク립トソースファイル名はコンパイル後のスク립トファイル名となる。
- ・スク립トソースファイルはUTF-8(BOMなし)で保存する。
- ・コマンド以外の文字列は無視されるため、コメントの書式は自由である。(//など)
- ・コマンド内のシンボル名、ラベル名、ID等々は[a-zA-Z0-9\_-]のみ使用可とする。

### 3-1.MSG コマンド

【役割】

ゲーム中のメッセージウインドウに、  
メッセージを表示する。

【書式】

.MSG  
表示するメッセージ

.

【解説】

.MSG というコマンド指定行の後、必ず改行を行う。  
次の行より実際に表示するメッセージを記載する。  
メッセージを記載し終わったら改行し、メッセージ終端文字として半角ピリオドを記載すること。

【使用例】

.MSG  
あいうえお。  
改行もできるよ。

.

### 3-2.IMG\_LOAD コマンド

#### 【役割】

NovelEngine 内の画像マネージャーに画像をロードする。

#### 【書式】

```
.IMG_LOAD  
{BMP シンボル名} {元画像名}
```

#### 【解説】

.IMG\_LOAD というコマンド指定行の後、必ず改行を行う。

次の行にオプションを記載する。

使用例は char1.jpg を char1 というシンボルで画像マネージャーにロードすることを意味する。

#### [オプション]

(BMP シンボル名 : String) …画像に関する他コマンドで使用する。重複してはいけない。

(元画像名 : String) …「N2ImageConverter」で結合した際の元画像名を記載する。拡張子が異なってもファイル名で重複してしまうので注意すること。

#### 【使用例】

```
.IMG_LOAD  
char1 char1
```

### 3-3.IMG\_BLEND コマンド

#### 【役割】

NovelEngine 内の画像マネージャーにロードされた画像を、プライマリスクリーンに座標指定して合成する。

#### 【書式】

.IMG\_BLEND

{BMP シンボル名} {srcX} {srcY} {dstX} {dstY} {w} {h}

#### 【解説】

.IMG\_BLEND というコマンド指定行の後、必ず改行を行う。

次の行にオプションを記載する。

使用例は char1 シンボルの画像の(0, 0)から幅 120px 高さ 480px の画像をプライマリスクリーンの(120, 0)へ合成することを意味する。

#### 【オプション】

(BMP シンボル名 : String) …IMG\_LOAD で指定したシンボル名。

(srcX : UINT) …元画像の左上座標 X。

(srcY : UINT) …元画像の左上座標 Y。

(dstX : UINT) …プライマリスクリーンの左上座標 X。

(dstY : UINT) …プライマリスクリーンの左上座標 Y。

(w : UINT) …画像の幅。

(h : UINT) …画像の高さ。

#### 【使用例】

.IMG\_BLEND

char1 0 0 240 0 120 480

### 3-4.IMG\_UPDATE コマンド

【役割】

プライマリスクリーンをセカンダリスクリーンに座標指定して転送する。

【書式】

```
.IMG_UPDATE  
{x} {y} {w} {h}
```

【解説】

.IMG\_UPDATE というコマンド指定行の後、必ず改行を行う。

次の行にオプションを記載する。

[オプション]

(x : UINT) …転送する左上座標 X。

(y : UINT) …転送する左上座標 Y。

(w : UINT) …転送する幅。

(h : UINT) …転送する高さ。

【使用例】

```
.IMG_UPDATE  
0 0 120 480
```



### 3-5.IMG\_VIEW コマンド

#### 【役割】

NovelEngine 内の画像マネージャーにロードされた画像を、メインウィンドウの子オブジェクトとして表示する。

#### 【書式】

```
.IMG_VIEW  
{BMP シンボル名} {LAYERID} {srcX} {srcY} {dstX} {dstY} {w} {h} {index}  
{alpha}
```

#### 【解説】

.IMG\_VIEW というコマンド指定行の後、必ず改行を行う。

次の行にオプションを記載する。

#### [オプション]

(BMP シンボル名 : String) …IMG\_LOAD で指定したシンボル名。

(LAYERID : String) …表示させる子オブジェクトの名前(任意)

(srcX : UINT) …元画像の左上座標 X。

(srcY : UINT) …元画像の左上座標 Y。

(dstX : UINT) …メインウィンドウの左上座標 X。

(dstY : UINT) …メインウィンドウの左上座標 Y。

(w : UINT) …画像の幅。

(h : UINT) …画像の高さ。

(index:UINT or String) …メインウィンドウ上の表示 Z-INDEX の値。TOP を指定で最前面、BOTTOM を指定で最背面、STAY を指定で LAYERID がすでに指定済みの場合、同じ Z-INDEX 上で入れ替わる。数値を入力の場合、すでに配置済みの Z-INDEX の範囲を守らないとランタイムエラーが発生する。

(alpha : UINT) …0 ~ 100 までの数値で入力します。0 で透明、100 で不透明となる。

#### 【使用例】

```
.IMG_VIEW  
char1 char1 0 0 240 0 120 480 TOP 100
```

### 3-6.SCR\_LOAD コマンド

【役割】

現在実行中のスクリプトファイルから指定したスクリプトファイルの先頭へ移動する。

【書式】

```
.SCR_LOAD  
{スクリプトファイル名}
```

【解説】

.SCR\_LOAD というコマンド指定行の後、必ず改行を行う。

次の行にオプションを記載する。

使用例は next\_script.n2s の先頭へ移動することを意味する。

[オプション]

(スクリプトファイル名 : String)…移動先のスクリプトファイル名。拡張子はいらない。

【使用例】

```
.SCR_LOAD  
next_script
```

### 3-7.LABEL コマンド

【役割】

実行中のスクリプトファイル内にラベルを設置する。ラベルは LABEL\_JUMP コマンドで移動するためのもの。IF コマンド使用時にも関連する。

【書式】

```
.LABEL  
{ラベル名}
```

【解説】

. LABEL というコマンド指定行の後、必ず改行を行う。

次の行にオプションを記載する。

[オプション]

(ラベル名 : String) … LABEL\_JUMP で使用する名。

【使用例】

```
.LABEL  
label1
```

### 3-8.LABEL\_JUMP コマンド

【役割】

LABEL コマンドで設置したラベルの位置にスクリプトの実行位置を移動する。ただし、そのスクリプト内で定義したラベルにしか移動できないので、注意すること。

【書式】

```
.LABEL_JUMP  
{ラベル名}
```

【解説】

. LABEL\_JUMP というコマンド指定行の後、必ず改行を行う。  
次の行にオプションを記載する。

[オプション]

(ラベル名 : String) … LABEL で作成したラベル名。

【使用例】

```
. LABEL_JUMP  
label1
```

### 3-9.WAIT コマンド

【役割】

メッセージウエイトを行う。

【書式】

```
.WAIT  
{待機時間}
```

【解説】

. WAIT というコマンド指定行の後、必ず改行を行う。

次の行にオプションを記載する。

使用例ではメッセージウエイト 1000ms を指定。

[オプション]

(待機時間：String)…メッセージウエイトの待機時間(ms)。

【使用例】

```
. WAIT  
1000
```

### 3-10.SET\_FLAG コマンド

【役割】

ゲーム内フラグを設定する。

【書式】

```
.SET_FLAG  
{フラグ名} {演算記号} {フラグ値}
```

【解説】

.SET\_FLAG というコマンド指定行の後、必ず改行を行う。

次の行にオプションを記載する。

使用例は flag1 という名前に値 1 を代入することを意味する。

[オプション]

(フラグ名 : String) … フラグの名前。

(演算記号 : String) … 「=」で代入、「+=」で足し算、「-=」で引き算、「\*=」で掛け算、「/=」で割り算、それ以外はコンパイルエラーを発生させる。

(フラグ値 : INT) … フラグの値。整数で設定する。

【使用例】

```
.WAIT  
flag1 = 1
```

## 3-11. IF コマンド

## 【役割】

分岐処理を行い、条件文には SET\_FLAG コマンドで設定したフラグを使用する。その後、移動処理を行うがスクリプト内の LABEL コマンドで指定したラベルにしか移動できない。

## 【書式】

```
.IF
{条件式}({条件式})
GOTO {ラベル}
(ELSEIF
{条件式}
GOTO {ラベル})
(ELSE
GOTO {ラベル})
ENDIF
```

## 【解説】

.IF というコマンド指定行の後、必ず改行を行う。  
次の行より分岐処理を記載する。  
分岐処理を記載し終えたら改行し、分岐処理終端文字として半角ピリオドを記載すること。  
コマンド内でネストする際は「.IF」ではなく「IF」となる点に注意すること。  
長くなるので使用例は次ページに掲載するがネストも合わせて条件を追うと label3 ラベルに移動後、end ラベルに移動する。

## 【オプション】

(条件式 : String) … 条件式。書式は  
{SET\_FLAG コマンドで設定したフラグ名} {記号(==,!=,>=<=><)} {フラグ値 : UINT}  
また「|」で複数条件を指定できるが AND 条件としか解釈はしない。OR 条件はネストで対応すること。また条件と条件の間の「|」にホワイトスペースを挟まないこと(使用例を見て確認)  
(ラベル : String) … LABEL コマンドで設置したラベル名。存在しないフラグを指定するとコンパイルエラーとなる。

【使用例】

```
.SET_FLAG
flag1 = 3
.IF
flag1 == 1|flag1 == 2
GOTO label1
ELSEIF
flag1 == 2
GOTO label2
ELSE
// ネスト
IF
flag1 == 3
GOTO label3
ELSE
GOTO label4
ENDIF
ENDIF
.
.LABEL
label1
.LABEL_JUMP
end
.LABEL
label2
.LABEL_JUMP
end
.LABEL
label3
.LABEL_JUMP
end
.LABEL
label4
.LABEL_JUMP
end
.LABEL end
```



### 3-12.CHOICE コマンド

#### 【役割】

選択肢を表示する。

#### 【書式】

```
.CHOICE  
{選択肢}  
GOTO {ラベル}  
{選択肢}  
GOTO {ラベル}
```

.

#### 【解説】

. CHOICE というコマンド指定行の後、必ず改行を行う。

次の行から選択肢オプションを記載する。

選択肢オプションを記載し終わったら改行し、分岐処理終端文字として半角ピリオドを記載すること。

#### [オプション]

(選択肢 : String) … 選択肢を表示するメッセージ文字列。複数行でも指定できる。

(ラベル : String) … LABEL コマンドで設置したラベル名。存在しないフラグを指定するとコンパイルエラーとなる。

#### 【使用例】

```
.CHOICE  
選択肢 1 だよ  
GOTO c1  
選択肢 2 だよ  
GOTO c2  
  
.  
.LABEL  
c1  
.LABEL_JUMP  
choend  
.LABEL  
c2  
.LABEL_JUMP  
choend  
.LABEL  
choend
```

### 3-13.SYS\_MENU コマンド

#### 【役割】

システムメニューを表示する。

通常はタイトル画面を表示したい時に使用するコマンドである。

#### 【書式】

```
.SYS_MENU  
START {スタート文字列}  
LOAD {ロード文字列}  
OPTION {コンフィグ文字列}  
EXIT {ゲーム終了文字列}
```

#### 【解説】

.SYS\_MENU というコマンド指定行の後、必ず改行を行う。

次の行からシステムメニューオプションを記載する。

システムメニューオプションを記載し終えたら改行し、分岐処理終端文字として半角ピリオドを記載すること。

システムメニューには4つの機能が搭載されている。それぞれ START、LOAD、OPTION、EXIT という文字列とその機能を表示する文字列とで成り立っており、順不同である。

#### 【オプション】

(スタート文字列：String)…START とセットで記載する。ゲームスタートボタンとなる。スタート文字列はそのボタンに記載される文字列である。

(ロード文字列：String)…LOAD とセットで記載する。セーブデータのロードボタンとなる。ロード文字列はそのボタンに記載される文字列である。

(コンフィグ文字列：String)…OPTION とセットで記載する。ゲーム設定変更ボタンとなる。コンフィグ文字列はそのボタンに記載される文字列である。

(ゲーム終了文字列：String)…EXIT とセットで記載する。ゲーム終了ボタンとなる。ゲーム終了文字列はそのボタンに記載される文字列である。

#### 【使用例】

```
.SYS_MENU  
START ゲームスタート  
LOAD ロード  
OPTION コンフィグ  
EXIT ゲーム終了
```

### 3-14.SND\_LOAD コマンド

【役割】

NovelEngine 内のサウンドマネージャーにサウンドをロードする。

【書式】

```
.SND_LOAD  
{SND シンボル名} {元サウンド名}
```

【解説】

.SND\_LOAD というコマンド指定行の後、必ず改行を行う。

次の行にオプションを記載する。

使用例は snd1.mp3 を snd1 というシンボルでサウンドマネージャーにロードすることを意味する。

[オプション]

(SND シンボル名 : String)…サウンドに関する他コマンドで使用する。重複してはいけない。

(元サウンド名 : String)…「N2SoundConverter」で結合した際の元サウンド名を記載する。拡張子が異なってもファイル名で重複してしまうので注意すること。

【使用例】

```
.SND_LOAD  
snd1 snd1
```

### 3-15.SND\_PLAY コマンド

#### 【役割】

NovelEngine 内のサウンドマネージャーからサウンドを再生する。ループ再生とカウント再生を実行できる。

#### 【書式】

```
.SND_PLAY  
{SND シンボル名} {サウンドタイプ} ({再生回数})
```

#### 【解説】

.SND\_PLAY というコマンド指定行の後、必ず改行を行う。

次の行にオプションを記載する。

使用例は se シンボルを 3 回再生する。

#### [オプション]

(SND シンボル名 : String)…SND\_LOAD コマンドでロードしたサウンドデータ。

(サウンドタイプ : String)…LOOP でループ再生、COUNT で回数制限再生。それ以外はコンパイルエラーを発生する。

(再生回数 : UINT)…サウンドタイプが COUNT の場合のみ指定する。LOOP の場合は省略可。

#### 【使用例】

```
.SND_PLAY  
se COUNT 3
```

### 3-16.SND\_STOP コマンド

【役割】

N2NovelEngine 内で現在再生中のサウンドを停止する。

【書式】

```
.SND_STOP  
{SND シンボル名}
```

【解説】

.SND\_STOP というコマンド指定行の後、必ず改行を行う。

次の行にオプションを記載する。

[オプション]

(SND シンボル名:String)…SND\_LOAD コマンドでロードしたサウンドデータ。

【使用例】

```
.SND_STOP  
snd1
```

### 3-17.SWF\_LOAD コマンド

【役割】

NovelEngine 内の SWF マネージャーに SWF をロードする。

【書式】

```
.SWF_LOAD  
{SWF シンボル名} {元 SWF 名}
```

【解説】

.SWF\_LOAD というコマンド指定行の後、必ず改行を行う。

次の行にオプションを記載する。

使用例は swf1.swf を swf1 というシンボルで SWF マネージャーにロードすることを意味する。

[オプション]

(SWF シンボル名 : String) … SWF に関する他コマンドで使用する。重複してはいけない。

(元 SWF 名 : String) … 「N2SwfConverter」で結合した際の際の元 SWF 名を記載する。

【使用例】

```
.SWF_LOAD  
swf1 swf1
```

### 3-18.SWF\_PLAY コマンド

#### 【役割】

NovelEngine 内の SWF マネージャーにロードされた SWF を、メインウィンドウの子オブジェクトとして表示する。

#### 【書式】

```
.SWF_PLAY  
{SWF シンボル名} {LAYERID} {x} {y} {index} ({alpha})
```

#### 【解説】

.SWF\_PLAY というコマンド指定行の後、必ず改行を行う。

次の行にオプションを記載する。

#### [オプション]

(SWF シンボル名 : String) …SWF\_LOAD で指定したシンボル名。

(LAYERID : String) …表示させる子オブジェクトの名前(任意)

(x : UINT) …元画像の左上座標 X。

(y : UINT) …元画像の左上座標 Y。

(index:UINT or String) …メインウィンドウ上の表示 Z-INDEX の値。TOP を指定で最前面、BOTTOM を指定で最背面、STAY を指定で LAYERID がすでに指定済みの場合、同じ Z-INDEX 上で入れ替わる。数値を入力の場合、すでに配置済みの Z-INDEX の範囲を守らないとランタイムエラーが発生する。

(alpha : UINT) …0～100までの数値で入力する。0で透明、100で不透明となる。

#### 【使用例】

```
.SWF_PLAY  
swf1 swf1 0 0 TOP 100
```

### 3-19.LAYER\_DEL コマンド

【役割】

IMG\_VIEW コマンドや SWF\_PLAY コマンドで作成したメインスクリーンの子オブジェクトを削除する。

【書式】

```
.LAYER_DEL  
{シンボル名}
```

【解説】

.LAYER\_DEL というコマンド指定行の後、必ず改行を行う。

次の行にオプションを記載する。

[オプション]

(シンボル名 : String)…IMG\_VIEW コマンドや SWF\_PLAY コマンドで作成したメインスクリーンの子オブジェクト名。

【使用例】

```
.LAYER_DEL  
swf1
```



### 3-20.SWF\_MOVIE コマンド

【役割】

SWF\_LOAD でロードした SWF を画面全体のムービーとして再生する。再生されたムービーは最終フレームまで再生し終わるか、左クリックでスキップされる。

【書式】

```
.SWF_MOVIE  
{SWF シンボル名}
```

【解説】

. SWF\_MOVIE というコマンド指定行の後、必ず改行を行う。

次の行にオプションを記載する。

[オプション]

(SWF シンボル名 : String) …SWF\_LOAD で指定したシンボル名。

【使用例】

```
.SWF_MOVIE  
swf1
```

### 3-21.FADE\_IN\_ALPHA コマンド

【役割】

プライマリスクリーンをセカンダリスクリーンに転送し、フェイドイン表示する。

【書式】

.FADE\_IN\_ALPHA

【解説】

.FADE\_IN\_ALPHA というコマンドを指定する

【使用例】

.FADE\_IN\_ALPHA

### 3-22.FADE\_OUT\_ALPHA コマンド

【役割】

プライマリスクリーンを背景色で塗りつぶし、その後セカンダリスクリーンに転送してフェイドアウト表示する。

【書式】

.FADE\_OUT\_ALPHA

【解説】

.FADE\_OUT\_ALPHA というコマンドを指定する

【使用例】

.FADE\_OUT\_ALPHA

### 3-23.BACK\_TITLE コマンド

【役割】

タイトル画面に戻る。エンディングの後などに使用する。

【書式】

.BACK\_TITLE

【解説】

.BACK\_TITLE というコマンドを指定する

【使用例】

.BACK\_TITLE