Local Variables Function Category

ローカル変数を扱うファンクション

Functions

aGetPLocalInt
パーティの整数型変数を調べ最大値を返す

aSetPLocalInt
パーティ全員のローカル変数をセットする

DeleteLocalFloat
浮動小数型のローカル変数を削除、開放する

DeleteLocalInt
整数型のローカル変数を削除・開放する

DeleteLocalLocation
ロケーション変数を削除・開放する

DeleteLocalObject
オブジェクト変数を削除・開放する

DeleteLocalString
文字列型変数を削除・開放する

GetLocalArrayInt
配列の整数値を読み出す

GetLocalArrayString
配列の文字列を読み出す

GetLocalFloat
オブジェクトに保存された浮動小数値を得る

GetLocalLocation
オブジェクトに保存されたロケーション型変数の値を得る

GetLocalObject
オブジェクトに保存されたオブジェクト型変数の値を得る

GetLocalString
オブジェクトに保存された文字列を得る

GetPLocalInt
パーティ内のあるPCが持っているローカル変数の値を得る

Global
クエストに関するローカルに格納されたオブジェクトを返す

SetArtifactItem
アーティファクトアイテムのタグを保管する

SetAssassinHead
アサシンの犠牲者の頭のタグをローカル情報に保存する

SetComplexItem
合成アイテムのタグをローカル情報に保存する

SetGlobal
クエスト関連の情報を保存するオブジェクトを設定する

SetLocalArrayInt
配列に整数値を保存する

SetLocalArrayString
配列に整数値を保存する

SetLocalFloat
オブジェクトに浮動小数型ローカル変数を保存する

SetLocalInt
オブジェクトに整数型ローカル変数を保存する

SetLocalLocation
オブジェクトにロケーション型ローカル変数を保存する

SetLocalObject
オブジェクトにオブジェクト型ローカル変数を保存する

SetLocalString
オブジェクトに文字列型ローカル変数を保存する

SetPLocalInt
PCパーティーに格納する変数を設定する

SetWorkingForPlayer
ヘンチマン（仲間になるNPC）が属するPCを設定する

See Also

Function Categories

aGetPLocalInt(object, string)

（パーティの整数型変数を調べ最大値を返す）
PCパーティメンバーに保存された変数の値を比較し、一番大きい値を返す。

```
int aGetPLocalInt(
    object oPC,
    string sLocalName
);
```

Parameters

oPC
PCパーティメンバーの１人。

sLocalName
Pcに保存された変数の名称

Description

戻り値
整数

PCパーティのメンバーに保存された変数の値を比較し、一番大きい値を返す。

この関数はパーティ全員をひとつのものとして GetLocalInt() と同じように取り扱うことができる。
PCパーティ全員の中で一番大きい値を返す。

Remarks

変数が保存されているオブジェクトとして各ＰＣを順に処理していくため、
パラメーターにオブジェクト指定はない。
マルチプレイヤー対応MODをさらに楽しくすることができそう。
報酬を吊り上げた額を比較するなんてことに使えるかも。

Requirements

#include "nw_j_assassin"

Version

1.28

See Also

functions: aSetPLocalInt | GetLocalInt | GetPLocalInt | SetPLocalInt
categories: Local Variables Functions | Module Functions | Party Functions

---

author: Michael Nork, editor: Charles Feduke, JP team: ngtaicho

aSetPLocalInt(object, string, int)

パーティ全員のローカル変数をセットする。

```
void aSetPLocalInt(
   object oPC,
   string sLocalName,
   int nValue
);
```

Parameters

oPC
ＰＣパーティの誰か

sLocalName
保存する変数の名称

nValue
保存する整数値

Description

この関数はＰＣパーティ全員のローカル変数に整数値を設定、保存する。事実上パーティ全体にSetLocalInt（オブジェクト、整数、文字列）を行っている。

Remarks

クエストのフラグセットなどに有効だと思う。

Requirements

#include "nw_j_assassin"

Version

1.28

See Also

functions: aGetPLocalInt | GetPLocalInt | SetPLocalInt
categories: Local Variables Functions | Module Functions | Party Functions

---

author: Michael Nork, editor: Charles Feduke, JP team: ngtaicho

DeleteLocalFloat(object, string)

オブジェクトに保持した浮動小数型のローカル変数を削除（開放）する。

```
void DeleteLocalFloat(
  object oObject,
  string sVarName
);
```

Parameters

oObject
変数を保存する対象オブジェクト

sVarName
削除する変数の名称

Description

oObjectに定義されたsVarNameという名前の浮動小数型の項目を削除・開放する。

Version

1.22

Example

```
// このスクリプトをエリアのOnEnterイベントに設定する。
// 入ってきたオブジェクトがプレイヤーであるかチェックし、
// ＰＣであれば浮動小数型ローカル変数fTimerを削除する。
object oPC = GetEnteringObject();
if (GetIsPC(oPC))
{
  DeleteLocalFloat(oPC, "fDegrees");
}
```

See Also

categories: Local Variables Functions

---

author: Michael Nork, editor: Jeff Lindsey, JP team: ngtaicho

DeleteLocalInt(object, string)
オブジェクトに設定された整数型のローカル変数を削除・開放する。

```
void DeleteLocalInt(
   object oObject,
   string sVarName
);
```

Parameters

oObject
変数を保存する対象オブジェクト

sVarName
削除する変数の名称

Description

oObjectに定義されたsVarNameという名称の整数型ローカル変数を削除・開放する。

Version

1.22

Example

```
// 会話しているプレイヤーの整数型ローカル変数nCounterを削除する。
object oPC = GetPCSpeaker();
DeleteLocalInt(oPC, "nCounter");
```

See Also

categories: Local Variables Functions

---

author: Michael Nork, editor: Jeff Lindsey, JP team: ngtaicho

DeleteLocalLocation(object, string)

オブジェクトに保持したロケーション変数を削除・開放する。

```
void DeleteLocalLocation(
   object oObject,
   string sVarName
);
```

Parameters

oObject
変数を保存する対象オブジェクト

sVarName
削除する変数の名称

Description

oObjectに定義されたsVarNameという名称のロケーション変数（位置情報）を削除・開放する。

Version

1.22

Example

```
// エリアのロケーション変数lMarkerを削除・開放する。
object oArea = GetArea(OBJECT_SELF);
DeleteLocalLocation(oArea, "lMarker");
```

See Also

categories: Local Variables Functions

---

author: Michael Nork, editor: Jeff Lindsey, JP team: ngtaicho

DeleteLocalObject(object, string)

オブジェクトに定義されたオブジェクト変数を削除・開放する

```
void DeleteLocalObject(
   object oObject,
   string sVarName
);
```

Parameters

oObject
変数を保存する対象オブジェクト

sVarName
削除する変数の名称

Description

oObjectに定義されたsVarNameという名称のオブジェクト変数を削除・開放する。

Version

1.22

Example

```
// モジュールを終了するときＰＣのローカルオブジェクト変数oFootlovkerを削除・開放する
object oPC = GetExitingObject();
if (GetIsPC(oPC))
{
  DeleteLocalObject(oPC, "oFootlocker");
}
```

See Also

categories: Local Variables Functions

---

author: Michael Nork, editor: Jeff Lindsey, JP team: ngtaicho

DeleteLocalString(object, string)

オブジェクトに定義された文字列型変数を削除・開放する

```
void DeleteLocalString(
   object oObject,
   string sVarName
);
```

Parameters

oObject
変数を保存する対象オブジェクト

sVarName
削除する変数の名称

Description

oObjectに定義されたsVarNameという名称の文字列型変数を削除・開放する

Version

1.22

Example

```
// sTeamNameという文字列型変数を削除する
DeleteLocalString(OBJECT_SELF, "sTeamName");
```

See Also

categories: Local Variables Functions

---

author: Michael Nork, editor: Jeff Lindsey, JP team: ngtaicho

GetLocalArrayString(object, string, int)

オブジェクトに保存された文字列型配列の値を得る動作をシミュレートする

```
string GetLocalArrayString(
   object oidObject,
   string sVarName,
   int nVarNum
);
```

Parameters

oidObject
文字列型配列が保存されるオブジェクト

sVarName
配列の名称

nVarNum
値を検索する配列の位置、インデックス、添え字

Description

oidObjectに保管したsVarNameという文字列型配列の中のnVarNumの位置にある値を返す

Remarks

nw_o0_itemmaker.nssの中の563行目にこの機能がある。
これは配列のシミュレーションであるが、１つのようにアクセスしたり、ループさせることもできる。
この配列シミュレーター関数で変数を作った場合、実際は"sVarNamen + VarNum"という名称の変数が作成されている。

GetLocalArrayString(OBJECT_SELF, "MyArray", 1);
GetLocalString（OBJECT_SELF,"MyArray1"）
上記２つの関数はまったく同じ動作をする。

仮に300までの配列を作ったとすると、事実上は300個の変数を作ったことになる。
この配列シミュレーター関数を使用すれば、乱雑さを軽減することができるかも知れない。

Requirements

#include "nw_o0_itemmaker"

Version

1.22

See Also

functions: GetLocalString | SetLocalArrayString
categories: Local Variables Functions

---

author: John Shuell, JP team: ngtaicho

GetLocalArrayInt(object, string, int)

オブジェクト上の配列から整数を得る動作をシミュレートする

```
int GetLocalArrayInt(
   object oidObject,
   string sVarName,
   int nVarNum
);
```

Parameters

oidObject
整数配列が保存されるオブジェクト

sVarName
配列の名称

nVarNum
値を検索する配列の位置、インデックス、添え字

Description

oidObjectに保管したsVarNameという整数型配列の中のnVarNumの位置にある値を返す

Remarks

nw_o0_itemmaker.nssの中の600行目にこの機能がある。
これは配列のシミュレーションであるが、１つのようにアクセスしたり、ループさせることもできる。
この配列シミュレーター関数で変数を作った場合、実際は"sVarNamen+VarNum"という名称の変数が作成されている。

GetLocalArrayInt(OBJECT_SELF, "MyArray", 1);
GetLocalInt （OBJECT_SELF,"MyArray1"）
上記２つの関数はまったく同じ動作をする。

仮に300までの配列を作ったとすると、事実上は300個の変数を作ったことになる。
この配列シミュレーター関数を使用すれば、乱雑さを軽減することができるかも知れない。

Requirements

#include "nw_o0_itemmaker"

Version

1.22

See Also

functions: GetLocalInt | SetLocalArrayInt
categories: Local Variables Functions

---

author: John Shuell, JP team: ngtaicho

GetLocalFloat(object, string)

オブジェクトに保存された浮動小数値を得る

```
float GetLocalFloat(
   object oObject,
   string sVarName
);
```

Parameters

oObject
数値を保管したオブジェクト

sVarName
変数の名称

Description

oObjectに保存されたsVarNameという名称の浮動小数型変数の値を読み込み返す。
この関数はSetLocalFloatとセットで使用される。
エラーの場合、関数は"0.0f"を返す。

Remarks

変数名は理論上どんな名前でもよいが、誤動作の可能性を避けるため、"N"と"X"ではじまるものは避けた方が無難。

Version

1.22

Example

```
// "Data"に保管された浮動小数値を返す
void main()
{
   float a = GetLocalFloat(OBJECT_SELF,"Data");
   SendMessageToPC(GetFirstPC(),FloatToString(a));
}
```

See Also

functions: SetLocalFloat
categories: Get Data Functions | Get Data from Creature Functions | Get Data from Object Functions | Local Variables Functions

---

author: GoLeM, editor: Kristian Markon, JP team: ngtaicho

GetLocalLocation(object, string)

オブジェクトに保存されているロケーション形のローカル変数の値を返す

```
location GetLocalLocation(
   object oObject,
   string sVarName
);
```

Parameters

oObject
ロケーション情報が保管されているオブジェクト

sVarName
変数の名称

Description

oObjectに保存されたsVarNameという名称のロケーション変数の値を読み込み返す。この関数はSetLocalLocationとセットで使用される。エラーの場合は無意味なデータを返す。

Remarks

変数名は理論上どんな名前でもよいが、誤動作の可能性を避けるため、"N"と"X"ではじまるものは避けた方が無難。
エラーの場合、無意味なデータが返るため、進行上エラーを引き起こす可能性があるので注意。

ロケーション変数を使って、元にいた位置にポータルで戻るといったことができそう。（ウェイポイントではなく）

Version

1.22

Example

```
// "Data"に保管された位置（ロケーション）にPCをジャンプさせるスクリプト
void main()
{
   location a = GetLocalLocation(OBJECT_SELF,"Data");
   AssignCommand(GetFirstPC(),ActionMoveToLocation(a,TRUE));
}
```

See Also

functions: SetLocalLocation
categories: Get Data Functions | Get Data from Creature Functions | Get Data from Object Functions | Local Variables Functions

---

author: GoLeM, editor: Kristian Markon, JP team: ngtaicho

GetLocalObject(object, string)

オブジェクトに保存されているオブジェクト型ローカル変数の値を返す

```
object GetLocalObject(
   object oObject,
   string sVarName
);
```

Parameters

oObject
オブジェクト情報が保管されているオブジェクト対象

sVarName
変数の名称

Description

戻り値
オブジェクト情報

oObjectに保存されたsVarNameという名称のオブジェクト変数の値を読み込み返す。
この関数はSetLocalObjectとセットで使用される。
エラー時はOBJECT_INVALIDを返す。

Remarks

変数名は理論上どんな名前でもよいが、誤動作の可能性を避けるため、"N"と"X"ではじまるものは避けた方が無難。

身包みはがされた装備を取り戻す、なんてことに使えそう。

Version

1.22

Example

```
// "Data"に保管されたオブジェクトがPCに渡される。
void main()
{
   object a = GetLocalObject(OBJECT_SELF,"Data");
   AssignCommand(GetFirstPC(),ActionMoveToObject(a,TRUE,1.0));
}
```

See Also

functions: SetLocalObject
categories: Get Data Functions | Get Data from Creature Functions | Get Data from Object Functions | Local Variables Functions

---

author: GoLeM, editor: Kristian Markon, JP team: ngtaicho

GetLocalString(object, string)

オブジェクトに保存された文字列を得る

```
string GetLocalString(
   object oObject,
   string sVarName
);
```

Parameters

oObject
文字列を保管したオブジェクト

sVarName
変数の名称

Description

返り値
文字列

oObjectに保存されたsVarNameという名称の浮動小数型変数の値を読み込み返す。
この関数はSetLocalStringとセットで使用される。
エラーの場合、関数はNull（なにもないもの""）を返す。

Remarks

変数名は理論上どんな名前でもよいが、誤動作の可能性を避けるため、"N"と"X"ではじまるものは避けた方が無難。

Version

1.22

Example

```
// "Data"に保管された文字列を返す
void main()
{
   string a = GetLocalString(OBJECT_SELF,"Data");
   SendMessageToPC(GetFirstPC(),a);
}
```

See Also

functions: GetLocalArrayString | SetLocalString
categories: Get Data Functions | Get Data from Creature Functions | Get Data from Object Functions | Local Variables Functions

---

author: GoLeM, editor: Kristian Markon, JP team: ngtaicho

GetPLocalInt(object, string)

パーティ内のあるPCが持っているローカル変数の値を得ます。

```
int GetPLocalInt(
   object oPC,
   string sLocalName
);
```

Parameters

oPC
ローカル変数を探知するパーティのPC

sLocalName
値を得るローカル変数名

Description

殆どのプロットの為に、パーティのプレイヤー全てを一つのものとして検査します。
これはBioWareプロットをより便利にするでしょう。

Remarks

絶対に必要だとは限りませんが、SetPLocalIntが使用されるべきです。SetPLocalIntは全てのパーティメンバーにローカル変数を設定します。

これら２つの関数でマルチプレイヤーゲームが、出て行くプレイヤーや途中参加の新しいプレイヤーを上手く扱えるようになります。

GetPLocalIntはGetPartyLocalIntを短縮したものです。
aGetPLocalInt(object, string)と同じ物になります。

Requirements

#include "nw_i0_plot"

Version

1.28

See Also

functions: aGetPLocalInt | aSetPLocalInt | SetPLocalInt
categories: Local Variables Functions | Module Functions | Party Functions

---

author: Tom Cassiotis, editor: Charles Feduke, JP team: Rainie

Global()

クエストに関する状況の情報を持ったobjectに格納されたローカルのobjectを返します。

```
object Global();
```

Description

NW_J_xxxx_MYGLOBALSの名を持つ呼出objectに、ローカルに格納されたオブジェクトを返します。xxxxはこの関数を呼び出す前に含まれるスクリプトによって決定されます。それが定義されるスクリプトのリストに関しては、Remarksを参照してください。このobjectはクエストに関連する情報および状態をすべて収容するために使用されます。SetPlotTag(sTag)、GetPlotTag()、SetxxxxItem(sTag)そしてGetxxxxItem()に使用されます。xxxxをCOMPLEX、STORYまたはARTIFACTと置き換えるだけです。

Remarks

関数は次のファイルで見つけることができます:
nw_j_artifact.nss: 29
nw_j_story.nss: 29
nw_j_complex.nss: 29

この関数の設定は関連したクエストを持っているNPCの情報を備えたストーリー・プロットアイテム周辺に集約されるように思います。この特殊な関数は利用可能な7つの他の関数から2つを単に利用します。これらの関数の全ては比較的率直に働き、一目で理解することが容易です。

Requirements

#include "nw_j_artifact"

Version

1.22

See Also

functions: GetArtifactItem | GetComplexItem | GetFetchItem | GetPlotTag | GetStoryItem | SetArtifactItem | SetComplexItem | SetGlobal | SetPlotTag | SetStoryItem | TakeStoryItem
categories: Get Data Functions | Local Variables Functions

---

author: John Shuell, JP team: Rainie

SetArtifactItem(string)

アーティファクトアイテムのタグを保管する。

```
void SetArtifactItem(
   string sTag
);
```

Parameters

sTag
アーティファクトアイテムのタグ

Description

GetArtifactItem関数とPlayerHasArtifactItem関数と一緒に使い、
後で使用できるようにアーティファクトアイテムのタグを文字列変数に保管する。

Remarks

nw_j_artifact.nssの97行目にある。

バイオウエア社公式モジュールで使用されるクエストと報酬システムの一部のため、流用は難しい。

Requirements

#include "nw_j_artifact"

Version

1.28

See Also

functions: GetArtifactItem | Global | PlayerHasArtifactItem | TakeArtifactItem
categories: Local Variables Functions | Module Specific Functions

---

author: Lilac Soul, JP team: ngtaicho

SetAssassinHead(string)

アサシンの犠牲者の頭のタグをローカル情報に保存する。

```
void SetAssassinHead(
   string sTag
);
```

Parameters

sTag
犠牲者の頭のタグ

Description

GetAssassinHead関数、PlayerHasHead関数とセットで使用される。アサシンの犠牲者の頭のタグをローカル情報に保管する。

Remarks

nw_j_assassin.nssの299行目で見つかる。

バイオウエア社公式モジュールで使用されるクエストと報酬システムの一部のため、流用は難しい。

誰かの顔を手土産に持っていったり、家の前に飾られたりするクエストの一部かも。

Requirements

#include "nw_j_assassin"

Version

1.28

See Also

functions: GetAssassinHead | PlayerHasHead | SetDoubleCrosserName
categories: Local Variables Functions | Module Specific Functions

---

author: Lilac Soul, editor: Jody Fletcher, JP team: ngtaicho

SetComplexItem(string)

合成アイテムのタグをローカル情報に保存する

```
void SetComplexItem(
   string sTag
);
```

Parameters

sTag
合成アイテムのタグ

Description

GetComplexItem関数、PlayerHasComplexItem関数と一緒に使用される。
合成アイテムのタグをローカル情報に保存する。

Remarks

nw_j_complex.nssの97行目にある。

バイオウエア社公式モジュールで使用されるクエストと報酬システムの一部のため、流用は難しい。

Requirements

#include "nw_j_complex"

Version

1.28

See Also

functions: GetComplexItem | Global | PlayerHasComplexItem | TakeArtifactItem | TakeComplexItem
categories: Local Variables Functions | Module Specific Functions

---

author: Lilac Soul, JP team: ngtaicho

SetGlobal(object, object)

クエスト関連の情報を保存するオブジェクトを設定する。

```
void SetGlobal(
    object oTarget,
    object oGlobal
);
```

Parameters

oTarget
クエスト関連の情報を保存するオブジェクト

oGlobal
Global()関数で作成されるクエスト情報が保持されたオブジェクト

Description

Global()関数を使って作られたクエスト関連の情報を保持するオブジェクトを、oTargetに作成する。

Remarks

クエスト関連の情報を保存して、単純なクエストを書くことを簡単にするために使われる一連のスクリプトの一部。

この特殊な関数は様々なファイルで定義されている。
nw_j_complex.nss：44行目
nw_j_story.nss：47行目
nw_j_theft.nss：44行目
nw_j_fetch.nss：47行目
nw_j_artifact.nss：47行目
nw_j_assassin.nss：107行目
nw_j_guard.nss：47行目
nw_j_rescue.nss：64行目

Requirements

#include "nw_j_artifact"

Version

1.22

See Also

functions: Global | TakeArtifactItem | TakeComplexItem | TakeStoryItem
categories: Local Variables Functions

---

author: John Shuell, JP team: ngtaicho

SetLocalArrayInt(object, string, int, int)

オブジェクトに設定された配列に数値をセットする動作をシミュレートする

```
void SetLocalArrayInt(
   object oidObject,
   string sVarName,
   int nVarNum,
   int nValue
);
```

Parameters

oidObject
配列を保存するオブジェクト

sVarName
配列の名称

nVarNum
配列上に保存する位置、インデックス、添え字

nValue
配列に保存する数値

Description

これは配列のシミュレーションであるが、１つのようにアクセスしたり、ループさせることもできる。
この配列シミュレーター関数で変数を作った場合、実際は"sVarNamen+VarNum"という名称の変数が作成されている。
SetLocalArrayInt(oPC, "myarray", 1, 10);
SetLocalInt(oPC, "myarray1", 10);
上記２つの関数はまったく同じ動作をする。

Remarks

nw_o0_itemmaker.nssの22行目に見つかる。
実際に配列は有効です。なぜなら：
a)コードの乱雑さを軽減することができる
b)変数名を個々に作るよりも、楽に連続して作ってまとめることができる

Known Bugs

SetLocalString(オブジェクト, 文字列, 文字列)と同じバグがある。

Requirements

#include "nw_o0_itemmaker"

Version

1.28

Example

```
//Sets 10 variables quickly on the PC, using a for loop.
//myarray1 is set to 1, myarray2 is set to 2, etc.
#include "nw_o0_itemmaker"
void main()
{
    object oPC = GetPCSpeaker();

    int nLoop;
    for (nLoop=1; nLoop<=10; nLoop++)
    {
        SetLocalArrayInt(oPC, "myarray", nLoop, nLoop);
    }
}
```

See Also

functions: GetLocalArrayInt | GetLocalInt | SetLocalInt
categories: Local Variables Functions

---

author: Lilac Soul, editor: Charles Feduke, additional contributor(s): Graziano Lenzi, JP team: geshi, ngtaicho

SetLocalArrayString(object, string, int, string)

オブジェクトに設定された配列に文字列をセットする動作をシミュレートする

```
void SetLocalArrayString(
   object oidObject,
   string sVarName,
   int nVarNum,
   string nValue
);
```

Parameters

oidObject
配列を保存するオブジェクト

sVarName
配列の名称

nVarNum
配列上に保存する位置、インデックス、添え字

nValue
配列に保存する文字列

Description

これは配列のシミュレーションであるが、１つのようにアクセスしたり、ループさせることもできる。
この配列シミュレーター関数で変数を作った場合、実際は"sVarNamen+VarNum"という名称の変数が作成されている。
SetLocalArrayString(oPC, "myarray", 1, "hello");
SetLocalString(oPC, "myarray1", "hello");
上記２つの関数はまったく同じ動作をする。

Remarks

nw_o0_itemmaker.nssの20行目に見つかる。
実際に配列は有効です。なぜなら：
a)コードの乱雑さを軽減することができる
b)変数名を個々に作るよりも、楽に連続して作ってまとめることができる

Known Bugs

SetLocalString(オブジェクト, 文字列, 文字列)と同じバグがあるらしい。

Requirements

#include "nw_o0_itemmaker"

Version

1.28

Example

```
// 10個の変数をPCにすばやくセットするループ
// myarray1に"number 1"をセット、
// myarray2に"number 2"をセット・・・
#include "nw_o0_itemmaker"
void main()
{
object oPC=GetPCSpeaker();

int nLoop;
for (nLoop=1; nLoop<=10; nLoop++)
  {
  SetLocalArrayString(oPC, "myarray", nLoop, "number "+IntToString(nLoop));
  }
}
```

See Also

functions: GetLocalArrayString | SetLocalString
categories: Local Variables Functions

---

author: Lilac Soul, editor: Charles Feduke, additional contributor(s): Graziano Lenzi, JP team: geshi, ngtaicho

SetLocalInt(object, string, int)

オブジェクトに整数型ローカル変数を保存する。

```
void SetLocalInt(
   object oObject,
   string sVarName,
   int nValue
);
```

Parameters

oObject
ローカル変数を保存するオブジェクト

sVarName
保存する変数の名称（ファイル名みたいなもの）

nValue
保存する整数値

Description

oObjectで指定されたオブジェクトにsVarNameという名前の整数型保管領域を確保し、nValueの値を保存する。同じオブジェクトの中に同じ名称で再度保存した場合、前の値は失われ、上書きされてしまう。

Remarks

これらの関数は一見、ローカル変数に値を代入する命令に見えるかも知れませんが、実際にはファイルの入出力に近いものと言えます。

SetLocalInt(オブジェクト, ローカル変数名, 整数値)
これを
SetLocalInt(記憶媒体, ファイル名, データ)
という感じで見たほうがわかりいいかも知れません。

オブジェクト（記憶媒体）が違えば、ローカル変数名（ファイル名）が同じでも別の値が保持できますし、整数値（データ）が入った変数名と、ローカル変数名（ファイル名）は一致させる必要もありません

Known Bugs

クリエイト関数で作成された直後のSetLocal関数はタイミングによって失敗する可能性を秘めている。詳しくはSetLocalString関数の説明にて。

Version

1.28

Example

```
void main()
{
    object oThis = OBJECT_SELF;
    string sKey = “ foo ” ;
    int iValue = 2;
    // "foo"という保存場所に2の値を保管する。
    // この値は後に GetLocalInt を使って読み出すことができる。
    SetLocalInt(oThis, sKey, iValue);
}
```

See Also

functions: GetLocalInt | SetLocalArrayInt
categories: Local Variables Functions

---

author: Daniel Beckman, editor: Charles Feduke, additional contributor(s): Xepherys, Graziano Lenzi, JP team: ngtaicho

SetLocalFloat(object, string, float)

オブジェクトに浮動小数型ローカル変数を保存する。

```
void SetLocalFloat(
   object oObject,
   string sVarName,
   float fValue
);
```

Parameters

oObject
ローカル変数を保存するオブジェクト

sVarName
保存する変数の名称（ファイル名みたいなもの）

fValue
保存する浮動小数値

Description

oObjectで指定されたオブジェクトにsVarNameという名前の整数型保管領域を確保し、fValueの値を保存する。同じオブジェクトの中に同じ名称で再度保存した場合、前の値は失われ、上書きされてしまう。

Remarks

これらの関数は一見、ローカル変数に値を代入する命令に見えるかも知れませんが、実際にはファイルの入出力に近いものと言えます。

SetLocalFloat(オブジェクト, ローカル変数名, 浮動小数値)
これを
SetLocalFloat(記憶媒体, ファイル名, データ)
という感じで見たほうがわかりいいかも知れません。

オブジェクト（記憶媒体）が違えば、ローカル変数名（ファイル名）が同じでも別の値が保持できますし、浮動小数値（データ）が入った変数名と、ローカル変数名（ファイル名）は一致させる必要もありません。

Known Bugs

クリエイト関数で作成された直後のSetLocal関数はタイミングによって失敗する可能性を秘めている。詳しくはSetLocalString関数の説明にて。

Version

1.28

Example

```
// SetLocalFloat構文の例
void main()
{
    object oThis = OBJECT_SELF;
    string sKey = “ foo ”;
    float fValue = 20.0;
    SetLocalFloat(oThis, sKey, fValue);
}
```

See Also

functions: GetLocalFloat
categories: Local Variables Functions

---

author: Daniel Beckman, editor: Charles Feduke, additional contributor(s): Graziano Lenzi, JP team: ngtaicho

SetLocalLocation(object, string, location)

オブジェクトにロケーション型ローカル変数を保存する。

```
void SetLocalLocation(
   object oObject,
   string sVarName,
   location lValue
);
```

Parameters

oObject
ローカル変数を保存するオブジェクト

sVarName
保存する変数の名称（ファイル名みたいなもの）

lValue
保存するロケーション情報

Description

oObjectで指定されたオブジェクトにsVarNameという名前のロケーション型保管領域を確保し、lValueの値を保存する。同じオブジェクトの中に同じ名称で再度保存した場合、前の値は失われ、上書きされてしまう。

Remarks

これらの関数は一見、ローカル変数に値を代入する命令に見えるかも知れませんが、実際にはファイルの入出力に近いものと言えます。

SetLocalLocation(オブジェクト, ローカル変数名, ロケーション情報)
これを
SetLocalLocation(記憶媒体, ファイル名, データ)
という感じで見たほうがわかりいいかも知れません。

オブジェクト（記憶媒体）が違えば、ローカル変数名（ファイル名）が同じでも別の値が保持できますし、ロケーション情報（データ）が入った変数名と、ローカル変数名（ファイル名）は一致させる必要もありません。

Known Bugs

クリエイト関数で作成された直後のSetLocal関数はタイミングによって失敗する可能性を秘めている。詳しくはSetLocalString関数の説明にて。

Version

1.28

Example

```
// SetLocalLocation関数の簡単な構文例
void main()
{
    object oThis = OBJECT_SELF;
    string sKey = “ foo ”;
    object oArea = GetArea(OBJECT_SELF);
    vector vPosition = Vector(0.0, 0.0, 0.0)
    location lValue = Location(oArea, vPosition, 0.0);
    SetLocalLocation(oThis, sKey, lValue);
}
```

See Also

functions: GetLocalLocation
categories: Local Variables Functions

---

author: Daniel Beckman, editor: Charles Feduke, additional contributor(s): Steve U., Graziano Lenzi, JP team: ngtaicho

SetLocalObject(object, string, object)

オブジェクトにオブジェクト型ローカル変数を保存する。

```
void SetLocalObject(
   object oObject,
   string sVarName,
   object oValue
);
```

Parameters

oObject
ローカル変数を保存するオブジェクト

sVarName
保存する変数の名称（ファイル名みたいなもの）

oValue
保存するオブジェクト情報

Description

oObjectで指定されたオブジェクトにsVarNameという名前のオブジェクト型保管領域を確保し、oValueの値を保存する。同じオブジェクトの中に同じ名称で再度保存した場合、前の値は失われ、上書きされてしまう。

Remarks

これらの関数は一見、ローカル変数に値を代入する命令に見えるかも知れませんが、実際にはファイルの入出力に近いものと言えます。

SetLocalObject(オブジェクト, ローカル変数名, オブジェクト情報)
これを
SetLocalObject(記憶媒体, ファイル名, データ)
という感じで見たほうがわかりいいかも知れません。

オブジェクト（記憶媒体）が違えば、ローカル変数名（ファイル名）が同じでも別の値が保持できますし、オブジェクト情報（データ）が入った変数名と、ローカル変数名（ファイル名）は一致させる必要もありません。

Known Bugs

クリエイト関数で作成された直後のSetLocal関数はタイミングによって失敗する可能性を秘めている。詳しくはSetLocalString関数の説明にて。

Version

1.28

Example

```
// モジュール内に"npc_henchment"とタグをつけられている
// オブジェクトがいると想定する。
void main()
{
    object oThis = OBJECT_SELF;
    object oObject = GetObjectByTag("npc_henchment");
    SetLocalObject(oThis, "hench", oObject);
}
```

See Also

functions: GetLocalObject
categories: Local Variables Functions

---

author: Daniel Beckman, editor: Charles Feduke, additional contributor(s): Xepherys, Graziano Lenzi, JP team: ngtaicho

SetLocalString(object, string, string)

オブジェクトに文字列型ローカル変数を保存する。

```
void SetLocalString(
   object oObject,
   string sVarName,
   string sValue
);
```

Parameters

oObject
ローカル変数を保存するオブジェクト

sVarName
保存する変数の名称（ファイル名みたいなもの）

sValue
保存する文字列

Description

oObjectで指定されたオブジェクトにsVarNameという名前の整数型保管領域を確保し、sValueの値を保存する。同じオブジェクトの中に同じ名称で再度保存した場合、前の値は失われ、上書きされてしまう。

Remarks

これらの関数は一見、ローカル変数に値を代入する命令に見えるかも知れませんが、実際にはファイルの入出力に近いものと言えます。

SetLocalString(オブジェクト, ローカル変数名, 文字列)
これを
SetLocalString(記憶媒体, ファイル名, データ)
という感じで見たほうがわかりいいかも知れません。

オブジェクト（記憶媒体）が違えば、ローカル変数名（ファイル名）が同じでも別の値が保持できますし、文字列（データ）が入った変数名と、ローカル変数名（ファイル名）は一致させる必要もありません。

Known Bugs

クリエイト関数で作成された直後のSet関数はタイミングによって失敗する可能性を秘めている。クリエイト関数で作られたばかりのオブジェクト、あるいはクリーチャーにこの関数を使用する場合、

SetLocalString(object, string, string)
のかわりに
ActionDoCommand(SetLocalString(object, string, string))
を使ってアクションキュー（行動待ち行列）に入れておいたほうが確実に関数が実行される。これは恐らく、オブジェクト／クリーチャー発生開始してすぐは直接の関数を受け付けない時間が存在するか、クリエート直後に設定の初期化が行われているためと思われる。

クリエート直後はActionDoCommand(SetLocal系関数)というふうにパターン化して使用しよう。

※このバグはSetLocal系の全ての関数に存在します。

Version

1.28

Example

```
// 後に下のGetLocal関数で読み出すことができる文字列を
// OBJECT_SELFにセットする
// string sValue = GetLocalString(OBJECT_SELF, "foo");
void main()
{
    object oThis = OBJECT_SELF;
    string sKey = “ foo ” ;
    string sValue = “ some string ” ;
    SetLocalString(oThis, sKey, sValue);
}

/* 以下の２つの例はnObjectType、sTemplate、lLocationが
     既に設定されているものとします。
     CreateObject(int, string, location, int)も参考にして下さい。
*/

// 不確実
// クリエイトオブジェクトされたものに反映されない可能性がある。
object oTarget = CreateObject(nObjectType, sTemplate, lLocation, FALSE);
SetLocalString(oTarget, "STRING_NAME", "This is my string value");

// 確実
// クリエイトオブジェクトされたものに確実に反映される
object oTarget = CreateObject(nObjectType, sTemplate, lLocation, FALSE);
action aSetString = ActionDoCommand(
    SetLocalString(oTarget, "STRING_NAME", "This is my string")
    );
AssignCommand(oTarget, aSetString);
```

See Also

functions: GetLocalString | SetLocalArrayString
categories: Local Variables Functions

---

author: Daniel Beckman, editor: Charles Feduke, additional contributor(s): Xepherys, Graziano Lenzi, JP team: ngtaicho

SetPLocalInt(object, string, int)

PCパーティーに格納する変数を設定します。

```
void SetPLocalInt(
   object oPC,
   string sLocalName,
   int nValue
);
```

Parameters

oPC
パーティ内のPC

sLocalName
変数名

nValue
持たせる変数の値

Description

oPCが所属しているパーティーに属する全てのPCに対して、nValueの値を持ったsLocalNameというローカルの整数を設定します。

Remarks

PCのパーティ全体の整数を探査するマルチプレイヤーを便利にする方法です。aSetPLocalInt(object, string, int)と同じです。

Requirements

#include "nw_i0_plot"

Version

1.28

See Also

functions: aGetPLocalInt | aSetPLocalInt | GetPLocalInt
categories: Local Variables Functions | Module Functions | Party Functions

---

author: Charles Feduke, additional contributor(s): Tom Cassiotis, JP team: Rainie

SetWorkingForPlayer(object)

ヘンチマン（仲間になるNPC）が属するPCを設定します。

```
void SetWorkingForPlayer(
   object oPC
);
```

Parameters

oPC
ヘンチマンを属させるプレイヤー

Description

（NW_L_HIRED＋ヘンチマンのtag）から10までのローカル変数を設定します。

Remarks

Script: nw_i0_henchman.nss: Line: 116

Requirements

#include "nw_i0_henchman"

Version

1.29

See Also

functions: GetFormerMaster | GetWorkingForPlayer | SetFormerMaster
categories: Henchmen/Familiars/Summoned Functions | Local Variables Functions

---

author: Jody Fletcher, JP team: Rainie