

StkThread.Lib

1. StkThread.Libについて

1.1 StkThread.Lib概要

1.2 システム要件

1.3 システム構成

2. ユーザープログラムで定義するコールバックメソッド

2.1 コールバックメソッド概要

2.2 コールバックメソッド詳細

2.2.1 int ElemStkThreadStart(int Id)

2.2.2 int ElemStkThreadStop(int Id)

2.2.3 int ElemStkThreadInit(int Id)

2.2.4 int ElemStkThreadMain(int Id)

2.2.5 int ElemStkThreadFinal(int Id)

2.2.6 void ProcBeforeFirstStkThreadStarts(void)

2.2.7 void ProcAfterLastStkThreadStops(void)

3. StkThread.Libのユーザープログラム用API

3.1 ユーザープログラム用API概要

3.2 ユーザープログラム用API詳細

3.2.1 int GetNumOfRunStkThread()

3.2.2 int GetNumOfStkThread()

3.2.3 void AddStkThread(int Id, TCHAR Name[MAX_LENGTH_OF_STKTHREAD_NAME], TCHAR Description[MAX_LENGTH_OF_STKTHREAD_DESCRIPTION], void* Init, void* Final, void* Main, void* Start, void* Stop)

3.2.4 void DeleteStkThread(int Id)

3.2.5 void SetProcBeforeFirstStkThreadStarts(void* Proc)

3.2.6 void SetProcAfterLastStkThreadStops(void* Proc)

3.2.7 int StartSpecifiedStkThreads(int Ids[MAX_NUM_OF_STKTHREADS], int NumOfIds)

3.2.8 int StopSpecifiedStkThreads(int Ids[MAX_NUM_OF_STKTHREADS], int NumOfIds)

3.2.9 void SetStkThreadInterval(int TargetId, int Millisec)

3.2.10 int GetStkThreadInterval(int TargetId)

3.2.11 int GetStkThreadIdByIndex(int Index)

3.2.12 int GetStkThreadStatus(int TargetId)

3.2.13 int GetStkThreadStatusByIndex(int Index)

3.2.14 int GetStkThreadName(int TargetId, TCHAR Name[MAX_LENGTH_OF_STKTHREAD_NAME])

3.2.15 int GetStkThreadNameByIndex(int Index, TCHAR Name[MAX_LENGTH_OF_STKTHREAD_NAME])

3.2.16 int GetStkThreadDescription(int TargetId, TCHAR Description[MAX_LENGTH_OF_STKTHREAD_DESCRIPTION])

3.2.17 int GetStkThreadDescriptionByIndex(int Index, TCHAR Description[MAX_LENGTH_OF_STKTHREAD_DESCRIPTION])

3.3 ユーザープログラム用定数値

5. その他

5.1 各管理リソースの上限値について

1. StkThread.Libについて

1.1 StkThread.Lib概要

StkThread.Libはスレッド管理用の機能・処理を提供する静的ライブラリで、C言語用APIを提供します。開発対象のアプリケーションの動作が複数のスレッドの動作に基づいている場合、StkThread.Libを利用することで開発コストを低減することができます。

本ドキュメントでは、StkThread.LibとStkThreadを以下のとおり定義します。

StkThread.Lib	スレッド管理用の機能・処理を提供する静的ライブラリ
StkThread	StkThread.Libで管理する個々のスレッド

1.2 システム要件

StkThread.Libは以下の環境で動作します。

オペレーティングシステム:

- Windows Vista Business/Ultimate SP2
- Windows 7 Professional/Ultimate SP1
- Windows 8 Pro
- Windows 8.1 Pro
- Windows Server 2008 Standard SP2
- Windows Server 2008 R2 Standard SP1
- Windows Server 2012 Standard
- Windows Server 2012 R2 Standard

前提となるソフトウェア:

- Microsoft Visual Studio 2008 (Visual C++)

1.3 システム構成

StkThread.Libは以下のファイルから構成されます。

stkthread.lib	StkThread.Libの静的ライブラリです。
stkthread.h	StkThread.Lib利用プログラムのためのヘッダファイルです。

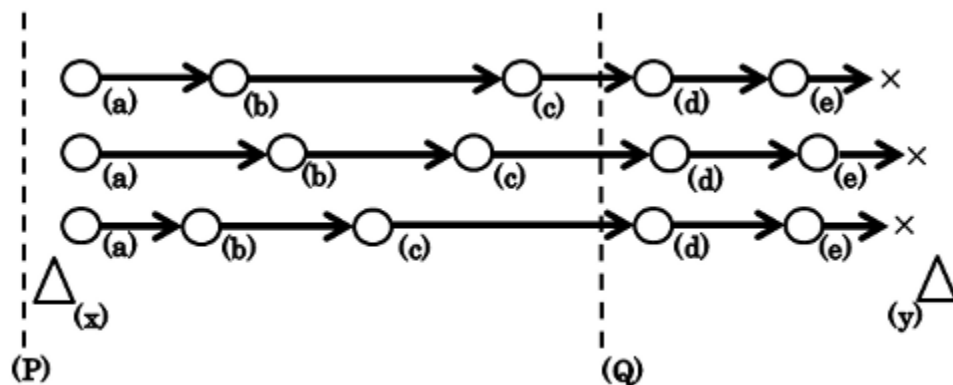
2. ユーザープログラムで定義するコールバックメソッド

2.1 コールバックメソッド概要

StkThread.Libでは、各スレッド(StkThread)の振る舞いをコールバックメソッドで定義します。
StkThread.Libの利用プログラムは、あらかじめ実装したコールバックメソッドをStkThread.Libに登録します。
コールバックメソッドの種類を以下に示します。

コールバックメソッド(仮名)	コールバックメソッドを呼び出すタイミング
ProcBeforeFirstStkThreadStarts	最初のスレッドが開始される直前
ProcAfterLastStkThreadStops	最後のスレッドが終了した直後
ElemStkThreadStart	スレッド開始要求受理時。各スレッド毎定義
ElemStkThreadStop	スレッド終了要求受理時。各スレッド毎定義
ElemStkThreadInit	スレッド開始直後。各スレッド毎定義
ElemStkThreadMain	スレッド実行時。繰り返し呼び出される。各スレッド毎定義
ElemStkThreadFinal	スレッド終了直前。各スレッド毎定義

時間が左側から右側に進むとして3つのStkThreadの各コールバックメソッドが呼び出されるタイミング(例)を以下に図示します。



- (a): ElemStkThreadStartのコールバック
- (b): ElemStkThreadInitのコールバック
- (c): ElemStkThreadMainのコールバック
- (d): ElemStkThreadStopのコールバック
- (e): ElemStkThreadFinalのコールバック
- (x): ProcBeforeFirstStkThreadStartsのコールバック
- (y): ProcAfterLastStkThreadStopsのコールバック
- (P): ユーザープログラムからのStartSpecifiedStkThreadsの呼び出し
- (Q): ユーザープログラムからのStopSpecifiedStkThreadsの呼び出し

StkThread.Libは、StartSpecifiedStkThreadsメソッドが呼び出されたことを契機にユーザープログラムで定義したElemStkThreadStartをコールバックしますが、これに先立ちProcBeforeFirstStkThreadStartsメソッドをコールバックします。

また、StkThread.Libは、StopSpecifiedStkThreadsメソッドが呼び出されたことを契機にElemStkThreadStopをコールバックし、最後にProcAfterLastStkThreadStopsメソッドをコールバックします。

2.2 コールバックメソッド詳細

2.2.1 int ElemStkThreadStart(int Id)

StkThread.Libは、スレッド開始要求受理時にユーザープログラムで定義した本コールバックメソッドを呼び出します。本コールバックメソッドはスレッド(StkThread)毎に定義することができます。
コールバックメソッドの名称はユーザーが自由に定義することができ、AddStkThreadメソッドでそのコールバックメソッドのポインタを指定します。

コールバックメソッドが呼び出される契機は、ユーザープログラムがStartSpecifiedStkThreadsを呼び出した直後（スレッド開始要求受理時）となります。

Id [入力]	スレッド開始対象となるStkThreadのID
[戻り値]	ユーザープログラムのコールバックメソッドはint型の値を返さなければなりませんが、この値はコールバックメソッドの呼び出し元では評価しません。特に意図が無ければ0を返すようにしてください。

2.2.2 int ElemStkThreadStop(int Id)

StkThread.Libは、スレッド終了要求受理時にユーザープログラムで定義した本コールバックメソッドを呼び出します。本コールバックメソッドはスレッド(StkThread)毎に定義することができます。

コールバックメソッドの名称はユーザーが自由に定義することができ、AddStkThreadメソッドでそのコールバックメソッドのポインタを指定します。

コールバックメソッドが呼び出される契機は、ユーザープログラムがStopSpecifiedStkThreadsを呼び出した直後（スレッド終了要求受理時）となります。

Id [入力]	スレッド終了対象となるStkThreadのID
[戻り値]	ユーザープログラムのコールバックメソッドはint型の値を返さなければなりませんが、この値はコールバックメソッドの呼び出し元では評価しません。特に意図が無ければ0を返すようにしてください。

2.2.3 int ElemStkThreadInit(int Id)

StkThread.Libは、スレッド開始直後にユーザープログラムで定義した本コールバックメソッドを呼び出します。

本コールバックメソッドはスレッド(StkThread)毎に定義することができます。

コールバックメソッドの名称はユーザーが自由に定義することができ、AddStkThreadメソッドでそのコールバックメソッドのポインタを指定します。

コールバックメソッドが呼び出される契機は、ユーザープログラムがStartSpecifiedStkThreadsを呼び出し、実際のスレッドが開始したとき（スレッド開始直後）となります。

Id [入力]	スレッド開始対象となるStkThreadのID
[戻り値]	ユーザープログラムのコールバックメソッドはint型の値を返さなければなりませんが、この値はコールバックメソッドの呼び出し元では評価しません。特に意図が無ければ0を返すようにしてください。

2.2.4 int ElemStkThreadMain(int Id)

StkThread.Libは、スレッド実行時にユーザープログラムで定義した本コールバックメソッドを定期的に呼び出します。本コールバックメソッドはスレッド(StkThread)毎に定義することができます。

コールバックメソッドの名称はユーザーが自由に定義することができ、AddStkThreadメソッドでそのコールバックメソッドのポインタを指定します。

コールバックメソッドが呼び出される契機は、ユーザープログラムがStartSpecifiedStkThreadsを呼び出し、実際のスレッドが開始した後（スレッド実行後）となります。また、コールバックメソッドはGetStkThreadIntervalで取得した間隔で定期的に呼び出されます。

本コールバックメソッドが呼び出される間隔はSetStkThreadIntervalメソッドで変更することができます。

Id [入力]	StkThreadのID
[戻り値]	ユーザープログラムのコールバックメソッドはint型の値を返さなければなりませんが、この値はコールバックメソッドの呼び出し元では評価しません。特に意図が無ければ0を返すようにしてください。

2.2.5 int ElemStkThreadFinal(int Id)

StkThread.Libは、スレッド終了直前にユーザープログラムで定義した本コールバックメソッドを呼び出します。

本コールバックメソッドはスレッド(StkThread)毎に定義することができます。
コールバックメソッドの名称はユーザーが自由に定義することができ、AddStkThreadメソッドでそのコールバックメソッドのポインタを指定します。
コールバックメソッドが呼び出される契機は、ユーザープログラムがStopSpecifiedStkThreadsを呼び出し、実際のスレッドが終了したとき(スレッド終了直前)となります。

Id [入力]	スレッド終了対象となるStkThreadのID
[戻り値]	ユーザープログラムのコールバックメソッドはint型の値を返さなければなりません、この値はコールバックメソッドの呼び出し元では評価しません。特に意図が無ければ0を返すようにしてください。

2.2.6 void ProcBeforeFirstStkThreadStarts(void)

StkThread.Libは、最初のスレッドが開始する直前にユーザープログラムで定義した本コールバックメソッドを呼び出します。
コールバックメソッドの名称はユーザーが自由に定義することができ、SetProcBeforeFirstStkThreadStartsメソッドでそのコールバックメソッドのポインタを指定します。
コールバックメソッドが呼び出される契機は、ユーザープログラムがStartSpecifiedStkThreadsを呼び出し、最初のスレッドが開始する直前となります。

2.2.7 void ProcAfterLastStkThreadStops(void)

StkThread.Libは、最後のスレッドが終了した直後にユーザープログラムで定義した本コールバックメソッドを呼び出します。
コールバックメソッドの名称はユーザーが自由に定義することができ、SetProcAfterLastStkThreadStopsメソッドでそのコールバックメソッドのポインタを指定します。
コールバックメソッドが呼び出される契機は、ユーザープログラムがStopSpecifiedStkThreadsを呼び出し、最後のスレッドが終了した直後となります。

3. StkThread.Libのユーザープログラム用API

3.1 ユーザープログラム用API概要

StkThreadのユーザープログラム用APIの一覧を以下に示します。

APIシグネチャ	概要
int GetNumOfRunStkThread()	現在実行中のスレッドの数を返却します。
int GetNumOfStkThread()	登録済の全てのスレッドの数を返却します。
void AddStkThread(int Id, TCHAR Name[MAX_LENGTH_OF_STKTHREAD_NAME], TCHAR Description[MAX_LENGTH_OF_STKTHREAD_DESCRIPTION], void* Init, void* Final, void* Main, void* Start, void* Stop)	新たなスレッドを登録します。
void DeleteStkThread(int Id)	登録されているスレッドを削除します。
void SetProcBeforeFirstStkThreadStarts(void* Proc)	コールバックメソッド ProcBeforeFirstStkThreadStarts のポインタを設定します。
void SetProcAfterLastStkThreadStops(void* Proc)	コールバックメソッド ProcAfterLastStkThreadStops のポインタを設定します。
int StartSpecifiedStkThreads(int Ids[MAX_NUM_OF_STKTHREADS], int NumOfIds)	指定されたIDのスレッドを開始します。
int StopSpecifiedStkThreads(int Ids[MAX_NUM_OF_STKTHREADS], int NumOfIds)	指定されたIDのスレッドを停止します。
void SetStkThreadInterval(int TargetId, int Millisec)	ElemStkThreadMainコールバックメソッドを定期的呼び出す間隔をミリ秒単位で指定します。
int GetStkThreadInterval(int TargetId)	ElemStkThreadMainコールバックメソッドを定期的呼び出す間隔を取得します。
int GetStkThreadIdByIndex(int Index)	指定した要素番号のIDを取得します。
int GetStkThreadStatus(int TargetId)	指定したIDのスレッドの状態を取得します。
int GetStkThreadStatusByIndex(int Index)	指定した要素番号のスレッドの状態を取得します。
int GetStkThreadName(int TargetId, TCHAR Name[MAX_LENGTH_OF_STKTHREAD_NAME])	指定したIDのスレッドの名前を取得します。
int GetStkThreadNameByIndex(int Index, TCHAR Name[MAX_LENGTH_OF_STKTHREAD_NAME])	指定した要素番号のスレッドの名前を取得します。
int GetStkThreadDescription(int TargetId, TCHAR Description[MAX_LENGTH_OF_STKTHREAD_DESCRIPTION])	指定したIDのスレッドの詳細説明を取得します。
int GetStkThreadDescriptionByIndex(int Index, TCHAR Description[MAX_LENGTH_OF_STKTHREAD_DESCRIPTION])	指定した要素番号のスレッドの詳細説明を取得します。

3.2 ユーザープログラム用API詳細

3.2.1 int GetNumOfRunStkThread()

現在実行中のスレッドの数を返却します。具体的にはElemStkThreadInitメソッドの開始直前からElemStkThreadFinalメソッドの終了直後までをスレッド実行中と解釈し、その間の処理を実行しているすべてのスレッドの数を返却します。

[入力]	なし
[戻り値]	現在実行中のスレッドの数

3.2.2 int GetNumOfStkThread()

登録済の全てのスレッドの数を返却します。スレッドの登録および削除は AddStkThreadメソッドおよび DeleteStkThreadメソッドを用いて実施することができます。

[入力]	なし
[戻り値]	全てのスレッドの数

3.2.3 void AddStkThread(int Id, TCHAR Name[MAX_LENGTH_OF_STKTHREAD_NAME], TCHAR Description[MAX_LENGTH_OF_STKTHREAD_DESCRIPTION], void* Init, void* Final, void* Main, void* Start, void* Stop)

新たなスレッドを登録します。

下記の何れかに該当する場合、本メソッドはスレッドを登録せず処理は呼び出し元に戻ります。

- すでに最大数のスレッドが登録されている場合
- 指定されたNameおよびDescriptionの文字長が最大長を超えている場合
- 指定されたIdと同じIDをもつスレッドがすでに登録されている場合

Id[入力]	StkThread ID (StkThreadが内部的に管理するスレッドのIDであり一般的なスレッドIDではない)
Name[入力]	スレッドの名前
Description[入力]	スレッドの詳細説明
Init[入力]	ElemStkThreadInit(コールバックメソッド)へのポインタ。NULLを指定した場合コールバックは無効。
Final[入力]	ElemStkThreadFinal(コールバックメソッド)へのポインタ。NULLを指定した場合コールバックは無効。
Main[入力]	ElemStkThreadMain(コールバックメソッド)へのポインタ。NULLを指定した場合コールバックは無効。
Start[入力]	ElemStkThreadStart(コールバックメソッド)へのポインタ。NULLを指定した場合コールバックは無効。
Stop[入力]	ElemStkThreadStop(コールバックメソッド)へのポインタ。NULLを指定した場合コールバックは無効。
[戻り値]	なし

3.2.4 void DeleteStkThread(int Id)

登録されているスレッドを削除します。

スレッドを削除するにあたり、本メソッドは内部的にStopSpecifiedStkThreadsメソッドを呼び出し、実行中のスレッドの停止処理を行います。

指定したIDのスレッドが存在しない場合、要求は無視され処理は呼び出し元に戻ります。

Id[入力]	StkThread ID (StkThreadが内部的に管理するスレッドのIDであり一般的なスレッドIDではない)
[戻り値]	なし

3.2.5 void SetProcBeforeFirstStkThreadStarts(void* Proc)

ProcBeforeFirstStkThreadStarts(コールバックメソッド)を設定します。

Id[入力]	ProcBeforeFirstStkThreadStarts(コールバックメソッド)へのポインタ。NULLを指定した場合コールバックは無効。
--------	--

[戻り値]	なし
-------	----

3.2.6 void SetProcAfterLastStkThreadStops(void* Proc)

ProcAfterLastStkThreadStops(コールバックメソッド)を設定します。

Id[入力]	ProcAfterLastStkThreadStops(コールバックメソッド)へのポインタ。NULLを指定した場合コールバックは無効。
[戻り値]	なし

3.2.7 int StartSpecifiedStkThreads(int Ids[MAX_NUM_OF_STKTHREADS], int NumOfIds)

指定されたID(StkThread ID)のスレッドを実行します。
このメソッドは最初に現在実行中のスレッドが存在するかどうかをチェックし、指定されたスレッドが最初の実行スレッドと判断した場合、ProcBeforeFirstStkThreadStartsをコールバックします。
その後ElemStkThreadStartおよびElemStkThreadInitのコールバックを経てElemStkThreadMainを定期的にコールバックします。
指定したIDのStkThreadが存在しない場合、要求は無視されます。

Ids[入力]	実行するスレッドのIDを含む配列。複数のIDを指定することができます。
NumOfIds[入力]	Idsに含まれるID(要素)の数
[戻り値]	本メソッドを呼び出す前に実行していたスレッドの数

3.2.8 int StopSpecifiedStkThreads(int Ids[MAX_NUM_OF_STKTHREADS], int NumOfIds)

指定されたID(StkThread ID)のスレッドの実行を停止します。
このメソッドはElemStkThreadStopおよびElemStkThreadFinalをコールバックしたあと、全てのスレッドが停止したかどうかをチェックし、全てのスレッドが停止したと判断した場合、ProcAfterLastStkThreadStopsをコールバックします。
指定したIDのStkThreadが存在しない場合、要求は無視されます。

Ids[入力]	停止するスレッドのIDを含む配列。複数のIDを指定することができます。
NumOfIds[入力]	Idsに含まれるID(要素)の数
[戻り値]	本メソッドを呼び出す前に実行していたスレッドの数

3.2.9 void SetStkThreadInterval(int TargetId, int Millisec)

StkThread.Libは、スレッド開始後定期的にElemStkThreadMainコールバックメソッドを呼び出します。
このメソッドはElemStkThreadMainコールバックメソッドを呼び出す間隔をミリ秒単位で指定します。

TargetId[入力]	間隔を指定するスレッドのID(StkThread ID)
Millisec[入力]	ElemStkThreadMainコールバックメソッドを呼び出す間隔(ミリ秒)
[戻り値]	なし

3.2.10 int GetStkThreadInterval(int TargetId)

このメソッドはElemStkThreadMainコールバックメソッドを呼び出す間隔を取得します。

TargetId[入力]	間隔を取得する対象のスレッドのID(StkThread ID)
[戻り値]	ElemStkThreadMainコールバックメソッドを呼び出す間隔(ミリ秒) -1: 指定したIDのStkThreadが存在しない場合

3.2.11 int GetStkThreadIdByIndex(int Index)

StkThreadはStkThread.Libの内部で配列の要素として管理されています。このメソッドは、配列のIndex番

目の要素であるStkThreadのIDを取得します。

Index[入力]	配列の要素番号(0からMAX_NUM_OF_STKTHREADS-1までの値)
[戻り値]	指定されたStkThreadのID -1: 指定したIDのStkThreadが存在しない場合

3.2.12 int GetStkThreadStatus(int TargetId)

指定したIDのスレッドの状態を取得します。

TargetId[入力]	StkThread ID (StkThreadが内部的に管理するスレッドのIDであり一般的なスレッドIDではない)
[戻り値]	StkThreadの状態。以下の値の何れかです。 STKTHREAD_STATUS_READY STKTHREAD_STATUS_STARTING STKTHREAD_STATUS_RUNNING STKTHREAD_STATUS_STOPPING -1: 指定したIDのStkThreadが存在しない場合

3.2.13 int GetStkThreadStatusByIndex(int Index)

StkThreadはStkThread.Libの内部で配列の要素として管理されています。このメソッドは、配列のIndex番目の要素であるStkThreadの状態を取得します。

Index[入力]	配列の要素番号(0からMAX_NUM_OF_STKTHREADS-1までの値)
[戻り値]	StkThreadの状態。以下の値の何れかです。 STKTHREAD_STATUS_READY STKTHREAD_STATUS_STARTING STKTHREAD_STATUS_RUNNING STKTHREAD_STATUS_STOPPING -1: 指定したIDのStkThreadが存在しない場合

3.2.14 int GetStkThreadName(int TargetId, TCHAR Name[MAX_LENGTH_OF_STKTHREAD_NAME])

指定したIDのスレッドの名前を取得します。

TargetId[入力]	StkThread ID (StkThreadが内部的に管理するスレッドのIDであり一般的なスレッドIDではない)
Name[出力]	指定したStkThreadの名前
[戻り値]	指定したIDのStkThreadが存在する場合0, そうでない場合-1が返ります。

3.2.15 int GetStkThreadNameByIndex(int Index, TCHAR Name[MAX_LENGTH_OF_STKTHREAD_NAME])

StkThreadはStkThread.Libの内部で配列の要素として管理されています。このメソッドは、配列のIndex番目の要素であるStkThreadの名前を取得します。

Index[入力]	配列の要素番号(0からMAX_NUM_OF_STKTHREADS-1までの値)
Name[出力]	指定したStkThreadの名前
[戻り値]	指定したIDのStkThreadが存在する場合0, そうでない場合-1が返ります。

3.2.16 int GetStkThreadDescription(int TargetId, TCHAR Description[MAX_LENGTH_OF_STKTHREAD_DESCRIPTION])

指定したIDのスレッドの詳細説明を取得します。

TargetId[入力]	StkThread ID (StkThreadが内部的に管理するスレッドのIDであり一般的なスレッドIDではない)
Description[出力]	指定したStkThreadの詳細説明

[戻り値]	指定したIDのStkThreadが存在する場合0, そうでない場合-1が返ります。
-------	---

3.2.17 int GetStkThreadDescriptionByIndex(int Index, TCHAR Description[MAX_LENGTH_OF_STKTHREAD_DESCRIPTION])

StkThreadはStkThread.Libの内部で配列の要素として管理されています。このメソッドは、配列のIndex番目の要素であるStkThreadの詳細説明を取得します。

Index[入力]	配列の要素番号(0からMAX_NUM_OF_STKTHREADS-1までの値)
Description[出力]	指定したStkThreadの詳細説明
[戻り値]	指定したIDのStkThreadが存在する場合0, そうでない場合-1が返ります。

3.3 ユーザープログラム用定数値

ユーザープログラムでは, stkthread.hに定義された下記の定数値を使用することができます。

定数	説明	定数値
MAX_NUM_OF_STKTHREADS	登録可能なスレッドの最大数	1,000
MAX_LENGTH_OF_STKTHREAD_NAME	スレッド名の最大文字数(TCHAR換算)	32
MAX_LENGTH_OF_STKTHREAD_DESCRIPTION	スレッドの説明の最大文字数(TCHAR換算)	256
STKTHREAD_STATUS_READY	スレッドの状態=待機中	100
STKTHREAD_STATUS_STARTING	スレッドの状態=開始中	101
STKTHREAD_STATUS_RUNNING	スレッドの状態=実行中	102
STKTHREAD_STATUS_STOPPING	スレッドの状態=停止中	104

5. その他

5.1 各管理リソースの上限値について

各管理リソースの上限値を以下に示します。

項目	上限値
登録可能なスレッドの数(登録可能なStkThreadの数)	1,000
スレッドの名前の最大文字長	32文字(TCHAR:終端NULL文字含む)
スレッドの詳細説明の最大文字長	256文字(TCHAR:終端NULL文字含む)