

# 動的プログラム呼び出し関数仕様書

## 名前(NAME)

GetProxyDynLang - 各動的言語のインターフェース取得

## 書式(SYNOPSIS)

```
#include <proxydynlang/IProxyDynLang.h>

IProxyDynLang* GetProxyDynLang()
```

## 説明(DESCRIPTION)

GetProxyDynLang()は、各動的プログラムの呼び出しのインターフェースを取得します。  
インターフェースとして次の関数が定義されている。

関数の呼び出し方法は、インターフェース名(IProxyDynLang)\_関数名()で、  
第一引数に、インターフェース構造体へのポインターを渡します。

init()

各動的プログラムの初期処理を行う

dynFileParse()

各動的プログラムファイルの解析を行う

funcall()

各動的プログラムファイル内の関数を実行する

classStaticMethodCall()

各動的プログラムファイル内スタティックメソッドを実行する

getInstance()

各動的プログラムファイル内のクラスインスタンスを生成する

classObjectMethodCall()

各動的プログラムファイル内のクラスメソッドを実行する

end()

各動的プログラムの終了処理を行う

## 戻り値(RETURN VALUE)

動的プログラムの呼び出しインターフェース構造体へのポインター

## 名前(NAME)

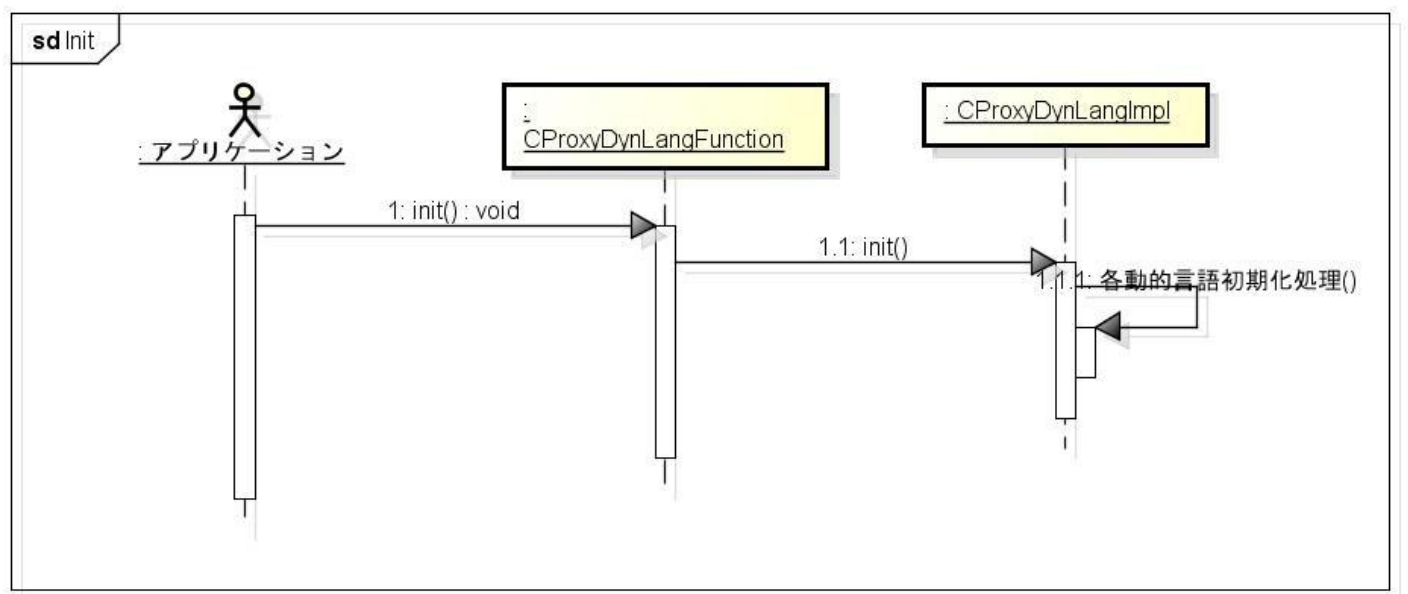
IProxyDynLang\_init 各動的プログラムの初期処理を行う

## 書式(SYNOPSIS)

```
#include <proxydynlang/IProxyDynLang.h>
void IProxyDynLang_init(IProxyDynLang* pThis)
```

## 説明(DESCRIPTION)

IProxyDynLang\_init()は、各動的プログラムの初期化処理を行う。



powered by Astah

## 戻り値(RETURN VALUE)

無し

## 名前(NAME)

IProxyDynLang\_dynFileParse - 各動的プログラムファイルの解析を行う

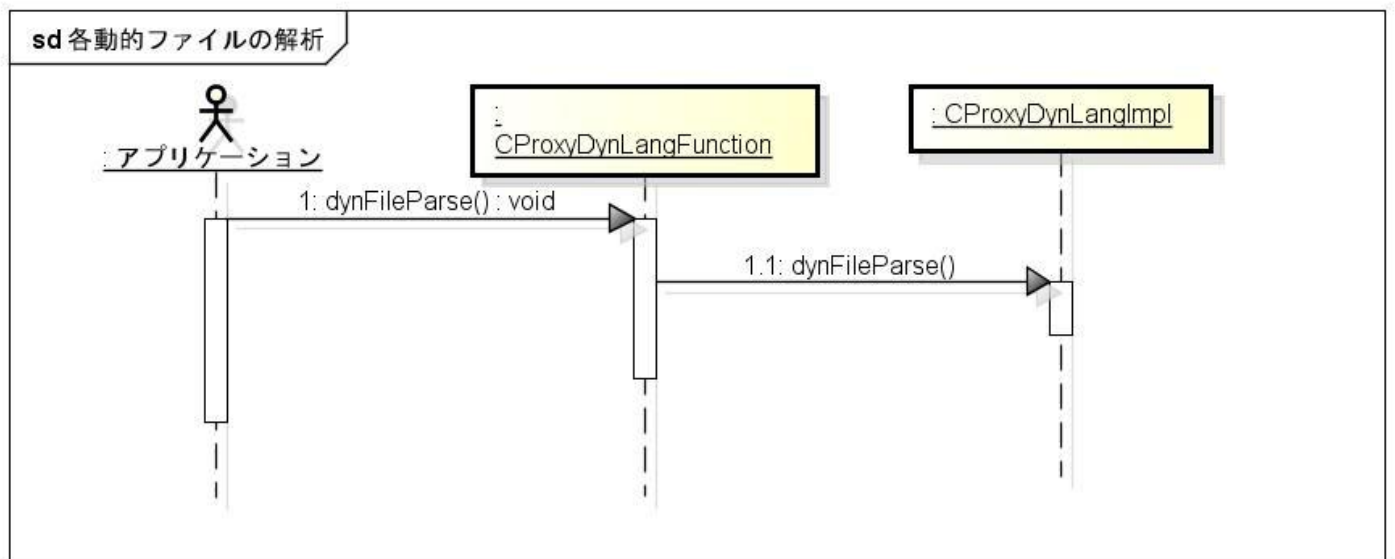
## 書式(SYNOPSIS)

```
#include <proxydynlang/IProxyDynLang.h>
```

```
void IProxyDynLang_dynFileParse(IProxyDynLang* pThis, char* pDynFileName)
```

## 説明(DESCRIPTION)

IProxyDynLang\_dynFileParse ()は、各動的プログラムの解析を行う。



powered by Astah

## 戻り値(RETURN VALUE)

無し

## 名前(NAME)

IProxyDynLang\_funcall - 各動的プログラム内の関数実行

## 書式(SYNOPSIS)

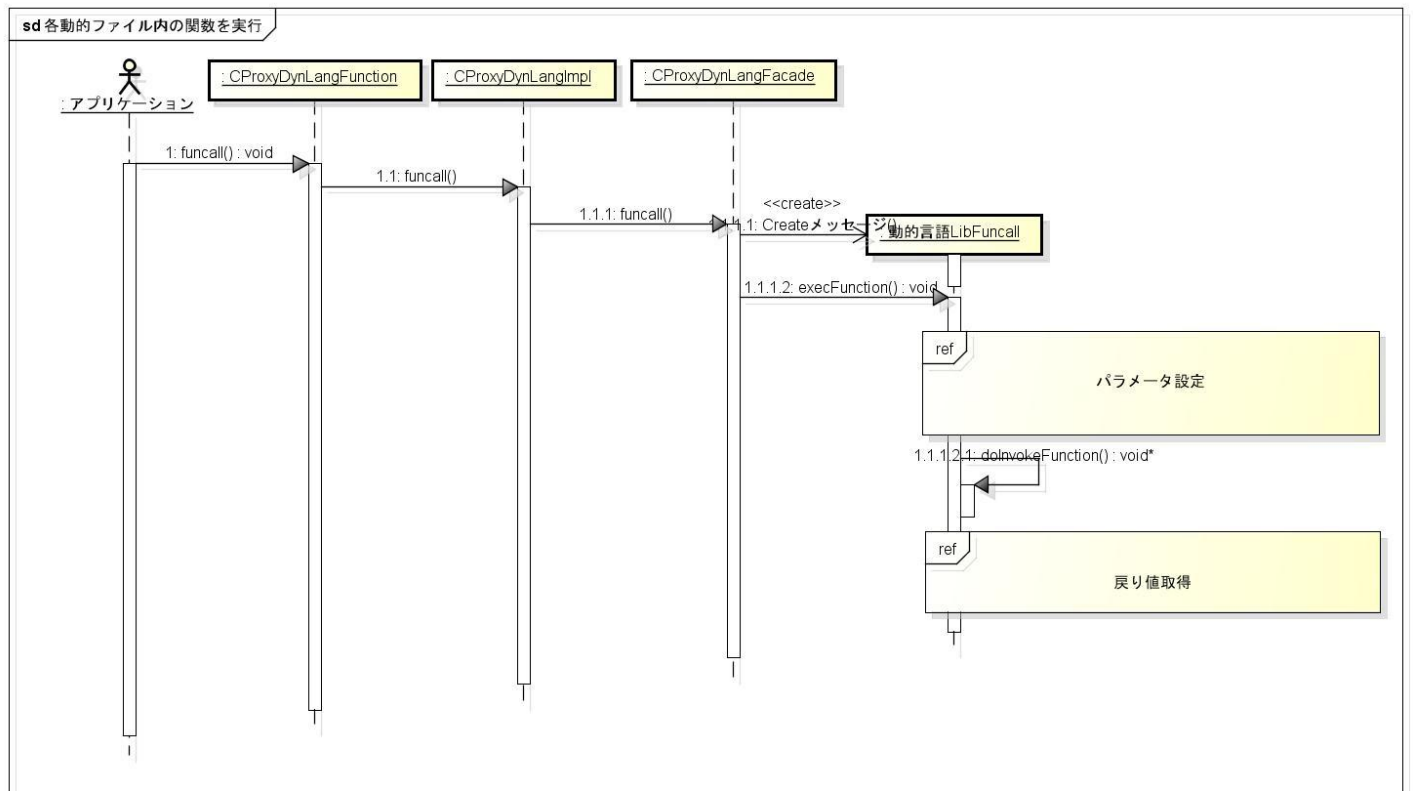
```
#include <proxydynlang/IProxyDynLang.h>
```

```
void IProxyDynLang_funcall(IProxyDynLang* pThis, DynLangReturn* pDynLangReturn, char*  
pFuncName, DynLangParams* pDynLangParams)
```

## 説明(DESCRIPTION)

IProxyDynLang\_funcall は、次の処理を実行する。

- pDynLangParams に従って、動的プログラム用に変換する(pDynLangParams != NULL)
- 各動的プログラム内の関数実行する
- 実行結果を動的プログラム用から pDynLangReturn に変換する



powered by Astah

## 戻り値(RETURN VALUE)

無し

## 名前(NAME)

IProxyDynLang\_classStaticMethodCall - 各動的プログラム内のクラススタティックメソッド実行

## 書式(SYNOPSIS)

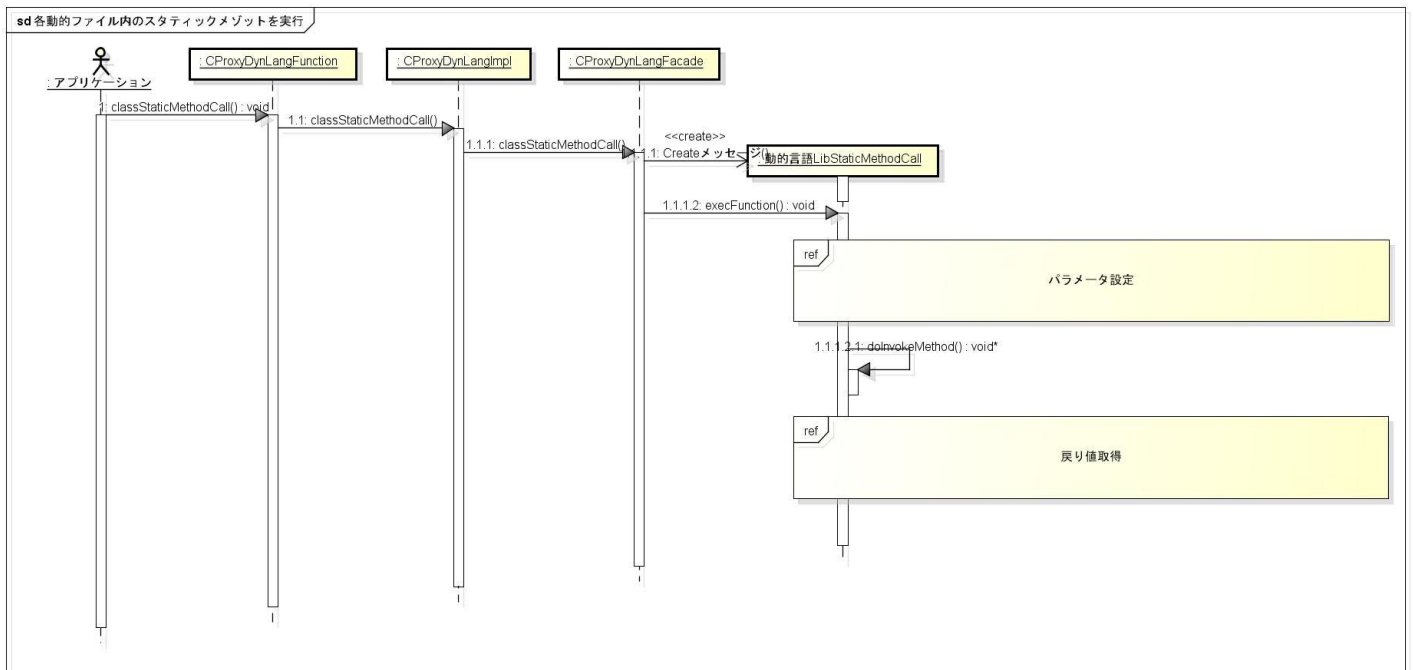
```
#include <proxydynlang/IProxyDynLang.h>
```

```
void IProxyDynLang_classStaticMethodCall(IProxyDynLang* pThis, DynLangReturn* pDynLangReturn,
char* pClassName, char* pFuncName, DynLangParams* pDynLangParams)
```

## 説明(DESCRIPTION)

IProxyDynLang\_classStaticMethodCall は、次の処理を実行する。

- pDynLangParams に従って、動的プログラム用に変換する(pDynLangParams != NULL)
- 各動的ファイル内のクラススタティックメソッド実行する
- 実行結果を動的プログラム用から pDynLangReturn に変換する



powered by Astah

## 戻り値(RETURN VALUE)

無し

## 名前(NAME)

IProxyDynLang\_getInstance -各動的ファイル内のクラスのインスタンス生成

## 書式(SYNOPSIS)

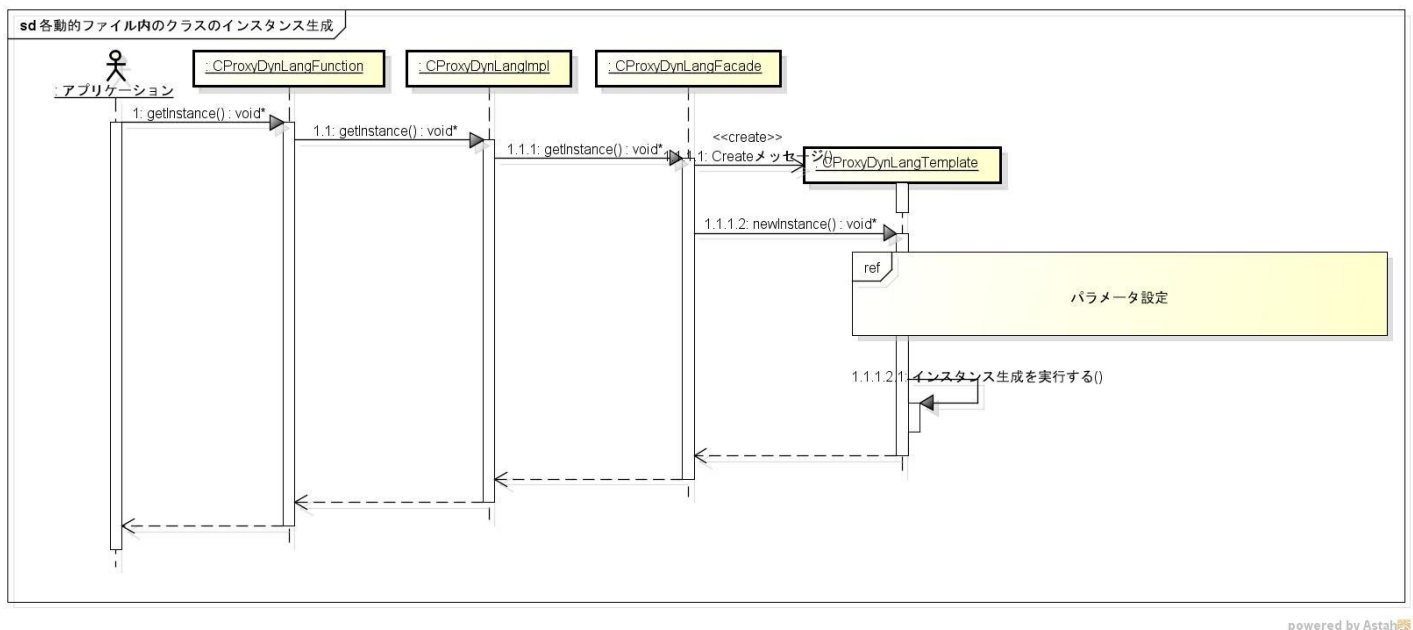
```
#include <proxydynlang/IProxyDynLang.h>
```

```
void* IProxyDynLang_getInstance(IProxyDynLang* pThis, char* pClassName, DynLangParams*  
pDynLangParams)
```

## 説明(DESCRIPTION)

IProxyDynLang\_getInstance は、次の処理を実行する。

- pDynLangParams に従って、動的プログラム用に変換する(pDynLangParams != NULL)
- 各動的ファイル内のクラスのインスタンス生成する



## 戻り値(RETURN VALUE)

生成したオブジェクト

## 名前(NAME)

IProxyDynLang\_classObjectMethodCall -オブジェクトに対するメソッド実行

## 書式(SYNOPSIS)

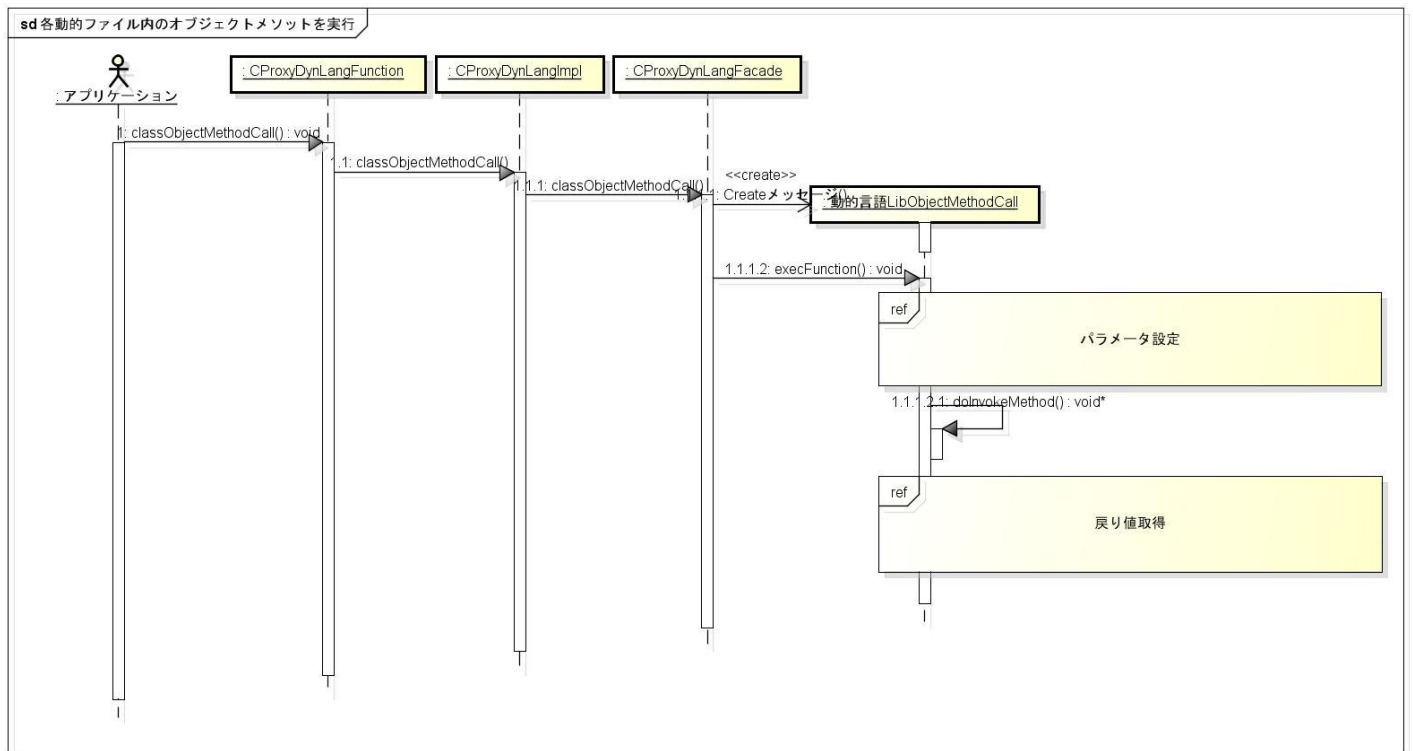
```
#include <proxydynlang/IProxyDynLang.h>
```

```
void IProxyDynLang_classObjectMethodCall(IProxyDynLang* pThis, DynLangReturn*
pDynLangReturn, void* pObj, char* pFuncName, DynLangParams* pDynLangParams)
```

## 説明(DESCRIPTION)

IProxyDynLang\_classObjectMethodCall は、次の処理を実行する。

- pDynLangParams に従って、動的プログラム用に変換する(pDynLangParams != NULL)
- オブジェクトに対するメソッドを実行する
- 実行結果を動的プログラム用から pDynLangReturn に変換する。



powered by Astah

## 戻り値(RETURN VALUE)

無し



## 名前(NAME)

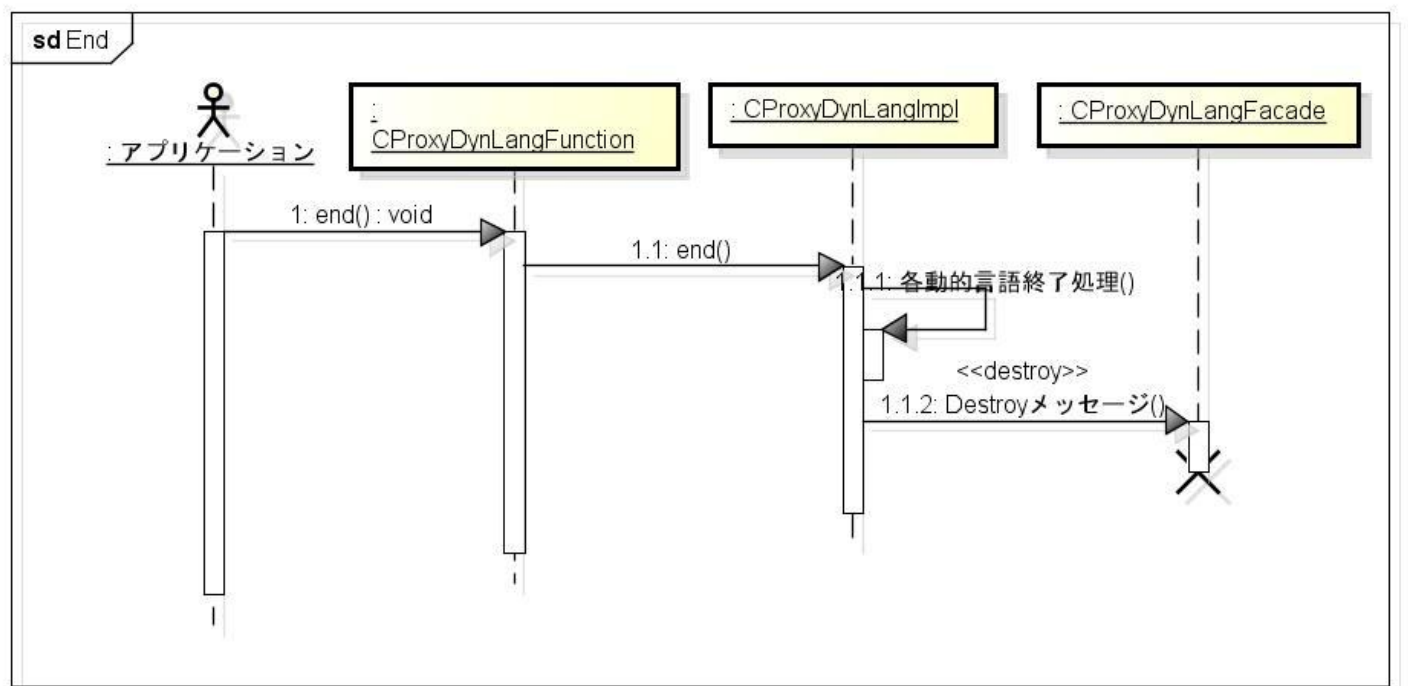
IProxyDynLang\_end -各動的言語の終了処理

## 書式(SYNOPSIS)

```
#include <proxydynlang/IProxyDynLang.h>
void IProxyDynLang_end(IProxyDynLang* pThis)
```

## 説明(DESCRIPTION)

IProxyDynLang\_end は、各動的言語の終了処理を行う。



powered by Astah

## 戻り値(RETURN VALUE)

無し

## 名前(NAME)

CProxyDynLangFunction\_ctor - CProxyDynLangFunction のコンストラクタ

## 書式(SYNOPSIS)

```
#include <proxydynlang/CProxyDynLangFunction.h>
void CProxyDynLangFunction_ctor(CProxyDynLangFunction* pThis, CProxyDynLangImpl* impl)
```

## 説明(DESCRIPTION)

CProxyDynLangFunction\_ctor ()は、impl を内部に保持する。

## 戻り値(RETURN VALUE)

無し

## 名前(NAME)

CProxyDynLangFunction\_dtor - CProxyDynLangFunction のデストラクタ

## 書式(SYNOPSIS)

```
#include <proxydynlang/CProxyDynLangFunction.h>
void CProxyDynLangFunction_dtor(CProxyDynLangFunction* pThis)
```

## 説明(DESCRIPTION)

CProxyDynLangFunction\_dtor ()は、CProxyDynLangFunction の解放を行う。

## 戻り値(RETURN VALUE)

無し

## 名前(NAME)

CProxyDynLangFunction\_init   各動的言語の初期化

## 書式(SYNOPSIS)

```
#include <proxydynlang/CProxyDynLangFunction.h>
void CProxyDynLangFunction_init(IProxyDynLang* pThis)
```

## 説明(DESCRIPTION)

CProxyDynLangFunction\_init ()は、CProxyDynLangImpl\_init 関数に委譲する。

## 戻り値(RETURN VALUE)

無し

## 名前(NAME)

CProxyDynLangFunction\_dynFileParse   -各動的ファイルのパーズ

## 書式(SYNOPSIS)

```
#include <proxydynlang/CProxyDynLangFunction.h>
void CProxyDynLangFunction_dynFileParse(IProxyDynLang* pThis, char* pDynFileName)
```

## 説明(DESCRIPTION)

CProxyDynLangFunction\_dynFileParse()は、CProxyDynLangImpl\_dynFileParse 関数に委譲する。

## 戻り値(RETURN VALUE)

無し

## 名前(NAME)

CProxyDynLangFunction\_funcall -各動的ファイル内の関数実行

## 書式(SYNOPSIS)

```
#include <proxydynlang/CProxyDynLangFunction.h>

void CProxyDynLangFunction_funcall(IProxyDynLang* pThis, DynLangReturn* pDynLangReturn,
char* pFuncName, DynLangParams* pDynLangParams)
```

## 説明(DESCRIPTION)

CProxyDynLangFunction\_funcall()は、CProxyDynLangImpl\_funcall 関数に委譲する。

## 戻り値(RETURN VALUE)

無し

## 名前(NAME)

CProxyDynLangFunction\_classStaticMethodCall -各動的ファイル内のクラススタティックメソッド実行

## 書式(SYNOPSIS)

```
#include <proxydynlang/CProxyDynLangFunction.h>

void CProxyDynLangFunction_classStaticMethodCall(IProxyDynLang* pThis, DynLangReturn*
pDynLangReturn, char* pClassName, char* pFuncName, DynLangParams* pDynLangParams)
```

## 説明(DESCRIPTION)

CProxyDynLangFunction\_classStaticMethodCall()は、CProxyDynLangImpl\_classStaticMethodCall 関数に委譲する。

## 戻り値(RETURN VALUE)

無し

## 名前(NAME)

CProxyDynLangFunction\_getInstance   -各動的ファイル内のクラスのインスタンス生成

## 書式(SYNOPSIS)

```
#include <proxydynlang/CProxyDynLangFunction.h>

void* CProxyDynLangFunction_getInstance(IProxyDynLang* pThis, char* pClassName,
DynLangParams* pDynLangParams)
```

## 説明(DESCRIPTION)

CProxyDynLangFunction\_classStaticMethodCall()は、CProxyDynLangImpl\_getInstance 関数に委譲する。

## 戻り値(RETURN VALUE)

生成したオブジェクト



## 名前(NAME)

CProxyDynLangFunction\_classObjectMethodCall 各オブジェクトに対するメソッドの呼び出し

## 書式(SYNOPSIS)

```
#include <proxydynlang/CProxyDynLangFunction.h>

void CProxyDynLangFunction_classObjectMethodCall(IProxyDynLang* pThis, DynLangReturn*
pDynLangReturn, void* pObj, char* pFuncName, DynLangParams* pDynLangParams)
```

## 説明(DESCRIPTION)

CProxyDynLangFunction\_classObjectMethodCall()は、CProxyDynLangImpl\_classObjectMethodCall 関数に委譲する。

## 戻り値(RETURN VALUE)

無し

## 名前(NAME)

CProxyDynLangFunction\_end   各動的言語の終了処理

## 書式(SYNOPSIS)

```
#include <proxydynlang/CProxyDynLangFunction.h>
void CProxyDynLangFunction_end(IProxyDynLang* pThis)
```

## 説明(DESCRIPTION)

CProxyDynLangFunction\_end()は、CProxyDynLangImpl\_end 関数に委譲し、CProxyDynLangFunction\_dtor()を呼び出す。

## 戻り値(RETURN VALUE)

無し

## 名前(NAME)

GetProxyDynLangImpl -各動的言語の呼び出しインターフェースを取得

## 書式(SYNOPSIS)

```
#include <proxydynlang/GetProxyDynLangImpl.h>
CProxyDynLangImpl* GetProxyDynLangImpl()
```

## 説明(DESCRIPTION)

GetProxyDynLangImpl ()は、各動的言語の呼び出しのインターフェースを取得します。  
インターフェースとして次の関数が定義されている。

関数の呼び出し方法は、クラス名(CProxyDynLangImpl)\_関数名()で、  
第一引数に、インターフェース構造体へのポインターを渡します。

init()

各動的言語の初期処理を行う

dynFileParse()

各動的言語ファイルのコンパイルを行う

funcall()

各動的言語ファイル内の関数を実行する

getInstance()

各動的言語ファイル内のクラスインスタンスを生成する

classObjectMethodCall()

各動的言語ファイル内のクラスメソッドを実行する

classStaticMethodCall()

各動的言語ファイル内スタティックメソッドを実行する

end()

各動的言語の終了処理を行う

## 戻り値(RETURN VALUE)

動的言語の呼び出しインターフェース構造体へのポインター

## 名前(NAME)

CProxyDynLangImpl\_init -各動的言語の初期化

## 書式(SYNOPSIS)

```
#include <proxydynlang/CProxyDynLangImpl.h>
void CProxyDynLangImpl_init(CProxyDynLangImpl* pThis)
```

## 説明(DESCRIPTION)

CProxyDynLangImpl\_init ()は、各動的プログラムの初期化を行う

## 戻り値(RETURN VALUE)

無し

## 名前(NAME)

CProxyDynLangImpl\_dynFileParse   -動的ファイルのパーズ

## 書式(SYNOPSIS)

```
#include <proxydynlang/CProxyDynLangImpl.h>
```

```
void CProxyDynLangImpl_dynFileParse(CProxyDynLangImpl* pThis, char* pDynFileName)
```

## 説明(DESCRIPTION)

CProxyDynLangImpl\_dynFileParse ()は、 IProxyDynLang\_dynFileParse 関数を参照

## 戻り値(RETURN VALUE)

無し

## 名前(NAME)

CProxyDynLangImpl\_funcall -各動的ファイル内の関数実行

## 書式(SYNOPSIS)

```
#include <proxydynlang/CProxyDynLangImpl.h>
```

```
void CProxyDynLangImpl_funcall(CProxyDynLangImpl* pThis, DynLangReturn* pDynLangReturn,  
char* pFuncName, DynLangParams* pDynLangParams)
```

## 説明(DESCRIPTION)

CProxyDynLangImpl\_funcall ()は、 IProxyDynLang\_funcall 関数を参照

## 戻り値(RETURN VALUE)

無し

## 名前(NAME)

CProxyDynLangImpl\_classStaticMethodCall   各動的ファイル内のクラススタティックメソッド実行

## 書式(SYNOPSIS)

```
#include <proxydynlang/CProxyDynLangImpl.h>

void CProxyDynLangImpl_classStaticMethodCall(CProxyDynLangImpl* pThis, DynLangReturn*
pDynLangReturn, char* pClassName, char* pFuncName, DynLangParams* pDynLangParams)
```

## 説明(DESCRIPTION)

CProxyDynLangImpl\_classStaticMethodCall ()は、IProxyDynLang\_classStaticMethodCall 関数を参照

## 戻り値(RETURN VALUE)

無し

## 名前(NAME)

CProxyDynLangImpl\_getInstance   -各動的ファイル内のクラスのインスタンス生成

## 書式(SYNOPSIS)

```
#include <proxydynlang/CProxyDynLangImpl.h>
```

```
void* CProxyDynLangImpl_getInstance(CProxyDynLangImpl* pThis, char* pClassName,  
DynLangParams* pDynLangParams)
```

## 説明(DESCRIPTION)

CProxyDynLangImpl\_getInstance ()は、 IProxyDynLang\_getInstance 関数を参照

## 戻り値(RETURN VALUE)

生成したオブジェクト



## 名前(NAME)

CProxyDynLangImpl\_classObjectMethodCall -オブジェクトに対するメソットの呼び出し

## 書式(SYNOPSIS)

```
#include <proxydynlang/CProxyDynLangImpl.h>

void CProxyDynLangImpl_classObjectMethodCall(CProxyDynLangImpl* pThis, DynLangReturn*
pDynLangReturn, void* pObj, char* pFuncName, DynLangParams* pDynLangParams)
```

## 説明(DESCRIPTION)

CProxyDynLangImpl\_classObjectMethodCall ()は、IProxyDynLang\_classObjectMethodCall 関数を参照

## 戻り値(RETURN VALUE)

無し

## 名前(NAME)

CProxyDynLangImpl\_end -各動的言語の終了処理

## 書式(SYNOPSIS)

```
#include <proxydynlang/CProxyDynLangImpl.h>
void CProxyDynLangImpl_end(CProxyDynLangImpl* pThis)
```

## 説明(DESCRIPTION)

CProxyDynLangImpl\_end()は、各動的プログラムの終了処理を行い、CProxyDynLangImpl の解放をする

## 戻り値(RETURN VALUE)

無し

## 名前(NAME)

GetProxyDynLangThread -各動的プログラムのスレッドインターフェース取得

## 書式(SYNOPSIS)

```
#include <proxydynlang/ IProxyDynLangThread>

IProxyDynLangThread* GetProxyDynLangThread ()
```

## 説明(DESCRIPTION)

GetProxyDynLangThread() は、各動的プログラムのスレッドインターフェースを取得する。インターフェースとして、次の関数が定義されている。

関数の呼び出し方法は、インターフェース名(IProxyDynLangThread)\_関数名()で、第一引数に、インターフェース構造体へのポインターを渡します。

### Create()

スレッドを生成する

### Creates()

複数スレッドを生成する

### Join()

スレッド終了待ちをする

### Joins()

複数スレッド終了待ちをする

### ThreadFuncParamFactory()

ThreadFuncParam 構造体を生成する

## 戻り値(RETURN VALUE)

各動的プログラムのスレッドインターフェースへのポインター

## 名前(NAME)

IProxyDynLangThread\_create – スレッドの生成を行う

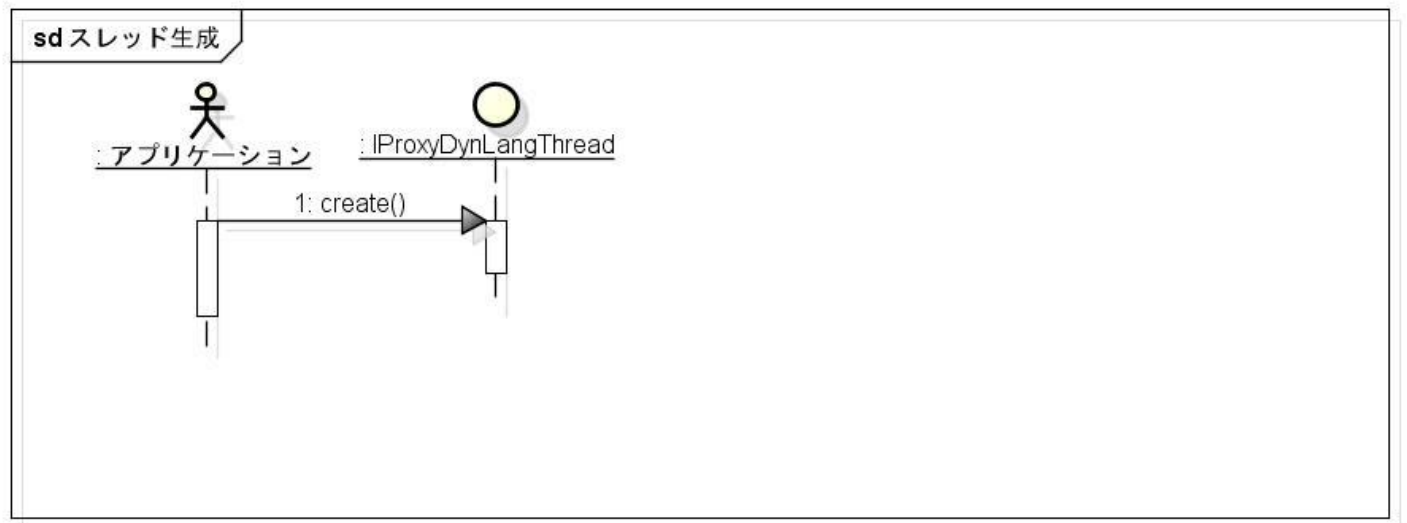
## 書式(SYNOPSIS)

```
#include <proxydynlang/ IProxyDynLangThread.h>
```

```
Long IProxyDynLangThread_create(IProxyDynLangThread* pThis, ThreadFunc* threadFunc)
```

## 説明(DESCRIPTION)

IProxyDynLangThread\_create0は、新しいスレッドを生成する



powered by Astah

## 戻り値(RETURN VALUE)

スレッド ID

## 名前(NAME)

IProxyDynLangThread\_creates – 複数スレッドの生成を行う

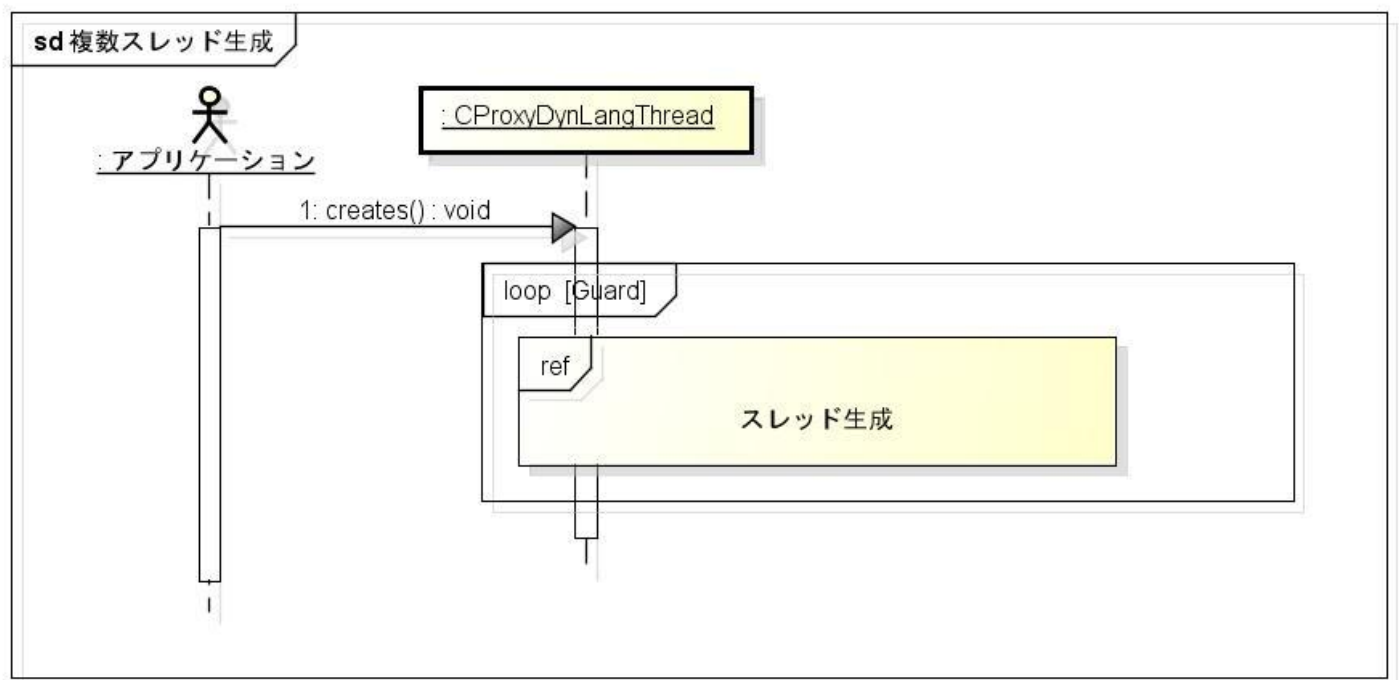
## 書式(SYNOPSIS)

```
#include <proxydynlang/ IProxyDynLangThread.h>
```

```
Void IProxyDynLangThread_creates(IProxyDynLangThread* pThis, ThreadFuncs* threadFuncs)
```

## 説明(DESCRIPTION)

IProxyDynLangThread\_creates ()は、複数の新しいスレッドを生成する



powered by Astah

## 戻り値(RETURN VALUE)

無し

## 名前(NAME)

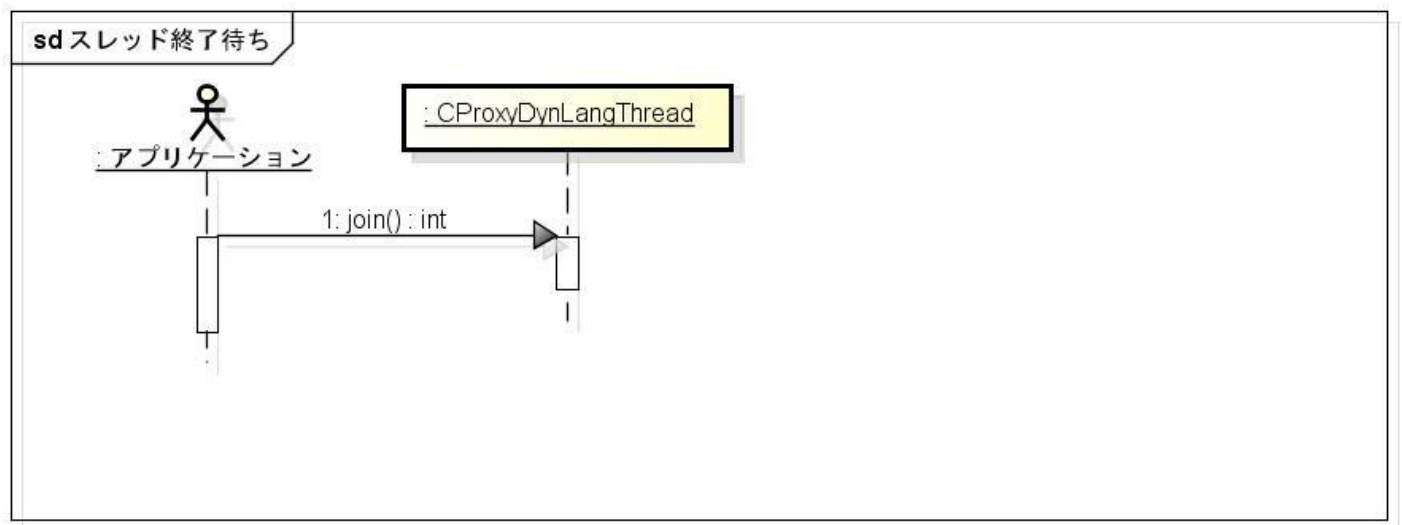
IProxyDynLangThread\_join – スレッド終了待ちをする

## 書式(SYNOPSIS)

```
#include <proxydynlang/ IProxyDynLangThread.h>
int IProxyDynLangThread_join(IProxyDynLangThread* pThis, long thid)
```

## 説明(DESCRIPTION)

IProxyDynLangThread\_join()は、thid で指定したスレッドの終了する待ちをする



powered by Astah

## 戻り値(RETURN VALUE)

終了ステータス

## 名前(NAME)

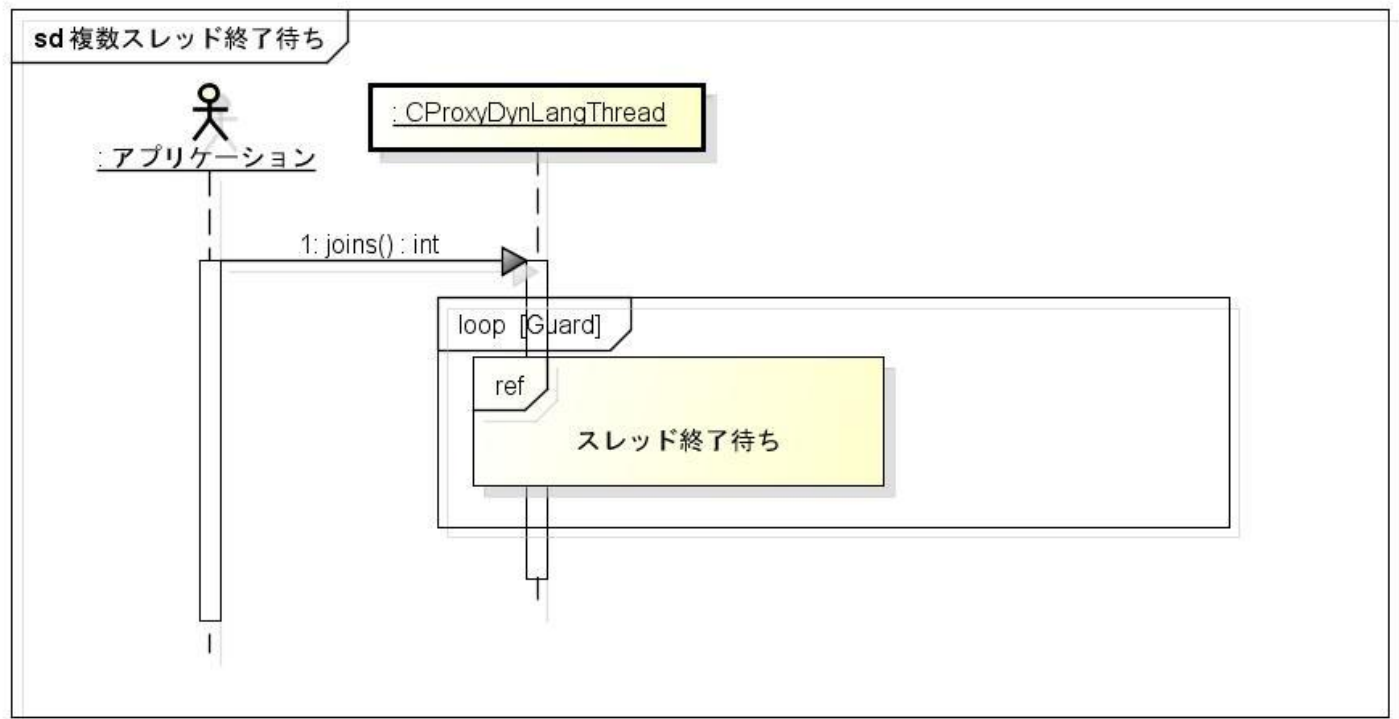
IProxyDynLangThread\_joins – 複数スレッドの終了待ちをする

## 書式(SYNOPSIS)

```
#include <proxydynlang/ IProxyDynLangThread.h>
int IProxyDynLangThread_joins (IProxyDynLangThread* pThis)
```

## 説明(DESCRIPTION)

IProxyDynLangThread\_joins()は、複数スレッドの終了待ちをする



powered by Astah

## 戻り値(RETURN VALUE)

終了ステータス

## 名前(NAME)

IProxyDynLangThread\_ThreadFuncParamFactory – ThreadFuncParam 構造体を生成する

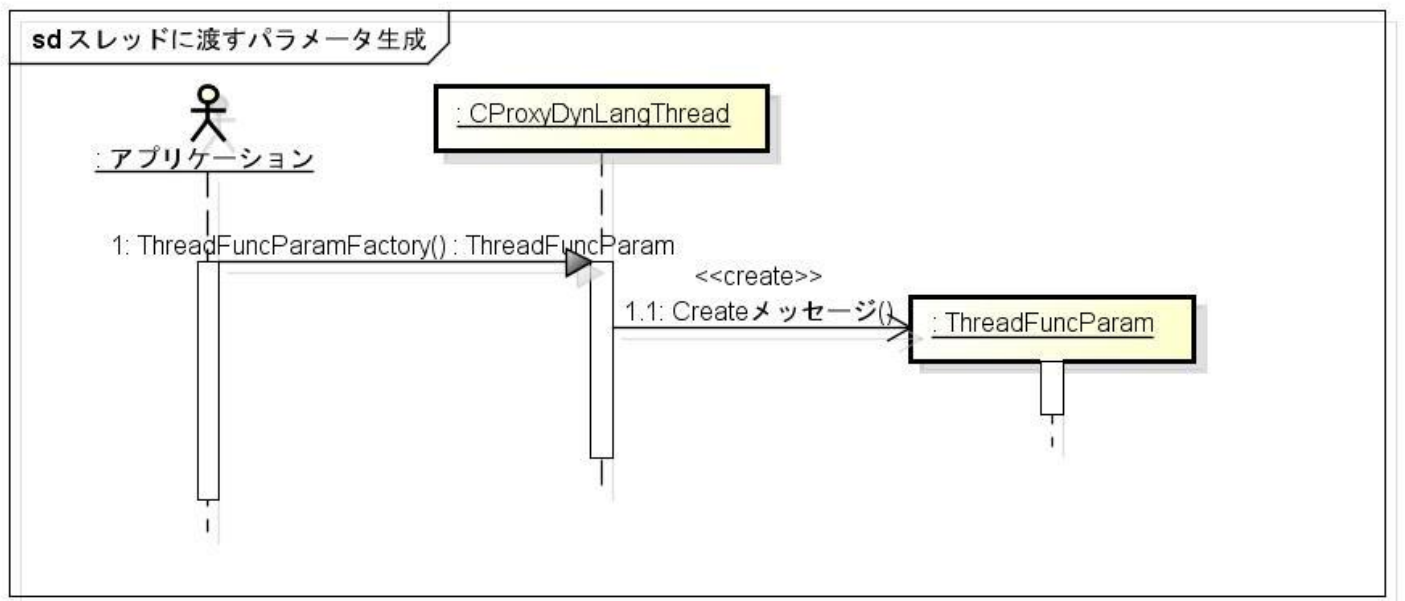
## 書式(SYNOPSIS)

```
#include <proxydynlang/ IProxyDynLangThread.h>
```

```
ThreadFuncParam* IProxyDynLangThread_ThreadFuncParamFactory(IProxyDynLangThread* pThis)
```

## 説明(DESCRIPTION)

IProxyDynLangThread\_ThreadFuncParamFactory ()は、ThreadFuncParam 構造体を生成する



powered by Astah

## 戻り値(RETURN VALUE)

ThreadFuncParam 構造体へのポインター



## 名前(NAME)

ThreadFuncParam\_ctor – ThreadFuncParam コンストラクタ

## 書式(SYNOPSIS)

```
#include <proxydynlang/ ThreadFuncParam.h>

void ThreadFuncParam_ctor(ThreadFuncParam* pThis, IProxyDynLang* proxyDynLang)
```

## 説明(DESCRIPTION)

ThreadFuncParam\_ctor()は、ThreadFuncParam 型の初期化を行う

## 戻り値(RETURN VALUE)

無し

## 名前(NAME)

ThreadFuncParam\_dtor – ThreadFuncParam デストラクタ

## 書式(SYNOPSIS)

```
#include <proxydynlang/ ThreadFuncParam.h>
void ThreadFuncParam_dtor(ThreadFuncParam* pThis)
```

## 説明(DESCRIPTION)

ThreadFuncParam\_dtor()は、ThreadFuncParam 型の解放を行う

## 戻り値(RETURN VALUE)

無し

## 名前(NAME)

ThreadFuncParam\_exitThread    呼び出しスレッドを終了する

## 書式(SYNOPSIS)

```
#include <proxydynlang/ ThreadFuncParam.h>
```

```
void ThreadFuncParam_exitThread(ThreadFuncParam* pThis, void* retval)
```

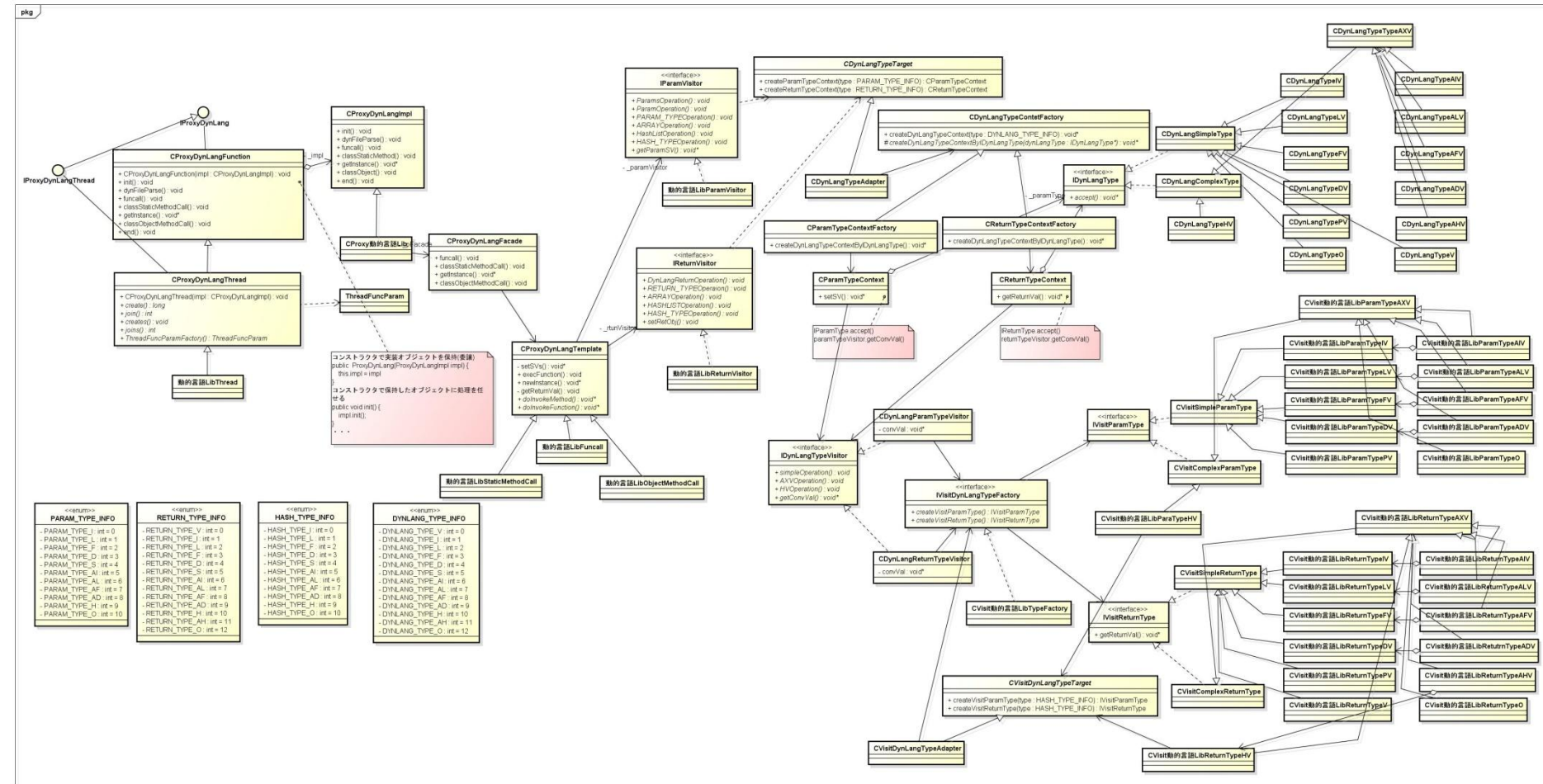
## 説明(DESCRIPTION)

ThreadFuncParam\_exitThread()は、呼び出しスレッドを終了する

## 戻り値(RETURN VALUE)

無し

クラス図



パラメータ構造体

```
typedef struct _DynLangParams      DynLangParams;
struct _DynLangParams {
    int      argc;                // DynLangParam の個数
    DynLangParam* argv; // 引数情報
};
```

```
typedef struct _DynLangParam      DynLangParam;
struct _DynLangParam {
    PARAM_TYPE_INFO param_type;
    void* param_value;
};
```

PARAM_TYPE_INFO	型名	param_value 値
PARAM_TYPE_I	Int 型	Int 型のポインター
PARAM_TYPE_L	Long 型	Long 型のポインター
PARAM_TYPE_F	Float 型	Float 型のポインター
PARAM_TYPE_D	Double 型	Double 型のポインター
PARAM_TYPE_S	文字列型	Char 型のポインター
PARAM_TYPE_AI	Int 型の配列	ARRAY_TYPE 型のポインター
PARAM_TYPE_AL	Long 型の配列	ARRAY_TYPE 型のポインター
PARAM_TYPE_AF	Float 型の配列	ARRAY_TYPE 型のポインター
PARAM_TYPE_AD	Double 型の配列	ARRAY_TYPE 型のポインター
PARAM_TYPE_H	Hash 型	HASHLIST_TYPE 型のポインター
PARAM_TYPE_O	Object 型	IProxyDynLang_getInstance()関数の戻り値

ARRAY\_TYPE 構造体

```
typedef struct _Array_Type      ARRAY_TYPE;
struct _Array_Type {
    int len;
    void* value;
};
```

PARAM_TYPE_INFO HASH_TYPE_INFO RETURN_TYPE_INFO	Value 値
PARAM_TYPE_AI HASH_TYPE_AI RETURN_TYPE_AI	Int 型のポインター
PARAM_TYPE_AL HASH_TYPE_AL RETURN_TYPE_AL	Long 型のポインター
PARAM_TYPE_AF HASH_TYPE_AF RETURN_TYPE_AF	Float 型のポインター
PARAM_TYPE_AD HASH_TYPE_AD RETURN_TYPE_AD	Double 型のポインター
RETURN_TYPE_AH	HASHLIST_TYPE 型のポインター

HASHLIST\_TYPE 構造体

```
typedef struct _HashList_Type    HASHLIST_TYPE;
struct _HashList_Type {
    HASHLIST_TYPE* next;
    char* key;                // ハッシュキー値
    HASH_TYPE_INFO value_type;
    void* value;
};
```

HASH_TYPE_INFO	型名	Value 値
HASH_TYPE_I	Int 型	Int 型のポインター
HASH_TYPE_L	Long 型	Long 型のポインター
HASH_TYPE_F	Float 型	Float 型のポインター
HASH_TYPE_D	Double 型	Double 型のポインター
HASH_TYPE_S	文字列型	文字列
HASH_TYPE_AI	Int 型の配列	ARRAY_TYPE 型のポインター
HASH_TYPE_AL	Long 型の配列	ARRAY_TYPE 型のポインター
HASH_TYPE_AF	Float 型の配列	ARRAY_TYPE 型のポインター
HASH_TYPE_AD	Double 型の配	ARRAY_TYPE 型のポインター
HASH_TYPE_H	ハッシュ型	HASHLIST_TYPE 型のポインター
HASH_TYPE_O	Object 型	IProxyDynLang_getInstance()関数の戻り値

戻り構造体

```
typedef struct _DynLangReturn    DynLangReturn;
struct _DynLangReturn {
    RETURN_TYPE_INFO return_type;
    void* return_value;
};
```

RETURN_TYPE_INFO	型名	return_value 値
RETURN_TYPE_V	戻り値無し	
RETURN_TYPE_I	Int 型	Int 型のポインター
RETURN_TYPE_L	Long 型	Long 型のポインター
RETURN_TYPE_F	Float 型	Float 型のポインター
RETURN_TYPE_D	Double 型	Doule 型のポインター
RETURN_TYPE_S	文字列型	文字列型
RETURN_TYPE_AI	Int 型の配列	ARRAY_TYPE 型のポインター
RETURN_TYPE_AL	Long 型の配列	ARRAY_TYPE 型のポインター
RETURN_TYPE_AF	Float 型の配列	ARRAY_TYPE 型のポインター
RETURN_TYPE_AD	Double 型の配列	ARRAY_TYPE 型のポインター
RETURN_TYPE_H	ハッシュ型	HASHLIST_TYPE 型のポインター
RETURN_TYPE_AH	ハッシュ型の配列	ARRAY_TYPE 型のポインター
RETURN_TYPE_O	オブジェクト型	IProxyDynLang_getInstance0関数の戻り値



ThreadFuncs 構造体

```
typedef struct _ThreadFuncs          ThreadFuncs
struct _ThreadFuncs
{
    int num;
    ThreadFunc* threadFunc;
}
```

```
typedef struct _ThreadFunc          ThreadFunc;
struct _ThreadFunc
{
    void (*func)(void* arg);
    void* arg;
}
```

ThreadFuncParam 構造体

```
typedef struct _ThreadFuncParam      ThreadFuncParam;
struct _ThreadFuncParam
{
    IProxyDynLang* proxyDynLang;
    void (*FP_thread_exit)(ThreadFuncParam* pThis, void* retval);
};
```