

HewTargetServer

ユーザーズマニュアル

本資料に記載の全ての情報は本資料発行時点のものであり、ルネサス エレクトロニクスは、予告なしに、本資料に記載した製品または仕様を変更することがあります。
ルネサス エレクトロニクスのホームページなどにより公開される最新情報をご確認ください。

ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
3. 当社製品を改造、改変、複製等しないでください。
4. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器の設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事事業の目的で使用しないでください。また、当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器に使用することができません。
6. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されていない用途に当社製品を使用したことによりお客様または第三者に生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。

標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット

高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）

特定水準： 航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害などを生じさせないようお客様の責任において冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、詳細につきましては製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

注 1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社とその総株主の議決権の過半数を直接または間接に保有する会社をいいます。

注 2. 本資料において使用されている「当社製品」とは、注 1 において定義された当社の開発、製造製品をいいます。

はじめに

HEW TargetServer (COM) は、ルネサス統合開発環境 High-performance Embedded Workshop (以下 High-performance Embedded Workshop) の機能を拡張するための機能です。市販の Windows アプリケーション開発ツールを利用して、High-performance Embedded Workshop のカスタマイズや他のアプリケーションとの連携を実現することができます。

本ユーザーズマニュアルでは、HEW TargetServer (COM) を使用するための基本的な情報を掲載しています。利用する Windows アプリケーション開発ツールの言語仕様、操作方法については、それぞれの製品マニュアルやオンラインヘルプをご参照ください。

登録商標

Microsoft, Visual Basic, Visual C++, Windows, Windows Vista は米国 Microsoft 社の米国およびその他の国における商標または登録商標です。

その他、記載されている製品名は各社の商標または登録商標です。

目次

1. 概要	1
1.1 利用する開発ツール	1
1.2 呼び出し可能なメソッド	1
2. HEW TargetServer (COM) 使用の準備	2
2.1 HEW TargetServer (COM) の登録	2
2.1.1 <i>EcHewTargetSever.dll</i> の登録	2
2.1.2 <i>HewTargetSever.exe</i> のレジストリへの登録	3
3. HEW TargetServer (COM) の使用方法	4
3.1 サンプルプログラム	4
3.2 プログラムの作成方法 (Visual C++ 6.0)	4
3.2.1 プロジェクトの生成	4
3.2.2 ボタンの作成	5
3.2.3 ソースコードの作成	5
3.3 プログラムの作成方法 (Visual C++ 2005)	11
3.3.1 プロジェクトの生成	11
3.3.2 ボタンの作成	12
3.3.3 ソースコードの作成	12
3.4 プログラムの作成方法 (Visual Basic 6.0)	15
3.4.1 プロジェクトの生成	15
3.4.2 タイプライブラリの指定	15
3.4.3 オブジェクトの生成	15
3.4.4 メソッドの呼び出し	16
3.5 プログラムの作成方法 (Visual Basic 2005)	18
3.5.1 プロジェクトの生成	18
3.5.2 タイプライブラリの指定	18
3.5.3 オブジェクトの生成	18
3.5.4 メソッドの呼び出し	19
3.6 Visual Basic 6.0からVisual Basic .NETに移行する場合の注意点	21
4. High-performance Embedded Workshopからのイベント取得	22
4.1 Visual C++のイベント取得	22
4.2 Visual Basicのイベント取得	25
5. メソッドの一覧	27
5.1 メソッド概要 (VC++のみ使用可)	27
5.1.1 実行制御	27
5.1.2 レジスタ操作	27
5.1.3 メモリ操作	28
5.1.4 ソフトウェアブレーク	28
5.1.5 変数ブレーク	28
5.1.6 変数トレース	28
5.1.7 割り込み	29
5.1.8 シンボル操作	29
5.1.9 ダウンロード	29
5.1.10 起動終了	29
5.1.11 ワークスペース	29
5.1.12 構成とセッション	30
5.1.13 プロジェクト	30
5.1.14 ビルド関係	30
5.1.15 ファイル関係	30
5.1.16 カバレッジ関係	30
5.1.17 その他	31
5.2 メソッド概要 (VB、VC++用)	32
5.2.1 実行制御	32
5.2.2 レジスタ操作	32
5.2.3 メモリ操作	32

5.2.4	ソフトウェアブレイク	33
5.2.5	変数ブレイク	33
5.2.6	変数トレース	33
5.2.7	割り込み	33
5.2.8	シンボル操作	34
5.2.9	ダウンロード	34
5.2.10	起動/終了	34
5.2.11	ワークスペース	34
5.2.12	構成とセッション	35
5.2.13	プロジェクト	35
5.2.14	ビルド関係	36
5.2.15	ファイル関係	37
5.2.16	カバレッジ関係	37
5.2.17	その他	37
5.3	メソッドの詳細 (VC++のみ使用可)	38
5.3.1	実行制御	38
	<i>GoTargetExec</i>	38
	<i>StopTargetExec</i>	39
	<i>ResetTargetExec</i>	40
	<i>InitializeTarget</i>	41
	<i>Step</i>	42
	<i>StepRate</i>	43
	<i>StepOver</i>	44
	<i>StepOut</i>	45
	<i>IsRunning</i>	46
5.3.2	レジスタ操作	47
	<i>GetPC</i>	47
	<i>SetPCAddress</i>	48
	<i>SetPCSource</i>	49
	<i>TestSetPC</i>	50
5.3.3	メモリ操作	51
	<i>GetMemory</i>	51
	<i>SetMemory</i>	52
	<i>GetDirectMemory</i>	53
5.3.4	ソフトウェアブレイク	54
	<i>SetPCBreakPt</i>	54
	<i>EnableBreakPt</i>	55
	<i>DeleteBreakPt</i>	56
	<i>GetAllBreakPt</i>	57
	<i>DeleteAllBreakPt</i>	58
5.3.5	変数ブレイク	59
	<i>SetDataBreakpoint</i>	59
	<i>EnableDataBreakpoint</i>	60
	<i>DeleteDataBreakpoint</i>	61
5.3.6	変数トレース	62
	<i>SetSymbolTrace</i>	62
	<i>ExecuteSymbolTrace</i>	63
	<i>DeleteSymbolTrace</i>	64
	<i>SaveSymbolTraceData</i>	65
5.3.7	割り込み	66
	<i>SendTrigger</i>	66
5.3.8	シンボル操作	67
	<i>GetRealTimeWatch</i>	67
	<i>GetQuickWatch</i>	68
	<i>SymbolToAddress</i>	69
	<i>AddressToSymbol</i>	70
	<i>GetLineFromAddr</i>	71
	<i>GetAddrFromLine</i>	72
5.3.9	ダウンロード	73
	<i>Download</i>	73
	<i>Unload</i>	74
5.3.10	起動/終了	75

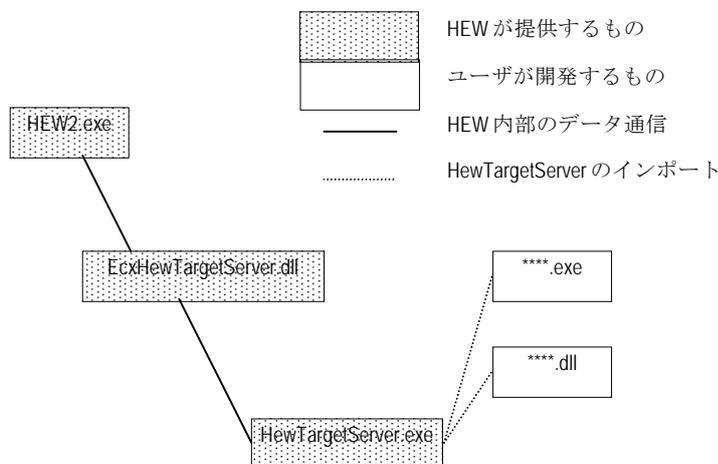
<i>InvokeHew</i>	75
<i>QuitHew</i>	76
5.3.11 ワークスペース	77
<i>Open Workspace</i>	77
<i>Close Workspace</i>	78
<i>Save Workspace</i>	79
5.3.12 構成とセッション	80
<i>SaveSession</i>	80
<i>GetCurrentConfiguration</i>	81
<i>SetCurrentConfiguration</i>	82
<i>GetConfigurations</i>	83
<i>GetCurrentSession</i>	84
<i>SetCurrentSession</i>	85
<i>GetSessions</i>	86
<i>GetCurrentProject</i>	87
<i>SetCurrentProject</i>	88
<i>GetProjects</i>	89
5.3.13 プロジェクト	90
<i>AddFile</i>	90
<i>AddFiles</i>	91
<i>DeleteFile</i>	92
<i>DeleteFiles</i>	93
5.3.14 ビルド関係	94
<i>BuildProject</i>	94
<i>RebuildProject</i>	95
<i>UpDateAllDependency</i>	96
<i>AddFileWithCompilerOption</i>	97
5.3.15 ファイル関係	98
<i>OpenFileAtLine</i>	98
<i>GetSourceFiles</i>	99
<i>GetDownloadModules</i>	100
<i>GetDependentFiles</i>	101
5.3.16 カバレッジ関係	102
<i>SetCoverageRange</i>	102
<i>GetCoverageRange</i>	103
<i>SetCoverageDisable</i>	104
<i>SetCoverageEnable</i>	105
<i>ClearCoverage</i>	106
<i>GetCoverageStatus</i>	107
<i>LoadCoverage</i>	108
<i>SaveCoverage</i>	109
5.3.17 その他	110
<i>GetErrorString</i>	110
<i>GetHewStatus</i>	111
<i>GetHewStatusEx</i>	112
<i>GetTargetName</i>	113
5.4 メソッドの詳細 (VB、VC++用)	114
5.4.1 実行制御	114
<i>GoTargetExec2</i>	114
<i>StopTargetExec2</i>	115
<i>ResetTargetExec2</i>	116
<i>InitializeTarget2</i>	117
<i>Step2</i>	118
<i>StepRate2</i>	119
<i>StepOver2</i>	120
<i>StepOut2</i>	121
<i>IsRunning2</i>	122
5.4.2 レジスタ操作	123
<i>GetPC2</i>	123
<i>SetPCAddress2</i>	124
<i>SetPCSource2</i>	125
<i>TestSetPC2</i>	126
5.4.3 メモリ操作	127

<i>GetMemory2</i>	127
<i>SetMemory2</i>	128
<i>GetDirectMemory2</i>	130
5.4.4 ソフトウェアブレイク	131
<i>SetPCBreakPt2</i>	131
<i>EnableBreakPt2</i>	132
<i>DeleteBreakPt2</i>	133
<i>GetAllBreakPt2</i>	134
<i>DeleteAllBreakPt2</i>	135
5.4.5 変数ブレイク.....	136
<i>SetDataBreakpoint2</i>	136
<i>EnableDataBreakpoint2</i>	137
<i>DeleteDataBreakpoint2</i>	138
5.4.6 変数トレース.....	139
<i>SetSymbolTrace2</i>	139
<i>ExecuteSymbolTrace2</i>	141
<i>DeleteSymbolTrace2</i>	142
<i>SaveSymbolTraceData2</i>	143
5.4.7 割り込み.....	144
<i>SendTrigger2</i>	144
5.4.8 シンボル操作.....	145
<i>GetRealTimeWatch2</i>	145
<i>GetQuickWatch2</i>	146
<i>SymbolToAddress2</i>	147
<i>AddressToSymbol2</i>	148
<i>GetLineFromAddr2</i>	149
<i>GetAddrFromLine2</i>	150
5.4.9 ダウンロード.....	151
<i>Download2</i>	151
<i>Unload2</i>	152
5.4.10 起動/終了.....	153
<i>InvokeHew2</i>	153
<i>QuitHew2</i>	154
<i>InvokeHewWithNoDialog</i>	155
5.4.11 ワークスペース.....	156
<i>OpenWorkspace2</i>	156
<i>CloseWorkspace2</i>	157
<i>SaveWorkspace2</i>	158
<i>GetWorkSpaceDirectory</i>	159
5.4.12 構成とセッション.....	160
<i>SaveSession2</i>	160
<i>GetCurrentConfiguration2</i>	161
<i>SetCurrentConfiguration2</i>	162
<i>GetConfigurations2</i>	163
<i>GetCurrentSession2</i>	164
<i>SetCurrentSession2</i>	165
<i>GetSessions2</i>	166
<i>GetCurrentProject2</i>	167
<i>SetCurrentProject2</i>	168
<i>GetProjects2</i>	169
5.4.13 プロジェクト.....	170
<i>AddFile2</i>	170
<i>AddFiles2</i>	171
<i>DeleteFile2</i>	172
<i>DeleteFiles2</i>	173
<i>AddProjectFileFolder</i>	174
<i>RemoveProjectFileFolder</i>	175
<i>AddFileToFolder</i>	176
5.4.14 ビルド関係.....	177
<i>BuildProject2</i>	177
<i>RebuildProject2</i>	178
<i>UpDateAllDependency2</i>	179
<i>AddFileWithCompilerOption2</i>	180

<i>GetLibraryOptions</i>	181
<i>SetLibraryOptions</i>	182
<i>GetLibraryFilesForConfiguration</i>	183
<i>SetLibraryFilesForConfiguration</i>	184
<i>GetIncludeFileDirectories</i>	185
<i>SetIncludeFileDirectories</i>	186
<i>GetCpuAndToolChainData</i>	187
<i>SetBuildExcludeFiles</i>	189
<i>SetBuildIncludeFiles</i>	190
5.4.15 ファイル関係.....	191
<i>OpenFileAtLine2</i>	191
<i>GetSourceFiles2</i>	192
<i>GetDownloadModules2</i>	193
<i>GetDependentFiles2</i>	194
5.4.16 カバレッジ関係.....	195
<i>SetCoverageRange2</i>	195
<i>GetCoverageRange2</i>	196
<i>SetCoverageDisable2</i>	197
<i>SetCoverageEnable2</i>	198
<i>ClearCoverage2</i>	199
<i>GetCoverageStatus2</i>	200
<i>LoadCoverage2</i>	201
<i>SaveCoverage2</i>	202
5.4.17 その他.....	203
<i>GetErrorString2</i>	203
<i>GetHewStatus2</i>	204
<i>GetHewStatusEx2</i>	206
<i>GetTargetName2</i>	207
<i>GetHewVersion</i>	208
<i>Command</i>	209
5.5 High-performance Embedded Workshopで取得可能なイベント.....	210

1. 概要

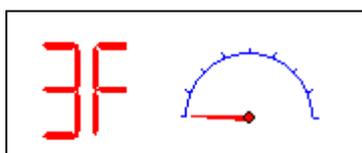
HEW TargetServer (COM) は、High-performance Embedded Workshop の機能を拡張するためのインタフェースを提供します。このインタフェースを使用することにより、High-performance Embedded Workshop 用のカスタマイズウィンドウの作成、他のアプリケーションとの連携環境を構築することができます。



1.1 利用する開発ツール

カスタマイズウィンドウの作成や他のアプリケーションと連携するには、Microsoft 社の Visual C++ や Visual Basic などの Windows アプリケーション開発ツールを使用します。

- 参考書籍が多数販売されています。アプリケーション作成のための情報も入手しやすくなっています。
- 豊富な GUI 部品を標準で備えています。この GUI 部品をユーザシステムのシミュレート部品として使用することが可能です。フリーウェアやシェアウェアのコントロール部品 (ActiveX コントロール) の利用も可能です。独自に作成することも可能です。



1.2 呼び出し可能なメソッド

HEW TargetServer (COM) のインタフェースを通じて呼び出しできるメソッドは、マイコンの実行制御、メモリやレジスタ内容の設定/参照、ソフトウェアブレークポイントの設定などです。これらのメソッドを利用し独自の環境を構築することが可能です。

2. HEW TargetServer (COM) 使用の準備

HEW TargetServer (COM) を使用し外部アプリケーションと連携する際には、使用する High-performance Embedded Workshop 環境にて HEW TargetServer (COM) を有効にする必要があります。本章では、HEW TargetServer (COM) 機能の登録、及び有効にする方法をご説明します。

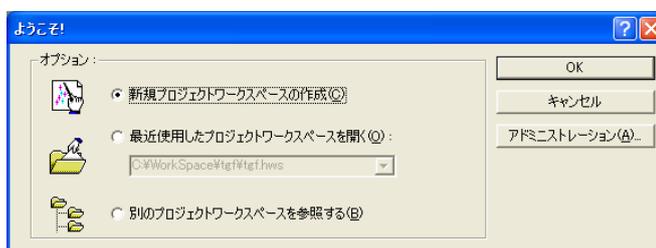
2.1 HEW TargetServer (COM) の登録

High-performance Embedded Workshop をインストールされた初期状態では、HEW TargetServer (COM) の機能をご使用頂くために必要となる EcxHewTargetServer.dll の登録が行われていません。

このため、HEW TargetServer (COM) を使用して作成した外部アプリケーションを起動しても High-performance Embedded Workshop を制御することができません。

2.1.1 EcxHewTargetSever.dllの登録

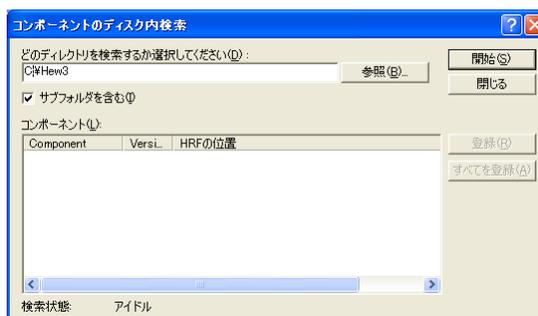
1. High-performance Embedded Workshop を起動してください。“ようこそ!” ダイアログボックスがオープンしますので、“アドミニストレーション” ボタンをクリックしてください。



2. “ツールアドミニストレーション” ダイアログボックスがオープンしたら登録済みコンポーネントの Extension Components のフォルダを選択し開いてください。初期状態では、HewTargetServer が登録されていない事を確認できます。次にディスク内検索ボタンをクリックしてください。



3. “コンポーネントのディスク内検索” ダイアログボックスが起動したら High-performance Embedded Workshop がインストールされているフォルダを指定し開始ボタンをクリックしてください。インストール済みのコンポーネントが表示されます。



4. 表示されたコンポーネントより HewTargetSever を選択し登録ボタンをクリックしてください。



以上で EcxHewTargetSever.dll の登録は終了しました。順次、ダイアログボックスを閉じてください。

注) EcxHewTargetServer.dll の登録後、使用する必要がなくなった場合は登録を解除してください。

2.1.2 HewTargetSever.exeのレジストリへの登録

High-performance Embedded Workshop は V. 4.05 より HewTargetServer.exe をインストーラにより自動的に登録・解除するようになりました。従来行っていた HewTargetServer.exe の登録・解除は不要です。

また、Install Manager Version 1.03 以上は、アクティブな High-performance Embedded Workshop を切り替える際の HewTargetServer.exe の解除・登録を自動で行います。

ただし、Install Manager を使用して複数の High-performance Embedded Workshop をインストールしている場合で、Install Manager Version 1.02 以下でアクティブ対象を変更する場合は、HewTargetServer.exe を手動で解除・登録する必要があります。

HewTargetServer.exe を解除・登録する方法：

現在アクティブである High-performance Embedded Workshop がインストールされているフォルダ内にある ALL_UNREGISTERSERVER.bat ファイルをダブルクリックして実行してください。レジストリから削除されます。

これからアクティブにする High-performance Embedded Workshop がインストールされているフォルダ内にある ALL_REGISTERSERVER.bat ファイルをダブルクリックし実行してください。HewTargetServer.exe がレジストリに登録されます。

Windows Vista の場合、まずコマンドプロンプトを管理者として起動し、cd コマンドで ALL_UNREGISTERSERVER.bat / ALL_REGISTERSERVER.bat ファイルのあるフォルダに移動した後、バッチファイルを実行してください。

(コマンドプロンプトを管理者として起動する方法: スタートメニューの全てのプログラム→アクセサリ→コマンドプロンプトを右クリックし、[管理者として実行]を選択します)

3. HEW TargetServer (COM) の使用方法

外部アプリケーションとの接続を行うために提供している HEW TargetServer (COM) をどのように使用するのかを説明します。

本章では、Visual C++6.0 を使用した High-performance Embedded Workshop のカスタマイズウィンドウの作成方法を、以下の簡単なサンプルウィンドウを例に説明します。

3.1 サンプルプログラム

本サンプルプログラムは、High-performance Embedded Workshop にて作成したプログラムに対してシミュレーション実行する際に外部ウィンドウから実行/停止を指示するウィンドウです。



3.2 プログラムの作成方法 (Visual C++ 6.0)

3.2.1 プロジェクトの生成

Visual C++で新しいプロジェクトを生成します。Visual C++のメニューより[ファイル]->[新規作成]を選択してください。[新規作成]ウィザードが起動します。本ウィザードで” MFC AppWizard(exe)” を選択し、OK ボタンをクリックしてください。



この例ではプロジェクト名を TypeLibraryDemo2 とします。

[ステップ 1]で作成するアプリケーションの種類を指定します。今回はダイアログベースのアプリケーションを作成します。選択後、次へボタンをクリックしてください。



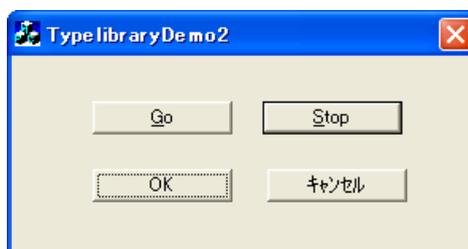
[ステップ 2]でオートメーションをチェックしてください。その他の設定はデフォルトのままかまいません。



これ以降のステップは、デフォルトのままに進めてください。

3.2.2 ボタンの作成

プロジェクトの作成が終了したら、ダイアログボックス上にボタンを作成します。ボタンを配置した際の各 ID とキャプションの設定は以下の通りです。



ID	キャプション	備考
IDC_BUTTON_GO	&Go	-
IDC_BUTTON_STOP	&Stop	-
IDOK	OK	デフォルト
IDCANCEL	キャンセル	デフォルト

3.2.3 ソースコードの作成

次にプロジェクトを作成し生成されたソースコードに記述を追加し HEW TargetServer (COM) を使用します。

(1) HewTargetSever.exe をインポートします。

修正ファイル : TypelibraryDemo2Dlg.h

```
//import type library  
#import "..\Hew2\HewTargetServer.exe" no_namespace
```

ここで指定している HewTargetSever.exe のパスは、ご使用の環境によって異なります。
コンパイラパッケージがインストールされているフォルダを指定してください。

(2) スマートポインタをメンバ変数として宣言します。

修正ファイル : TypelibraryDemo2Dlg.h

```
class CTypelibraryDemo2Dlg : public CDialog  
{  
    . . .  
public:  
    //declare smart pointer  
    IHewServer1Ptr pHewServer1;  
  
};
```

(3) コンストラクタによるスマートポインタの生成と初期化を追加します。

修正ファイル : TypelibraryDemo2Dlg.cpp

```
// TODO: 特別な初期化を行う時はこの場所に追加してください。  
// TODO: Add extra initialization here  
try{  
    //create smart pointer  
    IHewServer1Ptr ptr(_uuidof(HewServer1));  
    pHewServer1 = ptr;  
}
```

(4) クライアント終了時にスマートポインタを破棄するため、以下の記述を追加します。

修正ファイル : TypelibraryDemo2Dlg.cpp

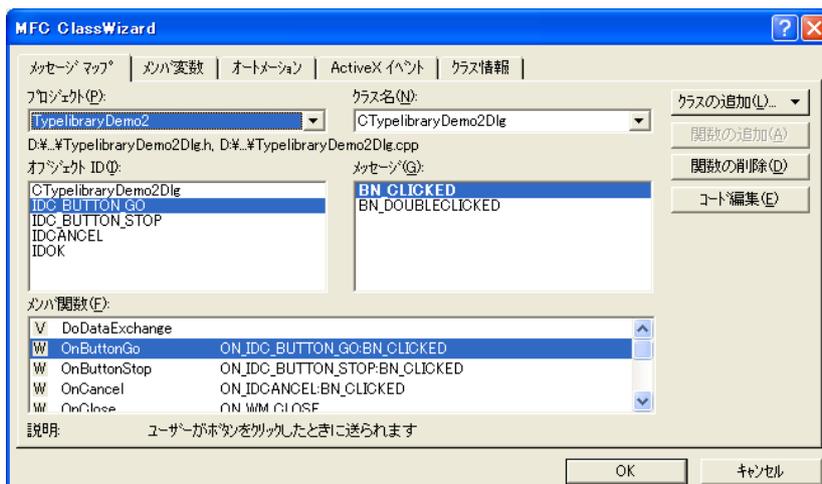
```
CTest1Dlg::~CTest1Dlg()
{
    // このダイアログ用のオートメーション プロキシがある場合は、このダイアログ
    // へのポインタを NULL に戻します、それによってダイアログが削除されたこと
    // がわかります。
    if (m_pAutoProxy != NULL)
        m_pAutoProxy->m_pDialog = NULL;
    //destroy smart pointer
    pHewServer1 = NULL;
}
...
void CTypelibraryDemo2Dlg::OnClose()
{
    if (CanExit())
        CDialog::OnClose();

    //destroy smart pointer
    pHewServer1 = NULL;
}
...
void CTypelibraryDemo2Dlg::OnOK()
{
    if (CanExit())
        CDialog::OnOK();

    //destroy smart pointer
    pHewServer1 = NULL;
}
...
void CTypelibraryDemo2Dlg::OnCancel()
{
    if (CanExit())
        CDialog::OnCancel();

    //destroy smart pointer
    pHewServer1 = NULL;
}
```

(5)次にボタンをクリックされたときに呼び出される関数を ClassWizard で追加します。



ボタンのオブジェクト、メッセージを選択して[関数の追加]ボタンをクリックすると、TypelibraryDemo2Dlg.cpp に以下のような空の関数が挿入されます。

```
void CTypelibraryDemo2Dlg::OnButtonGo()
{
    // TODO: この位置にコントロール通知ハンドラ用のコードを追加してください
}
```

注意事項 (Visual C++ 6.0/2005 共通)

High-performance Embedded Workshop で公開している関数を呼び出すときには、必ず try {}, catch {} を使用しなくてはなりません。インタフェース関数呼び出しでエラーが発生したときには、catch {} で COM システムからの COM エラーを取得することができます。catch {} を使用しない場合に、COM エラーが発行されると、クライアントプログラムは、アプリケーションエラーを引き起こします。

COM エラーには、以下の 3 種類のエラーがあります。

カスタムエラー (HewTargetServer.exe が発行するエラー)

本エラーには、High-performance Embedded Workshop が起動していないとき、ターゲットを接続していないとき、ロードモジュールがダウンロードされていないときに HewTargetServer.exe が呼ばれたときに返却するエラーが含まれます。

HEW エラー (HEW2.exe が発行するエラー)

High-performance Embedded Workshop がエラーを返却する場合は、呼び出されたインタフェースのパラメータが異常の可能性があります。

High-performance Embedded Workshop がエラーを返却したときには、GetErrorString() を呼び出すことにより、エラー内容を取得することができます。

システムエラー (COM システムが発行するエラー)

COM システムによってエラーが返却される場合は、RPC(Remote Procedure Call)環境に問題があるか、クライアントと HewTargetServer.exe のコミュニケーションに問題があります。

(6)OnButtonGo()関数作成

修正ファイル : TypelibraryDemo2Dlg.cpp

```
void CTypelibraryDemo2Dlg::OnButtonGo()
{
    HRESULT          hr = E_FAIL, hrErr = E_FAIL;
    CString          s1;

    //calling HewTargetServer function
    try
    {
        hr = pHewServer1->GoTargetExec(); // プログラムを実行するメソッドを記述します。
    }
    catch(_com_error e1)
    {
        if(e1.Description().length()>0) //display custom COM error
            AfxMessageBox(e1.Description());
        else
        {
            BSTR      bstrErrStr;

            try
            {
                hrErr = pHewServer1->GetErrorString(e1.Error(), &bstrErrStr);
                // エラー内容を取得するメソッドを記述します。
            }
            catch(_com_error e)
            {
            }
            if(SUCCEEDED(hrErr))
            {
                s1.Format("%s", CString(bstrErrStr));
                AfxMessageBox(s1);
            }
            else{ //display system error
                AfxMessageBox(e1.ErrorMessage());
            }
        }
    }
}
```

(7)OnButtonStop()関数作成

修正ファイル : TypelibraryDemo2Dlg.cpp

```
void CTypelibraryDemo2Dlg::OnButtonStop()
{
```

```
    HRESULT          hr = E_FAIL, hrErr = E_FAIL;
    CString          s1;

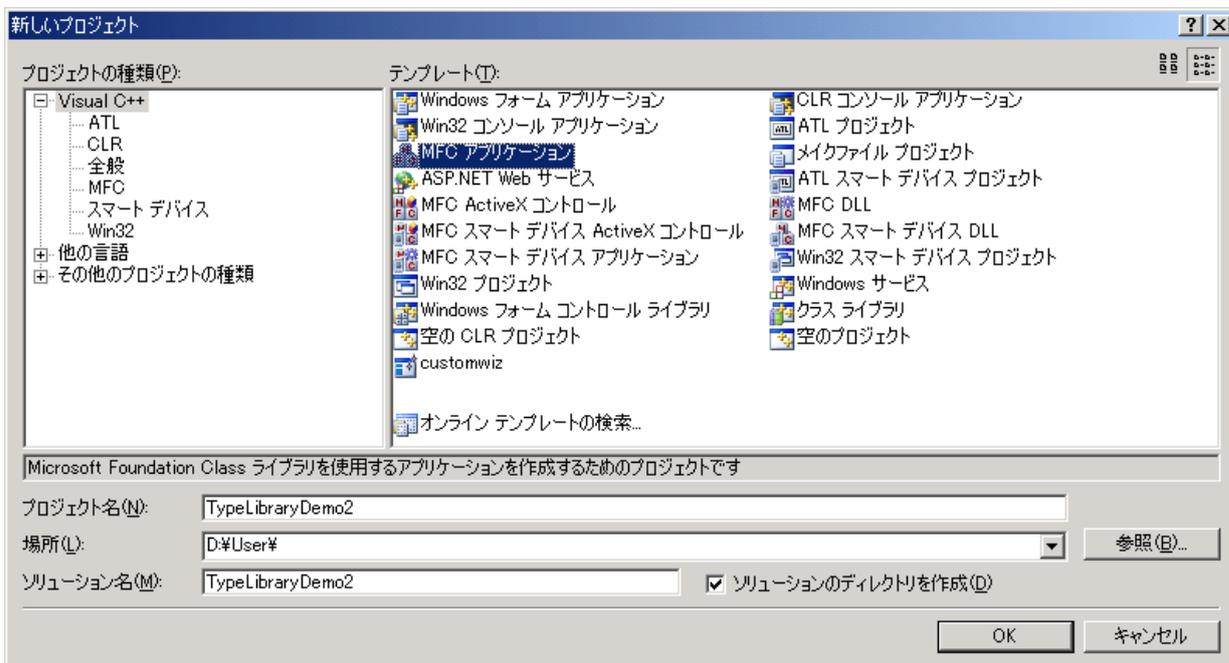
    //calling HewTargetServer function
    try
    {
        hr = pHewServer1->StopTargetExec(); // プログラムの実行を停止するメソッドを追加し
ます。
    }
    catch(_com_error e1)
    {
        if(e1.Description().length()>0) //display custom COM error
            AfxMessageBox(e1.Description());
        else
        {
            BSTR      bstrErrStr;

            try
            {
                hrErr = pHewServer1->GetErrorString(e1.Error(), &bstrErrStr);
                // エラー内容を取得するメソッドを追加します。
            }
            catch(_com_error e)
            {
            }
            if(SUCCEEDED(hrErr))
            {
                s1.Format("%s", CString(bstrErrStr));
                AfxMessageBox(s1);
            }
            else
            {
                //display system error
                AfxMessageBox(e1.ErrorMessage());
            }
        }
    }
}
```

3.3 プログラムの作成方法 (Visual C++ 2005)

3.3.1 プロジェクトの生成

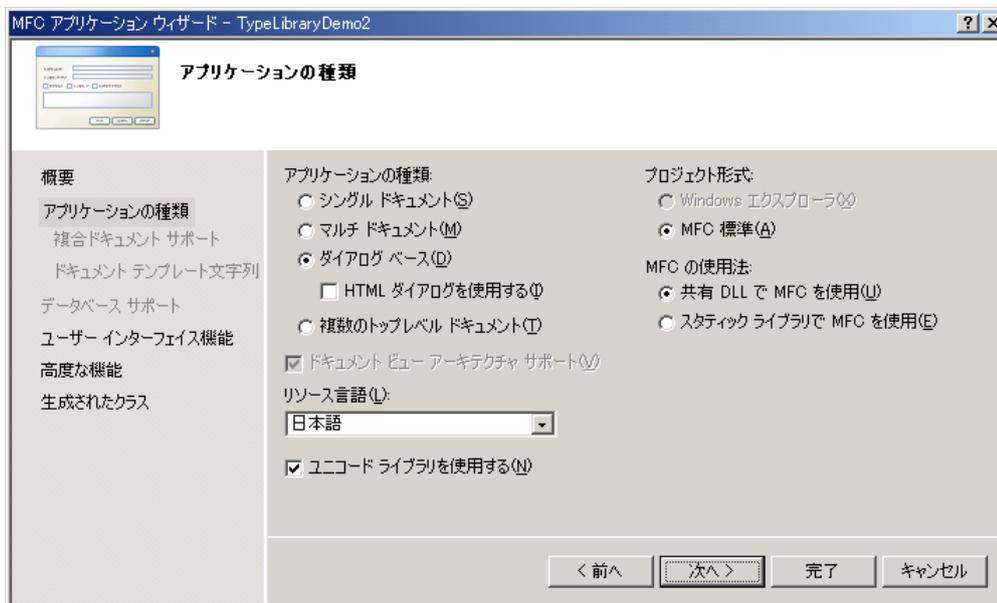
Visual C++で新しいプロジェクトを生成します。Visual C++のメニューより[ファイル]->[新規作成]->[プロジェクト]を選択してください。[新しいプロジェクト]ウィザードが起動します。本ウィザードで”MFC アプリケーション”を選択し、OK ボタンをクリックしてください。



この例ではプロジェクト名を TypeLibraryDemo2 とします。

[MFC アプリケーションウィザードによる]画面が表示されます。次へボタンをクリックしてください。

[アプリケーションの種類]でダイアログベースを選択し、次へボタンをクリックしてください。



[ユーザー インターフェイス機能]はデフォルトで次へボタンをクリックしてください。

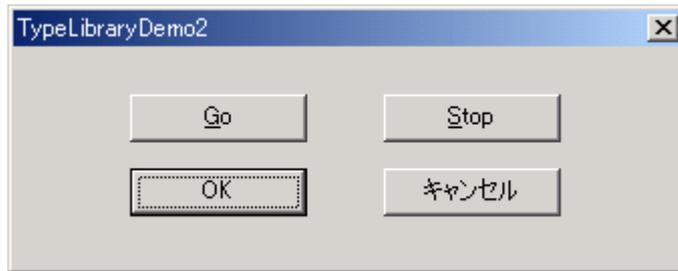
[高度な機能]でオートメーションをチェックしてください。その他の設定はデフォルトのままかまいません。



これ以降のステップは、デフォルトのままを進めてください。

3.3.2 ボタンの作成

プロジェクトの作成が終了したら、ダイアログボックス上にボタンを作成します。ボタンを配置した際の各 ID とキャプションの設定は以下の通りです。



ID	キャプション	備考
IDC_BUTTON_GO	&Go	-
IDC_BUTTON_STOP	&Stop	-
IDOK	OK	デフォルト
IDCANCEL	キャンセル	デフォルト

3.3.3 ソースコードの作成

次にプロジェクトを作成し生成されたソースコードに記述を追加し HEW TargetServer (COM) を使用します。

(1) HewTargetServe.exe をインポートします。

修正ファイル : TypelibraryDemo2Dlg.h

```
//import type library  
#import "..\Hew2\HewTargetServer.exe" no_namespace
```

ここで指定している HewTargetSever.exe のパスは、ご使用の環境によって異なります。コンパイラパッケージがインストールされているフォルダを指定してください。

(2) スマートポインタをメンバ変数として宣言します。

修正ファイル : TypelibraryDemo2Dlg.h

```

class CTypelibraryDemo2Dlg : public CDialog
{
    . . .
public:
    //declare smart pointer
    IHewServer1Ptr pHewServer1;

};

```

(3) コンストラクタによるスマートポインタの生成と初期化を追加します。

修正ファイル : TypelibraryDemo2Dlg.cpp

```

// TODO: 特別な初期化を行う時はこの場所に追加してください。
// TODO: Add extra initialization here
try{
    //create smart pointer
    IHewServer1Ptr ptr(_uuidof(HewServer1));
    pHewServer1 = ptr;
}

```

(4) クライアント終了時にスマートポインタを破棄するため、以下の記述を追加します。

修正ファイル : TypelibraryDemo2Dlg.cpp

```

CTest1Dlg::~CTest1Dlg()
{
    // このダイアログ用のオートメーション プロキシがある場合は、このダイアログ
    // へのポインタを NULL に戻します、それによってダイアログが削除されたこと
    // がわかります。
    if (m_pAutoProxy != NULL)
        m_pAutoProxy->m_pDialog = NULL;
    //destroy smart pointer
    pHewServer1 = NULL;
}
. . .
void CTypelibraryDemo2Dlg::OnClose()
{
    if (CanExit())
        CDialog::OnClose();

    //destroy smart pointer
    pHewServer1 = NULL;
}
. . .
void CTypelibraryDemo2Dlg::OnOK()
{
    if (CanExit())
        CDialog::OnOK();

    //destroy smart pointer
    pHewServer1 = NULL;
}
. . .
void CTypelibraryDemo2Dlg::OnCancel()
{
    if (CanExit())
        CDialog::OnCancel();

    //destroy smart pointer
    pHewServer1 = NULL;
}

```

(5)次にボタンをクリックされたときに呼び出される関数を追加します。

(5-1) クラスビューで `CTypeLibraryDemo2Dlg` クラスの右クリックメニューを表示し、プロパティを選択します。

(5-2) プロパティで  をクリックし、管理イベントを表示します。

(5-3) `IDC_BUTTON_GO` の `[+]` をクリックし、`BN_CLICKED` を選択します。

(5-4) ドロップダウンメニューの `<追加>OnBnClickedButtonGo` を選択します。

(5-5) 同様に、`IDC_BUTTON_STOP` についても追加します。

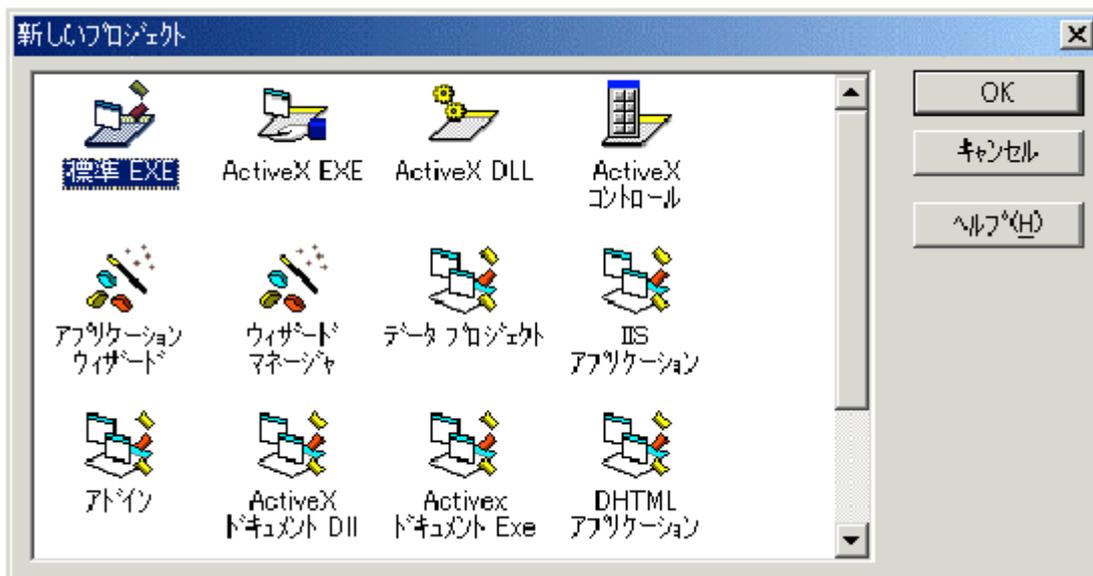
注意事項と(6)OnButtonGo関数作成 以降は Visual C++6.0 の記述(3.2.3 ソースコードの作成の p.9～) をご参照ください。ユニコードライブラリを使用する場合は文字列定数に `_T()` マクロを適用してください。

例) `s1.Format(_T("%s"), CString(bstrErrStr));`

3.4 プログラムの作成方法 (Visual Basic 6.0)

3.4.1 プロジェクトの生成

新しいプロジェクトを生成します。Visual Basic のメニュー[ファイル]→[新しいプロジェクト]を選択してください。“新しいプロジェクト”ダイアログボックスがオープンします。そのダイアログボックスで“標準 EXE”を選択し、OK ボタンをクリックしてください。



3.4.2 タイプライブラリの指定

Visual Basic の[Project]→[References...] メニューを選択し、“HEWTargetServer 1.7 Type Library”をチェックします。タイプライブラリの指定は、Visual Basic のプロジェクトごとに設定する必要があります。

3.4.3 オブジェクトの生成

VB のコードウィンドウで以下のように記述してください。このコードが HEWTargetServer の COM インタフェースを呼び出すための基本コードとなります。

```
1: Dim WithEvents hts As HEWTargetServerLib.HewServer1
2:
3: Private Sub Form_Load()
4:     Set hts = New HEWTargetServerLib.HewServer1
5: End Sub
6:
7: Private Sub Form_Unload(Cancel As Integer)
8:     Set hts = Nothing
9: End Sub
```

(各行の説明)

- 1 行目: ここでは、オブジェクト変数 hts が “HEWTargetServerLib.HewServer1” という型であることを宣言しています。これは、COM のインタフェース名です。また、WithEvents の記述は、プログラム実行開始やプログラム停止などの High-performance Embedded Workshop 側で発生したイベントを取得するために指定します。変数名 hts は任意です。
- 3~5 行目: Form_Load は、アプリケーション起動時(フォームオープン時)に呼び出されるプロシージャ(関数)です。ここでは、変数 hts に “HEWTargetServerLib.HewServer1” というオブジェクトを代入しています。HEWTargetServer のメソッドは、この変数 hts を通して、呼び出します。
- 7~9 行目: Form_Unload は、アプリケーション終了時(フォームクローズ時)に呼び出されるプロシージャ(関数)です。ここでは、変数 hts のオブジェクトを破棄しています。オブジェクトを破棄すると、HEWTargetServer のメソッドを呼び出すことはできなくなります。

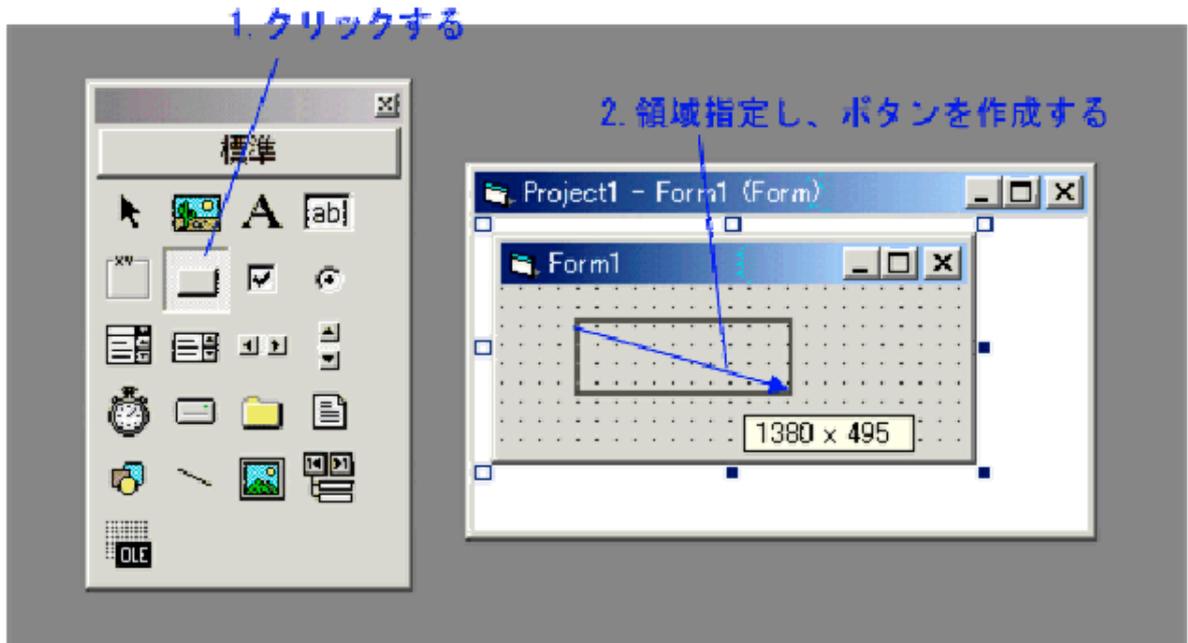
3.4.4 メソッドの呼び出し

サンプル: ユーザーターゲットをリセットする

ユーザーターゲットをリセットするカスタマイズウィンドウの作成方法を以下に示します。このアプリケーションでは、ボタンコントロールを1個使用します。

(1) ボタンの追加

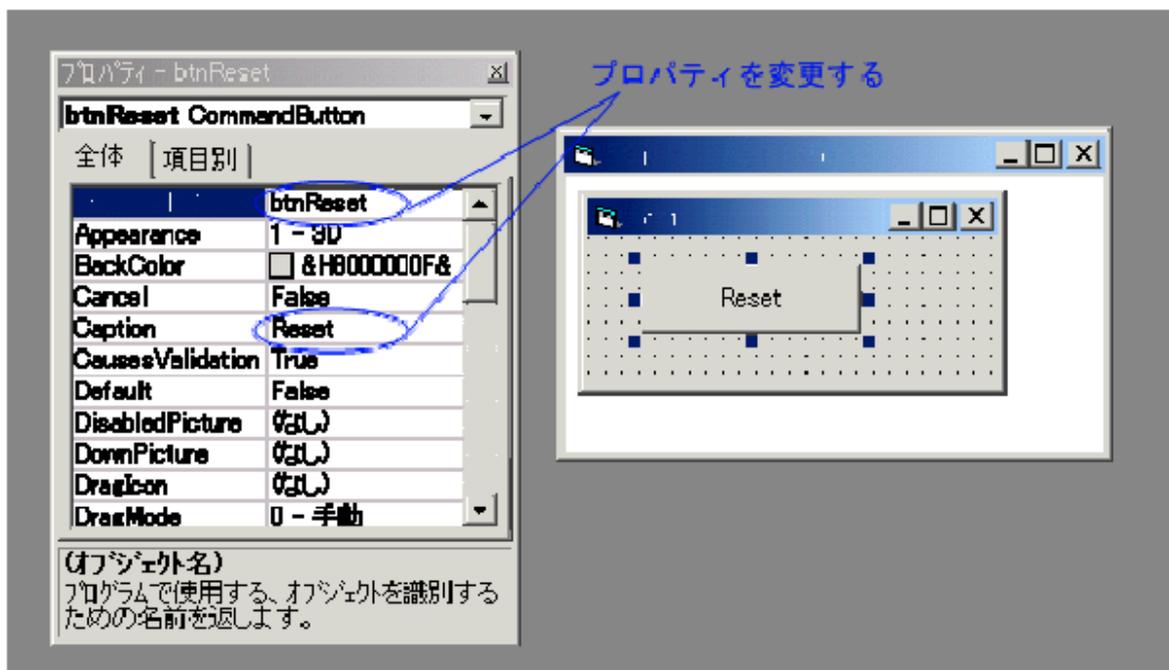
ツールボックスの CommandButton をクリックし、フォーム上にボタンコントロールを1つ作成してください。



(2) ボタンのプロパティ変更

作成したボタンコントロールのプロパティをプロパティウィンドウで変更します。

プロパティ	内容
(オブジェクト名)	btnReset
Caption	Reset



(3) ボタンの動作を記述

コードウィンドウで以下のように記述してください。太字箇所が追加した部分です。

```
Dim WithEvents hts As HEWTARGETSERVERLib.HewServer1
Private Sub Form_Load()
    Set hts = New HEWTARGETSERVERLib.HewServer1
End Sub

Private Sub Form_Unload(Cancel As Integer)
    Set hts = Nothing
End Sub

Private Sub btnReset_Click()
    Dim rtn As Long
    On Error GoTo HTS_error
    rtn = HTS.GoTargetExec2
    Exit Sub
HTS_error:
    MsgBox Err.Description
End Sub
```

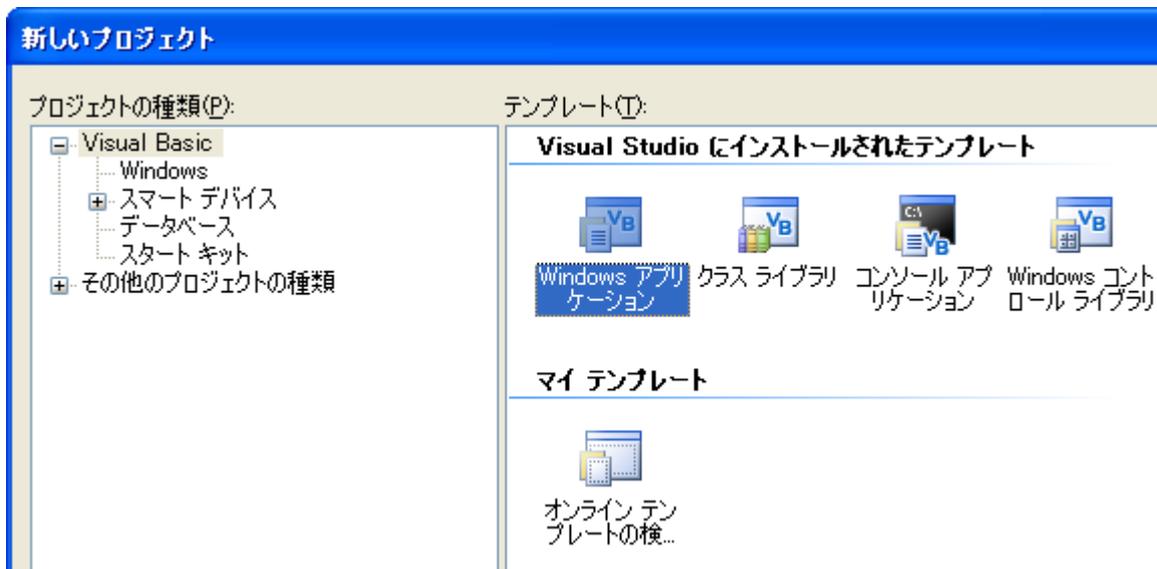
(4) 動作確認

アプリケーションの動作を確認します。まず、High-performance Embedded Workshop を起動してください。その後、Visual Basic のメニュー[実行]->[開始]を選択し、アプリケーションを実行してください。アプリケーション上の Reset ボタンをクリックすると、ユーザターゲットがリセットされます。

3.5 プログラムの作成方法 (Visual Basic 2005)

3.5.1 プロジェクトの生成

新しいプロジェクトを生成します。Visual Basic のメニュー[ファイル]→[新しいプロジェクト]を選択してください。“新しいプロジェクト”ダイアログボックスがオープンします。そのダイアログボックスで“標準 EXE”を選択し、OK ボタンをクリックしてください。



3.5.2 タイプライブラリの指定

Visual Basic の [プロジェクト]→[参照の追加...] メニューを選択し、[COM] タブで “HEWTargetServer 1.7 Type Library” を選択し、OK をクリックします。タイプライブラリの指定は、Visual Basic のプロジェクトごとに設定する必要があります。

3.5.3 オブジェクトの生成

VB のコードウィンドウで以下のように記述してください。このコードが HEWTargetServer の COM インタフェースを呼び出すための基本コードとなります。

```
1: Public Class Form1
2:     Dim WithEvents hts As HEWTARGETSERVERLib.HewServer1
3:
4:     Private Sub Form1_Load(ByVal sender As Object, ByVal e As System.EventArgs)
Handles Me.Load
5:         hts = New HEWTARGETSERVERLib.HewServer1
6:     End Sub
7:
8:     Private Sub Form1_FormClosed(ByVal sender As Object, ByVal e As System.
Windows.Forms.FormClosedEventArgs) Handles Me.FormClosed
9:         hts = Nothing
10:    End Sub
11: End Class
```

(各行の説明)

- 2 行目: ここでは、オブジェクト変数 hts が “HEWTargetServerLib.HewServer1” という型であることを宣言しています。これは、COM のインタフェース名です。また、WithEvents の記述は、プログラム実行開始やプログラム停止などの High-performance Embedded Workshop 側で発生したイベントを取得するために指定します。変数名 hts は任意です。
- 4~6 行目: Form_Load は、アプリケーション起動時(フォームオープン時)に呼び出されるプロシージャ(関数)です。ここでは、変数 hts に “HEWTargetServerLib.HewServer1” というオブジェクトを代入しています。HewTargetServer のメソッドは、この変数 hts を通して、呼び出します。
- 8~10 行目: Form_FormClosed は、アプリケーション終了時(フォームクローズ時)に呼び出されるプロシージャ(関数)です。ここでは、変数 hts のオブジェクトを破棄しています。オブジェクトを破棄すると、HewTargetServerのメソッドを呼び出すことはできなくなります。

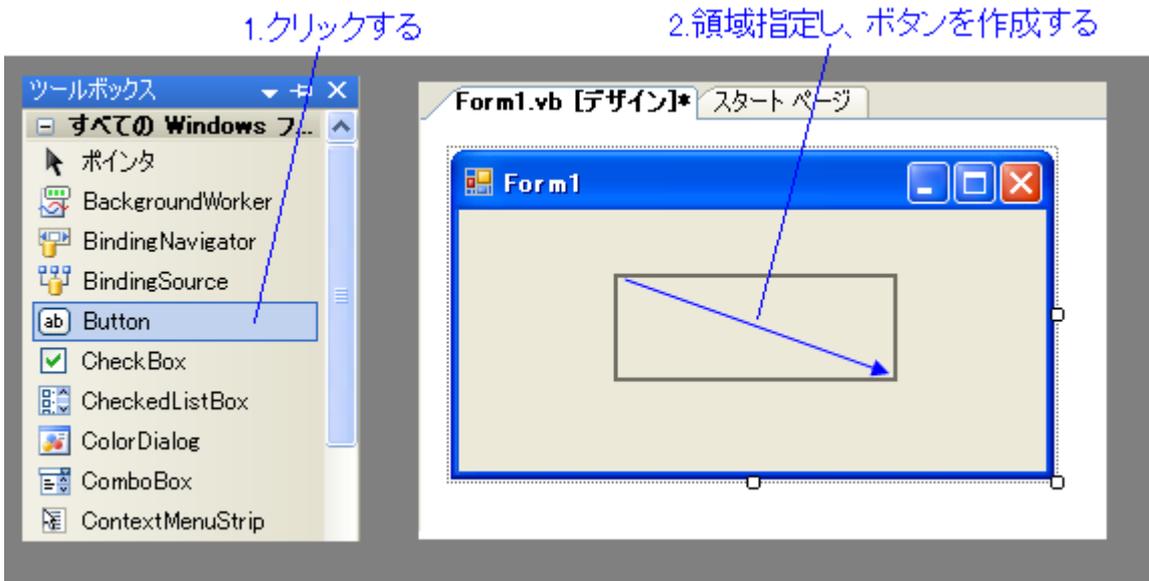
3.5.4 メソッドの呼び出し

サンプル: ユーザーターゲットをリセットする

ユーザーターゲットをリセットするカスタマイズウィンドウの作成方法を以下に示します。このアプリケーションでは、ボタンコントロールを1個使用します。

(1) ボタンの追加

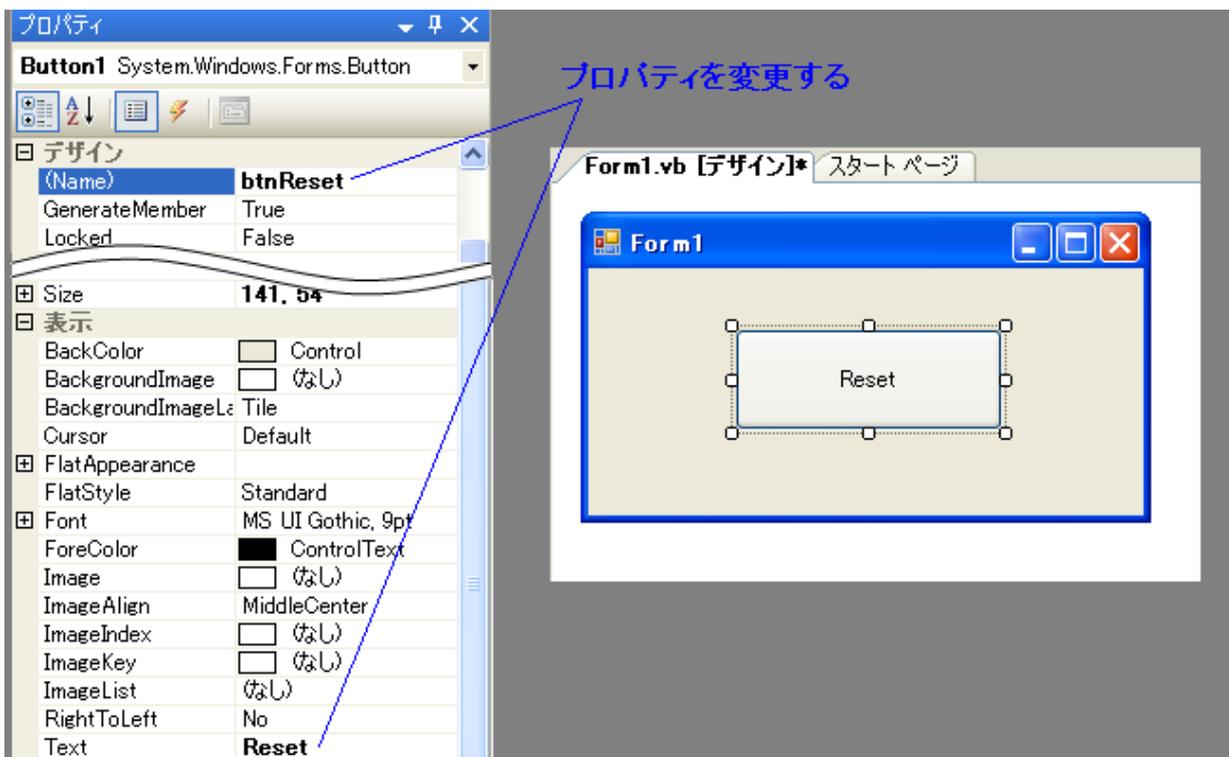
ツールボックスの Button をクリックし、フォーム上にボタンコントロールを1つ作成してください。



(2) ボタンのプロパティ変更

作成したボタンコントロールのプロパティをプロパティウィンドウで変更します。

プロパティ	内容
(Name)	btnReset
Text	Reset



(3) ボタンの動作を記述

btnReset ボタンの Click イベントを追加します。(詳細は VisualBasic のヘルプ等をご参照ください)
コードウィンドウで以下のように記述してください。太字箇所が追加した部分です。

```
Public Class Form1
    Dim WithEvents hts As HEWTARGETSERVERLib.HewServer1

    Private Sub Form1_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles
Me.Load
        hts = New HEWTARGETSERVERLib.HewServer1
    End Sub

    Private Sub Form1_FormClosed(ByVal sender As Object, ByVal e As System.Windows.
Forms.FormClosedEventArgs) Handles Me.FormClosed
        hts = Nothing
    End Sub

    Private Sub btnReset_Click(ByVal sender As System.Object, ByVal e As System. EventArgs)
Handles btnReset.Click
        Dim rtn As Integer

        Try
            rtn = hts.ResetTargetExec2()

            Catch ex As System.Runtime.InteropServices.COMException
                If ex.ErrorCode = &H8004FFFF Then
                    MessageBox.Show("Hew Target is not linked up")
                ...
            Catch ex As Exception
                MessageBox.Show(ex.ToString())
            End Try

    End Sub
End Class
```

(4) 動作確認

アプリケーションの動作を確認します。まず、High-performance Embedded Workshop を起動してください。その後、Visual Basic のメニュー[デバッグ]->[デバッグ開始]を選択し、アプリケーションを実行してください。アプリケーション上の Reset ボタンをクリックすると、ユーザターゲットがリセットされます。

3.6 Visual Basic 6.0 から Visual Basic .NET に移行する場合の注意点

Visual Basic 6.0 から Visual Basic .NET に移行する場合、Visual Basic 6.0 の非構造化例外処理 (On Error) を使用する時は、以下のように例外処理を修正してください。

(1) Visual Basic 6.0 の非構造化例外処理

```
HTS_error:
    HTSErrorMsgBox Err.Description
End Sub
```

(2) Visual Basic .NET の非構造化例外処理

```
HTS_error:
    strErrorMessage = ""
    If Err.Number = &H8004FFFF Then
        strErrorMessage = "Hew Target is not linked up"
    ElseIf Err.Number = &H80050000 Then
        strErrorMessage = "No module is downloaded"
    ElseIf Err.Number = &H80050001 Then
        strErrorMessage = "Invalid Break point handle"
    ElseIf Err.Number = &H80050002 Then
        strErrorMessage = "Error in pass in parameters"
    ElseIf Err.Number = &H80050003 Then
        strErrorMessage = "Invalid Begin and End address"
    ElseIf Err.Number = &H80050004 Then
        strErrorMessage = "No Interface"
    ElseIf Err.Number = &H80050005 Then
        strErrorMessage = "HEW was not found."
    Else
        HTS.GetErrorString2(Err.Number, strErrorMessage)
    End If

    HTSErrorMsgBox(strErrorMessage)
End Sub
```

4. High-performance Embedded Workshopからのイベント取得

ダイアログベースのアプリケーションを作成した場合、以下の方法でイベント(Event3_ToClient_Stop)を取得できます。本サンプルでは、High-performance Embedded Workshop でターゲットプログラムが停止したダイアログボックスを表示します。

注意事項

(1) 毎回イベントを取得する場合

ターゲットを実行中に何度も発生するイベントは、フラグを立ててイベントを毎回発行しないようにしています。以下のイベントは、一度イベントを発行してフラグが立っている場合、イベントを発行しません。

Event1_ToClient_TargetReset/Event4_ToClient_MemoryReset/Event5_ToClient_RegisterReset/Event8_ToClient_PlatformInitialize

フラグは、GetHewStatus() メソッドを呼び出すことでクリアされます。イベントが発行された後に GetHewStatus() メソッドを呼べば毎回イベントを取得できます。

4.1 Visual C++のイベント取得

(1) HewTargetServe.exe をインポートします。

修正ファイル : StdAfx.h

```
#import "HewTargetServer.exe"          no_namespace named_guids
```

(2) AfxOleInit() を追加します。

修正ファイル : CclientApp.cpp

```
BOOL CclientApp::InitInstance()
{
    AfxOleInit();
    . . .
}
```

(3) コンストラクタによるスマートポインタの生成と初期化をします。

修正ファイル : CclientDlg.cpp

```
#include "Afxctl.h"
```

. . .

```
CClientDlg::CClientDlg()
{
    . . .
    try{
        //creating smart pointer from new i/f: IHewServer2
        IHewServer2Ptr ptr(_uuidof(HewServer1));
        pHewServer1 = ptr;
    }
    catch(_com_error e)
    {
    }
    if(pHewServer1)
    {
        IUnknownPtr pUnk = pHewServer1;
        LPUNKNOWN pUnkHandler = this->GetIDispatch(FALSE);
        //The connection is established between source (HewTargetServer) and sink (Client).
        AfxConnectionAdvise(pUnk,  DIID_IHewServer2Events,  pUnkHandler,  FALSE,
        &m_dwCookie);
    }
}
```

(4) スマートポインタをメンバ変数として宣言します。

修正ファイル : ClientDlg.h

```

#include "EventHandler.h"
. . .
public:
    IHewServer2Ptr  pHewServer1;           //using smart pointer from new interface: IHewServer2
    protected:
        CEventHandler*  m_pHandler;
        DWORD          m_dwCookie;
. . .

```

(5) イベント取得ファイルの作成 1

修正ファイル : EventHandler.cpp (新規ファイル作成)

// EventHandler.cpp : implementation file

```

#include "stdafx.h"
#include "EventHandler.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

IMPLEMENT_DYNCREATE(CEventHandler, CCmdTarget)

CEventHandler::CEventHandler()
{
    EnableAutomation();
}

CEventHandler::~CEventHandler()
{
}

void CEventHandler::OnFinalRelease()
{
    CCmdTarget::OnFinalRelease();
}

BEGIN_MESSAGE_MAP(CEventHandler, CCmdTarget)
   //{{AFX_MSG_MAP(CEventHandler)
    //}}AFX_MSG_MAP
END_MESSAGE_MAP()

BEGIN_DISPATCH_MAP(CEventHandler, CCmdTarget)
   //{{AFX_DISPATCH_MAP(CEventHandler)
    DISP_FUNCTION_ID(CEventHandler, "Event3_ToClient_Stop", 3, OnHewStatusStop, VT_EMPTY,
0)    //}}AFX_DISPATCH_MAP
END_DISPATCH_MAP()

BEGIN_INTERFACE_MAP(CEventHandler, CCmdTarget)
    INTERFACE_PART(CEventHandler, DIID_IHewServer2Events, Dispatch)
END_INTERFACE_MAP()

////////////////////////////////////
// CEventHandler message handlers
void CEventHandler::OnHewStatusStop()
{
    AfxMessageBox("Event3_ToClient_Stop");
}

```

(6) イベント取得ファイルの作成 2

修正ファイル : EventHandler.h

(新規作成ファイル)

```
#if !defined(AFX_EVENTHANDLER_H__0F96FDDD_7167_457D_8069_73D9AEFCDF49_INCLUDED_)
#define AFX_EVENTHANDLER_H__0F96FDDD_7167_457D_8069_73D9AEFCDF49_INCLUDED_

#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000
// EventHandler.h : header file

////////////////////////////////////
// CEventHandler command target

class CEventHandler : public CCmdTarget
{
    DECLARE_DYNCREATE(CEventHandler)
    CEventHandler(); // protected constructor used by dynamic creation
public:
// Overrides
// ClassWizard generated virtual function overrides
//{{AFX_VIRTUAL(CEventHandler)
public:
    virtual void OnFinalRelease();
//}}AFX_VIRTUAL

// Implementation
public:
    virtual ~CEventHandler();

// Generated message map functions
//{{AFX_MSG(CEventHandler)
    afx_msg void OnHewStatusStop(); //event call back function
//}}AFX_MSG

    DECLARE_MESSAGE_MAP()
// Generated OLE dispatch map functions
//{{AFX_DISPATCH(CEventHandler)
    // NOTE - the ClassWizard will add and remove member functions here.
//}}AFX_DISPATCH
    DECLARE_DISPATCH_MAP()
    DECLARE_INTERFACE_MAP()
};
////////////////////////////////////
//{{AFX_INSERT_LOCATION}}
// Microsoft Visual C++ will insert additional declarations immediately before the previous line.

#endif
// !defined(AFX_EVENTHANDLER_H__0F96FDDD_7167_457D_8069_73D9AEFCDF49_INCLUDED_)
```

4.2 Visual Basicのイベント取得

High-performance Embedded Workshop 側で発生したイベント（プログラムの実行開始など）を取得するには、プロシージャ `hts_GotEventMessage` を使用してください。このプロシージャは、High-performance Embedded Workshop 側で発生したイベントを受け取ったときに呼び出されるサブルーチンです。（プロシージャ名の先頭に付いている“hts”は、プログラム先頭に位置するオブジェクトの変数名を表します。）

Visual Basic 6.0

```
Private Sub hts_GotEventMessage(ByVal action As Long)
End Sub
```

Visual Basic 2005

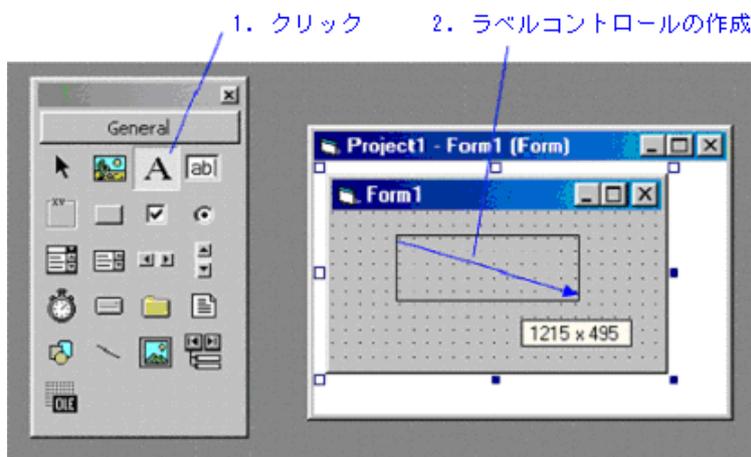
```
Private Sub hts_GotEventMessage(ByVal action As Integer)
End Sub
```

High-performance Embedded Workshop 側で発生したイベントの番号は、パラメータ“action”に保存されています。High-performance Embedded Workshop からイベントを取得したときに実行される処理を、このプロシージャで記述してください。

サンプル: High-performance Embedded Workshop 側で発生したイベントを取得してください。High-performance Embedded Workshop 側で発生したイベントを取得し、その番号を表示するためのカスタマイズウィンドウを作成する方法を、以下に示します。（操作法、図は Visual Basic 6.0）このアプリケーションでは、1 個のラベルコントロールが使用されています。

(1) ラベルコントロールの追加

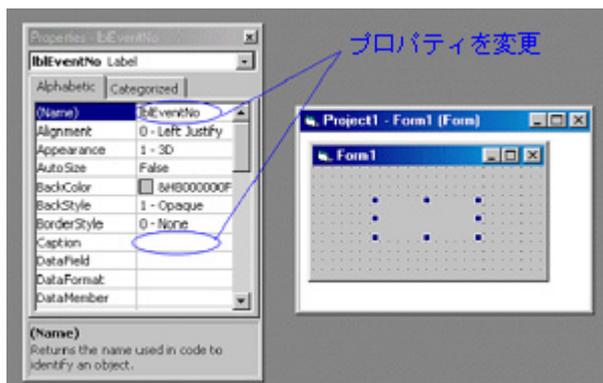
ツールボックスから[ラベル]を選択し、ラベルコントロールを作成してください。



(2) プロパティの変更

[プロパティ]ウィンドウで、作成したラベルコントロールのプロパティを変更してください。

プロパティ	内容
(Name)	lblEventNo
Caption	(nothing)



(3) イベント発生時の動作の記述

[コード]ウィンドウで以下のように記述してください。赤い字は追加部分を示します。

Visual Basic 6.0

```
Dim WithEvents hts As HEWTARGETSERVERLib.HewServer1
Private Sub Form_Load()
    Set hts = New HEWTARGETSERVERLib.HewServer1
End Sub
Private Sub Form_Unload(Cancel As Integer)
    Set hts = Nothing
End Sub
Private Sub hts_GotEventMessage(ByVal action As Long)
    lblEventNo.Caption = action
End Sub
```

Visual Basic 2005

```
Public Class Form1
    Dim WithEvents hts As HEWTARGETSERVERLib.HewServer1
    Private Sub Form1_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
        hts = New HEWTARGETSERVERLib.HewServer1
    End Sub
    Private Sub Form1_FormClosed(ByVal sender As Object, ByVal e As System.Windows.
Forms.FormClosedEventArgs) Handles Me.FormClosed
        hts = Nothing
    End Sub
    Private Sub hts_GotEventMessage(ByVal action As Integer)
        lblEventNo.Caption = action
    End Sub
End Class
```

(4) アクションの確認

アプリケーションのアクションを確認してください。まず High-performance Embedded Workshop を起動し、Visual Basic メニューの[EXEC]->[START]を選択して、アプリケーションを実行してください。High-performance Embedded Workshop を操作することによって（実行/停止など）、High-performance Embedded Workshop 内部で発生したイベントの番号が表示されます。

5. メソッドの一覧

HEW TargetServer (COM) は、以下のメソッドを公開しています。

注意事項

(1) VisualBasic6.0 のサポート

HewTargetServer は、High-performance Embedded Workshop (V4.02) から VisualBasic6.0 (以下 VB) に対応しました。既存のメソッドは、VB から使用できるように引数の型を変更しました。引数の型を変更したメソッドは、メソッド名に 2 を付けて区別します。

(例 1) Step(2, 1); // 既存の VC++ 用メソッド、引数の型は DWORD 型

(例 2) Step2(2, 1); // 変更した VB、VC++ 用メソッド、引数の型は long 型

High-performance Embedded Workshop (V4.02) から追加されるメソッドは、VB、VC++ 両方から使用できます。

(2) GetMemory() と GetDirectMemory() の違い

GetMemory() は、キャッシュからメモリ内容を取得します。

GetDirectMemory() は、MCU からメモリ内容を取得します。

ターゲット実行中にメモリを取得したい場合は、GetDirectMemory() を使ってください。

ターゲットがキャッシュをサポートしていない場合、GetMemory() と GetDirectMemory() は、同じ処理をします。

(3) 複数の High-performance Embedded Workshop を起動

High-performance Embedded Workshop が複数起動している場合、HEW Target Server は個々の High-performance Embedded Workshop を区別できません。

そのため、複数の High-performance Embedded Workshop は、同時に同じ処理を行います。

(4) HewTargetServer.exe のパスが変わる場合

High-performance Embedded Workshop の再インストール等で、HewTargetServer.exe のパスが変わる場合、HewTargetServer.exe をレジストリ登録ツール (ALL_REGISTERSERVER.bat) を使用して再登録してください。

(詳細は「2.1.2 HewTargetServer.exe のレジストリへの登録」をご参照下さい)

5.1 メソッド概要 (VC++のみ使用可)

5.1.1 実行制御

メソッド名	パラメータ	説明	ページ
GoTargetExec	(なし)	プログラムを実行する	P38
StopTargetExec	(なし)	プログラムを停止する	P39
ResetTargetExec	(なし)	プログラムをリセットする	P40
InitializeTarget	(なし)	ターゲットを初期化する	P41
Step	[in]DWORD _eMode [in]DWORD _dwStep	プログラムをステップ実行する	P42
StepRate	[in]DWORD _dwRate	ステップ実行速度を設定する	P43
StepOver	[in]DWORD _eMode [in]DWORD _dwStep	プログラムをステップオーバーで実行する	P44
StepOut	[in]DWORD _eMode	プログラムをステップアウトで実行する	P45
IsRunning	[out]long* p_bRunning	プログラムが実行中かチェックする	P46

5.1.2 レジスタ操作

メソッド名	パラメータ	説明	ページ
GetPC	[out]DWORD *p_dwPC	プログラムカウンタの値を取得する	P47
SetPCAddress	[in]ADDR aPCAddr	プログラムカウンタの値をアドレスで設定する	P48
SetPCSource	[in]BSTR bstrFileName [in]LINENO LineNum	プログラムカウンタの値を行番号で設定する	P49
TestSetPC	[out]long* p_bSetPCState	PC 設定のチェックをする	P50

5.1.3 メモリ操作

メソッド名	パラメータ	説明	ページ
GetMemory	[in]ADDR _aBegin [in]ADDR _aEnd [in]WORD _wDisplayWidth [out]SAGEARRAY (BYTE)* _ppbyBuff	メモリの値を取得する	P51
SetMemory	[in]ADDR _aBegin [in]ADDR _aEnd [in]WORD _wDisplayWidth [in]SAGEARRAY (BYTE) _ppbyBuff	メモリに値を設定する	P52
GetDirectMemory	[in]ADDR _aBegin [in]ADDR _aEnd [in]WORD _wDisplayWidth [out]SAGEARRAY (BYTE)* _ppbyBuff	ダイレクトメモリを取得する	P53

5.1.4 ソフトウェアブレイク

メソッド名	パラメータ	説明	ページ
SetPCBreakPt	[in]ADDR _aPCBreakAddr [out]BHANDLE* p_Bhandle	ブレイクポイントを設定する	P54
EnableBreakPt	[in] BHANDLE p_Bhandle [in]VARIANT_BOOL bEnable	ブレイクポイントの有効/無効を設定する	P55
DeleteBreakPt	[in] BHANDLE BHandle	ブレイクポイントを削除する	P56
GetAllBreakPt	[out] long *p_index [out] VARIANT *p_vAllBreakPt	現在設定されているブレイクポイントを取得する	P57
DeleteAllBreakPt	(なし)	現在設定されているブレイクポイントを削除する	P58

5.1.5 変数ブレイク

メソッド名	パラメータ	説明	ページ
SetDataBreakpoint	[in]DWORD _aSymbol [in]DWORD _eSize [in]DWORD _eType [in]DWORD _dwData [out]DWORD *p_dwBreakDataNo	ブレイクデータポイントを設定する	P59
EnableDataBreakpoint	[in]DWORD dwBreakDataNo [in]VARIANT_BOOL _bEnable	ブレイクデータポイントの有効/無効を設定する	P60
DeleteDataBreakpoint	[in]DWORD dwBreakDataNo	ブレイクデータポイントを削除する	P61

5.1.6 変数トレース

メソッド名	パラメータ	説明	ページ
SetSymbolTrace	[in]ADDR _aSymbol [in]DWORD _eConditon [in]DWORD _eSize [in]DWORD _eType [in]DWORD _dwData [out]DWORD *p_dwTraceNo	変数トレース条件を設定する	P62
ExecuteSymbolTrace	[in]VARIANT_BOOL _bEnable	変数トレースの有効/無効を設定する	P63
DeleteSymbolTrace	[in]DWORD _dwTrace	変数トレース条件を削除する	P64
SaveSymbolTraceDeta	[in]BSTR _bstrFileName	変数トレースの結果を指定したファイルに保存する	P65

5.1.7 割り込み

メソッド名	パラメータ	説明	ページ
SendTrigger	[in]DWORD _dwTriggerNo [in]DWORD _dwTriggerType1 [in]DWORD _dwTriggerType2 [in]DWORD _dwPriority	割り込み条件を設定する	P66

5.1.8 シンボル操作

メソッド名	パラメータ	説明	ページ
GetRealTimeWatch	[in]ADDR _aSymbol [in]DWORD _eSize [out]DWORD *p_dwValue	リアルタイムウォッチ	P67
GetQuickWatch	[in]BSTR bstrVarName [out]DWORD* p_dwValueSize [out]BSTR* bstrByValue [out]EobjectTypeServer* p_eType [out]BSTR* bstrVariableAllocation	ラベル/シンボルの値を取得する	P68
SymbolToAddress	[in]BSTR bstrSymbolName [out]ADDR* p_aSymbolAddr	シンボルからアドレスに変換する	P69
AddressToSymbol	[in]ADDR aSymbolAddr [out]BSTR* p_bstrSymbolName	アドレスからシンボルに変換する	P70
GetLineFromAddr	[in]ADDR _aLineAddr [out]BSTR* p_bstrFileName [out]LINE0* p_LineNo	指定アドレスに該当する行番号を取得する	P71
GetAddrFromLine	[in]BSTR bstrFileName [in]LINE0 LineNo [out]ADDR* p_aLineAddr	指定行番号に該当するアドレスを取得する	P72

5.1.9 ダウンロード

メソッド名	パラメータ	説明	ページ
Download	[in]BSTR _bstrFileName	ターゲットプログラムをダウンロードする	P73
Unload	[in]BSTR _bstrFileName	ターゲットプログラムをアンロードする	P74

5.1.10 起動/終了

メソッド名	パラメータ	説明	ページ
InvokeHew	(なし)	High-performance Embedded Workshop アプリケーションを起動する	P75
QuitHew	(なし)	High-performance Embedded Workshop アプリケーションを終了する	P76

5.1.11 ワークスペース

メソッド名	パラメータ	説明	ページ
OpenWorkspace	[in]BSTR _bstrFileName	ワークスペースを開く	P77
CloseWorkspace	[in]DWORD _dwIgnoreChanges	ワークスペースを閉じる	P78
SaveWorkspace	(なし)	ワークスペースを保存する	P79

5.1.12 構成とセッション

メソッド名	パラメータ	説明	ページ
SaveSession	(なし)	セッションファイルを保存する	P80
GetCurrentConfiguration	[out]BSTR *p_bstrCurrentConfigurationName	現在のビルド構成を取得する	P81
SetCurrentConfiguration	[in]BSTR _bstrConfiguration	ビルド構成を設定する	P82
GetConfigurations	[out]BSTR *p_strConfigurations	登録されているビルド構成を取得する	P83
GetCurrentSession	[out]BSTR *p_bstrCurrentSessionName	現在のデバッグセッションを取得する	P84
SetCurrentSession	[in]BSTR _bstrSession	デバッグセッションを設定する	P85
GetSessions	[out]BSTR *p_bstrSessions	登録されているデバッグセッションを取得する	P86
GetCurrentProject	[out]BSTR *p_bstrCurrentProjectName	現在のプロジェクトを取得する	P87
SetCurrentProject	[in] BSTR _bstrProjectName	アクティブプロジェクトを設定する	P88
GetProjects	[out]BSTR *p_bstrProjectNames	プロジェクト名をすべて取得する	P89

5.1.13 プロジェクト

メソッド名	パラメータ	説明	ページ
AddFile	[in]BSTR _bstrFileName	プロジェクトにファイルを追加する	P90
AddFiles	[in]BSTR _bstrFileName	プロジェクトに複数のファイルを追加する	P91
DeleteFile	[in]BSTR _bstrFileName	プロジェクトからファイルを削除する	P92
DeleteFiles	[in]BSTR _bstrFileName	プロジェクトから複数のファイルを削除する	P93

5.1.14 ビルド関係

メソッド名	パラメータ	説明	ページ
BuildProject	(なし)	プロジェクトをビルドする	P94
RebuildProject	(なし)	プロジェクトをリビルドする	P95
UpdateAllDependency	(なし)	すべての依存関係を更新する	P96
AddFileWithCompilerOption	[in] BSTR _bstrFileName [in] BSTR _bstrIncludeDirectories [in] BSTR _bstrDefines	プロジェクトにコンパイラオプションを設定してファイルを追加する	P97

5.1.15 ファイル関係

メソッド名	パラメータ	説明	ページ
OpenFileAtLine	[in] BSTR _bstrOpenFileName [in] int _iLine	ファイル名、行番号を指定しファイルを開く	P98
GetSourceFiles	[out]BSTR *p_bstrSourceFiles	ソースファイル名を取得する	P99
GetDownloadModules	[out]BSTR *p_bstrDownloadModules	モジュールファイル名を取得する	P100
GetDependentFiles	[out]BSTR *p_bstrDependentFiles	依存ファイル名を取得する	P101

5.1.16 カバレッジ関係

メソッド名	パラメータ	説明	ページ
SetCoverageRange	[in] DWORD dwStartAddress [in] DWORD dwEndAddress	カバレッジの範囲を設定する	P102
GetCoverageRange	[out]DWORD *p_dwStartAddress [out] DWORD *p_dwEndAddress	カバレッジの範囲を取得する	P103
SetCoverageDisable	(なし)	カバレッジを無効に設定する	P104
SetCoverageEnable	(なし)	カバレッジを有効に設定する	P105
ClearCoverage	(なし)	カバレッジ情報をクリアする	P106
GetCoverageStatus	[out] int *p_iStatus	カバレッジの状態を取得する	P107
LoadCoverage	[in]BSTR _bstrLoadFileName	カバレッジ情報をロードする	P108
SaveCoverage	[in]BSTR _bstrSaveFileName	カバレッジ情報をセーブする	P109

5.1.17 その他

メソッド名	パラメータ	説明	ページ
GetErrorString	[in]HRESULT _IError [out]BSTR* _pbstrError	メソッド呼び出しで発生したエラー文字列を取得する	P110
GetHewStatus	[out]int* p_iTargetReset [out]int* p_iTagetExecStatus [out]int* p_iMemoryReset [out]int* p_iRegisterReset [out]int* iPlatformInitialize [out]int* p_iLoadingStatus	ステータスを取得する	P111
GetHewStatusEx	[out] int *p_iInvokeHew [out] int *p_iOpenWorkspace [out] int *p_iBuildProject	ステータス(起動, ワークスペースオープン, ビルド)を取得する	P112
GetTargetName	[out] BSTR* p_bstrName	ターゲット名を取得する	P113

5.2 メソッド概要 (VB、VC++用)

5.2.1 実行制御

メソッド名	パラメータ	説明	ページ
GoTargetExec2	(なし)	プログラムを実行する	P114
StopTargetExec2	(なし)	プログラムを停止する	P115
ResetTargetExec2	(なし)	プログラムをリセットする	P116
InitializeTarget2	(なし)	ターゲットを初期化する	P117
Step2	[in] long _lMode [in] long _lStep	プログラムをステップ実行する	P118
StepRate2	[in] long _lRate	ステップ実行速度を設定する	P119
StepOver2	[in] long _lMode [in] long _lStep	プログラムをステップオーバーで実行する	P120
StepOut2	[in] long _lMode	プログラムをステップアウトで実行する	P121
IsRunning2	[out] long *p_lRunning	プログラムが実行中かチェックする	P122

5.2.2 レジスタ操作

メソッド名	パラメータ	説明	ページ
GetPC2	[out] long *p_lPC	プログラムカウンタの値を取得する	P123
SetPCAddress2	[in] long lPCAddr	プログラムカウンタの値をアドレスで設定する	P124
SetPCSource2	[in] BSTR bstrFileName [in] long lLineNum	プログラムカウンタの値を行番号で設定する	P125
TestSetPC2	[out] long *p_lSetPCState	PC設定のチェックをする	P126

5.2.3 メモリ操作

メソッド名	パラメータ	説明	ページ
GetMemory2	[in] long lBegin [in] long lEnd [in] long lDisplayWidth [out] VARIANT *p_vMemData	メモリの値を取得する	P127
SetMemory2	[in] long lBegin [in] long lEnd [in] long lDisplayWidth [in] VARIANT vMemData	メモリに値を設定する	P128
GetDirectMemory2	[in] long lBegin [in] long lEnd [in] long lDisplayWidth [out] VARIANT *p_vMemData	ダイレクトメモリを取得する	P130

5.2.4 ソフトウェアブレーク

メソッド名	パラメータ	説明	ページ
SetPCBreakPt2	[in] long lPCBreakAddr [out] long *p_lHandle	ブレークポイントを設定する	P131
EnableBreakPt2	[in] long lHandle [in] long lEnable	ブレークポイントの有効/無効を設定する	P132
DeleteBreakPt2	[in] long lHandle	ブレークポイントを削除する	P133
GetAllBreakPt2	[out] long *p_index [out] VARIANT *p_vAllBreakPt	現在設定されているブレークポイントを取得する	P134
DeleteAllBreakPt2	(なし)	現在設定されているブレークポイントを削除する	P135

5.2.5 変数ブレーク

メソッド名	パラメータ	説明	ページ
SetDataBreakpoint2	[in] long _lSymbol [in] long _lSize [in] long _lType [in] long _lData [out] long *p_lBreakDataNo	ブレークデータポイントを設定する	P136
EnableDataBreakpoint2	[in] long lDataBreakNo [in] long _lEnable	ブレークデータポイントの有効/無効を設定する	P137
DeleteDataBreakpoint2	[in] long lDataBreakNo	ブレークデータポイントを削除する	P138

5.2.6 変数トレース

メソッド名	パラメータ	説明	ページ
SetSymbolTrace2	[in] long _lSymbol [in] long _lCondition [in] long _lSize [in] long _lType [in] long _lData [out] long *p_lTraceNo	変数トレース条件を設定する	P139
ExecuteSymbolTrace2	[in] long _lEnable	変数トレースの有効/無効を設定する	P141
DeleteSymbolTrace2	[in] long _lTraceNo	変数トレース条件を削除する	P142
SaveSymbolTraceData2	[in] BSTR _bstrFileName	変数トレースの結果を指定したファイルに保存する	P143

5.2.7 割り込み

メソッド名	パラメータ	説明	ページ
SendTrigger2	[in] long _lTriggerNo [in] long _lTriggerType1 [in] long _lTriggerType2 [in] long _lPriority	割り込み条件を設定する	P144

5.2.8 シンボル操作

メソッド名	パラメータ	説明	ページ
GetRealTimeWatch2	[in] long _lSymbol [in] long _lSize [out] long *p_lValue	リアルタイムウォッチ	P145
GetQuickWatch2	[in] BSTR bstrVarName [out] long *p_lValueSize [out] BSTR *bstrByValue [out] long *p_lType [out] BSTR *bstrTypeName [out] BSTR *bstrVarAllocation	ラベル/シンボルの値を取得する	P146
SymbolToAddress2	[in] BSTR bstrSymbolName [out] long *p_lSymbolAddr	シンボルからアドレスに変換する	P147
AddressToSymbol2	[in] long lSymbolAddr [out] BSTR *p_bstrSymbolName	アドレスからシンボルに変換する	P148
GetLineFromAddr2	[in] long lLineAddr [out] BSTR *p_bstrFileName [out] long *p_lLineNo	指定アドレスに該当する行番号を取得する	P149
GetAddrFromLine2	[in] BSTR bstrFileName [in] long lLineNo [out] long *p_lLineAddr	指定行番号に該当するアドレスを取得する	P150

5.2.9 ダウンロード

メソッド名	パラメータ	説明	ページ
Download2	[in] BSTR _bstrFileName	ターゲットプログラムをダウンロードする	P151
Unload2	[in] BSTR _bstrFileName	ターゲットプログラムをアンロードする	P152

5.2.10 起動/終了

メソッド名	パラメータ	説明	ページ
InvokeHew2	(なし)	High-performance Embedded Workshop アプリケーションを起動する	P153
QuitHew2	(なし)	High-performance Embedded Workshop アプリケーションを終了する	P154
InvokeHewWithNoDialog	(なし)	ようこそダイアログボックスを表示せずに High-performance Embedded Workshop アプリケーションを起動する	P155

5.2.11 ワークスペース

メソッド名	パラメータ	説明	ページ
OpenWorkspace2	[in] BSTR _bstrFileName	ワークスペースを開く	P156
CloseWorkspace2	[in] long _lIgnoreChanges	ワークスペースを閉じる	P157
SaveWorkspace2	(なし)	ワークスペースを保存する	P158
GetWorkSpaceDirectory	[out] BSTR *_pbstrCurrentWorkspaceDirectory	ワークスペースの絶対パスを取得する	P159

5.2.12 構成とセッション

メソッド名	パラメータ	説明	ページ
SaveSession2	(なし)	セッションファイルを保存する	P160
GetCurrentConfiguration2	[out] BSTR *p_bstrCurrentConfigurationName	現在のビルド構成を取得する	P161
SetCurrentConfiguration2	[in] BSTR _bstrConfiguration	ビルド構成を設定する	P162
GetConfigurations2	[out] BSTR *p_bstrConfigurations	登録されているビルド構成を取得する	P163
GetCurrentSession2	[out] BSTR *p_bstrCurrentSessionName	現在のデバッグセッションを取得する	P164
SetCurrentSession2	[in] BSTR _bstrSession	デバッグセッションを設定する	P165
GetSessions2	[out] BSTR *p_bstrSessions	登録されているデバッグセッションを取得する	P166
GetCurrentProject2	[out] BSTR *p_bstrCurrentProjectName	現在のプロジェクトを取得する	P167
SetCurrentProject2	[in] BSTR _bstrProjectName	アクティブプロジェクトを設定する	P168
GetProjects2	[out] BSTR *p_bstrProjectNames	プロジェクト名をすべて取得する	P169

5.2.13 プロジェクト

メソッド名	パラメータ	説明	ページ
AddFile2	[in] BSTR _bstrFileName	プロジェクトにファイルを追加する	P170
AddFiles2	[in] BSTR _bstrFileName	プロジェクトに複数のファイルを追加する	P171
DeleteFile2	[in] BSTR _bstrFileName	プロジェクトからファイルを削除する	P172
DeleteFiles2	[in] BSTR _bstrFileName	プロジェクトから複数のファイルを削除する	P173
AddProjectFileFolder	[in] BSTR _bstrFolderName	Projects ツリーにフォルダを追加する	P174
RemoveProjectFileFolder	[in] BSTR _bstrFolderName	Projects ツリーのフォルダを削除する	P175
AddFileToFolder	[in] BSTR _bstrFileName [in] BSTR _bstrFolderName	Projects ツリーにある指定したフォルダにファイルを登録する	P176

5.2.14 ビルド関係

メソッド名	パラメータ	説明	ページ
BuildProject2	(なし)	プロジェクトをビルドする	P177
RebuildProject2	(なし)	プロジェクトをリビルドする	P178
UpdateAllDependency2	(なし)	すべての依存関係を更新する	P179
AddFileWithCompilerOption2	[in] BSTR _bstrAddFileName [in] BSTR _bstrIncludeDirectories [in] BSTR _bstrDefines	プロジェクトにコンパイラオプションを設定してファイルを追加する	P180
GetLibraryOptions	[out] BSTR *p_bstrLibraryOption	カレントプロジェクトのリンクのライブラリオプションを取得する	P181
SetLibraryOptions	[in] BSTR _bstrLibraryOption	カレントプロジェクトのリンクのライブラリオプションを設定する	P182
GetLibraryFilesForConfiguration	[in] BSTR _bstrProjectName [in] BSTR _bstrConfiguration [out] BSTR *_pbstrLibraryFiles	指定したプロジェクトの指定したコンフィグレーションからライブラリオプションを取得する	P183
SetLibraryFilesForConfiguration	[in] BSTR _bstrProjectName [in] BSTR _bstrConfiguration [in] BSTR _bstrLibraryFiles	指定したプロジェクトの指定したコンフィグレーションにライブラリオプションを設定する	P184
GetIncludeFileDirectories	[in] BSTR _bstrProjectName [in] BSTR _bstrConfiguration [in] BSTR _bstrFileName [out] VARIANT *_pvtIncludeDirectories	指定したプロジェクトの指定したコンフィグレーションにあるファイルからインクルードファイルオプションを取得する	P185
SetIncludeFileDirectories	[in] BSTR _bstrProjectName [in] BSTR _bstrConfiguration [in] BSTR _bstrFileName [in] VARIANT _vtIncludeDirectories [in] long _lSettingMode	指定したプロジェクトの指定したコンフィグレーションにあるファイルにインクルードファイルオプションを設定する	P186
GetCpuAndToolChainData	[in] BSTR _bstrProjectName [out] BSTR *_pbstrCPUFamily [out] BSTR *_pbstrCPUSeries [out] BSTR *_pbstrCPUType [out] BSTR *_pbstrToolChainFamily [out] BSTR *_pbstrToolChainName [out] BSTR *_pbstrToolChainVersion	指定したプロジェクトから CPU 種別、CPU シリーズ名、CPU タイプ名、コンパイラ種別、コンパイラ名、コンパイラバージョンを取得する	P187
SetBuildExcludeFiles	[in] BSTR _bstrFileNames	指定したファイルをビルドから除外する	P189
SetBuildIncludeFiles	[in] BSTR _bstrFileNames	指定したファイルをビルド対象に含める	P190

5.2.15 ファイル関係

メソッド名	パラメータ	説明	ページ
OpenFileAtLine2	[in] BSTR _bstrOpenFileName [in] long _lLine	ファイル名、行番号を指定しファイルを開く	P191
GetSourceFiles2	[out] BSTR *p_bstrSourceFiles	ソースファイル名を取得する	P192
GetDownloadModules2	[out] BSTR *p_bstrDownloadModules	モジュールファイル名を取得する	P193
GetDependentFiles2	[out] BSTR *p_bstrDependentFiles	依存ファイル名を取得する	P194

5.2.16 カバレッジ関係

メソッド名	パラメータ	説明	ページ
SetCoverageRange2	[in] long _lStartAddress [in] long _lEndAddress	カバレッジの範囲を設定する	P195
GetCoverageRange2	[out] long *p_lStartAddress [out] long *p_lEndAddress	カバレッジの範囲を取得する	P196
SetCoverageDisable2	(なし)	カバレッジを無効に設定する	P197
SetCoverageEnable2	(なし)	カバレッジを有効に設定する	P198
ClearCoverage2	(なし)	カバレッジ情報をクリアする	P199
GetCoverageStatus2	[out] long *p_lStatus	カバレッジの状態を取得する	P200
LoadCoverage2	[in] BSTR _bstrLoadFilename	カバレッジ情報をロードする	P201
SaveCoverage2	[in] BSTR _bstrSaveFilename	カバレッジ情報をセーブする	P202

5.2.17 その他

メソッド名	パラメータ	説明	ページ
GetErrorString2	[in] long lError [out] BSTR *p_bstrError	メソッド呼び出しで発生したエラー文字列を取得する	P203
GetHewStatus2	[out] long *p_lTargetReset [out] long *p_lTargetExecStatus [out] long *p_lMemoryReset [out] long *p_lRegisterReset [out] long *p_lLinkStatus [out] long *p_lPlatformInitialize [out] long *p_lLoadingStatus	ステータスを取得する	P204
GetHewStatusEx2	[out] long *p_lInvokeHew [out] long *p_lOpenWorkspace [out] long *p_lBuildProject	ステータス(起動、ワークスペースオープン、ビルド)を取得する	P206
GetTargetName2	[out] BSTR *p_bstrName	ターゲット名を取得する	P207
GetHewVersion	[out] BSTR *p_bstrHewVersion	High-performance Embedded Workshop のバージョンを取得する	P208
Command	[in] BSTR _bstrCommandLine [out] BSTR *p_bCommandMessage	High-performance Embedded Workshop のコマンドを実行する	P209

5.3 メソッドの詳細 (VC++のみ使用可)

5.3.1 実行制御

GoTargetExec

内容

現プログラム位置より、プログラムを実行します。

パラメータ

パラメータはありません。

戻り値

戻り値は、HRESULT 型で返します。成功した場合は0を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例

```
HRESULT hr = E_FAIL;

try
{
    hr = pHewServer1->GoTargetExec();
}
}
```

StopTargetExec

内容

プログラムの実行を停止します。

パラメータ

パラメータはありません。

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例

```
HRESULT hr = E_FAIL;

try
{
    hr = pHewServer1->StopTargetExec();
}
```

ResetTargetExec

内容

ターゲットとしているデバッグ環境をリセットします。

パラメータ

パラメータはありません。

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例

```
HRESULT hr = E_FAIL;

try
{
    hr = pHewServer1->ResetTargetExec();
}
```

InitializeTarget

内容

ターゲットとしているデバッグ環境を初期化します。

パラメータ

パラメータはありません。

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例

```
HRESULT hr = E_FAIL;

try
{
    hr = pHewServer1->InitializeTarget();
}
```

Step

内容

プログラムをステップ実行します。

パラメータ

属性	型	内容
[in]	DWORD _eMode	ステップの種類 0x00000001 アセンブラの命令でステップ実行 0x00000002 ソースコードの行でステップ実行
[in]	DWORD _dwStep	実行するステップ数

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例

```
HRESULT hr = E_FAIL;
DWORD _eMode = 1;           //assembler:1, source:2
DWORD _dwStep;

try
{
    hr = pHewServer1->Step(_eMode, _dwStep);
}
```

StepRate

内容

ステップの実行速度を設定します。

パラメータ

属性	型	内容
[in]	DWORD _dwRate	ステップレート 0~6 の値を設定します。 0 : 3 seconds 1 : 2.5 seconds 2 : 2 seconds 3 : 1.5 seconds 4 : 1 seconds 5 : 0.5 seconds 6 : 0 seconds

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例

```
HRESULT          hr = E_FAIL;
DWORD _dwRate;

try
{
    hr = pHewServer1->StepRate(_dwRate);
}
```

StepOver

内容

ステップオーバーでプログラムを実行します。

パラメータ

属性	型	内容
[in]	DWORD _eMode	ステップの種類 0x00000001 アセンブラの命令でステップ実行 0x00000002 ソースコードの行でステップ実行
[in]	DWORD _dwStep	ステップ数

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例

```
HRESULT hr = E_FAIL;
DWORD _eMode = 1;           //assembler:1, source:2
DWORD _dwStep;

try
{
    hr = pHewServer1->StepOver(_eMode, _dwStep);
}
```

StepOut

内容

プログラムをステップアウトで実行します。

パラメータ

属性	型	内容
[in]	DWORD _eMode	ステップの種類 0x00000001 アセンブラの命令でステップ実行 0x00000002 ソースコードの行でステップ実行

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例

```
HRESULT hr = E_FAIL;
DWORD _eMode = 1;           //assembler:1, source:2

try
{
    hr = pHewServer1->StepOut(_eMode);
}
```

IsRunning

内容

ユーザプログラム実行中か否かの判定をします。

パラメータ

属性	型	内容
[out]	long* p_bRunning	ユーザプログラム実行中のときは1、それ以外は0

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例

```
HRESULT hr = E_FAIL;
long _bRunning;

try
{
    hr = pHewServer1->IsRunning(&_bRunning);
}
}
```

5.3.2 レジスタ操作

GetPC

内容

現在のプログラムカウンタの値を取得します。

パラメータ

属性	型	内容
[out]	DWORD *p_dwPC	PC (プログラムカウンタ) 値

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外 (例えば E_FAIL (0x80004005L)) を返します。

記述例

```
HRESULT hr = E_FAIL;
DWORD _dwPC = 0x0;

try
{
    hr = pHewServer1->GetPC(&_dwPC);
}
```

SetPCAddress

内容

プログラムカウンタを設定します。

パラメータ

属性	型	内容
[in]	ADDR aPCAddr	設定する PC (プログラムカウンタ) 値

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外 (例えば E_FAIL (0x80004005L)) を返します。

記述例

```
HRESULT hr = E_FAIL;
ADDR aPCAddr;

try
{
    hr = pHewServer1->SetPCAddress(aPCAddr);
}
}
```

SetPCSource

内容

ソースファイル、行を指定しプログラムカウンタを設定します。

パラメータ

属性	型	内容
[in]	BSTR bstrFileName	ファイル名
[in]	LINENO LineNum	行番号

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例

```
HRESULT hr = E_FAIL;
BSTR bstrSetPCSourceFile;
LINENO LineNum;

try
{
    hr = pHewServer1->SetPCSource(bstrSetPCSourceFile, LineNum);
}
```

TestSetPC

内容

プログラムカウンタに対して値の設定が可能か否かを判定します。

パラメータ

属性	型	内容
[out]	long* p_bSetPCState	PC 値設定可能なときは 1、それ以外は 0

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例

```
long    lSetPC = NULL;
HRESULT hr = E_FAIL;

try
{
    hr = pHewServer1->TestSetPC(&lSetPC);
}
```

5.3.3 メモリ操作

GetMemory

内容

指定した開始アドレス、終了アドレス、およびアクセスサイズによりメモリ内容を取得します。High-performance Embedded Workshop 内部で指定エリアのメモリ内容のデータを保持している場合は、ターゲットメモリへアクセスせずに High-performance Embedded Workshop 内部で保持しているメモリ内容を返却します。

パラメータ

属性	型	内容
[in]	ADDR _aBegin	メモリ内容取得の開始アドレス
[in]	ADDR _aEnd	メモリ内容取得の終了アドレス
[in]	WORD _wDisplayWidth	メモリへアクセスするサイズ(1, 2, 4, 8 が指定可能)
[out]	SAFEARRAY (BYTE)* _ppbyBuff	メモリ内容

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例

```
HRESULT hr = E_FAIL;
DWORD dwAddrBegin = (DWORD)strtol(m_GetMemoryStartAddress, NULL, 16);
DWORD dwAddrEnd = (DWORD)strtol(m_GetMemoryEndAddress, NULL, 16);
WORD wDisplayWidth = (WORD)m_GetMemorySize.GetCurSel();

if (wDisplayWidth == 0) wDisplayWidth = 1;
else if(wDisplayWidth == 1) wDisplayWidth = 2;
else if(wDisplayWidth == 2) wDisplayWidth = 4;
else if(wDisplayWidth == 3) wDisplayWidth = 8;
else wDisplayWidth = 1;

BYTE bTemp;

//array for storing data obtained from HewTargetServer
...
SAFEARRAY FAR* pHewArray = NULL;

try
{
    hr = pHewServer1->GetMemory(dwAddrBegin, dwAddrEnd, wDisplayWidth, &pHewArray);
}
}
```

SetMemory

内容

指定した開始アドレス、終了アドレス、およびアクセスサイズによりメモリ内容を設定します。

パラメータ

属性	型	内容
[in]	ADDR _aBegin	メモリ内容取得の開始アドレス
[in]	ADDR _aEnd	メモリ内容取得の終了アドレス
[in]	WORD _wDisplayWidth	メモリへアクセスするサイズ(1, 2, 4, 8 が指定可能)
[in]	SAFEARRAY (BYTE)* _ppbyBuff	メモリ内容

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例

```
HRESULT          hr = E_FAIL;

dwAddrBegin = (DWORD)strtol(m_SetMemoryStartAddress, NULL, 16);
dwAddrEnd = (DWORD)strtol(m_SetMemoryEndAddress, NULL, 16);

...

try
{
    hr = pHewServer1->SetMemory(dwAddrBegin, dwAddrEnd, wDisplayWidth, pHewArray);
}
}
```

GetDirectMemory

内容

指定した開始アドレス、終了アドレス、およびアクセスサイズによりメモリ内容を取得します。High-performance Embedded Workshop 内部で本指定エリアのメモリ内容のデータを保持している、いないにかかわらず、ターゲットメモリへアクセスしてメモリ内容を返却します。

パラメータ

属性	型	内容
[in]	ADDR _aBegin	メモリ内容取得の開始アドレス
[in]	ADDR _aEnd	メモリ内容取得の終了アドレス
[in]	WORD _wDisplayWidth	メモリへアクセスするサイズ(1, 2, 4, 8 が指定可能)
[out]	SAFEARRAY (BYTE)* _ppbyBuff	メモリ内容

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例

```
HRESULT hr = E_FAIL;
SAFEARRAY FAR* pHewArray = NULL;

...

try
{
    hr = pHewServer1->GetDirectMemory(dwAddrBegin, dwAddrEnd, wDisplayWidth, &pHewArray);
}
}
```

5.3.4 ソフトウェアブレイク

SetPCBreakPt

内容

指定アドレスにブレイクポイントを設定し、そのハンドル値を返却します。

パラメータ

属性	型	内容
[in]	ADDR _aPCBreakAddr	アドレス値
[out]	BHANDLE* p_BHandle	ブレイクポイントのハンドル値

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例

```
HRESULT hr = E_FAIL;
ADDR _aPCBreakAddr;
BHANDLE Bhandle;

try
{
    hr = pHewServer1->SetPCBreakPt(dwAddr, &Bhandle);
}
}
```

EnableBreakPt

内容

ブレイクポイントのハンドル値から、ブレイクポイントの有効化、無効化を設定します。

パラメータ

属性	型	内容
[in]	BHANDLE BHandle	ブレイクポイントのハンドル値
[in]	VARIANT_BOOL bEnable	有効化 (VARIANT_TRUE), 無効化 (VARIANT_FALSE)

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例

```
HRESULT          hr = E_FAIL;
BHANDLE BHandle;
VARIANT_BOOL    bEnable;

try
{
    hr = pHewServer1->EnableBreakPt(BHandle, bEnable);
}
```

DeleteBreakPt

内容

ブレークポイントのハンドル値のブレークポイントを削除します。

パラメータ

属性	型	内容
[in]	BHANDLE BHandle	ブレークポイントのハンドル値

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例

```
HRESULT          hr = E_FAIL;
BHANDLE BHandle;

try
{
    hr = pHewServer1->DeleteBreakPt(BHandle);
}
}
```

GetAllBreakPt

内容

現在設定されているブレイクポイントを取得します。

パラメータ

属性	型	内容
[out]	long *p_index	ブレイクポイントの数
[out]	VARIANT *p_vAllBreakPt	ブレイクポイントの配列

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例

```
HRESULT hr = E_FAIL;
long index;
VARIANT vAllBreakPt;
VariantInit( &vAllBreakPt );

//calling HewTargetServer function
try
{
    hr = pHewServer1->GetAllBreakPt(&index, &vAllBreakPt);
}
```

DeleteAllBreakPt

内容

現在設定されているブレイクポイントを削除します。

パラメータ

パラメータはありません。

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例

```
HRESULT    hr = E_FAIL;

//calling HewTargetServer function
try
{
    hr = pHewServer1->DeleteAllBreakPt();
}
```

5.3.5 変数ブレーク

SetDataBreakpoint

内容

ブレークデータポイントを設定します。

パラメータ

属性	型	内容
[in]	DWORD _aSymbol	シンボルのアドレス
[in]	DWORD _eSize	シンボルのサイズ(1/2/4) 0x00000001 - 1 0x00000002 - 2 0x00000004 - 4
[in]	DWORD _eType	ブレークする種類(Equal/Not Equal) 0x00000001 - Equal 0x00000002 - Not Equal
[in]	DWORD _dwData	シンボルの値
[out]	DWORD *p_dwBreakDataNo	変数ブレークの番号

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例

```
HRESULT hr = E_FAIL;
DWORD _aSymbol;
DWORD _eSize;
DWORD _eType;
DWORD _dwData;
DWORD _dwBreakDataNo;

...

try
{
    hr = pHewServer1->SetDataBreakpoint(_aSymbol, _eSize, _eType, _dwData,
&_dwBreakDataNo);
}
```

EnableDataBreakpoint

内容

ブレークデータポイントの有効/無効を設定します。

パラメータ

属性	型	内容
[in]	DWORD dwBreakDataNo	変数ブレークの番号
[in]	VARIANT_BOOL _bEnable	有効(True)/ 無効(False)

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例

```
HRESULT hr = E_FAIL;
DWORD dwBreakDataNo;
VARIANT_BOOL bEnable;

try
{
    hr = pHewServer1->EnableDataBreakpoint(dwBreakDataNo, bEnable);
}
}
```

DeleteDataBreakpoint

内容

ブレークデータポイントを削除します。

パラメータ

属性	型	内容
[in]	DWORD dwBreakDataNo	変数ブレークの番号

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例

```
HRESULT hr = E_FAIL;
DWORD dwBreakDataNo;

try
{
    hr = pHewServer1->DeleteDataBreakpoint(dwBreakDataNo);
}
}
```

5.3.6 変数トレース

SetSymbolTrace

内容

変数トレース条件を設定します。

パラメータ

属性	型	内容
[in]	ADDR _aSymbol	シンボルのアドレス
[in]	DWORD _eCondition	トレース条件(Read/Write) 0x00000001 - Read 0x00000002 - Write 0x00000003 - Read_Write
[in]	DWORD _eSize	シンボルのサイズ(1/2/4) 0x00000001 - 1 0x00000002 - 2 0x00000004 - 4
[in]	DWORD _eType	トレースする種類(Equal/Not Equal/No Specific) 0x00000001 - Equal 0x00000002 - Not Equal 0x00000003 - Not Specified
[in]	DWORD _dwData	シンボルの値
[out]	DWORD * p_dwTraceNo	変数トレースの番号

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例

```
HRESULT hr = E_FAIL;
ADDR _aSymbol;
DWORD _eSize;
DWORD _eType;
DWORD _dwData;
DWORD _dwTraceNo;

...

try
{
    hr = pHewServer1->SetSymbolTrace(aSymbol, 0x00000001, _eSize, _eType, _dwData, &
    _dwTraceNo);
}
```

ExecuteSymbolTrace

内容

変数トレースの有効/無効を設定します。

パラメータ

属性	型	内容
[in]	VARIANT_BOOL _bEnable	有効(True)/無効(False)

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例

```
HRESULT hr = E_FAIL;
VARIANT_BOOL _bEnable;

try
{
    hr = pHewServer1->ExecuteSymbolTrace(_bEnable);
}
}
```

DeleteSymbolTrace

内容

変数トレース条件を削除します。

パラメータ

属性	型	内容
[in]	DWORD _dwTraceNo	削除する変数トレースの番号

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例

```
HRESULT hr = E_FAIL;
DWORD _dwTraceNo;

try
{
    hr = pHewServer1->DeleteSymbolTrace(_dwTraceNo);
}
}
```

SaveSymbolTraceData

内容

変数トレースの結果を指定したファイルに保存します。

パラメータ

属性	型	内容
[in]	BSTR _bstrFileName	変数トレースデータの保存ファイル

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例

```
HRESULT          hr = E_FAIL;
BSTR bstrSaveSymbolTraceData;

try
{
    hr = pHewServer1->SaveSymbolTraceData(bstrSaveSymbolTraceData);
}
}
```

出力フォーマット例

トレース結果は、以下の内容をスペースで区切って出力します。

- アクセスした時間(シミュレータではサイクル数)
- アクセスしたアドレス
- アクセス属性(Read/Write/Read_Write)
- アクセス値
- アクセスサイズ

サンプル

```
1287539 0XFFFE5DC Write 0XEA 1
1287553 0XFFFE5DC Write 0X30 1
1288170 0XFFFE5DC Write 0XEA 1
1445327 0XFFFE5DC Write 0XE0 1
1445341 0XFFFE5DC Write 0X30 1
1445958 0XFFFE5DC Write 0XE0 1
1605377 0XFFFE5DC Write 0X4C 1
1605391 0XFFFE5DC Write 0X30 1
1606008 0XFFFE5DC Write 0X4C 1
1760876 0XFFFE5DC Write 0XF6 1
```

5.3.7 割り込み

SendTrigger

内容

割り込み条件を設定します。

パラメータ

属性	型	内容
[in]	DWORD _dwTriggerNo	トリガー番号
[in]	DWORD _dwTriggerType1	トリガー割り込み条件 1
[in]	DWORD _dwTriggerType2	トリガー割り込み条件 2
[in]	DWORD _dwPriority	割り込み優先順位 (0~17)

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例

```
HRESULT hr = E_FAIL;

DWORD _dwTriggerNo;
DWORD _dwTriggerType1;
DWORD _dwTriggerType2;
DWORD _dwPriority

try
{
    hr = pHewServer1->SendTrigger(
        _dwTriggerNo,
        _dwTriggerType1,
        _dwTriggerType2,
        _dwPriority
    );
}
```

5.3.8 シンボル操作

GetRealTimeWatch

内容

指定したデータ値を取得します。

パラメータ

属性	型	内容
[in]	ADDR _aSymbol	シンボル名のアドレス
[in]	DWORD _eSize	シンボルのサイズ(1/2/4) 0x00000001 - 1 0x00000002 - 2 0x00000004 - 4
[out]	DWORD *p_dwValue	シンボルの値

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例

```
HRESULT hr = E_FAIL;
ADDR _aSymbol;
DWORD _eSize;
DWORD p_dwValue;

try
{
    hr = GetRealTimeWatch( aSymbol, eSize, &p_dwValue);
}
}
```

GetQuickWatch

内容

変数名から、変数のサイズ、変数値、型、割付領域を取得します。

パラメータ

属性	型	内容
[in]	BSTR bstrVarName	変数名
[out]	DWORD* p_dwValueSize	変数のサイズ
[out]	BSTR* bstrByValue	変数値の文字列
[out]	EObjectTypeServer* p_eType	変数型
[out]	BSTR* bstrTypeName	変数型の文字列
[out]	BSTR* bstrVariableAllocation	変数割付領域の文字列

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例

```
HRESULT hr = E_FAIL;
BSTR bstrVarName;
DWORD dwValueSize;
BSTR bstrByValue;
EObjectTypeServer _eType;
BSTR bstrTypeName;
BSTR bstrVariableAllocation;

try
{
    hr = pHewServer1->GetQuickWatch(bstrVarName,
        &dwValueSize,
        &bstrByValue,
        &_eType,
        &bstrTypeName,
        &bstrVariableAllocation
    );
}
}
```

SymbolToAddress

内容

シンボル名から対応するアドレス値へ変換します。

パラメータ

属性	型	内容
[in]	BSTR bstrSymbolName	シンボル名
[out]	ADDR* p_aSymbolAddr	アドレス

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例

```
HRESULT hr = E_FAIL;
BSTR bstrSymbolName;
ADDR _aSymbolAddr;

try
{
    hr = pHewServer1->SymbolToAddress(bstrSymbolName, &_aSymbolAddr);
}
}
```

AddressToSymbol

内容

アドレス値から対応するシンボル名へ変換します。

パラメータ

属性	型	内容
[in]	ADDR aSymbolAddr	アドレス
[out]	BSTR* p_bstrSymbolName	シンボル名

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例

```
HRESULT hr = E_FAIL;
ADDR aSymbolAddr;
BSTR bstrSymbolName;

try
{
    hr = pHewServer1->AddressToSymbol(aSymbolAddr, &bstrSymbolName);
}
}
```

GetLineFromAddr

内容

アドレス値から対応するファイル、行へ変換します。

パラメータ

属性	型	内容
[in]	ADDR _aLineAddr	アドレス
[out]	BSTR* p_bstrFileName	ファイル名
[out]	LINENO* p_LineNo	行番号

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例

```
HRESULT hr = E_FAIL;
ADDR _aLineAddr;
BSTR bstrFileName;
LINENO _LineNo;

try
{
    hr = pHewServer1->GetLineFromAddr(_aLineAddr, &bstrFileName, &_LineNo);
}
}
```

GetAddrFromLine

内容

ファイル、行から対応するアドレス値へ変換します。

パラメータ

属性	型	内容
[in]	BSTR bstrFileName	ファイル名
[in]	LINENO LineNo	行番号
[out]	ADDR* p_aLineAddr	アドレス

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例

```
HRESULT hr = E_FAIL;
BSTR bstrFileName;
LINENO LineNo;
ADDR _aLineAddr;

try
{
    hr = pHewServer1->GetAddrFromLine(bstrFileName, LineNo, &_aLineAddr);
}
}
```

5.3.9 ダウンロード

Download

内容

ロードモジュールをダウンロードします。

パラメータ

属性	型	内容
[in]	BSTR _bstrFileName	ロードモジュール(パス名を含む)

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例

```
HRESULT hr = E_FAIL;
BSTR bstrDownloadFile;

try
{
    hr = pHewServer1->Download(bstrDownloadFile);
}
}
```

Unload

内容

ロードモジュールをアンロードします。

パラメータ

属性	型	内容
[in]	BSTR _bstrFileName	アンロードするモジュール(パス名を含む)

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例

```
HRESULT hr = E_FAIL;
BSTR bstrUnloadFile;

try
{
    hr = pHewServer1->Unload(bstrUnloadFile);
}
}
```

5.3.10 起動/終了

InvokeHew

内容

High-performance Embedded Workshop アプリケーションを起動します。(ワークスペースは開きません。)

パラメータ

パラメータはありません。

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外 (例えば E_FAIL (0x80004005L)) を返します。

記述例

```
HRESULT    hr = E_FAIL;

try
{
    hr = pHewServer1->InvokeHew();
}
```

内容

High-performance Embedded Workshop アプリケーションを終了します。

パラメータ

パラメータはありません。

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例

```
HRESULT    hr = E_FAIL;

try
{
    hr = pHewServer1->QuitHew();
}
```

5.3.11 ワークスペース

OpenWorkspace

内容

ワークスペースを開きます。

パラメータ

属性	型	内容
[in]	BSTR _bstrFileName	ファイル名(パス名を含む)

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例

```
HRESULT hr = E_FAIL;
BSTR bstrOpenWorkspace;

try
{
    hr = pHewServer1->OpenWorkspace(bstrOpenWorkspace);
}
}
```

CloseWorkspace

内容

ワークスペースを閉じます。

パラメータ

属性	型	内容
[in]	DWORD _dwIgnoreChanges	0x00000000:変更があった場合、ワークスペースを閉じない 0x00000001:変更を保存せずに、ワークスペースを閉じる

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例

```
HRESULT hr = E_FAIL
DWORD _dwIgnoreChanges;

try
{
    hr = pHewServer1->CloseWorkspace(_dwIgnoreChanges);
}
}
```

SaveWorkspace

内容

ワークスペースを保存します。

パラメータ

パラメータは、ありません。

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例

```
HRESULT hr = E_FAIL;

try
{
    hr = pHewServer1->SaveWorkspace();
}
```

SaveSession

内容

セッションファイルを保存します。

パラメータ

パラメータはありません。

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例

```
HRESULT    hr = E_FAIL

try
{
    hr = pHewServer1->SaveSession();
}
```

GetCurrentConfiguration

内容

現在のビルド構成を取得します。

パラメータ

属性	型	内容
[out]	BSTR *p_bstrCurrentConfigurationName	ビルド構成名

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例

```
HRESULT hr = E_FAIL;
CString strCurrentConfigurationName = _T("");
BSTR bstrCurrentConfigurationName = strCurrentConfigurationName.AllocSysString(); //CString ->
BSTR 変換

//calling HewTargetServer function
try
{
    hr = pHewServer1->GetCurrentConfiguration(&bstrCurrentConfigurationName);
    strCurrentConfigurationName = bstrCurrentConfigurationName;
}
```

SetCurrentConfiguration

内容

現在有効なビルド構成を設定します。

パラメータ

属性	型	内容
[in]	BSTR _bstrConfiguration	ビルド構成名

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例

```
HRESULT hr = E_FAIL;
BSTR bstrSetCurrentConfiguration;

try
{
    hr = pHewServer1->SetCurrentConfiguration(bstrSetCurrentConfiguration);
}
}
```

GetConfigurations

内容

プロジェクトにあるビルド構成をすべて取得します。

パラメータ

属性	型	内容
[out]	BSTR *p_bstrConfigurations	ビルド構成名 (名前が複数ある場合は、カンマで区切る (例) "DefaultSession, SimSessionSH-4")

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外 (例えば E_FAIL (0x80004005L)) を返します。

記述例

```
HRESULT hr = E_FAIL;
CString strTmp = _T("");
BSTR ol = strTmp.AllocSysString(); //CString -> BSTR 変換

//calling HewTargetServer function
CString sol;
try
{
    hr = pHewServer1->GetConfigurations(&ol);
    sol = ol;
}
}
```

GetCurrentSession

内容

現在のデバッグセッションを取得します。

パラメータ

属性	型	内容
[out]	BSTR *p_bstrCurrentSessionName	デバッグセッション名

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例

```
HRESULT hr = E_FAIL;
CString strCurrentSessionName = _T("");
BSTR bstrCurrentSessionName = strCurrentSessionName.AllocSysString(); //CString -> BSTR 変換

//calling HewTargetServer function
try
{
    hr = pHewServer1->GetCurrentSession(&bstrCurrentSessionName);
    strCurrentSessionName = bstrCurrentSessionName;
}
```

SetCurrentSession

内容

現在有効なデバッグセッションを設定します。

パラメータ

属性	型	内容
[in]	BSTR _bstrSession	デバッグセッション名

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例

```
BSTR bstrSetCurrentSession = m_SetCurrentSession.AllocSysString(); //CString -> BSTR 変換
HRESULT hr = E_FAIL;

//calling HewTargetServer function
try
{
    hr = pHewServer1->SetCurrentSession(bstrSetCurrentSession);
}
}
```

GetSessions

内容

プロジェクトにある全デバッグセッションを取得します。

パラメータ

属性	型	内容
[out]	BSTR *p_bstrSessions	デバッグセッション名(名前が複数ある場合は、カンマで区切る) (例) "DefaultSession, SimSessionSH-4"

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例

```
HRESULT hr = E_FAIL;

//calling HewTargetServer function
CString strTmp = _T("");
BSTR o2 = strTmp.AllocSysString(); //CString -> BSTR 変換

CString so2;
try
{
    hr = pHewServer1->GetSessions(&o2);
    so2 = o2;
}
}
```

GetCurrentProject

内容

現在のプロジェクトを取得します。

パラメータ

属性	型	内容
[out]	BSTR *p_bstrCurrentProjectName	プロジェクト名

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例

```
HRESULT hr = E_FAIL;
CString strCurrentProjectName = _T("");
BSTR bstrCurrentProjectName = strCurrentProjectName.AllocSysString(); //CString -> BSTR 変換

//calling HewTargetServer function
try
{
    hr = pHewServer1->GetCurrentProject(&bstrCurrentProjectName);
    strCurrentProjectName = bstrCurrentProjectName;
}
```

SetCurrentProject

内容

アクティブプロジェクトを設定します。

パラメータ

属性	型	内容
[in]	BSTR _bstrProjectName	プロジェクト名

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例

```
HRESULT hr = E_FAIL;

BSTR bstrSetCurrentProject = m_SetCurrentProject.AllocSysString(); //CString -> BSTR 変換
//calling HewTargetServer function
try
{
    hr = pHewServer1->SetCurrentProject(bstrSetCurrentProject);
}
}
```

GetProjects

内容

プロジェクト名をすべて取得します。

パラメータ

属性	型	内容
[out]	BSTR *p_bstrProjectNames	プロジェクト名 名前が複数ある場合は、カンマで区切る (例)"Project1, Project2"

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例

```
HRESULT    hr = E_FAIL;
CString    strProjectNames = _T("");
BSTR bstrProjectNames = strProjectNames.AllocSysString(); //CString -> BSTR 変換

//calling HewTargetServer function
try
{
    hr = pHewServer1->GetProjects(&bstrProjectNames);
    strProjectNames = bstrProjectNames;
}
```

5.3.13 プロジェクト

AddFile

内容

アクティブなプロジェクトにファイルを追加します。

パラメータ

属性	型	内容
[in]	BSTR _bstrFileName	ファイル名(パス名を含む)

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例

```
HRESULT hr = E_FAIL;
BSTR bstrAddFile;

try
{
    hr = pHewServer1->AddFile(bstrAddFile);
}
}
```

AddFiles

内容

アクティブなプロジェクトに複数のファイルを追加します。

パラメータ

属性	型	内容
[in]	BSTR _bstrFileName	ファイル名 (ファイルが複数ある場合は、カンマで区切る) (パス名を含む)

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外 (例えば E_FAIL (0x80004005L)) を返します。

記述例

```
HRESULT hr = E_FAIL;
BSTR bstrAddFiles;

try
{
    hr = pHewServer1->AddFiles(bstrAddFiles);
}
}
```

DeleteFile

内容

アクティブなプロジェクトからファイルを削除します。

パラメータ

属性	型	内容
[in]	BSTR _bstrFileName	ファイル名(パス名を含む)

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例

```
HRESULT          hr = E_FAIL

BSTR bstrDeleteFile = m_DeleteFile.AllocSysString(); //CString -> BSTR 変換
//calling HewTargetServer function
try
{
    hr = pHewServer1->DeleteFile(bstrDeleteFile);
}
```

DeleteFiles

内容

アクティブなプロジェクトから複数のファイルを削除します。

パラメータ

属性	型	内容
[in]	BSTR _bstrFileName	ファイル名 (ファイルが複数ある場合は、カンマで区切る) (パス名を含む)

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外 (例えば E_FAIL (0x80004005L)) を返します。

記述例

```
HRESULT          hr = E_FAIL

BSTR bstrDeleteFiles = m_DeleteFiles.AllocSysString(); //CString -> BSTR 変換
//calling HewTargetServer function
try
{
    hr = pHewServer1->DeleteFiles(bstrDeleteFiles);
}
}
```

BuildProject

内容

プロジェクトをビルドします。

パラメータ

パラメータはありません。

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例

```
HRESULT    hr = E_FAIL;

//calling HewTargetServer function
try
{
    hr = pHewServer1->BuildProject();
}
```

RebuildProject

内容

プロジェクトをリビルドします。

パラメータ

パラメータはありません。

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例

```
HRESULT    hr = E_FAIL;

//calling HewTargetServer function
try
{
    hr = pHewServer1->RebuildProject();
}
```

UpDateAllDependency

内容

すべての依存関係を更新します。

パラメータ

パラメータはありません。

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例

```
HRESULT hr = E_FAIL;
//calling HewTargetServer function
try
{
    hr = pHewServer1->UpDateAllDependency();
}
```

AddFileWithCompilerOption

内容

プロジェクトにコンパイラオプションを設定してファイルを追加します。

パラメータ

属性	型	内容
[in]	BSTR _bstrFileName	ファイル名(パス名を含む)
[in]	BSTR _bstrIncludeDirectories	インクルードディレクトリ名 ディレクトリ名が複数ある場合は、カンマで区切る(例) "C:¥tmp, D:¥work"
[in]	BSTR _bstrDefines	定義 定義が複数ある場合は、カンマで区切る(例) "TMP1=C:¥tmp, TMP2=D:¥work"

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外 (例えば E_FAIL (0x80004005L)) を返します。

記述例

```
HRESULT hr = E_FAIL;
BSTR bstrFileName;
BSTR bstrIncludeDirectories;
BSTR bstrDefines;

//calling HewTargetServer function
try
{
    hr =pHewServer1->AddFileWithCompilerOption(bstrFileName, bstrIncludeDirectories, bstrDefines);
}
```

5.3.15 ファイル関係

OpenFileAtLine

内容

ファイル名、行番号を指定しファイルを開きます。

パラメータ

属性	型	内容
[in]	BSTR _bstrOpenFileName	ファイル名(パス名を含む)
[in]	int _iLine	行番号

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例

```
HRESULT hr = E_FAIL;
CString strOpenFileName = _T("");
BSTR bstrOpenFileName = strOpenFileName.AllocSysString(); //CString -> BSTR 変換
int iLine = 1;

//calling HewTargetServer function
try
{
    hr = pHewServer1->OpenFileAtLine(bstrOpenFileName, iLine);
}
```

GetSourceFiles

内容

プロジェクトにあるソースファイル名 (*.cpp, *.src 等) をすべて取得します。
ファイル名は、絶対パスで出力します。

パラメータ

属性	型	内容
[out]	BSTR *p_bstrSourceFiles	ソースファイル名(名前が複数ある場合は、カンマで区切る (例)"c:\sample1.cpp, c:\sample2.cpp")

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外 (例えば E_FAIL (0x80004005L)) を返します。

記述例

```
HRESULT hr = E_FAIL;
CString strSourceFiles = _T("");
BSTR bstrSourceFiles = strSourceFiles.AllocSysString(); //CString -> BSTR 変換

//calling HewTargetServer function
try
{
    hr = pHewServer1->GetSourceFiles(&bstrSourceFiles);
    strSourceFiles = bstrSourceFiles;
}
```

GetDownloadModules

内容

プロジェクトにあるモジュールファイル名 (*.abs 等) をすべて取得します。
ファイル名は、絶対パスで出力します。

パラメータ

属性	型	内容
[out]	BSTR *p_bstrDownloadModules	モジュールファイル名(名前が複数ある場合は、カンマで区切る(例)"c:\¥sample1.abs, c:\¥sample2.abs")

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外 (例えば E_FAIL (0x80004005L)) を返します。

記述例

```
HRESULT hr = E_FAIL;
CString strDownloadModules = _T("");
BSTR bstrDownloadModules = strDownloadModules.AllocSysString(); //CString -> BSTR 変換

//calling HewTargetServer function
try
{
    hr = pHewServer1->GetDownloadModules(&bstrDownloadModules);
    strDownloadModules = bstrDownloadModules;
}
```

GetDependentFiles

内容

プロジェクトにある依存ファイル名(*.h, *.inc 等)をすべて取得します。
ファイル名は、絶対パスで出力します。

パラメータ

属性	型	内容
[out]	BSTR *p_bstrDependentFiles	依存ファイル名(名前が複数ある場合は、カンマで区切る (例)"c:\¥sample1.h, c:\¥sample2.h")

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例

```
HRESULT    hr = E_FAIL;
CString    strDependentFiles = _T("");
BSTR bstrDependentFiles = strDependentFiles.AllocSysString(); //CString -> BSTR 変換

//calling HewTargetServer function
try
{
    hr = pHewServer1->GetDependentFiles(&bstrDependentFiles);
    strDependentFiles = bstrDependentFiles;
}
}
```

5.3.16 カバレッジ関係

SetCoverageRange

内容

カバレッジの範囲を設定します。

パラメータ

属性	型	内容
[in]	DWORD dwStartAddress	開始アドレス
[in]	DWORD dwEndAddress	終了アドレス

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例

```
HRESULT hr = E_FAIL;
DWORD dwStartAddress;
DWORD dwEndAddress;

//calling HewTargetServer function
try
{
    hr = pHewServer1->SetCoverageRange(dwStartAddress, dwEndAddress);
}
```

注意事項

カバレッジの範囲を設定すると同時に、カバレッジが有効になります。

GetCoverageRange

内容

カバレッジの範囲を取得します。

パラメータ

属性	型	内容
[out]	DWORD *p_dwStartAddress	開始アドレス
[out]	DWORD *p_dwEndAddress	終了アドレス

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例

```
HRESULT    hr = E_FAIL;
DWORD      dwStartAddress;
DWORD      dwEndAddress;

//calling HewTargetServer function
try
{
    hr = pHewServer1->GetCoverageRange(&dwStartAddress, &dwEndAddress);
}
```

SetCoverageDisable

内容

カバレッジを無効に設定します。

パラメータ

パラメータはありません。

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例

```
HRESULT    hr = E_FAIL;

//calling HewTargetServer function
try
{
    hr = pHewServer1->SetCoverageDisable();
}
```

SetCoverageEnable

内容

カバレッジを有効に設定します。

パラメータ

パラメータはありません。

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例

```
HRESULT    hr = E_FAIL;

//calling HewTargetServer function
try
{
    hr = pHewServer1->SetCoverageEnable();
}
```

ClearCoverage

内容

カバレッジ情報をクリアします。

パラメータ

パラメータはありません。

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例

```
HRESULT    hr = E_FAIL;

//calling HewTargetServer function
try
{
    hr = pHewServer1->ClearCoverage();
}
```

GetCoverageStatus

内容

カバレッジの状態を取得します。

パラメータ

属性	型	内容
[out]	int *p_iStatus	カバレッジの状態(1:有効、0:無効)

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例

```
HRESULT hr = E_FAIL;
int iStatus;

//calling HewTargetServer function
try
{
    hr = pHewServer1->GetCoverageStatus(&iStatus);
}
```

LoadCoverage

内容

カバレッジ情報をロードします。

パラメータ

属性	型	内容
[in]	BSTR _bstrLoadFileName	ファイル名

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例

```
HRESULT hr = E_FAIL;
BSTR bstrLoadFileName;

//calling HewTargetServer function
try
{
    hr = pHewServer1->LoadCoverage(bstrLoadFileName);
}
```

SaveCoverage

内容

カバレッジ情報をセーブします。

パラメータ

属性	型	内容
[in]	BSTR _bstrSaveFileName	ファイル名

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例

```
HRESULT hr = E_FAIL;
BSTR bstrSaveFileName;

//calling HewTargetServer function
try
{
    hr = pHewServer1->SaveCoverage(bstrSaveFileName);
}
```

5.3.17 その他

GetErrorString

内容

エラー番号に対応したエラーメッセージを取得します。

パラメータ

属性	型	内容
[in]	HRESULT _lError	エラー番号
[out]	BSTR* _pbstrError	エラーメッセージ

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例

```
HRESULT    hrErr = E_FAIL;
HRESULT    _lError;
BSTR       bstrErrStr;

try
{
    hrErr = pHewServer1->GetErrorString(_lError, &bstrErrStr);
}
}
```

GetHewStatus

内容

現在の High-performance Embedded Workshop の状態を取得します。

パラメータ

属性	型	内容
[out]	int* p_iTargetReset	ターゲットリセット時には1が返却、それ以外は0*
[out]	int* p_iTargetExecStatus	ユーザプログラム実行中は1が返却、それ以外は0
[out]	int* p_iMemoryReset	メモリ内容が更新されているときには1が返却、それ以外は0*
[out]	int* p_iRegisterReset	レジスタ値が更新されているときには1が返却、それ以外は0*
[out]	int* p_iLinkStatus	ターゲット接続時には1が返却、それ以外は0
[out]	int* p_iPlatformInitialize	ターゲット初期化後には1が返却、それ以外は0*
[out]	int* p_iLoadingStatus	プログラムロード後には1が返却、それ以外は0

*:本関数が呼び出されると、本フラグは0にリセットされます。

戻り値

メソッドが正常に終了した場合は1、エラーが発生した場合は0を返します。

記述例

```
int      iTargetReset;
int      iTargetExecStatus;
int      iMemoryReset;
int      iRegisterReset;
int      iLinkStatus;
int      iPlatformInitialize;
int      iLoadingStatus;
HRESULT  hr;

//calling HewTargetServer function
try
{
    hr = pHewServer1->GetHewStatus(&iTargetReset, &iTargetExecStatus,
        &iMemoryReset,
        &iRegisterReset,
        &iLinkStatus,
        &iPlatformInitialize,
        &iLoadingStatus
    );
}
}
```

GetHewStatusEx

内容

High-performance Embedded Workshop のステータス (起動, ワークスペースオープン, ビルド) を取得します。

パラメータ

属性	型	内容
[out]	int *p_iInvokeHew	High-performance Embedded Workshop の起動状態 (0:未起動、1:起動済み)
[out]	int *p_iOpenWorkspace	ワークスペースの状態 (0:未オープン、1:オープン済み) ※ 実際にはターゲットと接続中のみ 1 になります。 ワークスペースオープン中かどうかは GetWorkspaceDirectory を呼び出して判定してください。
[out]	int *p_iBuildProject	ビルド状態 (0:ビルド停止、1:ビルド中)

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外 (例えば E_FAIL (0x80004005L)) を返します。

記述例

```
HRESULT hr = E_FAIL;
int iInvokeHew;
int iOpenWorkspace;
int iBuildProject;

//calling HewTargetServer function
try
{
    hr = pHewServer1->GetHewStatusEx(&iInvokeHew, &iOpenWorkspace, &iBuildProject);
}
```

GetTargetName

内容

現在接続しているターゲット名を取得します。

パラメータ

属性	型	内容
[out]	BSTR* p_bstrName	ターゲット名

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例

```
HRESULT hr = E_FAIL;
BSTR bStrTargetName;

//calling HewTargetServer function
try
{
    //get target name
    hr = pHewServer1->GetTargetName (&bStrTargetName);
}
}
```

5.4 メソッドの詳細 (VB、VC++用)

5.4.1 実行制御

GoTargetExec2

内容

現プログラム位置より、プログラムを実行します。

パラメータ

パラメータはありません。

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;

try
{
    hr = pHewServer1->GoTargetExec2();
}
```

記述例 (Visual Basic)

```
Dim ret As Long

ret = hts.GoTargetExec2
```

StopTargetExec2

内容

プログラムの実行を停止します。

パラメータ

パラメータはありません。

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;

try
{
    hr = pHewServer1->StopTargetExec2();
}
}
```

記述例 (Visual Basic)

```
Dim ret As Long

ret = hts.StopTargetExec2
```

ResetTargetExec2

内容

ターゲットとしているデバッグ環境をリセットします。

パラメータ

パラメータはありません。

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;

try
{
    hr = pHewServer1->ResetTargetExec2();
}
```

記述例 (Visual Basic)

```
Dim ret As Long

ret = hts.ResetTargetExec2
```

InitializeTarget2

内容

ターゲットとしているデバッグ環境を初期化します。

パラメータ

パラメータはありません。

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;

try
{
    hr = pHewServer1->InitializeTarget2();
}
```

記述例 (Visual Basic)

```
Dim ret As Long

ret = hts.InitializeTarget2
```

Step2

内容

プログラムをステップ実行します。

パラメータ

属性	型	内容
[in]	long _lMode	ステップの種類 0x00000001 アセンブラの命令でステップ実行 0x00000002 ソースコードの行でステップ実行
[in]	long _lStep	実行するステップ数

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
long _lMode = 1;           //assembler:1, source:2
long _lStep = 1;

try
{
    hr = pHewServer1->Step2(_lMode, _lStep);
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim lMode As Long
Dim lStep As Long
lMode = 1
lStep = 1

ret = hts.Step2(lMode, lStep)
```

StepRate2

内容

ステップの実行速度を設定します。

パラメータ

属性	型	内容
[in]	long _lRate	ステップレート 0~6 の値を設定します。

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT          hr = E_FAIL;
int _lRate = 0;

try
{
    hr = pHewServer1->StepRate2(_lRate);
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim lRate As Long
lRate = 0

ret = hts.StepRate2(lRate)
```

StepOver2

内容

ステップオーバーでプログラムを実行します。

パラメータ

属性	型	内容
[in]	long _lMode	ステップの種類 0x00000001 アセンブラの命令でステップ実行 0x00000002 ソースコードの行でステップ実行
[in]	long _lStep	ステップ数

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
long _lMode = 1;           //assembler:1, source:2
long _lStep = 1;

try
{
    hr = pHewServer1->StepOver2(_lMode, _lStep);
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim lMode As Long
Dim lStep As Long
lMode = 1
lStep = 1

ret = hts.StepOver2(lMode, lStep)
```

StepOut2

内容

プログラムをステップアウトで実行します。

パラメータ

属性	型	内容
[in]	long _lMode	ステップの種類 0x00000001 アセンブラの命令でステップ実行 0x00000002 ソースコードの行でステップ実行

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
long _lMode = 1;           //assembler:1, source:2

try
{
    hr = pHewServer1->StepOut2(_lMode);
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim lMode As Long
lMode = 1

ret = hts.StepOut2(lMode)
```

IsRunning2

内容

ユーザプログラム実行中か否かの判定をします。

パラメータ

属性	型	内容
[out]	long *p_lRunning	ユーザプログラム実行中のときは1、それ以外は0

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
long _lRunning;

try
{
    hr = pHewServer1->IsRunning2(&_lRunning);
}
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim p_lRunning As Long 'Visual Basic 2005 の場合は "As Integer"

ret = hts.IsRunning2(p_lRunning)
```

5.4.2 レジスタ操作

GetPC2

内容

プログラムカウンタの値を取得します。

パラメータ

属性	型	内容
[out]	long *p_lPC	PC (プログラムカウンタ) 値

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外 (例えば E_FAIL (0x80004005L)) を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
long _lPC = 0x0;

try
{
    hr = pHewServer1->GetPC2(&_lPC);
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim p_lPC As Long          'Visual Basic 2005 の場合は "As Integer"
p_lPC = 0

ret = hts.GetPC2(p_lPC)
```

SetPCAddress2

内容

プログラムカウンタを設定します。

パラメータ

属性	型	内容
[in]	long LPCAddr	設定する PC (プログラムカウンタ) 値

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外 (例えば E_FAIL (0x80004005L)) を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
long LPCAddr = 0x800;

try
{
    hr = pHewServer1->SetPCAddress2(LPCAddr);
}
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim LPCAddr As Long
LPCAddr = &H800

ret = hts.SetPCAddress2(LPCAddr)
```

SetPCSource2

内容

ソースファイル、行を指定しプログラムカウンタを設定します。

パラメータ

属性	型	内容
[in]	BSTR bstrFileName	ファイル名
[in]	long lLineNum	行番号

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
BSTR bstrFileName;
long lLineNum = 100;

try
{
    hr = pHewServer1->SetPCSource2(bstrFileName, lLineNum);
}
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim bstrFileName As String
Dim lLineNum As Long
lLineNum = 100

ret = hts.SetPCSource2(bstrFileName, lLineNum)
```

TestSetPC2

内容

プログラムカウンタに対して値の設定が可能か否かを判定します。

パラメータ

属性	型	内容
[out]	long *p_lSetPCState	PC 値設定可能なときは 1、それ以外は 0

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
long lSetPCState = 0;

try
{
    hr = pHewServer1->TestSetPC2(&lSetPCState);
}
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim p_lSetPCState As Long          ' Visual Basic 2005 の場合は "As Integer"
p_lSetPCState = 0

ret = hts.TestSetPC2(p_lSetPCState)
```

5.4.3 メモリ操作

GetMemory2

内容

指定した開始アドレス、終了アドレス、およびアクセスサイズによりメモリ内容を取得します。High-performance Embedded Workshop 内部で指定エリアのメモリ内容のデータを保持している場合は、ターゲットメモリへアクセスせずに High-performance Embedded Workshop 内部で保持しているメモリ内容を返却します。

パラメータ

属性	型	内容
[in]	long lBegin	メモリ内容取得の開始アドレス
[in]	long lEnd	メモリ内容取得の終了アドレス
[in]	long lDisplayWidth	メモリへアクセスするサイズ(1, 2, 4, 8 が指定可能)
[out]	VARIANT vMemData	メモリ内容

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
long lBegin = strtol(m_GetMemoryStartAddress, NULL, 16);
long lEnd = strtol(m_GetMemoryEndAddress, NULL, 16);
long lDisplayWidth = m_GetMemorySize.GetCurSel();

//array for string data obtained from HewTargetServer
...

try
{
    hr = pHewServer1->GetMemory2(lBegin, lEnd, lDisplayWidth, &vMemData);
}
}
```

記述例 (Visual Basic 6.0)

```
Dim ret As Long
Dim lBegin As Long
Dim lEnd As Long
Dim lDisplayWidth As Long
Dim vMemData As Variant

...

ret = hts.GetMemory2(lBegin, lEnd, lDisplayWidth, vMemData)
```

記述例 (Visual Basic 2005)

```
Dim ret As Integer
Dim iBegin As Integer
Dim iEnd As Integer
Dim iDisplayWidth As Integer
Dim vMemData As Object

...

ret = hts.GetMemory2(iBegin, iEnd, iDisplayWidth, vMemData)
```

SetMemory2

内容

指定した開始アドレス、終了アドレス、およびアクセスサイズによりメモリ内容を設定します。

パラメータ

属性	型	内容
[in]	long lBegin	メモリ内容取得の開始アドレス
[in]	long lEnd	メモリ内容取得の終了アドレス
[in]	long lDisplayWidth	メモリへアクセスするサイズ(1, 2, 4, 8 が指定可能)
[in]	VARIANT vMemData	メモリ内容

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT          hr = E_FAIL;

lBegin = strtol(m_SetMemoryStartAddress, NULL, 16);
lEnd = strtol(m_SetMemoryEndAddress, NULL, 16);

...

long length = lEnd - lBegin + 1;
long *pldataArray;
SAFEARRAY* psaData;
VARIANT vMemData;
VARIANT *p_vMemData = &vMemData;

SAFEARRAYBOUND bounds = {length, 0};
VariantInit(p_vMemData);
p_vMemData->vt = VT_ARRAY | VT_I4;
psaData = SafeArrayCreate(VT_I4, 1, &bounds);
SafeArrayAccessData(psaData, (void**)&pldataArray);

for (long j = 0 ; j < length ; j++ ) {
    CString strWork;
    // e.g. Set 0x00, 0x01, 0x02, 0x03, 0x00, 0x01, ...
    pldataArray[j] = j % 4;
}
SafeArrayUnaccessData(psaData);
p_vMemData->parray = psaData;

try
{
    hr = pHewServer1->SetMemory2(lBegin, lEnd, lDisplayWidth, vMemData);
}
...
SafeArrayDestroy(psaData);
```

記述例 (Visual Basic 6.0)

```
Dim ret As Long
Dim lBegin As Long
Dim lEnd As Long
Dim lDisplayWidth As Long
Dim i As Long
Dim length As Long
Dim vMemData As Variant
Dim memData(65535) As Long
. . .

length = lEnd - lBegin + 1
For i = 0 To length - 1
    ' e.g. Set 0x00, 0x01, 0x02, 0x03, 0x00, 0x01, ...
    memData(i) = i Mod 4
Next i
vMemData = memData

. . .

ret = hts.SetMemory2(lBegin, lEnd, lDisplayWidth, vMemData)
```

記述例 (Visual Basic 2005)

```
Dim ret As Integer
Dim iBegin As Integer
Dim iEnd As Integer
Dim iDisplayWidth As Integer
Dim i As Integer
Dim length As Integer
Dim vMemData As Object
Dim memData(65535) As Integer
. . .

length = iEnd - iBegin + 1
For i = 0 To length - 1
    ' e.g. Set 0x00, 0x01, 0x02, 0x03, 0x00, 0x01, ...
    memData(i) = i Mod 4
Next i
vMemData = memData

. . .

ret = hts.SetMemory2(iBegin, iEnd, iDisplayWidth, vMemData)
```

GetDirectMemory2

内容

指定した開始アドレス、終了アドレス、およびアクセスサイズによりメモリ内容を取得します。High-performance Embedded Workshop 内部で本指定エリアのメモリ内容のデータを保持している、いないにかかわらず、ターゲットメモリへアクセスしてメモリ内容を返却します

パラメータ

属性	型	内容
[in]	long lBegin	メモリ内容取得の開始アドレス
[in]	long lEnd	メモリ内容取得の終了アドレス
[in]	long lDisplayWidth	メモリへアクセスするサイズ(1, 2, 4, 8 が指定可能)
[out]	VARIANT *p_vMemData	メモリ内容

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
long lBegin = strtol(m_GetMemoryStartAddress, NULL, 16);
long lEnd = strtol(m_GetMemoryEndAddress, NULL, 16);
long lDisplayWidth = m_GetMemorySize.GetCurSel();

//array for string data obtained from HewTargetServer
. . .

try
{
    hr = pHewServer1->GetDirectMemory2(lBegin, lEnd, lDisplayWidth, &vMemData);
}
}
```

記述例 (Visual Basic 6.0)

```
Dim ret As Long
Dim lBegin As Long
Dim lEnd As Long
Dim lDisplayWidth As Long
Dim vMemData As Variant

. . .

ret = hts.GetDirectMemory2(lBegin, lEnd, lDisplayWidth, vMemData)
```

記述例 (Visual Basic 2005)

```
Dim ret As Integer
Dim iBegin As Integer
Dim iEnd As Integer
Dim iDisplayWidth As Integer
Dim vMemData As Object

. . .

ret = hts.GetDirectMemory2(iBegin, iEnd, iDisplayWidth, vMemData)
```

5.4.4 ソフトウェアブレイク

SetPCBreakPt2

内容

指定アドレスにブレイクポイントを設定し、そのハンドル値を返却します。

パラメータ

属性	型	内容
[in]	long lPCBreakAddr	アドレス値
[out]	long *p_lHandle	ブレイクポイントのハンドル値

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
long lPCBreakAddr;
long lHandle;

try
{
    hr = pHewServer1->SetPCBreakPt2(lPCBreakAddr, &lHandle);
}
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim lPCBreakAddr As Long
Dim p_lHandle As Long 'Visual Basic 2005 の場合は "As Integer"

ret = hts.SetPCBreakPt2(lPCBreakAddr, p_lHandle)
```

EnableBreakPt2

内容

ブレイクポイントのハンドル値から、ブレイクポイントの有効化、無効化を設定します。

パラメータ

属性	型	内容
[in]	long lHandle	ブレイクポイントのハンドル値
[in]	long lEnable	有効化(1), 無効化(0)

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT          hr = E_FAIL;
long lHandle;
long lEnable = 1;

try
{
    hr = pHewServer1->EnableBreakPt2(lHandle, lEnable);
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim lHandle As Long
Dim lEnable As Long
lEnable = 1

ret = hts.EnableBreakPt2(lHandle, lEnable)
```

DeleteBreakPt2

内容

ブレイクポイントのハンドル値のブレイクポイントを削除します。

パラメータ

属性	型	内容
[in]	long lHandle	ブレイクポイントのハンドル値

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT          hr = E_FAIL;
long             lHandle;

try
{
    hr = pHewServer1->DeleteBreakPt2(lHandle);
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim lHandle As Long

ret = hts.DeleteBreakPt2(lHandle)
```

GetAllBreakPt2

内容

現在設定されているブレイクポイントを取得します。

パラメータ

属性	型	内容
[out]	long *p_index	ブレイクポイントの数
[out]	VARIANT *p_vAllBreakPt	ブレイクポイントの配列

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
long index;
VARIANT vAllBreakPt;
VariantInit( &vAllBreakPt );

//calling HewTargetServer function
try
{
    hr = pHewServer1->GetAllBreakPt2(&index, &vAllBreakPt);
}
```

記述例 (Visual Basic 6.0)

```
Dim ret As Long
Dim p_index As Long
Dim p_vAllBreakPt As Variant

ret = hts.GetAllBreakPt2(p_index, p_vAllBreakPt)
```

記述例 (Visual Basic 2005)

```
Dim ret As Integer
Dim p_index As Integer
Dim p_vAllBreakPt As Variant

ret = hts.GetAllBreakPt2(p_index, p_vAllBreakPt)
```

DeleteAllBreakPt2

内容

現在設定されているブレイクポイントを削除します。

パラメータ

パラメータはありません。

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT    hr = E_FAIL;

//calling HewTargetServer function
try
{
    hr = pHewServer1->DeleteAllBreakPt2();
}
```

記述例 (Visual Basic)

```
Dim ret As Long

ret = hts.DeleteAllBreakPt2
```

5.4.5 変数ブレイク

SetDataBreakpoint2

内容

ブレイクデータポイントを設定します。

パラメータ

属性	型	内容
[in]	long _lSymbol	シンボルのアドレス
[in]	long _lSize	シンボルのサイズ(1/2/4) 0x00000001 - 1 0x00000002 - 2 0x00000004 - 4
[in]	long _lType	ブレイクする種類(Equal/Not Equal) 0x00000001 - Equal 0x00000002 - Not Equal
[in]	long _lData	シンボルの値
[out]	long *p_lBreakDataNo	変数ブレイクの番号

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
long _lSymbol;
long _lSize;
long _lType;
long _lData;
long _lBreakDataNo;

...

try
{
    hr = pHewServer1->SetDataBreakpoint2(_lSymbol, _lSize, _lType, _lData,
    &_lBreakDataNo);
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim lSymbol As Long
Dim lSize As Long
Dim lType As Long
Dim lData As Long
Dim p_lBreakDataNo As Long 'Visual Basic 2005 の場合は"As Integer"

...

ret = hts.SetDataBreakpoint2(lSymbol, lSize, lType, lData, p_lBreakDataNo)
```

EnableDataBreakpoint2

内容

ブレイクデータポイントの有効/無効を設定します。

パラメータ

属性	型	内容
[in]	long lDataBreakNo	変数ブレイクの番号
[in]	long _lEnable	有効(1)/ 無効(0)

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
long lDataBreakNo;
long _lEnable = 1;

try
{
    hr = pHewServer1->EnableDataBreakpoint2(lDataBreakNo, _lEnable);
}
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim lDataBreakNo As Long
Dim lEnable As Long
lEnable = 1

ret = hts.EnableDataBreakpoint2(lDataBreakNo, lEnable)
```

DeleteDataBreakpoint2

内容

ブレークデータポイントを削除します。

パラメータ

属性	型	内容
[in]	long lDataBreakNo	変数ブレークの番号

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
long lDataBreakNo;

try
{
    hr = pHewServer1->DeleteDataBreakpoint2(lDataBreakNo);
}
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim lDataBreakNo As Long

ret = hts.DeleteDataBreakpoint2(lDataBreakNo)
```

5.4.6 変数トレース

SetSymbolTrace2

内容

変数トレース条件を設定します。

パラメータ

属性	型	内容
[in]	long _lSymbol	シンボルのアドレス
[in]	long _lCondition	トレース条件(Read/Write) 0x00000001 - Read 0x00000002 - Write 0x00000003 - Read_Write
[in]	long _lSize	シンボルのサイズ(1/2/4) 0x00000001 - 1 0x00000002 - 2 0x00000004 - 4
[in]	long _lType	トレースする種類(Equal/Not Equal/No Specific) 0x00000001 - Equal 0x00000002 - Not Equal 0x00000003 - Not Specified
[in]	long _lData	シンボルの値
[out]	long *p_lTraceNo	変数トレースの番号

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
long _lSymbol;
long _lCondition;
long _lSize;
long _lType;
long _lData;
long _lTraceNo;

...

try
{
    hr = pHewServer1->SetSymbolTrace2(_lSymbol, _lCondition, _lSize, _lType, _lData, &
    _lTraceNo);
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim lSymbol As Long
Dim lCondition As Long
Dim lSize As Long
Dim lType As Long
Dim lData As Long
Dim p_lTraceNo As Long          'Visual Basic 2005 の場合は"As Integer"

. . .

ret = hts.SetSymbolTrace2(lSymbol, lCondition, lSize, lType, lData, p_lTraceNo)
```

ExecuteSymbolTrace2

内容

変数トレースの有効/無効を設定します。

パラメータ

属性	型	内容
[in]	long _lEnable	有効(1)/無効(0)

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
long _lEnable = 1;

try
{
    hr = pHewServer1->ExecuteSymbolTrace2(_lEnable);
}
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim lEnable As Long
lEnable = 1

ret = hts.ExecuteSymbolTrace2(lEnable)
```

DeleteSymbolTrace2

内容

変数トレース条件を削除する。

パラメータ

属性	型	内容
[in]	long _lTraceNo	削除する変数トレースの番号

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
long _lTraceNo;

try
{
    hr = pHewServer1->DeleteSymbolTrace2(_lTraceNo);
}
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim lTraceNo As Long

ret = hts.DeleteSymbolTrace2(lTraceNo)
```

SaveSymbolTraceData2

内容

変数トレースの結果を指定したファイルに保存します。

パラメータ

属性	型	内容
[in]	BSTR _bstrFileName	変数トレースデータの保存ファイル

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT          hr = E_FAIL;
BSTR bstrFileName;

try
{
    hr = pHewServer1->SaveSymbolTraceData2(bstrFileName);
}
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim bstrFileName As String

ret = hts.SaveSymbolTraceData2(bstrFileName)
```

出力フォーマット例

トレース結果は、以下の内容をスペースで区切って出力します。

- アクセスした時間(シミュレータではサイクル数)
- アクセスしたアドレス
- アクセス属性(Read/Write/Read_Write)
- アクセス値
- アクセスサイズ

サンプル

```
1287539 0XFFFE5DC Write 0XEA 1
1287553 0XFFFE5DC Write 0X30 1
1288170 0XFFFE5DC Write 0XEA 1
1445327 0XFFFE5DC Write 0XE0 1
1445341 0XFFFE5DC Write 0X30 1
1445958 0XFFFE5DC Write 0XE0 1
1605377 0XFFFE5DC Write 0X4C 1
1605391 0XFFFE5DC Write 0X30 1
1606008 0XFFFE5DC Write 0X4C 1
1760876 0XFFFE5DC Write 0XF6 1
```

5.4.7 割り込み

SendTrigger2

内容

割り込み条件を設定します。

パラメータ

属性	型	内容
[in]	long _lTriggerNo	トリガー番号
[in]	long _lTriggerType1	トリガー割り込み条件 1
[in]	long _lTriggerType2	トリガー割り込み条件 2
[in]	long _lPriority	割り込み優先順位 (0~17)

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
long _lTriggerNo;
long _lTriggerType1;
long _lTriggerType2;
long _lPriority;

try
{
    hr = pHewServer1->SendTrigger2(
        _lTriggerNo,
        _lTriggerType1,
        _lTriggerType2,
        _lPriority
    );
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim lTriggerNo As Long
Dim lTriggerType1As Long
Dim lTriggerType2As Long
Dim lPriority As Long

ret = hts.SendTrigger2(lTriggerNo, lTriggerType1, lTriggerType2, lPriority)
```

5.4.8 シンボル操作

GetRealTimeWatch2

内容

指定したデータ値を取得します。

パラメータ

属性	型	内容
[in]	long _lSymbol	シンボル名のアドレス
[in]	long _lSize	シンボルのサイズ(1/2/4) 0x00000001 - 1 0x00000002 - 2 0x00000004 - 4
[out]	long *p_lValue	シンボルの値

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
long _lSymbol;
long _lSize;
long _lValue;

try
{
    hr = pHewServer1->GetRealTimeWatch2(_lSymbol, _lSize, &_lValue);
}
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim lSymbol As Long
Dim lSize As Long
Dim p_lValue As Long 'Visual Basic 2005 の場合は"As Integer"

ret = hts.GetRealTimeWatch2(lSymbol, lSize, p_lValue)
```

GetQuickWatch2

内容

変数名から、変数のサイズ、変数値、型、割付領域を取得します。

パラメータ

属性	型	内容
[in]	BSTR bstrVarName	変数名
[out]	long *p_lValueSize	変数のサイズ
[out]	BSTR *bstrByValue	変数値の文字列
[out]	long *p_lType	変数型
[out]	BSTR *bstrTypeName	変数型の文字列
[out]	BSTR *bstrVarAllocation	変数割付領域の文字列

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
BSTR bstrVarName;
long lValueSize;
BSTR bstrByValue;
long lType;
BSTR bstrTypeName;
BSTR bstrVarAllocation;

try
{
    hr = pHewServer1->GetQuickWatch2(bstrVarName,
                                     &lValueSize,
                                     &bstrByValue,
                                     &lType,
                                     &bstrTypeName,
                                     &bstrVarAllocation
    );
}
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim bstrVarName As String
Dim p_lValueSize As Long 'Visual Basic 2005 の場合は "As Integer"
Dim p_bstrByValue As String
Dim p_lType As Long 'Visual Basic 2005 の場合は "As Integer"
Dim p_bstrTypeName As String
Dim p_bstrVarAllocation As String

ret = hts.GetQuickWatch2(bstrVarName, p_lValueSize, p_bstrByValue, p_lType, _
                        p_bstrTypeName, p_bstrVarAllocation)
```

SymbolToAddress2

内容

シンボル名から対応するアドレス値へ変換します。

パラメータ

属性	型	内容
[in]	BSTR bstrSymbolName	シンボル名
[out]	long *p_lSymbolAddr	アドレス

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
BSTR bstrSymbolName;
long lSymbolAddr;

try
{
    hr = pHewServer1->SymbolToAddress2(bstrSymbolName, &lSymbolAddr);
}
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim bstrSymbolName As String
Dim p_lSymbolAddr As Long 'Visual Basic 2005 の場合は "As Integer"

ret = hts.SymbolToAddress2(bstrSymbolName, p_lSymbolAddr)
```

AddressToSymbol2

内容

アドレス値から対応するシンボル名へ変換します。

パラメータ

属性	型	内容
[in]	long lSymbolAddr	アドレス
[out]	BSTR *p_bstrSymbolName	シンボル名

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
long lSymbolAddr;
BSTR bstrSymbolName;

try
{
    hr = pHewServer1->AddressToSymbol2(lSymbolAddr, &bstrSymbolName);
}
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim lSymbolAddr Long
Dim p_bstrSymbolName As String

ret = hts.AddressToSymbol2(lSymbolAddr, p_bstrSymbolName)
```

GetLineFromAddr2

内容

アドレス値から対応するファイル、行へ変換します。

パラメータ

属性	型	内容
[in]	long lLineAddr	アドレス
[out]	BSTR *p_bstrFileName	ファイル名
[out]	long *p_lLineNo	行番号

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
long lLineAddr;
BSTR bstrFileName;
long lLineNo;

try
{
    hr = pHewServer1->GetLineFromAddr2(lLineAddr, &bstrFileName, &lLineNo);
}
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim lLineAddr Long
Dim p_bstrFileName As String
Dim p_lLineNo As Long 'Visual Basic 2005 の場合は "As Integer"

ret = hts.GetLineFromAddr2(lLineAddr, p_bstrFileName, p_lLineNo)
```

GetAddrFromLine2

内容

ファイル、行から対応するアドレス値へ変換します。

パラメータ

属性	型	内容
[in]	BSTR bstrFileName	ファイル名
[in]	long lLineNo	行番号
[out]	long *p_lLineAddr	アドレス

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
BSTR bstrFileName;
long lLineNo;
long lLineAddr;

try
{
    hr = pHewServer1->GetAddrFromLine2(
        bstrFileName,
        lLineNo,
        &lLineAddr
    );
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim bstrFileName As String
Dim lLineNo As Long
Dim p_lLineAddr As Long 'Visual Basic 2005 の場合は "As Integer"

ret = hts.GetAddrFromLine2(bstrFileName, lLineNo, p_lLineAddr)
```

5.4.9 ダウンロード

Download2

内容

ロードモジュールをダウンロードします。

パラメータ

属性	型	内容
[in]	BSTR _bstrFileName	ロードモジュール(パス名を含む)

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外 (例えば E_FAIL (0x80004005L)) を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
BSTR bstrDownloadFile;

try
{
    hr = pHewServer1->Download2(bstrDownloadFile);
}
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim bstrDownloadFile As String

ret = hts.Download2(bstrDownloadFile)
```

Unload2

内容

ロードモジュールをアンロードします。

パラメータ

属性	型	内容
[in]	BSTR _bstrFileName	アンロードするモジュール(パス名を含む)

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
BSTR bstrUnloadFile;

try
{
    hr = pHewServer1->Unload2(bstrUnloadFile);
}
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim bstrUnloadFile As String

ret = hts.Unload2(bstrUnloadFile)
```

5.4.10 起動/終了

InvokeHew2

内容

High-performance Embedded Workshop アプリケーションを起動します。(ワークスペースは開きません。)

パラメータ

パラメータはありません。

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外 (例えば E_FAIL (0x80004005L)) を返します。

記述例 (Visual C++)

```
HRESULT    hr = E_FAIL;

try
{
    hr = pHewServer1->InvokeHew2();
}
```

記述例 (Visual Basic)

```
Dim ret As Long

ret = hts.InvokeHew2
```

QuitHew2

内容

High-performance Embedded Workshop アプリケーションを終了します。

パラメータ

パラメータはありません。

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;

try
{
    hr = pHewServer1->QuitHew2();
}
```

記述例 (Visual Basic)

```
Dim ret As Long

ret = hts.QuitHew2
```

InvokeHewWithNoDialog

内容

ようこそダイアログボックスを表示せずに High-performance Embedded Workshop アプリケーションを起動します。(ワークスペースは開きません。)

パラメータ

パラメータはありません。

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外 (例えば E_FAIL (0x80004005L)) を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
try
{
    hr = pHewServer1->InvokeHewWithNoDialog();
}
```

記述例 (Visual Basic)

```
Dim ret As Long
ret = hts.InvokeHewWithNoDialog
```

5.4.11 ワークスペース

OpenWorkspace2

内容

ワークスペースを開きます。

パラメータ

属性	型	内容
[in]	BSTR _bstrFileName	ファイル名(パス名を含む)

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
BSTR bstrFileName;

try
{
    hr = pHewServer1->OpenWorkspace2(bstrFileName);
}
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim bstrFileName As String

ret = hts.OpenWorkspace2(bstrFileName)
```

CloseWorkspace2

内容

ワークスペースを閉じます。

パラメータ

属性	型	内容
[in]	long _lIgnoreChanges	0x00000000:変更があった場合、ワークスペースを閉じない 0x00000001:変更を保存せずに、ワークスペースを閉じる

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
long _lIgnoreChanges = 1;

try
{
    hr = pHewServer1->CloseWorkspace2(_lIgnoreChanges);
}
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim lIgnoreChanges As Long
lIgnoreChanges = 1

ret = hts.CloseWorkspace2(lIgnoreChanges)
```

SaveWorkspace2

内容

ワークスペースを保存します。

パラメータ

パラメータは、ありません。

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;

try
{
    hr = pHewServer1->SaveWorkspace2();
}
```

記述例 (Visual Basic)

```
Dim ret As Long

ret = hts.SaveWorkspace2
```

GetWorkspaceDirectory

内容

現在のワークスペースの絶対パスを取得します。

パラメータ

属性	型	内容
[out]	BSTR *_pbstrCurrentWorkspaceDirectory	ワークスペースの絶対パス

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
CString strTmp = _T("");
BSTR bstrCurrentWorkspaceDirectory = strTmp.AllocSysString();
//Call HewTargetServer function
CString strCurrentWorkspaceDirectory;
try
{
    hr = pHewServer1->GetWorkspaceDirectory(&bstrCurrentWorkspaceDirectory);
    strCurrentWorkspaceDirectory = bstrCurrentWorkspaceDirectory;
}
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim bstrCurrentWorkspaceDirectory As String

ret = hts.GetWorkspaceDirectory(bstrCurrentWorkspaceDirectory)
```

SaveSession2

内容

セッションファイルを保存します。

パラメータ

パラメータはありません。

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL

try
{
    hr = pHewServer1->SaveSession2();
}
```

記述例 (Visual Basic)

```
Dim ret As Long

ret = hts.SaveSession2
```

GetCurrentConfiguration2

内容

現在のビルド構成を取得します。

パラメータ

属性	型	内容
[out]	BSTR *p_bstrCurrentConfigurationName	ビルド構成名

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
CString strCurrentConfigurationName = _T("");
BSTR bstrCurrentConfigurationName;

//calling HewTargetServer function
try
{
    hr = pHewServer1->GetCurrentConfiguration2(&bstrCurrentConfigurationName);
    strCurrentConfigurationName = bstrCurrentConfigurationName;
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim p_bstrCurrentConfigurationName As String

ret = hts.GetCurrentConfiguration2(p_bstrCurrentConfigurationName)
```

SetCurrentConfiguration2

内容

現在有効なビルド構成を設定します。

パラメータ

属性	型	内容
[in]	BSTR _bstrConfiguration	ビルド構成名

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
BSTR bstrSetCurrentConfiguration;

try
{
    hr = pHewServer1->SetCurrentConfiguration2(bstrSetCurrentConfiguration);
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim bstrSetCurrentConfiguration As String

ret = hts.SetCurrentConfiguration2(bstrSetCurrentConfiguration)
```

GetConfigurations2

内容

プロジェクトにあるビルド構成をすべて取得します。

パラメータ

属性	型	内容
[out]	BSTR *p_bstrConfigurations	ビルド構成名 (名前が複数ある場合は、カンマで区切る (例) "DefaultSession, SimSessionSH-4")

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外 (例えば E_FAIL (0x80004005L)) を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
CString strTmp = _T("");
BSTR ol = strTmp.AllocSysString(); //CString -> BSTR 変換

//calling HewTargetServer function
CString sol;
try
{
    hr = pHewServer1->GetConfigurations2(&ol);
    sol = ol;
}
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim p_bstrConfigurations As String

ret = hts.GetConfigurations2(p_bstrConfigurations)
```

GetCurrentSession2

内容

現在のデバッグセッションを取得します。

パラメータ

属性	型	内容
[out]	BSTR *p_bstrCurrentSessionName	デバッグセッション名

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
CString strCurrentSessionName = _T("");
BSTR bstrCurrentSessionName;

//calling HewTargetServer function
try
{
    hr = pHewServer1->GetCurrentSession2(&bstrCurrentSessionName);
    strCurrentSessionName = bstrCurrentSessionName;
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim p_bstrCurrentSessionName As String

ret = hts.GetCurrentSession2(p_bstrCurrentSessionName)
```

SetCurrentSession2

内容

現在有効なデバッグセッションを設定します。

パラメータ

属性	型	内容
[in]	BSTR _bstrSession	デバッグセッション名

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
BSTR bstrSession;

//calling HewTargetServer function
try
{
    hr = pHewServer1->SetCurrentSession2(bstrSession);
}
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim bstrSession As String

ret = hts.SetCurrentSession2(bstrSession)
```

GetSessions2

内容

プロジェクトにある全デバッグセッションを取得します。

パラメータ

属性	型	内容
[out]	BSTR *p_bstrSessions	デバッグセッション名(名前が複数ある場合は、カンマで区切る) (例) "DefaultSession, SimSessionSH-4"

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;

//calling HewTargetServer function
CString strTmp = _T("");
BSTR o2 = strTmp.AllocSysString(); //CString -> BSTR 変換

CString so2;
try
{
    hr = pHewServer1->GetSessions2(&o2);
    so2 = o2;
}
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim p_bstrSessions As String

ret = hts.GetSessions2(p_bstrSessions)
```

GetCurrentProject2

内容

現在のプロジェクトを取得します。

パラメータ

属性	型	内容
[out]	BSTR *p_bstrCurrentProjectName	プロジェクト名

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
CString strCurrentProjectName = _T("");
BSTR bstrCurrentProjectName;

//calling HewTargetServer function
try
{
    hr = pHewServer1->GetCurrentProject2(&bstrCurrentProjectName);
    strCurrentProjectName = bstrCurrentProjectName;
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim p_bstrCurrentProjectName As String

ret = hts.GetCurrentProject2(p_bstrCurrentProjectName)
```

SetCurrentProject2

内容

アクティブプロジェクトを設定します。

パラメータ

属性	型	内容
[in]	BSTR _bstrProjectName	プロジェクト名

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
BSTR bstrProjectName;

//calling HewTargetServer function
try
{
    hr = pHewServer1->SetCurrentProject2(bstrProjectName);
}
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim bstrProjectName As String

ret = hts.SetCurrentProject2(bstrProjectName)
```

GetProjects2

内容

プロジェクト名をすべて取得します。

パラメータ

属性	型	内容
[out]	BSTR *p_bstrProjectNames	プロジェクト名 名前が複数ある場合は、カンマで区切る (例)"Project1, Project2"

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
CString strProjectNames = _T("");
BSTR bstrProjectNames;

//calling HewTargetServer function
try
{
    hr = pHewServer1->GetProjects2(&bstrProjectNames);
    strProjectNames = bstrProjectNames;
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim p_bstrProjectNames As String

ret = hts.GetProjects2(p_bstrProjectNames)
```

5.4.13 プロジェクト

AddFile2

内容

アクティブなプロジェクトにファイルを追加します。

パラメータ

属性	型	内容
[in]	BSTR _bstrFileName	ファイル名(パス名を含む)

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外 (例えば E_FAIL (0x80004005L)) を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
BSTR bstrFileName;

try
{
    hr = pHewServer1->AddFile2(bstrFileName);
}
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim bstrFileName As String

ret = hts.AddFile2(bstrFileName)
```

AddFiles2

内容

アクティブなプロジェクトに複数のファイルを追加します。

パラメータ

属性	型	内容
[in]	BSTR _bstrFileName	ファイル名(ファイルが複数ある場合は、カンマで区切る) (パス名を含む)

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外 (例えば E_FAIL (0x80004005L)) を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
BSTR bstrFileName;

try
{
    hr = pHewServer1->AddFiles2(bstrFileName);
}
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim bstrFileName As String

ret = hts.AddFiles2(bstrFileName)
```

DeleteFile2

内容

アクティブなプロジェクトからファイルを削除します。

パラメータ

属性	型	内容
[in]	BSTR _bstrFileName	ファイル名(パス名を含む)

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
BSTR bstrFileName;

//calling HewTargetServer function
try
{
    hr = pHewServer1->DeleteFile2(bstrFileName);
}
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim bstrFileName As String

ret = hts.DeleteFile2(bstrFileName)
```

DeleteFiles2

内容

アクティブなプロジェクトから複数のファイルを削除します。

パラメータ

属性	型	内容
[in]	BSTR _bstrFileName	ファイル名(ファイルが複数ある場合は、カンマで区切る) (パス名を含む)

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT          hr = E_FAIL;
BSTR bstrFileName;

//calling HewTargetServer function
try
{
    hr = pHewServer1->DeleteFiles2(bstrFileName);
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim bstrFileName As String

ret = hts.DeleteFiles2(bstrFileName)
```

AddProjectFileFolder

内容

カレントプロジェクトの **Projects** ツリーにフォルダを作成、追加します。

パラメータ

属性	型	内容
[in]	BSTR _bstrFolderName	フォルダ名 (階層がある場合は、円記号で区切る (例) "Folder1¥Subfolder")

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外 (例えば E_FAIL (0x80004005L)) を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
BSTR bstrAddFolder = m_AddFolder.AllocSysString();
//Call HewTargetServer function
try
{
    hr = pHewServer1->AddProjectFileFolder(bstrAddFolder);
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim bstrAddFolder As String

ret = hts.AddProjectFileFolder(bstrAddFolder)
```

RemoveProjectFileFolder

内容

カレントプロジェクトの Projects ツリーのフォルダを削除します。

パラメータ

属性	型	内容
[in]	BSTR _bstrFolderName	フォルダ名 (階層がある場合は、円記号で区切る (例) "Folder1¥Subfolder")

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外 (例えば E_FAIL (0x80004005L)) を返します。

注意事項

ファイルまたはサブフォルダのあるフォルダは削除できません。
フォルダ名を階層で指定した場合、最後の階層が削除されます。
(例) " Folder1¥Subfolder" を指定した場合、Subfolder のみ削除されます。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
BSTR bstrFolderName = m_RemoveFolder.AllocSysString();
//Call HewTargetServer function
try
{
    hr = pHewServer1->RemoveProjectFileFolder(bstrFolderName);
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim bstrFolderName As String

ret = hts.RemoveProjectFileFolder(bstrFolderName)
```

AddFileToFolder

内容

カレントプロジェクトの Projects ツリーにある指定したフォルダにファイルを登録します。

パラメータ

属性	型	内容
[in]	BSTR _bstrFileName	ファイル名 (パス名を含む)
[in]	BSTR _bstrFolderName	フォルダ名 (階層がある場合は、円記号で区切る (例) "Folder1¥Subfolder")

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外 (例えば E_FAIL (0x80004005L)) を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
BSTR bstrFileName = m_AddFile.AllocSysString();
BSTR bstrFolderName = m_AddFolder.AllocSysString();
//Call HewTargetServer function
try
{
    hr = pHewServer1->AddFileToFolder(bstrFileName, bstrFolderName);
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim bstrFileName As String
Dim bstrFolderName As String

ret = hts.AddFileToFolder(bstrFileName, bstrFolderName)
```

BuildProject2

内容

プロジェクトをビルドします。

パラメータ

パラメータはありません。

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT    hr = E_FAIL;

//calling HewTargetServer function
try
{
    hr = pHewServer1->BuildProject2();
}
```

記述例 (Visual Basic)

```
Dim ret As Long

ret = hts.BuildProject2
```

RebuildProject2

内容

プロジェクトをリビルドします。

パラメータ

パラメータはありません。

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT    hr = E_FAIL;

//calling HewTargetServer function
try
{
    hr = pHewServer1->RebuildProject2();
}
}
```

記述例 (Visual Basic)

```
Dim ret As Long

ret = hts.RebuildProject2
```

UpDateAllDependency2

内容

すべての依存関係を更新します。

パラメータ

パラメータはありません。

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
//calling HewTargetServer function
try
{
    hr = pHewServer1->UpDateAllDependency2();
}
```

記述例 (Visual Basic)

```
Dim ret As Long

ret = hts.UpDateAllDependency2
```

AddFileWithCompilerOption2

内容

プロジェクトにコンパイラオプションを設定してファイルを追加します。

パラメータ

属性	型	内容
[in]	BSTR _bstrAddFileName	ファイル名(パス名を含む)
[in]	BSTR _bstrIncludeDirectories	インクルードディレクトリ名 ディレクトリ名が複数ある場合は、カンマで区切る(例) "C:¥tmp, D:¥work"
[in]	BSTR _bstrDefines	定義 定義が複数ある場合は、カンマで区切る(例) "TMP1=C:¥tmp, TMP2=D:¥work"

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外 (例えば E_FAIL (0x80004005L)) を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
BSTR _bstrAddFileName;
BSTR _bstrIncludeDirectories;
BSTR _bstrDefines;

//calling HewTargetServer function
try
{
    hr=pHewServer1->AddFileWithCompilerOption2(_bstrAddFileName, _bstrIncludeDirectories, _bstrDefines);
}
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim bstrAddFileName As String
Dim bstrIncludeDirectories As String
Dim bstrDefines As String

ret = hts.AddFileWithCompilerOption2(bstrAddFileName, bstrIncludeDirectories, _
    bstrDefines)
```

GetLibraryOptions

内容

カレントプロジェクトのリンカのライブラリオプションを取得します。

パラメータ

属性	型	内容
[out]	BSTR *p_bstrLibraryOption	カレントプロジェクトのリンカのライブラリオプション 例 SHC : "LIB=c:¥test¥test1.lib,c:¥temp¥test2.lib" 例 M16C : "-LD "D:¥V540" -L "nc30lib""

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
CString strTmp = _T("");
BSTR ol = strTmp.AllocSysString(); //CString -> BSTR Conversion

//Call HewTargetServer function
CString sol;
try
{
    hr = pHewServer1->GetLibraryOptions(&ol);
    sol = ol;
}
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim p_bstrLibraryOption As String

ret = hts.GetLibraryOptions(p_bstrLibraryOption)
```

SetLibraryOptions

内容

カレントプロジェクトのリンクのライブラリオプションを設定します。
設定済みのライブラリオプションは破棄されます。

パラメータ

属性	型	内容
[in]	BSTR _bstrLibraryOption	カレントプロジェクトに設定するリンクのライブラリオプション

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
BSTR bstrSetLibraryOption;  
HRESULT hr = E_FAIL;  
  
//Call HewTargetServer function  
try  
{  
    hr = pHewServer1->SetLibraryOptions(bstrSetLibraryOption);  
}
```

記述例 (Visual Basic)

```
Dim ret As Long  
Dim bstrLibraryOption As String  
  
ret = hts.SetLibraryOptions(bstrLibraryOption)
```

GetLibraryFilesForConfiguration

内容

指定したプロジェクトの指定したコンフィグレーションからライブラリオプションを取得します。

パラメータ

属性	型	内容
[in]	BSTR _bstrProjectName	プロジェクト (空文字列を指定した場合は現在のプロジェクト)
[in]	BSTR _bstrConfiguration	コンフィグレーションの名前 (空文字列を指定した場合は現在のコンフィグレーション)
[out]	BSTR *_pbstrLibraryFiles	リンクのライブラリオプション

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外 (例えば E_FAIL (0x80004005L)) を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
BSTR bstrProject = m_GetLibraryFilesForConfiguration_Project.AllocSysString();
BSTR bstrConfiguration = m_GetLibraryFilesForConfiguration_Configuration.AllocSysString();
CString strTmp = _T("");
BSTR bstrLibraryFiles = strTmp.AllocSysString();
CString strLibraryFiles;

//Call HewTargetServer function
try
{
    hr = pHewServer1->GetLibraryFilesForConfiguration(bstrProject, bstrConfiguration,
                                                    &bstrLibraryFiles);
    strLibraryFiles = bstrLibraryFiles;
}
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim bstrProject As String
Dim bstrConfiguration As String
Dim p_bstrLibraryFiles As String

ret = hts.GetLibraryFilesForConfiguration(bstrProject, bstrConfiguration, bstrLibraryFiles)
```

SetLibraryFilesForConfiguration

内容

指定したプロジェクトの指定したコンフィグレーションにライブラリオプションを設定します。
設定済みのライブラリオプションは破棄されます。

パラメータ

属性	型	内容
[in]	BSTR _bstrProjectName	プロジェクト(空文字列を指定した場合は現在のプロジェクト)
[in]	BSTR _bstrConfiguration	コンフィグレーションの名前(空文字列を指定した場合は現在のコンフィグレーション)
[in]	BSTR _bstrLibraryFiles	リンクのライブラリオプション

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外 (例えば E_FAIL (0x80004005L)) を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
BSTR bstrProject = m_SetLibraryFilesForConfiguration_Project.AllocSysString();
BSTR bstrConfiguration = m_SetLibraryFilesForConfiguration_Configuration.AllocSysString();
BSTR bstrLibraryFiles = m_SetLibraryFilesForConfiguration_LibraryFiles.AllocSysString();

//Call HewTargetServer function
try
{
    hr = pHewServer1->SetLibraryFilesForConfiguration(bstrProject, bstrConfiguration,
                                                    bstrLibraryFiles);
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim bstrProject As String
Dim bstrConfiguration As String
Dim bstrLibraryFiles As String

ret = hts. SetLibraryFilesForConfiguration(bstrProject, bstrConfiguration, bstrLibraryFiles)
```

GetIncludeFileDirectories

内容

指定したプロジェクトの指定したコンフィグレーションにあるファイルからインクルードファイルオプションを取得します。

パラメータ

属性	型	内容
[in]	BSTR _bstrProjectName	プロジェクト (空文字列を指定した場合は現在のプロジェクト)
[in]	BSTR _bstrConfiguration	コンフィグレーションの名前 (空文字列を指定した場合は現在のコンフィグレーション)
[in]	BSTR _bstrFileName	ファイル名 (パス名を含む)
[out]	VARIANT *_pvtIncludeDirectories	コンパイラのインクルードファイルオプション (VT_ARRAY VT_VARIANT。配列の各要素は VT_BSTR 型の VARIANT。)

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外 (例えば E_FAIL (0x80004005L)) を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
BSTR bstrProject = m_GetIncludeFileDirectories_Project.AllocSysString();
BSTR bstrConfiguration = m_GetIncludeFileDirectories_Configuration.AllocSysString();
BSTR bstrFile = m_GetIncludeFileDirectories_File.AllocSysString();
CString strTmp = _T("");
VARIANT variantIncludeDirectories;

//Call HewTargetServer function
try
{
    hr =
        pHewServer1->GetIncludeFileDirectories(bstrProject, bstrConfiguration, bstrFile,
                                                &variantIncludeDirectories);
}
}
```

記述例 (Visual Basic 6.0)

```
Dim ret As Long
Dim bstrProject As String
Dim bstrConfiguration As String
Dim bstrFile As String
Dim vtIncludeDirectories As Variant
ret =
    hts.GetIncludeFileDirectoriesGetIncludeFileDirectories(bstrProject, bstrConfiguration, bstrFile,
                                                            vtIncludeDirectories)
```

記述例 (Visual Basic 2005)

```
Dim ret As Integer
Dim bstrProject As String
Dim bstrConfiguration As String
Dim bstrFile As String
Dim vtIncludeDirectories As Object
ret =
    hts.GetIncludeFileDirectoriesGetIncludeFileDirectories(bstrProject, bstrConfiguration, bstrFile,
                                                            vtIncludeDirectories)
```

SetIncludeFileDirectories

内容

指定したプロジェクトの指定したコンフィグレーションにあるファイルにインクルードファイルオプションを設定します。

パラメータ

属性	型	内容
[in]	BSTR _bstrProjectName	プロジェクト (空文字列を指定した場合は現在のプロジェクト)
[in]	BSTR _bstrConfiguration	コンフィグレーションの名前 (空文字列を指定した場合は現在のコンフィグレーション)
[in]	BSTR _bstrFileName	ファイル名 (パス名を含む)
[in]	VARIANT _vtIncludeDirectories	コンパイラのインクルードファイルオプション (VT_ARRAY VT_BSTR)
[in]	long _lSettingMode	0: 既存のオプションに追加します 1: 既存のオプションを置き換えます

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外 (例えば E_FAIL (0x80004005L)) を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
long lMode;
BSTR bstrProject = m_SetIncludeFileDirectories_Project.AllocSysString();
BSTR bstrConfiguration = m_SetIncludeFileDirectories_Configuration.AllocSysString();
BSTR bstrFile = m_SetIncludeFileDirectories_File.AllocSysString();
VARIANT vtIncludeDirectories;

//Call HewTargetServer function
try
{
    hr = pHewServer1->SetIncludeFileDirectories(bstrProject, bstrConfiguration, bstrFile,
vtIncludeDirectories, lMode);
}
```

記述例 (Visual Basic 6.0)

```
Dim ret As Long
Dim lMode As Long
Dim bstrProject As String
Dim bstrConfiguration As String
Dim bstrFile As String
Dim vtIncludeDirectories As Variant

ret = hts.SetIncludeFileDirectories(bstrProject, bstrConfiguration, bstrFile,
vtIncludeDirectories, lMode)
```

記述例 (Visual Basic 2005)

```
Dim ret As Integer
Dim iMode As Integer
Dim bstrProject As String
Dim bstrConfiguration As String
Dim bstrFile As String
Dim vtIncludeDirectories As Object

ret = hts.SetIncludeFileDirectories(bstrProject, bstrConfiguration, bstrFile,
vtIncludeDirectories, iMode)
```

GetCpuAndToolChainData

内容

指定したプロジェクトから CPU 種別、CPU シリーズ名、CPU タイプ名、コンパイラ種別、コンパイラ名、コンパイラバージョンを取得します。

パラメータ

属性	型	内容
[in]	BSTR _bstrProjectName	プロジェクト名
[out]	BSTR *_pbstrCPUFamily	CPU 種別
[out]	BSTR *_pbstrCPUSeries	CPU シリーズ名
[out]	BSTR *_pbstrCPUType	CPU タイプ名
[out]	BSTR *_pbstrToolChainFamily	コンパイラ種別
[out]	BSTR *_pbstrToolChainName	コンパイラ名
[out]	BSTR *_pbstrToolChainVersion	コンパイラバージョン

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
BSTR bstrProjectName = m_GetCpuAndToolChainData_Project.AllocSysString();
CString strTmp = _T("");
BSTR bstrCPUFamily = strTmp.AllocSysString();
BSTR bstrCPUSeries = strTmp.AllocSysString();
BSTR bstrCPUType = strTmp.AllocSysString();
BSTR bstrToolChainFamily = strTmp.AllocSysString();
BSTR bstrToolChainName = strTmp.AllocSysString();
BSTR bstrToolChainVersion = strTmp.AllocSysString();

//Call HewTargetServer function
CString strCPUFamily;
CString strCPUSeries;
CString strCPUType;
CString strToolChainFamily;
CString strToolChainName;
CString strToolChainVersion;

try
{
    hr = pHewServer1->GetCpuAndToolChainData(bstrProjectName, &bstrCPUFamily,
&bstrCPUSeries, &bstrCPUType, &bstrToolChainFamily, &bstrToolChainName, &bstrToolChainVersion);
    strCPUFamily = bstrCPUFamily;
    strCPUSeries = bstrCPUSeries;
    strCPUType = bstrCPUType;
    strToolChainFamily = bstrToolChainFamily;
    strToolChainName = bstrToolChainName;
    strToolChainVersion = bstrToolChainVersion;
}
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim bstrProjectName As String
Dim bstrCPUFamily As String
Dim bstrCPUSeries As String
Dim bstrCPUType As String
Dim bstrToolChainFamily As String
Dim bstrToolChainName As String
Dim bstrToolChainVersion As String

ret = hts.GetCpuAndToolChainData(bstrProjectName, bstrCPUFamily, bstrCPUSeries, bstrCPUType,
bstrToolChainFamily, bstrToolChainName, bstrToolChainVersion)
```

SetBuildExcludeFiles

内容

指定したファイルをビルドから除外します。

パラメータ

属性	型	内容
[in]	BSTR _bstrFileNames	ファイル名(ファイルが複数ある場合は、カンマで区切る)

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外 (例えば E_FAIL (0x80004005L)) を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
BSTR bstrBuildExcludeFiles;

//Call HewTargetServer function
try
{
    hr = pHewServer1->SetBuildExcludeFiles(bstrBuildExcludeFiles);
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim bstrBuildExcludeFiles As String

ret = hts.SetBuildExcludeFiles(bstrBuildExcludeFiles)
```

SetBuildIncludeFiles

内容

指定したファイルをビルド対象に含めます。

パラメータ

属性	型	内容
[in]	BSTR _bstrFileNames	ファイル名(ファイルが複数ある場合は、カンマで区切る)

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
BSTR bstrBuildIncludeFiles;

//Call HewTargetServer function
try
{
    hr = pHewServer1->SetBuildIncludeFiles (bstrBuildIncludeFiles);
}
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim bstrBuildIncludeFiles As String

ret = hts. SetBuildIncludeFiles(bstrBuildIncludeFiles)
```

5.4.15 ファイル関係

OpenFileAtLine2

内容

ファイル名、行番号を指定しファイルを開きます。

パラメータ

属性	型	内容
[in]	BSTR _bstrOpenFileName	ファイル名(パス名を含む)
[in]	long _lLine	行番号

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
BSTR bstrOpenFileName;
long _lLine = 1;

//calling HewTargetServer function
try
{
    hr = pHewServer1->OpenFileAtLine2(bstrOpenFileName, _lLine);
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim bstrOpenFileName As String
Dim lLine As Long
lLine = 1

ret = hts.OpenFileAtLine2(bstrOpenFileName, lLine)
```

GetSourceFiles2

内容

プロジェクトにあるソースファイル名 (*.cpp, *.src 等) をすべて取得します。
ファイル名は、絶対パスで出力します。

パラメータ

属性	型	内容
[out]	BSTR *p_bstrSourceFiles	ソースファイル名(名前が複数ある場合は、カンマで区切る (例)"c:\¥sample1.cpp, c:\¥sample2.cpp")

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外 (例えば E_FAIL (0x80004005L)) を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
CString strSourceFiles = _T("");
BSTR bstrSourceFiles = strSourceFiles.AllocSysString(); //CString -> BSTR 変換

//calling HewTargetServer function
try
{
    hr = pHewServer1->GetSourceFiles2(&bstrSourceFiles);
    strSourceFiles = bstrSourceFiles;
}
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim p_bstrSourceFiles As String

ret = hts.GetSourceFiles2(p_bstrSourceFiles)
```

GetDownloadModules2

内容

プロジェクトにあるモジュールファイル名 (*.abs 等) をすべて取得します。
ファイル名は、絶対パスで出力します。

パラメータ

属性	型	内容
[out]	BSTR *p_bstrDownloadModules	モジュールファイル名(名前が複数ある場合は、カンマで区切る(例)"c:\¥sample1.abs, c:\¥sample2.abs")

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外 (例えば E_FAIL (0x80004005L)) を返します。

記述例 (Visual C++)

```
HRESULT    hr = E_FAIL;
CString    strDownloadModules = _T("");
BSTR bstrDownloadModules = strDownloadModules.AllocSysString(); //CString -> BSTR 変換

//calling HewTargetServer function
try
{
    hr = pHewServer1->GetDownloadModules2(&bstrDownloadModules);
    strDownloadModules = bstrDownloadModules;
}
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim p_bstrDownloadModules As String

ret = hts.GetDownloadModules2(p_bstrDownloadModules)
```

GetDependentFiles2

内容

プロジェクトにある依存ファイル名(*.h, *.inc 等)をすべて取得します。
ファイル名は、絶対パスで出力します。

パラメータ

属性	型	内容
[out]	BSTR *p_bstrDependentFiles	依存ファイル名(名前が複数ある場合は、カンマで区切る (例)"c:\¥sample1.h, c:\¥sample2.h")

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外 (例えば E_FAIL (0x80004005L)) を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
CString strDependentFiles = _T("");
BSTR bstrDependentFiles = strDependentFiles.AllocSysString(); //CString -> BSTR 変換

//calling HewTargetServer function
try
{
    hr = pHewServer1->GetDependentFiles2(&bstrDependentFiles);
    strDependentFiles = bstrDependentFiles;
}
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim p_bstrDependentFiles As String

ret = hts.GetDependentFiles2(p_bstrDependentFiles)
```

SetCoverageRange2

内容

カバレッジの範囲を設定します。

パラメータ

属性	型	内容
[in]	long _lStartAddress	開始アドレス
[in]	long _lEndAddress	終了アドレス

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
long _lStartAddress;
long _lEndAddress;

//calling HewTargetServer function
try
{
    hr = pHewServer1->SetCoverageRange2(_lStartAddress, _lEndAddress);
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim lStartAddress As Long
Dim lEndAddress As Long

ret = hts.SetCoverageRange2(lStartAddress, lEndAddress)
```

注意事項

カバレッジの範囲を設定すると同時に、カバレッジが有効になります。

GetCoverageRange2

内容

カバレッジの範囲を取得します。

パラメータ

属性	型	内容
[out]	long *p_lStartAddress	開始アドレス
[out]	long *p_lEndAddress	終了アドレス

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
long lStartAddress;
long lEndAddress;

//calling HewTargetServer function
try
{
    hr = pHewServer1->GetCoverageRange2(&lStartAddress, &lEndAddress);
}
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim p_lStartAddress As Long 'Visual Basic 2005 の場合は"As Integer"
Dim p_lEndAddress As Long 'Visual Basic 2005 の場合は"As Integer"

ret = hts.GetCoverageRange2(p_lStartAddress, p_lEndAddress)
```

SetCoverageDisable2

内容

カバレッジを無効に設定します。

パラメータ

パラメータはありません。

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT    hr = E_FAIL;

//calling HewTargetServer function
try
{
    hr = pHewServer1->SetCoverageDisable2();
}
```

記述例 (Visual Basic)

```
Dim ret As Long

ret = hts.SetCoverageDisable2
```

SetCoverageEnable2

内容

カバレッジを有効に設定します。

パラメータ

パラメータはありません。

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT    hr = E_FAIL;

//calling HewTargetServer function
try
{
    hr = pHewServer1->SetCoverageEnable2();
}
```

記述例 (Visual Basic)

```
Dim ret As Long

ret = hts.SetCoverageEnable2
```

ClearCoverage2

内容

カバレッジ情報をクリアします。

パラメータ

パラメータはありません。

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT    hr = E_FAIL;

//calling HewTargetServer function
try
{
    hr = pHewServer1->ClearCoverage2();
}
```

記述例 (Visual Basic)

```
Dim ret As Long

ret = hts.ClearCoverage2
```

GetCoverageStatus2

内容

カバレッジの状態を取得します。

パラメータ

属性	型	内容
[out]	long *p_lStatus	カバレッジの状態(1:有効、0:無効)

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
long lStatus;

//calling HewTargetServer function
try
{
    hr = pHewServer1->GetCoverageStatus2(&lStatus);
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim p_lStatus As Long 'Visual Basic 2005 の場合は"As Integer"

ret = hts.GetCoverageStatus2(p_lStatus)
```

LoadCoverage2

内容

カバレッジ情報をロードします。

パラメータ

属性	型	内容
[in]	BSTR _bstrLoadFileName	ファイル名

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
BSTR bstrLoadFileName;

//calling HewTargetServer function
try
{
    hr = pHewServer1->LoadCoverage2(bstrLoadFileName);
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim bstrLoadFileName As String

ret = hts.LoadCoverage2(bstrLoadFileName)
```

SaveCoverage2

内容

カバレッジ情報をセーブします。

パラメータ

属性	型	内容
[in]	BSTR _bstrSaveFileName	ファイル名

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
BSTR bstrSaveFileName;

//calling HewTargetServer function
try
{
    hr = pHewServer1->SaveCoverage2(bstrSaveFileName);
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim bstrSaveFileName As String

ret = hts.SaveCoverage2(bstrSaveFileName)
```

GetErrorString2

内容

エラー番号に対応したエラーメッセージを取得します。

パラメータ

属性	型	内容
[in]	long lError	エラー番号
[out]	BSTR *p_bstrError	エラーメッセージ

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
long lError;
BSTR bstrErr;

try
{
    hr = pHewServer1->GetErrorString2(lError, &bstrErr);
}
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim lError As Long
Dim p_bstrErr As String

ret = hts.GetErrorString2(lError, p_bstrErr)
```

GetHewStatus2

内容

現在の High-performance Embedded Workshop の状態を取得します。

パラメータ

属性	型	内容
[out]	long *p_lTargetReset	ターゲットリセット時には1が返却、それ以外は0*
[out]	long *p_lTargetExecStatus	ユーザプログラム実行中は1が返却、それ以外は0
[out]	long *p_lMemoryReset	メモリ内容が更新されているときには1が返却、それ以外は0*
[out]	long *p_lRegisterReset	レジスタ値が更新されているときには1が返却、それ以外は0*
[out]	long *p_lLinkStatus	ターゲット接続時には1が返却、それ以外は0
[out]	long *p_lPlatformInitialize	ターゲット初期化後には1が返却、それ以外は0*
[out]	long *p_lLoadingStatus	プログラムロード後には1が返却、それ以外は0

*:本関数が呼び出されると、本フラグは0にリセットされます。

戻り値

メソッドが正常に終了した場合は1、エラーが発生した場合は0を返します。

記述例 (Visual C++)

```
HRESULT hr;
long lTargetReset;
long lTargetExecStatus;
long lMemoryReset;
long lRegisterReset;
long lLinkStatus;
long lPlatformInitialize;
long lLoadingStatus;

//calling HewTargetServer function
try
{
    hr = pHewServer1->GetHewStatus2(&lTargetReset, &lTargetExecStatus,
        &lMemoryReset,
        &lRegisterReset,
        &lLinkStatus,
        &lPlatformInitialize,
        &lLoadingStatus
    );
}
}
```

記述例 (Visual Basic 6.0)

```
Dim ret As Long
Dim p_lTargetReset As Long
Dim p_lTargetExecStatus As Long
Dim p_lMemoryReset As Long
Dim p_lRegisterReset As Long
Dim p_lLinkStatus As Long
Dim p_lPlatformInitialize As Long
Dim p_lLoadingStatus As Long

ret = hts.GetHewStatus2(p_lTargetReset, p_lTargetExecStatus, p_lMemoryReset, _
    p_lRegisterReset, p_lLinkStatus, p_lPlatformInitialize, p_lLoadingStatus)
```

記述例 (Visual Basic 2005)

```
Dim ret As Integer
Dim p_iTargetReset As Integer
Dim p_iTargetExecStatus As Integer
Dim p_iMemoryReset As Integer
Dim p_iRegisterReset As Integer
Dim p_iLinkStatus As Integer
Dim p_iPlatformInitialize As Integer
Dim p_iLoadingStatus As Integer

ret = hts.GetHewStatus2(p_iTargetReset, p_iTargetExecStatus, p_iMemoryReset, _
    p_iRegisterReset, p_iLinkStatus, p_iPlatformInitialize, p_iLoadingStatus)
```

GetHewStatusEx2

内容

High-performance Embedded Workshop のステータス(起動、ワークスペースオープン、ビルド)を取得します。

パラメータ

属性	型	内容
[out]	long *p_lInvokeHew	High-performance Embedded Workshop の起動状態(0:未起動、1:起動済み)
[out]	long *p_lOpenWorkspace	ワークスペースの状態(0:未オープン、1:オープン済み) ※ 実際にはターゲットと接続中のみ 1 になります。 ワークスペースオープン中かどうかは GetWorkspaceDirectory を呼び出して判定してください。
[out]	long *p_lBuildProject	ビルド状態(0:ビルド停止、1:ビルド中)

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外(例えば E_FAIL (0x80004005L)) を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
long lInvokeHew;
long lOpenWorkspace;
long lBuildProject;

//calling HewTargetServer function
try
{
    hr = pHewServer1->GetHewStatusEx2(&lInvokeHew, &lOpenWorkspace, &lBuildProject);
}
}
```

記述例 (Visual Basic 6.0)

```
Dim ret As Long
Dim p_lInvokeHew As Long
Dim p_lOpenWorkspace As Long
Dim p_lBuildProject As Long

ret = hts.GetHewStatusEx2(p_lInvokeHew, p_lOpenWorkspace, p_lBuildProject)
```

記述例 (Visual Basic 2005)

```
Dim ret As Integer
Dim p_iInvokeHew As Integer
Dim p_iOpenWorkspace As Integer
Dim p_iBuildProject As Integer

ret = hts.GetHewStatusEx2(p_iInvokeHew, p_iOpenWorkspace, p_iBuildProject)
```

GetTargetName2

内容

現在接続しているターゲット名を取得します。

パラメータ

属性	型	内容
[out]	BSTR* p_bstrName	ターゲット名

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
BSTR bstrName;

//calling HewTargetServer function
try
{
    //get target name
    hr = pHewServer1->GetTargetName2(&bstrName);
}
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim p_bstrName As String

ret = hts.GetTargetName2(p_bstrName)
```

GetHewVersion

内容

High-performance Embedded Workshop のバージョンを取得します。

パラメータ

属性	型	内容
[out]	BSTR* p_bstrHewVersion	バージョン

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
CString strTmp = _T("");
BSTR bstrHewVersoin = strTmp.AllocSysString(); //CString -> BSTR Conversion
//Call HewTargetServer function
try
{
    hr = pHewServer1->GetHewVersion(&bstrHewVersoin);
}
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim p_bstrHewVersion As String

ret = hts.GetHewVersion(p_bstrHewVersion)
```

Command

内容

High-performance Embedded Workshop のコマンドを実行します。

パラメータ

属性	型	内容
[in]	BSTR _bstrCommandLine	コマンド
[out]	BSTR *p_bstrCommandMessage	コマンドが出力したメッセージ

戻り値

戻り値は、HRESULT 型で返します。成功した場合は 0 を、エラーが発生した場合は、0 以外（例えば E_FAIL (0x80004005L)）を返します。

記述例 (Visual C++)

```
HRESULT hr = E_FAIL;
BSTR bstrCommandLine= m_Command.AllocSysString();
CString strTmp = _T("");
BSTR bstrCommandMessage = strTmp.AllocSysString();
//Call HewTargetServer function
try
{
    hr = pHewServer1->Command(bstrCommandLine, &bstrCommandMessage);
}
```

記述例 (Visual Basic)

```
Dim ret As Long
Dim bstrCommandLine As String
Dim p_bstrCommandMessage As String

ret = hts.Command(bstrCommandLine, p_bstrCommandMessage)
```

注意事項

- (1) Log コマンドは指定できません。
- (2) Assemble コマンド(*)の変更モードでは、アセンブルするメモリアドレスを出力しません。
* コマンドのサポートはデバッガに依存します。
- (3) 本機能を使用した場合のコマンド実行の応答時間は、High-performance Embedded Workshop のコマンドラインウィンドウにおけるコマンド実行時の応答時間に比べて長くなります。

5.5 High-performance Embedded Workshopで取得可能なイベント

イベント種類	イベント発行タイミング
Event1_ToClient_TargetReset	ターゲットリセット時に発行
Event2_ToClient_Go	ターゲットプログラム実行時に発行
Event3_ToClient_Stop	ターゲットプログラム停止時に発行
Event4_ToClient_MemoryReset	SETMEMORY2 メソッドや、メモリウィンドウ、コマンドライン等、HEWによりメモリを更新した場合に発行
Event5_ToClient_RegisterReset	SetPCAddress2 メソッドや、レジスタウィンドウ、コマンドライン等、HEWによりレジスタを更新した場合に発行
Event6_ToClient_LinkUp	ターゲットリンクアップ時に発行
Event7_ToClient_LinkDown	ターゲットリンクダウン時に発行
Event8_ToClient_PlatformInitialize	プラットフォーム初期化時に発行
Event9_ToClient_Download	プログラムダウンロード時に発行
Event10_ToClient_Unload	プログラムアンロード時に発行
Event11_ToClient_HewInvoke	High-performance Embedded Workshop 起動時に発行
Event12_ToClient_WorkspaceOpen	ワークスペースオープン時に発行 ※ このイベントは実際にはターゲットと接続したときに発行されます。 ワークスペースオープンと同時にターゲットと接続するような設定でご使用ください。
Event13_ToClient_ProjectBuild	プロジェクトビルド時に発行

注意事項

イベントは毎回は送信しないようになっています。詳細は 4. 章の「注意事項 (1) 毎回イベントを取得する場合」をご参照下さい。

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2006.06.20	-	初版発行
2.00	2006.07.21	32, 34	説明の訂正 (GetMemory()と GetDirectMemory()メソッドの内容)
3.00	2006.10.20	-	新規メソッドの説明を追加
4.00	2007.09.20	- 13	新規メソッドの説明を追加 Visual Basic .NET 移行時の注意点追加
5.00	2008.11.05	-	Visual C++/Visual Basic 2005 用の記述追加 新規メソッドの説明を追加
6.00	2009.04.20	-	SetMemory2 の記述例を更新 FileName 引数の説明を更新 InvokeHew(WithNoDialog)メソッドの注意事項を削除
7.00	2010.07.01	-	旧社名の表記に関する変更、および「ご注意書き」の改訂
8.00	2010.11.01	-	GetHewStatusEx2() と WorkspaceOpen イベントに注意点を追加

HewTargetServer
ユーザーズマニュアル

発行年月日 2010年11月1日 Rev.8.00

発行 ルネサス エレクトロニクス株式会社
〒211-8668 神奈川県川崎市中原区下沼部 1753

編集 株式会社ルネサス ソリューションズ
ツール開発第一部



ルネサスエレクトロニクス株式会社

■営業お問合せ窓口

<http://www.renesas.com>

※営業お問合せ窓口の住所・電話番号は変更になることがあります。最新情報につきましては、弊社ホームページをご覧ください。

ルネサス エレクトロニクス販売株式会社 〒100-0004 千代田区大手町2-6-2 (日本ビル)

(03)5201-5307

■技術的なお問合せおよび資料のご請求は下記へどうぞ。
総合お問合せ窓口：<http://japan.renesas.com/inquiry>

HewTargetServer ユーザーズマニュアル