

---

JP1 Version 8

# JP1/Script ( Windows(R) 用 )

解説・文法書

3020-3-K55-20

## マニュアルの購入方法

このマニュアル，および関連するマニュアルをご購入の際は，  
巻末の「ソフトウェアマニュアルのサービス ご案内」をご参  
照ください。

# HITACHI

## 対象製品

P-2412-3F84 JP1/Script 08-00 (適用 OS : Windows Server 2003 x64 , 64 ビットバージョン Windows Server 2003 ( IPF ) , Windows Server 2003 , Windows XP )

P-2812-3F84 JP1/Script 08-00 (適用 OS : 64 ビットバージョン Windows Server 2003 ( IPF ) )

P-2A12-3F84 JP1/Script 08-00 (適用 OS : Windows Vista Business Edition/Enterprise Edition/Ultimate Edition , Windows Vista x64 Business Edition/Enterprise Edition/Ultimate Edition )

## 輸出時の注意

本製品を輸出される場合には、外国為替および外国貿易法ならびに米国の輸出管理関連法規などの規制をご確認の上、必要な手続きをお取りください。

なお、ご不明な場合は、弊社担当営業にお問い合わせください。

## 商標類

Itanium は、アメリカ合衆国および他の国におけるインテルコーポレーションまたはその子会社の登録商標です。

Microsoft は、米国およびその他の国における米国 Microsoft Corp. の登録商標です。

MS-DOS は、米国およびその他の国における米国 Microsoft Corp. の登録商標です。

Visual Basic は、米国およびその他の国における米国 Microsoft Corp. の登録商標です。

Windows は、米国およびその他の国における米国 Microsoft Corp. の登録商標です。

Windows Server は、米国およびその他の国における米国 Microsoft Corp. の商標です。

Windows Vista は、米国 Microsoft Corporation の米国及びその他の国における登録商標または商標です。

## 発行

2006 年 4 月 ( 第 1 版 ) 3020-3-K55

2007 年 6 月 ( 第 2 版 ) 3020-3-K55-20

## 著作権

All Rights Reserved. Copyright (C) 2006, 2007, Hitachi, Ltd.

## 変更内容

変更内容 (3020-3-K55-20) JP1/Script 08-00

追加・変更内容	変更箇所
Windows Vista に対応した。	図 1-1 , 表 1-2 , 1.8 , 2.1.1 , 2.1.2 , 2.1.3 , 2.1.6 , 2.1.7 , 2.6 , 3.1.12 , 3.1.13 , 3.1.14 , 3.2.13 , 3.4 , 3.4.3 , 3.7 , 4.1.8 , 4.1.9 , 4.1.10 , 4.1.14 , 4.1.20 , 4.1.22 , 4.1.24 , 4.1.25 , 4.1.26 , 4.1.27 , 4.1.28 , 表 5-7 , 表 5-8 , 表 6-2 , 表 6-6 , 8.2.3 , 8.5.1 , 8.5.2 , 8.5.3 , 8.5.4 , 8.5.10 , 8.5.11 , 8.5.12 , 8.5.13 , 8.5.15 , 8.5.16 , 8.5.17 , 8.5.18 , 8.5.19 , 8.5.20 , 8.5.21 , 8.5.22 , 8.5.23 , 8.5.24 , 8.5.30 , 8.5.33 , 8.6.2 , 8.9.4 , 8.9.5 , 8.9.6 , 8.9.7 , 8.9.8 , 8.9.9 , 8.10.1 , 8.10.2 , 8.14.3 , 9.2.1 , 9.2.2 , 9.2.3 , 9.2.4 , 9.3.1 , 9.5.3 , 9.5.4 , 9.5.5 , 9.5.6 , 9.5.7 , 9.5.8 , 9.5.9 , 9.5.12 , 9.7.1 , 9.7.2 , 9.7.3 , 9.7.4 , 9.8.1 , 9.8.2 , 9.8.3 , 9.9.1
実行するファイルパスと実行フォルダの変数名に変数の一つしか指定できない旨の説明を削除した。	4.4.37
資料採取ツールの記述を変更した。	5.5.1
Windows NT , Windows 2000 に関する内容を削除した。	5.5.2
JP1/Script の資料採取の内容を変更した。	5.5.4
Windows イベントログの採取を追加した。	5.5.5
_ERR_FILE_POSITION_ の説明中の数値を 2,147,483,648 から 2,147,483,647 に変更した。	表 6-2
_ERR_SERVICE_NOT_BEGIN_ を追加した。	表 6-2
コマンドラインのパラメタ指定が、スクリプト起動時とスクリプトの実行環境ファイルの両方にある場合の説明を追加した。	6.2.4
イベントコンソールについて参照するマニュアルを変更した。	表 8-1
Dim (配列) コマンドの説明を変更した。	8.2.2
注意事項に、取得する環境変数の値の説明を追加した。	8.2.4
注意事項に、フォルダ名を指定した場合の動作を追加した。	8.9.6
注意事項に、停止中の状態にならないサービスに対しての実行を禁止する説明を追加した。	9.5.4 , 9.5.6
注意事項に、実行中の状態にならないサービスに対しての実行を禁止する説明を追加した。	9.5.5
Code 省略時の値を 17 から 0 に変更した。	9.8.3

単なる誤字・脱字などはお断りなく訂正しました。



# はじめに

---

このマニュアルは、次に示すプログラムプロダクトの機能概要と使い方について説明したものです。

- P-2412-3F84 JP1/Script
- P-2812-3F84 JP1/Script
- P-2A12-3F84 JP1/Script

## 対象読者

このマニュアルは、次の方にお読みいただくことを前提に説明しています。

- ほかのシステムでジョブ制御の経験がある方
- 使用するオペレーティングシステムの基本的な操作、および Basic 言語を理解している方
- JP1 関連製品の機能を理解している方

## マニュアルの構成

このマニュアルは、次に示す編、章と付録から構成されています。

### 第 1 編 解説編

#### 第 1 章 JP1/Script の概要

JP1/Script の機能概要と特長、全体構成、プログラムの種類、ファイル、システム構成について説明しています。また、JP1/Script の操作の流れについても説明しています。

#### 第 2 章 JP1/Script を利用するための準備

JP1/Script のインストール方法、アンインストール方法、他のコンピュータとの通信時のアカウント、および各種システム環境での環境設定方法について説明しています。また、JP1/Script の起動と終了についても説明しています。

#### 第 3 章 JP1/Script の操作

JP1/Script のマネージャ、エディタ、簡易入力、トレースビューア、トレースファイル、メニューエディタ、プロセスビューア、実行環境ファイルコンバータの操作方法について説明しています。操作の中で表示されるダイアログの説明は、第 4 章で詳説しています。

#### 第 4 章 ダイアログの説明

Script マネージャウィンドウ、Script エディタウィンドウ、Script トレースビューアウィンドウ、および Script メニューエディタウィンドウのダイアログについて説明しています。

#### 第 5 章 トラブルへの対処方法

JP1/Script で発生するトラブルの種類と、その対処方法について説明しています。

## 第 2 編 文法編

### 第 6 章 JP1/Script の規則

JP1/Script で使用するスクリプトに関する規則，およびコマンドラインに関する規則について説明しています。

### 第 7 章 ステートメント

スクリプトを作成するときに使用できるステートメントについて説明しています。

### 第 8 章 基本コマンド

スクリプトを作成するときに使用できる基本コマンドについて説明しています。

### 第 9 章 特殊コマンド

スクリプトを作成するときに使用できる特殊コマンドについて説明しています。

### 第 10 章 スクリプト制御インタフェース

実行中のスクリプトを制御する，スクリプト制御インタフェース（API）について説明しています。

### 第 11 章 スクリプト OLE コントロール

スクリプトコマンドを実行する，スクリプト OLE コントロールについて説明しています。

### 付録 A Script トレースのファイル出力形式

Script トレースで扱うファイルの出力形式について説明しています。

### 付録 B サンプルファイル

サンプルファイルの概要，およびサンプルファイルの実行手順について説明しています。

### 付録 C エラー詳細コード

JP1/Script のコマンド実行後の予約変数 `_RTN_` に設定される値，およびエラー内容について説明しています。

### 付録 D 用語解説

JP1/Script で使用する用語の意味を解説しています。

## 関連マニュアル

このマニュアルの関連マニュアルを次に示します。必要に応じてお読みください。

- JP1 Version 8 JP1/Integrated Management - Manager システム構築・運用ガイド ( 3020-3-K01 )
- JP1 Version 8 JP1/Integrated Management - Manager リファレンス ( 3020-3-K02 )
- JP1 Version 8 JP1/Automatic Job Management System 2 解説 ( 3020-3-K21 )
- JP1 Version 8 JP1/Automatic Job Management System 2 設計・運用ガイド ( 3020-3-K22 )
- JP1 Version 8 JP1/Automatic Job Management System 2 セットアップガイド ( 3020-3-K23 )
- JP1 Version 8 JP1/Automatic Job Management System 2 操作ガイド ( 3020-3-K24 )
- JP1 Version 8 JP1/Automatic Job Management System 2 連携ガイド ( 3020-3-K27 )
- JP1 Version 8 JP1/Cm2/SNMP System Observer ( 3020-3-L22 )

- JP1 Version 7i JP1/Performance Management/SNMP System Observer ( 3020-3-F69 )
- JP1 Version 6 統合ネットワーク管理システム サーバシステム管理 ( 3000-3-749 )

## 図中で使用する記号

このマニュアルでは、図中で使用する記号を次のように定義しています。

●PC（サーバ又はクライアント）



●入出力の動作



●ホスト



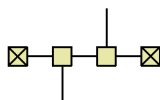
●プログラム



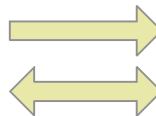
●ファイル



●バス形のLAN



●データの流れ



●制御の流れ



## このマニュアルでの表記

このマニュアルでは、製品名称または総称を次に示す略称で表記しています。

製品名称または総称	略称
Itanium(R) Processor Family	IPF
JP1/Automatic Job Management System 2	JP1/AJS2
JP1/Automatic Job Management System 2 - Client Toolkit	JP1/AJS2 - Client Toolkit
JP1/Automatic Job Management System 2 - Manager	JP1/AJS2 - Manager
JP1/Automatic Job Management System 2 - View	JP1/AJS2 - View
JP1/Automatic Job Scheduler	JP1/AJS
JP1/Cm2/SNMP System Observer	JP1/Cm2/SSO
JP1/Integrated Manager	JP1/IM
JP1/Integrated Management	
JP1/Integrated Manager - Console View	
JP1/Integrated Manager - Central Console	
JP1/Integrated Manager - Satellite Console	
JP1/Integrated Manager - SDK for Console	
JP1/Integrated Management - Manager	JP1/IM - Console
JP1/Network Batch Queuing System	
	JP1/NetBatch

製品名称または総称	略称
JP1/Performance Management/SNMP System Observer	JP1/PFM/SSO
Microsoft(R) MS-DOS(R) Operating System Version 5.0/V 以降	MS-DOS
64 ビットバージョン Microsoft(R) Windows Server(TM) 2003, Enterprise Edition	Windows Server 2003
Microsoft(R) Windows Server(TM) 2003, Enterprise Edition 日本語版	
Microsoft(R) Windows Server(TM) 2003, Standard Edition 日本語版	
Microsoft(R) Windows Server(TM) 2003 R2 , Enterprise Edition 日本語版	
Microsoft(R) Windows Server(TM) 2003 R2, Standard Edition 日本語版	
Microsoft(R) Windows Server(TM) 2003, Standard x64 Edition 日本語版	Windows Server 2003 x64
Microsoft(R) Windows Server(TM) 2003, Enterprise x64 Edition 日本語版	
Microsoft(R) Windows Server(TM) 2003 R2, Standard x64 Edition 日本語版	
Microsoft(R) Windows Server(TM) 2003 R2, Enterprise x64 Edition 日本語版	
Microsoft(R) Windows Vista(TM) Business Edition	Windows Vista
Microsoft(R) Windows Vista(TM) Enterprise Edition	
Microsoft(R) Windows Vista(TM) Ultimate Edition	
Microsoft(R) Windows Vista(TM) Business x64 Edition	Windows Vista x64
Microsoft(R) Windows Vista(TM) Enterprise x64 Edition	
Microsoft(R) Windows Vista(TM) Ultimate x64 Edition	
Microsoft(R) Windows(R) XP Professional Operating System	Windows XP

## 注

特に限定する必要がない場合は、Windows Server 2003、Windows Server 2003 x64、Windows Vista、Windows Vista x64、および Windows XP を合わせて Windows と表記しています。また、Windows Server 2003、および Windows Server 2003 x64 を合わせて Windows Server 2003 と表記し、Windows Vista、および Windows Vista x64 を合わせて Windows Vista と表記しています。

## このマニュアルで使用する記号

このマニュアルでは、次に示す記号を使用して、操作方法を説明しています。

記号	意味
[       ]	ウィンドウのボタン、またはメニューを示します。

記号	意味
[ A ] - [ B ]	A はメニューバーのメニューを示し、B は、A のメニューをクリックすると表示されるメニューを示します。

## オンラインヘルプについて

JP1/Script では、オンラインヘルプを提供しています。

オンラインヘルプは、このマニュアルの内容のうち、JP1/Script の機能、操作手順、および JP1/Script で使用できるコマンドを中心に説明しています。

## 常用漢字以外の漢字の使用について

このマニュアルでは、常用漢字を使用することを基本としていますが、次に示す用語については、常用漢字以外の漢字を使用しています。

個所（かしょ） 桁（けた） 進捗（しんちょく） 全て（すべて） 揃える（そろえる） 貼る（はる） 汎用（はんよう） 必須（ひつす）

## KB（キロバイト）などの単位表記について

1KB（キロバイト）、1MB（メガバイト）、1GB（ギガバイト）、1TB（テラバイト）はそれぞれ 1,024 バイト、1,024<sup>2</sup> バイト、1,024<sup>3</sup> バイト、1,024<sup>4</sup> バイトです。



# 目次

## 第 1 編 解説編

1	JP1/Script の概要	1
1.1	JP1/Script の特長	2
1.2	JP1/Script の全体構成	3
1.3	JP1/Script を構成するプログラムの種類	5
1.4	JP1/Script で扱うファイル	7
1.4.1	ファイルの種類	7
1.4.2	ファイルの容量	10
1.4.3	ラージファイル	11
1.5	JP1/Script のシステム構成	13
1.5.1	JP1/AJS2 と連携したシステム構成	13
1.5.2	JP1/Base と連携したシステム構成	14
1.5.3	JP1/IM-Console と連携したシステム構成	15
1.5.4	クラスタ環境で JP1/Script を使用する場合はシステム構成	16
1.5.5	ターミナルサービス環境で JP1/Script を使用する場合はシステム構成	17
1.6	JP1/Script の操作の流れ	19
1.7	スクリプトを作成する前に、実行する前に	21
1.8	Windows Vista 対応に関して	22
1.8.1	インストール先フォルダ	22
1.8.2	予約変数	22
1.8.3	JP1/Script サービスの開始と監視	22
1.8.4	コマンドの動作	24
1.8.5	マネージャの設定方法	27
1.8.6	レジストリの変更方法	27
1.8.7	メッセージボックスのレイアウト	27
1.8.8	管理者権限での実行	27
2	JP1/Script を利用するための準備	31
2.1	JP1/Script のインストール / アンインストール	32
2.1.1	プログラムのインストール先のフォルダ	32
2.1.2	Windows Server 2003 および Windows XP 版 JP1/Script ( P-2412-3F84 ) のインストール	33

2.1.3	Windows Server 2003 および Windows XP 版 JP1/Script ( P-2412-3F84 ) サービスの登録	37
2.1.4	64 ビットバージョン Windows Server 2003 ( IPF ) 版 JP1/Script ( P-2812-3F84 ) のインストール	38
2.1.5	64 ビットバージョン Windows Server 2003 ( IPF ) 版 JP1/Script ( P-2812-3F84 ) サービスの登録	39
2.1.6	Windows Vista 版 JP1/Script ( P-2A12-3F84 ) のインストール	39
2.1.7	Windows Vista 版 JP1/Script ( P-2A12-3F84 ) サービスの登録	40
2.1.8	JP1/Script のアンインストール	40
2.2	他のコンピュータとの通信時のアカウント	42
2.3	クラスタシステム環境での環境設定	44
2.4	ターミナルサービス環境での環境設定	46
2.5	ホスト名, または IP アドレスの変更手順	47
2.6	稼働監視プログラムでの JP1/Script の監視	48
2.7	JP1/Script の起動と終了	49
2.7.1	JP1/Script を起動する	49
2.7.2	JP1/Script を終了する	51

# 3

	JP1/Script の操作	55
3.1	マネージャの操作	56
3.1.1	Script マネージャウィンドウとメニュー	56
3.1.2	Script マネージャウィンドウでのマウスとキーの操作	61
3.1.3	使用するエディタを決める	62
3.1.4	スクリプトを作成・保存する	63
3.1.5	簡易入力機能を使ってスクリプトを作成する	65
3.1.6	スクリプトを編集する	70
3.1.7	スクリプトの文法をチェックする	71
3.1.8	スクリプトを複写する	72
3.1.9	スクリプトを追加する	73
3.1.10	スクリプトを削除する	75
3.1.11	スクリプトの名前を変更する	76
3.1.12	スクリプトの実行環境 ( すべての項目 ) を設定する	77
3.1.13	スクリプトの実行環境 ( 各項目 ) を設定する	80
3.1.14	スクリプトの自動起動を設定する	84
3.1.15	スクリプトを実行する	86
3.1.16	メニューフォームを作成する	89
3.1.17	エディタの起動と終了	91



3.2	エディタの操作	92
3.2.1	Script エディタウィンドウとメニュー	93
3.2.2	Script エディタウィンドウでのマウスとキーの操作	100
3.2.3	コメント行を設定・解除する	102
3.2.4	簡易入力機能を使う	105
3.2.5	メニューフォームを作成する	110
3.2.6	エディタの動作環境を設定する	111
3.2.7	スクリプトファイルの文法をチェックする	113
3.2.8	モニタリングを実行・中止する	115
3.2.9	スクリプトを実行する	123
3.2.10	モニタリング実行時のブレークポイントを設定・解除する	124
3.2.11	モニタリング実行時の動作環境を設定する	128
3.2.12	ウォッチへ変数を追加する	128
3.2.13	スクリプトの実行環境を設定する	129
3.2.14	文字列を検索する	133
3.2.15	文字列を置換する	135
3.3	簡易入力の操作	138
3.3.1	簡易入力の操作項目	138
3.3.2	コマンドやステートメントを入力する	139
3.3.3	JP1/Script の対象バージョンを指定する	140
3.3.4	1 行の最大文字数やインデントの桁数を指定する	141
3.4	トレースビューアの操作	143
3.4.1	Script トレースビューアウィンドウとメニュー	143
3.4.2	Script トレースビューアウィンドウでのマウスとキーの操作	147
3.4.3	トレースビューアの起動と終了	148
3.4.4	Script トレースビューアウィンドウでの操作	148
3.4.5	トレースファイルの表示・非表示を切り替える	149
3.4.6	トレースファイルの内容を表示する	150
3.4.7	トレースファイルを別名で保存する	151
3.4.8	トレースファイルを削除する	152
3.4.9	トレースファイルをクリアする	154
3.4.10	トレースファイルを印刷する	155
3.4.11	トレースファイルの内容を検索する	156
3.5	トレースファイルの表示操作	159
3.5.1	トレースファイルの表示ウィンドウとメニュー	159
3.5.2	Script トレース表示ウィンドウでのマウスとキーの操作	162
3.6	メニューエディタの操作	163

3.6.1	Script メニューエディタウィンドウとメニュー	164
3.6.2	Script メニューエディタウィンドウでのマウスとキーの操作	171
3.6.3	新しいメニューフォームを作成する	173
3.6.4	メニューフォームをコピーする	175
3.6.5	メニューフォームを貼り付ける	175
3.6.6	メニューフォームを削除する	176
3.6.7	コントロールの属性を一括して変更する	177
3.6.8	コントロールを一行に並べる	180
3.6.9	コントロールを均等な間隔で並べる	186
3.6.10	コントロールをメニューフォームの中央に並べる	189
3.6.11	コントロールの大きさをそろえる	191
3.6.12	コントロールをテキストの大きさに合わせる	195
3.6.13	メニューフォーム上にグリッドを表示する	196
3.6.14	タブの移動順序を設定する	198
3.6.15	選択したコントロールを前面に移動する	200
3.6.16	選択したコントロールを背面に移動する	202
3.6.17	コマンドのプロパティを設定する	203
3.6.18	メニューフォームをテスト表示する	204
3.6.19	メニューフォームの内容を印刷する	205
3.7	プロセスビューアの操作	207
3.7.1	プロセスビューアの操作項目	207
3.7.2	実行中のスクリプトプロセスを一覧表示する	209
3.7.3	スクリプトプロセスの表示更新間隔を指定する	210
3.8	実行環境ファイルコンバータの操作	212
3.8.1	実行環境ファイルコンバータの概要	212
3.8.2	実行環境ファイルの変換手順	212
3.8.3	実行環境ファイルコンバータの実行方法	213
3.8.4	実行環境構文ファイル (.SPU) の概要	216
3.8.5	実行環境構文ファイル (.SPU) の詳細	217

## 4

## ダイアログの説明 221

4.1	Script マネージャウィンドウのダイアログ	222
4.1.1	新規作成ダイアログ	222
4.1.2	ファイルの複写ダイアログ	224
4.1.3	ファイルの追加ダイアログ	225
4.1.4	ファイルの上書き確認ダイアログ	226

4.1.5	ファイルの削除の確認ダイアログ	227
4.1.6	名前の変更ダイアログ	227
4.1.7	実行ダイアログ	228
4.1.8	実行環境の設定（起動情報）ダイアログ	229
4.1.9	実行環境の設定（終了情報）ダイアログ	232
4.1.10	実行環境の設定（トレース情報）ダイアログ	234
4.1.11	実行環境（コマンドライン）の設定ダイアログ	236
4.1.12	コマンドラインの設定ダイアログ	238
4.1.13	実行環境（作業フォルダ）の設定ダイアログ	239
4.1.14	実行環境（起動日時）の設定ダイアログ	240
4.1.15	実行環境（打ち切り時間）の設定ダイアログ	242
4.1.16	実行環境（トレース出力先フォルダ）の設定ダイアログ	243
4.1.17	実行環境（ユーザトレース情報）の設定ダイアログ	245
4.1.18	フォルダの変更ダイアログ	246
4.1.19	ファイルのプロパティダイアログ	247
4.1.20	自動起動の設定ダイアログ	248
4.1.21	エディタの関連付けダイアログ	250
4.1.22	オプション（サーバ情報）ダイアログ	251
4.1.23	更新情報ダイアログ	253
4.1.24	オプション（互換性）ダイアログ	253
4.1.25	オプション（複数起動）ダイアログ	255
4.1.26	オプション（JP1/IM）ダイアログ	256
4.1.27	オプション（トレース）ダイアログ	258
4.1.28	オプション（クラスタ環境）ダイアログ	259
4.1.29	フォルダの選択ダイアログ	260
4.2	Script エディタウィンドウのダイアログ	262
4.2.1	値の更新ダイアログ	262
4.2.2	オプション（書式）ダイアログ	263
4.2.3	オプション（色）ダイアログ	264
4.2.4	オプション（互換性）ダイアログ	266
4.2.5	設定ダイアログ	267
4.2.6	検索ダイアログ	268
4.2.7	ファイル名の指定ダイアログ	269
4.2.8	ファイルバージョンの指定ダイアログ	270
4.2.9	コマンドラインパラメタの設定ダイアログ	271
4.2.10	変数の追加ダイアログ	272
4.3	Script トレースビューアウィンドウのダイアログ	274

4.3.1	削除の確認ダイアログ	274
4.3.2	クリアの確認ダイアログ	275
4.3.3	コンピュータの選択ダイアログ	275
4.3.4	検索ダイアログ	276
4.4	Script メニューエディタウィンドウのダイアログ	278
4.4.1	メニューフォームプロパティ (一般) ダイアログ	279
4.4.2	メニューフォームプロパティ (背景) ダイアログ	281
4.4.3	メニューフォームプロパティ (キー) ダイアログ	282
4.4.4	メニューフォームプロパティ (スタイル) ダイアログ	283
4.4.5	メニューフォームプロパティ (壁紙) ダイアログ	285
4.4.6	スタティックプロパティ (一般) ダイアログ	286
4.4.7	スタティックプロパティ (共通) ダイアログ	287
4.4.8	スタティックプロパティ (背景) ダイアログ	290
4.4.9	スタティックプロパティ (スタイル) ダイアログ	291
4.4.10	ボタンプロパティ (一般) ダイアログ	292
4.4.11	ボタンプロパティ (共通) ダイアログ	294
4.4.12	ボタンプロパティ (背景) ダイアログ	296
4.4.13	ボタンプロパティ (キー) ダイアログ	297
4.4.14	ボタンプロパティ (スタイル) ダイアログ	299
4.4.15	ファイル参照ボタンプロパティ (一般) ダイアログ	300
4.4.16	ファイル参照ボタンプロパティ (共通) ダイアログ	302
4.4.17	ファイル参照ボタンプロパティ (背景) ダイアログ	305
4.4.18	ファイル参照ボタンプロパティ (キー) ダイアログ	306
4.4.19	ファイル参照ボタンプロパティ (スタイル) ダイアログ	307
4.4.20	エディットコントロールプロパティ (一般) ダイアログ	309
4.4.21	エディットコントロールプロパティ (共通) ダイアログ	311
4.4.22	エディットコントロールプロパティ (キー) ダイアログ	313
4.4.23	エディットコントロールプロパティ (スタイル) ダイアログ	315
4.4.24	直線プロパティ (一般) ダイアログ	317
4.4.25	ファンクションキープロパティ (一般) ダイアログ	318
4.4.26	ファンクションキープロパティ (共通) ダイアログ	319
4.4.27	ファンクションキープロパティ (スタイル) ダイアログ	321
4.4.28	リストコントロールプロパティ (一般) ダイアログ	323
4.4.29	リストコントロールプロパティ (共通) ダイアログ	324
4.4.30	リストコントロールプロパティ (キー) ダイアログ	327
4.4.31	リストコントロールプロパティ (スタイル) ダイアログ	328
4.4.32	コンボボックスプロパティ (一般) ダイアログ	330

4.4.33	コンボボックスプロパティ（共通）ダイアログ	332
4.4.34	コンボボックスプロパティ（キー）ダイアログ	334
4.4.35	コンボボックスプロパティ（スタイル）ダイアログ	336
4.4.36	コマンドプロパティダイアログ	337
4.4.37	コマンドプロパティの設定ダイアログ	339
4.4.38	メニューフォーム名の指定ダイアログ	344
4.4.39	一括変更ダイアログ	344
4.4.40	グリッドの設定ダイアログ	346
4.4.41	タブオーダーの設定ダイアログ	347
4.4.42	メニューフォームの印刷情報ダイアログ	348

## 5

トラブルへの対処方法	351
------------	-----

5.1	対処の手順	352
5.2	トラブルシューティング	353
5.2.1	スクリプト実行時のトラブルと対処方法	353
5.2.2	コマンド実行時のトラブルと対処方法	354
5.2.3	トレースビューア実行時のトラブルと対処方法	357
5.2.4	プロセスビューア実行時のトラブルと対処方法	357
5.3	ログ情報	359
5.3.1	ログ情報の種類	359
5.3.2	ログファイルおよびディレクトリ一覧	360
5.4	トラブル発生時に採取が必要な資料	362
5.4.1	OS の情報	362
5.4.2	JP1/Script の情報	363
5.4.3	オペレーション内容	364
5.4.4	画面上の情報	364
5.5	資料の採取方法	365
5.5.1	資料採取ツール	365
5.5.2	OS システム情報の採取	365
5.5.3	ダンプ情報	366
5.5.4	JP1/Script の資料採取	366
5.5.5	Windows イベントログの採取	366
5.6	バックアップとリカバリー	367
5.6.1	JP1/Script で扱うファイルのバックアップとリカバリー	367
5.6.2	動作環境情報のバックアップとリカバリー	368

## 第 2 編 文法編

6	JP1/Script の規則	369
6.1	スクリプトに関する規則	370
6.1.1	変数の名前付け規則	370
6.1.2	利用できる変数の個数と容量	371
6.1.3	変数名として扱えないキーワード	371
6.1.4	予約変数	373
6.1.5	配列変数	376
6.1.6	定数	379
6.1.7	数字の記述規則	380
6.1.8	文字列の記述規則	380
6.1.9	演算規則	380
6.1.10	演算の優先順位	381
6.1.11	コーディング規則	381
6.1.12	JP1/Script の終了コード	384
6.1.13	JP1/Script のイベントログ	384
6.1.14	JP1/Script で発行するイベントログ	387
6.2	コマンドラインに関する規則	389
6.2.1	コマンドラインの形式	389
6.2.2	コマンドラインのパラメタの説明	390
6.2.3	コマンドラインの記述規則	391
6.2.4	注意事項	392
7	ステートメント	395
7.1	ステートメント一覧	396
7.1.1	=	396
7.1.2	Do...Loop	397
7.1.3	For...Next	398
7.1.4	For...End For	400
7.1.5	If...Then...Else	401
7.1.6	Select Case	403
7.1.7	While...End	404
7.1.8	Function	405
7.1.9	Sub	406

7.1.10	Call	407
7.1.11	Exit xx	408
7.1.12	GoTo	409
7.1.13	Continue	410
7.1.14	On Error	411

# 8

基本コマンド	413
--------	-----

8.1	基本コマンド一覧	414
8.1.1	基本コマンド一覧	414
8.2	変数操作コマンド	422
8.2.1	Dim (変数を宣言しメモリ領域を割り当てる)	422
8.2.2	Dim (配列) (配列変数を宣言しメモリ領域を割り当てる)	422
8.2.3	SetEnvironment または SetEnv (環境変数を設定する)	423
8.2.4	GetEnvironment または GetEnv (環境変数を取得する)	425
8.2.5	SetGV (グローバル変数を設定する)	426
8.2.6	GetGV (グローバル変数を取得する)	428
8.2.7	DeleteGV (グローバル変数を削除する)	429
8.2.8	GetArrayCount (配列変数の要素数を取得する)	430
8.3	文字列操作コマンド	432
8.3.1	InStr (文字列を検索し文字位置を返す)	432
8.3.2	InArray (配列変数の要素から文字列を検索し、インデックス番号を返す)	433
8.3.3	Len (文字列の文字数を返す)	434
8.3.4	LCase (半角アルファベットの大文字を小文字に変換する)	435
8.3.5	UCase (半角アルファベットの小文字を大文字に変換する)	436
8.3.6	Left (左端から指定した文字数分の文字列を返す)	436
8.3.7	Mid (指定した文字数分の文字列を返す)	437
8.3.8	Right (右端から指定した文字数分の文字列を返す)	438
8.3.9	Space (半角スペースからなる文字列を返す)	438
8.3.10	LTrim (先頭のスペースを削除した文字列を返す)	439
8.3.11	RTrim (末尾のスペースを削除した文字列を返す)	440
8.3.12	Trim (先頭および末尾のスペースを削除した文字列を返す)	441
8.3.13	文字列連結 (+) (文字列連結を行う)	441
8.3.14	文字列連結 (&) (文字列連結を行う)	442
8.3.15	文字列連結 (&=) (変数と式の文字列連結を行う)	443
8.3.16	AddStr (指定した連結文字で文字列連結を行う)	444
8.3.17	SeparateStrCount (文字列分割を行い、分割文字列の数を返す)	445

8.3.18	SeparateStr ( 文字列分割を行い, 分割文字列を返す )	446
8.3.19	Str ( 値を文字列で返す )	448
8.3.20	Format ( 値を書式化した文字列を返す )	448
8.3.21	IsLower ( 半角アルファベットの小文字かどうかを調べる )	450
8.3.22	IsUpper ( 半角アルファベットの大文字かどうかを調べる )	450
8.3.23	IsSingleChar ( 半角文字かどうかを調べる )	451
8.3.24	IsMultiChar ( 全角文字かどうかを調べる )	452
8.4	日付操作コマンド	453
8.4.1	Date ( 現在の日付を返す )	453
8.4.2	Time ( 現在の時刻を返す )	453
8.4.3	Year ( 指定された日付の年を返す )	453
8.4.4	Month ( 指定された日付の月を返す )	454
8.4.5	Day ( 指定された日付の日を返す )	454
8.4.6	Weekday ( 指定された日付の曜日を返す )	455
8.4.7	Hour ( 指定された時刻の時を返す )	456
8.4.8	Minute ( 指定された時刻の分を返す )	457
8.4.9	Second ( 指定された時刻の秒を返す )	458
8.4.10	CalcDate ( 指定された日付に加算減算を行う )	458
8.4.11	CompDate ( 二つの日付を比較する )	459
8.4.12	GetDateCount ( 二つの日付の経過日数を取得する )	461
8.4.13	CalcTime ( 指定された時刻に加算減算を行う )	462
8.4.14	CompTime ( 二つの時刻を比較する )	463
8.4.15	GetTimeCount ( 二つの時刻の経過時間を取得する )	464
8.4.16	IsLeapYear ( うるう年かどうかを調べる )	465
8.5	ファイル・フォルダ操作コマンド	467
8.5.1	IniRead ( 初期化ファイルから値を読み込む )	467
8.5.2	IniWrite ( 初期化ファイルに値を登録する )	468
8.5.3	TextFileReplace ( テキストファイルの文字列を置き換える )	469
8.5.4	TextOpen ( テキストファイルをオープンする )	470
8.5.5	TextClose ( テキストファイルをクローズする )	471
8.5.6	TextRead ( テキストファイルから 1 行分のデータを読み込む )	472
8.5.7	TextWrite ( テキストファイルにデータを書き込む )	474
8.5.8	TextSeek ( 読み書き開始位置を先頭または末尾に移動する )	475
8.5.9	GetTextPosition ( 読み書きの開始位置を返す )	476
8.5.10	MakeDir ( フォルダを作成する )	477
8.5.11	DeleteDir ( フォルダを削除する )	478
8.5.12	DeleteFile ( ファイルを削除する )	479



8.5.13	Rename (ファイル名を変更する)	480
8.5.14	TempDir (一時フォルダを取得する)	482
8.5.15	TempFile (一時ファイル名を作成する)	482
8.5.16	SetFileAttribute または SetFileAttr (フォルダ/ファイルの属性を設定する)	483
8.5.17	GetFileAttribute または GetFileAttr (フォルダ/ファイルの属性を取得する)	485
8.5.18	SetFileTime (ファイルの日付と時刻を設定する)	486
8.5.19	GetFileTime (ファイルの日付と時刻を取得する)	487
8.5.20	GetFileSize (ファイルの容量を取得する)	489
8.5.21	GetVersionInfo または GetVerInfo (ファイルのバージョン情報を取得する)	490
8.5.22	SplitFile (ファイルを分割する)	491
8.5.23	CatFiles (ファイルを統合する)	493
8.5.24	SetStandardFile または SetStdFile (標準入力・出力・エラーファイルを設定する)	494
8.5.25	ResetStandardFile または ResetStdFile (標準入力・出力・エラーファイルを解除する)	496
8.5.26	SplitPath (フルパスを解析する)	497
8.5.27	MakePath (フルパスを作成する)	498
8.5.28	SetPath (実行フォルダのパスを設定する)	499
8.5.29	GetPath (実行フォルダのパスを取得する)	499
8.5.30	SetVolumeLabel または SetVolLabel (ボリュームラベルを設定する)	500
8.5.31	GetVolumeLabel または GetVolLabel (ボリュームラベルを取得する)	501
8.5.32	GetDiskFreeSpace (ディスク空き容量を取得する)	502
8.5.33	Copy (ファイルをコピーする)	502
8.6	メッセージ出力コマンド	506
8.6.1	InputBox (メッセージとテキストボックスを表示する)	506
8.6.2	Message (ファイル/ウィンドウにテキストを出力する)	507
8.6.3	MessageBox (ダイアログボックスにメッセージを表示する)	511
8.6.4	MessageEventLog (イベントビューアにメッセージを出力する)	514
8.6.5	IMEventMessage (JP1/IM または JP1/Base にイベントを発行する)	515
8.7	メニュー呼び出しコマンド	519
8.7.1	Menu (ユーザ定義のメニューを表示する)	519
8.8	演算処理コマンド	522
8.8.1	+ 演算子 (加算) (二つの式の和を求める)	522
8.8.2	+= 演算子 (加算) (変数と式の和を変数に代入する)	523
8.8.3	- 演算子 (減算・マイナス符号) (二つの数値の差を求める)	524
8.8.4	-= 演算子 (減算) (変数と式の差を変数に代入する)	525
8.8.5	Mod 演算子 (剰余演算) (二つの数値の剰余を求める)	525
8.8.6	Mod= 演算子 (剰余演算) (変数と式の剰余を変数に代入する)	526

8.8.7	* 演算子 (乗算) (二つの数値の積を求める)	527
8.8.8	*= 演算子 (乗算) (変数と式の積を変数に代入する)	527
8.8.9	/ 演算子 (除算) (二つの数値の商を求める)	528
8.8.10	/= 演算子 (除算) (変数と式の商を変数に代入する)	529
8.8.11	¥ 演算子 (整数除算) (二つの数値の商を求める)	530
8.8.12	¥= 演算子 (整数除算) (変数と式の商を変数に代入する)	530
8.8.13	^ 演算子 (べき乗) (二つの数値のべき乗を求める)	531
8.8.14	^= 演算子 (べき乗) (変数と式のべき乗を変数に代入する)	532
8.8.15	比較演算子 (=, <>, <, <=, >, >=) (二つの式を比較する)	532
8.8.16	And (論理積) (二つの式の論理積を求める)	534
8.8.17	Or (論理和) (二つの式の論理和を求める)	535
8.8.18	Not (論理否定) (式の論理否定を求める)	536
8.9	チェック処理コマンド	538
8.9.1	IsEmpty (変数が Empty 値かどうかを調べる)	538
8.9.2	IsDefine または IsDef (変数が定義されているかどうかを調べる)	538
8.9.3	IsNumeric (数値として評価できるかを調べる)	539
8.9.4	IsEmptyDir (フォルダの中身が空かどうかを調べる)	539
8.9.5	ExistDir (フォルダが存在するかどうかを調べる)	540
8.9.6	ExistFile (ファイルが存在するかどうかを調べる)	541
8.9.7	IsWritableDir (フォルダが書き込み可能かどうかを調べる)	542
8.9.8	IsFileAttribute または IsFileAttr (フォルダ/ファイルの属性を調べる)	543
8.9.9	IsNew (バージョン情報/ファイル日付の新旧を比較する)	544
8.9.10	CheckDirName (フォルダ名の末尾が (¥) かどうかを調べる)	545
8.9.11	CheckDriveType (ドライブの種類を調べる)	546
8.10	外部プログラム呼び出しコマンド	547
8.10.1	Exec (自 PC 上の実行ファイルを呼び出す)	547
8.10.2	NetExec (自 PC / 他 PC 上の実行ファイルを呼び出す)	549
8.10.3	WaitForExec (実行ファイルの終了待ち/強制終了を行う)	553
8.10.4	GetExecStatus (実行ファイルの実行状態を取得する)	555
8.10.5	CallSpt (複数パラメタ指定で SPT ファイルを呼び出す)	556
8.11	自動起動処理コマンド	559
8.11.1	EntryStartUp (スクリプトファイルを自動起動に登録する)	559
8.11.2	CancelStartUp (スクリプトファイルの自動起動の登録を解除する)	561
8.12	JP1 呼び出しコマンド	562
8.12.1	JOBSubmit (キューサーバにスクリプトに登録する)	562
8.12.2	JOBWait (ジョブの終了を待つ)	564
8.12.3	JOBHold (ジョブを保留/解除する)	567

8.12.4	JOBCancel (ジョブをキャンセルする)	569
8.13	コメントコマンド	571
8.13.1	Rem または' (プログラム内にコメントを記述する)	571
8.14	その他のコマンド	572
8.14.1	Sleep (スクリプトの実行を中断する)	572
8.14.2	Alert (ユーザエラーを警告/解除する)	572
8.14.3	Beep (スピーカからビーブ音を鳴らす)	573
8.14.4	Exit (スクリプトを終了する)	574
8.14.5	GetErrorMessage (エラーメッセージを取得する)	575

## 9

特殊コマンド	577
9.1 特殊コマンド一覧	578
9.2 レジストリ操作コマンド	580
9.2.1 RegRead (レジストリから値を読み込む)	580
9.2.2 RegWrite (レジストリに値を登録する)	581
9.2.3 RegDelete (レジストリの値を削除する)	583
9.2.4 RegDeleteKey (レジストリのサブキーを削除する)	583
9.3 描画・表示処理コマンド	585
9.3.1 BitmapShow (ビットマップを表示する)	585
9.3.2 BitmapHide (ビットマップを消去する)	586
9.4 チェック処理コマンド	587
9.4.1 IsEmptyReg (レジストリサブキーの中身が空かどうかを調べる)	587
9.4.2 IsExistRegKey (レジストリサブキーが存在するかどうかを調べる)	588
9.4.3 IsExistService (サービスが存在するかどうかを調べる)	589
9.4.4 IsEmptyGroup (ショートカットが存在するかどうか調べる)	589
9.5 サービス操作コマンド	591
9.5.1 ServiceSetValue (サービス情報を設定する)	591
9.5.2 ServiceGetValue (サービスの情報を取得する)	594
9.5.3 ServiceCreate (サービスを登録する)	596
9.5.4 ServiceDelete (サービスを削除する)	597
9.5.5 ServiceStart (サービスを開始する)	598
9.5.6 ServiceStop (サービスを停止する)	599
9.5.7 ServicePause (サービスを一時停止する)	599
9.5.8 ServiceContinue (一時停止のサービスを再開する)	600
9.5.9 ServiceChange (サービスの設定情報を変更する)	601
9.5.10 ServiceQuery (サービスの設定情報を取得する)	602

9.5.11	ServiceRefer (サービスの現在の状態を取得する)	602
9.5.12	ServiceControl (サービスに制御コマンドを送る)	603
9.5.13	GetServiceName (サービス表示名からサービス名を取得する)	604
9.6	外部プログラム呼び出しコマンド	606
9.6.1	CallDll (DLL ファイルを呼び出す)	606
9.7	ショートカット操作コマンド	608
9.7.1	MakeGroup (プログラムグループを作成する)	608
9.7.2	DeleteGroup (プログラムグループを削除する)	609
9.7.3	MakeShortcut (ショートカットを作成する)	609
9.7.4	DeleteShortcut (ショートカットを削除する)	611
9.8	プロセス監視コマンド	613
9.8.1	GetProcessCount (プロセスの起動数を取得する)	613
9.8.2	GetProcessInfo (プロセスの情報を取得する)	614
9.8.3	TerminateProcess (プロセスを強制終了する)	615
9.9	その他のコマンド	617
9.9.1	ExitWindows (スクリプトを終了し Windows をログオフ/シャットダウンする)	617

## 10 スクリプト制御インタフェース 619

10.1	スクリプト制御インタフェースとは	620
10.2	スクリプト制御インタフェースの呼び出し	621
10.2.1	SPTHL.DLL のパスの取得方法	621
10.2.2	ロード方法	621
10.2.3	本機能がサポートされているかのチェック方法	622
10.3	スクリプト制御インタフェース一覧	623
10.3.1	SPTHOpen (スクリプト制御マネージャをオープンする)	623
10.3.2	SPTHTClose (スクリプト制御マネージャをクローズする)	624
10.3.3	SPTHGetErrorMessage (エラーメッセージを取得する)	624
10.3.4	SPTHTerminate (スクリプト実行を強制終了する)	625
10.4	スクリプト制御インタフェースの記述例	627

## 11 スクリプト OLE コントロール 629

11.1	スクリプト OLE コントロールとは	630
11.2	SPTO コントロール	631
11.2.1	RTN プロパティ (メソッドのエラー詳細コードを返す)	631

11.2.2 EXECRTN プロパティ（実行ファイルの終了コードを返す）	631
11.2.3 EXECID プロパティ（実行ファイルの識別子を返す）	632
11.2.4 EnableErrorMessage プロパティ（エラーの有無を返す）	632
11.2.5 SetGV メソッド（グローバル変数を設定する）	633
11.2.6 GetGV メソッド（グローバル変数を取得する）	634
11.2.7 DeleteGV メソッド（グローバル変数を削除する）	634
11.2.8 Exec メソッド（実行ファイルを呼び出す）	635

<b>付録</b>	<b>637</b>
付録 A Script トレースのファイル出力形式	638
付録 B サンプルファイル	648
付録 C エラー詳細コード	651
付録 D 用語解説	656

<b>索引</b>	<b>661</b>
-----------	------------

## 図目次

図 1-1	JP1/Script の全体構成	4
図 1-2	ジョブの実行運転・実行管理を行う場合の JP1/AJS2 との連携	13
図 1-3	クライアントからサーバ上の JP1/AJS2 のキューヘジョブを登録する場合の JP1/AJS2 との連携	14
図 1-4	JP1/Base と連携したシステム構成	15
図 1-5	JP1/IM-Console と連携したシステム構成	16
図 1-6	クラスタ環境で JP1/Script を使用する場合のシステム構成	17
図 1-7	ターミナルサービス環境で JP1/Script を使用する場合のシステム構成	18
図 1-8	JP1/Script の操作の流れ	20
図 3-1	Script マネージャウィンドウ	57
図 3-2	Script エディタウィンドウ	93
図 3-3	[ Script 簡易入力 ] ダイアログ	138
図 3-4	Script トレースビューアウィンドウ	143
図 3-5	Script トレース表示ウィンドウ	159
図 3-6	Script メニューエディタウィンドウ	164
図 3-7	コントロールボックス	167
図 3-8	[ Script プロセスビューア ] メッセージボックス	207
図 3-9	実行環境ファイルの移行の流れ	213
図 3-10	/SPT:ALL を指定した場合の実行環境ファイル作成の流れ	215
図 3-11	実行環境構文ファイルのフォーマット	217
図 A-1	解析トレースファイルの出力形式（複数起動が許可されていないスクリプトの場合）	638
図 A-2	解析トレースファイルの出力形式（複数起動が許可されているスクリプトの場合）	639
図 A-3	実行トレースファイルの出力形式（複数起動が許可されていないスクリプトの場合）	641
図 A-4	実行トレースファイルの出力形式（複数起動が許可されているスクリプトの場合）	642
図 A-5	ユーザトレースファイルの出力形式	643
図 A-6	サーバトレースファイルの出力形式	644
図 A-7	クライアント側エラーログの出力形式	646
図 A-8	サーバ側エラーログの出力形式	646

# 表目次

表 1-1	JP1/Script のプログラム	5
表 1-2	JP1/Script で扱うファイルの種類	7
表 1-3	アクセス拒否のエラーとなるコマンド	25
表 2-1	コマンド呼び出し元のログオンアカウントとパスワード	42
表 2-2	NetExec コマンドから呼び出す実行ファイルのアカウントとパスワード	42
表 3-1	Script マネージャウィンドウのメニュー一覧	59
表 3-2	ポップアップメニュー一覧	60
表 3-3	Script マネージャウィンドウでのマウス操作（対象がアイコンの場合）	61
表 3-4	Script マネージャウィンドウでのマウス操作（対象が背景の場合）	61
表 3-5	Script マネージャウィンドウでのキー操作	62
表 3-6	Script エディタのメニュー一覧	96
表 3-7	ポップアップメニュー一覧（編集モード）	98
表 3-8	ポップアップメニュー一覧（モニタリングモード）	99
表 3-9	Script エディタウィンドウでのマウス操作	100
表 3-10	Script エディタウィンドウでのキー操作	101
表 3-11	複数起動を許可されているスクリプトの場合	145
表 3-12	複数起動を許可されていないスクリプトの場合	145
表 3-13	トレースビューアのメニュー一覧	146
表 3-14	Script トレースビューアウィンドウでのマウス操作	147
表 3-15	Script トレースビューアウィンドウでのキー操作	147
表 3-16	Script トレース表示ウィンドウのメニュー一覧	161
表 3-17	Script トレース表示ウィンドウでのマウス操作	162
表 3-18	Script トレース表示ウィンドウでのキー操作	162
表 3-19	メニューエディタのメニュー一覧	168
表 3-20	レイアウトコマンドのカスケードメニュー一覧	170
表 3-21	ポップアップメニュー一覧	171
表 3-22	Script メニューエディタウィンドウでのマウス操作	171
表 3-23	Script メニューエディタウィンドウでのキー操作	172
表 3-24	実行環境ファイルコンバータの戻り値とメッセージ	215
表 3-25	実行環境構文ファイルの概要	217
表 3-26	指定値に対する曜日	218
表 5-1	スクリプト実行時のトラブルと対処方法	353
表 5-2	コマンド実行時のトラブルと対処方法	355

表 5-3	トレースビューア実行時のトラブルと対処方法	357
表 5-4	プロセスビューア実行時のトラブルと対処方法	358
表 5-5	ログの出力元，ログファイル名，およびディスク使用量	360
表 5-6	トラブル発生時に採取が必要な OS の情報	362
表 5-7	トラブル発生時に採取が必要な JP1/Script の情報	363
表 5-8	バックアップ対象ファイル	367
表 6-1	変数名として扱えないキーワード	371
表 6-2	予約変数一覧	374
表 6-3	定数一覧	379
表 6-4	演算の優先順位	381
表 6-5	レジストリに設定されている終了コード	384
表 6-6	イベント ID とイベントログ	384
表 6-7	JP1/Script が発行する JP1/IM イベント	387
表 6-8	出力する情報	388
表 7-1	ステートメント一覧	396
表 8-1	基本コマンド一覧	414
表 8-2	Exec，NetExec，および CallSpt コマンドの違い	421
表 9-1	特殊コマンド一覧	578



# 1

## JP1/Script の概要

JP1/Script は、Windows 上でジョブを制御するスクリプトを作成して実行するためのプログラムです。  
この章では、JP1/Script の特長、機能の概要、およびシステム構成について説明します。

- 
- 1.1 JP1/Script の特長
  - 1.2 JP1/Script の全体構成
  - 1.3 JP1/Script を構成するプログラムの種類
  - 1.4 JP1/Script で扱うファイル
  - 1.5 JP1/Script のシステム構成
  - 1.6 JP1/Script の操作の流れ
  - 1.7 スクリプトを作成する前に、実行する前に
  - 1.8 Windows Vista 対応に関して
-

## 1.1 JP1/Script の特長

---

JP1/Script は、Windows 上でジョブを制御するスクリプトを作成して実行するためのプログラムです。JP1/Script を使用することで、簡単にスクリプトを作成できます。

また、今までほかのシステムでジョブ制御を行っていた場合でも、ほかのシステムでのジョブ制御と同じようなイメージで、Windows 上でもジョブ制御を実現できます。

### 1. プログラムの作成 / 実行が容易

JP1/Script は、Basic 言語に似たコマンドで記述されたソースコードを逐次解釈しながら、実行するプログラムです。したがって、プログラムを作成 / 変更してから実行するまでの手間がなく、プログラムの作成 / 実行が容易です。

### 2. プログラム編集 / デバッグ時の負荷を軽減

JP1/Script の専用エディタでは、簡易入力機能を使用して入力したコマンドやステートメントをボタン一つでエディタに貼り付けることができ、コマンドやステートメントの知識がなくても手軽に間違いなく作成できます。また、モニタリング機能を使えば、スクリプトの動作を見ながら実行できるので、デバッグ時の負荷が軽減できます。

### 3. 目的に合った多様なコマンドを用意

JP1/Script では、ファイルやフォルダの操作やメッセージの出力などから、レジストリの操作、Windows のシャットダウンなど、業務に必要なさまざまなコマンドを用意しています。

### 4. メニュー機能による対話形式のプログラム作成が可能

JP1/Script では、メニュー機能を提供しています。メニュー機能を利用すれば、画面上で操作した内容を基にスクリプトを実行させることができ、対話業務にも適用できます。

### 5. さまざまな機能を提供

JP1/Script では、このほかにもさまざまな機能を提供しています。

例えば、作成したスクリプトを自動起動させたり、スクリプトから別のスクリプトやアプリケーションを自コンピュータ上や他コンピュータ上で実行させたりすることができます。

また、プログラムの実行結果を取得できます。実行結果を取得することで、次の処理を判断できるため、プログラムの切り替えなど、柔軟な連携処理を実現できます。

さらに、他の JP1 製品と連携することで統合的なジョブ管理が実現できます。

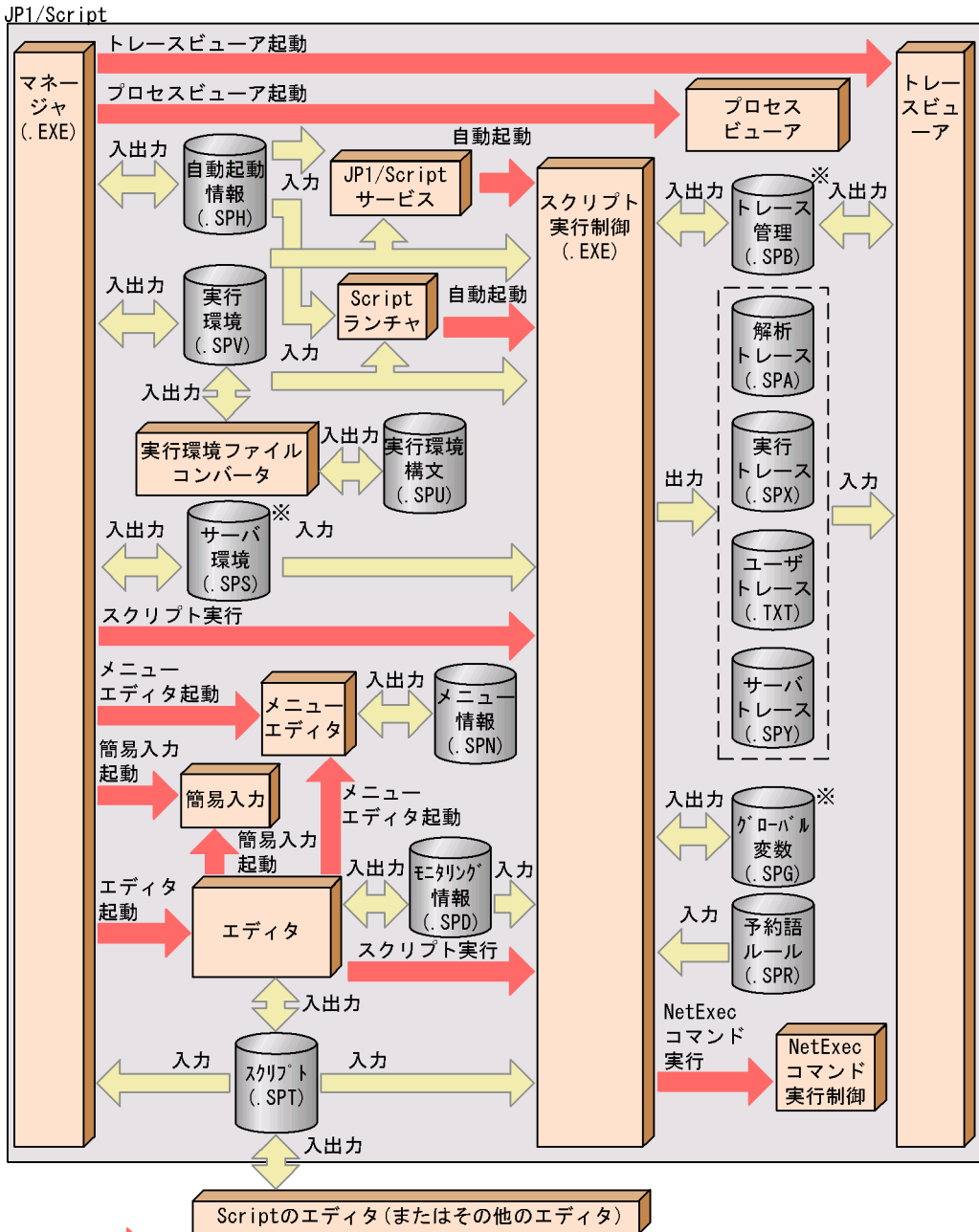
## 1.2 JP1/Script の全体構成

---

図 1-1 に JP1/Script の全体の構成を示します。

# 1. JP1/Script の概要

図 1-1 JP1/Script の全体構成



(凡例) → : プログラムの実行や起動

注※ → : ファイルの入出力

Windows Vistaでは、JP1/Scriptサービスに対するファイルの入力があります。

## 1.3 JP1/Script を構成するプログラムの種類

JP1/Script は、表 1-1 に示すプログラムによって構成されています。各プログラムの関連性については、「1.2 JP1/Script の全体構成」を参照してください。

表 1-1 JP1/Script のプログラム

プログラムの種類	アイコン	プログラム名	概要	プロセス監視の必要性
マネージャ		SPTM.exe	JP1/Script で作成したスクリプトや実行環境を管理するマネージャです。	×
スクリプト実行制御	-	Sptxe.exe	スクリプトファイル（.SPT）を解析して実行します。	×
NetExec コマンド実行制御	-	SPTXNetx.exe	NextExec コマンド実行時の接続先コンピュータとの通信処理を行います。	×
Script ランチャ		SPTHLNCH.exe	自動起動情報に登録されたスクリプトファイルのログオン空間での起動、および終了の制御や、NetExec コマンドのログオン空間での要求を制御します。	
JP1/Script サービス	-	SPTHSV.exe , SPTTMS.exe	自動起動情報に登録されたスクリプトファイルのサービス空間での起動、および終了の制御や、NetExec コマンドのサービス空間での要求を制御します。	
トレースビューア		SPTTM.exe	トレースファイルの状態、および内容を表示します。	×
簡易入力		Sptke.exe	コマンドやステートメントの入力作業を簡略化できる機能です。この機能を使うことで、コマンドの知識がなくても入力できます。記述したコマンドやステートメントの内容は、クリップボードやエディタに貼り付けられます。	×
エディタ		SPTIEdit.exe	JP1/Script 専用のエディタで、スクリプトファイルの作成、編集、保存に必要な機能を備えています。	×
メニューエディタ		SPTNEdit.exe	作成したスクリプトファイルのメニューフォーム（画面）を作成したり編集したりする機能を備えています。	×
プロセスビューア		SPTHView.exe	実行中のスクリプトプロセスを一覧表示したり、プロセスを監視・制御したりする機能を備えています。	×

## 1. JP1/Script の概要

プログラムの種類	アイコン	プログラム名	概要	プロセス監視の必要性
実行環境 ファイルコンバータ	-	SPTUspv.exe	実行環境ファイル（拡張子 SPV）と 実行環境構文ファイル（拡張子 SPU） を 相互変換できます。	×

（凡例）

- : 該当するアイコンはありません。
- : 監視する必要があります。
- × : 監視する必要はありません。

JP1/PFM/SSO などの稼働監視プログラムで JP1/Script のプロセスを監視する場合に必要なプロセスについて記載しています。プロセス監視については、「2.6 稼働監視プログラムでの JP1/Script の監視」を参照してください。

## 1.4 JP1/Script で扱うファイル

### 1.4.1 ファイルの種類

JP1/Script で扱うファイルを表 1-2 に示します。

なお、ファイルの容量の計算方法については、「1.4.2 ファイルの容量」を、トレースビューアが扱うファイルの出力形式については、「付録 A Script トレースのファイル出力形式」を参照してください。

表 1-2 JP1/Script で扱うファイルの種類

ファイルの種別	ファイル名	ファイルの拡張子	ファイルの形式	ファイルの内容
マネージャが扱うファイル	スクリプトファイル	.SPT	テキスト形式	スクリプト文を保存するファイルです。 ファイル名はユーザが任意に指定できます。
	実行環境ファイル	.SPV	バイナリ形式	一つのスクリプトファイルを実行するときの環境が設定されたファイルです。 ファイル名には、拡張子以外の部分はスクリプトファイルと同じものが付けられます。
	自動起動情報ファイル	.SPH	バイナリ形式	ランチャまたはサービスがスクリプトを自動的に実行できるようにするための、必要な情報が保存されたファイルです。 ファイル名は、SPTLNCH で固定です。 Windows XP および Windows Vista では、ユーザ簡易切替機能により複数ユーザの同時ログオンができるため、2 種類のファイルを用意しています。 <ul style="list-style-type: none"> <li>起動タイプがログオンのスクリプトが登録されるユーザ単位の自動起動情報ファイル</li> <li>起動タイプがサービスのスクリプトが登録される全ユーザ共通の自動起動情報ファイル</li> </ul> ファイル名は両ファイルとも SPTLNCH で固定です。

## 1. JP1/Script の概要

ファイルの種別	ファイル名	ファイルの拡張子	ファイルの形式	ファイルの内容
	サーバ環境ファイル	.SPS	バイナリ形式	サーバ環境が設定されているファイルです。 ファイル名は、SPTSV で固定です。
トレースビューが扱うファイル	解析トレースファイル	.SPA	テキスト形式	スクリプト文の構文解析の結果が保存されたファイルです。 ファイル名には、スクリプトファイルと同じものが付けられます。
	実行トレースファイル	.SPX	テキスト形式	スクリプト文のコマンドの実行結果が保存されたファイルです。 ファイル名には、スクリプトファイルと同じものが付けられます。
	ユーザトレースファイル (トレースファイル)	.TXT	テキスト形式	スクリプトのコマンドによって出力されたトレースが保存されたファイルです。 ファイル名はユーザが任意に指定できます。
	サーバトレースファイル	.SPY	テキスト形式	クライアントにあるスクリプト文のコマンドから呼び出されたサーバのコマンド実行結果を出力するファイルです。 ファイル名は、SPTSVTRC で固定です。
エディタが扱うファイル	モニタリング情報ファイル	.SPD	バイナリ形式	モニタリング機能で利用する情報ファイルです。 ファイル名には、拡張子以外の部分はスクリプトファイルと同じものが付けられます。
メニューエディタが扱うファイル	メニュー情報ファイル	.SPN	バイナリ形式	メニュー、および各コントロールのプロパティ定義が記述されたファイルです。メニューエディタの起動方法によって異なります。 <ul style="list-style-type: none"> <li>メニューエディタをマネージャやエディタのメニューから起動した場合 ファイル名は、拡張子以外の部分はスクリプトファイルと同じものが付けられます。</li> <li>メニューエディタをアイコンから起動した場合 ファイル名はユーザが任意に指定できます。</li> </ul>



ファイルの種別	ファイル名	ファイルの拡張子	ファイルの形式	ファイルの内容
コンバータが扱うファイル	実行環境構文ファイル	.SPU	テキスト形式	実行環境ファイルコンバータで利用するファイルです。実行環境ファイルに設定されている内容が文字形式で設定されています。ファイル名はユーザが任意に指定できます。
その他のファイル	トレース管理ファイル <sup>1</sup>	.SPB	バイナリ形式	トレースファイルを管理するファイルです。ファイル名は、SPTLOGDB で固定です。
	ワークファイル	.TMP	バイナリ形式	JP1/Script 内部で利用するワークファイルです。
	グローバル変数ファイル	.SPG	バイナリ形式	スクリプトの実行時に設定されたグローバル変数が保存されたファイルです。ファイル名は、SPTGV で固定です。
	予約語ルールファイル	.SPR	バイナリ形式	スクリプトファイルを解析・実行するために必要となる予約語に関する規則を格納したファイルです。ファイル名は、VERvvrr で固定です。vv は JP1/Script のバージョンを表し、rr は JP1/Script のリビジョンを表しています。

注 1 トレース管理ファイルは、以下のトレースファイルのファイル名、およびトレースファイルの書き出し位置などを管理しているファイルです。

- ・ 解析トレースファイル
- ・ 実行トレースファイル
- ・ サーバトレースファイル
- ・ ユーザトレースファイル

次の条件の場合、トレース管理ファイルで管理するトレースファイルが増えるためトレース管理ファイルの容量が増加します。

- ・ ファイル名の異なるスクリプトを実行する
- ・ Message コマンドで Target 引数に「Target\_File」を指定して新規のファイルに出力する

ファイル名の異なるスクリプトを実行した場合、そのスクリプトファイルの解析トレースファイル、および実行トレースファイルを管理するため、容量が増加します。同じスクリプトファイルを複数回実行する場合は、トレース管理ファイルに出力されている情報を使用しますので容量が増えることはありません。

Message コマンドで Target 引数に「Target\_File」を指定して新規のファイルに出力する場合、容量が増加します。

Message コマンドの OutputName 引数に指定したファイルが、すでに Message コマンドで Target 引数に「Target\_File」を指定して出力済のファイルの場合、トレース管理ファイルに出力されている情報を使用しますので容量が増えることはありません。

## 1. JP1/Script の概要

Message コマンドで Target 引数に「Target\_File」を指定して、ユニークなファイル名称を持ったユーザトレースファイルを大量に作成する場合、大量に作成されたファイル名、およびトレースファイルの書き出し位置などを管理するための情報が増えるため、トレース管理ファイルの容量が増大してしまいます。トレース管理ファイルの容量が増大すると、以下の現象が発生します。

- スクリプトファイルの実行性能が劣化する
- スクリプトファイルの実行が終了コード「20」で異常終了する
- コマンドの実行がメモリ不足でエラーになる

ユニークなファイル名称を持ったユーザトレースファイルを大量に作成する場合は、TextOpen/TextWrite/TextClose コマンドを使用して、トレースファイルを作成してください。TextOpen/TextWrite/TextClose コマンドで作成するファイルは、トレース管理ファイルの管理対象外のファイルになるため、トレース管理ファイルの容量が増大することはありません。

### 1.4.2 ファイルの容量

JP1/Script で扱う各ファイルの容量の計算方法、またはおよその容量を次に示します。

#### マネージャが扱うファイル

##### スクリプトファイル (.SPT)

スクリプトファイルは、ユーザがどれだけの量のスクリプトを記述するのかによって、容量が異なります。したがって、スクリプトファイルのファイル容量は、ユーザ任意となります。

##### 実行環境ファイル (.SPV)

ファイル容量は、8 ~ 12KB です。

##### 自動起動情報ファイル (.SPH)

自動起動情報ファイル、自動起動に登録した回数によって、容量が異なります。ただし、初期割り当てサイズは 36KB で、七つのスクリプトファイルを自動起動に登録できます。

##### サーバ環境ファイル (.SPS)

ファイル容量は、8 ~ 12KB です。

#### トレースビューアが扱うファイル

##### 解析トレースファイル (.SPA)

計算方法は次のとおりです。(単位: バイト)

最大行数 × (最大列数 + 4)

最大行数、および最大列数は、マネージャの [ ファイル ] - [ 実行の環境設定 ] の「トレース情報」ページで設定した値です。

##### 実行トレースファイル (.SPX)

計算方法は次のとおりです。(単位: バイト)

最大行数 × (最大列数 + 4)

最大行数、および最大列数は、マネージャの [ ファイル ] - [ 実行の環境設定 ]

の「トレース情報」ページで設定した値です。

ユーザトレースファイル (.TXT)

計算方法は次のとおりです。(単位: バイト)

最大行数 × (最大列数 + 4)

最大行数, および最大列数は, マネージャの [ ファイル ] - [ 実行の環境設定 ] の「トレース情報」ページで設定した値です。

サーバトレースファイル (.SPY)

ファイル容量は, 0.2 ~ 204KB です。

エディタが扱うファイル

モニタリング情報ファイル (.SPD)

モニタリング情報ファイルは, ウォッチ変数の個数とブレイクポイントの個数によって, 容量が異なります。ただし, 初期割り当てサイズは 4KB です。

メニューエディタが扱うファイル

メニュー情報ファイル (.SPN)

メニュー情報ファイルは, 定義したフォームの数やフォーム上に貼り付けたコントロールの数などによって, 容量が異なります。したがって, メニュー情報ファイルのファイル容量は任意です。

コンバータが扱うファイル

実行環境構文ファイル (.SPU)

実行環境ファイル (.SPV) の容量に依存します。

その他のファイル

トレース管理ファイル (.SPB)

初期割り当てサイズは, 12KB です。また, 拡張サイズは 12KB です。初期割り当てサイズは, マネージャの [ ツール ] - [ オプション ] の「トレース」ページで変更できます。なお, 初期割り当てサイズで, 8 個のトレースファイルを格納できます。

グローバル変数ファイル (.SPG)

初期割り当てサイズは, 20KB です。また, 拡張サイズは 20KB です。

### 1.4.3 ラージファイル

#### (1) ファイルシステム

FAT32, NTFS システムのラージファイルに対応します (FAT, FAT16 は対応しません)。

## (2) ファイルの最大容量

NTFS は 8EB ( 9,223,372,036,854,775,807 バイト ), FAT32 は 3.9GB ( 4,294,967,295 バイト ) となります。それ以上の場合の動作は保証しません。

## (3) エラーメッセージ

ラージファイル対応のエラーメッセージを、次に示します。

No.	コマンド	エラーメッセージ	予約変数 ( 値 )
1	GetFileSize	取得した値が変数の上限値を超えているため、値を変数に格納できません。	_ERR_FILE_SIZE_ ( 536904784 )
2	IniRead	指定したファイルの容量が制限値を超えています。	_ERR_NOT_LARGE_FILE_ ( 536904785 )
3	IniWrite	指定したファイルの容量が制限値を超えています。	_ERR_NOT_LARGE_FILE_ ( 536904785 )
4	TextFileReplace	指定したファイルの容量が制限値を超えています。	_ERR_NOT_LARGE_FILE_ ( 536904785 )
5	GetTextPosition	読み書き開始位置が、2,147,483,647 を超えています。	_ERR_FILE_POSITION_ ( 536904786 )

## 1.5 JP1/Script のシステム構成

JP1/Script を利用したシステム構成について説明します。

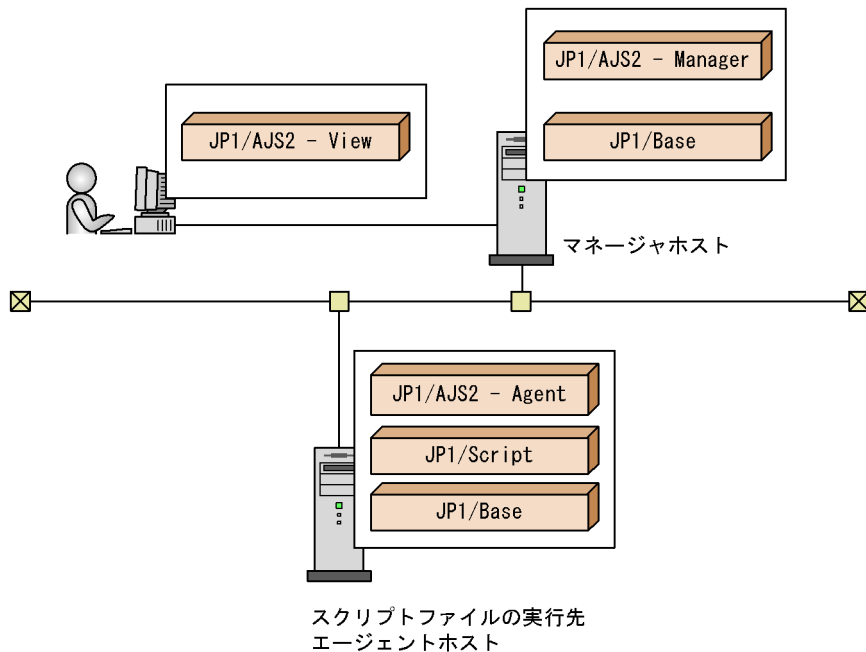
### 1.5.1 JP1/AJS2 と連携したシステム構成

#### (1) ジョブの自動運転・実行管理

JP1/AJS2 と連携してジョブの自動運転・実行管理を行う場合のシステム構成について説明します。JP1/AJS2 では、JP1/Script で作成したスクリプトファイルをジョブとして定義して、実行できます。

ジョブの自動運転・実行管理を行う場合の JP1/AJS2 との連携を次の図に示します。

図 1-2 ジョブの実行運転・実行管理を行う場合の JP1/AJS2 との連携



マネージャホストのジョブ定義で、エージェントホストで作成したスクリプトファイルの実行をエージェントホストに依頼するジョブを定義すると、実行時刻になった時点で、エージェントホストのスクリプトファイルが実行されます。

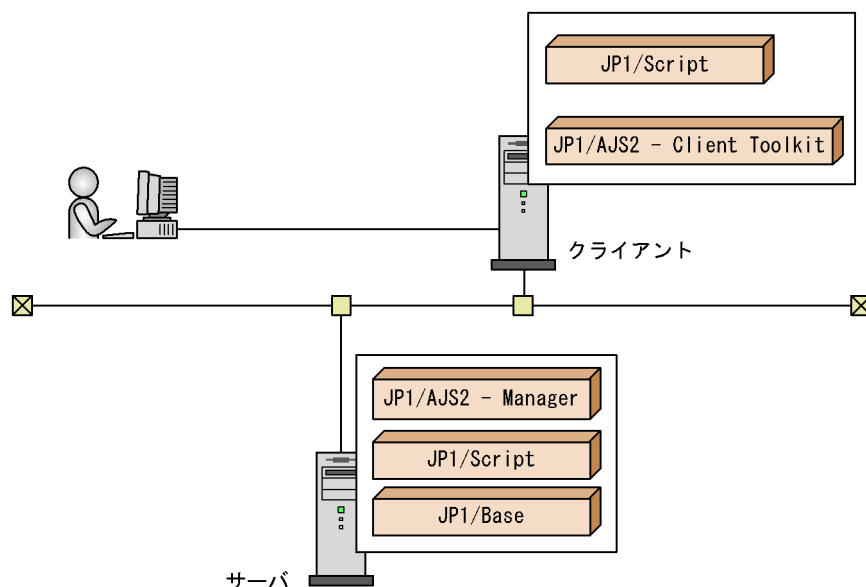
また、JP1/AJS2 では、GUI を表示し入力待ちになるプログラムをジョブとして実行できませんが、JP1/Script のスクリプトファイルを利用すると、GUI を持つプログラムを起動させることができます。詳細は、「8.10.2 NetExec (自 PC / 他 PC 上の実行ファイルを呼び出す)」を参照してください。

### (2) クライアントからサーバ上の JP1/AJS2 のキューへのジョブの登録

JP1/AJS2 と連携してクライアントからサーバ上の JP1/AJS2 のキューへジョブを登録する場合のシステム構成について説明します。JP1/Script では、JP1/AJS2 のキューに対してスクリプトファイルを登録できます。JP1/AJS2 のキューに対する操作コマンドについては、「8.12 JP1 呼び出しコマンド」を参照してください。

クライアントからサーバ上の JP1/AJS2 のキューへジョブを登録する場合の JP1/AJS2 との連携を次の図に示します。

図 1-3 クライアントからサーバ上の JP1/AJS2 のキューへジョブを登録する場合の JP1/AJS2 との連携

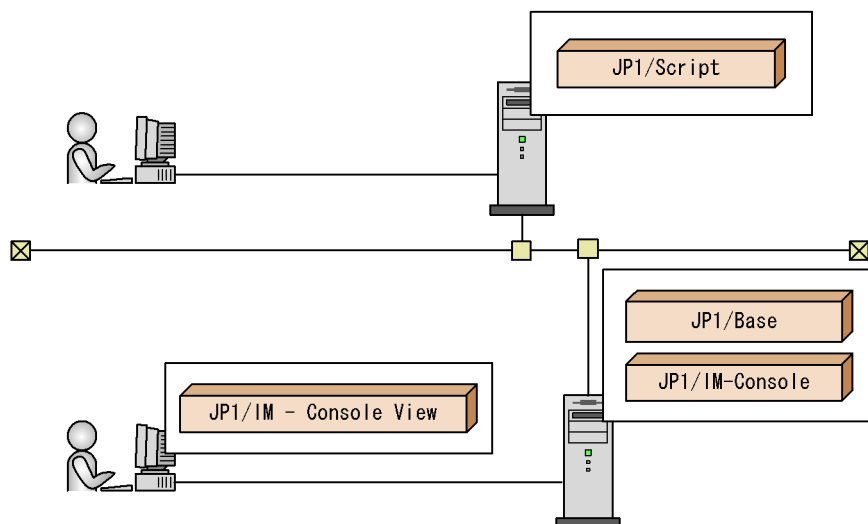


### 1.5.2 JP1/Base と連携したシステム構成

JP1/Base と連携して、JP1/Script から JP1/Base へ JP1 イベントを発行する場合のシステム構成について説明します。JP1/Script では、JP1/Base に対して JP1 イベントを発行できます。JP1 イベントの発行については、「4.1.26 オプション (JP1/IM) ダイアログ」および「8.6.5 IMEventMessage (JP1/IM または JP1/Base にイベントを発行する)」を参照してください。

JP1/Base と連携したシステム構成を次の図に示します。

図 1-4 JP1/Base と連携したシステム構成

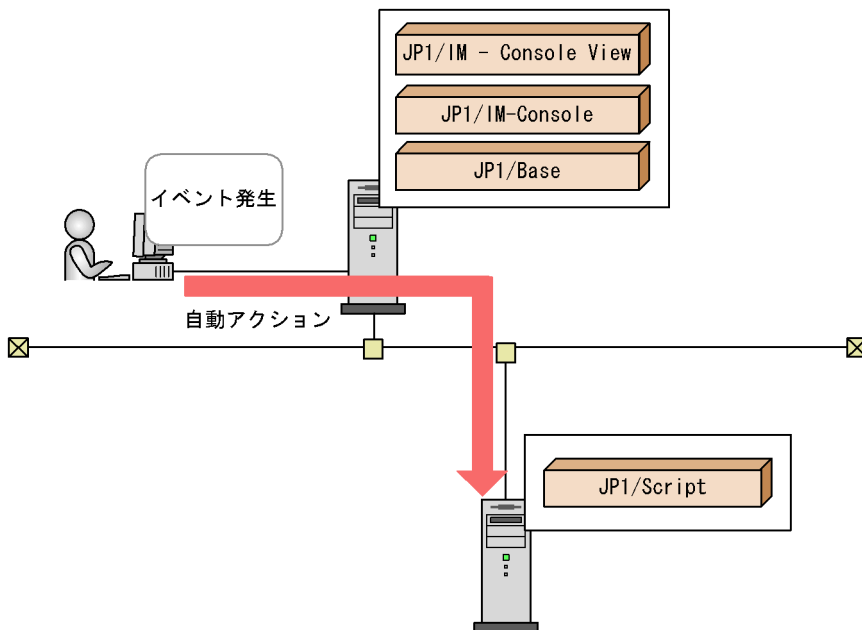


### 1.5.3 JP1/IM-Console と連携したシステム構成

JP1/IM-Console と連携した場合のシステム構成について説明します。JP1/IM-Console では、特定の JP1 イベントを受信したときに、自動的にコマンドを実行させることができる自動アクションで、JP1/Script で作成したスクリプトファイルを実行できます。

JP1/IM-Console と連携したシステム構成を次の図に示します。

図 1-5 JP1/IM-Console と連携したシステム構成



#### 1.5.4 クラスタ環境で JP1/Script を使用する場合のシステム構成

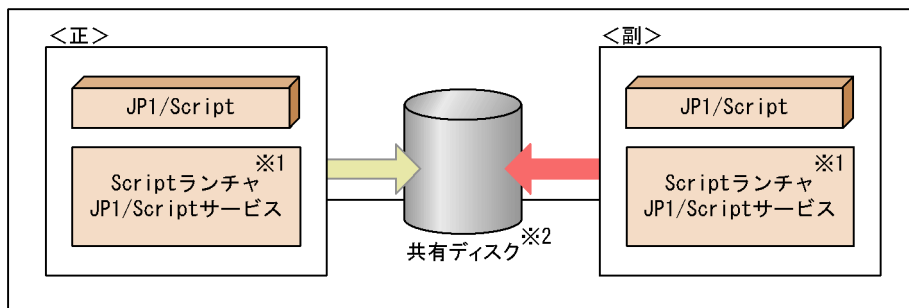
クラスタ環境で JP1/Script を使用する場合のシステム構成について説明します。JP1/Script では、クラスタ環境には対応していますが、フェールオーバーなどの実行状態の引継ぎなどはサポートしていません。クラスタ環境については「2.3 クラスタシステム環境での環境設定」を参照してください。

クラスタ環境で JP1/Script を使用する場合のシステム構成を次の図に示します。



図 1-6 クラスタ環境で JP1/Script を使用する場合のシステム構成

クラスタマシン



注※1 ScriptランチャおよびJP1/Scriptサービスはアクティブホスト上で稼働します。

注※2 JP1/Scriptの管理ファイルを格納します。

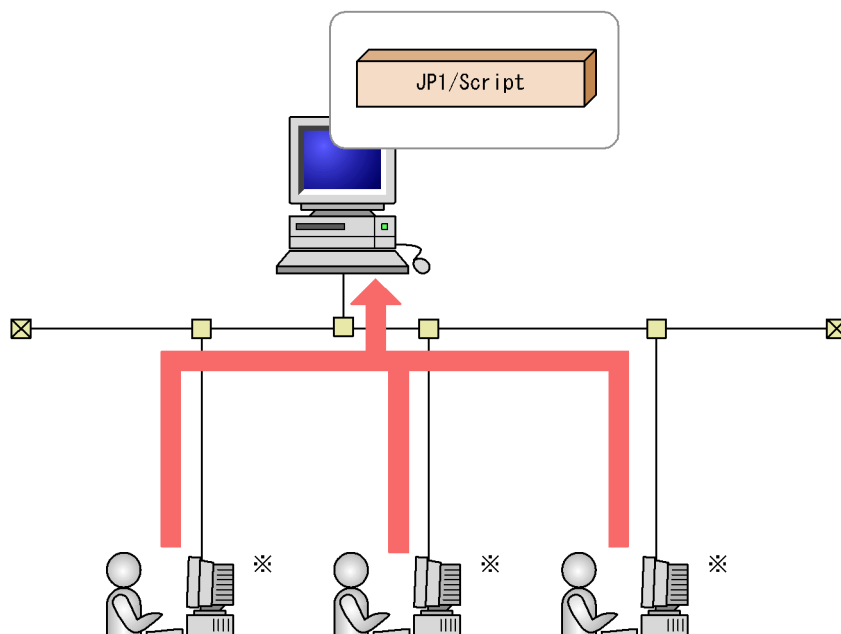
### 1.5.5 ターミナルサービス環境で JP1/Script を使用する場合のシステム構成

ターミナルサービス環境で JP1/Script を使用する場合のシステム構成について説明します。ターミナルサービス環境で JP1/Script を使用する場合は、JP1/Script アクセスライセンスを購入する必要があります。ターミナルサービス環境については「2.4 ターミナルサービス環境での環境設定」を参照してください。

ターミナルサービス環境で JP1/Script を使用する場合のシステム構成を次の図に示します。

## 1. JP1/Script の概要

図 1-7 ターミナルサービス環境で JP1/Script を使用する場合のシステム構成



注※ P-F2412-3F841, または P-F2812-3F841 JP1/Scriptアクセスライセンスを  
購入する必要があります。

## 1.6 JP1/Script の操作の流れ

---

JP1/Script の操作は、マネージャウィンドウでの操作と、トレースビューアウィンドウ（およびトレースファイルの表示ウィンドウ）での操作に大別できます。

スクリプトを完成させるまでの一連の操作は、マネージャウィンドウで実行できます。ただし、実際にスクリプトを記述する操作のときには、マネージャウィンドウからエディタ（またはユーザ指定のエディタ）を起動します。

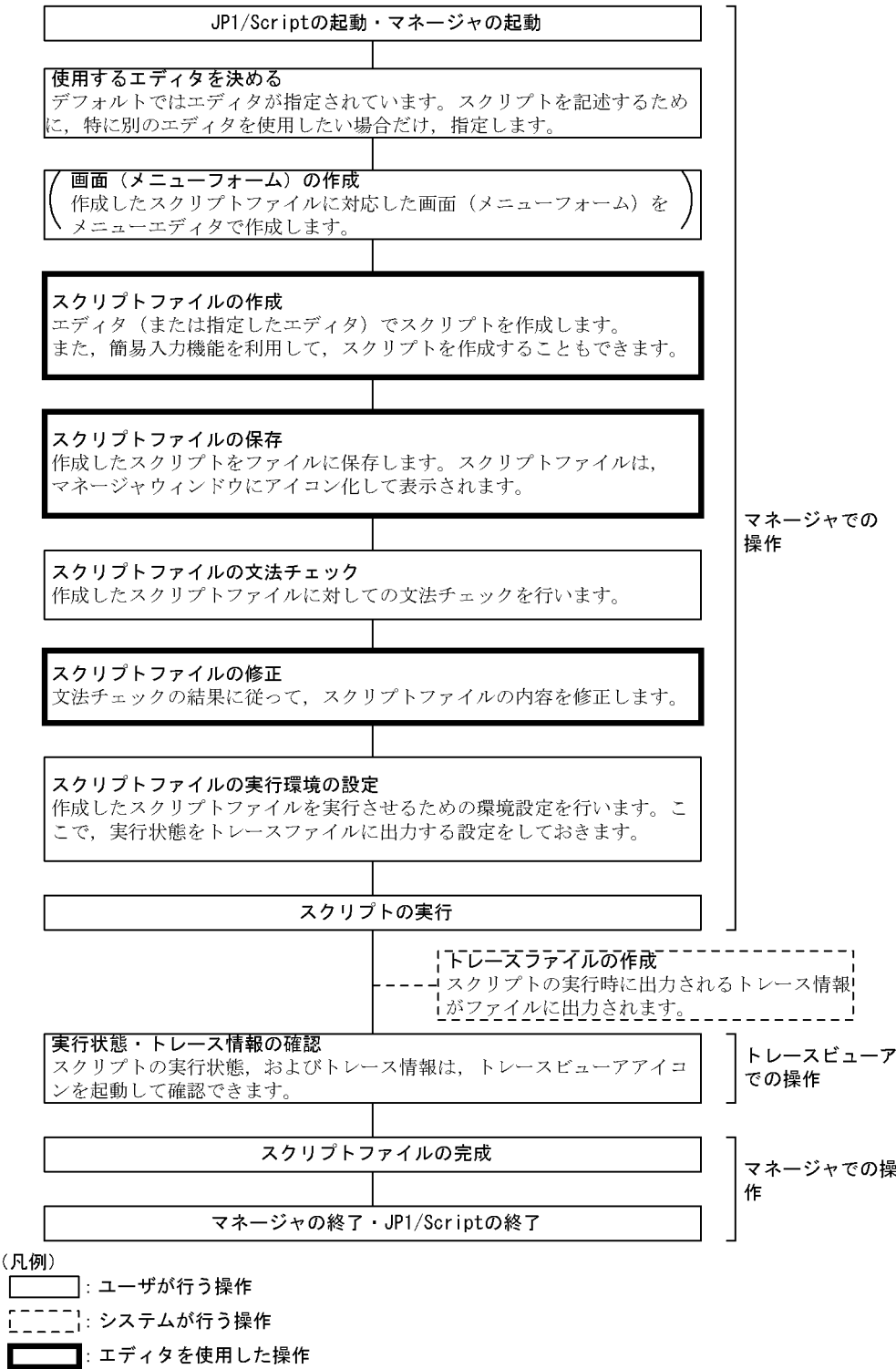
スクリプトの実行によって作成されたトレースファイルに関する操作は、トレースビューアウィンドウで行います。

ここでは、マネージャウィンドウでスクリプトファイルを完成させるまでの操作の流れを示します。

マネージャからエディタを起動して、スクリプトファイルを作成します。文法チェック機能を利用して、スクリプトの字句解析、構文解析を行えます。

1. JP1/Script の概要

図 1-8 JP1/Script の操作の流れ



## 1.7 スクリプトを作成する前に、実行する前に

---

### (1) Message コマンドについて

Message コマンドで Target 引数に「Target\_File」を指定して、ユニークなファイル名称を持ったユーザトレースファイルを大量に作成する場合、大量に作成されたファイル名、およびトレースファイルの書き出し位置などを管理するための情報が増えるため、トレース管理ファイルの容量が増えます。トレース管理ファイルの容量が増えると、以下の現象が発生します。

- スクリプトファイルの実行性能が劣化する
- スクリプトファイルの実行が終了コード「20」で異常終了する
- コマンドの実行がメモリ不足でエラーになる

ユニークなファイル名称を持ったユーザトレースファイルを大量に作成する場合は、TextOpen/TextWrite/TextClose コマンドを使用して、トレースファイルを作成してください。TextOpen/TextWrite/TextClose コマンドで作成するファイルは、トレース管理ファイルの管理対象外のファイルになるため、トレース管理ファイルの容量が増えることはありません。

### (2) 解析トレースファイルおよび実行トレースファイルについて

解析トレースファイル、および実行トレースファイルは標準でスクリプト実行時に発生するエラー情報を出力します。実行環境の設定によりトレースファイルを出力しない設定も可能ですが、スクリプト実行時に発生するステートメントやコマンドのエラーが出力されないため、原因が特定できない場合があります。このため、トレースファイルはできるだけ出力するようにしてください。トレースファイルを出力したくない場合は、各コマンドの実行結果を判定して、エラーの場合は、エラーとなったコマンド名、および \_RTN\_ 予約変数の内容をファイル等に出力して原因を特定する方法など、エラー処理設計を考慮したスクリプトを作成してください。

### (3) NetExec コマンドについて

NetExec コマンドを実行するスクリプトを複数実行した場合、NetExec コマンド単位で相手先コンピュータとの処理が行われるため、NetExec コマンドを実行するスクリプトを複数実行すると、スクリプトの実行性能が劣化します。NetExec コマンドを実行するスクリプトは大量に実行しないでください。

### (4) CallSpt コマンド、プロシージャの呼び出しについて

CallSPT コマンドで実行したスクリプトから CallSPT コマンドを実行したり、プロシージャから異なるプロシージャを呼び出すなど、階層構造のスクリプト処理を設計する場合、階層は2階層～3階層程度を目安として設計してください。階層が深くなったり、プロシージャの再帰的呼び出しを繰り返すとメモリ不足のエラーになります。

## 1.8 Windows Vista 対応に関して

Windows Vista で実装された機能や変更された機能により、Windows Server 2003 および Windows XP 上での動作と異なる機能があります。

ここでは、Windows Vista で従来の OS と異なる点、および管理者権限での実行について説明します。

### 1.8.1 インストール先フォルダ

Windows Vista ではインストール先のフォルダは、従来の OS と異なります。詳細は「2.1.1 プログラムのインストール先のフォルダ」を参照してください。

インストール先のフォルダが異なるため、次のフォルダが異なります。

- トラブル時の資料を採取するフォルダ
- バックアップおよびリカバリーするフォルダ
- SetGV コマンドで作成される「SPTGV.SPG」ファイルの作成先フォルダ
- NetExec コマンドのエラーログファイルの出力先フォルダ

### 1.8.2 予約変数

Windows Vista では OS に関する予約変数は次の値になります。

分類	予約変数	値
システム予約変数	_OS_	WIN_NT6.0
	_OS_PLATFORM_	WIN_NT
	_OS_REVISION_	0
	_OS_VERSION_	6

### 1.8.3 JP1/Script サービスの開始と監視

#### (1) JP1/Script サービスを開始する契機

Windows Vista では JP1/Script サービスが常時、実行中の状態でなければなりません。他サービスからスクリプトファイルを起動する場合、スクリプトファイルを起動するサービスよりも前に JP1/Script サービスを開始する必要があります。

JP1/Script サービスが実行中でない場合の影響

JP1/Script サービスが実行中の状態でない場合、次に示す影響があります。

スクリプトファイル起動時、解析 / 実行トレースが出力されない

スクリプトファイル起動時に JP1/Script サービスが実行中でない場合、イベントログ

に次の警告メッセージを出力します。

「ID 96: JP1/Script サービスの状態が実行中以外のため、解析 / 実行トレースに実行結果が出力されません。また Message/EntryStartUp/CancelStartUp/ グローバル変数操作コマンドの実行がエラーになります。JP1/Script サービスを開始してから実行してください。」

なお、JP1/Script サービスが実行中でない状態でもスクリプトファイルの起動は中止しません。

スクリプトファイルの実行中に JP1/Script サービスの状態が実行中になった場合、実行中になった以降の解析 / 実行トレースは出力されます。

Target\_File, Target\_SPXFile または Target\_SPXFile 指定の Messgase コマンド, EntryStartUp コマンド, CancelStartUp コマンド, グローバル変数操作コマンドの実行が失敗する

JP1/Script サービスの状態が実行中以外でエラーになった場合、\_RTN\_ 予約変数に \_ERR\_SERVICE\_NOT\_BEGIN\_ ( 値: 536903968 ) が設定されます。

サービスおよび他ユーザから実行したスクリプトプロセスを TerminateProcess コマンドで強制終了できない

指定されたプロセス ID のプロセスが存在しない場合と同様になるため、常に真 ( True ) を返します。

マネージャの [ ファイル ] - [ 実行環境の設定 ] - [ すべての項目 ] メニュー, [ ファイル ] - [ 実行環境の設定 ] - [ 起動日時 ] メニューおよび [ ツール ] - [ 自動起動の設定 ] メニューで自動起動情報の設定ができない

JP1/Script サービスが実行中でない状態で実行された場合、次のエラーダイアログを表示して設定ダイアログの表示を中止します。

「JP1/Script サービスの状態が実行中以外のため、実行できません。JP1/Script サービスを開始した後、再度実行してください。」

マネージャの [ ツール ] - [ オプション ] メニューで各種オプションを設定できない JP1/Script サービスが実行中でない状態で実行された場合、次のエラーダイアログを表示してオプションダイアログの表示を中止します。

「JP1/Script サービスの状態が実行中以外のため、実行できません。JP1/Script サービスを開始した後、再度実行してください。」

エディタの [ モニタリング ] - [ 実行環境の設定 ] メニューで自動起動情報を設定するダイアログを表示できない

自動起動情報の設定ができません。JP1/Script サービスが実行中でない状態で実行された場合、次のエラーダイアログを表示して設定ダイアログの表示を中止します。

「JP1/Script サービスの状態が実行中以外のため、実行できません。JP1/Script サービスを開始した後、再度実行してください。」

プロセスビューにサービスおよび他ユーザで実行中のプロセスが表示できない表示できない場合、次の警告ダイアログを表示した後、プロセスビューを起動します。

## 1. JP1/Script の概要

「JP1/Script サービスの状態が実行中以外のため、サービスから起動したプロセスおよび他ユーザで実行中のプロセスを表示しません。表示したい場合は、JP1/Script サービスを開始した後、再度実行してください。」

トレースビューアが起動できない

起動できない場合、次のエラーダイアログを表示してトレースビューアの起動を中止します。

「JP1/Script サービスの状態が実行中以外のため、実行できません。JP1/Script サービスを開始した後、再度実行してください。」

### (2) JP1/Script サービスを稼働監視プログラムで監視する条件

Windows Vista では必要に応じて JP1/Script サービスを稼働監視プログラムで監視してください。詳細は「2.6 稼働監視プログラムでの JP1/Script の監視」を参照してください。

## 1.8.4 コマンドの動作

Windows Vista では、サービスからスクリプトファイルを実行する場合、従来の OS と同様に、サービスに指定されているユーザアカウントの権限で動作します。しかし、マネージャからスクリプトファイルを実行する場合、管理者グループのユーザでログオンしても、OS のユーザアカウント制御 (UAC) により JP1/Script アプリケーションは標準ユーザで実行されます。

標準ユーザで実行されると影響を受けるコマンドを次に示します。

- アクセス拒否のエラーとなるコマンド
- ファイル操作系のコマンド
- Exec , NetExec コマンド
- TerminateProcess コマンド
- GetProcessInfo コマンド

### (1) アクセス拒否のエラーとなるコマンド

アクセス拒否のエラーとなるコマンドを表 1-3 に示します。

これらのコマンドを実行するスクリプトファイルは、管理者権限で実行する必要があります。標準ユーザで実行した場合、コマンドが次のエラーになります。

「0005: アクセスは拒否されました。ファイルの属性又はセキュリティを見直してください。」

スクリプトファイルを管理者権限で実行する方法については「1.8.8 管理者権限での実行」を参照してください。



表 1-3 アクセス拒否のエラーとなるコマンド

No.	コマンド	アクセス拒否のエラーとなる条件
1	SetEnviroment	第一引数 (Type) に SystemEnv を指定
2	Rename	第三引数 (Metod) に Reboot を指定
3	SetVollLabel	なし
4	RegWrite	第一引数 (RegKey) に HKEY_CURRENT_USER 以外を指定 <sup>1</sup>
5	RegDelete	第一引数 (RegKey) に HKEY_CURRENT_USER 以外を指定 <sup>2</sup>
6	RegDeleteKey	第一引数 (RegKey) に HKEY_CURRENT_USER 以外を指定 <sup>3</sup>
7	ServiceCreate	なし
8	ServiceDelete	なし
9	ServiceStart	なし
10	ServiceStop	なし
11	ServicePause	なし
12	ServiceContinue	なし
13	ServiceChange	なし
14	ServiceControl	なし
15	MakeGroup	第二引数 (RootType) に Lcl_Program を指定
16	DeleteGroup	第二引数 (RootType) に Lcl_Program を指定
17	MakeShortcut	第一引数 (RootPath) に Lcl_Desktop, Lcl_Startmenu, Lcl_Program または Lcl_Startup を指定
18	DeleteShortcut	第一引数 (RootPath) に Lcl_Desktop, Lcl_Startmenu, Lcl_Program または Lcl_Startup を指定
19	GetProcessCount	なし
20	GetProcessInfo	なし
21	TerminateProcess	なし

## 注 1

第一引数 (RegKey) に HKEY\_LOCAL\_MACHINE, 第二引数 (SubKey) の最初のキーに Software を指定した場合, HKEY\_CURRENT\_USER¥Software¥Classes¥VirtualStore¥Machine¥Software 下にリダイレクトされるためエラーになりません。RegWrite コマンドによってリダイレクトされたエントリは RegRead コマンドで優先して読み込まれます。

## 注 2

リダイレクトされた Software 下の値を削除する場合, リダイレクト先のキーを指定する必要があります。

## 注 3

リダイレクトされた Software 下のキーに対して実行した場合, キーは削除されません。

## (2) ファイル操作系のコマンド

ファイル操作系のコマンドで、環境変数 ProgramFiles および WinDir に設定されているフォルダ下のファイルに対して出力・更新した場合、ユーザごとの領域へリダイレクトされるときがあるので注意が必要です。リダイレクト先は環境変数 LocalAppData に設定されているパスの VirtualStore 下になります。リダイレクトされる条件は、ファイル操作系のコマンドを実行するスクリプトファイルを標準ユーザで実行することです。リダイレクトされないためにはスクリプトファイルを管理者権限で実行しなければなりません。スクリプトファイルを管理者権限で実行する方法については「1.8.8 管理者権限での実行」を参照してください。

標準ユーザで実行したスクリプトファイルからファイル操作系コマンドで入力・参照した場合、リダイレクト先のファイルが優先して操作対象になります。

## (3) Exec , NetExec コマンド

Exec , NetExec コマンドは Install.exe , Setup.exe などの管理者権限が必要な実行ファイルを呼び出すため、Exec , NetExec コマンドを実行するスクリプトファイルを標準ユーザから実行した場合、次の現象が発生します。

1. Exec , NetExec コマンドが次のエラーになる。  
「0740: 要求された操作には管理者特権が必要です。」
2. 権限昇格の [ ユーザアカウント制御 ] ダイアログが表示される。

通常は最初に Exec , NetExec コマンドが 1. のエラーになり、その後 OS のプログラム互換アシスタント機能が動作して 2. のダイアログが表示されます。

エラーおよびダイアログの表示を抑止したい場合は、スクリプトファイルを管理者権限で実行しなければなりません。スクリプトファイルを管理者権限で実行する方法については「1.8.8 管理者権限での実行」を参照してください。

権限昇格の [ ユーザアカウント制御 ] ダイアログで [ キャンセル ] ボタンをクリックした場合、Exec , NetExec コマンドが次のエラーになります。

「1223: この操作はユーザーによって取り消されました。」

## (4) TerminateProcess コマンド

他ユーザで実行中のプロセスは強制終了できません ( スクリプトプロセスは強制終了できます )。他ユーザのプロセスを指定してもエラーにならないで正常終了します ( 存在しないプロセスを指定したときと同じ動作です )。

## (5) GetProcessInfo コマンド

自ユーザで実行中のプロセス情報だけ取得できます。他ユーザで実行中のプロセス ID を指定した場合、アクセス拒否のエラーになります。また、サービス空間で実行中のスクリプトからログオン空間で実行中のプロセス ID を指定した場合もアクセス拒否のエラー

になります。

## 1.8.5 マネージャの設定方法

Windows Vista ではマネージャの [ ツール ] - [ オプション ] メニューで表示されるダイアログの各タブを設定する場合、マネージャを管理者権限で実行しなければなりません。管理者権限で実行しなかった場合、設定したユーザにだけ適用されます。マネージャを管理者権限で実行する方法については「1.8.8 管理者権限での実行」を参照してください。

## 1.8.6 レジストリの変更方法

Windows Vista では HKEY\_LOCAL\_MACHINE¥SOFTWARE¥HITACHI¥JP1/Script 下のレジストリキーの値を設定・変更する場合、レジストリエディタを管理者権限で実行しなければなりません。レジストリエディタを管理者権限で実行するには、管理者グループのユーザでログオンしたあとにレジストリエディタを起動してください。起動後、権限昇格の [ ユーザアカウント制御 ] ダイアログが表示されますので、[ 続行 ] ボタンをクリックしてください。

## 1.8.7 メッセージボックスのレイアウト

Windows Vista では MessageBox コマンドなど、JP1/Script で表示するメッセージボックスのボタンは右揃えで表示されます。

## 1.8.8 管理者権限での実行

OS でユーザアカウント制御 (UAC) を無効化すれば、ログオンしたユーザの権限で動作させることができますが、OS のセキュリティが低下するので、できる限り下記の操作をしてください。

### (1) マネージャの起動

管理者権限で起動したマネージャから実行するスクリプトファイル、エディタなどの各機能は、管理者権限で実行されます。この場合、エディタからのスクリプトファイルのモニタリング実行も、管理者権限で実行されます。

次のどちらかの方法によりマネージャを管理者権限で起動できます。

#### (a) [ 管理者権限として実行 ] を指定する

操作

1. マネージャのショートカットを右クリックして表示されるメニューの [ 管理者権限として実行 ] を選択する。  
権限昇格の [ ユーザアカウント制御 ] ダイアログが表示されます。

## 1. JP1/Script の概要

2. [ 続行 ] ボタンをクリックする。

3. 管理者グループでないユーザから操作した場合、管理者のパスワードを入力する。

(b) 特権レベル [ 管理者としてこのプログラムを実行する ] を指定する

操作

1. マネージャのショートカットを右クリックして表示されるメニューの [ プロパティ ] を選択する。

[ プロパティ ] ダイアログが表示されます。

2. [ 互換性タブ ] の特権レベル [ 管理者としてこのプログラムを実行する ] をチェックする。

3. マネージャを起動する。

権限昇格の [ ユーザアカウント制御 ] ダイアログが表示されます。

管理者グループでないユーザでは、2. の指定は有効にならないで標準ユーザで起動されます。

## (2) スクリプトファイルの実行

次のどちらかの方法によりスクリプトファイルを管理者権限で実行できます。

(a) 特権レベル [ 管理者としてこのプログラムを実行する ] を指定する

操作

1. JP1/Script インストールフォルダ下の sptxe.exe を右クリックして表示されるメニューの [ プロパティ ] を選択する。

[ プロパティ ] ダイアログが表示されます。

2. [ 互換性タブ ] の特権レベル [ 管理者としてこのプログラムを実行する ] をチェックする。

3. すべてのスクリプトファイルを実行する。

権限昇格の [ ユーザアカウント制御 ] ダイアログが表示されます。

管理者グループでないユーザでは、2. の指定は有効にならないで標準ユーザで起動されます。

(b) [ 管理者権限として実行 ] を指定する

操作

1. JP1/Script インストールフォルダ下の sptxe.exe のショートカットをデスクトップなどに作成する。

2. 作成したショートカットを右クリックして表示されるメニューの [ プロパティ ] を選択する。

[ プロパティ ] ダイアログが表示されます。

3. [ ショートカット ] タブの [ リンク先 ] の sptxe.exe の後ろにスクリプトファイル名を

指定する。

4. ショートカットを右クリックして表示されるメニューの [ 管理者権限として実行 ] を選択する。  
sptxe.exe が実行され、権限昇格の [ ユーザアカウント制御 ] ダイアログが表示されます。
5. [ 続行 ] ボタンをクリックする。



# 2

## JP1/Script を利用するための 準備

この章では、JP1/Script のインストール方法、アンインストール方法、他のコンピュータとの通信時のアカウント、各種システム環境での環境設定方法などについて説明します。

---

2.1 JP1/Script のインストール / アンインストール

---

2.2 他のコンピュータとの通信時のアカウント

---

2.3 クラスタシステム環境での環境設定

---

2.4 ターミナルサービス環境での環境設定

---

2.5 ホスト名、または IP アドレスの変更手順

---

2.6 稼働監視プログラムでの JP1/Script の監視

---

2.7 JP1/Script の起動と終了

---

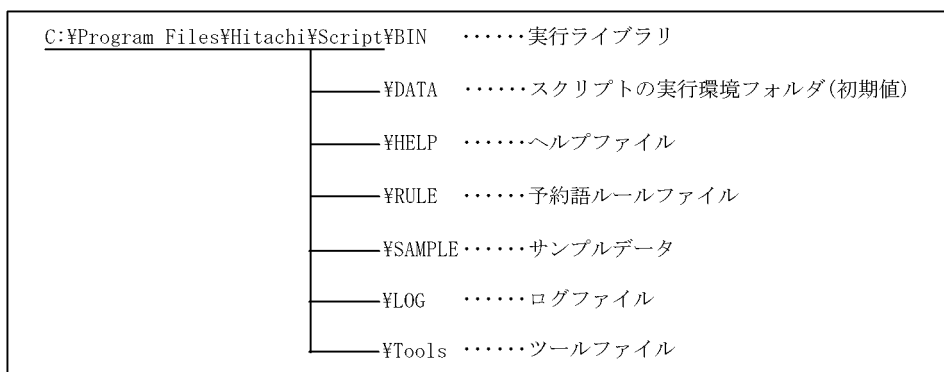
## 2.1 JP1/Script のインストール / アンインストール

ここでは、JP1/Script のインストールの方法、サービスの登録、およびアンインストールの方法について説明します。

### 2.1.1 プログラムのインストール先のフォルダ

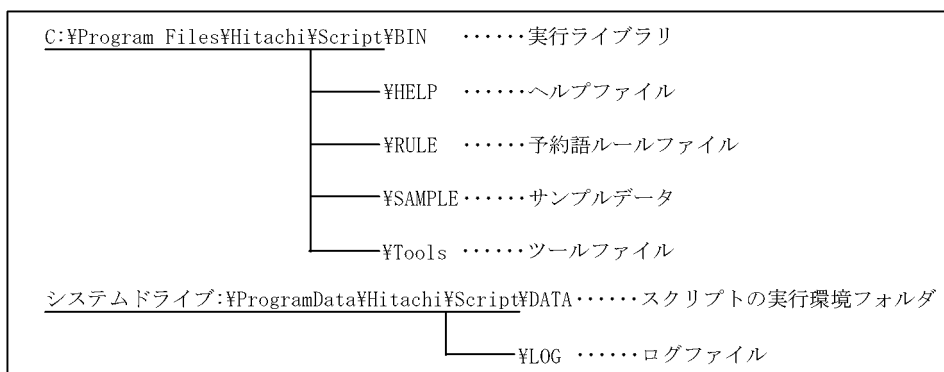
JP1/Script プログラムのインストール先のフォルダを次に示します。

#### (1) Windows Server 2003 または Windows XP の場合



(上記は Windows がインストールされているドライブのルートが C : の場合)

#### (2) Windows Vista の場合



(上記は Windows がインストールされているドライブのルートが C : の場合)

スクリプトの実行環境フォルダ ( ¥DATA ) およびログファイル ( ¥LOG ) は、無条件に上記フォルダとなります ( インストール時に変更できません )。システムドライブの容量



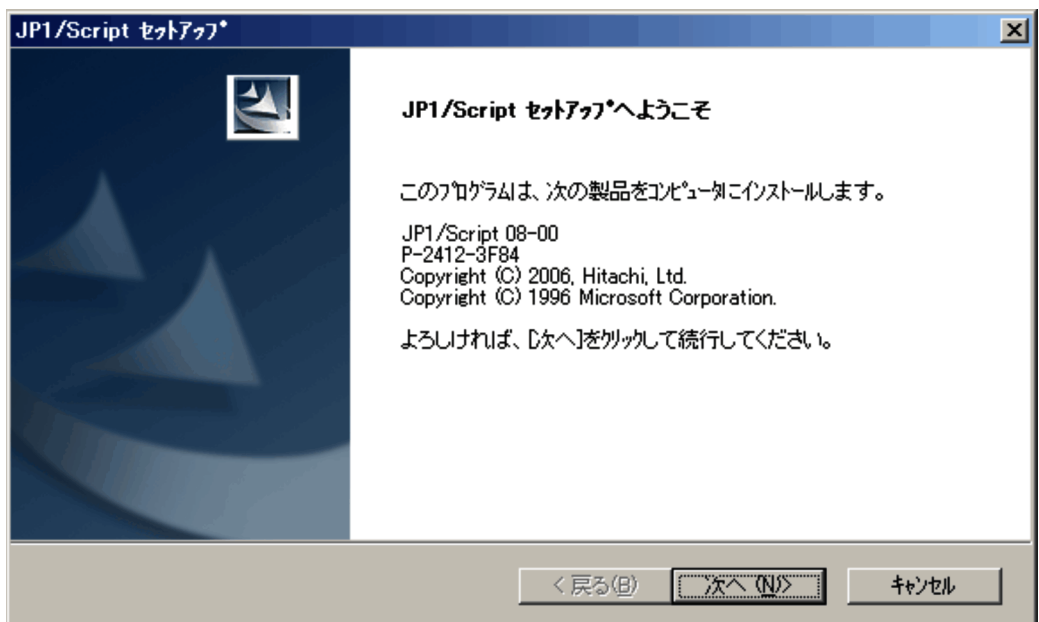
不足などで格納先を変更したい場合は、マネージャの [ ツール ] - [ オプション ] メニューで表示される [ オプション ] ダイアログの [ クラスタの環境設定 ] で変更できます。ただし、システムドライブの ¥Program Files フォルダ下に変更してはいけません。¥Program Files フォルダ下は OS によってリダイレクトされるため、プログラムが正常に動作しなくなります。

## 2.1.2 Windows Server 2003 および Windows XP 版 JP1/Script ( P-2412-3F84 ) のインストール

JP1/Script のインストールの手順を次に示します。インストールの前に管理者権限でログオンしておいてください。

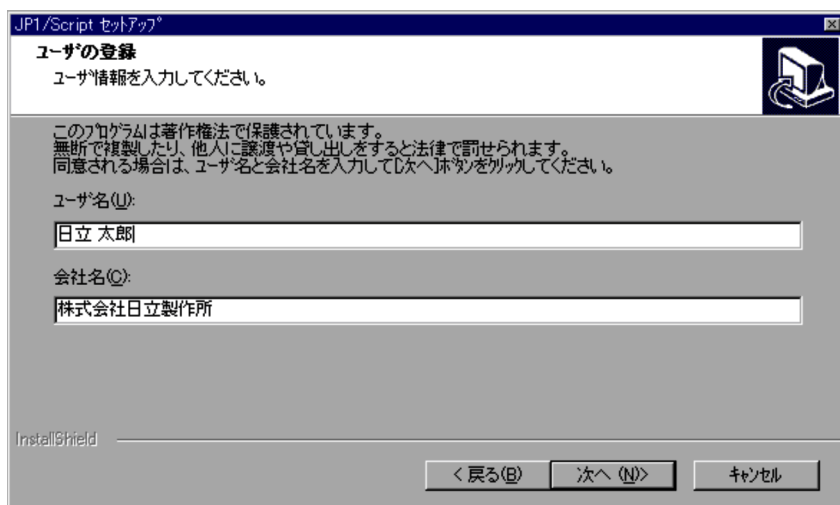
### 操作

1. インストールに使用する CD-ROM を CD-ROM ドライブにセットする。
2. CD-ROM に格納されている「INSTALL.EXE」を起動する。  
JP1/Script セットアップダイアログが表示されます。
3. インストールするプログラムを確認して [ 次へ ] ボタンをクリックする。

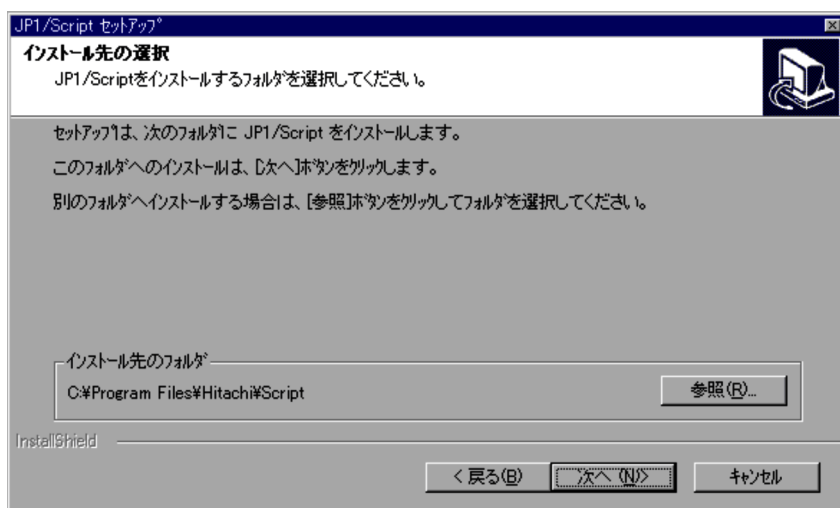


4. ユーザ情報（ユーザ名と会社名）を登録して [ 次へ ] ボタンをクリックする。  
指定したユーザ名と会社名がユーザ情報として、レジストリファイルに登録されます。

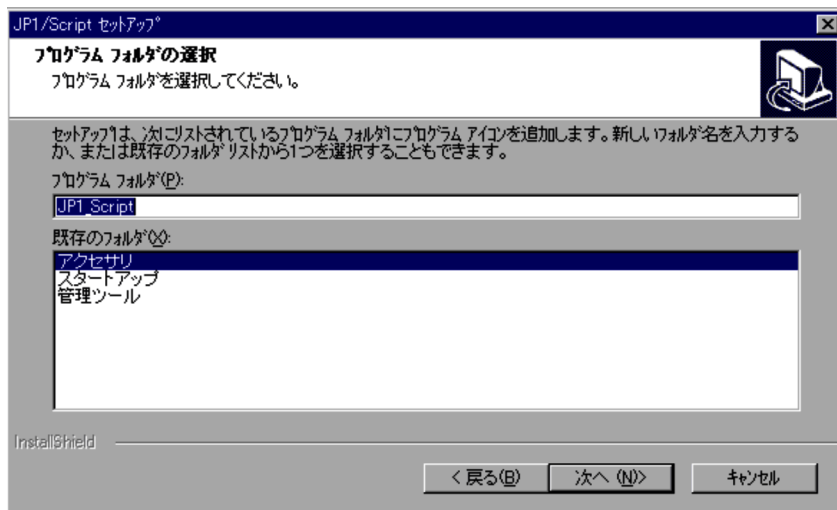
## 2. JP1/Script を利用するための準備



5. JP1/Script を格納するフォルダを指定して [ 次へ ] ボタンをクリックする。  
存在しないフォルダを指定した場合は、インストーラが自動的にフォルダを作成します。

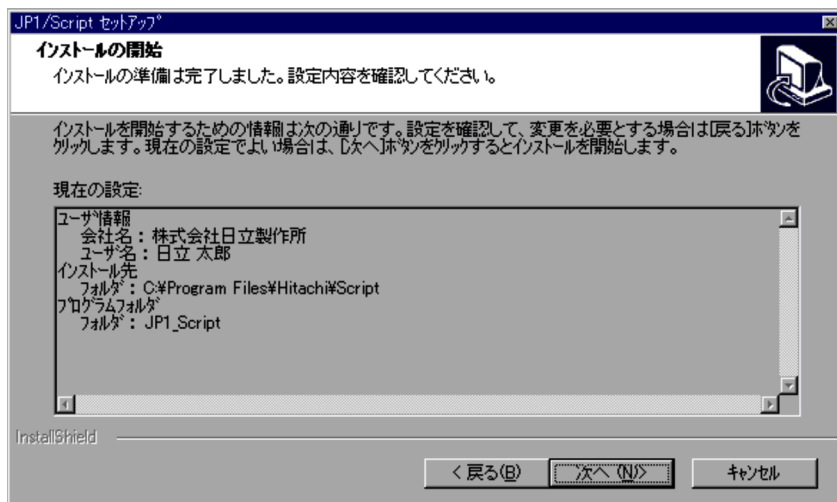


6. JP1/Script のプログラムアイコンを追加するプログラムフォルダを指定して [ 次へ ] ボタンをクリックする。



7. 操作 4. から操作 6. で指定した内容を確認する。

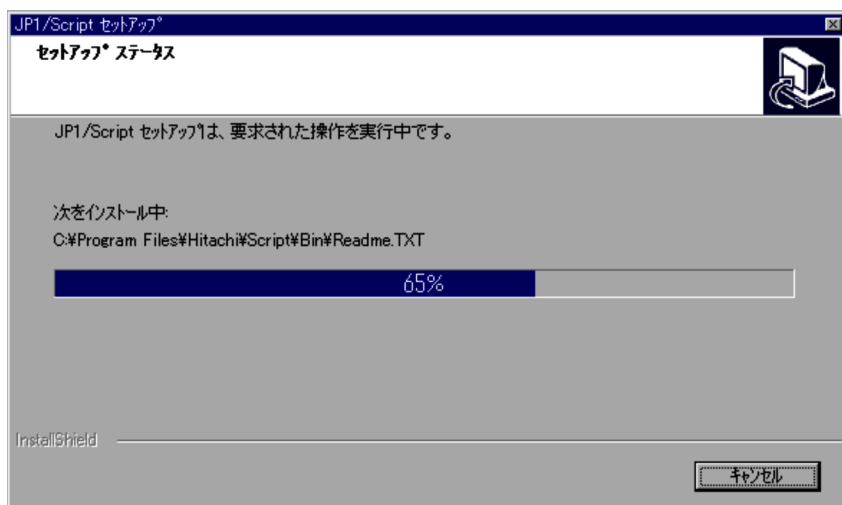
ダイアログに現在の設定が表示されるので、内容を確認します。この時点で、インストールの準備は完了しました。



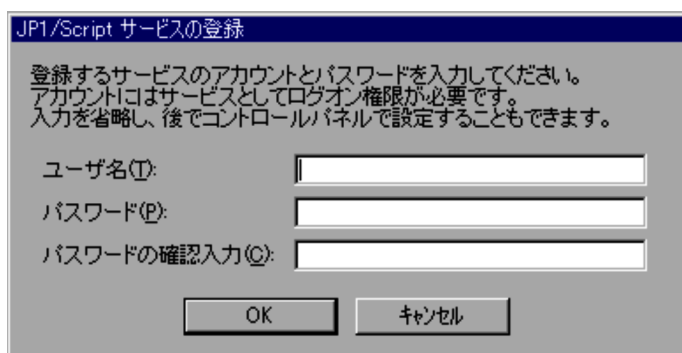
8. [ 次へ ] ボタンをクリックする。

インストールが開始され、インストール中のファイル名、およびインストールの進行状態を示すダイアログが表示されます。

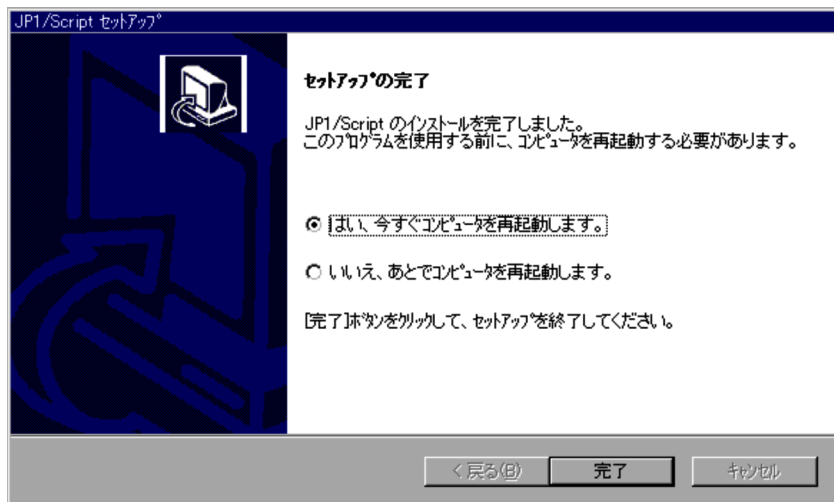
2. JP1/Script を利用するための準備



9. JP1/Script のサービス登録に必要なユーザ名とパスワードを指定して [ OK ] ボタンをクリックする。



10. [ README ファイルの参照 ] ダイアログに対して、[ 次へ ] ボタンをクリックする。  
README ファイルを参照するかどうかを選択して [ 次へ ] ボタンをクリックします。  
参照する場合は、README ファイルが表示されます。
11. セットアップの完了ダイアログに対して、[ 完了 ] ボタンをクリックする。



直ちにコンピュータを再起動するかどうかを選択して [ 完了 ] ボタンをクリックします。「はい、今すぐコンピュータを再起動します。」を選択した場合は、システムが再起動されます。

#### ！ 注意事項

操作 10. で README ファイルを参照しなかった場合は、インストール終了後、JP1/Script のインストールフォルダ下に格納されている README ファイルを必ず読んでください。

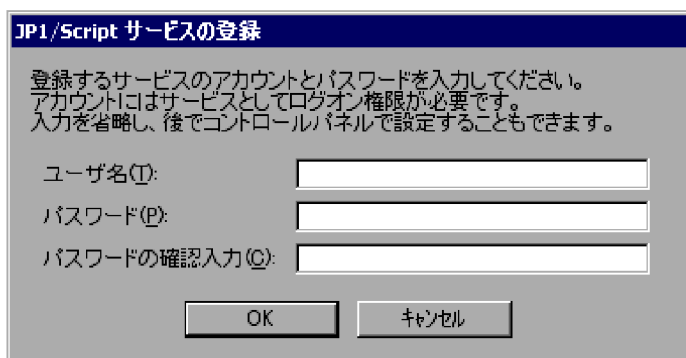
### 2.1.3 Windows Server 2003 および Windows XP 版 JP1/Script ( P-2412-3F84 ) サービスの登録

JP1/Script サービスに登録するアカウントには、サービスとしてログオンする権限が必要です。アカウント名の登録は、インストール時には省略できます。

インストール終了後にアカウント名を登録したり変更したりする場合は、コントロールパネルの [ サービス ] アイコンを選択してください。また、JP1/Script サービスから SetGV, GetGV, DeleteGV, NetExec コマンドで他のコンピュータを呼び出す場合、JP1/Script サービスに適切なアカウント名 ( 呼び出すコンピュータにログオンできるアカウント名 ) を設定してください。

適切なアカウント名を設定しないと上記コマンドは正常に機能しません。

## 2. JP1/Script を利用するための準備



JP1/Script サービスの登録

登録するサービスのアカウントとパスワードを入力してください。  
アカウントにはサービスとしてログオン権限が必要です。  
入力を省略し、後でコントロールパネルで設定することもできます。

ユーザ名(U):

パスワード(P):

パスワードの確認入力(C):

OK キャンセル

「OK」ボタン：ユーザ名とパスワードを登録する

「キャンセル」ボタン：インストールを中断する

SetGV、GetGV、および DeleteGV コマンドについては「8.2.5 SetGV (グローバル変数を設定する)」、「8.2.6 GetGV (グローバル変数を取得する)」、および「8.2.7 DeleteGV (グローバル変数を削除する)」を参照してください。

NetExec コマンドについては「8.10.2 NetExec (自 PC / 他 PC 上の実行ファイルを呼び出す)」を参照してください。

また、ログオンアカウントについては「2.2 他のコンピュータとの通信時のアカウント」を参照してください。

### ! 注意事項

アカウントを指定しなかった場合は、システムアカウントで実行されるため、他のシステムアカウントで実行されているサービスプログラムとシステムリソースが共用され影響を受ける場合があります。システムアカウントで動作している他のサービスプログラムが存在する場合は、アカウントを指定してください。

## 2.1.4 64 ビットバージョン Windows Server 2003 (IPF) 版 JP1/Script (P-2812-3F84) のインストール

JP1/Script のインストールの手順を次に示します。インストールの前に管理者権限でログインしておいてください。

### 操作

1. 提供媒体を CD-ROM ドライブに入れ、インストールする  
自動的に起動するインストーラの指示に従ってインストールします。  
新規インストールの場合は、次の項目を入力します。
  - ユーザ情報

- インストール先フォルダ

## 2. Windows を再起動する

### 2.1.5 64 ビットバージョン Windows Server 2003 ( IPF ) 版 JP1/Script ( P-2812-3F84 ) サービスの登録

JP1/Script サービスに登録するアカウントには、サービスとしてログオンする権限が必要です。なお、インストール時にはシステムアカウントが設定されます。

アカウント名を変更する場合は、管理ツールの [ サービスメニュー ] を選択してください。また、JP1/Script サービスから SetGV、GetGV、DeleteGV、NetExec コマンドで他のコンピュータを呼び出す場合、JP1/Script サービスに適切なアカウント ( 呼び出すコンピュータにログオンできるアカウント ) を設定してください。適切なアカウントを設定しないと上記コマンドは正常に機能しません。

また、ログオンアカウントについては、「2.2 他のコンピュータとの通信時のアカウント」を参照してください。

#### ! 注意事項

システムアカウントで実行すると、他のシステムアカウントで実行されているサービスプログラムとシステムリソースが共有され影響を受ける場合があります。システムアカウントで動作している他のサービスプログラムがある場合は、アカウントを指定してください。

### 2.1.6 Windows Vista 版 JP1/Script ( P-2A12-3F84 ) のインストール

JP1/Script のインストールの手順を次に示します。インストールの前に管理者権限でログオンしておいてください。

#### 操作

1. 提供媒体を CD-ROM ドライブに入れ、インストールする  
自動的に起動するインストーラの指示に従ってインストールします。  
新規インストールの場合は、次の項目を入力します。
  - ユーザ情報
  - インストール先フォルダ
2. Windows を再起動する

### 2.1.7 Windows Vista 版 JP1/Script ( P-2A12-3F84 ) サービスの登録

JP1/Script サービスに登録するアカウントには、サービスとしてログオンする権限が必要です。なお、インストール時にはシステムアカウントが設定されます。

アカウント名を変更する場合は、管理ツールの [ サービスメニュー ] を選択してください。また、JP1/Script サービスから SetGV, GetGV, DeleteGV, NetExec コマンドで他のコンピュータを呼び出す場合、JP1/Script サービスに適切なアカウント ( 呼び出すコンピュータにログオンできるアカウント ) を設定してください。適切なアカウントを設定しないと上記コマンドは正常に機能しません。

また、ログオンアカウントについては、「2.2 他のコンピュータとの通信時のアカウント」を参照してください。

#### ！ 注意事項

システムアカウントで実行すると、他のシステムアカウントで実行されているサービスプログラムとシステムリソースが共有され影響を受ける場合があります。システムアカウントで動作している他のサービスプログラムがある場合は、アカウントを指定してください。

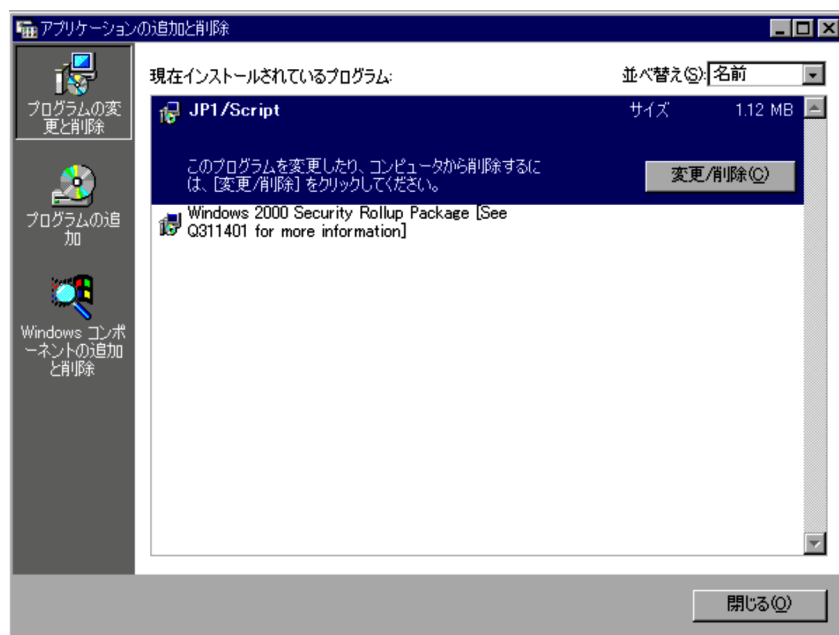
### 2.1.8 JP1/Script のアンインストール

JP1/Script のアンインストールの手順を次に示します。

#### 操作

1. 管理者権限でログオンする。  
管理者権限でログオンしていない場合は、正常にアンインストールされません。
2. 実行中の JP1/Script のプログラムをすべて終了する。  
すべて終了していない場合は、正常にアンインストールされません。
3. コントロールパネルの [ アプリケーションの追加と削除 ] を利用してアンインストールする。





#### アンインストールをするときの注意事項

- JP1/Script のインストール先フォルダのファイルは、すべて削除されます。必要なファイルなどは、バックアップしてください。
- JP1/Script のインストール先フォルダ内のファイルを、ほかのアプリケーションなどでアクセス中の場合、インストール先フォルダは、アンインストールしても削除されないことがあります。
- アンインストールを完全なものとするために、アンインストール終了後、システムを再起動してください。

## 2.2 他のコンピュータとの通信時のアカウント

ここでは、他のコンピュータとの通信時のアカウントについて説明します。

SetGV, GetGV, DeletGV, NetExec コマンドで他のコンピュータを呼び出す場合、呼び出し元のログオンアカウントとパスワードは適切なアカウント（呼び出すコンピュータにログオンできるアカウントとパスワード）でなければ正常に機能を果しません。

呼び出し元のログオンアカウントとパスワードはスクリプトの実行方法によって表 2-1 のようになります。

表 2-1 コマンド呼び出し元のログオンアカウントとパスワード

No.	スクリプトの実行方法	呼び出し元のログオンアカウントとパスワード
1	自動起動により JP1/Script サービスから実行します。	JP1/Script サービスに設定されたログオンアカウントとパスワード (システムアカウントは不可)
2	JP1/AJS2 のサービスプログラムから実行します。	スクリプトを実行する JP1 ユーザにマッピングされた OS ユーザのアカウントとパスワード
3	JP1/AJS2 以外のサービスプログラムから実行します。	各サービスに設定されたログオンアカウントとパスワード (システムアカウントは不可)
4	自動起動により Script ランチャから実行します。	コンピュータをログオンしているアカウントとパスワード
5	マネージャやエディタから実行します。	
6	エクスプローラなどから直接実行します。	

また、NetExec コマンドの場合、接続が成功して呼び出したコンピュータ上で実行ファイルが実行される際の実行アカウントとパスワードは表 2-2 のようになります。

表 2-2 NetExec コマンドから呼び出す実行ファイルのアカウントとパスワード

No.	NetExec コマンドの第 5 引数の値	呼び出す実行ファイルのログオンアカウントとパスワード
1	True (サービス空間で実行)	呼び出し先コンピュータの JP1/Script サービスのログオンアカウントとパスワード
2	False (ログオン空間で実行)	呼び出し先コンピュータをログオンしているアカウントとパスワード

SetGV, GetGV, および DeleteGV コマンドについては「8.2.5 SetGV (グローバル変数を設定する)」、「8.2.6 GetGV (グローバル変数を取得する)」、および「8.2.7 DeleteGV (グローバル変数を削除する)」を参照してください。

NetExec コマンドについては「8.10.2 NetExec ( 自 PC / 他 PC 上の実行ファイルを呼び出す )」を参照してください。

## 2.3 クラスタシステム環境での環境設定

---

二つのクラスタシステム環境のうち、片方を業務が実行できアクティブ状態になっている実行システム、もう片方をフェールオーバー時に実行システムに切り替わるまでスタンバイ状態になっている待機システムとします。このアクティブ・スタンバイ構成の場合、この環境下で JP1/Script を使用することはできますが、フェールオーバー時に実行中のスクリプトを継続して実行することはできません。

しかし、フェールオーバー時に実行環境を引き継ぐことができます。そのためには次のような環境設定を行います。

### 操作

1. JP1/Script をインストールする。

JP1/Script をインストールします。インストール先のフォルダはローカルドライブ上のフォルダにしてください。

2. JP1/Script サービスのスタートアップの種類を「手動」に変更する。

[ スタート ] - [ 設定 ] - [ コントロールパネル ] - [ サービス ] ダイアログで JP1/Script サービスのスタートアップの種類を「自動」から「手動」に変更します。

### 注

Windows 2000 では、[ コントロールパネル ] - [ 管理ツール ] - [ サービス ]

3. Script ランチャを終了し、スタートアップから削除する。

Script ランチャが起動している場合は終了させてください。Script ランチャを終了させる方法は、JP1/Script のインストール先フォルダにある "SPTHSTP.exe" にコマンドライン "/INST" を付けて起動してください。終了後、スタートアップに登録されている Script ランチャを削除します。

4. 使用する共有ディスクを決定する。

操作 7. で JP1/Script の管理ファイルの出力先に指定する共有ディスクを決めます。

5. クラスタアドミニストレータを起動する。

[ スタート ] - [ プログラム ] - [ 管理ツール ] - [ クラスタアドミニストレータ ] ウィンドウを起動します。

6. リソースを作成する。

[ クラスタアドミニストレータ ] ウィンドウで次に示す四つのリソースを作成します。

#### (1) 操作 4. で決定した共有ディスク リソース

- ・リソースの種類：物理ディスク
- ・リソース名：任意

#### (2) 論理 IP アドレス リソース

- ・リソースの種類：IP アドレス
- ・リソース名：任意
- ・リソースの依存関係：(1) のリソース

## (3) Script ランチャ リソース

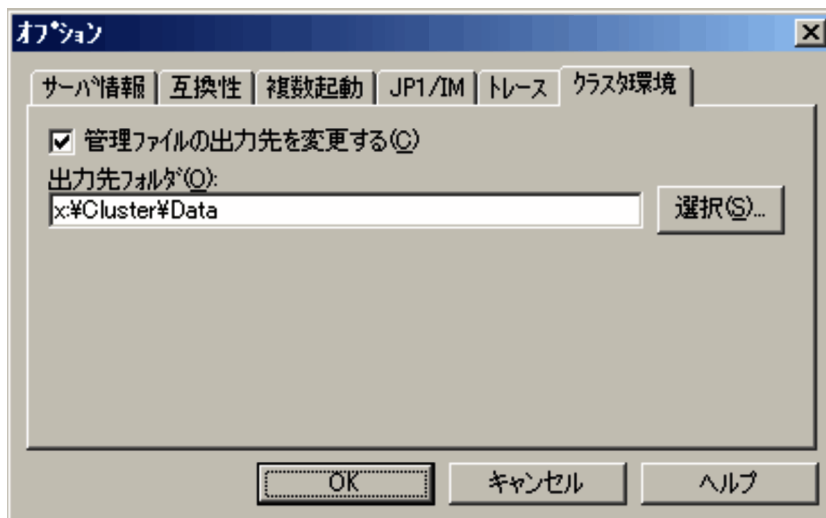
- ・リソースの種類：汎用アプリケーション
- ・リソース名：任意
- ・コマンドライン：JP1/Script のインストール先フォルダの "SPTHLNCH.exe"
- ・「デスクトップとの対話をアプリケーションに許可する」をチェック

## (4) JP1/Script サービス リソース

- ・リソースの種類：汎用サービス
- ・リソース名：任意
- ・リソースの依存関係：(1),(2),(3) のリソース
- ・サービス名：JP1\_Script

## 7. JP1/Script の管理ファイルの出力先を共有ディスク上のフォルダに変更する。

マネージャウィンドウの [ ツール ] - [ オプション ( クラスタ環境 ) ] ダイアログで管理ファイルの出力先を操作 4. で決定した共有ディスク上のフォルダに変更します。



## 8. システムを再起動する。

操作 7. で設定した内容を有効にするため、システムを再起動します。

操作 4. から操作 6. は二つのクラスタシステム環境のうちどちらかで設定します。それ以外の操作は二つのシステムでそれぞれ設定します。

## 処理詳細

二つのクラスタシステム環境ともアクティブ状態の実行システムになっているアクティブ・アクティブ構成の場合は、JP1/Script を使用することはできませんが、フェールオーバー時に実行中のスクリプトを継続して実行することはできません。また、フェールオーバー時に実行環境を引き継ぐこともできませんので、上記の環境設定は行わないでください。

## 2.4 ターミナルサービス環境での環境設定

---

JP1/Script では、Windows 2000 ターミナルサービス、または Windows Server 2003 ターミナルサービス上で JP1/Script を使用できます。管理者以外のユーザでリモートセッションからログインして JP1/Script を操作する場合、ログインするユーザに「グローバルオブジェクトの作成」の権利を OS のセキュリティの設定で割り当ててください。また、ターミナルサービス環境で JP1/Script を使用する場合は、次の点に注意してご使用ください。

- Script ランチャはターミナルサーバ側のプライマリセッションでだけ起動でき、ターミナルクライアント側のリモートセッションでは起動できません。なお、JP1/Script サービスは両セッションでサービスの開始 / 停止ができます。
- 自動起動に登録されたスクリプトファイルは起動のタイプにかかわらず、ターミナルサーバ側のプライマリセッションで起動されます。
- NetExec コマンドから呼び出す実行ファイルは実行する空間のタイプにかかわらず、ターミナルサーバ側のプライマリセッションで起動されます。

### 注意事項

Beep コマンドはターミナルサーバ側が Windows 2000 の場合に使用できます。それ以外で実行した場合、「0005：アクセスは拒否されました。ファイルの属性又はセキュリティを見直してください」などのエラーになります。ターミナルクライアント側のリモートセッションからは Beep コマンドを使用しないでください。

---

## 2.5 ホスト名，または IP アドレスの変更手順

---

ホスト名，または IP アドレスを変更する場合は，次に示す手順で作業を行ってください。

### 操作

1. 他 PC から変更する PC に対して実行しているスクリプトがないことを確認してください。
2. JP1/Script サービス，および Script ランチャを停止してください。  
Script ランチャを終了させる方法は，JP1/Script のインストール先フォルダにある "Spthstp.exe" にコマンドライン "/INST" を付加して起動してください。
3. 変更後，JP1/Script サービス，および Script ランチャを開始してください。  
Script ランチャの開始方法は，JP1/Script のプログラムフォルダから「Script ランチャ」を実行してください。

### 注意

- NetExec，GetGV，SetGV，DeleteGV コマンドのコンピュータ名を FQDN（完全修飾ドメイン名）形式で指定していてドメイン名を変更する場合，指定しているコンピュータ名を変更してください。
- JP1/Script サービスのアカウントにユーザアカウントを設定していてドメイン名を変更した場合，アカウントを見直してください。

## 2.6 稼働監視プログラムでの JP1/Script の監視

プロセス監視機能を使用して JP1/Script のプロセスを監視する場合は、下記の監視プロセス情報に従って JP1/Cm2/SSO のプロセス監視条件を設定してください。

- 監視 AP 名：JP1/Script V8.0(Windows)
- 付加情報：JP1/Script(Windows)
- 監視プロセス数：2

親プロセス					子プロセス				
プロセス数	種別 <sup>1</sup>	プロセス名	しきい値		プロセス数	種別 <sup>1</sup>	プロセス名	しきい値	
			下限	上限				下限	上限
2	2	SPTHSV	1	1	1	2	SPTTMS	1	1
	2	Spthlnch	1	1 2	0	-	-	-	-

注 1 プロセス名の種別（1：コマンドライン名，2：実行ファイル名）を示します。

注 2 ターミナルサービスを利用してログオンする場合，以下の値を指定してください。

上限 = 1 + ターミナルサービスを利用して同時にログオンするユーザ数

設定方法については，JP1/Cm2/SSO のマニュアル「JP1 Version 8 JP1/Cm2/SNMP System Observer」を参照してください。

なお，JP1/Cm2/SNMP System Observer は JP1/Performance Management/SNMP System Observer，または JP1/Server System Observer の後継製品です。JP1/Performance Management/SNMP System Observer をご使用の場合は，マニュアル「JP1 Version 7i JP1/Performance Management/SNMP System Observer」をまた JP1/Server System Observer をご使用の場合は，マニュアル「JP1 Version 6 統合ネットワーク管理システム サーバシステム管理」を参照してください。

このプロセスを監視する条件は，次のどれかです。

- スクリプトの自動起動を行っている。
- 他のコンピュータから通信を行うコマンド（SetGV / GetGV / DeleteGV / NetExec）を実行する。
- 他コンピュータからプロセスビューア / トレースビューアで参照する。
- Windows Vista で Target\_File，Target\_SPAMFile または Target\_SPXFile 指定の Message コマンドを実行する。
- Windows Vista で EntryStartUp コマンド，CancelStartUp コマンドを実行する。
- Windows Vista でグローバル変数操作コマンドを実行する。

条件に一致しない場合，監視の必要はありません。



## 2.7 JP1/Script の起動と終了

---

ここでは、JP1/Script の起動と終了方法について説明します。

### 2.7.1 JP1/Script を起動する

JP1/Script の起動方法を説明します。

#### (1) マネージャの起動方法

JP1/Script を使用してスクリプトファイルを作成したり、作成したスクリプトファイルの編集をしたりする場合は、最初に JP1/Script のマネージャを起動します。

操作

1. [ スタート ] から [ プログラム ] - [ JP1/Script ] を選択する
2. JP1/Script のグループから「マネージャ」を選択する  
マネージャには、スクリプトファイルの管理や実行環境の設定、スクリプトの文法チェックなどの機能もあります。

なお、マネージャについては、「3.1.1 Script マネージャウィンドウとメニュー」を参照してください。

#### (2) エディタの起動方法

実際にスクリプトファイルの作成や編集をする場合は、エディタを起動します。

エディタには、三つの起動方法があります。

##### (a) アイコンからの起動方法

操作

1. [ スタート ] から [ プログラム ] - [ JP1/Script ] を選択する
2. JP1/Script のグループから「エディタ」アイコンを選択する

##### (b) マネージャからの起動方法

操作

1. スクリプトファイルを選択する
2. マネージャウィンドウの [ ファイル ] - [ 編集 ] を選択する  
ただし、ほかのエディタが関連付けられている場合はそのエディタが起動されます。

##### (c) ドラックアンドドロップでの起動方法

操作

## 2. JP1/Script を利用するための準備

1. エクスプローラからスクリプトファイルをドラッグして、「エディタ」アイコン、またはすでに起動しているエディタウィンドウにドロップする  
なお、エディタについては、「3.2.1 Script エディタウィンドウとメニュー」を参照してください。

### (3) トレースビューアの起動方法

スクリプトファイルの実行時に出力されるトレースは、トレースファイルに出力されます。トレースの状態を知りたいときは、トレースビューアを起動します。トレースビューアには二つの起動方法があります。

#### (a) アイコンからの起動方法

操作

1. [スタート] から [プログラム] - [JP1/Script] を選択する
2. JP1/Script のグループから「トレースビューア」を選択する

#### (b) マネージャからの起動方法

操作

1. マネージャウィンドウの [ツール] - [トレースビューアの起動] を選択する  
トレースビューアを起動すると、トレースビューアウィンドウが表示されます。ウィンドウにはトレースファイルが一覧表示されています。

なお、トレースビューアについては、「3.4.1 Script トレースビューアウィンドウとメニュー」を参照してください。

### (4) メニューエディタの起動方法

メニューエディタには、次の五つの起動方法があります。

#### (a) アイコンからの起動方法

操作

1. [スタート] から [プログラム] - [JP1/Script] を選択する
2. JP1/Script のグループから「メニューエディタ」アイコンを選択する

#### (b) マネージャからの起動方法

操作

1. スクリプトファイルを選択する
2. [ツール] - [メニューエディタの起動] を選択する、または右クリックで表示されるポップアップメニューから [メニューエディタの起動] を選択する

#### (c) エディタからの起動方法

操作

1. [編集] - [メニューエディタ]を選択する、または右クリックで表示されるポップアップメニューから [メニューエディタ]を選択する

(d) ドラッグアンドドロップでの起動方法

操作

1. エクスプローラからメニュー情報ファイルをドラッグして、「メニューエディタ」アイコン、またはすでに起動しているメニューエディタウィンドウにドロップする  
ドロップした後、メニュー情報ファイルを更新中の場合は、更新中のメニュー情報ファイルを保存するかどうかメッセージが表示されます。

(e) エクスプローラからの起動方法

操作

1. エクスプローラからメニュー情報ファイルを選択する
2. [ファイル] - [開く]を選択する、またはダブルクリックする

なお、メニューエディタについては、「3.6.1 Script メニューエディタウィンドウとメニュー」を参照してください。

## (5) 自動起動の方法

JP1/Script は、スクリプトを自動的に起動して実行できます。

自動起動には、次の二つの方法があります。

- サービス空間で起動する方法
- ログオン空間で起動する方法

自動起動する場合は、マネージャで実行環境（起動情報）を設定する必要があります。  
設定方法の詳細は、「3.1.14 スクリプトの自動起動を設定する」を参照してください。

## 2.7.2 JP1/Script を終了する

JP1/Script の終了方法を説明します。

### (1) マネージャの終了方法

操作

1. Script マネージャウィンドウで、[ファイル] - [マネージャの終了]を選択する  
マネージャ機能が終了します。

### (2) エディタの終了方法

操作

1. Script エディタウィンドウで、[ファイル] - [終了]を選択する  
エディタ機能が終了します。

### (3) トレースビューアの終了方法

#### 操作

1. Script トレースビューアウィンドウで,[ファイル] - [トレースビューアの終了] を選択する  
トレースビューア機能が終了します。

### (4) メニューエディタの終了方法

#### 操作

1. Script メニューエディタウィンドウで,[ファイル] - [終了] を選択する  
メニューエディタ機能が終了します。

### (5) 実行しているスクリプトの強制終了方法

プロセスビューアを使って強制終了できます。

### (6) JP1/AJS2 からの JP1/Script の強制終了方法

JP1/AJS2 から JP1/Script プロセスを強制終了する場合、SPHTerminate を使用しています。dwOption には SPTH\_TERM\_CHILD を指定して、JP1/Script プロセスが起動した子プロセスも終了します。

また、起動した子プロセスが JP1/Script プロセスか JP1/Script プロセス以外かで終了される範囲が変わってきます。具体的には次の 3 パターンのようになります。

パターン	呼び出し順序	
		1. の JP1/AJS2 プロセスが強制終了した場合に終了されるプロセスの範囲
パターン 1	1.JP1/AJS2 プロセス 2.JP1/Script プロセス 3.JP1/Script プロセス以外	3. のプロセスまで
パターン 2	1.JP1/AJS2 プロセス 2.JP1/Script プロセス 3.JP/Script プロセス 4.JP1/Script プロセス以外	4. のプロセスまで
パターン 3	1.JP1/AJS2 プロセス 2.JP1/Script プロセス 3.JP/Script プロセス以外 4.JP1/Script プロセス以外	3. のプロセスまで ( 4. のプロセスは終了されない )

ただし、JP1/AJS2 からの強制終了要求から 30 秒経過しても JP1/Script プロセスが終了しない場合は、Win32 API の TerminateProcess 関数を使用して JP1/Script プロセスを強制終了します。TerminateProcess 関数を使用した場合、子プロセスは終了されませんので、各パターンの「2.JP1/Script プロセス」だけが終了されることになります。

**!** 注意事項

強制終了は、スクリプトの開発時に使用することを目的としているため、実際の業務での使用はお勧めできません。

---



# 3

## JP1/Script の操作

この章では，JP1/Script のマネージャ，エディタ，簡易入力，トレースビューア，トレースファイル，メニューエディタ，プロセスビューア，実行環境ファイルコンバータの操作方法について説明します。

- 
- 3.1 マネージャの操作
  - 3.2 エディタの操作
  - 3.3 簡易入力の操作
  - 3.4 トレースビューアの操作
  - 3.5 トレースファイルの表示操作
  - 3.6 メニューエディタの操作
  - 3.7 プロセスビューアの操作
  - 3.8 実行環境ファイルコンバータの操作
-

## 3.1 マネージャの操作

---

Script マネージャウィンドウでできる操作の一覧を次に示します。

- 使用するエディタを決める
- スクリプトを作成・保存する
- 簡易入力機能を使ってスクリプトを作成する
- スクリプトを編集する
- スクリプトの文法をチェックする
- スクリプトを複写する
- スクリプトを追加する
- スクリプトを削除する
- スクリプトの名前を変更する
- スクリプトの実行環境（すべての項目）を設定する
- スクリプトの実行環境（各項目）を設定する
- スクリプトの自動起動を設定する
- スクリプトを実行する
- メニューフォームを作成する
- マネージャを終了する
- フォルダを変更する
- ツールバーの表示・非表示を切り替える
- ステータスバーの表示・非表示を切り替える
- スクリプトファイルを大きい／小さいアイコンで表示する
- スクリプトファイルをリスト形式（一覧）で表示する
- クライアントエリアを最新の情報に更新する

注

これらの操作は、特に説明していません（これらは Windows の標準の操作に準じているか、メニュー一覧で該当するメニューを選択するだけで実行できる操作です）。

### 3.1.1 Script マネージャウィンドウとメニュー

JP1/Script でスクリプトを作成する場合は、まずマネージャを起動させます。マネージャアイコンを起動すると、Script マネージャウィンドウが表示されます。マネージャでは、作成したスクリプトの保存や文法のチェックなど、スクリプトに関する操作をします。

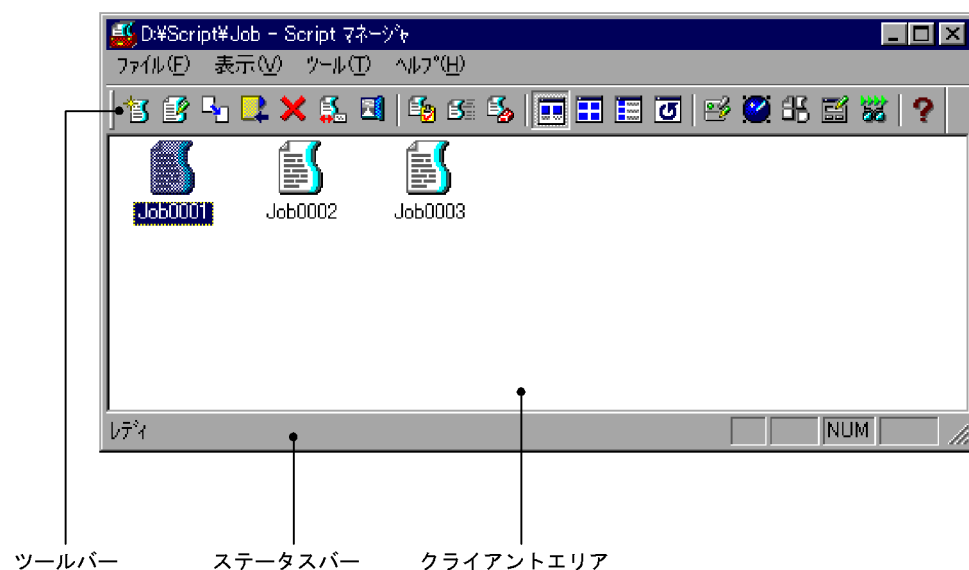
ここでは、マネージャを起動すると表示される Script マネージャウィンドウについて説明します。また、マネージャの機能をメニューごとに説明します。

#### （１）Script マネージャウィンドウ

Script マネージャウィンドウ、およびウィンドウの各部の名称を、図 3-1 に示します。



図 3-1 Script マネージャウィンドウ



## (a) ツールバー

ツールバーには、メニューバーから選択できるコマンドの中から、頻繁に使用するコマンドだけをボタンの形で表示しています。ボタンをマウスでクリックするだけで、該当するコマンドを実行できます。オプションメニューによって非表示にすることもできます。

Script マネージャウィンドウのツールバーには、次のコマンドのボタンが表示されます。

## [ 新規作成 ] ボタン

スクリプトファイルを新規作成します。

## [ 編集 ] ボタン

スクリプトファイルを編集します。

## [ 複写 ] ボタン

スクリプトファイルを複写します。

## [ 追加 ] ボタン

別のフォルダのスクリプトファイルを現在のフォルダに追加します。

## [ 削除 ] ボタン

スクリプトファイルを削除します。

## [ 名前の変更 ] ボタン

スクリプトファイル名を変更します。

## [ 終了 ] ボタン

マネージャを終了します。

[ 文法チェック ] ボタン

スクリプトファイルの文法をチェックします。

[ 実行 ] ボタン

スクリプトファイルを実行します。

[ 実行環境の設定 ] ボタン

スクリプトファイルの実行環境を設定します。

[ 大きいアイコン ] ボタン

スクリプトファイルを大きいアイコンで表示します。

[ 小さいアイコン ] ボタン

スクリプトファイルを小さいアイコンで表示します。

[ 一覧 ] ボタン

スクリプトファイルをリスト形式で表示します。

[ 最新 ] ボタン

クライアントエリアを最新の状態に更新します。

[ 自動起動の設定 ] ボタン

スクリプトファイルの自動起動を設定します。

[ トレースビューア ] ボタン

トレースビューアを起動します。

[ 簡易入力 ] ボタン

簡易入力を起動します。

[ メニューエディタ ] ボタン

メニューエディタを起動します。

[ プロセスビューア ] ボタン

プロセスビューアを起動します。

[ ヘルプ ] ボタン

JP1/Script のオンラインヘルプを表示します。

(b) ステータスバー

ステータスバーは、マネージャが現在実行している処理に関するメッセージや、処理終了後の状態に関するメッセージを表示するための領域です。

(c) クライアントエリア

クライアントエリアには、ファイルを抽象化したアイコンが表示されます。

ここに表示しきれないアイコンがある場合は、Script マネージャウィンドウにスクロールバーが表示されます。

## (2) Script マネージャウィンドウのメニュー

ここでは、Script マネージャウィンドウのメニューウィンドウのメニューバーに表示されるメニュー、および Script マネージャウィンドウで表示されるポップアップメニューについて説明します。

### (a) メニューバーのメニュー

Script マネージャウィンドウの機能を、メニューごとに説明します。Script マネージャウィンドウのメニュー一覧を表 3-1 に示します。

表 3-1 Script マネージャウィンドウのメニュー一覧

メニュー	コマンド (機能)	機能概要
[ ファイル ]	[ 新規作成 ]	スクリプトファイルを新規に作成するために、エディタを起動します。
	[ 編集 ]	既存のスクリプトファイルを編集するために、エディタを起動します。
	[ 複写 ]	現在表示されているフォルダのスクリプトファイルを別のフォルダへ複写します。
	[ 追加 ]	別のフォルダのスクリプトファイルを現在表示されているフォルダへ追加します。
	[ 削除 ]	作成されているスクリプトファイルを削除します。
	[ 名前の変更 ]	スクリプトファイルの名前を別の名前に変更します。
	[ 文法チェック ]	作成されているスクリプトファイルの文法をチェックします。
	[ 実行 ]	作成されているスクリプトファイルを実行します。
	[ 実行環境の設定 ] - [ すべての項目 ]	作成されているスクリプトファイルの実行環境を設定します。
	[ 実行環境の設定 ] - [ コマンドライン ]	作成されているスクリプトファイルの実行環境のコマンドラインだけを変更します。
	[ 実行環境の設定 ] - [ 作業フォルダ ]	作成されているスクリプトファイルの実行環境の作業フォルダだけを変更します。
	[ 実行環境の設定 ] - [ 起動日時 ]	作成されているスクリプトファイルの実行環境の起動日時だけを変更します。
	[ 実行環境の設定 ] - [ 打ち切り時間 ]	作成されているスクリプトファイルの実行環境の打ち切り時間だけを変更します。
	[ 実行環境の設定 ] - [ トレース出力先フォルダ ]	作成されているスクリプトファイルの実行環境のトレース出力先フォルダだけを変更します。
	[ 実行環境の設定 ] - [ ユーザトレース情報 ]	作成されているスクリプトファイルの実行環境のユーザトレース情報だけを変更します。
	[ フォルダの変更 ]	クライアントエリアに表示するフォルダを変更します。
	(フォルダ名)	直前に変更したフォルダ名が表示されます。
	[ マネージャの終了 ]	Script マネージャウィンドウのメニューマネージャを終了します。

### 3. JP1/Script の操作

メニュー	コマンド (機能)	機能概要
[ 表示 ]	[ ツールバー ]	ツールバーの表示・非表示を切り替えます。
	[ ステータスバー ]	ステータスバーの表示・非表示を切り替えます。
	[ 大きいアイコン ]	作成されているスクリプトファイルを大きいアイコンで表示します。
	[ 小さいアイコン ]	作成されているスクリプトファイルを小さいアイコンで表示します。
	[ 一覧 ]	作成されているスクリプトファイルをリスト形式で表示します。
	[ 最新の情報に更新 ]	クライアントエリアを最新の情報に更新します。
[ ツール ]	[ 自動起動の設定 ]	スクリプトファイルの自動起動を設定します。
	[ エディタの関連付け ]	スクリプトファイルを編集するエディタとの関連づけをします。
	[ トレースビューアの起動 ]	トレースビューアを起動します。
	[ 簡易入力 of 起動 ]	簡易入力機能を起動します。
	[ メニューエディタ of 起動 ]	メニューエディタを起動します。
	[ プロセスビューアの起動 ]	プロセスビューアを起動します。
	[ オプション ]	次の機能があります。 クライアントからのコマンド実行の認否設定 サーバに対するアクセス権の状況表示 利用するスクリプトエンジンのバージョン指定
[ ヘルプ ]	[ 目次 ]	JP1/Script オンラインヘルプの目次を表示します。
	[ キーワードで検索 ]	JP1/Script オンラインヘルプのキーワード一覧を表示します。
	[ ヘルプの使い方 ]	オンラインヘルプの使い方を表示します。
	[ バージョン情報 ]	マネージャのバージョン情報を表示します。

#### (b) ポップアップメニュー

Script マネージャウィンドウのクライアントエリアでマウスの右ボタンをクリックすると、ポップアップメニューが表示されます。Script マネージャウィンドウで表示されるポップアップメニューを表 3-2 に示します。

表 3-2 ポップアップメニュー一覧

ポップアップメニュー	機能概要
[ 新規作成 ]	スクリプトファイルを新規に作成します。
[ 編集 ]	スクリプトファイルを編集します。
[ 複写 ]	現在表示されているフォルダのスクリプトファイルを、別のフォルダへ複写します。
[ 追加 ]	別のフォルダのスクリプトファイルを、現在表示されているフォルダへ追加します。

ポップアップメニュー	機能概要
[ 削除 ]	作成されているスクリプトファイルを削除します。
[ 名前の変更 ]	スクリプトファイルの名前を別の名前に変更します。
[ 文法チェック ]	作成されているスクリプトファイルの文法をチェックします。
[ 実行 ]	作成されているスクリプトファイルを実行します。
[ 実行環境の設定 ] - [ すべての項目 ]	作成されているスクリプトファイルの実行環境を設定します。
[ 実行環境の設定 ] - [ コマンドライン ]	作成されているスクリプトファイルの実行環境のコマンドラインだけを変更します。
[ 実行環境の設定 ] - [ 作業フォルダ ]	作成されているスクリプトファイルの実行環境の作業フォルダだけを変更します。
[ 実行環境の設定 ] - [ 起動日時 ]	作成されているスクリプトファイルの実行環境の起動日時だけを変更します。
[ 実行環境の設定 ] - [ 打ち切り時間 ]	作成されているスクリプトファイルの実行環境の打ち切り時間だけを変更します。
[ 実行環境の設定 ] - [ トレース出力先フォルダ ]	作成されているスクリプトファイルの実行環境のトレース出力先フォルダだけを変更します。
[ 実行環境の設定 ] - [ ユーザトレース情報 ]	作成されているスクリプトファイルの実行環境のユーザトレース情報だけを変更します。
[ メニューエディタの起動 ]	メニューエディタを起動します。
[ プロパティ ]	作成されているスクリプトファイルの情報が表示されます。

### 3.1.2 Script マネージャウィンドウでのマウスとキーの操作

#### (1) マウス操作

Script マネージャウィンドウのクライアントエリアでのマウス操作について、対象がアイコンの場合を表 3-3 に、対象が背景の場合を表 3-4 に示します。

表 3-3 Script マネージャウィンドウでのマウス操作（対象がアイコンの場合）

操作	処理
クリック	それまでの選択を解除して、新たに対象を選択する。
ダブルクリック	スクリプトを実行する。
右ボタンシングルクリック	ポップアップメニューを表示する。

表 3-4 Script マネージャウィンドウでのマウス操作（対象が背景の場合）

操作	処理
クリック	それまでの選択を解除する。
ドラッグ	ドラッグした範囲内のアイコンを選択する。
右ボタンシングルクリック	ポップアップメニューを表示する。

## (2) キー操作

Script マネージャウィンドウのクライアントエリアでのキー操作を表 3-5 に示します。

表 3-5 Script マネージャウィンドウでのキー操作

操作	処理
Ctrl+N	新規にスクリプトファイルを作成する。
Ctrl+O	選択したスクリプトファイルを編集する。
DEL	選択したスクリプトファイルを削除する。
Enter	選択したスクリプトファイルを実行する。
Shift+Enter	選択したスクリプトファイルを文法チェックする。
Shift+Ctrl+Enter	選択したスクリプトファイルの実行環境を設定する。

### 3.1.3 使用するエディタを決める

JP1/Script には、スクリプトを作成するための専用のエディタがあります。デフォルトの設定ではこのエディタを使用するように設定されています。

ただし、特に別のエディタを使用してスクリプトを作成したい場合は、ここで説明する操作を行う必要があります。

操作

1. [ ツール ] - [ エディタの関連付け ] メニューを選択する。  
[ エディタの関連付け ] ダイアログが表示されます。



ダイアログの設定方法については、「4.1.21 エディタの関連付けダイアログ」を参照してください。

2. 専用のエディタ以外のほかのエディタを使用する場合は、[ 別のエディタを使用する ] ボタンをチェックし、[ 選択 ] ボタンをクリックして関連づけをするエディタを指定する。必要の場合は、エディタの作業フォルダを指定する。  
ここではメモ帳を関連づけた例を示します。



ほかのエディタを使用するのをやめて、専用のエディタを使用する場合は、[ 別のエディタを使用する ] ボタンのチェックを外します。

3. [ OK ] ボタンをクリックする。

エディタの関連づけが設定されます。

Windows 標準のワードパッドは使用できません。その他のプログラム（テキスト形式で保存できるプログラム）を設定してください。

操作 2. で作業フォルダの指定を省略すると、エディタのフォルダが仮定されます。

操作 3. で [ キャンセル ] ボタンをクリックすると、エディタの関連づけがキャンセルされます。

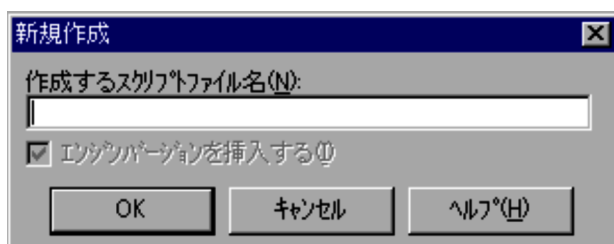
### 3.1.4 スクリプトを作成・保存する

スクリプトファイルを新規に作成します。

この操作を行う前に、エディタとの関連づけを設定しておく必要があります。設定の方法については、「3.1.3 使用するエディタを決める」を参照してください。

操作

1. [ ファイル ] - [ 新規作成 ] メニューを選択する。  
[ 新規作成 ] ダイアログが表示されます。



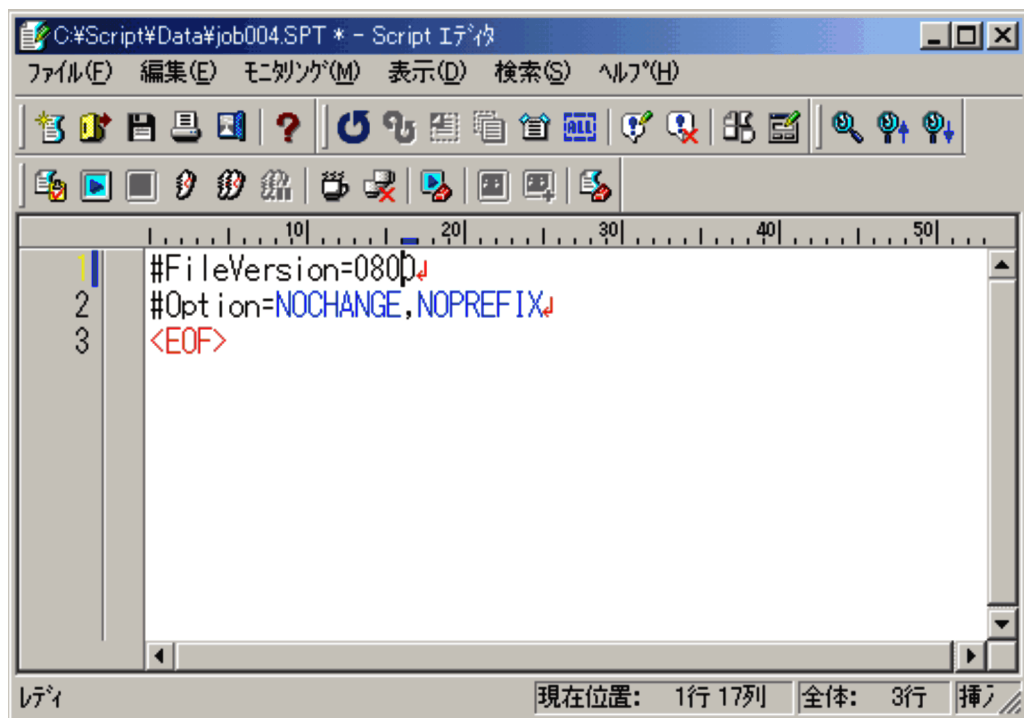
ダイアログの設定方法については、「4.1.1 新規作成ダイアログ」を参照してください。

2. 作成するスクリプトファイルの名前（拡張子は不要）を入力して、[ OK ] ボタンをク

### 3. JP1/Script の操作

リックする。

関連づけされているエディタが起動します。



3. スクリプトを作成して上書き保存し、エディタを終了する。

クライアントエリアに、作成したスクリプトファイルのアイコンが表示されます。



使用するエディタが専用のエディタではなくほかのエディタの場合、作成したスクリプトファイルをクライアントエリアに反映させるためには、[ 表示 ] - [ 最新の情報に更新 ] メニューを実行してください。

Windows 標準のワードパッドは使用できません。エディタとしてワードパッドが設定されている場合は、「3.1.3 使用するエディタを決める」を参照して設定をやり直し



てください。

操作 2. でスクリプトファイルの名前を入力するときに、拡張子は指定しないでください。プログラムが自動的に拡張子（.SPT）を設定します。

マネージャとエディタは独立に動きます。エディタを起動した後、マネージャを終了しても、エディタは終了しません。

### 3.1.5 簡易入力機能を使ってスクリプトを作成する

スクリプトを記述する際の、コマンドやステートメントの入力作業を、簡易入力機能を使って行います。

この機能を使うことで、コマンドの知識がなくても手軽に間違いなく入力できます。

記述したコマンドやステートメントの内容を、クリップボードを経由して使用しているエディタに貼り付けます。

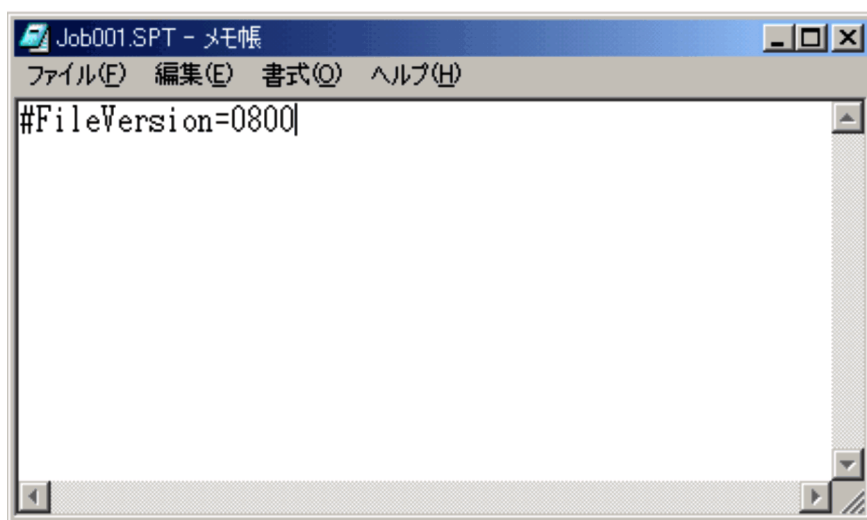
#### 操作

操作手順は、使用するエディタに専用のエディタが関連づけられている場合と、ほかのエディタが関連づけられている場合とで異なります。

使用するエディタに専用のエディタが関連づけられている場合  
「3.2.4 簡易入力機能を使う」を参照してください。

使用するエディタにほかのエディタが関連づけられている場合

1. [ ファイル ] - [ 新規作成 ] メニュー，またはスクリプトファイルを選択し [ ファイル ] - [ 編集 ] メニューを選択して，エディタを起動する。  
関連づけられているエディタが起動します（メモ帳が関連づけられている例）。



2. マネージャの [ ツール ] - [ 簡易入力の起動 ] メニューを選択するか，JP1/Script グ

### 3. JP1/Script の操作

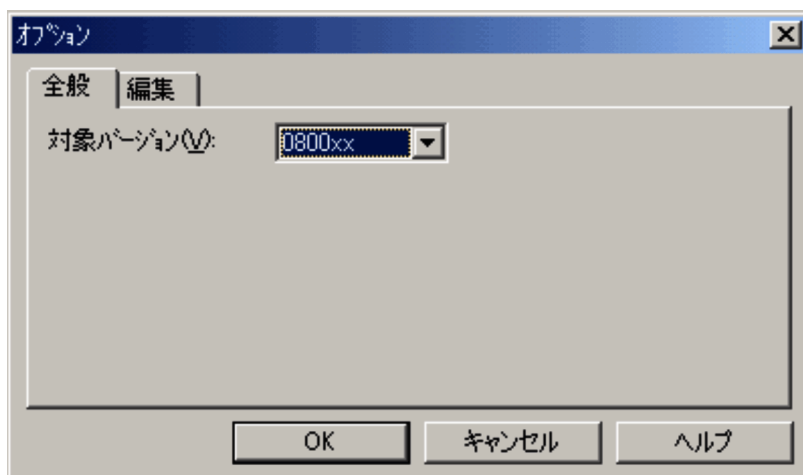
ループの簡易入力アイコンをダブルクリックして、簡易入力を起動する。

[Script 簡易入力] ダイアログが表示されます。



ダイアログの設定方法については「3.3 簡易入力の操作」を参照してください。

3. [オプション] ボタンをクリックして、[オプション (全般)] ダイアログでスクリプトファイルの作成バージョンを設定する。



ダイアログの設定方法については「3.3.3 JP1/Script の対象バージョンを指定する」を参照してください。

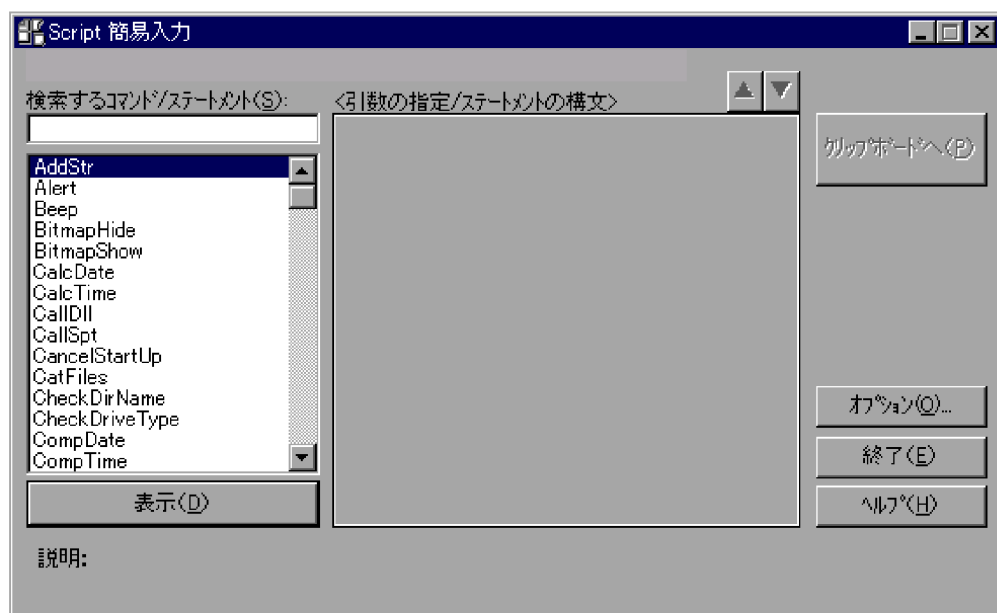
4. [編集] タブをクリックして、[オプション (編集)] ダイアログでクリップボードへの出力形式を設定する。



ダイアログの設定方法については「3.3.4 1行の最大文字数やインデントの桁数を指定する」を参照してください。

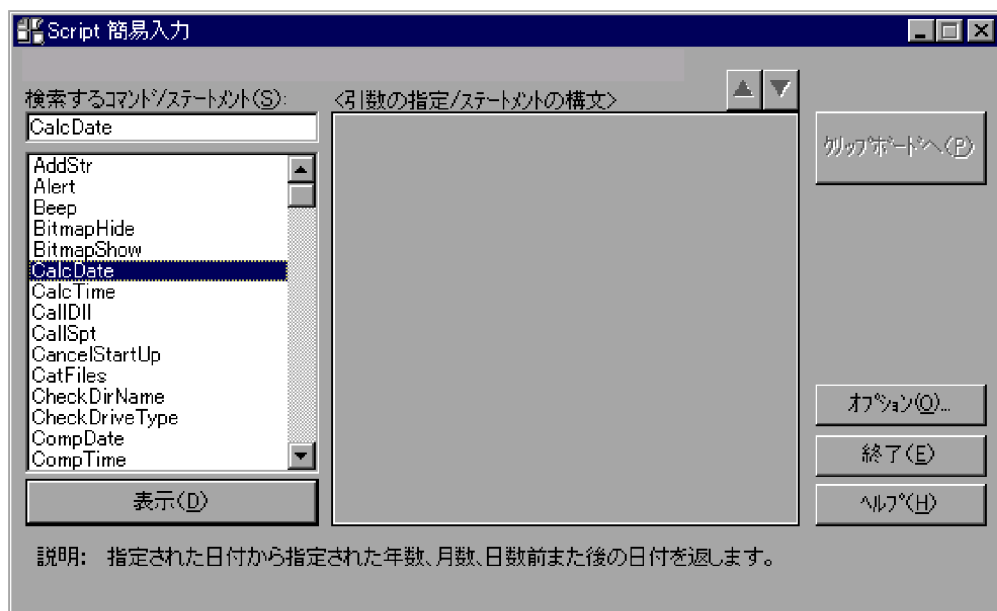
5. [ OK ] ボタンをクリックする。

スクリプトファイルの作成バージョン、およびクリップボードへの出力形式が設定され、[ Script 簡易入力 ] ダイアログに戻ります。



6. 使用するコマンド、またはステートメントを検索し、選択する。  
該当するコマンド、またはステートメントにフォーカスが移動します。

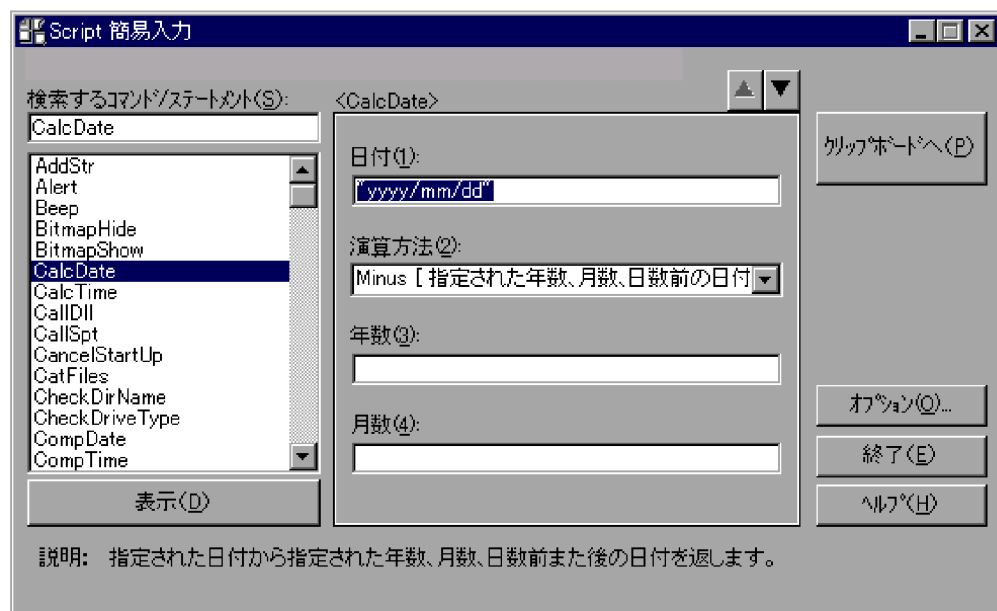
### 3. JP1/Script の操作

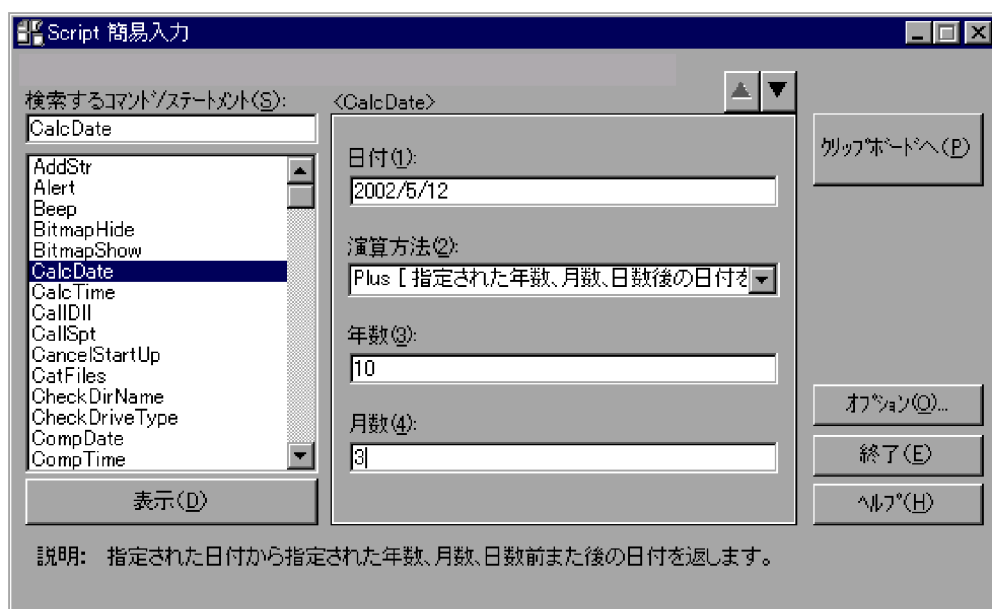


7. [ 表示 ] ボタンをクリックするか、コマンド、またはステートメントをダブルクリックする。

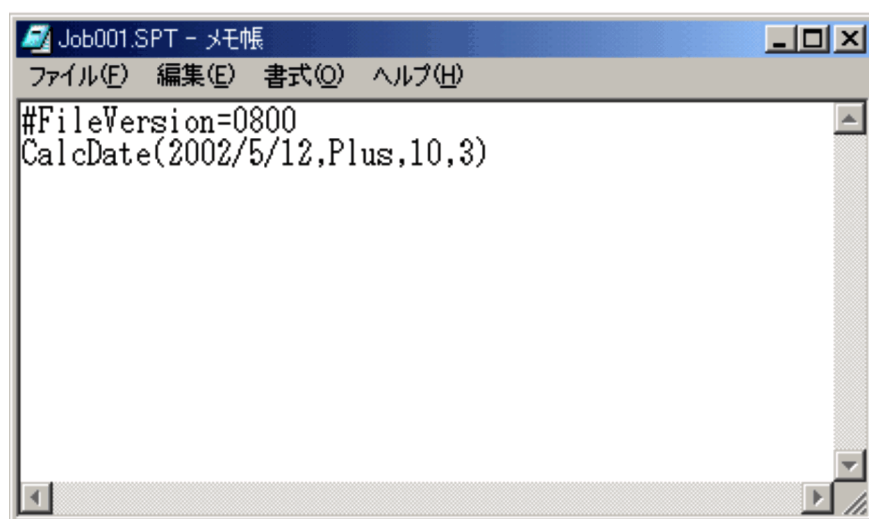
「引数の指定 / ステートメントの構文」に、選択したコマンドの引数を入力する項目が表示されます。

ステートメントの場合は、ステートメントの構文が表示されます。





9. [クリップボードへ] ボタンをクリックする。  
「引数の指定 / ステートメントの構文」の内容がクリップボードに貼り付けられます。
10. クリップボードからエディタに貼り付ける。



- ・「検索するコマンド / ステートメント」の一覧 (リストボックス) で、コマンド名、またはステートメント名を選択し、マウスの右ボタンをクリックすると、選択されたコマンド、またはステートメントのヘルプが表示されます。
- ・操作 7. で引数がないコマンドの場合は、「引数の指定 / ステートメントの構文」に「このコマンドに引数はありません。」というメッセージが表示されます。

### 3.1.6 スクリプトを編集する

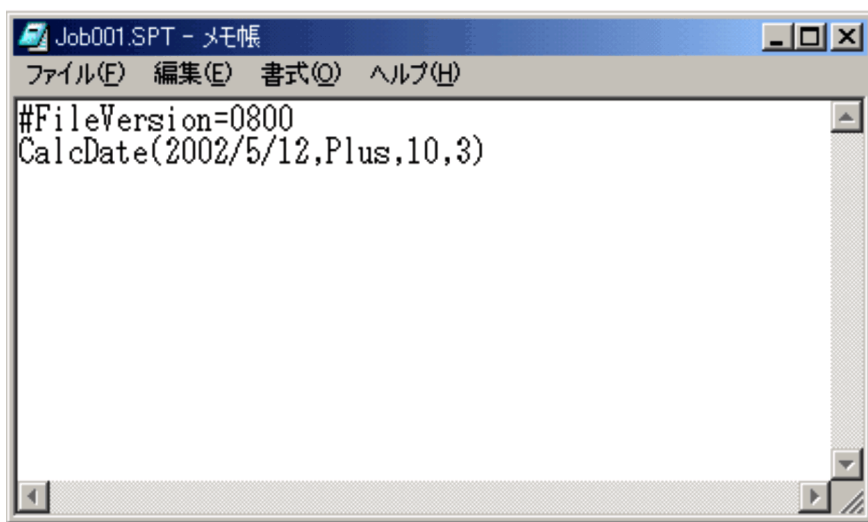
作成したスクリプトファイルを編集します。

操作

1. クライアントエリアから、編集するスクリプトファイルのアイコンを一つ選択する。



2. [ ファイル ] - [ 編集 ] メニューを選択する。  
関連づけされているエディタが起動します。



3. スクリプトを編集して上書き保存し、エディタを終了する。

Windows 標準のワードパッドは使用できません。エディタとしてワードパッドが設定されている場合は、「3.1.3 使用するエディタを決める」を参照して設定をやり直してください。

スクリプトを編集するエディタとしてデフォルトでは、専用のエディタが関連づけられています。

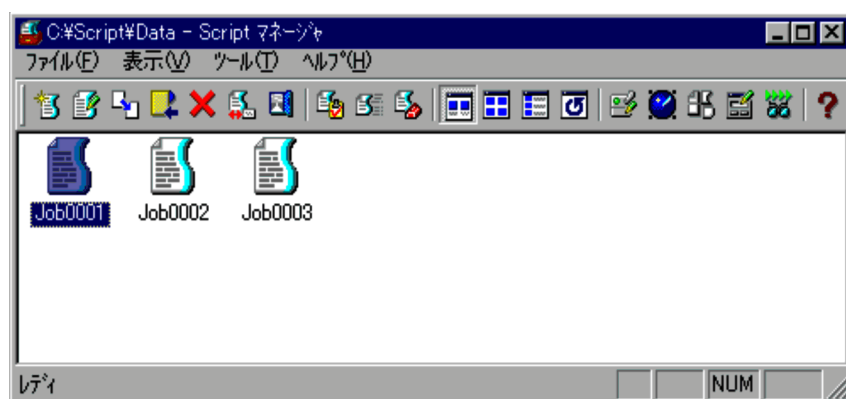
### 3.1.7 スクリプトの文法をチェックする

作成したスクリプトの文法をチェックします。

文法チェックとは、字句解析や構文解析だけを行って、コマンドは実行しないモードのことです。

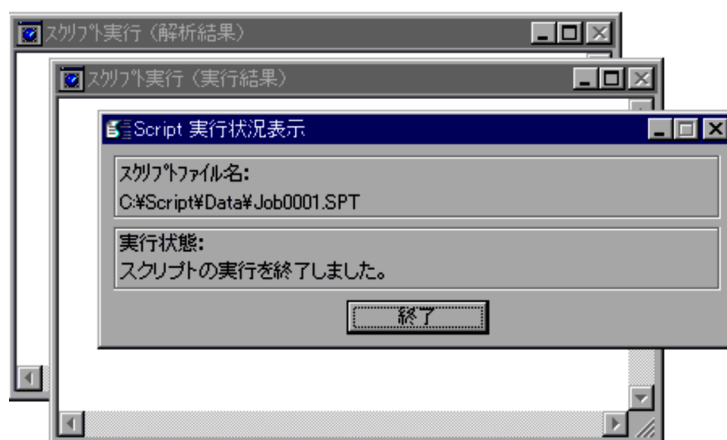
操作

1. クライアントエリアから、文法をチェックするスクリプトファイルのアイコンを一つ選択する。



2. [ファイル] - [文法チェック] メニューを選択する。

文法チェックが実行され、実行状況、実行トレース、解析トレースを表示するウィンドウが表示されます。文法チェックの結果は、解析トレースとして表示されます。文法チェックが終了すると、実行状況を表示するウィンドウの[終了]ボタンが選択可能になります。



3. 解析トレースの内容を確認する。

このとき、[ファイル] - [編集] メニューを選択してエディタを起動し、解析結果を見ながらスクリプトを編集することもできます。

4. 実行状況を表示しているウィンドウの [ 終了 ] ボタンをクリックする。  
三つのウィンドウが閉じます。

文法チェックを実行すると、解析トレースファイルが作成されます。解析トレースファイルを印刷したい場合は、トレースビューアを起動し、印刷したいファイルを選択してから、[ ファイル ] - [ 印刷 ] メニューを選択してください。詳細は、「3.4.10 トレースファイルを印刷する」を参照してください。

### 3.1.8 スクリプトを複写する

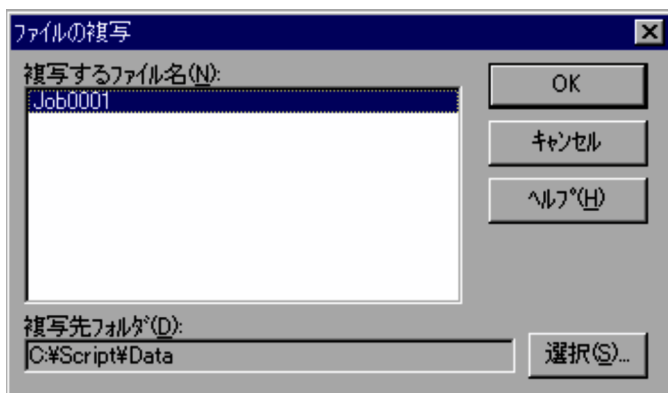
現在表示されているフォルダのスクリプトファイルを、他のフォルダへ複写します。

操作

1. クライアントエリアから、複写するスクリプトファイルのアイコンを一つ、または複数選択する。



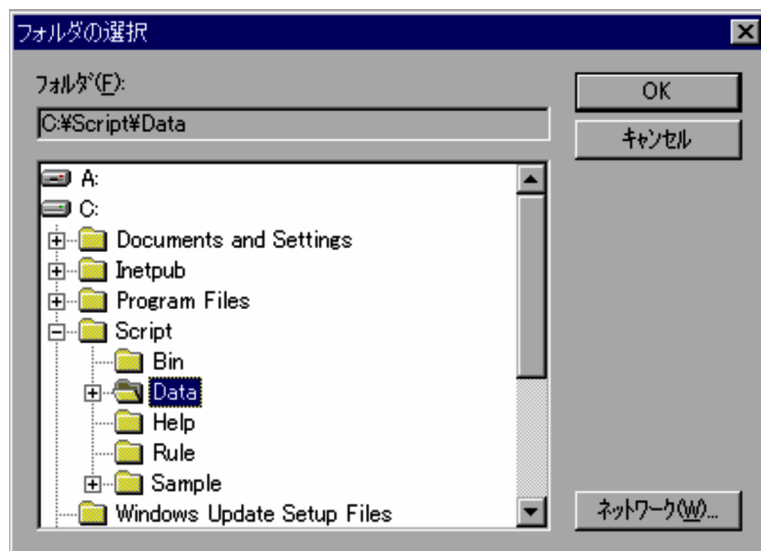
2. [ ファイル ] - [ 複写 ] メニューを選択する。  
[ ファイルの複写 ] ダイアログが表示されます。



ダイアログの設定方法については「4.1.2 ファイルの複写ダイアログ」を参照してください。



3. [ 選択 ] ボタンをクリックして,[ フォルダの選択 ] ダイアログから複写先フォルダを選択する。



ダイアログの設定方法については「4.1.29 フォルダの選択ダイアログ」を参照してください。

4. [ OK ] ボタンをクリックする。

複写が実行されます。

[ ファイルの複写 ] ダイアログを表示してから、一部のファイルの複写をとりやめたい場合は,[ ファイルの複写 ] ダイアログで、とりやめたいファイルをクリックしてください。ファイルの選択状態を解除し、複写の対象から外すことができます。

操作 4. で [ キャンセル ] ボタンをクリックすると、ファイルの複写を中断してダイアログを閉じます。

複写先フォルダに同名のファイルがある場合は,[ ファイルの上書き確認 ] ダイアログが表示されます。

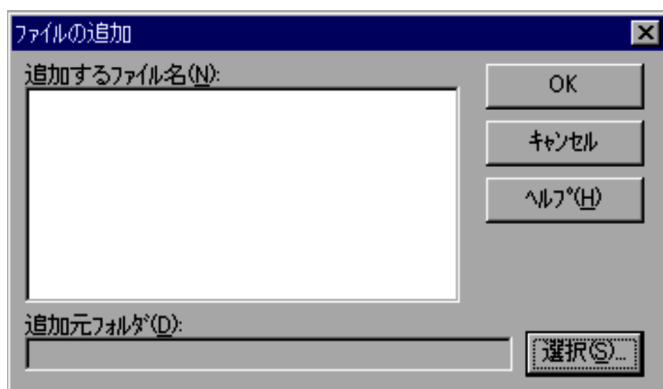
### 3.1.9 スクリプトを追加する

他のフォルダにあるスクリプトファイルを現在表示されているフォルダに追加します。

操作

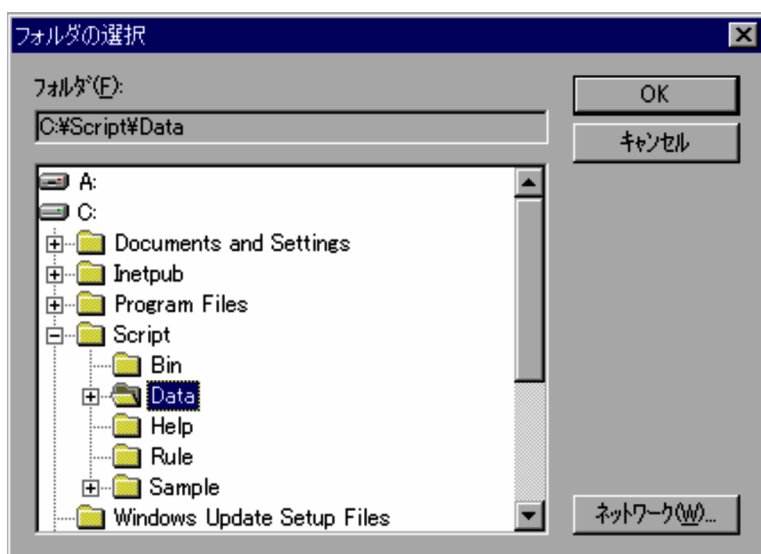
1. [ ファイル ] - [ 追加 ] メニューを選択する。  
[ ファイルの追加 ] ダイアログが表示されます。

### 3. JP1/Script の操作



ダイアログの設定方法については「4.1.3 ファイルの追加ダイアログ」を参照してください。

2. [ 選択 ] ボタンをクリックして、[ フォルダの選択 ] ダイアログから追加元フォルダ（追加したいファイルがあるフォルダ）を選択する。



ダイアログの設定方法については「4.1.29 フォルダの選択ダイアログ」を参照してください。

3. 追加したいファイルを一つ、または複数選択する。

4. [ OK ] ボタンをクリックする。  
ファイルの追加が実行されます。

操作 4. で [ キャンセル ] ボタンをクリックすると、ファイルの追加を中断してダイアログを閉じます。

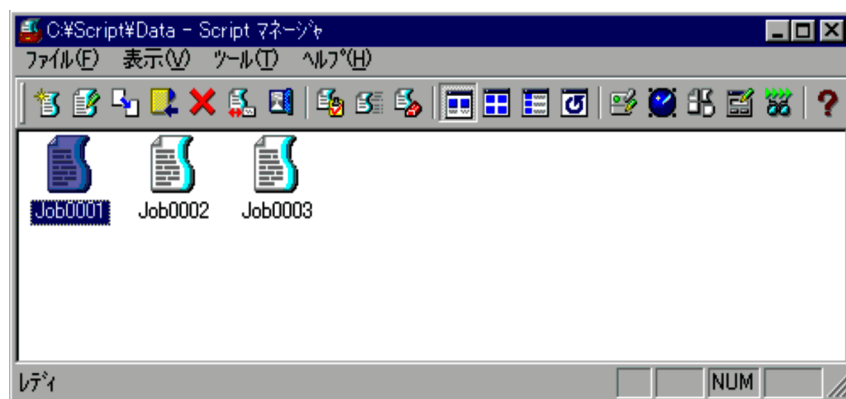
追加先フォルダに同名のファイルがある場合は、[ ファイルの上書き確認 ] ダイアログが表示されます。

### 3.1.10 スクリプトを削除する

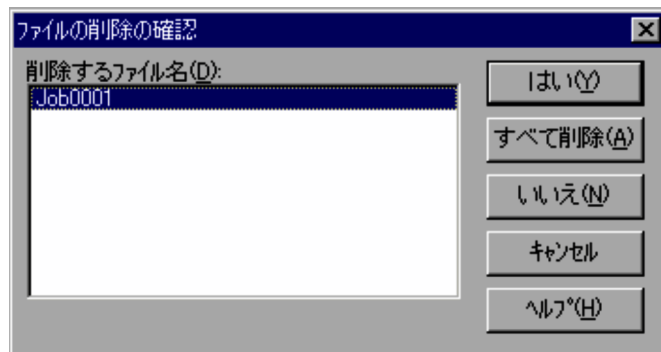
作成されているスクリプトファイルを削除します。

操作

1. クライアントエリアから、削除するスクリプトファイルのアイコンを一つ、または複数選択する。



2. [ ファイル ] - [ 削除 ] メニューを選択する。  
[ ファイルの削除の確認 ] ダイアログが表示されます。



ダイアログの設定方法については「4.1.5 ファイルの削除の確認ダイアログ」を参照してください。

3. [ はい ] ボタンをクリックする。  
スクリプトファイルが一つ削除され、次に選択されているファイルに処理が移ります。  
必要な回数だけ、操作を繰り返します。選択されているすべてのファイルが削除されると、ダイアログが閉じます。

操作 3. で [ すべて削除 ] ボタンをクリックすると、選択されているすべてのファイルが削除されます。

操作 3. で [ キャンセル ] ボタンをクリックすると、ファイルの削除を中断してダイア

ログを閉じます。

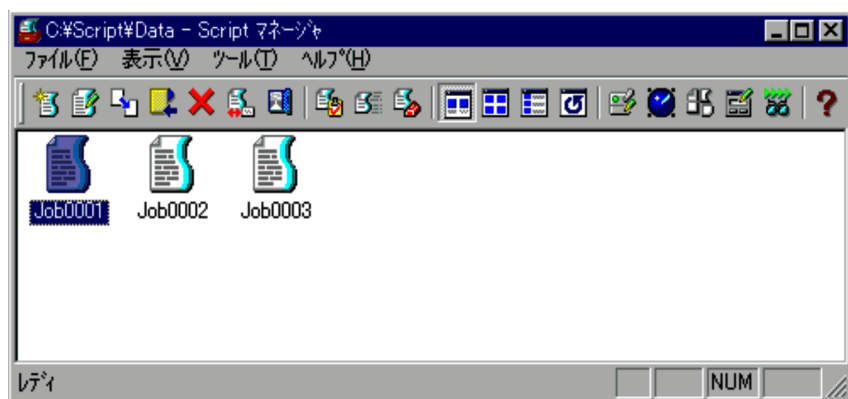
[ファイルの削除の確認] ダイアログを表示してから、一部のファイルの削除をとりやめたい場合は、[ファイルの削除の確認] ダイアログで、とりやめたいファイルをクリックしてください。ファイルの選択状態を解除し、削除の対象から外すことができます。

### 3.1.11 スクリプトの名前を変更する

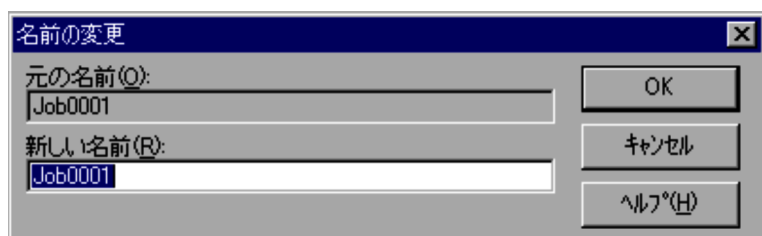
選択したスクリプトファイルの名前を別の名前に変更します。

操作

1. クライアントエリアから、名前を変更するスクリプトファイルのアイコンを一つ選択する。

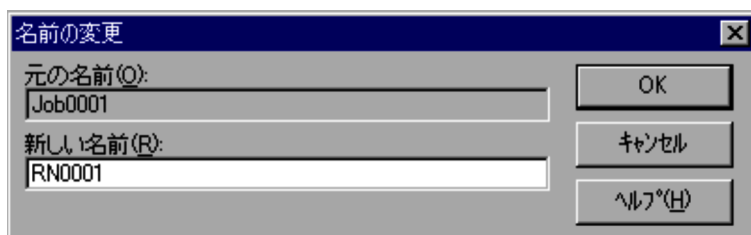


2. [ファイル] - [名前の変更] メニューを選択する。  
[名前の変更] ダイアログが表示されます。



ダイアログの設定方法については「4.1.6 名前の変更ダイアログ」を参照してください。

3. 新しい名前を入力する。



4. [ OK ] ボタンをクリックする。  
 名前の変更が実行されて、ダイアログが閉じます。



操作 4. で [ キャンセル ] ボタンをクリックすると、名前の変更を中断してダイアログを閉じます。

### 3.1.12 スクリプトの実行環境（すべての項目）を設定する

作成されているスクリプトファイルの実行環境を設定します。このほかに、エディタから操作する方法もあります。

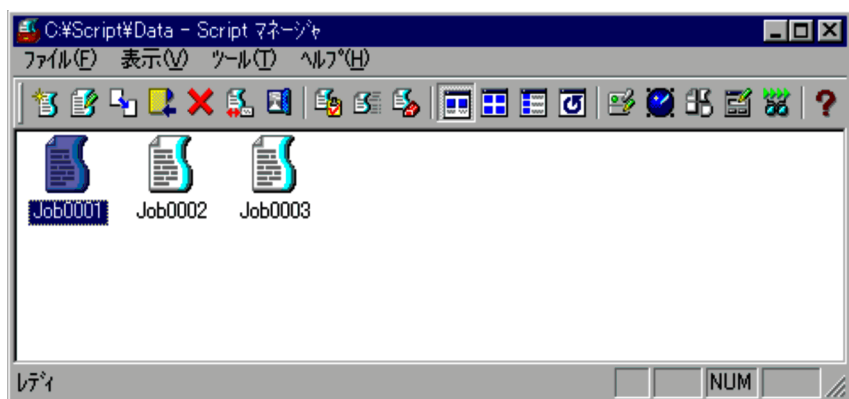
エディタから操作する方法については「3.2.13 スクリプトの実行環境を設定する」を参照してください。

操作

Script マネージャウィンドウから操作する方法

1. クライアントエリアから、実行環境を設定するスクリプトファイルのアイコンを一つ、または複数選択する。

### 3. JP1/Script の操作



2. [ ファイル ] - [ 実行環境の設定 ] - [ すべての項目 ] メニューを選択する。  
[ 実行環境 ( 起動情報 ) の設定 ] ダイアログが表示されます。



ダイアログの設定方法については、「4.1.8 実行環境の設定 ( 起動情報 ) ダイアログ」を参照してください。

3. 必要な起動情報を設定する。  
実行時に位置変数が必要な場合、および解析トレース、実行トレースの出力形式などを設定する場合は、コマンドラインでの指定が必要です。

コマンドラインの指定方法については、「6.2 コマンドラインに関する規則」を参照してください。

4. [ 終了情報 ] タブをクリックする。

[ 実行環境の設定 ( 終了情報 ) ] ダイアログが表示されます。



ダイアログの設定方法については、「4.1.9 実行環境の設定 ( 終了情報 ) ダイアログ」を参照してください。

5. 打ち切り時間を設定する。

打ち切り時間による強制終了をしない場合は、打ち切り時間が「0」になっていることを確認してください。

6. [ トレース情報 ] タブをクリックする。

[ 実行環境の設定 ( トレース情報 ) ] ダイアログが表示されます。



ダイアログの設定方法については、「4.1.10 実行環境の設定（トレース情報）ダイアログ」を参照してください。

7. 必要なトレース情報を設定する。  
トレースファイルの出力行数などを特に指定する必要がない場合は、このダイアログでの操作は必要ありません。
8. [ OK ] ボタンをクリックする。  
実行環境が設定され、ダイアログが閉じます。

#### 補足

Windows Vista では、JP1/Script サービスが実行中以外の状態でスクリプトの実行環境を設定しようとすると、次のエラーダイアログを表示します。  
「JP1/Script サービスの状態が実行中以外のため、実行できません。JP1/Script サービスを開始した後、再度実行してください。」

### 3.1.13 スクリプトの実行環境（各項目）を設定する

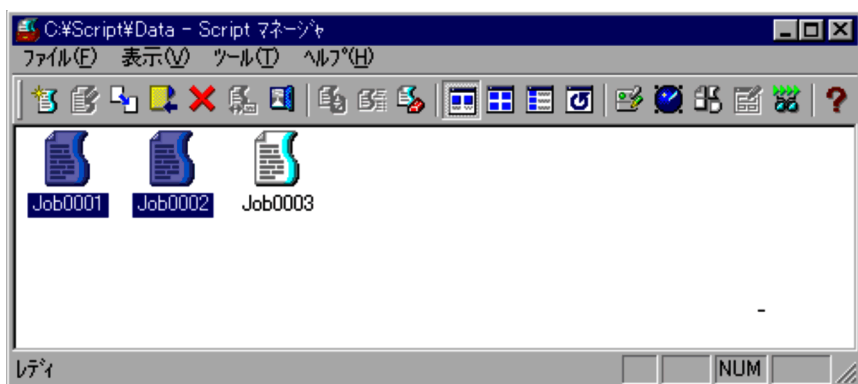
作成されているスクリプトファイルの実行環境を項目ごとに設定します。

#### 操作

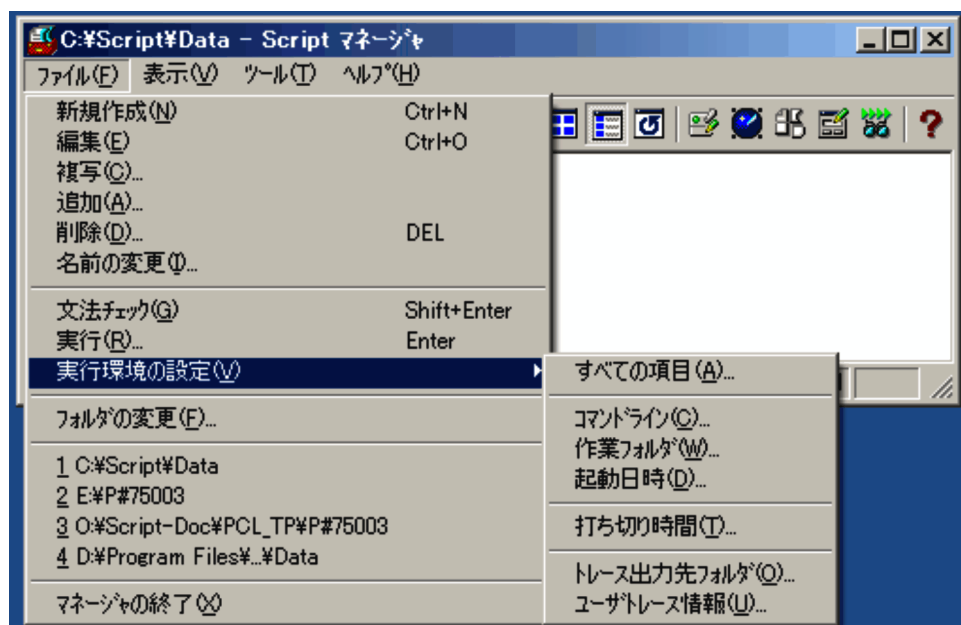
1. クライアントエリアから、実行環境を設定するスクリプトファイルのアイコンを一



つ、または複数選択する。



2. [ファイル] - [実行環境の設定] - [コマンドライン] メニューなどの各項目を選択する。



[ファイル] - [実行環境の設定] から次のメニューを選択します。

- コマンドライン
  - 作業フォルダ
  - 起動日時
  - 打ち切り時間
  - トレース出力先フォルダ
  - ユーザトレース情報
- [コマンドライン] を選択した場合  
 [実行環境 (コマンドライン) の設定] ダイアログが表示されます。



ダイアログの設定方法については「4.1.11 実行環境（コマンドライン）の設定ダイアログ」を参照してください。

- [ 作業フォルダ ] を選択した場合  
[ 実行環境（作業フォルダ）の設定 ] ダイアログが表示されます。



ダイアログの設定方法については「4.1.13 実行環境（作業フォルダ）の設定ダイアログ」を参照してください。

- [ 起動日時 ] を選択した場合  
[ 実行環境（起動日時）の設定 ] ダイアログが表示されます。



ダイアログの設定方法については「4.1.14 実行環境（起動日時）の設定ダイアログ」を参照してください。

Windows Vista では、JP1/Script サービスが実行中以外の状態でスクリプトの実行環境（起動日時）を設定しようとすると、次のエラーダイアログを表示します。

「JP1/Script サービスの状態が実行中以外のため、実行できません。JP1/Script サービスを開始した後、再度実行してください。」

- [ 打ち切り時間 ] を選択した場合  
[ 実行環境（打ち切り時間）の設定 ] ダイアログが表示されます。



ダイアログの設定方法については「4.1.15 実行環境（打ち切り時間）の設定ダイアログ」を参照してください。

- [ トレース出力先フォルダ ] を選択した場合  
[ 実行環境（トレース出力先フォルダ）の設定 ] ダイアログが表示されます。



ダイアログの設定方法については「4.1.16 実行環境（トレース出力先フォルダ）の設定ダイアログ」を参照してください。

- [ ユーザトレース情報 ] を選択した場合  
[ 実行環境（ユーザトレース情報）の設定 ] ダイアログが表示されます。



ダイアログの設定方法については「4.1.17 実行環境（ユーザトレース情報）の設定ダイアログ」を参照してください。

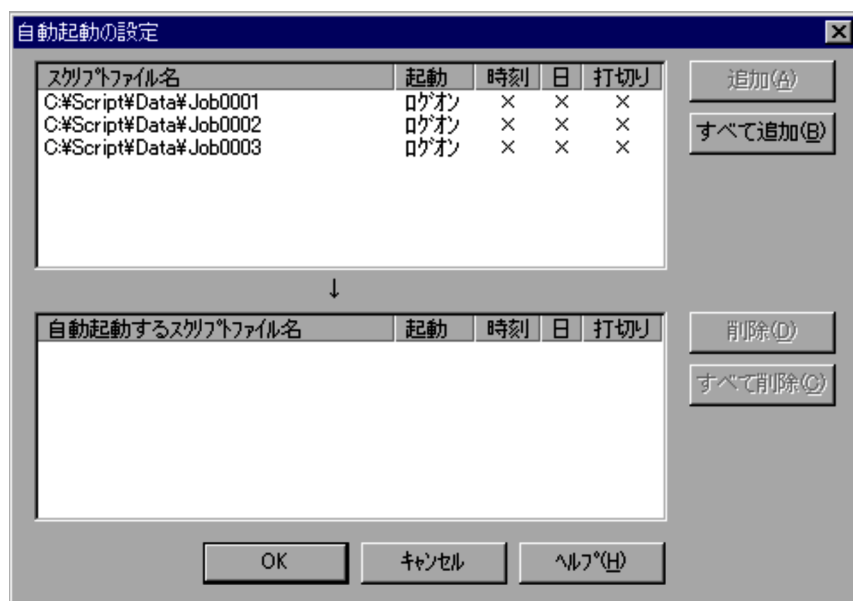
1. 必要な情報を設定する。
2. [ OK ] ボタンをクリックする。  
項目ごとの実行環境が設定され、ダイアログが閉じます。

### 3.1.14 スクリプトの自動起動を設定する

ランチャ、またはサービスによって自動起動されるスクリプトファイルの名前を設定します。

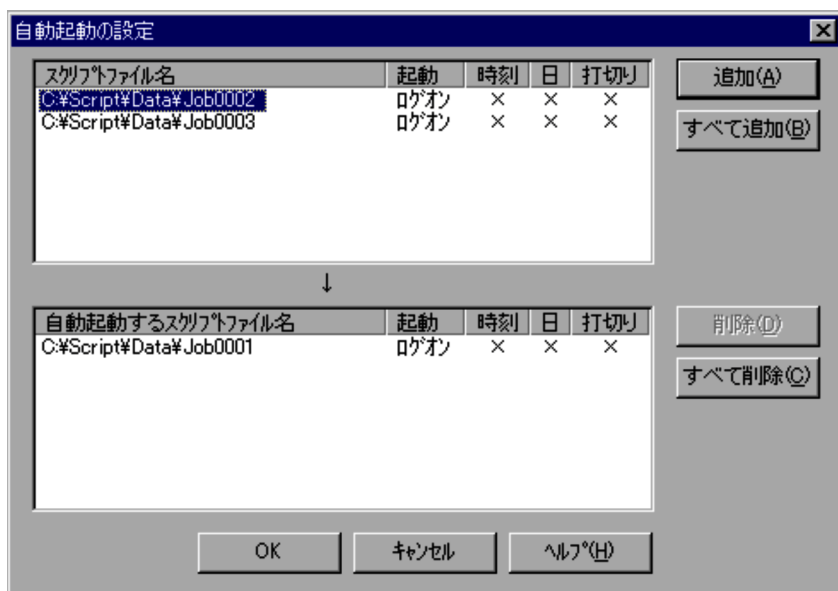
操作

1. [ ツール ] - [ 自動起動の設定 ] メニューを選択する。  
[ 自動起動の設定 ] ダイアログが表示されます。



ダイアログの設定方法については、「4.1.20 自動起動の設定ダイアログ」を参照してください。

2. スクリプトファイル名を選択し [ 追加 ] ボタンをクリックするか、または [ すべて追加 ] ボタンをクリックする。  
選択したスクリプトファイル名が、[ 自動起動するスクリプトファイル名 ] の一覧 ( リストボックス ) に追加されます。



3. [ OK ] ボタンをクリックする。

[ 自動起動するスクリプトファイル名 ] の一覧 ( リストボックス ) の内容が自動起動情報ファイルとして出力されます。

- [ 自動起動するスクリプトファイル名 ] の一覧 ( リストボックス ) に不要なスクリプトファイルがある場合は、それを一つ、または複数選択して [ 削除 ], または [ すべて削除 ] ボタンをクリックします。削除したスクリプトファイル名は、[ スクリプトファイル名 ] の一覧 ( リストボックス ) に追加されます。
- 操作 3. で [ キャンセル ] ボタンをクリックすると、自動起動に設定しないでダイアログを閉じます。設定は初期状態に戻ります。

#### 補足

Windows Vista では、JP1/Script サービスが実行中以外の状態でスクリプトの自動起動を設定しようとすると、次のエラーダイアログを表示します。

「JP1/Script サービスの状態が実行中以外のため、実行できません。JP1/Script サービスを開始した後、再度実行してください。」

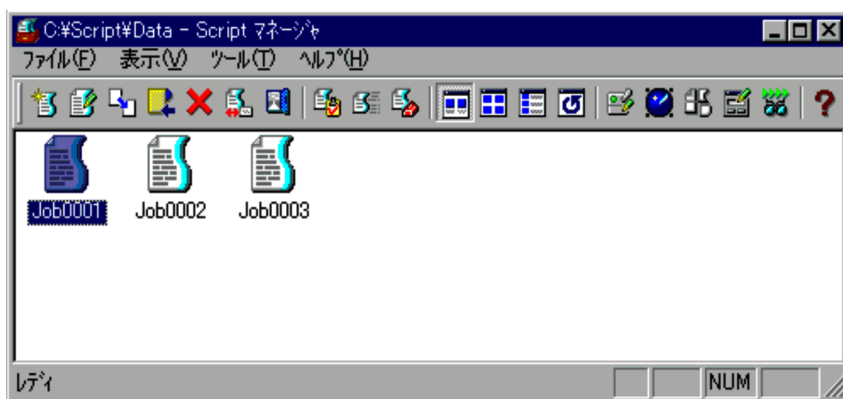
### 3.1.15 スクリプトを実行する

作成されているスクリプトファイルを実行します。

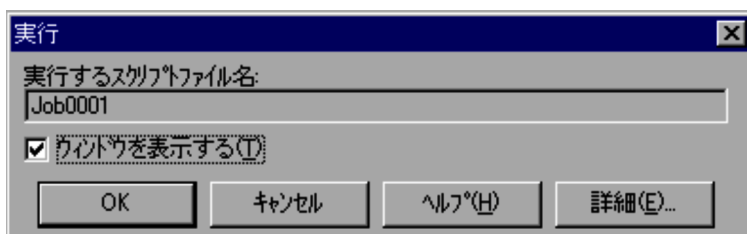
Script マネージャウィンドウからの実行は、通常は実行トレースなどを確認するために行います。

#### 操作

1. クライアントエリアで、実行するスクリプトファイルのアイコンを一つ選択する。



2. [ ファイル ] - [ 実行 ] メニューを選択する。またはアイコンをダブルクリックする。  
[ 実行 ] ダイアログが表示されます。



ダイアログの設定方法については、「4.1.7 実行ダイアログ」を参照してください。

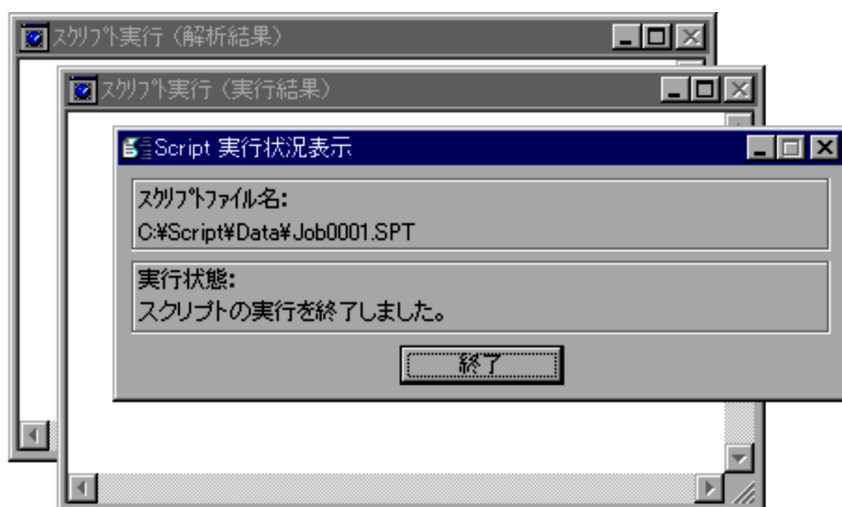
3. 実行環境の設定が必要な場合は、[ 詳細 ] ボタンをクリックし、[ 実行の環境設定 (起動情報) ] ダイアログでコマンドラインなどを設定する。



ダイアログの設定方法については、「4.1.8 実行環境の設定（起動情報）ダイアログ」を参照してください。

4. [実行] ダイアログで [OK] ボタンをクリックする。

スクリプトが実行され、実行状況、実行トレース、解析トレースを表示するウィンドウが表示されます。



スクリプト実行の結果は、実行トレースとして表示されます。



実行が終了すると、実行状況を表示するウィンドウの [ 終了 ] ボタンが選択できるようになります。

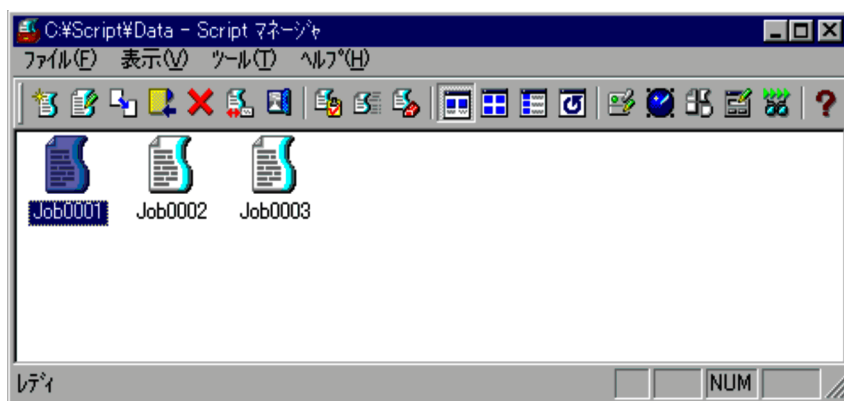
5. 実行トレースの内容を確認する。
6. 実行状況を表示しているウィンドウの [ 終了 ] ボタンをクリックする。  
三つのウィンドウが閉じます。
  - Script マネージャウィンドウでスクリプトを実行すると、実行トレースファイルが作成されます。実行トレースファイルを印刷したい場合は、トレースビューアを起動し、印刷したいファイルを選択してから、[ ファイル ] - [ 印刷 ] メニューを選択してください。詳細は、「3.4.10 トレースファイルを印刷する」参照してください。
  - 操作 4. で [ キャンセル ] ボタンをクリックすると、スクリプトを実行しないでダイアログを閉じます。
  - 実行状況、実行トレース、解析トレースを表示するウィンドウを表示させたい場合は、[ ウィンドウを表示する ] をチェックします。デフォルトではチェックされています。

### 3.1.16 メニューフォームを作成する

メニューフォームを作成します。

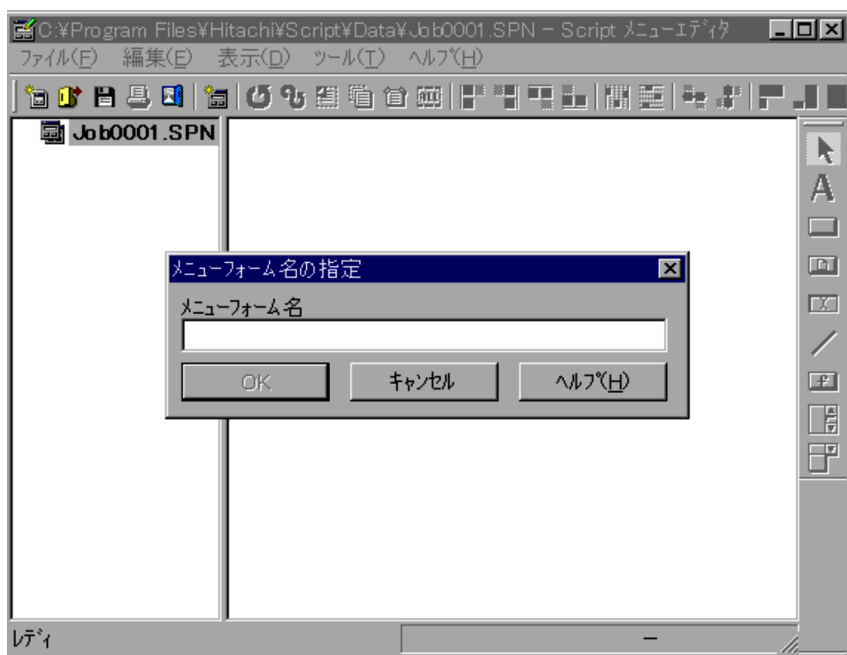
操作

1. クライアントエリアで、編集、または実行するスクリプトファイルのアイコンを一つ選択する。



2. [ ツール ] - [ メニューエディタの起動 ] メニューを選択する。  
Script メニューエディタウィンドウが表示されます。  
メニューフォームが作成されていない場合は、Script メニューエディタウィンドウの手前に、[ メニューフォーム名の指定 ] ダイアログが表示されます。なお、画面左下にはコマンドプロパティダイアログボックスが表示されます。

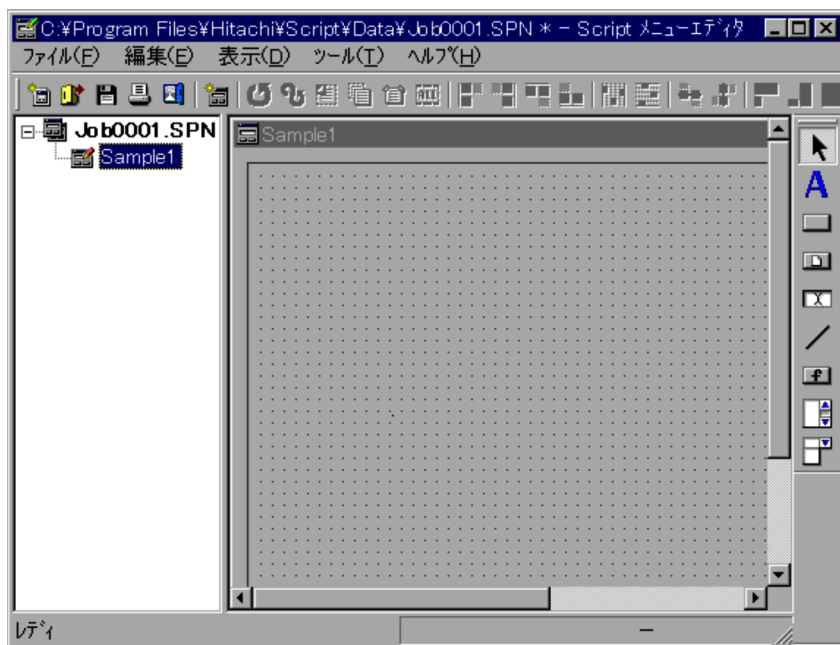
### 3. JP1/Script の操作



メニューフォームが作成されている場合は、操作 4 . に進んでください。

#### 3. メニューフォーム名を入力する。

フォームビューに新しいメニューフォームが表示されます。操作 5 . に進んでください。



4. [ 編集 ] - [ メニューフォームの新規作成 ] メニューを選択する。  
フォームビューに新しいメニューフォームが表示されます。
5. メニューフォームを作成する。  
作成方法については、「3.6.3 新しいメニューフォームを作成する」を参照してください。
6. メニューエディタを終了する。
  - マネージャとメニューエディタは独立に動きます。メニューエディタを起動した後、マネージャを終了しても、メニューエディタは終了しません。

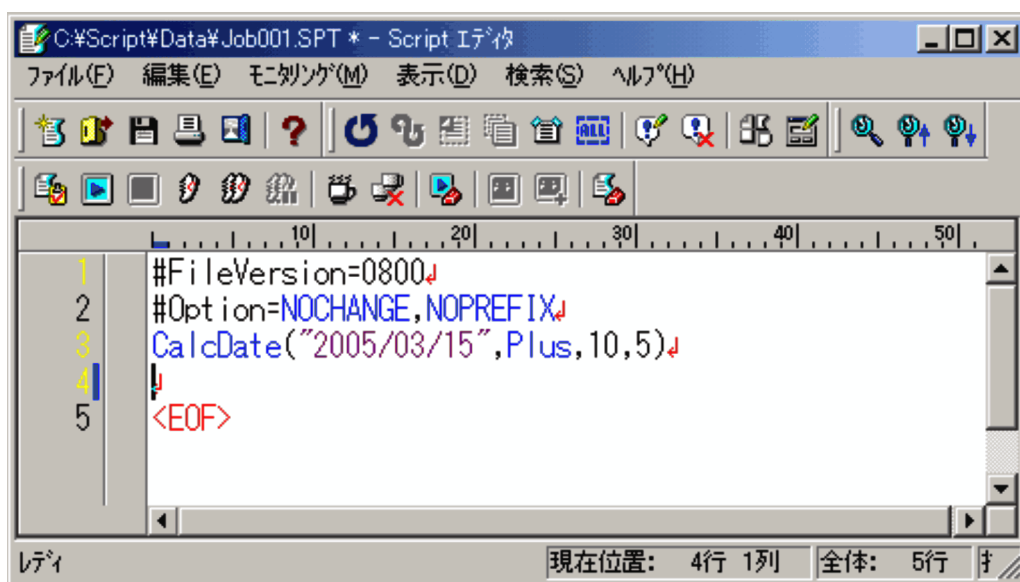
### 3.1.17 エディタの起動と終了

エディタを起動する

実際にスクリプトファイルの作成や編集をする場合は、エディタを起動します。

操作

1. クライアントエリアでスクリプトファイルのアイコンを一つ選択する。
2. Script マネージャウィンドウの [ ファイル ] - [ 編集 ] を選択する。



別のエディタが関連づけられている場合はそのエディタが起動されます。

エディタを終了する

Script エディタウィンドウの [ ファイル ] - [ 終了 ] を選択してください。エディタが終了します。

## 3.2 エディタの操作

---

Script エディタウィンドウでできる操作の一覧を次に示します。

- コメント行を設定・解除する
- 簡易入力機能を使う
- Script メニューフォームを作成する
- エディタの動作環境を設定する
- スクリプトファイルの文法をチェックする
- モニタリングを実行・中止する
- スクリプトを実行する
- モニタリング実行時のブレークポイントを設定・解除する
- モニタリング実行時の動作環境を設定する
- ウォッチへ変数を追加する
- スクリプトファイルの実行環境を設定する
- 文字列を検索する
- 文字列を置換する
- 新しいスクリプトファイルを作成する
- 既存のスクリプトファイルを開く
- スクリプトファイルを保存する
- スクリプトファイルに名前を付けて保存する
- スクリプトファイルの内容を印刷する
- 直前の操作を元に戻す
- 直前の操作をやり直す
- 選択した文字列をクリップボードに切り抜く
- 選択した文字列をクリップボードにコピーする
- クリップボードの文字列を指定の位置に貼り付ける
- すべての文字列を選択する
- 実行ポイント行へジャンプする
- ツールバーの表示・非表示を切り替える
- ステータスバーの表示・非表示を切り替える
- ツールバーを整列させる
- ルーラーの表示・非表示を切り替える
- 縦スクロールバーの表示・非表示を切り替える
- 行番号の表示・非表示を切り替える
- ファイルの先頭を表示する
- ファイルの末尾を表示する
- ウォッチウィンドウの表示・非表示を切り替える
- ヘルプを表示する

## 注

これらの操作は特に説明していません（これらは、Windows の標準の操作に準じているか、メニュー一覧で該当するメニューを選択するだけで実行できる操作です）。

### 3.2.1 Script エディタウィンドウとメニュー

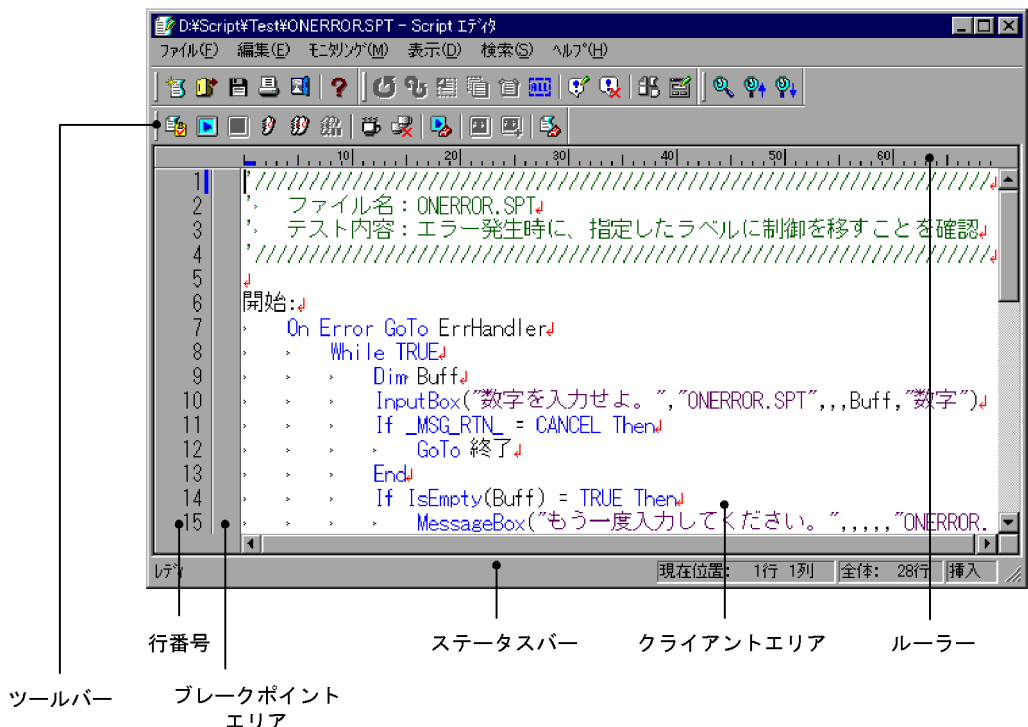
エディタはスクリプトを新規に作成したり、既存のスクリプトを編集したりするためのプログラムです。

ここでは、エディタを起動すると表示される Script エディタウィンドウについて説明します。また、エディタの機能をメニューごとに説明します。

#### (1) Script エディタウィンドウ

Script エディタウィンドウ、およびウィンドウの各部の名称を、図 3-2 に示します。

図 3-2 Script エディタウィンドウ



#### (a) ツールバー

ツールバーには、メニューバーから選択できるコマンドの中から、頻繁に使用するコマンドだけをボタンの形で表示しています。ボタンをマウスでクリックするだけで、該当するコマンドを実行できます。表示メニューによって非表示にすることもできます。

Script エディタウィンドウのツールバーには、次のコマンドのボタンが表示されます。

#### 標準ツールバー

- [ 新規作成 ] ボタン  
スクリプトファイルを新規に作成します。
- [ 開く ] ボタン  
スクリプトファイルを開きます。
- [ 保存 ] ボタン  
スクリプトファイルを保存します。
- [ 印刷 ] ボタン  
スクリプトファイルを印刷します。
- [ 終了 ] ボタン  
Script エディタを終了します。
- [ ヘルプ ] ボタン  
JP1/Script のオンラインヘルプを表示します。

#### 編集ツールバー

- [ 元に戻す ] ボタン  
直前の操作を無効にします。
- [ やり直し ] ボタン  
直前の操作をやり直します。
- [ 切り抜き ] ボタン  
選択した文字列をクリップボードに切り抜きます。
- [ コピー ] ボタン  
選択した文字列をクリップボードにコピーします。
- [ 貼り付け ] ボタン  
クリップボードの文字列を指定の位置に貼り付けます。
- [ すべて選択 ] ボタン  
記述されている文字列をすべて選択します。
- [ コメント設定 ] ボタン  
選択した行をコメント行にします。
- [ コメント解除 ] ボタン  
選択した行をコメント行の設定から解除します。
- [ 簡易入力 ] ボタン  
コマンド、またはステートメントを簡易入力します。
- [ メニューエディタ ] ボタン  
メニューエディタを起動します。

#### モニタリングツールバー

- [ 文法チェック ] ボタン  
記述したスクリプトの文法をチェックします。

- [ ブレークポイントまで実行 ] ボタン  
ブレークポイントまでの実行スタート、およびリスタートをします。
- [ モニタリングの中止 ] ボタン  
モニタリングの実行を中止します。
- [ ステップ実行 ] ボタン  
コマンド、またはステートメントを一つずつ実行します。
- [ 連続ステップ実行 ] ボタン  
コマンド、またはステートメントを連続して一つずつ実行します。
- [ 連続ステップ実行の一時停止 ] ボタン  
連続ステップ実行を一時停止します。
- [ ブレークポイントの設定 / 解除 ] ボタン  
ブレークポイントを設定、または解除します。
- [ ブレークポイントをすべて解除 ] ボタン  
設定されているブレークポイントをすべて解除します。
- [ 設定 ] ボタン  
モニタリング実行時の環境を設定します。
- [ ウォッチウィンドウの表示 / 非表示 ] ボタン  
ウォッチウィンドウの表示・非表示を切り替えます。
- [ ウォッチへ追加 ] ボタン  
指定した変数をウォッチウィンドウに追加します。
- [ 実行環境の設定 ] ボタン  
スクリプトファイルの環境を設定します。

#### 検索ツールバー

- [ 検索 ] ボタン  
検索する文字列を入力します。
- [ 前検索 ] ボタン  
検索文字列を上に向かって検索します。
- [ 次検索 ] ボタン  
検索文字列を下に向かって検索します。



#### (b) ルーラー

行のカラム数を表示する目盛です。

#### (c) 行番号

スクリプトの行番号を表示するための領域です。

#### (d) ブレークポイントエリア

ブレークポイントを示す記号 (  ) と、次に実行する位置を示す記号 (  ) を表示するための領域です。

## (e) ステータスバー

ステータスバーは、Script エディタが現在実行している処理に関するメッセージや、処理終了後の状態に関するメッセージを表示するための領域です。

## (f) クライアントエリア

クライアントエリアには、スクリプトファイルの内容が表示されます。

## (2) Script エディタウィンドウのメニュー

ここでは、Script エディタウィンドウのメニューバーに表示されるメニュー、および Script エディタウィンドウで表示されるポップアップメニューについて説明します。

## (a) メニューバーのメニュー

エディタの機能を、メニューごとに説明します。エディタのメニュー一覧を表 3-6 に示します。

表 3-6 Script エディタのメニュー一覧

メニュー	コマンド (機能)	機能概要
[ ファイル ]	[ 新規作成 ]	スクリプトファイルを新規に作成します。
	[ 開く ]	既存のスクリプトファイルを開きます。
	[ 保存 ]	スクリプトファイルを上書き保存します。
	[ 名前を付けて保存 ]	スクリプトファイルを別名で保存します。
	[ 印刷 ]	スクリプトファイルを印刷します。
	[ 終了 ]	エディタを終了します。
	( ファイル名 )	直前に保存したスクリプトファイルが最大 9 個表示されます。
[ 編集 ]	[ 元に戻す ]	編集する直前の状態に戻します。
	[ やり直し ]	直前の操作をやり直します。
	[ 切り抜き ]	選択した文字列をクリップボードに切り抜きます。
	[ コピー ]	選択した文字列をクリップボードにコピーします。
	[ 貼り付け ]	クリップボードの文字列を指定した位置に貼り付けます。
	[ すべて選択 ]	記述されているすべての文字列を選択します。
	[ コメント設定 ]	選択した行をコメント行に設定します。
	[ コメント解除 ]	選択したコメント行のコメント属性を解除します。
	[ 簡易入力 ]	簡易入力機能を使って、コマンド、またはステートメントを入力します。
	[ メニューエディタ ]	メニューエディタを起動します。
	[ オプション ]	エディタの動作環境を設定します。
[ モニタリング ]	[ 文法チェック ]	スクリプトの文法をチェックします。



メニュー	コマンド (機能)	機能概要
	[ モニタリングの実行 ] - [ 実行 ]	モニタリングモードで、ブレークポイントまで実行、またはリスタートします。
	[ モニタリングの実行 ] - [ ステップ実行 ]	モニタリングモードで、コマンド、またはステートメントを一つずつ実行し、一つ実行するごとに一時停止します。
	[ モニタリングの実行 ] - [ 連続ステップ実行 ]	モニタリングモードで、コマンド、またはステートメントを一つずつ実行し、[ 連続ステップ実行の一時停止 ] が選択された時点で一時停止します。
	[ モニタリングの実行 ] - [ 連続ステップ実行の一時停止 ]	連続ステップ実行を一時停止します。
	[ 実行 ]	スクリプトファイルを最後まで実行します。
	[ モニタリングの中止 ]	モニタリングモードでの実行を中止します。
	[ ブレークポイントの設定 / 解除 ]	ブレークポイントを設定、または解除します。
	[ ブレークポイントをすべて解除 ]	設定されているブレークポイントをすべて解除します。
	[ 設定 ]	モニタリング実行時の環境を設定します。
	[ ウォッチへ変数の追加 ]	指定した変数をウォッチウィンドウへ追加します。
	[ 実行環境の設定 ]	スクリプトファイルの環境を設定します。
	[ 実行ポイント行へジャンプ ]	モニタリングモードで現在実行している行を表示します。
[ 表示 ]	[ ツールバー ] - [ 標準ツールバー ]	標準ツールバーの表示・非表示を切り替えます。
	[ ツールバー ] - [ 編集ツールバー ]	編集ツールバーの表示・非表示を切り替えます。
	[ ツールバー ] - [ モニタリングツールバー ]	モニタリングツールバーの表示・非表示を切り替えます。
	[ ツールバー ] - [ 検索ツールバー ]	検索ツールバーの表示・非表示を切り替えます。
	[ ステータスバー ]	ステータスバーの表示・非表示を切り替えます。
	[ ツールバーの整列 ]	ツールバーを整列させます。
	[ ルーラー ]	ルーラーの表示・非表示を切り替えます。
	[ 縦スクロールバー ]	縦スクロールバーの表示・非表示を切り替えます。
	[ 横スクロールバー ]	横スクロールバーの表示・非表示を切り替えます。
	[ 行番号を表示 ]	行番号の表示・非表示を切り替えます。
	[ ファイルの先頭を表示 ]	スクリプトファイルの先頭を表示します。
	[ ファイルの末尾を表示 ]	スクリプトファイルの末尾を表示します。
	[ ウォッチウィンドウを表示 ]	ウォッチウィンドウの表示・非表示を切り替えます。

### 3. JP1/Script の操作

メニュー	コマンド（機能）	機能概要
[ 検索 ]	[ 検索 ]	検索，または置換する文字列，および方法を入力します。
	[ 置換 ]	検索文字列を，指定した文字列に置換します。
	[ 前検索 ]	検索文字列を上に向かって検索，または置換します。
	[ 次検索 ]	検索文字列を下に向かって検索，または置換します。
[ ヘルプ ]	[ 目次 ]	JP1/Script オンラインヘルプの目次を表示します。
	[ キーワードで検索 ]	JP1/Script オンラインヘルプのキーワード一覧を表示します。
	[ ヘルプの使い方 ]	オンラインヘルプの使い方を表示します。
	[ バージョン情報 ]	エディタのバージョン情報を表示します。

#### （b）ポップアップメニュー

Script エディタウィンドウのクライアントエリアでマウスの右ボタンをクリックすると，ポップアップメニューが表示されます。ポップアップメニューの内容は編集モードの場合とモニタリングモードの場合とで異なります。編集モードの場合に表示されるポップアップメニューを表 3-7 に，モニタリングモードの場合に表示されるポップアップメニューを表 3-8 に示します。

表 3-7 ポップアップメニュー一覧（編集モード）

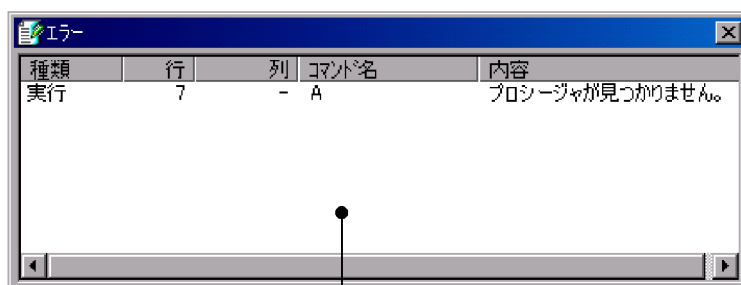
ポップアップメニュー	機能概要
[ 新規作成 ]	スクリプトファイルを新規に作成します。
[ 開く ]	既存のスクリプトファイルを開きます。
[ 保存 ]	スクリプトファイルを上書き保存します。
[ 元に戻す ]	編集する直前の状態に戻します。
[ やり直し ]	直前の操作をやり直します。
[ 切り抜き ]	選択した文字列をクリップボードに切り抜きます。
[ コピー ]	選択した文字列をクリップボードにコピーします。
[ 貼り付け ]	クリップボードの文字列を指定の位置に貼り付けます。
[ すべて選択 ]	記述されている文字列をすべて選択します。
[ コメント設定 ]	選択した行をコメント行に設定します。
[ コメント解除 ]	選択したコメント行のコメント属性を解除します。
[ 簡易入力 ]	簡易入力機能を使って，コマンド，またはステートメントを入力します。
[ メニューエディタ ]	メニューエディタを起動します。

表 3-8 ポップアップメニュー一覧（モニタリングモード）

ポップアップメニュー	機能概要
[ コピー ]	選択した文字列をクリップボードにコピーします。
[ ブレークポイントの設定 / 解除 ]	ブレークポイントを設定、または解除します。
[ ウォッチへ追加 ]	選択した変数をウォッチウィンドウへ追加します。

### （３）エラーウィンドウ

モニタリング実行で発生したエラーを表示します。



クライアントエリア

#### （a）クライアントエリア

##### 種類

エラーの種類を表示します。この欄をダブルクリックすると、該当するエラーの個所にカーソルが移動します。

##### 行

エラー発生個所の行を表示します。

##### 列

エラー発生個所の列を表示します。

##### コマンド名

エラーの発生したコマンドを表示します。

##### 内容

エラーの内容を表示します。

#### （b）エラーウィンドウの操作

エラーウィンドウ上でマウスの右ボタンをシングルクリックすると、次のポップアップメニューが表示されます。

- ジャンプ

選択した行 / 列位置へジャンプします。

(4) ウォッチウィンドウ

モニタリング実行時に変数名とその変数の値を表示します。

ウォッチウィンドウへの変数の追加方法は「3.2.12 ウォッチへ変数を追加する」を参照してください。



(a) クライアントエリア

変数名

変数名が表示されます。

値

変数名が無効のときは「ErrorInvalid Data」と表示されます。また、配列変数名だけを指定したインデックス番号が有効な範囲にない場合、指定した配列変数の次元が異なる場合は値を参照できないため、「ErrorInvalid Data」と表示されます。

(b) ウォッチウィンドウの操作

ウォッチウィンドウ上でマウスの右ボタンをシングルクリックすると、次のポップアップメニューが表示されます。

- 値の更新  
選択した変数名の値を更新します。
- 変数名の削除  
選択した変数名を削除します。

3.2.2 Script エディタウィンドウでのマウスとキーの操作

(1) マウス操作

Script エディタウィンドウのクライアントエリアでのマウス操作を表 3-9 に示します。

表 3-9 Script エディタウィンドウでのマウス操作

操作	処理
クリック	それまでの選択を解除して、新たに対象を選択する。

操作	処理
ダブルクリック	文字列を選択する。
右ボタンシングルクリック	ポップアップメニューを表示する。

## (2) キー操作

Script エディタウィンドウのクライアントエリアでのキー操作と、モードごとの操作の可否を表 3-10 に示します。

表 3-10 Script エディタウィンドウでのキー操作

操作	編集モード	モニタリングモード	処理
Ctrl+A			記述されているすべての文字列を選択する。
Ctrl+C			選択された文字列をコピーする。
Ctrl+E		×	メニューエディタを起動する。
Ctrl+F			検索、置換する情報を入力する。
Ctrl+H		×	検索文字列を指定した文字列に置換する。
Ctrl+K		×	コマンド、またはステートメントを簡易入力する。
Ctrl+M		×	選択された行をコメント行にする。
Ctrl+N		×	スクリプトファイルを新規に作成する。
Ctrl+O		×	スクリプトファイルを開く。
Ctrl+P		×	スクリプトファイルを印刷する。
Ctrl+S		×	スクリプトファイルを保存する。
Ctrl+T		×	モニタリング実行時の動作環境を設定する。
Ctrl+V		×	クリップボードの内容を貼り付ける。
Ctrl+X		×	選択された文字列を切り抜く。
Ctrl+Z		×	編集される直前の状態に戻る。
Ctrl+F5		×	スクリプトファイルを実行する。
Ctrl+F11			連続ステップ実行を開始する。
Ctrl+Home			スクリプトファイルの先頭を表示する。
Ctrl+End			スクリプトファイルの末尾を表示する。
F1			Script ヘルプを表示する。

### 3. JP1/Script の操作

操作	編集モード	モニタリングモード	処理
F3			検索文字列を下に向かって検索，置換する。
F5		×	ブレークポイントまで実行をスタート，またはリスタートさせる。
F7		×	文法チェックを行う。
F9			ブレークポイントを設定，または解除する。
F11	×		次のコマンド，またはステートメントを一つずつ実行する。
Alt+0	×		ウォッチウィンドウへ指定された変数を追加する。
Alt+1	×		ウォッチウィンドウの表示，非表示を切り替える。
Alt+F4			Script エディタを終了する。
Shift+F3			検索文字列を上に向かって検索，置換する。
Shift+F5	×		モニタリング実行を中止する。
Shift+F9			指定ブレークポイントをすべて解除する。
Shift+Ctrl+M		×	選択された行のコメント化を解除する。
Shift+Ctrl+Z		×	直前に行った操作をやり直す。
Shift+Ctrl+F11	×		連続ステップ実行を一時停止する。
Shift+Ctrl+Enter		×	スクリプトファイルの環境を設定する。

（凡例）

：操作できる機能

：一部操作できる機能

×：操作できない機能

注 カースルがコマンド，またはステートメントの位置にある状態で F1 キーを押すと，そのコマンド，またはステートメントのヘルプが表示されます。

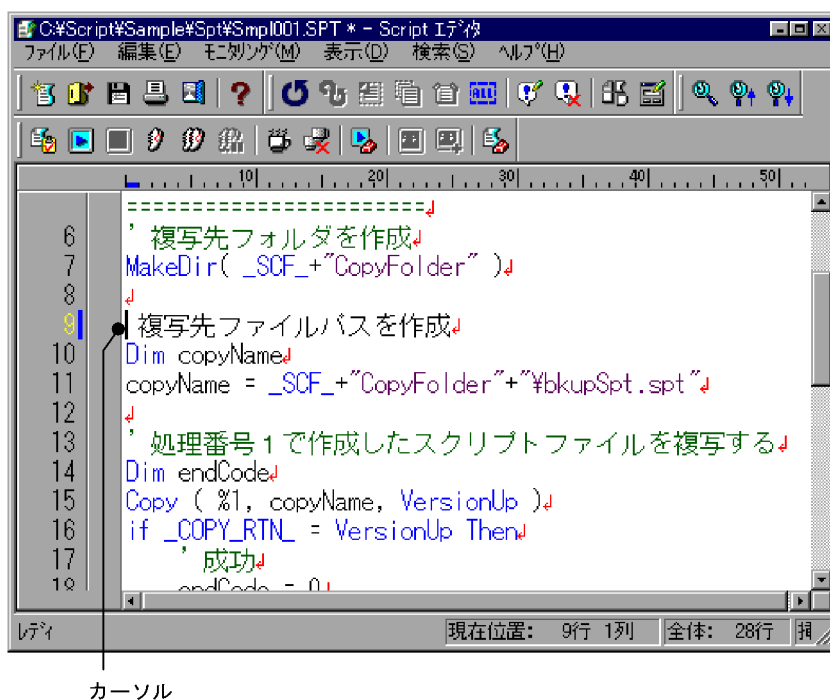
#### 3.2.3 コメント行を設定・解除する

コメント行を設定，または解除します。

操作

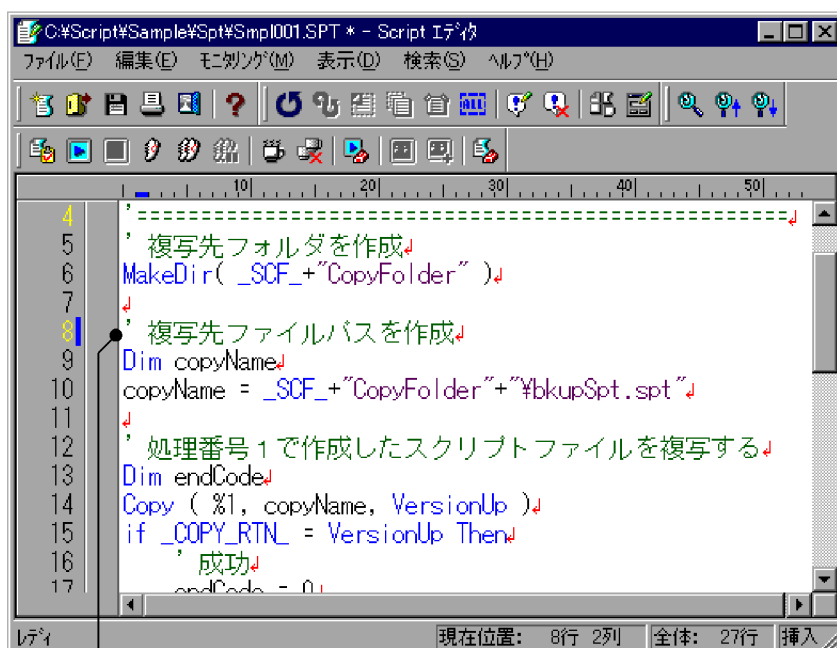
コメント行を設定する

1. コメントにしたい行にカーソルを移動させる。



2. [ 編集 ] - [ コメント設定 ] メニューを選択する。  
カーソルのある行がコメント行になります。

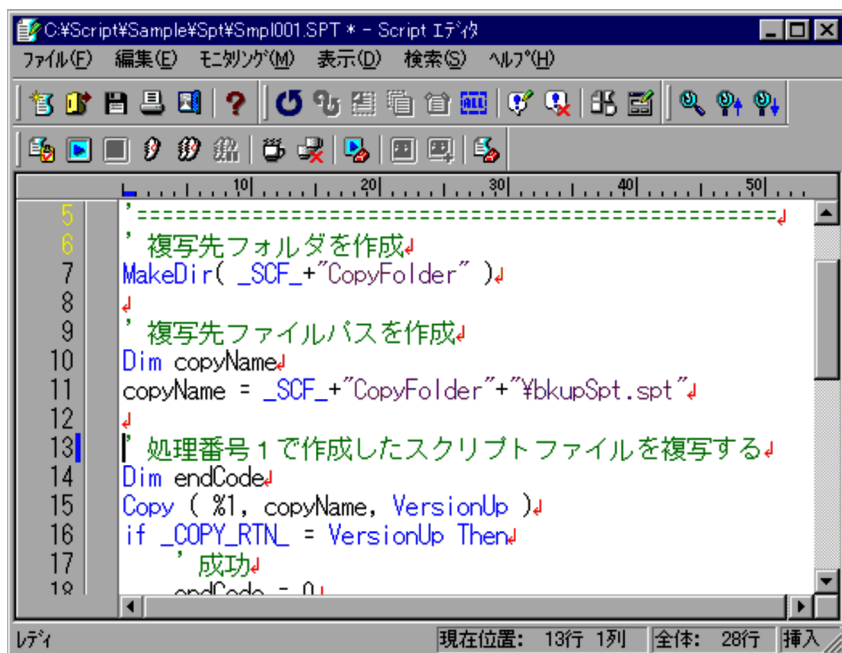
### 3. JP1/Script の操作



コメントを表すアポストロフィ ( ' ) が付く

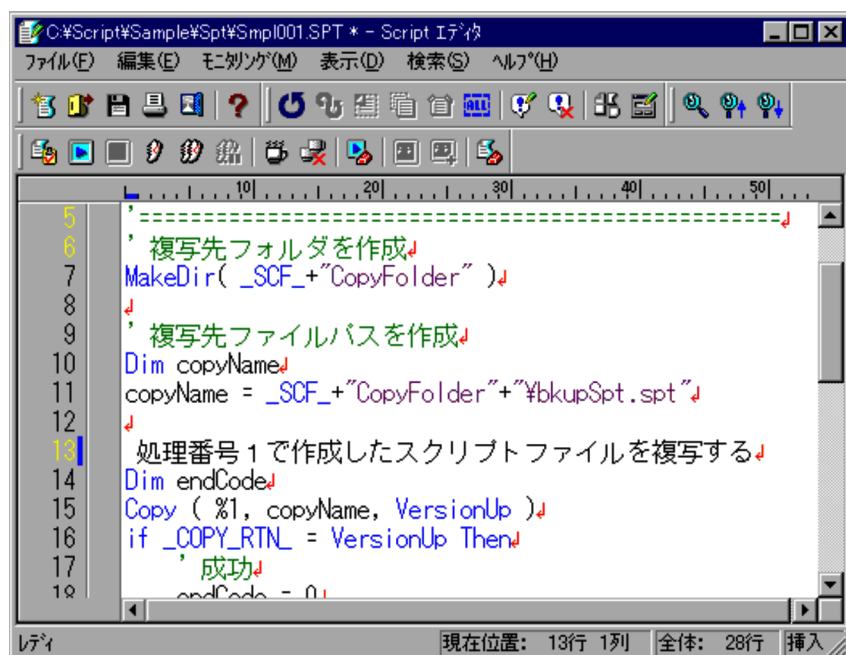
コメント行を解除する

1. コメント行の設定を解除したい行にカーソルを移動させる。





2. [ 編集 ] - [ コメント解除 ] メニューを選択する。  
カーソルのある行がコメント行の設定から解除されます。



- 複数行を選択して操作すると、選択した行すべてに対してコメント行が設定、または解除されます。
- モニタリングモードの場合は、メニューがグレイアウトされて [ 編集 ] - [ コメント設定 ], および [ コメント解除 ] メニューは選択できません。

### 3.2.4 簡易入力機能を使う

スクリプトを記述する際の、コマンドやステートメントの入力作業を、簡易入力機能を使って行います。

この機能を使うことで、コマンドの知識がなくても手軽に間違いなく入力できます。

記述したコマンドやステートメントの内容は、[ エディタへ ] ボタンでエディタに貼り付けます。

操作

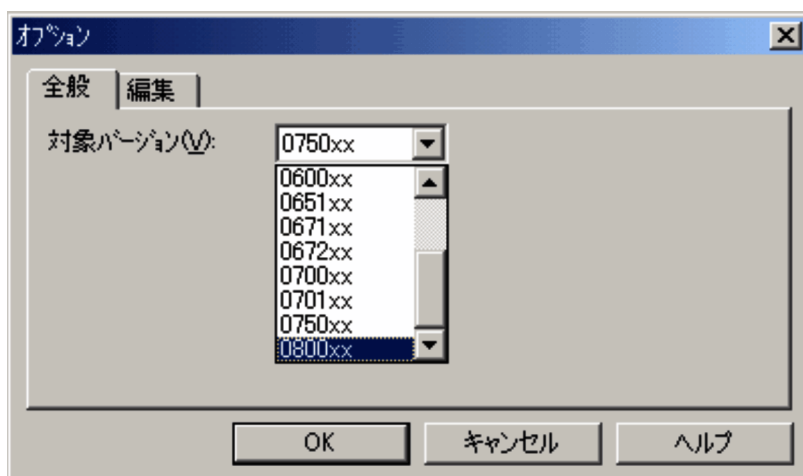
1. [ 編集 ] - [ 簡易入力 ] メニューを選択して、簡易入力を起動する。  
[ Script 簡易入力 ] ダイアログが表示されます。

### 3. JP1/Script の操作



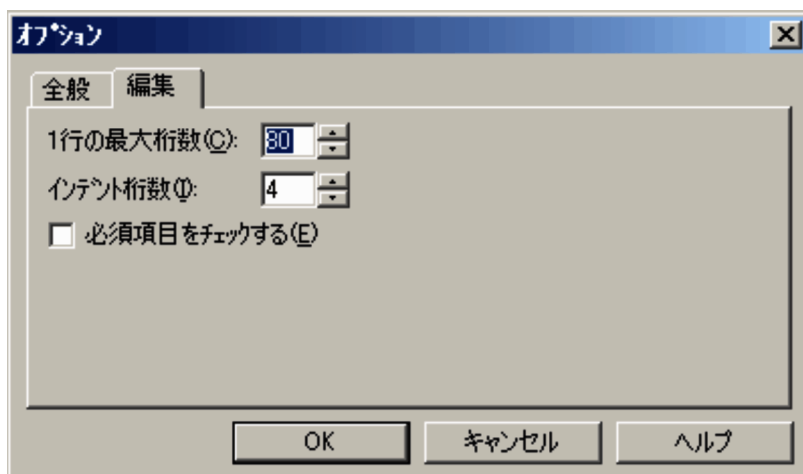
ダイアログの設定方法については、「3.3 簡易入力の操作」を参照してください。

2. [ オプション ] ボタンをクリックして、[ 簡易入力オプション (全般) ] ダイアログでスクリプトファイルの作成バージョンを設定する。



ダイアログの設定方法については、「3.3.3 JP1/Script の対象バージョンを指定する」を参照してください。

3. [ 編集 ] タブをクリックして、[ 簡易入力オプション (編集) ] ダイアログでエディタへの出力形式を設定する。



ダイアログの設定方法については、「3.3.4 1行の最大文字数やインデントの桁数を指定する」を参照してください。

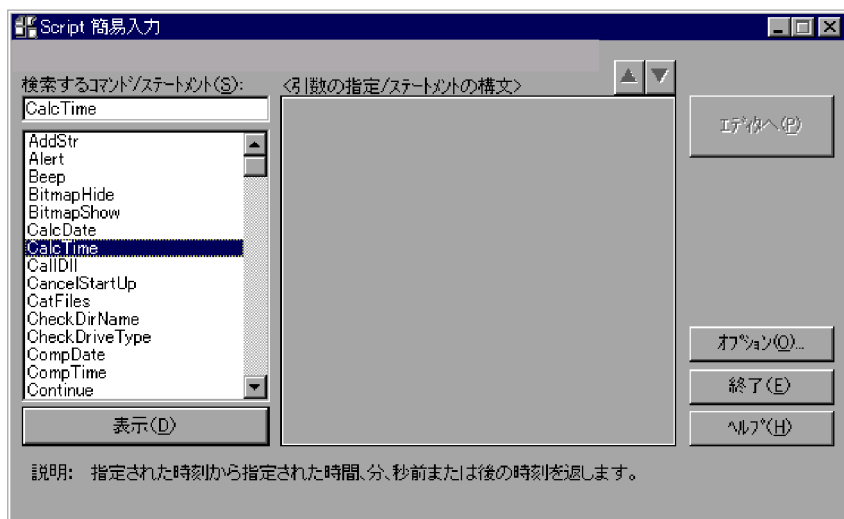
4. [ OK ] ボタンをクリックする。

スクリプトファイルの作成バージョン、およびエディタへの出力形式が設定され、[ Script 簡易入力 ] ダイアログに戻ります。

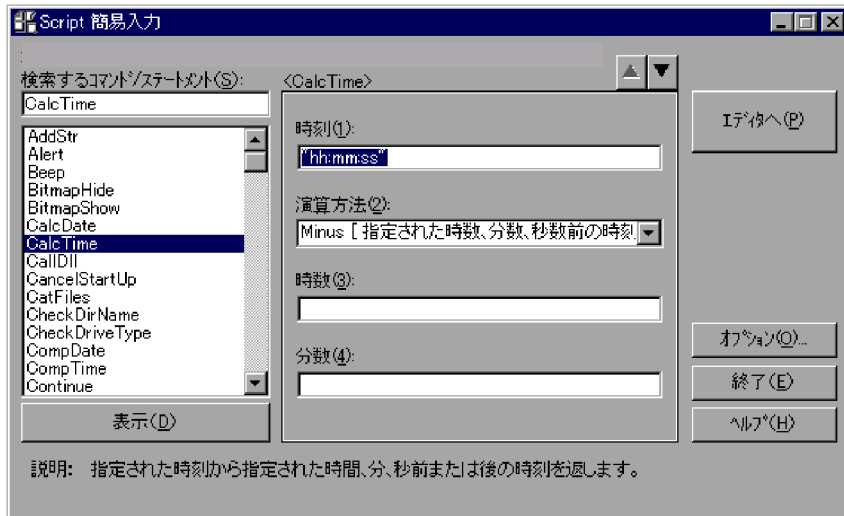


5. 使用するコマンド、またはステートメントを検索し、選択する。  
該当するコマンド、またはステートメントにフォーカスが移動します（ここでは、「CalcTime」を選択）。

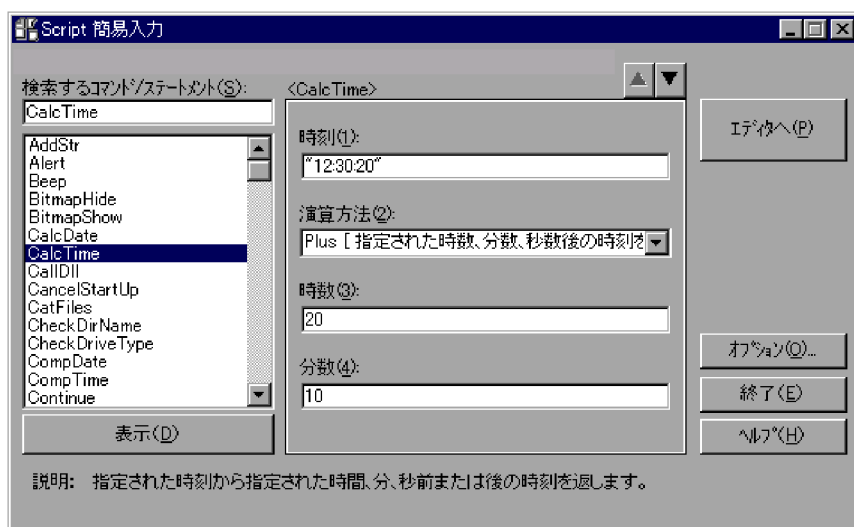
### 3. JP1/Script の操作



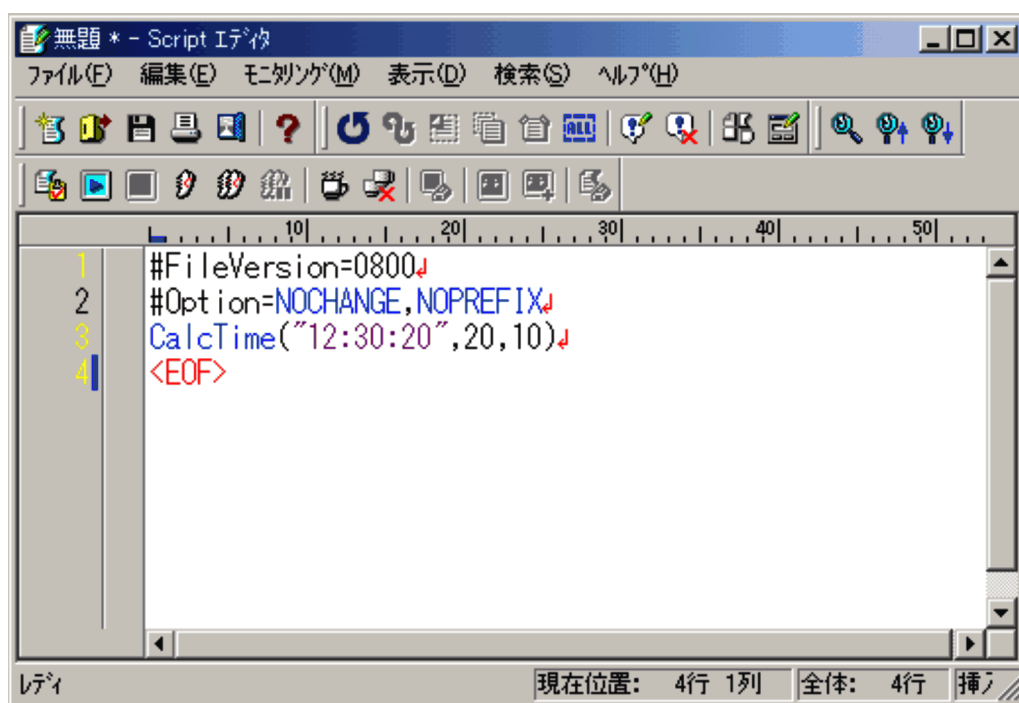
6. [ 表示 ] ボタンをクリックするか、コマンド、またはステートメントをダブルクリックする。
- 「引数の指定 / ステートメントの構文」に、選択したコマンドの引数を入力する項目が表示されます。
- ステートメントの場合は、ステートメントの構文が表示されます。



7. コマンドの引数の項目を入力する。



8. [ エディタへ ] ボタンをクリックする。  
「引数の指定 / ステートメントの構文」の内容がエディタに貼り付けられます。



- 「検索するコマンド / ステートメント」の一覧 (リストボックス) で、コマンド名、またはステートメント名を選択し、マウスの右ボタンをクリックすると、選択されたコマンド、またはステートメントのヘルプが表示されます。
- 操作 5. で引数がないコマンドの場合は、「引数の指定 / ステートメントの構文」に

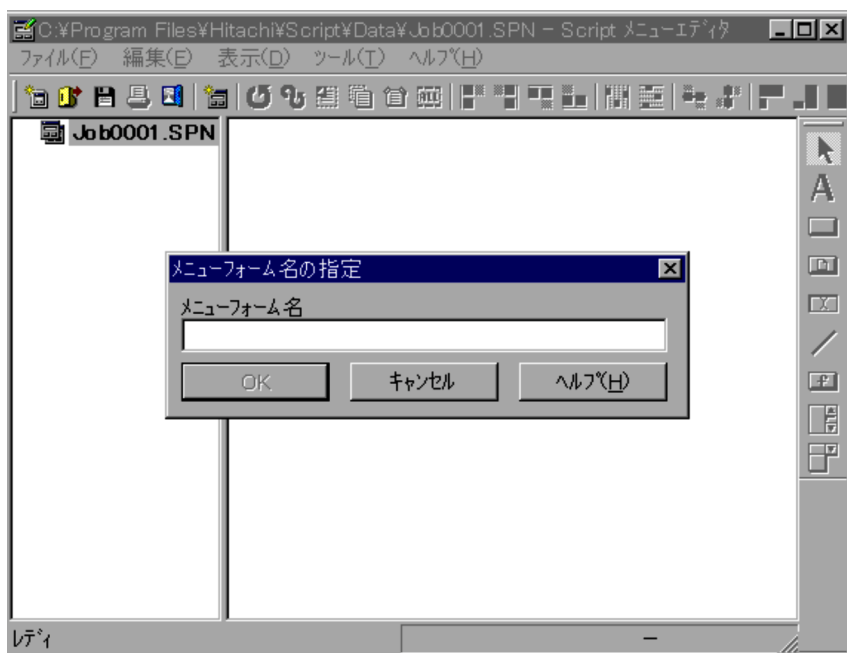
「このコマンドに引数はありません。」というメッセージが表示されます。

### 3.2.5 メニューフォームを作成する

メニューフォームを作成します。

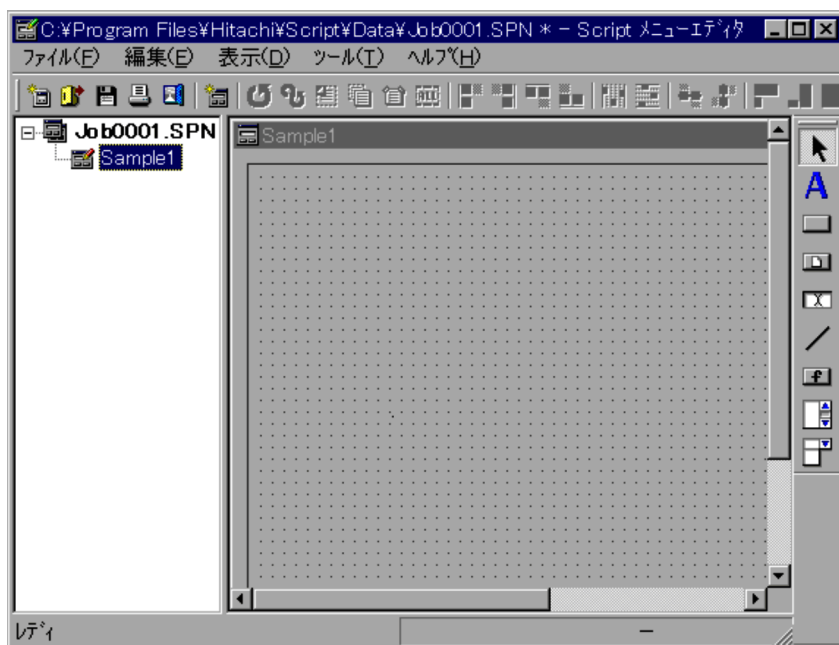
操作

1. [ 編集 ] - [ メニューエディタ ] メニューを選択する。  
Script メニューエディタウィンドウが表示されます。  
メニューフォームが作成されていない場合は、Script メニューエディタウィンドウの手前に、[ メニューフォーム名の指定 ] ダイアログが表示されます。なお、画面左下にはコマンドプロパティダイアログボックスが表示されます。



メニューフォームが作成されている場合は、操作 3. に進んでください。

2. メニューフォーム名を入力する。  
フォームビューに新しいメニューフォームが表示されます。操作 4. に進んでください。



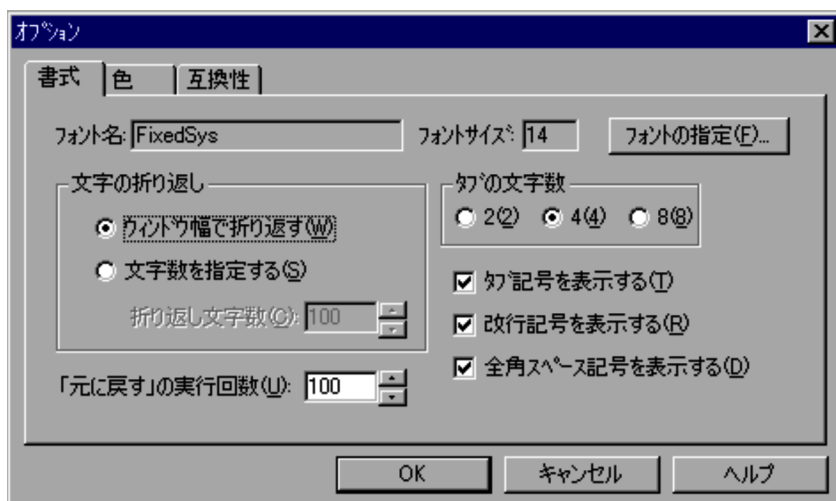
3. [編集] - [メニューフォームの新規作成] メニューを選択する。  
フォームビューに新しいメニューフォームが表示されます。
4. メニューフォームを作成する。  
作成方法については、「3.6.3 新しいメニューフォームを作成する」を参照してください。
5. メニューエディタを終了する。
  - エディタがモニタリングモードになっている場合は、[編集] - [メニューエディタ] メニューはグレースアウトされて選択できません。

### 3.2.6 エディタの動作環境を設定する

エディタの動作環境を設定します。

操作

1. [編集] - [オプション] メニューを選択する。  
[オプション (書式)] ダイアログが表示されます。



ダイアログの設定方法については「4.2.2 オプション（書式）ダイアログ」を参照してください。

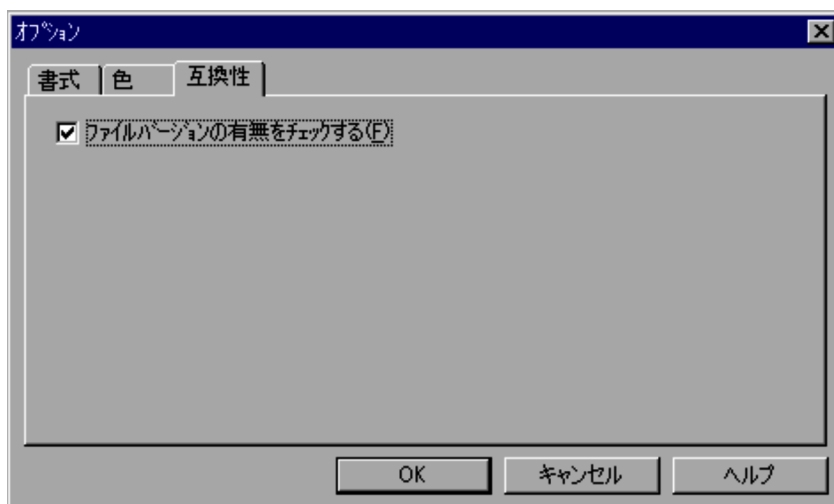
2. 書式に関する情報を設定する。
3. [ 色 ] タブをクリックする。  
[ オプション（色）] ダイアログが表示されます。



4. 表示色に関する情報を設定する。  
すべての項目の表示色をデフォルトに戻すには [ すべてリセット ] を選択してください。  
ダイアログの設定方法については「4.2.3 オプション（色）ダイアログ」を参照してください。
5. [ 互換性 ] タブをクリックする。



[ オプション ( 互換性 ) ] ダイアログが表示されます。



6. ファイルバージョンの有無をチェックするかどうかを設定する。  
ダイアログの設定方法については「4.2.4 オプション ( 互換性 ) ダイアログ」を参照してください。
7. [ OK ] ボタンをクリックする。  
エディタの動作環境が設定され、ダイアログが閉じます。

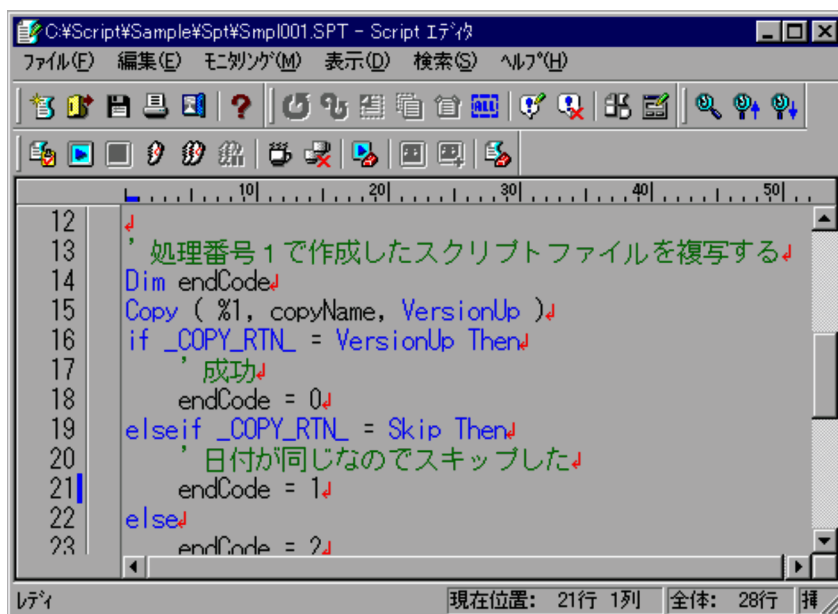
### 3.2.7 スクリプトファイルの文法をチェックする

スクリプトファイルの文法をチェックします。

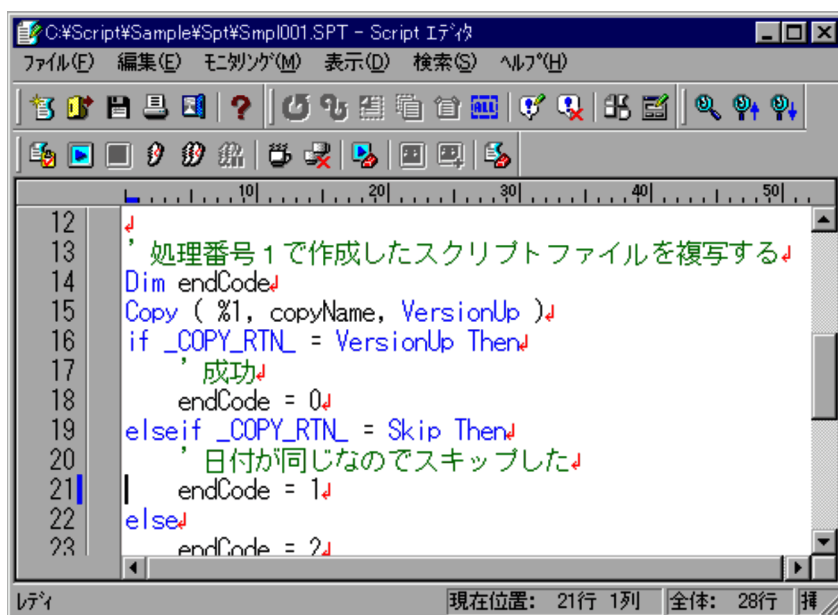
操作

1. [ モニタリング ] - [ 文法チェック ] メニューを選択する。  
エディタがモニタリングモードになり、文法チェックが開始されます。  
文法チェック時は一瞬、暗くなります。

### 3. JP1/Script の操作



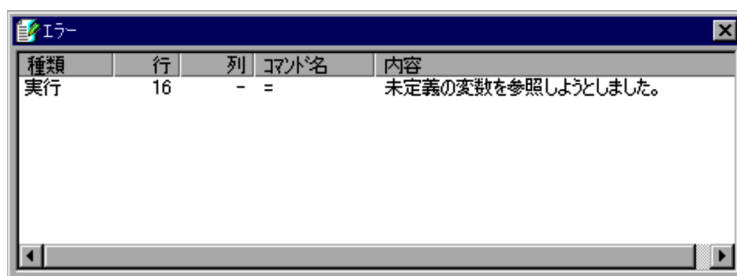
(文法チェック中の表示)



(文法チェック終了の表示)

文法エラーがある場合はエラーウィンドウにエラー内容が表示されます。

#### 2. エラーウィンドウの内容を確認する。



- ・モニタリングモードのときは、メニューがグレイアウトされて [ モニタリング ] - [ 文法チェック ] は選択できません。
- ・ファイル名のないスクリプトファイルに対して文法チェックを実行しようすると、ファイル名を付けて保存するためのダイアログが表示されます。ファイル名を付けて保存しないと、文法チェックは実行できません。
- ・スクリプトファイルの内容が更新されている場合は、更新された内容で文法チェックを実行しますが、ファイルへの保存は行いません。

### 3.2.8 モニタリングを実行・中止する



スクリプトファイルの動作を確認しながら実行することをモニタリングといいます。

モニタリングの方法には、次の 3 種類があります。

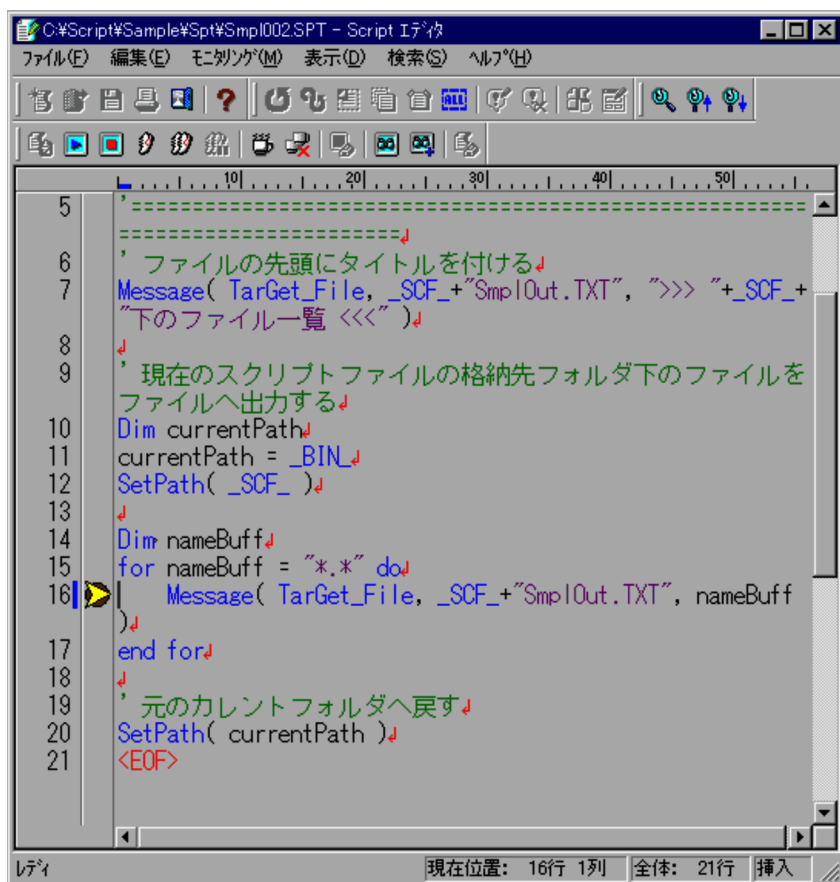
メニュー	機能
実行	ブレークポイントまで実行、またはリスタートします。
ステップ実行	コマンド、またはステートメントを一つずつ実行します。
連続ステップ実行	コマンド、またはステートメントを一つずつ実行し、[ 連続ステップ実行の一時停止 ] が選択された時点で一時停止します。

#### 操作

「実行」の場合

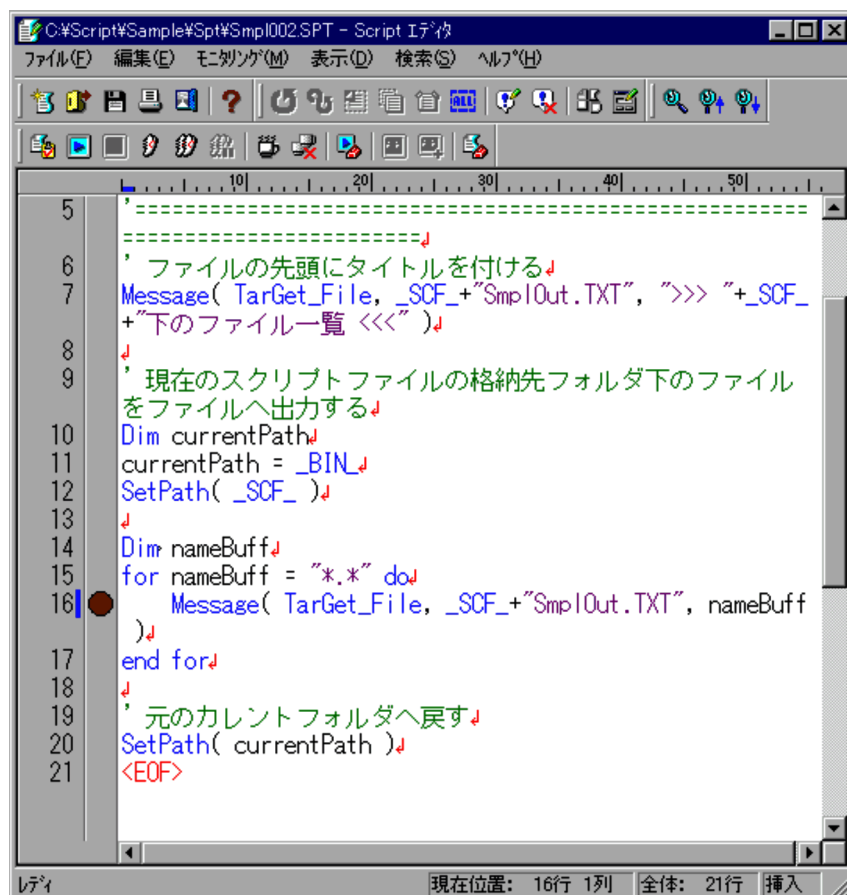
1. [ モニタリング ] - [ モニタリングの実行 ] - [ 実行 ], またはツールバーの [ ブレークポイントまで実行 ] ボタンを選択する。  
Script エディタがモニタリングモードになり、モニタリングが始まります。次に実行される行の左側には、実行する位置を示す記号 (  ) が表示されます。コメント行やスペース行は無視されます。  
ブレークポイント (  ) の指定されている行まで実行されると、実行が一時的に停止されます。

### 3. JP1/Script の操作



ブレークポイントのダイアログの設定方法については、「3.2.10 モニタリング実行時のブレークポイントを設定・解除する」を参照してください。

2. モニタリングを中止したい場合は、[モニタリング] - [モニタリングの中止]メニュー、またはツールバーの[モニタリングの中止]ボタンを選択する。  
メニューを選択した時点で、モニタリングが中止され、エディタは編集モードに戻ります。

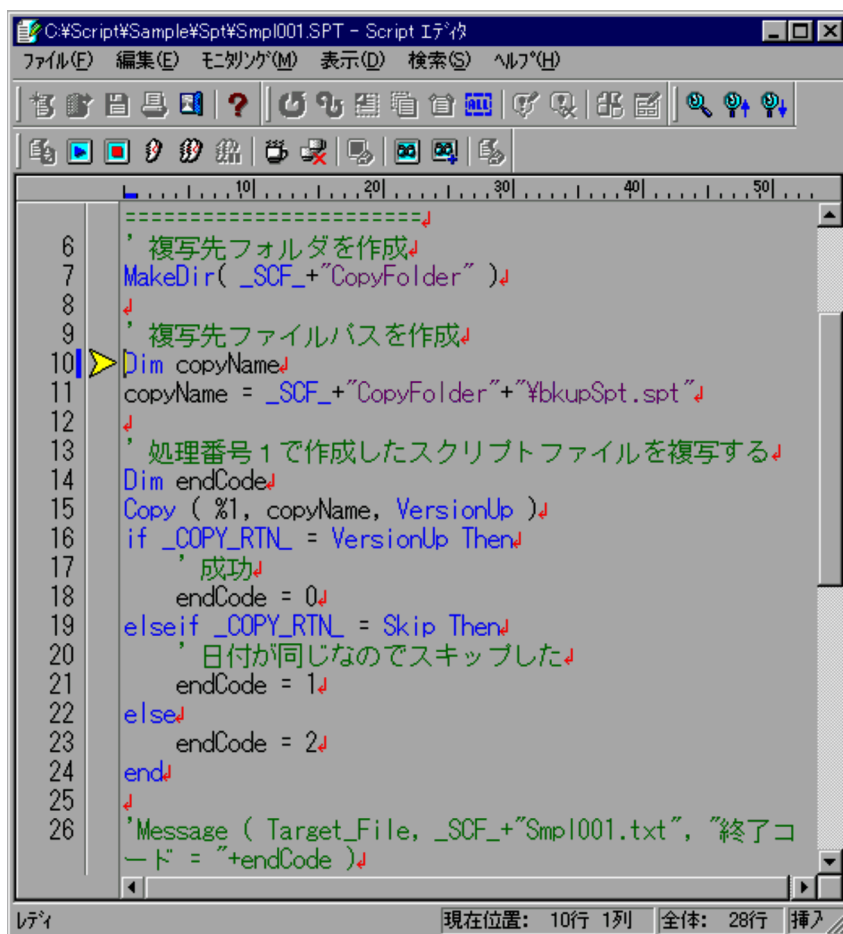


「ステップ実行」の場合

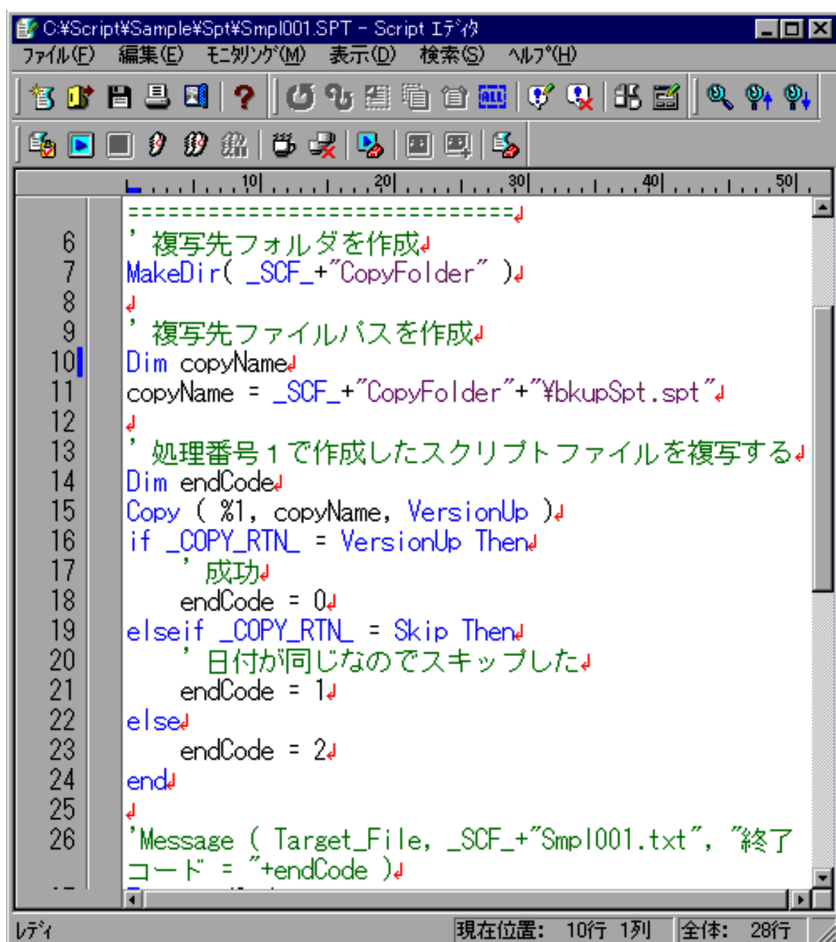
1. [ モニタリング ] - [ モニタリングの実行 ] - [ ステップ実行 ], またはツールバーの [ ステップ実行 ] ボタンを選択する。

エディタがモニタリングモードになり、モニタリングが始まります。次に実行される行の左側には、実行する位置を示す記号 ( ▶ ) が表示されます。コメント行やスペース行は無視されます。

### 3. JP1/Script の操作



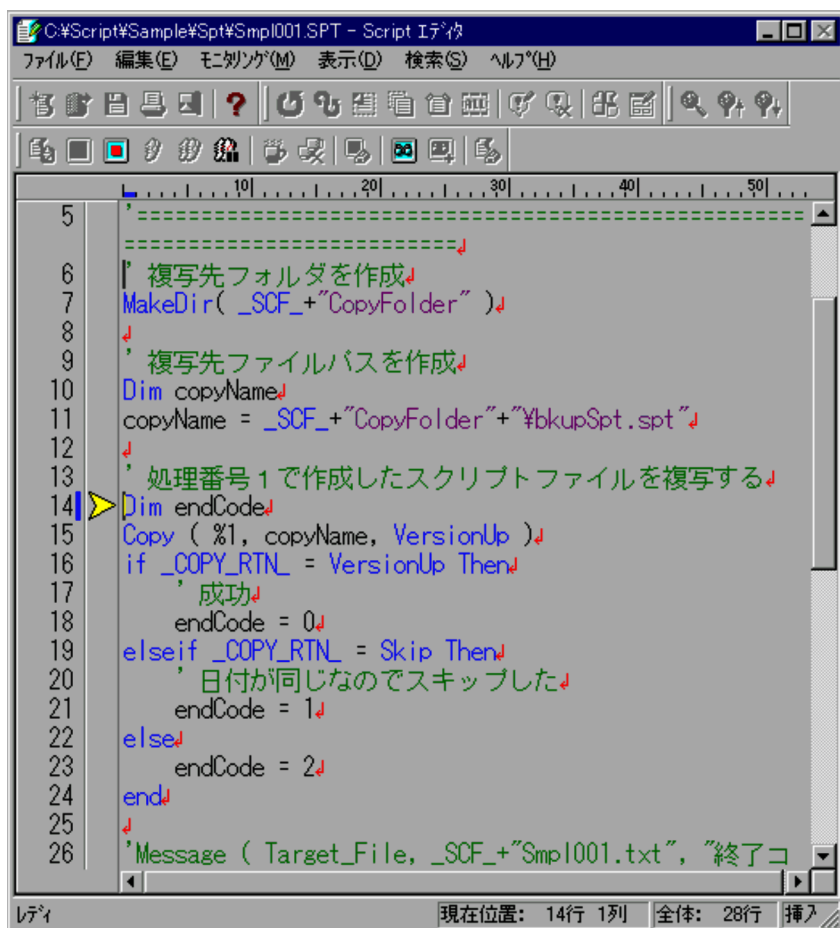
2. モニタリングを中止したい場合は、[ モニタリング ] - [ モニタリングの中止 ] メニュー，またはツールバーの [ モニタリングの中止 ] ボタンを選択する。  
メニューを選択した時点で，モニタリングが中止され，エディタは編集モードに戻ります。



「連続ステップ実行」の場合

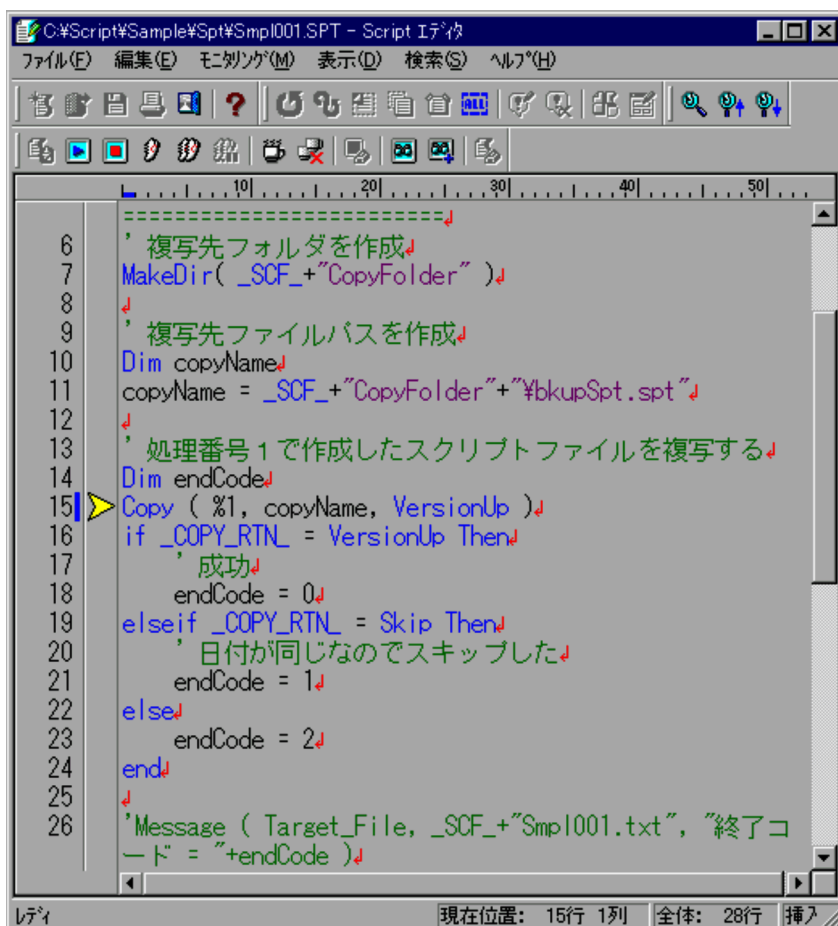
1. [ モニタリング ] - [ モニタリングの実行 ] - [ 連続ステップ実行 ], またはツールバーの [ 連続ステップ実行 ] ボタンを選択する。  
エディタがモニタリングモードになり、モニタリングが始まります。次に実行される行の左側には、実行する位置を示す記号 ( ➤ ) が表示されます。コメント行やスペース行は無視されます。

### 3. JP1/Script の操作

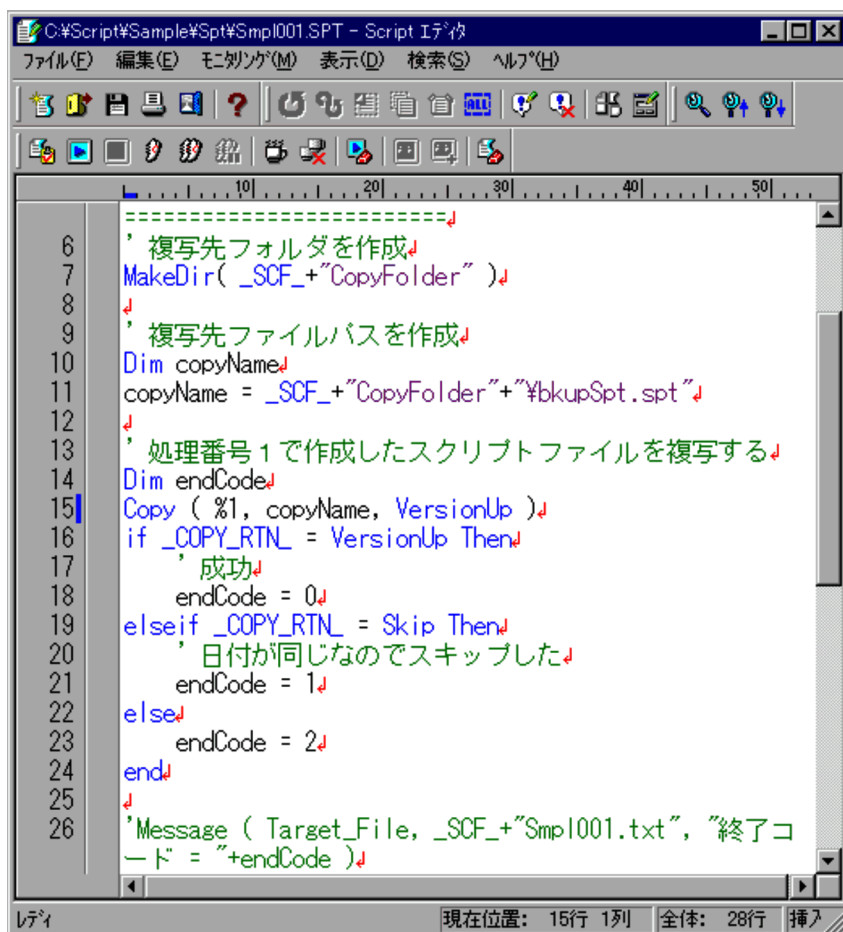


2. モニタリングを一時停止したいところで [ モニタリング ] - [ モニタリングの実行 ]  
- [ 連続ステップ実行の一時停止 ] メニュー , またはツールバーの [ 連続ステップ実行の一時停止 ] ボタンを選択する。  
メニューを選択した時点で、モニタリングが一時停止されます。





3. モニタリングを中止したい場合は、[ モニタリング ] - [ モニタリングの中止 ] メニュー，またはツールバーの [ モニタリングの中止 ] ボタンを選択する。  
メニューを選択した時点で、モニタリングが中止され、エディタは編集モードに戻ります。



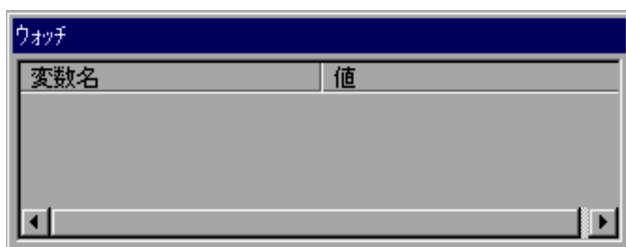
- 無限ループになるスクリプトファイルの連続ステップ実行中に、[ モニタリング ] - [ モニタリングの実行 ] - [ 連続ステップ実行の一時停止 ] メニューを選択しても、連続ステップ実行は一時停止されません。このような場合は、[ モニタリング ] - [ モニタリングの中止 ] メニューを選択してステップ実行を中止してください。
- モニタリングで検出された解析エラーや実行エラーは、エラーウィンドウに表示されます。また、ウォッチウィンドウに追加されている変数の値は、実行中に逐一更新されて、表示されます。

(エラーウィンドウの例)



- ウォッチウィンドウを表示したい場合は、モニタリング実行中に [表示] - [ウォッチウィンドウを表示] メニューをチェックしてください。このメニューは、モニタリング実行中にしかチェックできません。

(ウォッチウィンドウの例)



- ファイル名のないスクリプトファイルに対してモニタリングを実行しようとする、ファイル名を付けて保存するためのダイアログが表示されます。ファイル名を付けて保存しないと、モニタリングは実行できません。
- スクリプトファイルの内容が更新されている場合は、更新された内容でモニタリングを実行しますが、ファイルへの保存は行いません。

### 3.2.9 スクリプトを実行する

スクリプトファイルを最後まで実行します。モニタリングではなく、通常のスクリプト実行です。

操作

1. [モニタリング] - [実行] を選択する。  
スクリプトの実行が始まります。モニタリングではないので、次に実行される行を示す (▶) は表示されません。  
スクリプトの最後の行まで実行すると、実行が終了します。
- モニタリングではないので、[表示] - [ウォッチウィンドウ] メニューがチェックされていても、実行中にウォッチウィンドウは表示されません。

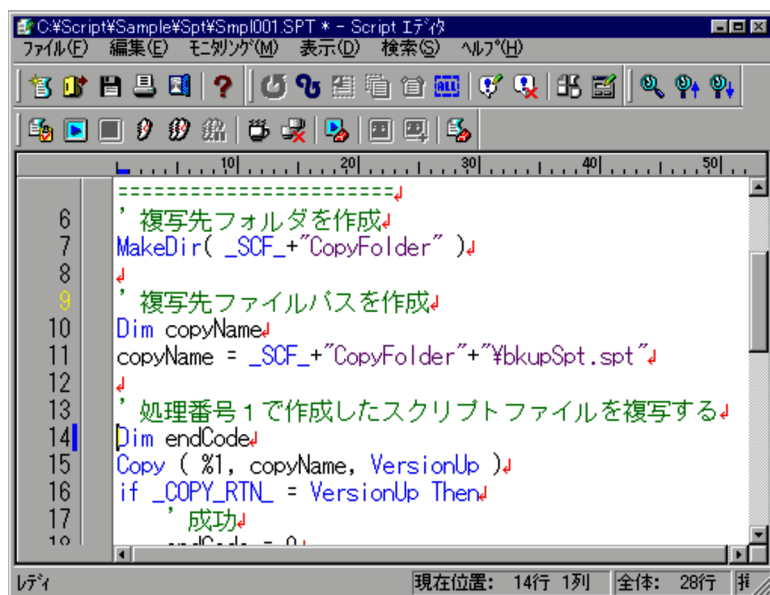
### 3.2.10 モニタリング実行時のブレークポイントを設定・解除する


モニタリングの実行時に、一時的に実行を停止させる位置を設定、または解除します。

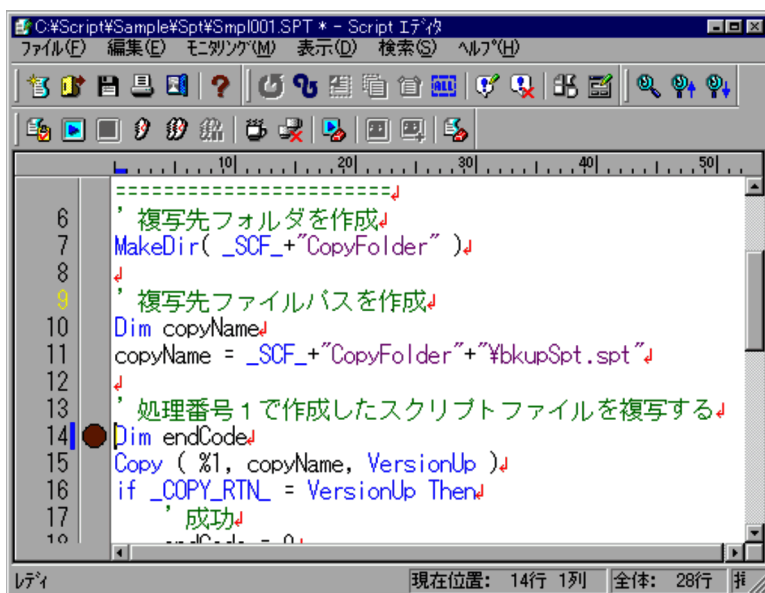
操作

ブレークポイントを設定する

1. ブレークポイントを設定したい行にカーソルを移動させる。

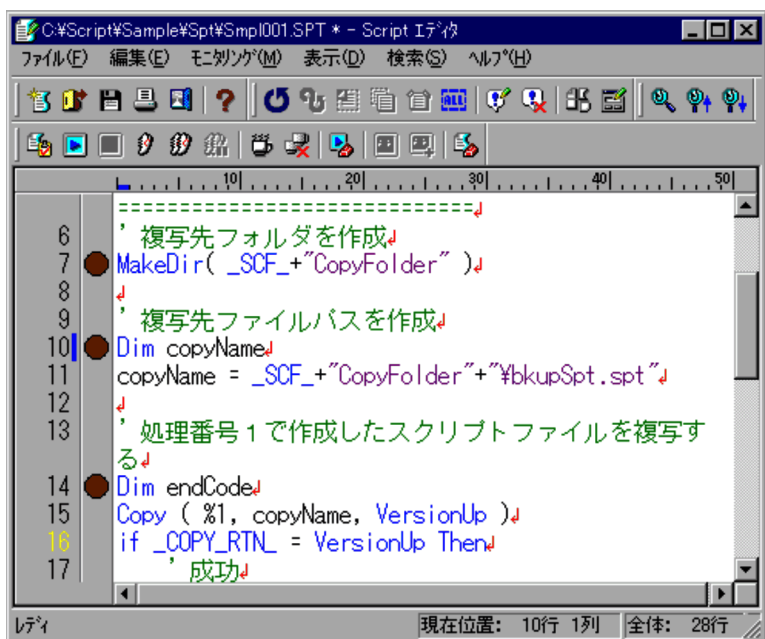


2. [ モニタリング ] - [ ブレークポイントの設定 ] メニューを選択する。  
カーソルのある行にブレークポイントが設定されます。ブレークポイントは、行の左側に  で表示されます。



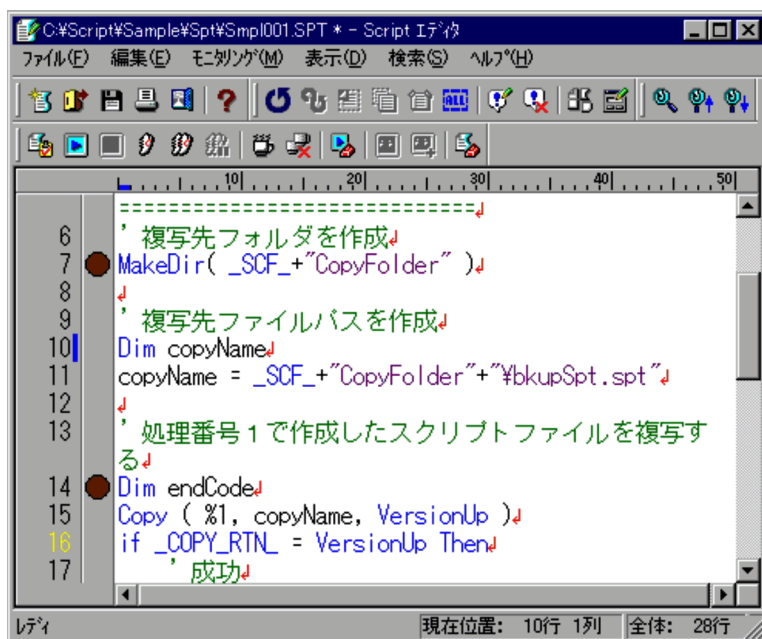
一部のブレークポイントを解除する

1. ブレークポイントを解除したい行にカーソルを移動させる。



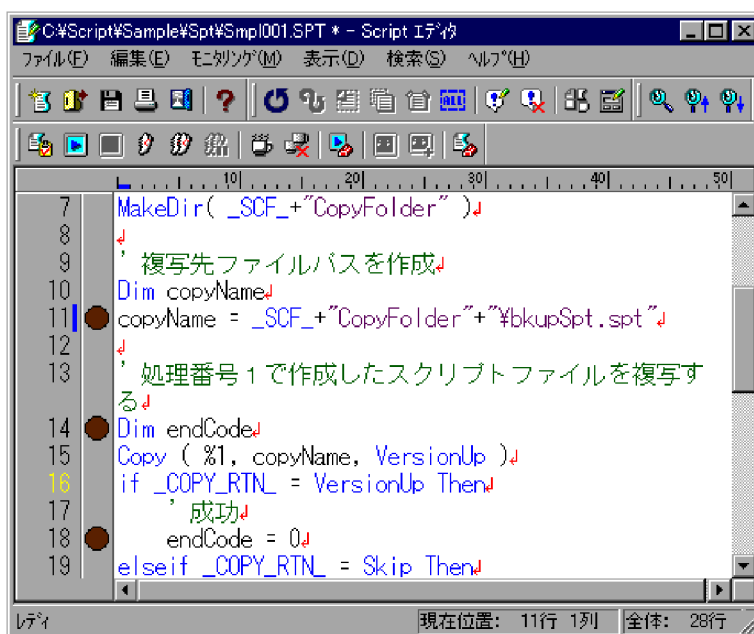
2. [ モニタリング ] - [ ブレークポイントの解除 ] メニューを選択する。  
カーソルのある行のブレークポイントが解除されます。

### 3. JP1/Script の操作



すべてのブレークポイントを解除する

1. [ モニタリング ] - [ ブレークポイントをすべて解除 ] メニューを選択する。

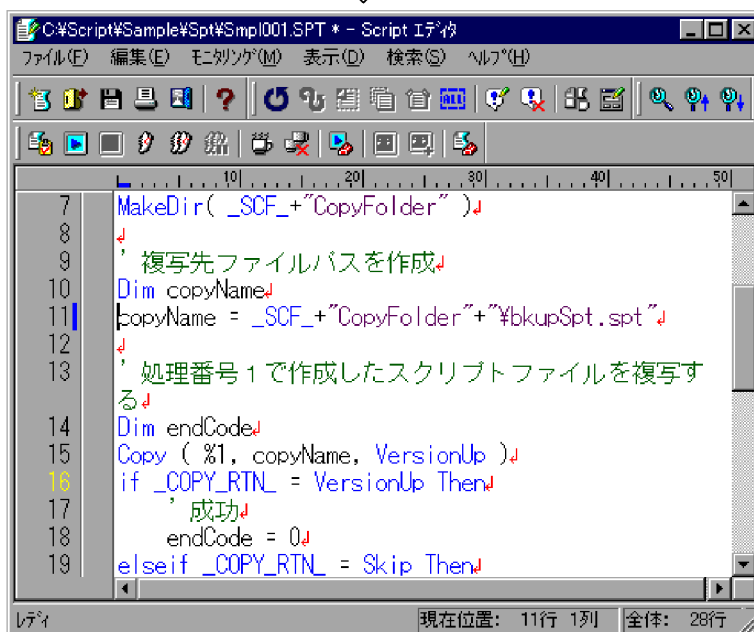


```

7  MakeDir( _SCF_+"CopyFolder" )↓
8
9  ' 複写先ファイルパスを作成↓
10 Dim copyName↓
11 copyName = _SCF_+"CopyFolder"+"¥bkupSpt.spt"↓
12
13 ' 処理番号 1 で作成したスクリプトファイルを複写す
   る↓
14 Dim endCode↓
15 Copy ( %1, copyName, VersionUp )↓
16 if _COPY_RTNL = VersionUp Then↓
17     ' 成功↓
18     endCode = 0↓
19 elseif _COPY_RTNL = Skip Then↓

```

現在位置: 11行 1列 全体: 28行

```

7  MakeDir( _SCF_+"CopyFolder" )↓
8
9  ' 複写先ファイルパスを作成↓
10 Dim copyName↓
11 copyName = _SCF_+"CopyFolder"+"¥bkupSpt.spt"↓
12
13 ' 処理番号 1 で作成したスクリプトファイルを複写す
   る↓
14 Dim endCode↓
15 Copy ( %1, copyName, VersionUp )↓
16 if _COPY_RTNL = VersionUp Then↓
17     ' 成功↓
18     endCode = 0↓
19 elseif _COPY_RTNL = Skip Then↓

```

現在位置: 11行 1列 全体: 28行

表示されているスクリプトファイルのブレークポイントがすべて解除されます。

- 編集モードの場合は、任意の行にブレークポイントを設定できます。モニタリングモードの場合、ブレークポイントを設定できるのは、実行対象となっているコマンド、またはステートメント単位に限られます。
- コメント行やスペース行にブレークポイントが設定されている場合は、エディタが自

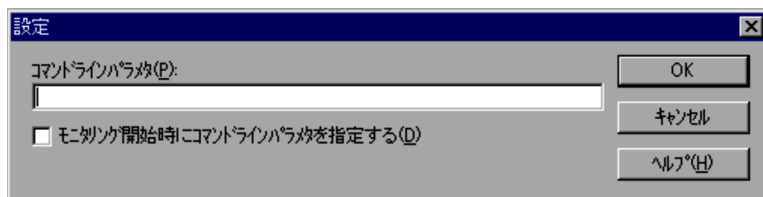
動的に下方向に適切な位置を検索して、そこにブレークポイントを設定します。適切な位置が見つからなかった場合、ブレークポイントは設定されないで、スクリプトファイルの先頭行で実行が一時的に停止されます。

### 3.2.11 モニタリング実行時の動作環境を設定する

モニタリング実行時の動作環境を設定します。

操作

1. [ モニタリング ] - [ 設定 ] メニューを選択する。  
[ 設定 ] ダイアログが表示されます。



ダイアログの設定方法については、「4.2.5 設定ダイアログ」を参照してください。

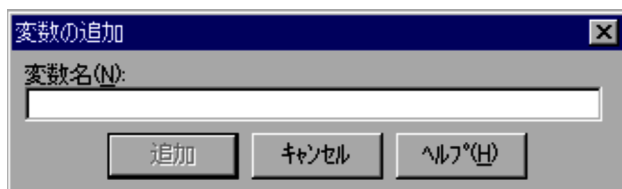
2. 必要な情報を入力する。  
モニタリング実行時に必要な情報を設定します。
3. [ OK ] ボタンを選択する。  
モニタリング時の動作環境が設定され、ダイアログが閉じます。

### 3.2.12 ウォッチへ変数を追加する

ウォッチウィンドウに指定した変数を追加します。

操作

1. モニタリングモードで、[ モニタリング ] - [ ウォッチへ変数の追加 ] メニューを選択する。  
[ 変数の追加 ] ダイアログが表示されます。

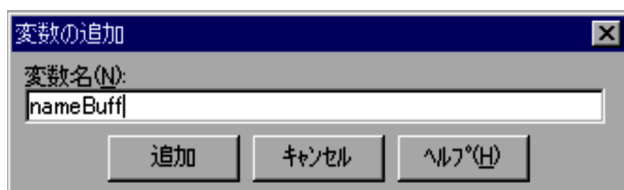


ダイアログの設定方法について、「4.2.10 変数の追加ダイアログ」を参照してください。

2. 追加したい変数の名前を入力する。  
入力できるのは半角で 99 文字までです。これを超えた変数名を入力することはでき

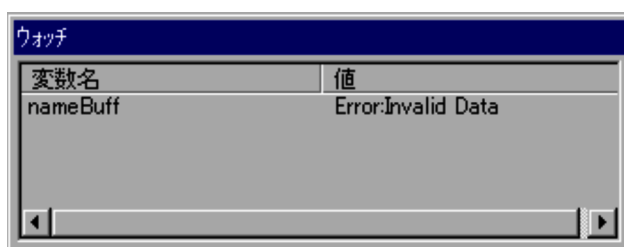


ません。



3. [ 追加 ] ボタンを選択する。

入力した変数の名前がウォッチウィンドウに追加されます。



4. [ キャンセル ] ボタンを選択する。

[ 変数の追加 ] ダイアログが閉じられます。

- モニタリングモードで、Script エディタウィンドウのクライアントエリアで文字列を選択してから [ モニタリング ] - [ ウォッチへ変数の追加 ] メニューを選択すると、選択した文字列が「変数名」に表示されます。
- 追加した変数をウォッチウィンドウから削除する場合は、変数名を右クリックし、[ 変数名の削除 ] メニューを選択します。

### 3.2.13 スクリプトの実行環境を設定する

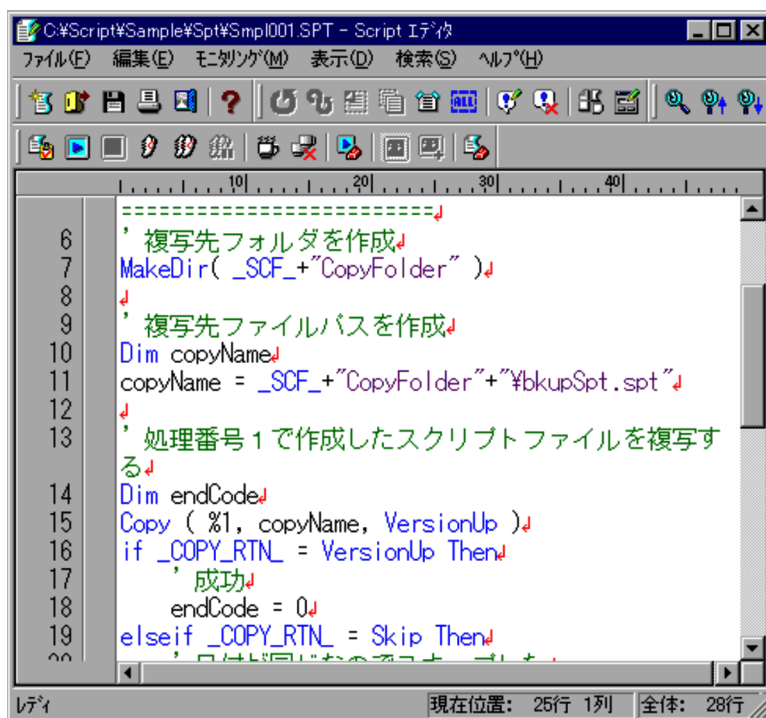
作成されているスクリプトファイルの実行環境を設定します。このほかに、Script マネージャウィンドウから操作する方法もあります。

操作

1. 実行環境を設定するスクリプトファイルをクライアントエリアに表示する。  
表示する方法には次の三つがあります。

- [ ファイル ] - [ 開く ] メニューで既存のスクリプトファイルを開く。
- 無題で編集集中だったスクリプトを保存する。
- マネージャの新規作成、編集でエディタを起動した状態で、エクスプローラからアイコンをドラッグ&ドロップする。

### 3. JP1/Script の操作



2. [ モニタリング ] - [ 実行環境の設定 ] メニューを選択する。  
[ 実行環境の設定 ( 起動情報 ) ] ダイアログが表示されます。



ダイアログの設定方法については、「4.1.8 実行環境の設定（起動情報）ダイアログ」を参照してください。

3. 必要な起動情報を設定する。

実行時に位置変数が必要な場合、および解析トレース、実行トレースの出力形式などを設定する場合は、コマンドラインでの指定が必要です。

コマンドラインでの指定については、「6.2 コマンドラインに関する規則」を参照してください。

4. [ 終了情報 ] タブをクリックする。

[ 実行環境の設定（終了情報）] ダイアログが表示されます。

### 3. JP1/Script の操作



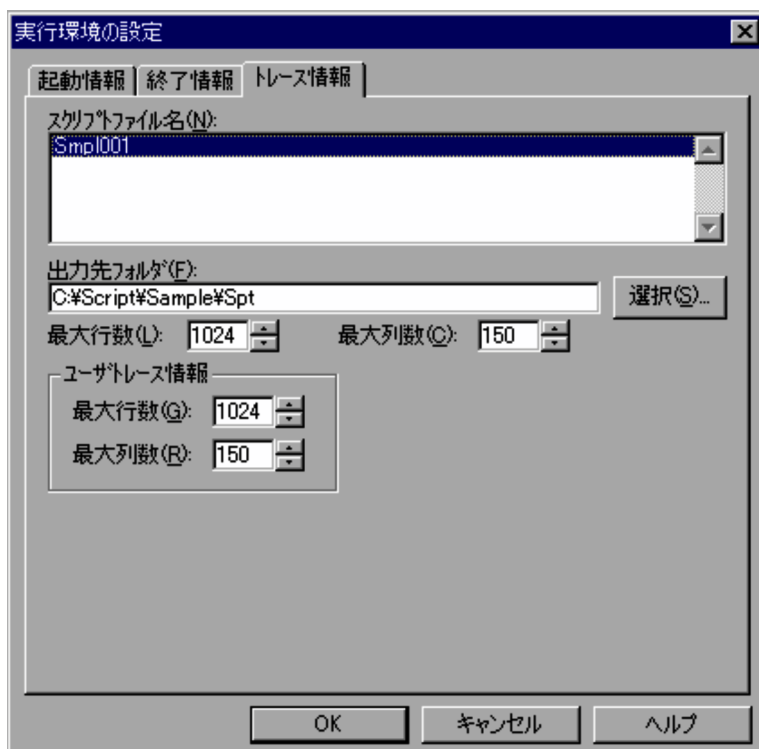
ダイアログの設定方法については、「4.1.9 実行環境の設定（終了情報）ダイアログ」を参照してください。

5. 打ち切り時間を設定する。

打ち切り時間による強制終了をしない場合は、打ち切り時間が「0」になっていることを確認してください。

6. [トレース情報] タブをクリックする。

[実行環境の設定（トレース情報）] ダイアログが表示されます。



ダイアログの設定方法については、「4.1.10 実行環境の設定（トレース情報）ダイアログ」を参照してください。

7. 必要なトレース情報を設定する。  
トレースファイルの出力行数などを特に指定する必要がない場合は、このダイアログでの操作は必要ありません。
8. [ OK ] ボタンをクリックする。  
実行環境が設定され、ダイアログが閉じます。

#### 補足

Windows Vista では、JP1/Script サービスが実行中以外の状態でスクリプトの実行環境を設定しようとすると、次のエラーダイアログを表示します。

「JP1/Script サービスの状態が実行中以外のため、実行できません。JP1/Script サービスを開始した後、再度実行してください。」

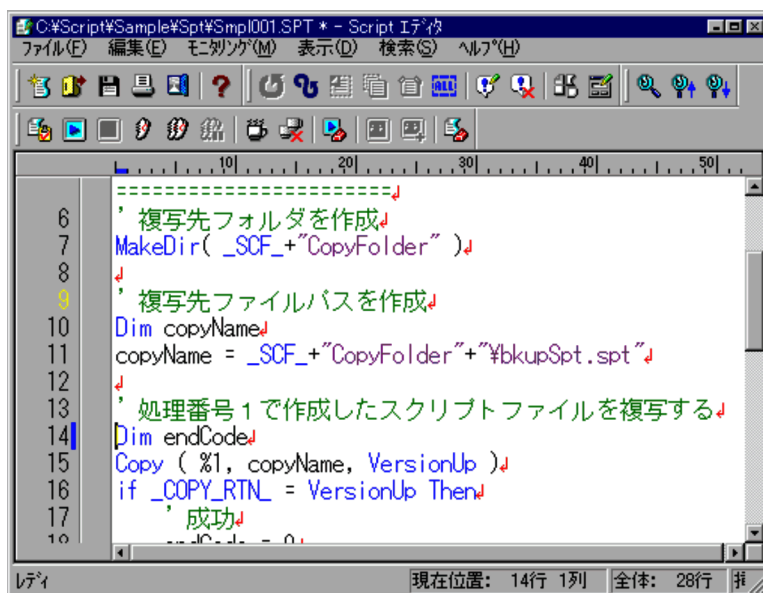
### 3.2.14 文字列を検索する

スクリプトファイル中の文字列を検索します。

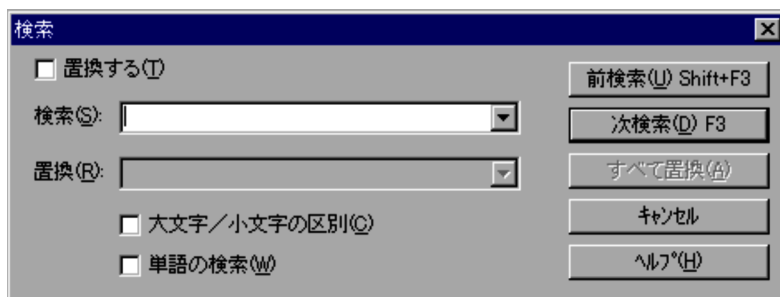
#### 操作

1. 検索するスクリプトファイルを表示する。

### 3. JP1/Script の操作

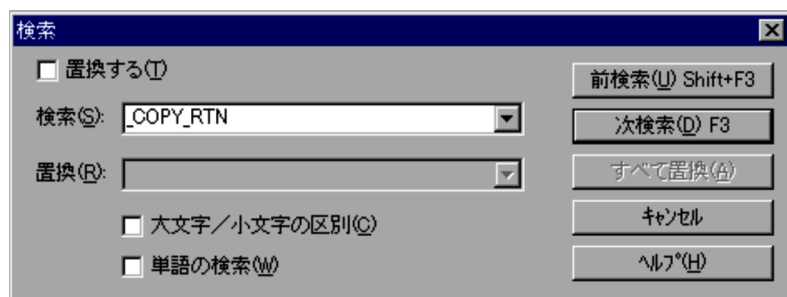


2. [ 検索 ] - [ 検索 ] メニューを選択する。  
[ 検索 ] ダイアログが表示されます。

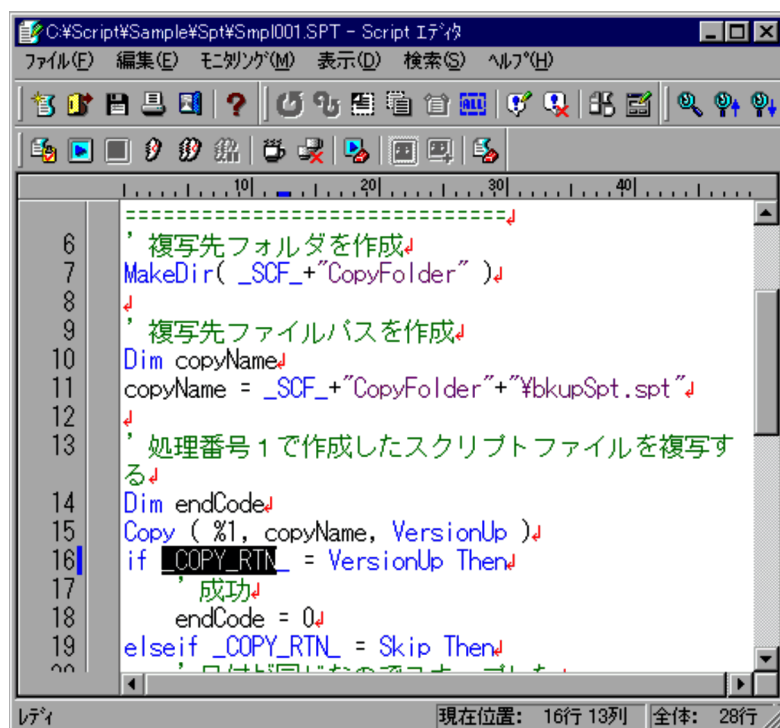


ダイアログの設定方法については、「4.2.6 検索ダイアログ」を参照してください。

3. [ 置換する ] がチェックされていないことを確認する。  
[ 置換する ] がチェックされている場合は、チェックを外してください。
4. [ 検索 ] に検索文字列を指定する。また、必要に応じて [ 大文字 / 小文字の区別 ], および [ 単語の検索 ] をチェックする。



5. [ 前検索 ] ボタン, または [ 次検索 ] ボタンを選択する。  
 検索文字列が検索されます。検索文字列がない場合は, ビープ音が鳴ります。



6. 検索を終了する場合は, [ キャンセル ] ボタンを選択する。  
 [ 検索 ] ダイアログが閉じられます。

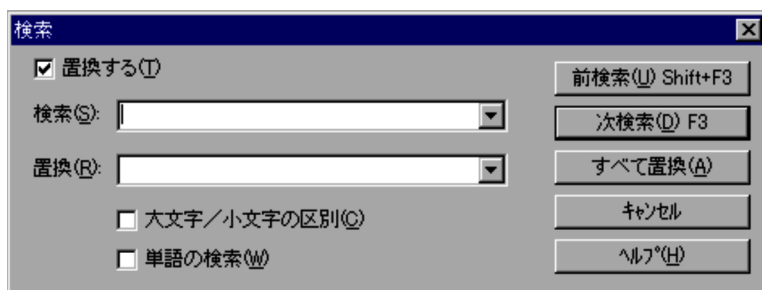
### 3.2.15 文字列を置換する

スクリプトファイル中の文字列を置換します。

操作

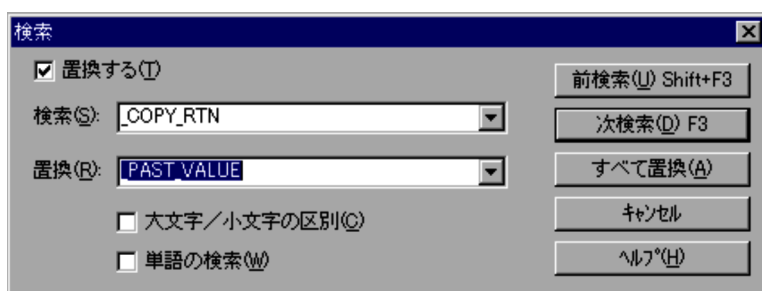
1. 置換するスクリプトファイルを表示する。
2. [ 検索 ] - [ 置換 ] メニューを選択する。

[ 検索 ] ダイアログが表示されます。



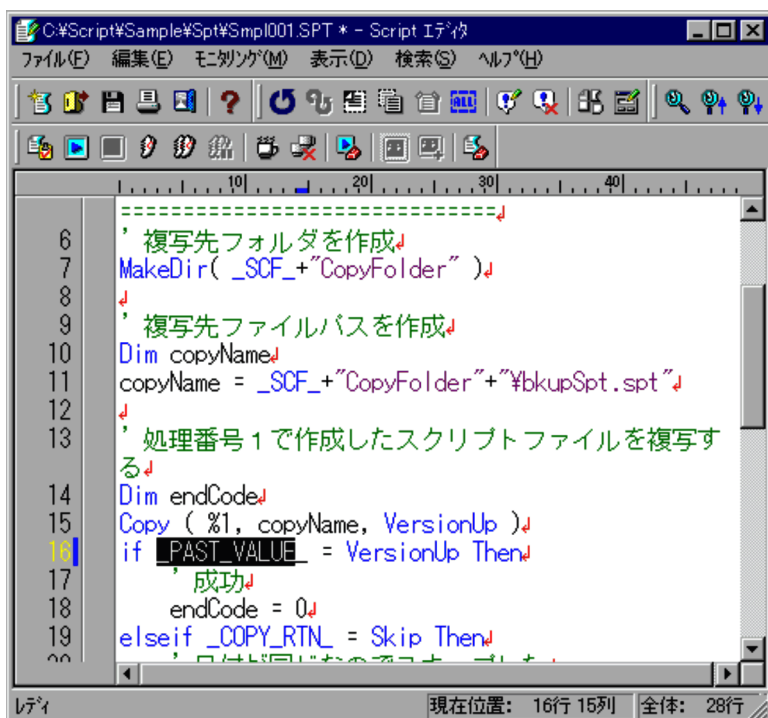
ダイアログの設定方法については、「4.2.6 検索ダイアログ」を参照してください。

3. [ 置換する ] がチェックされていることを確認する。  
[ 置換する ] がチェックされていない場合は、チェックしてください。
4. [ 検索 ] に置換前の文字列を指定する。また、必要に応じて [ 大文字 / 小文字の区別 ], および [ 単語の検索 ] をチェックする。



5. [ 置換 ] に置換後の文字列を指定する。
6. [ 前検索 ] ボタン, または [ 次検索 ] ボタンを選択する。  
指定した内容で置換が始まります。置換すべき文字列がない場合は、ピープ音が鳴ります。





7. 置換を終了する場合は、[ キャンセル ] ボタンを選択する。  
[ 検索 ] ダイアログが閉じられます。

## 3.3 簡易入力操作

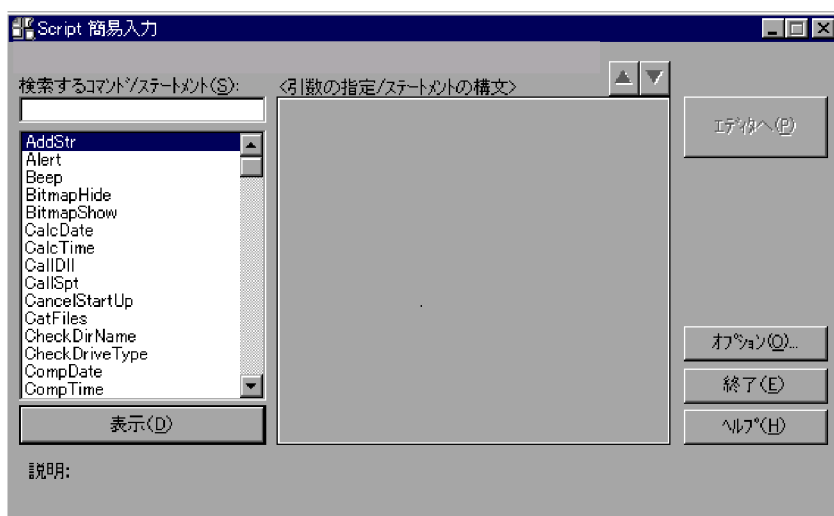
コマンドやステートメントの入力作業を簡略化して入力できます。この機能を使うことで、コマンドの知識がなくても手軽に間違いなく入力できます。また、記述したコマンドやステートメントの内容は、クリップボードに貼り付けられます。

### 3.3.1 簡易入力の操作項目

簡易入力アイコンを起動する、エディタの [ 編集 ] - [ 簡易入力 ] を選択する、またはマネージャの [ ツール ] - [ 簡易入力の起動 ] を選択すると、[ Script 簡易入力 ] ダイアログが表示されます。

[ Script 簡易入力 ] ダイアログを、図 3-3 に示します。

図 3-3 [ Script 簡易入力 ] ダイアログ



#### (a) 処理対象バージョン

現在指定されているスクリプトエンジンのバージョンが表示されます。「検索するコマンド/ステートメント」の一覧 (リストボックス) には、指定されているスクリプトエンジンのバージョンに対応する、コマンド、およびステートメントが表示されます。

#### (b) 検索するコマンド/ステートメント

一覧 (リストボックス) には、指定されているスクリプトエンジンのバージョンに対応する、コマンド、およびステートメントが表示されます。選択したいコマンド、およびステートメントが表示されていない場合は、コマンド名、ステートメント名、またはコマンド名、ステートメント名の一部を入力して検索してください。

## (c) 説明

「検索するコマンド / ステートメント」で選択されたコマンド、またはステートメントの簡単な説明が表示されます。

## (d) 対象 OS

現在選択しているコマンド、またはステートメントをサポートしている OS が表示されます。

## (e) 引数の指定 / ステートメントの構文

「検索するコマンド / ステートメント」で選択されたコマンドの引数を入力する項目が表示されます。ステートメントの場合は、ステートメントの構文が表示されます。

[ Script 簡易入力 ] ダイアログには、次のボタンが表示されます。

## [ 表示 ] ボタン

「引数の指定 / ステートメントの構文」に、選択されたコマンドの引数を入力する項目、またはステートメントの構文が表示されます。ただし、選択されたコマンドの引数がない場合は、「このコマンドには引数はありません。」が表示されます。

## [ クリップボードへ ] ボタン (または [ エディタへ ] ボタン)

簡易入力をエディタから起動した場合は [ エディタへ ]、マネージャ、または簡易入力アイコンから起動した場合は [ クリップボードへ ] と表示されます。

「引数の指定 / ステートメントの構文」に表示された内容が、エディタ、またはクリップボードに貼り付けられます。

## [ オプション ] ボタン

[ 簡易入力オプション (全般情報) 設定 ] ダイアログが表示されます。ダイアログの設定方法については、「3.3.3 JP1/Script の対象バージョンを指定する」を参照してください。

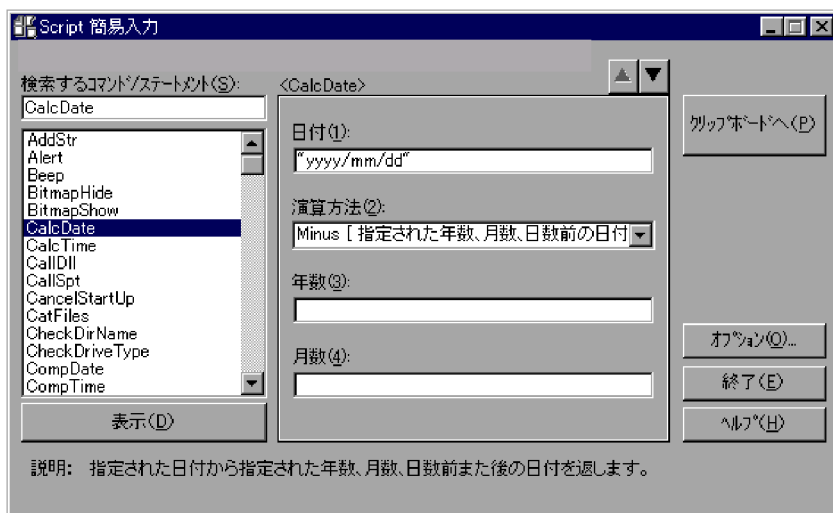
## [ 終了 ] ボタン

[ Script 簡易入力 ] ダイアログを閉じます。

### 3.3.2 コマンドやステートメントを入力する

スクリプトを記述する際の、コマンドやステートメントの入力作業を、このダイアログで行います。ここでは、CalcDate の表示例を示します。

### 3. JP1/Script の操作





### (1) 項目

#### 対象バージョン

スクリプトファイルを JP1/Script のどのバージョンで作成するかを指定します。  
デフォルトではインストールした JP1/Script のバージョンが設定されています。

### (2) 操作

- [ OK ] ボタンを選択すると、指定した内容が設定されて [ オプション ] ダイアログを閉じます。
- [ キャンセル ] ボタンを選択すると、内容を変更しないでダイアログを閉じます。

## 3.3.4 1 行の最大文字数やインデントの桁数を指定する

[ Script 簡易入力 ] ダイアログで [ オプション ] ボタンを選択すると、[ オプション ] ダイアログが表示されます。

[ オプション ] ダイアログには、「全般」「編集」の二つのタブがあります。

「編集」タブを選択すると、[ オプション (編集) ] ダイアログが表示されます。



## (1) 項目

### 1 桁の最大桁数

Script エディタ，またはクリップボードに貼り付ける編集内容の 1 行の最大数を指定します。

64 ～ 999 の範囲で指定してください。デフォルトでは「80」が設定されています。

### インデント桁数

クリップボードに貼り付ける編集内容のインデントの値を指定します。

0 ～ 16 の範囲で指定してください。デフォルトでは「4」が設定されています。

### 必須項目をチェックする

コマンドの必ず入力しなければならない引数の項目に，正しくデータが入力されているかどうかをチェックします。

デフォルトではチェックされていません。

## (2) 操作

- [ OK ] ボタンを選択すると，指定した内容が設定されて [ オプション ] ダイアログを閉じます。
- [ キャンセル ] ボタンを選択すると，内容を変更しないでダイアログを閉じます。

## 3.4 トレースビューア の操作

トレースビューアは、マネージャで作成したスクリプトの実行状態、およびトレース情報を表示するためのプログラムです。スクリプトの実行状態、およびトレース情報は、トレースファイルに出力されます。

### 補足

Windows Vista では、JP1/Script サービスが実行中以外の状態でトレースビューアを起動しようとすると、次の警告ダイアログを表示した後、トレースビューアの起動を中止します。

「JP1/Script サービスの状態が実行中以外のため、実行できません。JP1/Script サービスを開始した後、再度実行してください。」

### 3.4.1 Script トレースビューアウィンドウとメニュー

トレースビューアを起動すると表示される Script トレースビューアウィンドウ、およびトレースビューアのメニューの種類について説明します。また、トレースファイルを開いたときに表示されるウィンドウ、およびメニューの種類についても説明します。

#### (1) Script トレースビューアウィンドウ

Script トレースビューアウィンドウ、およびウィンドウの各部の名称を、図 3-4 に示します。

図 3-4 Script トレースビューアウィンドウ



#### (a) ツールバー

ツールバーには、メニューバーから選択できるコマンドの中から、頻繁に使用するコマンドだけをボタンの形で表示します。ボタンをマウスでクリックするだけで、該当するコマンドを実行できます。オプションメニューによって非表示にすることもできます。

### 3. JP1/Script の操作

Script トレースビューアウィンドウのツールバーには、次のコマンドのボタンが表示されます。

[ 保存 ] ボタン

トレースファイルを別名で保存します。

[ 表示 ] ボタン

トレースファイルの内容を表示します。

[ 削除 ] ボタン

トレースファイルを削除します。

[ 印刷 ] ボタン

トレースファイルを印刷します。

[ 終了 ] ボタン

トレースビューアを終了します。

[ クリア ] ボタン

トレースファイルを空 ( ファイルサイズ 0 ) にします。

[ すべて選択 ] ボタン

すべての項目を選択します。

[ 解析トレース ] ボタン

解析トレース一覧の表示・非表示を切り替えます。

[ 実行トレース ] ボタン

実行トレース一覧の表示・非表示を切り替えます。

[ ユーザトレース ] ボタン

ユーザトレース一覧の表示・非表示を切り替えます。

[ サーバトレース ] ボタン

サーバトレース一覧の表示・非表示を切り替えます。

[ 最新 ] ボタン

表示しているトレースファイルの情報を最新の情報に更新します。

[ コンピュータの選択 ] ボタン

表示するトレースファイルの接続先コンピュータ名を設定します。

[ ヘルプ ] ボタン

JP1/Script のオンラインヘルプを表示します。

(b) ステータスバー









ステータスバーは、トレースビューアが現在実行している処理に関するメッセージや、処理終了後の状態に関するメッセージを表示するための領域です。



## (c) クライアントエリア



ここには、トレースファイル情報（種別、ファイル名、行数、更新日時、格納先フォルダ）が表示されます。トレースファイル情報の左端には、各トレースファイルの実行状態がアイコンで表示されます。表示されるアイコンは、複数起動を許可されているスクリプトの場合と、複数起動を許可されていないスクリプトの場合とで異なります。各アイコンの見方を次に示します。





表 3-11 複数起動を許可されているスクリプトの場合

表示	種別	意味
 (白, 黄緑)	実行中 (一部終了)	一つ以上のスクリプトが実行中であることを表します。
 (白, 青)	正常終了 (一部終了)	一つ以上のスクリプトが正常に終了したことを表します。
 (白, 赤)	システムエラー (一部終了)	一つ以上のスクリプトのコマンド実行中にエラーが発生し、スクリプトの実行を中断したことを表します。
 (白, 黄)	ユーザエラー (一部終了)	一つ以上のスクリプトで Alert コマンド、または ExitWindows コマンドを実行したこと、または強制的に終了されたことを表します (Alert コマンド実行後にシステムエラーになる場合もあります)。
 (水色, 青)	正常終了 (すべて終了)	すべてのスクリプトが正常に終了したことを表します。
 (水色, 黒)	未実行 (すべて終了)	すべてのスクリプトが 1 回も実行されていないことを表します (構文解析などのエラーでコマンドが実行されなかった場合などに表示されます)。
 (水色, 赤)	システムエラー (すべて終了)	すべてのスクリプトが終了しましたが、一つ以上のスクリプトでコマンド実行中にエラーが発生し、スクリプト実行を中断したことを表します。
 (水色, 黄)	ユーザエラー (すべて終了)	すべてのスクリプトが終了しましたが、一つ以上のスクリプトで Alert コマンド、または ExitWindows コマンドを実行したこと、または強制的に終了されたことを表します (Alert コマンド実行後にシステムエラーになる場合もあります)。

注 表示欄の括弧内はアイコンの丸印の色を表します。

表 3-12 複数起動を許可されていないスクリプトの場合

表示	種別	意味
 (黒)	管理対象外	実行状態を管理しないファイル (解析トレース、ユーザトレース) であることを表します。
 (黒)	未実行	スクリプトが 1 回も実行されていないことを表します。

表示	種別	意味
 (黄緑)	実行中	スクリプトが実行中であることを表します。
 (青)	正常終了	スクリプトが正常に終了したことを表します。
 (赤)	異常終了 (システムエラー)	スクリプトのコマンド実行中にエラーが発生し、スクリプトの実行を中断したことを表します。
 (黄)	異常終了 (ユーザエラー)	Alert コマンド、または ExitWindows コマンドを実行したこと、または強制的に終了されたことを表します (Alert コマンド実行後にシステムエラーになる場合もあります。)

注 表示欄の括弧内はアイコンの丸印の色を表します。

## (2) Script トレースビューアウィンドウのメニュー

トレースビューアの機能をメニューごとに説明します。トレースビューアのメニュー一覧を表 3-13 に示します。

表 3-13 トレースビューアのメニュー一覧

メニュー	コマンド (機能)	機能概要
[ ファイル ]	[ 名前を付けて保存 ]	選択されているトレースファイルに別の名前を付けて保存します。
	[ トレースファイルの表示 ]	選択されているトレースファイルの内容を表示します。
	[ トレースファイルの削除 ]	選択されているトレースファイルを削除します。
	[ 印刷 ]	選択されているトレースファイルの内容を印刷します。
	[ トレースビューアの終了 ]	トレースビューアを終了します。
[ 編集 ]	[ トレースファイルのクリア ]	トレースファイルを空 (ファイルサイズ 0) にします。
	[ すべて選択 ]	すべての項目を選択します。
[ 表示 ]	[ ツールバー ]	ツールバーの表示・非表示を切り替えます。
	[ ステータスバー ]	ステータスバーの表示・非表示を切り替えます。
	[ 未使用時に最小化 ]	未使用のとき、最小化します。
	[ 解析トレース ]	解析トレース一覧の表示・非表示を切り替えます。
	[ 実行トレース ]	実行トレース一覧の表示・非表示を切り替えます。
	[ ユーザトレース ]	ユーザトレース一覧の表示・非表示を切り替えます。
	[ サーバトレース ]	サーバトレース一覧の表示・非表示を切り替えます。

メニュー	コマンド（機能）	機能概要
	[ 最新の情報に更新 ]	トレースファイルの情報を最新の情報に更新します。
[ オプション ]	[ コンピュータの選択 ]	表示するトレースファイルの接続先コンピュータ名を設定します。
[ ヘルプ ]	[ 目次 ]	JP1/Script オンラインヘルプの目次を表示します。
	[ キーワードで検索 ]	JP1/Script オンラインヘルプのキーワード一覧を表示します。
	[ ヘルプの使い方 ]	オンラインヘルプの使い方を表示します。
	[ バージョン情報 ]	トレースビューアのバージョン情報を表示します。

### 3.4.2 Script トレースビューアウィンドウでのマウスとキーの操作

Script トレースビューアウィンドウでのマウス操作とキーの操作について説明します。

#### （１）マウス操作

Script トレースビューアウィンドウのクライアントエリアでのマウス操作を表 3-14 に示します。

表 3-14 Script トレースビューアウィンドウでのマウス操作

操作	処理
クリック	それまでの選択を解除して、新たに対象を選択する。
ダブルクリック	トレースファイルの内容表示ウィンドウを表示する。

#### （２）キー操作

Script トレースビューアウィンドウのクライアントエリアでのキー操作を表 3-15 に示します。

表 3-15 Script トレースビューアウィンドウでのキー操作

操作	処理
Ctrl + A	すべてのトレースファイルを選択する。
Ctrl + P	トレースファイルを印刷する。
Del	トレースファイルを削除する。
Enter	トレースファイルの内容表示ウィンドウを表示する。

### 3.4.3 トレースビューアの起動と終了

#### (1) トレースビューアを起動する

スクリプトファイルの実行時に出力されるトレースは、トレースファイルに出力されます。トレースの状態を知りたいときは、トレースビューアを起動します。

次の2種類の起動方法があります。

##### (a) スタートメニューから起動する方法

操作

1. [ スタート ] から [ プログラム ] - [ JP1/Script ] を選択し、JP1/Script のグループからトレースビューアを選択する。  
Script トレースビューアウィンドウが表示されます。ウィンドウにはトレースファイルが一覧表示されています。

補足

Windows Vista では、JP1/Script サービスが実行中以外の状態でトレースビューアを起動しようとすると、次のエラーダイアログを表示します。

「JP1/Script サービスの状態が実行中以外のため、実行できません。JP1/Script サービスを開始した後、再度実行してください。」

##### (b) Script マネージャウィンドウから起動する方法

操作

1. Script マネージャウィンドウで、[ ツール ] - [ トレースビューアの起動 ] を選択する。  
トレースビューアを起動できます。

補足

Windows Vista では、JP1/Script サービスが実行中以外の状態でトレースビューアを起動しようとすると、次のエラーダイアログを表示します。

「JP1/Script サービスの状態が実行中以外のため、実行できません。JP1/Script サービスを開始した後、再度実行してください。」

#### (2) トレースビューアを終了する

[ ファイル ] - [ トレースビューアの終了 ] を選択してください。トレースビューアが終了して、Script トレースビューアウィンドウが閉じます。

### 3.4.4 Script トレースビューアウィンドウでの操作

Script トレースビューアウィンドウ（およびトレースファイルの表示ウィンドウ）でできる操作の一覧を次に示します。

- トレースファイルの表示・非表示を切り替える
  - トレースファイルの内容を表示する
  - トレースファイルを別名で保存する
  - トレースファイルを印刷する
  - トレースファイルを削除する
  - トレースファイルをクリアする
  - トレースファイルの内容を検索する
  - ツールバーの表示・非表示を切り替える
  - ステータスバーの表示・非表示を切り替える
  - 未使用時に最小化する
  - クライアントエリアを最新の情報に更新する
  - クライアントエリアに表示するトレースファイルの接続先コンピュータ名を設定する
- 
- トレースファイルの表示ウィンドウを閉じる
  - トレースファイルの行番号の表示・非表示を切り替える
  - トレースファイルの表示ウィンドウを縦（または横）に並べる

#### 注

これらの操作は特に説明していません（これらは Windows の標準の操作に準じているか、メニュー一覧で該当するメニューを選択するだけで実行できる操作です）。

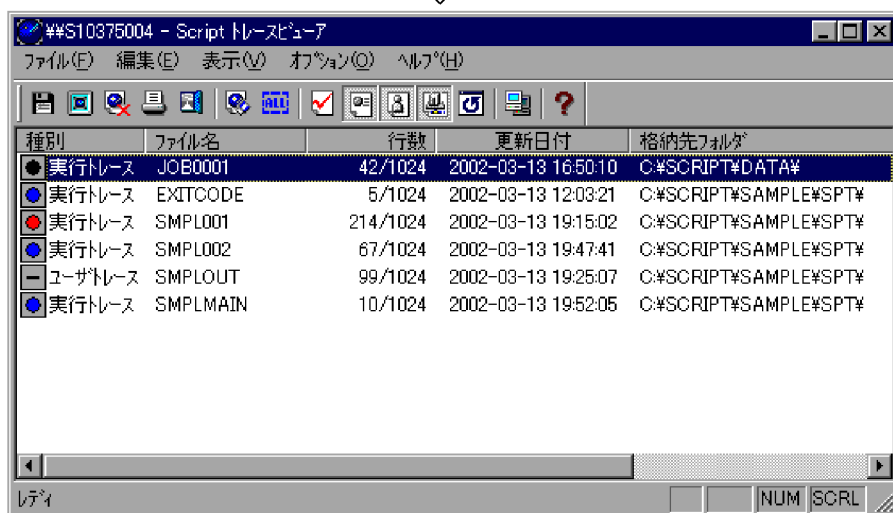
### 3.4.5 トレースファイルの表示・非表示を切り替える

Script トレースビューアウィンドウに一覧表示される実行トレースファイル、解析トレースファイル、ユーザトレースファイル、サーバトレースファイルの表示・非表示を切り替えます。例えば、スクリプトファイルが完成し、業務で実行する場合には、解析トレースは必要ないので非表示にします。

#### 操作

1. 実行トレースの表示・非表示を切り替える場合は、[ 表示 ] - [ 実行トレース ] メニューを選択する。
2. 解析トレースの表示・非表示を切り替える場合は、[ 表示 ] - [ 解析トレース ] メニューを選択する。
3. ユーザトレースの表示・非表示を切り替える場合は、[ 表示 ] - [ ユーザトレース ] メニューを選択する。
4. サーバトレースの表示・非表示を切り替える場合は、[ 表示 ] - [ サーバトレース ] メニューを選択する。

解析トレースの表示・非表示の切り替え例

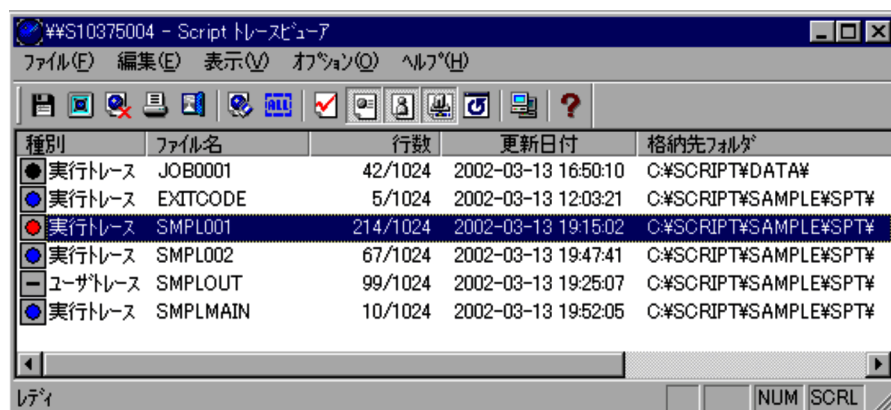


### 3.4.6 トレースファイルの内容を表示する

Script トレースビューアウィンドウで選択したトレースファイルの内容を、トレースファイルの表示ウィンドウに表示します。一度に複数のファイルを選択することもできます。

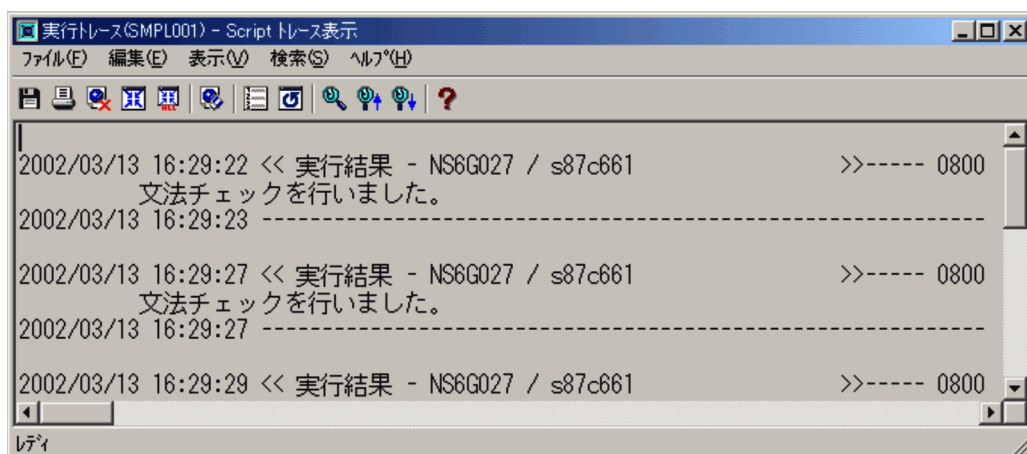
操作

1. Script トレースビューアウィンドウで、トレースファイルの一つ、または複数選択する。



2. [ファイル] - [トレースファイルの表示] メニューを選択する。

Script トレース表示ウィンドウに、トレースファイルの内容が表示されます。



3. Script トレース表示ウィンドウで、[ファイル] - [表示ウィンドウを閉じる]、または [ファイル] - [全ての表示ウィンドウを閉じる] を選択する。

Script トレース表示ウィンドウが閉じて、Script トレースビューアウィンドウに戻ります。

- Script トレースビューアウィンドウで、[表示] - [未使用時に最小化] メニューが選択されている場合は、操作 2. でトレースファイルの表示ウィンドウが表示されたときに、Script トレースビューアウィンドウが最小化します。

### 3.4.7 トレースファイルを別名で保存する

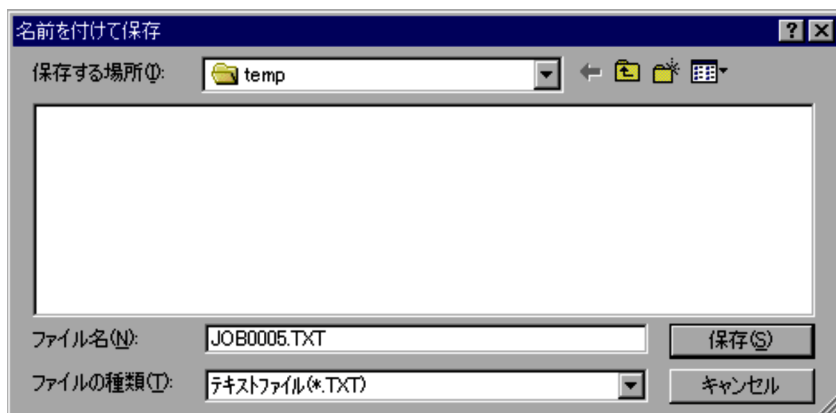
選択したトレースファイルを別名で保存します。

操作

- Script トレースビューアウィンドウのクライアントエリアから、別名で保存したいトレースファイルを一つ選択する。または、トレースファイルの表示ウィンドウに、別名で保存したいファイルが表示されていることを確認する。



- [ファイル] - [名前を付けて保存] メニューを選択する。表示されるダイアログで名前を指定し、[保存] ボタンをクリックする。



選択したトレースファイルが指定した名前で保存されます。

- 操作 2. で [キャンセル] ボタンをクリックすると、別名での保存を中断してダイアログを閉じます。

### 3.4.8 トレースファイルを削除する

Script トレースビューアウィンドウで選択されているトレースファイルを削除します。またはトレースファイルの表示ウィンドウに表示されているトレースファイルを削除します。

操作

- Script トレースビューアウィンドウのクライアントエリアから、削除するトレース



ファイルの一つ，または複数選択する。または，トレースファイルの表示ウィンドウに，削除したいファイルが表示されていることを確認する。



2. [ファイル] - [トレースファイルの削除] メニューを選択する。  
[確認] ダイアログが表示されます。



ダイアログの設定方法については、「4.3.1 削除の確認ダイアログ」を参照してください。

3. [はい] ボタンをクリックする。  
トレースファイルが一つ削除され，次に選択されているファイルに処理が移ります（複数のファイルが選択されている場合）。



必要な回数だけ、操作を繰り返します。選択されているすべてのファイルが削除されると、ダイアログが閉じます。

- 操作 3. で [すべて] ボタンをクリックすると、選択されているすべてのファイルが削除されます。
- 操作 3. で [キャンセル] ボタンをクリックすると、ファイルの削除を中断してダイアログを閉じます。

### 3.4.9 トレースファイルをクリアする

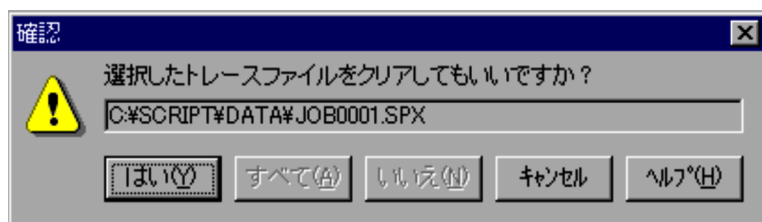
Script トレースビューアウィンドウで選択されているトレースファイルを空（ファイルサイズ 0）にします。または、トレースファイルの表示ウィンドウに表示されているトレースファイルを空にします。

操作

1. Script トレースビューアウィンドウのクライアントエリアから、クリアするトレースファイルの一つ、または複数選択する。または、トレースファイルの表示ウィンドウに、クリアしたいトレースファイルが表示されていることを確認する。



2. [編集] - [トレースファイルのクリア] メニューを選択する。  
[確認] ダイアログが表示されます。



ダイアログの設定方法については、「4.3.2 クリアの確認ダイアログ」を参照してください。

3. [ はい ] ボタンをクリックする。  
 トレースファイルが一つクリアされます。



複数のファイルが選択されている場合、次に選択されているファイルに処理が移ります。

必要な回数だけ、操作を繰り返します。選択されているすべてのファイルがクリアされると、ダイアログが閉じます。

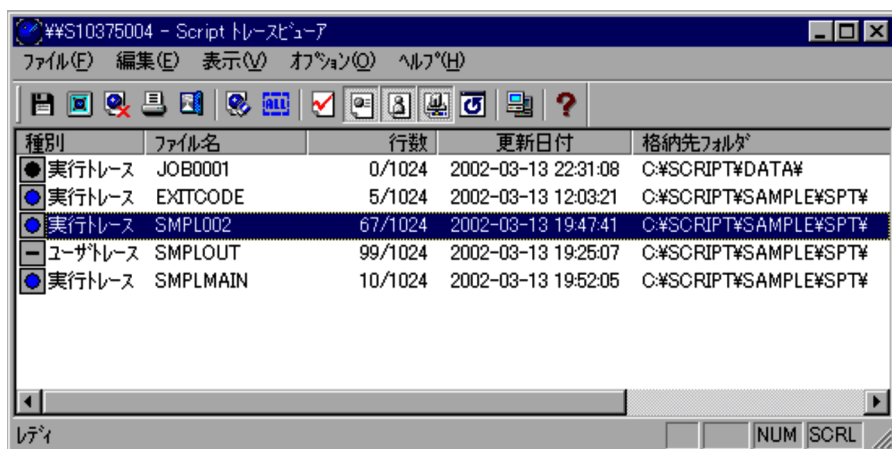
- 操作 3. で [ すべて ] ボタンをクリックすると、選択されているすべてのファイルがクリアされます。
- 操作 3. で [ キャンセル ] ボタンをクリックすると、ファイルのクリアを中断してダイアログを閉じます。

### 3.4.10 トレースファイルを印刷する

Script トレースビューアウィンドウで選択したトレースファイルの内容を印刷します。  
 または、トレースファイルの表示ウィンドウの内容を印刷します。

操作

1. Script トレースビューアウィンドウのクライアントエリアから、印刷するトレースファイルの一つを選択する。または、トレースファイルの表示ウィンドウに、印刷したいトレースファイルが表示されていることを確認する。



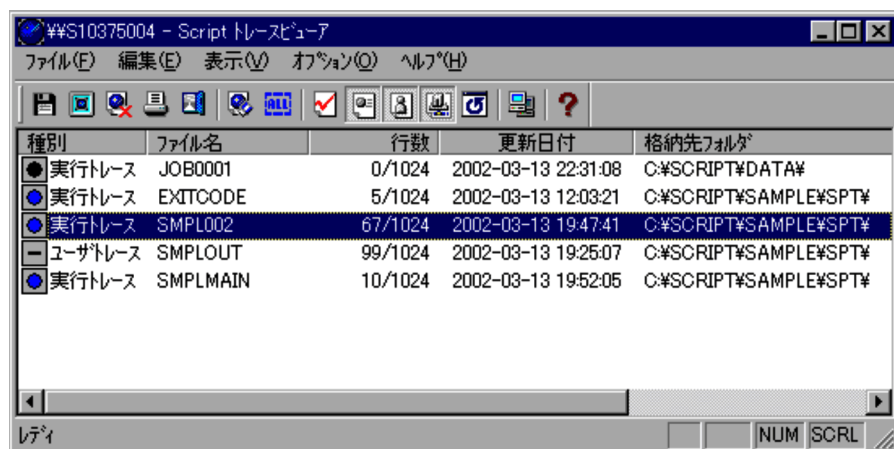
2. [ ファイル ] - [ 印刷 ] メニューを選択する。表示される [ 印刷 ] ダイアログに必要な情報を設定する。
  3. [ OK ] ボタンをクリックする。  
トレースファイルの内容が印刷されます。
- 操作 3. で [ キャンセル ] ボタンをクリックすると、印刷を中断してダイアログを閉じます。
  - タブ文字は、半角スペース 4 個分を仮定して印刷します。
  - 指定された用紙の範囲に収まらない場合は、折り返して印刷します。

### 3.4.11 トレースファイルの内容を検索する

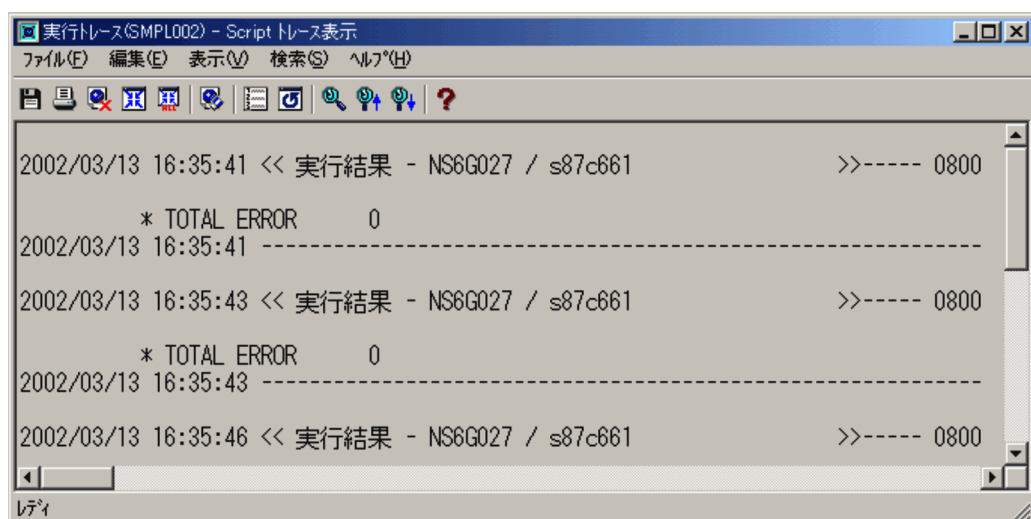
表示しているトレース情報から、指定した文字列を検索します。

操作

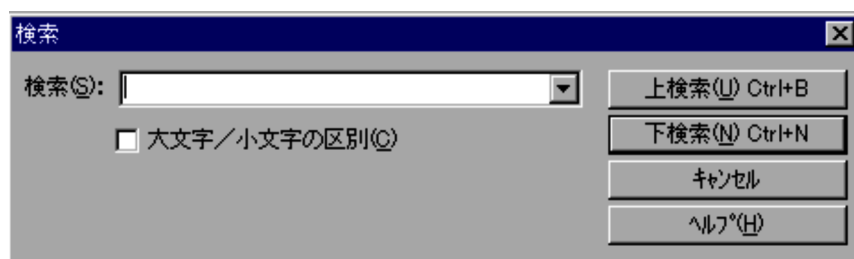
1. Script トレースビューアウィンドウのクライアントエリアから、検索するトレースファイルの一つを選択する。



2. [ファイル] - [トレースファイルの表示] メニューを選択し、トレースファイルを表示する。



3. [検索] - [検索] メニューを選択する。  
[検索] ダイアログが表示されます。

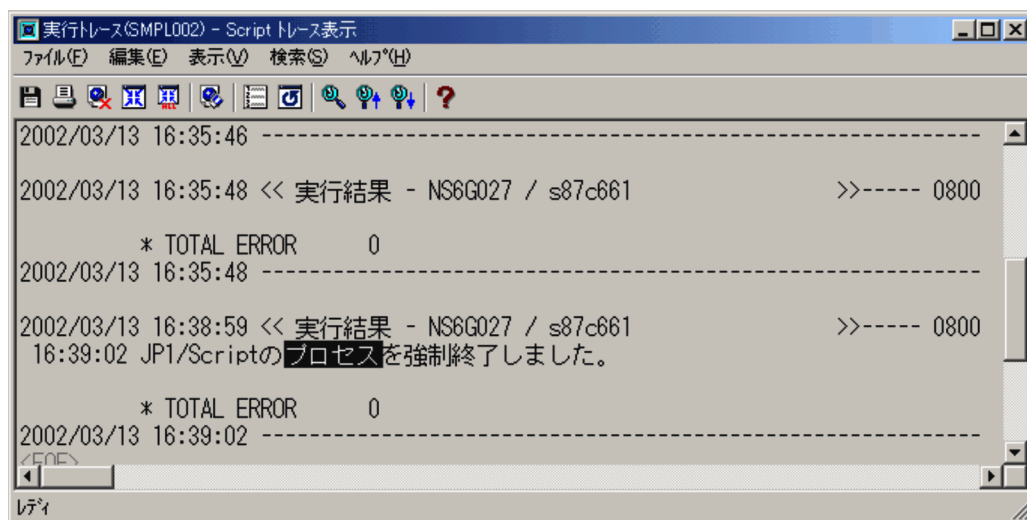


ダイアログの設定方法については、「4.3.4 検索ダイアログ」を参照してください。

### 3. JP1/Script の操作

4. 検索する文字列を指定して,[ 上検索 ] ボタン, または [ 下検索 ] ボタンをクリックする。

検索処理が実行されます。



- 操作 3. で [ キャンセル ] ボタンをクリックすると, 検索を実行しないでダイアログを閉じます。
- 検索文字列を指定しないで検索を実行しようとすると, ビープ音がなります。

## 3.5 トレースファイルの表示操作

### 3.5.1 トレースファイルの表示ウィンドウとメニュー

トレースファイルの表示ウィンドウは、スクリプトの実行状態、およびトレース情報を表示するためのウィンドウです。

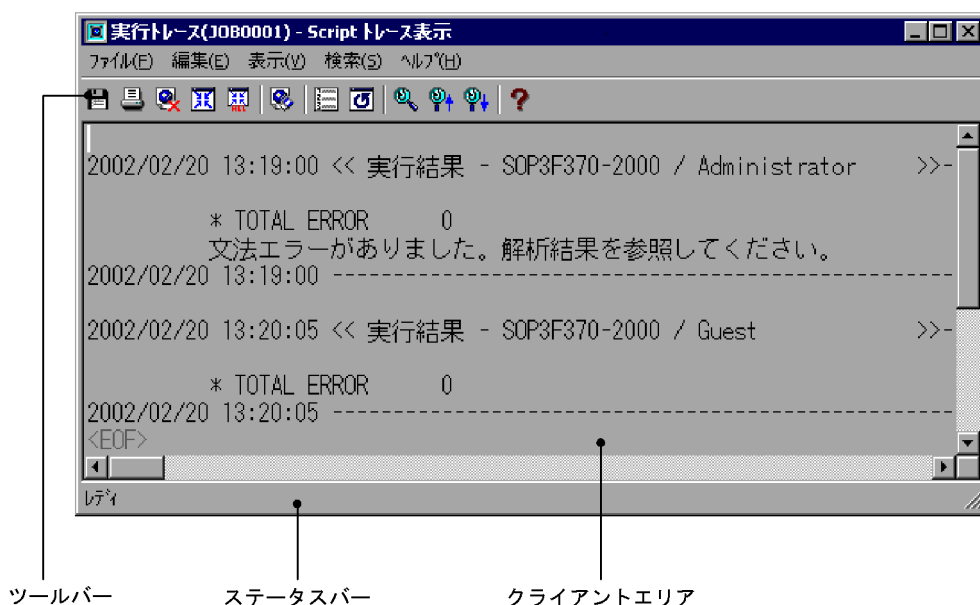
ここでは、トレースファイルを開いたときに表示されるウィンドウ、およびメニューの種類について説明します。

トレースビューアの出力形式については、「付録 A Script トレースのファイル出力形式」を参照してください。

#### (1) Script トレース表示ウィンドウ

Script トレースビューアウィンドウに表示されているトレースファイルをダブルクリックすると、Script トレース表示ウィンドウが表示されます。Script トレース表示ウィンドウ、およびウィンドウの各部の名称を図 3-5 に示します。

図 3-5 Script トレース表示ウィンドウ



#### (a) ツールバー

ツールバーには、メニューバーから選択できるコマンドの中から、頻繁に使用するコマンドだけをボタンの形で表示します。ボタンをマウスでクリックするだけで、該当するコマンドを実行できます。オプションメニューによって、非表示にすることもできます。

Script トレース表示ウィンドウのツールバーには、次のコマンドのボタンが表示されます。

[ 保存 ] ボタン

トレースファイルを別名で保存します。

[ 印刷 ] ボタン

トレースファイルを印刷します。

[ 削除 ] ボタン

トレースファイルを削除します。

[ 閉じる ] ボタン

トレースファイルの表示ウィンドウを終了します。

[ すべて閉じる ] ボタン

現在表示されているすべての Script トレース表示ウィンドウを終了します。

[ クリア ] ボタン

トレースファイルを空 ( ファイルサイズ 0 ) にします。

[ 行番号 ] ボタン

行番号の表示・非表示を切り替えます。

[ 最新 ] ボタン

トレースファイルの内容を最新の状態にします。

[ 検索 ] ボタン

検索情報を設定します。

[ 上検索 ] ボタン

検索文字列を現在の位置から上方向に検索します。

[ 下検索 ] ボタン

検索文字列を現在の位置から下方向に検索します。

[ ヘルプ ] ボタン

JP1/Script のオンラインヘルプを表示します。

(b) ステータスバー

ステータスバーは、トレースビューアが現在実行している処理に関するメッセージや、処理終了後の状態に関するメッセージを表示するための領域です。

(c) クライアントエリア

クライアントエリアには、トレースファイルの内容が表示されます。

ここに表示しきれない内容がある場合は、スクロールバーが表示されます。

このクライアントエリアに表示する文字の表示フォント、およびフォントサイズを変更



できます。

#### 〔レジストリキー〕

HKEY\_CURRENT\_USER¥SOFTWARE¥HITACHI¥JP1/Script¥SPTTM¥TraceWindow

#### 〔値〕

Font(文字列): フォント

FontSize(DWORD): フォントサイズ

## (2) Script トレース表示ウィンドウのメニュー

Script トレース表示ウィンドウの機能を、メニューごとに説明します。Script トレース表示ウィンドウのメニュー一覧を表 3-16 に示します。

表 3-16 Script トレース表示ウィンドウのメニュー一覧

メニュー	コマンド (機能)	機能概要
〔ファイル〕	〔名前を付けて保存〕	選択されているトレースファイルに別の名前を付けて保存します。
	〔トレースファイルの削除〕	選択されているトレースファイルを削除します。
	〔印刷〕	選択されているトレースファイルの内容を印刷します。
	〔トレース表示を閉じる〕	現在表示されているトレースファイルの表示ウィンドウを終了します。
	〔すべてのトレース表示を閉じる〕	現在表示されているすべてのトレースファイルの表示ウィンドウを終了します。
〔編集〕	〔トレースファイルのクリア〕	トレースファイルを空 (ファイルサイズ 0) にします。
〔表示〕	〔ツールバー〕	ツールバーの表示・非表示を切り替えます。
	〔ステータスバー〕	ステータスバーの表示・非表示を切り替えます。
	〔行番号を表示〕	行番号の表示・非表示を切り替えます。
	〔縦に並べる〕	表示されているトレースファイルの表示ウィンドウを、デスクトップ上に縦に並べます。
	〔横に並べる〕	表示されているトレースファイルの表示ウィンドウを、デスクトップ上に横に並べます。
	〔すべてのトレース表示を最小化〕	表示されているトレースファイルの表示ウィンドウを、すべて最小化します。
	〔最新の情報に更新〕	表示しているトレースファイルの内容を、最新状態にします。
〔検索〕	〔検索〕	検索する情報を入力します。
	〔上検索〕	検索文字列を上に向かって検索します。
	〔下検索〕	検索文字列を下に向かって検索します。
〔ヘルプ〕	〔目次〕	JP1/Script オンラインヘルプの目次を表示します。

メニュー	コマンド (機能)	機能概要
	[ キーワードで検索 ]	JP1/Script オンラインヘルプのキーワード一覧を表示します。
	[ ヘルプの使い方 ]	オンラインヘルプの使い方を表示します。
	[ バージョン情報 ]	JP1/Script トレース表示のバージョン情報を表示します。

## 3.5.2 Script トレース表示ウィンドウでのマウスとキーの操作

### (1) マウス操作

Script トレース表示ウィンドウのクライアントエリアでのマウス操作を表 3-17 に示します。

表 3-17 Script トレース表示ウィンドウでのマウス操作

操作	処理
クリック	それまでの選択を解除して、新たに対象を選択する。

### (2) キー操作

Script トレース表示ウィンドウでのマウスとキーの操作のクライアントエリアでのキー操作を表 3-18 に示します。

表 3-18 Script トレース表示ウィンドウでのキー操作

操作	処理
Ctrl + B	検索文字列を上方向に向かって検索する。
Ctrl + C	選択した部分をクリップボードへコピーする。
Ctrl + F	[ 検索情報指定 ] ダイアログを表示する。
Ctrl + N	検索文字列を下方向に向かって検索する。
Ctrl + P	トレースファイルを印刷する。
Del	トレースファイルを削除する。

## 3.6 メニューエディタの操作

---

Script メニューエディタウィンドウでできる操作の一覧を次に示します。

- 新しいメニューフォームを作成する
- メニューフォームをコピーする
- メニューフォームを貼り付ける
- メニューフォームを削除する
- コントロールの属性を一括して変更する
- コントロールを一行に並べる
- コントロールを均等な間隔で並べる
- コントロールをメニューフォームの中央に並べる
- コントロールの大きさをそろえる
- コントロールをテキストの大きさに合わせる
- メニューフォーム上にグリッドを表示する
- タブの移動順序を設定する
- 選択したコントロールを前面に移動する
- 選択したコントロールを背面に移動する
- コマンドのプロパティを設定する
- メニューフォームをテスト表示する
- メニューフォームの内容を印刷する
- 新しいメニュー情報ファイルを作成する
- 既存のメニュー情報ファイルを開く
- メニュー情報ファイルを保存する
- メニュー情報ファイルに名前をつけて保存する
- 直前の編集作業を無効にする
- 直前の編集作業をやり直す
- 選択したコントロールをクリップボードに切り抜く
- 選択したコントロールをクリップボードにコピーする
- クリップボードのコントロールをメニューフォームに貼り付ける
- メニューフォーム上のすべてのコントロールを選択する
- 選択したコントロールを削除する
- ツールバーの表示 / 非表示を切り替える
- ステータスバーの表示 / 非表示を切り替える
- コントロールボックスの表示 / 非表示を切り替える
- プロパティダイアログの表示 / 非表示を切り替える
- コマンドプロパティダイアログの表示 / 非表示を切り替える
- ヘルプを表示する

注

これらの操作は特に説明していません（これらは Windows の標準の操作に準じてい

るか、メニュー一覧で該当するメニューを選択するだけで実行できる操作です。

### 3.6.1 Script メニューエディタウィンドウとメニュー

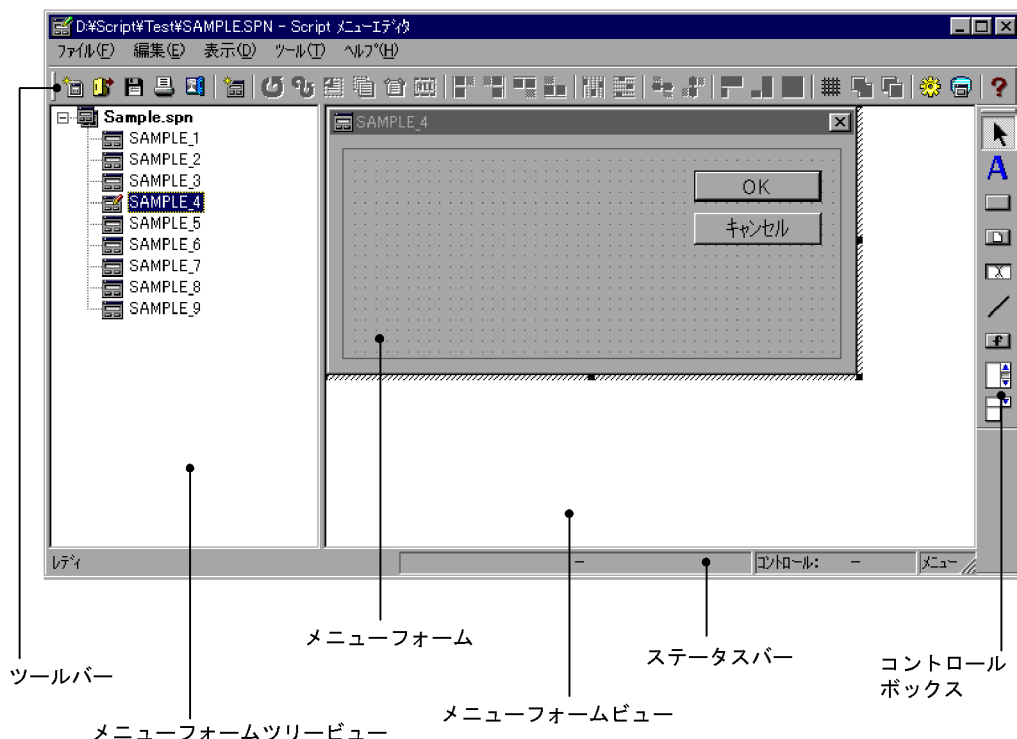
メニューエディタは、作成したスクリプトファイルに対応した画面（メニューフォーム）を作成するためのプログラムです。メニューエディタで作成した画面（メニューフォーム）は、メニュー情報ファイルとして保存されます。

ここでは、メニューエディタを起動すると表示される Script メニューエディタウィンドウについて説明します。また、メニューエディタの機能をメニューごとに説明します。

#### （１）Script メニューエディタウィンドウ

Script メニューエディタウィンドウ、およびウィンドウの各部の名称を、図 3-6 に示します。

図 3-6 Script メニューエディタウィンドウ



#### （a）ツールバー

ツールバーには、メニューバーから選択できるコマンドの中から、頻繁に使用するコマンドだけをボタンの形で表示しています。ボタンをマウスでクリックするだけで、該当するコマンドを実行できます。表示メニューによって非表示にすることもできます。

Script メニューエディタウィンドウのツールバーには、次のコマンドのボタンが表示されます。

[ 新規作成 ] ボタン

メニュー情報ファイルを新規に作成します。

[ 開く ] ボタン

メニュー情報ファイルを開きます。

[ 保存 ] ボタン

メニュー情報ファイルを保存します。

[ メニューフォームの印刷 ] ボタン

メニューフォームの定義内容を印刷します。

[ 終了 ] ボタン

メニューエディタを終了します。

[ メニューフォームの新規作成 ] ボタン

メニューフォームを新規に作成します。

[ 元に戻す ] ボタン

直前の操作を無効にします。

[ やり直し ] ボタン

直前の操作をやり直します。

[ 切り抜き ] ボタン

選択したコントロールをクリップボードに切り抜きます。

[ コピー ] ボタン

選択したコントロールをクリップボードにコピーします。

[ 貼り付け ] ボタン

クリップボードのコントロールをメニューフォームに貼り付けます。

[ すべて選択 ] ボタン

メニューフォームに記述されているコントロールをすべて選択します。

[ 左寄せ配置 ] ボタン

コントロールを、基準となるコントロールの左端に揃えます。

[ 右寄せ配置 ] ボタン

コントロールを、基準となるコントロールの右端に揃えます。

[ 上寄せ配置 ] ボタン

コントロールを、基準となるコントロールの上端に揃えます。

[ 下寄せ配置 ] ボタン

コントロールを、基準となるコントロールの下端に揃えます。

### 3. JP1/Script の操作

#### [ 左右均等配置 ] ボタン

コントロールを左右のスペースが均等になるように配置します。

#### [ 上下均等配置 ] ボタン

コントロールを上下のスペースが均等になるように配置します。

#### [ 上下中央配置 ] ボタン

コントロールを、メニューフォーム内で垂直方向の中央に配置します。

#### [ 左右中央配置 ] ボタン

コントロールを、メニューフォーム内で水平方向の中央に配置します。

#### [ 幅を合わせる ] ボタン

コントロールをほかのコントロールの幅に合わせます。

#### [ 高さを合わせる ] ボタン

コントロールをほかのコントロールの高さに合わせます。

#### [ サイズを合わせる ] ボタン

コントロールをほかのコントロールのサイズに合わせます。

#### [ グリッドの設定 ] ボタン

グリッドを設定します。

#### [ 前面へ移動 ] ボタン

選択されたコントロールを前面に移動します。

#### [ 背面へ移動 ] ボタン

選択されたコントロールを背面に移動します。

#### [ コマンドプロパティの設定 ] ボタン

コマンドのプロパティ定義を設定します。

#### [ テスト表示 ] ボタン

作成した画面（メニューフォーム）をテスト表示します。

#### [ ヘルプ ] ボタン

JP1/Script のオンラインヘルプを表示します。

#### (b) ステータスバー

ステータスバーは、メニューエディタが現在実行している処理に関するメッセージや、処理終了後の状態に関するメッセージを表示するための領域です。

#### (c) メニューフォーム

画面を作成するシートです。ここで作成した画面の情報がメニュー情報ファイルに記述されます。

## (d) メニューフォームツリービュー

メニュー情報ファイルに記述されている、すべてのメニューフォーム名をツリー形式で表示する領域です。メニューフォーム名をクリックすると、メニューフォームビューにメニューフォームが表示されます。

## (e) メニューフォームビュー

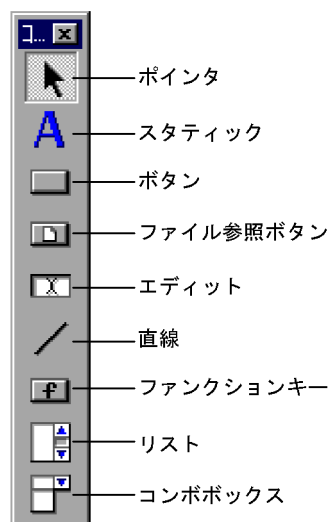
メニュー情報ファイルに記述されているメニューフォームを表示する領域です。

## (f) コントロールボックス

利用できるコントロールがボタンで表示されています。コントロールのボタンを選択してからメニューフォーム上にドラッグアンドドロップすると、選択したコントロールがメニューフォーム上に貼り付けられます。

コントロールボックスを、図 3-7 に示します。

図 3-7 コントロールボックス



コントロールボックスの各コントロールについて、次に説明します。

## [ ポインタ ]

コントロールを選択するとき使います。

## [ スタティク ]

スタティク（見出し）を、メニューフォーム上に貼り付ける場合に選択します。

## [ ボタン ]

ボタンを、メニューフォーム上に貼り付ける場合に選択します。

## [ ファイル参照ボタン ]

ファイル参照ボタンを、メニューフォーム上に貼り付ける場合に選択します。

#### [ エディット ]

エディットを、メニューフォーム上に貼り付ける場合に選択します。

#### [ 直線 ]

直線を、メニューフォーム上に貼り付ける場合に選択します。

#### [ ファンクションキー ]

ファンクションキーを、メニューフォーム上に貼り付ける場合に選択します。このコントロールは、キーボードのファンクションキーと対応させるためのコントロールです。

#### [ リスト ]

リストを、メニューフォーム上に貼り付ける場合に選択します。

#### [ コンボボックス ]

コンボボックスを、メニューフォーム上に貼り付ける場合に選択します。

### (2) Script メニューエディタウィンドウのメニュー

ここでは、Script メニューエディタウィンドウのメニューバーに表示されるメニュー、および Script メニューエディタウィンドウで表示されるポップアップメニューについて説明します。

#### (a) メニューバーのメニュー

メニューエディタの機能を、メニューごとに説明します。メニューエディタのメニュー一覧を表 3-19 に、レイアウトコマンドのカスケードメニュー一覧を表 3-20 に示します。

表 3-19 メニューエディタのメニュー一覧

メニュー	コマンド ( 機能 )	機能概要
[ ファイル ]	[ 新規作成 ]	メニュー情報ファイルを新規に作成します。
	[ 開く ]	メニュー情報ファイルを開きます。
	[ 保存 ]	メニュー情報ファイルを保存します。
	[ 名前を付けて保存 ]	選択されているメニュー情報ファイルに別の名前を付けて保存します。
	[ メニューフォームの印刷 ]	メニューフォームの定義内容を印刷します。
	[ 終了 ]	メニューエディタを終了します。
	( メニュー情報ファイル名 )	直前に保存したメニュー情報ファイル名が表示されます。
[ 編集 ]	[ メニューフォームの新規作成 ]	メニューフォームを新規に作成します。
	[ メニューフォームのコピー ]	既存のメニューフォームをクリップボードにコピーします。
	[ メニューフォームの貼り付け ]	クリップボードのメニューフォームを貼り付けます。



メニュー	コマンド (機能)	機能概要
	[ メニューフォームの削除 ]	クリップボードのメニューフォームを削除します。
	[ 元に戻す ]	編集する直前の状態に戻します。
	[ やり直し ]	直前の操作をやり直します。
	[ 切り抜き ]	選択したコントロールをクリップボードに切り抜きます。
	[ コピー ]	選択したコントロールをクリップボードにコピーします。
	[ 貼り付け ]	クリップボードのコントロールをメニューフォーム上に貼り付けます。
	[ すべて選択 ]	メニューフォーム上に記述されているすべてのコントロールを選択します。
	[ 削除 ]	選択したコントロールを削除します。
	[ 一括変更 ]	選択したコントロールの属性を一括して変更します。
	[ レイアウト ]	メニューフォームに貼り付けたコントロールのレイアウトを整えます。また、[ レイアウト ] コマンドの下には、カスケードメニューが表示されます。カスケードメニューについては、表 3-20 を参照してください。
[ 表示 ]	[ ツールバー ]	ツールバーの表示・非表示を切り替えます。
	[ ステータスバー ]	ステータスバーの表示・非表示を切り替えます。
	[ コントロールボックス ]	コントロールボックスの表示・非表示を切り替えます。
	[ プロパティ ]	プロパティダイアログの表示・非表示を切り替えます。
	[ コマンドプロパティ ]	コマンドプロパティダイアログの表示・非表示を切り替えます。
[ ツール ]	[ コマンドプロパティの設定 ]	コマンドのプロパティを定義します。
	[ テスト表示 ]	作成した画面 (メニューフォーム) をテスト表示します。
[ ヘルプ ]	[ 目次 ]	JP1/Script オンラインヘルプの目次を表示します。
	[ キーワードで検索 ]	JP1/Script オンラインヘルプのキーワード一覧を表示します。
	[ ヘルプの使い方 ]	オンラインヘルプの使い方を表示します。
	[ バージョン情報 ]	JP1/Script メニューエディタのバージョン情報を表示します。

表 3-20 レイアウトコマンドのカスケードメニュー一覧

カスケードメニュー	コマンド (機能)	機能概要
[ 整列 ]	[ 左端 ]	コントロールを、基準となるコントロールの左端に揃えます。
	[ 右端 ]	コントロールを、基準となるコントロールの右端に揃えます。
	[ 上端 ]	コントロールを、基準となるコントロールの上端に揃えます。
	[ 下端 ]	コントロールを、基準となるコントロールの下端に揃えます。
	[ 上下中央 ]	コントロールを、基準となるコントロールの垂直方向の中央に配置します。
	[ 左右中央 ]	コントロールを、基準となるコントロールの水平方向の中央に配置します。
[ スペースの均等化 ]	[ 左右方向 ]	コントロールを、左右のスペースが均等になるように配置します。
	[ 上下方向 ]	コントロールを、上下のスペースが均等になるように配置します。
[ メニューフォーム内での整列 ]	[ 上下中央 ]	コントロールを、メニューフォーム上で垂直方向の中央に配置します。
	[ 左右中央 ]	コントロールを、メニューフォーム上で水平方向の中央に配置します。
[ 同じサイズに設定 ]	[ 幅 ]	コントロールを、ほかのコントロールの幅に合わせます。
	[ 高さ ]	コントロールを、ほかのコントロールの高さに合わせます。
	[ 幅と高さ ]	コントロールを、ほかのコントロールのサイズに合わせます。
[ テキストへのサイズ合わせ ]	-	コントロールをテキストのサイズに合わせます。
[ グリッドの設定 ]	-	グリッドを設定します。
[ タブオーダの設定 ]	-	タブオーダを設定します。
[ 前面へ移動 ]	-	選択したコントロールを前面に移動します。
[ 背面へ移動 ]	-	選択したコントロールを背面に移動します。

## (b) ポップアップメニュー

Script メニューエディタウィンドウのメニューフォームビューでマウスの右ボタンをクリックすると、ポップアップメニューが表示されます。Script メニューエディタウィンドウで表示されるポップアップメニューを表 3-21 に示します。

表 3-21 ポップアップメニュー一覧

ポップアップメニュー	機能概要
[ 新規作成 ]	メニュー情報ファイルを新規に作成します。
[ 開く ]	既存のメニュー情報ファイルを開きます。
[ 保存 ]	メニュー情報ファイルを上書き保存します。
[ メニューフォームの印刷 ]	メニューフォームの定義内容を印刷します。
[ メニューフォームのコピー ]	既存のメニューフォームをクリップボードにコピーします。
[ メニューフォームの貼り付け ]	クリップボードのメニューフォームを貼り付けます。
[ メニューフォームの削除 ]	既存のメニューフォームを削除します。
[ 元に戻す ]	編集する直前の状態に戻します。
[ やり直し ]	直前の操作をやり直します。
[ 切り抜き ]	選択したコントロールを切り抜きます。
[ コピー ]	選択したコントロールをクリップボードにコピーします。
[ 貼り付け ]	クリップボードのコントロールをメニューフォーム上に貼り付けます。
[ すべて選択 ]	メニューフォーム上に記述されているコントロールすべてを選択します。
[ 削除 ]	選択したコントロールを削除します。
[ 一括変更 ]	選択したコントロールの属性を一括して変更します。
[ レイアウト ]	メニューフォームに貼り付けたコントロールのレイアウトを整えます。また,[レイアウト]コマンドの下には、カスケードメニューが表示されます。カスケードメニューについては、表 3-20 を参照してください。
[ コマンドプロパティの設定 ]	コマンドのプロパティを定義します。
[ テスト表示 ]	作成した画面(メニューフォーム)をテスト表示します。
[ プロパティ ]	選択したコントロールのプロパティダイアログを表示します。

## 3.6.2 Script メニューエディタウィンドウでのマウスとキーの操作

### (1) マウス操作

メニューフォームビュー上でのマウス操作を表 3-22 に示します。

表 3-22 Script メニューエディタウィンドウでのマウス操作

操作	処理
クリック	新たに対象を選択する。またはそれまでの選択を解除する。
ダブルクリック	メニューフォーム、または各コントロールのプロパティダイアログを表示する。

### 3. JP1/Script の操作

操作	処理
右ボタンシングルクリック	ポップアップメニューを表示する。
ドラッグ	各コントロールを移動，または選択する。複数のコントロールを選択状態にした場合，選択順序はコントロールの座標位置を YX ソートした結果と同じになる。
Ctrl+ ドラッグ	選択されたコントロールを複写する。

## (2) キー操作

Script メニューエディタウィンドウのメニューフォームビューでのキー操作を表 3-23 に示します。

表 3-23 Script メニューエディタウィンドウでのキー操作

操作	処理
Alt+1	コントロールボックスの表示，非表示を切り替える。
Alt+2	プロパティダイアログの表示，非表示を切り替える。
Alt+3	コマンドプロパティダイアログの表示，非表示を切り替える。
Alt+F4	メニューエディタを終了する。
Alt+R	コントロールを左右のスペースが均等になるように配置する。
Alt+D	コントロールを上下のスペースが均等になるように配置する。
Ctrl+A	メニューフォーム上のすべてのコントロールを選択する。
Ctrl+C	選択されたコントロールをコピーする。
Ctrl+F	メニューフォームを新規に作成する。
Ctrl+K	コマンドのプロパティを定義する。
Ctrl+N	メニュー情報ファイルを新規に作成する。
Ctrl+O	メニュー情報ファイルを開く。
Ctrl+P	メニューフォームの内容を印刷する。
Ctrl+S	メニュー情報ファイルを保存する。
Ctrl+V	クリップボードの内容をメニューフォームに貼り付ける。
Ctrl+X	選択されたコントロールを切り抜く。
Ctrl+Z	編集される直前の状態に戻す。
Ctrl+F9	コントロールをダイアログ内で垂直方向の中央に配置する。
Ctrl+L	コントロールをほかのコントロールの左端にそろえる。
Ctrl+R	コントロールをほかのコントロールの右端にそろえる。
Ctrl+U	コントロールをほかのコントロールの上端にそろえる。
Ctrl+D	コントロールをほかのコントロールの下端にそろえる。
Delete	選択されたメニューフォーム，またはコントロールを削除する。
F5	メニューフォームをテスト表示する。

操作	処理
F7	コントロールをテキストの大きさに合わせる。
F9	コントロールをほかのコントロールの垂直方向の中央に配置する。
Shift+F9	コントロールをほかのコントロールの水平方向の中央に配置する。
Shift+Ctrl+Z	直前の操作をやり直す。
Shift+Ctrl+F9	コントロールをダイアログ内で水平方向の中央に配置する。
	選択された一つ以上のコントロールを左方向に移動する（グリッド単位）。
	選択された一つ以上のコントロールを右方向に移動する（グリッド単位）。
	選択された一つ以上のコントロールを上方向に移動する（グリッド単位）。
	選択された一つ以上のコントロールを下方向に移動する（グリッド単位）。
Alt+	選択された一つ以上のコントロールを左方向に移動する（ピクセル単位）。
Alt+	選択された一つ以上のコントロールを右方向に移動する（ピクセル単位）。
Alt+	選択された一つ以上のコントロールを上方向に移動する（ピクセル単位）。
Alt+	選択された一つ以上のコントロールを下方向に移動する（ピクセル単位）。
Shift+	選択された一つのコントロールを左上を基準にして左方向にサイズ変更する（グリッド単位）。
Shift+	選択された一つのコントロールを左上を基準にして右方向にサイズ変更する（グリッド単位）。
Shift+	選択された一つのコントロールを左上を基準にして上方向にサイズ変更する（グリッド単位）。
Shift+	選択された一つのコントロールを左上を基準にして下方向にサイズ変更する（グリッド単位）。
Shift+Alt+	選択された一つのコントロールを左上を基準にして上方向にサイズ変更する（ピクセル単位）。
Shift+Alt+	選択された一つのコントロールを左上を基準にして下方向にサイズ変更する（ピクセル単位）。
Shift+Alt+	選択された一つのコントロールを左上を基準にして左方向にサイズ変更する（ピクセル単位）。
Shift+Alt+	選択された一つのコントロールを左上を基準にして右方向にサイズ変更する（ピクセル単位）。
Enter	選択された一つのコントロールのプロパティを表示する。

注 選択されたコントロールが直線の場合，操作は無効になります。

### 3.6.3 新しいメニューフォームを作成する

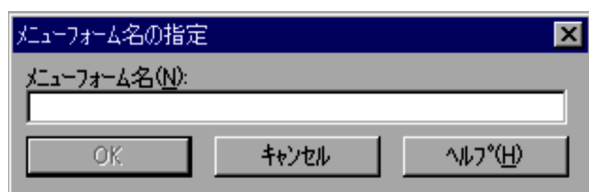
新しいメニューフォームを作成します。

メニューフォームとは，作成されたスクリプトファイルを実行したときに画面上に表示させるウィンドウのことです。メニューフォーム上には，コントロールと呼ばれる，ポ

タンやリストボックスなどを配置できます。

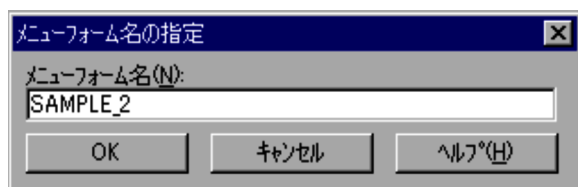
操作

1. [ 編集 ] - [ メニューフォームの新規作成 ] メニューを選択する。  
[ メニューフォーム名の指定 ] ダイアログが表示されます。

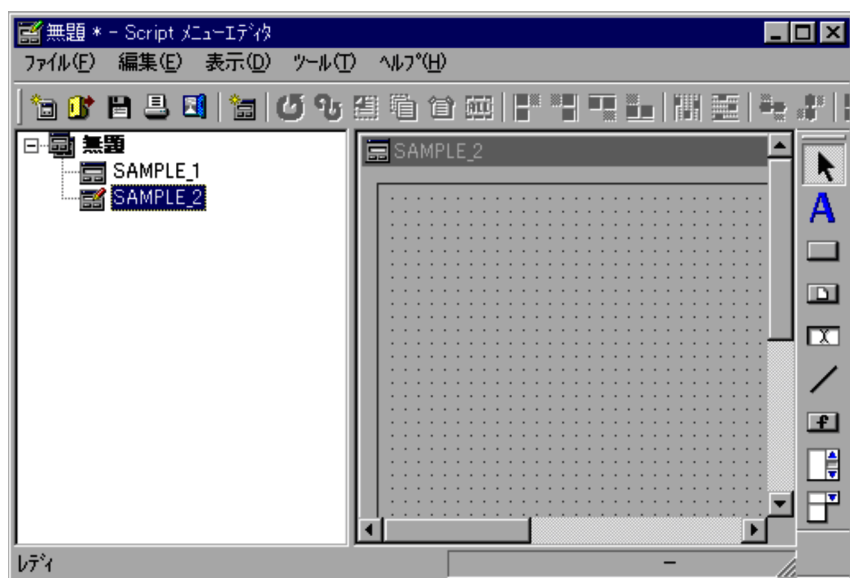


ダイアログの設定方法については、「4.4.38 メニューフォーム名の指定ダイアログ」を参照してください。

2. メニューフォーム名を入力する。



3. [ OK ] ボタンをクリックする。



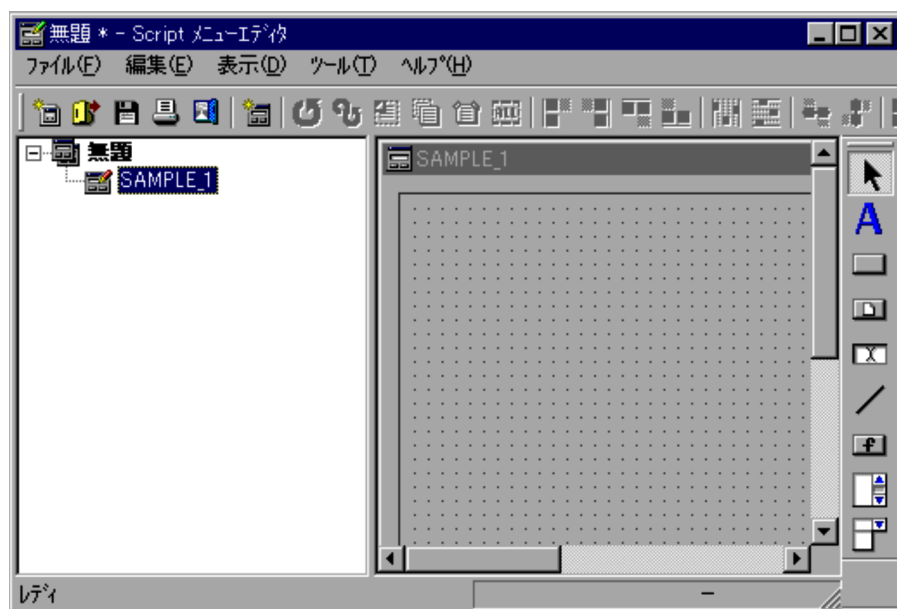
メニューフォームビューには新しいメニューフォームが表示され、メニューフォームツリービューには操作 2. で入力したメニューフォーム名

### 3.6.4 メニューフォームをコピーする

既存のメニューフォームをコピーします。

操作

1. メニューフォームツリービューからメニューフォームを選択する。



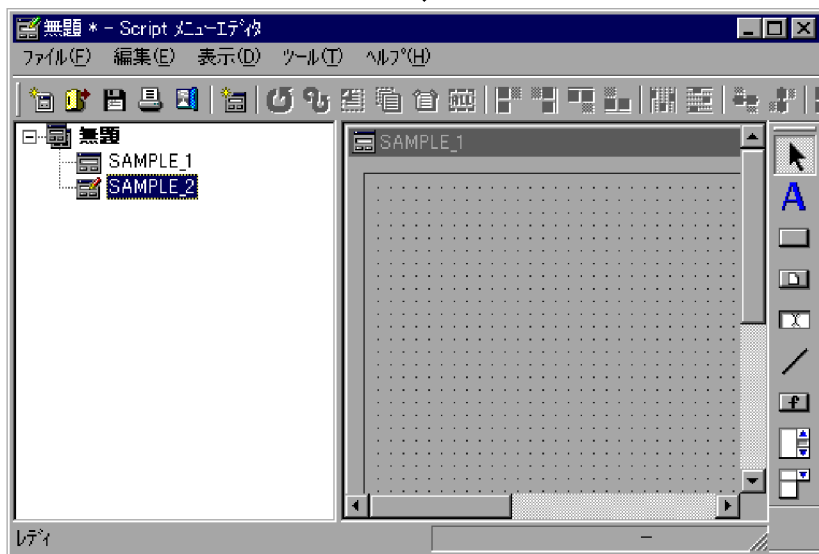
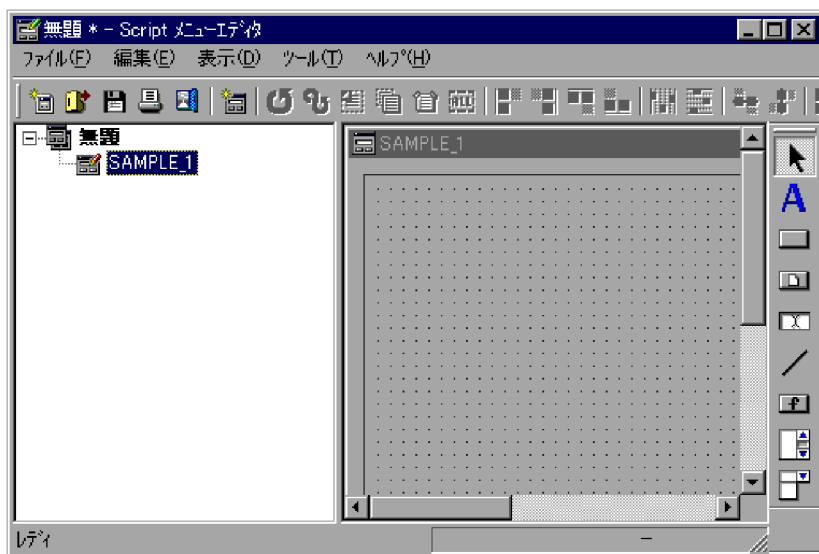
2. [ 編集 ] - [ メニューフォームのコピー ] を選択する。  
選択されたメニューフォームがクリップボードにコピーされます。

### 3.6.5 メニューフォームを貼り付ける

クリップボードのメニューフォームを貼り付けます。

操作

1. [ 編集 ] - [ メニューフォームの貼り付け ] メニューを選択する。  
クリップボードのメニューフォームが、メニューフォームビューに表示されます。また、貼り付けたメニューフォームの名前がメニューフォームツリービューに表示されます。



貼り付けるメニューフォームの名前がすでにある場合、貼り付けられる前の名前の末尾に「1」以降の数字が付加されます。(例：Name Name1, Name1 Name2)

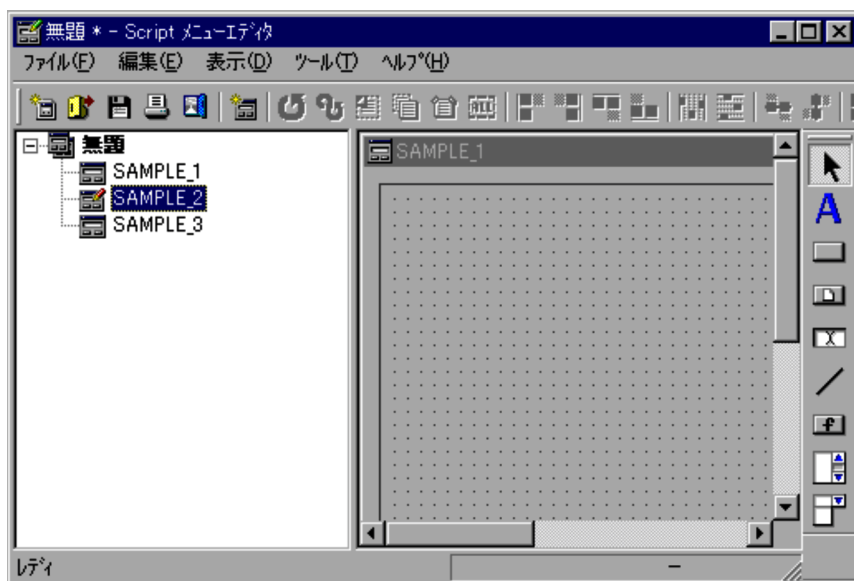
### 3.6.6 メニューフォームを削除する

作成したメニューフォームを削除します。

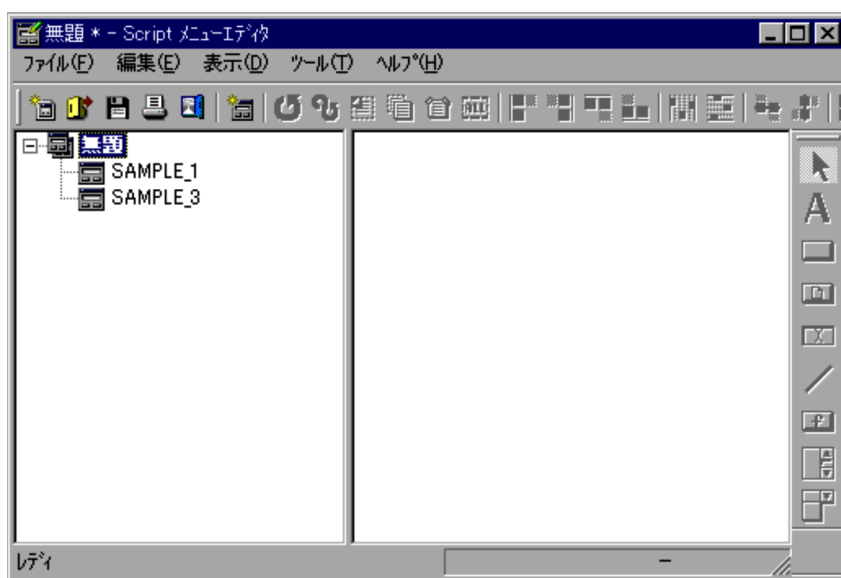
操作

1. メニューフォームツリービューからメニューフォームを選択する。





2. [ 編集 ] - [ メニューフォームの削除 ] を選択する。  
 選択されたメニューフォームが削除されます。



### 3.6.7 コントロールの属性を一括して変更する

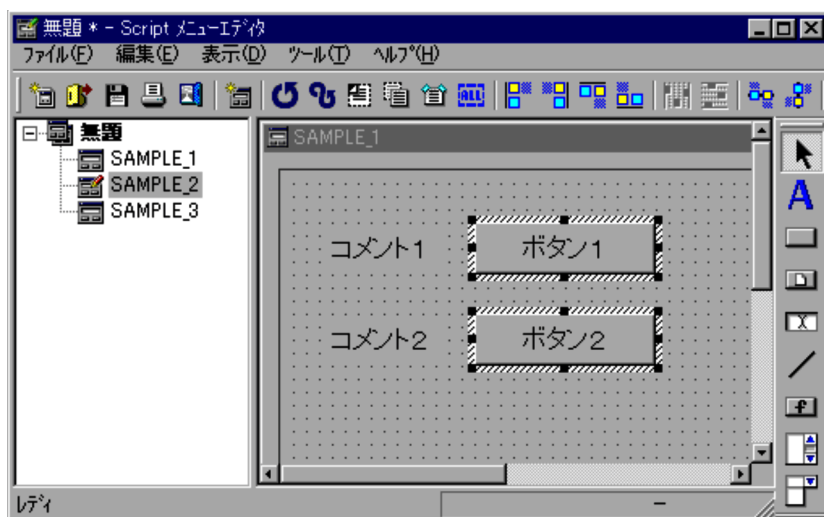
メニューフォーム上のコントロールの属性を一括して変更します。

操作

1. メニューフォーム上でコントロールを選択する。

### 3. JP1/Script の操作

ここでは、ボタン 1 とボタン 2 を選択します。

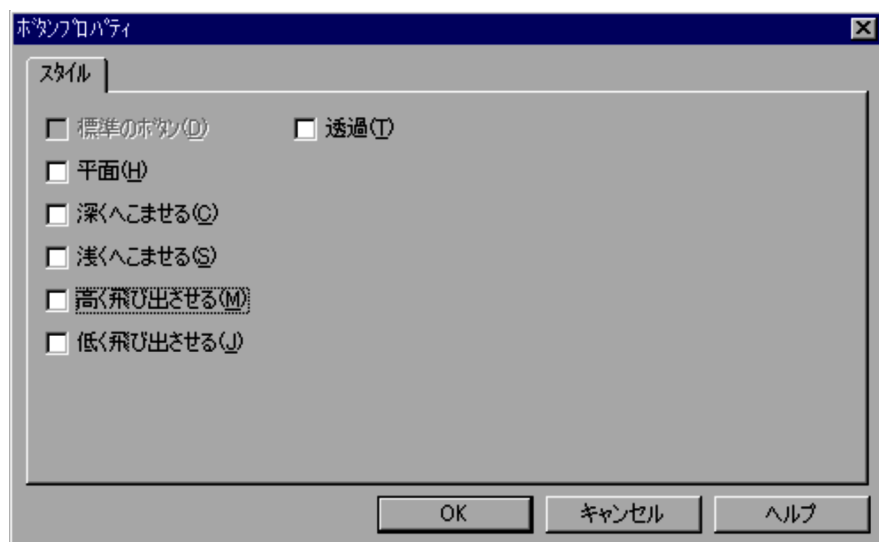


2. [編集] - [一括変更] メニューを選択する。  
[一括変更] ダイアログが表示されます。

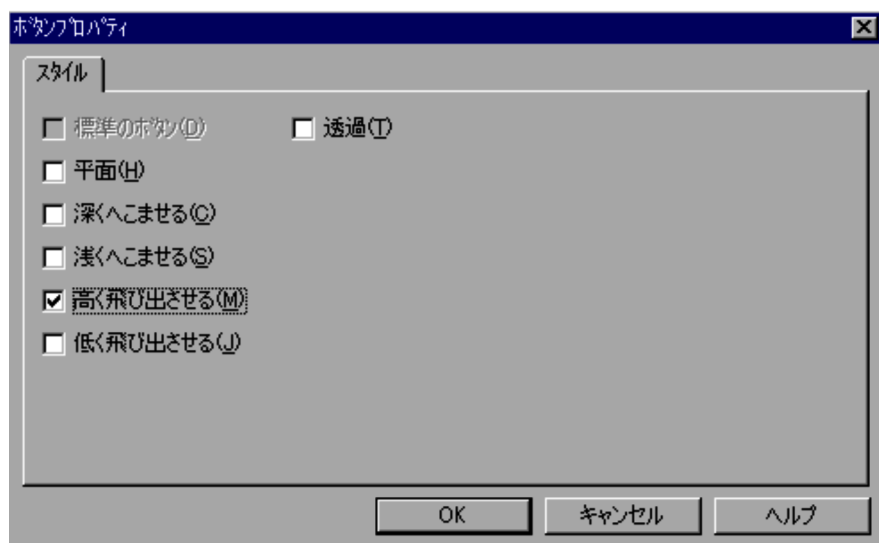


ダイアログの設定方法については、「4.4.39 一括変更ダイアログ」を参照してください。

3. 変更内容を設定する。  
ここでは、[スタイル] ボタンをクリックし、スタイルを変更します。



4. [ ボタンプロパティ ] ダイアログで、「高く飛び出させる」を選択します。

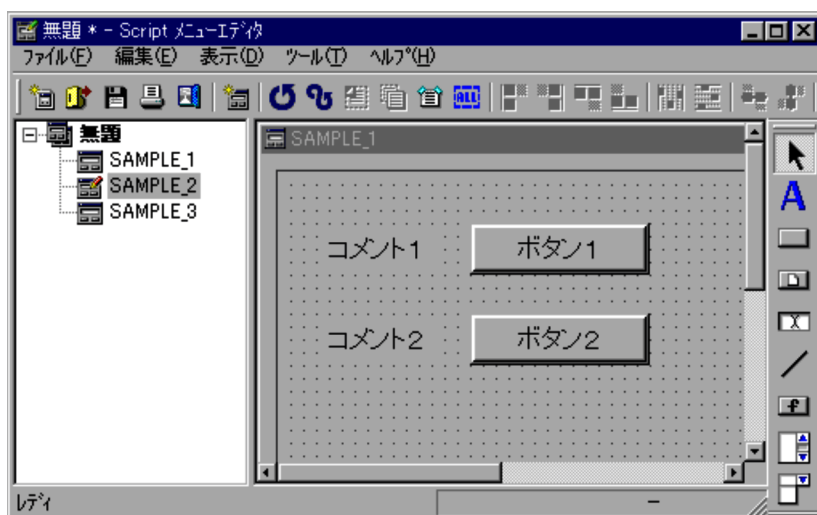


5. [ OK ] をクリックする。  
[ 一括変更 ] ダイアログに戻ります。



6. [OK] をクリックする。

操作 3. で変更した新しい内容でコントロールが設定されます。



- コントロールが選択されていないと、メニューがグレースアウトされてメニューを選択できません。
- 選択されたコントロールに直線が含まれる場合にはメニューがグレースアウトされてメニューを選択できません。

### 3.6.8 コントロールを一行に並べる

2 個以上のコントロールを一行に並べます。

最初に選択したコントロールの位置が基準になり、あとから選択したコントロールが移

動されます。

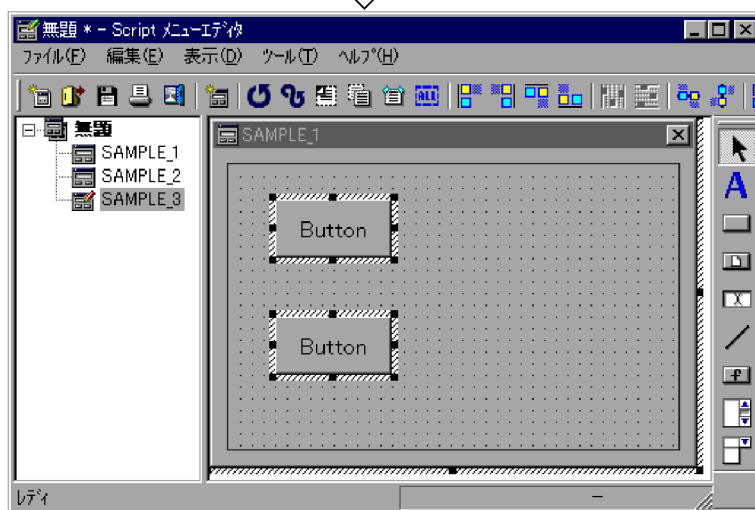
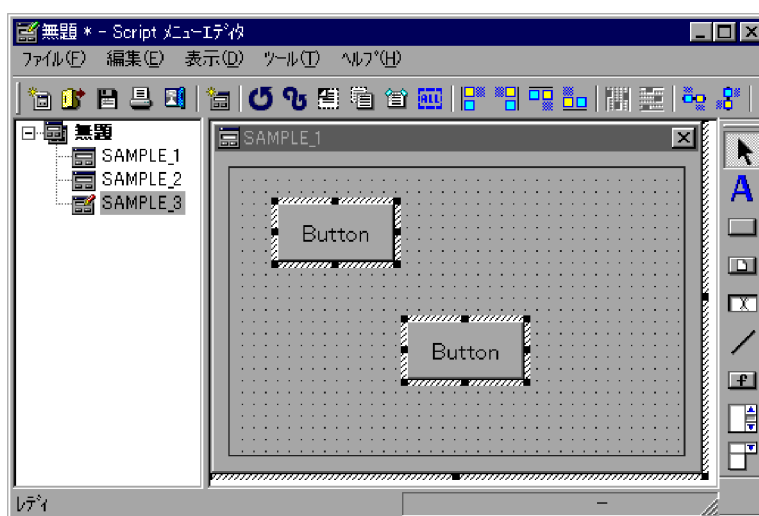
操作

1. メニューフォーム上でコントロールを 2 個以上選択する。
2. あとから選択したコントロールを、最初を選択した基準となるコントロールに合わせる。

並べ方によって、次のように操作が異なります。

左端にそろえる場合

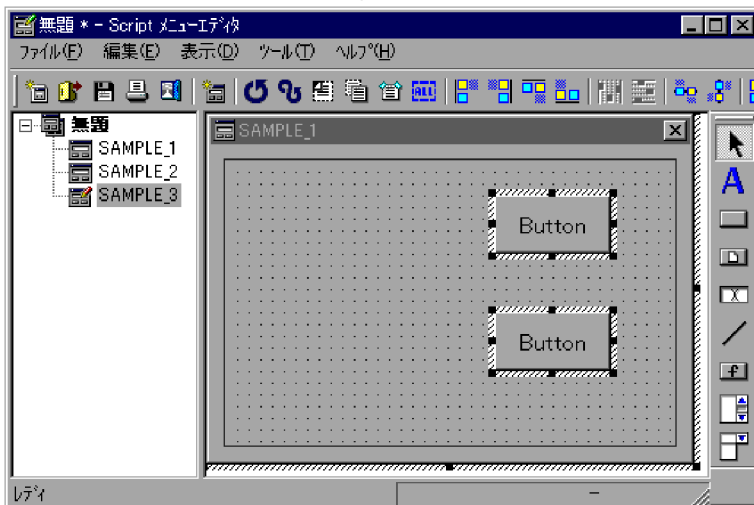
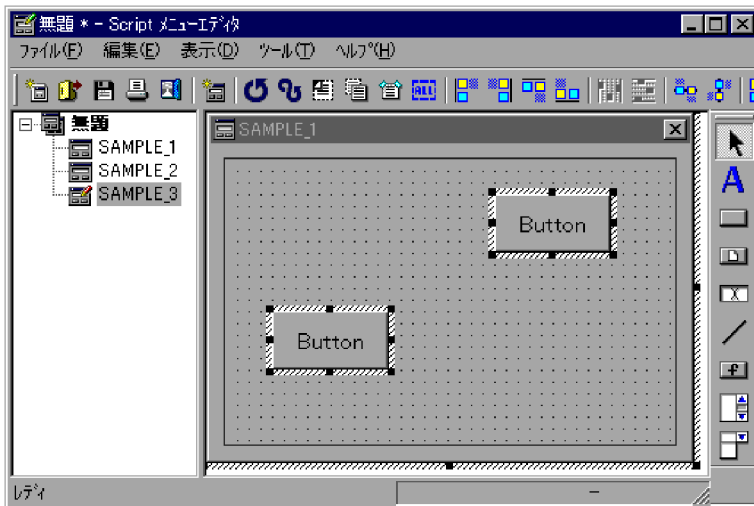
: [編集] - [レイアウト] - [整列] - [左端]



### 3. JP1/Script の操作

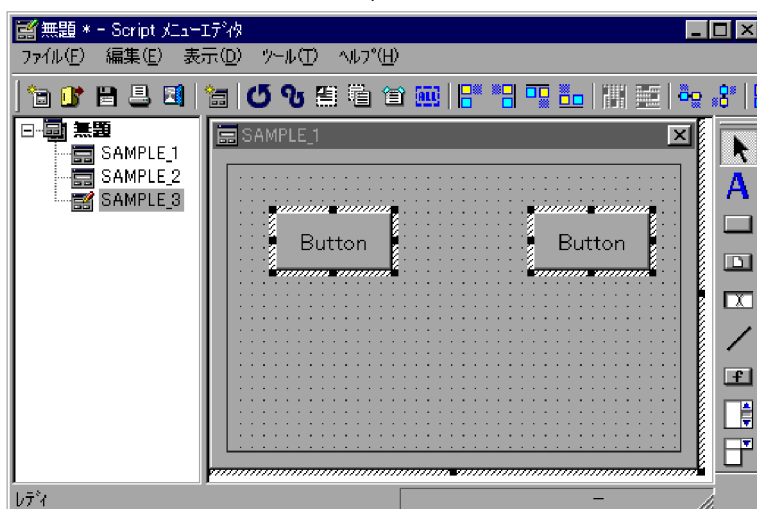
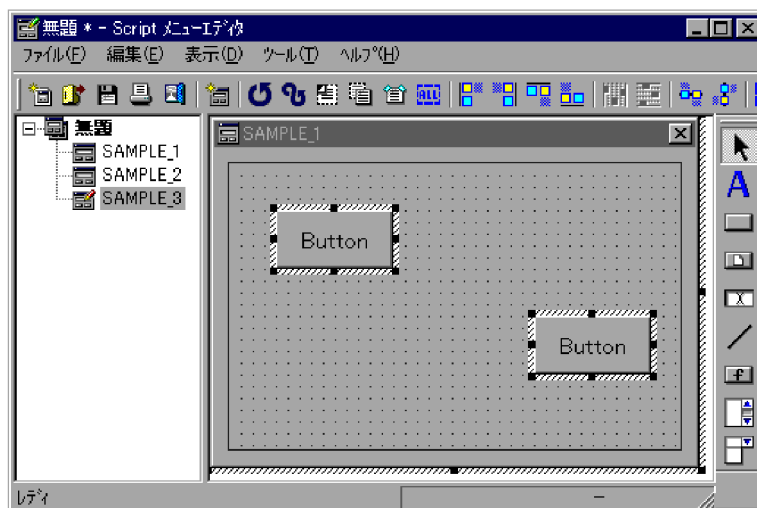
右端にそろえる場合

: [編集] - [レイアウト] - [整列] - [右端]



上端にそろえる場合

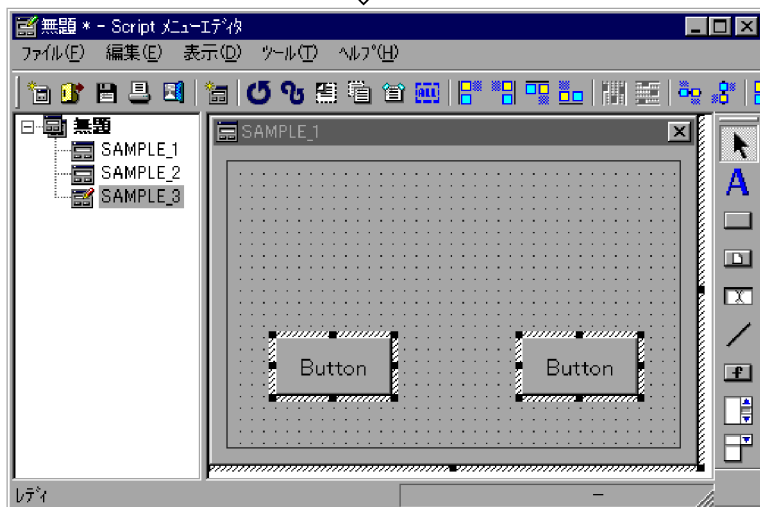
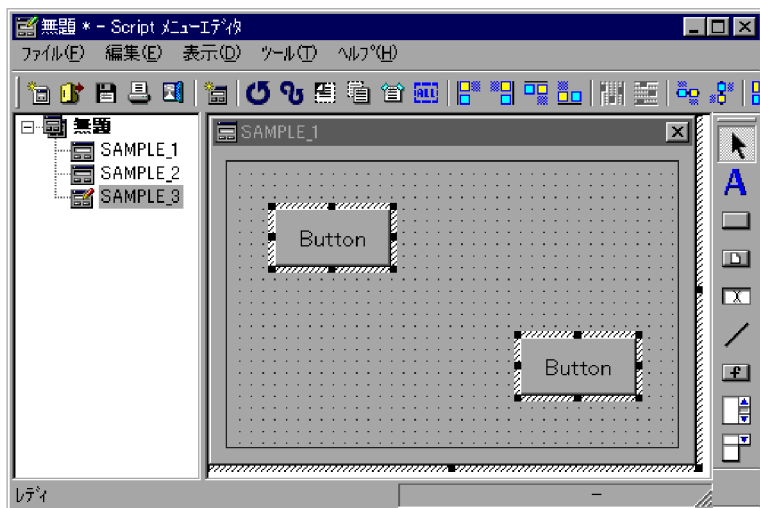
: [編集] - [レイアウト] - [整列] - [上端]



### 3. JP1/Script の操作

下端にそろえる場合

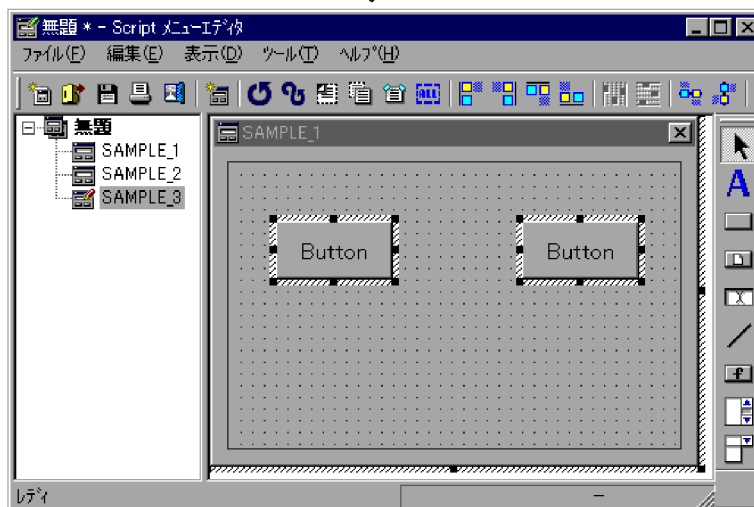
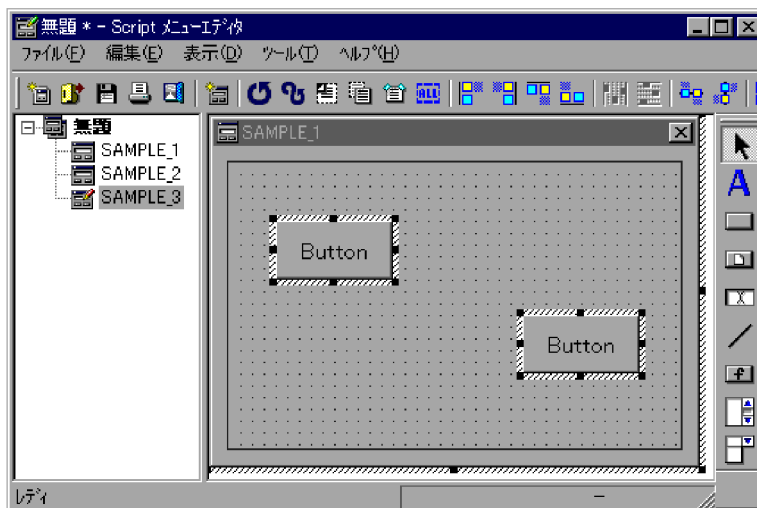
: [編集] - [レイアウト] - [整列] - [下端]





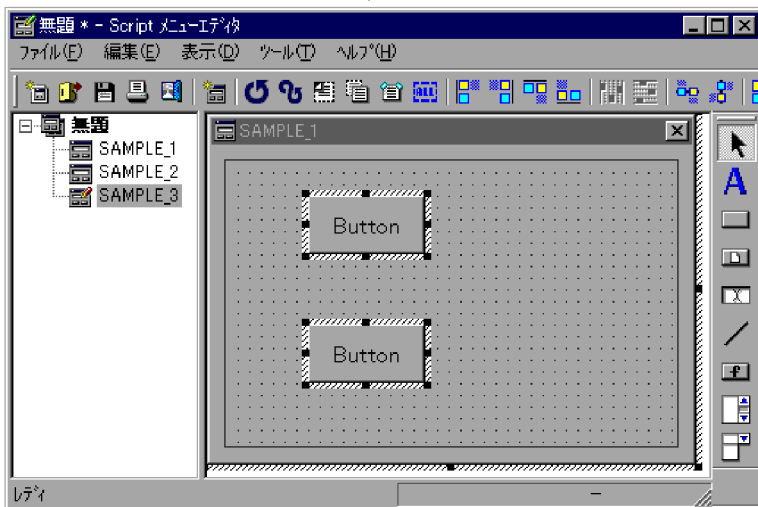
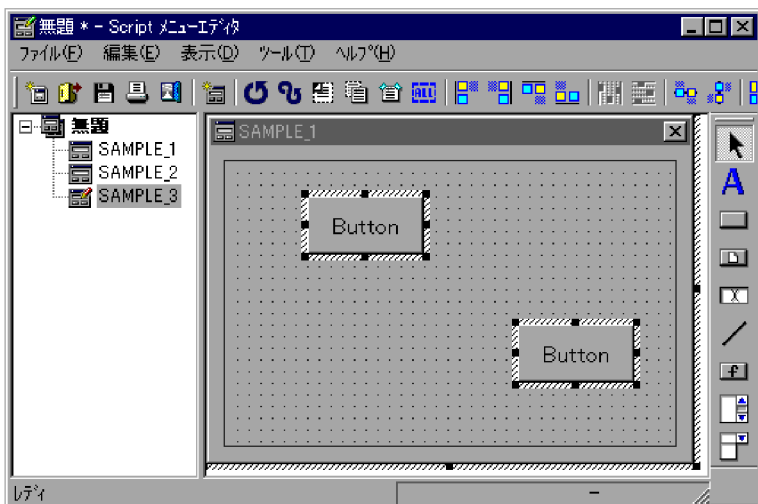
垂直方向の中央にそろえる場合

: [編集] - [レイアウト] - [整列] - [上下中央]



水平方向の中央にそろえる場合

: [編集] - [レイアウト] - [整列] - [左右中央]



選択したコントロールが一列に並びます。

- コントロールを 2 個以上選択しないと、メニューがグレイアウトされてメニューを選択できません。

### 3.6.9 コントロールを均等な間隔で並べる

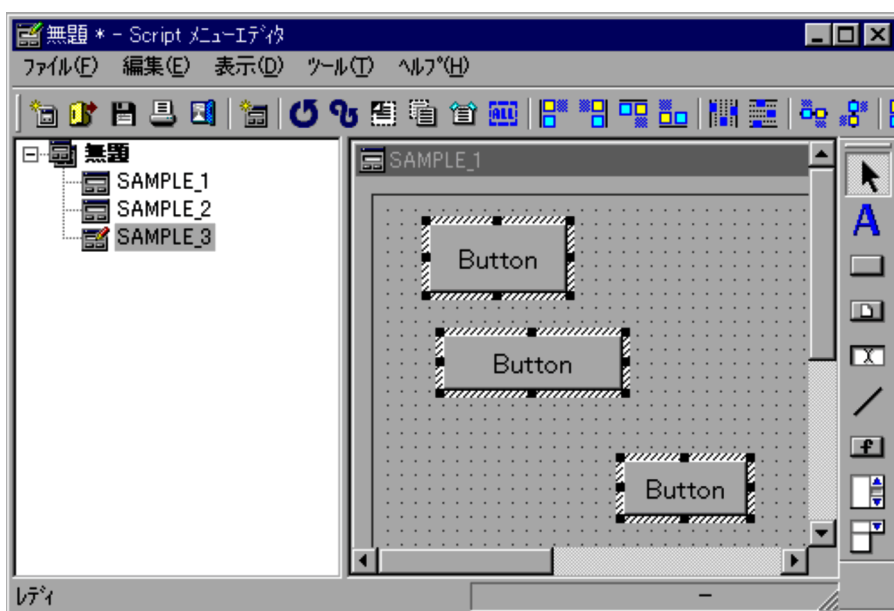
選択した 3 個以上のコントロールを均等な間隔で並べます。

左右方向に並べる方法と、上下方向に並べる方法があります。

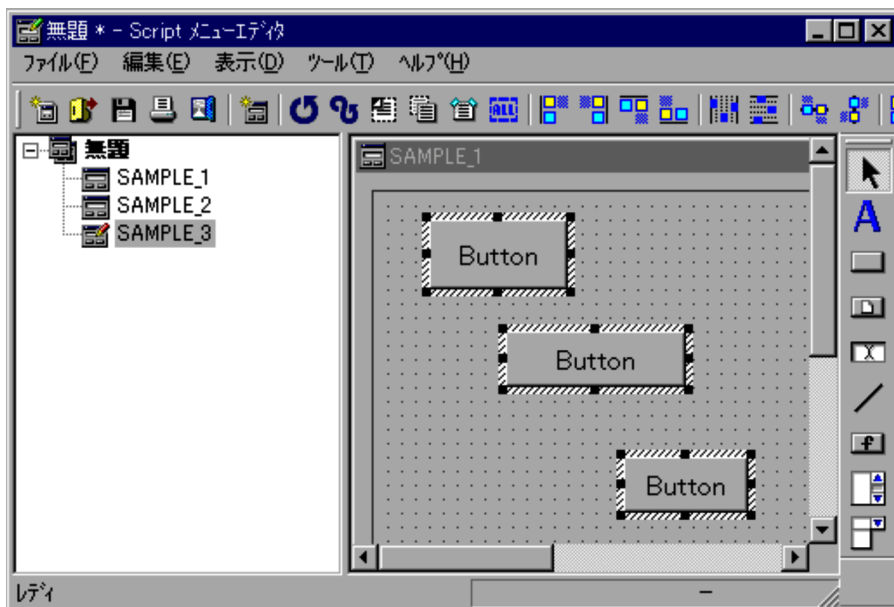
操作

コントロールを左右方向に均等な間隔で並べる

1. メニューフォーム上でコントロールを 3 個以上選択する。



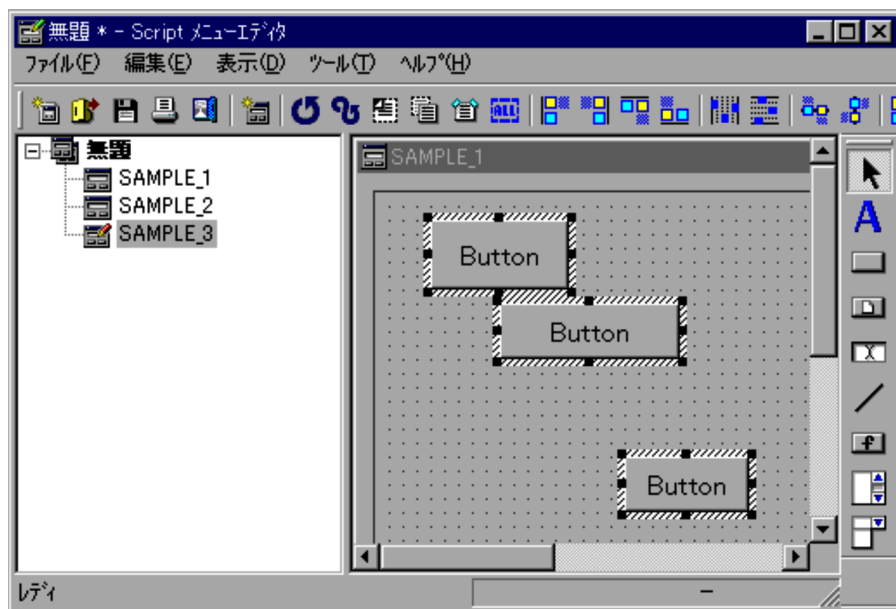
2. [編集] - [レイアウト] - [スペースの均等化] - [左右方向] を選択する。  
 選択したうちでいちばん左にあるコントロールといちばん右にあるコントロールを基準として、その間にあるコントロールが均等に並びます。



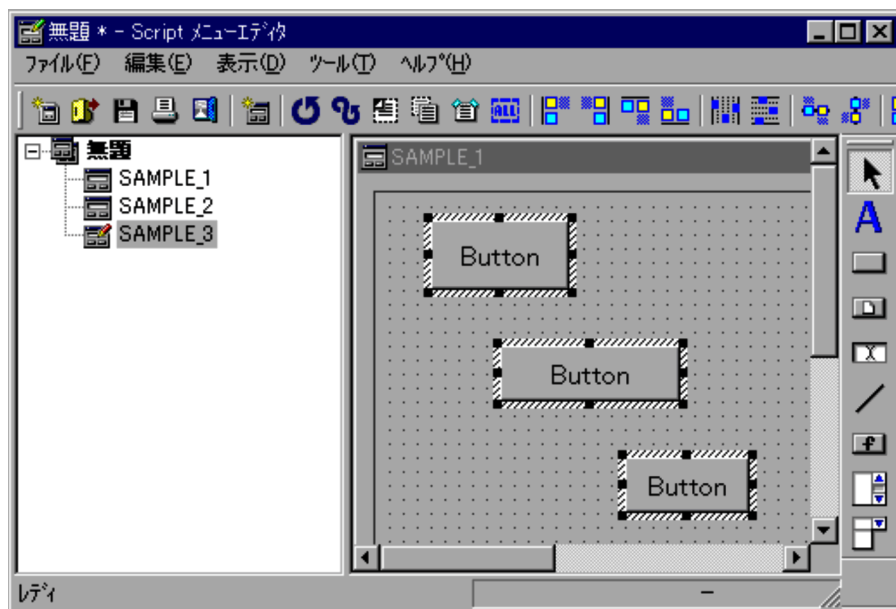
コントロールを上下方向に均等な間隔で並べる

### 3. JP1/Script の操作

1. メニューフォーム上でコントロールを 3 個以上選択する。



2. [編集] - [レイアウト] - [スペースの均等化] - [上下方向] を選択する。  
選択したうちでいちばん上にあるコントロールといちばん下にあるコントロールを基準として、その間にあるコントロールが均等に並びます。



- コントロールを 3 個以上選択しないと、メニューがグレイアウトされてメニューを選択できません。

### 3.6.10 コントロールをメニューフォームの中央に並べる

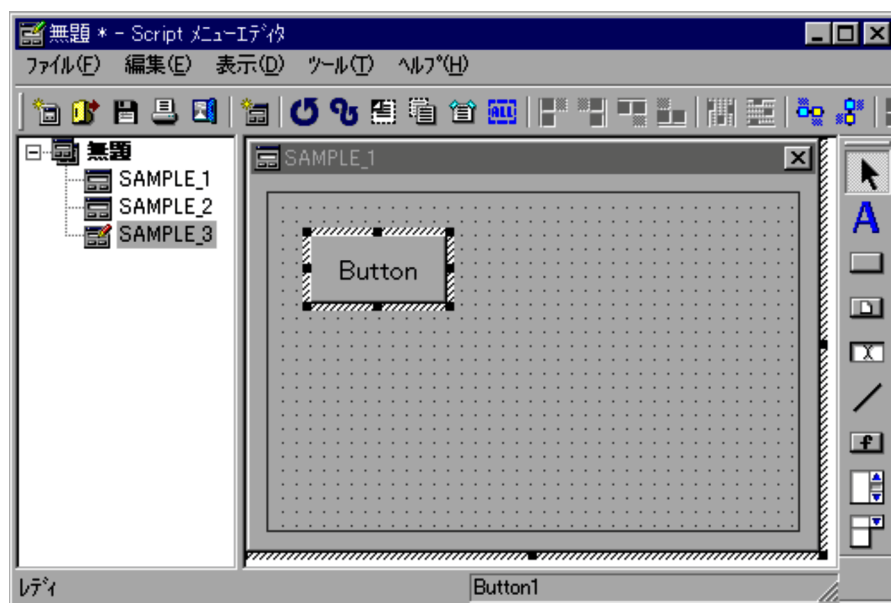
選択したコントロールをメニューフォームの中央に並べます。

メニューフォームの水平方向の中央に並べる方法と、垂直方向の中央に並べる方法があります。

操作

水平方向の中央に並べる

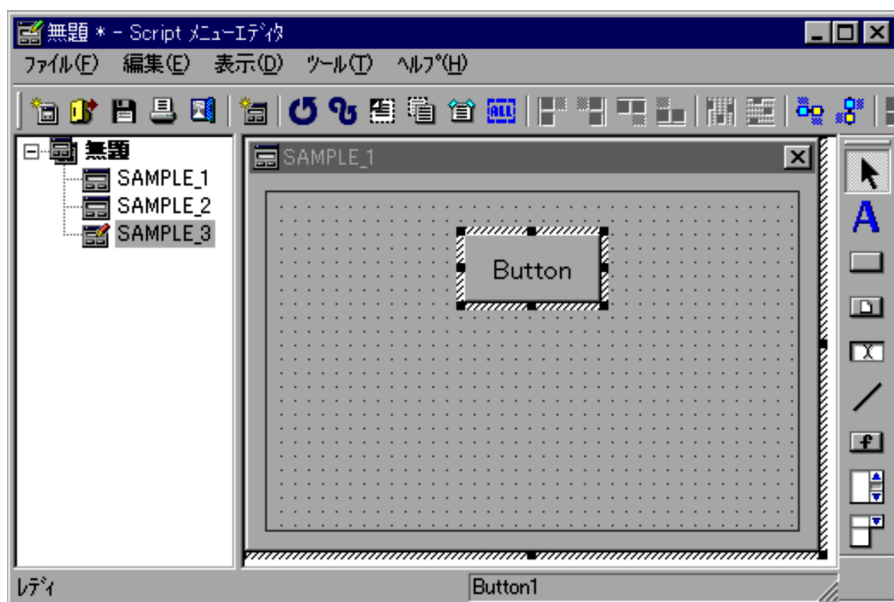
1. メニューフォーム上でコントロールを選択する。



2. [編集] - [レイアウト] - [メニューフォーム内での整列] - [左右中央] メニューを選択する。

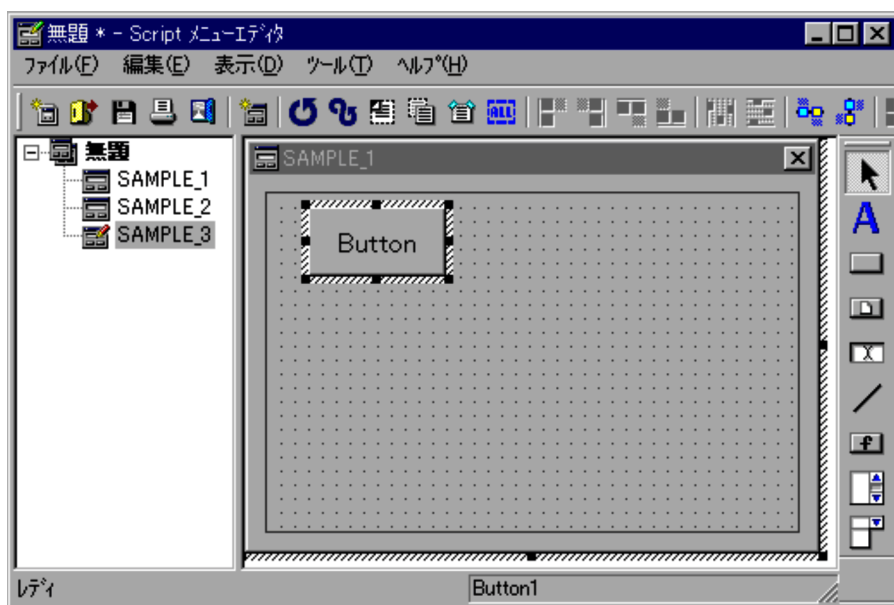
メニューフォームの水平方向の中央に選択したコントロールが並びます。

### 3. JP1/Script の操作



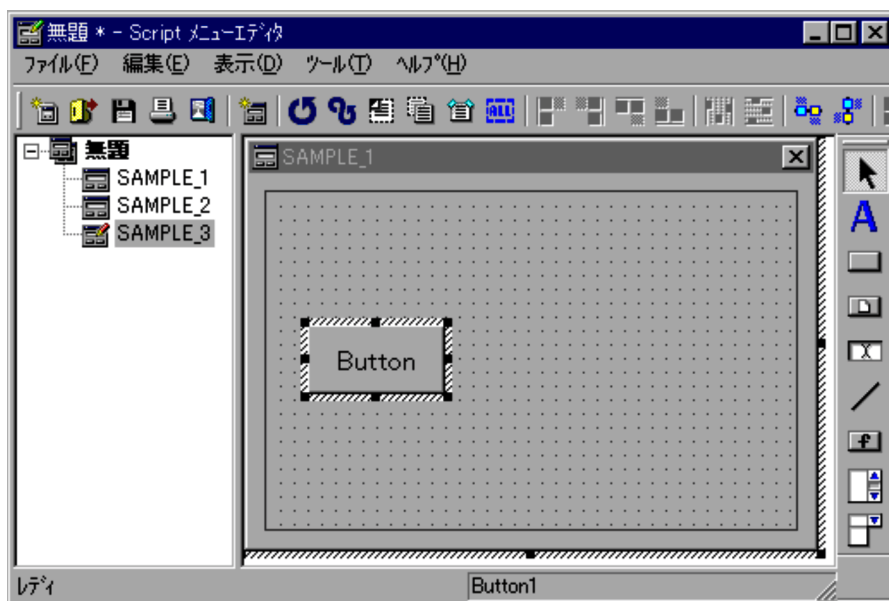
垂直方向の中央に並べる

1. メニューフォーム上でコントロールを選択する。



2. [編集] - [レイアウト] - [メニューフォーム内での整列] - [上下中央]メニューを選択する。

メニューフォームの垂直方向の中央に選択したコントロールが並びます。



### 3.6.11 コントロールの大きさをそろえる

コントロールの大きさを他のコントロールの大きさにそろえます。

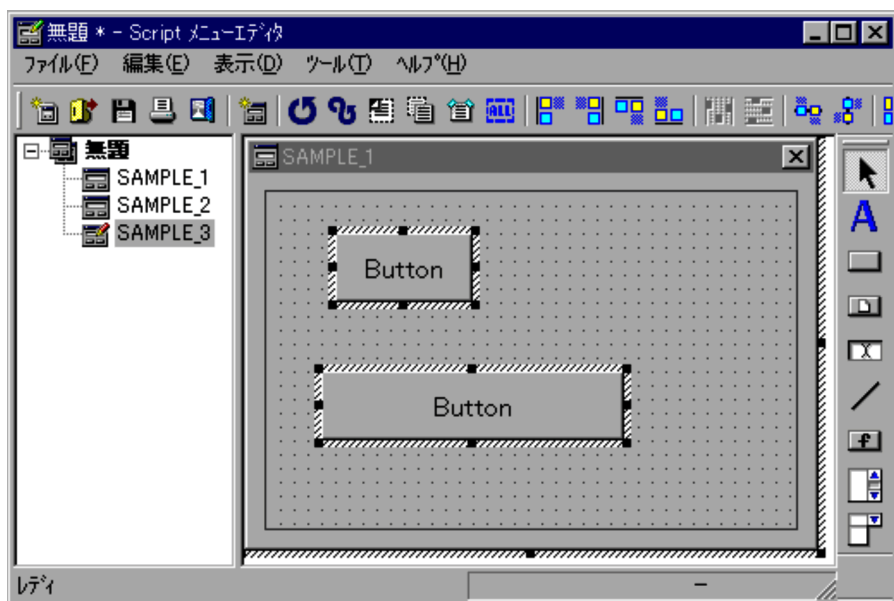
幅をそろえる方法，高さをそろえる方法，幅と高さをそろえる方法があります。

操作

幅をそろえる

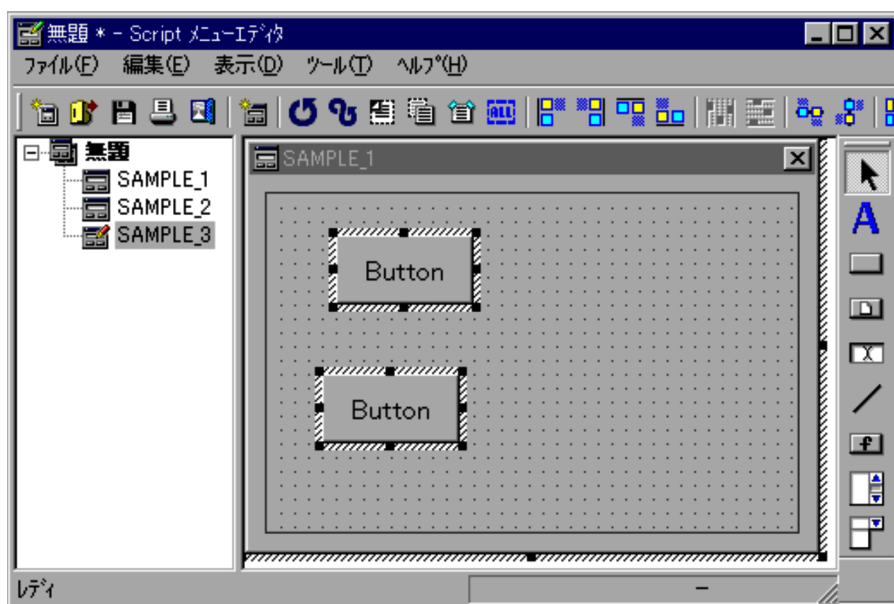
1. メニューフォーム上で 2 個以上コントロールを選択する。

### 3. JP1/Script の操作



2. [ 編集 ] - [ レイアウト ] - [ 同じサイズに設定 ] - [ 幅 ] を選択する。

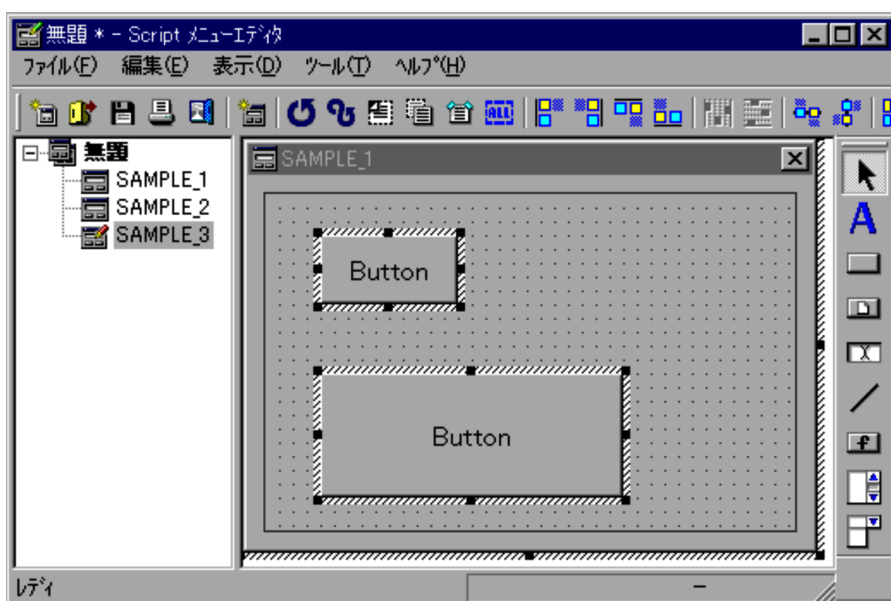
最初に選択したコントロールと同じ幅に、あとから選択したコントロールの幅がそろいます。



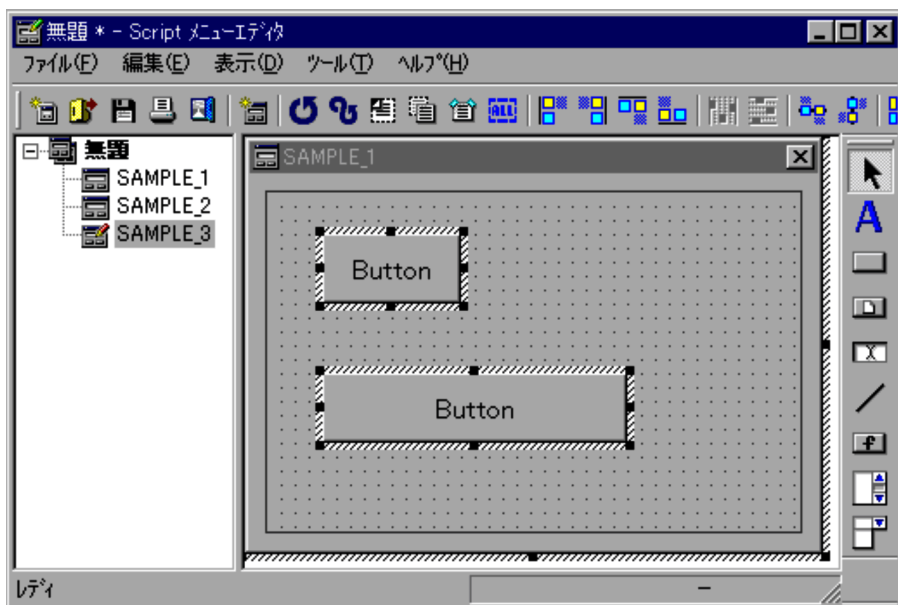


高さをそろえる

1. メニューフォーム上で 2 個以上コントロールを選択する。



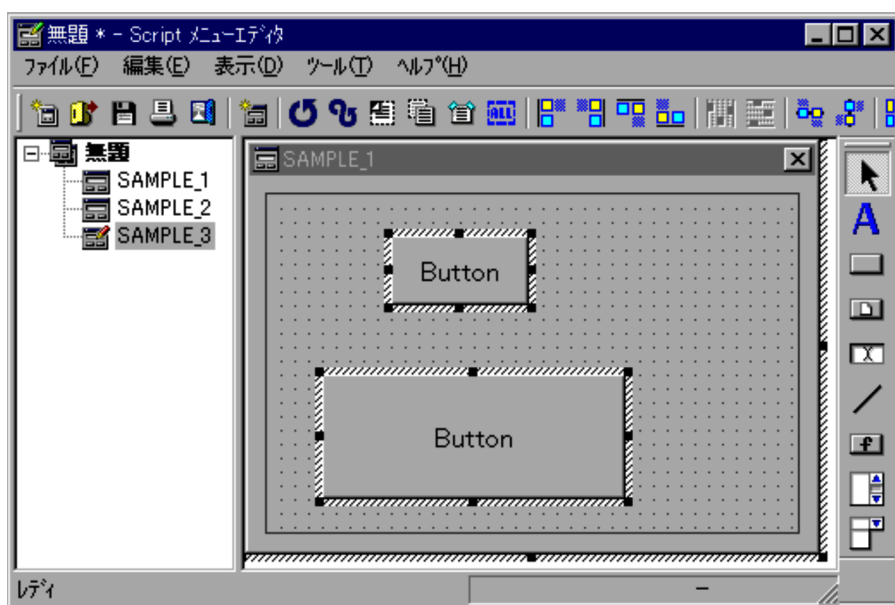
2. [編集] - [レイアウト] - [同じサイズに設定] - [高さ] を選択する。  
最初を選択したコントロールと同じ高さに、あとから選択したコントロールの高さがそろいます。



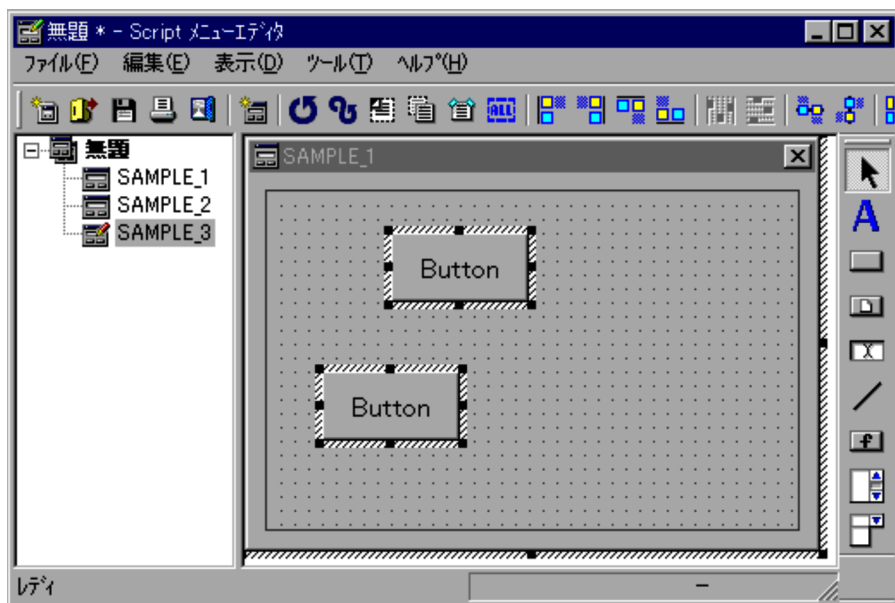
### 3. JP1/Script の操作

幅と高さをそろえる

1. メニューフォーム上で 2 個以上コントロールを選択する。



2. [ 編集 ] - [ レイアウト ] - [ 同じサイズに設定 ] - [ 幅と高さ ] を選択する。  
最初を選択したコントロールと同じ幅と高さに、あとから選択したコントロールの幅と高さがそろいます。



- コントロールを 2 個以上選択しないと、メニューがグレーアウトされてメニューを選

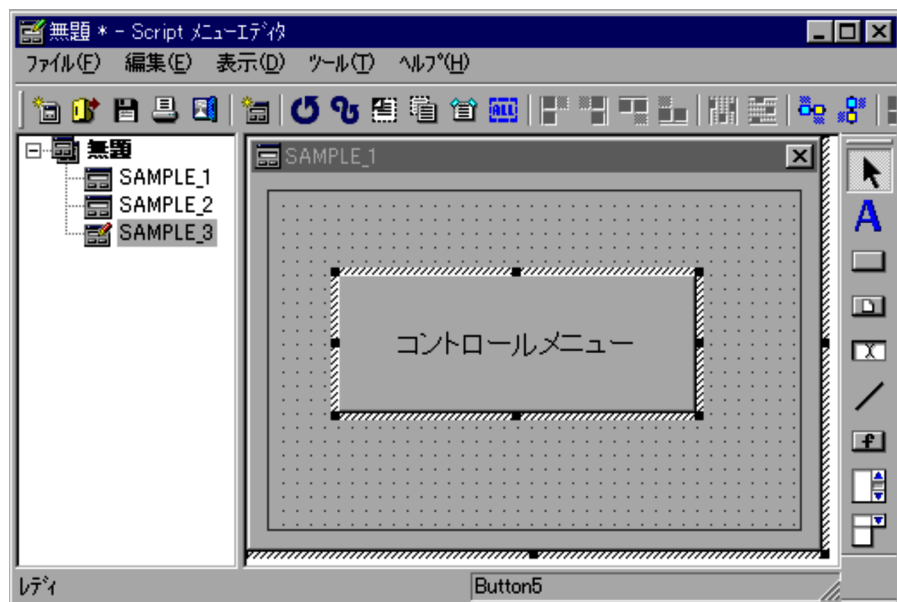
扱できません。

### 3.6.12 コントロールをテキストの大きさに合わせる

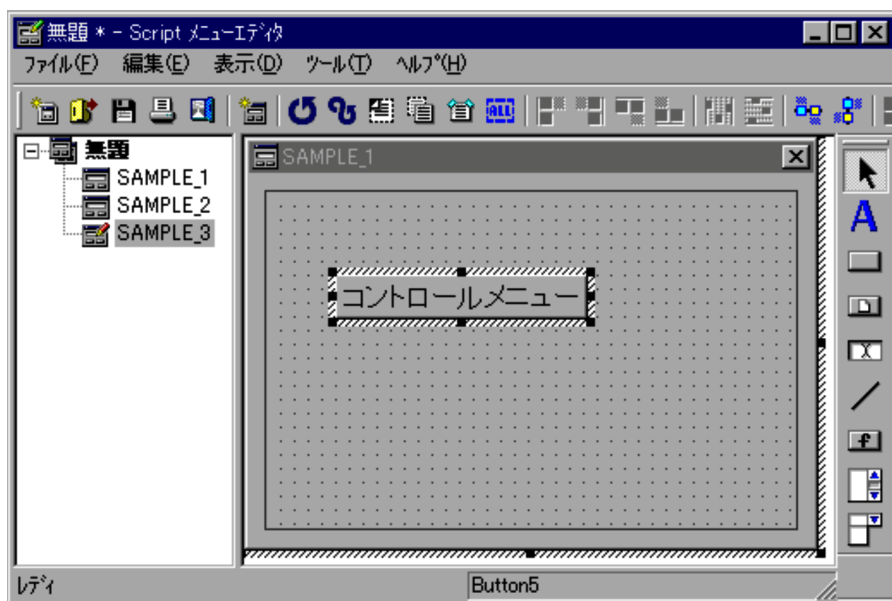
選択したコントロールの大きさをテキストの大きさに合わせます。

操作

1. メニューフォーム上でコントロールを選択する。



2. [編集] - [レイアウト] - [テキストへのサイズ合わせ] メニューを選択する。  
選択したコントロールの大きさがテキストの大きさにそろいます。



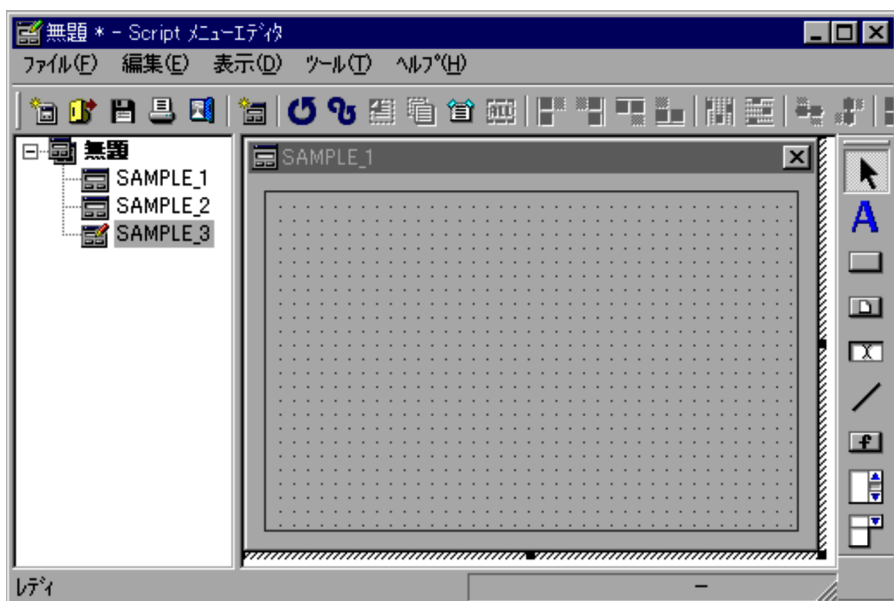
- コントロールが選択されていないと、メニューがグレースアウトされてメニューを選択できません。
- 選択されたコントロールが、スタティックコントロール、ボタンコントロール、またはファイル参照ボタンコントロール以外の場合にはメニューがグレースアウトされてメニューを選択できません。

### 3.6.13 メニューフォーム上にグリッドを表示する

メニューフォーム上にグリッドを表示します。グリッドの間隔を設定できます。

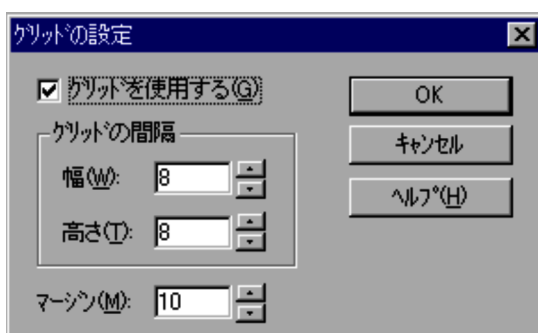
操作

1. [ 編集 ] - [ レイアウト ] - [ グリッドの設定 ] を選択する。  
[ グリッドの設定 ] ダイアログが表示されます。



ダイアログの設定方法については、「4.4.40 グリッドの設定ダイアログ」を参照してください。

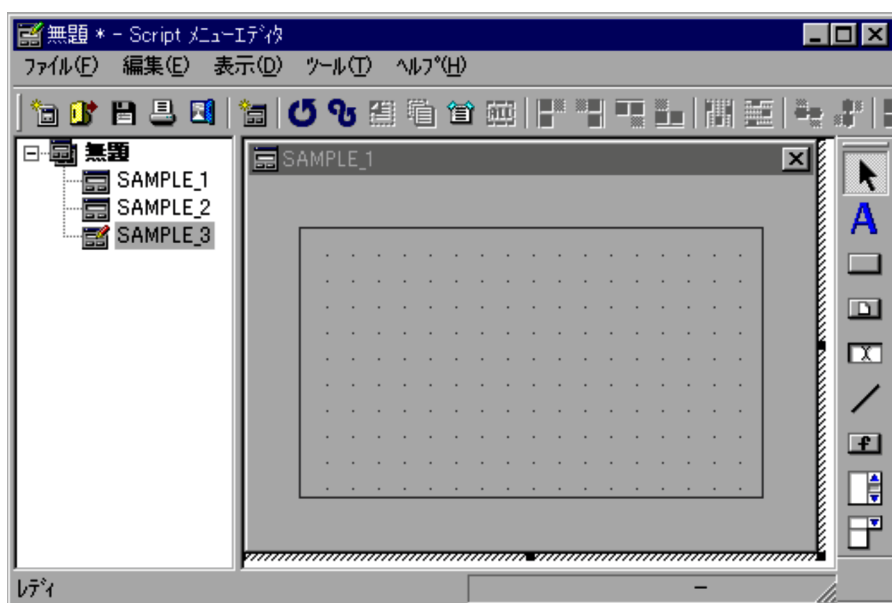
2. [ グリッドを使用する ] をチェックする。



3. [ 幅 ], [ 高さ ], および [ マージン ] を設定する。  
[ 幅 ], [ 高さ ], および [ マージン ] は、ピクセル単位で設定できます。



4. [ OK ] をクリックする。  
グリッドがメニューフォーム上に表示されます。



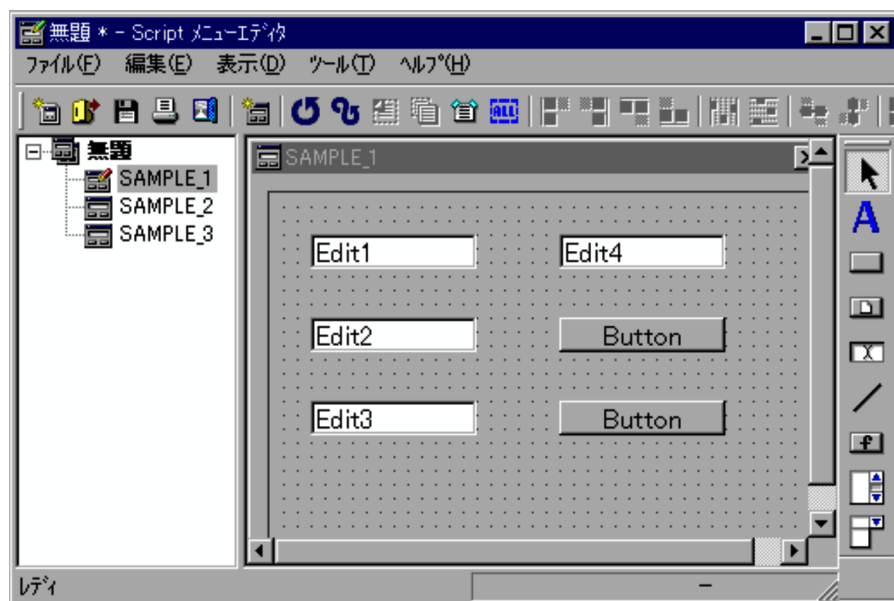
- コントロールをメニューフォーム上にドラッグすると、コントロールはグリッド上だけに配置できません。グリッドとは関係なくコントロールを配置する場合には、Alt キーを押しながらコントロールをドラッグしてください。
- Shift キーを押しながらコントロールをドラッグすると、コントロールを縦方向と横方向にだけ移動できます。
- グリッドを表示させたくない場合は、「グリッドを使用する」のチェックを外してください。

### 3.6.14 タブの移動順序を設定する

タブの移動順序を設定します。

操作

1. メニューフォームを表示する。



2. [ 編集 ] - [ レイアウト ] - [ タブオーダーの設定 ] を選択する。  
[ タブオーダーの設定 ] ダイアログが表示されます。



ダイアログの設定方法については、「4.4.41 タブオーダーの設定ダイアログ」を参照してください。

3. フィールド名を選択する。  
移動したいフィールド名を選択します（ここでは、Edit4 を選択）。



4. フィールドを移動させる。

選択したフィールドを上に移動させたい場合には [ 上に移動 ] ボタン，下に移動させたい場合には [ 下に移動 ] ボタン，フィールドの座標の位置の順序でソートしたい場合には [ 座標位置でソート ] を選択します。



5. [ OK ] をクリックする。

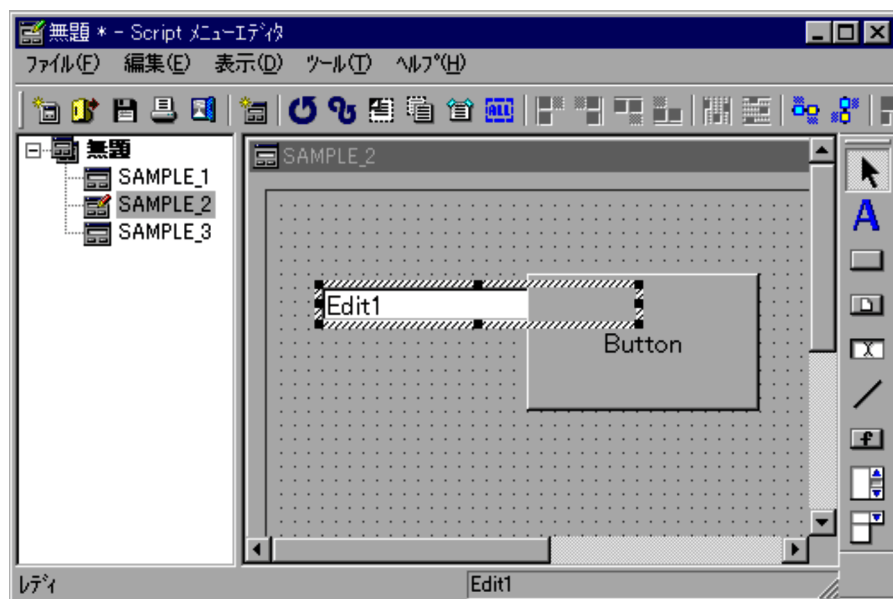
### 3.6.15 選択したコントロールを前面に移動する

選択したコントロールを前面に移動します。

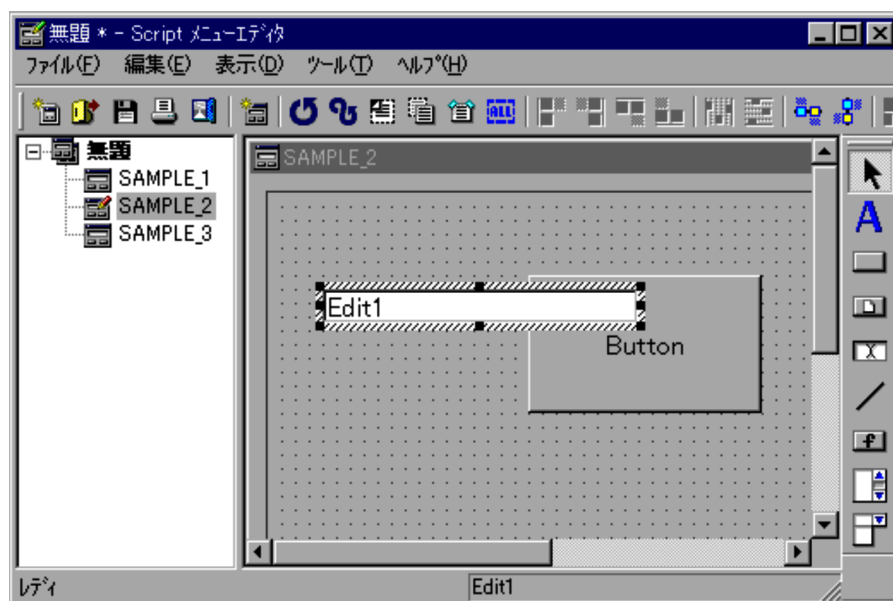
操作

1. メニューフォーム上のコントロールを選択する。





2. [編集] - [レイアウト] - [前面へ移動]を選択する。  
 選択したコントロールがいちばん手前に移動されます。



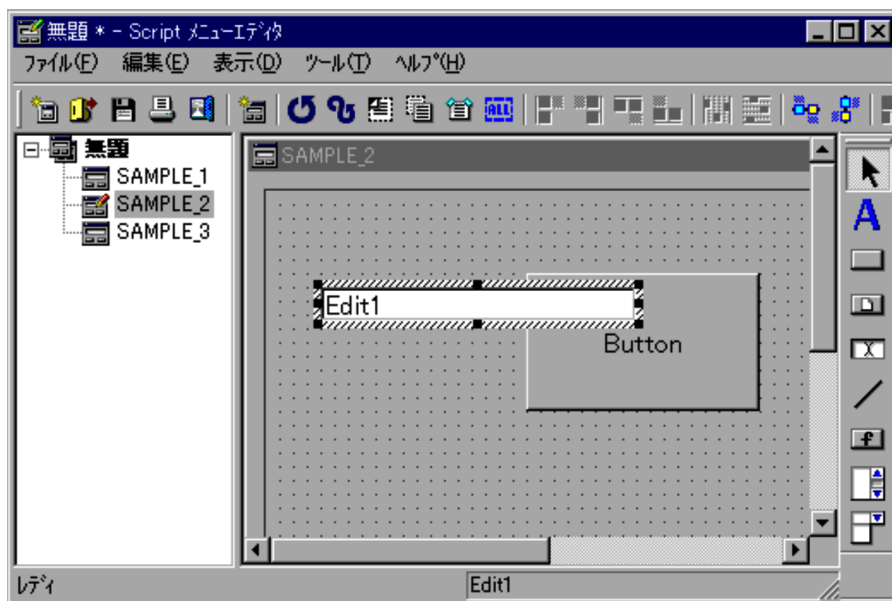
- ・ コントロールが選択されていないと、メニューがグレースアウトされてメニューを選択できません。

### 3.6.16 選択したコントロールを背面に移動する

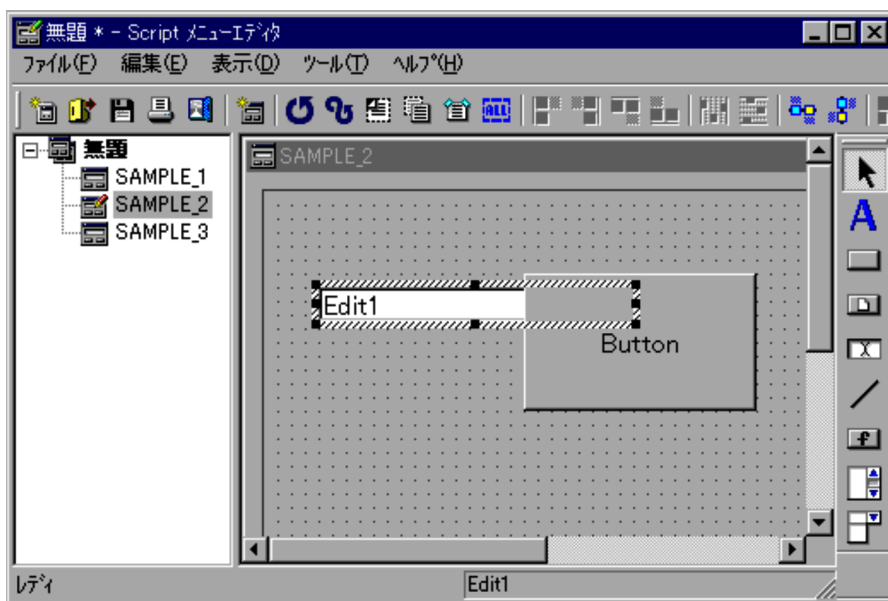
選択したコントロールをいちばん後ろに移動します。

操作

1. メニューフォーム上のコントロールを選択する。



2. [編集] - [レイアウト] - [背面へ移動]メニューを選択する。  
選択したコントロールが背面に移動します。



- コントロールが選択されていないと、メニューがグレイアウトされてメニューを選択できません。

### 3.6.17 コマンドのプロパティを設定する

コマンドのプロパティを設定します。

操作

1. [ ツール ] - [ コマンドプロパティの設定 ] を選択する。  
[ コマンドプロパティの設定 ] ダイアログが表示されます。



ダイアログの設定方法については、「4.4.37 コマンドプロパティの設定ダイアログ」を参照してください。

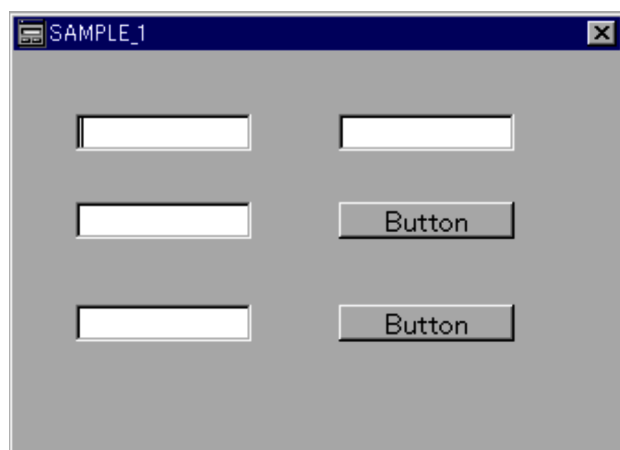
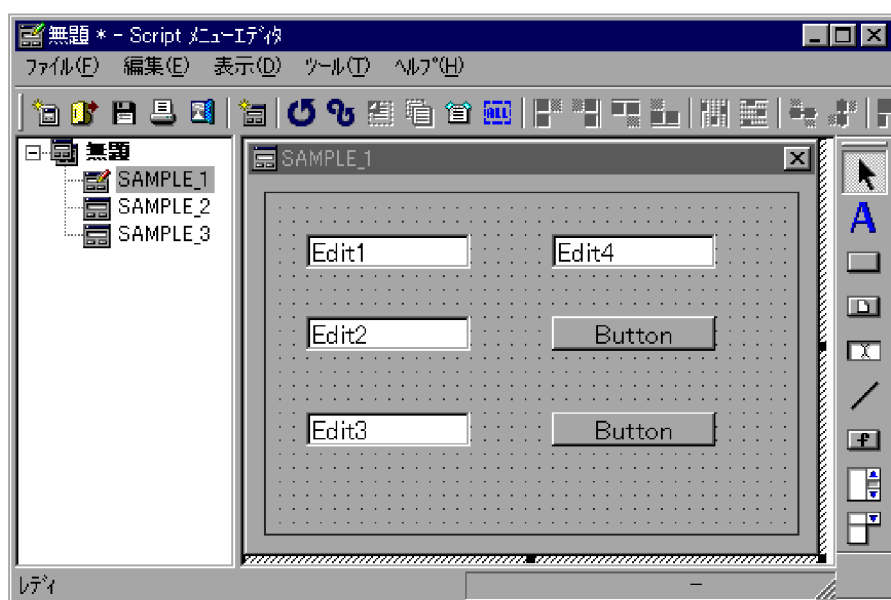
2. 必要な情報を入力する。
3. [ OK ] をクリックする。  
コマンドのプロパティが設定されます。

### 3.6.18 メニューフォームをテスト表示する

メニューフォームをテスト表示します。

操作

1. [ ツール ] - [ テスト表示 ] を選択する。



作成したメニューフォームが表示されます。

### 3.6.19 メニューフォームの内容を印刷する

作成したメニューフォームの内容を印刷します。

操作

1. [ ファイル ] - [ メニューフォームの印刷 ] メニューを選択する。  
[ メニューフォームの印刷情報 ] ダイアログが表示されます。

### 3. JP1/Script の操作

2. 必要な情報を設定する。

3. [ 印刷実行 ] ボタンをクリックする。

メニューフォームの内容が印刷されます。

- 操作 3. で [ キャンセル ] ボタンをクリックすると , ダイアログを閉じます。

## 3.7 プロセスビューア の操作

実行中のスクリプトプロセスを監視・制御できます。

また、他のコンピュータで実行中のプロセスも監視・制御できます。

補足

Windows Vista では、JP1/Script サービスが実行中以外の状態でプロセスビューアを起動しようとする、次の警告ダイアログを表示した後、プロセスビューアを起動します。

「JP1/Script サービスの状態が実行中以外のため、サービスから起動したプロセスおよび他ユーザで実行中のプロセスが表示できません。」

### 3.7.1 プロセスビューア の操作項目

プロセスビューアアイコンを起動する、またはマネージャの [ ツール ] - [ プロセスビューアの起動 ] を選択すると、[ Script プロセスビューア ] メッセージボックスが表示されます。

[ Script プロセスビューア ] メッセージボックスを、図 3-8 に示します。

図 3-8 [ Script プロセスビューア ] メッセージボックス



クライアントエリア

#### (1) コンピュータ名

接続先コンピュータの名称を指定します。[ 選択 ] ボタンをクリックして、表示されるコンピュータの選択ダイアログで、接続するコンピュータの名称を選択できます。指定したコンピュータとの接続でエラーが発生した場合は、エラーメッセージが表示されます。また、指定したコンピュータと接続できない場合は、未接続状態になりクライアントエリアには何も表示されません。なお、接続したコンピュータの名称は履歴としてコンボ

ボックスに残ります。

## (2) クライアントエリア

接続先のコンピュータで実行されているスクリプトファイルの情報（プロセス）が表示されます。

### ファイル名

現在実行されているスクリプトプロセスのスクリプトファイル名が表示されます。  
NetExec コマンドから呼び出された実行ファイルは "NetExec - ファイル名" の形式で表示されます。

### ステータス

スクリプトプロセスの実行状態が表示されます。  
実行状態を示す値は次のとおりです。  
実行中：プロセスは実行中です。  
終了中：プロセスは終了処理中です。

### PID

スクリプトプロセスのプロセス ID が表示されます。

### 起動日

スクリプトプロセスの起動日が "mm/dd" の形式（mm: 月, dd: 日）で表示されます。

### 起動時刻

スクリプトプロセスの起動時刻が "HH:MM:SS" の形式（HH: 時, MM: 分, SS: 秒）で表示されます。

### 稼働時間

スクリプトプロセスを起動してからの稼働時間が "HH:MM:SS" の形式（HH: 時, MM: 分, SS: 秒）で表示されます。

### 起動種別

スクリプトプロセスの起動種別が表示されます。  
ログオン：ログオン空間で起動されています。  
サービス：サービス空間で起動されています。

### フォルダ

スクリプトファイルのフォルダ名が表示されます（カレントフォルダ）。

[ Script プロセスビューア ] メッセージボックスには、次のボタンが表示されます。

### [ オプション ] ボタン

クライアントエリアのプロセス表示の更新間隔を設定します。更新の間隔は 1 秒から 60 秒の間で設定します。

### [ プロセスの終了 ] ボタン

接続先のコンピュータで実行されているスクリプトファイルを終了します（プロセ



スの終了)。

プロセスを終了するときは、クライアントエリアで終了するプロセスを選択して [ プロセスの終了 ] ボタンをクリックします。表示される [ JP1/Script プロセスの終了 ] メッセージで [ はい ] ボタンをクリックすると、クライアントエリアで選択したプロセスが終了します。

このボタンを無効にしたい場合は、次に示すレジストリキーの値に 1 を設定してください。

[ レジストリキー ]

```
HKEY_LOCAL_MACHINE¥SOFTWARE¥HITACHI¥JP1/  
Script¥SPTHView¥Option
```

[ 値 ]

```
TermButton (DWORD)
```

[ 値のデータ ]

0 : [ プロセスの終了 ] ボタンを有効にする (初期値)

1 : [ プロセスの終了 ] ボタンを無効にする

[ 終了 ] ボタン

[ Script プロセスビューア ] メッセージボックスが閉じます。

[ ヘルプ ] ボタン

[ Script プロセスビューア ] メッセージボックスのオンラインヘルプが表示されます。

### 3.7.2 実行中のスクリプトプロセスを一覧表示する

現在実行されているスクリプトプロセスを一覧表示し、プロセスの監視や制御ができます。他のコンピュータで実行されているスクリプトプロセスを一覧表示することもできます。



#### 操作

- [ オプション ] ボタンを選択すると, [ Script プロセスビューアオプション ] ダイアログが表示されます。ダイアログの設定方法については「3.7.3 スクリプトプロセスの表示更新間隔を指定する」を参照してください。
- 一覧表示されたスクリプトプロセスを一つ, または複数選択して [ プロセスの終了 ] ボタンを選択すると, 選択したスクリプトプロセスを強制終了します。
- [ 終了 ] ボタンを選択すると, [ Script プロセスビューア ] メッセージボックスを閉じます。

### 3.7.3 スクリプトプロセスの表示更新間隔を指定する

[ Script プロセスビューア ] メッセージボックスで [ オプション ] ボタンを選択すると, [ オプション ] ダイアログが表示されます。



#### ダイアログの項目

##### 表示の更新間隔

[ Script プロセスビューア ] メッセージボックスで一覧表示されているスクリプトプロセスの表示の更新間隔を秒単位で指定します。

1 ~ 60 の範囲で指定してください。デフォルトでは「1」が設定されています。

ダイアログでの操作

- [ OK ] ボタンを選択すると、指定した内容が設定されて [ Script プロセスビュー オプション ] ダイアログを閉じます。
- [ キャンセル ] ボタンを選択すると、内容を変更しないで [ Script プロセスビュー オプション ] ダイアログを閉じます。

## 3.8 実行環境ファイルコンバータの操作

---

### 3.8.1 実行環境ファイルコンバータの概要

UNIX 版 JP1/Script への実行環境ファイル移行のためのツールです。このプログラムによって実行環境ファイルから実行環境構文ファイル（実行環境ファイル内容がテキストファイル形式で記述されているファイル）を作成できます。

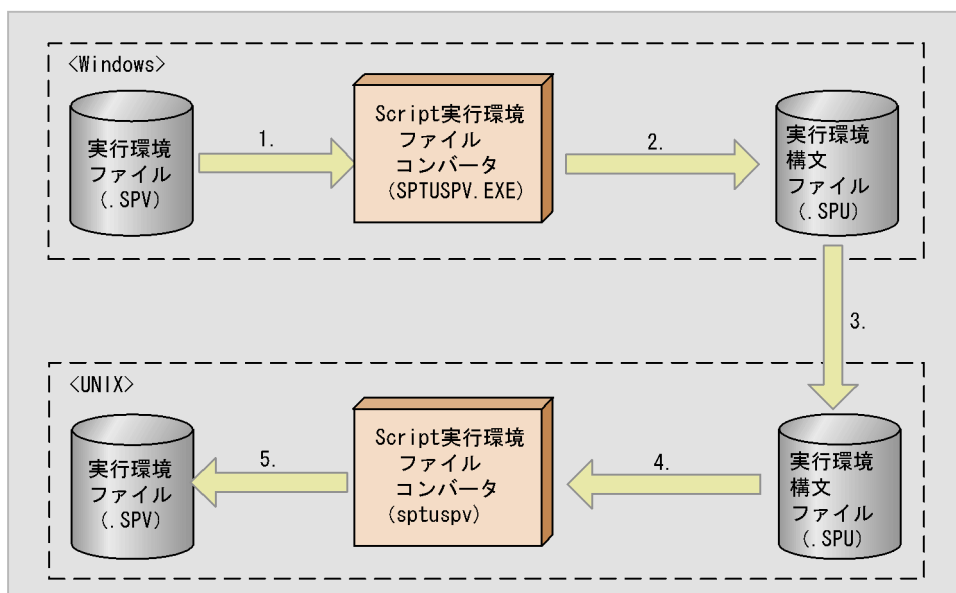
### 3.8.2 実行環境ファイルの変換手順

Windows 上の実行環境ファイルを UNIX 上に移行する手順は次のとおりです。

操作

1. 変換する実行環境ファイル（.SPV）を「Script 実行環境ファイルコンバータ」で指定して実行します。
2. 実行環境構文ファイル（.SPU）が作成されます。
3. 作成された実行環境構文ファイル（.SPU）を FTP 転送で UNIX 上に移動します。
4. 手順 3. で移動した実行環境ファイル（.SPV）を「Script 実行環境ファイルコンバータ」に指定して実行します。
5. UNIX 上で使用できる実行環境ファイル（.SPV）が作成されます。

図 3-9 実行環境ファイルの移行の流れ



(凡例)

実行環境ファイル：スクリプトの実行環境が設定されているファイル。

実行環境構文ファイル：スクリプトの実行環境が設定されているファイル。

### 3.8.3 実行環境ファイルコンバータの実行方法

実行環境ファイルコンバータを実行するには、コマンドプロンプトを起動してからコマンドラインに次の形式でコマンドを入力します。

#### (1) 実行環境ファイル (.SPV) から実行環境構文ファイル (.SPU) への変換

変換元ファイルに実行環境ファイルの拡張子「.SPV」を指定した場合、この変換を行います。

形式

```
sptuspv /i fromFileName( /o toDirName)
```

説明

*fromFileName*

実行環境構文ファイル (.SPU) に変換する実行環境ファイル (.SPV) を指定します。ラベル名にワイルドカード指定ができます。拡張子に指定できるのは「.SPU」、または「.SPV」だけです。これらの拡張子以外を指定した場合はエラーになります。

*toDirName*

変換後の実行環境構文ファイル (.SPV) が出力されるフォルダを指定します。  
指定を省略した場合、変換元ファイルと同じフォルダに出力されます。変換後のファイルがすでにある場合は、今までの内容はすべて失われます。

## (2) 実行環境構文ファイル (.SPU) から実行環境ファイル (.SPV) への変換

変換元ファイルに実行環境構文ファイルの拡張子「.SPU」を指定した場合、この変換を行います。

### 形式

```
sptuspv /i fromFileName( /o toDirName)( operand)
```

### 説明

#### *fromFileName*

実行環境ファイル (.SPV) に変換する実行環境構文ファイル (.SPU) を指定します。ラベル名にワイルドカード指定ができます。拡張子に指定できるのは「.SPU」、または「.SPV」だけです。これらの拡張子以外を指定した場合はエラーになります。

#### *toDirName*

変換後の実行環境ファイル (.SPV) が出力されるフォルダを指定します。指定を省略した場合、変換元ファイルと同じフォルダに出力されます。変換後のファイルがすでにある場合は、今までの内容はすべて失われます。

#### *operand*

次のオペランドを指定できます。

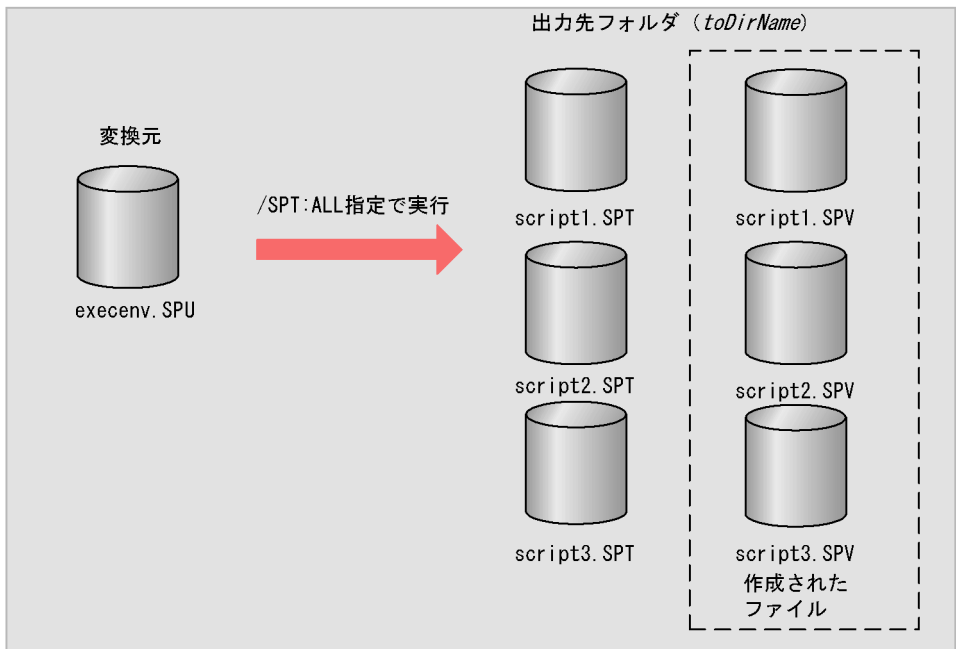
/SPT:ALL

一つの実行環境構文ファイル (.SPU) から出力先フォルダに存在するすべてのスクリプトファイル (.SPT) に適用させたい場合に指定します。 *toDirName* , または *toDirName* 省略時は *fromFileName* の後ろに指定します。

/SPT:ALL を指定した場合、変換元ファイル名にワイルドカードは指定できません ( オペランドの組み合わせエラーになります )。

/SPT:ALL を指定した場合の実行環境ファイル作成の流れを次の図に示します。

図 3-10 /SPT:ALL を指定した場合の実行環境ファイル作成の流れ



変換元ファイルにexecenv. SPU、出力先フォルダにDir、オペランド/SPT:ALLを指定して実行した場合、出力先に存在したscript1～script3. SPTに対して、実行環境ファイルscript1～script3. SPVがexecenv. SPUの内容で作成されます。オペランドに/SPT:ALLを指定しない場合は、出力先フォルダにexecenv. SPUだけが作成されます。

#### 戻り値

正常に終了した場合は、0 が返されます。正常に終了しなかった場合は、そのエラーコードが返されます。

#### 表示するメッセージ

sptuspv では、表 3-24 に示すメッセージがコンソールに表示されます。コンソールがない状態で実行する場合などは、実行時のオペランドに標準出力装置を割り当て、実行後に割り当てたファイルを参照してエラー内容を確認します。メッセージはファイルの変換単位で表示されます。

実行環境ファイル、実行環境構文ファイルにエラーがある場合は、そのファイルの変換を中止し、次のファイルを変換します。

表 3-24 実行環境ファイルコンバータの戻り値とメッセージ

項番	戻り値	戻り値の意味	メッセージ
1	0	正常に終了しました(ワイルドカード指定時、対象ファイルすべての変換が正常に終了しました)	ファイルを変換しました。(変換されたファイル名)

項番	戻り値	戻り値の意味	メッセージ
2	1	実行環境構文ファイルの構文に誤りがあります。	実行環境構文ファイルの構文に誤りがあります。(エラーとなったファイル名)
3	2	範囲外の数値が指定されています。	実行環境構文ファイル中に範囲外の値が設定されています。(エラーとなったファイル名)
4	3	ファイル形式に誤りがあります。	ファイルが壊れているかまたはファイルの形式が異なります。(エラーとなったファイル名)
5	-	出力先フォルダに変換するファイルがすでに存在します。	変換後のファイルが出力先ディレクトリにすでに存在しました。このファイルの内容はすべて失われました。(上書きされたファイル名)
6	4	入力ファイルが存在しません。	指定されたファイルが見つかりません。(エラーとなったファイル名)
7	5	入力ファイルアクセスエラー。	システムエラーメッセージ。(エラーとなったファイル名)
8	20	出力先ディレクトリが見つかりません。	出力先のディレクトリが見つかりません。(エラーとなったディレクトリ名)
9	21	出力ファイルアクセスエラー。	システムエラーメッセージ。(エラーとなったファイル名)
10	22	SPV, SPU 以外のファイルを指定しました。	指定できないファイルです。(エラーとなったファイル名)
11	23	パラメタを何も指定していないか、指定した内容が不正です。	パラメタの指定が不正です。
12	24	オペランドが不正です。	指定したパラメタの組み合わせに誤りがあります。
13	25	メモリが不足しています。	メモリが不足しました。
14	26	変換対象ファイルがありません。	出力先のディレクトリに対象となるスクリプトファイルが存在しませんでした。
15	27	指定したディレクトリが見つかりません。	実行環境構文ファイル中に指定しているディレクトリが見つかりません。(エラーとなったファイル名)
16	-	出力ファイルの上書き時に書き込みエラーが発生しました。	既に存在するファイルに対して書き込みエラーが発生したため、ファイルを削除しました。(エラーとなったファイル名)
17	99	一部変換が失敗しました。	-

## 注

- 入力ファイルのワイルドカード指定時、項番 2, 3, 4, 6, 7 が発生した場合に返ります。
- そのファイルの変換は行われなかったが、そのほかの対象ファイルの変換は行った場合に返ります。
- 項番 2, 3, 4, 6, 7 以外のエラー時は、その時点で処理が中断されます。

## 3.8.4 実行環境構文ファイル (.SPU) の概要

実行環境構文ファイル (.SPU) とは、スクリプトの実行環境が設定されている実行環境ファイルの内容をテキスト形式で表したファイルのことです。



実行環境構文ファイルの概要を表 3-25 に示します。

表 3-25 実行環境構文ファイルの概要

ファイルの種類	ファイル名	拡張子	ファイルの内容
コンバータが扱うファイル	実行環境構文ファイル	.SPU	実行環境ファイルの内容をテキスト形式で表したファイルです。実行環境の設定をテキストエディタから行うことを目的としています。

### 3.8.5 実行環境構文ファイル (.SPU) の詳細

実行環境構文ファイルの形式を図 3-11 に示します。

図 3-11 実行環境構文ファイルのフォーマット

```

Start_CommandLine=/ABC/123      ... 1.
Start_Work_Dir=C:¥Temp          ... 2.
Start_MultiRun=0                ... 3.
Start_Type=0                    ... 4.
Start_Date_Appo=0               ... 5.
Start_Week_Appo=0               ... 6.
Start_Week=1                    ... 7.
Start_Day_Appo=0                ... 8.
Start_Day=1                     ... 9.
Start_TimeAppo=0                ...10.
Start_Time_Hour=9               ...11.
Start_Time_Minuts=00            ...12.
End_StopTimes=0                 ...13.
Trace_Dir=C:¥temp               ...14.
Trace_MaxLines=1024             ...15.
Trace_MaxColumns=150            ...16.
Trace_user_MaxLines=1024        ...17.
Trace_user_MaxColumns=150       ...18.

```

#### 説明

1. コマンドライン。  
トレースファイル，およびイベントログの出力の有無が Windows 版の形式で記述されている場合は，UNIX 版の形式で変換されます。
2. 作業フォルダ。実行環境ファイルから実行環境構文ファイルに変換した場合，常に空とします。実行環境ファイルから実行環境ファイルに変換した場合はこの値は有効とし，存在しないフォルダ名の場合実行環境ファイル変換時に構文エラーとなり実行環境ファイルは作成されません。
3. 複数起動の許可。許可する場合は 1，許可しない場合は 0 を記述します。それ以外を記述した場合実行環境ファイル変換時に構文エラーとなり実行環境ファイルは作成されません。
4. 起動のタイプ。ログオンの場合は 0，サービスの場合は 1 を記述します。それ以外を記述した場合実行環境ファイル変換時に構文エラーとなり実行環境ファイル

は作成されません。

Windows 版 Script コンバータの場合だけ有効です。

5. 自動起動で日の指定を行うかどうかを指定します。指定する場合は 1，指定しない場合は 0 を記述します。それ以外を記述した場合実行環境ファイル変換時に構文エラーとなり実行環境ファイルは作成されません。

Windows 版 Script コンバータの場合だけ有効です。

6. 曜日指定。指定する場合は 1，指定しない場合は 0 を記述します。それ以外を記述した場合実行環境ファイル変換時に構文エラーとなり実行環境ファイルは作成されません。

Windows 版 Script コンバータの場合だけ有効です。

7. 曜日。指定できる値は 1 から 7 までです。範囲以外の値が指定されている場合実行環境ファイル変換時に構文エラーとなり実行環境ファイルは作成されません。

Windows 版 Script コンバータの場合だけ有効です。

表 3-26 指定値に対する曜日

値	曜日
1	月曜
2	火曜
3	水曜
4	木曜
5	金曜
6	土曜
7	日曜

8. 日付指定。指定する場合は 1，指定しない場合は 0 を記述します。それ以外を記述した場合実行環境ファイル変換時に構文エラーとなり実行環境ファイルは作成されません。

Windows 版 Script コンバータの場合だけ有効です。

9. 日付。指定できる値は 1 から 31 までです。範囲外の値が指定されている場合実行環境ファイル変換時に構文エラーとなり実行環境ファイルは作成されません。

Windows 版 Script コンバータの場合だけ有効です。

10. 時刻指定。指定する場合は 1，指定しない場合は 0 を記述します。それ以外を記述した場合実行環境ファイル変換時に構文エラーとなり実行環境ファイルは作成されません。

Windows 版 Script コンバータの場合だけ有効です。

11. 時刻（時間）。指定できる値は 0 から 23 までです。範囲外の値が指定されている場合実行環境ファイル変換時に構文エラーとなり実行環境ファイルは作成されません。

Windows 版 Script コンバータの場合だけ有効です。

- 12.時刻（分）。指定できる値は0から59までです。範囲外の値が指定されている場合実行環境ファイル変換時に構文エラーとなり実行環境ファイルは作成されません。

Windows 版 Script コンバータの場合だけ有効です。

- 13.打ち切り時間。指定できる値は0から1,440までです。範囲外の値が指定されている場合実行環境ファイル変換時に構文エラーとなり実行環境ファイルは作成されません。

Windows 版 Script コンバータの場合だけ有効です。

- 14.トレースの出力先フォルダ。存在しないフォルダの場合、解析エラーとします。実行環境ファイルから実行環境構文ファイルに変換した場合、常に空とします。実行環境ファイルから実行構文ファイルに変換した場合はこの値は有効とし、存在しないフォルダ名の場合実行環境ファイル変換時に構文エラーとなり環境ファイルは作成されません。

- 15.トレース最大行数。指定できる値は100から9,999までです。範囲外の値が指定されている場合実行環境ファイル変換時に構文エラーとなり実行環境ファイルは作成されません。

- 16.トレース最大列数。指定できる値は128から1,024までです。範囲外の値が指定されている場合実行環境ファイル変換時に構文エラーとなり実行環境ファイルは作成されません。

- 17.ユーザトレース最大行数。指定できる値は100から9,999までです。範囲外の値が指定されている場合実行環境ファイル変換時に構文エラーとなり実行環境ファイルは作成されません。

- 18.ユーザトレース最大列数。指定できる値は128から1,024までです。範囲外の値が指定されている場合実行環境ファイル変換時に構文エラーとなり実行環境ファイルは作成されません。

注 UNIX 版の実行環境構文ファイルで記述した（してある）場合、実行環境ファイル（.SPV）の変換ではデフォルト値となります。

その他

実行環境構文ファイル（.SPU）の記述規則は次のとおりです。

- すべて項目は省略できることとし、記述しなかった項目にはデフォルト値を仮定します。
- 同じ項目を二つ以上記述した場合、あとに記述されている値が有効となります。
- 指定項目の記述する順序は自由です。
- 行の先頭に“#”を記述することによって、その行はコメントとして扱えます。



# 4

## ダイアログの説明

この章では、Script マネージャウィンドウ、Script エディタウィンドウ、Script トレースビューアウィンドウ、および Script メニューエディタウィンドウのダイアログについて説明します。

- 
- 4.1 Script マネージャウィンドウのダイアログ
  - 4.2 Script エディタウィンドウのダイアログ
  - 4.3 Script トレースビューアウィンドウのダイアログ
  - 4.4 Script メニューエディタウィンドウのダイアログ
-

## 4.1 Script マネージャウィンドウのダイアログ

---

Script マネージャウィンドウの操作中表示されるダイアログの一覧を、次に示します。

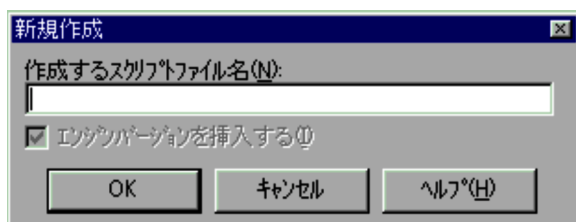
各ダイアログの項目、および操作については、4.1.1 項以降に説明します。

- 新規作成ダイアログ
- ファイルの複写ダイアログ
- ファイルの追加ダイアログ
- ファイルの上書き確認ダイアログ
- ファイルの削除の確認ダイアログ
- 名前の変更ダイアログ
- 実行ダイアログ
- 実行環境の設定（起動情報）ダイアログ
- 実行環境の設定（終了情報）ダイアログ
- 実行環境の設定（トレース情報）ダイアログ
- 実行環境（コマンドライン）の設定ダイアログ
- コマンドラインの設定ダイアログ
- 実行環境（作業フォルダ）の設定ダイアログ
- 実行環境（起動日時）の設定ダイアログ
- 実行環境（打ち切り時間）の設定ダイアログ
- 実行環境（トレース出力先フォルダ）の設定ダイアログ
- 実行環境（ユーザトレース情報）の設定ダイアログ
- フォルダの変更ダイアログ
- ファイルのプロパティダイアログ
- 自動起動の設定ダイアログ
- エディタの関連付けダイアログ
- オプション（サーバ情報）ダイアログ
- 更新情報ダイアログ
- オプション（互換性）ダイアログ
- オプション（複数起動）ダイアログ
- オプション（JP1/IM）ダイアログ
- オプション（トレース）ダイアログ
- オプション（クラスタ環境）ダイアログ
- フォルダの選択ダイアログ

### 4.1.1 新規作成ダイアログ

Script マネージャウィンドウで [ ファイル ] - [ 新規作成 ] を選択したときに表示されるダイアログです。

作成するスクリプトファイル名を設定します。



### (1) ダイアログの項目

作成するスクリプトファイル名

作成するスクリプトファイル名を設定します。

エンジンバージョンを挿入する

スクリプトファイルの先頭に、スクリプトファイルを実行するスクリプトエンジンのバージョンを挿入するときに指定します。なおこの指定は、エディタの関連付けダイアログで、専用のエディタ以外が関連付けられている場合にだけ有効です。

### (2) ダイアログでの操作

- 作成するスクリプトファイル名を入力して [ OK ] ボタンをクリックすると、[ ツール ] - [ エディタの関連付け ] で指定したエディタが起動します。
- [ キャンセル ] ボタンをクリックすると新規作成を中断します。

### (3) 処理詳細

- ファイル名は、Windows ファイルシステムの名前の規則に従ってください。ただし、次の半角文字は使用できません。  
「=」, 「:」, 「,」, 「.」, 「{」, 「}」
- 使用するネットワーク OS によっては、ファイルの名前の付け方が異なります。各ネットワーク OS のファイルシステムの名前の規則に従ってください。
- 使用するエディタが、[ ツール ] - [ エディタの関連付け ] メニューで指定されていないときは、その旨のメッセージを表示して処理を中断します。
- 専用のエディタで作成したスクリプトファイルを、Script マネージャウィンドウのクライアントエリアに反映するには、[ 表示 ] - [ 最新の状態に更新 ] メニューを実行する必要があります。
- [ エンジンバージョンを挿入する ] を指定すると、スクリプトファイルの先頭行に、次の内容が挿入されます。  
「#FileVersion=VRRR ( VV : JP1/Script のバージョン , RR : JP1/Script のリビジョン )」  
VRRR にはインストールされている JP1/Script のバージョンリビジョンが設定されます。  
上記の指定がない場合は、Script マネージャウィンドウの [ オプション ( 互換性 ) ] ダイアログで設定されたバージョンで、スクリプトファイルが実行されます。

##### ! 注意事項

Windows 標準のワードパットはエディタとして使用できません。そのほかの、TXT 形式で保存できるエディタをご利用ください。

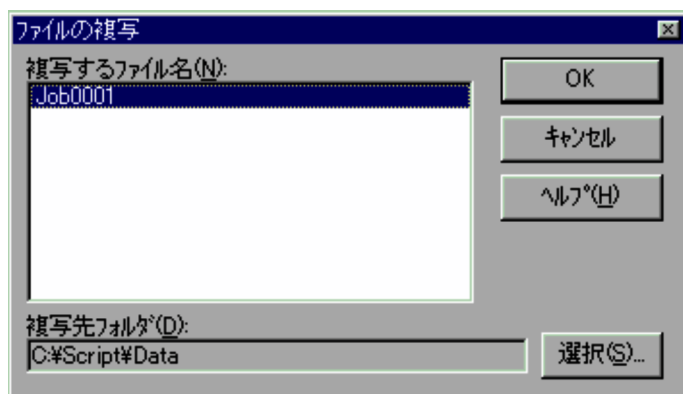
##### ! 注意事項

スクリプトを実行させるエンジンのバージョンとは、JP1/Script のインストールバージョンではありません。

### 4.1.2 ファイルの複写ダイアログ

Script マネージャウィンドウで、アイコンを一つ選択してから、[ ファイル ] - [ 複写 ] メニューを選択すると表示されるダイアログです。

現在表示されているフォルダから別のフォルダへ複写するスクリプトファイルを選択します。



#### (1) ダイアログの項目

##### 複写するファイル名

マネージャウィンドウであらかじめ選択した一つ、または複数のファイル名が表示されます。

##### 複写先のフォルダ

複写先のフォルダ名を指定します。

[ 選択 ] ボタンを選択すると [ フォルダの選択 ] ダイアログが表示されます。

##### 選択

このボタンを選択すると、複写先のフォルダを選択する [ フォルダの選択 ] ダイアログが表示されます。



## (2) ダイアログでの操作

- 複写先のフォルダ名を入力し、複写するファイル名から一つ、または複数のファイルを選択します。[ 選択 ] ボタンを押して、複写先のフォルダを選択することもできます。
- [ OK ] ボタンをクリックするとファイルが複写されます。[ キャンセル ] ボタンをクリックすると、ファイルの複写を中断します。

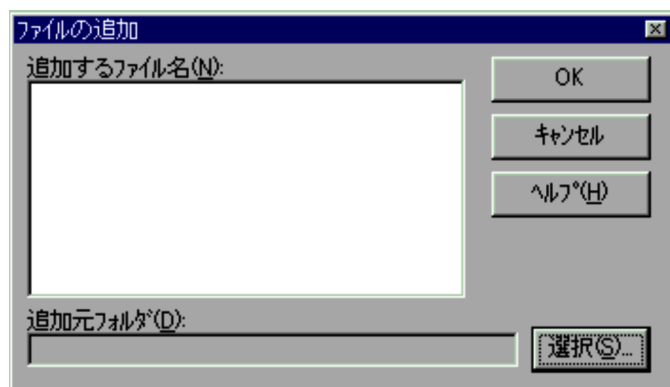
## (3) 処理詳細

- 複写処理中は進捗ダイアログに複写状況が表示されます。
- 複写先に同じ名前のファイルがある場合は、[ ファイルの上書き確認 ] ダイアログが表示されます。

## 4.1.3 ファイルの追加ダイアログ

Script マネージャウィンドウで、アイコンを一つ選択してから、[ ファイル ] - [ 追加 ] メニューを選択すると表示されるダイアログです。

ほかのフォルダのスク립トファイルを現在表示されているフォルダに追加します。



## (1) ダイアログの項目

## 追加するファイル名

追加元フォルダにあるスク립トファイルが表示されます。  
ここで、追加するファイル名を一つ、または複数選択します。

## 追加元フォルダ

追加するファイルがあるフォルダ名を指定します。

## 選択

このボタンを選択すると、追加元のフォルダを選択する [ フォルダの選択 ] ダイアログが表示されます。

## (2) ダイアログでの操作

- 複写先のフォルダ名を入力し、複写するファイル名から一つ、または複数のファイルを選択します。[ 選択 ] ボタンを押して、複写先のフォルダを選択することもできます。
- [ OK ] ボタンをクリックするとファイルが複写されます。[ キャンセル ] ボタンをクリックすると、ファイルの複写を中断します。

## (3) 処理詳細

- 追加処理中は進捗ダイアログに追加状況が表示されます。
- 追加先（現在のフォルダ）に同じ名前のファイルがある場合は、[ ファイルの上書き確認 ] ダイアログが表示されます。

### 4.1.4 ファイルの上書き確認ダイアログ

ファイルの複写やファイルの追加で、ファイルが上書きされるときに表示されるダイアログです。



## (1) ダイアログの項目

### 元のファイル

複写、または追加するファイルの名前が表示されます。上書きはできません。  
ファイル名の下には、ファイルの容量、更新日付・時刻が表示されます。

### 新しいファイル

上書きされるファイルの名前が表示されます。上書きしたくない場合は、新しい名前を指定します。

[ はい ] ボタンをクリックした場合だけ有効となります。ファイル名の下には、ファイルの容量、更新日付・時刻が表示されます。

## (2) ダイアログでの操作

- ファイルを上書きする場合は、[ はい ]、または [ すべて上書き ] ボタンをクリックします。[ すべて上書き ] ボタンをクリックした場合は、以降のファイルに対して上書き確認されなくなります。
- [ いいえ ] ボタンをクリックするとファイルは上書きされないで処理が続行されます。

- [キャンセル] ボタンをクリックすると処理は中断されます。

### 4.1.5 ファイルの削除の確認ダイアログ

Script マネージャウィンドウで、アイコンを一つ選択してから、[ファイル] - [削除] メニューを選択すると表示されるダイアログです。

作成されているスクリプトファイルを削除します。



#### (1) ダイアログの項目

削除するファイル名

Script マネージャウィンドウであらかじめ選択した一つ、または複数のファイル名が表示されます。

#### (2) ダイアログでの操作

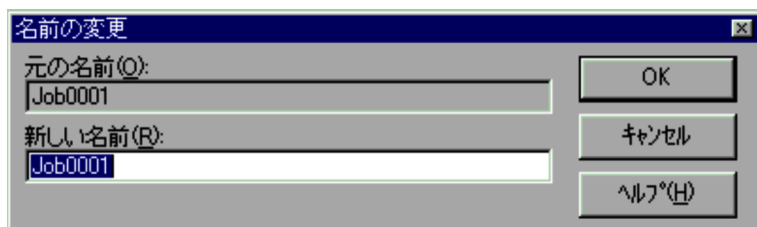
- 削除するファイル名の一覧から、一つ、または複数のファイルを選択します。[はい] ボタンを選択すると、ファイルが削除されて、Script マネージャウィンドウのクライアントエリアからアイコンが消去されます。複数のファイルを選択した場合は、次のファイルに処理が移ります。
- [すべて削除] ボタンを選択すると、削除するファイル名の一覧に表示されているすべてのファイルが削除されます。
- [いいえ] ボタンを選択すると、ファイルは削除されません。複数のファイルを選択した場合は、次のファイルに処理が移ります。
- [キャンセル] ボタンをクリックすると、ファイルの削除を中止します。

### 4.1.6 名前の変更ダイアログ

[ファイル] - [名前の変更] メニューを選択すると表示されるダイアログです。

Script マネージャウィンドウであらかじめ選択したスクリプトファイルの名前を別の名前に変更します。

#### 4. ダイアログの説明



##### (1) ダイアログの項目

###### 元の名前

Script マネージャウィンドウで選択したファイル (アイコン) の名前が表示されます。

###### 新しい名前

変更する名前を表示します。

##### (2) ダイアログでの操作

- 新しい名前を入力して [ OK ] ボタンを選択すると、ファイル名が変更されます。
- [ キャンセル ] ボタンを選択すると、ダイアログを閉じて名前の変更を中止します。

##### (3) 処理詳細

- ファイル名は、Windows ファイルシステムの名前の規則に従ってください。ただし、次の半角文字は使用できません。  
「=」, 「:」, 「,」, 「.」, 「{」, 「}」
- 使用するネットワーク OS によっては、ファイルの名前の付け方が異なります。各ネットワーク OS のファイルシステムの名前の規則に従ってください。

### 4.1.7 実行ダイアログ

Script マネージャウィンドウで、アイコンを一つ選択してから、[ ファイル ] - [ 実行 ] メニューを選択すると表示されるダイアログです。



##### (1) ダイアログの項目

実行するスクリプトファイル名

Script マネージャウィンドウで選択したファイル（アイコン）の名前が表示されます。

ウィンドウを表示する

解析トレース、実行トレース、および実行状況のウィンドウを表示するかどうかを指定します。

## （２）ダイアログでの操作

- [ OK ] ボタンを選択すると、スクリプトが実行されます。
- [ キャンセル ] ボタンを選択すると、スクリプトの実行が中止されます。
- [ 詳細 ] ボタンを選択すると、実行環境を設定できます。設定した内容は、実行環境ファイルに登録されます。

実行環境の設定方法については、「4.1.8 実行環境の設定（起動情報）ダイアログ」、  
「4.1.9 実行環境の設定（終了情報）ダイアログ」、  
「4.1.10 実行環境の設定（トレース情報）ダイアログ」を参照してください。

## （３）処理詳細

- 実行中は、実行状態を表す状態表示画面が表示されます。

### 4.1.8 実行環境の設定（起動情報）ダイアログ

Script マネージャウィンドウでアイコンを一つ、または複数選択してから、[ ファイル ]  
- [ 実行環境の設定 ] - [ すべての項目 ] メニューを選択すると、[ 実行環境の設定 ] ダイアログが表示されます。

また、[ 実行 ] ダイアログから [ 詳細 ] ボタンを選択しても表示されます。

[ 実行環境の設定 ] ダイアログには、「起動情報」「終了情報」「トレース情報」の三つのタブがあります。

「起動情報」タブを選択すると、[ 実行環境の設定（起動情報）] ダイアログが表示されます。



#### 補足

Windows Vista では、JP1/Script サービスが実行中以外の状態で実行環境の設定（起動情報）をしようとすると、次のエラーダイアログを表示します。  
「JP1/Script サービスの状態が実行中以外のため、実行できません。JP1/Script サービスを開始した後、再度実行してください。」

### （１）ダイアログの項目

[ 実行環境の設定（起動情報）] ダイアログでは、選択したスクリプトファイルに対して、次の項目を設定できます。

#### スクリプトファイル名

Script マネージャウィンドウで選択したファイル（アイコン）の名前が表示されます。  
アイコンを複数選択した場合は複数のファイルの名前を昇順にソートして表示します。

#### コマンドライン

実行時に必要なコマンドラインを指定します。  
位置変数や、トレースファイル、およびイベントログの出力の有無などを指定できます。  
[ 設定 ] ボタンを選択すると、[ コマンドラインの設定 ] ダイアログが表示されます。

詳細は、「6.2 コマンドラインに関する規則」を参照してください。

#### 作業フォルダ

実行時のカレントドライブ、またはフォルダを指定します。

[ 選択 ] ボタンを選択すると、[ フォルダの選択 ] ダイアログが表示されます。

指定がない場合は、スクリプトファイルのフォルダが選択されます。

#### 起動情報

自動起動する場合の起動先の空間を指定します。曜日・日・時刻の設定が無い場合、次に示す条件で起動されます。

ログオン：OS にログオンした時

サービス：電源投入時

起動先がログオンで曜日・日・時刻を設定していても OS にログオンしていなければ、起動されません。

#### 複数起動を許可する

一つのスクリプトを複数同時に実行することを許可する場合にチェックします。

同時実行数の上限はありません。しかし、同時実行数が多くなれば、その分システムリソースを消費します。そのため、同時実行するスクリプト数に注意してください。

この値をチェックした場合、トレース出力先フォルダは [ 実行環境の設定 (トレース情報) ] ダイアログの「出力先フォルダ」ではなく、[ オプション (複数起動) ] ダイアログの「トレースファイルの出力先フォルダ」で指定します。ダイアログの設定方法については、「4.1.25 オプション (複数起動) ダイアログ」を参照してください。

#### 自動起動情報に登録する

ランチャによって自動起動する場合にチェックします。

#### 曜日の設定をする

指定曜日を決めて起動する場合に選択します。

「自動起動情報に登録する」がチェックされていない場合は、グレースアウトされて選択できません。

#### 曜日

毎週何曜日に起動するかを指定します。

「曜日の設定をする」がチェックされていない場合は、グレースアウトされて指定できません。

#### 日の設定をする

指定日を決めて起動する場合に選択します。

「自動起動情報に登録する」がチェックされていない場合は、グレースアウトされて選択できません。

#### 日

毎月何日に起動するかを指定します。

## 4. ダイアログの説明

1 ~ 31 の範囲で指定してください。なお、31 日を指定した場合、31 日のない月は実行されません。

「日の設定をする」がチェックされていない場合は、グレースアウトされて指定できません。

### 時刻の設定をする

指定時刻を決めて起動する場合に選択します。

「自動起動情報に登録する」がチェックされていない場合は、グレースアウトされて選択できません。

### 時刻

何時何分に起動するかを指定します。

00:00 ~ 23:59 の範囲で指定してください。なお、指定時刻がランチャの起動時刻より前の場合は、翌日に起動されます。

「時刻の設定をする」がチェックされていない場合は、グレースアウトされて指定できません。

注 JP1/Script サービスの起動時、またはランチャの起動時にだけ有効になる情報です。

## (2) ダイアログでの操作

- 必要に応じて、実行環境の情報を設定します。
- [ OK ] を選択すると、ダイアログで設定した情報が各実行環境ファイルとして作成されます。
- [ キャンセル ] を選択すると、実行環境ファイルを作成しないでダイアログを閉じます。

## (3) 処理詳細

- アイコンを複数選択した場合の各項目のデフォルトでは、「スクリプトファイル名」の先頭に表示されたファイルの実行環境の情報が設定されています。また、「スクリプトファイル名」に表示されたファイルの名前を一つ選択するとそのファイルの情報が各項目に設定されます。
- 日、および時刻が設定されていない場合、JP1/Script サービスの起動時、またはランチャの起動時に起動されます。
- Windows XP では、起動のタイプがログオンのスクリプトファイルに対して自動起動情報の登録をした場合、登録したユーザでしか自動起動されません。

## 4.1.9 実行環境の設定（終了情報）ダイアログ

Script マネージャウィンドウでアイコンを一つ、または複数選択してから、[ ファイル ] - [ 実行環境の設定 ] - [ すべての項目 ] メニューを選択すると、[ 実行環境の設定 ] ダイアログが表示されます。

また、[ 実行 ] ダイアログから [ 詳細 ] ボタンを選択しても表示されます。



[ 実行環境の設定 ] ダイアログには、「起動情報」「終了情報」「トレース情報」の三つのタブがあります。

「終了情報」タブを選択すると、[ 実行環境の設定 ( 終了情報 ) ] ダイアログが表示されます。



#### 補足

Windows Vista では、JP1/Script サービスが実行中以外の状態で実行環境の設定 ( 終了情報 ) をしようとすると、次のエラーダイアログを表示します。

「JP1/Script サービスの状態が実行中以外のため、実行できません。JP1/Script サービスを開始した後、再度実行してください。」

#### ( 1 ) ダイアログの項目

[ 実行環境の設定 ( 終了情報 ) ] ダイアログでは、選択したスクリプトファイルに対して、次の項目を設定できます。

##### スクリプトファイル名

Script マネージャウィンドウで選択したファイル ( アイコン ) の名前が表示されます。

アイコンを複数選択した場合は複数のファイルの名前を昇順にソートして表示します。

## 4. ダイアログの説明

### 打ち切り時間

実行するスクリプトファイルを何分後に強制終了させるかを指定します。

0 ~ 1,440 の範囲で指定してください。ただし、0 が指定された場合は打ち切り時間なしとみなします。

### (2) ダイアログでの操作

- 必要に応じて、実行環境の情報を設定します。
- [OK] を選択すると、ダイアログで設定した情報が各実行環境ファイルとして作成されます。
- [キャンセル] を選択すると、実行環境ファイルを作成しないでダイアログを閉じます。

### (3) 処理詳細

- アイコンを複数選択した場合の各項目のデフォルトでは、「スクリプトファイル名」の先頭に表示されたファイルの実行環境の情報が設定されています。また、「スクリプトファイル名」に表示されたファイルの名前を一つ選択するとそのファイルの情報が各項目に設定されます。
- 「打ち切り時間」を設定しておけば、特定の時間内に終了するはずの処理が何らかの異常によって終了しなかった場合に、自動的に強制終了させることができます。

## 4.1.10 実行環境の設定 (トレース情報) ダイアログ

Script マネージャウィンドウでアイコンを一つ、または複数選択してから、[ファイル] - [実行環境の設定] - [すべての項目] メニューを選択すると、[実行環境の設定] ダイアログが表示されます。

また、[実行] ダイアログから [詳細] ボタンを選択しても表示されます。

[実行環境の設定] ダイアログには、「起動情報」「終了情報」「トレース情報」の三つのタブがあります。

「トレース情報」タブを選択すると、[実行環境の設定 (トレース情報)] ダイアログが表示されます。



#### 補足

Windows Vista では、JP1/Script サービスが実行中以外の状態で実行環境の設定（トレース情報）をしようとすると、次のエラーダイアログを表示します。

「JP1/Script サービスの状態が実行中以外のため、実行できません。JP1/Script サービスを開始した後、再度実行してください。」

### （１）ダイアログの項目

[ 実行環境の設定（トレース情報）] ダイアログでは、選択したスクリプトファイルに対して、次の項目を設定できます。

#### スクリプトファイル名

Script マネージャウィンドウで選択したファイル（アイコン）の名前が表示されます。

アイコンを複数選択した場合は複数のファイルの名前を昇順にソートして表示します。

#### 出力先フォルダ

解析トレースファイル、および実行トレースファイルを出力する先のフォルダを指定します。

[ 選択 ] ボタンを選択すると、[ フォルダの選択 ] ダイアログが表示されます。

なお、[ 実行環境の設定（起動情報）] ダイアログで「複数起動を許可する」が

#### 4. ダイアログの説明

チェックされた場合にはグレースアウトされ指定できません。

##### 最大行数

解析トレースファイル，および実行トレースファイルの最大出力行数を指定します。  
100 ~ 9,999 の範囲で指定してください。

##### 最大列数

解析トレースファイル，および実行トレースファイルの最大出力列数を指定します。  
128 ~ 1,024 の範囲で指定してください。

##### ユーザトレース情報

###### 最大行数

ユーザトレースファイルの最大出力行数を指定します。  
100 ~ 9,999 の範囲で指定してください。

###### 最大列数

ユーザトレースファイルの最大出力列数を指定します。  
128 ~ 1,024 の範囲で指定してください。

#### (2) ダイアログでの操作

- 必要に応じて，実行環境の情報を設定します。
- [ OK ] を選択すると，ダイアログで設定した情報が各実行環境ファイルとして作成されます。
- [ キャンセル ] を選択すると，実行環境ファイルを作成しないでダイアログを閉じます。

#### (3) 処理詳細

- アイコンを複数選択した場合の各項目のデフォルトでは，「スクリプトファイル名」の先頭に表示されたファイルの実行環境の情報が設定されています。また，「スクリプトファイル名」に表示されたファイルの名前を一つ選択するとそのファイルの情報が各項目に設定されます。

### 4.1.11 実行環境（コマンドライン）の設定ダイアログ

Script マネージャウィンドウでアイコンを一つ，または複数選択してから，[ ファイル ] - [ 実行環境の設定 ] - [ コマンドライン ] メニューを選択すると，[ 実行環境（コマンドライン）の設定 ] ダイアログが表示されます。



### (1) ダイアログの項目

[ 実行環境 (コマンドライン) の設定 ] ダイアログでは、選択したスクリプトファイルに対して、次の項目を設定できます。

#### スクリプトファイル名

Script マネージャウィンドウで選択したファイル (アイコン) の名前が表示されます。

アイコンを複数選択した場合は複数のファイルの名前を昇順にソートして表示します。

#### コマンドライン

実行時に必要なコマンドラインを指定します。

位置変数や、トレースファイル、およびイベントログの出力の有無などを指定できます。

[ 設定 ] ボタンを選択すると、[ コマンドラインの設定 ] ダイアログが表示されます。詳細は、「6.2 コマンドラインに関する規則」を参照してください。

### (2) ダイアログでの操作

- 必要に応じて、実行環境の情報を設定します。
- [ OK ] を選択すると、ダイアログで設定した情報で各実行環境ファイルが変更されます。
- [ キャンセル ] を選択すると、各実行環境ファイルを変更しないでダイアログを閉じます。

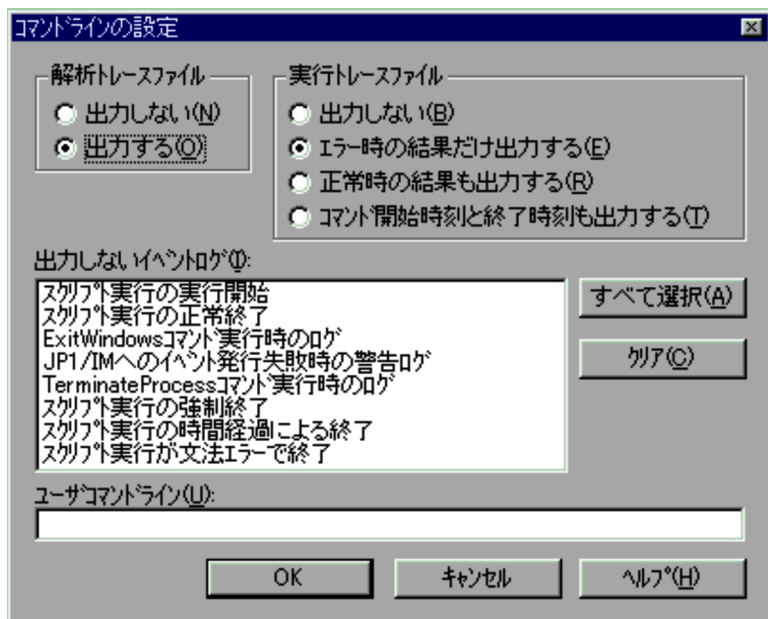
### (3) 処理詳細

- アイコンを複数選択した場合の各項目のデフォルトでは、「スクリプトファイル名」の先頭に表示されたファイルの実行環境の情報が設定されています。また、「スクリプトファイル名」に表示されたファイルの名前を一つ選択するとそのファイルの情報が各項目に設定されます。

### 4.1.12 コマンドラインの設定ダイアログ

[ 実行環境の設定 ( 起動情報 ) ], または [ 実行環境 ( コマンドライン ) の設定 ] ダイアログで, [ 設定 ] ボタンをクリックすることによって表示されるダイアログです。

実行時に必要なコマンドラインを設定します。



#### (1) ダイアログの項目

##### 解析トレースファイル

解析トレースファイルの出力の有無を指定します。

デフォルトでは「出力する」が設定されています。

ここで指定した内容は, [ 実行環境の設定 ( 起動情報 ) ] ダイアログのコマンドラインに "/SPALV(n)" , または "/spalv(n)" の形式で設定されます。

##### 実行トレースファイル

実行トレースファイルの出力の有無, および出力レベルを指定します。

デフォルトでは「エラー時の結果だけ出力する」が設定されています。

ここで指定した内容は, [ 実行環境の設定 ( 起動情報 ) ] ダイアログのコマンドラインに "/SPXLV(n)" , または "/spxlv(n)" の形式で設定されます。

##### 出力しないイベントログ

実行時に出力されるイベントログの出力を抑止する場合に選択します。

[ すべて選択 ] ボタンを選択すると, すべてのイベントログが選択されます。[ クリア ] ボタンを選択すると, 選択したイベントログを選択解除します。

デフォルトではすべての項目が選択されていません。

ここで指定した内容は, [ 実行環境の設定 ( 起動情報 ) ] ダイアログのコマンドライ

ンに "/NOEVLOG" , または "/noevlog" , または "/NOEVLOG(n , n , ...)" , または "/noevlog(n , n , ...)" の形式で設定されます。

#### ユーザコマンドライン

ユーザ指定のコマンドラインを指定します。

### (2) ダイアログでの操作

- 必要に応じて、コマンドラインの情報を設定します。
- [ OK ] を選択すると、ダイアログで設定した情報が [ 実行環境の設定 ( 起動情報 ) ] ダイアログのコマンドラインに設定されます。
- [ キャンセル ] を選択すると、内容を変更しないでダイアログを閉じます。

#### 処理詳細

- 「出力しないイベントログ」のリストには、実行時に出力されるイベントログのうちログの種類がエラー以外のイベントログが表示されています。

## 4.1.13 実行環境 ( 作業フォルダ ) の設定ダイアログ

Script マネージャウィンドウでアイコンを一つ、または複数選択してから、[ ファイル ] - [ 実行環境の設定 ] - [ 作業フォルダ ] メニューを選択すると、[ 実行環境 ( 作業フォルダ ) の設定 ] ダイアログが表示されます。



### (1) ダイアログの項目

[ 実行環境 ( 作業フォルダ ) の設定 ] ダイアログでは、選択したスクリプトファイルに対して、次の項目を設定できます。

#### スクリプトファイル名

Script マネージャウィンドウで選択したファイル ( アイコン ) の名前が表示されます。

アイコンを複数選択した場合は複数のファイルの名前を昇順にソートして表示します。

#### 4. ダイアログの説明

##### 作業フォルダ

実行時のカレントドライブ、またはフォルダを指定します。指定がない場合は、スクリプトファイルのフォルダが選択されます。

##### (2) ダイアログでの操作

- 必要に応じて、実行環境の情報を設定します。
- [OK] を選択すると、ダイアログで設定した情報で各実行環境ファイルが変更されます。
- [キャンセル] を選択すると、各実行環境ファイルを変更しないでダイアログを閉じます。

##### (3) 処理詳細

- アイコンを複数選択した場合の各項目のデフォルトでは、「スクリプトファイル名」の先頭に表示されたファイルの実行環境の情報が設定されています。また、「スクリプトファイル名」に表示されたファイルの名前を一つ選択するとそのファイルの情報が各項目に設定されます。

#### 4.1.14 実行環境（起動日時）の設定ダイアログ

Script マネージャウィンドウでアイコンを一つ、または複数選択してから、[ファイル] - [実行環境の設定] - [起動日時] メニューを選択すると、[実行環境（起動日時）の設定] ダイアログが表示されます。



##### 補足

Windows Vista では、JP1/Script サービスが実行中以外の状態で実行環境（起動日時）を設定しようとすると、次のエラーダイアログを表示します。

「JP1/Script サービスの状態が実行中以外のため、実行できません。JP1/Script



サービスを開始した後、再度実行してください。」

## (1) ダイアログの項目

[ 実行環境（起動日時）の設定 ] ダイアログでは、選択したスクリプトファイルに対して、次の項目を設定できます。

### スクリプトファイル名

Script マネージャウィンドウで選択したファイル（アイコン）の名前が表示されます。

アイコンを複数選択した場合は複数のファイルの名前を昇順にソートして表示します。

### 起動のタイプ

#### ログオン・サービス

ログオン時に起動するか、サービス（電源投入時）で起動するかを選択します。

### 自動起動情報に登録する

ランチャによって自動起動する場合にチェックします。

### 曜日の設定をする

指定曜日を決めて起動する場合に選択します。

「自動起動情報に登録する」がチェックされていない場合は、グレースアウトされて選択できません。

### 曜日

毎週何曜日に起動するかを指定します。

「曜日の設定をする」がチェックされていない場合は、グレースアウトされて指定できません。

### 日の設定をする

指定日を決めて起動する場合に選択します。

「自動起動情報に登録する」がチェックされていない場合は、グレースアウトされて選択できません。

### 日

毎月何日に起動するかを指定します。

1 ~ 31 の範囲で指定してください。なお、31 日を指定した場合、31 日のない月は実行されません。

「日の設定をする」がチェックされていない場合は、グレースアウトされて指定できません。

### 時刻の設定をする

指定時刻を決めて起動する場合に選択します。

「自動起動情報に登録する」がチェックされていない場合は、グレースアウトされて選択できません。

#### 4. ダイアログの説明

##### 時刻

何時何分に起動するかを指定します。

00:00 ~ 23:59 の範囲で指定してください。なお、指定時刻がランチャの起動時刻より前の場合は、翌日に起動されます。

「時刻の設定をする」がチェックされていない場合は、グレーアウトされて指定できません。

注 JP1/Script サービスの起動時、またはランチャの起動時にだけ有効になる情報です。

##### (2) ダイアログでの操作

- 必要に応じて、実行環境の情報を設定します。
- [ OK ] を選択すると、ダイアログで設定した情報で各実行環境ファイルが変更されます。
- [ キャンセル ] を選択すると、各実行環境ファイルを変更しないでダイアログを閉じます。

##### (3) 処理詳細

- アイコンを複数選択した場合の各項目のデフォルトでは、「スクリプトファイル名」の先頭に表示されたファイルの実行環境の情報が設定されています。また、「スクリプトファイル名」に表示されたファイルの名前を一つ選択するとそのファイルの情報が各項目に設定されます。
- 日、および時刻が設定されていない場合、JP1/Script サービスの起動時、またはランチャの起動時に起動されます。

### 4.1.15 実行環境（打ち切り時間）の設定ダイアログ

Script マネージャウィンドウでアイコンを一つ、または複数選択してから、[ ファイル ] - [ 実行環境の設定 ] - [ 打ち切り時間 ] メニューを選択すると、[ 実行環境（打ち切り時間）の設定 ] ダイアログが表示されます。



### (1) ダイアログの項目

[ 実行環境 ( 打ち切り時間 ) の設定 ] ダイアログでは、選択したスクリプトファイルに対して、次の項目を設定できます。

#### スクリプトファイル名

Script マネージャウィンドウで選択したファイル ( アイコン ) の名前が表示されます。

アイコンを複数選択した場合は複数のファイルの名前を昇順にソートして表示します。

#### 打ち切り時間

実行するスクリプトファイルを何分後に強制終了させるかを指定します。

0 ~ 1,440 の範囲で指定してください。ただし、0 が指定された場合は打ち切り時間なしとみなします。

### (2) ダイアログでの操作

- 必要に応じて、実行環境の情報を設定します。
- [ OK ] を選択すると、ダイアログで設定した情報で各実行環境ファイルが変更されます。
- [ キャンセル ] を選択すると、各実行環境ファイルを変更しないでダイアログを閉じます。

### (3) 処理詳細

- アイコンを複数選択した場合の各項目のデフォルトでは、「スクリプトファイル名」の先頭に表示されたファイルの実行環境の情報が設定されています。また、「スクリプトファイル名」に表示されたファイルの名前を一つ選択するとそのファイルの情報が各項目に設定されます。
- 「打ち切り時間」を設定しておけば、特定の時間内に終了するはずの処理が何らかの異常によって終了しなかった場合に、自動的に強制終了させることができます。

## 4.1.16 実行環境 ( トレース出力先フォルダ ) の設定ダイアログ

Script マネージャウィンドウでアイコンを一つ、または複数選択してから、[ ファイル ] - [ 実行環境の設定 ] - [ トレース出力先フォルダ ] メニューを選択すると、[ 実行環境 ( トレース出力先フォルダ ) の設定 ] ダイアログが表示されます。

#### 4. ダイアログの説明



##### (1) ダイアログの項目

[ 実行環境 (トレース出力先フォルダ) の設定 ] ダイアログでは、選択したスクリプトファイルに対して、次の項目を設定できます。

##### スクリプトファイル名

Script マネージャウィンドウで選択したファイル (アイコン) の名前が表示されます。

アイコンを複数選択した場合は複数のファイルの名前を昇順にソートして表示します。

##### 複数起動を許可する

複数のプロセスから同時に実行することを許可する場合にチェックします。

この値をチェックした場合、トレース出力先フォルダは「出力先フォルダ」ではなく、[ オプション (複数起動) ] ダイアログの「トレースファイルの出力先フォルダ」で指定します。

同時実行数の上限はありません。しかし、同時実行数が多くなれば、その分システムリソースを消費します。そのため、同時実行するスクリプト数に注意してください。

##### 出力先フォルダ

解析トレースファイル、および実行トレースファイルを出力する先のフォルダを指定します。

[ 選択 ] ボタンを選択すると、[ フォルダの選択 ] ダイアログが表示されます。

「複数起動を許可する」がチェックされた場合にはグレースアウトされ指定できません。

##### 最大行数

解析トレースファイル、および実行トレースファイルの最大出力行数を指定します。

100 ~ 9,999 の範囲で指定してください。

## 最大列数

解析トレースファイル、および実行トレースファイルの最大出力列数を指定します。  
128 ~ 1,024 の範囲で指定してください。

## (2) ダイアログでの操作

- 必要に応じて、実行環境の情報を設定します。
- [OK] を選択すると、ダイアログで設定した情報で各実行環境ファイルが変更されます。
- [キャンセル] を選択すると、各実行環境ファイルを変更しないでダイアログを閉じます。

## (3) 処理詳細

- アイコンを複数選択した場合の各項目のデフォルトでは、「スクリプトファイル名」の先頭に表示されたファイルの実行環境の情報が設定されています。また、「スクリプトファイル名」に表示されたファイルの名前を一つ選択するとそのファイルの情報が各項目に設定されます。

## 4.1.17 実行環境（ユーザトレース情報）の設定ダイアログ

Script マネージャウィンドウでアイコンを一つ、または複数選択してから、[ファイル]

- [実行環境の設定] - [ユーザトレース情報] メニューを選択すると、[実行環境（ユーザトレース情報）の設定] ダイアログが表示されます。



## (1) ダイアログの項目

[実行環境（ユーザトレース情報）の設定] ダイアログでは、選択したスクリプトファイルに対して、次の項目を設定できます。

## スクリプトファイル名

Script マネージャウィンドウで選択したファイル（アイコン）の名前が表示されます。

アイコンを複数選択した場合は複数のファイルの名前を昇順にソートして表示しま

## 4. ダイアログの説明

す。

### 最大行数

ユーザトレースファイルの最大出力行数を指定します。

100 ~ 9,999 の範囲で指定してください。

### 最大列数

ユーザトレースファイルの最大出力列数を指定します。

128 ~ 1,024 の範囲で指定してください。

## (2) ダイアログでの操作

- 必要に応じて、実行環境の情報を設定します。
- [OK] を選択すると、ダイアログで設定した情報で各実行環境ファイルが変更されます。
- [キャンセル] を選択すると、各実行環境ファイルを変更しないでダイアログを閉じます。

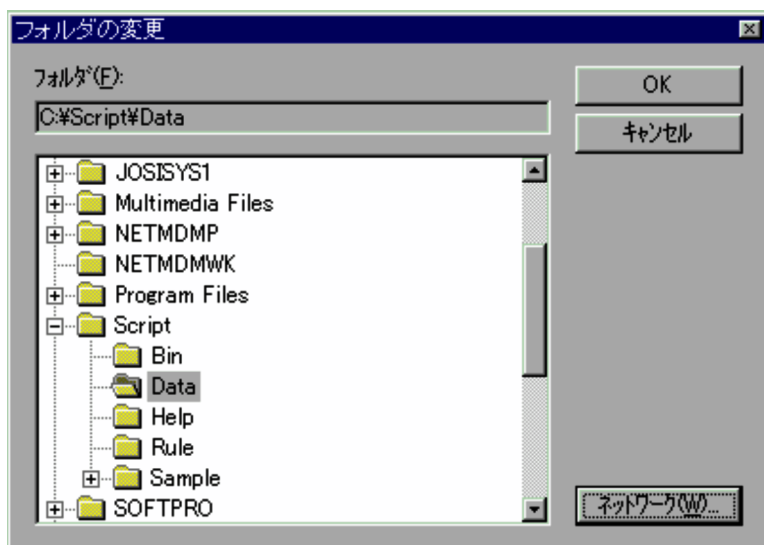
## (3) 処理詳細

- アイコンを複数選択した場合の各項目のデフォルトでは、「スクリプトファイル名」の先頭に表示されたファイルの実行環境の情報が設定されています。また、「スクリプトファイル名」に表示されたファイルの名前を一つ選択するとそのファイルの情報が各項目に設定されます。

## 4.1.18 フォルダの変更ダイアログ

[ファイル] - [フォルダの変更] メニューを選択すると表示されるダイアログです。

クライアントエリアに表示中のフォルダ（現在選択されているフォルダ）を、指定されたフォルダに変更します。クライアントエリアには、指定されたフォルダに格納されているスクリプトファイルが表示されます。



### (1) ダイアログの項目

#### フォルダ

上の表示領域には、選択しているフォルダ名が表示されます。  
 フォルダ名が表示領域より長い場合は、最終部分が表示されます。  
 下の表示領域には、ドライブ名とフォルダ名がツリー構造で表示されます。

#### ネットワーク

[ ネットワークドライブの割り当て ] ダイアログを表示します。

### (2) ダイアログでの操作

- ・ フォルダを選択してから [ OK ] ボタンをクリックすると、フォルダが変更され、変更後のフォルダの情報がクライアントエリアに表示されます。
- ・ [ キャンセル ] ボタンをクリックすると、フォルダを変更しないでダイアログを閉じます。

### (3) 補足

直前に変更したフォルダは、このダイアログを表示しなくても変更できます。その場合は、[ ファイル ] メニューにあるフォルダ名を直接選択してください。

## 4.1.19 ファイルのプロパティダイアログ

Script マネージャウィンドウでアイコンを一つ選択してから、マウスの右ボタンをクリックし、ポップアップメニューから [ プロパティ ] を選択すると表示されるダイアログです。

#### 4. ダイアログの説明



##### (1) ダイアログの項目

###### スクリプトファイル名

選択されたファイル（アイコン）の名前が表示されます。

###### 作成日時

ファイルの作成日と時刻が表示されます。

###### 更新日時

ファイルの更新日と時刻が表示されます。

###### ファイル容量

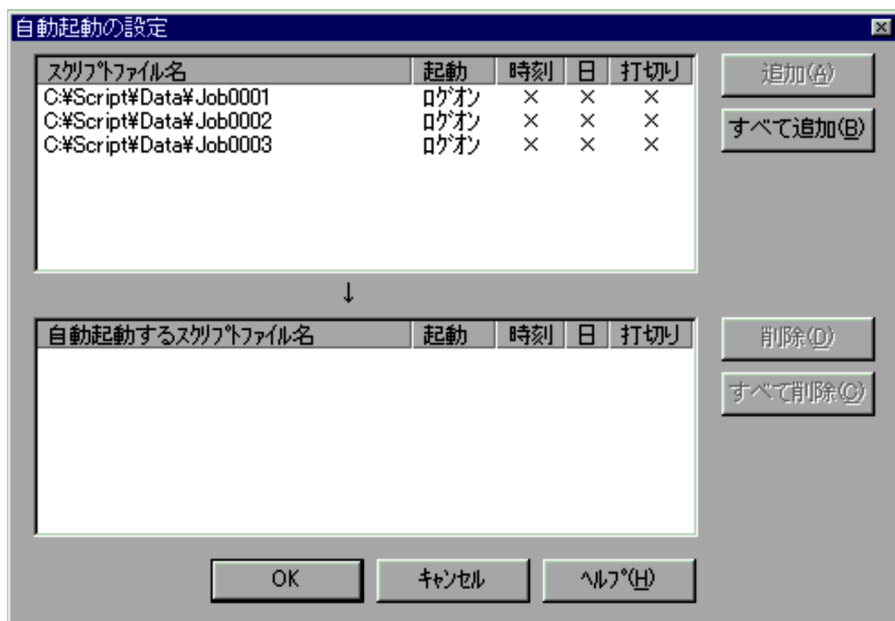
ファイルの容量が表示されます。

#### 4.1.20 自動起動の設定ダイアログ

Script マネージャウィンドウで、[ ツール ] - [ 自動起動の設定 ] メニューを選択すると表示されるダイアログです。

ランチャによって自動起動されるスクリプトファイルの名前を設定します。





#### 補足

Windows Vista では、JP1/Script サービスが実行中以外の状態で自動起動を設定しようとすると、次のエラーダイアログを表示します。

「JP1/Script サービスの状態が実行中以外のため、実行できません。JP1/Script サービスを開始した後、再度実行してください。」

### (1) ダイアログの項目

#### スクリプトファイル名

現在の実行フォルダ下にある実行環境ファイルのうち「自動起動情報へ登録する」が定義されていないスクリプトファイルのファイル名が表示されます。

#### 自動起動するスクリプトファイル名

実行環境ファイルで「自動起動情報へ登録する」が定義されているスクリプトファイルのファイル名が表示されます。

#### 起動

起動のタイプが表示されます。

#### 時刻

時刻による起動が設定されているかどうかを表示します。

#### 日

日による起動が設定されているかどうかを表示します。

#### 打切り

打ち切り時間の設定がされているかどうかを表示します。

### (2) ダイアログでの操作

- 「スクリプトファイル名」のリストボックスから自動起動させたいスクリプトファイルを選択して [ 追加 ] ボタンを選択すると、選択したファイルが「自動起動するスクリプトファイル名」のリストボックスに追加され、「スクリプトファイル名」のリストボックスから削除されます。
- [ すべて追加 ] ボタンを選択すると、すべてのファイルが「自動起動するスクリプトファイル名」のリストボックスに追加され、「スクリプトファイル名」のリストボックスから削除されます。
- 「自動起動するスクリプトファイル名」のリストボックスからファイルを選択して [ 削除 ] ボタンを選択すると、選択したファイルが「自動起動するスクリプトファイル名」のリストボックスから削除され、「スクリプトファイル名」のリストボックスに追加されます。
- [ すべて削除 ] ボタンを選択すると、すべてのファイルが「自動起動するスクリプトファイル名」のリストボックスから削除され、「スクリプトファイル名」のリストボックスに追加されます。
- [ OK ] ボタンを選択すると、「自動起動するスクリプトファイル名」のリストボックスの内容を自動起動情報ファイルとして出力します。
- [ キャンセル ] ボタンを選択すると、自動起動情報ファイルを出力しないでダイアログを閉じます。

### (3) 処理詳細

- 最初にダイアログを開いたときは、その時点の自動起動情報ファイルの内容がリストボックスに表示されます。
- 時刻、日、打ち切りの情報は、 が設定されていることを、×が設定されていないことを意味します。
- 現在の実行フォルダ下にあるスクリプトファイルのうち実行環境ファイルがない場合は、このダイアログを開いた時点で作成されます。[ OK ] ボタン、[ キャンセル ] ボタンのどちらを選択しても実行環境ファイルは作成されます。
- Windows XP では、起動のタイプがログオンのスクリプトファイルを追加した場合、追加したユーザでしか自動起動されません。

## 4.1.21 エディタの関連付けダイアログ

Script マネージャウィンドウで、[ ツール ] - [ エディタの関連付け ] メニューを選択すると表示されるダイアログです。



### (1) ダイアログの項目

別のエディタを使用する

Script エディタ以外のエディタを使用する場合はチェックします。

アプリケーション名

関連付けするエディタの名前を表示します。

エディタは、[ 選択 ] ボタンを選択して表示されるアプリケーションの一覧から選択できます。

作業フォルダ

関連付けするエディタの作業フォルダを表示します。

作業フォルダは、[ 選択 ] ボタンを選択して表示される一覧から選択できます。

### (2) ダイアログでの操作

- 関連付けをするアプリケーション名を選択して [ OK ] ボタンを選択すると、エディタを関連付けします。
- [ キャンセル ] ボタンを選択すると、エディタを関連付けしないでダイアログを閉じます。

### (3) 処理詳細

- 作業フォルダの指定を省略した場合、次のフォルダが仮定されます。

[ 別のエディタを使用する ] をチェックしたとき：

そのエディタが格納されているフォルダが仮定されます。

[ 別のエディタを使用する ] をチェックしないとき：

編集対象のスクリプトファイルが格納されているフォルダが仮定されます。

## 4.1.22 オプション (サーバ情報) ダイアログ

Script マネージャウィンドウで、[ ツール ] - [ オプション ] メニューを選択すると、[ オプション ] ダイアログが表示されます。

[ オプション ] ダイアログには、「サーバ情報」「互換性」「複数起動」「JP1/IM」「トレース」「クラスタ環境」の六つのタブがあります。

#### 4. ダイアログの説明

「サーバ情報」タブを選択すると、[ オプション (サーバ情報) ] ダイアログが表示されます。

##### 補足

Windows Vista では、JP1/Script サービスが実行中以外の状態でオプション (サーバ情報) を設定しようとすると、次のエラーダイアログを表示します。

「JP1/Script サービスの状態が実行中以外のため、実行できません。JP1/Script サービスを開始した後、再度実行してください。」



#### (1) ダイアログの項目

##### コマンド名

クライアントによる実行の認否を設定するコマンド名が表示されます。

##### 認否

クライアントによる各コマンドの実行を許可するかどうかが表示されます。  
が許可を，×が禁止を意味します。

##### 更新

このボタンを選択すると，[ 更新情報 ] ダイアログが表示されます。

##### サーバトレースを出力する

サーバ上で実行したコマンドのトレース情報を出力するかどうかを設定します。  
デフォルトではチェックされています。

出力先は，インストールフォルダの "DATA" フォルダ下で "SPTSVTRC.SPY" 固定です。

ただし，[ ツール ] - [ オプション (クラスタ環境) ] ダイアログで管理ファイルの出力先が変更されている場合，指定されたフォルダ下に出力します。

##### プロセスビューを受け付ける

クライアントによるプロセスビューの操作を受け付けるかどうかを設定します。  
デフォルトではチェックされています。

## (2) ダイアログでの操作

- ・[OK] ボタンを選択すると、指定した内容が設定されて[オプション] ダイアログを閉じます。
- ・[キャンセル] ボタンを選択すると、内容を変更しないでダイアログを閉じます。

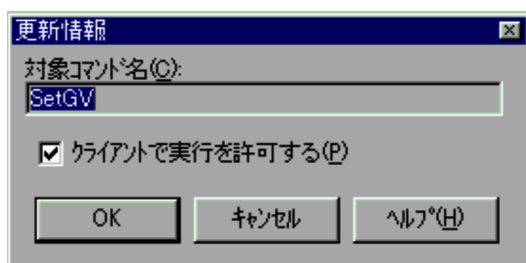
## (3) 処理詳細

- ・このダイアログで設定した「プロセスビューアを受け付ける」以外のサーバの環境は、サーバ環境ファイル(SPTSV.SPS)に保存されます。

## 4.1.23 更新情報ダイアログ

[オプション(サーバ情報)] ダイアログで,[更新] ボタンをクリックすることによって表示されるダイアログです。

クライアントによる各コマンドの実行を許可するかどうかを設定します。



## (1) ダイアログの項目

対象コマンド名

更新の対象となるコマンド名が表示されます。

クライアントで実行を許可する

クライアントによる各コマンドの実行を許可するかどうかを設定します。  
デフォルトでは、チェックされています。

## (2) ダイアログでの操作

- ・[OK] ボタンを選択すると、指定した内容が設定されて[更新情報] ダイアログを閉じ,[オプション(サーバ情報)] ダイアログに戻ります。
- ・[キャンセル] ボタンを選択すると、内容を変更しないでダイアログを閉じ,[オプション(サーバ情報)] ダイアログに戻ります。

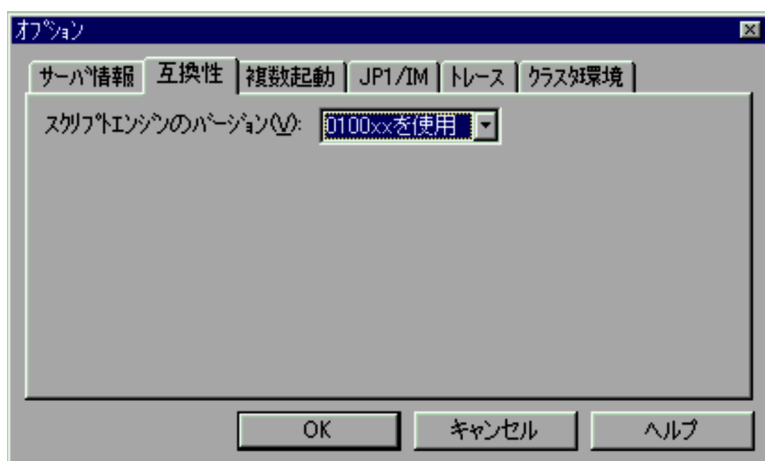
## 4.1.24 オプション(互換性)ダイアログ

Script マネージャウィンドウで,[ツール] - [オプション] メニューを選択すると,[オプション] ダイアログが表示されます。

[オプション] ダイアログには、「サーバ情報」「互換性」「複数起動」「JP1/IM」「トレー

ス」「クラスタ環境」の六つのタブがあります。

「互換性」タブを選択すると、[ オプション ( 互換性 ) ] ダイアログが表示されます。



#### 補足

Windows Vista では、JP1/Script サービスが実行中以外の状態でオプション ( 互換性 ) を設定しようとすると、次のエラーダイアログを表示します。

「JP1/Script サービスの状態が実行中以外のため、実行できません。JP1/Script サービスを開始した後、再度実行してください。」

### ( 1 ) ダイアログの項目

#### スクリプトエンジンのバージョン

スクリプトファイルを実行するときに使用するスクリプトエンジンのバージョンを指定します。

デフォルトでは、「0100xx を使用」が設定されています。

#### ! 注意事項

スクリプトエンジンのバージョンとは、JP1/Script のインストールバージョンではありません。

### ( 2 ) ダイアログでの操作

- ・ [ OK ] ボタンを選択すると、指定した内容が設定されて [ オプション ] ダイアログを閉じます。
- ・ [ キャンセル ] ボタンを選択すると、内容を変更しないでダイアログを閉じます。

### ( 3 ) 処理詳細

- ・ 実行するスクリプトファイルの先頭にスクリプトエンジンのバージョンが記述されていない場合、ここで指定したバージョンのスクリプトエンジンが使用されます。

### 4.1.25 オプション（複数起動）ダイアログ

Script マネージャウィンドウで、[ ツール ] - [ オプション ] メニューを選択すると、[ オプション ] ダイアログが表示されます。

[ オプション ] ダイアログには、「サーバ情報」「互換性」「複数起動」「JP1/IM」「トレース」「クラスタ環境」の六つのタブがあります。

「複数起動」タブを選択すると、[ オプション（複数起動）] ダイアログが表示されます。



#### 補足

Windows Vista では、JP1/Script サービスが実行中以外の状態でオプション（複数起動）を設定しようとすると、次のエラーダイアログを表示します。

「JP1/Script サービスの状態が実行中以外のため、実行できません。JP1/Script サービスを開始した後、再度実行してください。」

#### （１）ダイアログの項目

##### トレースファイルの出力先フォルダ

複数起動時に出力するトレースファイルの出力先フォルダを指定します。

指定できるのは、ローカルドライブにあるフォルダだけです。デフォルトでは「(JP1/Script インストールディレクトリ)¥DATA¥」が設定されています。

ここで指定した内容は、[ 実行環境の設定 ] - [ すべての項目 ] ダイアログの [ 起動情報 ] タブページ、または [ 実行環境の設定 ] - [ トレース出力先フォルダ ] ダイアログの「複数起動を許可する」がチェックされている場合にだけ有効になります。

#### （２）ダイアログでの操作

- [ OK ] ボタンを選択すると、指定した内容が設定されて [ オプション ] ダイアログを閉じます。
- [ キャンセル ] ボタンを選択すると、内容を変更しないでダイアログを閉じます。

### (3) 処理詳細

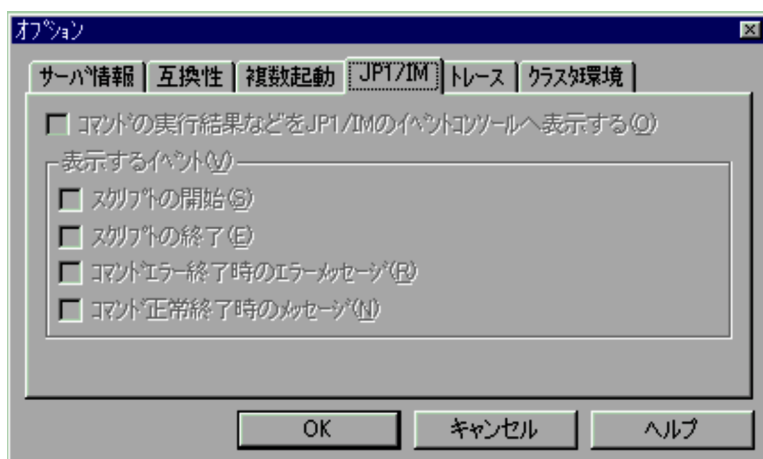
- ・[ ツール ] - [ オプション ( クラスタ環境 ) ] ダイアログで「管理ファイルの出力先を変更する」がチェックされている場合、「トレースファイルの出力先フォルダ」はグレーアウトされて指定された値は無効になります。

## 4.1.26 オプション ( JP1/IM ) ダイアログ

Script マネージャウィンドウで、[ ツール ] - [ オプション ] メニューを選択すると、[ オプション ] ダイアログが表示されます。

[ オプション ] ダイアログには、「サーバ情報」「互換性」「複数起動」「JP1/IM」「トレース」「クラスタ環境」の六つのタブがあります。

「JP1/IM」タブを選択すると、[ オプション ( JP1/IM ) ] ダイアログが表示されます。



### 補足

Windows Vista では、JP1/Script サービスが実行中以外の状態でオプション ( JP1/IM ) を設定しようとすると、次のエラーダイアログを表示します。

「JP1/Script サービスの状態が実行中以外のため、実行できません。JP1/Script サービスを開始した後、再度実行してください。」

### (1) ダイアログの項目

コマンドの実行結果などを JP1/IM のイベントコンソールへ表示する

コマンドの実行結果などを JP1/IM のイベントコンソールに表示する場合にチェックします。

表示するイベント

スクリプトの開始

スクリプトの開始を表すイベントを、イベントコンソールに表示させる場合に



チェックします。

#### スクリプトの終了

スクリプトの終了を表すイベントを、イベントコンソールに表示させる場合にチェックします。

#### コマンドエラー終了時のエラーメッセージ

コマンドのエラー終了を表すイベントを、イベントコンソールに表示させる場合にチェックします。

#### コマンド正常終了時のメッセージ

コマンドの正常終了を表すイベントを、イベントコンソールに表示させる場合にチェックします。

### (2) ダイアログでの操作

- [ OK ] ボタンを選択すると、指定した内容が設定されて [ オプション ] ダイアログを閉じます。
- [ キャンセル ] ボタンを選択すると、内容を変更しないでダイアログを閉じます。

### (3) 処理詳細

- 「コマンドの実行結果などを JP1/IM のイベントコンソールへ表示する」の項目は、ご使用の PC に JP1/IM、または JP1/Base がインストールされている場合にだけ設定できます。インストールされていない場合には、グレーアウトされて設定できません。
- ご使用の PC に JP1/IM、または JP1/Base がインストールされていれば、JP1/IM、または JP1/Base のサービスが起動されているかどうかには関係なく、このダイアログで設定した内容が有効になります。
- スクリプトをマネージャ（ウィンドウを表示する）や専用のエディタから起動した場合は、表示するイベントのうち「スクリプトの開始」や「スクリプトの終了」を指定していても、JP1/IM のイベントコンソールへ表示しません。
- この機能で実行結果を表示できるコマンドは次のとおりです。

#### (a) 基本コマンド

TextOpen ,TextClose ,MakeDir ,DeleteDir ,DeleteFile ,Rename ,SetFileAttribute ,  
GetFileAttribute ,SetFileTime ,GetFileTime ,GetFileSize ,GetVersionInfo ,  
SplitFile ,CatFiles ,SetStandardFile ,ResetStandardFile ,SetVolumeLabel ,  
GetVolumeLabel ,Copy ,Exec ,NetExec ,WaitForExec ,EntryStartUp ,  
CancelStartUp ,JOBSubmit ,JOBWait ,JOBHold ,JOBCancel

#### (b) 特殊コマンド

ServiceCreate ,ServiceDelete ,ServiceStart ,ServiceStop ,ServicePause ,  
ServiceContinue ,ServiceChange ,ServiceControl ,CallDll ,MakeGroup ,  
DeleteGroup ,MakeShortcut ,DeleteShortcut ,TerminateProcess ,ExitWindows

### 4.1.27 オプション（トレース）ダイアログ

Script マネージャウィンドウで、[ ツール ] - [ オプション ] メニューを選択すると、[ オプション ] ダイアログが表示されます。

[ オプション ] ダイアログには、「サーバ情報」「互換性」「複数起動」「JP1/IM」「トレース」「クラスタ環境」の六つのタブがあります。

「トレース」タブを選択すると、[ オプション（トレース）] ダイアログが表示されます。



#### 補足

Windows Vista では、JP1/Script サービスが実行中以外の状態でオプション（トレース）を設定しようとすると、次のエラーダイアログを表示します。

「JP1/Script サービスの状態が実行中以外のため、実行できません。JP1/Script サービスを開始した後、再度実行してください。」

#### （１）ダイアログの項目

##### トレース管理ファイルの初期割り当てデータ数

トレース管理ファイルに登録できるトレースファイルの初期割り当て件数を指定します。

16 ~ 32,000 の範囲で指定してください。デフォルトでは「1024」が設定されています。

#### （２）ダイアログでの操作

- ・ [ OK ] ボタンを選択すると、指定した内容が設定されて [ オプション ] ダイアログを閉じます。
- ・ [ キャンセル ] ボタンを選択すると、内容を変更しないでダイアログを閉じます。

#### （３）処理詳細

- ・ 初期割り当てデータ数を変更するとトレース管理ファイル（SPTLOGDB.SPB）の内

容はクリアされます。このため、トレースビューアウィンドウに表示されていたデータは表示されなくなります。また、トレースファイルの表示ウィンドウの表示が不正になることもあります。この場合は、トレースファイルの表示ウィンドウを開き直してください。

### 4.1.28 オプション（クラスタ環境）ダイアログ

Script マネージャウィンドウで、[ ツール ] - [ オプション ] メニューを選択すると、[ オプション ] ダイアログが表示されます。

[ オプション ] ダイアログには、「サーバ情報」「互換性」「複数起動」「JP1/IM」「トレース」「クラスタ環境」の六つのタブがあります。

「クラスタ環境」タブを選択すると、[ オプション（クラスタ環境）] ダイアログが表示されます。



#### 補足

Windows Vista では、JP1/Script サービスが実行中以外の状態でオプション（クラスタ環境）を設定しようとすると、次のエラーダイアログを表示します。

「JP1/Script サービスの状態が実行中以外のため、実行できません。JP1/Script サービスを開始した後、再度実行してください。」

#### （１）ダイアログの項目

##### 管理ファイルの出力先を変更する

管理ファイルの出力先を通常出力先以外に変更する場合にチェックします。通常の出力先はインストールフォルダの "DATA" フォルダ下です。

管理ファイルとは、JP1/Script で扱うファイルのうち自動起動情報ファイル、サーバ環境ファイル、サーバトレースファイル、トレース管理ファイル、グローバル変数ファイルのことを指します。

##### 出力先フォルダ

「管理ファイルの出力先を変更する」がチェックされている場合、管理ファイルの出力先フォルダを指定します。クラスタ環境の共有ディレクトリ上のフォルダを指定します。

指定できるのは、インストールフォルダの "DATA" フォルダ以外のフォルダです。

##### (2) ダイアログでの操作

- [OK] ボタンを選択すると、変更と再起動を確認するダイアログが表示され、[再起動] ボタンを選択すると指定した内容が設定されて [オプション] ダイアログを閉じ、システムを再起動します。
- [キャンセル] ボタンを選択すると、内容を変更しないでダイアログを閉じます。

##### (3) 処理詳細

- クラスタ環境で JP1/Script を使用することはできますが、フェールオーバー時に実行中のスクリプトを継続して実行することはできません。しかし、このダイアログで管理ファイルの出力先を共有ディスク上に変更している場合、実行環境を引き継ぐことができます。詳細は、「2.3 クラスタシステム環境での環境設定」を参照してください。
- このダイアログで管理ファイルの出力先を変更した場合、新しい設定を有効にするためにシステムの再起動が必要です。
- 管理ファイルの出力先を変更する以前にすでに使用していた各管理ファイルがある場合、それらのファイルは変更前フォルダに残り、それまでのファイルの内容は無効になります。  
指定された「出力先フォルダ」に各管理ファイルがすでにある場合はそのファイルを参照し、存在しない場合は新規に作成します。
- 「出力先フォルダ」に出力されたファイルは複数のコンピュータから同時に参照することはできません。
- 「管理ファイルの出力先を変更する」がチェックされている場合、複数起動時に出力するトレースファイルの出力先フォルダは「出力先フォルダ」の "Trace" フォルダ下になります。[ツール] - [オプション (複数起動)] ダイアログの「トレースファイルの出力先フォルダ」で指定された値は無効になります。
- 管理ファイルの出力先を変更するとトレース管理ファイル (SPTLOGDB.SPB) の内容と各トレースファイルの内容が不一致になる場合があります。そのためトレースビューアウィンドウに表示される内容 (行数と更新日付) が正しくない場合があります。

#### 4.1.29 フォルダの選択ダイアログ

各ダイアログでフォルダを指定する必要がある場合に、[選択] ボタンを選択すると表示されるダイアログです。



### (1) ダイアログの項目

#### フォルダ

上の表示領域には、選択しているフォルダ名が表示されます。

フォルダ名が表示領域より長い場合は、最終部分が表示されます。

下の表示領域には、ドライブ名とフォルダ名がツリー構造で表示されます。

#### ネットワーク

[ ネットワークドライブの割り当て ] ダイアログを表示します。

### (2) ダイアログでの操作

- [ OK ] ボタンをクリックすると、指定したフォルダの情報が取り込まれます。
- [ キャンセル ] をクリックするとフォルダの情報を取り込まないでダイアログを閉じます。

## 4.2 Script エディタウィンドウのダイアログ

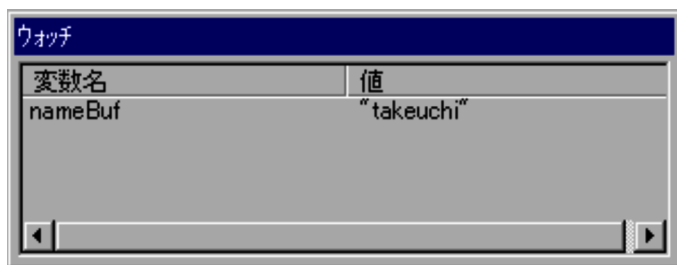
Script エディタウィンドウの操作中表示されるダイアログの一覧を、次に示します。

各ダイアログの項目、および操作については、4.2.1 項以降に説明します。

- 値の更新ダイアログ
- オプション（書式）ダイアログ
- オプション（色）ダイアログ
- オプション（互換性）ダイアログ
- 設定ダイアログ
- 検索ダイアログ
- ファイル名の指定ダイアログ
- ファイルバージョンの指定ダイアログ
- コマンドラインパラメタの設定ダイアログ
- 変数の追加ダイアログ

### 4.2.1 値の更新ダイアログ

[ ウォッチウィンドウ ] で変数名をダブルクリックすると、[ 値の更新 ] ダイアログが表示されます。



#### (1) ダイアログの項目

変数名

更新する対象の変数名が表示されます。

値

更新したい値を入力します。値は 1,024 バイトまで入力できます。

更新する対象の変数名が無効なときは、入力できません。

#### (2) ダイアログでの操作

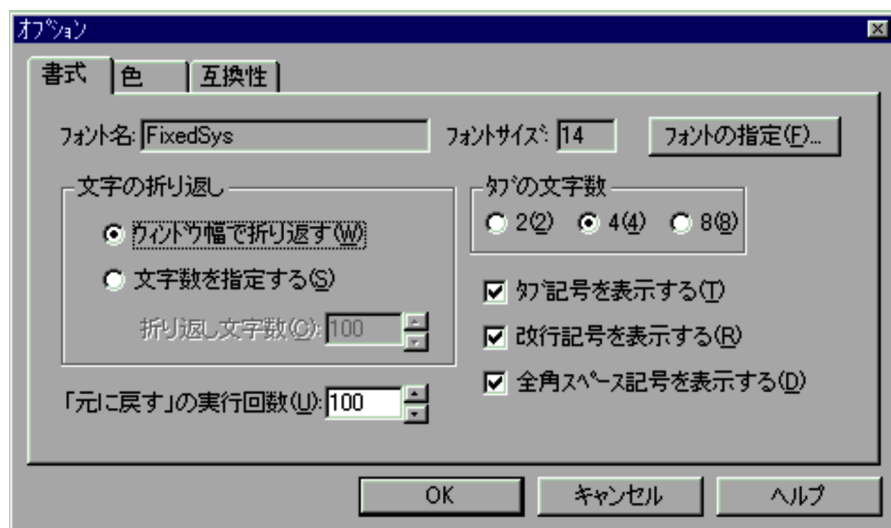
- [ 更新 ] ボタンを選択すると、指定した変数の値が更新されて [ 値の更新 ] ダイアログが閉じます。
- [ キャンセル ] ボタンを選択すると、変数の値を更新しないでダイアログを閉じます。

## 4.2.2 オプション（書式）ダイアログ

Script エディタウィンドウで、[ 編集 ] - [ オプション ] メニューを選択すると、[ オプション ] ダイアログが表示されます。

[ オプション ] ダイアログには、「書式」と「色」と「互換性」の三つのタブがあります。

「書式」タブを選択すると、[ オプション（書式） ] ダイアログが表示されます。



### （１）ダイアログの項目

#### フォント名

文字のフォント名が表示されます。フォントを変更するには、[ フォントの指定 ] ボタンを選択します。

デフォルトでは「FixedSys」が設定されています。

#### フォントサイズ

文字のフォントサイズを指定します。フォントサイズを変更するには、[ フォントの指定 ] ボタンを選択します。

デフォルトでは「14」が設定されています。

#### 文字の折り返し

文字を折り返す方法を選択します。

デフォルトでは「ウィンドウ幅で折り返す」が選択されています。

#### ウィンドウ幅で折り返す

文字をウィンドウ幅で折り返す場合に選択します。

#### 文字数を指定する

文字を固定の文字数で折り返す場合に選択します。

#### 4. ダイアログの説明

##### 折り返し文字数

折り返す文字数を指定します。

「文字数を指定する」が選択されている場合にだけ指定できます。

20 ~ 512 の範囲で指定してください。

##### タブの文字数

タブの文字数を選択します。

デフォルトでは「4」が選択されています。

##### タブ記号を表示する

タブを示す記号の表示 / 非表示を指定します。

デフォルトではチェックされています。

##### 改行記号を表示する

改行を示す記号の表示 / 非表示を指定します。

デフォルトではチェックされています。

##### 全角スペース記号を表示する

全角スペースを示す記号の表示 / 非表示を指定します。

デフォルトではチェックされています。

##### 「元に戻す」の実行回数

[ 編集 ] - [ 元に戻す ] を何回まで実行できるようにするかを指定します。

10 ~ 999 の範囲で指定してください。デフォルトでは「100」が指定されています。

#### (2) ダイアログでの操作

- [ OK ] ボタンを選択すると、指定した書式が設定されて [ オプション ] ダイアログが閉じます。
- [ キャンセル ] ボタンを選択すると、書式を変更しないでダイアログを閉じます。

### 4.2.3 オプション ( 色 ) ダイアログ

Script エディタウィンドウで、[ 編集 ] - [ オプション ] メニューを選択すると、[ オプション ] ダイアログが表示されます。

[ オプション ] ダイアログには、「書式」と「色」と「互換性」の三つのタブがあります。

「色」タブを選択すると、[ オプション ( 色 ) ] ダイアログが表示されます。





### (1) ダイアログの項目

#### テキスト

テキストの色を指定します。

デフォルトでは「システムカラー」が設定されます。

#### 編集時の背景

編集モードの時の背景色を指定します。

デフォルトでは「システムカラー」が設定されます。

#### モニタリング時の背景

モニタリングモードの時の背景色を指定します。

デフォルトでは「灰色」が設定されます。

#### コメント行

コメント行の色を指定します。

デフォルトでは「緑色」が設定されます。

#### 改行記号

改行記号の色を指定します。

デフォルトでは「赤色」が設定されます。

#### コマンド

コマンドの色を指定します。

デフォルトでは「青色」が設定されます。

#### ステートメント

ステートメントの色を指定します。

デフォルトでは「青色」が設定されます。

#### 4. ダイアログの説明

##### キーワード

キーワードの色を指定します。  
デフォルトでは「青色」が設定されます。

##### 文字列

文字列の色を指定します。  
デフォルトでは「暗紫色」が設定されます。

##### 更新行番号

更新行番号の色を指定します。  
デフォルトでは「黄色」が設定されます。

##### 現在位置カーソル

現在位置を示すカーソルの色を指定します。  
デフォルトでは「青色」が設定されます。

##### すべてリセット

すべての項目に対する色の指定をデフォルト値に戻します。

#### (2) ダイアログでの操作

- [すべてリセット] 以外の各項目のボタンを選択すると、[色の設定] ダイアログが表示され、色が選択できます。さらに、[色の設定] ダイアログの [色の作成] ボタンを選択すると、[色の作成] ダイアログが表示され、色を作成して指定できます。
- [OK] ボタンを選択すると、指定した色が設定されて [オプション] ダイアログが閉じます。
- [キャンセル] ボタンを選択すると、色を変更しないでダイアログを閉じます。

### 4.2.4 オプション (互換性) ダイアログ

Script エディタウィンドウで、[編集] - [オプション] メニューを選択すると、[オプション] ダイアログが表示されます。

[オプション] ダイアログには、「書式」と「色」と「互換性」の三つのタブがあります。

「互換性」タブを選択すると、[オプション (互換性)] ダイアログが表示されます。



### (1) ダイアログの項目

ファイルバージョンの有無をチェックする

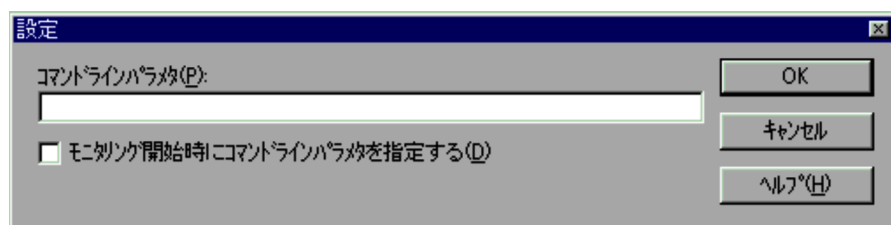
スクリプトファイルの保存時に、ファイルの先頭にファイルバージョンの有無をチェックするかどうかを指定します。

### (2) ダイアログでの操作

- [ OK ] ボタンを選択すると、指定した内容が設定されて [ オプション ] ダイアログが閉じます。
- [ キャンセル ] ボタンを選択すると、内容を変更しないでダイアログを閉じます。

## 4.2.5 設定ダイアログ

Script エディタウィンドウで、[ モニタリング ] - [ 設定 ] メニューを選択すると、[ 設定 ] ダイアログが表示されます。



### (1) ダイアログの項目

コマンドラインパラメータ

モニタリング時に必要となるパラメータを指定します。

#### 4. ダイアログの説明

モニタリング開始時にコマンドラインパラメタを指定する

モニタリング開始時に [ コマンドラインパラメタの設定 ] ダイアログを表示し、コマンドラインパラメタを設定する場合にチェックします。

#### (2) ダイアログでの操作

- [ OK ] ボタンを選択すると、指定した内容が設定されて [ 設定 ] ダイアログを閉じます。
- [ キャンセル ] ボタンを選択すると、内容を変更しないでダイアログを閉じます。

#### (3) 処理詳細

- 指定できるコマンドラインパラメタは半角で 1,024 文字までです。
- [ 設定 ] ダイアログとスクリプトの実行環境ファイルの両方のコマンドラインに指定がある場合、モニタリング実行時には両方の指定が有効になります。  
この場合、[ 設定 ] ダイアログで指定されたコマンドラインの値が先に、実行環境ファイルで指定されたコマンドラインの値が後になる順序でモニタリング実行します。  
ただし、ダイアログに "/SPXLV(3)"、実行環境に "/SPXLV(0)" のように同種で異なる指定が両方にある場合はダイアログの指定が優先されます。

< 例 >

次のように指定された場合、位置変数 %1 は ABC、%2 は 123 となり、/SPXLV(3) が優先され、/NOEVLOG が有効になります。

ダイアログで指定されたコマンドライン：/SPXLV(3) ABC

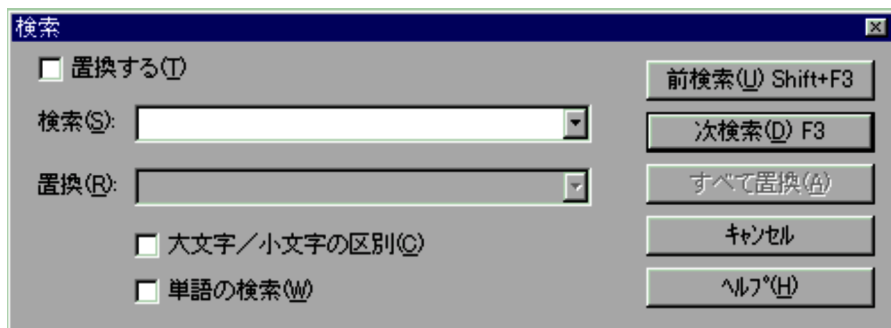
実行環境で指定されたコマンドライン：/SPXLV(0) 123 /NOEVLOG

- [ 設定 ] ダイアログに設定された内容はエディタ終了時にモニタリング情報ファイルに格納されます。

### 4.2.6 検索ダイアログ

Script エディタウィンドウで、[ 検索 ] - [ 検索 ] メニューを選択すると、[ 検索 ] ダイアログが表示されます。

[ 検索 ] - [ 検索 ] メニューを選択しないで [ 検索 ] - [ 前検索 ]、または [ 後検索 ] メニューを選択した場合も、[ 検索 ] ダイアログが表示されます。



## (1) ダイアログの項目

### 置換する

文字列を置換する場合はチェックします。  
デフォルトではチェックされていません。

### 検索

検索する文字列を指定します。

### 置換

検索文字列を置き換える文字列を指定します。「置換する」がチェックされている場合だけ、指定できます。

### 大文字 / 小文字の区別

大文字と小文字を区別して検索する場合はチェックします。  
デフォルトではチェックされていません。

### 単語の検索

単語だけを検索する場合にチェックします。  
デフォルトではチェックされていません。

## (2) ダイアログでの操作

- ・[ 前検索 ] ボタンを選択すると、検索文字列を上方向に検索します。
- ・[ 次検索 ] ボタンを選択すると、検索文字列を下方向に検索します。
- ・[ すべて置換 ] ボタンを選択すると、スクリプトファイル中にあるすべての検索文字列を「置換」で指定した文字列に置換します。
- ・[ キャンセル ] ボタンを選択すると、ダイアログを閉じます。

## (3) 処理詳細

- ・「置換」、および「検索」に指定した文字列は、過去 10 件分だけドロップダウンリストに記憶されます。
- ・検索文字列がない場合はピープ音が鳴ります。

## 4.2.7 ファイル名の指定ダイアログ

Script エディタウィンドウで、ファイル名の付いていないスクリプトファイルをモニタリング実行、または実行しようとする、[ ファイル名の指定 ] ダイアログが表示されます。



### (1) ダイアログの項目

スクリプトファイル名

スクリプトファイル名を指定します。

### (2) ダイアログでの操作

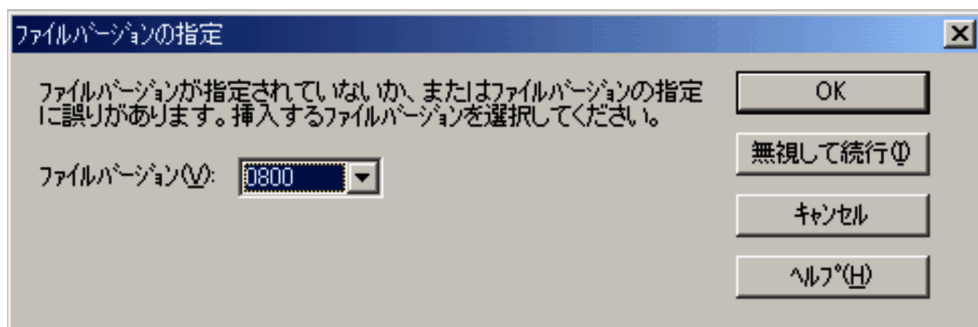
- ・[参照] ボタンを選択すると,[ファイルを開く] ダイアログが表示されます。
- ・[OK] ボタンを選択すると,そのファイル名でスクリプトファイルが保存され,モニタリング実行,または実行が開始されます。
- ・[キャンセル] ボタンを選択すると,ファイル名を設定しないでダイアログを閉じます。
- ・[環境の設定] ボタンを選択すると,[実行環境の設定] ダイアログが表示されます。

## 4.2.8 ファイルバージョンの指定ダイアログ

- ・オプション(互換性)ダイアログの「ファイルバージョンの有無をチェックする」がチェックされている
- ・スクリプトファイルの先頭にファイルバージョンが記述されていない

これら二つの条件を満たすスクリプトファイルに対して,Script エディタウィンドウで,次のうちのどれかの操作をすると,[ファイルバージョンの指定] ダイアログが表示されます。

- a.[ファイル] - [保存] を選択
- b.[ファイル] - [名前を付けて保存] を選択
- c.別のスクリプトファイルを表示
- d.[モニタリング] - [文法チェック] を選択
- e.[モニタリング] - [モニタリングの実行] - [実行] を選択
- f.[モニタリング] - [モニタリングの実行] - [ステップ実行] を選択
- g.[モニタリング] - [モニタリングの実行] - [連続ステップ実行] を選択
- h.[モニタリング] - [実行] を選択
- i.エディタを終了



### (1) ダイアログの項目

#### ファイルバージョン

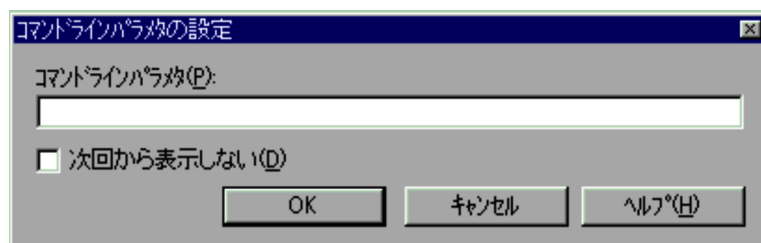
スクリプトファイルの先頭に記述するファイルバージョンを指定します。

### (2) ダイアログでの操作

- [OK] ボタンを選択すると、指定したバージョンをスクリプトファイルの先頭に記述して、ファイルが保存されます。
- [無視して続行] ボタンを選択すると、バージョンを記述しないでスクリプトファイルが保存され、<説明> a ~ i のうち、実行しようとしていた操作が続行されます。
- [キャンセル] ボタンを選択すると、<説明> a ~ i のうち、実行しようとしていた操作が取り消されます。この場合、スクリプトファイルの先頭にバージョンは記述されません。また、スクリプトファイルは保存されません。

## 4.2.9 コマンドラインパラメタの設定ダイアログ

Script エディタウィンドウで、[ モニタリング ] - [ 設定 ] メニューを選択すると表示される [ 設定 ] ダイアログで「モニタリング開始時にコマンドラインパラメタを指定する」がチェックされている状態で、モニタリング実行、ステップ実行、連続ステップ実行のどれかをした場合に [ コマンドラインパラメタの設定 ] ダイアログが表示されます。



### (1) ダイアログの項目

#### コマンドラインパラメタ

モニタリング時に必要となるパラメタを指定します。

#### 次回から表示しない

次回モニタリング開始時に [ コマンドラインパラメタの設定 ] ダイアログを表示させない場合にチェックします。

### (2) ダイアログでの操作

- [OK] ボタンを選択すると、指定した内容が設定されて [ コマンドラインパラメタの設定 ] ダイアログを閉じ、モニタリングを実行します。
- [キャンセル] ボタンを選択すると、内容を変更しないでダイアログを閉じ、モニタリングを実行します。

### (3) 処理詳細

- 指定できるコマンドラインパラメタは半角で 1,024 文字までです。
- [ コマンドラインパラメタの設定 ] ダイアログとスクリプトの実行環境ファイルの両方のコマンドラインに指定がある場合、モニタリング実行時には両方の指定が有効になります。

この場合、[ コマンドラインパラメタの設定 ] ダイアログで指定されたコマンドラインの値が先に、実行環境ファイルで指定されたコマンドラインの値が後になる順序でモニタリング実行します。

ただし、ダイアログに "/SPXLV(3)"、実行環境に "/SPXLV(0)" のように同種で異なる指定が両方にある場合はダイアログの指定が優先されます。

< 例 >

次のように指定された場合、位置変数 %1 は ABC、%2 は 123 となり、/

SPXLV(3) が優先され、/NOEVLOG が有効になります。

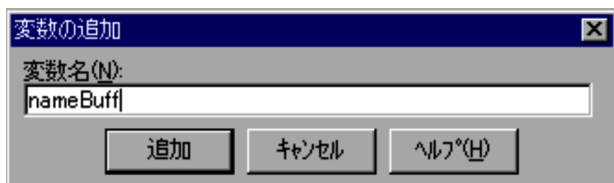
ダイアログで指定されたコマンドライン：/SPXLV(3) ABC

実行環境で指定されたコマンドライン：/SPXLV(0) 123 /NOEVLOG

- [ コマンドラインパラメタの設定 ] ダイアログに設定された内容はエディタ終了時にモニタリング情報ファイルに格納されます。

## 4.2.10 変数の追加ダイアログ

Script エディタウィンドウで、[ モニタリング ] - [ ウォッチへ変数の追加 ] メニューを選択すると、[ 変数の追加 ] ダイアログが表示されます。



### (1) ダイアログの項目

変数名

追加する変数名を指定します。

### (2) ダイアログでの操作

- [ 追加 ] ボタンを選択すると、指定した変数名がウォッチウィンドウに追加されて [ 変数の追加 ] ダイアログが閉じます。
- [ キャンセル ] ボタンを選択すると、変数を追加しないでダイアログを閉じます。

### (3) 処理詳細

- [ モニタリング ] - [ ウォッチへ変数の追加 ] メニューはモニタリングモードの場合にだけ、選択できます。
- 同じ変数を複数指定しても、存在しない変数を指定してもエラーにはなりません。



- 指定できる変数名は半角で 99 文字までです。これを超えた部分は切り捨てられます。

## 4.3 Script トレースビューアウィンドウのダイアログ

Script トレースビューアウィンドウのダイアログトレースビューアウィンドウの操作中表示されるダイアログの一覧を、次に示します。

各ダイアログの項目、および操作については、4.3.1 項以降に説明します。

削除の確認ダイアログ

クリアの確認ダイアログ

コンピュータの選択ダイアログ

検索ダイアログ

### 4.3.1 削除の確認ダイアログ

トレースファイルの一つ、または複数選択してから、[ ファイル ] - [ トレースファイルの削除 ] メニューを選択すると表示されるダイアログです。



#### (1) ダイアログの項目

選択したトレースファイルを削除してもいいですか？

選択されているトレースファイル名が一つ表示されます。

#### (2) ダイアログでの操作

- [ はい ] ボタンを選択すると、表示されているトレースファイルが削除されます。複数のファイルが選択されている場合は、次のファイルに処理が移ります。
- [ すべて ] ボタンを選択すると、選択されているすべてのトレースファイルが削除されます。
- [ いいえ ] ボタンを選択すると、表示されているトレースファイルを削除しません。複数のファイルが選択されている場合は、次のファイルに処理が移ります。
- [ キャンセル ] ボタンをクリックすると、ファイルの削除を中止してダイアログを閉じます。

### 4.3.2 クリアの確認ダイアログ

トレースファイルを一つ、または複数選択してから、[ 編集 ] - [ トレースファイルのクリア ] メニューを選択すると表示されるダイアログです。

選択されているトレースファイルを空（ファイルサイズ 0）にします。



#### (1) ダイアログの項目

選択したトレースファイルをクリアしてもいいですか？

選択されているトレースファイル名が一つ表示されます。

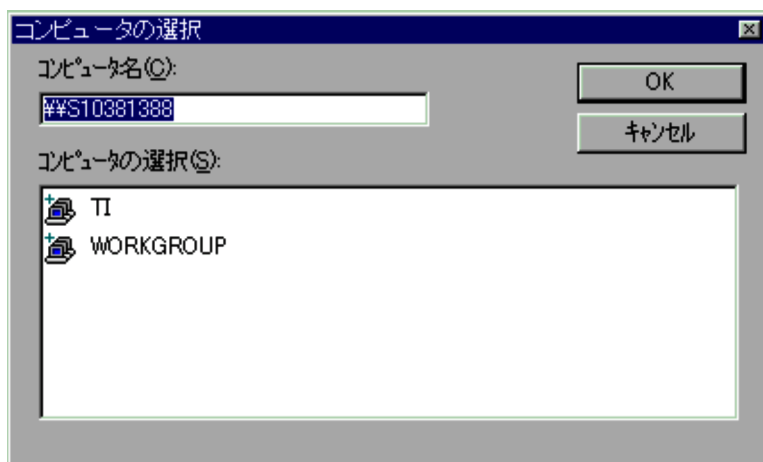
#### (2) ダイアログでの操作

- [ はい ] ボタンを選択すると、表示されているトレースファイルがクリアされます。複数のファイルが選択されている場合は、次のファイルに処理が移ります。
- [ すべて ] ボタンを選択すると、選択されているすべてのトレースファイルがクリアされます。
- [ いいえ ] ボタンを選択すると、表示されているトレースファイルをクリアしません。複数のファイルが選択されている場合は、次のファイルに処理が移ります。
- [ キャンセル ] ボタンをクリックすると、ファイルのクリアを中止してダイアログを閉じます。

### 4.3.3 コンピュータの選択ダイアログ

[ オプション ] - [ コンピュータの選択 ] メニューを選択すると表示されるダイアログです。

接続するコンピュータを変更します。



### (1) ダイアログの項目

#### コンピュータ名

接続先のコンピュータ名（最大 15 文字）を設定します。

#### コンピュータの選択

自コンピュータと接続されているコンピュータの一覧が表示されます。

### (2) ダイアログでの操作

- 一覧からコンピュータを選択して [ OK ] ボタンを選択すると、指定されたコンピュータと接続します。
- [ キャンセル ] ボタンをクリックすると、コンピュータの接続先を変更しないでダイアログを閉じます。

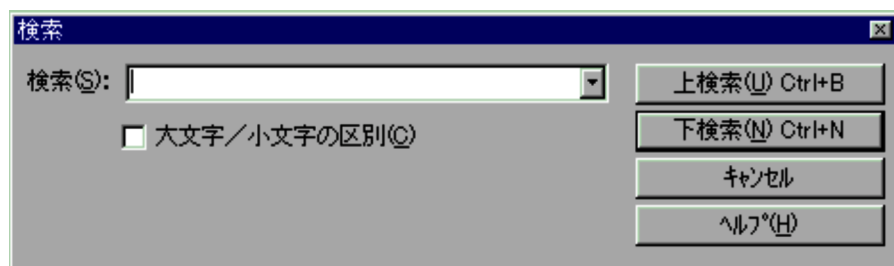
### (3) 処理詳細

- コンピュータ名は最大 15 文字です。
- 指定したコンピュータと接続できない場合は、クライアントエリアには何も表示されません。

## 4.3.4 検索ダイアログ

トレースファイルの表示ウィンドウで、[ 検索 ] - [ 検索 ] メニューを選択すると表示されるダイアログです。

表示されているトレースファイルから、指定した文字列を検索します。



### (1) ダイアログの項目

#### 検索

検索したい文字列を指定します。

#### 大文字 / 小文字の区別

検索する条件として、大文字と小文字を区別する場合にチェックします。

### (2) ダイアログでの操作

- 検索する文字列を指定して [ 上検索 ] ボタンをクリックすると、現在の位置から上に向かって検索を実行します。[ 下検索 ] ボタンをクリックすると、現在の位置から下に向かって検索を実行します。
- [ キャンセル ] ボタンをクリックすると、検索を実行しないでダイアログを閉じます。

### (3) 処理詳細

- 表示中の文字列を選択している状態で [ 検索 ] メニューを選択すると、選択中の文字列を検索文字列と仮定します。

## 4.4 Script メニューエディタウィンドウのダイアログ

---

Script メニューエディタウィンドウの操作中に表示されるダイアログの一覧を、次に示します。各ダイアログの項目、および操作については、4.4.1 項以降に説明します。

- ・メニューフォームプロパティ（一般）ダイアログ
- ・メニューフォームプロパティ（背景）ダイアログ
- ・メニューフォームプロパティ（キー）ダイアログ
- ・メニューフォームプロパティ（スタイル）ダイアログ
- ・メニューフォームプロパティ（壁紙）ダイアログ
- ・スタティックプロパティ（一般）ダイアログ
- ・スタティックプロパティ（共通）ダイアログ
- ・スタティックプロパティ（背景）ダイアログ
- ・スタティックプロパティ（スタイル）ダイアログ
- ・ボタンプロパティ（一般）ダイアログ
- ・ボタンプロパティ（共通）ダイアログ
- ・ボタンプロパティ（背景）ダイアログ
- ・ボタンプロパティ（キー）ダイアログ
- ・ボタンプロパティ（スタイル）ダイアログ
- ・ファイル参照ボタンプロパティ（一般）ダイアログ
- ・ファイル参照ボタンプロパティ（共通）ダイアログ
- ・ファイル参照ボタンプロパティ（背景）ダイアログ
- ・ファイル参照ボタンプロパティ（キー）ダイアログ
- ・ファイル参照ボタンプロパティ（スタイル）ダイアログ
- ・エディットコントロールプロパティ（一般）ダイアログ
- ・エディットコントロールプロパティ（共通）ダイアログ
- ・エディットコントロールプロパティ（キー）ダイアログ
- ・エディットコントロールプロパティ（スタイル）ダイアログ
- ・直線プロパティ（一般）ダイアログ
- ・ファンクションキープロパティ（一般）ダイアログ
- ・ファンクションキープロパティ（共通）ダイアログ
- ・ファンクションキープロパティ（スタイル）ダイアログ
- ・リストコントロールプロパティ（一般）ダイアログ
- ・リストコントロールプロパティ（共通）ダイアログ
- ・リストコントロールプロパティ（キー）ダイアログ
- ・リストコントロールプロパティ（スタイル）ダイアログ
- ・コンボボックスプロパティ（一般）ダイアログ
- ・コンボボックスプロパティ（共通）ダイアログ
- ・コンボボックスプロパティ（キー）ダイアログ
- ・コンボボックスプロパティ（スタイル）ダイアログ
- ・コマンドプロパティダイアログ

- ・ コマンドプロパティの設定ダイアログ
- ・ メニューフォーム名の指定ダイアログ
- ・ 一括変更ダイアログ
- ・ グリッドの設定ダイアログ
- ・ タブオーダーの設定ダイアログ
- ・ メニューフォームの印刷情報ダイアログ

#### 4.4.1 メニューフォームプロパティ（一般）ダイアログ

Script メニューエディタウィンドウのメニューフォームビューで、メニューフォームを選択してダブルクリックすると、[メニューフォームプロパティ] ダイアログが表示されます。

[メニューフォームプロパティ] ダイアログには、「一般」、「背景」、「キー」、「スタイル」、「壁紙」の五つのタブがあります。

「一般」タブを選択すると、[メニューフォームプロパティ（一般）] ダイアログが表示されます。



##### （１）ダイアログの項目

[メニューフォームプロパティ（一般）] ダイアログでは、選択したメニューフォームに対して、次の項目を設定できます。

##### キャプション

メニューフォームのタイトルのキャプションを指定します。

##### 変数名

「キャプション」に変数名を指定する場合にチェックします。

#### 4. ダイアログの説明

「キャプション」は変数名で指定してください。この変数名が配列変数の場合、メニューフォーム実行時にエラーになります。

##### メニューフォーム名

メニューフォーム名を指定します。

メニューフォーム名は、変数の名前付け規則に従って指定してください。

##### 終了監視時間

メニューフォームの終了時間を秒単位で指定します。

0 ~ 86,400 の範囲で指定してください。ただし、0 が指定された場合は終了監視しないものとみなします。

##### 表示位置を指定する

メニューフォームを起動したときの表示位置を指定する場合にチェックします。

##### 幅

メニューフォームの幅をピクセル単位で指定します。

115 ~ 9,999 の範囲で指定してください。デフォルトでは「600」が設定されています。

##### 高さ

メニューフォームの高さをピクセル単位で指定します。

22 ~ 9,999 の範囲で指定してください。デフォルトでは「400」が設定されています。

##### 背景色

メニューフォームの背景色を色の設定ダイアログから指定します。

デフォルトでは「灰色」が設定されています。

##### X座標・Y座標

メニューフォームを起動したときの表示位置の X 座標と Y 座標をピクセル単位で指定します。

「表示位置を指定する」が選択されている場合にだけ指定できます。

「X座標」も「Y座標」もデフォルトでは「0」が設定されています。

##### 変数で指定

「X座標」と「Y座標」を変数で指定する場合にチェックします。

「表示位置を指定する」がチェックされている場合にだけチェックできます。

「X座標」も「Y座標」も変数名で指定してください。この変数名が配列変数の場合、メニューフォーム実行時にエラーになります。

#### (2) ダイアログでの操作

- [OK] ボタンを選択すると、指定した内容が設定されて [メニューフォームプロパティ] ダイアログが閉じます。
- [キャンセル] ボタンを選択すると、内容を変更しないでダイアログを閉じます。



## (3) 処理詳細

- ・「表示位置を指定する」がチェックされていない場合は、起動したときにメニューフォームは画面の中央に表示されます。

## 4.4.2 メニューフォームプロパティ（背景）ダイアログ

Script メニューエディタウィンドウのメニューフォームビューで、メニューフォームを選択してダブルクリックすると、[メニューフォームプロパティ] ダイアログが表示されます。

[メニューフォームプロパティ] ダイアログには、「一般」、「背景」、「キー」、「スタイル」、「壁紙」の五つのタブがあります。

「背景」タブを選択すると、[メニューフォームプロパティ（背景）] ダイアログが表示されます。



## (1) ダイアログの項目

[メニューフォームプロパティ（背景）] ダイアログでは、選択したメニューフォームの背景について、次の項目を設定できます。

## 表示形式

背景に画像を表示させる方法をドロップダウンリストから選択します。  
デフォルトでは「画像を表示しない」が設定されています。

## ファイル名

背景に表示させる画像のファイル名を指定します。  
ただし、「表示形式」で「画像を表示しない」が選択されている場合は指定できません。

##### 参照

画像ファイルを指定します。

ただし、「表示形式」で「画像を表示しない」が選択されている場合は指定できません。

#### (2) ダイアログでの操作

- [OK] ボタンを選択すると、指定した内容が設定されて [メニューフォームプロパティ] ダイアログが閉じます。
- [キャンセル] ボタンを選択すると、内容を変更しないでダイアログを閉じます。

### 4.4.3 メニューフォームプロパティ (キー) ダイアログ

Script メニューエディタウィンドウのメニューフォームビューで、メニューフォームを選択してダブルクリックすると、[メニューフォームプロパティ] ダイアログが表示されます。

[メニューフォームプロパティ] ダイアログには、「一般」、「背景」、「キー」、「スタイル」、「壁紙」の五つのタブがあります。

「キー」タブを選択すると、[メニューフォームプロパティ (キー)] ダイアログが表示されます。



#### (1) ダイアログの項目

[メニューフォームプロパティ (キー)] ダイアログでは、選択したメニューフォームのキーについて、次の項目を設定できます。

##### キー

終了キーとして設定するキーを選択します。

キーとして選択できるのは、Esc、Enter、および F1 ~ F12 です。

Ctrl

モデファイキーに Ctrl キーを指定する場合にチェックします。

Shift

モデファイキーに Shift キーを指定する場合にチェックします。

Alt

モデファイキーに Alt キーを指定する場合にチェックします。

>> 追加

選択したキーを終了キーとして追加する場合にクリックします。

<< 削除

選択したキーを終了キーから削除する場合にクリックします。

終了キー

終了キーから削除するキーを選択します。

デフォルトでは「Esc」が設定されています。

## (2) ダイアログでの操作

- [OK] ボタンを選択すると、指定した内容が設定されて [メニューフォームプロパティ] ダイアログが閉じます。
- [キャンセル] ボタンを選択すると、内容を変更しないでダイアログを閉じます。

## (3) 処理詳細

- 終了キーが「Esc」の場合、入力されたデータのチェック処理や変数への値の代入などはしないでメニューフォームを終了します。
- メニューフォーム上にファンクションキーがある場合、F1 ~ F12 は終了キーとして選択できないため、「キー」のリストには表示されません。

また、次の場合には、「終了キー」の F1 ~ F12 の設定は無効になります。メニューフォームを開いた時点で警告メッセージが表示されます。

- すでに F1 ~ F12 を「終了キー」に設定している状態でファンクションキーを定義した場合。
- JP1/Script 06-71 より前のバージョンで作成されたメニューフォームで、ファンクションキーが定義され、「終了キー」に F1 ~ F12 が設定されている場合。

### 4.4.4 メニューフォームプロパティ (スタイル) ダイアログ

Script メニューエディタウィンドウのメニューフォームビューで、メニューフォームを選択してダブルクリックすると、[メニューフォームプロパティ] ダイアログが表示されます。

#### 4. ダイアログの説明

[メニューフォームプロパティ] ダイアログには、「一般」、「背景」、「キー」、「スタイル」、「壁紙」の五つのタブがあります。

「スタイル」タブを選択すると、[メニューフォームプロパティ (スタイル)] ダイアログが表示されます。



##### (1) ダイアログの項目

[メニューフォームプロパティ (スタイル)] ダイアログでは、選択したメニューフォームについて、次の項目を設定できます。

###### 最小化ボタン

メニューフォームに最小化ボタンを付ける場合にチェックします。  
「タイトルバー」がチェックされている場合にだけチェックできます。

###### システムメニュー

メニューフォームにシステムメニューを設定する場合にチェックします。  
「タイトルバー」がチェックされている場合にだけチェックできます。  
デフォルトではチェックされています。

###### フォアグラウンドに表示

メニューフォームをフォアグラウンドに表示させる場合にチェックします。

###### タイトルバー

メニューフォームにタイトルバーを付ける場合にチェックします。  
デフォルトではチェックされています。

##### (2) ダイアログでの操作

- ・[OK] ボタンを選択すると、指定した内容が設定されて [メニューフォームプロパ

ティ] ダイアログが閉じます。

- [キャンセル] ボタンを選択すると、内容を変更しないでダイアログを閉じます。

#### 4.4.5 メニューフォームプロパティ ( 壁紙 ) ダイアログ

Script メニューエディタウィンドウのメニューフォームビューで、メニューフォームを選択してダブルクリックすると、[メニューフォームプロパティ] ダイアログが表示されます。

[メニューフォームプロパティ] ダイアログには、「一般」、「背景」、「キー」、「スタイル」、「壁紙」の五つのタブがあります。

「壁紙」タブを選択すると、[メニューフォームプロパティ ( 壁紙 )] ダイアログが表示されます。



##### ( 1 ) ダイアログの項目

[メニューフォームプロパティ ( 壁紙 )] ダイアログでは、選択したメニューフォームに貼る壁紙について、次の項目を設定できます。

##### 表示形式

壁紙となる画像の表示方法をドロップダウンリストから選択します。

デフォルトでは「画像を表示しない」が設定されています。

##### タスクバーを隠さず表示

タスクバーを隠さないで画像を壁紙として表示させる場合にチェックします。

##### ファイル名

壁紙として表示させる画像のファイル名を指定します。

## (2) ダイアログでの操作

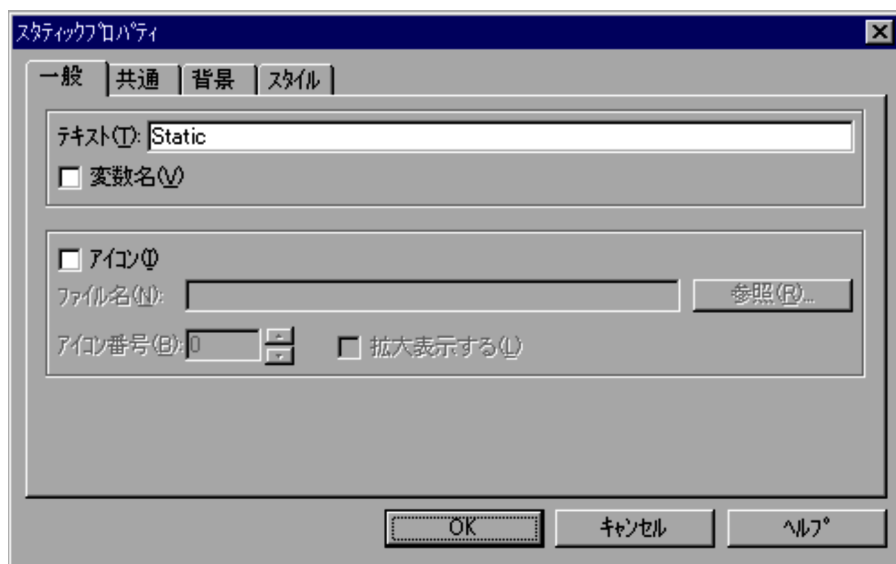
- [OK] ボタンを選択すると、指定した内容が設定されて [メニューフォームプロパティ] ダイアログが閉じます。
- [キャンセル] ボタンを選択すると、内容を変更しないでダイアログを閉じます。

## 4.4.6 スタティックプロパティ (一般) ダイアログ

Script メニューエディタウィンドウのメニューフォームビューで、スタティックコントロールを選択してダブルクリックすると、[スタティックプロパティ] ダイアログが表示されます。

[スタティックプロパティ] ダイアログには、「一般」、「共通」、「背景」、「スタイル」の四つのタブがあります。

「一般」タブを選択すると、[スタティックプロパティ (一般)] ダイアログが表示されます。



## (1) ダイアログの項目

[スタティックプロパティ (一般)] ダイアログでは、スタティックコントロールに表示させるテキストについて、次の項目を設定できます。

### テキスト

表示させるテキストを指定します。

### 変数名

「テキスト」に変数名を指定する場合にチェックします。

「テキスト」は変数名で指定してください。この変数名が配列変数の場合、メニュー

フォーム実行時にエラーになります。

#### アイコン

スタティックコントロールにアイコンを表示させる場合にチェックします。

#### ファイル名

スタティックコントロールに表示させるアイコンファイル名を指定します。

「アイコン」がチェックされている場合にだけ設定できます。

#### 参照

[ ファイルを開く ] ダイアログからアイコンファイルを指定します。

「アイコン」がチェックされている場合にだけ設定できます。

#### アイコン番号

アイコン番号を 0 から始まる数値で指定します。

「アイコン」がチェックされている場合にだけ設定できます。

#### 拡大表示する

スタティックコントロールに 2 倍拡大したアイコンを表示させる場合にチェックします。

「アイコン」がチェックされている場合にだけ設定できます。

### (2) ダイアログでの操作

- [ OK ] ボタンを選択すると、指定した内容が設定されて [ スタティックプロパティ ] ダイアログが閉じます。
- [ キャンセル ] ボタンを選択すると、内容を変更しないでダイアログを閉じます。

## 4.4.7 スタティックプロパティ ( 共通 ) ダイアログ

Script メニューエディタウィンドウのメニューフォームビューで、スタティックコントロールを選択してダブルクリックすると、[ スタティックプロパティ ] ダイアログが表示されます。

[ スタティックプロパティ ] ダイアログには、「一般」、「共通」、「背景」、「スタイル」の四つのタブがあります。

「共通」タブを選択すると、[ スタティックプロパティ ( 共通 ) ] ダイアログが表示されます。



### (1) ダイアログの項目

[スタティックプロパティ (共通)] ダイアログでは、スタティックコントロールに表示させるテキストについて、次の項目を設定できます。

#### フィールド名

フィールドの名称を指定します。

フィールド名は、変数の名前付け規則に従って指定してください。

#### テキストの配置

##### 縦方向・横方向

テキストを配置するときの縦方向と横方向の位置をドロップダウンリストから選択します。

デフォルトでは「縦方向」は「上寄り」、「横方向」は「左寄り」が設定されています。

#### テキストカラー

テキストの色を指定します。

デフォルトでは「黒色」が設定されています。

#### 背景色

スタティックコントロールの背景の色を指定します。

デフォルトでは「灰色」が設定されています。

#### フォント名

現在選択されているフォント名を表示します。

フォントを変更するには、[フォント] ボタンを選択します。

デフォルトでは「MS Pゴシック」が設定されています。



### スタイル

現在選択されているフォントのスタイルを表示します。  
フォントのスタイルを変更するには、[ フォント ] ボタンを選択します。  
デフォルトでは「標準」が設定されています。

### サイズ

現在選択されているフォントのサイズを表示します。  
フォントのサイズを変更するには、[ フォント ] ボタンを選択します。  
デフォルトでは「12」が設定されています。

### 取り消し線

取り消し線の指定の有無を表示します。  
取り消し線を変更するには、[ フォント ] ボタンを選択します。  
デフォルトでは「なし」が設定されています。

### 下線

下線の指定の有無を表示します。  
下線を変更するには、[ フォント ] ボタンを選択します。  
デフォルトでは「なし」が設定されています。

### 表示状態を制御する

コントロールの表示状態を制御する場合にチェックします。

### 変数名

コントロールの表示状態を示す値を格納する変数名を指定します。  
「表示状態を制御する」がチェックされている場合にだけ設定できます。  
この変数名が配列変数の場合、メニューフォーム実行時にエラーになります。  
コントロールの表示状態を示す値は次のとおりです。

#### DISABLE

コントロールはメニューフォーム上に無効状態で表示されます。

#### HIDE

コントロールはメニューフォーム上には表示されません。

#### その他

コントロールはメニューフォーム上に有効状態で表示されます。

### ! 注意事項

JP1/Script 06-71 以降では、[ 起動時にフォーカスを位置付ける ] は無効状態で表示されるようになりました。

## (2) ダイアログでの操作

- [ OK ] ボタンを選択すると、指定した内容が設定されて [ スタティックプロパティ ] ダイアログが閉じます。
- [ キャンセル ] ボタンを選択すると、内容を変更しないでダイアログを閉じます。

### (3) 処理詳細

- 「表示状態を制御する」がチェックされていない場合、コントロールはメニューフォーム上に有効状態で表示されます。
- テスト表示では「表示状態を制御する」の指定は無効になります。

## 4.4.8 スタティックプロパティ (背景) ダイアログ

Script メニューエディタウィンドウのメニューフォームビューで、スタティックコントロールを選択してダブルクリックすると、[スタティックプロパティ] ダイアログが表示されます。

[スタティックプロパティ] ダイアログには、「一般」、「共通」、「背景」、「スタイル」の四つのタブがあります。

「背景」タブを選択すると、[スタティックプロパティ (背景)] ダイアログが表示されます。



### (1) ダイアログの項目

[スタティックプロパティ (背景)] ダイアログでは、スタティックコントロールの背景に表示させる画像について、次の項目を設定できます。

#### 表示形式

背景に画像を表示させる方法をドロップダウンリストから選択します。

デフォルトでは「画像を表示しない」が設定されています。

#### ファイル名

背景に表示させる画像のファイル名を指定します。

ただし、「表示形式」で「画像を表示しない」が選択されている場合は指定できません。

ん。

#### 参照

画像ファイルを指定します。

ただし、「表示形式」で「画像を表示しない」が選択されている場合は指定できません。

#### (2) ダイアログでの操作

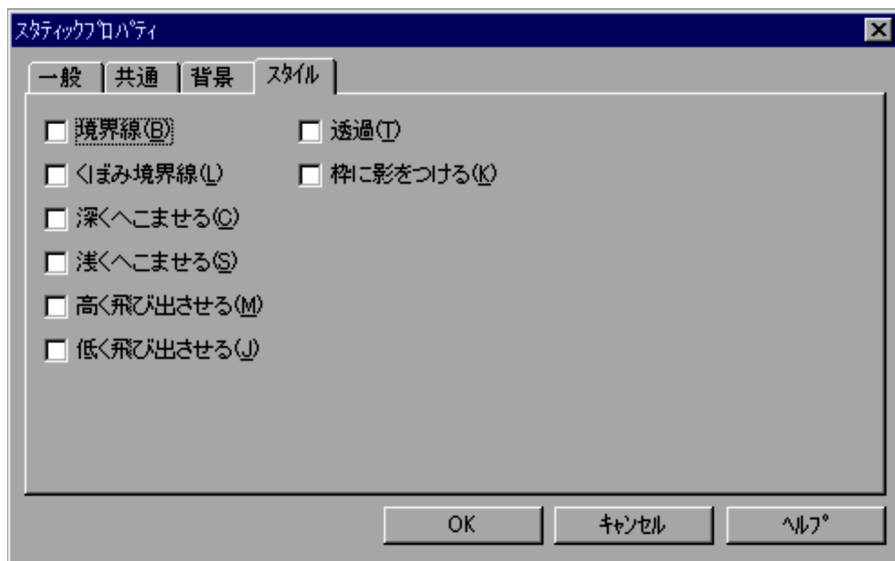
- [OK] ボタンを選択すると、指定した内容が設定されて [スタティックプロパティ] ダイアログが閉じます。
- [キャンセル] ボタンを選択すると、内容を変更しないでダイアログを閉じます。

### 4.4.9 スタティックプロパティ (スタイル) ダイアログ

Script メニューエディタウィンドウのメニューフォームビューで、スタティックコントロールを選択してダブルクリックすると、[スタティックプロパティ] ダイアログが表示されます。

[スタティックプロパティ] ダイアログには、「一般」、「共通」、「背景」、「スタイル」の四つのタブがあります。

「スタイル」タブを選択すると、[スタティックプロパティ (スタイル)] ダイアログが表示されます。



#### (1) ダイアログの項目

[スタティックプロパティ (スタイル)] ダイアログでは、スタティックコントロールのスタイルについて、次の項目を設定できます。

#### 4. ダイアログの説明

##### 境界線

コントロールの周囲に境界線を作成する場合にチェックします。

##### くぼみ境界線

コントロールの周囲にくぼんだ境界線を作成する場合にチェックします。

##### 深くへこませる

コントロールの周囲を深くへこませる場合にチェックします。

##### 浅くへこませる

コントロールの周囲を浅くへこませる場合にチェックします。

##### 高く飛び出させる

コントロールの周囲を高く飛び出させる場合にチェックします。

##### 低く飛び出させる

コントロールの周囲を低く飛び出させる場合にチェックします。

##### 透過

コントロールを透過させる場合にチェックします。

##### 枠に影をつける

コントロールの枠に影を付ける場合にチェックします。

#### (2) ダイアログでの操作

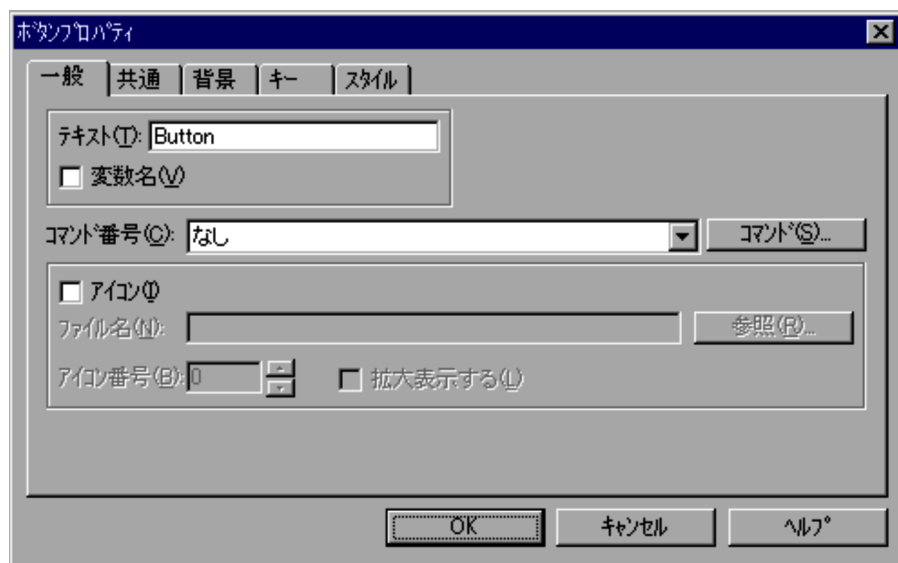
- [OK] ボタンを選択すると、指定した内容が設定されて [ スタティックプロパティ ] ダイアログが閉じます。
- [キャンセル] ボタンを選択すると、内容を変更しないでダイアログを閉じます。

### 4.4.10 ボタンプロパティ (一般) ダイアログ

Script メニューエディタウィンドウのメニューフォームビューで、ボタンコントロールを選択してダブルクリックすると、[ ボタンプロパティ ] ダイアログが表示されます。

[ ボタンプロパティ ] ダイアログには、「一般」、「共通」、「背景」、「キー」、「スタイル」の五つのタブがあります。

「一般」タブを選択すると、[ ボタンプロパティ (一般) ] ダイアログが表示されます。



### (1) ダイアログの項目

[ ボタンプロパティ (一般) ] ダイアログでは、ボタンコントロールについて、次の項目を設定できます。

#### テキスト

ボタンコントロールに表示するテキストを指定します。

#### 変数名

「テキスト」に変数名を指定する場合にチェックします。

「テキスト」は変数名で指定してください。この変数名が配列変数の場合、メニューフォーム実行時にエラーになります。

#### コマンド番号

ボタンをクリックしたときに実行されるコマンドの番号を指定します。

#### コマンド

このボタンを選択すると、[ コマンドプロパティの設定 ] ダイアログが表示されます。

コマンドのプロパティを設定します。

#### アイコン

ボタンにアイコンを表示させる場合にチェックします。

#### ファイル名

ボタンに表示させるアイコンファイル名を指定します。

「アイコン」がチェックされている場合にだけ設定できます。

#### 参照

#### 4. ダイアログの説明

[ ファイルを開く ] ダイアログからアイコンファイルを指定します。  
「アイコン」がチェックされている場合にだけ設定できます。

##### アイコン番号

アイコンの番号を 0 から始まる数値で指定します。

「アイコン」がチェックされている場合にだけ設定できます。

##### 拡大表示する

ボタンに 2 倍拡大したアイコンを表示させる場合にチェックします。

「アイコン」がチェックされている場合にだけ設定できます。

#### (2) ダイアログでの操作

- [ OK ] ボタンを選択すると、指定した内容が設定されて [ ボタンプロパティ ] ダイアログが閉じます。
- [ キャンセル ] ボタンを選択すると、内容を変更しないでダイアログを閉じます。

### 4.4.11 ボタンプロパティ ( 共通 ) ダイアログ

Script メニューエディタウィンドウのメニューフォームビューで、ボタンコントロールを選択してダブルクリックすると、[ ボタンプロパティ ] ダイアログが表示されます。

[ ボタンプロパティ ] ダイアログには、「一般」、「共通」、「背景」、「キー」、「スタイル」の五つのタブがあります。

「共通」タブを選択すると、[ ボタンプロパティ ( 共通 ) ] ダイアログが表示されます。



#### (1) ダイアログの項目

[ ボタンプロパティ ( 共通 ) ] ダイアログでは、ボタンコントロールについて、次の項目

を設定できます。

#### フィールド名

フィールドの名称を指定します。変数の名前付け規則に従って指定してください。

#### テキストの配置

##### 縦方向・横方向

テキストを配置するときの縦方向と横方向の位置をドロップダウンリストから選択します。

デフォルトでは、「縦方向」「横方向」とも、「中央」が設定されています。

#### 起動時にフォーカスを位置付ける

メニューフォームを起動するとき、フォーカスを位置付ける場合にチェックします。

#### 再表示時にフォーカスを位置付ける

メニューフォームを再表示するとき、フォーカスを位置付ける場合にチェックします。

#### テキストカラー

テキストの色を指定します。

デフォルトでは「黒色」が設定されています。

#### 背景色

ボタンコントロールの背景の色を指定します。

デフォルトでは「灰色」が設定されています。

#### フォント名

現在選択されているフォント名を表示します。

フォントを変更するには、[ フォント ] ボタンを選択します。

デフォルトでは「MS P ゴシック」が設定されています。

#### スタイル

現在選択されているフォントのスタイルを表示します。

フォントのスタイルを変更するには、[ フォント ] ボタンを選択します。

デフォルトでは「標準」が設定されています。

#### サイズ

現在選択されているフォントのサイズを表示します。

フォントのサイズを変更するには、[ フォント ] ボタンを選択します。

デフォルトでは「12」が設定されています。

#### 取り消し線

取り消し線の指定の有無を表示します。

取り消し線を変更するには、[ フォント ] ボタンを選択します。

デフォルトでは「なし」が設定されています。

#### 下線

#### 4. ダイアログの説明

下線の指定の有無を表示します。

下線を変更するには、[ フォント ] ボタンを選択します。

デフォルトでは「なし」が設定されています。

##### 表示状態を制御する

コントロールの表示状態を制御する場合にチェックします。

##### 変数名

コントロールの表示状態を示す値を格納する変数名を指定します。

「表示状態を制御する」がチェックされている場合にだけ設定できます。

この変数名が配列変数の場合、メニューフォーム実行時にエラーになります。

コントロールの表示状態を示す値は次のとおりです。

##### DISABLE

コントロールはメニューフォーム上に無効状態で表示されます。

##### HIDE

コントロールはメニューフォーム上には表示されません。

##### その他

コントロールはメニューフォーム上に有効状態で表示されます。

#### (2) ダイアログでの操作

- [ OK ] ボタンを選択すると、指定した内容が設定されて [ ボタンプロパティ ] ダイアログが閉じます。
- [ キャンセル ] ボタンを選択すると、内容を変更しないでダイアログを閉じます。

#### (3) 処理詳細

- 「表示状態を制御する」がチェックされていない場合、コントロールはメニューフォーム上に有効状態で表示されます。
- テスト表示では「表示状態を制御する」の指定は無効になります。

### 4.4.12 ボタンプロパティ ( 背景 ) ダイアログ

Script メニューエディタウィンドウのメニューフォームビューで、ボタンコントロールを選択してダブルクリックすると、[ ボタンプロパティ ] ダイアログが表示されます。

[ ボタンプロパティ ] ダイアログには、「一般」、「共通」、「背景」、「キー」、「スタイル」の五つのタブがあります。

「背景」タブを選択すると、[ ボタンプロパティ ( 背景 ) ] ダイアログが表示されます。





### (1) ダイアログの項目

[ ボタンプロパティ ( 背景 ) ] ダイアログでは、ボタンコントロールの背景について、次の項目を設定できます。

#### 表示形式

背景に画像を表示させる方法をドロップダウンリストから選択します。  
デフォルトでは「画像を表示しない」が設定されています。

#### ファイル名

背景に表示させる画像のファイル名を指定します。  
ただし、「表示形式」で「画像を表示しない」が選択されている場合は指定できません。

#### 参照

画像ファイルを指定します。  
ただし、「表示形式」で「画像を表示しない」が選択されている場合は指定できません。

### (2) ダイアログでの操作

- [ OK ] ボタンを選択すると、指定した内容が設定されて [ ボタンプロパティ ] ダイアログが閉じます。
- [ キャンセル ] ボタンを選択すると、内容を変更しないでダイアログを閉じます。

## 4.4.13 ボタンプロパティ ( キー ) ダイアログ

Script メニューエディタウィンドウのメニューフォームビューで、ボタンコントロールを選択してダブルクリックすると、[ ボタンプロパティ ] ダイアログが表示されます。

#### 4. ダイアログの説明

[ ボタンプロパティ ] ダイアログには、「一般」、「共通」、「背景」、「キー」、「スタイル」の五つのタブがあります。

「キー」タブを選択すると、[ ボタンプロパティ ( キー ) ] ダイアログが表示されます。



##### ( 1 ) ダイアログの項目

[ ボタンプロパティ ( キー ) ] ダイアログでは、ボタンコントロールのキーについて、次の項目を設定できます。

##### アクセラレータキーを指定する

アクセラレータキーを指定する場合にチェックします。

##### Ctrl

モデファイキーに Ctrl キーを指定する場合にチェックします。

「アクセラレータキーを指定する」がチェックされている場合にだけチェックできます。

##### Shift

モデファイキーに Shift キーを指定する場合にチェックします。

「アクセラレータキーを指定する」がチェックされている場合にだけチェックできます。

##### Alt

モデファイキーに Alt キーを指定する場合にチェックします。

「アクセラレータキーを指定する」がチェックされている場合にだけチェックできます。

##### キー

指定するキーをドロップダウンリストから選択します。

「アクセラレータキーを指定する」がチェックされている場合にだけ選択できます。

## (2) ダイアログでの操作

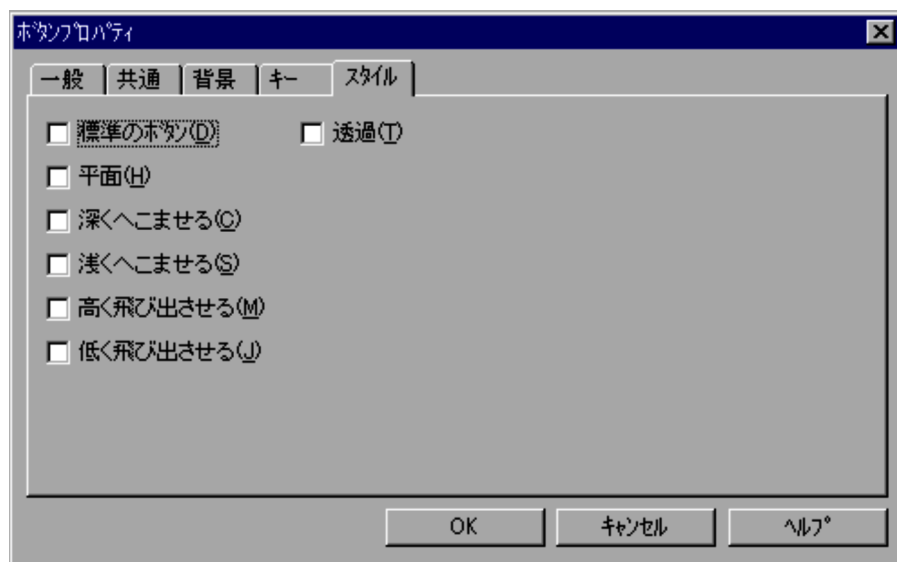
- [OK] ボタンを選択すると、指定した内容が設定されて [ ボタンプロパティ ] ダイアログが閉じます。
- [キャンセル] ボタンを選択すると、内容を変更しないでダイアログを閉じます。

### 4.4.14 ボタンプロパティ (スタイル) ダイアログ

Script メニューエディタウィンドウのメニューフォームビューで、ボタンコントロールを選択してダブルクリックすると、[ ボタンプロパティ ] ダイアログが表示されます。

[ ボタンプロパティ ] ダイアログには、「一般」、「共通」、「背景」、「キー」、「スタイル」の五つのタブがあります。

「スタイル」タブを選択すると、[ ボタンプロパティ (スタイル) ] ダイアログが表示されます。



## (1) ダイアログの項目

[ ボタンプロパティ (スタイル) ] ダイアログでは、ボタンコントロールの背景について、次の項目を設定できます。

### 標準のボタン

標準のボタンを指定する場合にチェックします。

標準のボタンはメニューフォーム上で一つだけ指定できます。

### 平面

#### 4. ダイアログの説明

ボタンを 3D 表示しない場合にチェックします。

深くへこませる

コントロールの周囲を深くへこませる場合にチェックします。

浅くへこませる

コントロールの周囲を浅くへこませる場合にチェックします。

高く飛び出させる

コントロールの周囲を高く飛び出させる場合にチェックします。

低く飛び出させる

コントロールの周囲を低く飛び出させる場合にチェックします。

透過

コントロールを透過させる場合にチェックします。

#### (2) ダイアログでの操作

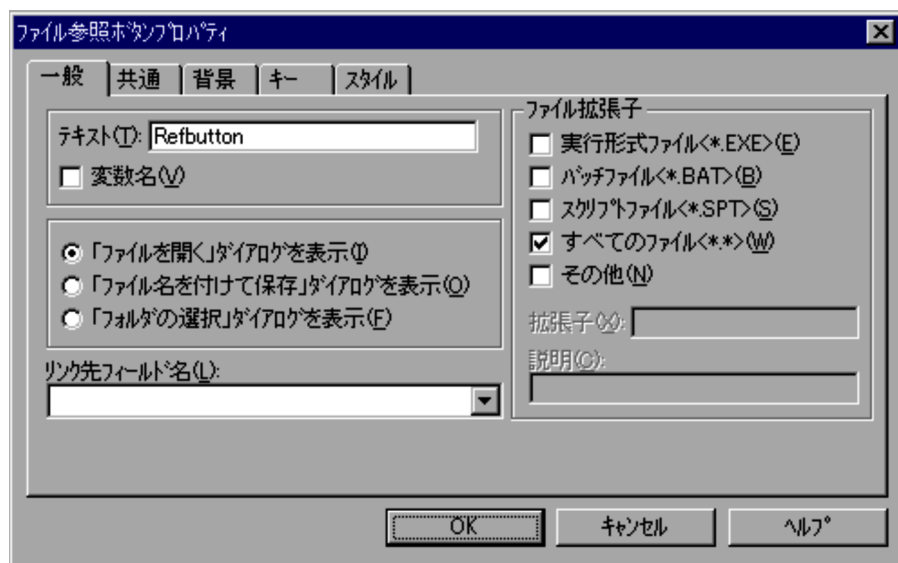
- [OK] ボタンを選択すると、指定した内容が設定されて [ ボタンプロパティ ] ダイアログが閉じます。
- [キャンセル] ボタンを選択すると、内容を変更しないでダイアログを閉じます。

### 4.4.15 ファイル参照ボタンプロパティ (一般) ダイアログ

Script メニューエディタウィンドウのメニューフォームビューで、ファイル参照ボタンを選択してダブルクリックすると、[ ファイル参照ボタンプロパティ ] ダイアログが表示されます。

[ ファイル参照ボタンプロパティ ] ダイアログには、「一般」、「共通」、「背景」、「キー」、「スタイル」の五つのタブがあります。

「一般」タブを選択すると、[ ファイル参照ボタンプロパティ (一般) ] ダイアログが表示されます。



### (1) ダイアログの項目

[ファイル参照ボタンプロパティ (一般)] ダイアログでは、ファイル参照ボタンについて、次の項目を設定できます。

#### テキスト

ファイル参照ボタンに表示させるテキストを指定します。

#### 変数名

「テキスト」に、変数名を指定する場合にチェックします。

「テキスト」は変数名で指定してください。この変数名が配列変数の場合、メニューフォーム実行時にエラーになります。

#### 「ファイルを開く」ダイアログを表示

[ファイルを開く] ダイアログを表示させる場合にチェックします。

#### 「ファイル名を付けて保存」ダイアログを表示

[ファイル名を付けて保存] ダイアログを表示させる場合にチェックします。

#### 「フォルダの選択」ダイアログを表示

[フォルダの選択] ダイアログを表示させる場合にチェックします。

#### リンク先フィールド名

リンクしているコントロールのフィールド名を指定します。必ず指定してください。

#### ファイル拡張子

[ファイルを開く] ダイアログ、または [ファイル名をつけて保存] ダイアログに表示させる、ファイルのフィルタの形式を指定します。次の拡張子から選択します。

実行形式ファイル <\*.EXE>

#### 4. ダイアログの説明

フィルタに実行形式ファイルを指定する場合にチェックします。

バッチファイル <\*.BAT>

フィルタにバッチファイルを指定する場合にチェックします。

スクリプトファイル <\*.SPT>

フィルタにスクリプトファイルを指定する場合にチェックします。

すべてのファイル <\*. \*>

フィルタにすべてのファイルを指定する場合にチェックします。

その他

フィルタに上記以外のファイルを指定する場合にチェックします。

拡張子

フィルタに指定する拡張子を指定します。「その他」をチェックした場合にだけ使用できます。

説明

フィルタに表示されるファイルの説明を指定します。「その他」をチェックした場合にだけ使用できます。

ただし、「フォルダの選択」ダイアログを表示」がチェックされている場合にはグレーアウトされて選択できません。

#### (2) ダイアログでの操作

- [OK] ボタンを選択すると、指定した内容が設定されて [ ファイル参照ボタンプロパティ ] ダイアログが閉じます。
- [キャンセル] ボタンを選択すると、内容を変更しないでダイアログを閉じます。

#### (3) 処理詳細

- 「「ファイルを開く」ダイアログを表示」、または「「ファイル名を付けて保存」ダイアログを表示」が定義されている場合、リンク先のフィールドに "¥" や "/" などを含む文字列を指定して [ ファイル参照 ] ボタンを選択してもダイアログは表示されません。

### 4.4.16 ファイル参照ボタンプロパティ ( 共通 ) ダイアログ

Script メニューエディタウィンドウのメニューフォームビューで、ファイル参照ボタンを選択してダブルクリックすると、[ ファイル参照ボタンプロパティ ] ダイアログが表示されます。

[ ファイル参照ボタンプロパティ ] ダイアログには、「一般」、「共通」、「背景」、「キー」、「スタイル」の五つのタブがあります。

「共通」タブを選択すると、[ ファイル参照ボタンプロパティ ( 共通 ) ] ダイアログが表示されます。



### (1) ダイアログの項目

[ファイル参照ボタンプロパティ (共通)] ダイアログでは、ファイル参照ボタンに表示させるテキストについて、次の項目を設定できます。

#### フィールド名

フィールドの名称を指定します。

フィールド名は、変数の名前付け規則に従って指定してください。

#### テキストの配置

##### 縦方向・横方向

テキストを配置するときの縦方向と横方向の位置をドロップダウンリストから選択します。

デフォルトでは「縦方向」「横方向」とも、「中央」が設定されています。

#### 起動時にフォーカスを位置付ける

メニューフォームを起動するとき、フォーカスを位置付ける場合にチェックします。

#### 再表示時にフォーカスを位置付ける

メニューフォームを再表示するとき、フォーカスを位置付ける場合にチェックします。

#### テキストカラー

テキストの色を指定します。

デフォルトでは「黒色」が設定されています。

#### 背景色

ファイル参照ボタンの背景の色を指定します。

デフォルトでは「灰色」が設定されています。

#### 4. ダイアログの説明

##### フォント名

現在選択されているフォント名を表示します。  
フォントを変更するには、[ フォント ] ボタンを選択します。  
デフォルトでは「MS P ゴシック」が設定されています。

##### スタイル

現在選択されているフォントのスタイルを表示します。  
フォントのスタイルを変更するには、[ フォント ] ボタンを選択します。  
デフォルトでは「標準」が設定されています。

##### サイズ

現在選択されているフォントのサイズを表示します。  
フォントのサイズを変更するには、[ フォント ] ボタンを選択します。  
デフォルトでは「12」が設定されています。

##### 取り消し線

取り消し線の指定の有無を表示します。  
取り消し線を変更するには、[ フォント ] ボタンを選択します。  
デフォルトでは「なし」が設定されています。

##### 下線

下線の指定の有無を表示します。  
下線を変更するには、[ フォント ] ボタンを選択します。  
デフォルトでは「なし」が設定されています。

##### 表示状態を制御する

コントロールの表示状態を制御する場合にチェックします。

##### 変数名

コントロールの表示状態を示す値を格納する変数名を指定します。  
「表示状態を制御する」がチェックされている場合にだけ設定できます。  
この変数名が配列変数の場合、メニューフォーム実行時にエラーになります。  
コントロールの表示状態を示す値は次のとおりです。

##### DISABLE

コントロールはメニューフォーム上に無効状態で表示されます。

##### HIDE

コントロールはメニューフォーム上には表示されません。

##### その他

コントロールはメニューフォーム上に有効状態で表示されます。

#### (2) ダイアログでの操作

- [ OK ] ボタンを選択すると、指定した内容が設定されて [ ファイル参照ボタンプロパティ ] ダイアログが閉じます。
- [ キャンセル ] ボタンを選択すると、内容を変更しないでダイアログを閉じます。



## (3) 処理詳細

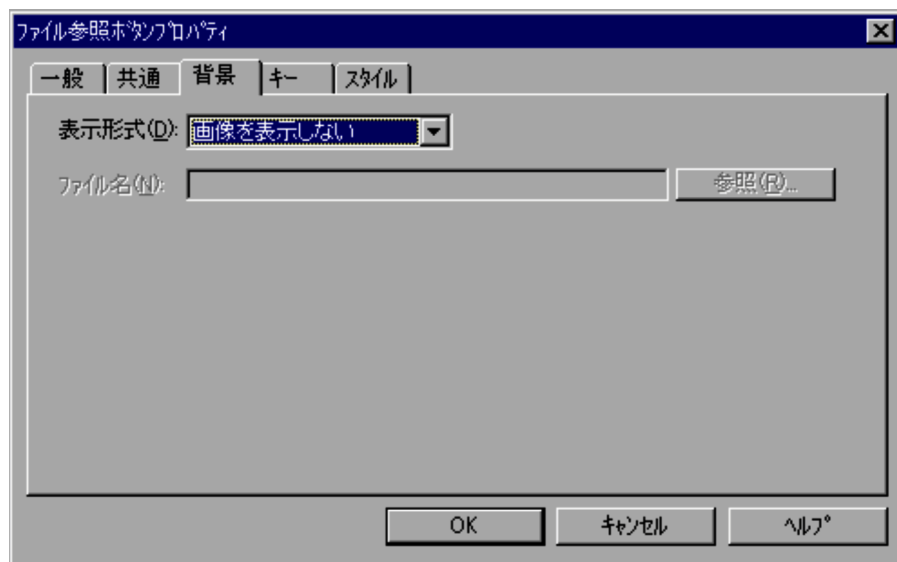
- 「表示状態を制御する」がチェックされていない場合、コントロールはメニューフォーム上に有効状態で表示されます。
- テスト表示では「表示状態を制御する」の指定は無効になります。

## 4.4.17 ファイル参照ボタンプロパティ (背景) ダイアログ

Script メニューエディタウィンドウのメニューフォームビューで、ファイル参照ボタンを選択してダブルクリックすると、[ ファイル参照ボタンプロパティ ] ダイアログが表示されます。

[ ファイル参照ボタンプロパティ ] ダイアログには、「一般」、「共通」、「背景」、「キー」、「スタイル」の五つのタブがあります。

「背景」タブを選択すると、[ ファイル参照ボタンプロパティ (背景) ] ダイアログが表示されます。



## (1) ダイアログの項目

[ ファイル参照ボタンプロパティ (背景) ] ダイアログでは、ファイル参照ボタンの背景について、次の項目を設定できます。

## 表示形式

背景に画像を表示させる方法をドロップダウンリストから選択します。  
デフォルトでは「画像を表示しない」が設定されています。

## ファイル名

背景に表示させる画像のファイル名を指定します。  
ただし、「表示形式」で「画像を表示しない」が選択されている場合は指定できません。

ん。

#### 参照

画像ファイルを指定します。

ただし、「表示形式」で「画像を表示しない」が選択されている場合は指定できません。

#### (2) ダイアログでの操作

- [OK] ボタンを選択すると、指定した内容が設定されて [ ファイル参照ボタンプロパティ ] ダイアログが閉じます。
- [キャンセル] ボタンを選択すると、内容を変更しないでダイアログを閉じます。

### 4.4.18 ファイル参照ボタンプロパティ (キー) ダイアログ

Script メニューエディタウィンドウのメニューフォームビューで、ファイル参照ボタンを選択してダブルクリックすると、[ ファイル参照ボタンプロパティ ] ダイアログが表示されます。

[ ファイル参照ボタンプロパティ ] ダイアログには、「一般」、「共通」、「背景」、「キー」、「スタイル」の五つのタブがあります。

「キー」タブを選択すると、[ ファイル参照ボタンプロパティ (キー) ] ダイアログが表示されます。



#### (1) ダイアログの項目

[ ファイル参照ボタンプロパティ (キー) ] ダイアログでは、ファイル参照ボタンのキーについて、次の項目を設定できます。

**アクセラレータキーを指定する**

アクセラレータキーを指定する場合にチェックします。

**Ctrl**

モデファイキーに Ctrl キーを指定する場合にチェックします。

「アクセラレータキーを指定する」がチェックされている場合にだけチェックできます。

**Shift**

モデファイキーに Shift キーを指定する場合にチェックします。

「アクセラレータキーを指定する」がチェックされている場合にだけチェックできます。

**Alt**

モデファイキーに Alt キーを指定する場合にチェックします。

「アクセラレータキーを指定する」がチェックされている場合にだけチェックできます。

**キー**

指定するキーをドロップダウンリストから選択します。

「アクセラレータキーを指定する」がチェックされている場合にだけ選択できます。

**(2) ダイアログでの操作**

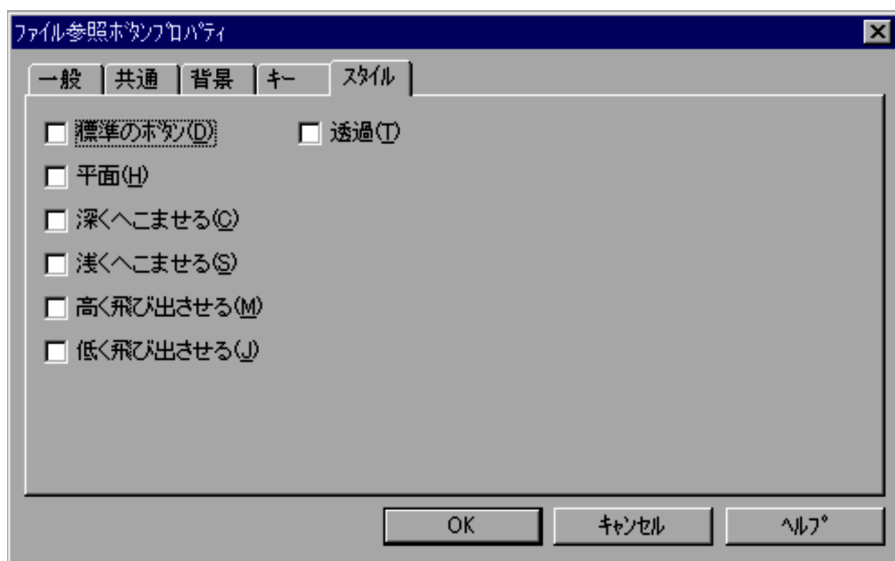
- [ OK ] ボタンを選択すると、指定した内容が設定されて [ ファイル参照ボタンプロパティ ] ダイアログが閉じます。
- [ キャンセル ] ボタンを選択すると、内容を変更しないでダイアログを閉じます。

## 4.4.19 ファイル参照ボタンプロパティ (スタイル) ダイアログ

Script メニューエディタウィンドウのメニューフォームビューで、ファイル参照ボタンを選択してダブルクリックすると、[ ファイル参照ボタンプロパティ ] ダイアログが表示されます。

[ ファイル参照ボタンプロパティ ] ダイアログには、「一般」、「共通」、「背景」、「キー」、「スタイル」の五つのタブがあります。

「スタイル」タブを選択すると、[ ファイル参照ボタンプロパティ (スタイル) ] ダイアログが表示されます。



### (1) ダイアログの項目

[ファイル参照ボタンプロパティ (スタイル)] ダイアログでは、ファイル参照ボタンのキーについて、次の項目を設定できます。

#### 標準のボタン

標準のファイル参照ボタンを指定する場合にチェックします。  
標準ボタンはメニューフォーム上で一つだけ指定できます。

#### 平面

ファイル参照ボタンを 3D 表示しない場合にチェックします。

#### 深くへこませる

ファイル参照ボタンの周囲を深くへこませる場合にチェックします。

#### 浅くへこませる

ファイル参照ボタンの周囲を浅くへこませる場合にチェックします。

#### 高く飛び出させる

ファイル参照ボタンの周囲を高く飛び出させる場合にチェックします。

#### 低く飛び出させる

ファイル参照ボタンの周囲を低く飛び出させる場合にチェックします。

#### 透過

ファイル参照ボタンを透過させる場合にチェックします。

### (2) ダイアログでの操作

- [OK] ボタンを選択すると、指定した内容が設定されて [ファイル参照ボタンプロパ

ティ] ダイアログが閉じます。

- [キャンセル] ボタンを選択すると、内容を変更しないでダイアログを閉じます。

#### 4.4.20 エディットコントロールプロパティ（一般）ダイアログ

Script メニューエディタウィンドウのメニューフォームビューで、エディットコントロールを選択してダブルクリックすると、[エディットコントロールプロパティ] ダイアログが表示されます。

[エディットコントロールプロパティ] ダイアログには、「一般」、「共通」、「キー」、「スタイル」の四つのタブがあります。

「一般」タブを選択すると、[エディットコントロールプロパティ（一般）] ダイアログが表示されます。



##### （１）ダイアログの項目

[エディットコントロールプロパティ（一般）] ダイアログでは、エディットコントロールについて、次の項目を設定できます。

##### テキスト

エディットコントロールに表示させるテキストを指定します。

必ず変数名を指定してください。この変数名が配列変数の場合、メニューフォーム実行時にエラーになります。

##### 入力桁数をチェックする

入力桁数をチェックする場合にチェックします。

#### 4. ダイアログの説明

##### 最大桁数

入力桁数の最大値を指定します。

「入力桁数をチェックする」がチェックされている場合にだけ設定できます。

1 ~ 1,024 の範囲で指定してください。デフォルトでは「10」が設定されています。

##### 入力完了時の動作を指定する

入力完了時の動作をチェックする場合にチェックします。

「入力桁数をチェックする」がチェックされている場合にだけ設定できます。

ただし、「スタイル」タブの「コマンド選択フィールド」と同時に指定することはできません。

##### コマンドを実行する

入力完了時にコマンドを実行する場合に選択します。

「入力完了時の動作を指定する」がチェックされている場合にだけ選択できます。

##### コマンド番号

入力完了時に実行するコマンドのコマンド番号を指定します。

「コマンドを実行する」が選択されている場合にだけ設定できます。

##### コマンド

[ コマンドプロパティの設定 ] ダイアログからコマンドのプロパティを設定します。

「コマンドを実行する」が選択されている場合にだけ設定できます。

##### 他のコントロールにフォーカスを移動する

入力完了時にほかのコントロールにフォーカスを移動させる場合に選択します。

「入力完了時の動作を指定する」がチェックされている場合にだけ選択できます。

##### タブオーダに従う

入力完了時に移動するフォーカスの移動順序をタブオーダに従わせる場合にチェックします。

「他のコントロールにフォーカスを移動する」が選択されている場合にだけチェックできます。

##### 移動先のフィールド名

入力完了時にフォーカスが移動するコントロールのフィールド名を指定します。

メニューフォーム上にあるボタン・ファイル参照ボタン・エディットコントロール・リストコントロール・コンボボックスのフィールド名が一覧表示されます。

「他のコントロールにフォーカスを移動する」が選択され、かつ「タブオーダに従う」が選択されていない場合にだけ設定できます。

#### (2) ダイアログでの操作

- [ OK ] ボタンを選択すると、指定した内容が設定されて [ エディットコントロールプロパティ ] ダイアログが閉じます。
- [ キャンセル ] ボタンを選択すると、内容を変更しないでダイアログを閉じます。

## (3) 処理詳細

- エディットコントロールに入力された半角文字 1 文字は 1 桁、全角文字 1 文字は 2 桁とみなします。
- エディットコントロールに入力できる桁数が残り 1 桁の状態で全角文字を入力した場合、文字の入力は受け付けられないでフォーカスも移動しません。
- 「タブオーダーに従う」がチェックされている場合、入力桁数に達した後フォーカスはタブオーダーに従って移動します。
- 「移動先のフィールド名」に指定されたフィールド名がメニューフォーム上にない場合、フォーカスはタブオーダーに従って移動します。

## 4.4.21 エディットコントロールプロパティ（共通）ダイアログ

Script メニューエディタウィンドウのメニューフォームビューで、エディットコントロールを選択してダブルクリックすると、[エディットコントロールプロパティ] ダイアログが表示されます。

[エディットコントロールプロパティ] ダイアログには、「一般」、「共通」、「キー」、「スタイル」の四つのタブがあります。

「共通」タブを選択すると、[エディットコントロールプロパティ（共通）] ダイアログが表示されます。



## (1) ダイアログの項目

[エディットコントロールプロパティ（共通）] ダイアログでは、エディットコントロールについて、次の項目を設定できます。

#### 4. ダイアログの説明

##### フィールド名

フィールドの名称を指定します。

フィールド名は、変数の名前付け規則に従って指定してください。

##### テキストの配置

「縦方向」は「上寄り」,「横方向」は「左寄り」が設定されています。

##### 起動時にフォーカスを位置付ける

メニューフォームを起動するとき、フォーカスを位置付ける場合にチェックします。

##### 再表示時にフォーカスを位置付ける

メニューフォームを再表示するとき、フォーカスを位置付ける場合にチェックします。

##### テキストカラー

テキストの色を指定します。

デフォルトでは「黒色」が設定されています。

##### 背景色

エディットコントロールの背景の色を指定します。

デフォルトでは「白色」が設定されています。

##### フォント名

現在選択されているフォント名を表示します。

フォントを変更するには、[ フォント ] ボタンを選択します。

デフォルトでは「MS P ゴシック」が設定されています。

##### スタイル

現在選択されているフォントのスタイルを表示します。

フォントのスタイルを変更するには、[ フォント ] ボタンを選択します。

デフォルトでは「標準」が設定されています。

##### サイズ

現在選択されているフォントのサイズを表示します。

フォントのサイズを変更するには、[ フォント ] ボタンを選択します。

デフォルトでは「12」が設定されています。

##### 取り消し線

取り消し線の指定の有無を表示します。

取り消し線を変更するには、[ フォント ] ボタンを選択します。

デフォルトでは「なし」が設定されています。

##### 下線

下線の指定の有無を表示します。

下線を変更するには、[ フォント ] ボタンを選択します。

デフォルトでは「なし」が設定されています。

##### 表示状態を制御する



コントロールの表示状態を制御する場合にチェックします。

#### 変数名

コントロールの表示状態を示す値を格納する変数名を指定します。

「表示状態を制御する」がチェックされている場合にだけ設定できます。

この変数名が配列変数の場合、メニューフォーム実行時にエラーになります。

コントロールの表示状態を示す値は次のとおりです。

#### DISABLE

コントロールはメニューフォーム上に無効状態で表示されます。

#### HIDE

コントロールはメニューフォーム上には表示されません。

#### その他

コントロールはメニューフォーム上に有効状態で表示されます。

### (2) ダイアログでの操作

- [OK] ボタンを選択すると、指定した内容が設定されて [エディットコントロールプロパティ] ダイアログが閉じます。
- [キャンセル] ボタンを選択すると、内容を変更しないでダイアログを閉じます。

### (3) 処理詳細

- 「表示状態を制御する」がチェックされていない場合、コントロールはメニューフォーム上に有効状態で表示されます。
- テスト表示では「表示状態を制御する」の指定は無効になります。

## 4.4.22 エディットコントロールプロパティ (キー) ダイアログ

Script メニューエディタウィンドウのメニューフォームビューで、エディットコントロールを選択してダブルクリックすると、[エディットコントロールプロパティ] ダイアログが表示されます。

[エディットコントロールプロパティ] ダイアログには、「一般」、「共通」、「キー」、「スタイル」の四つのタブがあります。

「キー」タブを選択すると、[エディットコントロールプロパティ (キー)] ダイアログが表示されます。



### (1) ダイアログの項目

[エディットコントロールプロパティ (キー)] ダイアログでは、エディットコントロールのキーについて、次の項目を設定できます。

#### アクセラレータキーを指定する

アクセラレータキーを指定する場合にチェックします。

#### Ctrl

モデファイキーに Ctrl キーを指定する場合にチェックします。

「アクセラレータキーを指定する」がチェックされている場合にだけチェックできます。

#### Shift

モデファイキーに Shift キーを指定する場合にチェックします。

「アクセラレータキーを指定する」がチェックされている場合にだけチェックできます。

#### Alt

モデファイキーに Alt キーを指定する場合にチェックします。

「アクセラレータキーを指定する」がチェックされている場合にだけチェックできます。

#### キー

指定するキーをドロップダウンリストから選択します。

「アクセラレータキーを指定する」がチェックされている場合にだけ選択できます。

## (2) ダイアログでの操作

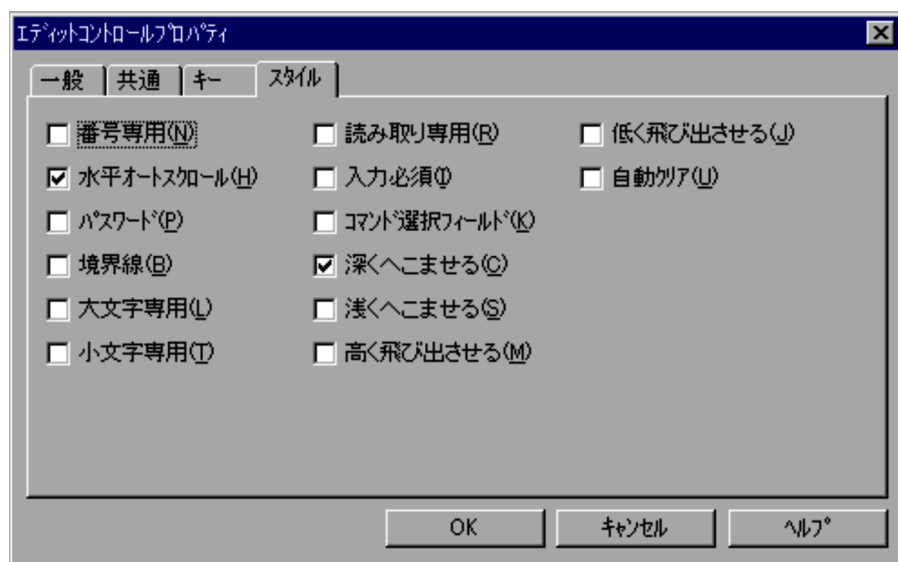
- [OK] ボタンを選択すると、指定した内容が設定されて [エディットコントロールプロパティ] ダイアログが閉じます。
- [キャンセル] ボタンを選択すると、内容を変更しないでダイアログを閉じます。

### 4.4.23 エディットコントロールプロパティ (スタイル) ダイアログ

Script メニューエディタウィンドウのメニューフォームビューで、エディットコントロールを選択してダブルクリックすると、[エディットコントロールプロパティ] ダイアログが表示されます。

[エディットコントロールプロパティ] ダイアログには、「一般」、「共通」、「キー」、「スタイル」の四つのタブがあります。

「スタイル」タブを選択すると、[エディットコントロールプロパティ (スタイル)] ダイアログが表示されます。



## (1) ダイアログの項目

[エディットコントロールプロパティ (スタイル)] ダイアログでは、エディットコントロールについて、次の項目を設定できます。

## 番号専用

エディットコントロールに数字以外の文字の入力を禁止する場合にチェックします。

## 水平オートスクロール

テキストを入力しきれない場合、テキストを自動的にスクロールさせるときに

#### 4. ダイアログの説明

チェックします。  
デフォルトではチェックされています。

##### パスワード

入力するテキストの内容が見えないように、入力するテキストをすべて \*（アスタリスク）で表示させる場合にチェックします。

##### 境界線

エディットコントロールの周囲に境界線を作成する場合にチェックします。

##### 大文字専用

入力されたテキストをすべて大文字に変換して表示する場合にチェックします。

##### 小文字専用

入力されたテキストをすべて小文字に変換して表示する場合にチェックします。

##### 読み取り専用

エディットコントロールに対しての入力や編集を禁止する場合にチェックします。

##### 入力必須

エディットコントロールに必ず入力させる場合にチェックします。この項目をチェックすると、値が入力されなかった場合に警告メッセージを出します。

##### コマンド選択フィールド

コマンド選択フィールドのエディットコントロールの場合にチェックします。

##### 深くへこませる

エディットコントロールの周囲を深くへこませる場合にチェックします。  
デフォルトではチェックされています。

##### 浅くへこませる

エディットコントロールの周囲を浅くへこませる場合にチェックします。

##### 高く飛び出させる

エディットコントロールの周囲を高く飛び出させる場合にチェックします。

##### 低く飛び出させる

エディットコントロールの周囲を低く飛び出させる場合にチェックします。

##### 自動クリア

メニューフォームが表示された時にエディットコントロールを常に空の状態にさせる場合にチェックします。

#### (2) ダイアログでの操作

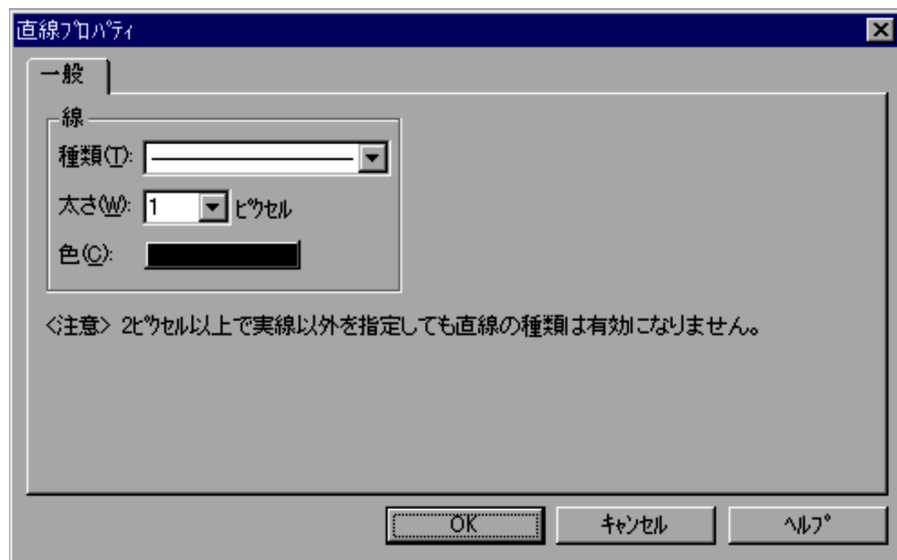
- [OK] ボタンを選択すると、指定した内容が設定されて [エディットコントロールプロパティ] ダイアログが閉じます。
- [キャンセル] ボタンを選択すると、内容を変更しないでダイアログを閉じます。

## (3) 処理詳細

- ・「大文字専用」と「小文字専用」は同時にはチェックできません。
- ・「読み取り専用」と「自動クリア」は同時にはチェックできません。

## 4.4.24 直線プロパティ（一般）ダイアログ

Script メニューエディタウィンドウのメニューフォームビューで、直線を選択してダブルクリックすると、[直線プロパティ（一般）] ダイアログが表示されます。



## (1) ダイアログの項目

[直線プロパティ（一般）] ダイアログでは、直線について、次の項目を設定できます。

## 線

## 種類

直線の種類をドロップダウンリストから選択します。  
デフォルトでは「実線」が設定されています。

## 太さ

直線の太さをピクセル単位で指定します。  
デフォルトでは「1」が設定されています。

## 色

直線の色を指定します。  
デフォルトでは「黒色」が設定されています。

## (2) ダイアログでの操作

- ・[OK] ボタンを選択すると、指定した内容が設定されて [直線プロパティ] ダイアロ

グが閉じます。

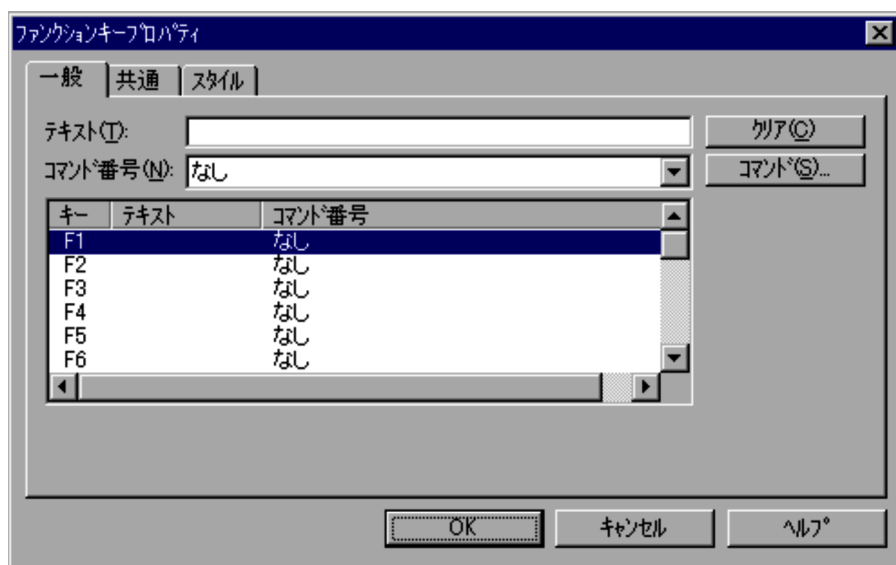
- [キャンセル] ボタンを選択すると、内容を変更しないでダイアログを閉じます。

#### 4.4.25 ファンクションキープロパティ (一般) ダイアログ

Script メニューエディタウィンドウのメニューフォームビューで、ファンクションキーコントロールを選択してダブルクリックすると、[ファンクションキープロパティ] ダイアログが表示されます。

[ファンクションキープロパティ] ダイアログには、「一般」、「共通」、「スタイル」の三つのタブがあります。

「一般」タブを選択すると、[ファンクションキープロパティ (一般)] ダイアログが表示されます。



##### (1) ダイアログの項目

[ファンクションキープロパティ (一般)] ダイアログでは、ファンクションキーコントロールについて、次の項目を設定できます。

###### テキスト

表示するテキストを指定します。

###### コマンド番号

ボタンをクリックしたときに実行されるコマンドの番号を指定します。

###### クリア

リストの中から、選択したデータの内容をクリアします。

コマンド

コマンドのプロパティを設定します。

## (2) ダイアログでの操作

- [OK] ボタンを選択すると、指定した内容が設定されて [ ファンクションキープロパティ ] ダイアログが閉じます。
- [キャンセル] ボタンを選択すると、内容を変更しないでダイアログを閉じます。

### 4.4.26 ファンクションキープロパティ ( 共通 ) ダイアログ

Script メニューエディタウィンドウのメニューフォームビューで、ファンクションキーコントロールを選択してダブルクリックすると、[ ファンクションキープロパティ ] ダイアログが表示されます。

[ ファンクションキープロパティ ] ダイアログには、「一般」、「共通」、「スタイル」の三つのタブがあります。

「共通」タブを選択すると、[ ファンクションキープロパティ ( 共通 ) ] ダイアログが表示されます。



## (1) ダイアログの項目

[ ファンクションキープロパティ ( 共通 ) ] ダイアログでは、ファンクションキーコントロールについて、次の項目を設定できます。

フィールド名

「FunctionKey」が設定されています。

テキストの配置

#### 4. ダイアログの説明

##### 縦方向・横方向

テキストを配置するときの縦方向と横方向の位置をドロップダウンリストから選択します。

デフォルトでは、「縦方向」「横方向」とも、「中央」が設定されています。

##### テキストカラー

テキストの色を指定します。

デフォルトでは「黒色」が設定されています。

##### 背景色

ファンクションキーコントロールの背景の色を指定します。

デフォルトでは「灰色」が設定されています。

##### フォント名

現在選択されているフォント名を表示します。

フォントを変更するには、「[ フォント ]」ボタンを選択します。

デフォルトでは「MS P ゴシック」が設定されています。

##### スタイル

現在選択されているフォントのスタイルを表示します。

フォントのスタイルを変更するには、「[ フォント ]」ボタンを選択します。

デフォルトでは「標準」が設定されています。

##### サイズ

現在選択されているフォントのサイズを表示します。

フォントのサイズを変更するには、「[ フォント ]」ボタンを選択します。

デフォルトでは「12」が設定されています。

##### 取り消し線

取り消し線の指定の有無を表示します。

取り消し線を変更するには、「[ フォント ]」ボタンを選択します。

デフォルトでは「なし」が設定されています。

##### 下線

下線の指定の有無を表示します。

下線を変更するには、「[ フォント ]」ボタンを選択します。

デフォルトでは「なし」が設定されています。

##### 表示状態を制御する

コントロールの表示状態を制御する場合にチェックします。

##### 変数名

コントロールの表示状態を示す値を格納する変数名を指定します。

「表示状態を制御する」がチェックされている場合にだけ設定できます。

この変数名が配列変数の場合、メニューフォーム実行時にエラーになります。

コントロールの表示状態を示す値は次のとおりです。



## DISABLE

コントロールはメニューフォーム上に無効状態で表示されます。

## HIDE

コントロールはメニューフォーム上には表示されません。

## その他

コントロールはメニューフォーム上に有効状態で表示されます。

**!** 注意事項

JP1/Script 06-71 以降では、[ 起動時にフォーカスを位置付ける ] は無効状態で表示されるようになりました。

## (2) ダイアログでの操作

- [ OK ] ボタンを選択すると、指定した内容が設定されて [ ファンクションキープロパティ ] ダイアログが閉じます。
- [ キャンセル ] ボタンを選択すると、内容を変更しないでダイアログを閉じます。

## (3) 処理詳細

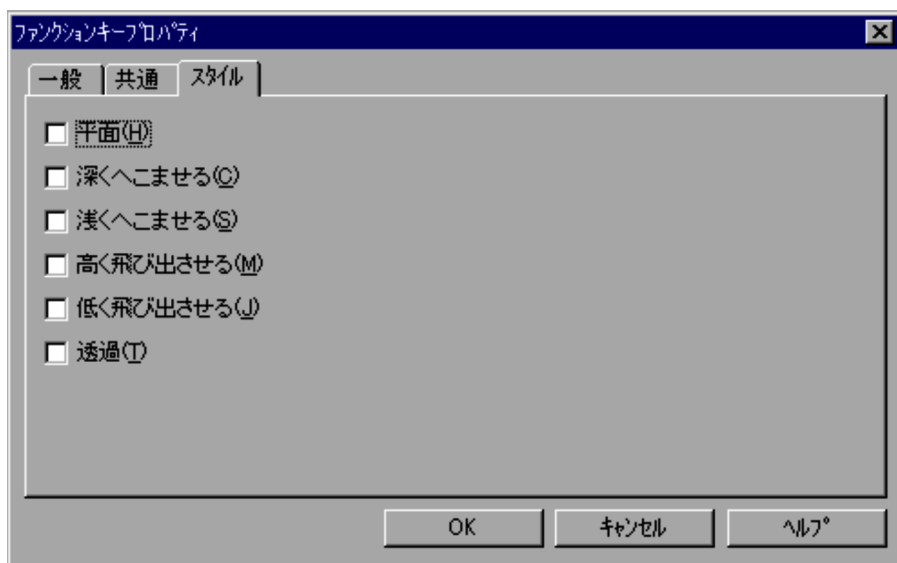
- 「表示状態を制御する」がチェックされていない場合、コントロールはメニューフォーム上に有効状態で表示されます。
- テスト表示では「表示状態を制御する」の指定は無効になります。

### 4.4.27 ファンクションキープロパティ (スタイル) ダイアログ

Script メニューエディタウィンドウのメニューフォームビューで、ファンクションキーコントロールを選択してダブルクリックすると、[ ファンクションキープロパティ ] ダイアログが表示されます。

[ ファンクションキープロパティ ] ダイアログには、「一般」、「共通」、「スタイル」の三つのタブがあります。

「スタイル」タブを選択すると、[ ファンクションキープロパティ (スタイル) ] ダイアログが表示されます。



### (1) ダイアログの項目

[ファンクションキープロパティ (スタイル)] ダイアログでは、ファンクションキーコントロールについて、次の項目を設定できます。

#### 平面

ボタンを 3D 表示させない場合にチェックします。

#### 深くへこませる

コントロールの周囲を深くへこませる場合にチェックします。

#### 浅くへこませる

コントロールの周囲を浅くへこませる場合にチェックします。

#### 高く飛び出させる

コントロールの周囲を高く飛び出させる場合にチェックします。

#### 低く飛び出させる

コントロールの周囲を低く飛び出させる場合にチェックします。

#### 透過

コントロールを透過させる場合にチェックします。

### (2) ダイアログでの操作

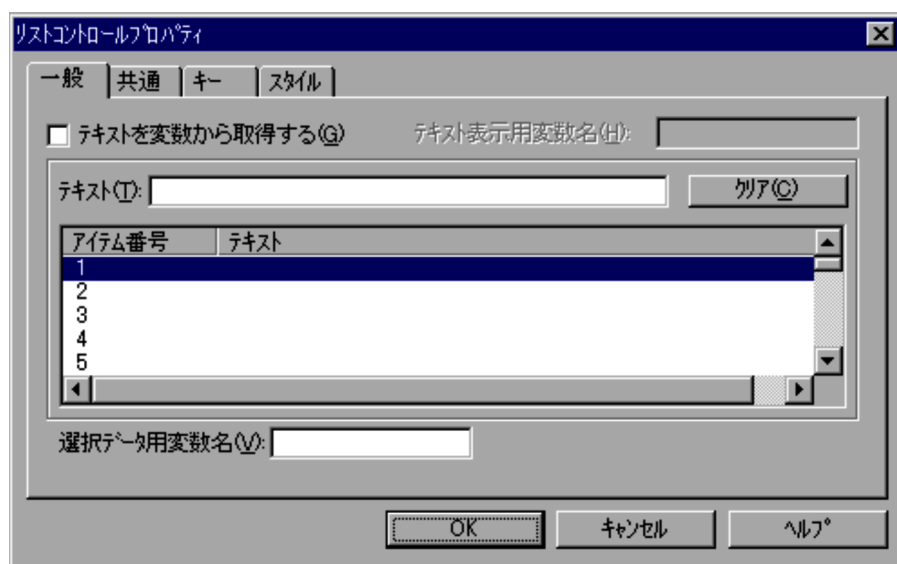
- [OK] ボタンを選択すると、指定した内容が設定されて [ファンクションキープロパティ] ダイアログが閉じます。
- [キャンセル] ボタンを選択すると、内容を変更しないでダイアログを閉じます。

#### 4.4.28 リストコントロールプロパティ（一般）ダイアログ

Script メニューエディタのメニューフォームビューで、リストコントロールを選択してダブルクリックすると、[リストコントロールプロパティ] ダイアログが表示されます。

[リストコントロールプロパティ] ダイアログには、「一般」、「共通」、「キー」、「スタイル」の四つのタブがあります。

「一般」タブを選択すると、[リストコントロールプロパティ（一般）] ダイアログが表示されます。



##### (1) ダイアログの項目

[リストコントロールプロパティ（一般）] ダイアログでは、リストコントロールについて、次の項目を設定できます。

##### テキストを変数から取得する

リストに表示させるテキストを変数から取得する場合にチェックします。

##### テキスト表示用変数名

リストに表示させるテキストを設定する変数名を指定します。

「テキストを変数から取得する」がチェックされている場合にだけ設定できます。この変数名が2次元の配列変数の場合、メニューフォーム実行時にエラーになります。

##### テキスト

リストに表示させるテキストを指定します。

ただし、「テキストを変数から取得する」がチェックされている場合は設定できません。

#### 4. ダイアログの説明

##### クリア

データをリストから削除します。

ただし、「テキストを変数から取得する」がチェックされている場合は削除できません。

##### 選択データ用変数名

リストのステータスを設定する変数名を指定します。

この変数名が次の場合、メニューフォーム実行時にエラーになります。

- 「テキストを変数から取得する」がチェックされないで、かつこの変数名が配列変数の場合。
- 「テキストを変数から取得する」がチェックされ、複数選択ができないリストで、かつ、この変数名が配列変数の場合。
- 「テキストを変数から取得する」がチェックされ、複数選択できるリストで、かつこの変数名が2次元の配列変数の場合。

#### (2) ダイアログでの操作

- [OK] ボタンを選択すると、指定した内容が設定されて [ リストコントロールプロパティ ] ダイアログが閉じます。
- [キャンセル] ボタンを選択すると、内容を変更しないでダイアログを閉じます。

#### (3) 処理詳細

- 「テキストを変数から取得する」がチェックされている場合、「テキスト表示用変数名」の要素で値が格納されていない要素はリストに表示されません。
- 「選択データ用変数名」に格納される値は次のようになります。

「テキストを変数から取得する」がチェックされている場合：

選択されたデータの「テキスト表示用変数名」でのインデックス番号が格納される。

「テキストを変数から取得する」がチェックされていない場合：

複数選択不可能なリストの場合：アイテム番号が格納される。

複数選択可能なリストの場合："001...100" 形式の数字列が格納される。

### 4.4.29 リストコントロールプロパティ ( 共通 ) ダイアログ

Script メニューエディタウィンドウのメニューフォームビューで、リストコントロールを選択してダブルクリックすると、[ リストコントロールプロパティ ] ダイアログが表示されます。

[ リストコントロールプロパティ ] ダイアログには、「一般」、「共通」、「キー」、「スタイル」の四つのタブがあります。

「共通」タブを選択すると、[ リストコントロールプロパティ ( 共通 ) ] ダイアログが表示されます。



### (1) ダイアログの項目

[リストコントロールプロパティ (共通)] ダイアログでは、リストコントロールについて、次の項目を設定できます。

#### フィールド名

フィールドの名称を指定します。

フィールド名は、変数の名前付け規則に従って指定してください。

#### テキストの配置

##### 縦方向・横方向

テキストを配置するときの縦方向と横方向の位置をドロップダウンリストから選択します。

デフォルトでは「縦方向」は「中央」、「横方向」は「左寄り」が設定されています。

#### 起動時にフォーカスを位置付ける

メニューフォームを起動するとき、フォーカスを位置付ける場合にチェックします。

#### 再表示時にフォーカスを位置付ける

メニューフォームを再表示するとき、フォーカスを位置付ける場合にチェックします。

#### テキストカラー

テキストの色を指定します。

デフォルトでは「黒色」が設定されています。

#### 背景色

リストコントロールの背景の色を指定します。

#### 4. ダイアログの説明

デフォルトでは「白色」が設定されています。

##### フォント名

現在選択されているフォント名を表示します。

フォントを変更するには、[ フォント ] ボタンを選択します。

デフォルトでは「MS P ゴシック」が設定されています。

##### スタイル

現在選択されているフォントのスタイルを表示します。

フォントのスタイルを変更するには、[ フォント ] ボタンを選択します。

デフォルトでは「標準」が設定されています。

##### サイズ

現在選択されているフォントのサイズを表示します。

フォントのサイズを変更するには、[ フォント ] ボタンを選択します。

デフォルトでは「12」が設定されています。

##### 取り消し線

取り消し線の指定の有無を表示します。

取り消し線を変更するには、[ フォント ] ボタンを選択します。

デフォルトでは「なし」が設定されています。

##### 下線

下線の指定の有無を表示します。

下線を変更するには、[ フォント ] ボタンを選択します。

デフォルトでは「なし」が設定されています。

##### 表示状態を制御する

コントロールの表示状態を制御する場合にチェックします。

##### 変数名

コントロールの表示状態を示す値を格納する変数名を指定します。

「表示状態を制御する」がチェックされている場合にだけ設定できます。

この変数名が配列変数の場合、メニューフォーム実行時にエラーになります。

コントロールの表示状態を示す値は次のとおりです。

##### DISABLE

コントロールはメニューフォーム上に無効状態で表示されます。

##### HIDE

コントロールはメニューフォーム上には表示されません。

##### その他

コントロールはメニューフォーム上に有効状態で表示されます。

#### (2) ダイアログでの操作

- [ OK ] ボタンを選択すると、指定した内容が設定されて [ リストコントロールプロパティ ] ダイアログが閉じます。

- [キャンセル] ボタンを選択すると、内容を変更しないでダイアログを閉じます。

### (3) 処理詳細

- 「表示状態を制御する」がチェックされていない場合、コントロールはメニューフォーム上に有効状態で表示されます。
- テスト表示では「表示状態を制御する」の指定は無効になります。

## 4.4.30 リストコントロールプロパティ (キー) ダイアログ

Script メニューエディタウィンドウのメニューフォームビューで、リストコントロールを選択してダブルクリックすると、[リストコントロールプロパティ] ダイアログが表示されます。

[リストコントロールプロパティ] ダイアログには、「一般」、「共通」、「キー」、「スタイル」の四つのタブがあります。

「キー」タブを選択すると、[リストコントロールプロパティ (キー)] ダイアログが表示されます。



### (1) ダイアログの項目

[リストコントロールプロパティ (キー)] ダイアログでは、リストコントロールについて、次の項目を設定できます。

#### アクセラレータキーを指定する

アクセラレータキーを指定する場合にチェックします。

#### Ctrl

モデファイキーに Ctrl キーを指定する場合にチェックします。

「アクセラレータキーを指定する」がチェックされている場合にだけチェックできます。

##### Shift

モデファイキーに Shift キーを指定する場合にチェックします。

「アクセラレータキーを指定する」がチェックされている場合にだけチェックできます。

##### Alt

モデファイキーに Alt キーを指定する場合にチェックします。

「アクセラレータキーを指定する」がチェックされている場合にだけチェックできます。

##### キー

指定するキーをドロップダウンリストから選択します。

「アクセラレータキーを指定する」がチェックされている場合にだけ選択できます。

#### (2) ダイアログでの操作

- [ OK ] ボタンを選択すると、指定した内容が設定されて [ リストコントロールプロパティ ] ダイアログが閉じます。
- [ キャンセル ] ボタンを選択すると、内容を変更しないでダイアログを閉じます。

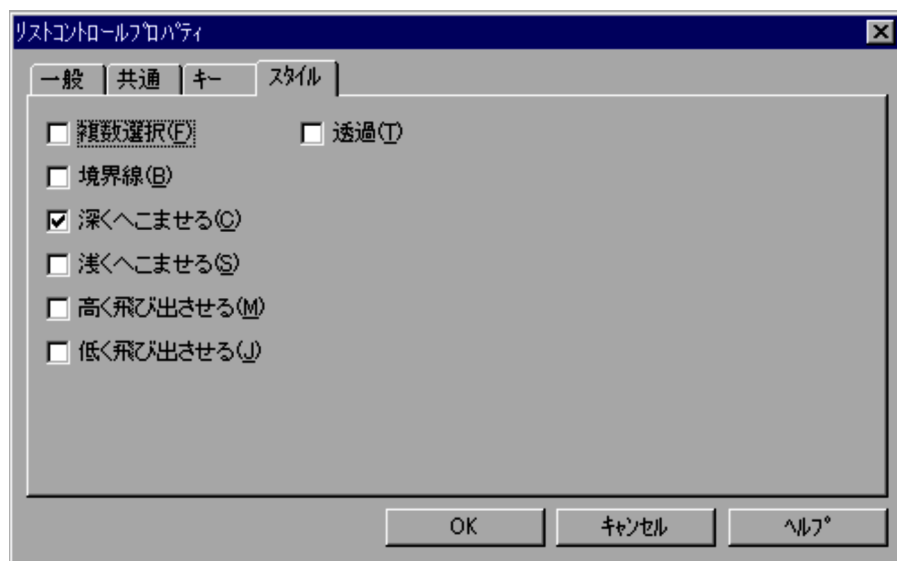
### 4.4.31 リストコントロールプロパティ (スタイル) ダイアログ

Script メニューエディタウィンドウのメニューフォームビューで、リストコントロールを選択してダブルクリックすると、[ リストコントロールプロパティ ] ダイアログが表示されます。

[ リストコントロールプロパティ ] ダイアログには、「一般」、「共通」、「キー」、「スタイル」の四つのタブがあります。

「スタイル」タブを選択すると、[ リストコントロールプロパティ (スタイル) ] ダイアログが表示されます。





### (1) ダイアログの項目

[リストコントロールプロパティ (スタイル)] ダイアログでは、リストコントロールについて、次の項目を設定できます。

#### 複数選択

複数のデータを選択する場合にチェックします。

#### 境界線

リストコントロールの周囲に境界線を作成する場合にチェックします。

#### 深くへこませる

リストコントロールの周囲を深くへこませる場合にチェックします。  
デフォルトではチェックされています。

#### 浅くへこませる

リストコントロールの周囲を浅くへこませる場合にチェックします。

#### 高く飛び出させる

リストコントロールの周囲を高く飛び出させる場合にチェックします。

#### 低く飛び出させる

リストコントロールの周囲を低く飛び出させる場合にチェックします。

#### 透過

リストコントロールを透過させる場合にチェックします。

### (2) ダイアログでの操作

- ・ [OK] ボタンを選択すると、指定した内容が設定されて [リストコントロールプロパ

ティ] ダイアログが閉じます。

- [キャンセル] ボタンを選択すると、内容を変更しないでダイアログを閉じます。

## 4.4.32 コンボボックスプロパティ (一般) ダイアログ

Script メニューエディタウィンドウのメニューフォームビューで、コンボボックスを選択してダブルクリックすると、[コンボボックスプロパティ] ダイアログが表示されます。

[コンボボックスプロパティ] ダイアログには、「一般」、「共通」、「キー」、「スタイル」の四つのタブがあります。

「一般」タブを選択すると、[コンボボックスプロパティ (一般)] ダイアログが表示されます。



### (1) ダイアログの項目

[コンボボックスプロパティ (一般)] ダイアログでは、コンボボックスについて、次の項目を設定できます。

テキストを変数から取得する

コンボボックスに表示させるテキストを変数から取得する場合にチェックします。

テキスト表示用変数名

コンボボックスに表示させるテキストを設定する変数名を指定します。

「テキストを変数から取得する」がチェックされている場合にだけ設定できます。

この変数名が2次元の配列変数の場合、メニューフォーム実行時にエラーになります。

#### テキスト

コンボボックスに表示させるテキストを指定します。

ただし、「テキストを変数から取得する」がチェックされている場合は設定できません。

#### リストされるデータ

コンボボックスのリストに表示されるデータの一覧です。

ただし、「テキストを変数から取得する」がチェックされている場合は有効になりません。

#### 追加

「リストされるデータ」の末尾にデータを追加します。

ただし、「テキストを変数から取得する」がチェックされている場合は有効になりません。

#### 挿入

「リストされるデータ」にデータを挿入します。

ただし、「テキストを変数から取得する」がチェックされている場合は有効になりません。

#### 更新

「リストされるデータ」のデータを更新します。

ただし、「テキストを変数から取得する」がチェックされている場合は有効になりません。

#### 削除

「リストされるデータ」の中で選択されているデータを削除します。

ただし、「テキストを変数から取得する」がチェックされている場合は有効になりません。

#### 上に移動

「リストされるデータ」の中で選択されているデータを上に移動します。

ただし、「テキストを変数から取得する」がチェックされている場合は有効になりません。

#### 下に移動

「リストされるデータ」の中で選択されているデータを下に移動します。

ただし、「テキストを変数から取得する」がチェックされている場合は有効になりません。

#### 選択データ用変数名

コンボボックスのステータスを設定する変数名を指定します。

この変数名が配列変数の場合、メニューフォーム実行時にエラーになります。

#### ドロップダウンされる行数

コンボボックスをドロップダウンしたときに表示されるデータの数を指定します。

1 ~ 100 の範囲で指定してください。デフォルトでは「5」が設定されています。

### (2) ダイアログでの操作

- [OK] ボタンを選択すると、指定した内容が設定されて [ コンボボックスプロパティ ] ダイアログが閉じます。
- [キャンセル] ボタンを選択すると、内容を変更しないでダイアログを閉じます。

### (3) 処理詳細

- 「テキストを変数から取得する」がチェックされている場合、「テキスト表示用変数名」の要素で値が格納されていない要素はドロップダウンリストに表示されません。
- 「選択データ用変数名」にはコンボボックスで選択、または入力された文字列が格納されます。

## 4.4.33 コンボボックスプロパティ ( 共通 ) ダイアログ

Script メニューエディタウィンドウのメニューフォームビューで、コンボボックスを選択してダブルクリックすると、[ コンボボックスプロパティ ] ダイアログが表示されます。

[ コンボボックスプロパティ ] ダイアログには、「一般」、「共通」、「キー」、「スタイル」の四つのタブがあります。

「共通」タブを選択すると、[ コンボボックスプロパティ ( 共通 ) ] ダイアログが表示されます。



### (1) ダイアログの項目

[ コンボボックスプロパティ ( 共通 ) ] ダイアログでは、コンボボックスについて、次の項目を設定できます。

#### フィールド名

フィールドの名称を指定します。

フィールド名は、変数の名前付け規則に従って指定してください。

#### テキストの配置

「縦方向」は「中央」,「横方向」は「左寄り」が設定されています。

#### 起動時にフォーカスを位置付ける

メニューフォームを起動するとき、フォーカスを位置付ける場合にチェックします。

#### 再表示時にフォーカスを位置付ける

メニューフォームを再表示するとき、フォーカスを位置付ける場合にチェックします。

#### テキストカラー

テキストの色を指定します。

デフォルトでは「黒色」が設定されています。

#### 背景色

コンボボックスの背景の色を指定します。

デフォルトでは「白色」が設定されています。

#### フォント名

現在選択されているフォント名を表示します。

フォントを変更するには、[ フォント ] ボタンを選択します。

デフォルトでは「MS P ゴシック」が設定されています。

#### スタイル

現在選択されているフォントのスタイルを表示します。

フォントのスタイルを変更するには、[ フォント ] ボタンを選択します。

デフォルトでは「標準」が設定されています。

#### サイズ

現在選択されているフォントのサイズを表示します。

フォントのサイズを変更するには、[ フォント ] ボタンを選択します。

デフォルトでは「12」が設定されています。

#### 取り消し線

取り消し線の指定の有無を表示します。

取り消し線を変更するには、[ フォント ] ボタンを選択します。

デフォルトでは「なし」が設定されています。

#### 下線

下線の指定の有無を表示します。

下線を変更するには、[ フォント ] ボタンを選択します。

デフォルトでは「なし」が設定されています。

#### 表示状態を制御する

コントロールの表示状態を制御する場合にチェックします。

##### 変数名

コントロールの表示状態を示す値を格納する変数名を指定します。

「表示状態を制御する」がチェックされている場合にだけ設定できます。

この変数名が配列変数の場合、メニューフォーム実行時にエラーになります。

コントロールの表示状態を示す値は次のとおりです。

##### DISABLE

コントロールはメニューフォーム上に無効状態で表示されます。

##### HIDE

コントロールはメニューフォーム上には表示されません。

##### その他

コントロールはメニューフォーム上に有効状態で表示されます。

#### (2) ダイアログでの操作

- [OK] ボタンを選択すると、指定した内容が設定されて [ コンボボックスプロパティ ] ダイアログが閉じます。
- [キャンセル] ボタンを選択すると、内容を変更しないでダイアログを閉じます。

#### (3) 処理詳細

- 「表示状態を制御する」がチェックされていない場合、コントロールはメニューフォーム上に有効状態で表示されます。
- テスト表示では「表示状態を制御する」の指定は無効になります。

### 4.4.34 コンボボックスプロパティ (キー) ダイアログ

Script メニューエディタウィンドウのメニューフォームビューで、コンボボックスを選択してダブルクリックすると、[ コンボボックスプロパティ ] ダイアログが表示されます。

[ コンボボックスプロパティ ] ダイアログには、「一般」、「共通」、「キー」、「スタイル」の四つのタブがあります。

「キー」タブを選択すると、[ コンボボックスプロパティ (キー) ] ダイアログが表示されます。



### (1) ダイアログの項目

[コンボボックスプロパティ (キー)] ダイアログでは、コンボボックスについて、次の項目を設定できます。

#### アクセラレータキーを指定する

アクセラレータキーを指定する場合にチェックします。

#### Ctrl

モデファイキーに Ctrl キーを指定する場合にチェックします。

「アクセラレータキーを指定する」がチェックされている場合にだけチェックできます。

#### Shift

モデファイキーに Shift キーを指定する場合にチェックします。

「アクセラレータキーを指定する」がチェックされている場合にだけチェックできます。

#### Alt

モデファイキーに Alt キーを指定する場合にチェックします。

「アクセラレータキーを指定する」がチェックされている場合にだけチェックできます。

#### キー

指定するキーをドロップダウンリストから選択します。

「アクセラレータキーを指定する」がチェックされている場合にだけ選択できます。

## (2) ダイアログでの操作

- [OK] ボタンを選択すると、指定した内容が設定されて [コンボボックスプロパティ] ダイアログが閉じます。
- [キャンセル] ボタンを選択すると、内容を変更しないでダイアログを閉じます。

### 4.4.35 コンボボックスプロパティ (スタイル) ダイアログ

Script メニューエディタウィンドウのメニューフォームビューで、コンボボックスを選択してダブルクリックすると、[コンボボックスプロパティ] ダイアログが表示されます。

[コンボボックスプロパティ] ダイアログには、「一般」、「共通」、「キー」、「スタイル」の四つのタブがあります。

「スタイル」タブを選択すると、[コンボボックスプロパティ (スタイル)] ダイアログが表示されます。



## (1) ダイアログの項目

[コンボボックスプロパティ (スタイル)] ダイアログでは、コンボボックスについて、次の項目を設定できます。

### 編集可能

編集可能なコンボボックスを作成する場合にチェックします。

### 水平オートスクロール

テキストを入力しきれない場合、テキストを自動的にスクロールさせるときにチェックします。



**大文字専用**

入力されたテキストをすべて大文字に変換して表示させる場合にチェックします。

**小文字専用**

入力されたテキストをすべて小文字に変換して表示させる場合にチェックします。

**入力必須**

コンボボックスに必ず入力させる場合にチェックします。

**入力データをチェックする**

コンボボックスのリストの一覧にないデータが入力されたときにエラーにする場合にチェックします。

「編集可能」がチェックされている場合にだけチェックできます。

**深くへこませる**

コンボボックスの周囲を深くへこませる場合にチェックします。

**浅くへこませる**

コンボボックスの周囲を浅くへこませる場合にチェックします。

**高く飛び出させる**

コンボボックスの周囲を高く飛び出させる場合にチェックします。

**低く飛び出させる**

コンボボックスの周囲を低く飛び出させる場合にチェックします。

**入力桁数をチェックする**

入力桁数をチェックする場合にチェックします。

**最大桁数**

入力桁数の最大値を指定します。

「入力桁数をチェックする」がチェックされている場合にだけ設定できます。

1 ~ 1,024 の範囲で指定してください。デフォルトでは「10」が設定されています。

**(2) ダイアログでの操作**

- ・[OK] ボタンを選択すると、指定した内容が設定されて [コンボボックスプロパティ] ダイアログが閉じます。
- ・[キャンセル] ボタンを選択すると、内容を変更しないでダイアログを閉じます。

**(3) 処理詳細**

- ・「大文字専用」と「小文字専用」は、同時にはチェックできません。
- ・「入力データをチェックする」で指定されたデータチェックでは、大文字と小文字を区別しません。

**4.4.36 コマンドプロパティダイアログ**

ボタンコントロールをクリックしたときのコマンドの実行について、定義するダイアロ

#### 4. ダイアログの説明

グです。

Script メニューエディタウィンドウで、[ 表示 ] - [ コマンドプロパティ ] を選択すると、[ コマンドプロパティ ] ダイアログの表示 / 非表示を切り替えられます。

デフォルトではコマンド番号 1 には「実行内容」は「何もしない」、「メニューフォームの状態」は「メニューフォームをクローズする」が、コマンド番号 2 には「実行内容」は「何もしない」、「メニューフォームの状態」は「直前のメニューフォームに戻る」が設定されています。



##### (1) ダイアログの項目

[ コマンドプロパティ ] ダイアログでは、コマンドの実行について、次の項目を設定できます。

###### コマンド一覧

[ コマンドプロパティの設定 ] ダイアログで設定されたコマンドの一覧が表示されます。

一覧には次の項目が表示されます。

###### 番号

コマンドの番号が表示されます。

###### 実行内容

コマンドの実行内容が表示されます。

###### メニューフォームの状態

コマンドを実行した後のメニューフォームの状態が表示されます。

###### 実行対象

コマンドの実行対象が表示されます。

###### 実行フォルダ

コマンドの実行フォルダが表示されます。

###### 閉じる

コマンドプロパティダイアログを閉じます。

### クリア

選択したコマンド番号の、プロパティ定義の内容をクリアします。

### 設定

コマンドのプロパティを定義したいときにクリックします。

[ コマンドプロパティの設定 ] ダイアログが表示されます。

## (2) ダイアログでの操作

- ・[ 閉じる ] ボタンをクリックすると、[ コマンドプロパティ ] ダイアログが閉じます。

## 4.4.37 コマンドプロパティの設定ダイアログ

ボタンコントロールやファンクションキーコントロールをクリックしたときのコマンドのプロパティを定義するダイアログです。

[ コマンドプロパティの設定 ] ダイアログは、[ ツール ] - [ コマンドプロパティの設定 ] を選択すると表示されます。

## (1) ダイアログの項目

[ コマンドプロパティの設定 ] ダイアログでは、コマンドについて、次の項目を設定できます。

#### 4. ダイアログの説明

##### コマンド番号

コマンドの番号を指定します。

##### ボタン押下時の動作

###### 実行内容・メニューフォームの状態

コマンドの実行内容と、コマンドを実行した後の実行状態をドロップダウンリストから選択します。

デフォルトでは、「実行内容」は「外部コマンドを実行する」、「メニューの状態」は「メニューをクローズする」が設定されています。

「ボタン押下時の動作」の指定については、「(3) ボタン押下時の動作」を参照してください。

##### 実行するファイルパス

実行する対象のファイルパスを指定します。

ファイルパスは「実行内容」が「外部コマンドを実行する」の場合にだけ指定できます。

##### 変数名

「実行するファイルパス」を変数名で指定する場合にチェックします。

「実行するファイルパス」は変数名で指定してください。この変数名が配列変数の場合、メニューフォーム実行時にエラーになります。

##### 参照

「実行するファイルパス」を、[ ファイルを開く ] ダイアログから指定します。

##### パラメータ

実行対象へ渡すパラメータを指定します。

なお、パラメータに次の文字列を指定した場合、起動するアプリケーション側でのウィンドウ表示時のデフォルトとして扱われます。明示的にウィンドウを表示するアプリケーションには有効になりません。

パラメータ	意味
"/SPT:HIDE"	アプリケーションのウィンドウを非表示にします。
"/SPT:MIN"	アプリケーションのウィンドウをアイコン化します。
"/SPT:MAX"	アプリケーションのウィンドウを最大表示します。

パラメータは「実行内容」が「外部コマンドを実行する」の場合にだけ指定できます。

##### 変数名

「パラメータ」を変数名で指定する場合にチェックします。

「パラメータ」は変数名で指定してください。この変数名が配列変数の場合、メニューフォーム実行時にエラーになります。

また、変数は一つしか指定できません。変数として二つ以上のパラメータを渡したい場合は、一つの変数に値を複数設定してください(値をスペースで区切る)。

**実行フォルダ**

コマンドの実行フォルダを指定します。

実行フォルダは「実行内容」が「外部コマンドを実行する」の場合にだけ指定できます。

「実行フォルダ」を省略すると、現在のフォルダが内部的に仮定されます。

**変数名**

「実行フォルダ」を変数名で指定する場合にチェックします。

「実行フォルダ」は変数名で指定してください。この変数名が配列変数の場合、メニューフォーム実行時にエラーになります。

**参照**

[ フォルダの選択 ] ダイアログから「実行フォルダ」を指定します。

**実行するメニューフォーム名**

実行するメニューフォームの名称を指定します。

メニューフォームの名称は「実行内容」が「ファイル内のメニューフォームを表示する」の場合にだけ指定できます。

**実行するサブルーチン名**

実行するサブルーチンの名称を指定します。

サブルーチンの名称は「実行内容」が「ファイル内のサブルーチンを呼び出す」の場合にだけ指定できます。

**(2) ダイアログでの操作**

- ・[ OK ] ボタンを選択すると、指定した内容が設定されて [ コマンドプロパティの設定 ] ダイアログが閉じます。
- ・[ キャンセル ] ボタンを選択すると、内容を変更しないでダイアログを閉じます。

**(3) ボタン押下時の動作**

[ コマンドプロパティの設定 ] ダイアログで設定する「ボタン押下時の動作」の「実行内容」と「メニューフォームの状態」の指定の組み合わせと指定時の処理動作を次に示します。

**(a) 実行内容「外部コマンドを実行する」の場合**

メニューフォームの状態	指定	処理動作
「メニューフォームをクローズする」		外部コマンドを実行し、終了を待たないでメニューフォームを終了する。
「キャンセルして 直前のメニューフォームに戻る」	×	
「直前のメニューフォームに戻る」		外部コマンドを実行し、終了を待たないで直前のメニューフォームに戻る。
「表示したまま終了を待つ」		外部コマンドを実行し、メニューフォームを使用不能な状態で表示して終了を待つ。

#### 4. ダイアログの説明

メニューフォームの状態	指定	処理動作
「非表示にして終了を待つ」		外部コマンドを実行し、メニューフォームを非表示にして終了を待つ。
「最小化して待つ」		外部コマンドを実行し、メニューフォームを最小化して終了を待つ。
「アイドル状態にする」		外部コマンドを実行するだけで、終了待ちはしない。

#### (b) 実行内容「ファイル内のメニューフォームを表示する」の場合

メニューフォームの状態	指定	処理動作
「メニューフォームをクローズする」	×	
「キャンセルして直前のメニューフォームに戻る」	×	
「直前のメニューフォームに戻る」	×	
「表示したまま終了を待つ」		ファイル内の新しいメニューフォームを表示し、メニューフォームを使用不能な状態で表示して終了を待つ。
「非表示にして終了を待つ」		ファイル内の新しいメニューフォームを表示し、メニューフォームを非表示にして終了を待つ。
「最小化して待つ」	×	
「アイドル状態にする」	×	

#### (c) 実行内容「ファイル内のサブルーチンを呼び出す」の場合

メニューフォームの状態	指定	処理動作
「メニューフォームをクローズする」		ファイル内のサブルーチンを呼び出し、メニューフォームを使用不能な状態で表示して終了を待つ。サブルーチンが Function ステートメントでその戻り値が True の場合はメニューフォームを終了し、True 以外の場合は何もしない。
「キャンセルして直前のメニューフォームに戻る」	×	
「直前のメニューフォームに戻る」		ファイル内のサブルーチンを呼び出し、メニューフォームを使用不能な状態で表示して終了を待つ。サブルーチンが Function ステートメントでその戻り値が True の場合は直前のメニューフォームに戻り、True 以外の場合は何もしない。
「表示したまま終了を待つ」		ファイル内のサブルーチンを呼び出し、メニューフォームを使用不能な状態で表示して終了を待つ。
「非表示にして終了を待つ」		ファイル内のサブルーチンを呼び出し、メニューフォームを非表示にして終了を待つ。

メニューフォームの状態	指定	処理動作
「最小化して待つ」	×	
「アイドル状態にする」	×	

## (d) 実行内容「ファンクションキーを切り替える」の場合

メニューフォームの状態	指定	処理動作
「メニューフォームをクローズする」	×	
「キャンセルして 直前のメニューフォームに戻る」	×	
「直前のメニューフォームに戻る」	×	
「表示したまま終了を待つ」	×	
「非表示にして終了を待つ」	×	
「最小化して待つ」	×	
「アイドル状態にする」	×	

## (e) 実行内容「何もしない」の場合

メニューフォームの状態	指定	処理動作
「メニューフォームをクローズする」		メニューフォームを終了する。
「キャンセルして 直前のメニューフォームに戻る」		入力されたデータのチェック、および変数への 値の代入をしないで、直前のメニューフォーム に戻る。
「直前のメニューフォームに戻る」		直前のメニューフォームに戻る。
「表示したまま終了を待つ」	×	
「非表示にして終了を待つ」	×	
「最小化して待つ」		最小化する。
「アイドル状態にする」		何もしない。

## (凡例)

: 組み合わせ可能な指定

×: 組み合わせ不可能な指定

## (4) 処理詳細

- ・「実行内容」を「外部コマンドを実行する」に設定している場合、予約変数には次の値が格納されます。

メニューフォームの状態	予約変数	格納される値
「アイドル状態にする」	_EXEC_RTN_	なし
	_EXEC_ID_	実行ファイルの識別子
「アイドル状態にする」以外	_EXEC_RTN_	外部コマンドの終了コード

#### 4. ダイアログの説明

- Menu コマンドから実行された外部コマンドの終了コードに 0 以外が返された場合、メッセージボックスを表示します。メッセージボックスを表示しない場合は、次に示すレジストリの値に 0 を設定してください。

〔レジストリキー〕

HKEY\_LOCAL\_MACHINE¥Software¥Hitachi¥JP1/Script¥SPTX

〔値名〕

Menu\_EmgMsgBox

〔値のデータタイプ〕

DWORD

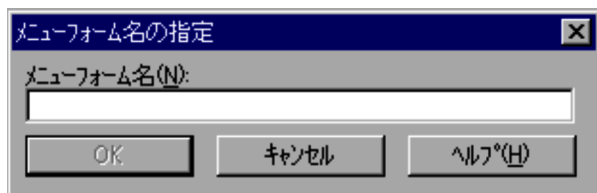
〔値〕

- 0：メッセージボックスを表示しない
- 1：メッセージボックスを表示する（初期値）

#### 4.4.38 メニューフォーム名の指定ダイアログ

新しいメニューフォームを作成するときに、メニューフォーム名を指定するダイアログです。

〔メニューフォーム名の指定〕ダイアログは、[編集] - [メニューフォームの新規作成] を選択すると表示されます。



##### （１）ダイアログの項目

〔メニューフォーム名の指定〕ダイアログでは、次の項目を設定できます。

メニューフォーム名

メニューフォーム名を指定します。

##### （２）ダイアログでの操作

- [OK] ボタンを選択すると、指定した内容が設定されて [メニューフォーム名の指定] ダイアログが閉じます。
- [キャンセル] ボタンを選択すると、内容を変更しないでダイアログを閉じます。

#### 4.4.39 一括変更ダイアログ

メニューフォーム上のコントロールの属性を一括して変更するダイアログです。



[一括変更] ダイアログは、[編集] - [一括変更] を選択すると表示されます。



### (1) ダイアログの項目

[一括変更] ダイアログでは、次の項目を設定できます。

#### テキストの配置

テキストの配置を一括変更する場合にチェックします。

ただし、選択されたコントロールにエディットコントロール、またはコンボボックスが含まれる場合にはグレーアウトされてチェックできません。

#### 縦方向・横方向

テキストの縦方向と横方向の位置をドロップダウンリストから選択します。

「テキストの配置」がチェックされている場合にだけ選択できます。

#### テキストカラー

テキストの色を一括変更する場合にチェックし、色の設定ダイアログからテキストの色を指定します。

#### 背景色

背景の色を一括変更する場合にチェックし、色の設定ダイアログから背景の色を指定します。

#### 背景

背景の情報を一括変更する場合にチェックし、選択したコントロールのプロパティ (背景) ダイアログで背景の情報を指定します。

ただし、選択されたコントロールにスタティック、ボタン、またはファイル参照ボタン以外のコントロールが含まれる場合にはグレーアウトされてチェックできません。

#### 4. ダイアログの説明

##### スタイル

スタイルを一括変更する場合にチェックし、選択したコントロールのプロパティ（スタイル）ダイアログでスタイルの情報を指定します。

選択されたコントロールがすべて同じ種類の場合にだけチェックできます。

ただし、選択されたコントロールの種類がボタン、またはファイル参照ボタンの場合には「標準のボタン」の設定内容は変更できません。

また、選択されたコントロールの種類がエディットコントロールの場合には「コマンド選択フィールド」の設定内容は変更できません。

##### フォント

フォントを一括変更する場合にチェックし、フォントダイアログからフォントの情報を指定します。

##### フォント名

現在選択されているフォント名を表示します。

##### スタイル

現在選択されているフォントのスタイルを表示します。

##### サイズ

現在選択されているフォントのサイズを表示します。

##### 取り消し線

取り消し線の指定の有無を表示します。

##### 下線

下線の指定の有無を表示します。

#### （２）ダイアログでの操作

- ・ [ OK ] ボタンを選択すると、指定した内容が設定されて [ 一括変更 ] ダイアログが閉じます。
- ・ [ キャンセル ] ボタンを選択すると、内容を変更しないでダイアログを閉じます。

#### （３）処理詳細

- ・ 各項目の初期値には最初に選択されたコントロールの内容が設定されます。
- ・ すべての項目がチェックされていない場合には、[ OK ] ボタンはグレースアウトされて表示されます。

### 4.4.40 グリッドの設定ダイアログ

メニューフォーム上でグリッドを表示させる場合に、グリッドの間隔を設定するダイアログです。

[ グリッドの設定 ] ダイアログは、[ 編集 ] - [ レイアウト ] - [ グリッドの設定 ] を選択すると表示されます。



### (1) ダイアログの項目

[ グリッドの設定 ] ダイアログでは、次の項目を設定できます。

グリッドを使用する

グリッドを使用する場合にチェックします。

グリッドの間隔

幅・高さ

グリッドの幅と高さをピクセル単位で指定します。

1 ~ 100 の範囲で指定してください。デフォルトでは「8」が指定されています。

マージン

マージンをピクセル単位で設定します。

0 ~ 100 の範囲で指定してください。デフォルトでは「10」が指定されています。

### (2) ダイアログでの操作

- [ OK ] ボタンを選択すると、指定した内容が設定されて [ グリッドの設定 ] ダイアログが閉じます。
- [ キャンセル ] ボタンを選択すると、内容を変更しないでダイアログを閉じます。

## 4.4.41 タブオーダの設定ダイアログ

タブの移動順序を設定するダイアログです。

[ タブオーダの設定 ] ダイアログは、[ 編集 ] - [ レイアウト ] - [ タブオーダの設定 ] を選択すると表示されます。



### (1) ダイアログの項目

[ タブオーダーの設定 ] ダイアログでは、次の項目を設定できます。

#### フィールド名

メニューフォーム上にあるフィールドのフィールド名が表示されます。  
表示されているフィールド名の上から順番にタブが移動します。

#### 上に移動

選択したフィールドのタブの移動順序が、一つ上に設定されます。

#### 下に移動

選択したフィールドのタブの移動順序が、一つ下に設定されます。

#### 座標位置でソート

タブの移動順序が、メニューフォーム上にあるフィールドの座標位置の順序に設定されます。

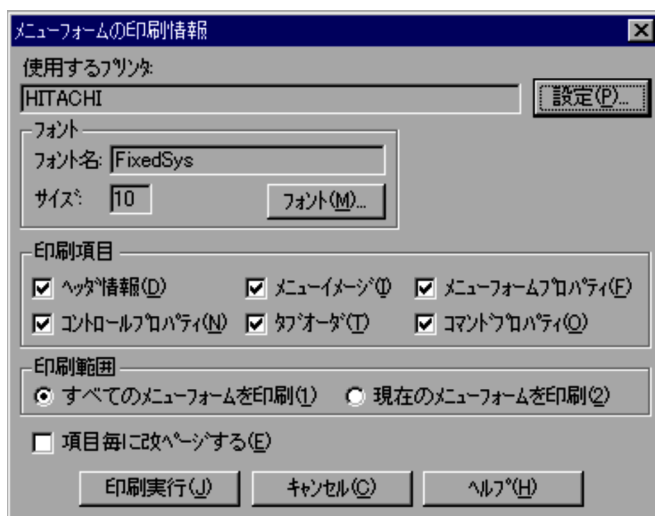
### (2) ダイアログでの操作

- [ OK ] ボタンを選択すると、指定した内容が設定されて [ タブオーダーの設定 ] ダイアログが閉じます。
- [ キャンセル ] ボタンを選択すると、内容を変更しないでダイアログを閉じます。

## 4.4.42 メニューフォームの印刷情報ダイアログ

作成したメニューフォームの内容の印刷情報を設定するダイアログです。

[ メニューフォームの印刷情報 ] ダイアログは、[ ファイル ] - [ メニューフォームの印刷 ] を選択すると表示されます。



### (1) ダイアログの項目

[メニューフォームの印刷情報] ダイアログでは、次の項目を設定できます。

#### 使用するプリンタ

使用するプリンタ名が表示されます。

プリンタを変更するには、[設定] ボタンを選択します。

デフォルトでは「デフォルトプリンタ」が設定されています。

#### フォント名

印刷に使用するフォント名が表示されます。

フォントを変更するには、[フォント] ボタンを選択します。

デフォルトでは「FixedSys」が設定されています。

#### サイズ

印刷に使用するフォントのサイズが表示されます。

フォントのサイズを変更するには、[フォント] ボタンを選択します。

デフォルトでは「10」が設定されています。

#### 印刷項目

##### ヘッダ情報

ヘッダ情報を印刷する場合にチェックします。

印刷される内容は次のとおりです。

- メニュー情報ファイル名
- メニューフォーム名
- 更新日付 (yyyy/mm/dd/ HH:MM:SS の形式 )
- 印刷日付 (yyyy/mm/dd/ HH:MM:SS の形式 )
- ページ

#### 4. ダイアログの説明

yyyy: 西暦, mm: 月, dd: 日, HH: 時, MM: 分, SS: 秒を意味します。

##### メニューイメージ

メニューフォームのテスト表示イメージを印刷する場合にチェックします。  
用紙の横幅中央に印刷します。用紙に収まらない場合は、縮小して印刷します。

##### メニューフォームプロパティ

メニューフォームのプロパティを印刷する場合にチェックします。

##### コントロールプロパティ

コントロールのプロパティを印刷する場合にチェックします。

##### タブオーダ

タブオーダを印刷する場合にチェックします。

##### コマンドプロパティ

コマンドプロパティを印刷する場合にチェックします。

デフォルトではすべての項目がチェックされています。

##### 印刷範囲

###### すべてのメニューフォームを印刷

すべてのメニューフォームの内容を印刷する場合に選択します。

###### 現在のメニューフォームを印刷

現在、選択されているメニューフォームの内容を印刷する場合に選択します。  
メニューフォームが選択されていない場合は「現在のメニューフォームを印刷」は  
グレーアウトされて選択できません。  
デフォルトでは「すべてのメニューフォームを印刷」が選択されています。

##### 項目毎に改ページする

「印刷項目」で選択した項目ごとに改ページをする場合にチェックします。  
デフォルトではチェックされていません。

#### (2) ダイアログでの操作

- [印刷実行] ボタンを選択すると、指定した内容が設定されて印刷を実行します。印刷終了後に [メニューフォームの印刷情報] ダイアログを閉じます。
- [キャンセル] ボタンを選択すると、内容を変更しないでダイアログを閉じます。

#### (3) 処理詳細

- メニューフォームがない場合には、[ファイル] - [メニューフォームの印刷] はグレーアウトされて選択できません。

# 5

## トラブルへの対処方法

この章では、JP1/Script で発生するトラブルの種類と、その対処方法について説明します。

---

5.1 対処の手順

---

5.2 トラブルシューティング

---

5.3 ログ情報

---

5.4 トラブル発生時に採取が必要な資料

---

5.5 資料の採取方法

---

5.6 バックアップとリカバリー

---

## 5.1 対処の手順

---

JP1/Script でトラブルが発生した場合の対処の手順を次に示します。

### 現象の確認

トラブルが発生したときの現象を確認してください。メッセージが出力されている場合は、メッセージの内容を確認してください。

また、JP1/Script 使用時のトラブルシューティングについては、「5.2 トラブルシューティング」を参照してください。JP1/Script が出力するログ情報については、「5.3 ログ情報」を参照してください。

### 資料の採取

トラブルの要因を調べるために資料の採取が必要です。「5.4 トラブル発生時に採取が必要な資料」、および「5.5 資料の採取方法」を参照して必要な資料を採取してください。

### 問題の調査

採取した資料を基に問題の要因を調査し、問題が発生している部分、または問題の範囲を切り分けてください。



## 5.2 トラブルシューティング

ここでは、JP1/Script 使用時のトラブルシューティングについて説明します。JP1/Script を使用しているときにトラブルが発生した場合、まず、この節で説明している現象が発生していないか確認してください。

### 5.2.1 スクリプト実行時のトラブルと対処方法

スクリプト実行時のトラブルと対処方法を表 5-1 に示します。

表 5-1 スクリプト実行時のトラブルと対処方法

トラブルの種類	要因	対処方法
スクリプト実行中にトレース情報を出力する領域やコマンドを処理する領域がボリューム上に確保できない。	ボリューム上に十分な空き領域がない。	JP1/Script のマネージャ、またはエクスプローラなどで不要なファイルを削除して、空き領域を確保する。
		トレースファイルの出力先や作業フォルダに別のボリュームを指定する。
スクリプトを実行すると「アプリケーションを正しく初期化できませんでした。」や終了コード「99」でスクリプトの実行が失敗する。	スクリプト実行時に必要なメモリが不足している。	メモリを増設する、またはページファイルのサイズを大きくする。ページファイルのサイズを大きくしてもメモリ不足が発生する場合は、他のアプリケーションを終了させて、空き容量を増やす。
		サービスから実行している場合、サービスアカウントを他のサービスで使用していないアカウントに変更する。
スクリプトを実行すると「ファイルが壊れているか、またはファイルの形式が異なります」となる。	指定したファイルに誤りがある。	指定したファイルに誤りがないか確認する。
	前回の実行中に異常終了している場合は、ファイルが正しく保存されていない可能性がある。	JP1/Script のマネージャ、またはエクスプローラなどでファイルを削除する。
スクリプトファイルが実行中のままとなる。	JP1/AJS2 などのサービスプログラムから起動して、実行スクリプト内でメッセージボックスやウィンドウなどの対話操作が必要な処理を動作させている。	JP1/AJS2 などのサービスプログラムから起動して対話操作をする場合は、NetExec コマンドを使ってログオン空間でスクリプトを実行する。
	Exec コマンド、または NetExec コマンドで呼び出したプログラムがループしている。	実行プログラムに誤りがないか調査する。
	トレース管理ファイル (SPTLOGDB.SPB) が破壊されている。	エクスプローラなどでトレース管理ファイルを削除する。

## 5. トラブルへの対処方法

トラブルの種類	要因	対処方法
ユーザプログラムからスクリプトを実行すると終了コード「19」となる。	スクリプトで文法エラーが発生している。	JP1/Script のマネージャ、または Script エディタから実行して、エラー内容を確認する。
		スクリプトエンジンのバージョンが古いバージョンになっていないか確認する。
スクリプトの実行がエラーになる。	スクリプトファイルの実行中にエラーを検出した。	JP1/Script のトレースビューアを起動してエラーの内容を確認する。
スクリプトの実行速度が遅くなる。	同時に実行するスクリプトの数が多過ぎる。	同時に実行するスクリプトの数を減らす。
	トレース管理ファイル (SPTLOGDB.SPB) の容量が肥大化している。	スクリプトが動作していない状態でトレース管理ファイル (SPTLOGDB.SPB) を削除して再実行する。Message コマンドでユニークなファイル名称を持ったユーザトレースファイルを大量に作成する場合、Message コマンドから TextOpen/TextWrite/TextClose コマンドによるトレースファイルの作成を検討する。
スクリプト実行が、メモリ不足で異常終了 (例：スクリプト実行が終了コード「20」で異常終了する) したり、コマンド実行がメモリ不足でエラー (例：Copy コマンドで「メモリ不足が発生しました」となる) になる。	トレース管理ファイル (SPTLOGDB.SPB) の容量が肥大化している。	スクリプトが動作していない状態でトレース管理ファイル (SPTLOGDB.SPB) を削除して再実行する。Message コマンドでユニークなファイル名称を持ったユーザトレースファイルを大量に作成する場合、Message コマンドから TextOpen/TextWrite/TextClose コマンドによるトレースファイルの作成を検討する。

### 5.2.2 コマンド実行時のトラブルと対処方法

コマンド実行時のトラブルと対処方法を表 5-2 に示します。

表 5-2 コマンド実行時のトラブルと対処方法

トラブルの種類	要因	対処方法
NetExec コマンドを実行すると「サーバが接続要求に対して応答しません。」となる。	接続先コンピュータに JP1/Script がインストールされていない。	接続先コンピュータに JP1/Script をインストールして、Windows を再起動する。
	ログオン空間で実行している場合、接続先コンピュータ上で Script ランチャが起動されていない。	Script ランチャを起動する。
	サービス空間で実行している場合、接続先コンピュータ上で JP1/Script サービスが起動されていない。	JP1/Script サービスを起動する。
	接続先コンピュータに論理ホスト名（論理 IP アドレス）を指定し、論理ホスト上でフェールオーバーが発生した。	接続先コンピュータ論理ホスト名（論理 IP アドレス）ではなく、物理ホスト名（物理 IP アドレス）を指定する。 接続先コンピュータに論理ホスト名を指定する場合は、クラスタに JP1/Script を登録する。

## 5. トラブルへの対処方法

トラブルの種類	要因	対処方法
NetExec コマンドを実行すると「ユーザ名を認識できないか、またはパスワードが誤っています。」となる。	JP1/Script のマネージャ、単体起動、または起動のタイプが「ログオン」のスクリプト自動起動でスクリプトを実行している場合、Windows にログオンしたユーザアカウントが接続先コンピュータ上に存在しない。または、Guest アカウントが無効になっている。	接続先コンピュータで Guest アカウントが無効になっていないか確認する。  Windows にログオンしたユーザアカウントが接続先コンピュータ上に存在しているか確認する。
	起動のタイプが「サービス」のスクリプト自動起動から実行している場合、JP1/Script サービスに設定したユーザアカウントが接続先コンピュータ上に存在しない。または、Guest アカウントが無効になっている。	接続先コンピュータで Guest アカウントが無効になっていないか確認する。  JP1/Script サービスに設定したユーザアカウントが接続先コンピュータ上に存在しているか確認する。
	JP1/AJS2 のサービスプログラムから実行している場合、スクリプトを実行する JP1 ユーザにマッピングされた OS ユーザのアカウントが接続先コンピュータ上に存在しない。または、Guest アカウントが無効になっている。	接続先コンピュータで Guest アカウントが無効になっていないか確認する。  スクリプトを実行する JP1 ユーザにマッピングされた OS ユーザのアカウントが接続先コンピュータ上に存在しているかを確認する。
	JP1/AJS2 以外のサービスプログラムから実行している場合、各サービスアカウントに設定したユーザアカウントが接続先コンピュータ上に存在しない。または、Guest アカウントが無効になっている。	接続先コンピュータで Guest アカウントが無効になっていないか確認する。  各プログラムのサービスに設定したユーザアカウントが接続先コンピュータ上に存在しているか確認する。
SetGV, GetGV, DeleteGV, または NetExec コマンドを実行すると「このコマンドの実行は許可されていません。」となる。	接続先コンピュータで各コマンドのクライアントでの実行を許可しない設定になっている。	接続先コンピュータの JP1/Script のマネージャで各コマンドのクライアントでの実行を許可する設定になっているか確認する。
NetExec コマンドを実行すると「アクセスは拒否されました。ファイルの属性、またはセキュリティを見直してください。」となる。	スクリプトの自動起動、または上位などのサービスプログラムから実行している場合、サービスアカウントにローカルシステムアカウントを設定している。	スクリプトの自動起動、または上位などのサービスプログラムから実行している場合、そのサービスアカウントに接続先コンピュータ上に存在しているユーザアカウントを設定する。

トラブルの種類	要因	対処方法
Exec, NetExec コマンドを実行後、予約変数 _EXEC_RTN_ に予測しないコードが設定される。	起動した実行ファイルに異常が発生した。	起動した実行ファイルを単体で実行して異常が無いか、実行ファイルが _EXEC_RTN_ に設定されたリターンコードを返さないか確認する。 実行ファイルでアプリケーションエラーが発生した場合、-1073741819 (16進 C0000005) などの値が設定される。

### 5.2.3 トレースビューア実行時のトラブルと対処方法

トレースビューア実行時のトラブルと対処方法を表 5-3 に示します。

表 5-3 トレースビューア実行時のトラブルと対処方法

トラブルの種類	要因	対処方法
トレースビューアで他コンピュータに接続すると「トレースビューアサーバが起動されていません。または、応答がありません。」となる。	接続先コンピュータに JP1/Script がインストールされていない。	接続先コンピュータに JP1/Script をインストールして、Windows を再起動する。
	JP1/Script サービスが起動されていない。	JP1/Script サービスを起動する。
	接続先コンピュータに論理ホスト名 (論理 IP アドレス) を指定し、論理ホスト上でフェールオーバーが発生した。	接続先コンピュータに論理ホスト名 (論理 IP アドレス) ではなく、物理ホスト名 (物理 IP アドレス) を指定する。 接続先コンピュータに論理ホスト名を指定する場合は、クラスタに JP1/Script を登録する。

### 5.2.4 プロセスビューア実行時のトラブルと対処方法

プロセスビューア実行時のトラブルと対処方法を表 5-4 に示します。

## 5. トラブルへの対処方法

表 5-4 プロセスビューア実行時のトラブルと対処方法

トラブルの種類	要因	対処方法
プロセスビューアで他コンピュータに接続すると「サーバが接続要求に対して応答しません。」となる。	接続先コンピュータに JP1/Script がインストールされていない。	接続先コンピュータに JP1/Script をインストールして、Windows を再起動する。
	JP1/Script サービスが起動されていない。	JP1/Script サービスを起動する。
	接続先コンピュータに論理ホスト名（論理 IP アドレス）を指定し、論理ホスト上でフェールオーバーが発生した。	接続先コンピュータに論理ホスト名（論理 IP アドレス）ではなく、物理ホスト名（物理 IP アドレス）を指定する。
		接続先コンピュータに論理ホスト名を指定する場合は、クラスタに JP1/Script を登録する。
プロセスビューアで他コンピュータに接続すると「サーバが接続要求に対して拒否の応答をしました。」となる。	接続先コンピュータでプロセスビューアを受け付けない設定になっている。	接続先コンピュータの JP1/Script のマネージャでプロセスビューアを受け付ける設定になっているか確認する。

## 5.3 ログ情報

---

JP1/Script でトラブルが発生した場合、ログ情報を確認して対処方法を検討します。

JP1/Script で出力されるログ情報には次の 6 種類があります。

解析トレースファイル <sup>1</sup>

実行トレースファイル <sup>2</sup>

ユーザトレースファイル <sup>3</sup>

サーバトレースファイル <sup>4</sup>

NetExec エラーログファイル <sup>5</sup>

イベントログ <sup>6</sup>

注 1 解析トレースファイルを出力する場合だけ

注 2 実行トレースファイルを出力する場合だけ

注 3 Message コマンドで第一引数に Target\_File を指定している場合だけ

注 4 SetGV, GetGV, deleteGV, NetExec コマンドで他コンピュータを指定した場合だけ

注 5 NetExec コマンドを実行した場合だけ

注 6 イベントログを出力する場合だけ

ここでは、6 種類のログ情報について説明します。

### 5.3.1 ログ情報の種類

#### (1) 解析トレースファイル

解析トレースファイルとは、コマンドの解析結果を出力するログ情報のことです。

設定方法については、「4.1.12 コマンドラインの設定ダイアログ」、および「6.2 コマンドラインに関する規則」を参照してください。また、出力形式については、「付録 A Script トレースのファイル出力形式」を参照してください。

#### (2) 実行トレースファイル

実行トレースファイルとは、コマンドの実行結果を出力するログ情報のことです。実行トレースファイルには出力レベルがあり、出力レベルによって出力されるログ情報が異なります。設定方法については、「4.1.12 コマンドラインの設定ダイアログ」、および「6.2 コマンドラインに関する規則」を参照してください。また、出力形式については、「付録 A Script トレースのファイル出力形式」を参照してください。

### (3) ユーザトレースファイル

ユーザトレースファイルとは、Message コマンドの第一引数に Target\_File を指定して出力するログ情報のことです。設定方法については、「8.6.2 Message (ファイル/ウィンドウにテキストを出力する)」を参照してください。また、出力形式については、「付録 A Script トレースのファイル出力形式」を参照してください。

### (4) サーバトレースファイル

サーバトレースファイルとは、SetGV、GetGV、DeleteGV、および NetExec コマンドで他コンピュータ名を指定した場合、サーバ上でコマンドを実行した結果を出力するログ情報のことです。設定方法については、「4.1.22 オプション (サーバ情報) ダイアログ」を参照してください。また、出力形式については、「付録 A Script トレースのファイル出力形式」を参照してください。

### (5) NetExec エラーログファイル

NetExec エラーログファイルとは、NetExec コマンド実行時に発生したエラーを出力するログ情報のことです。NetExec エラーログファイルについては、「付録 A Script トレースのファイル出力形式」を参照してください。

### (6) イベントログ

イベントログとは、システムの状態やトラブルを通知するログ情報のことです。Windows のイベントビューアで参照することができます。設定方法については、「4.1.12 コマンドラインの設定ダイアログ」、および「6.2 コマンドラインに関する規則」を参照してください。

## 5.3.2 ログファイルおよびディレクトリ一覧

ここでは、JP1/Script から出力されるログ情報について、ログの出力元、ログファイル名、およびディスク使用量を表 5-5 に示します。

表 5-5 ログの出力元、ログファイル名、およびディスク使用量

ログ情報の種類	ログファイル名	ディスク 使用量 (KB)
解析トレースファイル	スクリプト実行フォルダ <sup>1</sup> ¥ スクリプトファイル名 .SPA	154 <sup>2</sup>
実行トレースファイル	スクリプト実行フォルダ <sup>1</sup> ¥ スクリプトファイル名 .SPX	154 <sup>2</sup>
ユーザトレースファイル	Message コマンドの OutputName に指定したファイルパス	154 <sup>2</sup>
サーバトレースファイル	インストール先フォルダ ¥Data <sup>3</sup> ¥SPTSVTRC.SPY	204



ログ情報の種類	ログファイル名	ディスク 使用量 (KB)
NetExec エラーログファイル	インストール先フォルダ ¥Log¥STXNetExec_Client¥ <sup>4</sup> STXNetExec_C_xx <sup>5</sup> .log	10 <sup>6</sup>
	インストール先フォルダ ¥Log¥STXNetExec_Server¥ <sup>4</sup> STXNetExec_S_xx <sup>5</sup> .log	10 <sup>6</sup>

## 注 1

出力先フォルダは変更できます。設定方法については、「4.1.10 実行環境の設定（トレース情報）ダイアログ」、または「4.1.25 オプション（複数起動）ダイアログ」を参照してください。

## 注 2

デフォルト値です。ディスク使用量については「1.4.2 ファイルの容量」を参照してください。

## 注 3

出力先フォルダは変更できます。設定方法については、「4.1.28 オプション（クラスタ環境）ダイアログ」を参照してください。

## 注 4

出力先フォルダは変更できます。設定方法については、「8.10.2 NetExec（自 PC / 他 PC 上の実行ファイルを呼び出す）」を参照してください。なお、STXNetExec\_Client フォルダは NetExec コマンドを実行したクライアント側、STXNetExec\_Server フォルダは NetExec コマンドで指定したサーバコンピュータ側に作成されます。

## 注 5

xx は 01 ~ 30 までの数値。

## 注 6

一つのファイルの上限値。ファイルは最大 30 ファイル作成されるので、最大で約 300KB を必要とします。

## 5.4 トラブル発生時に採取が必要な資料

「5.2 トラブルシューティング」に示した対処をしてもトラブルを解決できなかった場合、トラブルの要因を調べるための資料を採取し、システム管理者に連絡する必要があります。この節では、トラブル発生時に採取が必要な資料について説明します。

### 5.4.1 OS の情報

トラブル発生時に採取が必要な OS の情報を表 5-6 に示します。

表 5-6 トラブル発生時に採取が必要な OS の情報

情報の種類	概要	デフォルトのファイル名	採取方法
イベントログ (アプリケーションログ)	Windows イベント ログ	-	イベントビューアからログファイル保存
イベントログ (システムログ)			
OS 情報	システム 情報	-	「5.5.2 OS システム情報の採取」を参照のこと
ダンプ情報 <sup>1</sup>	ワトソン ログファ イル	<ul style="list-style-type: none"> <li>Windows NT の場合 システムドライブ¥システムインストールディレクトリ ¥drwtsn32.log <sup>2</sup></li> <li>Windows 2000 の場合 システムドライブ ¥Documents and Settings¥All Users¥Documents¥DrWatson¥drwtsn32.log <sup>2</sup> システムドライブ ¥Documents and Settings¥All Users¥Documents¥DrWatson¥user.dump <sup>2</sup></li> <li>Windows XP, または Windows Server 2003 の場合 システムドライブ ¥Documents and Settings¥All Users¥Application Data¥Microsoft¥Dr Watson¥drwtsn32.log <sup>2</sup> システムドライブ ¥Documents and Settings¥All Users¥Application Data¥Microsoft¥Dr Watson¥user.dump <sup>2</sup></li> </ul>	-
	プロセス ダンプ <sup>3</sup>	-	Microsoft(R) サポート技術情報を参照してください。

## 注 1

アプリケーションエラーが発生した場合に採取してください。

## 注 2

別のフォルダにログファイルが出力されるように設定している場合は、該当するフォルダから資料を採取してください。

## 注 3

スクリプトプロセス（SPTXE.EXE）が実行中のままとなった場合に採取してください。

## 5.4.2 JP1/Script の情報

JP1/Script に関する次の情報の採取が必要です。また、ネットワーク接続でのトラブルの場合、接続先マシン上のファイルの採取も必要です。

トラブル発生時に採取が必要な JP1/Script の情報を表 5-7 に示します。

表 5-7 トラブル発生時に採取が必要な JP1/Script の情報

情報の種類	概要	格納場所
スクリプトファイル	スクリプト実行時のスクリプトソース	スクリプト実行フォルダ ¥ スクリプトファイル名 .SPT
スクリプト実行環境ファイル	スクリプト実行時の環境が設定されているファイル	スクリプト実行フォルダ ¥ スクリプトファイル名 .SPV
モニタリング情報ファイル	モニタリング実行時の情報が設定されているファイル	スクリプト実行フォルダ ¥ スクリプトファイル名 .SPD
メニューファイル	Menu コマンド実行時のメニューファイル	メニューファイル名 .SPN
解析トレースファイル	スクリプト実行時の解析トレースファイル	スクリプト実行フォルダ ¥ スクリプトファイル名 .SPA
実行トレースファイル	スクリプト実行時の実行トレースファイル	スクリプト実行フォルダ ¥ スクリプトファイル名 .SPX
ユーザトレースファイル	Message コマンド実行時のトレースファイル	Message コマンドの OutputName に指定したファイルパス
サーバトレースファイル	NetExec , SetGV , GetGV , DeleteGV コマンドで他コンピュータを指定した場合に出力されるサーバ上の実行結果ログ	インストール先フォルダ ¥Data¥SPTSVTRC.SPY

## 5. トラブルへの対処方法

情報の種類	概要	格納場所
NetExec エラーログ ファイル	NetExec コマンド実行時のクライアントとサーバ上に出力されるエラー情報	インストール先フォルダ ¥Log¥STXNetExec_Client¥STXNetExec_C_xx.log インストール先フォルダ ¥Log¥STXNetExec_Server¥STXNetExec_S_xx.log
自動起動情報ファイル	自動起動情報を格納したファイル	Windows XP（全ユーザ共通）の場合： インストール先フォルダ ¥Data¥SPTHLNCH.SPH Windows XP（ユーザごと）の場合： インストール先フォルダ ¥Data¥ 現在のユーザ名 ¥SPTHLNCH.SPH Windows Vista の場合： システムドライブ ¥ProgramData¥Hitachi¥Script¥Data¥SPTHLNCH.SPH
サーバ環境ファイル	サーバ環境が設定されているファイル	インストール先フォルダ ¥Data¥SPTSV.SPS
トレース管理ファイル	トレースファイルを管理するファイル	インストール先フォルダ ¥Data¥SPTLOGDB.SPB
グローバル変数ファイル	グローバル変数が保存されたファイル	インストール先フォルダ ¥Data¥SPTGV.SPG
レジストリ情報	JP1/Script のレジストリ情報	HKEY_LOCAL_MACHINE¥Software¥Hitachi¥JP1/Script¥ 配下
その他	スクリプト処理で扱っている ini ファイルやスクリプト処理で作成しているログファイルなど	-

### 5.4.3 オペレーション内容

トラブル発生時のオペレーション内容で必要な情報を次に示します。

- ・オペレーション内容の詳細
- ・トラブル発生時刻
- ・マシン構成（各 OS のバージョン，ホスト名など）
- ・再現性の有無

### 5.4.4 画面上の情報

次に示すハードコピーを採取してください。

- ・アプリケーションエラーが発生した場合は，操作画面のハードコピー
- ・エラーメッセージダイアログボックスのハードコピー（画面全体）
- ・プロセスビューアのハードコピー（画面全体）
- ・タスクマネージャの SPTXE プロセスのハードコピー（[ 表示 ] - [ 列の選択 ] で [ スレッドの数 ] をチェックし，スレッド数も採取してください）

## 5.5 資料の採取方法

トラブルが発生したときに資料を採取する方法を次に示します。

### 5.5.1 資料採取ツール

#### (1) 資料採取ツールの準備

資料採取ツールはインストールフォルダの Tools フォルダ下の Sptras ファイルになります。エクスプローラなどを使って Sptras ファイルを任意のフォルダに複写してください。次に、複写先のファイルの拡張子を「.bat」に変更してください。なお、インストール先がシステムドライブの ¥Program Files¥HITACHI¥Script 以外の場合、複写先のファイルをエディタで編集する必要があります。次に示す行の右辺にインストール先を指定してください。

```
@set INST_DIR_SCRIPT=%SystemDrive%¥Program files¥hitachi¥Script
```

#### (2) 資料採取ツールの実行

資料採取ツールを実行します。次にツールの実行例を示します。

```
c:¥>c:¥temp¥Sptras.bat
```

資料採取ツールの実行結果は、デフォルトでは %TEMP%¥jplscript フォルダ下に出力されます。このフォルダ下を採取してください。バッチファイルの実行結果を出力したファイルが作成済みの場合、情報の上書きを確認するメッセージが出力されるので「y」で応答してください。

#### ! 注意事項

資料採取ツールは、スクリプトファイルを実行していない時にバッチファイルを実行してください。Script プロセスビューアで実行しているスクリプトファイルを確認できます。また、資料採取ツールが終了するまでスクリプトを実行しないように注意してください。

### 5.5.2 OS システム情報の採取

- Windows XP の場合  
[ コンピュータの管理 ] の [ システム情報 ] の内容、または [ アクセサリ ] - [ システムツール ] - [ システム情報 ] の内容
- Windows Server 2003 の場合  
[ アクセサリ ] - [ システムツール ] - [ システム情報 ] の内容

### 5.5.3 ダンプ情報

アプリケーションエラーが発生した場合は、「5.4.1 OS の情報」に示すダンプ情報を採取してください。

### 5.5.4 JP1/Script の資料採取

トラブルが発生したスクリプトファイル（拡張子：SPT）、およびそのスクリプトファイルの実行環境ファイル（拡張子：SPV）、モニタリング情報ファイル（拡張子：SPD）、メニューファイル（拡張子：SPN）、解析トレースファイル（拡張子：SPA）、実行トレースファイル（拡張子：SPX）、ユーザトレースファイル（Message コマンドで出力したファイル）、スクリプトファイル中で処理しているファイル（ini ファイル、TextOpen コマンドに指定しているファイルなど）を採取してください。

### 5.5.5 Windows イベントログの採取

Windows の [ イベントビューア ] ウィンドウで、Windows イベントログ（アプリケーションログ/システムログ）をイベント形式とテキスト形式でそれぞれファイルに保存してください。

## 5.6 バックアップとリカバリー

JP1/Script で扱うファイルや動作環境情報のバックアップとリカバリーについて説明します。

### 5.6.1 JP1/Script で扱うファイルのバックアップとリカバリー

バックアップ対象ファイルを表 5-8 に示します。バックアップとリカバリーは必要に応じて利用者が実施してください。

なお、スクリプトファイル以外のファイルは、スクリプトの定義後または実行後に作成されるため、存在しない場合があります。

表 5-8 バックアップ対象ファイル

項番	ファイル名	ファイル名または拡張子	ファイルの所在
1	スクリプトファイル	.SPT	任意のフォルダ。ユーザが作成する。
2	実行環境ファイル	.SPV	スクリプト実行フォルダ。スクリプトの定義後または実行後に作成される。
3	メニュー情報ファイル	.SPN	
4	モニタリング情報ファイル	.SPD	
5	解析トレースファイル	.SPA	
6	実行トレースファイル	.SPX	
7	ユーザトレースファイル	.TXT など	Message コマンドの OutputName に指定したファイルパス。
8	サーバトレースファイル	SPTSVTRC.SPY	インストール先フォルダ ¥Data フォルダ下。 <sup>1</sup> <sup>2</sup> スクリプトの定義後または実行後に作成される。
9	トレース管理ファイル	SPTLOGDB.SPB	
10	サーバ環境ファイル	SPTSV.SPS	
11	グローバル変数ファイル	SPTGV.SPG	
12	自動起動情報ファイル	SPTLNCH.SPH	
13	実行環境構文ファイル	.SPU	実行環境ファイルと同じフォルダまたは toDirName で指定したフォルダ。実行環境ファイルコンバータを使用しない場合は存在しない。

注 1

## 5. トラブルへの対処方法

Script マネージャの [ ツール ] - [ オプション ] - [ クラスタ環境 ] で管理ファイルの出力先が変更されている場合、変更先のフォルダ下に作成されます。

### 注 2

Windows Vista の場合、システムドライブの ¥ProgramData¥Hitachi¥Script¥Data フォルダ下に作成されます。

### 注 3

Windows XP の場合、インストール先フォルダ ¥Data フォルダの下のフォルダ下にも作成されます。

Windows Vista の場合、システムドライブの ¥ProgramData¥Hitachi¥Script¥Data フォルダ下と、その下のフォルダ下に作成されます。詳細は、「5.4.2 JP1/Script の情報」を参照してください。

## 5.6.2 動作環境情報のバックアップとリカバリー

- Script マネージャの [ ツール ] - [ オプション ] で設定した内容  
Script マネージャの [ ツール ] - [ オプション ] ダイアログで設定した内容は、「サーバ情報」以外はレジストリに書き込まれる内容なのでバックアップは利用者が記録し、その内容をダイアログで設定することでリカバリーできます。
- JP1/Script サービスのアカウント  
バックアップは JP1/Script サービスのアカウントを利用者が記録し、その内容を再設定することでリカバリーできます。
- その他  
利用者がレジストリに設定した内容は、その内容を再設定することでリカバリーできます。



# 6

## JP1/Script の規則

JP1/Script では，ステートメント，基本コマンド，および特殊コマンドを使用して，スクリプトを作成します。この章では，スクリプトに関する規則，およびコマンドラインに関する規則について説明します。

---

### 6.1 スクリプトに関する規則

---

### 6.2 コマンドラインに関する規則

---

## 6.1 スクリプトに関する規則

---

ここでは、コマンドを使用してスクリプトを作成する場合に、知っておいていただきたいスクリプトに関する規則を、次に示す項目に分けて説明します。

- 変数の名前付け規則
- 利用できる変数の個数と容量
- 変数名として扱えないキーワード
- 予約変数
- 配列変数
- 定数
- 数字の記述規則
- 文字列の記述規則
- 演算規則
- 演算の優先順位
- コーディング規則
- JP1/Script の終了コード
- JP1/Script のイベントログ

### 6.1.1 変数の名前付け規則

JP1/Script の変数の名前付け規則は次のとおりです。

- 変数名の先頭文字は、必ず半角の英字 (A ~ Z, a ~ z), または全角文字でなければなりません。先頭以外の文字には、半角の英字, 数字, アンダースコア (\_), および全角文字を使用できます。
- 変数名は、半角で 32 文字以内でなければなりません。33 文字以上指定した場合は、33 文字以降は無視されます。
- 変数名, およびキーワードの大文字と小文字は区別されません。
- 変数名は 2 行以上にわたって記述することはできません。
- キーワードを変数名に使用することはできません。
- 変数宣言を行った適用範囲内で、同じ変数名を複数使用できます。
- 関数名は、変数の名前付け規則に従います。ただし、関数名と変数名に同じ名前は指定できません。
- プロシージャ名は変数の名前付け規則に従います。ただし、プロシージャ名と変数名に同じ名前を設定することはできません。

注

プロシージャとは、実行時に一つの単位として処理されるコマンドの集まりのことをいいます。JP1/Script では、Function ステートメントと Sub ステートメントで定義された一連の処理がこれに該当します。「サブルーチン」と同意語です。

## 6.1.2 利用できる変数の個数と容量

JP1/Script で利用できる変数の個数と容量は次のとおりです。

- 一つのプロシージャで利用できる変数は、1,024 個までです。
- プロシージャの外部で利用できる変数は、1,024 個までです（ただし、予約変数、および位置変数は含まれません）。
- 変数に格納できる容量は、最大 1,024 バイトです。1,024 バイト以降のデータは無視されます。

## 6.1.3 変数名として扱えないキーワード

変数名として扱えないキーワード（JP1/Script の予約語）を、表 6-1 に示します。

表 6-1 変数名として扱えないキーワード

見出し	変数名として扱えないキーワード
A	Abort , AbortAll , AbortRetryIgnore , Abs , AddStr , After , Alert , Alertness , AllDq , AllGV , Alt , And , Anyway , Append , AppliModal , ApplicationModal , Array , Asc , AscB , AscW , Atn , ATTR_ARCHIVE , ATTR_COMPRESSED , ATTR_HIDDEN , ATTR_NORMAL , ATTR_OFF , ATTR_OFFLINE , ATTR_ON , ATTR_READONLY , ATTR_SUBDIR , ATTR_SYSTEM , ATTR_TEMPORARY , AuditFailure , AuditSuccess
B	Backup , Beep , Before , BitmapHide , BitmapShow , Boolean , ByRef , Byte , ByVal
C	CalcDate , CalcTime , Call , CallDll , CallSpt , Cancel , CancelStartUp , CancelUserErr , Case , CatFiles , CBool , CByte , CDate , CDbl , CDROM , CheckDirName , CheckDriveType , Chr , ChrB , ChrW , Cint , Clear , CLng , Close , Command , CompDate , CompTime , Continue , Copy , Cos , Create , CreateObject , Critical , CSng , CStr , Ctrl , Ctrl_Alt , Currency , Cur_Desktop , Cur_Program , Cur_Startmenu , Cur_Startup
D	Date , DateSerial , DateValue , Delete , DeleteGV , Day , DayU , Debug , DeleteDir , DeleteFile , DeleteGroup , DeleteShortcut , DependG , DependM , Description , Destroy , Dim , DISABLE , DispName , Do , Double
E	Each , Else , ElseIf , Emergence , Emergency , Empty , End , Enter , EntryStartUp , Equal , Eqv , Erase , Err , Errctl , Error , ErrSkip , ErrSkip2 , Esc , Ex , Exclamation , ExclDir , Exec , EXEC_RUNNING , EXEC_STOPPED , Exit , ExitWindows , Exp , Explicit
F	False , FileTime , Fix , FIXED , For , Force , Format , FreeExt , Function , F1 , F2 , F3 , F4 , F5 , F6 , F7 , F8 , F9 , F10 , F11 , F12
G	GetArrayCount , GetDateCount , GetDiskFreeSpace , GetEnv , GetEnvironment , GetErrorMessage , GetExecStatus , GetFileAttr , GetFileAttribute , GetFileSize , GetFileTime , GetGV , GetPath , GetProcessCount , GetProcessInfo , GetServiceName , GetTextPosition , GetTimeCount , GetVerInfo , GetVersionInfo , GetVolLabel , GetVolumeLabel , Goto , Group

## 6. JP1/Script の規則

見出し	変数名として扱えないキーワード
H	HelpContext , HelpFile , Hex , HIDE , HKEY_CLASSES_ROOT , HKEY_CURRENT_USER , HKEY_LOCAL_MACHINE , HKEY_USERS , Hour , HourU
I	If , IfEmpty , Ignore , IMEventMessage , Imp , InArray , Info , Information , IniRead , IniWrite , InputBox , InputLine , InStr , InStrB , Int , Integer , Is , IsArray , IsDate , IsDef , IsDefine , IsEmpty , IsEmptyDir , IsEmptyGroup , IsEmptyReg , IsExistDir , IsExistFile , IsExistRegKey , IsExistService , IsFileAttr , IsFileAttribute , IsLeapYear , IsLower , IsMultiChar , IsNew , IsNull , IsNumeric , IsObject , IsSingleChar , IsUpper , IsWriteableDir
J	JOBCancel , JOBHold , JOBSubmit , JOBWait
K	KB
L	LBound , LCase , Lcl_Desktop , Lcl_Program , Lcl_Startmenu , Lcl_Startup , Left , LeftB , Len , LenB , Log , Logoff , Logon , LOGON_FAILED , Long , Loop , Ltrim
M	MakeDir , MakeGroup , MakePath , MakeShortcut , Max , MB , Menu , Message , MessageBox , MessageEventLog , Mid , MidB , Min , Minus , Minute , MinuteU , Mod , Mod= , Modify , Month , MonthU , Move
N	Name , NeedDq , NetExec , Next , No , NOCHANGE , None , NoOverwrite , NoReplace , Normal , Not , NotEqual , Nothing , Notice , Now , Null , Number
O	Object , Oct , OK , OKCancel , On , Option , Or , Overwrite , OverwriteOnly
P	Password , Path , Pause , Pile , Plus , Poweroff , Preserve , Private , ProcessEnv , Public
Q	Question
R	Raise , RAMDISK , Randomize , ReadOnly , ReadWrite , Reboot , ReDim , REG_BINARY , RegDelete , RegDeleteKey , REG_DWORD , REG_DWORD_BIG_ENDIAN , REG_EXPAND_SZ , REG_LINK , REG_MULTI_SZ , REG_NONE , RegRead , REG_RESOURCE_LIST , REG_SZ , RegWrite , REMOTE , REMOVABLE , Release , Rem , Remove , Rename , Replace , ResetStandardFile , ResetStdFile , Restart , Resume , Retry , RetryCancel , Right , RightB , Rnd , Rtrim

見出し	変数名として扱えないキーワード
S	Second , SecondU , Security , Select , SplitFile , SeparateStr , SeparateStrCount , Service , ServiceChange , ServiceContinue , SERVICE_AUTO_START , SERVICE_BOOT_START , SERVICE_CONTINUE_PENDING , ServiceControl , SERVICE_CONTROL_CONTINUE , SERVICE_CONTROL_PAUSE , SERVICE_CONTROL_STOP , ServiceCreate , ServiceDelete , SERVICE_DEMAND_START , SERVICE_DISABLED , SERVICE_ERROR_CRITICAL , SERVICE_ERROR_IGNORE , SERVICE_ERROR_NORMAL , SERVICE_ERROR_SEVERE , SERVICE_FILE_SYSTEM_DRIVER , ServiceGetValue , SERVICE_KERNEL_DRIVER , ServicePause , SERVICE_PAUSE , SERVICE_PAUSE_PENDING , ServiceQuery , ServiceRefer , SERVICE_RUNNING , ServiceSetValue , ServiceStart , SERVICE_START_PENDING , ServiceStop , SERVICE_STOPPED , SERVICE_STOP_PENDING , SERVICE_SYSTEM_START , SERVICE_WIN32_OWN_PROCESS , SERVICE_WIN32_SHARE_PROCESS , Set , SetEnv , SetEnvironment , SetFileAttr , SetFileAttribute , SetFileTime , SetGV , SetPath , SetStandardFile , SetStdFile , SetVolLabel , Shutdown , SetVolumeLabel , Shift , Shift_Alt , Shift_Ctrl , Shift_Ctrl_Alt , Sgn , Sin , Single , Skip , Sleep , Source , Space , SplitPath , Sqr , Start , StartName , StdError , StdInput , StdOutput , Step , Stop , Str , StrComp , String , StringJ , Sub , SubDirToo , Submit , Syntax , Sysmodal , SystemEnv , SystemModal
T	Tan , Target_DispClear , Target_DispOff , Target_DispOn , Target_File , Target_SPAMFile , Target_SPXFile , TempDir , TempFile , TerminateProcess , TextClose , TextFileReplace , TextOnly , TextOpen , TextRead , TextSeek , TextWrite , Then , Time , TimeSerial , TimeValue , To , ToBegin , ToEnd , Trace , Trim , True , Twice , Type , TypeOf
U	UBound , UCase , UnSubmit , Until , Update , UserEnv , UserErr
V	Val , Variant , VarType , Version , VersionUp
W	Wait , WaitAll , WaitForExec , Warning , Weekday , Wend , While , WriteOnly
X	Xor
Y	Year , Yes , YesNo , YesNoCancel
記号	^ , ^ = , - , -= , * , *= , / , /= , ¥ , ¥ = , + , += , & , & = , ? , = , < > , < , > , < = , > =

## 注

これらのキーワードは、今後のエンハンスで順次サポートする予定のため、予備として確保してあります。ご使用にならないでください。

### 6.1.4 予約変数

JP1/Script には、特定のデータ（システム情報、およびコマンドの戻り値）を参照できる「予約変数」が用意されています。

予約変数の種類と意味を表 6-2 に示します。

表 6-2 予約変数一覧

分類	予約変数	意味
システム予約変数	_ALLRIGHT_	アドミニストレータ権限がある場合は YES になり、権限がない場合は Empty 値になります。
	_BIN_	実行時の起動フォルダ名です。末尾に ¥ が付きます。
	_COMP_	現在のシステムにログオンしているコンピュータ名です。
	_DOMAIN_	ログオンしているドメイン名です。
	_OS_	OS の種類とバージョンです。 <ul style="list-style-type: none"> <li>• Windows 2000 の場合：WIN_NT5.0</li> <li>• Windows XP の場合：WIN_NT5.1</li> <li>• Windows Server 2003 の場合：WIN_NT5.2</li> <li>• Windows Vista の場合：WIN_NT6.0</li> </ul>
	_OS_PLATFORM_	OS の種類です。 <ul style="list-style-type: none"> <li>• WIN_NT</li> </ul>
	_OS_REVISION_	OS のマイナーバージョンです。 <ul style="list-style-type: none"> <li>• Windows 2000 および Windows Vista の場合：0</li> <li>• Windows XP の場合：1</li> <li>• Windows Server 2003 の場合：2</li> </ul>
	_OS_VERSION_	OS のメジャーバージョンです。 <ul style="list-style-type: none"> <li>• Windows 2000, Windows XP および Windows Server 2003 の場合：5</li> <li>• Windows Vista の場合：6</li> </ul>
	_SCF_	実行中のスクリプトファイルのフォルダ名です。末尾に ¥ が付きます。
	_SCF_FIL_	実行中のスクリプトファイルのファイル名です。拡張子は付きません。
	_SCF_EXT_	スクリプトファイルの拡張子 (.SPT) です。
	_SNF_EXT_	メニュー情報ファイルの拡張子 (.SPN) です。
	_SVF_EXT_	実行環境ファイルの拡張子 (.SPV) です。
	_SDF_EXT_	モニタリング情報ファイルの拡張子 (.SPD) です。
	_SYS_	MS-DOS の起動フォルダ名です。末尾に ¥ が付きます。
	_TEMP_	一時ファイル用のフォルダ名です。末尾に ¥ が付きます。
	_USER_	現在のシステムにログオンしているユーザ名です。
	_WIN_	Windows フォルダ名です。末尾に ¥ が付きます。
	_WINSYS_	Windows ドライバが存在するフォルダ名です。末尾に ¥ が付きます。

分類	予約変数	意味
プロセス予約変数	_PROC_ID_	実行中のスクリプトのプロセス識別子です。
	_ARGV_	%1 以降の位置変数を格納した配列変数です。%0 は含みません。_ARGV_(n) の形式で参照 (n は 1 からの数字) できます。
	_ARGV_CNT_	%1 以降の位置変数の合計数です。%0 は含みません。
コマンド戻り値予約変数	_COPY_RTN_	Copy コマンドの実行結果です。
	_COPY_CNT_	Copy コマンドでコピーしたファイルの数です。
	_COPY_SKIP_CNT_	Copy コマンドでコピーしなかったファイルの数です。
	_COPY_SKIP2_CNT_	Copy コマンドの ErrSkip2 指定で無視されたファイルの数です。
	_EXEC_RTN_	Exec, NetExec, または CallSpt コマンドの戻り値です。符号付きの数値です。
	_EXEC_ID_	Exec, または NetExec コマンドの実行ファイル識別子です。ただし、Flag に False を設定した場合 (呼び出したプログラムの終了指定がなしの場合) にだけ、有効になります。
	_JOB_RTN_	ジョブ操作系コマンドの戻り値です。
	_MSG_RTN_	InputDialog, または MessageBox コマンドの戻り値です。
	_DLL_RTN_	CallDll コマンドの外部関数の戻り値です。
	_FORM_TERM_KEY_	画面 (メニューフォーム) を終了させた要因です。
	_FORM_TERM_CMDNO_	コマンドを実行して終了したときのコマンド番号です。
	_FORM_MODIFY_KEY_	画面 (メニューフォーム) 終了キーのモディファイキーです。
	_FORM_FIELD_NAME_	最後にフォーカスがあったメニュー名です。
	_RTN_	エラー詳細コードです。符号付きの数値です。コマンドの戻り値が False の場合、設定されます。ただし、コマンドの戻り値が特定の戻り値 (文字列操作コマンド, 日付操作コマンド, その他のコマンドなど) の場合は、予約変数 _RTN_ には常に 0 (ゼロ) が設定されます。また、エラー詳細コードの内容は、トレースファイルに出力されるエラーメッセージの内容となります。
	_RTNxx_	CallDll コマンドの戻り値 (xx は 00 からの数字) です。
	_SVC_RTN_	サービス操作系コマンドの戻り値です。

分類	予約変数	意味
文字コード 予約変数	_NL_	改行文字です。
	_TAB_	タブ文字です。
エラー詳細 コード予約 変数	_NO_ERR_	エラーはありません。
	_ERR_EOF_	ファイルの終わりに達しました。
	_ERR_TIMEOUT_	タイムアウト時間を経過しました。
	_ERR_FILE_	ファイルが見つかりません。
	_ERR_PATH_	パスが見つかりません。
	_ERR_ACCESS_	アクセスが拒否されました。
	_ERR_PROTECT_	書き込みが禁止されています。
	_ERR_READY_	デバイスが準備できていません。
	_ERR_EXCLUSIVE_	ファイルは他でアクセス中です。
	_ERR_SVR_CONNECT_	サーバが接続要求に応答しません。
	_ERR_SVR_TIMEOUT_	サーバからの応答待ちでタイムアウトエラーが発生しました。
	_ERR_SVR_RECEIVEDATA_	サーバからデータ受信中にエラーが発生しました。
	_ERR_SVR_NODATA_	サーバとの送受信中の無通信時間が許容時間を超えました。
	_ERR_FILE_SIZE_	取得した値が変数の上限値を超えているため、値を変数に格納できません。
	_ERR_NOT_LARGE_FILE_	指定したファイルの容量が制限値を超えています。
	_ERR_FILE_POSITION_	読み書き開始位置が 2,147,483,647 を超えています。
	_ERR_SERVICE_NOT_BEGIN_	JP1/Script サービスが実行中の状態になっていません。

### 6.1.5 配列変数

配列変数のインデックス番号は 1 から始まります。一次元配列変数のインデックス番号は要素を表します。二次元配列変数の 1 番目のインデックスは行要素を表し、2 番目のインデックス番号は列要素を表します。

次に二次元配列変数「T ( 5 , 6 )」のデータ構造例を示します。



T(1, 1)	T(1, 2)	T(1, 3)	T(1, 4)	T(1, 5)	T(1, 6)
T(2, 1)	T(2, 2)	T(2, 3)	T(2, 4)	T(2, 5)	T(2, 6)
T(3, 1)	T(3, 2)	T(3, 3)	T(3, 4)	T(3, 5)	T(3, 6)
T(4, 1)	T(4, 2)	T(4, 3)	T(4, 4)	T(4, 5)	T(4, 6)
T(5, 1)	T(5, 2)	T(5, 3)	T(5, 4)	T(5, 5)	T(5, 6)

## ( 1 ) 配列変数の記述規則

配列変数の記述規則を次に示します。

配列変数名は変数の名前付け規則に従います。

配列変数の要素（二次元配列の場合は，行要素，および列要素）を表すインデックス番号は 1 から始まります。また，「( )」，または「[]」で囲んで指定します。

配列変数の宣言時にインデックス番号を指定する場合は，配列変数の要素数は固定となります。インデックス番号を指定しない場合は，配列変数の要素数は可変となります。

配列変数の次元数は，一次元，および二次元まで定義できます。

配列変数の要素数の上限は，一つの配列変数につき 65,536 個，一つのスクリプトファイルにつき 131,072 個です。

一度宣言した配列変数の次元数は変更できません。

( 例 )

Dim A ( 5 )

Dim A ( 5 , 10 )    ( エラー ) 次元数は変更できない。

一度宣言した配列変数の次元数が等しい場合は，要素数を変更できます。また，要素数が固定な配列変数から可変な配列変数へ，または可変な配列変数から固定な配列変数へ変更できます。変更した場合，今までの設定値は Empty 値となります。

( 例 )

Dim A ( 10 )

Dim A ( 5 )    要素数を変更できる。

Dim A ( )    可変な要素数に変更できる。

非配列変数から配列変数へ，または配列変数から非配列変数への再定義はできません。

( 例 )

Dim A

Dim A ( )    ( エラー ) A は非配列変数として定義されているため再定義できない。

代入文で配列変数のすべての要素を一括して代入することができます。ただし，左辺と右辺の配列変数の要素数が異なる場合，一括して代入することはできません。

( 例 )

```
Dim A ( 5 ) , B ( ) , C ( 10 )
For cnt = 1 To 5
    A ( cnt ) = Time
Next
```

B = A    A のすべての要素を一括して B に代入できる。

C = B    (エラー) B と C の要素数が異なるため一括して代入できない。

代入文で配列変数の一つの行要素だけを一括して代入することができます。ただし、一つの列要素だけを一括して代入することはできません。

(例)

```
Dim A ( 2 , 5 ) , B ( )
For cnt = 1 To 5
    A ( 1 , cnt ) = Time
Next
```

B = A ( 1 )    A の第 1 行目の要素を一括して B に代入できる。

比較文で配列変数のすべての要素が等しいかどうかを判定することができます。ただし、大小比較はできません。

(例 1)

```
Dim A ( 5 ) , B ( 5 )
:
```

If A = B Then    A と B が等しいかどうかを比較できる。

(例 2)

```
Dim A ( 5 ) , B ( 5 )
:
```

If A < B Then    (エラー) A と B の大小比較はできない。

要素数が可変な配列変数で値が未設定な中間の要素に対し代入を行った場合、先頭から中間の要素までの未設定であった要素は Empty 値になります。

(例 1)

```
Dim A ( )
A ( 1 ) = Time
```

A ( 5 ) = Time    A ( 2 ) から A ( 4 ) までの要素は Empty 値になる。

(例 2)

```
Dim A ( , )
```

A ( 2 , 5 ) = Time    A ( 1 , 1 ) と A ( 2 , 1 ) から A ( 2 , 4 ) の要素は Empty 値になる。

(例 3)

```
Dim A ( 2 , )
```

A ( 2 , 5 ) = Time    A ( 2 , 1 ) から A ( 2 , 4 ) の要素は Empty 値になる。

(例 4)

Dim A ( ,5)

A (2 ,5)=Time A (1 ,1) から A (1 ,5) と A (2 ,1) から A (2 ,4) の要素は Empty 値になる。

## (2) 配列変数のデータ構造例

配列変数のデータ構造例を次に示します。

一次元の配列変数の場合、インデックス番号は要素を表します。

(例)

Dim A (5) と宣言した場合

下図のような五つ分の要素が割り当てられます。

(第1要素)	(第2要素)	(第3要素)	(第4要素)	(第5要素)
A(1)	A(2)	A(3)	A(4)	A(5)

二次元の配列変数の場合、一番目のインデックス番号は行要素を、二番目のインデックス番号は列要素を表します。

(例1)

Dim B (1 ,5) と宣言した場合

下図のような1行5列分の要素が割り当てられます。

	(第1列)	(第2列)	(第3列)	(第4列)	(第5列)
(第1行)	B(1,1)	B(1,2)	B(1,3)	B(1,4)	B(1,5)

(例2)

Dim C (3 ,4) と宣言した場合

下図のような3行4列分の要素が割り当てられます。

	(第1列)	(第2列)	(第3列)	(第4列)
(第1行)	C(1,1)	C(1,2)	C(1,3)	C(1,4)
(第2行)	C(2,1)	C(2,2)	C(2,3)	C(2,4)
(第3行)	C(3,1)	C(3,2)	C(3,3)	C(3,4)

## 6.1.6 定数

JP1/Script で特別な意味を持つ定数の種類と意味を、表 6-3 に示します。

表 6-3 定数一覧

定数	意味
Empty	空の値 ("" ) を示します。
True	真 (0 以外) の値を示します。

定数	意味
False	偽 ( 0 ) の値を示します。

### 6.1.7 数字の記述規則

数字の記述規則を次に示します。

- 「0」, 「+ 0」, 「- 0」は同値とみなします。
- 「0」だけが連続している数字は「0」とみなします。
- 扱える数字は整数だけです。
- コンマ ( , ) の有無は, 数字をダブルコーテーション ( " ) で囲んだ場合には区別されません。  
( 例 ) "10,000" と 10000 は同値とみなします。
- 数字の先頭の「+」の有無は区別されません。例えば, 600 と +600 は同値とみなします。
- 扱える数値は, -2,147,483,647 ~ 2,147,483,647 の範囲です。
- 数字をダブルクォーテーション ( " ) で囲んでも文字列としては扱われません。例えば, "10000" と 10000 は同じ値とみなします。文字列として扱うには Str コマンドを使用してください。

### 6.1.8 文字列の記述規則

文字列の記述規則を次に示します。

- 数値以外の文字列を記述する場合はダブルクォーテーション ( " ) で囲みます。
- 変数, 予約語などをダブルクォーテーション ( " ) で囲むと文字列として扱われます。  
ただし, 予約語を指定するコマンドの引数に指定された場合は, 予約語として扱われます。
- 扱える文字列の長さは最大 1,024 文字です。
- 何も囲んでいないダブルクォーテーション ( " ) は Empty 値とみなします。
- ダブルクォーテーション ( " ) で囲まれている数値にコンマ ( , ) 以外の文字列が含まれると文字列として扱われます。

### 6.1.9 演算規則

スクリプトの演算規則を次に示します。

- ( ) 内の演算は, 最も内側のものから行われます。
- 変数は, 基本的に文字列型として扱われますが, 数字だけで構成されている文字列同士の演算の場合は数字として扱われ, 演算が行われます。
- 単項演算子は「+」, 「-」だけです。

## 6.1.10 演算の優先順位

演算の優先順位を表 6-4 に示します。

表 6-4 演算の優先順位

優先度	演算子	内容
高い	^	指数演算
	+, -	単項演算子
	*, /, ¥, Mod	乗算, 除算, 整数除算, 剰余演算
	+, -	加算, 減算
	&	文字列連結
	=, <>, <, <=, >, >=	比較演算
	Not	論理否定
	And	論理積
低い	Or	論理和

## 6.1.11 コーディング規則

JP1/Script のコーディング規則を次に示します。

1 文（ステートメントを除く）が複数行にわたるときは、アンダースコア（\_）を文の最後に記述します。アンダースコアは、前に 1 個以上の半角スペースを入れてから記述します。なお、引数を伴うコマンド文で引数を括弧で囲んである場合は、アンダースコアを記述する必要はありません。

（例）引数を括弧で囲む場合

```
Command (A,B,C,
         D,E,F)
```

（例）引数を括弧で囲まない場合

```
Command A,B,C, _
         D,E,F
```

コマンド名を、複数行にわたって記述することはできません。

コマンドの大文字と小文字は区別されません。

1 行に記述できるスクリプトは最大 10,240 バイトです。10,240 バイトを超えた部分のスクリプトは無視されます。

スクリプトに記述できる文字列は最大 1,024 バイトです。1,024 バイトを超えた部分のスクリプトは無視されます。

コマンドの引数の記述は、以下の形式のうちのどれかで記述します。引数が複数個存在する場合は、「,」（コンマ）、またはスペースで区切ります。

（例）引数の記述（ : スペース）

```
Command (A)
Command (A)
Command A
Command (A,B)
Command (A,B)
Command A B
Command A,B
```

コマンド、または関数の引数に、引数を伴うコマンド、または関数を記述する場合には、引数として記述したコマンド、または関数の引数を、必ず括弧で囲まなくてはなりません。

(例) 必ず括弧で囲む場合の記述

```
Command( Function( A,B,C),D,E)
Command Function( A,B,C),D,E
Command Function( A,B,C)D E
```

演算子の左辺または右辺に、引数を伴うコマンドまたは関数を記述する場合は、引数を必ず括弧で囲まなくてはなりません。

(例) M=Command(A,B,C) + Function(D)

スクリプトを実行させるスクリプトエンジンのバージョンを指定するときには、各スクリプトファイルの先頭に、#FileVersion=VVR (VV: JP1/Script のバージョン, RR: JP1/Script のリビジョン) と記述しなければなりません。この記述がない場合、または存在しないバージョンが指定された場合は、マネージャウィンドウの [ オプション ( 互換性情報 ) の設定 ] ダイアログで設定されたバージョンで、スクリプトファイルが実行されます。なお、新規にスクリプトファイルを作成するときには、先頭にスクリプトエンジンのバージョンを自動的に挿入するかどうかを、マネージャウィンドウの [ 新規作成 ] ダイアログで指定できます。

(例) #FileVersion = 0700

GoTo 文や On Error 文などの制御の分岐先は、ラベル名の直後にコロン ( : ) を付加したものを指定しなければなりません。

(例) LabelName:

文字列中に含まれる "¥r", "¥n", "¥t", "¥¥" をそれぞれ対応するコントロールコードに変換しない場合は、スクリプトファイルの先頭に、#Option = NOCHANGE と記述しなければなりません (ただし、#FileVersion = VVR が指定されている場合はその下の行に記述しなければなりません)。この記述がない場合には文字列中に含まれる "¥r", "¥n", "¥t", "¥¥" はそれぞれ対応するコントロールコードに変換されます。

("¥r" は復帰, "¥n" は改行と復帰, "¥t" はタブ, "¥¥" は一つの "¥" に変換)

なお、#Option = NOCHANGE の記述があっても、スクリプトエンジンの実行バージョンが 0500 以前のものであれば、無条件に文字列中に含まれる "¥r", "¥n", "¥t", "¥¥" をそれぞれ対応するコントロールコードに変換します。

(例)

```
#FileVersion = 0700
#Option = NOCHANGE
MessageBox( "C:¥¥WkDir¥¥Script.SPT" )
```

[メッセージボックスに出力される文字列]  
 C:¥¥WkDir¥¥Script.SPT

( 例 )

```
#FileVersion = 0700
MessageBox( "C:¥¥WkDir¥¥Script.SPT" )
```

[メッセージボックスに出力される文字列]  
 C:¥WkDir¥Script.SPT

次のコマンドの場合、文字列中に含まれる "&" 文字は表示されないでその次の文字が下線付きで表示されます。文字列中に含まれる "&" 文字をそのまま表示させる場合は、スクリプトファイルの先頭に、#Option = NOPREFIX と記述してください。#Option = NOCHANGE の記述がある場合は、#Option = NOCHANGE, NOPREFIX とコンマで区切って指定します。(指定は順不同です。ただし、#FileVersion = VVRR が指定されている場合はその下の行に記述してください。)

- Message コマンドでウィンドウ出力の場合のメッセージ
- InputBox コマンドでメッセージとテキストボックスのキャプション

なお、スクリプトエンジンの実行バージョンが 0520 より前のものである場合、#Option = NOPREFIX の記述があっても、文字列中に含まれる "&" 文字は表示されないでその次の文字が下線付きで表示されます。"&" 文字を表示する場合は "&&" のように記述してください。

( 例 )

```
#FileVersion = 0700
#Option = NOCHANGE,NOPREFIX
Message( Target_DispOn,"Display","Client&Server" )
```

[ウィンドウに出力される文字列]  
 Client&Server

( 例 )

```
#FileVersion = 0700
Message( Target_DispOn,"Display","Client&Server" )
```

[ウィンドウに出力される文字列]  
 Client\_Server

n 個のダブルクォーテーション ( " ) を文字としてそのまま表す場合は、文字列中に ( n \* 2 ) 個のダブルクォーテーションを指定します。

( 例 )

```
MessageBox( "Error code:""99"" " )
```

[メッセージボックスに出力される文字列]  
 Error code:"99"

### 6.1.12 JP1/Script の終了コード

スクリプト実行の終了コードには、基本的に Exit コマンド、または ExitWindows コマンドで指定された値が返されます。これらのコマンド、またはこれらのコマンドの引数を省略した場合、終了コードには「0」が返されます。ただし、スクリプト実行がエラーになった場合、JP1/Script の終了コードが返されることもあります。

終了コードとしてレジストリに設定されている値を、表 6-5 に示します。なお、設定されているレジストリは次のとおりです。

HKEY\_LOCAL\_MACHINE¥SOFTWARE¥Hitachi¥JP1/Script¥SPTX¥ExitCode

表 6-5 レジストリに設定されている終了コード

値名	デフォルト値	意味
AlreadyRun	16	指定されたスクリプトファイルは既に起動されています。
Terminate	17	JP1/Script のプロセスを強制終了しました。
TimeOut	18	打ち切り時間を経過したため JP1/Script のプロセスを終了しました。
GrammarError	19	文法エラーが発生しました。
ExAbortError	20	JP1/Script のプロセスを中断する実行エラーが発生しました。(メモリ不足、未定義の変数を参照しているエラー、プロシージャが見つからないエラー、ステートメントで発生するエラーなど)
Error	99	JP1/Script のプロセス開始前にエラーが発生しました。(指定されたスクリプトファイルが見つからないエラーなど)
	21 ~ 32	将来の予備

#### ! 注意事項

Exit コマンドや ExitWindows コマンドで終了コードを指定する場合は、JP1/Script の終了コードを重複しないように注意してください。

### 6.1.13 JP1/Script のイベントログ

JP1/Script では状況に応じてイベントログを出力します。また、ユーザ側でも MessageEventLog コマンドを使ってイベントログの出力を指定することができます。

出力されるイベント ID とイベントログ、および内容を、表 6-6 に示します。

表 6-6 イベント ID とイベントログ

イベント ID	イベントログ	出力される内容 <sup>1</sup>
1	スクリプト実行開始時の情報ログ	スクリプトの実行を開始しました。(スクリプトファイル名)



イベント ID	イベントログ	出力される内容 <sup>1</sup>
2	スクリプト実行終了時の情報ログ	スクリプトの実行を終了しました。(スクリプトファイル名)
3	MessageEventLog コマンドを使ったユーザ指定のログ	MessageEventLog コマンドで指定されたメッセージ
4	JP1/Script サービスや Script ランチャ実行時のエラーログ, および警告ログ	JP1/Script のインストールに失敗している可能性があります。JP1/Script をインストールし直してください。」
		自動起動情報ファイルの内容が不正です。自動起動情報ファイルを再作成してください。 (自動起動情報ファイル名)
		発生したエラーのエラーメッセージ
5	ExitWindows コマンド実行時の警告ログ	ExitWindows (コマンドで指定されたシャットダウン操作の種別) コマンドが発行されました。(スクリプトファイル名)
6	JP1/IM, または JP1/Base へのイベント発行失敗時の警告ログ	JP1/Integrated Manager の関数でエラーが発生しました。(JP1/IM または JP1/Base の関数名 ::JP1/IM または JP1/Base の関数のエラー状態コード)
7	TerminateProcess コマンド実行時の情報ログ	TerminateProcess (Id= コマンドで指定されたプロセス ID) コマンドが発行されました。 (スクリプトファイル名)

イベント ID	イベントログ	出力される内容 <sup>1</sup>
8 <sup>2</sup>	JP1/Script サービスとランチャの実行開始，および実行終了時の情報ログ	Script ランチャを開始します。(日付 時刻 システム起動時からの経過時間)
		Script ランチャを終了します。(日付 時刻 システム起動時からの経過時間)
		Script ランチャの NetExec スレッドを開始します。(日付 時刻 システム起動時からの経過時間)
		Script ランチャの NetExec スレッドを終了します。(日付 時刻 システム起動時からの経過時間)
		Script 終了制御ランチャが実行されました。(日付 時刻 システム起動時からの経過時間)
		JP1/Script サービスを開始します。(日付 時刻 システム起動時からの経過時間)
		JP1/Script サービスを終了します。(日付 時刻 システム起動時からの経過時間)
		JP1/Script サービスの NetExec スレッドを開始します。(日付 時刻 システム起動時からの経過時間)
		JP1/Script サービスの NetExec スレッドを終了します。(日付 時刻 システム起動時からの経過時間)
		JP1/Script ファイルマッピングオブジェクト制御スレッドを開始します。(日付 時刻 システム起動時からの経過時間)
		JP1/Script ファイルマッピングオブジェクト制御スレッドを終了します。(日付 時刻 システム起動時からの経過時間)
16	複数スクリプト起動時のエラーログ	指定されたスクリプトファイルは既に起動されています。(スクリプトファイル名)
17	スクリプト強制終了時の警告ログ	JP1/Script のプロセスを強制終了しました。(スクリプトファイル名)
18	打ち切り時間経過によるスクリプト終了時の警告ログ	打ち切り時間を経過したため JP1/Script のプロセスを終了しました。(スクリプトファイル名)
19	文法エラーによるスクリプト終了時の情報ログ	スクリプトの実行を終了しました。文法エラーがありました。(スクリプトファイル名)
96	スクリプト実行開始時，JP1/Script サービスの状態が実行中以外だった	JP1/Script サービスの状態が実行中以外のため，解析 / 実行トレースに実行結果が出力されません。また Message/EntryStartUp/CancelStartUp/ グローバル変数操作コマンドの実行がエラーになります。JP1/Script サービスを開始してから実行してください。(スクリプトファイル名)

イベント ID	イベントログ	出力される内容 <sup>1</sup>
98	解析トレース, および実行トレース未出力時のエラーログ	ユーザ名 : スクリプト実行時のユーザ名 スクリプトファイル名 : スクリプトファイル名 次のようなエラーがありました。 エラーの行位置 行目 : この部分には発生したエラーのエラーメッセージが出力されます。
99	スクリプト実行開始時のエラーログ	発生したエラーのエラーメッセージ

## 注 1

斜体になっている部分はログごとに内容が変わる埋め字を示しています。また日付は yyyy/mm/dd, 時刻は hh:mm:ss.ss, システム起動時からの経過時間はミリ秒で表示します。

## 注 2

このイベントログの出力を抑止する場合は, 次に示すレジストリキーの値を次のように設定してください。

## [ レジストリキー ]

HKEY\_LOCAL\_MACHINE¥SOFTWARE¥HITACHI¥JP1/Script¥SPTF¥SPV

## [ 値 ]

CommandLine( 文字列 )

## [ 値のデータ ]

/NOEVLOG(8)

これらのイベントログのうち, ログの種類がエラー以外のイベントログは, 出力を抑止することができます。出力抑止を指定する場合は, 「3.1.13 スクリプトの実行環境 (各項目) を設定する」の「[ コマンドライン ] を選択した場合」を参照してください。

## 6.1.14 JP1/Script で発行するイベントログ

JP1/Script で発行する JP1/IM イベントを表 6-7 に示します。

表 6-7 JP1/Script が発行する JP1/IM イベント

イベント ID	重大度	イベントの内容	発行契機	出力する情報 ( 表 6-8 参照 )
00003B00	情報	スクリプトの開始 / 終了イベント	スクリプト実行の開始時, および終了時	C1 ~ C5, C8, C9
00003B01	エラー	メモリアクセスエラー	メモリアクセスエラー発生時	C1 ~ C5, C8
00003B03	ユーザ指定値	IMEventMessage コマンドによるイベント	IMEventMessage コマンド実行時	C1 ~ C9
00003B09	情報	コマンド正常終了時イベント	コマンドの正常終了時	C1 ~ C5, C8, C9
00003B10	エラー	コマンドエラー終了時イベント	コマンドの解析エラー終了時	C1 ~ C5, C8, C9

## 6. JP1/Script の規則

イベント ID	重大度	イベントの内容	発行契機	出力する情報 (表 6-8 参照)
00003B11	エラー	コマンドエラー終了時 イベント	コマンドの実行エ ラー終了時	C1 ~ C5,C8,C9

注 マネージャの [ ツール ] - [ オプション ] で表示される [ オプション ] ダイアログの [ JP1/  
IM ] タブで表示をする指定を行った場合に、発行されます。

表 6-8 出力する情報

記号	重大度
C1	重大度
C2	ユーザ名
C3	PP 名
C4	登録タイプ名
C5	登録名
C6	事象種別
C7	開始時刻
C8	終了時刻
C9	終了コード

## 6.2 コマンドラインに関する規則

ここでは、スクリプトファイルを実行するときの、コマンドラインの指定方法について説明します。

[ 実行環境 ( 起動情報 ) の設定 ] ダイアログボックスのコマンドラインの指定は、次の場合に必要です。

- スクリプトファイルの実行に位置変数が必要な場合
- 解析トレースファイルの出力の有無 ( デフォルトは出力する ) を指定する場合
- 実行トレースファイルの出力の有無 ( デフォルトは出力する ) , および出力レベルを指定する場合
- イベントログの出力抑止 ( デフォルトは出力する ) を指定する場合

### 6.2.1 コマンドラインの形式

コマンドラインの形式は次の 3 通りがあります。

( 1 ) スクリプトファイルの [ 実行環境の設定 ] ダイアログボックスでコマンドラインを指定する場合

位置変数    /SPALV ( n )    /SPXLV ( n )    /NOEVLOG ( n , n , ... )

この場合、スクリプトファイルは先に選択されているので、コマンドラインでスクリプトファイル名を指定する必要はありません。

なお、位置変数については、「付録 D 用語解説」を参照してください。

( 2 ) Exec コマンドや NetExec コマンドからパラメタを指定してスクリプトファイルを呼び出す場合

Exec ( スクリプトファイル名 , 終了待ちの有無 ,  
" 位置変数 " , " /SPALV ( n ) " , " /SPXLV ( n ) " , " /NOEVLOG ( n , n , ... ) " )

なお、位置変数については「付録 D 用語解説」を参照してください。

( 3 ) ユーザプログラムなどから実行形式 ( SPTXE.EXE ) でコマンドラインを指定してスクリプトファイルを実行する場合

SPTXE.EXE    スクリプトファイル名    位置変数    /SPALV ( n )    /SPXLV ( n )  
/NOEVLOG ( n , n , ... )

この場合、スクリプトファイル名はコマンドラインの先頭に指定しなければなりません。

なお、位置変数については「付録 D 用語解説」を参照してください。

## 6.2.2 コマンドラインのパラメタの説明

### (1) 位置変数

%n (n は正の整数) で表され, %0 はスクリプトファイル名 (フルパス), %1 は第 1 位置変数 (%n は第 n 位置変数) となります。

複数の位置変数を指定する場合は, コマンドラインの記述規則に従ってください。コマンドラインの記述規則については, 「6.2.3 コマンドラインの記述規則」を参照してください。

以下の例で使用しているフォルダ名の Program File は間に空白を含んでいます。

#### (例 1)

スクリプトファイルの [ 実行環境の設定 ] ダイアログボックスでコマンドラインを指定する場合

[ ダイアログのコマンドラインの指定形式 ]

```
ABC 123 "C:¥Program Files¥"
```

[ 各位置変数に設定される値 ]

```
%0 : スクリプトファイル名
%1 : ABC
%2 : 123
%3 : C:¥Program Files¥
```

#### (例 2)

Exec コマンドや NetExec コマンドからパラメタを指定してスクリプトファイルを呼び出す場合

[ コマンドラインの指定形式 ]

```
Exec ("C:¥Temp¥Test.SPT", True, "ABC", "123", "\"C:¥Program Files¥\"")
```

[ 各位置変数に設定される値 ]

```
%0 : C:¥Temp¥Test.SPT
%1 : ABC
%2 : 123
%3 : C:¥Program Files¥
```

#### (例 3)

ユーザプログラムなどから実行形式 (SPTXE.EXE) でコマンドラインを指定してスクリプトファイルを実行する場合

[ ユーザプログラムの指定形式 ]

```
SPTXE.EXE C:¥Temp¥Test.SPT ABC 123 "C:¥Program Files¥"
```

[ 各位置変数に設定される値 ]

```
%0 : C:¥Temp¥Test.SPT
%1 : ABC
%2 : 123
%3 : C:¥Program Files¥
```

### (2) /SPALV (n) (または /spalv (n))

解析トレースファイルの出力の有無を指定します。このパラメタの指定を省略すると、解析トレースファイルが出力されます。n には 0 以上の整数を入力します。デフォルトは 0 以外です。

- n の値が 0 の場合：解析トレースファイルを出力しない。
- n の値が 0 以外の場合：解析トレースファイルを出力する。

### (3) /SPXLV (n) (または /spxlv (n))

実行トレースファイルの出力の有無，および出力レベルを指定します。このパラメタの指定を省略すると，実行トレースファイルにエラー時の結果だけが出力されます。n には 0 から 3 の整数を入力します。デフォルトは 1 です。

- n の値が 0 の場合：実行トレースファイルが出力されない。
- n の値が 1 の場合：実行トレースファイルにエラー時の結果だけが出力される。
- n の値が 2 の場合：実行トレースファイルに正常時の結果も出力される。
- n の値が 3 の場合：実行トレースファイルに，コマンド開始時刻と終了時刻も出力される。

### (4) /NOEVLOG (または /noevlog)

イベントログを出力させたくない場合に指定します。このパラメタの指定を省略すると，すべてのイベントログが出力されます。イベントログの種類については，「6.1.13 JP1/Script のイベントログ」を参照してください。

このパラメタの指定を省略すると，すべてのイベントログが出力されます。

/NOEVLOG または /noevlog

ログの種類がエラーの場合にだけログを出力する。

/NOEVLOG (n,n,...) または /noevlog (n,n,...)

n で指定したイベント ID のログは出力しない。n には複数のイベント ID を順不同で指定できる。複数のイベント ID をコンマで区切って指定する。

## 6.2.3 コマンドラインの記述規則

コマンドラインを記述するときの規則を，次に説明します。

- 複数のパラメタを指定する場合は，空白で区切ってください。
- 区切りではない空白を含むパラメタを指定する場合は，パラメタ全体をダブルクォーテーション (") で囲んでください。
- 一つのスクリプトファイルで利用できる位置変数は 255 個まで (%254 まで) です。それ以降の指定は無視されます。
- 位置変数に格納できる文字数は半角文字で 1,024 文字以内です。それ以降の文字は無

視されます。

なお、1 個目と 2 個目の規則は Exec コマンドや NetExec コマンドからスクリプトファイル以外の実行ファイルと呼び出すときにも該当します。

## 6.2.4 注意事項

コマンドラインを指定するときの注意事項について、次に示します。

ユーザプログラムなどから実行形式 (SPTXE.EXE) でスクリプトファイルを実行する場合は、実行するスクリプトファイル名の指定が必要です。スクリプトファイル名は、コマンドラインの先頭に指定しなければなりません。

〔コマンドラインの指定形式〕

```
SPTXE.EXE スクリプトファイル名 位置変数 /SPALV(n) /SPXLV(n)
/NOEVLOG(n,n,...)
```

指定したコマンドラインの値はスクリプトファイル単位で有効ですが、実行コンピュータのすべてのスクリプトファイルに対して有効にしたい場合は、次のレジストリにコマンドラインの値を設定してください。

〔レジストリキー〕

```
HKEY_LOCAL_MACHINE¥Software¥Hitachi¥JP1/Script¥SPTF¥SPV
```

〔値〕

```
CommandLine (文字列)
```

注意

ダイアログとレジストリの両方にコマンドラインを指定している場合は、両方の指定が有効になります。ただし、ダイアログに "/SPXLV(3)", レジストリに "/SPXLV(0)" のように、同じパラメタに異なる指定をしている場合はダイアログの指定が優先されます。

サービスとランチャのイベントログの出力を抑止する場合は、実行環境ファイルのコマンドラインではなく、上記レジストリに /NOEVLOG、または /NOEVLOG(4) を指定してください。

実行環境のコマンドラインに /NOEVLOG、または /NOEVLOG(n, n, ...) が指定された場合、/SPALV(n) や /SPXLV(n) と異なり位置変数として扱われます。位置変数として扱わない場合は、次のレジストリの値に 1 を設定してください。

〔レジストリキー〕

```
HKEY_LOCAL_MACHINE¥Software¥Hitachi¥JP1/Script¥SPTX
```

〔値名〕

```
NoEvLog_Opt (DWORD)
```

〔値〕

1 : /NOEVLOG、または /NOEVLOG(n,n,...) を位置変数として扱わない



n 個のダブルクォーテーション (") を文字としてそのままパラメタに含める場合は、  
(n\*4) 個のダブルクォーテーション (") を指定し、パラメタ全体をダブルクォーテーション (") で囲まなければなりません。

なお、次のレジストリの値に 1 を設定した場合は、(n\*2) 個のダブルクォーテーション (") を指定します。

〔レジストリキー〕

HKEY\_LOCAL\_MACHINE¥Software¥Hitachi¥JP1/Script¥SPTX

〔値名〕

DQString\_Opt (DWORD)

〔値〕

1: パラメタ文字列に n 個のダブルクォーテーション (") を含む場合、(n\*2) 個のダブルクォーテーション (") を指定し、パラメタ全体をダブルクォーテーション (") で囲む。

このレジストリを設定しない場合、指定方法が複雑になりますので、新規のシステム開発時などは、この設定をすることをお勧めします。

注

ユーザプログラムなどから実行形式 (SPXE.EXE) でコマンドラインを指定してスクリプトを実行する場合は、(n\*4) 個のダブルクォーテーション (") とパラメタ全体を囲むダブルクォーテーション (") のそれぞれを 2 倍のダブルクォーテーション (") で指定します。また、NetExec コマンドでは、((((n\*4)\*2)\*2)\*2) 個のダブルクォーテーション (") と、パラメタ全体を囲む 15 個のダブルクォーテーション (") で指定します。なお、上記のレジストリの値に 1 を設定した場合は、Exec コマンドや NetExec コマンドでも (n\*2) 個のダブルクォーテーション (") を指定し、パラメタ全体をダブルクォーテーション (") で囲みます。

コマンドラインのパラメタの指定が、スクリプト起動時とスクリプトの実行環境ファイルの両方にある場合、両方の指定が有効になります。この場合、スクリプト起動時のコマンドラインのパラメタが先、実行環境ファイルのコマンドラインのパラメタが後になります。



# 7

## ステートメント

この章では、スクリプトを作成するときに使用できるステートメントについて説明します。

---

### 7.1 ステートメント一覧

# 7.1 ステートメント一覧

スクリプトを作成するときに使用できる、ステートメントの一覧を表 7-1 に示します。  
なお、ステートメントの後ろに続く括弧内の数字は、対象バージョンを表します（括弧がないものは、01-00 からサポートしています）。

表 7-1 ステートメント一覧

ステートメント	意味
=	右辺の値を左辺の変数に代入します。
Do...Loop ( 05-10 以降 )	指定した条件が真 ( True ) である間、または条件が真 ( True ) になるまで、一連のステートメントを繰り返します。
For...Next	指定した回数だけ、一連のステートメントを繰り返します。
For...End For	指定したフォルダ内のすべてのファイルに対して、一連のステートメントを繰り返します。
If..Then..Else	式の値に基づいて、条件付きの実行をします。
Select Case	条件式の値に従って、複数のステートメントブロックのどれかを実行します。
While...End	指定した条件が真 ( True ) である間、一連のステートメントの実行を繰り返します。
Function	Function プロシージャの名前、引数を宣言し、Function プロシージャの始まりを示します。
Sub	Sub プロシージャの名前、引数を宣言し、Sub プロシージャの始まりを示します。
Call	Sub プロシージャ、および Function プロシージャに制御を渡します。
Exit xx	While...End ループ、For...Next ループ、For...End For ループ、Do...Loop ループ、Function プロシージャ、または Sub プロシージャから抜け出します。
Goto ( 05-00 以降 )	指定したラベルに処理を分岐させます。
Continue ( 05-00 以降 )	While...End ループ、For...Next ループ、For...End For ループ、Do...Loop ループの先頭に戻ります。
On Error ( 05-00 以降 )	エラー発生時に、指定したラベルに制御を移すステートメントです。なお、エラーが発生しても指定したラベルに制御を移動しないようにすることもできます。

## 7.1.1 =

機能

右辺の値を左辺の変数に代入します。

形式

*Result* = *Expression*

## 指定項目

*Result*

値を受け取る変数名を指定します。

*Expression*

任意の式を指定します。

## 説明

式 *Expression* の値を変数 *Result* に代入します。

式 *Expression* の値が Empty 値のときは、変数 *Result* には Empty 値が代入されます。Dim コマンドで宣言していない変数に値を代入した場合は、自動的に変数を割り当てて値を代入します。

## 例

```
' 変数result1には"ABCDE"が格納される。
result1 = "ABCDE"

' 変数result2には200が格納される。
result2 = 50 + 150

' 変数result3には現在の日付が格納される。
result3 = Date
```

## 対象バージョン

JP1/Script 01-00 以降

## 7.1.2 Do...Loop

## 機能

指定した条件が真 (True) である間、または条件が真 (True) になるまで、一連のステートメントを繰り返し実行するフロー制御ステートメントです。

## 形式

```
Do
  { Statements }
[ Exit Do ]
{ Statements }
Loop [ { While | Until } Condition]
```

## 指定項目

*Statements*

*Condition* が真 (True) である間、または *Condition* が真 (True) になるまで繰り返し実行される一つ、または複数のステートメントを指定します。複数のステートメントを記述する場合は、ステートメントごとに改行します。

*Condition*

真 (True)、または偽 (False) を評価する条件式を指定します。

## 説明

## 7. ステートメント

キーワード *While* が記述されている間は *Condition* が真 (True) である間、一連のステートメントが繰り返し実行され、キーワード *Until* が記述されている場合は *Condition* が真 (True) になるまで、一連のステートメントが繰り返し実行されます。

*Exit Do* ステートメントは、制御構造 *Do ...Loop* 内だけで使われ、*Condition* で指定した以外の条件で *Do ...Loop* を終了させることができます。*Exit Do* ステートメントは、*Do ...Loop* 内の任意の場所に何回でも指定できます。*Exit Do* は条件の評価 (例えば *If...Then* ステートメントなど) とともに使われることが多く、*Loop* ステートメントの直後のステートメントに制御を渡します。

*Do ...Loop* ステートメントはネスト (入れ子) 構造にできます。つまり、*Do ...Loop* の内部に別の *Do ...Loop* を記述できます。ネスト構造の場合に *Exit Do* が実行されると、その *Exit Do* を囲んでいる最も内側のループから抜け出します。

*Do ...Loop* ステートメントは *While...End* ステートメントとは異なり、一連のステートメントが必ず一度は実行される後判定の制御構造しています。一連のステートメントが実行される前に条件の判定を行いたい場合は、*While...End* ステートメントを使用してください。

### 例

```
' Reverseは文字列を反転した値を返す。
' resultには"EDCBA"が格納される。
result = Reverse ( "ABCDE" )
MessageBox ( result )

Function Reverse ( chrValue )
    Dim chrString ,chrLength
    cnt = 0
    chrLength = Len ( chrValue )
    Do
        chrString = chrString + Mid ( chrValue ,chrLength - cnt , 1 )
        cnt = cnt + 1
    Loop While ( cnt < chrLength )
    Reverse = chrString
End Function
```

### 対象バージョン

JP1/Script 05-10 以降

## 7.1.3 For...Next

### 機能

指定した回数だけ、一連のステートメントを繰り返すフロー制御ステートメントです。

### 形式

```
For Counter = Start To End [Step Step]
    [Statements]
[Exit For]
[Statements]
```

Next

指定項目

*Counter*

ループカウンタに使う数値を格納する変数を指定します。変数の名前付け規則に従って名前を付けます。この変数には、配列変数や配列変数の一要素は指定できません。

*Start*

*Counter* の初期値を指定します。

*End*

*Counter* の最終値を指定します。

*Step*

ループを繰り返すごとに *Counter* に加算される値を指定します。*Step* を省略すると、ループを繰り返すごとに *Counter* に 1 が加算されます。*Counter* には正の数、または負の数を指定できます。*Step* で指定した値によりループの実行は次のように制御されます。

値	実行条件
正の数または 0	<i>Counter</i> <i>End</i>
負の数	<i>Counter</i> <i>End</i>

*Statements*

ループ内で実行される一連のステートメントを指定します。For と Next の間にこれらのステートメントを記述します。ここに記述したステートメントは、For...Next で指定した回数だけ実行されます。*Statements* には複数の文を記述できます。複数の文を記述する場合は、文ごとに改行します。

説明

プログラムの実行がループに移り、ループ内の一連のステートメントがすべて実行されると、*Step* の値を *Counter* に加算した値が終了条件を満たしていない場合は、*Step* の値を *Counter* に加算し、ループ内のステートメントが再び実行されます。終了条件を満たしている場合は、ループから抜け出して Next ステートメントの次のステートメントに制御が移ります。

Exit For ステートメントは、制御構造 For...Next と For...End For ループ内だけで使用できます。Exit For ステートメントを使用すると、指定した回数以外の条件で For...Next ループを終了させることができます。Exit For ステートメントは、For...Next ループ内の任意の位置で何回でも指定できます。Exit For ステートメントは、条件の評価（例えば If...Then ステートメントなど）とともに使われることが多く、Next の直後のステートメントに制御を渡します。

For...Next ループはネスト（入れ子）構造にすることができます。つまり、For...Next ループの内部に別の For...Next ループを入れることができます。ループ

## 7. ステートメント

をネスト（入れ子）させる場合は、それぞれの *Counter* に別の変数名を指定してください。

例

```
For I = 1 To 10
  For J = 1 To 10
    For K = 1 To 10
      :
    Next
  Next
Next
```

対象バージョン

JP1/Script 01-00 以降

### 7.1.4 For...End For

機能

指定したフォルダ内のすべてのファイルに対して、一連のステートメントを繰り返すフロー制御ステートメントです。

形式

```
For VarName = Filemask Do
  { Statements }
  [ Exit For ]
  { Statements }
End [ For ]
```

指定項目

*VarName*

ファイル名を格納する変数名を指定します。変数の名前付け規則に従って名前を付けます。配列変数や配列変数の一要素は指定できません。

*Filemask*

ファイル名をフルパスで指定します。ファイル名にはワイルドカードを含めることもできます。フルパス名は、ダブルクォーテーションで囲んでください。

*Statements*

ループ内で実行される一連のステートメントを指定します。For と End For の間にこれらのステートメントを記述します。ここに記述したステートメントは、For...End For で指定したパスに該当するファイルが見つからなくなるまで繰り返されます。

*Statements* には複数の文を記述できます。複数の文を記述する場合は、文ごとに改行します。

説明

指定したフォルダ内のすべてのファイルに対して、一連のステートメントを繰り返します。指定したパスに該当するファイルが見つからなくなった時、End For ステートメントの次のステートメントに制御が移ります。



Exit For ステートメントは、制御構造 For...End For と For...Next ループ内だけで使用できます。Exit For ステートメントを使用すると、直ちに For...End For ループを終了させることができます。Exit For ステートメントは、For...End For ループ内の任意の位置で何回でも指定できます。

For...End For ループはネスト（入れ子）構造にすることができます。つまり、For...End For ループの内部に別の For...End For ループを入れることができます。ループをネスト（入れ子）させる場合は、それぞれの *VarName* に別の変数名を指定してください。

*Filemask* に "\*.\*)" を指定した場合、*VarName* には ".", ".." 以外のファイル名が格納されます。

#### 補足

- For...End For ループのファイルの検索は、システムが互換性のために保存している MS-DOS ファイル形式に変換した短いファイル名も対象になります。
- For...End For ループのファイル検索順序は保証していません。

#### 例

・ 実行フォルダ下のスクリプトファイルを別フォルダにバックアップする。

```
Dim path1 ,bkupDir
bkupDir = _BIN_+"BKUP¥"
MakeDir ( bkupDir )

' スクリプトファイルを検索
For path1 = _BIN_+"*.SPT" Do
    ' フォルダは無視
    If IsExistDir ( _BIN_+path1 ) = False Then
        ' 別フォルダにバックアップ
        Copy ( _BIN_+path1 ,bkupDir+path1 )
    End If
End For
```

#### 対象バージョン

JP1/Script 01-00 以降

## 7.1.5 If...Then...Else

#### 機能

式の値に基づいて、条件付きの実行を行うフロー制御ステートメントです。

#### 形式 1

```
If Condition Then
    {Statements}
[Else
    {ElseStatements}]
End If
```

#### 形式 2

```
If Condition Then
    {Statements}
```

## 7. ステートメント

```
{ ElseIf Condition-n Then
  { ElseIfStatements } ...
{ Else
  { ElseStatements } }
End { If }
```

### 指定項目

#### *Condition*

真 (True) か偽 (False) を評価する条件式。

#### *Statements*

*Condition* が真 (True) の場合、実行される文です。

*Statements* には複数の文を記述できます。複数の文を記述する場合は、文ごとに改行します。

#### *Condition-n*

引数 *Condition* に同じ。

#### *ElseifStatements*

*Condition-n* が真 (True) の場合に実行される一連のステートメントを指定します。

*ElseifStatements* には複数の文を記述できます。複数の文を記述する場合は、文ごとに改行します。

#### *ElseStatements*

Else 以前に定義されている条件 (*Condition* または *Condition-n*) がどれも真 (True) でない場合に実行される一連のステートメントを指定します。

*ElseStatements* には複数の文を記述できます。複数の文を記述する場合は、文ごとに改行します。

### 説明

最初に *Condition* が評価されます。*Condition* が真 (True) の場合、Then に続くステートメントが実行されます。*Condition* が偽 (False) の場合、「形式 1」では Else 節がある場合 Else 節が実行されます。「形式 2」では ElseIf で指定した条件 (*Condition-n*) が評価されます。どちらかの条件が真 (True) の場合、対応する Then に続くステートメントが実行されます。ElseIf で指定された条件式がどちらも偽 (False) の場合 (または、ElseIf 節がないとき) は、Else に続くステートメントが実行されます。Then、または Else に続くステートメントの実行が終わると、End の次のステートメントからプログラムの実行が続けられます。

Else 節と ElseIf 節はどちらも必要に応じて定義します。また、「形式 2」の If では、ElseIf 節は幾つでも指定できます。ただし、Else 節の後ろに ElseIf 節を指定することはできません。また、If はネスト (入れ子) 構造にすることができます。

### 対象バージョン

JP1/Script 01-00 以降

## 7.1.6 Select Case

### 機能

条件式の値に従って、複数のステートメントブロックのどれかを実行させるフロー制御ステートメントです。

### 形式

```
Select Case TestExpression
  [ Case ExpressionList-n
    [ Statements-n ] ] ...
  [ Case Else
    [ ElseStatements-n ] ]
End [ Select ]
```

### 指定項目

*TestExpression*

任意の条件式を指定します。

*ExpressionList-n*

Case 節を記述する場合は必ず指定します。一つ、または複数の式をコンマ( , ) で区切って指定します。

*Statements-n*

*TestExpression* が *ExpressionList-n* のどれかと一致するとき、一致した *Statements-n* のステートメントが実行されます。*Statements-n* には複数の文を記述できます。複数の文を記述する場合は、文ごとに改行します。

*ElseStatements-n*

*TestExpression* がどの Case 節とも一致しないとき、*Elsestatements* のステートメントが実行されます。*Elsestatements-n* には複数の文を記述できます。複数の文を記述する場合は、文ごとに改行します。

### 説明

*TestExpression* が Case 節のどちらかの式 *ExpressionList* に一致すると、次の Case 節、または End Select ステートメントまでの *Statements* が実行されます。ブロックの実行が終わると、制御は End Select ステートメントの次のステートメントに移ります。*TestExpression* が複数の Case 節に一致するときは、最初に一致した Case 節に続くステートメントだけが実行されます。

Case Else 節は、*TestExpression* がどの Case 節の *ExpressionList* にも一致しなかったときに実行するステートメント *ElseStatements* を指定します。Case Else ステートメントは必ずしも必要ではありませんが、予測できない *TestExpression* の値を処理するために、Select Case ブロックに Case Else ステートメントを記述することを推奨します。Case 節の式 *ExpressionList* が *TestExpression* と一致しない場合で Case Else ステートメントが指定されていないときは、End Select ステートメントの次のステートメントから実行が続けられます。

Select Case ステートメントは、ネスト(入れ子)構造にすることができます。この

## 7. ステートメント

とき、各 Select Case ステートメントには、それぞれ対応する End Select ステートメントが必要です。

### 対象バージョン

JP1/Script 01-00 以降

### 7.1.7 While...End

#### 機能

指定した条件が真 (True) である間、一連のステートメントの実行を繰り返すフロー制御ステートメントです。

#### 形式

```
While Condition
  { Statements }
  [ Exit While ]
End { While }
```

#### 指定項目

##### *Condition*

真 (True)、または偽 (False) を評価する条件式を指定します。

##### *Statements*

*Condition* が真 (True) の間に実行する一つ、または複数のステートメントを指定します。複数のステートメントを記述する場合は、ステートメントごとに改行します。

#### 説明

*Condition* が真 (True) の場合は、End ステートメントに達するまで、*Statements* 内のすべてのステートメントが実行されます。実行が End ステートメントに達すると、制御は再び While ステートメントに戻り、*Condition* が評価されます。

*Condition* が真 (True) の間、この処理が繰り返されます。真 (True) でない場合は、End ステートメントの次のステートメントに制御が移ります。

Exit While ステートメントは、制御構造 While...End ループ内だけで使用できます。Exit While ステートメントを使用すると、指定した以外の条件で While...End ループを終了させることができます。Exit While ステートメントは、While...End ループ内の任意の位置で何回でも指定できます。

While...End ループは、任意のレベルでネスト (入れ子) 構造にすることができます。End ステートメントは最後に実行された While ステートメントに対応します。

### 対象バージョン

JP1/Script 01-00 以降

## 7.1.8 Function

### 機能

Function プロシージャの名前，引数を宣言し，Function プロシージャの始まりを示します。

### 形式

```
Function  Name { (ArgList) }
    { Statements }
    { Name = Expression }
    { Exit Function }
    { Statements }
    { Name = Expression }
End { Function }
```

### 指定項目

#### *Name*

定義する Function プロシージャの名前を指定します。変数の名前付け規則に従って名前を付けます。

#### *ArgList*

Function プロシージャを呼び出すときに，Function プロシージャに渡す引数を表す変数のリストを指定します。複数の変数を指定する場合はコンマ(，)で区切ります。引数がある場合には，引数を必ずカッコで囲んでください。

#### *Statements*

Function プロシージャ内で実行される一連のステートメントを指定します。

#### *Expression*

Function プロシージャの戻り値を指定します。

### 説明

Function プロシージャはパブリックプロシージャなので，スクリプト内の他のすべてのプロシージャからも参照できます。

実行可能なコードはすべてプロシージャ内に記述する必要があります。また，

Function プロシージャを他の Function プロシージャ，または Sub プロシージャの中で定義することはできません。

Exit Function ステートメントは，Function プロシージャを直ちに終了します。プログラムの実行は，その Function プロシージャを呼び出したステートメントの次のステートメントから継続されます。Exit Function ステートメントは，Function プロシージャ内の任意の位置で何回でも指定できます。

Function プロシージャの呼び出し方法については，Call ステートメントを参照してください。

Function プロシージャから値を返すには，値を Function プロシージャ名 *Name* に代入してください。*Name* への代入は，Function プロシージャ内の任意の場所が必要に応じて何回でも行うことができます。*Name* に値を代入しない場合は，既定の

## 7. ステートメント

戻り値が返されます。既定の戻り値は、長さ 0 の文字列 ("" ) です。

Function プロシージャで使う変数には、Function プロシージャ内で明示的に宣言される変数と、それ以外の変数の 2 種類があります。プロシージャ内で Dim コマンドなどを使用して明示的に宣言された変数は、そのプロシージャの中だけで有効なローカル変数になります。プロシージャ内で明示的に宣言しないで使われている変数も、そのプロシージャの外部のさらに上のレベルで明示的に宣言されていないかぎりには、ローカル変数となります。Function プロシージャ内のローカル変数の値は、プロシージャの実行が終了すると破棄されます。

プロシージャ内で明示的に宣言していない変数をプロシージャ内で使うことはできませんが、その変数と同じ名前の変数などがスクリプトレベルで定義されている場合、その変数はグローバルな変数として扱われます。

例

```
Function BinarySearch (...)  
:  
' 値が見つかりません。値としてFalseを返します。  
If lower > upper Then  
    BinarySearch = False  
    Exit Function  
End  
:  
End Function
```

対象バージョン

JP1/Script 01-00 以降

### 7.1.9 Sub

機能

Sub プロシージャの名前、引数を宣言し、Sub プロシージャの始まりを示します。

形式

```
Sub Name [ (ArgList) ]  
    [ Statements ]  
    [ Exit Sub ]  
    [ Statements ]  
End [ Sub ]
```

指定項目

*Name*

定義する Sub プロシージャの名前を指定します。変数の名前付け規則に従って名前を付けます。

*ArgList*

Sub プロシージャを呼び出すときに、Sub プロシージャに渡す引数を表す変数のリストを指定します。複数の変数を指定するときは、コンマ ( , ) で区切ります。引数がある場合には、引数を必ずカッコで囲んでください。

*Statements*

Sub プロシージャ内で実行される，一連のステートメントを指定します。

## 説明

Sub プロシージャはパブリックプロシージャなので，スクリプト内の他のすべてのプロシージャからも参照できます。

実行可能なコードはすべてプロシージャ内に記述する必要があります。また，Sub プロシージャを他の Sub プロシージャ，Function プロシージャの中で定義することはできません。

Exit Sub ステートメントは，Sub プロシージャを直ちに終了します。プログラムの実行は，その Sub プロシージャを呼び出したステートメントの次のステートメントから継続されます。Exit Sub ステートメントは，Sub プロシージャ内の任意の場所に必要に応じて幾つでも指定できます。

Sub プロシージャと Function プロシージャには，似た点と異なる点があります。引数を受け取って一連のステートメントを実行し，引数の値を変えることができる点は似ています。しかし，Sub プロシージャは，値を返す Function プロシージャと異なり，値を返さないため，式の中に記述することはできません。

Sub プロシージャを呼び出すには，プロシージャ名の後ろに引数リストを付けて指定します。Sub プロシージャの呼び出し方法については，Call ステートメントを参照してください。

Sub プロシージャで使う変数には，Sub プロシージャ内で明示的に宣言される変数と，それ以外の変数の 2 種類があります。プロシージャ内で Dim コマンドなどを使用して明示的に宣言された変数は，そのプロシージャの中だけで有効なローカル変数になります。プロシージャ内で明示的に宣言しないで使われている変数も，そのプロシージャの外部のさらに上のレベルで明示的に宣言されていないかぎりには，ローカル変数となります。Sub プロシージャ内のローカル変数の値は，プロシージャの実行が終了すると破棄されます。

プロシージャ内で明示的に宣言されていない変数をプロシージャ内で使うことはできますが，その変数と同じ名前の変数などがスクリプトレベルで定義されている場合，その変数はグローバルな変数として扱われます。

## 対象バージョン

JP1/Script 01-00 以降

## 7.1.10 Call

## 機能

Sub プロシージャ，および Function プロシージャに制御を渡すフロー制御ステートメントです。

## 形式

`[ Call ] Name [ (ArgumentList) ]`

## 7. ステートメント

### 指定項目

#### Call

省略可能なキーワードです。

#### Name

呼び出すプロシージャの名前を指定します。

#### ArgumentList

プロシージャに引き渡す変数リスト，または式のどちらかを指定します。

*ArgumentList* を複数指定するときは，コンマ ( , ) で区切ります。

### 説明

プロシージャを呼び出すとき，キーワード Call は省略できます。引数を必要とするプロシージャを呼び出す場合は，*ArgumentList* をカッコで囲んでも囲まなくてもかまいません。ただし，プロシージャを他のプロシージャやコマンドの引数として呼び出したり，式の途中で呼び出す場合には，引数を必ずカッコで囲んでください。Call ステートメントの構文で組み込み関数，またはユーザー定義型関数を呼び出す場合，その関数の戻り値を取得することはできません。Function プロシージャと Sub プロシージャで同じ名前が定義されている場合，Function プロシージャを優先して呼び出します。

### 例

```
Call MyProc ( 0 )   または   MyProc 0
```

### 対象バージョン

JP1/Script 01-00 以降

## 7.1.11 Exit xx

### 機能

While...End ループ，For...Next ループ，For...End For ループ，Do...Loop ループ，Function プロシージャ，または Sub プロシージャから抜け出すためのフロー制御ステートメントです。

### 形式

Exit While

Exit For

Exit Do

Exit Function

Exit Sub

### 説明

各 Exit ステートメントの機能は次のとおりです。



**Exit While**

While...End ループを抜け出し、End ステートメントの次のステートメントに制御を移します。While...End ステートメントの中だけで使用できます。

While...End ステートメントがネスト（入れ子）構造になっている場合は、Exit While のあるループの一つ外側のループに制御を移します。

**Exit For**

For...Next ループを抜け出し、Next ステートメントの次のステートメントに制御を移します。For...Next と For...End For ループの中だけで使用できます。

For...Next、または For...End For ループがネスト（入れ子）構造になっている場合は、Exit For ステートメントが含まれているループの一つ外側のループに制御を移します。

**Exit Do（05-10以降）**

Do...Loop ループを抜け出し、Loop ステートメントの次のステートメントに制御を移します。Do...Loop ステートメントの中だけで使用できます。Do...Loop ステートメントがネスト（入れ子）構造になっている場合は、Exit Do ステートメントのあるループの一つ外側のループに制御を移します。

**Exit Function**

このステートメントのある Function プロシージャを直ちに抜け出ます。制御は Function プロシージャを呼び出したステートメントの次のステートメントに移ります。

**Exit Sub**

このステートメントのある Sub プロシージャを直ちに抜け出ます。制御は Sub プロシージャを呼び出したステートメントの次のステートメントに移ります。

**対象バージョン**

JP1/Script 01-00 以降

## 7.1.12 GoTo

**機能**

指定したラベルに処理を分岐するステートメントです。

**形式**

GoTo *LabelName*

**指定項目**

*LabelName*

分岐先のラベル名を指定します。ラベル名には変数の名前付け規則に準じた名前を指定します。

**説明**

指定されたラベル先に実行の制御を移し、ラベルの次の行から実行が再開されます。

## 7. ステートメント

ラベル名に同位のレベルで同じ名前が複数個指定されている場合、文法エラーになります。

分岐先のラベルは任意の位置に記述することができますが、GoTo ステートメントがプロシージャの内部で指定されている場合、分岐先は同じプロシージャの内部に存在しなければなりません。

例

```
If Exec ("NOTEPAD.EXE" ,True ,"Logging.txt" ) = False
Then
    GoTo  ErrorExit
End
```

```
ErrorExit:      ' 分岐先
:
```

対象バージョン

JP1/Script 05-00 以降

### 7.1.13 Continue

機能

While...End ループ、For...Next ループ、For...End For ループ、Do...Loop ループの先頭に戻るためのフロー制御ステートメントです。

形式

Continue

説明

While...End ループ、For...Next ループ、For...End For ループ、Do...Loop ループの中でだけ記述することができます。それ以外の位置で記述した場合は文法エラーとなります。

Continue ステートメントは While...End ループ、For...Next ループ、For...End For ループ、Do...Loop ループ内の任意の位置で何回でも指定できます。

While...End ループ、For...Next ループ、For...End For ループ、Do...Loop ループがネスト（入れ子）構造になっている場合は、Continue ステートメントを含んでいる最も内側のループの先頭に制御を移します。

例

```
For I = 1 To 10
    For J = 1 To 10          ' ここに戻る
        :
        Continue
        :
    Next
Next
```

対象バージョン

JP1/Script 05-00 以降

## 7.1.14 On Error

### 機能

エラー発生時に、指定したラベルに制御を移すステートメントです。エラー発生時に、指定したラベルに制御を移さないようにすることもできます。

### 形式

`On Error GoTo LabelName`

### 指定項目

*LabelName*

分岐先のラベル名を指定します。ラベル名には変数の名前付け規則に準じた名前を指定します。

*LabelName* に 0 を指定すると、エラー発生時に、指定したラベルに制御が移らなくなります。

### 説明

On Error ステートメントは、ラベルの次の行から始まるエラー発生時の処理を有効にします。実行時にエラーが発生すると、ここで設定した分岐先に制御が移り、処理が実行されます。ただし、構文エラーが発生した場合には設定した分岐先には制御が移りません。分岐先のラベルは On Error ステートメントと同じプロシージャ内に存在しなければなりません。

On Error GoTo 0 を指定しなければ、ラベルの次の行から始まるエラー発生時の処理はプロシージャの終了時に自動的に無効になります。

### 補足

On Error ステートメントで指定したラベルに分岐する場合は、基本的に、実行エラーになりうるステートメントやコマンドが実行エラーとなったときです。

ただし、例外として以下の場合は実行エラーとなっても分岐しません。

- TextRead コマンドで「ファイルの終わりに達した」のエラー時
- WaitForExec コマンドで「タイムアウト時間を経過した」のエラー時

また、IsDefine コマンドや IsExistFile コマンドのように実行エラーになりえないコマンドはコマンドの戻り値にかかわらず分岐しません。

### 例

```
On Error GoTo ErrHandler '以降エラー発生時にラベルの次の行に制御が移る
Copy ( InDir+"CTL3D32.DLL" ,OutDir+"CTL3D32.DLL" ,VersionUp )
:
On Error GoTo 0 '以降エラー発生時にラベルの次の行に制御を移さない
:

ErrHandler:          'エラー発生時の分岐先
:
```

### 対象バージョン

JP1/Script 05-00 以降



# 8

## 基本コマンド

この章では、スクリプトを作成するときに使用できる基本コマンドについて説明します。

- 
- |      |                 |
|------|-----------------|
| 8.1  | 基本コマンド一覧        |
| 8.2  | 変数操作コマンド        |
| 8.3  | 文字列操作コマンド       |
| 8.4  | 日付操作コマンド        |
| 8.5  | ファイル・フォルダ操作コマンド |
| 8.6  | メッセージ出力コマンド     |
| 8.7  | メニュー呼び出しコマンド    |
| 8.8  | 演算処理コマンド        |
| 8.9  | チェック処理コマンド      |
| 8.10 | 外部プログラム呼び出しコマンド |
| 8.11 | 自動起動処理コマンド      |
| 8.12 | JP1 呼び出しコマンド    |
| 8.13 | コメントコマンド        |
| 8.14 | その他のコマンド        |
-

## 8.1 基本コマンド一覧

### 8.1.1 基本コマンド一覧

スクリプトを作成するときに使用できる，基本コマンドの一覧を表 8-1 に示します。なお，コマンド欄の括弧内の数字は，対象バージョンを表します（括弧がないものは，01-00 からサポートしています）。

表 8-1 基本コマンド一覧

分類	コマンド	意味
変数操作	Dim	変数を宣言してメモリ領域を割り当てます。
	Dim ( 配列 ) ( 06-00 以降 )	配列変数を宣言してメモリ領域を割り当てます。
	SetEnvironment または SetEnv	環境変数を設定します。
	GetEnvironment または GetEnv	環境変数を取得します。
	SetGV	自 PC，およびネットワーク上の他 PC のグローバル変数を設定します。
	GetGV	自 PC，およびネットワーク上の他 PC のグローバル変数を取得します。
	DeleteGV ( 06-00 以降 )	自 PC，およびネットワーク上の他 PC のグローバル変数を削除します。
	GetArrayCount ( 06-00 以降 )	配列変数の要素数（二次元配列の場合は行数または列数）を取得します。
文字列操作	InStr ( 05-00 以降 )	文字列の中から指定された文字列を検索し，最初に見つかった文字位置（先頭からその位置までの文字数）を返します。
	InArray ( 06-00 以降 )	配列変数の要素の中から指定された文字列を検索し，最初に見つかったインデックス番号を返します。
	Len	文字列の文字数を返します。
	Lcase	文字列の半角英字の大文字を小文字に変換します。
	Ucase	文字列の半角英字の小文字を大文字に変換します。
	Left	文字列の左端から指定された文字数分の文字列を返します。

分類	コマンド	意味
	Mid	文字列から指定された文字数分の文字列を返します。
	Right	文字列の右端から指定された文字数分の文字列を返します。
	Space	指定された数の半角スペースからなる文字列を返します。
	Ltrim ( 05-00 以降 )	文字列から先頭のスペースを削除した文字列を返します。
	Rtrim ( 05-00 以降 )	文字列から末尾のスペースを削除した文字列を返します。
	Trim ( 05-00 以降 )	文字列から先頭, および末尾のスペースを削除した文字列を返します。
	+ ( 文字列連結 )	二つの式に対して文字列連結を行います。
	& ( 文字列連結 ) ( 06-00 以降 )	二つの式に対して文字列連結を行います。
	&= ( 文字列連結 ) ( 06-51 以降 )	変数と式の値に対して文字列連結を行い, 連結した文字列を変数に代入します。
	AddStr ( 05-00 以降 )	指定した二つ以上の文字列に対して, 指定した区切り文字を挿入し連結した文字列を返します。
	SeparateStrCount ( 05-00 以降 )	指定した区切り文字によって分割された文字列の数を返します。
	SeparateStr ( 05-00 以降 )	指定した区切り文字によって分割された文字列を返します。
	Str ( 05-20 以降 )	指定した値を文字列で返します。
	Format ( 05-20 以降 )	指定した値を書式化した文字列で返します。
	IsLower ( 06-00 以降 )	文字列が半角英字の小文字かどうかを調べ, 結果を真 ( True ) または偽 ( False ) で返します。
	IsUpper ( 06-00 以降 )	文字列が半角英字の大文字かどうかを調べ, 結果を真 ( True ), または偽 ( False ) で返します。
	IsSingleChar ( 06-00 以降 )	文字列が半角文字かどうかを調べ, 結果を真 ( True ), または偽 ( False ) で返します。
	IsMultiChar ( 06-00 以降 )	文字列が全角文字かどうかを調べ, 結果を真 ( True ), または偽 ( False ) で返します。
日付操作	Date	現在の日付を返します。

## 8. 基本コマンド

分類	コマンド	意味
	Time	現在の時刻を返します。
	Year	指定された日付の年を表す値を、4桁の数値で返します。
	Month	指定された日付の月を表す値を、1～12の範囲の2桁の数値で返します。
	Day	指定された日付の日を表す値を、1～31の範囲の2桁の数値で返します。
	Weekday	指定された日付の曜日を表す値を、1(日曜)～7(土曜)の範囲の1桁の数値、または曜日を表す文字列で返します。
	Hour	指定された時刻の時を表す値を、0～23の範囲の2桁の数値で返します。
	Minute	指定された時刻の分を表す値を、0～59の範囲の2桁の数値で返します。
	Second	指定された時刻の秒を表す値を、0～59の範囲の2桁の数値で返します。
	CalcDate ( 05-20 以降 )	指定した日付に、指定した値を加算、または減算した結果(日付)を返します。
	CompDate ( 05-20 以降 )	指定した二つの日付を比較し、結果を真( True ), または偽( False )で返します。
	GetDateCount ( 05-20 以降 )	指定した二つの日付の経過日数を返します。
	CalcTime ( 05-20 以降 )	指定した時刻に、指定した値を加算、および減算した結果(時刻)を返します。
	CompTime ( 05-20 以降 )	指定した二つの時刻を比較し、結果を真( True ), または偽( False )で返します。
	GetTimeCount ( 05-20 以降 )	指定した二つの時刻の経過時間を返します。
ファイル・フォルダ操作	IsLeapYear ( 05-20 以降 )	指定した西暦がうるう年かどうかを調べ、結果を真( True ), または偽( False )で返します。
	IniRead	初期化ファイルから値を読み込みます。
	IniWrite	初期化ファイルの値を設定します。



分類	コマンド	意味
	TextFileReplace	テキストファイルの中の特定の文字列を置き換えます。
	TextOpen ( 05-00 以降 )	テキスト形式ファイルをオープンします。
	TextClose ( 05-00 以降 )	テキスト形式ファイルをクローズします。
	TextRead ( 05-00 以降 )	テキスト形式ファイルの 1 行分のデータを読み込みます。
	TextWrite ( 05-00 以降 )	テキスト形式ファイルにデータを書き込みます。
	TextSeek ( 05-00 以降 )	テキスト形式ファイルのデータの読み書き開始位置を移動します。
	GetTextPosition ( 05-00 以降 )	テキスト形式ファイルの現在の読み書き開始位置を返します。
	MakeDir	フォルダを作成します。
	DeleteDir	フォルダを削除します。
	DeleteFile	ファイルを削除します。
	Rename	ファイル名を変更します。
	TempDir	一時フォルダを取得します。
	TempFile	一時ファイル名を取得します。
	SetFileAttribute または SetFileAttr	ファイルの属性を設定します。
	GetFileAttribute または GetFileAttr ( 06-00 以降 )	ファイルの属性を取得します。
	SetFileTime ( 05-10 以降 )	ファイルに日付と時刻を設定します。
	GetFileTime ( 05-10 以降 )	ファイルの日付と時刻を取得します。
	GetFileSize ( 06-00 以降 )	ファイルの容量を取得します。
	GetVersionInfo または GetVerInfo ( 06-00 以降 )	ファイルのバージョン情報を取得します。
	SplitFile ( 05-10 以降 )	指定したサイズでファイルを分割します。
	CatFiles ( 05-10 以降 )	複数のファイルを一つに統合します。
	SetStandardFile または SetStdFile ( 05-10 以降 )	Exec コマンドで呼び出すプロセスの標準入力, 標準出力, および標準エラーファイルを設定します。
	ResetStandardFile または ResetStdFile ( 05-10 以降 )	Exec コマンドで呼び出すプロセスの標準入力, 標準出力, および標準エラーファイルを解除します。
	SplitPath	フルパスを解析します。

## 8. 基本コマンド

分類	コマンド	意味
	MakePath	フルパスを作成します。
	SetPath	実行フォルダのパスを設定します。
	GetPath	実行フォルダのパスを取得します。
	SetVolumeLabel または SetVolLabel ( 05-10 以降 )	ディスクのボリュームラベルを設定します。
	GetVolumeLabel または GetVolLabel ( 05-10 以降 )	ディスクのボリュームラベルを取得します。
	GetDiskFreeSpace ( 05-10 以降 )	ディスクの空き容量を取得します。
	Copy	ファイルをコピーします。
メッセージ出力	InputBox	ダイアログボックスにメッセージとテキストボックスを表示し、テキストが入力されるか、またはボタンがクリックされると、テキストボックスの内容を返します。
	Message	ファイル、またはウィンドウへ指定されたテキストを出力します。また、テキストを出力したウィンドウを消去します。
	MessageBox	ダイアログボックスに指定されたメッセージを表示します。ボタンやアイコンの付加を指定できます。ボタンを付加する場合、ボタンがクリックされるのを待って、どのボタンがクリックされたのかを示す値を返します。
	MessageEventLog ( 01-01 以降 )	イベントビューアのアプリケーションログにメッセージを出力します。
	IMEventMessage ( 05-20 以降 )	JP1/IM、または JP1/Base ヘイイベントを発行します。JP1/IM のイベントコンソールについては、マニュアル「JP1 Version 8 JP1/Integrated Management - Manager システム構築・運用ガイド」を参照してください。
メニュー呼び出し	Menu ( 05-20 以降 )	ユーザ定義の画面 ( メニューフォーム ) を表示します。
演算処理	+ 演算子 ( 加算 )	二つの数値の和を求めます。
	+ = 演算子 ( 加算 ) ( 06-51 以降 )	変数と式の値の和を求め、結果を変数に代入します。
	- 演算子 ( 減算 )	二つの数値の差を求めます。数式の符号を反転した値を指定します。
	- = 演算子 ( 減算 ) ( 06-51 以降 )	変数と式の値の差を求め、結果を変数に代入します。

分類	コマンド	意味
	Mod 演算子 (剰余演算)	二つの数値の除算を行い、その剰余を返します。
	Mod= 演算子 (剰余演算) (06-51 以降)	変数と式の値の除算を行い、その剰余を変数に代入します。
	* 演算子 (乗算)	二つの数値の積を求めます。
	*= 演算子 (乗算) (06-51 以降)	変数と式の値の積を求め、結果を変数に代入します。
	/ 演算子 (除算)	二つの数値の商を計算し、結果を整数で返します。
	/= 演算子 (除算) (06-51 以降)	変数と式の値の商を計算し、結果を整数で変数に代入します。
	¥ 演算子 (整数除算)	二つの数値の商を計算し、結果を整数で返します。
	¥= 演算子 (整数除算) (06-51 以降)	変数と式の値の商を計算し、結果を整数で変数に代入します。
	^ 演算子 (べき乗) (06-00 以降)	二つの数値のべき乗を求めます。
	^= 演算子 (べき乗) (06-51 以降)	変数と式の値のべき乗を求め、結果を変数に代入します。
	比較演算子 (=, <>, <, <=, >, >=)	二つの式を比較します。
	論理積 (And)	二つの式の論理積を求めます。
	論理和 (Or)	二つの式の論理和を求めます。
	論理否定 (Not) (06-00 以降)	式の論理否定を求めます。
チェック処理	IsEmpty	変数が Empty 値かどうかを調べ、結果を真 (True) または偽 (False) で返します。
	IsDefine または IsDef	変数が定義されているかどうかを調べ、結果を真 (True), または偽 (False) で返します。
	IsNumeric (01-01 以降)	値が数値として評価できるかどうかを調べ、結果を真 (True), または偽 (False) で返します。
	IsEmptyDir	フォルダの中身が空かどうかをチェックし、結果を真 (True), または偽 (False) で返します。
	IsExistDir	フォルダが存在するかどうかをチェックし、結果を真 (True), または偽 (False) で返します。
	IsExistFile	ファイルが存在するかどうかをチェックし、結果を真 (True), または偽 (False) で返します。

## 8. 基本コマンド

分類	コマンド	意味
	IsWriteableDir	フォルダが書き込み可能かどうかをチェックし、結果を真 (True), または偽 (False) で返します。
	IsFileAttribute または IsFileAttr	ファイルの属性が指定された属性か、どうかをチェックし、結果を真 (True), または偽 (False) で返します。
	IsNew ( 05-10 以降 )	二つのファイルのバージョン情報, またはファイル日付の新旧を比較し、結果を真 (True), または偽 (False) で返します。
	CheckDirName	フォルダの末尾 ( ¥ ) かどうかをチェックします。
	CheckDriveType	ドライブの種類をチェックします。
外部プログラム呼び出し	Exec	実行ファイルを呼び出します。複数パラメタの指定が可能です。
	NetExec ( 05-00 以降 )	自 PC, およびネットワーク上の他の PC 上の実行ファイルを呼び出します。複数パラメタの指定が可能です。
	WaitForExec ( 01-01 以降 )	Exec, および NetExec コマンドで呼び出した実行ファイルの終了待ちや強制終了を行います。
	GetExecStatus ( 06-00 以降 )	Exec, および NetExec コマンドで呼び出した実行ファイルの現在の実行状態を取得します。
	CallSpt ( 06-71 以降 )	現在のプロセスが SPT ファイルをプロシージャのように呼び出します。複数パラメタの指定が可能です。
自動起動処理	EntryStartUp ( 05-10 以降 )	スクリプトファイルを自動起動に登録します。
	CancelStartUp ( 05-10 以降 )	スクリプトファイルの自動起動の登録を解除します。
JP1 呼び出し	JOBSubmit	JP1 のキューサーバに対してスクリプトを登録します。
	JOBWait	JOBSubmit コマンドで登録したジョブが終了するのを待ちます。
	JOBHold	JOBSubmit コマンドで登録したジョブを保留, または保留解除にします。
	JOBCancel	JOBSubmit コマンドで登録したジョブをキャンセルします。
コメント	Rem または 「    」 ( シングル引用符 )	プログラム内にコメントを記述するときに指定します。

分類	コマンド	意味
その他	Sleep	指定された時間、処理を中断します。
	Alert	トレースビューアのステータスに、警告・警告解除を知らせます。
	Beep	スピーカからビープ音を鳴らします。
	Exit	スクリプトの実行を終了します。
	GetErrorMessage ( 01-01 以降 )	指定されたエラー詳細コードのエラーメッセージを返します。

注 Exec , NetExec , および CallSpt コマンドの違いを表 8-2 に示します。

表 8-2 Exec , NetExec , および CallSpt コマンドの違い

コマンド	呼び出せる実行ファイル	呼び出した実行ファイルの終了待ち	呼び出された外部ファイルのプロセス		用途
			特徴	プロセス環境変数	
Exec	EXE ファイル , BAT ファイル , COM ファイル , SPT ファイル , CMD ファイル , 関連付けられたファイル	終了待ちをするか しないかを指定 できます。	新しいプロセス で実行され ます。	呼び出し側のコピー を継承するので、呼 び出し側の値は認識 できるが、それを更 新しても呼び出し側 では反映されませ ん。	実行ファイル を呼び出す場 合
NetExec	EXE ファイル , BAT ファイル , COM ファイル , SPT ファイル , CMD ファイル , 関連付けられた ファイル	終了待ち をするか しないかを 指定でき ます。	新しいプロセス で実行され ます。	呼び出し側のコピー を継承するので、呼 び出し側の値は認識 できるが、それを更 新しても呼び出し側 では反映されませ ん。	他のコン ピュータで実 行ファイル を呼び出す 場合や、実 行する空間 をサービス かログオン かを指定し て実行ファ イルを呼び 出す場合
CallSpt	SPT ファイル	指定でき ません ( 常に終 了待ちと なりま す )	現在のプロセス で実行され ます。	呼び出し側と共有 するため、参照 できます。また、 更新した値を呼 び出し側で参照 できます。	現在の処理の 一部として実 行ファイルを 呼び出す場合

## 8.2 変数操作コマンド

---

### 8.2.1 Dim ( 変数を宣言しメモリ領域を割り当てる )

#### 機能

変数を宣言してメモリ領域を割り当てます。

#### 形式

```
Dim VarName [ , VarName , ... ]
```

#### 指定項目

*VarName*

定義する変数名を指定します。変数の名前付け規則に従って名前を付けます。

#### 説明

このコマンドを使ってスクリプトレベルで宣言した変数は、宣言したスクリプト内のすべてのプロシージャから参照できます。プロシージャレベルで宣言した変数は、宣言したプロシージャ内だけで参照できます。

ただし、スクリプト、またはプロシージャの Do...Loop , For...Next , For...End For , While...End ループ内で変数の宣言、または宣言していない変数への代入により割り当てた変数はループ内だけで参照できます。

Dim コマンドで宣言していない変数に値を代入した場合は、自動的に変数を宣言して、メモリ領域を割り当て、値を代入します。

変数の宣言時には、変数は Empty 値になります。

#### 対象バージョン

JP1/Script 01-00 以降

### 8.2.2 Dim ( 配列 ) ( 配列変数を宣言しメモリ領域を割り当てる )

#### 機能

配列変数を宣言してメモリ領域を割り当てます。

#### 形式

```
Dim VarName ( [ Number1 ] [ , Number2 ] ) [ , VarName ( [ Number1 ]
[ , Number2 ] ) , ... ]
Dim VarName [ [ Number1 ] [ , Number2 ] ] [ , VarName [
[ Number1 ] [ , Number2 ] ] , ... ]
```

#### 指定項目

*VarName*

定義する配列変数の変数名を指定します。変数の名前付け規則に従って名前を付けます。

*Number1*

配列変数の要素数（二次元配列の場合は行要素数）を数値、または値を格納した変数で指定します。

省略した場合、要素数は可変となります。

*Number2*

二次元配列の列要素数を数値、または値を格納した変数で指定します。

省略した場合、要素数は可変となります。

## 説明

このコマンドを使ってスクリプトレベルで宣言した配列変数は、宣言したスクリプト内のすべてのプロシージャから参照できます。プロシージャレベルで宣言した配列変数は、宣言したプロシージャ内だけで参照できます。

ただし、スクリプトレベル、またはプロシージャレベルの Do...Loop, For...Next, For...End For, While...End ループ内で配列変数の宣言により割り当てた配列変数はループ内だけで参照できます。

配列変数の宣言時には、配列変数のすべての要素は Empty 値になります。

## 例

```
' 二次元の固定な配列変数table1の宣言
Dim table1( 5 ,6 )

' 一次元の固定な配列変数table2から一次元の可変な配列変数table3への
' 一括代入
Dim table2( 6 ) ,table3( )
For i = 1 To 6
    table2( i ) = Time( )
Next
table3 = table2

' 一次元の固定な配列変数table4と配列変数table5の比較
Dim table4( 6 ) ,table5( 6 )
:
If table4 = table5 Then
:
' 次のような大小比較はできないため、実行エラーになる。
If table4 <= table5 Then
:
:
```

## 対象バージョン

JP1/Script 06-00 以降

## 8.2.3 SetEnvironment または SetEnv （環境変数を設定する）

## 機能

環境変数を設定します。

## 形式

```
SetEnvironment ( Type , EnvironmentName [ , Value ] )
```

## 8. 基本コマンド

`SetEnv ( Type , EnvironmentName [ , Value ] )`

### 指定項目

#### *Type*

設定する環境変数がシステム環境変数かユーザ環境変数か、または現在のプロセスの環境変数かを指定します。  
この値は次のどれかになります。

値	意味
SystemEnv	システム環境変数 設定する環境変数はコマンド実行後のシステム再起動後から有効になります。ただし、GetEnvironment コマンドではコマンド実行後から取得できます。
UserEnv	ユーザ環境変数 設定する環境変数はコマンド実行後のシステム再起動後から有効になります。ただし、GetEnvironment コマンドではコマンド実行後から取得できます。
ProcessEnv	現在のプロセスの環境変数 設定する環境変数はコマンド実行後の現在のプロセス内で有効になります。

SystemEnv を指定した場合、Windows Vista では管理者権限でスクリプトファイルを実行してください。詳細は、「1.8.4 コマンドの動作」を参照してください。

#### *EnvironmentName*

設定する環境変数名を文字列、または値を格納した変数名で指定します。

#### *Value*

設定する値を文字列、または値を格納した変数名で指定します。  
この値は省略できます。省略した場合、長さ 0 の文字列 ("" ) を仮定します。  
この値に展開前の環境変数への参照 ("%PATH%" など) を含んでいてもそのままの値で設定します。  
例えば *Value* で指定された値が "%PATH%Data" の場合、"%PATH%Data" を設定します。しかし、この環境変数を参照するときに、システムは %PATH% の部分を環境変数 PATH の現在の値に置き換えた値で取得します。  
ただし、プロセス環境変数の場合、現在の値に置き換えられませんので注意してください。環境変数 PATH が見つからない場合 "%PATH%" の部分はそのまま取得します。

### 説明

指定された環境変数に指定された値を設定します。コマンドが正常に実行された場合は真 (True) を、エラーが発生した場合は偽 (False) を、コマンドの実行結果として返します。

指定された環境変数が存在しない場合は、新規に環境変数を作成し値を設定します。環境変数が存在する場合は、指定された値で更新します。

### 例

```
SetEnvironment ( ProcessEnv , "I1_Path" , "C:¥ABC¥" )
```



Exec ( "ABC.EXE" , True )

対象バージョン

JP1/Script 01-00 以降

## 8.2.4 GetEnvironment または GetEnv （環境変数を取得する）

機能

環境変数を取得します。

形式

GetEnvironment ( Type , EnvironmentName [ , Option ] )  
GetEnv ( Type , EnvironmentName [ , Option ] )

指定項目

*Type*

取得する環境変数がシステム環境変数かユーザ環境変数か、または現在のプロセスの環境変数かを指定します。  
この値は次のどれかになります。

値	意味
SystemEnv	システム環境変数
UserEnv	ユーザ環境変数
ProcessEnv	現在のプロセスの環境変数

*EnvironmentName*

取得する環境変数を文字列、または値を格納した変数名で指定します。

*Option* ( 05-20 以降 )

オプションを指定します。  
この値には次の値を指定できます。

値	意味
Expand	取得する値に展開前の環境変数への参照 ("%PATH%" など) を含む場合、展開した値で取得します。

説明

指定された環境変数を取得し、値をコマンドの実行結果として返します。  
指定された環境変数が存在しない場合は、長さ 0 の文字列 ("") を返します。

**!** 注意事項

取得する環境変数の値が、長さ 1025 バイト以上の文字列の場合、コマンドの実行結果として返される値は 1024 バイト目までです。1025 バイト目以降は無視されます。

## 例

```
' 環境変数pathの値が"C:¥Winnt"の場合、変数buff1と変数buff2
'  には"C:¥Winnt"が格納される。
Dim buff1 ,buff2
buff1 = GetEnvironment ( SystemEnv ,"path" )
buff2 = GetEnvironment ( SystemEnv ,"path" ,Expand )

' 環境変数pathの値が"%SystemRoot%"で 環境変数SystemRoot
'   の値が"C:¥Winnt"の場合、
'   変数buff3には"%SystemRoot%"が、変数buff4には"C:¥Winnt"が
'   格納される。
Dim buff3 ,buff4
buff3 = GetEnvironment ( SystemEnv ,"path" )
buff4 = GetEnvironment ( SystemEnv ,"path" ,Expand )
```

## 対象バージョン

JP1/Script 01-00 以降

## 8.2.5 SetGV（グローバル変数を設定する）

## 機能

グローバル変数を設定します。

JP1/Script 05-00 以降では、他のコンピュータ上のグローバル変数も設定できます。

## 形式

```
SetGV ( GlobalName , [ Value ] [ , CompName ] )
```

## 指定項目

*GlobalName*

設定するグローバル変数を文字列、または値を格納した変数名で指定します。

*Value*

設定する値を文字列、数値、または値を格納した変数名で指定します。

この値は省略できます。省略した場合、長さ 0 の文字列 ("" ) を仮定します。

*CompName* ( 05-00 以降 )

設定するグローバル変数が存在するコンピュータのコンピュータ名を文字列、または値を格納した変数名で指定します。

この値は省略できます。省略した場合、このコマンドを実行するコンピュータのコンピュータ名を仮定します。

なお、指定する他のコンピュータの OS は Windows NT , Windows 2000 , および Windows XP で、JP1/Script がインストールされ、JP1/Script サービスが開始されていないとなりません。また、アクセスするためのアカウントについ

ては「2.2 他のコンピュータとの通信時のアカウント」を参照してください。

#### 説明

指定されたグローバル変数に指定された値を設定します。コマンドが正常に実行された場合は真 (True) を、エラーが発生した場合は偽 (False) を、コマンドの実行結果として返します。

指定されたグローバル変数が存在しない場合は、新規にグローバル変数を作成し値を設定します。グローバル変数が存在する場合は、指定された値で更新します。

*CompName* に他のコンピュータを指定し、「グローバル変数」、および「設定する値」に  $n$  個のダブルクォーテーション (") を文字としてそのまま含めるには、 $(n*8)$  個のダブルクォーテーション (") を指定しなければなりません。ただし、次のレジストリの値に 1 を設定した場合は、 $(n*2)$  個のダブルクォーテーション (") を指定してください。

#### 〔レジストリキー〕

HKEY\_LOCAL\_MACHINE¥SOFTWARE¥Hitachi¥JP1/Script¥SPTX

#### 〔値名〕

ParseDQ (DWORD)

#### 〔値〕

1: パラメタ文字列に  $n$  個のダブルクォーテーション (") を含む場合、 $(n*2)$  個のダブルクォーテーション (") を指定する。

なお、〔値〕に 1 と 2 (SeparateStrCount, SeparateStr コマンドを参照) の両方を指定する場合は 3 を指定してください。

#### 補足

このコマンドは、インストールフォルダの "DATA" フォルダ下に、グローバル変数ファイル (SPTGV.SPG) を作成します。ただし、JP1/Script 06-51 以降では、マネージャの [ ツール ] - [ オプション (クラスタ環境) ] ダイアログで管理ファイルの出力先が変更されている場合、指定されたフォルダ下にグローバル変数ファイルを作成します。このファイルはスクリプトの実行が終了しても残りますので、グローバル変数を初期化する場合は、ファイルを削除してください。

また、*CompName* に他のコンピュータを指定する場合、LAN 環境下での動作だけを保証しています。ファイアウォールには対応していません。

#### 例

・ 自コンピュータ上のグローバル変数を設定する。

```
SetGV ( "Debug_Mode" , True )
```

・ コンピュータ "SOP4A065 ( SCRIPT )" 上のグローバル変数を設定する。

```
SetGV ( "Debug_Mode" , True , "SOP4A065 ( SCRIPT )" )
```

#### 対象バージョン

JP1/Script 01-00 以降

## 8.2.6 GetGV ( グローバル変数を取得する )

### 機能

グローバル変数を取得します。

JP1/Script 05-00 以降では、他のコンピュータ上のグローバル変数も取得できます。

### 形式

GetGV ( *GlobalName* [ , *CompName* ] )

### 指定項目

*GlobalName*

取得するグローバル変数を文字列、または値を格納した変数名で指定します。

*CompName* ( 05-00 以降 )

取得するグローバル変数が存在するコンピュータのコンピュータ名を文字列、または値を格納した変数名で指定します。

この値は省略できます。省略した場合、このコマンドを実行するコンピュータのコンピュータ名を仮定します。

なお、指定する他のコンピュータの OS は Windows NT、Windows 2000、および Windows XP で、JP1/Script がインストールされ、JP1/Script サービスが開始されていなければなりません。また、アクセスするためのアカウントについては「2.2 他のコンピュータとの通信時のアカウント」を参照してください。

### 説明

指定されたグローバル変数を取得し、値をコマンドの実行結果として返します。

指定されたグローバル変数が存在しない場合は、長さ 0 の文字列 ( "" ) を返します。

*CompName* に他のコンピュータを指定し、「グローバル変数」、および「設定する値」に *n* 個のダブルクォーテーション ( " ) を文字としてそのまま含めるには、( *n*\*8 ) 個のダブルクォーテーション ( " ) を指定しなければなりません。ただし、次のレジストリの値に 1 を設定した場合は、( *n*\*2 ) 個のダブルクォーテーション ( " ) を指定してください。

[ レジストリキー ]

HKEY\_LOCAL\_MACHINE¥SOFTWARE¥Hitachi¥JP1/Script¥SPTX

[ 値名 ]

ParseDQ (DWORD)

[ 値 ]

1 : パラメタ文字列に *n* 個のダブルクォーテーション ( " ) を含む場合、( *n*\*2 ) 個のダブルクォーテーション ( " ) を指定する

なお、[ 値 ] に 1 と 2 ( SeparateStrCount、SeparateStr コマンドを参照 ) の両方を指定する場合は 3 を指定してください。

### 補足

*CompName* に他のコンピュータを指定する場合、LAN 環境下での動作だけを保証

しています。ファイアウォールには対応していません。

例

```
' 自コンピュータ上のグローバル変数を取得し、レジストリを操作する。
If GetGV ( "Debug_Mode" ) = True Then
  Dbflag = "DebugFlag"
  RegWrite ( HKEY_CURRENT_USER _
    , "Software¥Hitachi¥Script¥Option" , Dbflag , True )
End
' コンピュータ"SOP4A065 ( SCRIPT ) "上のグローバル変数を取得し、
' 更新する。
If GetGV ( "Debug_Mode" , "SOP4A065 ( SCRIPT ) " ) = True Then
  SetGV ( "Debug_Mode" , False , "SOP4A065 ( SCRIPT ) " )
End
```

対象バージョン

JP1/Script 01-00 以降

8.2.7 DeleteGV （グローバル変数を削除する）

機能

グローバル変数を削除します。  
他のコンピュータ上のグローバル変数も削除できます。

形式

```
DeleteGV ( GlobalName [ , CompName ] )
```

指定項目

*GlobalName*

削除するグローバル変数を文字列、または値を格納した変数名で指定します。  
また、この値には次の値を指定できます。

値	意味
AllGV	すべてのグローバル変数を削除します。ただし、グローバル変数ファイル（SPTGV.SPG）は残ります。

*CompName*

削除するグローバル変数が存在するコンピュータのコンピュータ名を文字列、または値を格納した変数名で指定します。  
この値は省略できます。省略した場合、このコマンドを実行するコンピュータのコンピュータ名を仮定します。  
なお、指定する他のコンピュータの OS は Windows NT、Windows 2000、および Windows XP で、JP1/Script がインストールされ、JP1/Script サービスが開始されていない必要ありません。また、アクセスするためのアカウントについては「2.2 他のコンピュータとの通信時のアカウント」を参照してください。

説明

指定されたグローバル変数を削除します。コマンドが正常に実行された場合は真 (True) を、エラーが発生した場合は偽 (False) を、コマンドの実行結果として返します。

指定されたグローバル変数が存在しない場合は、何もしないで常に真 (True) を返します。

CompName に他のコンピュータを指定し、「グローバル変数」、および「設定する値」に  $n$  個のダブルクォーテーション (") を文字としてそのまま含めるには、 $(n*8)$  個のダブルクォーテーション (") を指定しなければなりません。ただし、次のレジストリの値に 1 を設定した場合は、 $(n*2)$  個のダブルクォーテーション (") を指定してください。

〔レジストリキー〕

HKEY\_LOCAL\_MACHINE¥SOFTWARE¥Hitachi¥JP1/Script¥SPTX

〔値名〕

ParseDQ (DWORD)

〔値〕

1: パラメタ文字列に  $n$  個のダブルクォーテーション (") を含む場合、 $(n*2)$  個のダブルクォーテーション (") を指定する

なお、〔値〕に 1 と 2 (SeparateStrCount, SeparateStr コマンドを参照) の両方を指定する場合は 3 を指定してください。

補足

CompName に他のコンピュータを指定する場合、LAN 環境下での動作だけを保証しています。ファイアウォールには対応していません。

例

・自コンピュータ上のグローバル変数をすべて削除する。

DeleteGV ( AllGV )

・コンピュータ"SOP4A065 (SCRIPT)"上のグローバル変数"prm01"を削除する。

SetGV ( "prm01" ,10 , "SOP4A065 ( SCRIPT ) " )

Exec ( \_SCF\_+"ABC.SPT" ,True )

DeleteGV ( "prm01" , "SOP4A065 ( SCRIPT ) " )

対象バージョン

JP1/Script 06-00 以降

## 8.2.8 GetArrayCount ( 配列変数の要素数を取得する )

機能

配列変数の要素数 ( 二次元配列の場合は行数または列数 ) を取得します。

形式

GetArrayCount ( ArrayName )

指定項目

*ArrayName*

配列変数を変数名で指定します。

## 説明

指定された配列変数の要素数（二次元配列の場合は行数または列数）を取得します。コマンドが正常に実行された場合は要素数（二次元配列の場合は行数または列数）を、エラーが発生した場合は長さ 0 の文字列（""）を、コマンドの実行結果として返します。

指定された配列変数が可変である場合は、すでに設定された要素数（二次元配列の場合は行数または列数）を返します。

## 例

```
' [ 一次元の固定な配列変数Aの場合 ]
' 変数result1には要素数10が格納される。
Dim A( 10 )
result1 = GetArrayCount ( A )

' [ 二次元の固定な配列変数Bの場合 ]
' 変数result2と変数result3には列数10が、変数result4には
' 行数5が格納される。
Dim B( 5 ,10 )
result2 = GetArrayCount ( B( 1 ) )
result3 = GetArrayCount ( B( 5 ) )
result4 = GetArrayCount ( B )

' [ 二次元の可変な配列変数Cの場合 ]
' 変数result5には列数1が、変数result6には列数7が、変数result7
' には行数2が格納される。
Dim C( , )
C( 2 ,1 ) = "SUN"
C( 2 ,2 ) = "MON"
C( 2 ,3 ) = "TUE"
C( 2 ,4 ) = "WED"
C( 2 ,5 ) = "THU"
C( 2 ,6 ) = "FRI"
C( 2 ,7 ) = "SAT"
result5 = GetArrayCount ( C( 1 ) )
result6 = GetArrayCount ( C( 2 ) )
result7 = GetArrayCount ( C )
```

## 対象バージョン

JP1/Script 06-00 以降

## 8.3 文字列操作コマンド

### 8.3.1 InStr （文字列を検索し文字位置を返す）

機能

指定された文字列の中から指定された文字列を検索し、最初に見つかった文字位置（先頭からその位置までの文字数）を返します。

形式

InStr ( *String* , *SearchStr* , [ *Start* ] [ , *Compare* ] )

指定項目

*String*

検索対象となる文字列を文字列、または値を格納した変数名で指定します。  
この値が長さ 0 の文字列 ( "" ) の場合は、0 を返します。

*SearchStr*

検索する文字列を文字列、または値を格納した変数名で指定します。  
この値が長さ 0 の文字列 ( "" ) の場合は、0 を返します。

*Start*

*String* で指定した文字列の先頭を 1 として、どの位置から検索開始するかを先頭からの文字数で指定します。  
この値が *String* で指定した文字列の文字数を超える場合は、0 を返します。  
この値は省略できます。省略した場合、1 を仮定します。

*Compare*

文字列の比較方法を指定します。  
この値は次のどれかになります。

値	意味
真 ( True )	半角文字も全角文字も大文字と小文字を区別して比較します。
偽 ( False )	半角文字も全角文字も大文字と小文字を区別しないで比較します。
Twice	半角文字の大文字と小文字は区別しないで全角文字の大文字と小文字は区別して比較します。

この値は省略できます。省略した場合、Twice を仮定します。

説明

指定された文字列の中から指定された文字列を検索し、最初に見つかった文字位置（先頭からその位置までの文字数）を返します。  
全角文字も半角文字も同じ 1 文字となります。  
指定された文字列が見つからない場合は、0 を返します。



例

```
' 変数point1には6が格納される。
Dim point1
point1 = InStr ( "ファイルをABC順にならべる" , "abc" , 3 , False )
```

対象バージョン

JP1/Script 05-00 以降

### 8.3.2 InArray （配列変数の要素から文字列を検索し、インデックス番号を返す）

機能

指定された配列変数の要素の中から指定された文字列を検索し、最初に見つかったインデックス番号を返します。

形式

```
InArray ( ArrayName , SearchStr , [Start] [, Compare] )
```

指定項目

*ArrayName*

検索対象となる配列変数を変数名で指定します。  
この配列変数の要素数が 0 の場合は、0 を返します。

*SearchStr*

検索する文字列を文字列、または値を格納した変数名で指定します。  
この値が長さ 0 の文字列 ( "" ) の場合は、0 を返します。

*Start*

*ArrayName* で指定した配列変数のどの位置から検索開始するかをインデックス番号の数値、または値を格納した変数名で指定します。  
この値が *ArrayName* で指定した配列変数の要素数を超える場合は、0 を返します。  
この値は省略できます。省略した場合、先頭の要素を仮定します。

*Compare*

文字列の比較方法を指定します。  
この値は次のどれかになります。

値	意味
真 ( True )	半角文字も全角文字も大文字と小文字を区別して比較します。
偽 ( False )	半角文字も全角文字も大文字と小文字を区別しないで比較します。
Twice	半角文字の大文字と小文字は区別しないで全角文字の大文字と小文字は区別して比較します。

この値は省略できます。省略した場合、Twice を仮定します。

## 8. 基本コマンド

### 説明

指定された配列変数の要素の中から指定された文字列を検索し、最初に見つかったインデックス番号（1 から始まる数値）を返します。

指定された文字列が見つからない場合は、0 を返します。

### 例 1

・ 下図のように値を格納した配列変数closeDayから指定のデータを検索する。

	(第1列)	(第2列)	(第3列)	...	(第11列)	(第12列)
(第1行)	"January"	"February"	"March"	...	"November"	"December"
(第2行)	5	2	1	...	1	6
(第3行)	19	16	15	...	15	20

```
Dim closeDay( 3 ,12 )
:
( 配列変数closeDayへの値格納処理 )
:
monthName = "March"
buff = InArray ( closeDay( 1 ) ,monthName ,1 ,False )
If buff > 0 Then
    firstDay = closeDay( 2 ,buff )
    MessageBox ( monthName+"の第1定休日は"+firstDay+"です。" )
End If
```

### 例 2

・ 二次元の配列変数の全要素の中から指定のデータを検索する。

```
Dim array1( , )
:
( 配列変数array1への値格納処理 )
:
allCnt = GetArrayCount ( array1 )
For line = 1 To allCnt
    buff = InArray ( array1( line ) ,"1999" ,1 ,False )
    If 0 < buff Then
        Exit For
    End If
Next
```

### 対象バージョン

JP1/Script 06-00 以降

## 8.3.3 Len （文字列の文字数を返す）

### 機能

文字列の文字数を返します。

### 形式

Len ( *String* )

### 指定項目

*String*

文字列，または値を格納した変数名を指定します。

説明

指定された文字列の文字数，または指定された変数に格納された文字列の文字数を返します。

全角文字も半角文字も同じ 1 文字となります。

例

```
' 変数length1には6が格納される。
Dim length1
length1 = Len ( "ABCDEF" )

' 変数length2には4が格納される。
Dim length2
length2 = Len ( "ファイル" )
```

対象バージョン

JP1/Script 01-00 以降

### 8.3.4 LCase （半角アルファベットの大文字を小文字に変換する）

機能

文字列の半角アルファベットの大文字を小文字に変換し，変換した文字列を返します。

形式

LCase ( *String* )

指定項目

*String*

文字列，または値を格納した変数名を指定します。

説明

指定された文字列の半角アルファベットの大文字を小文字に変換し，変換した文字列を返します。

指定された文字列の大文字だけが小文字に変換されます。大文字の半角アルファベット以外の文字は影響を受けません。

例

```
' 変数string1には "abcdef" が格納される。
Dim string1
string1 = LCase ( "abcDEF" )
```

対象バージョン

JP1/Script 01-00 以降

### 8.3.5 UCase （半角アルファベットの小文字を大文字に変換する）

機能

文字列の半角アルファベットの小文字を大文字に変換し、変換した文字列を返します。

形式

UCase ( *String* )

指定項目

*String*

文字列、または値を格納した変数名を指定します。

説明

指定された文字列の半角アルファベットの小文字を大文字に変換し、変換した文字列を返します。

指定された文字列の小文字だけが大文字に変換されます。小文字の半角アルファベット以外の文字は影響を受けません。

例

```
' 変数string1には "ABCDEF" が格納される。
Dim string1
string1 = UCase ( "abcDEF" )
```

対象バージョン

JP1/Script 01-00 以降

### 8.3.6 Left （左端から指定した文字数分の文字列を返す）

機能

文字列の左端から指定された文字数分の文字列を返します。

形式

Left ( *String* , *Length* )

指定項目

*String*

文字列、または値を格納した変数名を指定します。

*Length*

*String* で指定した文字列から取り出す文字数を数値、または値を格納した変数名で指定します。

この値に 0 を指定した場合は、長さ 0 の文字列 ( "" ) を返します。また、

*String* で指定した文字列全体の文字数以上の値を指定した場合は、*String* で指定した文字列全体を返します。

## 説明

指定された文字列の左端から指定された文字数分の文字列を返します。  
 全角文字も半角文字も同じ 1 文字となります。  
 文字列の文字数を調べるには、Len コマンドを使用します。

## 例

```
' 変数string1には "ABCD" が格納される。
Dim string1
string1 = Left ( "ABCDEFGH" ,4 )

' 変数string2には "ABC順" が格納される。
Dim string2
string2 = Left ( "ABC順にならべる",4 )
```

## 対象バージョン

JP1/Script 01-00 以降

### 8.3.7 Mid （指定した文字数分の文字列を返す）

## 機能

文字列の中間から指定された文字数分の文字列を返します。

## 形式

Mid ( *String* , *Start* [, *Length*] )

## 指定項目

*String*

文字列、または値を格納した変数名を指定します。

*Start*

*String* で指定した文字列の先頭の位置を 1 として、どの位置から文字列を取り出すかを先頭からの文字数で指定します。

この値が *String* で指定した文字列の文字数を超える場合は、長さ 0 の文字列 ("" ) を返します。

*Length*

取り出す文字数を数値、または値を格納した変数名で指定します。

この値を省略した場合、または文字列内にこの値より短い文字数しかない場合には、*Start* から後のすべての文字列が返されます。

## 説明

指定された文字列の中間から指定された文字数分の文字列を返します。  
 全角文字も半角文字も同じ 1 文字となります。  
 文字列の文字数を調べるには、Len コマンドを使用します。

## 例

```
' 変数string1には "CDEFG" が格納される。
Dim string1
```

## 8. 基本コマンド

```
string1 = Mid ( "ABCDEFGH" , 3 , 5 )  
  
' 変数string2には "c順になら" が格納される。  
Dim string2  
string2 = Mid ( "ABC順にならべる" , 3 , 5 )
```

対象バージョン

JP1/Script 01-00 以降

### 8.3.8 Right （右端から指定した文字数分の文字列を返す）

機能

文字列の右端から指定された文字数分の文字列を返します。

形式

Right ( *String* , *Length* )

指定項目

*String*

文字列、または値を格納した変数名を指定します。

*Length*

*String* で指定した文字列から取り出す文字数を、数値、または値を格納した変数名で指定します。

この値に 0 を指定した場合は、長さ 0 の文字列 ("" ) を返します。また、*String* で指定した文字列全体の文字数以上の値を指定した場合は、*String* で指定した文字列全体を返します。

説明

指定された文字列の右端から指定された文字数分の文字列を返します。

全角文字も半角文字も同じ 1 文字となります。

文字列の文字数を調べるには、Len コマンドを使用します。

例

```
' 変数string1には "EFGH" が格納される。  
Dim string1  
string1 = Right ( "ABCDEFGH" , 4 )  
  
' 変数string2には "ならべる" が格納される。  
Dim string2  
string2 = Right ( "ABC順にならべる" , 4 )
```

対象バージョン

JP1/Script 01-00 以降

### 8.3.9 Space （半角スペースからなる文字列を返す）

機能

指定された数の半角スペースからなる文字列を返します。

#### 形式

Space ( *Number* )

#### 指定項目

##### *Number*

スペースの数を数値，または値を格納した変数名で指定します。指定できる値の範囲は，0 から 1,024 までです。0 を指定した場合は，長さ 0 の文字列 ("" ) を返します。また，1,024 より大きい値を指定した場合は，1,024 を仮定します。

#### 説明

指定された数の半角スペースからなる文字列を返します。

#### 例

```
' 変数string1には "   " (半角スペース3文字) が格納される。
Dim string1
string1 = Space ( 3 )
```

#### 対象バージョン

JP1/Script 01-00 以降

### 8.3.10 LTrim ( 先頭のスペースを削除した文字列を返す )

#### 機能

文字列から先頭のスペースを削除した文字列を返します。

#### 形式

LTrim ( *String* [, *Option*] )

#### 指定項目

##### *String*

文字列，または値を格納した変数名を指定します。  
文字列の先頭にスペースが見つからない場合は，String で指定した文字列全体を返します。

##### *Option*

オプションを指定します。  
この値には次の値を指定できます。

値	意味
Twice	全角スペースも削除します。

#### 説明

オプションが省略された場合，指定された文字列から先頭の半角スペースを削除した文字列を返します。

オプションに `Twice` が指定されている場合は、半角スペースだけでなく全角スペースも削除します。

例

```
' 変数string1には"ABC DEFG  "が格納される。
Dim string1
string1 = LTrim ( "  ABC DEFG  " ,Twice )
```

対象バージョン

JP1/Script 05-00 以降

8.3.11 RTrim （末尾のスペースを削除した文字列を返す）

機能

文字列から末尾のスペースを削除した文字列を返します。

形式

```
RTrim ( String [, Option] )
```

指定項目

*String*

文字列、または値を格納した変数名を指定します。  
文字列の末尾にスペースが見つからない場合は、*String* で指定した文字列全体を返します。

*Option*

オプションを指定します。  
この値には次の値を指定できます。

値	意味
Twice	全角スペースも削除します。

説明

オプションが省略された場合、指定された文字列から末尾の半角スペースを削除した文字列を返します。  
オプションに `Twice` が指定されている場合は、半角スペースだけでなく全角スペースも削除します。

例

```
' 変数string1には"  ABC DEFG"が格納される。
Dim string1
string1 = RTrim ( "  ABC DEFG  " ,Twice )
```

対象バージョン

JP1/Script 05-00 以降



### 8.3.12 Trim（先頭および末尾のスペースを削除した文字列を返す）

#### 機能

文字列から先頭、および末尾のスペースを削除した文字列を返します。

#### 形式

Trim ( String [, Option] )

#### 指定項目

##### String

文字列、または値を格納した変数名を指定します。

文字列の先頭、および末尾にスペースが見つからない場合は、Stringで指定した文字列全体を返します。

##### Option

オプションを指定します。

この値には次の値を指定できます。

値	意味
Twice	全角スペースも削除します。

#### 説明

オプションが省略された場合、指定された文字列から先頭、および末尾の半角スペースを削除した文字列を返します。

オプションに Twice が指定されている場合は、半角スペースだけでなく全角スペースも削除します。

#### 例

変数string1と変数string2と変数string3には"ABC DEFG"が格納される。

```
Dim string1 ,string2 ,string3
string1 = Trim ( "   ABC DEFG   " ,Twice )
string2 = Trim ( "       ABC DEFG" ,Twice )
string3 = Trim ( "ABC DEFG       " ,Twice )
```

#### 対象バージョン

JP1/Script 05-00 以降

### 8.3.13 文字列連結（+）（文字列連結を行う）

#### 機能

二つの式に対して文字列の連結を行います。

#### 形式

Result = Expression1 + Expression2

#### 指定項目

8. 基本コマンド

*Result*

結果を受け取る変数名を指定します。

*Expression1*

任意の式を指定します。

*Expression2*

任意の式を指定します。

説明

式の内部処理形式により、+ 演算子の動作は次のように異なります。

条件	行われる演算
両方の式が文字	文字列連結
両方の式が数値	加算
両方の式が数値だけからなる文字列	加算
一方の式が数値,他方が文字列	文字列連結
一方の式が文字列,他方が数値だけからなる文字列	文字列連結
一方の式が数値,他方が数値だけからなる文字列	加算

両方の式が Empty 値のときは、演算結果 *Result* は数値 0 になります。ただし、一方の式だけが Empty 値のときは、他方の式がそのまま演算結果 *Result* として返されます。

補足

*Result* と *Expression1* が同じ変数である場合、+= 演算子を使用して書き換えることができます。

例

```
' 変数result1には"ABCDEF"が格納される。
result1 = "ABC" + "DEF"

' 変数result2には12が格納される。
result2 = 7 + 5
```

対象バージョン

JP1/Script 01-00 以降

8.3.14 文字列連結 (&) (文字列連結を行う)

機能

二つの式に対して文字列連結を行います。

形式

*Result* = *Expression1* & *Expression2*

指定項目

*Result*

結果を受け取る変数名を指定します。

*Expression1*

任意の式を指定します。

*Expression2*

任意の式を指定します。

## 説明

式の内部処理形式により、& 演算子の動作は次のようになります。

条件	行われる演算
両方の式が文字	文字列連結
両方の式が数値	文字列連結
両方の式が数値だけからなる文字列	文字列連結
一方の式が数値,他方が文字列	文字列連結
一方の式が文字列,他方が数値だけからなる文字列	文字列連結
一方の式が数値,他方が数値だけからなる文字列	文字列連結

両方の式が Empty 値のときは、演算結果 *Result* は長さ 0 の文字列 ("" ) になります。ただし、一方の式だけが Empty 値のときは、他方の式がそのまま演算結果 *Result* として返されます。

## 補足

*Result* と *Expression1* が同じ変数である場合、&= 演算子を使用して書き換えることができます。

## 例

```
' 変数result1には"ABCDEF"が格納される。
result1 = "ABC" & "DEF"

' 変数result2には75が格納される。
result2 = 7 & 5
```

## 対象バージョン

JP1/Script 06-00 以降

### 8.3.15 文字列連結 (&=) (変数と式の文字列連結を行う)

## 機能

変数の値と式の値に対して文字列連結を行い、連結した文字列を変数に代入します。

## 形式

*Result* &= *Expression*

## 指定項目

## 8. 基本コマンド

### *Result*

結果を受け取る変数名を指定します。

### *Expression*

任意の式を指定します。

### 説明

&= 演算子は変数 *Result* と式 *Expression* のすべての値を文字列とみなして文字列連結を行います。

両方の値が Empty 値のときは、変数 *Result* には Empty 値が代入されます。変数 *Result* が Empty 値、または未定義のときは、式 *Expression* の値がそのまま変数 *Result* に代入されます。式 *Expression* の値が Empty 値のときは、変数 *Result* の値は変わりません。

### 例

' 変数result1には"ABCDEF"が格納される。

```
result1 = "ABC"
```

```
result1 &= "DEF"
```

' 変数result2には"01"が格納される。

```
result2 = 0
```

```
result2 &= 1
```

' 変数result3には"100"が格納される。

```
Dim result3
```

```
result3 &= 100
```

### 対象バージョン

JP1/Script 06-51 以降

## 8.3.16 AddStr （指定した連結文字で文字列連結を行う）

### 機能

指定された二つ以上の文字列に対して、指定された連結文字で区切った文字列連結を行い、連結した文字列を返します。

### 形式

```
AddStr ( [SeparateChar] , [Option] , String1 , String2 [ ,  
String3 , ... ] )
```

### 指定項目

#### *SeparateChar*

連結文字を文字列、または値を格納した変数名で指定します。

この値は省略できます。省略した場合、連結文字を挿入しません。

#### *Option*

オプションを指定します。

この値には次の値を指定できます。

値	意味
NeedDq	<i>String1</i> ~ 5で指定された文字列中に連結文字とみなされるコンマ、タブ、スペースを含んでいるときだけ文字列をダブルクォーテーションで囲みます。
AllDq	<i>String1</i> ~ 5で指定された文字列を無条件にダブルクォーテーションで囲みます。

この値は省略できます。省略した場合、NeedDq を仮定します。

*String1* ~ 5

連結する文字列を文字列、または値を格納した変数名で指定します。文字列は最大五つ指定できます。

#### 説明

指定された複数の文字列に対して、指定された連結文字で区切った文字列連結を行い、連結した文字列を返します。連結した文字列の文字数が半角文字で 1,024 文字を超える場合、1,024 文字で切り捨てられます。

#### 例

```
' 変数string1には"コード 291" 価格 "3,000"が格納される。
Dim string1
string1 = AddStr ( " " , "コード 291" , "価格" , "3,000" )

' 変数string2には"コード 291" "価格" "3,000"が格納される。
Dim string2
string2 = AddStr ( " " , AllDq , "コード 291" , "価格" , "3,000" )
```

#### 対象バージョン

JP1/Script 05-00 以降

### 8.3.17 SeparateStrCount （文字列分割を行い、分割文字列の数を返す）

#### 機能

指定された文字列に対して、指定された区切り文字で文字列分割を行い、分割した文字列の数を返します。

#### 形式

SeparateStrCount ( *String* , *SeparateChar* )

#### 指定項目

*String*

分割する文字列を文字列、または値を格納した変数名で指定します。

*SeparateChar*

区切り文字を文字列、または値を格納した変数名で指定します。

#### 説明

指定された文字列に対して、指定された区切り文字で文字列分割を行い、分割した

文字列の数を返します。

区切り文字が見つからないため分割できない場合は、1 を返します。

区切り文字が *String* の先頭や末尾にある場合は、その区切り文字の直前や直後に長さ 0 の文字列 ("" ) があるものとして分割した文字列の数を返します。

また、*String* で指定した文字列が長さ 0 の文字列 ("" ) である場合は、0 を返します。

分割した文字列を取得するには、SeparateStr コマンドを使用します。

「分割する文字列」に  $n$  個のダブルクォーテーション ( " ) を文字としてそのまま含めるには、 $(n * 8)$  個のダブルクォーテーション ( " ) を指定しなければなりません。

ただし、次のレジストリの値に 1 を設定した場合は、 $(n * 2)$  個のダブルクォーテーション ( " ) を指定してください。

[ レジストリキー ]

HKEY\_LOCAL\_MACHINE¥SOFTWARE¥Hitachi¥JP1/Script¥SPTX

[ 値名 ]

ParseDQ (DWORD)

[ 値 ]

2 : パラメタ文字列に  $n$  個のダブルクォーテーション ( " ) を含む場合、 $(n * 2)$  個のダブルクォーテーション ( " ) を指定する

なお、[ 値 ] に 1 ( SetGV , GetGV , DeleteGV コマンドを参照 ) と 2 の両方を指定する場合は 3 を指定してください。

例

' 変数count1には3が格納される。

```
Dim count1
count1 = SeparateStrCount ( "JP1 Script 01-00,01-01" , " " )
```

' 変数count2には3が格納される。

```
Dim count2 ,param
param = ""コード 100"" 価格 300"
count2 = SeparateStrCount ( param , " " )
```

' 変数count3には4が格納される。

```
Dim count3
count3 = SeparateStrCount ( "コード100;コード200;コード300;" , ";" )
```

対象バージョン

JP1/Script 05-00 以降

### 8.3.18 SeparateStr ( 文字列分割を行い、分割文字列を返す )

機能

指定された文字列に対して、指定された区切り文字で文字列分割を行い、分割した

文字列を返します。

#### 形式

`SeparateStr ( String , SeparateChar [ , Position ] )`

#### 指定項目

##### *String*

分割する文字列を文字列、または値を格納した変数名で指定します。

##### *SeparateChar*

区切り文字を文字列、または値を格納した変数名で指定します。

##### *Position*

分割した文字列の並びの先頭を 1 として、どの位置の文字列を取り出すかを先頭からの文字列位置で指定します。区切り文字は含みません。この値が分割した文字列の数よりも大きい場合は、長さ 0 の文字列 ("" ) が返されます。この値は省略できます。省略した場合、1 を仮定します。

#### 説明

指定された文字列に対して、指定された区切り文字で文字列分割を行い、指定された位置の分割した文字列を返します。

区切り文字が 2 文字以上連続している場合に分割した文字列は長さ 0 の文字列 ("" ) になります。

分割した文字列の数を取得するには、SeparateStrCount コマンドを使用します。

「分割する文字列」に n 個のダブルクォーテーション (") を文字としてそのまま含めるには、(n \* 8) 個のダブルクォーテーション (") を指定しなければなりません。

ただし、次のレジストリの値に 1 を設定した場合は、(n \* 2) 個のダブルクォーテーション (") を指定してください。

[ レジストリキー ]

HKEY\_LOCAL\_MACHINE¥SOFTWARE¥Hitachi¥JP1/Script¥SPTX

[ 値名 ]

ParseDQ (DWORD)

[ 値 ]

2 : パラメタ文字列に n 個のダブルクォーテーション (") を含む場合、(n\*2) 個のダブルクォーテーション (") を指定する

なお、[ 値 ] に 1 ( SetGV , GetGV , DeleteGV コマンドを参照 ) と 2 の両方を指定する場合は 3 を指定してください。

#### 例

' 変数string1には"01-00,01-01"が格納される。

```
Dim string1
```

```
string1 = SeparateStr ( "JP1 Script 01-00,01-01" , " " , 3 )
```

' 変数string2には長さ0の文字列""が格納される。

## 8. 基本コマンド

```
Dim string2
string2 = SeparateStr ( "JP1,Script,01-00,,01-01" ,"," ,4 )
```

対象バージョン

JP1/Script 05-00 以降

### 8.3.19 Str （ 値を文字列で返す ）

機能

指定された値を文字列で返します。

形式

Str ( *Number* )

指定項目

*Number*

文字列に変換する値，または値を格納した変数名で指定します。

この値に文字列を指定した場合は，*Number*で指定した値をそのまま返します。

説明

指定された値を文字列で返します。

例

・ 変数string1には2が格納される。

```
Dim string1
string1 = 1 + 2 - 1
```

・ 変数string2には"12-1"が格納される。

```
Dim string2
string2 = Str ( 1 ) + Str ( 2 ) + Str ( -1 )
```

・ 変数string3には11が格納される。

```
Dim string3
string3 = Str ( 1 ) + 2 - 1
```

対象バージョン

JP1/Script 05-20 以降

### 8.3.20 Format （ 値を書式化した文字列を返す ）

機能

指定された値を書式化した文字列で返します。

形式

Format ( *Form* , *Arg1* [ , *Arg2* , ... ] )

指定項目

*Form*

*Arg1* ~ *32*で指定された値をどのように書式化するかを文字列，または値を格



納した変数名で指定します。

*Form* の中身の文字はそのまま表され、書式化の指示の始まりは % で指定します。

この値は次のような指定の組み合わせになります。

指定	意味
%d	数値を 10 進数で表す
%x	数値を 16 進数（小文字）で表す
%X	数値を 16 進数（大文字）で表す
%o	数値を 8 進数で表す
%s	文字列をそのまま表す
%c	1 文字をそのまま表す なお、全角文字も半角文字も同じ 1 文字となる。また、 <i>Arg1</i> ~ <i>32</i> で指定された書式化する値が 2 文字以上の場合でも、先頭の 1 文字だけ表す。
%5d , %10s など	書式化する値の最大桁数を指定する
%05d	書式化する値の最大桁数を指定する場合に、最大桁数に満たないときは前ゼロで埋める
%-5d , %-10s など	値を左詰で表す

なお、% を書式化の指示の始まりではなく、文字として表す場合は %% と指定します。

*Arg1* ~ *32*

書式化する値を文字列、数値、または値を格納した変数名で指定します。*Form* で指定された書式の順に複数の値を指定できます。

#### 説明

指定された値を書式化した文字列で返します。

#### 例

```
' 前ゼロ10進数5桁の識別子をパラメタとして実行ファイルABC.EXEへ渡す。
Dim numID , strID
numID = GetGV ( "seqNo" )
If IsEmpty ( numID ) Then
    numID = 1
End If
strID = Format ( "%05d" , numID )
Exec ( _SCF_ + "ABC.EXE" , True , strID )
```

#### 対象バージョン

JP1/Script 05-20 以降

### 8.3.21 IsLower ( 半角アルファベットの小文字かどうかを調べる )

#### 機能

指定された文字列が半角アルファベットの小文字かどうかを調べ、結果を真 ( True ) か偽 ( False ) で返します。

#### 形式

IsLower ( *String* )

#### 指定項目

*String*

調べる文字列、または値を格納した変数名を指定します。

この値に長さ 0 の文字列 ( "" ) を指定した場合は、偽 ( False ) を返します。

#### 説明

指定された文字列のすべての文字が半角アルファベットの小文字かどうかを調べ、結果をコマンドの実行結果として返します。すべての文字が半角アルファベットの小文字である場合は真 ( True ) を、それ以外の場合は偽 ( False ) を返します。

#### 例

```
Dim buff
buff = GetGV ( "Level" )
If buff = "A" Then
    If IsLower ( buff ) Then
        MsgBox ( "Small A Level." )
    Else
        MsgBox ( "Large A Level." )
    End If
End If
```

#### 対象バージョン

JP1/Script 06-00 以降

### 8.3.22 IsUpper ( 半角アルファベットの大文字かどうかを調べる )

#### 機能

指定された文字列が半角アルファベットの大文字かどうかを調べ、結果を真 ( True ) か偽 ( False ) で返します。

#### 形式

IsUpper ( *String* )

#### 指定項目

*String*

調べる文字列、または値を格納した変数名を指定します。

この値に長さ 0 の文字列 ("" ) を指定した場合は、偽 ( False ) を返します。

#### 説明

指定された文字列のすべての文字が半角アルファベットの大文字かどうかを調べ、結果をコマンドの実行結果として返します。すべての文字が半角アルファベットの大文字である場合は真 ( True ) を、それ以外の場合は偽 ( False ) を返します。

#### 例

```
Dim buff
buff = GetGV ( "Level" )
If buff = "A" Then
    If IsUpper ( buff ) Then
        MsgBox ( "Large A Level." )
    Else
        MsgBox ( "Small A Level." )
    End If
End If
```

#### 対象バージョン

JP1/Script 06-00 以降

## 8.3.23 IsSingleChar ( 半角文字かどうかを調べる )

#### 機能

指定された文字列が半角文字かどうかを調べ、結果を真 ( True ) か偽 ( False ) で返します。

#### 形式

IsSingleChar ( *String* )

#### 指定項目

*String*

調べる文字列、または値を格納した変数名を指定します。

この値に長さ 0 の文字列 ("" ) を指定した場合は、偽 ( False ) を返します。

#### 説明

指定された文字列のすべての文字が半角文字かどうかを調べ、結果をコマンドの実行結果として返します。すべての文字が半角文字である場合は真 ( True ) を、それ以外の場合は偽 ( False ) を返します。

#### 例

```
Dim buff
buff = GetGV ( "#" )
If IsSingleChar ( buff ) Then
    MsgBox ( "#" + buff )
Else
    MsgBox ( “ 項番 ” + buff )
End If
```

対象バージョン

JP1/Script 06-00 以降

## 8.3.24 IsMultiChar （全角文字かどうかを調べる）

機能

指定された文字列が全角文字かどうかを調べ、結果を真（True）か偽（False）で返します。

形式

IsMultiChar ( *String* )

指定項目

*String*

調べる文字列、または値を格納した変数名を指定します。

この値に長さ 0 の文字列（""）を指定した場合は、偽（False）を返します。

説明

指定された文字列のすべての文字が全角文字かどうかを調べ、結果をコマンドの実行結果として返します。すべての文字が全角文字である場合は真（True）を、それ以外の場合は偽（False）を返します。

例

```
Dim buff
buff = GetGV ( "#" )
If IsMultiChar ( buff ) Then
    MsgBox ( " 項番 " + buff )
Else
    MsgBox ( "#" + buff )
End If
```

対象バージョン

JP1/Script 06-00 以降

## 8.4 日付操作コマンド

---

### 8.4.1 Date （現在の日付を返す）

機能

現在の日付を返します。

形式

Date

このコマンドに引数はありません。

説明

現在の日付を yyyy/mm/dd の形式で返します。

対象バージョン

JP1/Script 01-00 以降

### 8.4.2 Time （現在の時刻を返す）

機能

現在の時刻を返します。

形式

Time

このコマンドに引数はありません。

説明

現在の時刻を hh:mm:ss の形式で返します。

対象バージョン

JP1/Script 01-00 以降

### 8.4.3 Year （指定された日付の年を返す）

機能

指定された日付の年を表す値を 4 桁の数値で返します。

形式

Year ( [Date] )

指定項目

Date

日付を yyyy/mm/dd の形式で指定します。

## 8. 基本コマンド

yyyy に 0 ~ 69 の範囲の数値を指定すると、2000 ~ 2069 と読み替えられます。

mm に不正な値を指定した場合はエラーになりますが、dd に不正な値を指定した場合はエラーになりません。実在する日付に置き換えて結果を返します。例えば、1999/12/32（実在しない日付）を指定した場合、2000/01/01（実在する日付）に置き換えて、2000 を返します。

この値は省略できます。省略した場合、現在の日付を仮定します。

### 説明

指定された日付の年を表す値を 4 桁の数値で返します。

### 対象バージョン

JP1/Script 01-00 以降

## 8.4.4 Month（指定された日付の月を返す）

### 機能

指定された日付の月を表す値を 1 ~ 12 の範囲の 1 ~ 2 桁の数値で返します。

### 形式

Month（[Date]）

### 指定項目

*Date*

日付を yyyy/mm/dd の形式で指定します。

yyyy に 0 ~ 69 の範囲の数値を指定すると、2000 ~ 2069 と読み替えられます。

mm に不正な値を指定した場合はエラーになりますが、dd に不正な値を指定した場合はエラーになりません。実在する日付に置き換えて結果を返します。例えば、1999/12/33（実在しない日付）を指定した場合、2000/01/02（実在する日付）に置き換えて、1 を返します。

この値は省略できます。省略した場合、現在の日付を仮定します。

### 説明

指定された日付の月を表す値を 1 ~ 12 の範囲の 1 ~ 2 桁の数値で返します。

### 対象バージョン

JP1/Script 01-00 以降

## 8.4.5 Day（指定された日付の日を返す）

### 機能

指定された日付の日を表す値を 1 ~ 31 の範囲の 1 ~ 2 桁の数値で返します。

## 形式

Day ( [Date] )

## 指定項目

*Date*

日付を yyyy/mm/dd の形式で指定します。

yyyy に 0 ~ 69 の範囲の数値を指定すると、2000 ~ 2069 と読み替えられます。

mm に不正な値を指定した場合はエラーになりますが、dd に不正な値を指定した場合はエラーになりません。実在する日付に置き換えて結果を返します。例えば、1999/12/33 (実在しない日付) を指定した場合、2000/01/02 (実在する日付) に置き換えて、2 を返します。

この値は省略できます。省略した場合、現在の日付を仮定します。

## 説明

指定された日付の日を表す値を 1 ~ 31 の範囲の 1 ~ 2 桁の数値で返します。

## 対象バージョン

JP1/Script 01-00 以降

## 8.4.6 Weekday ( 指定された日付の曜日を返す )

## 機能

指定された日付の曜日を表す値を 1 (日曜) ~ 7 (土曜) の範囲の 1 桁の数値、または曜日を表す文字列で返します。

## 形式

Weekday ( [Date] [, Option] )

## 指定項目

*Date*

日付を yyyy/mm/dd の形式で指定します。

yyyy に 0 ~ 69 の範囲の数値を指定すると、2000 ~ 2069 と読み替えられます。

mm に不正な値を指定した場合はエラーになりますが、dd に不正な値を指定した場合はエラーになりません。実在する日付に置き換えて結果を返します。例えば、1999/12/33 (実在しない日付) を指定した場合、2000/01/02 (実在する日付) に置き換えて、1 (日曜) を返します。

この値は省略できます。省略した場合、現在の日付を仮定します。

*Option*

オプションを指定します。

この値には次の値を指定できます。

8. 基本コマンド

値	意味
String	曜日を表す値をアルファベットの文字列で返します。
StringJ	曜日を表す値を日本語の文字列で返します。

説明

オプションが省略された場合、指定された日付の曜日を表す値を 1 (日曜) ~ 7 (土曜) の範囲の 1 桁の数値で返します。

この場合、コマンドは次のどれかの値を返します。

値	内容
1	日曜
2	月曜
3	火曜
4	水曜
5	木曜
6	金曜
7	土曜

オプションが指定されている場合、指定された日付の曜日を表す値を文字列で返します。

この場合、コマンドは次のどれかの値を返します。

String を指定	StringJ を指定	内容
"SUN"	" 日 "	日曜
"MON"	" 月 "	月曜
"TUE"	" 火 "	火曜
"WED"	" 水 "	水曜
"THU"	" 木 "	木曜
"FRI"	" 金 "	金曜
"SAT"	" 土 "	土曜

対象バージョン

JP1/Script 01-00 以降

8.4.7 Hour ( 指定された時刻の時を返す )

機能

指定された時刻の時を表す値を 0 ~ 23 の範囲の 1 ~ 2 桁の数値で返します。



## 形式

Hour ( [ *Time* ] )

## 指定項目

*Time*

時刻を hh:mm:ss の形式で指定します。

hh に不正な値を指定した場合はエラーになりますが、mm と ss に不正な値を指定した場合はエラーになりません。

例えば、19:61:00 ( 実在しない時刻 ) を指定した場合、20:01:00 ( 実在する時刻 ) と置き換えて、20 を返します。

この値は省略できます。省略した場合、現在の時刻を仮定します。

## 説明

指定された時刻の時を表す値を 0 ~ 23 の範囲の 1 ~ 2 桁の数値で返します。

## 対象バージョン

JP1/Script 01-00 以降

## 8.4.8 Minute ( 指定された時刻の分を返す )

## 機能

指定された時刻の分を表す値を 0 ~ 59 の範囲の 1 ~ 2 桁の数値で返します。

## 形式

Minute ( [ *Time* ] )

## 指定項目

*Time*

時刻を hh:mm:ss の形式で指定します。

hh に不正な値を指定した場合はエラーになりますが、mm と ss に不正な値を指定した場合はエラーになりません。実在する時刻を置き換えて結果を返します。

例えば、19:61:00 ( 実在しない時刻 ) を指定した場合、20:01:00 ( 実在する時刻 ) と置き換えて、1 を返します。

この値は省略できます。省略した場合、現在の時刻を仮定します。

## 説明

指定された時刻の分を表す値を 0 ~ 59 の範囲の 1 ~ 2 桁の数値で返します。

## 対象バージョン

JP1/Script 01-00 以降

### 8.4.9 Second （ 指定された時刻の秒を返す ）

機能

指定された時刻の秒を表す値を 0 ～ 59 の範囲の 1 ～ 2 桁の数値で返します。

形式

Second ( [ Time ] )

指定項目

*Time*

時刻を hh:mm:ss の形式で指定します。

hh に不正な値を指定した場合はエラーになりますが、mm と ss に不正な値を指定した場合はエラーになりません。実在する時刻を置き換えて結果を返します。

例えば、19:00:65（実在しない時刻）を指定した場合、19:01:05（実在する時刻）と置き換えて、5 を返します。

この値は省略できます。省略した場合、現在の時刻を仮定します。

説明

指定された時刻の秒を表す値を 0 ～ 59 の範囲の 1 ～ 2 桁の数値で返します。

対象バージョン

JP1/Script 01-00 以降

### 8.4.10 CalcDate （ 指定された日付に加算減算を行う ）

機能

指定された年数、月数、日数を、指定された日付に足したり、指定された日付から引いたりして求めた日付を返します。

形式

CalcDate ( Date , Calc , [ Years ] , [ Months ] [ , Days ] )

指定項目

*Date*

日付を yyyy/mm/dd の形式で指定します。

yyyy に 0 ～ 69 の範囲の数値を指定すると、2000 ～ 2069 と読み替えられます。

mm に不正な値を指定した場合はエラーになりますが、dd に不正な値を指定した場合はエラーになりません。実在する日付に置き換えて結果を返します。

例えば、1999/12/33（実在しない日付）を指定した場合、2000/01/02（実在する日付）に置き換えます。

*Calc*

演算方法を指定します。

この値は、次のどれかになります。

値	意味
Minus	Date で指定された日付から、 <i>Years</i> 、 <i>Months</i> 、 <i>Days</i> で指定された年数、月数、日数を引いた日付を返します。(減算)
Plus	Date で指定された日付に、 <i>Years</i> 、 <i>Months</i> 、 <i>Days</i> で指定された年数、月数、日数を足した日付を返します。(加算)

#### *Years*

年数を 0 以上の数値、または値を格納した変数名で指定します。  
この値は省略できます。省略した場合、0 を仮定します。

#### *Months*

月数を 0 以上の数値、または値を格納した変数名で指定します。  
この値は省略できます。省略した場合、0 を仮定します。

#### *Days*

日数を 0 以上の数値、または値を格納した変数名で指定します。  
この値は省略できます。省略した場合、0 を仮定します。

#### 説明

指定された年数、月数、日数を、指定された日付に足したり、指定された日付から引いたりして求め、yyyy/mm/dd の形式で日付を返します。  
求めた日付が存在しない日付になった場合、その日付以降の存在する日付を返します。例えば、求めた日付が 1999/02/29 (存在しない日付) になった場合、1999/03/01 (存在する日付) を返します。  
エラーが発生した場合は、長さ 0 の文字列 ("" ) を返します。

#### 例

```
' 本日から10日前の日付を表示する。
Dim result1
result1 = CalcDate ( Date() ,Minus , , ,10 )
MessageBox ( "今日から10日前は"+result1+"です。" )
```

#### 対象バージョン

JP1/Script 05-20 以降

## 8.4.11 CompDate (二つの日付を比較する)

#### 機能

指定された二つの日付を比較して、結果を真 (True)、または偽 (False) で返します。

#### 形式

```
CompDate ( Date1 , Comp , Date2 )
```

#### 指定項目

## 8. 基本コマンド

### *Date1*

比較する一方の日付を yyyy/mm/dd の形式で指定します。yyyy に 0 ~ 69 の範囲の数値を指定すると、2000 ~ 2069 と読み替えられます。

mm に不正な値を指定した場合はエラーになりますが、dd に不正な値を指定した場合はエラーになりません。実在する日付に置き換えて結果を返します。

例えば、1999/12/33 (実在しない日付) を指定した場合、2000/01/02 (実在する日付) に置き換えます。

### *Comp*

比較方法を指定します。

この値は、次のどれかになります。

値	意味
Equal	等しい ( = )
NotEqual	等しくない ( <> )
Before	Date1 より、Date2 の方が前の日付である ( > )
After	Date1 より、Date2 の方が後の日付である ( < )

### *Date2*

比較するもう一方の日付を yyyy/mm/dd の形式で指定します。

yyyy に 0 ~ 69 の範囲の数値を指定すると、2000 ~ 2069 と読み替えられます。また、mm に不正な値を指定した場合はエラーになりますが、dd に不正な値を指定した場合は、実在する日付に置き換えます。

### 説明

二つの日付を *Comp* で指定された方法で比較し、結果をコマンドの実行結果として返します。比較結果が *Comp* で指定した値と一致する場合は真 (True) を、それ以外の場合は偽 (False) を返します。

エラーが発生した場合は、長さ 0 の文字列 ("" ) を返します。

### 例

・ 作成日付が今日から30日前のファイルを削除する。

```
Dim fileName ,delDate ,creDate
delDate = CalcDate ( Date() ,Minus ,0 ,0 ,29 )

For fileName = _TEMP_+"*.*" Do
    GetFileTime ( _TEMP_+fileName ,creDate , ,Create )
    If CompDate ( creDate ,After ,deldate ) = True Then
        DeleteFile ( _TEMP_+fileName )
    End If
End For
```

### 対象バージョン

JP1/Script 05-20 以降

## 8.4.12 GetDateCount （二つの日付の経過日数を取得する）

### 機能

指定された二つの日付の経過日数を取得します。

### 形式

`GetDateCount ( StartDate , EndDate [ , UnitofDate ] )`

### 指定項目

#### *StartDate*

経過日数を求めるときの開始日付を yyyy/mm/dd の形式で指定します。

yyyy に 0 ~ 69 の範囲の数値を指定すると、2000 ~ 2069 と読み替えられます。

mm に不正な値を指定した場合はエラーになりますが、dd に不正な値を指定した場合はエラーになりません。実在する日付に置き換えて結果を返します。

例えば、1999/12/33（実在しない日付）を指定した場合、2000/01/02（実在する日付）に置き換えます。

#### *EndDate*

経過日数を求めるときの終了日付を yyyy/mm/dd の形式で指定します。

yyyy に 0 ~ 69 の範囲の数値を指定すると、2000 ~ 2069 と読み替えられます。

また、mm に不正な値を指定した場合はエラーになりますが、dd に不正な値を指定した場合は、実在する日付に置き換えます。

#### *UnitofDate*

取得する経過日数の単位を指定します。

この値には、次のどれかを指定します。

値	意味
YearU	年の単位で取得します。 例えば、開始日付が 1998/4/1 で、終了日付が 1999/4/1 である場合は 1 になり、終了日付が 1999/3/1 である場合は 0 になります。
MonthU	月の単位で取得します。 例えば、開始日付が 1998/4/10、終了日付が 1998/5/10 である場合は 1 になり、終了日付が 1998/5/9 である場合は 0 になります。
DayU	日の単位で取得します。

この値は省略できます。省略した場合、DayU を仮定します。

### 説明

指定された二つの日付の経過日数を取得し、コマンドの実行結果として返します。

*StartDate* で指定された日付が *EndDate* で指定された日付よりもあとの日付である場合、経過日数は負の値になります。

### 例

8. 基本コマンド

```
' 1998年4月1日から1999年3月31日までの経過日数を年,月,日の単位で求める。
Dim date1 ,date2 ,yBuff ,mBuff ,dBuff
date1 = "1998/4/1"
date2 = "1999/3/31"

' yBuffには0,mBuffには11,dBuffには364が格納される。
yBuff = GetDateCount ( date1 ,date2 ,YearU )
mBuff = GetDateCount ( date1 ,date2 ,MonthU )
dBuff = GetDateCount ( date1 ,date2 ,DayU )
```

対象バージョン  
JP1/Script 05-20 以降

8.4.13 CalcTime （ 指定された時刻に加算減算を行う ）

機能  
指定された時刻に、指定された時、分、または秒を足したり引いたりした時刻を返します。

形式  
CalcTime ( Time , Calc , [Hours] , [Minutes] , [Seconds] [, DaysBuff] )

指定項目

Time  
時刻を hh:mm:ss の形式で指定します。  
hh に不正な値を指定した場合はエラーになりますが、mm と ss に不正な値を指定した場合はエラーになりません。実在する時刻に置き換えます。  
例えば、19:00:65（実在しない時刻）を指定した場合、19:01:05（実在する時刻）と置き換えます。

Calc  
演算方法を指定します。  
この値には、次のどれかを指定します。

値	意味
Minus	Time で指定された時刻から、Hours、Minutes、Seconds で指定された時、分、秒を引いた日付を返します。（減算）
Plus	Time で指定された時刻に、Hours、Minutes、Seconds で指定された時、分、秒を足した日付を返します。（加算）

Hours  
時を 0 以上の数値、または値を格納した変数名で指定します。  
この値は省略できます。省略した場合、0 を仮定します。

Minutes  
分を 0 以上の数値、または値を格納した変数名で指定します。

この値は省略できます。省略した場合、0 を仮定します。

#### *Seconds*

秒を 0 以上の数値、または値を格納した変数名で指定します。

この値は省略できます。省略した場合、0 を仮定します。

#### *DaysBuff*

求めた時刻が *Time* で指定された時刻から日をまたがった場合、またがった日を受け取る変数名を指定します。必要がない場合は省略します。

日のまたがりがない場合、0 を返します。

#### 説明

指定された時刻に、指定された時、分、または秒を足したり引いたりした時刻を求め、求めた時刻を hh:mm:ss の形式で返します。

エラーが発生した場合は、長さ 0 の文字列 ("" ) を返します。

#### 例

・ スクリプトファイルを現在の時刻から7時間45分後に自動起動するように登録する。

```
Dim resTime ,resdate ,daysBuff
resTime = CalcTime ( Time() ,Plus ,7 ,45 , ,daysBuff )
resDate = CalcDate ( Date() , Plus , , ,daysBuff )
EntryStartUp ( _SCF_+"ABC.SPT" , , ,resTime , ,Day(resDate) )
```

#### 対象バージョン

JP1/Script 05-20 以降

## 8.4.14 CompTime (二つの時刻を比較する)

#### 機能

指定された二つの時刻を比較して、結果を真 (True) , または偽 (False) で返します。

#### 形式

```
CompTime ( Time1 , Comp , Time2 )
```

#### 指定項目

##### *Time1*

比較する一方の時刻を hh:mm:ss の形式で指定します。

hh に不正な値を指定した場合はエラーになりますが、mm と ss に不正な値を指定した場合はエラーになりません。実在する時刻に置き換えます。

例えば、19:00:65 ( 実在しない時刻 ) を指定した場合、19:01:05 ( 実在する時刻 ) と置き換えます。

##### *Comp*

比較方法を指定します。

この値は、次のどれかになります。

## 8. 基本コマンド

値	意味
Equal	等しい ( = )
NotEqual	等しくない ( <> )
Before	<i>Time1</i> より , <i>Time2</i> の方が前の時刻である ( > )
After	<i>Time1</i> より , <i>Time2</i> の方が後の時刻である ( < )

*Time2*

比較するもう一方の日付を hh:mm:ss の形式で指定します。

mm と ss に不正な値を指定した場合は、実在する時刻に置き換えます。

### 説明

二つの時刻を *Comp* で指定された方法で比較し、結果をコマンドの実行結果として返します。比較結果が *Comp* で指定された値と一致する場合は真 ( True ) を、それ以外の場合は偽 ( False ) を返します。

エラーが発生した場合は、長さ 0 の文字列 ( "" ) を返します。

### 例

・ 作成時間が今日の午前中のファイルを削除する。

```
Dim fileName , creDate , creTime

For fileName = _TEMP_ + "*. *" Do
  GetFileTime ( _TEMP_ + fileName , creDate , creTime , Create )
  If CompDate ( creDate , Equal , Date() ) Then
    If CompTime ( creTime , After , "12:00:00" ) = True Then
      DeleteFile ( _TEMP_ + fileName )
    End If
  End If
End For
```

### 対象バージョン

JP1/Script 05-20 以降

## 8.4.15 GetTimeCount ( 二つの時刻の経過時間を取得する )

### 機能

指定された二つの時刻の経過時間を取得します。

### 形式

```
GetTimeCount ( StartTime , EndTime [ , UnitofTime ] )
```

### 指定項目

*StartTime*

経過時間を求めるときの開始時刻を hh:mm:ss の形式で指定します。

hh に不正な値を指定した場合はエラーになりますが、mm と ss に不正な値を指定した場合はエラーになりません。実在する時刻に置き換えます。



例えば、19:00:65（実在しない時刻）を指定した場合、19:01:05（実在する時刻）と置き換えます。

#### *EndTime*

経過時間を求めるときの終了時刻を hh:mm:ss の形式で指定します。

mm と ss に不正な値を指定した場合は、実在する時刻に置き換えます。

#### *UnitofTime*

取得する経過時間の単位を指定します。

この値には、次のどれかを指定します。

値	意味
HourU	時の単位で取得します。 例えば、開始時刻が 9:00:00 で、終了時刻が 10:00:00 である場合は 1 になり、終了時刻が 9:59:59 である場合は 0 になります。
MinuteU	分の単位で取得します。 例えば、開始時刻が 9:10:00、終了時刻が 9:11:00 である場合は 1 になり、終了時刻が 9:10:59 である場合は 0 になります。
SecondU	秒の単位で取得します。

この値は省略できます。省略した場合、SecondU を仮定します。

#### 説明

指定された二つの時刻の経過時間を取得し、コマンドの実行結果として返します。  
*StartTime* で指定され時刻が *EndTime* で指定された時刻よりも後の時刻である場合、経過時間は負の値になります。

#### 例

' 9:10:30から10:09:20までの経過時間を時,分,秒の単位で求める。

```
Dim time1 ,time2 ,hBuff ,mBuff ,sBuff
```

```
time1 = "9:10:30"
```

```
time2 = "10:09:20"
```

' hBuffには0,mBuffには58,sBuffには3530が格納される。

```
hBuff = GetTimeCount ( time1 ,time2 ,HourU )
```

```
mBuff = GetTimeCount ( time1 ,time2 ,MinuteU )
```

```
sBuff = GetTimeCount ( time1 ,time2 ,SecondU )
```

#### 対象バージョン

JP1/Script 05-20 以降

## 8.4.16 IsLeapYear（うるう年かどうかを調べる）

#### 機能

指定された西暦年がうるう年かどうかを調べ、結果を真（True）、または偽（False）で返します。

#### 形式

## 8. 基本コマンド

IsLeapYear ( *Year* )

### 指定項目

*Year*

西暦年を数値，または値を格納した変数名で指定します。

西暦年に 0 ~ 69 の範囲の数値を指定すると，2000 ~ 2069 と読み替えられます。

### 説明

指定された西暦年がうるう年かどうかを調べ，結果をコマンドの実行結果として返します。うるう年である場合には真 ( True ) を，うるう年でない場合には偽 ( False ) を返します。

### 例

・ 今年がうるう年かどうかを調べる。

```
Dim nowYear
nowYear = Year ( Date() )
If IsLeapYear ( nowYear ) Then
    MessageBox ( "今年はうるう年です。" )
Else
    MessageBox ( "今年はうるう年ではありません。" )
End If
```

### 対象バージョン

JP1/Script 05-20 以降

## 8.5 ファイル・フォルダ操作コマンド

### 8.5.1 IniRead （初期化ファイルから値を読み込む）

#### 機能

初期化ファイル（.INI ファイル）から値を読み込みます。

#### 形式

`IniRead ( FilePath , SectionName , ValueBuff , EntryName )`

#### 指定項目

##### *FilePath*

初期化ファイル名のフルパスを文字列、または値を格納した変数名で指定します。

##### *SectionName*

セクション名を文字列、または値を格納した変数名で指定します。

##### *ValueBuff*

読み込まれた値を受け取る変数名を指定します。

##### *EntryName*

エントリ名を文字列、数値、または値を格納した変数名で指定します。

#### 説明

指定された初期化ファイル（.INI ファイル）の特定のセクションから、エントリされている値を読み込み、指定された変数に格納します。コマンドが正常に実行された場合は真（True）を、エラーが発生した場合は偽（False）を、コマンドの実行結果として返します。

*FilePath* で指定された初期化ファイル、または *SectionName* で指定された初期化ファイルの特定のセクションが存在しない場合は、指定された変数に長さ 0 の文字列（""）を格納します。

#### 補足

Windows Vista で環境変数 ProgramFiles（通常はシステムドライブの Program Files フォルダ）、または WinDir（通常はシステムドライブの Windows フォルダ）に設定されているフォルダ下の初期化ファイルを指定する場合、注意が必要です。詳細は、「1.8.4 コマンドの動作」を参照してください。

#### 例

起動フォルダ下の初期化ファイル "ABC.INI" の内容が次のような場合、以下のスクリプトで内容が読み込める。

```
; ABC.INI
[Files]
File01=Readme.txt
```

## 8. 基本コマンド

```
File02=Abc.exe
```

・ スクリプトファイル

・ 1行目で変数file1に"Readme.txt"が格納される。

・ 2行目で変数file2に"Abc.exe"が格納される。

```
Dim file1 ,file2
```

```
IniRead ( _BIN_"ABC.INI" , "Files" , file1 , "File01" )
```

```
IniRead ( _BIN_"ABC.INI" , "Files" , file2 , "File02" )
```

対象バージョン

JP1/Script 01-00 以降

### 8.5.2 IniWrite （初期化ファイルに値を登録する）

機能

初期化ファイル（.INI ファイル）に値を登録します。

形式

```
IniWrite ( FilePath , SectionName , EntryName , Value )
```

指定項目

*FilePath*

初期化ファイル名のフルパスを文字列、または値を格納した変数名で指定します。

*SectionName*

セクション名を文字列、または値を格納した変数名で指定します。

*EntryName*

エントリ名を文字列、数値、または値を格納した変数名で指定します。

*Value*

登録する値を文字列、数値、または値を格納した変数名で指定します。

説明

指定された初期化ファイルの特定のセクションに値を登録します。コマンドが正常に実行された場合は真（True）を、エラーが発生した場合は偽（False）を、コマンドの実行結果として返します。

*FilePath* で指定された初期化ファイルが存在しない場合は、ファイルを新規作成します。また、*SectionName* で指定された初期化ファイルの特定のセクションが存在しない場合は、セクションを新規作成します。

補足

Windows Vista で環境変数 ProgramFiles（通常はシステムドライブの Program Files フォルダ）、または WinDir（通常はシステムドライブの Windows フォルダ）に設定されているフォルダ下の初期化ファイルを指定する場合、注意が必要です。詳細は、「1.8.4 コマンドの動作」を参照してください。

## 例

次のようなスクリプトで Windows フォルダ下の初期化ファイル "ABC.INI" に情報を書き込める。

## ' スクリプトファイル

```
IniWrite ( _WIN_+"ABC.INI" , "Files" , "File01" , "Readme.txt")
IniWrite ( _WIN_+"ABC.INI" , "Files" , "File02" , "Abc.exe")

; ABC.INI
[Files]
File01 = Readme.txt
File02 = Abc.exe
```

## 対象バージョン

JP1/Script 01-00 以降

### 8.5.3 TextFileReplace ( テキストファイルの文字列を置き換える )

## 機能

テキストファイルの中の特定の文字列を置き換えます。

## 形式

```
TextFileReplace ( FileName , OldText , NewText [ , ReplaceCntBuff ]
)
```

## 指定項目

*FileName*

テキストファイル名を文字列、または値を格納した変数名で指定します。  
ファイル名の拡張子が省略された場合、自動的に拡張子 ( .TXT ) を付加します。ただし、ファイル名の末尾が拡張子のピリオド ( . ) である場合、JP1/Script 06-00 以降では、拡張子のないファイル名とみなします。

*OldText*

変更前文字列を文字列、または値を格納した変数名で指定します。

*NewText*

変更後文字列を文字列、または値を格納した変数名で指定します。

*ReplaceCntBuff* ( 06-00 以降 )

置き換えた文字列の数を受け取る変数名を指定します。必要がない場合は省略します。

## 説明

指定された変更前文字列をテキストファイルから検索し、別の文字列に置き換えます。コマンドが正常に実行された場合は真 ( True ) を、エラーが発生した場合は偽 ( False ) を、コマンドの実行結果として返します。

ラージファイルには、使用できません。ファイル分割 ( SplitFile ) から、使用して

ください。

### 補足

Windows Vista で環境変数 ProgramFiles ( 通常はシステムドライブの Program Files フォルダ ), または WinDir ( 通常はシステムドライブの Windows フォルダ ) に設定されているフォルダ下のファイルを指定する場合, 注意が必要です。詳細は, 「1.8.4 コマンドの動作」を参照してください。

### ! 注意事項

変換前文字列に改行コードを指定してもファイル中の改行コードは変換後文字列に変換されません。

### 例

#### [ 実行前 ]

```
' TextFileReplaceのテスト
TextFileReplace ( _BIN_+"ABC.TXT" , "テスト" , "test" )
```

#### [ 実行後 ]

```
' TextFileReplaceのtest
TextFileReplace ( _BIN_+"ABC.TXT" , "test" , "test" )
```

### 対象バージョン

JP1/Script 01-00 以降

## 8.5.4 TextOpen ( テキストファイルをオープンする )

### 機能

テキスト形式ファイルをオープンします。  
JP1/Script 07-50 以降では, ラージファイルが使用できます。

### 形式

TextOpen ( *FilePath* [ , *Mode* ] )

### 指定項目

#### *FilePath*

テキスト形式ファイル名のフルパスを文字列, または値を格納した変数名で指定します。

ファイル名の拡張子が省略された場合, 自動的に拡張子 ( .TXT ) を付加します。ただし, ファイル名の末尾が拡張子のピリオド ( . ) である場合, JP1/Script 06-00 以降では, 拡張子のないファイル名とみなします。

#### *Mode*

ファイルオープン時の動作を指定します。  
この値は次のどれかになります。

値	意味
Create	ファイルが無条件に作成
ReadOnly	ファイルを読み込みモードでオープン
WriteOnly	ファイルを書き込みモードでオープン
ReadWrite	ファイルを読み書きモードでオープン

*Mode* に *ReadOnly* を指定した場合は共用モードで、*Create*、*WriteOnly*、および *ReadWrite* を指定した場合は排他モードでオープンします。

この値は省略できます。省略した場合、*Create* を仮定します。

#### 説明

指定されたテキスト形式ファイルをオープンします。コマンドが正常に実行された場合はファイル識別子を、エラーが発生した場合は 0 を、コマンドの実行結果として返します。

またコマンドが正常に実行された場合は、現在の読み書き開始位置は先頭の 0 になります。

#### 補足

Windows Vista で環境変数 *ProgramFiles* (通常はシステムドライブの *Program Files* フォルダ)、または *WinDir* (通常はシステムドライブの *Windows* フォルダ) に設定されているフォルダ下のファイルを指定する場合、注意が必要です。詳細は、「1.8.4 コマンドの動作」を参照してください。

#### 例

```
Dim file1
file1 = TextOpen ( _BIN_+"Readme.txt" ,ReadOnly )
If file1 = 0 Then
    MessageBox ( _BIN_+"Readme.txtのオープンに失敗" ,OK )
Else
    Dim buff1
    If TextRead ( file1 ,buff1 ) Then
        MessageBox ( buff1,OK )
    End
    TextClose ( file1 )
End
```

#### 対象バージョン

JP1/Script 05-00 以降

## 8.5.5 TextClose (テキストファイルをクローズする)

#### 機能

テキスト形式ファイルをクローズします。

JP1/Script 07-50 以降では、ラージファイルが使用できます。

形式

```
TextClose ( [ FileId ] )
```

指定項目

*FileId*

ファイル識別子を数値、または値を格納した変数名で指定します。  
このファイル識別子は、TextOpen コマンドで実行結果として返される値です。  
この値は省略できます。省略、または 0 の場合、オープン中のファイルをすべてクローズします。

説明

指定されたファイル識別子のテキスト形式ファイルをクローズします。コマンドが正常に実行された場合は真 (True) を、エラーが発生した場合は偽 (False) を、コマンドの実行結果として返します。

例

```
Dim file1 ,file2
file1 = TextOpen ( _BIN_+"Readme.txt" ,ReadOnly )
If file1 = 0 Then
    MessageBox ( _BIN_+"Readme.txtのオープンに失敗" ,OK )
    Exit
End
file2 = TextOpen ( _TEMP_+"Backup.txt" ,Create )
If file2 = 0 Then
    MessageBox ( _TEMP_+"Backup.txtのオープンに失敗" ,OK )
    TextClose ( file1 )
    Exit
End

Dim buff1
If TextRead ( file1 ,buff1 ) Then
    TextWrite ( file2 ,buff1 )
End
TextClose ( file1 )
TextClose ( file2 )
```

対象バージョン

JP1/Script 05-00 以降

## 8.5.6 TextRead ( テキストファイルから 1 行分のデータを読み込む )

機能

テキスト形式ファイルから 1 行分のデータを読み込みます。  
JP1/Script 07-50 以降では、ラージファイルが使用できます。

形式

```
TextRead ( FileId , Buff )
```



## 指定項目

*FileId*

ファイル識別子を数値，または値を格納した変数名で指定します。  
 このファイル識別子は，TextOpen コマンドで実行結果として返される値です。  
 なお，TextOpen コマンドのファイルオープン時の動作には，WriteOnly（ファイルを書き込みモードでオープン）以外を指定してください。

*Buff*

読み込まれた 1 行分のデータを受け取る変数名を指定します。受け取るデータの文字数が半角文字で 1,024 文字を超える場合，1,024 文字で切り捨てられます。

## 説明

指定されたファイル識別子のテキスト形式ファイルの現在の読み書き開始位置から 1 行分のデータを読み込み，指定された変数に格納します。コマンドが正常に実行された場合は真（True）を，エラーが発生した場合は偽（False）を，コマンドの実行結果として返します。

コマンドの実行結果が真（True）の場合は，読み書き開始位置は次行になります。また，ファイルの終わりに達した場合はコマンドの実行結果に偽（False）を返し，\_RTN\_ 予約変数には \_ERR\_EOF\_ 予約変数と等しい値が設定されます。しかし，On Error ステートメントで指定したラベルには分岐しません。

**!** 注意事項

TextRead コマンドでファイルの終わりに達した場合，JP1/Script ではコマンド実行時に異常が発生した場合と同じようにエラーとして処理します。したがって，スクリプト処理が正常終了した場合でも，実行トレースログにはエラーとして記録されます。また，トレースビューア上での表示が異常終了となるので，TextRead コマンドの EOF 検出でのエラーは無視してください。

また，TextRead コマンドで読み込まれた 1 行分のデータを受け取る変数には，直前まで格納していた値が格納されているので，EOF 検出後に変数内容が未設定になる状態を期待する処理は作り込まないでください。

## 例

- ・ 実行フォルダ下のテキストファイル"Readme.txt"の内容を，
- ・ 以下のスクリプトで1行づつ読み込み内容を表示する。

```
Dim file1 ,buff1 ,readRtn

On Error GoTo ErrorBranch
file1 = TextOpen ( _BIN_+"Readme.txt" ,ReadOnly )
TextRead ( file1 ,buff1 )
readRtn = _RTN_
While readRtn <> _ERR_EOF_
  MessageBox ( buff1 )
  TextRead ( file1 ,buff1 )
  readRtn = _RTN_
ErrorBranch:
```

## 8. 基本コマンド

```
End

TextClose ( file1 )
Exit ( 0 )

ErrorBranch:
    MessageBox ( _BIN_+"Readme.txtのファイル操作失敗" ,OK )
    TextClose ( file1 )
    Exit ( -1 )
```

### 対象バージョン

JP1/Script 05-00 以降

## 8.5.7 TextWrite ( テキストファイルにデータを書き込む )

### 機能

テキスト形式ファイルにデータを書き込みます。

JP1/Script 07-50 以降では、ラージファイルが使用できます。

### 形式

```
TextWrite ( FileId , Data , [NewLine] [ , Position] )
```

### 指定項目

#### *FileId*

ファイル識別子を数値，または値を格納した変数名で指定します。

このファイル識別子は，TextOpen コマンドで実行結果として返される値です。

なお，TextOpen コマンドのファイルオープン時の動作には，ReadOnly ( ファイルを読み込みモードでオープン ) 以外を指定してください。

#### *Data*

書き込むデータを文字列，または値を格納した変数名で指定します。

#### *NewLine*

データを書き込んだ後に改行をするときには真 ( True ) を，改行をしないときには偽 ( False ) を指定します。

この値は省略できます。省略した場合，真 ( True ) を仮定します。

#### *Position*

ファイルの先頭を 0 とした読み書き開始位置を数値，または値を格納した変数名で指定します。

この値は省略できます。省略した場合，現在の読み書き開始位置を仮定します。

### 説明

指定されたファイル識別子のテキスト形式ファイルに対しデータを現在の読み書き開始位置，または指定された読み書き開始位置から書き込みます。コマンドが正常に実行された場合は真 ( True ) を，エラーが発生した場合は偽 ( False ) を，コマンドの実行結果として返します。

読み書き開始位置は，コマンドの実行結果が真 ( True ) で *NewLine* に真 ( True )

が指定してある場合は次行になり、*NewLine* に偽 (False) が指定してある場合は書き込んだデータの直後になります。

### ! 注意事項

すでに存在するファイルに対して *TextOpen* コマンドの *Mode* オペランドに *WriteOnly*、または *ReadWrite* を指定し、*TextWrite* コマンドを実行した場合、*TextWrite* コマンドは、単純に指定された文字数分の上書きを行います。したがって、出力する文字数 (改行コードを含む) が既存ファイルより少ない場合、少ない文字数分、オープン前の内容がファイルに残ります。

### 例

次のようなスクリプトで実行フォルダ下のテキストファイル "Readme.txt" にデータを書き込める。

・ スクリプトファイル

```
Dim file1
file1 = TextOpen ( _BIN_+"Readme.txt" ,Create )
If file1 = 0 Then
    MsgBox ( _BIN_+"Readme.txtのオープンに失敗" ,OK )
Else
    TextWrite ( file1 , "JP1/Scriptについての追加情報" )
    TextWrite ( file1 , "目次" ,False )
    TextClose ( file1 )
End

; Readme.txt
JP1/Scriptについての追加情報
目次
```

### 対象バージョン

JP1/Script 05-00 以降

## 8.5.8 TextSeek (読み書き開始位置を先頭または末尾に移動する)

### 機能

テキスト形式ファイルの読み書き開始位置をファイルの先頭、または最後に移動します。

JP1/Script 07-50 以降では、ラージファイルが使用できます。

### 形式

```
TextSeek ( FileId [, Point] )
```

### 指定項目

*FileId*

ファイル識別子を数値、または値を格納した変数名で指定します。

このファイル識別子は、*TextOpen* コマンドで実行結果として返される値です。

## 8. 基本コマンド

なお、TextOpen コマンドのファイルオープン時の動作には、WriteOnly（ファイルを書き込みモードでオープン）以外を指定してください。

### Point

読み書き開始位置を指定します。  
この値は次のどれかになります。

値	意味
ToBegin	ファイルの先頭に移動します。
ToEnd	ファイルの最後に移動します。

この値は省略できます。省略した場合、ToBegin を仮定します。

### 説明

指定されたファイル識別子のテキスト形式ファイルに対し読み書き開始位置を指定された位置に移動します。コマンドが正常に実行された場合は真（True）を、エラーが発生した場合は偽（False）を、コマンドの実行結果として返します。

### 例

```
' 実行フォルダ下のテキストファイル"Readme.txt"の
' 最後に "*** END OF FILE ***"と書き込む。
Dim file1
file1 = TextOpen ( _BIN_+"Readme.txt" ,ReadWrite )
If file1 = 0 Then
    MessageBox ( _BIN_+"Readme.txtのオープンに失敗" ,OK )
Else
    If TextSeek ( file1 ,ToEnd ) Then
        TextWrite ( file1 , "*** END OF FILE ***" ,False )
    End
    TextClose ( file1 )
End
```

### 対象バージョン

JP1/Script 05-00 以降

## 8.5.9 GetTextPosition（読み書きの開始位置を返す）

### 機能

テキスト形式ファイルの現在の読み書き開始位置を返します。

### 形式

GetTextPosition ( *FileId* )

### 指定項目

#### *FileId*

ファイル識別子を数値、または値を格納した変数名で指定します。  
このファイル識別子は、TextOpen コマンドで実行結果として返される値です。

## 説明

指定されたファイル識別子のテキスト形式ファイルの現在の読み書き開始位置を、先頭を 0 とした値で取得します。コマンドが正常に実行された場合は現在の読み書き開始位置を、エラーが発生した場合は長さ 0 の文字列 ("" ) を、コマンドの実行結果として返します。

読み書き開始位置が 2,147,483,647 を超える場合は、エラーとなります。

## 例

```
' 実行フォルダ下のテキストファイル"Readme.txt"から読み込んだデータが、
' "JP1/Scriptについて"であれば" JP1/Scriptの説明 "に書き換える。
Dim file1
file1 = TextOpen ( _BIN_+"Readme.txt" ,ReadWrite )
If file1 = 0 Then
    MessageBox ( _BIN_+"Readme.txtのオープンに失敗" ,OK )
Else
    Dim line ,position ,buff
    For line = 1 To 10
        position = GetTextPosition ( file1 )
        If TextRead ( file1 ,buff ) Then
            If buff = "JP1/Scriptについて" Then
                TextWrite ( file1 ," JP1/Scriptの説明 " ,True ,position )
            End
        Else
            Exit For
        End
    Next
    TextClose ( file1 )
End
```

## 対象バージョン

JP1/Script 05-00 以降

## 8.5.10 MakeDir ( フォルダを作成する )

## 機能

フォルダを作成します。

## 形式

MakeDir ( *DirPath* )

## 指定項目

*DirPath*

作成するフォルダパスを文字列、または値を格納した変数名で指定します。

## 説明

指定されたフォルダを作成します。コマンドが正常に実行された場合は真 ( True ) を、エラーが発生した場合は偽 ( False ) を、コマンドの実行結果として返します。指定されたフォルダまでのパスが存在しない場合でも、途中のパスを作成してフォルダを作成します。

補足

Windows Vista で環境変数 ProgramFiles ( 通常はシステムドライブの Program Files フォルダ ), または WinDir ( 通常はシステムドライブの Windows フォルダ ) に設定されているフォルダ下のフォルダを指定する場合, 注意が必要です。詳細は, 「1.8.4 コマンドの動作」を参照してください。

例

```
' Windowsフォルダ下に"HITACHI"フォルダを作成する。
MakeDir ( _WIN_+"HITACHI" )
```

対象バージョン

JP1/Script 01-00 以降

8.5.11 DeleteDir ( フォルダを削除する )

機能

フォルダを削除します。

形式

```
DeleteDir ( DirPath [ , Option ] )
```

指定項目

*DirPath*

削除するフォルダパスを文字列, または値を格納した変数名で指定します。

*Option ( 06-00 以降 )*

オプションを指定します。  
この値には次の値を指定できます。

値	意味
Anyway	フォルダの中身が空の場合も空でない場合も中身ごとフォルダを削除します。
IfEmpty	フォルダの中身が空の場合はフォルダを削除し, 空でない場合はフォルダを削除しません。 なお, フォルダを削除しなかった場合, 偽 ( False ) をコマンドの実行結果として返します。

この値は省略できます。省略した場合, Anyway を仮定しますので注意が必要です。

説明

指定されたフォルダを削除します。コマンドが正常に実行された場合は真 ( True ) を, エラーが発生した場合は偽 ( False ) を, コマンドの実行結果として返します。また, 指定されたフォルダが存在しない場合は, 常に真 ( True ) を返します。

補足

Windows Vista で環境変数 ProgramFiles ( 通常はシステムドライブの Program

Files フォルダ), または WinDir (通常はシステムドライブの Windows フォルダ) に設定されているフォルダ下のフォルダを指定する場合, 注意が必要です。詳細は, 「1.8.4 コマンドの動作」を参照してください。

**!** 注意事項

オプションに IfEmpty の指定がない場合, フォルダの中身が空ではなくても, 中身ごとフォルダを削除してしまいます。使用する場合は十分に注意してください。

例

```
' Windowsフォルダ下の"HITACHI"フォルダを無条件に削除する。
DeleteDir ( _WIN_+"HITACHI" )

' Windowsフォルダ下の"HITACHI"フォルダ下が空であれば削除する。
DeleteDir ( _WIN_+"HITACHI" ,IfEmpty )
```

対象バージョン

JP1/Script 01-00 以降

### 8.5.12 DeleteFile ( ファイルを削除する )

機能

ファイルを削除します。  
JP1/Script 07-50 以降では, ラージファイルが使用できます。

形式

```
DeleteFile ( PathName [ , Option] )
```

指定項目

*PathName*

削除するフォルダ, またはファイルのフルパスを文字列, または値を格納した変数名で指定します。ファイル名にワイルドカードを指定することもできます。

*Option ( 05-10 以降 )*

オプションを指定します。  
この値には次の値を指定できます。

値	意味
ExclDir	<i>PathName</i> で指定されたフォルダ下のファイルだけを削除し, フォルダは残します。

説明

指定されたフォルダ, またはファイルを削除します。オプションが省略された場合, フォルダの中身が空ではなくても, 中身ごとフォルダを削除します。オプションに ExclDir が指定されている場合は, 指定されたフォルダ下のファイルだけを削除します。  
コマンドが正常に実行された場合は真 ( True ) を, エラーが発生した場合は偽

## 8. 基本コマンド

( False ) を、コマンドの実行結果として返します。

ただし、「指定されたファイルが見つからない」( \_ERR\_FILE\_ 予約変数と等しい値 )、または「指定されたネットワーク名またはパスが見つからない」( \_ERR\_PATH\_ 予約変数と等しい値 ) のエラーが発生した場合は、常に真 ( True ) を返します。

### 補足

Windows Vista で環境変数 ProgramFiles ( 通常はシステムドライブの Program Files フォルダ )、または WinDir ( 通常はシステムドライブの Windows フォルダ ) に設定されているフォルダ下のファイルを指定する場合、注意が必要です。詳細は、「1.8.4 コマンドの動作」を参照してください。

### 例

```
' 一時フォルダ下のファイル"tempfile.tmp"を削除する。
DeleteFile ( _TEMP_+"tempfile.tmp" )

' 一時フォルダ下の拡張子が".tmp"であるファイルを削除する。
DeleteFile ( _TEMP_+"*.tmp" )

' 一時フォルダ下のすべてのファイルを削除し、一時フォルダは残す。
DeleteFile ( _TEMP_ ,ExclDir )
```

### 対象バージョン

JP1/Script 01-00 以降

## 8.5.13 Rename ( ファイル名を変更する )

### 機能

ファイル名を変更します。

JP1/Script 07-50 以降では、ラージファイルが使用できます。

### 形式

```
Rename ( OldName , [ NewName ] [ , Method ] )
```

### 指定項目

*OldName*

変更前ファイル名を文字列、または値を格納した変数名で指定します。

*NewName*

変更後ファイル名を文字列、または値を格納した変数名で指定します。この値を省略し、変更方法 *Method* に Reboot を指定した場合、ファイルは削除されます。

*Method*

変更方法を指定します。

この値は次のどれかになります。



値	意味
Replace	NewName で指定された名称のファイルがすでに存在する場合もファイルを置き換えます。
NoReplace ( 06-00 以降 )	NewName で指定された名称のファイルがすでに存在する場合はエラーになります。
Reboot	システムの再起動後にファイルを置き換えます。
FreeExt	拡張子を .000 ~ .999 のうち、空いているものに変更します。このオプション使用時は、変更後ファイル名を指定しても無視されます。また、このオプションは Copy コマンドの Pile オプションと異なって、拡張子が .000 ~ .999 のファイルがすでに存在している場合、内容がまったく同じものがあるときは OldName で指定された変更前ファイルを削除するだけとなります。

この値は省略できます。省略した場合、Replace が仮定されます。

Reboot を指定した場合、Windows Vista では管理者権限でスクリプトファイルを実行してください。詳細は、「1.8.4 コマンドの動作」を参照してください。

#### 説明

OldName で指定されたファイルを NewName のファイル名に変更します。コマンドが正常に実行された場合は真 (True) を、エラーが発生した場合は偽 (False) を、コマンドの実行結果として返します。

NewName を省略、または長さ 0 の文字列 ("" ) を指定し、変更方法 Method に Reboot を指定するとファイルは削除されます。

#### 補足

Windows Vista で環境変数 ProgramFiles ( 通常はシステムドライブの Program Files フォルダ )、または WinDir ( 通常はシステムドライブの Windows フォルダ ) に設定されているフォルダ下のファイルを指定する場合、注意が必要です。詳細は、「1.8.4 コマンドの動作」を参照してください。

#### 例

```
' ファイル名を変更する。
Dim outDir1
outDir1 = _BIN_
Rename ( outDir1+"DEFAULT.DAT" ,outDir1+"DEFAULT.000" ,FreeExt )
Rename ( _TEMP_+"WORK.DAT" ,outDir1+"DEFAULT.DAT" )

' 自分自身のコピーを作り、元のファイルを削除する。
' (普通には削除できないため)
Copy ( _BIN_+"ABC.EXE" , _BIN_+"XYZ.EXE" ,Overwrite )
Rename ( _BIN_+"ABC.EXE" ,"" ,Reboot )
```

#### 対象バージョン

JP1/Script 01-00 以降

### 8.5.14 TempDir （一時フォルダを取得する）

機能

一時フォルダを取得します。

形式

TempDir ( *DirNameBuff* )

指定項目

*DirNameBuff*

一時フォルダを受け取る変数名を指定します。

説明

システムの一時フォルダを取得し、変数に格納します。コマンドが正常に実行された場合は真 (True) を、エラーが発生した場合は偽 (False) を、コマンドの実行結果として返します。

フォルダ名の後ろには ( ¥ ) が付加されます。

対象バージョン

JP1/Script 01-00 以降

### 8.5.15 TempFile （一時ファイル名を作成する）

機能

一時ファイル名を作成します。

形式

TempFile ( *FileNameBuff* , [ *Prefix* ] [ , *DirName* ] )

指定項目

*FileNameBuff*

作成した一時ファイル名を受け取る変数名を指定します。

*Prefix*

プレフィックスを文字列、または値を格納した変数名で指定します。プレフィックスは先頭の 3 文字が有効となります。

この値は省略できます。省略した場合、"STX" が仮定されます。

*DirName*

一時ファイルを作成するフォルダ名を文字列、または値を格納した変数名で指定します。

この値は省略できます。省略した場合、システムの一時フォルダを仮定します。

説明

一時ファイル名を作成します。コマンドが正常に実行された場合は真 (True) を、エラーが発生した場合は偽 (False) を、コマンドの実行結果として返します。

このコマンドは、作成されたファイル名でファイルサイズ 0 バイトのファイルを実際に作成します。

#### 補足

Windows Vista で環境変数 ProgramFiles (通常はシステムドライブの Program Files フォルダ)、または WinDir (通常はシステムドライブの Windows フォルダ) に設定されているフォルダ下のフォルダを指定する場合、注意が必要です。詳細は、「1.8.4 コマンドの動作」を参照してください。

#### 例

```
Dim bkupFileName ,outDir1
outDir1 = _TEMP_
TempFile ( bkupFileName , "BUP" ,outDir1 )
Copy ( outDir1+"MODEM.INF" ,bkupFileName )
```

#### 対象バージョン

JP1/Script 01-00 以降

## 8.5.16 SetFileAttribute または SetFileAttr ( フォルダ / ファイルの属性を設定する )

#### 機能

フォルダ、またはファイルの属性を設定します。  
JP1/Script 07-50 以降では、ラージファイルが使用できます。

#### 形式

```
SetFileAttribute ( PathName , Attribute1 [ , Attribute2 , ... ] )
SetFileAttr ( PathName , Attribute1 [ , Attribute2 , ... ] )
```

#### 指定項目

##### *PathName*

設定するフォルダ、またはファイルのフルパスを文字列、または値を格納した変数名で指定します。

##### *Attribute1 ~ 8*

設定する属性を指定します。複数の属性を指定できます。  
ただし、ATTR\_NORMAL は単独で指定した場合にだけ有効です。  
この値は次のようになります。

値	意味
ATTR_READONLY	読み込み専用ファイル
ATTR_HIDDEN	隠しファイル
ATTR_ARCHIVE	アーカイブファイル
ATTR_SYSTEM	システムファイル

## 8. 基本コマンド

値	意味
ATTR_TEMPORARY	一時ファイル ただし、PathName にフォルダを指定した場合は、この属性は指定できません。
ATTR_NORMAL	すべての属性を取り除きます。この属性は、単独で指定したときだけ有効です。

*PathName* で指定されたフォルダ、またはファイルに対して、*Attribute1* ~ *8* で指定された属性だけを操作する場合、次のどれかを指定します。

値	意味
ATTR_ON (05-10以降)	<i>Attribute1</i> ~ <i>8</i> で指定された属性だけを追加します。
ATTR_OFF (05-10以降)	<i>Attribute1</i> ~ <i>8</i> で指定された属性だけを取り除きます。

JP1/Script 06-00 以降では、複数の属性を指定する場合、設定する属性を一つずつ格納した一次元の配列変数を変数名で指定することもできます。

### 説明

指定されたフォルダ、またはファイルに指定された属性を設定します。複数の属性を設定できます。コマンドが正常に実行された場合は真 (True) を、エラーが発生した場合は偽 (False) を、コマンドの実行結果として返します。

フォルダ、またはファイルの複数の属性を取得するには、GetFileAttribute コマンドを、フォルダ、またはファイルの一つの属性をチェックするには、IsFileAttribute コマンドを使用します。

### 補足

Windows Vista で環境変数 ProgramFiles (通常はシステムドライブの Program Files フォルダ)、または WinDir (通常はシステムドライブの Windows フォルダ) に設定されているフォルダ、フォルダ下のフォルダ、またはフォルダ下のファイルを指定する場合、注意が必要です。詳細は、「1.8.4 コマンドの動作」を参照してください。

### 例

・ ファイルのアーカイブ属性を取り除いておき、ファイル操作後にアーカイブ属性をチェックして、ファイルが更新されているかどうかを判定する。

```
Dim file1
file1 = "C:¥TEMP¥logging.tmp"
If IsFileAttribute ( file1 ,ATTR_ARCHIVE ) = True Then
    SetFileAttribute ( file1 ,ATTR_ARCHIVE ,ATTR_OFF )
End
:
(ファイル操作)
:
If IsFileAttribute ( file1 ,ATTR_ARCHIVE ) = True Then
    MessageBox ( file1+"は更新されました。" )
Else
```

```

    MessageBox ( file1+"は更新されませんでした。" )
End

```

対象バージョン  
JP1/Script 01-00 以降

### 8.5.17 GetFileAttribute または GetFileAttr （フォルダ / ファイルの属性を取得する）

**機能**  
フォルダ，またはファイルの属性を取得します。  
JP1/Script 07-50 以降では，ラージファイルが使用できます。

**形式**  
 GetFileAttribute ( PathName , AttrArrayBuff )  
 GetFileAttr ( PathName , AttrArrayBuff )

**指定項目**

*PathName*  
取得するフォルダ，またはファイルのフルパスを文字列，または値を格納した変数名で指定します。

*AttrArrayBuff*  
取得する属性を受け取る要素数が可変な一次元の配列変数の変数名を指定します。  
格納される値は次のようになります。

値	意味
ATTR_READONLY	読み込み専用ファイル
ATTR_HIDDEN	隠しファイル
ATTR_ARCHIVE	アーカイブファイル
ATTR_SUBDIR	サブディレクトリ
ATTR_SYSTEM	システムファイル
ATTR_TEMPORARY	一時ファイル
ATTR_COMPRESSED	圧縮ファイル
ATTR_NORMAL	ファイルには，特に属性はありません。

格納後，配列変数の要素数は属性の数に置き換わります。

**説明**  
指定されたフォルダ，またはファイルに指定された属性を取得します。コマンドが正常に実行された場合は真（True）を，エラーが発生した場合は偽（False）を，コマンドの実行結果として返します。  
フォルダ，またはファイルの属性を設定するには，SetFileAttribute コマンドを，

フォルダ、またはファイルの一つの属性をチェックするには、IsFileAttribute コマンドを使用します。

#### 補足

Windows Vista で環境変数 ProgramFiles (通常はシステムドライブの Program Files フォルダ)、または WinDir (通常はシステムドライブの Windows フォルダ) に設定されているフォルダ、フォルダ下のフォルダ、またはフォルダ下のファイルを指定する場合、注意が必要です。詳細は、「1.8.4 コマンドの動作」を参照してください。

#### 例

```
' 実行フォルダ下の一時ファイルを削除する。
Dim file_attr( )
For file = _SCF_+"*.*" Do
    ' ファイル属性を取得
    GetFileAttribute ( file ,file_attr )
    aryCnt = GetArrayCount ( file_attr )
    ' 属性を格納した配列変数から一時ファイル属性を検索
    For i = 1 To aryCnt
        If file_attr( i ) = ATTR_TEMPORARY Then
            ' ファイル削除
            DeleteFile ( file , ExclDir )
        Exit For
    End If
Next
End For
```

#### 対象バージョン

JP1/Script 06-00 以降

## 8.5.18 SetFileTime ( ファイルの日付と時刻を設定する )

#### 機能

ファイルの日付と時刻を設定します。  
JP1/Script 07-50 以降では、ラージファイルが使用できます。

#### 形式

```
SetFileTime ( PathName , [Date] , [Time] [, Type] )
```

#### 指定項目

##### *PathName*

設定するファイルのフルパスを文字列、または値を格納した変数名で指定します。

##### *Date*

設定する日付を yyyy/mm/dd の形式の文字列、または値を格納した変数名で指定します。yyyy に 0 ~ 69 の範囲の数値を指定すると、2000 ~ 2069 と読み替えられます。

*Time*

設定する時刻を hh:mm:ss の形式の文字列、または値を格納した変数名で指定します。

*Type*

設定するファイル日付の種別を指定します。  
この値は次のどちらかになります。

値	意味
Create	作成日付
Update	更新日付

この値は省略できます。省略した場合、Update を仮定します。

## 説明

指定されたファイルに日付と時刻を設定します。コマンドが正常に実行された場合は真 (True) を、エラーが発生した場合は偽 (False) を、コマンドの実行結果として返します。

なお、指定されたファイルが FAT システムのファイルである場合、1980 年 1 月 1 日から 2107 年 12 月 31 日までの日付しか設定できません。

## 補足

Windows Vista で環境変数 ProgramFiles (通常はシステムドライブの Program Files フォルダ)、または WinDir (通常はシステムドライブの Windows フォルダ) に設定されているフォルダ下のファイルを指定する場合、注意が必要です。詳細は、「1.8.4 コマンドの動作」を参照してください。

## 例

```
Dim file1
file1 = _SCF_+"User.txt"
If IsFileAttr ( file1 ,ATTR_ARCHIVE ) Then
    SetFileAttr ( file1 ,ATTR_ARCHIVE ,ATTR_OFF )
    SetFileTime ( file1 ,Date() ,Time() ,Update )
End If
```

## 対象バージョン

JP1/Script 05-10 以降

## 8.5.19 GetFileTime ( ファイルの日付と時刻を取得する )

## 機能

ファイルの日付と時刻を取得します。  
JP1/Script 07-50 以降では、ラージファイルが使用できます。

## 形式

```
GetFileTime ( PathName , [DateBuff] , [TimeBuff] [, Type] )
```

8. 基本コマンド

指定項目

*PathName*

取得するファイルのフルパスを文字列、または値を格納した変数名で指定します。

*DateBuff*

日付を受け取る変数名を指定します。日付は yyyy/mm/dd の形式で返します。必要がない場合は省略します。

*TimeBuff*

時刻を受け取る変数名を指定します。時刻は hh:mm:ss の形式で返します。必要がない場合は省略します。

*Type*

取得するファイル日付の種別を指定します。  
この値は次のどちらかになります。

値	意味
Create	作成日付
Update	更新日付

この値は省略できます。省略した場合、Update を仮定します。

説明

指定されたファイルの日付と時刻を取得します。コマンドが正常に実行された場合は真 (True) を、エラーが発生した場合は偽 (False) を、コマンドの実行結果として返します。

補足

Windows Vista で環境変数 ProgramFiles (通常はシステムドライブの Program Files フォルダ)、または WinDir (通常はシステムドライブの Windows フォルダ) に設定されているフォルダ下のファイルを指定する場合、注意が必要です。詳細は、「1.8.4 コマンドの動作」を参照してください。

例

```
Dim file1 ,file2 ,dateBuff ,timeBuff
file1 = _SCF_+"User.txt"
file2 = _SCF_+"UserBkup.txt"
If GetFileTime ( file1 ,dateBuff ,timeBuff ,Create ) = True Then
    SetFileTime ( file2 ,dateBuff ,timeBuff ,Create )
End
```

対象バージョン

JP1/Script 05-10 以降



## 8.5.20 GetFileSize ( ファイルの容量を取得する )

### 機能

ファイルの容量を取得します。

JP1/Script 07-50 以降では、ラージファイルが使用できます。

### 形式

GetFileSize ( *PathName* [ , *UnitofByte* ] )

### 指定項目

#### *PathName*

取得するファイルのフルパスを文字列、または値を格納した変数名で指定します。

#### *UnitofByte*

取得する容量の単位を指定します。

この値は次のどれかになります。

値	意味
Byte	バイト
KB	キロバイト
MB	メガバイト

*UnitofByte* に KB、または MB が指定された場合、切り上げた値を返します。  
例えば、*UnitofByte* に KB が指定され、容量が 1 キロバイトに満たない場合は、1 を返します。

この値は省略できます。省略した場合、Byte を仮定します。

ラージファイルでも指定された容量の単位に変換した結果が、JP1/Script で扱える数値の上限値 2,147,483,647 を超えると、エラーとなります。

### 説明

指定されたファイルの容量を取得し、コマンドの実行結果として返します。

エラーが発生した場合は、0 を返します。

### 補足

Windows Vista で環境変数 ProgramFiles ( 通常はシステムドライブの Program Files フォルダ )、または WinDir ( 通常はシステムドライブの Windows フォルダ ) に設定されているフォルダ下のファイルを指定する場合、注意が必要です。詳細は、「1.8.4 コマンドの動作」を参照してください。

### 例

```
' コピー元ファイルの容量を調べ、コピー先ディスクに書き込める
' 空き容量があればファイルをコピーする。
Dim path1 , path2 , fileSize , freeSz
path1 = _SCF_ + "Bkup.txt"
```

## 8. 基本コマンド

```
path2 = "A:"  
fileSz = GetFileSize ( path1 )  
freeSz = GetDiskFreeSpace ( path2 ,KB )  
If ( freeSz * 1024 ) >= fileSz Then  
    Copy ( path1 ,path2 )  
End If
```

対象バージョン

JP1/Script 06-00 以降

### 8.5.21 GetVersionInfo または GetVerInfo ( ファイルのバージョン情報を取得する )

機能

ファイルのバージョン情報を取得します。

形式

```
GetVersionInfo ( PathName , {Version1Buff} , {Version2Buff} ,  
    {DescriptionBuff} , {LegalCopyrightBuff} , {CompanyNameBuff}  
    , {OriginalFilenameBuff} , {ProductVersionBuff} ,  
    {ProductNameBuff} { , InternalNameBuff} )  
GetVerInfo ( PathName , {Version1Buff} , {Version2Buff} ,  
    {DescriptionBuff} , {LegalCopyrightBuff} , {CompanyNameBuff}  
    , {OriginalFilenameBuff} , {ProductVersionBuff} ,  
    {ProductNameBuff} { , InternalNameBuff} )
```

指定項目

*PathName*

取得するファイルのフルパスを文字列、または値を格納した変数名で指定します。

*Version1Buff*

9999.9999.9999.9999 の形式のファイルバージョンを受け取る変数名を指定します。必要がない場合は省略します。

ファイルバージョンは 9999.9999.9999.9999 の形式の文字列 ( 9999 は前ゼロ 4 桁の数字列 ) で返します。

*Version2Buff*

ファイルバージョンを受け取る変数名を指定します。必要がない場合は省略します。

ファイルバージョンは文字列で返します。

*DescriptionBuff*

説明を受け取る変数名を指定します。必要がない場合は省略します。

*LegalCopyrightBuff*

著作権を受け取る変数名を指定します。必要がない場合は省略します。

*CompanyNameBuff*

会社名を受け取る変数名を指定します。必要がない場合は省略します。

*OriginalFilenameBuff*

正式ファイル名を受け取る変数名を指定します。必要がない場合は省略します。

*ProductVersionBuff*

製品バージョンを受け取る変数名を指定します。必要がない場合は省略します。

*ProductNameBuff*

製品名を受け取る変数名を指定します。必要がない場合は省略します。

*InternalNameBuff*

内部名を受け取る変数名を指定します。必要がない場合は省略します。

## 説明

指定されたファイルのバージョン情報を取得し、各変数に格納します。コマンドが正常に実行された場合は真 (True) を、エラーが発生した場合は偽 (False) をコマンドの実行結果として返します。

## 補足

Windows Vista で環境変数 ProgramFiles (通常はシステムドライブの Program Files フォルダ)、または WinDir (通常はシステムドライブの Windows フォルダ) に設定されているフォルダ下のファイルを指定する場合、注意が必要です。詳細は、「1.8.4 コマンドの動作」を参照してください。

## 例

```
' ファイルバージョンを比較する。
Dim path1 ,ver1 ,ver2
path1 = "C:¥Program Files¥Hitachi¥Script¥Bin¥SPTXE.EXE"
ver0520 = "0005.0020.0000.0000"
GetVersionInfo ( path1 , ver1 , ver2)
If ver0520 < ver1 Then
    MessageBox ( ver0520 + "よりも新しい。" + _NL_ + _
        "ファイルバージョン：" + ver2 )
End If
```

## 対象バージョン

JP1/Script 06-00 以降

## 8.5.22 SplitFile ( ファイルを分割する )

## 機能

ファイルを指定されたサイズで分割します。  
JP1/Script 07-50 以降では、ラージファイルが使用できます。

## 形式

```
SplitFile ( FilePath , SplitSize , [DirPath] , [Option] [ , SplitCnt] )
```

## 指定項目

*FilePath*

## 8. 基本コマンド

分割するファイルのフルパスを文字列、または値を格納した変数名で指定します。

### *SplitSize*

分割するサイズを数値、または値を格納した変数名で指定します。サイズはバイト単位で指定します。

ラージファイルでも、分割するサイズの最大値は、2,147,483,647 バイトです。

### *DirPath*

分割したファイルを格納するフォルダ名を文字列、または値を格納した変数名で指定します。

この値は省略できます。省略した場合、*FilePath* で指定されたファイルが存在するフォルダを仮定します。

### *Option*

オプションを指定します。

この値には次の値を指定できます。

値	意味
Delete	ファイルを分割後、 <i>FilePath</i> で指定されたファイルを削除します。

### *SplitCnt*

分割したファイル数を受け取る変数名を指定します。必要がない場合は省略します。

## 説明

指定されたファイルを指定されたサイズで分割し、分割したファイルを指定されたフォルダに格納します。コマンドが正常に実行された場合は真 (True) を、エラーが発生した場合は偽 (False) を、コマンドの実行結果として返します。

分割したファイルのファイル名は *FilePath* で指定されたファイル名の末尾に拡張子 .XXX (XXX は 001 からの数字で、999 を超えた場合 1000 からの数字) を付加した値になります。

## 補足

Windows Vista で環境変数 ProgramFiles (通常はシステムドライブの Program Files フォルダ)、または WinDir (通常はシステムドライブの Windows フォルダ) に設定されているフォルダ下のファイル、またはフォルダを指定する場合、注意が必要です。詳細は、「1.8.4 コマンドの動作」を参照してください。

## 例

```
' ファイル"C:\¥FDCOPY.TXT"を分割して一時フォルダ下に"FDCOPY.TXT.001"
' からsplCnt個の分割されたファイルを作成し、FDに格納する。
' (参考)1.44MBのFD = 1423KB = 1423 * 1024B
'          1.25MBのFD = 1221KB = 1221 * 1024B
'          720KBのFD = 713KB = 713 * 1024B
'          640KBのFD = 640KB = 640 * 1024B
Dim file1 ,size1 ,splCnt ,filePath
```

```

file1 = "FDCOPY.TXT"
size1 = 1423 * 1024
SplitFile ( "C: ¥"+file1 ,size1 ,_TEMP_ , ,splCnt )

Dim cnt1
For cnt1 = 1 To splCnt
    ' 分割後ファイル名の作成
    filePath = _TEMP_+file1
    Select Case Len ( cnt1 )
        Case 1
            filePath = filePath+".00"+cnt1
        Case 2
            filePath = filePath+".0"+cnt1
        Case Else
            filePath = filePath+"."+cnt1
    End Select

    ' 一時フォルダからFDへコピー
    Copy ( filePath , "A:¥" , Overwrite )
    If splCnt - cnt1 <= 0 Then
        MsgBox ( "FDへの複写が終了しました。" , OK )
        Exit For
    End
    MsgBox ( cnt1+1+"枚目のFDに交換してください。" , OKCancel )
    If _MSG_RTN_ = Cancel Then
        Exit For
    End
Next

```

対象バージョン

JP1/Script 05-10 以降

## 8.5.23 CatFiles （ファイルを統合する）

機能

ファイルを統合します。

JP1/Script 07-50 以降では、ラージファイルが使用できます。

形式

```

CatFiles ( PathName , [Option] , FilePath1 , [FilePath2] [,
FilePath3 , ...] )

```

指定項目

*PathName*

統合したファイルを格納するフォルダ名、またはファイル名を、文字列、または値を格納した変数名で、指定します。指定したファイル名がすでに存在している場合は上書きします。

*Option*

オプションを指定します。

この値には次の値を指定できます。

値	意味
Delete	ファイルを統合後、 <i>FilePathn</i> で指定されたファイルを削除します。

*FilePathn*

統合するファイルのフルパスを文字列、または値を格納した変数名で指定します。ファイル名にワイルドカードを指定することもできます。

ファイル名は最大 10 個指定でき、指定した順序で統合します。

JP1/Script 06-00 以降では、複数のファイルのフルパスの文字列を一つずつ格納した一次元の配列変数を変数名で指定することもできます。この場合は、11 個以上のファイル名を指定することができます。

ファイル名にワイルドカードが指定された場合は、該当するファイル名を昇順にソートして統合します。

## 説明

指定したファイルを統合し、統合したファイルを指定されたフォルダに格納します。コマンドが正常に実行された場合は真 (True) を、エラーが発生した場合は偽 (False) を、コマンドの実行結果として返します。

## 補足

Windows Vista で環境変数 ProgramFiles (通常はシステムドライブの Program Files フォルダ)、または WinDir (通常はシステムドライブの Windows フォルダ) に設定されているフォルダ下のファイル、またはフォルダを指定する場合、注意が必要です。詳細は、「1.8.4 コマンドの動作」を参照してください。

## 例

```
' FDに格納されたファイル"A:¥FDCOPY.TXT.*"を統合して、
' "C:¥SCRIPT¥FDCOPY.TXT"を作成する。
Dim dir1 ,file1
dir1 = "C:¥SCRIPT¥"
file1 = "FDCOPY.TXT"
CatFiles ( dir1+file1 , , "A:¥"+file1+"*" )
```

## 対象バージョン

JP1/Script 05-10 以降

## 8.5.24 SetStandardFile または SetStdFile (標準入力・出力・エラーファイルを設定する)

## 機能

Exec コマンドで呼び出すプロセスの標準入力、標準出力、および標準エラーファイルをオープンして設定します。

JP1/Script 07-50 以降では、ラージファイルが使用できます。

ただし、ファイル容量が 4,294,967,296 バイト以上のファイルは指定できません。

## 形式

```
SetStandardFile ( FilePath , Type [, Mode] )
SetStdFile ( FilePath , Type [, Mode] )
```

## 指定項目

*FilePath*

設定するファイルのフルパスを文字列、または値を格納した変数名で指定します。

*Type*

設定するファイルの種別を指定します。  
この値は次のどれかになります。

値	意味
StdInput	標準入力
StdOutput	標準出力
StdError	標準エラー

*Mode*

*Type* で指定されたファイルの種別が StdOutput、または StdError の場合、  
ファイル作成時の動作を指定します。  
この値は次のどれかになります。

値	意味
Create	新規モード
Append	追加モード

この値は省略できます。省略した場合、Append を仮定します。

*Type* に StdInput を指定した場合は常に新規モードとなるため、この値は無効になります。

## 説明

このコマンドを実行した後の Exec コマンドで呼び出すプロセスの標準入力、標準出力、および標準エラーファイルをオープンして設定します。コマンドが正常に実行された場合は真 (True) を、エラーが発生した場合は偽 (False) を、コマンドの実行結果として返します。

設定されたファイルは ResetStandardFile コマンドが実行されるまで有効になり、オープンされたままです。

なお、このコマンドで設定したファイルは NetExec コマンドでは有効になりません。

## 補足

Windows Vista で環境変数 ProgramFiles (通常はシステムドライブの Program Files フォルダ)、または WinDir (通常はシステムドライブの Windows フォルダ)

に設定されているフォルダ下のファイルを指定する場合、注意が必要です。詳細は、「1.8.4 コマンドの動作」を参照してください。

**！ 注意事項**

標準入力、標準出力、および標準エラーのうち 1 種類でも設定した場合に、Exec コマンドで実行するプロセスが正常に実行されないことがあります。その場合は、空のファイルを用意し、標準入力、標準出力、および標準エラーの 3 種類すべてを設定してください。

例

```
SetStandardFile ( _SCF_+"BatOut.TXT" ,StdOutput ,Create )
SetStandardFile ( _SCF_+"BatErr.TXT" ,StdError )
If Exec ( _SCF_+"Backup.BAT" ,True ) Then
    MessageBox ( "Execコマンド 成功！" )
Else
    MessageBox ( "Execコマンド 失敗！" )
End
```

対象バージョン

JP1/Script 05-10 以降

8.5.25 ResetStandardFile または ResetStdFile （標準入力・出力・エラーファイルを解除する）

機能

Exec コマンドで呼び出すプロセスの標準入力、標準出力、および標準エラーファイルをクローズして解除します。  
JP1/Script 07-50 以降では、ラージファイルが使用できます。

形式

```
ResetStandardFile ( [Type] )
ResetStdFile ( [Type] )
```

指定項目

*Type*  
設定を解除するファイルの種別を指定します。  
この値は次のどれかになります。

値	意味
StdInput	標準入力
StdOutput	標準出力
StdError	標準エラー

この値は省略できます。省略した場合、設定されているすべてのファイルを解除します。



## 説明

SetStandardFile コマンドで設定されたプロセスの標準入力、標準出力、および標準エラーファイルをクローズして解除します。コマンドが正常に実行された場合は真 (True) を、エラーが発生した場合は偽 (False) を、コマンドの実行結果として返します。

## 例

```
SetStandardFile ( _SCF_+"BatOut1.TXT" ,StdOutput )
SetStandardFile ( _SCF_+"BatErr.TXT" ,StdError )
Exec ( _SCF_+"Backup.BAT" ,True )

ResetStandardFile ( StdOutput )
SetStandardFile ( _SCF_+"BatOut2.TXT" ,StdOutput )
Exec ( _SCF_+"Build.BAT" ,True )
```

## 対象バージョン

JP1/Script 05-10 以降

## 8.5.26 SplitPath (フルパスを解析する)

## 機能

フルパスを解析します。

## 形式

```
SplitPath ( FullPath , {DrvNameBuff} , {DirNameBuff} ,
{LblNameBuff} [, ExtNameBuff] )
```

## 指定項目

*FullPath*

解析するフルパスを文字列、または値を格納した変数名で指定します。

*DrvNameBuff*

ドライブ名を受け取る変数名を指定します。必要がない場合は省略します。

*DirNameBuff*

フォルダ名を受け取る変数名を指定します。必要がない場合は省略します。

*LblNameBuff*

ファイルラベル名を受け取る変数名を指定します。必要がない場合は省略します。

*ExtNameBuff*

拡張子を受け取る変数名を指定します。必要がない場合は省略します。

## 説明

与えられたフルパスを、ドライブ名、フォルダ名、ファイルラベル名、拡張子に分解して各変数に格納します。コマンドが正常に実行された場合は真 (True) を、エラーが発生した場合は偽 (False) を、コマンドの実行結果として返します。ドライブ名にはコロン (:) が付加されます。フォルダ名には前後に円記号 (¥) が

## 8. 基本コマンド

付加されます。拡張子にはピリオド ( . ) が付加されます。

### 例

```
Dim drv1 ,dir1 ,lbl1 ,ext1
SplitPath ( "C:¥WINDOWS¥WINFILE.EXE" ,drv1 ,dir1 ,lbl1 ,ext1 )

drv1  C:
dir1  ¥WINDOWS¥
lbl1  WINFILE
ext1  .EXE
```

### 対象バージョン

JP1/Script 01-00 以降

## 8.5.27 MakePath （フルパスを作成する）

### 機能

フルパスを作成します。

### 形式

```
MakePath ( FullPathBuff , DrvName , DirName , LblName , ExtName )
```

### 指定項目

*FullPathBuff*

作成したフルパスを受け取る変数名を指定します。

*DrvName*

ドライブ名を文字列、または値を格納した変数名で指定します。

*DirName*

フォルダ名を文字列、または値を格納した変数名で指定します。

*LblName*

ファイルラベル名を文字列、または値を格納した変数名で指定します。

*ExtName*

拡張子を文字列、または値を格納した変数名で指定します。

### 説明

ドライブ名、フォルダ名、ファイルラベル名、拡張子を結合してフルパスを作成し、変数に格納します。コマンドが正常に実行された場合は真 ( True ) を、エラーが発生した場合は偽 ( False ) を、コマンドの実行結果として返します。

ドライブ名のコロン ( : ) を省略しても自動的に付加されます。フォルダ名の前後の円記号 ( ¥ ) を省略しても自動的に付加されます。拡張子のピリオド ( . ) を省略しても自動的に付加されます。

### 例

```
SplitPath  OriginalFile  Drv  Dir  Lbl  Ext
MakePath   BackupFile   Drv  Dir  Lbl  ".BAK"
```

対象バージョン

JP1/Script 01-00 以降

## 8.5.28 SetPath （実行フォルダのパスを設定する）

機能

実行フォルダのパスを設定します。

形式

SetPath ( [ *DirPath* ] )

指定項目

*DirPath*

設定するフォルダパスを文字列、または値を格納した変数名で指定します。  
この値は省略できます。省略した場合、カレントフォルダを仮定します。

説明

指定されたフォルダパスを実行フォルダのパスに設定します。\_BIN\_ 予約変数も指定されたフォルダパスに変更されます。

コマンドが正常に実行された場合は真 ( True ) を、エラーが発生した場合は偽 ( False ) を、コマンドの実行結果として返します。

指定されたフォルダパスが存在しない場合は、何もしないで、常に偽 ( False ) を返します。

例

```
OutDir = %1
If IsEmptyDir ( OutDir ) Then
    SetPath ( OutDir )      '実行フォルダを設定する。
End
Exec ( OutDir+"ABC.EXE" ,True )
SetPath                    '実行フォルダをカレントに戻す。
```

対象バージョン

JP1/Script 01-00 以降

## 8.5.29 GetPath （実行フォルダのパスを取得する）

機能

実行フォルダのパスを取得します。

形式

GetPath

このコマンドに引数はありません。

説明

現在の実行フォルダのパスを取得し、値をコマンドの実行結果として返します。

## 8. 基本コマンド

実行フォルダがルートフォルダの場合を除き、フォルダ名の末尾には(¥)は付加されません。フォルダ名の末尾に(¥)を付加する場合は、次のレジストリに値を設定してください。

〔レジストリキー〕

HKEY\_LOCAL\_MACHINE¥Software¥Hitachi¥JP1/Script¥SPTX

〔値名〕

GetPath\_Opt

〔値のデータタイプ〕

DWORD

〔値〕

1：末尾に(¥)を付加する

例

```
Dim drv1
SplitPath ( GetPath ,drv1 )
If drv1 <> "C:" Then
    SetPath
End
```

対象バージョン

JP1/Script 01-00 以降

### 8.5.30 SetVolumeLabel または SetVolLabel （ボリュームラベルを設定する）

機能

ディスクのボリュームラベルを設定します。

形式

```
SetVolumeLabel ( DiskName [, LabelName] )
SetVolLabel ( DiskName [, LabelName] )
```

指定項目

*DiskName*

ディスクのボリュームを文字列、または値を格納した変数名で指定します。  
ディスクのコロン(：)は省略してもかまいません。

*LabelName*

設定するボリュームラベル名を文字列、または値を格納した変数名で指定します。  
この値は省略できます。省略した場合、長さ 0 の文字列("")を仮定します。

説明

ディスクのボリュームラベルを設定します。コマンドが正常に実行された場合は真 (True) を、エラーが発生した場合は偽 (False) を、コマンドの実行結果として返

します。

#### 補足

Windows Vista では、管理者権限でスクリプトファイルを実行してください。詳細は、「1.8.4 コマンドの動作」を参照してください。

#### 例

```
Dim drvName ,volName
drvName = "E:"
volName = GetVolumeLabel ( drvName )
If volName = Empty Then
    SetVolumeLabel ( drvName ,"Script共有ドライブ" )
End
```

#### 対象バージョン

JP1/Script 05-10 以降

## 8.5.31 GetVolumeLabel または GetVolLabel （ボリュームラベルを取得する）

#### 機能

ディスクのボリュームラベルを取得します。

#### 形式

```
GetVolumeLabel ( DiskName [, FileSystemBuff] )
GetVolLabel ( DiskName [, FileSystemBuff] )
```

#### 指定項目

##### *DiskName*

取得するディスクのボリュームを文字列、または値を格納した変数名で指定します。ディスクのコロン（:）は省略してもかまいません。

##### *FileSystemBuff*

ファイルシステム名（FAT ,HPFS ,NTFS など）を受け取る変数名を指定します。必要がない場合は省略します。

#### 説明

指定されたディスクのボリュームラベルを取得し、コマンドの実行結果として返します。

指定されたディスクが存在しない場合は、長さ 0 の文字列（""）を返します。

#### 例

```
Dim volName ,fileSystem
volName = GetVolumeLabel ( "E:" ,fileSystem )
```

#### 対象バージョン

JP1/Script 05-10 以降

### 8.5.32 GetDiskFreeSpace （ディスク空き容量を取得する）

機能

ディスクの空き容量を取得します。

形式

GetDiskFreeSpace ( *DiskName* [ , *UnitofByte* ] )

指定項目

*DiskName*

取得するディスクのボリュームを文字列、または値を格納した変数名で指定します。ディスクのコロン (:) は省略してもかまいません。

*UnitofByte*

取得する容量の単位を指定します。  
この値は次のどれかになります。

値	意味
KB	キロバイト
MB	メガバイト

空き容量は切り捨てた値を返します。例えば、UnitofByte に MB が指定され、容量が 1 メガバイトに満たない場合は、0 を返します。  
この値は省略できます。省略した場合、KB を仮定します。

説明

指定されたディスクの空き容量を取得し、コマンドの実行結果として返します。  
エラーが発生した場合は、0 を返します。

例

```
Dim MSize, BSize
MSize = GetDiskFreeSpace ( "A:" ,MB )
BSize = MSize * 1024 *1024
MessageBox ( "FDの空き容量は"+ BSize +"バイトです。" )
```

対象バージョン

JP1/Script 05-10 以降

### 8.5.33 Copy （ファイルをコピーする）

機能

ファイルをコピーします。  
JP1/Script 07-50 以降では、ラージファイルが使用できます。

形式

Copy ( *OldFileName* , *NewPathName* , [ *Option1* ] , [ *Option2* ] ,

```
{Option3} , {Option4} , {Option5} , {ExceptFileName} ,  
{Option6} { , Option7} )
```

#### 指定項目

##### *OldFileName*

コピー元（移動元）ファイル名を文字列、または値を格納した変数名で指定します。

指定されたファイル名の最後の文字がアンダースコア（`_`）の場合、`COMPRESS` コマンドで圧縮されたファイルとみなし、展開しながらコピーします。ただし、*NewPathName* でコピー先ファイル名の指定がないかぎり、ファイル名は変更しません。最後の文字はアンダースコア（`_`）のままです。ファイル名にワイルドカードを指定することもできます。

##### *NewPathName*

コピー先（移動先）フォルダ名、またはファイル名を、文字列、または値を格納した変数名で指定します。

##### *Option1 ~ 5*

コピーオプションを指定します。

複数のコピーオプションを指定できます。

この値には次の値を指定できます。

値	意味
VersionUp	新しいファイルだけを上書きします。 バージョン情報を比較し、バージョン情報が取得できない場合は、ファイルの日付で新旧を比較します。
Overwrite	すべて上書きします。
NoOverwrite（05-10以降）	コピー先フォルダにすでに存在するファイルはコピーしません。
OverwriteOnly（05-10以降）	コピー先フォルダに存在するファイルだけをコピーします。
File	既存のファイルは拡張子を .000 ~ .999 のうち空いているものに変更して残し、コピーします。
Backup（05-10以降）	アーカイブ属性の付いたファイルだけをコピーし、コピー後にコピー元ファイルのアーカイブ属性をクリアします
Move（05-10以降）	ファイルを移動します。コピーは行いません。 これ以外の Versionup, Overwrite, NoOverwrite とともに指定でき、移動先フォルダにすでにファイルが存在する場合はともに指定されたオプションに従って移動します。
SubDirToo（05-10以降）	下位のフォルダまで、その構造を保ったままコピーします。 ファイルが存在しない下位のフォルダはコピーしません。これ以外のオプションとともに指定できます。
Trace（05-10以降）	コピーしたファイル名を実行トレースファイルに出力します。 これ以外のオプションと同時に指定できます。
ErrSkip（05-10以降）	コピー元ファイルの排他エラーやアクセスエラーが発生しても無視してコピーを続行します。これ以外のオプションと同時に指定できます。

## 8. 基本コマンド

この値は省略できます。省略した場合、Overwrite を仮定します。

### *ExceptFileName (05-10 以降)*

*OldFileName* で指定したファイル名の中でコピー（移動）の対象にしないファイルがある場合、そのファイル名を文字列、または値を格納した変数名で指定します。

複数のファイル名を指定する場合は、セミコロン（;）で区切った文字列、または値を格納した変数名で指定します。

JP1/Script 06-00 以降では、複数のファイル名の文字列を一つずつ格納した一次元の配列変数を変数名で指定することもできます。

ファイル名にワイルドカードを指定することもできます。

### *Option6 ~ 7 (06-51 以降)*

コピーオプションを指定します。

この値には次の値を指定できます。

値	意味
Security	ファイルにセキュリティ情報がある場合、セキュリティ情報もコピーします。これ以外のオプションと同時に指定できます。
ErrSkip2 (06-71 以降)	コピー先ファイルの排他エラーやアクセスエラーが発生しても無視してコピーを続行します。 無視したファイルの数は <code>_COPY_SKIP2_CNT</code> 予約変数に格納します。 これ以外のオプションと同時に指定できます。

## 説明

指定されたコピーオプションでファイルをコピーします。コマンドが正常に実行された場合は真（True）を、エラーが発生した場合は偽（False）を、コマンドの実行結果として返します。

コピーの結果は `_COPY_RTN` 予約変数に格納されます。格納される値は次のどれかになります。

値	意味
Overwrite	ファイルのバージョンをチェックしないでコピーした。
VersionUp	ファイルのバージョンをチェックしてコピーした。
Pile	既存ファイルの拡張子を変更して残しコピーした。
Backup (05-10 以降)	アーカイブ属性の付いたファイルだけをコピーし、アーカイブ属性をクリアした。
Move (05-10 以降)	ファイルを移動した。
Skip	ファイルをコピーしなかった。

コピーしたファイルの数は `_COPY_CNT` 予約変数に格納されます。コピーしなかったファイルの数は `_COPY_SKIP_CNT` 予約変数に格納されます。

## 補足



Windows Vista で環境変数 ProgramFiles (通常はシステムドライブの Program Files フォルダ), または WinDir (通常はシステムドライブの Windows フォルダ) に設定されているフォルダ下のファイル, またはフォルダを指定する場合, 注意が必要です。詳細は, 「1.8.4 コマンドの動作」を参照してください。

例

```
Dim inDir ,outDir
inDir = _SCF_"Inst¥"
outDir = _SCF_"Inst_Backup¥"
Copy ( inDir+"CTL3D32.DLL" ,outDir+"CTL3D32.DLL" ,VersionUp )
Copy ( inDir+"SCRIPT.EX_" ,outDir+" SCRIPT.EXE" ,Overwrite )
Copy ( inDir+"SCR*.*" ,outDir )
Copy ( inDir+"*" ,outDir ,Backup ,ErrSkip , , , _
,inDir+"*.EXE;"+"inDir+"*.DLL" )
```

対象バージョン

JP1/Script 01-00 以降

## 8.6 メッセージ出力コマンド

---

### 8.6.1 InputBox (メッセージとテキストボックスを表示する)

#### 機能

ダイアログボックスにメッセージとテキストボックスを表示します。テキストが入力されるか、ボタンがクリックされるとテキストボックスの内容を返します。

#### 形式

```
InputBox ( [Text] , [Title] , [xPos] , [yPos] , InputBuff1 ,
           [Caption1] , [InputBuff2] , [Caption2] , ... )
```

#### 指定項目

##### *Text*

ダイアログボックス内にメッセージとして表示する文字列を文字列、または値を格納した変数名で指定します。メッセージに含まれる文字列、"`¥r`"、"`¥n`"、"`¥t`"、"`¥¥`" は、それぞれ対応するコントロールコードとして処理されます。コントロールコードとして処理しない場合については、「6.1.11 コーディング規則」を参照してください。

##### *Title*

メッセージボックスのタイトルバーに表示するタイトルを文字列、または値を格納した変数名で指定します。  
この値は省略できます。省略した場合、"`JP1/Script InputBox`" が表示されます。

##### *xPos*

画面の左端からダイアログボックスの左端までの水平方向の距離を数値、または値を格納した変数名で指定します。距離は左端を 0 としたピクセル単位で指定します。この値を省略すると、ダイアログボックスは水平方向に対して画面の中央の位置に配置されます。また、負の値を指定した場合、0 を仮定します。

##### *yPos*

画面の上端からダイアログボックスの上端までの垂直方向の距離を数値、または値を格納した変数名で指定します。距離は上端を 0 としたピクセル単位で指定します。この値を省略すると、ダイアログボックスは垂直方向に対して画面の中央の位置に配置されます。また、負の値を指定した場合、0 を仮定します。

##### *InputBuff1 ~ 4*

テキストボックスに入力された文字列を格納する変数名を指定します。テキストボックスは最大 4 個指定できます。この変数にあらかじめ値を設定しておくと初期値としてテキストボックスに表示することができます。テキストボックスに何も入力されなかった場合は、長さ 0 の文字列 ("") を格納します。

*Caption1 ~ 4*

*InputBuff1 ~ 4* で指定したテキストボックスのキャプションを文字列、または値を格納した変数名で指定します。指定できる文字数は、半角文字で 40 文字以内です。

この値は省略できます。省略した場合、キャプションは表示されません。

## 説明

指定されたパラメタでメッセージとテキストボックスを表示します。テキストボックスは最大 4 個指定できます。

コマンドが正常に実行された場合は真 (True) を、エラーが発生した場合は偽 (False) を、コマンドの実行結果として返します。

メッセージボックスには「OK」ボタンと「キャンセル」ボタンが付加されます。

「OK」ボタン、または Enter キーが押されると、テキストボックスの内容を返します。「キャンセル」ボタンが押された場合は、長さ 0 の文字列 ("" ) を返します。

入力されたボタンの値は `_MSG_RTN_` 予約変数に格納されます。格納される値は次のどれかになります。

値	入力されたボタン
OK	[OK]
CANCEL	[キャンセル]

## 補足

スクリプトをサービスで起動する場合は、このコマンドは使用できません。使用すると実行エラーになります。

## 例

```
Dim compName
Dim userName
InputBox ( "会社名と氏名を入力してください。", "データ登録", 100, 100, _
    compName, "会社名", userName, "氏名" )
```

## 対象バージョン

JP1/Script 01-00 以降

## 8.6.2 Message ( ファイル / ウィンドウにテキストを出力する )

## 機能

ファイル、またはウィンドウにテキストを出力します。

また、出力したウィンドウやテキストを消去します。

ウィンドウ出力については、JP1/Script 01-01 からサポートしています。

## 形式

`Message ( Target , [OutputName] , [Text] , [LineLength] ,`

## 8. 基本コマンド

[MaxLines] , [xPos] [, yPos] )

### 指定項目

#### Target

テキストの出力先を指定します。  
この値は次のどれかになります。

値	意味
Target_File	ファイルに出力します。
Target_Dispon ( 01-01 以降 )	ウィンドウに出力します。
Target_Dispclear ( 01-01 以降 )	ウィンドウに表示しているメッセージテキストを消去します。
Target_Dispoﬀ ( 01-01 以降 )	表示しているウィンドウを消去します。
Target_SPASFile ( 05-20 以降 )	実行中スクリプトの解析トレースファイルに出力します。
Target_SPXFile ( 05-20 以降 )	実行中スクリプトの実行トレースファイルに出力します。

#### OutputName

- Target に Target\_File を指定した場合  
出力先ファイルのフルパスを文字列、または値を格納した変数名で指定します。ファイル名の拡張子が省略された場合、自動的に拡張子 (.TXT) を付加します。ただし、ファイル名の末尾が拡張子のピリオド ( . ) である場合、JP1/Script 06-00 以降では、拡張子のないファイル名とみなします。
- Target に Target\_Dispon , Target\_Dispclear , Target\_Dispoﬀ を指定した場合  
ウィンドウのタイトル名を文字列、または値を格納した変数名で指定します。
- Target に Target\_SPASFile , Target\_SPXFile を指定した場合  
実行中スクリプトの各トレースファイルが出力先ファイルになるため、この値は省略できます。値を省略しないで指定した場合、この値をウィンドウタイトル名とみなし各トレースファイルとウィンドウの両方に出力します。

#### Text

- Target に Target\_File , Target\_Dispon , Target\_SPASFile , Target\_SPXFile を指定した場合  
出力するメッセージテキストを文字列、または値を格納した変数名で指定します。  
メッセージテキストに含まれる文字列、"%r" , "%n" , "%t" , "%¥" は、それぞれ対応するコントロールコードとして処理されます。コントロールコードとして処理しない場合については、「6.1.11 コーディング規則」を参照してください。
- Target に Target\_Dispclear , Target\_Dispoﬀ を指定した場合  
この値は無効になります。

#### LineLength

- *Target* に *Target\_File* を指定した場合  
出力するメッセージテキストの 1 行の長さを数値, または値を格納した変数名で指定します。メッセージテキストに含まれる "¥r", "¥n" は除いた値を指定します。  
この値は *OutputName* で指定されたファイル名に対して, 初めて Message コマンドを実行するときに指定します。2 回目以降の Message コマンドを実行するときに *LineLength* の値が変更された場合, 1 行の長さを変更して *OutputName* で指定されたファイルを新しく作成します。  
この値は省略できます。省略した場合, 実行環境ファイルで定義されたユーザトレース情報の最大列数を仮定します。
- *Target* に *Target\_Dispon*, *Target\_Dispclear*, *Target\_Dispooff* を指定した場合  
この値は無効になります。
- *Target* に *Target\_SPAMfile*, *Target\_SPXFile* を指定した場合  
この値は無効になり, 実行環境ファイルで定義されたトレース情報の最大列数を仮定します。

#### *MaxLines*

- *Target* に *Target\_File* を指定した場合  
出力するメッセージテキストの最大行数を数値, または値を格納した変数名で指定します。  
この値は *LineLength* と同様に, *OutputName* で指定されたファイル名に対して, 初めて Message コマンドを実行するときに指定します。2 回目以降の Message コマンドを実行するときに *MaxLines* の値が変更された場合, 最大行数を変更して *OutputName* で指定されたファイルを新しく作成します。  
この値は省略できます。省略した場合, 実行環境ファイルで定義されたユーザトレース情報の最大行数を仮定します。
- *Target* に *Target\_Dispon*, *Target\_Dispclear*, *Target\_Dispooff* を指定した場合  
この値は無効になります。
- *Target* に *Target\_SPAMfile*, *Target\_SPXFile* を指定した場合  
この値は無効になり, 実行環境ファイルで定義されたトレース情報の最大行数を仮定します。

#### *xPos*

- *Target* に *Target\_Dispon*, *Target\_Dispclear* を指定した場合
- *Target* に *Target\_SPAMfile*, または *Target\_SPXFile* を指定してウィンドウにも出力する場合  
画面の左端からウィンドウの左端までの水平方向の距離を数値, または値を格納した変数名で指定します。  
距離は左端を 0 としたピクセル単位で指定します。  
この値を省略すると, ウィンドウは水平方向に対して画面の中央の位置に配置されます。また, 負の値を指定した場合, 0 を仮定します。

## 8. 基本コマンド

- *Target* に *Target\_File* , *Target\_DispOff* を指定した場合
- *Target* に *Target\_SPAMFile* , または *Target\_SPXFile* を指定してウィンドウには出力しない場合  
この値は無効になります。

### *yPos*

- *Target* に *Target\_Dispon* , *Target\_Dispclear* を指定した場合
- *Target* に *Target\_SPAMFile* または *Target\_SPXFile* を指定してウィンドウにも出力する場合  
画面の上端からウィンドウの上端までの垂直方向の距離を数値 , または値を格納した変数名で指定します。  
距離は上端を 0 としたピクセル単位で指定します。  
この値を省略すると , ウィンドウは垂直方向に対して画面の中央の位置に配置されます。また , 負の値を指定した場合 , 0 を仮定します。
- *Target* に *Target\_File* , *Target\_DispoFF* を指定した場合
- *Target* に *Target\_SPAMFile* , または *Target\_SPXFile* を指定してウィンドウには出力しない場合  
この値は無効になります。

### 説明

指定されたファイル , またはウィンドウに指定されたメッセージテキストを出力します。

また , 出力したウィンドウの消去 , テキストメッセージの消去をします。コマンドが正常に実行された場合は真 ( True ) を , エラーが発生した場合は偽 ( False ) を , コマンドの実行結果として返します。

### 補足

- スクリプトをサービスで起動する場合は , このコマンドのウィンドウ出力は使用できません。使用すると実行エラーになります。
- Windows Vista で環境変数 ProgramFiles ( 通常はシステムドライブの Program Files フォルダ ) , または WinDir ( 通常はシステムドライブの Windows フォルダ ) に設定されているフォルダ下のファイルを指定する場合 , 注意が必要です。詳細は , 「1.8.4 コマンドの動作」を参照してください。

**！ 注意事項**

- 実行環境で複数起動の指定をしたスクリプト中に、Target\_SPXFile、または Target\_SPXFile を指定した Message コマンドがあり、そのスクリプトファイルを同時に複数実行した場合、同一の解析トレースファイル、または実行トレースファイルにメッセージを出力します。このため、ファイルに排他制御がかかり、実行性能が低下することがありますので注意が必要です。また、Target\_File で出力するファイル名が固定の場合も同様です。
- Message コマンドで出力するファイルは JP1/Cm2/Extensible Agent や JP1/Base のログファイルトラップ機能の対象にはなりません。
- Target\_File で作成したファイルを他のアプリケーションでテキスト出力しないでください。その後の Message コマンドの出力結果が保証されません。
- Message コマンドで Target\_File を指定して、ユニークなファイル名称を持ったユーザトレースファイルを大量に作成する場合、トレース管理ファイルの増大によって、スクリプトの実行性能に影響を与える場合があります。また、以下の例のようにスクリプト実行が、メモリ不足で異常終了したり、コマンド実行がメモリ不足でエラーになる場合があります。  
スクリプト実行が終了コード「20」で異常終了する。  
Copy コマンドで「メモリ不足が発生しました」となる。  
このような場合は、Message コマンドではなく TextOpen/TextWrite/TextClose コマンドによってトレースファイルを作成してください。

**例**

```

・ スクリプト実行の履歴を "Logging.txt" に書き込む。
Message ( Target_File , _BIN_+"Logging.txt" , "実行開始" , 30 , 100 )
:
Message ( Target_File , _BIN_+"Logging.txt" , "実行終了" )

・ スクリプト実行の履歴状況をディスプレイに表示する。
Message ( Target_DispOn , "実行状況" , "開始しました。" , , , 100 , 100 )
:
Message ( Target_DispOn , "実行状況" , "終了しました。" , , , 100 , 100 )
Sleep ( 3000 )
Message ( Target_DispOff , "実行状況" )

```

**対象バージョン**

JP1/Script 01-00 以降

### 8.6.3 MessageBox （ダイアログボックスにメッセージを表示する）

**機能**

ダイアログボックスにメッセージを表示します。メッセージボックスにはボタンやアイコンの付加を指定できます。ボタンがクリックされるのを待って、どのボタンがクリックされたかを示す値を返します。

**形式**

8. 基本コマンド

```
MessageBox ( Text , [Buttons] , [DefaultBtn] , [IconStyle] ,  
[ModalStyle] [, Title] )
```

指定項目

*Text*

ダイアログボックス内にメッセージとして表示する文字列を文字列、または値を格納した変数名で指定します。メッセージに含まれる文字列、"**%i**"、"**%n**"、"**%t**"、"**%s**" は、それぞれ対応するコントロールコードとして処理されます。コントロールコードとして処理しない場合については、「6.1.11 コーディング規則」を参照してください。

*Buttons*

メッセージボックスに付加するボタンの種類と個数を指定します。  
この値は次のどれかになります。

値	意味
OK	[OK] ボタンだけを表示します。
OKCancel	[OK] ボタンと [キャンセル] ボタンを表示します。
YesNo	[はい] ボタンと [いいえ] ボタンを表示します。
YesNoCancel	[はい]、[いいえ]、および [キャンセル] の三つのボタンを表示します。
RetryCancel	[再試行] ボタンと [キャンセル] ボタンを表示します。
AbortRetryIgnore または Abort	[中止]、[再試行]、および [無視] の三つのボタンを表示します。

この値は省略できます。省略した場合は、[OK] ボタンだけを表示します。

*DefaultBtn*

*Buttons* で指定されたボタンのなかでどのボタンが標準ボタンになるかを指定します。  
この値は次のどれかになります。

値	意味
1	第 1 ボタンを標準ボタンにします。
2	第 2 ボタンを標準ボタンにします。
3	第 3 ボタンを標準ボタンにします。

この値は省略できます。省略した場合、1 が仮定されます。

*IconStyle*

メッセージボックスに付加するアイコンのスタイルを指定します。  
この値は次のどれかになります。



値	意味
Exclamation または Ex	感嘆符「!」
Information または Info	円の中に「i」の字
Question	疑問符「?」
Stop	「STOP」

この値は省略できます。省略した場合、アイコンは表示されません。

#### *ModalStyle*

メッセージボックスがモーダルかどうかを指定します。

この値は次のどれかになります。

値	意味
ApplicationModal または AppliModal	アプリケーションモーダルに設定します。ほかのアプリケーションのウィンドウに移動して作業することはできません。
SystemModal または SysModal	システムモーダルに設定します。ユーザがすぐ気付く必要があるような重大なエラーを通知するために使用します。

この値は省略できます。省略した場合、ApplicationModal が仮定されます。

#### *Title*

メッセージボックスのタイトルバーに表示するタイトルを文字列、または値を格納した変数名で指定します。

この値は省略できます。省略した場合、"JP1/Script MessageBox" が表示されます。

#### 説明

指定されたパラメタでメッセージボックスを表示します。コマンドが正常に実行された場合は真 (True) を、エラーが発生した場合は偽 (False) を、コマンドの実行結果として返します。

入力されたボタンの値は \_MSG\_RTN\_ 予約変数に格納されます。格納される値は次のどれかになります。

値	入力されたボタン
OK	[OK]
CANCEL	[キャンセル]
ABORT	[中止]
RETRY	[再試行]
IGNORE	[無視]
YES	[はい]

8. 基本コマンド

値	入力されたボタン
NO	[いいえ]

[キャンセル] ボタンが表示されているダイアログボックスでは、Esc キーを押すと、  
[キャンセル] ボタンをクリックしたときと同じ結果になります。

補足

スクリプトをサービスで起動する場合は、このコマンドは使用できません。使用すると実行エラーになります。

例

```
MessageBox ( "終了してもいいですか?" ,OKCancel , ,Question )
If _MSG_RTN_ <>CANCEL Then
    Exit
End
```

対象バージョン

JP1/Script 01-00 以降

8.6.4 MessageEventLog （イベントビューアにメッセージを出力する）

機能

イベントビューアのアプリケーションログにメッセージを出力します。

形式

```
MessageEventLog ( Text , [LogType] [, Option] )
```

指定項目

*Text*

イベントビューアのアプリケーションログに出力するメッセージを文字列、または値を格納した変数名で指定します。メッセージに含まれる文字列、"%r", "%n", "%t", "%¥" は、それぞれ対応するコントロールコードとして処理されます。コントロールコードとして処理しない場合については、「6.1.11 コーディング規則」を参照してください。

*LogType*

出力するメッセージのイベントの種類を指定します。  
この値は次のどれかになります。

値	意味
Error, または Err	エラー
Warning	警告
Information, または Info	情報

値	意味
AuditSuccess	成功の監査
AuditFailure	失敗の監査

この値は省略できます。省略した場合、Information を仮定します。

*Option (06-51 以降)*

オプションを指定します。

この値には次の値を指定できます。

値	意味
TextOnly	メッセージの先頭にユーザ名と実行したスクリプトファイル名を付加しません。Text で指定されたメッセージだけを出力します。

#### 説明

指定されたパラメタでイベントビューアのアプリケーションログにメッセージを出力します。コマンドが正常に実行された場合は真 (True) を、エラーが発生した場合は偽 (False) を、コマンドの実行結果として返します。

このコマンドを使用して出力したイベントログのイベント ID は 3 になります。

また、イベントログの出力されるメッセージの先頭にはユーザ名と実行したスクリプトファイル名が付加されます。オプションに TextOnly が指定されている場合は付加しません。

なお、実行環境ファイルで定義されたコマンドラインやレジストリのコマンドラインに /NOEVLOG や /NOEVLOG(3) などの指定がある場合、この指定にかかわらずコマンドを実行します。

#### 例

```
If Exec ( "ABC.EXE" ,True ) = False Then
  MessageEventLog ( "ABC.EXEの呼び出し失敗" ,Error )
End
```

#### 対象バージョン

JP1/Script 01-01 以降

## 8.6.5 IMEventMessage ( JP1/IM または JP1/Base にイベントを発行する )

#### 機能

JP1/IM、または JP1/Base にイベントを発行します。

#### 形式

```
IMEventMessage ( Text , [Severity] , [UserName] ,
  [RootObjName] , [Occurrence] , [StartDate] , [StartTime] ,
  [EndDate] , [EndTime] , [ResultCode] )
```

## 8. 基本コマンド

### 指定項目

#### *Text*

JP1/IM のイベントコンソールにイベントとして表示するメッセージを文字列、または値を格納した変数名で指定します。

#### *Severity*

イベントの重大度を指定します。  
この値は次のどれかになります。

値	意味
Emergency または Emergence	緊急 パニック状態を表します。 注意 Emergency ではなく Emergence を指定した場合、JP1/IM-Console の自動アクションでは JP1 イベントとみなされないため動作しません。
Alertness	警戒 即座に修正を必要とする状態を表します。
Critical	致命的 重大なエラーを表します。
Error または Err	アプリケーションでのエラーを表します。
Warning	警告 警告メッセージを表します。
Notice	通知 エラー状態ではないが、特別な処理を必要とすることを表します。
Information または Info	情報 情報メッセージを表します。
Debug	デバッグ プログラムをデバッグするときに使用する情報を表します。

この値は省略できます。省略した場合、イベントコンソールの重大度の欄には Information が仮定されます。

#### *UserName*

イベントのユーザ名を文字列、または値を格納した変数名で指定します。  
この値は省略できます。省略した場合、イベントコンソールのユーザ名の欄は空欄になります。

#### *RootObjName*

イベントの登録名を文字列、または値を格納した変数名で指定します。  
この値は省略できます。省略した場合、イベントコンソールの登録名の欄は空欄になります。

#### *Occurrence*

イベントの事象種別を指定します。  
この値は次のどれかになります。

値	意味
Start	実行開始
End	実行終了
Create	定義の作成
Destroy	定義の削除
Modify	定義内容の変更
Submit	実行可能状態への変更
UnSubmit	実行可能状態からの離脱
Pause	実行の一時停止
Release	一時停止の解除
Restart	再実行の開始

この値は省略できます。省略した場合、イベントコンソールの事象種別の欄は空欄になります。

#### *StartDate*

イベントの開始日付を yyyy/mm/dd の形式で指定します。

この値は省略できます。省略した場合、イベントコンソールの開始時刻の欄は空欄になります。

#### *StartTime*

イベントの開始時刻を hh:mm:ss の形式で指定します。

この値は省略できます。省略した場合、イベントコンソールの開始時刻の欄は空欄になります。

#### *EndDate*

イベントの終了日付を yyyy/mm/dd の形式で指定します。

この値は省略できます。省略した場合、イベントコンソールの終了日付の欄は空欄になります。

#### *EndTime*

イベントの終了時刻を hh:mm:ss の形式で指定します。

この値は省略できます。省略した場合、イベントコンソールの終了時刻の欄は空欄になります。

#### *ResultCode*

イベントの終了コードを数値、または値を格納した変数名で指定します。

この値は省略できます。省略した場合、イベントコンソールの終了コードの欄は空欄になります。

### 説明

指定されたパラメタで JP1/IM、または JP1/Base にイベントを発行します。コマンドが正常に実行された場合は真 (True) を、エラーが発生した場合は偽 (False) を、コマンドの実行結果として返します。

## 8. 基本コマンド

このコマンドを使用して発行したイベントのイベント ID は 00003B03 になります。  
また、イベントのプロダクト名は /HITACHI/JP1/SCRIPT に、登録名タイプは  
PROCESS になります。

### 補足

JP1/IM - Console のイベントコンソールの拡張属性説明ファイルは、インストール  
フォルダの「DATA」フォルダ下に格納されます。ファイル名は、  
"SCRIPT\_BASE.ja" です。

### ！ 注意事項

JP1/Base へイベントを発行する場合、JP1/Script は物理的ホストに対してだけ発行しており、論理ホストへの発行はできません。このため、JP1/IM - Console を論理ホスト環境で運用している場合には、JP1/Script が発行する JP1 イベントは表示されません。表示するには、JP1/Base の物理ホストの転送設定ファイルに JP1/Script で発行する JP1 イベントを転送する設定を行ってください。

### 例

```
Dim se_Date ,s_Time ,e_Time ,rtn1
se_Date = Date()
s_Time = Time()
rtn1 = Exec ( _SCF_+"ABC.EXE" ,True )
e_Time = Time()

If rtn1 Then
    IMEventMessage ( "ABC.EXEは正常終了" , ,_USER_ ,_SCF_FIL_ , _
        End ,se_Date ,s_Time ,se_Date ,e_Time ,_EXEC_RTN_)
Else
    IMEventMessage ( "ABC.EXEはエラー終了" , ,_USER_ ,_SCF_FIL_ , _
        End ,se_Date ,s_Time ,se_Date ,e_Time ,_RTN_)
End
```

### 対象バージョン

JP1/Script 05-20 以降

## 8.7 メニュー呼び出しコマンド

### 8.7.1 Menu（ユーザ定義のメニューを表示する）

#### 機能

ユーザ定義のメニューを表示します。

#### 形式

Menu ( *FormName* , [ *SpnFilePath* ] [ , *FocusField* ] )

#### 指定項目

##### *FormName*

表示するメニューのフォーム名称を、文字列、または値を格納した変数名で指定します。

##### *SpnFilePath*

表示するメニューの情報が格納されているメニュー情報ファイルのフルパスを、文字列、または格納した変数名で指定します。

この値は省略できます。省略した場合、実行中のスクリプトファイルのメニュー情報ファイルを仮定します。

CallSpt コマンドから呼び出されたスクリプトの実行時は、呼び出されたスクリプトファイルのメニュー情報ファイルを仮定します。

##### *FocusField*

メニュー起動時にフォーカスを位置付けるフィールド名称を文字列、または格納した変数名で指定します。

この値は省略できます。省略した場合、メニュー情報ファイルで定義されたフィールドにフォーカスが位置付きます。

#### 説明

ユーザが定義したメニューフォームを表示します。コマンドが正常に実行された場合は真 ( True ) を、エラーが発生した場合は偽 ( False ) を、コマンドの実行結果として返します。

メニューフォーム終了時に、メニューフォームの終了要因を表す文字列が `_FORM_TERM_KEY_` 予約変数に格納されます。格納される値は次のどれかです。

値	意味
Enter	Enter キーが押されて終了しました。
Esc	Esc キーが押されて終了しました。
F1 ~ F12	ファンクションキー ( F1 ~ F12 ) が押されて終了しました。
Command	コマンドを実行して終了しました。
Close	Alt + F4 またはクローズボタンが押されて終了しました。
Timeout	終了監視時間を経過したため終了しました。

8. 基本コマンド

メニューの終了要因が Command である場合、実行したコマンド番号を `_FORM_TERM_CMDNO_` 予約変数に格納します。

また、終了キーのモデファイキーを表す文字列が `_FORM_MODIFY_KEY_` 予約変数に格納されます。格納される値は次のどれかです。

値	意味
Shift	Shift キー
Ctrl	Ctrl キー
Alt	Alt キー
Shift_Ctrl	Shift キーと Ctrl キー
Ctrl_Alt	Ctrl キーと Alt キー
Shift_Alt	Shift キーと Alt キー
Shift_Ctrl_Alt	Shift キーと Ctrl キーと Alt キー

また、メニューフォーム終了時にフォーカスの位置付いていたフィールド名を `_FORM_FIELD_NAME_` 予約変数に格納します。

補足

スクリプトをサービスで起動する場合は、このコマンドは使用できません。使用すると実行エラーになります。

Menu コマンドから実行された外部コマンドの終了コードに 0 以外が返された場合、メッセージボックスを表示します。メッセージボックスを表示しない場合は、次に示すレジストリキーの値に 0 を設定してください。

```
[ レジストリキー ]
HKEY_LOCAL_MACHINE\SOFTWARE\HITACHI\JP1/Script\SPTX

[ 値 ]
Menu_EmgMsgBox (DWORD)
```

```
[ 値のデータ ]
0 : メッセージボックスを表示しない
1 : メッセージボックスを表示する ( 初期値 )
```

例

```
！ メニューがEscキーを押されて終了した場合、スクリプトの実行を中断する。
Menu ( "MyMenu" )
If _FORM_TERM_KEY_ = "Esc" Then
  Exit
Else
  :
  :
  :
End
```

対象バージョン



JP1/Script 05-20 以降

## 8.8 演算処理コマンド

### 8.8.1 + 演算子（加算）（二つの式の和を求める）

機能

二つの数値の和を求めます。

形式

*Result* = *Expression1* + *Expression2*

指定項目

*Result*

二つの数値の和を受け取る変数名を指定します。

*Expression1*

任意の式を指定します。

*Expression2*

任意の式を指定します。

説明

式 *Expression1* と *Expression2* を加算し、その和を演算結果 *Result* として返します。

式の内部処理形式により、+ 演算子の動作は次のように異なります。

条件	行われる演算
両方の式が文字列	文字列連結
両方の式が数値	加算
両方の式が数値だけからなる文字列	加算
一方の式が数値,他方が文字列	文字列連結
一方の式が文字列,他方が数値だけからなる文字列	文字列連結
一方の式が数値,他方が数値だけからなる文字列	加算

両方の式が Empty 値のとき、演算結果 *Result* は数値 0 になります。ただし、一方の式だけが Empty 値のときは、他方の式がそのまま演算結果 *Result* として返されます。

補足

*Result* と *Expression1* が同じ変数である場合、+= 演算子を使用して書き換えることができます。

例

' 変数 *result1* には "ABCDEF" が格納される。  
*result1* = "ABC" + "DEF"

```
' 変数result2には12が格納される。
result2 = 7 + 5

' 変数result3には"10min."が格納される。
result3 = 10 + "min."
```

対象バージョン  
JP1/Script 01-00 以降

8.8.2 += 演算子（加算）（変数と式の和を変数に代入する）

機能  
変数の値と式の値を加算し、その和を変数に代入します。

形式  
*Result* += *Expression*

指定項目  
*Result*  
結果を受け取る変数名を指定します。  
*Expression*  
任意の式を指定します。

説明  
変数 *Result* の値と式 *Expression* の値を加算し、その和を演算結果として変数 *Result* に代入します。  
+= 演算子の動作は、+ 演算子の場合と同じで次のようになります。

条件	行われる演算
両方の値が文字列	文字列連結
両方の値が数値	加算
両方の値が数値だけからなる文字列	加算
一方の値が数値、他方が文字列	文字列連結
一方の値が文字列、他方が数値だけからなる文字列	文字列連結
一方の値が数値、他方が数値だけからなる文字列	加算

両方の値が Empty 値のときは、変数 *Result* には数値 0 が代入されます。変数 *Result* が Empty 値、または未定義のときは、式 *Expression* の値がそのまま変数 *Result* に代入されます。式 *Expression* の値が Empty 値のときは、変数 *Result* の値は変わりません。

例  

```
' 変数result1には"ABCDEF"が格納される。
result1 = "ABC"
```

## 8. 基本コマンド

```
result1 += "DEF"

' 変数result2には11が格納される。
result2 = 10
result2 += 1

' 変数result3には1が格納される。
Dim result3
result3 += 1
```

対象バージョン

JP1/Script 06-51 以降

### 8.8.3 - 演算子（減算・マイナス符号）（二つの数値の差を求める）

機能

二つの数値の差を求めます。または、数式の符号を反転した値を指定します。

形式 1

```
Result = Number1 - Number2
```

形式 2

```
-Number
```

指定項目

*Result*

二つの数値の差を受け取る変数名を指定します。

*Number*

任意の数式を指定します。

*Number1*

任意の数式を指定します。

*Number2*

任意の数式を指定します。

説明

形式 1 では、数式 *Number1* から *Number2* を減算し、その差を演算結果 *Result* として返します。

形式 2 では、*-* 演算子は単項マイナス符号演算子として、式の符号を反転した値を指定するために使用します。

Empty 値を持つ式は、0 として扱われます。

補足

*Result* と *Number1* が同じ変数である場合、*-=* 演算子を使用して書き換えることができます。

## 例

```
' 変数result1には2が格納される。
result1 = 7 - 5
```

## 対象バージョン

JP1/Script 01-00 以降

## 8.8.4 -= 演算子（減算）（変数と式の差を変数に代入する）

## 機能

変数の値から式の値を減算し、その差を変数に代入します。

## 形式

```
Result -= Number
```

## 指定項目

*Result*

結果を受け取る変数名を指定します。

*Number*

任意の数式を指定します。

## 説明

変数 *Result* の値から数式 *Number* の値を減算し、その差を演算結果として変数 *Result* に代入します。

両方の値が Empty 値のときは、変数 *Result* には数値 0 が代入されます。変数 *Result* が Empty 値、または未定義のときは、数式の符号を反転した値  $-Number$  が変数 *Result* に代入されます。数式 *Number* の値が Empty 値のときは、変数 *Result* の値は変わりません。

## 例

```
' 変数result1には-1が格納される。
Dim result1
result1 -= 1

' 変数result2には9が格納される。
result2 = 10
result2 -= 1
```

## 対象バージョン

JP1/Script 06-51 以降

## 8.8.5 Mod 演算子（剰余演算）（二つの数値の剰余を求める）

## 機能

二つの数値の除算を行い、その剰余を返します。

## 8. 基本コマンド

### 形式

*Result* = *Number1* Mod *Number2*

### 指定項目

*Result*

二つの数値の剰余を受け取る変数名を指定します。

*Number1*

任意の数式を指定します。

*Number2*

任意の数式を指定します。

### 説明

数式 *Number1* を *Number2* で除算し、その余りを演算結果 *Result* として返します。

Empty 値を持つ式は、0 として扱われます。

### 補足

*Result* と *Number1* が同じ変数である場合、*Mod=* 演算子を使用して書き換えることができます。

### 例

' 変数 *result1* には5が格納される。  
*result1* = 19 Mod 7

### 対象バージョン

JP1/Script 01-00 以降

## 8.8.6 Mod= 演算子（剰余演算）（変数と式の剰余を変数に代入する）

### 機能

変数の値を式の値で除算し、その剰余を変数に代入します。

### 形式

*Result* Mod= *Number*

### 指定項目

*Result*

結果を受け取る変数名を指定します。

*Number*

任意の数式を指定します。

### 説明

変数 *Result* の値を数式 *Number* の値で除算し、その余りを演算結果として変数 *Result* に代入します。

Empty 値は、0 として扱われます。変数 *Result* が Empty 値、または未定義のときは、0 が変数 *Result* に代入されます。数式 *Number* の値が Empty 値のときは、0 で除算を行うことになるため実行エラーになります。

例

```
' 変数result1には5が格納される。
result1 = 19
result1 Mod= 7
```

対象バージョン

JP1/Script 06-51 以降

### 8.8.7 \* 演算子（乗算）（二つの数値の積を求める）

機能

二つの数値の積を求めます。

形式

*Result* = *Number1* \* *Number2*

指定項目

*Result*

二つの数値の積を受け取る変数名を指定します。

*Number1*

任意の数式を指定します。

*Number2*

任意の数式を指定します。

説明

数式 *Number1* に *Number2* を乗算し、その積を演算結果 *Result* として返します。  
Empty 値を持つ式は、0 として扱われます。

補足

*Result* と *Number1* が同じ変数である場合、\*= 演算子を使用して書き換えることができます。

例

```
' 変数result1には35が格納される。
result1 = 7 * 5
```

対象バージョン

JP1/Script 01-00 以降

### 8.8.8 \*= 演算子（乗算）（変数と式の積を変数に代入する）

機能

## 8. 基本コマンド

変数の値に式の値を乗算し、その積を変数に代入します。

形式

*Result* \*= *Number*

指定項目

*Result*

結果を受け取る変数名を指定します。

*Number*

任意の数式を指定します。

説明

変数 *Result* の値に数式 *Number* の値を乗算し、その積を演算結果として変数 *Result* に代入します。

Empty 値は、0 として扱われます。変数 *Result* が Empty 値、または未定義のときや数式 *Number* の値が Empty 値のときは、0 が変数 *Result* に代入されます。

例

```
' 変数result1には35が格納される。
result1 = 7
result1 *= 5
```

対象バージョン

JP1/Script 06-51 以降

### 8.8.9 / 演算子（除算）（二つの数値の商を求める）

機能

二つの数値の商を計算し、結果を整数で返します（¥ 演算子と同じ機能です）。

形式

*Result* = *Number1* / *Number2*

指定項目

*Result*

二つの数値の商を受け取る変数名を指定します。

*Number1*

任意の数式を指定します。

*Number2*

任意の数式を指定します。

説明

数式 *Number1* を *Number2* で除算し、その商を演算結果 *Result* として整数で返します。

Empty 値を持つ式は、0 として扱われます。



## 補足

*Result* と *Number1* が同じ変数である場合、*/=* 演算子を使用して書き換えることができます。

## 例

```
' 変数result1と変数result2には2が格納される。
result1 = 14 / 7
result2 = 19 / 7
```

## 対象バージョン

JP1/Script 01-00 以降

## 8.8.10 */=* 演算子（除算）（変数と式の商を変数に代入する）

## 機能

変数の値を式の値で除算し、その商を整数で変数に代入します（*/=* 演算子と同じ機能です）。

## 形式

*Result* */=* *Number*

## 指定項目

*Result*

結果を受け取る変数名を指定します。

*Number*

任意の数式を指定します。

## 説明

変数 *Result* の値を数式 *Number* の値で除算し、その商を演算結果として整数で変数 *Result* に代入します。

Empty 値は、0 として扱われます。変数 *Result* が Empty 値、または未定義のときは、0 が変数 *Result* に代入されます。数式 *Number* の値が Empty 値のときは、0 で除算を行うことになるため実行エラーになります。

## 例

```
' 変数result1と変数result2には2が格納される。
Dim result1, result2
result1 = 14
result2 = 19
result1 /= 7
result2 /= 7
```

## 対象バージョン

JP1/Script 06-51 以降

### 8.8.11 ¥ 演算子（整数除算）（二つの数値の商を求める）

機能

二つの数値の商を計算し、結果を整数で返します（/ 演算子と同じ機能です）。

形式

*Result* = *Number1* ¥ *Number2*

指定項目

*Result*

二つの数値の商を受け取る変数名を指定します。

*Number1*

任意の数式を指定します。

*Number2*

任意の数式を指定します。

説明

数式 *Number1* を *Number2* で除算し、その商を演算結果 *Result* として整数で返します。

Empty 値を持つ式は、0 として扱われます。

補足

*Result* と *Number1* が同じ変数である場合、¥= 演算子を使用して書き換えることができます。

例

変数 *result1* と変数 *result2* には2が格納される。

```
result1 = 14 ¥ 7
```

```
result2 = 19 ¥ 7
```

対象バージョン

JP1/Script 01-00 以降

### 8.8.12 ¥= 演算子（整数除算）（変数と式の商を変数に代入する）

機能

変数の値を式の値で除算し、その商を整数で変数に代入します（/= 演算子と同じ機能です）。

形式

*Result* ¥= *Number*

指定項目

*Result*

結果を受け取る変数名を指定します。

*Number*

任意の数式を指定します。

#### 説明

変数 *Result* の値を数式 *Number* の値で除算し、その商を演算結果として整数で変数 *Result* に代入します。

Empty 値は、0 として扱われます。変数 *Result* が Empty 値、または未定義のときは、0 が変数 *Result* に代入されます。数式 *Number* の値が Empty 値のときは、0 で除算を行うことになるため実行エラーになります。

#### 例

```
' 変数result1と変数result2には2が格納される。
Dim result1, result2
result1 = 14
result2 = 19
result1 ¥= 7
result2 ¥= 7
```

#### 対象バージョン

JP1/Script 06-51 以降

## 8.8.13 ^ 演算子（べき乗）（二つの数値のべき乗を求める）

#### 機能

二つの数値のべき乗を計算し、結果を整数で返します。

#### 形式

*Result* = *Number1* ^ *Number2*

#### 指定項目

*Result*

二つの数値のべき乗を受け取る変数名を指定します。

*Number1*

任意の数式を指定します。

*Number2*

任意の数式を指定します。

#### 説明

数式 *Number1* と *Number2* のべき乗を計算し、その演算結果を *Result* に整数で返します。

Empty 値を持つ式は、0 として扱われます。

#### 補足

*Result* と *Number1* が同じ変数である場合、^= 演算子を使用して書き換えることが

## 8. 基本コマンド

できます。

例

```
' 変数result1には8が格納される。  
result1 = 2 ^ 3
```

対象バージョン

JP1/Script 06-00 以降

### 8.8.14 ^= 演算子（べき乗）（変数と式のべき乗を変数に代入する）

機能

変数の値と式の値のべき乗を計算し、結果を整数で変数に代入します。

形式

```
Result ^= Number
```

指定項目

*Result*

結果を受け取る変数名を指定します。

*Number*

任意の数式を指定します。

説明

変数 *Result* の値と数式 *Number* の値のべき乗を計算し、その演算結果を整数で変数 *Result* に代入します。

Empty 値は、0 として扱われます。変数 *Result* が Empty 値、または未定義のときは、0 が変数 *Result* に代入されます。数式 *Number* の値が Empty 値のときは、0 でべき乗を計算することになるため常に 1 になります。

例

```
' 変数result1には8が格納される。  
result1 = 2  
result1 ^= 3
```

対象バージョン

JP1/Script 06-51 以降

### 8.8.15 比較演算子（=, <>, <, <=, >, >=）（二つの式を比較する）

機能

二つの式の比較を行います。

形式

$$Result = Expression1 \text{ Comparisonoperator } Expression2$$

#### 指定項目

*Result*

結果を受け取る変数名を指定します。

*Expression1*

任意の式を指定します。

*Expression2*

任意の式を指定します。

*Comparisonoperator*

任意の比較演算子を指定します。

#### 説明

二つの式の比較を行います。

各比較演算子の演算結果 *Result* が、真 (True)、偽 (False) となる条件を、次に示します。

= (等しい)

真となる条件  $Expression1 = Expression2$

偽となる条件  $Expression1 \neq Expression2$

<> (等しくない)

真となる条件  $Expression1 \neq Expression2$

偽となる条件  $Expression1 = Expression2$

< (より小さい)

真となる条件  $Expression1 < Expression2$

偽となる条件  $Expression1 \geq Expression2$

<= (以下)

真となる条件  $Expression1 \leq Expression2$

偽となる条件  $Expression1 > Expression2$

> (より大きい)

真となる条件  $Expression1 > Expression2$

偽となる条件  $Expression1 \leq Expression2$

>= (以上)

真となる条件  $Expression1 \geq Expression2$

偽となる条件  $Expression1 < Expression2$

保持しているデータ型によってどのような比較が行われるか、または比較結果がどうなるかを次に示します。

条件	行われる演算
両方の式が数値データ型	数値比較

8. 基本コマンド

条件	行われる演算
両方の式が文字列型	文字列比較
一方の式が数値データ型、他方の式が文字列型	文字列比較
一方の式が Empty 値、他方の式が数値データ型	Empty 値を 0 とみなして数値比較を行います。 ただし、等しいかどうかを判定する場合は、0 と みなしません。
一方の式が Empty 値、他方の式が文字列型	Empty 値を長さ 0 の文字列 ("" ) とみなして文字 列比較を行います。
両方の式が Empty 値	二つの式は等しいという結果になります。

対象バージョン

JP1/Script 01-00 以降

8.8.16 And ( 論理積 ) ( 二つの式の論理積を求める )

機能

二つの式の論理積を求めます。

形式

*Result* = *Expression1* And *Expression2*

指定項目

*Result*

結果を受け取る変数名を指定します。

*Expression1*

真 ( True ) , または偽 ( False ) を評価する任意の式を指定します。

*Expression2*

真 ( True ) , または偽 ( False ) を評価する任意の式を指定します。

説明

二つの式の論理積を求めます。

演算結果 *Result* の値は次のようになります。

<i>Expression1</i>	<i>Expression2</i>	<i>Result</i>
True	True	True
True	False	False
False	True	False
False	False	False
文字列	True	True
文字列	False	False
文字列	文字列	True

<i>Expression1</i>	<i>Expression2</i>	<i>Result</i>
Empty 値	True	False
Empty 値	False	False
Empty 値	Empty 値	False

例

```

' 変数number1の値が1以上10以下のとき,実行ファイル"Day10.SPT"を
' 呼び出す。
number1 = Day ( )
If 1 <= number1 And number1 <= 10 Then
    Exec ( "Day10.SPT", True )
End If

```

対象バージョン

JP1/Script 01-00 以降

## 8.8.17 Or ( 論理和 ) ( 二つの式の論理和を求める )

機能

二つの式の論理和を求めます。

形式

*Result* = *Expression1* Or *Expression2*

指定項目

*Result*

結果を受け取る変数名を指定します。

*Expression1*

真 ( True ) , または偽 ( False ) を評価する任意の式を指定します。

*Expression2*

真 ( True ) , または偽 ( False ) を評価する任意の式を指定します。

説明

二つの式の論理和を求めます。

演算結果 *Result* の値は次のようになります。

<i>Expression1</i>	<i>Expression2</i>	<i>Result</i>
True	True	True
True	False	True
False	True	True
False	False	False
文字列	True	True

8. 基本コマンド

<i>Expression1</i>	<i>Expression2</i>	<i>Result</i>
文字列	False	True
文字列	文字列	True
Empty 値	True	True
Empty 値	False	False
Empty 値	Empty 値	False

例

```
' 変数number1の値が7以下,または20以上のとき,実行ファイル
' "Backup.BAT"を呼び出す。
number1 = Hour ( )
If number1 <= 7 Or 20 <= number1 Then
  Exec ( "Backup.BAT", True )
End If
```

対象バージョン

JP1/Script 01-00 以降

8.8.18 Not（論理否定）（式の論理否定を求める）

機能

式の論理否定を求めます。

形式

```
Result = Not Expression
```

指定項目

*Result*

結果を受け取る変数名を指定します。

*Expression*

真（True）, または偽（False）を評価する任意の式を指定します。

説明

式の論理否定を求めます。

演算結果 *Result* の値は次のようになります。

<i>Expression</i>	<i>Result</i>
True	False
False	True
文字列	False
Empty 値	True

例



```
' 変数number1の値が10以下のとき,実行ファイル"Day10.SPT"を呼び出す。  
number1 = Day ( )  
If Not 10 < number1 Then  
    Exec ( "Day10.SPT", True )  
End If
```

#### 対象バージョン

JP1/Script 06-00 以降

## 8.9 チェック処理コマンド

---

### 8.9.1 IsEmpty （変数が Empty 値かどうかを調べる）

#### 機能

変数が Empty 値かどうかを調べ、結果を真 (True) か偽 (False) で返します。

#### 形式

IsEmpty ( VarName )

#### 指定項目

*VarName*

調べる変数を変数名で指定します。

#### 説明

指定された変数が Empty 値かどうかを調べ、結果をコマンドの実行結果として返します。Empty 値の場合は真 (True) を、それ以外の場合は偽 (False) を返します。また、指定された変数が定義されていない場合は、常に偽 (False) を返します。

#### 例

```
' _ALLRIGHT_ 予約変数が Empty 値かどうかを調べる。  
If IsEmpty ( _ALLRIGHT_ ) Then  
  MessageBox ("管理者権限があるIDでログオンし直してください。")  
End If
```

#### 対象バージョン

JP1/Script 01-00 以降

### 8.9.2 IsDefine または IsDef （変数が定義されているかどうかを調べる）

#### 機能

変数が定義されているかどうかを調べ、結果を真 (True) か偽 (False) で返します。

#### 形式

IsDefine ( VarName )  
IsDef ( VarName )

#### 指定項目

*VarName*

調べる変数を変数名で指定します。

#### 説明

指定された変数が定義されているかどうかを調べ、結果をコマンドの実行結果とし

て返します。定義されている場合は真 (True) を、それ以外の場合は偽 (False) を返します。

例

```
' 位置変数%1が指定されたかどうかで処理を分岐する。
If IsDefine ( %1 ) Then
    Exec ( "CallProc01.SPT" ,True ,%1 )
Else
    Exec ( "CallProc02.SPT" ,True )
End If
```

対象バージョン

JP1/Script 01-00 以降

### 8.9.3 IsNumeric ( 数値として評価できるかを調べる )

機能

値が数値として評価できるかどうかを調べ、結果を真 (True) か偽 (False) で返します。

形式

```
IsNumeric ( CheckValue )
```

指定項目

*CheckValue*

調べる値、または値を格納した変数名を指定します。

説明

指定された値が数値として評価できるかどうかを調べ、結果をコマンドの実行結果として返します。数値として評価できる場合は真 (True) を、それ以外の場合は偽 (False) を返します。

対象バージョン

JP1/Script 01-01 以降

### 8.9.4 IsEmptyDir ( フォルダの中身が空かどうかを調べる )

機能

フォルダの中身が空かどうかを調べ、結果を真 (True) か偽 (False) で返します。

形式

```
IsEmptyDir ( DirName )
```

指定項目

*DirName*

チェックするフォルダ名を文字列、または値を格納した変数名で指定します。

## 8. 基本コマンド

### 説明

指定されたフォルダの中にファイル，またはフォルダが存在するかどうかをチェックし，結果をコマンドの実行結果として返します。中身が空の場合は真（True）を，それ以外の場合は偽（False）を返します。

また，指定されたフォルダが存在しない場合は，常に偽（False）を返します。

### 補足

Windows Vista で環境変数 ProgramFiles（通常はシステムドライブの Program Files フォルダ），または WinDir（通常はシステムドライブの Windows フォルダ）に設定されているフォルダ，またはフォルダ下のフォルダを指定する場合，注意が必要です。詳細は，「1.8.4 コマンドの動作」を参照してください。

### 例

```
' "BKUP" フォルダの中身が空かどうかを調べる。
outDir = _SCF_ + "BKUP¥"
If IsExistDir ( outDir ) = True Then
    If IsEmptyDir ( outDir ) = True Then
        MessageBox ( "Empty !" )
    Else
        MessageBox ( "Not Empty !" )
    End
Else
    MessageBox ( "Not Exist !" )
End
```

### 対象バージョン

JP1/Script 01-00 以降

## 8.9.5 IsExistDir（フォルダが存在するかどうかを調べる）

### 機能

フォルダが存在するかどうかを調べ，結果を真（True）か偽（False）で返します。

### 形式

```
IsExistDir ( DirName )
```

### 指定項目

*DirName*

チェックするフォルダ名を文字列，または値を格納した変数名で指定します。

### 説明

指定されたフォルダが存在するかどうかをチェックし，結果をコマンドの実行結果として返します。存在する場合は真（True）を，それ以外の場合は偽（False）を返します。

### 補足

Windows Vista で環境変数 ProgramFiles（通常はシステムドライブの Program

Files フォルダ), または WinDir (通常はシステムドライブの Windows フォルダ) に設定されているフォルダ, またはフォルダ下のフォルダを指定する場合, 注意が必要です。詳細は, 「1.8.4 コマンドの動作」を参照してください。

例

```
' 実行フォルダ下に "SCRIPT" フォルダが存在しない場合は作成する。
path1 = _BIN_+"SCRIPT"
If IsExistDir ( path1 ) = False Then
    MakeDir ( path1 )
End
```

対象バージョン

JP1/Script 01-00 以降

## 8.9.6 IsExistFile ( ファイルが存在するかどうかを調べる )

機能

ファイルが存在するかどうかをチェックし, 結果を真 (True) か偽 (False) で返します。

形式

```
IsExistFile ( FilePath )
```

指定項目

*FilePath*

チェックするファイルのフルパスを文字列, または値を格納した変数名で指定します。

説明

指定されたファイルが存在するかどうかをチェックし, 結果をコマンドの実行結果として返します。存在する場合は真 (True) を, それ以外の場合は偽 (False) を返します。

補足

Windows Vista で環境変数 ProgramFiles (通常はシステムドライブの Program Files フォルダ), または WinDir (通常はシステムドライブの Windows フォルダ) に設定されているフォルダ下のファイルを指定する場合, 注意が必要です。詳細は, 「1.8.4 コマンドの動作」を参照してください。

### ! 注意事項

フォルダ名を指定した場合, フォルダが存在するかどうかをチェックし, 結果を返します。存在する場合は真 (True) を返します。

例

```
path1 = _BIN_+"SCRIPT¥Logging.txt"
```

## 8. 基本コマンド

```
If IsExistFile ( path1 ) = True Then
  Copy ( path1 , _TEMP_+"ScpLog.txt" )
  If _COPY_RTN_ <> Skip Then
    DeleteFile ( path1 )
  End
End
```

対象バージョン

JP1/Script 01-00 以降

### 8.9.7 IsWriteableDir ( フォルダが書き込み可能かどうかを調べる )

機能

フォルダが書き込み可能かどうかをチェックし、結果を真 ( True ) か偽 ( False ) で返します。

形式

IsWriteableDir ( *DirName* )

指定項目

*DirName*

チェックするフォルダ名を文字列、または値を格納した変数名で指定します。

説明

指定されたフォルダが書き込み可能かどうかをチェックし、結果をコマンドの実行結果として返します。書き込み可能の場合は真 ( True ) を、それ以外の場合は偽 ( False ) を返します。

また、指定されたフォルダが存在しない場合は、常に偽 ( False ) を返します。

補足

Windows Vista で環境変数 ProgramFiles ( 通常はシステムドライブの Program Files フォルダ )、または WinDir ( 通常はシステムドライブの Windows フォルダ ) に設定されているフォルダ、またはフォルダ下のフォルダを指定する場合、注意が必要です。詳細は、「1.8.4 コマンドの動作」を参照してください。

例

```
' "BKUP"フォルダが書き込み可能かどうかを調べる。
outDir = _SCF_+"BKUP¥"
writeFlag = "NG"
If IsWriteableDir ( OutDir ) = True Then
  writeFlag ="OK"
End
```

対象バージョン

JP1/Script 01-00 以降

### 8.9.8 IsFileAttribute または IsFileAttr （フォルダ / ファイルの属性を調べる）

#### 機能

フォルダ，またはファイルの属性が指定された属性かどうかをチェックし，結果を真（True）か偽（False）で返します。

#### 形式

```
IsFileAttribute ( PathName , Attribute )
IsFileAttr ( PathName , Attribute )
```

#### 指定項目

##### *PathName*

チェックするフォルダ，またはファイルのフルパスを文字列，または値を格納した変数名で指定します。

##### *Attribute*

チェックする属性を指定します。  
この値は次のどれかになります。

値	意味
ATTR_READONLY	読み込み専用ファイル
ATTR_HIDDEN	隠しファイル
ATTR_ARCHIVE	アーカイブファイル
ATTR_SUBDIR	サブディレクトリ
ATTR_SYSTEM	システムファイル
ATTR_TEMPORARY	一時ファイル
ATTR_COMPRESSED （05-20以降）	圧縮ファイル

#### 説明

指定されたフォルダ，またはファイルが指定された属性かどうかをチェックし，結果をコマンドの実行結果として返します。指定された属性の場合は真（True）を，それ以外の場合は偽（False）を返します。

フォルダ，またはファイルの属性を設定するには，SetFileAttribute コマンドを，フォルダ，またはファイルの複数の属性を取得するには，GetFileAttribute コマンドを使用します。

#### 補足

Windows Vista で環境変数 ProgramFiles（通常はシステムドライブの Program Files フォルダ），または WinDir（通常はシステムドライブの Windows フォルダ）に設定されているフォルダ，フォルダ下のフォルダ，またはフォルダ下のファイルを指定する場合，注意が必要です。詳細は，「1.8.4 コマンドの動作」を参照してください。

例

```
' ファイルが一時ファイル属性ならば削除する。
Dim file1
file1 = "C:¥TEMP¥logging.tmp"
If IsFileAttribute ( file1 ,ATTR_TEMPORARY ) = True Then
  DeleteFile ( file1 )
End
```

対象バージョン

JP1/Script 01-00 以降

8.9.9 IsNew （バージョン情報 / ファイル日付の新旧を比較する）

機能

二つのファイルのバージョン情報，またはファイル日付の新旧を比較し，結果を真（True）か偽（False）で返します。

形式

```
IsNew ( PathName1 , PathName2 [, Option] )
```

指定項目

*PathName1*

比較する一方のファイルのフルパスを文字列，または値を格納した変数名で指定します。

*PathName2*

比較するもう一方のファイルのフルパスを文字列，または値を格納した変数名で指定します。

*Option*

オプションを指定します。  
この値には次の値を指定できます。

値	意味
Version	バージョン情報だけで比較します。
FileTime	ファイル日付だけで比較します。

説明

オプションが省略された場合，二つのファイルのバージョン情報を比較し，バージョン情報が取得できない場合はファイルの日付の新旧を比較し，*PathName1*で指定されたファイルが *PathName2*で指定されたファイルよりも新しい場合は真（True）を，古い，または等しい場合は偽（False）を，コマンドの実行結果として返します。  
オプションが指定されている場合は，各オプションの比較方法で比較します。  
エラーが発生した場合は，長さ 0 の文字列（""）を返します。



補足

Windows Vista で環境変数 ProgramFiles ( 通常はシステムドライブの Program Files フォルダ ), または WinDir ( 通常はシステムドライブの Windows フォルダ ) に設定されているフォルダ下のファイルを指定する場合 , 注意が必要です。詳細は , 「 1.8.4 コマンドの動作 」を参照してください。

例

```
Dim file1 ,file2
file1 = _SCF_+"TEST.SPT"
file2 = "C:¥BKUP¥TEST.SPT"
If IsNew ( file1 ,file2 ) = False Then
    ' 古いファイルに戻す
    Copy ( file2 ,file1 )
End
```

対象バージョン

JP1/Script 05-10 以降

8.9.10 CheckDirName ( フォルダ名の末尾が ( ¥ ) かどうかを調べる )

機能

フォルダ名の末尾が ( ¥ ) であるかどうかを調べます。

形式

```
CheckDirName ( DirNameBuff [ , Option ] )
```

指定項目

*DirNameBuff*

チェックするフォルダ名を格納している変数名を指定します。

*Option*

オプションを指定します。

この値には次の値を指定できます。

値	意味
Remove	末尾の ( ¥ ) を取り去ります。

説明

オプションが省略された場合 , 指定された変数に格納されているフォルダパスの末尾の文字を調べ , ( ¥ ) 以外の文字の場合は , 末尾に ( ¥ ) を付加します。オプションに Remove が指定されている場合は , 末尾の ( ¥ ) を取り去ります。コマンドが正常に実行された場合は真 ( True ) を , エラーが発生した場合は偽 ( False ) を , コマンドの実行結果として返します。

例

## 8. 基本コマンド

```
' フォルダパスが( ¥ )で終わることを保証してからフルパス名を作成
CheckDirName ( outDir )
CheckDirName ( inDir )
Copy ( inDir+"FILE.DAT" ,outDir+"FILE.DAT" ,Overwrite )
```

### 対象バージョン

JP1/Script 01-00 以降

## 8.9.11 CheckDriveType (ドライブの種類を調べる)

### 機能

ドライブの種類を調べます。

### 形式

```
CheckDriveType ( DrvTypeBuff , PathName )
```

### 指定項目

*DrvTypeBuff*

チェックした結果を受け取る変数名を指定します。

格納される値は次のどれかになります。

値	意味
REMOVABLE	交換可能なメディア (FD ドライブなど)
FIXED	ハードディスク
REMOTE	LAN/ リモートドライブ
CDROM	CD-ROM ドライブ
RAMDISK	RAM ディスク
""	判定できない

*PathName*

チェックするパス名を文字列、または値を格納した変数名で指定します。

### 説明

指定されたパスが指すドライブの種類をチェックします。コマンドが正常に実行された場合は真 (True) を、エラーが発生した場合は偽 (False) を、コマンドの実行結果として返します。

### 例

```
Dim result1
CheckDriveType ( result1 , "D:¥" )
If result1 <> CDROM Then
    MessageBox ( "CD-ROMではありません。" ,OK )
End
```

### 対象バージョン

JP1/Script 01-00 以降

## 8.10 外部プログラム呼び出しコマンド

### 8.10.1 Exec（自 PC 上の実行ファイルを読み出す）

#### 機能

複数パラメタ指定で実行ファイル（EXE ファイル，BAT ファイル，COM ファイル，SPT ファイル，CMD ファイル，関連付けられたファイル）を読み出します。

終了待ちの有無を指定できます。

関連付けられたファイルの読み出しについては，JP1/Script 05-00 以降でサポートしています。

#### 形式

```
Exec ( FileName , Flag [ , Param1 , Param2 , ... ] )
```

#### 指定項目

##### *FileName*

読み出す実行ファイル名を文字列，または値を格納した変数名で指定します。  
実行ファイルに指定できるファイルの種類は次のとおりです。

- 実行形式ファイル（.EXE）
- MS-DOS バッチファイル（.BAT）
- MS-DOS 実行形式ファイル（.COM）
- JP1/Script ファイル（.SPT）
- コマンドスクリプト（.CMD）
- 関連付けられたファイル

##### *Flag*

*FileName* で指定された実行ファイルの終了待ちをするときは真（True）を，  
終了待ちをしないときは偽（False）を指定します。

##### *Param1 ~ 31*

*FileName* で指定されたファイルの実行に必要なパラメタがある場合，パラメタを文字列，数値，または値を格納した変数名で指定します。

複数のパラメタを指定できます。

JP1/Script 06-00 以降では，複数のパラメタを一つずつ格納した一次元の配列変数を変数名で指定することもできます。

パラメタの記述規則については，「6.2 コマンドラインに関する規則」を参照してください。

なお，パラメタに次の文字列を指定した場合，起動するアプリケーション側でのウィンドウ表示時のデフォルトとして扱われます。明示的にウィンドウを表示するアプリケーションには有効になりません。

## 8. 基本コマンド

パラメタ	意味
"/SPT:HIDE"	アプリケーションのウィンドウを非表示にします。
"/SPT:MIN"	アプリケーションのウィンドウをアイコン化します。
"/SPT:MAX"	アプリケーションのウィンドウを最大表示します。

パラメタはコマンドラインパラメタとして実行ファイルに渡されますが、"/SPT:HIDE"、"/SPT:MIN"、および "/SPT:MAX" パラメタはコマンドラインパラメタとしては渡されません。

### 説明

指定された実行ファイルを実行します。

Flag に真 (True) が指定された場合は、呼び出したアプリケーションが終了するまで待機し、偽 (False) が指定された場合は、アプリケーションが終了するまで待機しないで次のコマンドを処理します。

コマンドが正常に実行された場合は真 (True) を、それ以外の場合は偽 (False) を、コマンドの実行結果として返します。

アプリケーションの終了待ちをし、コマンドの実行結果が真 (True) の場合は、実行ファイルの終了コードを \_EXEC\_RTN\_ 予約変数に符号付きの数値で格納します。終了待ちをしない場合やコマンドの実行結果が偽 (False) の場合は、\_EXEC\_RTN\_ 予約変数には何も格納されません。

また、アプリケーションの終了待ちをしない場合、実行ファイルの識別子を \_EXEC\_ID\_ 予約変数に格納します。

### 補足

- Exec コマンドは、呼び出す実行ファイルの種類が異なると、同等のエラーであってもコマンドの実行結果が異なる場合があります。例えば、存在しない EXE ファイルを呼び出した場合は偽 (False) を、存在しない SPT ファイルを呼び出した場合は真 (True) をコマンドの実行結果として返します。  
Flag に真 (True) が指定され、終了待ちをする Exec コマンドが正常終了したかどうかを判定するためには、コマンドの実行結果だけでなく、\_EXEC\_RTN\_ 予約変数に格納された実行ファイルの終了コードの判定も必要です。例 1 のような判定方法を推奨します。  
Flag に偽 (False) が指定され、スクリプト終了時にアプリケーションがまだ終了していない場合は、その時点でアプリケーションが終了するまで待機します。このような終了待ちをしないでスクリプトを終了するには、Exit コマンド (オプションに Skip 指定) を使用します。例 2 を参照してください。
- Windows Vista では、管理者権限が必要な実行ファイルを実行する場合、OS のユーザアカウント制御ダイアログが表示されることがあります。詳細は、「1.8.4 コマンドの動作」を参照してください。

### 例 1

```

' スクリプトファイル"ABC.SPT"を呼び出す。
rtn = Exec ( _SCF_+"ABC.SPT", True )
rtnCode = _RTN_

' 実行結果を判定する。
If rtn = True Then
    If _EXEC_RTN_ = 0 Then
        MsgBox ( "正常終了しました。" )
    Else
        MsgBox ( "異常終了しました。終了コード=" + _EXEC_RTN_)
        Exit ( _EXEC_RTN_)
    End
Else
    MsgBox ( "異常終了しました。エラーコード=" + rtnCode)
    Exit (rtnCode)
End

```

#### 例 2

```

' 実行ファイル"ABC.EXE"を終了待ちなしで呼び出す。
rtn1 = Exec ( "ABC.EXE", False )
:
' ABC.EXEの終了を待たずにスクリプトを終了する。
Exit ( 0, Skip )

```

#### 対象バージョン

JP1/Script 01-00 以降

## 8.10.2 NetExec ( 自 PC / 他 PC 上の実行ファイルを呼び出す )

#### 機能

複数パラメタ指定で実行ファイル ( EXE ファイル, BAT ファイル, COM ファイル, SPT ファイル, CMD ファイル, 関連付けられたファイル ) を呼び出します。ほかのコンピュータ上の実行ファイルも呼び出せます。終了待ちの有無を指定できます。

#### 形式

```

NetExec ( [CompName] , FileName , Flag , [ExecDirName] ,
[ExecPlace] , [Option] [ , Param1 , Param2 , ... ] )

```

#### 指定項目

##### *CompName*

呼び出すコンピュータ名を文字列, または値を格納した変数名で指定します。この値は省略できます。省略した場合, このコマンドを実行するコンピュータのコンピュータ名を仮定します。なお, 指定する他のコンピュータの OS は Windows NT, Windows 2000, Windows Server 2003, および Windows XP で, JP1/Script がインストールされていなければなりません。また, アクセスするためのアカウントについては「2.2 他のコンピュータとの通信時のアカウント」を参照してください。

### *FileName*

呼び出す実行ファイル名を文字列、または値を格納した変数名で指定します。  
実行ファイルに指定できるファイルの種類は次のとおりです。

- 実行形式ファイル (.EXE)
- MS-DOS バッチファイル (.BAT)
- MS-DOS 実行形式ファイル (.COM)
- JP1/Script スクリプトファイル (.SPT)
- コマンドスクリプト (.CMD)
- 関連付けられたファイル

空白を含むパス、またはファイル名を指定する場合に、パス、またはファイル名をダブルクォーテーション (") で囲む必要はありません。

### *Flag*

*FileName* で指定された実行ファイルの終了待ちをするときは真 (True) を、終了待ちをしないときは偽 (False) を指定します。

### *ExecDirName*

*FileName* で指定された実行ファイルのカレントディレクトリ名を文字列、または値を格納した変数名で指定します。

この値は省略できます。省略した場合、実行ファイルのディレクトリ名を仮定します。

なお、この値は *FileName* で指定された実行ファイルが JP1/Script ファイル以外の場合に有効となります。

### *ExecPlace*

*FileName* で指定された実行ファイルをサービス空間で実行するときには真 (True) を、ログオン空間で実行するときには偽 (False) を指定します。

この値は省略できます。省略した場合、真 (True) を仮定します。

なお、*CompName* で指定されたコンピュータでは、この値に真 (True) が指定された場合には JP1/Script サービスが、偽 (False) が指定された場合には Script ランチャが開始されていなければなりません。

### *Option*

*FileName* で指定された実行ファイルが JP1/Script スクリプトファイル (.SPT) の場合、オプションを指定します。

この値には次の値を指定できます。

値	意味
CopyFile	<p>呼び側に存在するスクリプトファイルを <i>CompName</i> で指定されたコンピュータにコピーして実行します。</p> <p>この場合、インストールフォルダの "DATA" フォルダ下に呼び側コンピュータ名のフォルダを作成し、その下にスクリプトファイルをコピーします。コピーするスクリプトファイルの実行環境ファイル (.SPV) はコピーしません。よってコピーしたスクリプトは実行環境ファイルのない状態で実行します。また、コピーしたスクリプトファイルと同名の実行環境ファイルが呼び側コンピュータ名のフォルダ下に存在する場合はコピー時に削除します。</p> <p>JP1/Script 06-00 以降では、コピーするスクリプトファイルと同名のメニュー情報ファイル (.SPN) が存在する場合メニュー情報ファイルも上記のフォルダ下にコピーします。同名のファイルが呼び側に存在する場合はコピー時に削除します。</p> <p>このコマンドの実行終了後、コピーしたスクリプトファイル、およびメニュー情報ファイルは削除し、コピーしたスクリプトの実行後に出力される解析トレースファイル、および実行トレースファイルは呼び側にコピーして戻します。</p> <p>なお、オプションに CopyFile を指定した場合、<i>Flag</i> で指定した値にかかわらずコピーしたスクリプトの実行が終了するまで Windows のログオフやシャットダウンはしないでください。</p>

この値は省略できます。省略した場合、*CompName* で指定されたコンピュータに存在するスクリプトファイルを実行します。

また、*FileName* で指定された実行ファイルが JP1/Script スクリプトファイル (.SPT) 以外の場合は無効になります。

#### Param1 ~ 31

*FileName* で指定されたファイルの実行に必要なパラメタがある場合、パラメタを文字列、数値、または値を格納した変数名で指定します。

複数のパラメタを指定できます。

JP1/Script 06-00 以降では、複数のパラメタを一つずつ格納した一次元の配列変数を変数名で指定することもできます。

パラメタの記述規則に関しては、「6.2 コマンドラインに関する規則」を参照してください。

なお、パラメタに次の文字列を指定した場合、起動するアプリケーション側でのウィンドウ表示時のデフォルトとして扱われます。明示的にウィンドウを表示するアプリケーションには有効になりません。

パラメタ	意味
"/SPT:HIDE"	アプリケーションのウィンドウを非表示にします。
"/SPT:MIN"	アプリケーションのウィンドウをアイコン化します。
"/SPT:MAX"	アプリケーションのウィンドウを最大表示します。

パラメタはコマンドラインパラメタとして実行ファイルに渡されますが、"/SPT:HIDE"、"/SPT:MIN"、および "/SPT:MAX" パラメタはコマンドラインパラメタとしては渡されません。

説明

## 8. 基本コマンド

指定されたコンピュータ上の実行ファイルを実行します。

*Flag* に真 (True) が指定された場合は、呼び出したアプリケーションが終了するまで待機し、偽 (False) が指定された場合は、アプリケーションが終了するまで待機しないで次のコマンドを処理します。

コマンドが正常に実行された場合は真 (True) を、それ以外の場合は偽 (False) を、コマンドの実行結果として返します。

アプリケーションの終了待ちをし、コマンドの実行結果が真 (True) の場合は、実行ファイルの終了コードを `_EXEC_RTN_` 予約変数に符号付きの数値で格納します。

終了待ちをしない場合やコマンドの実行結果が偽 (False) の場合は、

`_EXEC_RTN_` 予約変数には何も格納されません。

また、アプリケーションの終了待ちをしない場合、実行ファイルの識別子を

`_EXEC_ID_` 予約変数に格納します。

### 補足 1

*Flag* に偽 (False) が指定され、スクリプト終了時にアプリケーションがまだ終了していない場合は、その時点でアプリケーションが終了するまで待機します。このような終了待ちをしないでスクリプトを終了するには、Exit コマンド (オプションに Skip 指定) を使用します。Exec コマンドの例 2 を参照してください。

また、*CompName* にほかのコンピュータを指定する場合、LAN 環境下での動作だけを保証しています。ファイアウォールには対応していません。

JP1/Script 07-00 以降では、NetExec コマンド実行時にインストールフォルダの "LOG" フォルダ下 "STXNetExec\_Client"、および "STXNetExec\_Server" フォルダを作成し、エラーが発生した場合、このフォルダ下にエラーログファイルを出力します。

07-00 より前のバージョンでエラーログの出力先フォルダがインストールフォルダの "DATA¥LOG" であるバージョンに 07-00 を上書きインストールした場合、前バージョンでの出力先フォルダ "DATA¥LOG" 下にエラーログファイルを出力します。

エラーログファイルの出力を抑止する場合は、次に示すレジストリの値に 0 を設定してください。

#### 〔レジストリキー〕

HKEY\_LOCAL\_MACHINE¥SOFTWARE¥Hitachi¥JP1/Script¥SPTX¥  
Option

#### 〔値名〕

Net\_Trace

#### 〔値のデータタイプ〕

REG\_DWORD

#### 〔値〕

0 : エラーログファイルを出力しない

1 : エラーログファイルを出力する (初期値)



また、エラーログファイルの出力先フォルダを変更する場合は、次に示すレジストリの値にフォルダ名を指定してください。

〔値名〕

Net\_Trace\_Directory

〔値のデータタイプ〕

REG\_SZ

〔値〕

エラーログファイルの出力先フォルダ

## 補足 2

Windows Vista では、管理者権限が必要な実行ファイルを実行する場合、OS のユーザアカウント制御ダイアログが表示されることがあります。詳細は、「1.8.4 コマンドの動作」を参照してください。

## 例

```
' コンピュータ"SOP4A065 (SCRIPT)"上のスクリプトファイルを
' ログオン空間で起動する。
comp1="SOP4A065 (SCRIPT) "
prm1="/SPALV(2) "
prm2="/SPXLV(2) "
rtn1=NetExec ( comp1,_TEMP_+"NETWORK.SPT" ,True , ,False , _
,prm1 ,prm2 )
' スクリプトを終了する。
If rtn1 = True Then
' NetExecコマンドが正常終了の場合は呼び出したスクリプト
' ファイルの終了コードを返す。
Exit ( _EXEC_RTN_ )
Else
' NetExecコマンドが異常終了の場合はエラー詳細コードを返す。
Exit ( _RTN_ )
End
```

## 対象バージョン

JP1/Script 05-00 以降

## 8.10.3 WaitForExec ( 実行ファイルの終了待ち / 強制終了を行う )

### 機能

Exec, および NetExec コマンドの終了待ちなしで呼び出した実行ファイルの終了待ちや強制終了を行います。

### 形式

WaitForExec ( [Option] , [ExecId] [, Time] )

### 指定項目

Option

## 8. 基本コマンド

オプションを指定します。  
この値には次の値を指定できます。

値	意味
Wait	Exec, および NetExec コマンドで呼び出した実行ファイルが終了するまで待ちます。
WaitAll	Exec, および NetExec コマンドで呼び出したすべての実行ファイルが終了するまで待ちます。
Abort	Exec, および NetExec コマンドで呼び出した実行ファイルを強制終了します。
AbortAll	Exec, および NetExec コマンドで呼び出したすべての実行ファイルを強制終了します。

この値は省略できます。省略した場合、Wait を仮定します。

### *ExecId*

*Option* に Wait, または Abort が指定された場合、Exec, および NetExec コマンドの終了待ちなしで呼び出した実行ファイルの識別子を文字列、または値を格納した変数名で指定します。この識別子は、Exec, および NetExec コマンド実行時に `_EXEC_ID_` 予約変数に格納される値です。

この値は省略できます。省略した場合、現在の `_EXEC_ID_` 予約変数に格納されている値を仮定します。

### *Time ( 06:00 以降 )*

待ち時間を 0 からの数値、または値を格納した変数名で指定します。

*Option* に Wait, または WaitAll が指定された場合は、待ち時間に達するまで実行ファイルの終了待ちを行います。

*Option* に Abort, または AbortAll が指定された場合は、待ち時間に達したら強制終了を行います。

待ち時間は秒単位で指定します。必要がない場合は省略します。

なお、*Option* の指定にかかわらず、待ち時間に達して終了した場合は真 ( True ) を、待ち時間に達しても終了しない場合は偽 ( False ) をコマンドの実行結果として返し、`_RTN_` 予約変数には `_ERR_TIMEOUT_` 予約変数と等しい値が設定されます。しかし、On Error ステートメントで指定したラベルには分岐しません。

### 説明

Exec, および NetExec コマンドの終了待ちなしで呼び出した実行ファイルの終了待ちや強制終了を行い、コマンドが正常に実行された場合は真 ( True ) を、エラーが発生した場合は偽 ( False ) を、コマンドの実行結果として返します。

このコマンドを呼び出した時点で実行ファイルが終了していた場合は、常に真 ( True ) を返します。

### 例

```
Dim exec1_ID , exec2_ID
If Exec ( _WIN_ + "NOTEPAD.EXE" , False , _SCF_ + "ABC.SPT" ) Then
```

```

    exec1_ID = _EXEC_ID_
    :
End
If Exec ( _WIN_+"NOTEPAD.EXE" ,False ,_BIN_+"Memo.TXT" ) Then
    exec2_ID = _EXEC_ID_
    :
End

WaitForExec ( Wait ,exec1_ID ) ' 終了するまで待つ。
WaitForExec ( Abort ,exec2_ID ,60 )' 60秒後に強制終了する。

```

#### 対象バージョン

JP1/Script 01-01 以降

### 8.10.4 GetExecStatus （実行ファイルの実行状態を取得する）

#### 機能

Exec, および NetExec コマンドの終了待ちなしで呼び出した実行ファイルの現在の実行状態を取得します。

#### 形式

GetExecStatus ( [ *ExecId* ] )

#### 指定項目

##### *ExecId*

Exec, および NetExec コマンドの終了待ちなしで呼び出した実行ファイルの識別子を文字列, または値を格納した変数名で指定します。この識別子は, Exec, および NetExec コマンド実行時に \_EXEC\_ID\_ 予約変数に格納される値です。この値は省略できます。省略した場合, 現在の \_EXEC\_ID\_ 予約変数に格納されている値を仮定します。

#### 説明

Exec, および NetExec コマンドの終了待ちなしで呼び出した実行ファイルの現在の実行状態を取得します。コマンドが正常に実行された場合は実行状態を, エラーが発生した場合は長さ 0 の文字列 ("") を, コマンドの実行結果として返します。実行状態として返される値は次のどれかになります。

値	意味
EXEC_RUNNING	実行ファイルは実行中です。
EXEC_STOPPED	実行ファイルは実行中ではありません。

#### 例

```

' Execコマンドの終了待ちなしで呼び出した実行ファイルの処理が終了してから,
' 次のExecコマンドの処理を行う。
Dim exec1_ID
prm1 = _SCF_+"Result.TXT"

```

## 8. 基本コマンド

```
If Exec ( _SCF_+"Sum.SPT" ,False ,prm1 ) Then
    exec1_ID = _EXEC_ID_
    :
End
While GetExecStatus ( exec1_ID ) = EXEC_RUNNING
    Sleep ( 100 )
End While
Exec ( _WIN_+"NOTEPAD.EXE" ,True ,prm1)
```

対象バージョン

JP1/Script 06-00 以降

### 8.10.5 CallSpt ( 複数パラメタ指定で SPT ファイルを呼び出す )

機能

複数パラメタ指定で SPT ファイルを呼び出します。Exec コマンドと異なり、現在のプロセスと同一のプロセスで実行します。

形式

```
CallSpt ( SptFileName [,Param1 ,Param2 , ... ,Param31] )
```

指定項目

*SptFileName*

呼び出す JP1/Script スクリプトファイル ( .SPT ) 名を文字列、または値を格納した変数名で指定します。ファイル名の拡張子 ( .SPT ) を省略しても自動的に付加して動作します。

*Param1 ~ 31*

SptFileName で指定されたファイルの実行に必要なパラメタがある場合、パラメタを文字列、数値、または値を格納した変数名で指定します。

複数のパラメタを指定できます。

複数のパラメタを一つずつ格納した一次元の配列変数を変数名で指定することもできます。パラメタの記述規則については、「6.2.3 コマンドラインの記述規則」を参照してください。n 個のダブルクォーテーション ( " ) を文字としてそのままパラメタに含める場合は、( n \* 2 ) 個のダブルクォーテーションを指定し、パラメタ全体をダブルクォーテーションで囲んでください。

なお、Exec コマンドや NetExec コマンドで特別な意味を持つ " /SPT:HIDE ", " /SPT:MIN ", " /SPT:MAX " パラメタ、および、" /SPALV ( n ) ", " /SPXLV ( n ) ", " /NOEVLOG " パラメタを指定してもコマンドラインパラメタとして渡され、文字列として扱われます。特別な意味は持ちません。

説明

指定されたスクリプトファイルを実行します。

Exec コマンドと異なり、現在のプロセスと同一のプロセスで実行します。したがって、プロシージャのように現在のプロセスの環境を共有し、プロセス環境変数を更

新することができます。

CallSpt コマンドを実行するスクリプトファイル（以下「親スクリプト」という）と、*SptFileName* で指定されたスクリプトファイル（以下「子スクリプト」という）の間での変数やプロシージャの関係について次に示します。

- 子スクリプト内で Dim コマンドなどを使用して明示的に宣言された変数は、子スクリプトの中だけで有効なローカル変数になります。
- 子スクリプト内で明示的に宣言しないで使われている変数も、親スクリプトの CallSpt コマンドよりも前、または親スクリプトのさらに上のレベルの親スクリプトで定義されていないかぎりには、ローカル変数となります。親スクリプトの CallSpt コマンドよりも前、または親スクリプトのさらに上のレベルの親スクリプトで定義されている場合はグローバルな変数として扱われます。
- 親スクリプト内、または親スクリプトのさらに上のレベルの親スクリプトで定義されているプロシージャは子スクリプト内で使用することができます。
- 親スクリプトと子スクリプトで同じ名前のプロシージャが定義されている場合、子スクリプト内では子スクリプトで定義されたプロシージャが有効になります。ただし、親スクリプト内のプロシージャが Function プロシージャで子スクリプト内のプロシージャが Sub プロシージャである場合、親スクリプトの Function プロシージャが有効になります。
- 子スクリプト内で定義されたローカル変数やプロシージャ、子スクリプトの位置変数は、子スクリプトの実行が終了する（＝親スクリプトの CallSpt コマンドが終了する）と破棄されます。
- 子スクリプトを単体で実行する場合、親スクリプトの変数、プロシージャは使用できません。

子スクリプト中の #FileVersion や #Option の記述、および実行環境ファイル（.SPV）の内容はすべて無視し、親スクリプトの指定で実行します。

コマンドが正常に実行された場合は真（True）を、それ以外の場合は偽（False）を、コマンドの実行結果として返します。コマンドの実行結果が真（True）の場合は、実行ファイルの終了コードを \_EXEC\_RTN\_ 予約変数に符号付の数値で格納します。

#### 補足

ほかのコマンドと異なり CallSpt コマンドの文法チェックを行った場合は、*SptFileName* で指定されたスクリプトファイルの文法チェックも行います。

#### 例

```
' スクリプトファイル"Environment.SPT"を実行し、
' プロセス環境変数を更新・参照する。
Dim Path01
rtn = CallSpt(_SCF_+"Environment.SPT", "Get", "Path01", _
Path01)
:
rtn = CallSpt(_SCF_+"Environment.SPT", "Set", "Path01", _
_TEMP_)

' Environment.SPTの処理
```

## 8. 基本コマンド

```
Select Case %1
Case "Set"
    SetEnv ( ProcessEnv, %2, %3 )
Case "Get"
    %3 = GetEnv ( ProcessEnv, %2 )
End Select
```

対象バージョン

JP1/Script 06-71 以降

## 8.11 自動起動処理コマンド

### 8.11.1 EntryStartUp（スクリプトファイルを自動起動に登録する）

#### 機能

スクリプトファイルを自動起動に登録します。

#### 形式

```
EntryStartUp ( FilePath , [ MultiExec ] , [ ExecType ] , [ ExecTime ]
, [ ExecWeek ] , [ ExecDay ] [ , TerminateTime ] )
```

#### 指定項目

##### *FilePath*

登録するスクリプトファイルのフルパスを文字列、または値を格納した変数名で指定します。ファイル名の拡張子（.SPT）を省略しても自動的に付加して動作します。

##### *MultiExec*

スクリプトファイルの複数起動を許可するかどうかを指定します。複数起動を許可する場合は真（True）を、許可しない場合は偽（False）を指定します。この値は省略できます。省略した場合、実行環境ファイルで定義された値を仮定します。

##### *ExecType*

起動のタイプを指定します。  
この値は次のどれかになります。

値	意味
Logon	ログオン時に起動
Service	JP1/Script のサービス開始時に起動

この値は省略できます。省略した場合、実行環境ファイルで定義された値を仮定します。

##### *ExecTime*

起動する時刻を設定する場合、時刻を hh:mm:ss の形式で指定します。ss には必ず 00 を指定してください。それ以外の値を指定しても 00 とおきかえて動作します。

起動する時刻の設定をしない場合は、省略します。

##### *ExecWeek*

起動する曜日を設定する場合、起動する曜日を表す値を数値、または値を格納した変数名で指定します。

## 8. 基本コマンド

この値は次のどれかになります。

値	意味
1	日曜
2	月曜
3	火曜
4	水曜
5	木曜
6	金曜
7	土曜

起動する曜日を設定しない場合は、省略します。

### *ExecDay*

起動する日付を設定する場合、日付を数値、または値を格納した変数名で指定します。指定できる値の範囲は 1 から 31 までです。

起動する日付を設定しない場合は、省略します。

*ExecWeek* に値を指定した場合は、無効になります。

### *TerminateTime*

打ち切り時間を数値、または値を格納した変数名で指定します。時間は分単位で指定します。指定できる値の範囲は、0 から 1,440 までです。

この値は省略できます。省略した場合、実行環境ファイルで定義された値を仮定します。

## 説明

スクリプトファイルを自動起動に登録します。コマンドが正常に実行された場合は真 (True) を、エラーが発生した場合は偽 (False) をコマンドの実行結果として返します。

このコマンドは *FilePath* で指定されたスクリプトファイルの実行環境ファイル (.SPV) の内容を更新します。実行環境ファイルがない場合は、新しく実行環境ファイルを作成します。

## ! 注意事項

Windows XP の場合、異なるユーザで同じスクリプトに対してログオン空間の自動起動情報を登録した場合、前に登録したユーザの自動起動の条件が無効になり、あとのユーザで登録したときの条件になります。無条件に更新されるので十分注意してください。

## 例

```
' D:¥Work¥Script 下にあるスクリプトファイルで、実行環境ファイルが存在する  
' ものを自動起動に登録する。  
wkPath = "D:¥Work¥Script¥"  
For FileName = wkPath + "*" + _SCF_EXT_ Do
```



```

FileName = wkPath + FileName
FilePath = Left ( FileName ,Len ( FileName ) - 4 ) + _SVF_EXT_
If IsExistFile ( FilePath ) = True Then
    EntryStartUp ( FileName )
End
End For

```

対象バージョン

JP1/Script 05-10 以降

## 8.11.2 CancelStartUp ( スクリプトファイルの自動起動の登録を解除する )

機能

スクリプトファイルの自動起動の登録を解除します。

形式

```
CancelStartUp ( [ FilePath ] )
```

指定項目

*FilePath*

登録を解除するスクリプトファイルのフルパスを文字列、または値を格納した変数名で指定します。ファイル名の拡張子 ( .SPT ) を省略しても自動的に付加して動作します。

この値は省略できます。省略した場合、自動起動に登録されているすべてのスクリプトファイルを解除します。

説明

指定されたスクリプトファイルの自動起動の登録を解除します。コマンドが正常に実行された場合は真 ( True ) を、エラーが発生した場合は偽 ( False ) を、コマンドの実行結果として返します。

また、指定されたスクリプトファイルが存在しない場合や指定されたスクリプトファイルの実行環境ファイルが自動起動に登録されていない場合は、常に真 ( True ) を返します。

例

```

' このスクリプトが起動されてから1時間後に、自動起動に登録されている
' すべてのスクリプトファイルを解除する。
Sleep ( 60 * 60 * 1000 )
CancelStartUp ( )

```

対象バージョン

JP1/Script 05-10 以降

## 8.12 JP1 呼び出しコマンド

---

### 8.12.1 JOBSUBMIT ( キューサーバにスクリプトを登録する )

機能

JP1/AJS2, または JP1/NetBatch のキューサーバに対してスクリプトを登録します。

形式

```
JOBSUBMIT ( [ DomainName ] , QueSrvName , QueName , UserName ,  
[ JobName ] , Priority , FilePath , [ CommandLine ] , JobNoBuff  
[ , Flag ] )
```

指定項目

*DomainName*

*QueSrvName* で指定するキューサーバが参加しているドメイン名, またはワークグループ名を, 文字列, または値を格納した変数名で指定します。指定できる文字数は, ドメイン名の最大文字数です。スタンドアロンで運用している場合は省略できます。省略する場合は, 長さ 0 の文字列 ("" ) を指定してください。

*QueSrvName*

*QueName* で指定するキューが作成されているサーバ名を文字列, または値を格納した変数名で指定します。指定できる文字数は, コンピュータ名の最大文字数です。

*QueName*

ジョブを投入するキュー名称を文字列, または値を格納した変数名で指定します。指定できる文字数は, *DomainName* と *QueSrvName* と *QueName* を合わせて半角文字で 65 文字以内 ( 全角文字で 32 文字以内 ) です。

*UserName*

ジョブを投入するユーザの名称を文字列, または値を格納した変数名で指定します。指定できる文字数は, 半角文字で 20 文字以内 ( 全角文字で 10 文字以内 ) です。

*JobName*

投入するジョブに付ける名称を文字列, または値を格納した変数名で指定します。指定できる文字数は, 半角文字で 32 文字以内 ( 全角文字で 16 文字以内 ) です。

この値を省略する場合は, 長さ 0 の文字列 ("" ) を指定してください。このように指定すると *FilePath* で指定したファイル名の先頭 32 文字がジョブ名として扱われます。

*Priority*

投入するジョブの実行優先順位を指定します。  
この値は次のどれかになります。

値	意味
0	対話処理と比較して低いことを表します。
1	対話処理と同等であることを表します。
2	対話処理と比較して高いことを表します。

*FilePath*

ジョブとして実行するスクリプトファイル名のフルパスを文字列、または値を格納した変数名で指定します。指定できる文字数は、半角文字で 259 文字以内（全角文字で 129 文字以内）です。

スペースを含むフルパスを指定しないでください。指定した場合は、正しく動作しません。

*CommandLine*

*FilePath* で指定したファイルに対するパラメタを文字列、または値を格納した変数名で指定します。サブミットするジョブにパラメタが不要な場合は、省略できます。省略する場合は、長さ 0 の文字列（""）を指定してください。

*JobNoBuff*

投入したジョブに割り当てられたジョブ番号を受け取る変数名を指定します。  
このジョブ番号は、JOBWait コマンド、JOBHold コマンド、JOBCancel コマンドで使用します。

*Flag (01-01 以降)*

内部的に JP1/AJS2、または JP1/Net Batch の関数 NetBatchQueueOpen、および NetBatchQueueClose を呼び出して指定されたキューの開閉をする場合は真（True）を、呼び出さないでキューの開閉をしない場合は偽（False）を指定します。

この値は省略できます。省略した場合、偽（False）を仮定します。

同じキューに対して、ほかのアプリケーションと並行してこのコマンドを実行する場合、偽（False）を指定することを推奨します。

## 説明

指定されたキューに対して指定されたスクリプトファイルを登録します。

このコマンドは、内部的には JP1/AJS2、または JP1/NetBatch の関数

NetBatchQueueOpen、NetBatchJobSubmitWithID、NetBatchJobGetWithID、NetBatchQueueClose が実行されています。

内部的に呼び出す関数が成功した場合は真（True）を、それ以外の場合は偽（False）を、コマンドの実行結果として返します。コマンドの実行結果が真（True）の場合は、関数の終了コードを \_JOB\_RTN\_ 予約変数に格納します。コマンドの実行結果が偽（False）の場合は、\_JOB\_RTN\_ 予約変数には何も格納されません。

## 8. 基本コマンド

`_JOB_RTN_` 予約変数に格納される値は次のどれかになります。

値	意味
0	正常終了。
1	パラメタに誤りがある。
2	メモリが不足していたり、TCP/IP が正しく設定されていないなど実行環境に不備がある。
3	ネットワークに障害が生じた。
4	次のどれかの状態を示す。 <ul style="list-style-type: none"><li>指定したキューが存在しない。</li><li>キュー名称に誤りがある。</li><li>パッチジョブ運用が停止処理中、または停止状態である。</li><li>JP1/NetBatch サービスが初期化中である。</li><li>キューが閉じられている。</li></ul>
10	JP1/NetBatch の関数 <code>NetBatchQueueOpen</code> 、または <code>NetBatchQueueClose</code> を使用するためのアクセス権がない。 またはキューに投入できるジョブ数の最大値を超えたことによって、キューが強制的に閉じられたため、関数では操作できない。

なお、この値は JP1/NetBatch 02-00 の時点での関数の終了コードです。それ以降の JP1/NetBatch、または JP1/AJS2 での関数の終了コードに関しては、JP1/NetBatch、または JP1/AJS2 を参照してください。

### 例

```
Dim rtn1 , jobNo
rtn1 = JOBSUBMIT ( _DOMAIN_ , _COMP_ , "Que1" , _USER_ , _
                  " " , 0 , _BIN_ + "SCRIPT.SPT" , " " , jobNo )
If rtn1 = True And _JOB_RTN_ <> 0 Then
    MessageBox ( "JOBSUBMITで失敗しました！" , OK , , Ex )
End
```

### 対象バージョン

JP1/Script 01-00 以降

## 8.12.2 JOBWait (ジョブの終了を待つ)

### 機能

JOBSUBMIT コマンドで登録したジョブが終了するのを待ちます。

### 形式

```
JOBWait ( [DomainName] , QueSrvName , QueName , UserName , JobNo
, [JobNameBuff] , [ExitCodeBuff] , [JobSubmitTimeBuff] ,
[JobStartTimeBuff] , [JobEndTimeBuff] , [FilePathBuff] ,
[CommandLineBuff] [ , Time] )
```

### 指定項目

*DomainName*

*QueSrvName* で指定するキューサーバが参加しているドメイン名、またはワー

クグループ名を文字列、または値を格納した変数名で指定します。指定できる文字数は、ドメイン名の最大文字数です。スタンドアロンで運用している場合は省略できます。省略する場合は、長さ 0 の文字列 ("" ) を指定してください。

#### *QueSrvName*

*QueName* で指定するキューが作成されているサーバ名を文字列、または値を格納した変数名で指定します。指定できる文字数は、コンピュータ名の最大文字数です。

#### *QueName*

ジョブを投入したキュー名称を文字列、または値を格納した変数名で指定します。指定できる文字数は、*DomainName* と *QueSrvName* と *QueName* を合わせて半角文字で 65 文字以内 ( 全角文字で 32 文字以内 ) です。

#### *UserName*

ジョブを投入したユーザの名称を文字列、または値を格納した変数名で指定します。指定できる文字数は、半角文字で 20 文字以内 ( 全角文字で 10 文字以内 ) です。

#### *JobNo*

実行を保留、または保留解除するジョブのジョブ番号を数値、または値を格納した変数名で指定します。このジョブ番号は、JOBSUBMIT コマンドで取得した番号です。

#### *JobNameBuff*

ジョブ名称を受け取る変数名を指定します。  
この値は省略できます。

#### *ExitCodeBuff*

*JobNo* で指定したジョブの終了コードを受け取る変数名を指定します。  
この値は省略できます。

#### *JobSubmitTimeBuff*

*JobNo* で指定したジョブの投入時刻を受け取る変数名を指定します。変数には yyyy/mm/dd/w/HH:MM:SS:sss の形式 ( yyyy: 西暦 , mm: 月 , dd: 日 , w: 曜日 ( 0: 日曜 ~ 6: 土曜 ) , HH: 時 , MM: 分 , SS: 秒 , sss: ミリ秒 ) で格納されます。  
この値は省略できます。

#### *JobStartTimeBuff*

*JobNo* で指定したジョブの実行が開始された時刻を受け取る変数名を指定します。変数には yyyy/mm/dd/w/HH:MM:SS:sss の形式 ( yyyy: 西暦 , mm: 月 , dd: 日 , w: 曜日 ( 0: 日曜 ~ 6: 土曜 ) , HH: 時 , MM: 分 , SS: 秒 , sss: ミリ秒 ) で格納されます。  
この値は省略できます。

#### *JobEndTimeBuff*

*JobNo* で指定したジョブの実行が終了した時刻を受け取る変数名を指定します。変数には yyyy/mm/dd/w/HH:MM:SS:sss の形式 ( yyyy: 西暦 , mm: 月 , dd: 日 ,

8. 基本コマンド

w: 曜日 ( 0: 日曜 ~ 6: 土曜 ), HH: 時 , MM: 分 , SS: 秒 , sss: ミリ秒 ) で格納されます。  
この値は省略できます。

*FilePathBuff*

*JOBSubmit* コマンドで指定したジョブとして実行するスクリプトファイル名のフルパスを受け取る変数名を指定します。  
この値は省略できます。

*CommandLineBuff*

*JOBSubmit* コマンドで指定したジョブとして実行するファイルに対するパラメタを受け取る変数名を指定します。  
この値は省略できます。

*Time*

ジョブの終了を監視する間隔を数値 , または値を格納した変数名で指定します。  
間隔はミリ秒単位で指定します。  
この値は省略できます。省略した場合 , 1,000 ミリ秒を仮定します。

説明

*JOBSubmit* コマンドで登録したジョブの実行が終了するのを待ちます。  
このコマンドは , 内部的には JP1/AJS2 , または JP1/NetBatch の関数 *NetBatchJobGetWithID* が実行されています。  
内部的に呼び出す関数が成功した場合は真 ( True ) を , それ以外の場合は偽 ( False ) を , コマンドの実行結果として返します。コマンドの実行結果が真 ( True ) の場合は , 関数の終了コードを *\_JOB\_RTN\_* 予約変数に格納します。コマンドの実行結果が偽 ( False ) の場合は , *\_JOB\_RTN\_* 予約変数には何も格納されません。  
*\_JOB\_RTN\_* 予約変数に格納される値は次のどれかになります。

値	意味
0	正常終了。
1	パラメタに誤りがある。
2	メモリが不足していたり , TCP/IP が正しく設定されていないなど実行環境に不備がある。
3	ネットワークに障害が生じた。
4	次のどれかの状態を示す。 <ul style="list-style-type: none"><li>指定したキューが存在しない。</li><li>キュー名称に誤りがある。</li><li>バッチジョブ運用が停止状態である。</li><li>JP1/AJS2 , または JP1/Net Batch サービスが初期化中である。</li></ul>
5	ジョブ番号で指定したジョブが存在しない。

なお , この値は JP1/NetBatch 02-00 の時点での関数の終了コードです。それ以降の JP1/NetBatch , または JP1/AJS2 での関数の終了コードに関しては , JP1/NetBatch , または JP1/AJS2 を参照してください。

例

```
Dim rtn1 ,jobNo
rtn1 = JOBSUBMIT ( _DOMAIN_ , _COMP_ , "Que1" , _USER_ , _
    " " , 0 , _BIN_ + "SCRIPT.SPT" , " " , jobNo )
If rtn1 = True And _JOB_RTN_ = 0 Then
    rtn1 = JOBWAIT ( _DOMAIN_ , _COMP_ , "Que1" , _USER_ , jobNo )
    If rtn1 = True And _JOB_RTN_ = 0 Then
        MessageBox ( "ジョブの実行は終了しました。" , OK , , Information )
    End
End
```

対象バージョン

JP1/Script 01-00 以降

### 8.12.3 JOBHold （ジョブを保留 / 解除する）

機能

JOBSUBMIT コマンドで登録したジョブを保留・保留解除します。

形式

```
JOBHold ( [ DomainName ] , QueSrvName , QueName , UserName , JobNo
    , Flag )
```

指定項目

*DomainName*

*QueSrvName* で指定するキューサーバが参加しているドメイン名、またはワークグループ名を文字列、または値を格納した変数名で指定します。指定できる文字数は、ドメイン名の最大文字数です。スタンドアロンで運用している場合は省略できます。省略する場合は、長さ 0 の文字列 ( "" ) を指定してください。

*QueSrvName*

*QueName* で指定するキューが作成されているサーバ名を文字列、または値を格納した変数名で指定します。指定できる文字数は、コンピュータ名の最大文字数です。

*QueName*

ジョブを投入したキュー名称を文字列、または値を格納した変数名で指定します。

指定できる文字数は、*DomainName* と *QueSrvName* と *QueName* を合わせて半角文字で 65 文字以内（全角文字で 32 文字以内）です。

*UserName*

ジョブを投入したユーザの名称を文字列、または値を格納した変数名で指定します。

指定できる文字数は、半角文字で 20 文字以内（全角文字で 10 文字以内）です。

*JobNo*

実行を保留、または保留解除するジョブのジョブ番号を数値、または値を格納

## 8. 基本コマンド

した変数名で指定します。このジョブ番号は、JOBSubmit コマンドで取得した番号です。

### Flag

ジョブの実行を保留するときには真 (True) を、ジョブの保留を解除するときには偽 (False) を指定します。

### 説明

JOBSubmit コマンドで登録したジョブの実行を保留、または保留解除します。ジョブが実行待ち状態の場合にだけ、実行を保留できます。またジョブが保留状態の場合にだけ、保留を解除できます。

このコマンドは、内部的には JP1/AJS2、または JP1/NetBatch の関数 NetBatchJobHoldWithID が実行されています。

内部的に呼び出す関数が成功した場合は真 (True) を、それ以外の場合は偽 (False) を、コマンドの実行結果として返します。コマンドの実行結果が真 (True) の場合は、関数の終了コードを \_JOB\_RTN\_ 予約変数に格納します。コマンドの実行結果が偽 (False) の場合は、\_JOB\_RTN\_ 予約変数には何も格納されません。\_JOB\_RTN\_ 予約変数に格納される値は次のどれかになります。

値	意味
0	正常終了。
1	パラメタに誤りがある。
2	メモリが不足していたり、TCP/IP が正しく設定されていないなど実行環境に不備がある。
3	ネットワークに障害が生じた。
4	次のどれかの状態を示す。 <ul style="list-style-type: none"><li>指定したキューが存在しない。</li><li>キュー名称に誤りがある。</li><li>バッチジョブ運用が停止状態である。</li><li>JP1/AJS2、または JP1/Net Batch サービスが初期化中である。</li></ul>
5	ジョブ番号で指定したジョブが存在しない。
6	ジョブ番号で指定したジョブは、現在実行中である。

なお、この値は JP1/NetBatch 02-00 の時点での関数の終了コードです。それ以降の JP1/NetBatch、または JP1/AJS2 での関数の終了コードに関しては、JP1/NetBatch、または JP1/AJS2 を参照してください。

### 例

```
Dim rtn1 , jobNo
rtn1 = JOBSubmit ( _DOMAIN_ , _COMP_ , "Que1" , _USER_ , _
    " " , 0 , _BIN_ + "SCRIPT.SPT" , " " , jobNo )
If rtn1 = True And _JOB_RTN_ = 0 Then
    rtn1 = JOBHold ( _DOMAIN_ , _COMP_ , "Que1" , _USER_ , jobNo , True )
    If rtn1 = True And _JOB_RTN_ = 0 Then
        MessageBox ( "ジョブの実行を保留しています。" , OK , , Information )
    End
```



End

対象バージョン

JP1/Script 01-00 以降

## 8.12.4 JOBCancel（ジョブをキャンセルする）

機能

JOBSsubmit コマンドで登録したジョブをキャンセルします。

形式

JOBCancel ( [ *DomainName* ] , *QueSrvName* , *QueName* , *UserName* ,  
*JobNo* )

指定項目

*DomainName*

*QueSrvName* で指定するキューサーバが参加しているドメイン名、またはワークグループ名を文字列、または値を格納した変数名で指定します。指定できる文字数は、ドメイン名の最大文字数です。スタンドアロンで運用している場合は省略できます。省略する場合は、長さ 0 の文字列 ( "" ) を指定してください。

*QueSrvName*

*QueName* で指定するキューが作成されているサーバ名を文字列、または値を格納した変数名で指定します。指定できる文字数は、コンピュータ名の最大文字数です。

*QueName*

ジョブを投入したキュー名称を文字列、または値を格納した変数名で指定します。指定できる文字数は、*DomainName* と *QueSrvName* と *QueName* を合わせて半角文字で 65 文字以内（全角文字で 32 文字以内）です。

*UserName*

ジョブを投入したユーザの名称を文字列、または値を格納した変数名で指定します。指定できる文字数は、半角文字で 20 文字以内（全角文字で 10 文字以内）です。

*JobNo*

実行をキャンセルするジョブのジョブ番号を数値、または値を格納した変数名で指定します。このジョブ番号は、JOBSsubmit コマンドで取得した番号です。

説明

JOBSsubmit コマンドで登録したジョブの実行をキャンセルします。ジョブが実行待ち状態の場合だけでなく実行中のジョブもキャンセルできます。

このコマンドは、内部的には JP1/AJS2、または JP1/NetBatch の関数 NetBatchJobCancelWithID が実行されています。

内部的に呼び出す関数が成功した場合は真 ( True ) を、それ以外の場合は偽

## 8. 基本コマンド

( False ) を、コマンドの実行結果として返します。コマンドの実行結果が真 ( True ) の場合は、関数の終了コードを \_JOB\_RTN\_ 予約変数に格納します。コマンドの実行結果が偽 ( False ) の場合は、\_JOB\_RTN\_ 予約変数には何も格納されません。  
\_JOB\_RTN\_ 予約変数に格納される値は次のどれかになります。

値	意味
0	正常終了。
1	パラメタに誤りがある。
2	メモリが不足していたり、TCP/IP が正しく設定されていないなど実行環境に不備がある。
3	ネットワークに障害が生じた。
4	次のどれかの状態を示す。 <ul style="list-style-type: none"><li>・指定したキューが存在しない。</li><li>・キュー名称に誤りがある。</li><li>・バッチジョブ運用が停止状態である。</li><li>・JP1/AJS2、または JP1/Net Batch サービスが初期化中である。</li></ul>
5	ジョブ番号で指定したジョブが存在しない。

なお、この値は JP1/NetBatch 02-00 の時点での関数の終了コードです。それ以降の JP1/NetBatch、または JP1/AJS2 での関数の終了コードに関しては、JP1/NetBatch、または JP1/AJS2 を参照してください。

### 例

```
Dim rtn1 , jobNo
rtn1 = JOBSUBMIT ( _DOMAIN_ , _COMP_ , "Que1" , _USER_ , _
                  " " , 0 , _BIN_ + "SCRIPT.SPT" , " " , jobNo )
If rtn1 = True And _JOB_RTN_ = 0 Then
    rtn1 = JOBCANCEL ( _DOMAIN_ , _COMP_ , "Que1" , _USER_ , jobNo )
    If rtn1 = True And _JOB_RTN_ = 0 Then
        MsgBox ( "ジョブの実行をキャンセルしました。 " , OK , , Ex )
    End
End
```

### 対象バージョン

JP1/Script 01-00 以降

## 8.13 コメントコマンド

---

### 8.13.1 Rem または ' ( プログラム内にコメントを記述する )

#### 機能

プログラム内にコメントを記述するときに指定します。

#### 形式

```
Rem Comment
' Comment
```

#### 指定項目

*Comment*

プログラムをわかりやすくするための説明文 ( コメント ) を記述します。Rem と *Comment* の間には、半角文字で 1 文字以上のスペースを入力する必要があります。

#### 説明

Rem の代わりにクォーテーション ( ' ) を使うことができます。同じ行のほかのステートメント、またはコマンドの後にクォーテーション ( ' ) や Rem を記述できません。

#### 例

```
Rem ファイルにメッセージを出力する。
Message ( Target_File , _SCF_+"履歴.TXT" , "実行開始" )

' ウィンドウにメッセージを出力する。
Message ( Target_DispOn , "履歴" , "実行開始" )
```

#### 対象バージョン

JP1/Script 01-00 以降

# 8.14 その他のコマンド

## 8.14.1 Sleep （スクリプトの実行を中断する）

機能

指定された時間，スクリプトの実行を中断します。

形式

Sleep ( *Time* )

指定項目

*Time*

停止する時間を数値，または値を格納した変数名で指定します。  
時間はミリ秒単位で指定します。

説明

指定された時間だけスクリプトの実行が中断されます。

例

```
If debug = True Then
  Sleep ( 3000 )
End
```

対象バージョン

JP1/Script 01-00 以降

## 8.14.2 Alert （ユーザエラーを警告 / 解除する）

機能

トレースビューアのステータスにユーザエラーを警告・警告解除します。

形式

Alert ( [ *Mode* ] )

指定項目

*Mode*

トレースビューアのステータスを指定します。  
この値は次のどれかになります。

値	意味
UserErr ( 05-10 以降 ) または 真 ( True )	ユーザエラーを警告します。
CancelUserErr ( 05-10 以降 ) または 偽 ( False )	ユーザエラーの警告を解除し，実行中に戻します。
Normal ( 05-10 以降 )	正常終了に戻します。

この値は省略できます。省略した場合は真 ( True ) が仮定されます。

#### 説明

トレースビューアのステータスにユーザエラーを警告します。また警告を解除し、トレースビューアのステータスを実行中に戻します。  
ステータスを正常終了に戻すこともできます。

#### 例

```
path1 = _BIN_+"Logging.txt"
Message ( Target_File ,path1 , "実行開始¥n" , 30 , 100 )
:
:
Message ( Target_File ,path1 , "実行終了" )

If IsExistFile ( path1 ) Then
  Alert ( True )
End
```

#### 対象バージョン

JP1/Script 01-00 以降

### 8.14.3 Beep ( スピーカからビープ音を鳴らす )

#### 機能

指定された時間、スピーカからビープ音を鳴らします。

#### 形式

Beep ( [ *Frequency* ] [ , *Time* ] )

#### 指定項目

##### *Frequency*

ビープ音の周波数を数値、または値を格納した変数名で指定します。  
周波数はヘルツ単位で指定します。指定できる値の範囲は、37 から 32767 までです。  
この値は省略できます。省略した場合、440 ヘルツを仮定します。

##### *Time*

ビープ音の周期を数値、または値を格納した変数名で指定します。  
周期はミリ秒単位で指定します。  
この値は省略できます。省略した場合、100 ミリ秒を仮定します。

#### 説明

スピーカからビープ音を鳴らします。

**!** 注意事項

- ・ターミナルサービスで実行する場合、ターミナルサーバ側が Windows 2000 の場合に使用できます。それ以外で実行した場合、「0005: アクセスは拒否されました。ファイルの属性又はセキュリティを見直してください。」などのエラーになります。ターミナルクライアント側のリモートセッションからは Beep コマンドを使用しないでください。
- ・Windows Vista のサービス空間で実行した場合、「0001: ファンクションが間違っています」のエラーになります。Windows Vista のサービス空間から Beep コマンドを使用しないでください。

例

Beep ( 300 ,200 )

対象バージョン

JP1/Script 01-00 以降

## 8.14.4 Exit （スクリプトを終了する）

機能

スクリプトの実行を終了します。

形式

Exit ( [Code] [, Option] )

指定項目

*Code*

終了コードを数値、または値を格納した変数名で指定します。  
この値を省略した場合は、0 が仮定されます。

*Option*

オプションを指定します。  
この値には、次の値を指定できます。

値	意味
Skip	Exec コマンドで呼び出した実行ファイルが終了するのを待たないでスクリプトを終了します。
Abort または Stop	Exec コマンドで呼び出した実行ファイルをスクリプト終了時にすべて強制終了します。

この値は省略できます。省略した場合、スクリプト終了時に Exec コマンドで呼び出した実行ファイルがすべて終了するまで待ちます。

説明

スクリプトの実行を終了します。

なお、JP1/Script の終了コードについては、「6.1.12 JP1/Script の終了コード」を参照してください。

## 例

```

MessageBox ( "スクリプトの実行を終了します。" ,OK )
Exit ( 0 )

```

## 対象バージョン

JP1/Script 01-00 以降

## 8.14.5 GetErrorMessage ( エラーメッセージを取得する )

## 機能

指定されたエラー詳細コードのエラーメッセージを取得します。

## 形式

```
GetErrorMessage ( [ Code ] )
```

## 指定項目

*Code*

エラー詳細コードを数値, または値を格納した変数名で指定します。  
この値は省略できます。省略した場合, 現在の `_RTN_` 予約変数に格納されている値を仮定します。

## 説明

指定されたエラー詳細コードのエラーメッセージを取得し, メッセージを実行結果として返します。

また, エラーメッセージの文字数が半角文字で 1,024 文字を超える場合, 1,024 文字で切り捨てられます。

## 例

```

If Exec ( _WIN_+"NOTEPAD.EXE" ,True ,_BIN_+"Logging.txt" ) Then
  Exit ( _EXEC_RTN_ )
Else
  Dim errMsg
  errMsg = GetErrorMessage ( _RTN_ )
  Message ( Target_File ,_BIN_+"ErrLog.txt" ,errMsg )
  Exit ( 1 )
End

```

## 対象バージョン

JP1/Script 01-01 以降





# 9

## 特殊コマンド

この章では、スクリプトを作成するときに使用できる特殊コマンドについて説明します。

- 
- 9.1 特殊コマンド一覧
  - 9.2 レジストリ操作コマンド
  - 9.3 描画・表示処理コマンド
  - 9.4 チェック処理コマンド
  - 9.5 サービス操作コマンド
  - 9.6 外部プログラム呼び出しコマンド
  - 9.7 ショートカット操作コマンド
  - 9.8 プロセス監視コマンド
  - 9.9 その他のコマンド
-

## 9.1 特殊コマンド一覧

特殊コマンドは、レジストリ操作、サービス操作など、上級者向けのスクリプトをサポートするためのコマンドです。

スクリプトを作成するときに使用できる、特殊コマンドの一覧を表 9-1 に示します。なお、コマンド欄の括弧内の数字は、対象バージョンを表します（括弧がないものは、01-00 からサポートしています）。

レジストリ操作やサービス操作のコマンドで製品の非公開情報を使用するケースでは、あらかじめその製品の開発部署、または販売元に問い合わせをし、合意を得る必要があります。

表 9-1 特殊コマンド一覧

分類	コマンド	意味
レジストリ操作	RegRead	レジストリから値を読み込みます。
	RegWrite	レジストリに値を設定します。
	RegDelete	レジストリの値を削除します。
	RegDeleteKey	レジストリのサブキーを削除します。
画像・表示処理	BitmapShow (05-00 以降)	ビットマップを描画します。
	BitmapHide (05-00 以降)	ビットマップを消去します。
チェック処理	IsEmptyReg	レジストリサブキーの中身が空かどうかをチェックし、結果を真 (True) か偽 (False) で返します。
	IsExistRegKey	レジストリサブキーが存在するかどうかをチェックし、結果を真 (True) か偽 (False) で返します。
	IsExistService	サービスが存在するかどうかをチェックし、結果を真 (True) か偽 (False) で返します。
	IsEmptyGroup (05-20 以降)	プログラムグループの中身が空かどうかをチェックし、結果を真 (True) か偽 (False) で返します。
サービス操作	ServiceSetValue	サービス操作に必要な情報を、サービス情報というオブジェクトとして設定します。
	ServiceGetValue	サービス情報に格納されている情報を取得します。
	ServiceCreate	サービスを登録します。
	ServiceDelete	サービスを削除します。
	ServiceStart	登録しているサービスを開始します。
	ServiceStop	登録しているサービスを停止します。

分類	コマンド	意味
	ServicePause	登録しているサービスを一時停止します。
	ServiceContinue	一時停止しているサービスを再開します。
	ServiceChange	サービスの設定情報を変更します。
	ServiceQuery	サービスの設定情報を取得します。
	ServiceRefer	サービスの現在の情報を取得します。
	ServiceControl	サービスに制御コマンドを送ります。
	GetServiceName ( 05-10 以降 )	サービスの表示名からサービス名を取得します。
外部プログラム呼び出し	CallDll	DLL ファイルを呼び出します。複数パラメタの指定が可能です。
ショートカット操作	MakeGroup ( 05-20 以降 )	プログラムグループを作成します。
	DeleteGroup ( 05-20 以降 )	プログラムグループを削除します。
	MakeShortcut ( 05-20 以降 )	ショートカットを作成します。
	DeleteShortcut ( 05-20 以降 )	ショートカットを削除します。
プロセス監視	GetProcessCount ( 05-20 以降 )	指定したプロセスの起動数とプロセス ID を取得します。
	GetProcessInfo ( 05-20 以降 )	指定したプロセス ID のプロセスの情報を取得します。
	TerminateProcess ( 05-20 以降 )	指定したプロセス ID のプロセスを強制終了します。
その他	ExitWindows ( 05-10 以降 )	スクリプトの実行を終了し、Windows をログオフ、またはシャットダウンします。

## 9.2 レジストリ操作コマンド

---

### 9.2.1 RegRead (レジストリから値を読み込む)

#### 機能

レジストリから値を読み込みます。

#### 形式

```
RegRead ( RegKey , SubKey , ValueBuff , [ EntryName ] ,  
          [ TypeBuff ] [ , Option ] )
```

#### 指定項目

##### *RegKey*

レジストリキーを指定します。  
この値は次のどれかになります。

- HKEY\_CLASSES\_ROOT
- HKEY\_CURRENT\_USER
- HKEY\_LOCAL\_MACHINE
- HKEY\_USERS

HKEY\_LOCAL\_MACHINE を指定し、SubKey の最初のキーに Software を指定した場合、  
HKEY\_CURRENT\_USER¥Software¥Classes¥VirtualStore¥Machine¥Software 下にリダイレクトされます。リダイレクトされたエントリは優先して読み込まれます。

##### *SubKey*

レジストリサブキーを文字列、または値を格納した変数名で指定します。

##### *ValueBuff*

読み込んだ値を受け取る変数名を指定します。

##### *EntryName*

エントリ名を文字列、または値を格納した変数名で指定します。  
この値は省略できます。省略した場合、エントリ名のない値を仮定します。

##### *TypeBuff*

型を受け取る変数名を指定します。  
この変数に格納される値は次のどれかになります。

- REG\_BINARY
- REG\_DWORD
- REG\_DWORD\_BIG\_ENDIAN
- REG\_EXPAND\_SZ
- REG\_LINK
- REG\_MULTI\_SZ

- REG\_NONE
- REG\_RESOURCE\_LIST
- REG\_SZ

この値は省略できます。

*Option*

オプションを指定します。  
この値には次の値を指定できます。

値	意味
10	値が REG_DWORD 型、または REG_DWORD_BIG_ENDIAN 型の場合、読み込んだ値を 10 進数で返します。
16	値が REG_DWORD 型、または REG_DWORD_BIG_ENDIAN 型の場合、読み込んだ値を 16 進数で返します。

この値は省略できます。省略した場合、16 を仮定します。

説明

レジストリから値を読み込み、指定された変数に格納します。コマンドが正常に実行された場合は真 (True) を、エラーが発生した場合は偽 (False) を、コマンドの実行結果として返します。

注意

読み込んだレジストリの型が REG\_MULTI\_SZ の場合、先頭行の値だけ変数に格納されます。

例

```
Dim DBflag
RegRead ( HKEY_LOCAL_MACHINE
          , "Software¥Hitachi¥JP1/Script¥Option" _
          , DBflag , "Debug" )
If DBflag = 1 Then
    Exit
End
```

対象バージョン

JP1/Script 01-00 以降

## 9.2.2 RegWrite ( レジストリに値を登録する )

機能

レジストリに値を登録します。

形式

RegWrite ( RegKey , SubKey , [ EntryName ] , [ Value ] [ , Type ] )

指定項目

### *RegKey*

レジストリキーを指定します。  
この値は次のどれかになります。

- HKEY\_CLASSES\_ROOT
- HKEY\_CURRENT\_USER
- HKEY\_LOCAL\_MACHINE
- HKEY\_USERS

HKEY\_CURRENT\_USER 以外を指定した場合、Windows Vista では管理者権限でスクリプトファイルを実行してください。詳細は、「1.8.4 コマンドの動作」を参照してください。

### *SubKey*

レジストリサブキーを文字列、または値を格納した変数名で指定します。

### *EntryName*

エントリ名を文字列、または値を格納した変数名で指定します。  
この値は省略できます。省略した場合、エントリ名のない値を仮定します。

### *Value*

登録する値を文字列、数値、または値を格納した変数名で指定します。  
この値は省略できます。省略した場合、長さ 0 の文字列、または 0 を仮定します。

### *Type*

登録する値の型を指定します。  
この値は次のどれかになります。

- REG\_DWORD
- REG\_DWORD\_BIG\_ENDIAN
- REG\_EXPAND\_SZ
- REG\_SZ

この値は省略できます。省略した場合、REG\_SZ を仮定します。

### 説明

指定されたレジストリエントリに値を設定します。コマンドが正常に実行された場合は真 (True) を、エラーが発生した場合は偽 (False) を、コマンドの実行結果として返します。エントリが存在しない場合は新規に作成します。

### 例

```
RegWrite ( HKEY_CURRENT_USER _  
           , "Software¥Hitachi¥Script" , "CurrentVersion" _  
           , "0100" , REG_SZ )
```

### 対象バージョン

JP1/Script 01-00 以降

### 9.2.3 RegDelete （レジストリの値を削除する）

#### 機能

レジストリの値を削除します。

#### 形式

`RegDelete ( RegKey , SubKey , EntryName )`

#### 指定項目

##### *RegKey*

レジストリキーを指定します。  
この値は次のどれかになります。

- HKEY\_CLASSES\_ROOT
- HKEY\_CURRENT\_USER
- HKEY\_LOCAL\_MACHINE
- HKEY\_USERS

HKEY\_CURRENT\_USER 以外を指定した場合、Windows Vista では管理者権限でスクリプトファイルを実行してください。詳細は、「1.8.4 コマンドの動作」を参照してください。

##### *SubKey*

レジストリサブキーを文字列、または値を格納した変数名で指定します。

##### *EntryName*

削除するエントリ名を文字列、または値を格納した変数名で指定します。

#### 説明

指定されたレジストリのエントリを削除します。コマンドが正常に実行された場合は真 (True) を、エラーが発生した場合は偽 (False) を、コマンドの実行結果として返します。

#### 例

```
RegDelete ( HKEY_LOCAL_MACHINE _
            , "Software¥Hitachi¥Script" , "CurrentVersion" )
```

#### 対象バージョン

JP1/Script 01-00 以降

### 9.2.4 RegDeleteKey （レジストリのサブキーを削除する）

#### 機能

レジストリのサブキーを削除します。

#### 形式

`RegDeleteKey ( RegKey , SubKey , DelKey )`

## 指定項目

### *RegKey*

レジストリキーを指定します。  
この値は次のどれかになります。

- HKEY\_CLASSES\_ROOT
- HKEY\_CURRENT\_USER
- HKEY\_LOCAL\_MACHINE
- HKEY\_USERS

HKEY\_CURRENT\_USER 以外を指定した場合、Windows Vista では管理者権限でスクリプトファイルを実行してください。詳細は、「1.8.4 コマンドの動作」を参照してください。

### *SubKey*

レジストリサブキーを文字列、または値を格納した変数名で指定します。

### *DelKey*

削除キーを文字列、または値を格納した変数名で指定します。

## 説明

指定されたレジストリサブキー下の削除キーと、そのキーに含まれるすべてのサブキー、およびエントリを削除します。コマンドが正常に実行された場合は真 ( True ) を、エラーが発生した場合は偽 ( False ) を、コマンドの実行結果として返します。

## 例

```
RegDeleteKey ( HKEY_LOCAL_MACHINE  
               , "Software¥Hitachi¥Script" , "Debug" )
```

## 対象バージョン

JP1/Script 01-00 以降



## 9.3 描画・表示処理コマンド

### 9.3.1 BitmapShow (ビットマップを表示する)

#### 機能

ビットマップを表示します。

#### 形式

BitmapShow ( *BmpFileName* , [ *xPos* ] [ , *yPos* ] )

#### 指定項目

##### *BmpFileName*

表示するビットマップファイルのフルパスを文字列、または値を格納した変数名で指定します。

##### *xPos*

画面の左端からビットマップを表示するウィンドウの左端までの水平方向の距離を数値、または値を格納した変数名で指定します。

距離は左端を 0 としたピクセル単位で指定します。

この値を省略すると、ウィンドウは水平方向に対して画面の中央の位置に配置されます。また、負の値を指定した場合、0 を仮定します。

##### *yPos*

画面の上端からビットマップを表示するウィンドウの上端までの垂直方向の距離を数値、または値を格納した変数名で指定します。

距離は上端を 0 としたピクセル単位で指定します。

この値を省略すると、ウィンドウは垂直方向に対して画面の中央の位置に配置されます。また、負の値を指定した場合、0 を仮定します。

#### 説明

指定されたビットマップを表示します。コマンドが正常に実行された場合は真 (True) を、エラーが発生した場合は偽 (False) を、コマンドの実行結果として返します。

表示したビットマップを消去する場合は BitmapHide コマンドを使用します。

#### 補足

- スクリプトをサービスで起動する場合は、このコマンドは使用できません。使用すると実行エラーになります。
- Windows Vista で環境変数 ProgramFiles (通常はシステムドライブの Program Files フォルダ)、または WinDir (通常はシステムドライブの Windows フォルダ) に設定されているフォルダ下のビットマップファイルを指定する場合、注意が必要です。詳細は、「1.8.4 コマンドの動作」を参照してください。

#### 例

## 9. 特殊コマンド

```
bmpFile = _WIN_+"ABC.BMP"  
BitmapShow ( bmpFile ,100 ,100 )  
Sleep ( 10000 )  
BitmapHide ( bmpFile )
```

### 対象バージョン

JP1/Script 05-00 以降

## 9.3.2 BitmapHide ( ビットマップを消去する )

### 機能

ビットマップを消去します。

### 形式

```
BitmapHide ( BmpFileName )
```

### 指定項目

*BmpFileName*

消去するビットマップファイルのフルパスを文字列、または値を格納した変数名で指定します。

### 説明

BitmapShow コマンドで表示したビットマップを消去します。コマンドが正常に実行された場合は真 ( True ) を、エラーが発生した場合は偽 ( False ) を、コマンドの実行結果として返します。

また、指定されたビットマップファイルが表示されていない場合は、常に真 ( True ) を返します。

### 補足

スクリプトをサービスで起動する場合は、このコマンドは使用できません。使用すると実行エラーになります。

### 例

```
bmpFile = _WIN_+"ABC.BMP"  
BitmapShow ( bmpFile ,100 ,100 )  
Sleep ( 10000 )  
BitmapHide ( bmpFile )
```

### 対象バージョン

JP1/Script 05-00 以降

## 9.4 チェック処理コマンド

### 9.4.1 IsEmptyReg ( レジストリサブキーの中身が空かどうかを調べる )

#### 機能

レジストリサブキーの中身が空かどうかをチェックし、結果を真 ( True ) か偽 ( False ) で返します。

チェックの際に例外にするレジストリサブキーを指定できます。

#### 形式

```
IsEmptyReg ( RegKey , SubKey [ , ExceptSubKey1 , ExceptSubKey2 , ... ] )
```

#### 指定項目

##### *RegKey*

レジストリキーを指定します。  
この値は次のどれかになります。

- HKEY\_CLASSES\_ROOT
- HKEY\_CURRENT\_USER
- HKEY\_LOCAL\_MACHINE
- HKEY\_USERS

##### *SubKey*

チェックするレジストリサブキーを文字列、または値を格納した変数名で指定します。

##### *ExceptSubKey1 ~ 27*

チェックのときに例外にするレジストリサブキーがある場合、例外にするレジストリサブキーを文字列、または値を格納した変数名で指定します。  
複数のレジストリサブキーを指定できます。

#### 説明

指定されたレジストリサブキーの中にレジストリサブキー、またはエントリが存在するかどうかをチェックし、結果をコマンドの実行結果として返します。中身が空の場合は真 ( True ) を、それ以外の場合は偽 ( False ) を返します。

例外にするレジストリサブキーが指定されている場合は、例外キー以外のレジストリサブキー、またはエントリが存在するかどうかをチェックします。

#### 例

```
' JP1/Scriptのレジストリサブキーの中身が空かどうかを調べる。
Dim regKey ,subKey ,isEmp
regKey = HKEY_LOCAL_MACHINE
subKey = "Software¥Hitachi¥JP1/Script"
isEmp = IsEmptyReg( regKey ,subKey ,"PathName" )
```

## 9. 特殊コマンド

```
If isEmp = True Then
    MsgBox( "Empty !" )
Else
    MsgBox ( "Not Empty !" )
End
```

### 対象バージョン

JP1/Script 01-00 以降

## 9.4.2 IsExistRegKey （レジストリサブキーが存在するかどうかを調べる）

### 機能

レジストリサブキーが存在するかどうかをチェックし、結果を真（True）か偽（False）で返します。

### 形式

IsExistRegKey ( *RegKey* , *SubKey* )

### 指定項目

#### *RegKey*

レジストリキーを指定します。  
この値は次のどれかになります。

- HKEY\_CLASSES\_ROOT
- HKEY\_CURRENT\_USER
- HKEY\_LOCAL\_MACHINE
- HKEY\_USERS

#### *SubKey*

チェックするレジストリサブキーを文字列、または値を格納した変数名で指定します。

### 説明

指定されたレジストリサブキーが存在するかどうかをチェックし、結果をコマンドの実行結果として返します。存在する場合は真（True）を、それ以外の場合は偽（False）を返します。

### 例

```
' JP1/Scriptのレジストリサブキーが存在していたら、削除する。
Dim regKey ,subKey
regKey = HKEY_LOCAL_MACHINE
subKey = "Software¥Hitachi¥JP1/Script"
If IsExistRegKey ( regKey ,subKey ) = True Then
    RegDeleteKey ( regKey ,subKey )
End
```

### 対象バージョン

JP1/Script 01-00 以降

### 9.4.3 IsExistService ( サービスが存在するかどうかを調べる )

#### 機能

サービスが存在するかどうかをチェックし、結果を真 ( True ) か偽 ( False ) で返します。

#### 形式

IsExistService ( *ServiceInfoName* )

#### 指定項目

*ServiceInfoName*

チェックするサービス情報名、またはサービス名を文字列、または値を格納した変数名で指定します。

#### 説明

指定されたサービスが登録されているかどうかをチェックし、結果をコマンドの実行結果として返します。存在する場合は真 ( True ) を、それ以外の場合は偽 ( False ) を返します。

サービス情報名の代わりにサービス名を指定することもできますが、できるだけサービス情報を使ってください。サービス情報とは、サービスに関する情報をカプセル化したオブジェクトのことです。

#### 例

```
' JP1/Script サービスが存在していたら、サービスを削除する。
ServiceSetValue ( "JSService" ,Name::"JP1_Script" )
If IsExistService ( "JSService" ) Then
    ServiceDelete ( "JSService" )
End
```

#### 対象バージョン

JP1/Script 01-00 以降

### 9.4.4 IsEmptyGroup ( ショートカットが存在するかどうか調べる )

#### 機能

プログラムグループにショートカットが存在するかどうかをチェックし、結果を真 ( True ) か偽 ( False ) で返します。

#### 形式

IsEmptyGroup ( *GroupName* [ , *RootType* ] )

#### 指定項目

*GroupName*

プログラムグループ名を、文字列、または値を格納した変数名で指定します。

9. 特殊コマンド

*RootType*

プログラムグループの種類を指定します。  
この値は次のどちらかになります。

値	意味
Lcl_Program	プログラム (共通グループ)
Cur_Program	プログラム (ユーザ別グループ)

この値は省略できます。省略した場合、Cur\_Program を仮定します。

説明

指定されたプログラムグループの中にショートカットが存在するかチェックし、結果をコマンドの実行結果として返します。ショートカットが存在しない場合は真 (True) を、存在する場合は偽 (False) を返します。

指定されたプログラムグループが存在しない場合は、常に偽 (False) を返します。

例

```
' Windows 2000でプログラムグループ"アルファベット"の中に
' ショートカットが存在しない場合は、そのグループを削除する。
If _OS_ = "WIN_NT5.0" Then
  If IsEmptyGroup ( "アルファベット" ) Then
    DeleteGroup ( "アルファベット" )
  End If
End If
```

対象バージョン

JP1/Script 05-20 以降

## 9.5 サービス操作コマンド

### 9.5.1 ServiceSetValue（サービス情報を設定する）

#### 機能

サービス操作に必要な情報をサービス情報というオブジェクトとして設定します。

#### 形式

```
ServiceSetValue ( ServiceInfoName , Keyword1 :: Value1 , Keyword2
:: Value2 , ... )
```

#### 指定項目

##### *ServiceInfoName*

サービス情報名を文字列，または値を格納した変数名で指定します。指定できる文字数は，半角文字で 31 文字以内です。

##### *Keyword1 ~ 11*

設定するキーワードを指定します。

##### *Value1 ~ 11*

設定する値を指定します。

*Keyword1 ~ 11* と *Value1 ~ 11* の区切りに文字には ( :: ) を指定してください。

*Keyword1 ~ 11* で指定するキーワード，および *Value1 ~ 11* で指定する値は次のとおりです。

キーワード	値と意味
Name	サービス名 指定できる文字数は，半角文字で 255 文字以内です。なお，スラッシュ (/) と円記号 (¥) は，サービス名に使用できません。
DispName	表示名 指定できる文字数は，半角文字で 255 文字以内です。
Type	サービスの種類 1 または SERVICE_WIN32_OWN_PROCESS サービスが独立した Win32 プロセスで動作することを指定します。  2 または SERVICE_WIN32_SHARE_PROCESS サービスが Win32 プロセスを他のサービスと共有することを指定します。  3 または SERVICE_KERNEL_DRIVER Windows デバイス ドライバであることを指定します。  4 または SERVICE_FILE_SYSTEM_DRIVER Windows ファイル システム ドライバであることを指定します。

キーワード	値と意味
Start	<p>サービスの起動時期</p> <p>1 または SERVICE_BOOT_START オペレーティングシステム ロードがデバイスドライバを起動することを指定します。</p> <p>この値は、Type で指定されたサービスの種類が次のどちらかのときだけ有効です。</p> <ul style="list-style-type: none"> <li>• 3 または SERVICE_KERNEL_DRIVER</li> <li>• 4 または SERVICE_FILE_SYSTEM_DRIVER</li> </ul> <p>2 または SERVICE_SYSTEM_START IoInitSystem 関数がデバイス ドライバを起動することを指定します。</p> <p>この値は、Type で指定されたサービスの種類が次のどちらかのときだけ有効です。</p> <ul style="list-style-type: none"> <li>• 3 または SERVICE_KERNEL_DRIVER</li> <li>• 4 または SERVICE_FILE_SYSTEM_DRIVER</li> </ul> <p>3 または SERVICE_AUTO_START サービス制御マネージャがシステムの起動時にデバイスドライバや Win32 サービスを自動的に起動することを指定します。</p> <p>4 または SERVICE_DEMAND_START プロセスが StartService 関数を呼び出したときにサービス制御マネージャがデバイスドライバや Win32 サービスを起動することを指定します。</p> <p>5 または SERVICE_DISABLED デバイスドライバや Win32 サービスを起動しないことを指定します。</p>
ErrorCtl	<p>システム起動時にサービスが起動できなかったときのエラーの重大度</p> <p>1 または SERVICE_ERROR_IGNORE 起動（ブート）プログラムは、エラーを記録して、起動処理を続行します。</p> <p>2 または SERVICE_ERROR_NORMAL 起動プログラムは、エラーを記録してメッセージボックスポップアップを表示し、起動処理を続行します。</p> <p>3 または SERVICE_ERROR_SEVERE 起動プログラムはエラーを記録します。</p> <p>4 または SERVICE_ERROR_CRITICAL 起動プログラムは、可能ならばエラーを記録します。</p>
Path	サービスプログラムのフルパス
Group	サービスが所属するロード順序グループ サービスがグループに所属しないときは省略してください。
DependG	依存グループ名 このサービスより前に起動しなければならないロード順序グループ名を指定します。
DependM	依存サービス名 このサービスより前に起動しなければならないサービス名を指定します。このサービスが他のサービスに依存しないときは省略してください。



キーワード	値と意味
StartName	<p>サービスアカウント名 Type で指定されたサービスの種類が 1 または SERVICE_WIN32_OWN_PROCESS の場合 起動時にサービスプロセスがログオンに使用する "DomainName ¥Username" という形式のサービスアカウント名でなければなりません。サービスアカウント名が組み込みドメインに所属するときは、"¥Username" のように指定してください。</p> <p>Type で指定されたサービスの種類が 2 または SERVICE_WIN32_SHARE_PROCESS の場合 LocalSystem 以外のサービスアカウント名を指定することはできません。この値を省略した場合、サービスは "LocalSystem" アカウントでログオンします。その場合、Password の値は省略されていなくてはなりません。</p> <p>Type で指定されたサービスの種類が 3 または SERVICE_KERNEL_DRIVER か、4 または SERVICE_FILE_SYSTEM_DRIVER の場合 この値は、入出力 (I/O) システムがデバイスドライバをロードするときに使用するドライバオブジェクトの名前 (¥FileSystem¥Rdr や、¥Driver¥Xns など) です。この値を省略した場合、I/O システムは、サービス名に基づいたデフォルトのオブジェクト名を作成してドライバを起動します。</p>
Password	<p>アカウントのパスワード Type で指定されたサービスの種類が 1 または SERVICE_WIN32_OWN_PROCESS か、2 または SERVICE_WIN32_SHARE_PROCESS の場合 StartName で指定されたサービスアカウント名のパスワードを指定します。</p> <p>Type で指定されたサービスの種類が 3 または SERVICE_KERNEL_DRIVER か、4 または SERVICE_FILE_SYSTEM_DRIVER の場合 この値は無視されます。</p> <p>サービスにパスワードがないときは省略してください。</p>

### 説明

コマンドの実行結果は、\_SVC\_RTN\_ 予約変数に格納します。コマンドの実行が成功した場合は真 (True) を、それ以外の場合は偽 (False) を返します。

サービス情報とは、サービスに関する情報をカプセル化したオブジェクトのことです。その属性は、上記のキーワードで設定・取得できます。

他のサービス操作コマンドを利用する場合、あらかじめこのコマンドを使って、サービス情報に必要なキーワードの情報を設定しておきます。必要のないキーワードは省略できます。

### 例

```
OutDir = "C:¥Program Files¥Hitachi¥Script¥BIN¥"
ServiceSetValue ( "JSService" ,Name::"JP1_Script" _
                  ,DispName::"JP1/Script" _
                  ,Type::1 _
                  ,Start::3 _
                  ,Errctl::2 _
                  ,Path::OutDir+"SPTHSV.EXE" )
```

または

```
ServiceSetValue ( "JSService" ,Name::"JP1_Script" _
```

```
,DispName::"JP1/Script"  
,Type::SERVICE_WIN32_OWN_PROCESS _  
,Start::SERVICE_AUTO_START _  
,Errctl::SERVICE_ERROR_NORMAL _  
,Path::OutDir+"SPTHSV.EXE" )
```

対象バージョン  
JP1/Script 01-00 以降

9.5.2 ServiceGetValue （サービスの情報を取得する）

機能  
サービス情報に格納されている情報を取得します。

形式  
ServiceGetValue ( ServiceInfoName , Keyword1 :: ValueBuff1 ,  
Keyword2 :: ValueBuff2 , ... )

指定項目  
*ServiceInfoName*  
サービス情報名を文字列，または値を格納した変数名で指定します。  
*Keyword1 ~ 11*  
キーワードを指定します。  
*ValueBuff1 ~ 11*  
値を受け取る変数名を指定します。  
*Keyword1 ~ 11*で指定するキーワード，および *ValueBuff1 ~ 11*で受け取る  
値は次のとおりです。

キーワード	値と意味
Name	サービス名
DispName	表示名
Type	サービスの種類 1または SERVICE_WIN32_OWN_PROCESS サービスが独立した Win32 プロセスで動作することを意味します。  2または SERVICE_WIN32_SHARE_PROCESS サービスが Win32 プロセスを他のサービスと共有することを意味します。  3または SERVICE_KERNEL_DRIVER Windows NT デバイス ドライバであることを意味します。  4または SERVICE_FILE_SYSTEM_DRIVER Windows NT ファイル システム ドライバであることを意味します。

キーワード	値と意味
Start	<p>サービスの起動時期</p> <p>1 または SERVICE_BOOT_START オペレーティングシステム ロードがデバイスドライバを起動することを意味します。</p> <p>2 または SERVICE_SYSTEM_START IoInitSystem 関数がデバイス ドライバを起動することを意味します。</p> <p>3 または SERVICE_AUTO_START サービス制御マネージャがシステムの起動時にデバイスドライバや Win32 サービスを自動的に起動することを意味します。</p> <p>4 または SERVICE_DEMAND_START プロセスが StartService 関数を呼び出したときにサービス制御マネージャがデバイスドライバや Win32 サービスを起動することを意味します。</p> <p>5 または SERVICE_DISABLED デバイスドライバや Win32 サービスを起動しないことを意味します。</p>
Errorctl	<p>システム起動時にサービスが起動できなかったときのエラーの重大度</p> <p>1 または SERVICE_ERROR_IGNORE 起動（ブート）プログラムは、エラーを記録して、起動処理を続行することを意味します。</p> <p>2 または SERVICE_ERROR_NORMAL 起動プログラムは、エラーを記録してメッセージボックスポップアップを表示し、起動処理を続行することを意味します。</p> <p>3 または SERVICE_ERROR_SEVERE 起動プログラムはエラーを記録することを意味します。</p> <p>4 または SERVICE_ERROR_CRITICAL 起動プログラムは、可能ならばエラーを記録することを意味します。</p>
Path	サービスプログラムのフルパス
Group	サービスが所属するロード順序グループ
DependG	依存グループ名
DependM	依存サービス名
StartName	サービスアカウント名
Password	アカウントのパスワード

#### 説明

サービス情報に格納されている情報のうち、必要なキーワードの情報だけを変数に格納します。

コマンドの実行結果は、\_SVC\_RTN\_ 予約変数に格納します。コマンドの実行が成功した場合は真（True）を、それ以外の場合は偽（False）を返します。

#### 例

```
' JP1/Script サービスの情報を取得する。
```

## 9. 特殊コマンド

```
Dim svDispName ,svType ,svStart ,svErrctl ,svPath
ServiceSetValue ( "JSService" ,Name::"JP1_Script" )
ServiceQuery ( "JSService" )
ServiceGetValue ( "JSService" ,DispName::svDispName _
                  ,Type::svType _
                  ,Start::svStart _
                  ,Errctl::svErrctl _
                  ,Path::svPath )
```

対象バージョン

JP1/Script 01-00 以降

### 9.5.3 ServiceCreate （サービスを登録する）

機能

サービスを登録します。

形式

ServiceCreate ( *ServiceInfoName* )

指定項目

*ServiceInfoName*

登録するサービスのサービス情報名を文字列、または値を格納した変数名で指定します。

説明

サービス情報に格納されている情報を使って、システムにサービスを登録します。サービスの登録には少なくともキーワード Name, DispName, Type, Start, Errctl, Path の値が指定されている必要があります。これらのキーワードの詳細は「9.5.1 ServiceSetValue ( サービス情報を設定する )」を参照してください。コマンドの実行結果は、\_SVC\_RTN\_ 予約変数に格納します。コマンドの実行が成功した場合は真 ( True ) を、それ以外の場合は偽 ( False ) を返します。サービスの登録にはシステム管理者権限のあるユーザでログオンしている必要があります。

補足

Windows Vista では管理者権限でスクリプトファイルを実行してください。詳細は、「1.8.4 コマンドの動作」を参照してください。

例

```
' JP1/Script サービスが存在しなかったらサービスを登録する。
ServiceSetValue ( "JSService" ,Name::"JP1_Script" )
If IsExistService ( "JSService" ) = False Then
  OutDir = "C:¥Program Files¥Hitachi¥Script¥BIN¥"
  ServiceSetValue ( "JSService" ,Name::"JP1_Script" _
                  ,DispName::"JP1/Script" _
                  ,Type::SERVICE_WIN32_OWN_PROCESS _
                  ,Start::SERVICE_AUTO_START _
```

```

,Errctl::SERVICE_ERROR_NORMAL
,Path::OutDir+"SPTHSV.EXE" )
ServiceCreate ( "JSService" )
End

```

対象バージョン

JP1/Script 01-00 以降

## 9.5.4 ServiceDelete （サービスを削除する）

機能

登録されているサービスを削除します。

形式

```
ServiceDelete ( ServiceInfoName )
```

指定項目

*ServiceInfoName*

登録を解除するサービスのサービス情報名を文字列、または値を格納した変数名で指定します。

説明

サービス情報に格納されているキーワード Name の値を使って、システムに登録されているサービスを削除します。

コマンドの実行結果は、\_SVC\_RTN\_ 予約変数に格納します。コマンドの実行が成功した場合は真 ( True ) を、それ以外の場合は偽 ( False ) を返します。

補足

Windows Vista では管理者権限でスクリプトファイルを実行してください。詳細は、「1.8.4 コマンドの動作」を参照してください。

### ！ 注意事項

停止中の状態にならないサービスに対してこのコマンドを実行しないでください。このコマンドは、サービスを削除するため指定したサービスに対して停止要求をして、指定したサービスが停止中の状態になり削除できるまで終了しません。

例

```

' JP1/Script サービスが存在していたらサービスを削除する。
ServiceSetValue ( "JSService" ,Name::"JP1_Script" )
If IsExistService ( "JSService" ) Then
  ServiceDelete ( "JSService" )
End

```

対象バージョン

JP1/Script 01-00 以降

### 9.5.5 ServiceStart ( サービスを開始する )

#### 機能

登録されているサービスを開始します。

#### 形式

ServiceStart ( *ServiceInfoName* )

#### 指定項目

*ServiceInfoName*

開始するサービスのサービス情報名を文字列、または値を格納した変数名で指定します。

#### 説明

サービス情報に格納されているキーワード Name の値を使って、システムに登録されているサービスを開始します。

コマンドの実行結果は、\_SVC\_RTN\_ 予約変数に格納します。コマンドの実行が成功した場合は真 ( True ) を、アカウント不正のためログオンに失敗しサービスを開始できなかった場合は LOGON\_FAILED を、それ以外の場合は偽 ( False ) を返します。

#### 補足

Windows Vista では管理者権限でスクリプトファイルを実行してください。詳細は、「1.8.4 コマンドの動作」を参照してください。

#### ! 注意事項

実行中の状態にならないサービスに対してこのコマンドを実行しないでください。このコマンドは、指定したサービスに対して開始要求をして、実行中の状態になるまで終了しません。

#### 例

```
' JP1/Script サービスが存在しなかったら登録し、サービスを開始する。
ServiceSetValue ( "JSService" ,Name::"JP1_Script" )
If IsExistService ( "JSService" ) = False Then
  OutDir = "C:¥Program Files¥Hitachi¥Script¥BIN¥"
  ServiceSetValue ( "JSService" ,Name::"JP1_Script" _
    ,DispName::"JP1/Script" _
    ,Type::SERVICE_WIN32_OWN_PROCESS _
    ,Start::SERVICE_AUTO_START _
    ,Errctl::SERVICE_ERROR_NORMAL _
    ,Path::OutDir+"SPTHSV.EXE" )
  ServiceCreate ( "JSService" )
End
ServiceStart ( "JSService" )
```

#### 対象バージョン

JP1/Script 01-00 以降

## 9.5.6 ServiceStop ( サービスを停止する )

### 機能

開始されているサービスを停止します。

### 形式

`ServiceStop ( ServiceInfoName )`

### 指定項目

*ServiceInfoName*

停止するサービスのサービス情報名を文字列, または値を格納した変数名で指定します。

### 説明

サービス情報に格納されているキーワード `Name` の値を使って, すでに起動されているサービスを停止します。

コマンドの実行結果は, `_SVC_RTN_` 予約変数に格納します。コマンドの実行が成功した場合は真 (True) を, それ以外の場合は偽 (False) を返します。

### 補足

Windows Vista では管理者権限でスクリプトファイルを実行してください。詳細は, 「1.8.4 コマンドの動作」を参照してください。

### ！ 注意事項

停止中の状態にならないサービスに対してこのコマンドを実行しないでください。このコマンドは, 指定したサービスに対して停止要求をして, 停止中の状態になるまで終了しません。

### 例

```
' 実行中のJP1/Script サービスを停止する。
Dim svStatus
ServiceSetValue ( "JSService" ,Name::"JP1_Script" )
ServiceRefer ( "JSService" ,svStatus )
If svStatus = SERVICE_RUNNING Then
    ServiceStop ( "JSService" )
End
```

### 対象バージョン

JP1/Script 01-00 以降

## 9.5.7 ServicePause ( サービスを一時停止する )

### 機能

開始されているサービスを一時停止します。

### 形式

## 9. 特殊コマンド

`ServicePause ( ServiceInfoName )`

### 指定項目

*ServiceInfoName*

一時停止するサービスのサービス情報名を文字列、または値を格納した変数名で指定します。

### 説明

サービス情報に格納されているキーワード `Name` の値を使って、すでに起動されているサービスを一時停止します。

このコマンドは、内部的には `ServiceControl` コマンドが実行されています。

コマンドの実行結果は、`_SVC_RTN_` 予約変数に格納します。コマンドの実行が成功した場合は真 (True) を、それ以外の場合は偽 (False) を返します。

### 補足

Windows Vista では管理者権限でスクリプトファイルを実行してください。詳細は、「1.8.4 コマンドの動作」を参照してください。

### 例

```
' 実行中のJP1/Script サービスを一時停止する。
Dim svStatus
ServiceSetValue ( "JSService" ,Name::"JP1_Script" )
ServiceRefer ( "JSService" ,svStatus )
If svStatus = SERVICE_RUNNING Then
    ServicePause ( "JSService" )
End
```

### 対象バージョン

JP1/Script 01-00 以降

## 9.5.8 ServiceContinue (一時停止のサービスを再開する)

### 機能

一時停止しているサービスを再開します。

### 形式

`ServiceContinue ( ServiceInfoName )`

### 指定項目

*ServiceInfoName*

再開するサービスのサービス情報名を文字列、または値を格納した変数名で指定します。

### 説明

サービス情報に格納されているキーワード `Name` の値を使って、一時停止しているサービスを再開します。

このコマンドは、内部的には `ServiceControl` コマンドが実行されています。



コマンドの実行結果は、\_SVC\_RTN\_ 予約変数に格納します。コマンドの実行が成功した場合は真 (True) を、それ以外の場合は偽 (False) を返します。

#### 補足

Windows Vista では管理者権限でスクリプトファイルを実行してください。詳細は、「1.8.4 コマンドの動作」を参照してください。

#### 例

```
' 一時停止しているJP1/Script サービスを再開する。
Dim svStatus
ServiceSetValue ( "JSService" ,Name::"JP1_Script" )
ServiceRefer ( "JSService" ,svStatus )
If svStatus = SERVICE_PAUSED Then
    ServiceContinue ( "JSService" )
End
```

#### 対象バージョン

JP1/Script 01-00 以降

## 9.5.9 ServiceChange ( サービスの設定情報を変更する )

#### 機能

サービスの設定情報を変更します。

#### 形式

```
ServiceChange ( ServiceInfoName )
```

#### 指定項目

*ServiceInfoName*

変更するサービス情報名を文字列、または値を格納した変数名で指定します。

#### 説明

サービス情報に格納されている情報を使って、現在のサービスの設定情報を変更します。

コマンドの実行結果は、\_SVC\_RTN\_ 予約変数に格納します。コマンドの実行が成功した場合は真 (True) を、それ以外の場合は偽 (False) を返します。

#### 補足

Windows Vista では管理者権限でスクリプトファイルを実行してください。詳細は、「1.8.4 コマンドの動作」を参照してください。

#### 例

```
' JP1/Script サービスの設定情報(サービスの起動時期)を変更する。
Dim svStart
ServiceSetValue ( "JSService" ,Name::"JP1_Script" )
ServiceQuery ( "JSService" )
ServiceGetValue ( "JSService" ,Start::svStart )
If svStart = 3 Or svStart = SERVICE_AUTO_START Then
```

## 9. 特殊コマンド

```
svStart = SERVICE_DEMAND_START
ServiceSetValue ( "JSService" ,Start::svStart )
ServiceChange ( "JSService" )
End
```

### 対象バージョン

JP1/Script 01-00 以降

## 9.5.10 ServiceQuery ( サービスの設定情報を取得する )

### 機能

サービスの設定情報を取得します。

### 形式

```
ServiceQuery ( ServiceInfoName )
```

### 指定項目

*ServiceInfoName*

取得するサービス情報名を文字列、または値を格納した変数名で指定します。

### 説明

現在のサービスの設定情報を取得し、サービス情報に格納します。

情報を取得するサービスのサービス名をあらかじめサービス情報のキーワード  
Name に設定しておく必要があります。

コマンドの実行結果は、\_SVC\_RTN\_ 予約変数に格納します。コマンドの実行が成功した場合は真 ( True ) を、それ以外の場合は偽 ( False ) を返します。

### 例

```
' JP1/Script サービスの設定情報(サービスの種類)を取得する。
Dim svType
ServiceSetValue ( "JSService" ,Name::"JP1_Script" )
ServiceQuery ( "JSService" )
ServiceGetValue ( "JSService" ,Type::svType )
MessageBox ( "JP1/Scriptサービスの種類："+svType )
```

### 対象バージョン

JP1/Script 01-00 以降

## 9.5.11 ServiceRefer ( サービスの現在の状態を取得する )

### 機能

サービスの現在の状態を取得します。

### 形式

```
ServiceRefer ( ServiceInfoName , StatusBuff )
```

### 指定項目

*ServiceInfoName*

サービス情報名を文字列，または値を格納した変数名で指定します。

#### *StatusBuff*

現在の状態を受け取る変数名を指定します。

この変数に格納される値は次のどれかになります。

値	意味
SERVICE_STOPPED	サービスは実行中ではありません。
SERVICE_START_PENDING	サービスは起動中です。
SERVICE_STOP_PENDING	サービスは停止中です。
SERVICE_RUNNING	サービスは実行中です。
SERVICE_CONTINUE_PENDING	サービスの再開は未処理です。
SERVICE_PAUSE_PENDING	サービスの一時停止は未処理です。
SERVICE_PAUSED	サービスは一時停止されています。

#### 説明

現在のサービスの状態を取得し，結果を変数に格納します。

状態を取得するサービスのサービス名をあらかじめサービス情報のキーワード Name に設定しておく必要があります。

コマンドの実行結果は，\_SVC\_RTN\_ 予約変数に格納します。コマンドの実行が成功した場合は真（True）を，それ以外の場合は偽（False）を返します。

#### 例

```
' JP1/Script サービスの現在の状態を取得し，サービスを開始する。
Dim svStatus
ServiceSetValue ( "JSService" ,Name::"JP1_Script" )
ServiceRefer ( "JSService" ,svStatus )
If svStatus = SERVICE_STOPPED Then
    ServiceStart ( "JSService" )
End
```

#### 対象バージョン

JP1/Script 01-00 以降

## 9.5.12 ServiceControl （サービスに制御コマンドを送る）

#### 機能

サービスに制御コマンドを送ります。

#### 形式

```
ServiceControl ( ServiceInfoName , Control )
```

#### 指定項目

##### *ServiceInfoName*

サービス情報名を文字列，または値を格納した変数名で指定します。

*Control*

サービスに送る制御コマンドを指定します。  
この値は次のどれかになります。

値	意味
SERVICE_CONTROL_STOP	サービスを停止します。 ServiceStop コマンドと異なり、サービスが開始されていない場合はエラーになります。
SERVICE_CONTROL_PAUSE	サービスを一時停止します。 ServicePause コマンドと同じ機能です。
SERVICE_CONTROL_CONTINUE	サービスを再開します。 ServiceContinue コマンドと同じ機能です。

## 説明

サービス情報に格納されているキーワード *Name* の値を使って、サービスに制御コマンドを送ります。  
制御するサービスのサービス名をあらかじめサービス情報の *Name* に設定しておく必要があります。  
コマンドの実行結果は、*\_SVC\_RTN\_* 予約変数に格納します。コマンドの実行が成功した場合は真 (True) を、それ以外の場合は偽 (False) を返します。

## 補足

Windows Vista では管理者権限でスクリプトファイルを実行してください。詳細は、「1.8.4 コマンドの動作」を参照してください。

## 例

```
' JP1/Script サービスを一時停止する。
ServiceSetValue ( "JSService" ,Name::"JP1_Script" )
ServiceControl ( "JSService" ,SERVICE_CONTROL_PAUSE )
```

## 対象バージョン

JP1/Script 01-00 以降

## 9.5.13 GetServiceName ( サービス表示名からサービス名を取得する )

## 機能

サービスの表示名からサービス名を取得します。

## 形式

```
GetServiceName ( DispName )
```

## 指定項目

*DispName*

サービスの表示名を文字列、または値を格納した変数名で指定します。

**説明**

指定されたサービスの表示名からサービス名を取得します。コマンドが正常に実行された場合はサービス名を、エラーが発生した場合は長さ 0 の文字列 ("" ) を、コマンドの実行結果として返します。

**例**

```
' 表示名が"JP1/Script"であるサービスを停止する。
svName = GetServiceName ( "JP1/Script" )
If IsEmpty ( svName ) = False Then
    ServiceSetValue ( "JSService" ,Name::svName )
    If IsExistService ( "JSService" ) Then
        ServiceStop ( "JSService" )
    End
End
```

**対象バージョン**

JP1/Script 05-10 以降

## 9.6 外部プログラム呼び出しコマンド

### 9.6.1 CallDll ( DLL ファイルを呼び出す )

#### 機能

DLL ファイルを呼び出します。

#### 形式

CallDll ( *DllFileName* , *FunctionName* , *Param1* , *Param2* , ... )

#### 指定項目

##### *DllFileName*

DLL ファイル名を、文字列、または値を格納した変数名で指定します。

##### *FunctionName*

呼び出す外部関数名を、文字列、または値を格納した変数名で指定します。

##### *Param1 ~ 32*

関数のパラメタを、文字列、または値を格納した変数名で指定します。

複数のパラメタを指定できます。

JP1/Script 06-00 以降では、複数のパラメタを一つずつ格納した配列変数を変数名で指定することもできます。

#### 説明

指定された DLL をロードし、外部関数をコールします。コマンドが正常に実行された場合は真 ( True ) を、エラーが発生した場合は偽 ( False ) を、コマンドの実行結果として返します。

外部関数は次のインタフェースにしてください。また、モジュール定義ファイル ( .DEF ) の EXPORTS セクションに外部参照宣言が必要になります。

#### 宣言

```
BOOL WINAPI MyFunc( HWND hParent, int argc, char * argv[],
int * rtnc, char * * rtnv[] );
```

#### 引数

```
HWND hParent;      ' 親ウィンドウのハンドル
int  argc;         ' スクリプトから渡されるパラメタの数
char * argv[];     ' スクリプトから渡されるパラメタの配列
int * rtnc;        ' スクリプトに返す文字列の数を格納するポインタ
char * * rtnv[];   ' スクリプトに返す文字列の配列を格納するポインタ
```

#### 戻り値

外部関数 MyFunc の戻り値は \_DLL\_RTN\_ 予約変数に格納します。

\_DLL\_RTN\_ 予約変数が True の場合、引数 rtn に格納されたスクリプトに返す文字列の配列はスクリプト上で \_RTNxx\_ 予約変数を使って参照できます。xx は 00 から順に引数 rtnc に格納された数だけ割り振られます。外部関数

MyFunc の戻り値が False の場合, \_RTNxx\_ 予約変数は定義されません。  
 スクリプトに返す文字列, および文字列配列は, 関数終了後も参照できるように, 関数のスコープ内のローカルバッファには格納しないでください。

例

```
' TEST.DLLの関数部分(Favorite.c)
#include "windows.h"

#define FAV_SPORT1 "SKI"
#define FAV_SPORT2 "BASKETBALL"
#define FAV_FOOD1 "STEAK"
#define FAV_FOOD2 "PASTA"
char * g_ret[2];

BOOL WINAPI GetFavorite( HWND hParent, int argc, char * argv[],
int * rtnnc, char ** rtnv[] )
{
    if (lstrcmp(argv[0], "SPORT") == 0)
    {
        g_ret[0] = FAV_SPORT1;
        g_ret[1] = FAV_SPORT2;
    }
    else if (lstrcmp(argv[0], "FOOD") == 0)
    {
        g_ret[0] = FAV_FOOD1 ;
        g_ret[1] = FAV_FOOD2;
    }

    *rtnnc = 2;
    *rtnv = g_ret;

    return ( TRUE );
}

' スクリプト側の処理(abc.SPT)
Dim Quest
Quest = "SPORT"
CallDll(_BIN_+"TEST.DLL" ,"GetFavorite" ,Quest)
Dim Msg
Msg = "私の好きな"+Quest+"は"+_RTN00_+"と"+_RTN01_+"です"
MessageBox(Msg ,OK,,Information)
Exit(0)
```

対象バージョン

JP1/Script 01-00 以降

## 9.7 ショートカット操作コマンド

### 9.7.1 MakeGroup （プログラムグループを作成する）

#### 機能

プログラムグループを作成します。

#### 形式

MakeGroup ( *GroupName* [, *RootType*] )

#### 指定項目

##### *GroupName*

プログラムグループ名を文字列，または値を格納した変数名で指定します。  
Lcl\_Program を指定した場合，Windows Vista では管理者権限でスクリプト  
ファイルを実行してください。詳細は，「1.8.4 コマンドの動作」を参照してく  
ださい。

##### *RootType*

プログラムグループの種類を指定します。  
この値は次のどれかになります。

値	意味
Lcl_Program	プログラム（共通グループ）
Cur_Program	プログラム（ユーザ別グループ）

この値は省略できます。省略した場合，Cur\_Program を仮定します。

#### 説明

指定されたプログラムグループを作成します。コマンドが正常に実行された場合は  
真（True）を，エラーが発生した場合は偽（False）をコマンドの実行結果として返  
します。

指定されたプログラムグループがすでに存在している場合，何もしないで，常に真  
（True）を返します。

#### 例

```
' Windows 2000でプログラムグループ"アルファベット"を作成し，
' そのグループにショートカット"ABC"を登録する。
If _OS_ = "WIN_NT5.0" Then
    MakeGroup ( "アルファベット" )
    MakeShortcut ( Cur_Program , "アルファベット¥ABC", _
        _BIN_+"ABC.SPT" )
End If
```

#### 対象バージョン

JP1/Script 05-20 以降



## 9.7.2 DeleteGroup （プログラムグループを削除する）

### 機能

プログラムグループを削除します。

### 形式

DeleteGroup ( *GroupName* [ , *RootType* ] )

### 指定項目

#### *GroupName*

プログラムグループ名を文字列、または値を格納した変数名で指定します。  
Lcl\_Program を指定した場合、Windows Vista では管理者権限でスクリプト  
ファイルを実行してください。詳細は、「1.8.4 コマンドの動作」を参照してく  
ださい。

#### *RootType*

プログラムグループの種類を指定します。  
この値は次のどちらかになります。

値	意味
Lcl_Program	プログラム（共通グループ）
Cur_Program	プログラム（ユーザ別グループ）

この値は省略できます。省略した場合、Cur\_Program を仮定します。

### 説明

指定されたプログラムグループを削除します。コマンドが正常に実行された場合は  
真（True）を、エラーが発生した場合は偽（False）をコマンドの実行結果として返  
します。  
指定されたプログラムグループが存在しない場合、何もしないで常に真（True）を  
返します。

### 例

```
' Windows 2000でプログラムグループ"アルファベット"を削除する。
If _OS_ = "WIN_NT5.0" Then
    DeleteGroup( "アルファベット" )
End If
```

### 対象バージョン

JP1/Script 05-20 以降

## 9.7.3 MakeShortcut （ショートカットを作成する）

### 機能

ショートカットを作成します。

## 9. 特殊コマンド

### 形式

```
MakeShortcut ( RootPath , SubPath , LinkPath , [ Param ] ,  
               [ WorkDirPath ] , [ IconPath ] , [ IconIndex ] [ , IconFlag ] )
```

### 指定項目

#### *RootPath*

ショートカットの作成場所を指定します。  
この値は次のどれかになります。

値	意味
None	現在の実行フォルダ下に作成されます。
Lcl_Desktop	デスクトップ（共通グループ）
Lcl_Startmenu	スタートメニュー（共通グループ）
Lcl_Program	プログラム（共通グループ）
Lcl_Startup	スタートアップ（共通グループ）
Cur_Desktop	デスクトップ（ユーザ別グループ）
Cur_Startmenu	スタートメニュー（ユーザ別グループ）
Cur_Program	プログラム（ユーザ別グループ）
Cur_Startup	スタートアップ（ユーザ別グループ）

Lcl\_Desktop , Lcl\_Startmenu , Lcl\_Program または Lcl\_Startup を指定した場合、Windows Vista では管理者権限でスクリプトファイルを実行してください。詳細は、「1.8.4 コマンドの動作」を参照してください。

#### *SubPath*

ショートカット名を、文字列、または値を格納した変数名で指定します。  
*RootPath* で指定されたショートカットの作成場所が Lcl\_Program , または Cur\_Program の場合、次のように指定してください。  
プログラムグループ¥ショートカット名

#### *LinkPath*

ショートカットのリンク先を、文字列、または値を格納した変数名で指定します。

#### *Param*

*LinkPath* で指定されたファイルの実行に必要なパラメタがある場合、パラメタを文字列、または値を格納した変数名で指定します。

#### *WorkDirPath*

作業フォルダ名を文字列、または値を格納した変数名で指定します。  
この値は省略できます。省略した場合、*LinkPath* で指定されたリンク先のフォルダを仮定します。

#### *IconPath*

ショートカットに使うアイコンが登録されているアイコンファイル名を文字列、

または値を格納した変数名で指定します。

この値は省略できます。省略した場合、*LinkPath* で指定されたリンク先を仮定します。

#### *IconIndex*

*IconPath* で指定されたアイコンファイル中のアイコンのインデックスを 0 から始まる数値、または値を格納した変数名で指定します。

この値は省略できます。省略した場合、0 を仮定します。

#### *IconFlag*

ショートカットを起動したときのウィンドウの表示方法を指定します。

この値は次のどれかになります。

値	意味
Normal	通常のウィンドウ
Min	最小化（タスクバーのボタン）
Max	最大化（画面全体）

この値は省略できます。省略した場合、Normal を仮定します。

#### 説明

指定されたショートカットを作成します。コマンドが正常に実行された場合は真（True）を、エラーが発生した場合は偽（False）をコマンドの実行結果として返します。

指定された名前のショートカットがすでに存在する場合、そのショートカットは削除され、指定されたパラメタで登録し直します。

#### 例

```
' Windows 2000でスタートアップに"ABC.SPT"を呼び出す
' ショートカットを作成する。
If _OS_ = "WIN_NT5.0" Then
    MakeShortcut ( Cur_Startup , "ABC" , _BIN_ + "ABC.SPT" _
                  , , , _TEMP_ , , , Min )
End If
```

#### 対象バージョン

JP1/Script 05-20 以降

## 9.7.4 DeleteShortcut （ショートカットを削除する）

#### 機能

ショートカットを削除します。

#### 形式

```
DeleteShortcut ( RootPath , SubPath )
```

#### 指定項目

## 9. 特殊コマンド

### *RootPath*

ショートカットの存在場所を指定します。  
この値は次のどれかになります。

値	意味
None	現在の実行フォルダ下
Lcl_Desktop	デスクトップ（共通グループ）
Lcl_Startmenu	スタートメニュー（共通グループ）
Lcl_Program	プログラム（共通グループ）
Lcl_Startup	スタートアップ（共通グループ）
Cur_Desktop	デスクトップ（ユーザ別グループ）
Cur_Startmenu	スタートメニュー（ユーザ別グループ）
Cur_Program	プログラム（ユーザ別グループ）
Cur_Startup	スタートアップ（ユーザ別グループ）

Lcl\_Desktop , Lcl\_Startmenu , Lcl\_Program または Lcl\_Startup を指定した場合、Windows Vista では管理者権限でスクリプトファイルを実行してください。詳細は、「1.8.4 コマンドの動作」を参照してください。

### *SubPath*

ショートカット名を、文字列、または値を格納した変数名で指定します。  
*RootPath* で指定されたショートカットの存在場所が Lcl\_Program , または Cur\_Program の場合、次のように指定してください。  
プログラムグループ¥ショートカット名

### 説明

指定されたショートカットを削除します。コマンドが正常に実行された場合は真（True）を、エラーが発生した場合は偽（False）を、コマンドの実行結果として返します。

指定された名前のショートカットが存在しない場合、何もしないで、常に真（True）を返します。

### 例

```
' Windows 2000でスタートアップからショートカット"ABC"を削除する。
If _OS_ = "WIN_NT5.0" Then
    DeleteShortcut ( Cur_Startup , "ABC" )
End If
```

### 対象バージョン

JP1/Script 05-20 以降

## 9.8 プロセス監視コマンド

### 9.8.1 GetProcessCount ( プロセスの起動数を取得する )

#### 機能

プロセスの起動数を取得します。

#### 形式

```
GetProcessCount ( ProcessName [ , ProcessIdBuff ] )
```

#### 指定項目

##### *ProcessName*

プロセス名を文字列、または値を格納した変数名で指定します。

プロセス名にはフォルダ名や ".EXE" などの拡張子を付けずにラベル名だけを指定してください。

##### *ProcessIdBuff*

プロセス ID を受け取る変数名を指定します。

JP1/Script 06-00 以降では、要素数が可変な一次元の配列変数の変数名を指定することもできます。

必要がない場合は省略します。

*ProcessName* で指定されたプロセスが複数起動されている場合は、複数のプロセス ID を起動順にソートして、次のどちらかのように格納します。

- *ProcessIdBuff* に変数が指定された場合は、複数のプロセス ID をセミコロン (;) で区切った文字列にして格納します。
- *ProcessIdBuff* に配列変数が指定された場合は、複数のプロセス ID を先頭の要素から順に一つずつ格納します。格納後、配列変数の要素数はプロセス ID の数に置き換わります。

#### 説明

指定されたプロセスの起動数を取得し、コマンドの実行結果として返します。

*ProcessIdBuff* が指定されている場合はプロセス ID も返します。

指定されたプロセスが起動されていない場合は 0 を返し、*ProcessIdBuff* には長さ 0 の文字列 ("" ) を格納します。エラーが発生した場合は長さ 0 の文字列 ("" ) を、コマンドの実行結果として返します。

#### 補足

- ローカルシステムアカウントで起動したプロセスについては完全な名前が取得できないという Windows の仕様上の制限によって GetProcessCount コマンドにローカルシステムアカウントで起動した 16 文字以上のプロセス名 ( 拡張子を含める ) のラベル名を指定して、ローカルシステムアカウント以外のアカウントから実行した場合、該当するプロセスが起動中にもかかわらず、起動数、およびプロセス ID に値が返らないという現象が起こります。

この現象を回避するためには、*ProcessName* に拡張子を含めたプロセス名の 15 文字までの部分を指定してください。詳細は、例 2 を参照してください。

- Windows Vista では管理者権限でスクリプトファイルを実行してください。詳細は、「1.8.4 コマンドの動作」を参照してください。

例 1

```
' プロセス"ABC"のプロセスID一覧を作成する。
Dim pCnt ,pIdBuff ,seppId
pCnt = GetProcessCount ( "ABC" ,pIdBuff )
For i = 1 To pCnt
    seppId = SeparateStr ( pIdBuff ,";" ,i )
    Message ( Target_File ,_SCF_+"PIDLIST.TXT" ,seppId )
Next
```

例 2

```
Dim PLName, Pname, PnameWork, Rc, IdBuff
PLName = "ABCDEFGHijkl"
Pname = "ABCDEFGHijkl.exe"
' まず、通常の指定方法（ラベル名を指定）で実行する。
Rc = GetProcessCount(PLName, IdBuff)
' 起動数が0の場合、プロセスがローカルシステムアカウントから
' 起動されている場合を考慮した処理を行う。
If Rc = 0 Then
    If Len(Pname) > 15 Then
        PnameWork = Left(Pname, 15)
    End If
    Rc = GetProcessCount(PnameWork, IdBuff)
End If
```

対象バージョン

JP1/Script 05-20 以降

## 9.8.2 GetProcessInfo （プロセスの情報を取得する）

機能

指定されたプロセス ID のプロセスの情報を取得します。

形式

```
GetProcessInfo ( ProcessId , {StartTimeBuff} { ,
ProcessorTimeBuff} )
```

指定項目

*ProcessId*

プロセス ID を数値、または値を格納した変数名で指定します。

このプロセス ID は、GetProcessCount コマンドで引数に返される値です。

また、JP1/Script 06-00 以降では、Exec、および NetExec コマンド実行時に返される \_EXEC\_ID\_ 予約変数を指定することができます。この場合、Exec、および NetExec コマンドの終了待ちなしで呼び出した実行ファイルのプロセスの情報を取得します。

*StartTimeBuff*

プロセスの起動時刻を受け取る変数名を指定します。必要がない場合は省略します。

この変数に格納される値は hh:mm:ss の形式です。

*ProcessorTimeBuff*

プロセスの稼働時間を受け取る変数名を指定します。必要がない場合は省略します。

この変数に格納される値は秒単位です。

## 説明

指定されたプロセス ID のプロセスの情報を取得し、各変数に格納します。コマンドが正常に実行された場合は真 (True) を、エラーが発生した場合は偽 (False) を、コマンドの実行結果として返します。

## 補足

Windows Vista では管理者権限でスクリプトファイルを実行してください。詳細は、「1.8.4 コマンドの動作」を参照してください。

## 例

```
' 起動中のプロセス"ABC"の情報を取得する。
Dim pCnt ,pIdBuff ,seppId ,infBuff1 ,infBuff2
pCnt = GetProcessCount ( "ABC" ,pIdBuff )
For i = 1 To pCnt
    seppId = SeparateStr ( pIdBuff ,";" ,i )
    GetProcessInfo ( seppId ,infBuff1 ,infBuff2 )
    :
    :
Next
```

## 対象バージョン

JP1/Script 05-20 以降

### 9.8.3 TerminateProcess ( プロセスを強制終了する )

## 機能

指定されたプロセス ID のプロセスを強制終了します。

## 形式

```
TerminateProcess ( ProcessId [ , Code ] )
```

## 指定項目

*ProcessId*

プロセス ID を数値、または値を格納した変数名で指定します。

このプロセス ID は、GetProcessCount コマンドで引数に返される値です。

また、JP1/Script 06-00 以降では、Exec、および NetExec コマンド実行時に返される \_EXEC\_ID\_ 予約変数を指定することができます。この場合、Exec、お

## 9. 特殊コマンド

よび NetExec コマンドの終了待ちなしで呼び出した実行ファイルのプロセスを強制終了します。

### Code

強制終了するプロセスに対する終了コードを数値、または値を格納した変数名で指定します。

この値を省略した場合は、0 が仮定されます。

### 説明

指定されたプロセス ID のプロセスを強制終了します。コマンドが正常に実行された場合は真 (True) を、エラーが発生した場合は偽 (False) を、コマンドの実行結果として返します。

また、指定されたプロセス ID のプロセスが存在しない場合は、常に真 (True) を返します。

### 補足

- このコマンドが実行された場合、イベントビューアのアプリケーションログにイベントログを出力します。イベントログのイベント ID は 7 になります。  
ただし、実行環境で定義されたコマンドラインやレジストリのコマンドラインに /NOEVLOG や /NOEVLOG(7) などの指定がある場合、イベントログの出力を抑止します。
- Windows Vista では管理者権限でスクリプトファイルを実行してください。詳細は、「1.8.4 コマンドの動作」を参照してください。また、管理者権限で実行しなかった場合、他ユーザで実行中のプロセスを強制終了できません (スクリプトプロセスは強制終了できます)。

### 例

・ 起動中のプロセス "ABC" のうち最初に起動されたプロセス以外のプロセスを強制終了する。

```
Dim pCnt, pIdBuff, seppId
pCnt = GetProcessCount ( "ABC" ,pIdBuff )
If pCnt > 1 Then
    For i = 2 To pCnt
        seppId = SeparateStr ( pIdBuff ,";" ,i )
        TerminateProcess ( seppId )
    Next
End If
```

### 対象バージョン

JP1/Script 05-20 以降



## 9.9 その他のコマンド

### 9.9.1 ExitWindows（スクリプトを終了し Windows をログオフ/シャットダウンする）

#### 機能

スクリプトの実行を終了し、Windows をログオフ、またはシャットダウンします。

#### 形式

ExitWindows ( [Code] [, Option1 , Option2] )

#### 指定項目

##### Code

終了コードを数値、または値を格納した変数名で指定します。  
この値を省略した場合は、0 が仮定されます。

##### Option1

シャットダウン操作の種別を指定します。  
この値には次の値を指定できます。

値	意味
Logoff	すべてのプロセスをシャットダウン後、カレントユーザをログオフします。
Poweroff	システムをシャットダウンし、電源を切ります。 シャットダウン権限を持っていない限りなりません。また、システムがソフトウェアから電源を切る機能をサポートしていない限りなりません。
Reboot	システムをシャットダウン後、システムを再起動します。 シャットダウン権限を持っていない限りなりません。
Shutdown	システムをシャットダウンし、電源を切っても安全な状態にします。 シャットダウン権限を持っていない限りなりません。
Force	シャットダウンの通知に応答しないアプリケーションがあっても「アプリケーションが応答しません」というダイアログを表示しないで、プロセスを強制的に終了させます。

この値は省略できます。省略した場合、Logoff を仮定します。

##### Option2

Option1 で Force 以外を指定した場合に Option1 とともにシャットダウン操作の種別を指定します。  
この値には次の値を指定できます。

値	意味
Force	シャットダウンの通知に応答しないアプリケーションがあっても「アプリケーションが応答しません」というダイアログを表示しないで、プロセスを強制的に終了させます。

説明

スクリプトの実行を終了し、指定されたシャットダウン操作の種別で Windows を終了させます。

なお、JP1/Script の終了コードについては、「6.1.12 JP1/Script の終了コード」を参照してください。

補足

このコマンドが実行された場合、イベントビューアのアプリケーションログにイベントログを出力します。イベントログのイベント ID は 5 になります。

ただし、実行環境で定義されたコマンドラインやレジストリのコマンドラインに /NOEVLOG や /NOEVLOG(5) などの指定がある場合、イベントログの出力を抑止します。

注意事項

他のプログラムやサービスを実行中に ExitWindows コマンドを実行すると、他のプログラムが正常に動作しなくなる危険性があります。使用する場合は十分に注意してください。

また、次の場合、ExitWindows コマンドは有効になりません。

- Windows 2000 Server、Windows 2000 Advanced Server、および Windows Server 2003 でシステムアカウント以外を設定しているサービスからコマンドを実行し、かつログオンしていない状態の場合。
- シャットダウン権限が無いアカウントを設定しているサービスからコマンドを実行した場合。
- シャットダウン権限が無いアカウントでログオンしているコンピュータのログオン空間からコマンドを実行した場合。
- コマンドの Option1、または Option2 に Force の指定が無い場合に動作中のアプリケーションがシャットダウンを拒否したとき。ただし、Windows Vista ではサービスからコマンドを実行した場合、Force の指定の有無にかかわらず有効になります (Logoff は除く)。
- コマンドの Option1、または Option2 に Force の指定が無く、かつログオンしているコンピュータがロック状態である場合。ただし、Windows Vista ではサービスからコマンドを実行した場合、Force の指定の有無にかかわらず有効になります (Logoff は除く)。

例

```
MessageBox ( "スクリプトの実行を終了し、シャットダウンします。" _  
            , OKCancel )  
If _MSG_RTN_ = OK Then  
    ExitWindows ( 0 ,Shutdown )  
Else  
    Exit ( 0 )  
End
```

対象バージョン

JP1/Script 05-10 以降

# 10 スクリプト制御インタフェース

この章では、実行中のスクリプトを制御する、スクリプト制御インタフェース（API）について説明します。

---

10.1 スクリプト制御インタフェースとは

---

10.2 スクリプト制御インタフェースの呼び出し

---

10.3 スクリプト制御インタフェース一覧

---

10.4 スクリプト制御インタフェースの記述例

---

## 10.1 スクリプト制御インタフェースとは

---

スクリプト制御インタフェース（API）とは、実行中のスクリプトを制御するインタフェースです。JP1/Script の一部としてインストールされる SPTHL.DLL に含まれ、他のプログラムから LoadLibraryEx を使用して呼び出すことができます。

## 10.2 スクリプト制御インタフェースの呼び出し

---

スクリプト制御インタフェースを、他のプログラムから LoadLibraryEx を使用して呼び出す方法を次に示します。

### 10.2.1 SPTHL.DLL のパスの取得方法

レジストリから SPTHL.DLL のパスを取得するには、次の二つの方法があります。

#### (1) 方法 1

JP1/Script 05-00 以降では、次のレジストリから取得できます。

〔サブキー〕

HKEY\_LOCAL\_MACHINE¥SOFTWARE¥Hitachi¥JP1/Script¥PathName

〔値名〕

Path01

〔値のデータタイプ〕

REG\_SZ

#### (2) 方法 2

JP1/Script 05-10 以降では、次のレジストリが設定されています。

JP1/Script 05-00 以前には次のレジストリキーは存在しないため、この機能がサポートされているかチェックする必要はありません。

〔サブキー〕

HKEY\_LOCAL\_MACHINE¥SOFTWARE¥Hitachi¥JP1/  
Script¥CancelInformation

〔値名〕

LoadPath

〔値のデータタイプ〕

REG\_SZ

### 10.2.2 ロード方法

LoadLibraryEx で SPTHL.DLL をロードします。

その際、dwFlag に LOAD\_WITH\_ALTERED\_SEARCH\_PATH を指定してください。

### 10.2.3 本機能がサポートされているかのチェック方法

次のレジストリキーを参照し、JP1/Script のバージョンが 05-00 より後かどうかを確認してください。

〔サブキー〕

HKEY\_LOCAL\_MACHINE¥SOFTWARE¥Hitachi¥JP1/Script

〔値名〕

CurrentVersion

〔値のデータタイプ〕

REG\_SZ

〔値のデータ形式〕

JP1/Script 05-00 の場合は "0500" , 05-00-/A の場合は "0500/A" の形式で設定されます。

## 10.3 スクリプト制御インタフェース一覧

スクリプト制御インタフェースの一覧を次に示します。

コマンド	意味
SPTHOpen	スクリプト制御マネージャをオープンします。
SPTHTClose	スクリプト制御マネージャをクローズします。
SPTHGetErrorMessage	エラーメッセージを取得します。
SPTHTerminate	スクリプト実行を強制終了します。

### 10.3.1 SPTHOpen（スクリプト制御マネージャをオープンする）

#### 機能

スクリプト制御マネージャをオープンします。  
スクリプト制御マネージャとは、スクリプト実行プロセスの制御を行うプログラムです。

#### プロトタイプ宣言

```

BOOL  APIENTRY  SPTHOpen
(
    LPCSTR  lpszComputerName ,
    HWND    hWnd ,
    LPHANDLE lphScript
) ;

```

#### 引数

*lpszComputerName*

スクリプトのコンピュータ名を指定します。  
この値に NULL を指定した場合、現在のコンピュータ名を仮定します。

*hWnd*

コール元のウィンドウハンドルを指定します。

*lphScript*

スクリプト制御ハンドルを受け取る領域のポインタを指定します。

#### 戻り値

成功した場合は TRUE を、失敗した場合は FALSE を返します。拡張エラーコードは GetLastError 関数で取得できます。

#### 対象バージョン

JP1/Script 05-00 以降

### 10.3.2 SPTHClose ( スクリプト制御マネージャをクローズする )

#### 機能

スクリプト制御マネージャをクローズします。

#### プロトタイプ宣言

```
void APIENTRY SPTHClose  
(  
    HANDLE hScript  
);
```

#### 引数

*hScript*

SPTHOpen 関数で受け取ったスクリプト制御ハンドルを指定します。

#### 戻り値

この関数に戻り値はありません。

#### 対象バージョン

JP1/Script 05-00 以降

### 10.3.3 SPTHGetErrorMessage ( エラーメッセージを取得する )

#### 機能

エラーメッセージを取得します。

#### プロトタイプ宣言

```
DWORD APIENTRY SPTHGetErrorMessage  
(  
    DWORD dwErrorCode ,  
    LPSTR lpszMessage ,  
    DWORD dwSize  
);
```

#### 引数

*dwErrorCode*

GetLastError 関数で取得したエラーコードを指定します。

*lpszMessage*

エラーメッセージを受け取るバッファのポインタを指定します。

*dwSize*

エラーメッセージを受け取るバッファのバイト数 ( ' ¥0 ' を含む ) を指定します。バッファサイズが不足した場合は、入る部分だけが設定されます。

#### 戻り値



エラーメッセージの長さ（ ' ¥0 ' を含まない）を返します。

対象バージョン

JP1/Script 05-00 以降

### 10.3.4 SPTHTerminate （スクリプト実行を強制終了する）

機能

スクリプト実行を強制終了します。

プロトタイプ宣言

```
BOOL  APIENTRY  SPTHTerminate
(
    HANDLE  hScript ,
    LPCSTR  lpszFileName ,
    UINT    uProcessID ,
    DWORD   dwOption
);
```

引数

*hScript*

SPTHOpen 関数で受け取ったスクリプト制御ハンドルを指定します。

*lpszFileName*

スクリプトファイル名をフルパスで指定します。

この値は SPTHOpen 関数で指定したコンピュータから見たファイルパスでなければなりません。

*uProcessID* を指定した場合、この値を省略（NULL を指定）することができます。

*uProcessID*

スクリプトのプロセス ID を指定します。

*lpszFileName* を指定した場合、この値を省略（0 を指定）することができます。

*dwOption*

オプションを 0、または次の値の組み合わせで指定します。

値	意味
SPTH_TERM_CHILD	スクリプトが Exec や NetExec コマンドから起動した子プロセスも強制終了する。要求してから 3 分経過しても子プロセスが終了しない場合は、Win32 API の TerminateProcess 関数を使用して強制終了する。
SPTH_DQ_FILENAME	<i>lpszFileName</i> で指定された値の間に空自が含まれている場合、または値がダブルクォーテーションで囲まれる場合は、空自もファイル名の一部とみなし、値がダブルクォーテーションで囲まれていない場合は、空自までをファイル名とみなす。

### 説明

成功した場合は TRUE を、失敗した場合は FALSE を返します。拡張エラーコードは GetLastError 関数で取得できます。

なお、終了すべきプロセスが存在しなくても（すでに終了していても）この関数は正常終了（TRUE を返す）します。

正常終了した場合でも実際にプロセス終了処理を行ったかどうかは GetLastError 関数の戻り値で判定できます。プロセス終了処理を行った場合は NO\_ERROR を返し、そうでない場合はそれ以外の値を返します。

### 備考

*lpzFileName* と *uProcessID* は、どちらかを省略することができますが、両方を指定した方がより確実に終了すべきスクリプトプロセスを特定することができるため、やむを得ない事情が無い限り両方を指定してください。

なお、複数起動したスクリプトプロセスを終了させる場合は、必ず *uProcessID* を指定してください。

この関数によってスクリプトの実行を強制的に終了させた場合、スクリプトプロセスの戻り値は以下で示されるレジストリの値（デフォルトは 17）になります。

〔レジストリキー〕

```
HKEY_LOCAL_MACHINE¥SOFTWARE¥Hitachi¥JP1/  
Script¥SPTX¥ExitCode
```

〔値名〕

Terminate

〔値のデータタイプ〕

REG\_DWORD

### 補足

JP1/AJS2 から、JP1/Script プロセスを強制終了する場合、SPTHTerminate を使用しています。詳細は「2.7.2(6) JP1/AJS2 からの JP1/Script の強制終了方法」を参照してください。

### 対象バージョン

JP1/Script 05-00 以降

## 10.4 スクリプト制御インタフェースの記述例

スクリプト制御インタフェースの記述例を次に示します。

<例>

```
#include <windows.h>
#include <stdio.h>
#include <stdlib.h>
#include "sthapi.h"

BOOL TerminateScript ( LPCSTR lpszNetName, LPCSTR lpszFileName,
UINT uProcessID )
{
    HKEY          hKey;
    LPCSTR        subKey = "SOFTWARE¥¥Hitachi¥¥JP1/
Script¥¥PathName";
    char          valueName[32];
    char          value[_MAX_PATH];
    DWORD        valueSize = sizeof ( value ) ;
    HINSTANCE     hAPIInstance;
    tSPTHOpen     pSPTHOpen;
    tSPTHClose     pSPTHClose;
    tSPTHTerminate pSPTHTerminate;
    HANDLE        hScript;

    // レジストリオープン
    if ( RegOpenKeyEx( HKEY_LOCAL_MACHINE, subKey, 0,
KEY_QUERY_VALUE, &hKey ) != ERROR_SUCCESS )
    {
        // エラー処理
        return ( FALSE ) ;
    }

    // "SPTHL.DLL"のパスを取得
    strcpy ( valueName, "Path01" ) ;
    if ( RegQueryValueEx ( hKey, valueName, 0, NULL, ( LPBYTE )
value, &valueSize ) != ERROR_SUCCESS)
    {
        // エラー処理
        return ( FALSE ) ;
    }

    // "SPTHL.DLL"をロード
    strcat ( value, "¥¥SPTHL.DLL" ) ;
    hAPIInstance = LoadLibraryEx ( value, NULL,
LOAD_WITH_ALTERED_SEARCH_PATH ) ;
    if ( hAPIInstance == NULL )
    {
        // エラー処理
        return ( FALSE ) ;
    }

    // 各関数のエントリーポイントを取得
    pSPTHOpen      = ( tSPTHOpen )GetProcAddress ( hAPIInstance,
"SPTHOpen" ) ;
    pSPTHClose     = ( tSPTHClose ) GetProcAddress ( hAPIInstance,
```

## 10. スクリプト制御インタフェース

```
"SPTHClose" ) ;
    pSPTHTerminate = ( tSPTHTerminate ) GetProcAddress (
hAPIInstance, "SPTHTerminate" ) ;

    // オープン
    if ( pSPTHOpen ( lpszNetName, NULL, &hScript ) == FALSE )
    {
        FreeLibrary ( hAPIInstance ) ;
        return ( FALSE ) ;
    }

    // スクリプト実行の強制終了
    if (pSPTHTerminate ( hScript, lpszFileName, uProcessID,
SPTH_TERM_CHILD ) == FALSE )
    {
        pSPTHClose ( hScript ) ;
        FreeLibrary ( hAPIInstance ) ;
        return ( FALSE ) ;
    }

    // クローズ
    pSPTHClose ( hScript ) ;

    // APIのインスタンスハンドルを解放する
    FreeLibrary ( hAPIInstance ) ;

    return ( TRUE ) ;
}
```

# 11 スクリプト OLE コントロール

この章では、スクリプトコマンドを実行する、スクリプト OLE コントロールについて説明します。

---

11.1 スクリプト OLE コントロールとは

---

11.2 SPTO コントロール

---

## 11.1 スクリプト OLE コントロールとは

---

スクリプト OLE コントロールとは、スクリプトコマンドを実行する OLE コントロールです。本 OLE コントロール（ SPTO.OCX ）は、JP1/Script の一部としてインストールされ、Visual Basic のアプリケーションなどに組み込むことができます。

なお、スクリプト OLE コントロールは、JP1/Script 05-00 以降でサポートしています。

## 11.2 SPTO コントロール

---

### 機能

SPTO コントロールはスクリプトコマンドの実行を提供します。

### 構文

SPTO

### プロパティ

RTN  
EXECRTN  
EXECID  
EnableErrorMessage

### メソッド

SetGV  
GetGV  
DeleteGV  
Exec

### 対象バージョン

JP1/Script 05-00 以降

### 11.2.1 RTN プロパティ（メソッドのエラー詳細コードを返す）

#### 機能

メソッドのエラー詳細コードを返します。

#### 構文

*Object* . RTN

#### 指定項目

*Object*  
SPTO コントロールへの参照を表すオブジェクト式を指定します。

#### 説明

メソッドのエラー詳細コードを長整数型（long）で返します。

#### 対象バージョン

JP1/Script 05-00 以降

### 11.2.2 EXECRTN プロパティ（実行ファイルの終了コードを返す）

#### 機能

Exec メソッド（終了待ちあり）から呼び出した実行ファイルの終了コードを返します。

### 構文

*Object* . EXECRTN

### 指定項目

*Object*

SPTO コントロールへの参照を表すオブジェクト式を指定します。

### 説明

Exec メソッドから呼び出すアプリケーションの終了待ちをし、メソッドの実行結果が True の場合、アプリケーションの終了コードを長整数型（long）で格納します。終了待ちをしない場合やメソッドの実行結果が False の場合は何も格納されません。

### 対象バージョン

JP1/Script 05-00 以降

## 11.2.3 EXECID プロパティ（実行ファイルの識別子を返す）

### 機能

Exec メソッド（終了待ちなし）から呼び出した実行ファイルの識別子を返します。

### 構文

*Object* . EXECID

### 指定項目

*Object*

SPTO コントロールへの参照を表すオブジェクト式を指定します。

### 説明

Exec メソッドから呼び出すアプリケーションの終了待ちをしない場合、実行ファイルの識別子を文字列型で格納します。

### 対象バージョン

JP1/Script 05-00 以降

## 11.2.4 EnableErrorMessage プロパティ（エラーの有無を返す）

### 機能

SPTO コントロール内にエラーがあるかどうかを返します。

### 構文

*Object* . EnableErrorMessage



## 指定項目

*Object*

SPTO コントロールへの参照を表すオブジェクト式を指定します。

## 説明

SPTO コントロール内にエラーがない場合は False を、エラーがある場合は True を返します。

## 対象バージョン

JP1/Script 05-00 以降

## 11.2.5 SetGV メソッド（グローバル変数を設定する）

## 機能

グローバル変数を設定します。

## 構文

```
Object . SetGV ( GlobalName , Value )
```

## 指定項目

*Object*

SPTO コントロールへの参照を表すオブジェクト式を指定します。

*GlobalName*

設定するグローバル変数を文字列型で指定します。

*Value*

設定する値を文字列型で指定します。

## 説明

指定されたグローバル変数に指定された値を設定します。メソッドが正常に実行された場合は True を、エラーが発生した場合は False を、メソッドの実行結果として返します。

指定されたグローバル変数が存在しない場合は、新規にグローバル変数を作成し値を設定します。グローバル変数が存在する場合は、指定された値で更新します。

## 補足

このメソッドは、インストールフォルダの "DATA" フォルダ下に、グローバル変数ファイル (SPTGV.SPG) を作成します。ただし、JP1/Script 06-51 以降では、マネージャの [ ツール ] - [ オプション ( クラスタ環境 ) ] ダイアログで管理ファイルの出力先が変更されている場合、指定されたフォルダ下にグローバル変数ファイルを作成します。このファイルはスクリプトの実行が終了しても残りますので、グローバル変数を初期化する場合は、ファイルを削除してください。

## 対象バージョン

JP1/Script 05-00 以降

# 11.2.6 GetGV メソッド（グローバル変数を取得する）

機能

グローバル変数を取得します。

構文

*Object* . GetGV ( *GlobalName* )

指定項目

*Object*

SPTO コントロールへの参照を表すオブジェクト式を指定します。

*GlobalName*

取得するグローバル変数を文字列型で指定します。

説明

指定されたグローバル変数を取得し、値をメソッドの実行結果として返します。  
指定されたグローバル変数が存在しない場合は、長さ 0 の文字列 ("" ) を返します。

対象バージョン

JP1/Script 05-00 以降

# 11.2.7 DeleteGV メソッド（グローバル変数を削除する）

機能

グローバル変数を削除します。

構文

*Object* . DeleteGV ( *GlobalName* )

指定項目

*Object*

SPTO コントロールへの参照を表すオブジェクト式を指定します。

*GlobalName*

削除するグローバル変数を文字列型で指定します。  
また、この値には次の値を指定できます。

値	意味
AllGV	すべてのグローバル変数を削除します。ただし、グローバル変数ファイル (SPTGV.SPG) は残ります。

説明

指定されたグローバル変数を削除します。メソッドが正常に実行された場合は True を、エラーが発生した場合は False を、メソッドの実行結果として返します。  
指定されたグローバル変数が存在しない場合は、何もしないで常に True を返しま

す。

対象バージョン

JP1/Script 06-51 以降

## 11.2.8 Exec メソッド（実行ファイル呼び出す）

機能

複数パラメタ指定で実行ファイル（EXE ファイル，BAT ファイル，COM ファイル，SPT ファイル，CMD ファイル，関連付けられたファイル）を呼び出します。終了待ちの有無を指定できます。

構文

```
Object . Exec ( FileName , Flag , Parameter )
```

指定項目

*Object*

SPTO コントロールへの参照を表すオブジェクト式を指定します。

*FileName*

呼び出す実行ファイル名を文字列型で指定します。  
実行ファイルに指定できるファイルの種類は次のとおりです。

- 実行形式ファイル（.EXE）
- MS-DOS バッチファイル（.BAT）
- MS-DOS 実行形式ファイル（.COM）
- JP1/Script ファイル（.SPT）
- コマンドスクリプト（.CMD）
- 関連付けられたファイル

*Flag*

*FileName* で指定された実行ファイルの終了待ちをするかどうかをブール型で指定します。終了待ちをするときは True を，終了待ちをしないときは False を指定します。

*Parameter*

*FileName* で指定されたファイルの実行に必要なパラメタがある場合，パラメタを文字列型で指定します。  
複数のパラメタを指定する場合は，各パラメタを空白で区切って指定します。  
なお，パラメタに次の文字列を指定した場合，特別な意味を持ちます。

パラメタ	意味
"/SPT:HIDE"	アプリケーションのウィンドウを非表示にします。
"/SPT:MIN"	アプリケーションのウィンドウをアイコン化します。
"/SPT:MAX"	アプリケーションのウィンドウを最大表示します。

パラメタはコマンドラインパラメタとして実行ファイルに渡されますが、"/SPT:HIDE", "/SPT:MIN", および "/SPT:MAX" パラメタはコマンドラインパラメタとしては渡されません。

### 説明

指定された実行ファイルを実行します。

*Flag* に True が指定された場合は、呼び出したアプリケーションが終了するまで待機し、False が指定された場合は、アプリケーションが終了するまで待機しないで次の処理に移ります。

アプリケーションが正常終了した場合は True を、それ以外の場合は False を、メソッドの実行結果として返します。

アプリケーションの終了待ちをし、メソッドの実行結果が True の場合は、実行ファイルの終了コードを EXECRTN プロパティに長整数型 (long) で格納します。終了待ちをしない場合やメソッドの実行結果が False の場合は、EXECRTN プロパティには何も格納されません。

また、アプリケーションの終了待ちをしない場合、実行ファイルの識別子を EXECID プロパティに文字列型で格納します。

### 対象バージョン

JP1/Script 05-00 以降

# 付録

---

付録 A Script トレースのファイル出力形式

---

付録 B サンプルファイル

---

付録 C エラー詳細コード

---

付録 D 用語解説

---

## 付録 A Script トレースのファイル出力形式

Script トレースで扱う次のファイルの出力形式について説明します。

- 解析トレースファイル
- 実行トレースファイル
- ユーザトレースファイル (トレースファイル)
- サーバトレースファイル

### (1) 解析トレースファイルの出力形式

解析トレースファイルは、コマンドの解析結果を出力するファイルです。対象となるスクリプトが複数起動を許可されているかどうかによって、出力形式が異なります。

#### (a) 複数起動が許可されていないスクリプトの場合

ファイル名

ファイル名 (拡張子 .SPA) は、マネージャの [ ファイル ] - [ 実行の環境設定 ] の「トレース情報」ページで指定した、フォルダ名とスクリプトファイルのラベル名を組み合わせた名称となります。

出力形式

解析トレースファイルの出力形式を図 A-1 に示します。

図 A-1 解析トレースファイルの出力形式 (複数起動が許可されていないスクリプトの場合)

```

2002/02/07 14:23:58 << 解析結果 - SOP3F370-2000 / Administrator >>-----0671
a
  14:23:58      7.9      : Message      : 省略できない引数が省略されています。
  b          c          d          e
TOTAL ERROR      1
f
2002/02/07 14:23:58 -----
g
2002/02/07 14:25:00 << 解析結果 - SOP3F370-2000 / Guest >>-----0671

TOTAL ERROR      0

2002/02/07 14:26:01 -----

```

a. トレース出力を開始した日時、実行したコンピュータ名、実行したユーザ名、およびスクリプトエンジンのバージョンを出力します。

b. 解析時エラーとなった時間を出力します。

c. 解析時エラーとなった行の位置、列の位置を出力します。

d. 解析エラーとなったコマンドを出力します。

- e. 解析エラーの内容を出力します。
- f. 解析エラーの合計値を出力します。
- g. トレース出力を終了した日時を出力します。

#### 行数および列数の範囲

解析トレースファイルに出力するトレース情報は、行数 100 ~ 9,999、列数 128 ~ 1,024 の範囲で出力できます。

行数、および列数の出力範囲は、マネージャの [ ファイル ] - [ 実行の環境設定 ] の「トレース情報」ページで、任意に値を変更できます。

なお、出力されるトレース情報が、行数の最大範囲を超えた場合は、先頭の行に戻って、先頭の行からトレース情報が出力されます（すでに出力されているトレース情報が上書きされます）。また、列数の最大範囲を超えた場合は、範囲を超えた部分が削除されます。

#### (b) 複数起動が許可されているスクリプトの場合

##### ファイル名

ファイル名（拡張子 .SPA）は、マネージャの [ ツール ] - [ オプション ] の「複数起動」ページで指定した、フォルダ名とスクリプトファイルのラベル名を組み合わせた名称となります。

##### 出力形式

解析トレースファイルの出力形式を図 A-2 に示します。

図 A-2 解析トレースファイルの出力形式（複数起動が許可されているスクリプトの場合）

2002/02/07 14:56:14 149		S 0671 - SOP3F370-2000 / Administrator	
a			
2002/02/07 14:56:14 149	5,9	: NetExec	: 引数が足りません。
b c	d e	f	
2002/02/07 14:56:14 2244		S 0671 - SOP3F370-2000 / Guest	
2002/02/07 14:56:14 2244	5,9	: NetExec	: 引数が足りません。
2002/02/07 14:56:17 149	8,9	: Message	: 省略できない引数が省
略されています。			
2002/02/07 14:56:17 149	E	TOTAL ERROR	2
g	h		
2002/02/07 14:56:17 2244	8,9	: Message	: 省略できない引数が省
略されています。			
2002/02/07 14:56:17 2244	E	TOTAL ERROR	2

- a. 実行したプロセスのトレース出力を開始した日時、プロセス識別子、開始を意味する「S」、スクリプトエンジンのバージョン、および実行したコンピュータ名と実行したユーザ名を出力します。
- b. 解析時エラーとなった時間を出力します。
- c. 解析時エラーとなったプロセスのプロセス識別子を出力します。

- d. 解析時エラーとなった行の位置、列の位置を出力します。
- e. 解析エラーとなったコマンドを出力します。
- f. 解析エラーの内容を出力します。
- g. 実行したプロセスのトレース出力を終了した日時、プロセス識別子、および終了を意味する「E」を出力します。
- h. 解析エラーの合計値を出力します。

#### 行数および列数の範囲

解析トレースファイルに出力するトレース情報は、行数 100 ~ 9,999、列数 128 ~ 1,024 の範囲で出力できます。

行数、および列数の出力範囲は、マネージャの [ ファイル ] - [ 実行の環境設定 ] の「トレース情報」ページで、任意に値を変更できます。

なお、出力されるトレース情報が、行数の最大範囲を超えた場合は、先頭の行に戻って、先頭の行からトレース情報が出力されます（すでに出力されているトレース情報が上書きされます）。また、列数の最大範囲を超えた場合は、範囲を超えた部分が削除されます。

## （２）実行トレースファイルの出力形式

実行トレースファイルは、スクリプト実行プログラムが、コマンドの実行結果を出力するファイルです。対象となるスクリプトが複数起動を許可されているかどうかによって、出力形式が異なります。

### （a）複数起動が許可されていないスクリプトの場合

#### ファイル名

ファイル名（拡張子 .SPX）は、マネージャの [ ファイル ] - [ 実行の環境設定 ] の「トレース情報」ページで指定した、フォルダ名とスクリプトファイルのラベル名を組み合わせた名称となります。

#### 出力形式

実行トレースファイルの出力形式を図 A-3 に示します。



図 A-3 実行トレースファイルの出力形式（複数起動が許可されていないスクリプトの場合）

```

2002/02/07 14:23:58 << 実行結果 - SOP3F370-2000 / Administrator >>----- 0671
a
    * TOTAL ERROR      0
    文法エラーがありました。解析結果を参照してください。
2002/02/07 14:23:58 -----

2002/02/07 14:25:00 << 実行結果 - SOP3F370-2000 / Guest >>----- 0671
14:24:23 *      11: DeleteFile      : 0032: プロセスはファイルにアクセスできません。
                                   別のプロセスが使用中です。(d:¥Script¥job¥job004.txt)
14:24:27 *      13: MakeDir        : 指定されたディレクトリの作成に失敗しました。(A:¥
Script)
14:25:45 *      25: Exec           : 0002: システムは指定されたファイルを見つけることがで
きません。
                                   (D:¥Script¥Job¥Job004.exe)
b          c      d      e          f
          * TOTAL ERROR      3
          g
2002/02/07 14:26:01 -----
h

```

- a．トレースの出力を開始した日時，および実行したコンピュータ名，実行したユーザ名，およびスクリプトエンジンのバージョンを出力します。
- b．実行時エラーとなった時間を出力します。
- c．エラーとなったときに出力します。
- d．実行時エラーとなったコマンドの行位置を出力します。
- e．実行エラーのコマンド名を出力します。
- f．エラーの内容を出力します。
- g．実行エラーの合計値を出力します。
- h．トレース出力を終了した日時を出力します。

#### 行数および列数の範囲

実行トレースファイルに出力するトレース情報は，行数 100 ～ 9,999，列数 128 ～ 1,024 の範囲で出力できます。

行数，および列数の出力範囲は，マネージャの「ファイル」 - 「実行の環境設定」の「トレース情報」ページで，任意に値を変更できます。

なお，出力されるトレース情報が，行数の最大範囲を超えた場合は，先頭の行に戻って，先頭の行からトレース情報が出力されます（すでに出力されているトレース情報が上書きされます）。また，列数の最大範囲を超えた場合は，範囲を超えた部分が削除されます。

#### (b) 複数起動が許可されているスクリプトの場合

ファイル名

ファイル名 (拡張子 .SPX) は、マネージャの [ ツール ] - [ オプション ] の「複数起動」ページで指定した、フォルダ名とスクリプトファイルのラベル名を組み合わせた名称となります。

#### 出力形式

実行トレースファイルの出力形式を図 A-4 に示します。

図 A-4 実行トレースファイルの出力形式 (複数起動が許可されているスクリプトの場合)

2002/02/07 15:08:46 596	S	0671	-	SOP3F370-2000	/	Administrator	
a							
2002/02/07 15:08:47 596	*	5:	Message				: 0123:ファイル名、ディレクトリ名、またはボリューム ラベルの構文が間違っています。
b	c	d	e	f		g	
2002/02/07 15:08:47 2296	S	0671	-	SOP3F370-2000	/	Guest	
2002/02/07 15:08:47 2296	*	5:	Message				: 0123:ファイル名、ディレクトリ名、またはボリューム ラベルの構文が間違っています。
2002/02/07 15:08:48 596	*	7:	MakeDir				: 指定されたディレクトリの作成に失敗しました。(A:¥Script)
2002/02/07 15:08:48 2296	*	7:	MakeDir				: 指定されたディレクトリの作成に失敗しました。(A:¥Script)
2002/02/07 15:08:48 596	*	10:	Exec				: 0002:システムは指定されたファイルを見つけることができません。(D:¥Script¥job¥job004.exe)
2002/02/07 15:08:48 2296	*	10:	Exec				: 0002:システムは指定されたファイルを見つけることができません。(D:¥Script¥job¥job004.exe)
2002/02/07 15:10:39 596	*	13:	SetGV				: 0053:ネットワークパスが見つかりません。(SOP3F370-NT)
2002/02/07 15:10:39 596	E *	TOTAL ERROR				4	
h							
2002/02/07 15:10:44 2296	*	13:	SetGV				: 0053:ネットワークパスが見つかりません。(SOP3F370-NT)
2002/02/07 15:10:44 2296	E *	TOTAL ERROR				4	

a. 実行したプロセスのトレースの出力を開始した日時、プロセス識別子、開始を意味する「S」、およびスクリプトエンジンのバージョン、および実行したコンピュータ名と実行したユーザ名を出力します。

b. 実行時エラーとなった時間を出力します。

c. 実行時エラーとなったプロセスのプロセス識別子を出力します。

d. エラーとなったときに出力します。

e. 実行時エラーとなったコマンドの行位置を出力します。

f. 実行エラーのコマンド名を出力します。

g. エラーの内容を出力します。

h. 実行したプロセスのトレース出力を終了した日時、プロセス識別子、および終了を意味する「E」を出力します。

i. 実行エラーの合計値を出力します。

行数および列数の範囲

実行トレースファイルに出力するトレース情報は、行数 100 ~ 9,999、列数 128 ~ 1,024 の範囲で出力できます。

行数、および列数の出力範囲は、マネージャの [ ファイル ] - [ 実行の環境設定 ] の「トレース情報」ページで、任意に値を変更できます。

なお、出力されるトレース情報が、行数の最大範囲を超えた場合は、先頭の行に戻って、先頭の行からトレース情報が出力されます（すでに出力されているトレース情報が上書きされます）。また、列数の最大範囲を超えた場合は、範囲を超えた部分が削除されます。

### （３）ユーザトレースファイルの出力形式

ユーザトレースファイルは、Message コマンドの第一引数に Target\_File を指定して出力したファイルです。

#### （a）ファイル名

ファイル名（.TXT）は、ユーザが任意で付けられます。

#### （b）出力形式

ユーザトレースファイルの出力形式を図 A-5 に示します。

図 A-5 ユーザトレースファイルの出力形式

2002/02/24 12:00:01 開始	
2002/02/24 12:01:10 Gyoumu001 の呼出し開始	
2002/02/24 12:01:15 Gyoumu001 の呼出し完了（正常）	
<u>2002/02/24 12:03:17 スクリプト実行完了</u>	
a	b

a．トレース出力した日付と時間を出力します。

b．出力したメッセージです。

#### （c）行数および列数の範囲

実行トレースファイルに出力するトレース情報は、行数 100 ~ 9,999、列数 128 ~ 1,024 の範囲で出力できます。

行数、および列数の出力範囲は、マネージャの [ ファイル ] - [ 実行の環境設定 ] の「トレース情報」ページで、任意に値を変更できます。

なお、出力されるトレース情報が、行数の最大範囲を超えた場合は、先頭の行に戻って、先頭の行からトレース情報が出力されます（すでに出力されているトレース情報が上書きされます）。また、列数の最大範囲を超えた場合は、範囲を超えた部分が削除されます。

(4) サーバトレースファイルの出力形式

サーバトレースファイルは、クライアントから SetGV、GetGV、DeleteGV、および NetExec コマンドで呼び出されたサーバがコマンドを実行した結果を出力するファイルです。

(a) ファイル名

ファイル名は、SPTSVTRC.SPY で固定です。インストールフォルダの DATA フォルダ下に作成されます。

(b) 出力形式

サーバトレースファイルの出力形式を図 A-6 に示します。

図 A-6 サーバトレースファイルの出力形式

2002/02/19	22:19:03	22:19:03	SetGV			OK	"A65(T)"	"TAK"	"D:¥SCRYVARYN_SGV. SPT(23)"
2002/02/19	22:19:15	22:19:15	GetGV			ER(2)	"A65(T)"	"TAK"	"D:¥SCRYVARYN_GGV. SPT(21)"
						g			
2002/02/19	22:20:16	22:20:22	NetExec	S 151		OK	"A65(T)"	"TAK"	"D:¥SCRYEXEC¥N_PRM. SPT(29)"
a	b	c	d	e f		g	h	i	j
2002/02/19	22:20:16	22:20:22	NetExec	E 151		0			
	k	l		e		m			

- a. サーバでコマンドを実行した日付を出力します。
- b. サーバでコマンドの実行を開始した時間を出力します。
- c. サーバでコマンドの実行を終了した時間を出力します。
- d. サーバで実行したコマンド名を出力します。
- e. サーバで実行したコマンドがプロセスを呼び出す場合、2 行のトレースを出力します。  
この 1 行目と 2 行目を識別する記号を出力します。  
  
S : 1 行目のトレース  
  
E : 2 行目のトレース
- f. サーバで実行したコマンドがプロセスを呼び出す場合、プロセス識別子を出力します。
- g. サーバで実行したコマンドの実行結果を出力します。  
  
OK : 正常  
  
ER : 実行エラー (括弧内の数値は、エラー詳細コード)
- h. クライアントのコンピュータ名を出力します ("" で囲まれます)。

- i. クライアントのユーザ名を出力します ( "" で囲まれます )。
- j. クライアントで実行したスクリプトファイル名をフルパスで出力します ( "" で囲まれます )。括弧内の数値は、サーバで実行したコマンドのスクリプトファイル中の行番号です。
- k. サーバで実行したコマンドがプロセスを呼び出す場合、プロセスを開始した時間を出力します。
- l. サーバで実行したコマンドがプロセスを呼び出す場合、プロセスを終了した時間を出力します。
- m. サーバで実行したコマンドがプロセスを呼び出す場合、プロセスの終了コードを出力します ( サーバで実行したコマンドの実行結果が OK のときだけ出力されます )。

(c) 行数および列数の範囲

サーバトレースファイルに出力する実行結果は、行数 1 ~ 1,000、列数 200 で、行数、および列数の値は変更できません。なお、出力される実行結果が行数の最大値を超えた場合は、先頭の行に戻って、先頭の行から実行結果が出力されます (すでに出力されている実行結果が上書きされます)。また、列数の最大値を超えた場合は、改行されます。

(d) 出力指定

ユーザは、マネージャの [ オプションダイアログ ] の [ サーバ情報 ] ページで、サーバトレースファイルの出力の有無を指定できます。

## (5) NetExec エラーログファイルの出力形式

NetExec エラーログファイルは、NetExec コマンド実行時に発生したエラー内容を出力します。

(a) ファイル名

- クライアント側

フォルダ名：レジストリ「Net\_Trace\_Directory」に指定されたフォルダ下に

「Log¥STXNetExec\_Client」<sup>1</sup> というフォルダを作成し、そのフォルダ下。

ファイル名：「STXNetExec\_C\_xx.LOG」

xx は 01 ~ 30 までの数値。NetExec コマンド実行制御を行う SPTXNetx.exe の 1 回のプロセス実行中にエラーが発生すると、一つのトレースファイルを出力します。

- サーバ側

フォルダ名：レジストリ「Net\_Trace\_Directory」に指定されたフォルダ下に

「Log¥STXNetExec\_Server」<sup>2</sup> というフォルダを作成し、そのフォルダ下。

ファイル名：「STXNetExec\_S\_xx.LOG」

xx は 01 ~ 30 までの数値。NetExec コマンド実行制御を行う SPTXNetx.exe の 1 回のプロセス実行中にエラーが発生すると一つのトレースファイルを出力します。

注 1 JP1/Script 07-00 以降では「STXNetExec\_Client」というフォルダ名になり

ます。

注 2 JP1/Script 07-00 以降では「STXNetExec\_Server」というフォルダ名になります。

#### (b) 出力形式

クライアント側エラーログの出力形式を図 A-7 に、サーバ側エラーログの出力形式を図 A-8 に示します。

図 A-7 クライアント側エラーログの出力形式

```
11:44:57:659 2003-02-07 *** CallCompName=SOP3F370-2000, CallUserName=WADA
11:44:57:659 2003-02-07 CallFileName=D:\TEST_005.SPT, CallFileLine=0
11:44:57:659 2003-02-07 * NetExec ->CompName=10.210.104.103, FileName=D:\TEST\MgBox.spt
11:44:57:659 2003-02-07 Flag=1, ExecDirName=
ExecPlace=0, Option=1325
11:44:57:574 2003-02-07 <<< CxNamePipeClient::Connect Error End 2 (RC=2)
11:44:57:850 2003-02-07 <<< CxNamePipeBasic::SendData Error End 4 (RC=1431698578)
11:44:57:850 2003-02-07 <<< STXDNetworkProcessClient::CheckGetResultNetExec Error End 1
(RC=1431698578)
11:44:57:850 2003-02-07 <<< STXDNetExecApp::Exitinstance End (ExitCode=-1431698578)
```

図 A-8 サーバ側エラーログの出力形式

```
0035653767 0.000 19:02:15:087 2003-02-07 >>> CxNamedPipeServer::Create -> PipeName=JP1
Script-NetworkExec-ByLogon, TimeOut=60000
0035653767 0.000 19:02:15:187 2003-02-07 <<< STXDNetworkProcessServer::OnRequestDoNet
Exec Error End 1
```

#### (c) ファイル容量

エラーログのファイル容量は、クライアント側、サーバ側のそれぞれで、次のとおりです。

1 ファイルのファイル容量上限値 : ( 100 バイト / 1 行 ) \* 100 行 = 約 10KB ( キロバイト )

ファイル数の上限値 : 30 ファイル

合計の上限値 : 約 10KB ( キロバイト ) \* 30 ファイル = 約 300KB ( キロバイト )

## (d) 出力指定

インストーラは次のレジストリを設定し、デフォルトでエラーログを出力します。

## 〔レジストリキー〕

HKEY\_LOCAL\_MACHINE¥SOFTWARE¥Hitachi¥SPTX¥Option

## 〔値名 1〕

Net\_Trace (DWORD)

## 〔値 1〕

トレースの出力の有無，および出力レベル

0：トレース出力しない。

1：エラートレースだけ出力し，正常終了時はファイルを削除し残さない（デフォルト）

## 〔値名 2〕

Net\_Trace\_Directory (文字側)

## 〔値 2〕

トレース出力先フォルダ名（デフォルトは，インストールログフォルダ下）

---

## 付録 B サンプルファイル

JP1/Script では、次の 2 種類のサンプルファイルを提供しています。

- ・メニュー画面を表示し、選択した番号に応じた処理をするサンプルファイル
- ・プログラムを呼び出して、終了コードをチェックするサンプルファイル

ここでは、これらのサンプルファイルの機能概要、ファイル構成、および実行手順について説明します。なお、サンプルファイルの格納先については、「2.1.1 プログラムのインストール先のフォルダ」を参照してください。

### (1) メニュー画面を表示し、選択した番号に応じた処理をするサンプルファイル

#### (a) 機能概要

サンプルファイルを起動するとメニュー画面が表示されます。メニュー画面の番号を選択すると、次の操作ができます。

##### 処理番号 1

エディタを起動します。エディタが終了するまで待ちます。

##### 処理番号 2

新規にフォルダを作成し、そのフォルダに、メニュー画面で指定したスクリプトファイル名を複写します。複写するファイル名は「bkupSpt.spt」です。

##### 処理番号 3

スクリプトファイルのあるフォルダ下のファイル一覧をユーザトレースファイルへ出力します。出力するファイル名は「SmplOut.TXT」です。

##### 処理番号 9

サンプルスクリプトを終了します。

#### (b) ファイル構成

SmplMain.Spt : このサンプルファイルのメインスクリプトファイル

Smpl001.Spt : 処理番号 2 に対応しているスクリプトファイル

Smpl002.Spt : 処理番号 3 に対応しているスクリプトファイル

Sample.Spt : エディタ起動時のスクリプトファイル

#### (c) サンプルファイルの実行手順

1. サンプルファイル「SmplMain.Spt」を起動する  
処理メニューが表示されます。
2. メニューに従って、実行したい処理番号を入力して [ OK ] ボタンをクリックする



([ キャンセル ] ボタンをクリックするとサンプルプログラムが終了します)

- ・ 処理番号「1」を入力すると、エディタが起動されます。
- ・ 処理番号「2」, および「3」を入力すると、別のスクリプトが起動されて処理が行われます。

それぞれの処理が正常に完了すると、再度メニュー画面が表示されます。

それぞれの処理でエラーが発生した場合、ピープ音を鳴らすコマンド、警告コマンド、およびメッセージボックスを表示するコマンドが発行されて、処理が終了します。

## (2) プログラムを呼び出して、終了コードをチェックするサンプルファイル

### (a) 機能概要

プログラムを呼び出して、終了コードをチェックします。呼び出したプログラムの終了コードは変更できます。

なお、このサンプルファイルを実行する前には、次の情報を変更してください。

#### [ レジストリキー ]

HKEY\_CURRENT\_USER¥SOFTWARE¥HITACHI¥JP1/Script¥Sample¥ExitCode

#### [ 値 ]

Mode(DWORD)

対話モードで起動する場合は「1」、バッチモードで起動する場合は「1 以外の値」を指定する。

EndCode(DWORD)

終了コードを指定する。

SleepTime(DWORD)

バッチモードの場合に起動している時間(単位: ms)を指定する。

### (b) ファイル構成

ExitCode.Spt: このサンプルファイルのメインスクリプトファイル

ExitCode.Exe: 呼び出し用プログラム

### (c) サンプルファイルの実行手順

1. マネージャから ExitCode.SPT を起動する、または ExitCode.SPT を直接起動する  
どちらの方法で起動する場合も、パラメタとして次のうちどちらかの値を指定します。  
B: バッチモードで起動する  
T: 対話モードで起動する  
パラメタの指定がないと、スクリプトがエラーとなります。
2. 対話モードで起動した場合、画面が表示されるので画面の内容にしたがってデータを

入力する

バッチモードで起動した場合は、画面が表示されないで、設定したレジストリの内容で動作します。

## 付録 C エラー詳細コード

### (1) JP1/Script が設定するエラーコード

NetExec コマンド実行後，「RTN」に設定されるリターンコードの意味を次の表に示します。

エラーコード	エラーの内容
-1431698655	サーバからの応答待ちでタイムアウトエラー発生。
-1431698656	サーバが接続要求に対して応答しない。
-1431698579	サーバからデータ受信中にエラー発生。
-1431698578	サーバとの送受信中の無通信時間が許容時間を超えた。

### (2) OS のエラーコード

Windows2000 で設定されるエラーコードを，ご参考までに次の表に示します。ただし，OS の種類により異なる場合もありますので，この内容は必ず保証されるものではありません。

エラーコード	エラーの内容
-004	出力ファイル用のメモリ空間が足りません。
-006	グローバルハンドルが不正です。
0001	ファンクションが間違っています。
0002	システムは指定されたファイルを見つけることができません。
0003	システムは指定されたネットワーク名またはディレクトリパスを見つけることができません。
0004	ファイルを開くことができません。
0005	アクセスは拒否されました。ファイルの属性またはセキュリティを見直してください。
0006	ハンドルが無効です。
0007	記憶域制御ブロックが壊れています。
0008	このコマンドを実行するのに十分な記憶域がありません。
0009	記憶域制御ブロックのアドレスが無効です。
0010	環境が間違っています。
0011	間違ったフォーマットのプログラムを読み込もうとしました。
0012	アクセスコードが無効です。
0013	データが無効です。
0014	この操作を完了するのに十分な記憶域がありません。
0015	指定されたドライブが見つかりません。

エラーコード	エラーの内容
0016	ディレクトリを削除できません。
0017	ファイルを別のディスクドライブに移動できません。
0018	これ以上ファイルがありません。
0019	このメディアは書き込み禁止になっています。
0020	指定されたデバイスが見つかりません。
0021	デバイスの準備ができていません。
0022	デバイスがコマンドを認識できません。
0023	データエラー（巡回冗長検査 (CRC) エラー）です。
0024	プログラムはコマンドを発行しましたが、コマンドの長さが間違っています。
0025	指定されたディスクの領域またはトラックが見つかりません。
0026	指定されたディスクまたはフロッピーディスクにアクセスできません。
0027	要求されたセクタが見つかりません。
0029	指定されたデバイスに書き込めません。
0030	指定されたデバイスから読み取れません。
0031	システムに接続されたデバイスが機能していません。
0032	プロセスはファイルにアクセスできません。別のプロセスが使用中です。
0033	プロセスはファイルにアクセスできません。別のプロセスがファイルの一部をロックしています。
0036	開かれている共有ファイルが多過ぎます。
0038	ファイルの終わりに達しました。
0039	ディスクがいっぱいです。
0050	ネットワーク要求はサポートされていません。
0051	リモートコンピュータは利用できません。
0052	ネットワーク上に同じ名前があります。
0053	ネットワークパスが見つかりません。
0054	ネットワークがビジーです。
0055	指定されたネットワークリソースまたはデバイスは利用できません。
0056	ネットワーク BIOS コマンドが制限値に達しました。
0057	ネットワークアダプタのハードウェアエラーが発生しました。
0058	指定されたサーバは、要求された操作を実行できません。
0059	予期しないネットワークエラーが発生しました。
0060	リモートアダプタは互換性がありません。
0064	指定されたネットワーク名は利用できません。
0065	ネットワークアクセスは拒否されました。
0066	ネットワークリソースの種類が正しくありません。

エラーコード	エラーの内容
0067	ネットワーク名が見つかりません。
0068	ローカルコンピュータのネットワークアダプタカードに対する名前数が制限値を超えました。
0069	ネットワーク BIOS セッションの数が制限値を超えました。
0070	リモートサーバは一時停止されているか、起動途中です。
0071	コンピュータへの接続数が最大値に達しているため、これ以上このリモートコンピュータに接続できません。
0072	指定されたプリンタまたはディスクデバイスは一時停止されています。
0080	ファイルがあります。
0082	ディレクトリまたはファイルを作成できません。
0084	この要求を処理するための記憶域がありません。
0085	ローカル デバイス名は既に使用されています。
0086	指定されたネットワーク パスワードが間違っています。
0087	パラメータが間違っています。
0088	ネットワークで書き込みエラーが発生しました。
0089	現在、ほかのプロセスを起動できません。
0110	指定されたデバイスまたはファイルを開けません。
0111	ファイル名が長過ぎます。
0112	ディスクに十分な空き領域がありません。
0123	ファイル名、ディレクトリ名、またはボリュームラベルの構文が間違っています。
0144	このディレクトリはルートディレクトリのサブディレクトリではありません。
0145	ディレクトリが空ではありません。
0148	指定されたパスは現在使用できません。
0161	指定されたパスは無効です。
0164	このシステムでは、これ以上のスレッドを作成できません。
0167	ファイルの領域をロックできません。
0170	要求されたリソースは使用中です。
0183	すでに存在するファイルを作成することはできません。
0196	このオペレーティングシステムではこのアプリケーションプログラムは実行されません。
0197	オペレーティングシステムは現在このアプリケーションを実行するように構成されていません。
0199	このオペレーティングシステムではこのアプリケーションプログラムは実行されません。
0206	ファイル名または拡張子が長過ぎます。
0230	パイプの状態が無効です。

エラーコード	エラーの内容
0231	すべてのパイプインスタンスがビジーです。
0232	パイプを閉じています。
0240	セッションは取り消されました。
0267	ディレクトリ名が無効です。
1053	そのサービスは指定時間内に開始要求または制御要求に応答しませんでした。
1054	サービスに対してスレッドを作成できませんでした。
1056	サービスインスタンスはすでに実行されています。
1057	アカウント名が無効であるか、または存在しません。あるいは、指定したアカウント名のパスワードが無効です。
1058	指定されたサービスは無効であるか、または有効なデバイスが関連づけられていないため、開始できません。
1060	指定されたサービスはインストールされたサービスとして存在しません。
1062	そのサービスを開始できませんでした。
1069	ログオンに失敗したため、サービスを開始できませんでした。
1070	サービスは開始後に開始待ち状態でハングしました。
1073	指定されたサービスはすでに開始されています。
1078	その名前は、サービス名またはサービス表示名としてすでに使われています。
1114	ダイナミックリンクライブラリ (DLL) 初期化ルーチンの実行に失敗しました。
1115	システムシャットダウンが実行中です。
1117	I/O デバイスエラーが発生したため、要求を実行できませんでした。
1123	フロッピーディスクのセクタ ID フィールドとフロッピーディスクコントローラトラックアドレスが一致しません。
1124	フロッピーディスクドライバが認識しないエラーがフロッピーディスクコントローラから報告されました。
1125	フロッピーディスクコントローラは矛盾する結果をレジスタに返しました。
1130	このコマンドを処理するのに必要な記憶域をサーバで確保できません。
1131	デッドロック発生の可能性が検出されました。
1176	置換するファイルを置換されるファイルに移動できません。置換されるファイルの名前は、元のままです。
1177	置換するファイルを置換されるファイルに移動できません。置換されるファイルの名前は、バックアップ名に変更されています。
1326	ログオン失敗：ユーザー名を認識できないか、またはパスワードが間違っています。
1327	ログオン失敗：ユーザー アカウントの制限。
1328	ログオン失敗：アカウントのログオン時間の制約違反。
1329	ログオン失敗：ユーザはこのコンピュータへのログオンを許可されていません。

エラーコード	エラーの内容
1330	ログオン失敗：指定されたアカウント，パスワードの有効期間が切れています。
1331	ログオン失敗：アカウントは現在無効に設定されています。
1380	ログオン失敗：要求された種類のログオンは，このコンピュータではユーザーに許可されていません。
1385	ログオン失敗：要求された種類のログオンは，このコンピュータではユーザーに許可されていません。
1450	システムリソースが不足しているため，要求されたサービスを完了できません。
1451	システムリソースが不足しているため，要求されたサービスを完了できません。
1452	システムリソースが不足しているため，要求されたサービスを完了できません。
7006	同じ名前のサービス名が，すでにシステムにあります。

コマンド実行後に設定されたリターンコードのエラー内容は，GetErrorMessage コマンドを使用して取得できます。次に例を示します。

(例)

```
Dim ErrMsg
Dim ErrCode
ErrCode=21
ErrMsg=GetErrorMessage(ErrCode)
Message( Target_File, _BIN_+"ErrMsg.txt", ErrMsg )
```

---

## 付録 D 用語解説

### ( 英字 )

---

#### JP1/Automatic Job Management System2 ( JP1/AJS2 )

JP1 関連製品の一つです。JP1/AJS2 は、JP1 Version6 での製品体系変更で、従来の JP1/AJS や JP1/NetBatch などが統合されたプログラムです。JP1/Script は、JP1/AJS2 と連携することで、複数の PC 間での分散処理が実現できます。

#### JP1/Automatic Job Scheduler ( JP1/AJS )

JP1 関連製品の一つです。JP1/AJS は、PC 上での複雑な業務運転の自動化・省力化を図るため、ジョブの自動運転を行うなどの機能を提供するプログラムです。JP1/Script は、JP1/AJS と連携することで、複数の PC 間での分散処理が実現できます。

#### JP1/Base

JP1 関連製品の一つです。JP1/Base は、JP1 Version6 での製品体系変更で、従来の JP1/IM のイベントサービスが移行されたプログラムです。JP1/Script は、コマンドの実行結果などを JP1/Base にイベント発行することができます。製品中に「JP1/IM」と表記している場合、JP1/Base のことも指しています。

#### JP1/Integrated Manager ( JP1/IM )

JP1 関連製品の一つです。JP1/IM は、分散システムの統合管理を行うためのプログラムです。JP1/Script は、コマンドの実行結果などを JP1/IM のイベントコンソールへ表示できます。JP1/IM のイベントサービスは、JP1 Version6 での製品体系変更で、JP1/Base へ移行されました。製品中に「JP1/IM」と表記している場合、JP1/Base のことも指しています。

#### JP1/Network Batch Queuing System ( JP1/NetBatch )

JP1 関連製品の一つです。JP1/NetBatch は、PC 上でバッチジョブを効率良く運用するためのプログラムです。JP1/Script は、JP1/NetBatch と連携することで、複数の PC 間での分散処理が実現できます。

#### Script エンジン

JP1/Script を構成しているプログラムの一つです。スクリプトエンジンによって、作成したスクリプトファイルのコマンドの文法チェック（字句解析および構文解析）が行われます。

#### Script 実行

JP1/Script を構成しているプログラムの一つです。スクリプトファイルの実行を行います。また、JP1/Script の各コマンドに対する処理も行います。

### ( ア行 )

---

#### 位置変数

コマンドラインからスクリプトを起動するときに指定するスクリプトファイル名、およびパラメタ



を格納する変数です。この位置変数は、「%0,%1,%2,%n,(nには数字が入る)…」のように"%"と正の整数で構成され、コマンドラインに指定したスクリプトファイル名、およびパラメタが,%0から順番に格納されます。

コマンドラインに「C:\¥TEST.SPT ABC 123」と指定した場合、位置変数%0,%1,%2には次の値が代入されます。

%0=C:\¥TEST.SPT

%1=ABC

%2=123

このように位置変数を利用すると、スクリプトの起動時に変数の値を設定することができます。

## インデックス番号

配列変数の要素（二次元配列の場合は、行要素および列要素）を表すための1から始まる数値のことです。

## エディタ

JP1/Script 付属のエディタです。簡易入力機能と連携できるので、効率良くスクリプトを作成できます。また、スクリプトの動作を見ながら実行できるモニタリング機能を使えば、デバッグ時の負荷も軽減できます。

## (カ行)

---

### 解析トレースファイル

スクリプトの構文解析結果が保存されているファイルです。

### 簡易入力

JP1/Script を構成しているプログラムの一つです。コマンドを対話形式で入力でき、入力したコマンドはエディタ、またはクリップボードに出力できます。

### キュー

実行するジョブを登録するところです。ジョブをキューに投入すると、一時的にキューに登録されます。キューに登録されたジョブは、キューから順次取り出されて実行されます。

### グローバル変数ファイル

スクリプトの実行時に設定されたグローバル変数が保存されているファイルです。

### コントロール

メニューフォーム上に配置するボタンやリストボックスなどの総称です。

## (サ行)

---

### サーバ環境ファイル

サーバ環境が設定されているファイルです。

### サーバトレースファイル

クライアントにあるスクリプト文のコマンドから呼び出されたサーバのコマンド実行結果が保存されているファイルです。

## サブルーチン

「プロシージャ」と同意語です。

## 実行環境ファイル

スクリプトファイルを実行するときの環境設定が保存されているファイルです。

## 実行環境ファイルコンバータ

Windows 上の JP1/Script の実行環境ファイルを UNIX 上の JP1/Script へ移行するためのツールです。

## 実行トレースファイル

スクリプトの実行結果が保存されているファイルです。

## 自動起動情報ファイル

ランチャがスクリプトの実行を自動的に行えるようにするための、必要な情報が保存されているファイルです。

## スクリプト（スクリプト文）

用意されたコマンドを使用して、命令文などを記述したものを指しています。

## スクリプトファイル

作成したスクリプトを保存したファイルです。

## （タ行）

---

### トレース管理ファイル

トレースファイルを管理するファイルです。

### トレースビューア

JP1/Script を構成しているプログラムの一つです。トレースビューアを起動すると、スクリプトファイルの実行時に出力されるトレース情報、スクリプトファイルの実行状態を見ることができます。

## （ハ行）

---

### バッチジョブ

バッチ処理で実現するジョブのことをいいます。

### バッチ処理

ジョブを一括して実行する処理のことです。対話操作を必要としない業務や、多量のデータを処理する業務に適しています。

### フィールド名

各コントロールに対して設定する名称です。作成されたメニューフォーム上には表示されず、メニュー情報ファイルに記述されます。

## プロシージャ

実行時に一つの単位として処理されるコマンドの集まりのことをいいます。JP1/Script では、Function ステートメントと Sub ステートメントで定義された一連の処理がこれに当たります。「サブルーチン」と同意語です。

## プロセスビューア

JP1/Script を構成しているプログラムの一つです。現在実行されているスクリプトプロセスを一覧表示し、プロセスの監視や制御ができます。他のコンピュータで実行されているスクリプトプロセスも一覧表示できます。

## 編集モード

エディタの状態の一つで、スクリプトファイルを作成している状態のことを指します。エディタ起動時には編集モードが設定されています。[ モニタリング ] - [ モニタリングの実行 ] コマンドを選択するとモニタリングモードになり、[ モニタリング ] - [ モニタリングの中止 ] コマンドを選択すると編集モードに戻ります。

## (マ行)

---

## マネージャ

JP1/Script を構成しているプログラムの一つです。スクリプトファイルの作成、編集、および動作環境の設定などの操作をします。

## メニューエディタ

JP1/Script を使用して作成したスクリプトファイルのメニューフォームを作成したり編集したりする場合に、メニューエディタを利用します。また、メニューエディタには、コントロールのプロパティの設定、およびメニューフォームのテスト表示機能もあります。

## メニュー情報ファイル

メニューフォームを構成するフォーム、および各コントロールのプロパティ定義が記述されたファイルです。メニューエディタでメニューフォームを編集すると、メニュー情報ファイルは自動的に作成され、テキスト形式で記述されます。

## メニューフォーム

一般に「ウィンドウ」と呼ばれている GUI と同じ働きをします。メニューエディタを利用して、目的に応じた画面が作成できます。メニューフォームの上にはボタンやリストボックスなどを配置できます。

## モニタリング情報ファイル

モニタリングに必要な情報を格納したファイルです。

## モニタリングモード

エディタの状態の一つで、作成したスクリプトファイルをモニタリング実行している状態を指します。エディタ起動時には編集モードが設定されています。[ モニタリング ] - [ モニタリングの実行 ] コマンドを選択するとモニタリングモードになり、[ モニタリング ] - [ モニタリングの中止 ] コマンドを選択すると編集モードに戻ります。

## ( ヤ行 )

---

### ユーザトレースファイル ( トレースファイル )

スクリプトのコマンドによって出力されたトレースを保存するファイルです。

### 予約語ルールファイル

スクリプトファイルを解析・実行するために必要となる予約語に関する規則を格納したファイルです。

## ( ラ行 )

---

### ラージファイル

容量が 2,147,483,648 バイト以上のファイルです。

### ランチャ

JP1/Script を構成しているプログラムの一つです。あらかじめ設定されている自動起動の情報に従って、スクリプトファイルの起動、および終了を制御します。

## ( ワ行 )

---

### ワークファイル

JP1/Script 内部で利用するワークファイルです。

### ワイルドカード

ワイルドカードは \* と ? で記述します。\* は任意の文字列を、? は任意の 1 文字を表します。

JP1/Script では、ワイルドカードをフォルダ名、およびファイル名中のどこに記述してもかまいません。ただし、複数フォルダにわたって指定する場合は、ワイルドカードを指定文字列の末尾に記述してください。

( 例 ) フォルダ "\_TEMP\_" の下のファイル "ABCDE.TXT" が該当する記述の例

```
_TEMP_+ "*"
_TEMP_+ "*.*"
_TEMP_+ "ABC*"
_TEMP_+ "*TXT"
_TEMP_+ "*E"
```

# 索引

## 記号

' 571

\*= 演算子 (乗算) 527

\* 演算子 (乗算) 527

+= 演算子 (加算) 523

+ 演算子 (加算) 522

-= 演算子 (減算) 525

- 演算子 (減算・マイナス符号) 524

.SPA 8

.SPB 9

.SPD 8

.SPG 9

.SPH 7

.SPN 8

.SPR 9

.SPS 8

.SPT 7

.SPU 9

.SPV 7

.SPX 8

.SPY 8

.TMP 9

.TXT 8

/= 演算子 (除算) 529

/NOEVLOG (または /noevlog) 391

/SPALV (n) (または /spalv (n)) 391

/SPXLV (n) (または /spxlv (n)) 391

/ 演算子 (除算) 528

= 396

¥= 演算子 (整数除算) 530

¥ 演算子 (整数除算) 530

^= 演算子 (べき乗) 532

^ 演算子 (べき乗) 531

\_ALLRIGHT\_ 374

\_ARGV\_ 375

\_ARGV\_CNT\_ 375

\_BIN\_ 374

\_COMP\_ 374

\_COPY\_CNT\_ 375

\_COPY\_RTN\_ 375

\_COPY\_SKIP\_CNT\_ 375

\_COPY\_SKIP2\_CNT\_ 375

\_DLL\_RTN\_ 375

\_DOMAIN\_ 374

\_ERR\_ACCESS\_ 376

\_ERR\_EOF\_ 376

\_ERR\_EXCLUSIVE\_ 376

\_ERR\_FILE\_ 376

\_ERR\_FILE\_POSITION\_ 376

\_ERR\_FILE\_SIZE\_ 376

\_ERR\_NOT\_LARGE\_FILE\_ 376

\_ERR\_PATH\_ 376

\_ERR\_PROTECT\_ 376

\_ERR\_READY\_ 376

\_ERR\_SERVICE\_NOT\_BEGIN\_ 23,376

\_ERR\_SVR\_CONNECT\_ 376

\_ERR\_SVR\_NODATA\_ 376

\_ERR\_SVR\_RECEIVEDATA\_ 376

\_ERR\_SVR\_TIMEOUT\_ 376

\_ERR\_TIMEOUT\_ 376

\_EXEC\_ID\_ 375

\_EXEC\_RTN\_ 375

\_FORM\_FIELD\_NAME\_ 375

\_FORM\_MODIFY\_KEY\_ 375

\_FORM\_TERM\_CMDNO\_ 375

\_FORM\_TERM\_KEY\_ 375

\_JOB\_RTN\_ 375

\_MSG\_RTN\_ 375

\_NL\_ 376

\_NO\_ERR\_ 376

\_OS\_ 374

\_OS\_PLATFORM\_ 374

\_OS\_REVISION\_ 374

\_OS\_VERSION\_ 374

\_PROC\_ID\_ 375

\_RTN\_ 375

\_RTNxx\_ 375

\_SCF\_ 374

\_SCF\_EXT\_ 374

\_SCF\_FIL\_ 374

\_SDF\_EXT\_ 374

\_SNF\_EXT\_ 374  
\_SVC\_RTN\_ 375  
\_SVF\_EXT\_ 374  
\_SYS\_ 374  
\_TAB\_ 376  
\_TEMP\_ 374  
\_USER\_ 374  
\_WIN\_ 374  
\_WINSYS\_ 374

## 数字

---

1 行の最大文字数やインデントの桁数を指定する 141  
64 ビットバージョン Windows Server 2003 (IPF) 版 JP1/Script (P-2812-3F84) サービスの登録 39  
64 ビットバージョン Windows Server 2003 (IPF) 版 JP1/Script (P-2812-3F84) のインストール 38

## A

---

AddStr 444  
Alert 572  
AlreadyRun 384  
And (論理積) 534

## B

---

Beep 573  
BitmapHide 586  
BitmapShow 585

## C

---

CalcDate 458  
CalcTime 462  
Call 407  
CallDll 606  
CallSpt 556  
CancelStartUp 561  
CatFiles 493  
CheckDirName 545  
CheckDriveType 546

CompDate 459  
CompTime 463  
Continue 410  
Copy 502

## D

---

Date 453  
Day 454  
DeleteDir 478  
DeleteFile 479  
DeleteGroup 609  
DeleteGV 429  
DeleteGV メソッド 634  
DeleteShortcut 611  
Dim 422  
Dim (配列) 422  
DLL ファイルを呼び出す 606  
Do...Loop 397

## E

---

EnableErrorMessage プロパティ 632  
EntryStartUp 559  
Error 384  
ExAbortError 384  
Exec 547  
EXECID プロパティ 632  
Exec, NetExec, および CaalSpt コマンドの違い 421  
EXECRTN プロパティ 631  
Exec コマンドや NetExec コマンドからパラメタを指定してスクリプトファイルを呼び出す場合 389  
Exec メソッド 635  
Exit 574  
ExitWindows 617  
Exit xx 408

## F

---

For...End For 400  
For...Next 398  
Format 448  
Function 405

## G

---

GetArrayCount 430  
 GetDateCount 461  
 GetDiskFreeSpace 502  
 GetEnvironment または GetEnv 425  
 GetErrorMessage 575  
 GetExecStatus 555  
 GetFileAttr 485  
 GetFileAttribute 485  
 GetFileSize 489  
 GetFileTime 487  
 GetGV 428  
 GetGV メソッド 634  
 GetPath 499  
 GetProcessCount 613  
 GetProcessInfo 614  
 GetServiceName 604  
 GetTextPosition 476  
 GetTimeCount 464  
 GetVerInfo 490  
 GetVersionInfo 490  
 GetVolLabel 501  
 GetVolumeLabel 501  
 GoTo 409  
 GrammarError 384

## H

---

Hour 456

## I

---

If...Then...Else 401  
 IMEventMessage 515  
 InArray 433  
 IniRead 467  
 IniWrite 468  
 InputBox 506  
 InStr 432  
 IsDef 538  
 IsDefine 538  
 IsEmpty 538  
 IsEmptyDir 539  
 IsEmptyGroup 589

IsEmptyReg 587  
 IsExistDir 540  
 IsExistFile 541  
 IsExistRegKey 588  
 IsExistService 589  
 IsFileAttr 543  
 IsFileAttribute 543  
 IsLeapYear 465  
 IsLower 450  
 IsMultiChar 452  
 IsNew 544  
 IsNumeric 539  
 IsSingleChar 451  
 IsUpper 450  
 IsWriteableDir 542

## J

---

JOBCancel 569  
 JOBHold 567  
 JOBSubmit 562  
 JOBWait 564  
 JP1/AJS2 からの JP1/Script の強制終了方法 52  
 JP1/AJS2 と連携したシステム構成 13  
 JP1/Base と連携したシステム構成 14  
 JP1/IM-Console と連携したシステム構成 15  
 JP1/IM または JP1/Base にイベントを発行する 515  
 JP1/Script が設定するエラーコード 651  
 JP1/Script サービス 5  
 JP1/Script で扱うファイル 7  
 JP1/Script で扱うファイルのバックアップとリカバリー 367  
 JP1/Script のアンインストール 40  
 JP1/Script のイベントログ 384  
 JP1/Script のインストール/アンインストール 32  
 JP1/Script の概要 1  
 JP1/Script の規則 369  
 JP1/Script の起動と終了 49  
 JP1/Script のシステム構成 13  
 JP1/Script の終了コード 384  
 JP1/Script の全体構成 3

JP1/Script の操作 55  
JP1/Script の操作の流れ 19  
JP1/Script の対象バージョンを指定する 140  
JP1/Script の特長 2  
JP1/Script を起動する 49  
JP1/Script を構成するプログラムの種類 5  
JP1/Script を終了する 51  
JP1/Script を利用するための準備 31  
JP1 呼び出しコマンド 562

## L

---

LCase 435  
Left 436  
Len 434  
LTrim 439

## M

---

MakeDir 477  
MakeGroup 608  
MakePath 498  
MakeShortcut 609  
Menu 519  
Message 507  
MessageBox 511  
MessageEventLog 514  
Mid 437  
Minute 457  
Mod= 演算子 ( 剰余演算 ) 526  
Mod 演算子 ( 剰余演算 ) 525  
Month 454

## N

---

NetExec 549  
NetExec エラーログファイル 360  
NetExec エラーログファイルの出力形式 645  
NetExec コマンド実行制御 5  
Not ( 論理否定 ) 536

## O

---

On Error 411  
Or ( 論理和 ) 535  
OS のエラーコード 651

## R

---

RegDelete 583  
RegDeleteKey 583  
RegRead 580  
RegWrite 581  
Rem 571  
Rename 480  
ResetStandardFile 496  
ResetStdFile 496  
Right 438  
RTN プロパティ 631  
RTrim 440

## S

---

Script エディタウィンドウ 93  
Script エディタウィンドウでのマウスとキーの操作 100  
Script エディタウィンドウとメニュー 93  
Script エディタウィンドウのダイアログ 262  
Script エディタウィンドウのメニュー 96  
Script トレースのファイル出力形式 638  
Script トレースビューアウィンドウ 143  
Script トレースビューアウィンドウでの操作 148  
Script トレースビューアウィンドウでのマウスとキーの操作 147  
Script トレースビューアウィンドウとメニュー 143  
Script トレースビューアウィンドウのダイアログ 274  
Script トレースビューアウィンドウのメニュー 146  
Script トレース表示ウィンドウ 159  
Script トレース表示ウィンドウでのマウスとキーの操作 162  
Script トレース表示ウィンドウのメニュー 161



Script マネージャウィンドウ 56  
Script マネージャウィンドウでのマウスと  
キーの操作 61  
Script マネージャウィンドウとメニュー 56  
Script マネージャウィンドウのダイアログ  
222  
Script マネージャウィンドウのメニュー 59  
Script メニューエディタウィンドウ 164  
Script メニューエディタウィンドウでのマウ  
スとキーの操作 171  
Script メニューエディタウィンドウとメ  
ニュー 164  
Script メニューエディタウィンドウのダイア  
ログ 278  
Script メニューエディタウィンドウのメ  
ニュー 168  
Script ランチャ 5  
Second 458  
Select Case 403  
SeparateStr 446  
SeparateStrCount 445  
ServiceChange 601  
ServiceContinue 600  
ServiceControl 603  
ServiceCreate 596  
ServiceDelete 597  
ServiceGetValue 594  
ServicePause 599  
ServiceQuery 602  
ServiceRefer 602  
ServiceSetValue 591  
ServiceStart 598  
ServiceStop 599  
SetEnvironment または SetEnv 423  
SetFileAttr 483  
SetFileAttribute 483  
SetFileTime 486  
SetGV 426  
SetGV メソッド 633  
SetPath 499  
SetStandardFile 494  
SetStdFile 494  
SetVolLabel 500

SetVolumeLabel 500  
Sleep 572  
Space 438  
SplitFile 491  
SplitPath 497  
SPTHClose 624  
SPTHGetErrorMessage 624  
SPTH.L.DLL のパスの取得方法 621  
SPTHOpen 623  
SPTHTerminate 625  
SPTO コントロール 631  
Str 448  
Sub 406

---

## T

TempDir 482  
TempFile 482  
Terminate 384  
TerminateProcess 615  
TextClose 471  
TextFileReplace 469  
TextOpen 470  
TextRead 472  
TextSeek 475  
TextWrite 474  
Time 453  
TimeOut 384  
Trim 441

---

## U

UCase 436

---

## W

WaitForExec 553  
Weekday 455  
While...End 404  
Windows Server 2003 および Windows XP  
版 JP1/Script ( P-2412-3F84 ) サービスの登  
録 37  
Windows Server 2003 および Windows XP  
版 JP1/Script ( P-2412-3F84 ) のインストー  
ル 33

Windows Vista 版 JP1/Script ( P-2A12-3F84 ) サービスの登録 40  
Windows Vista 版 JP1/Script ( P-2A12-3F84 ) のインストール 39

## Y

---

Year 453

## あ

---

値の更新ダイアログ 262  
値を書式化した文字列を返す 448  
値を文字列で返す 448  
新しいメニューフォームを作成する 173

## い

---

一時停止のサービスを再開する 600  
一時ファイル名を作成する 482  
一時フォルダを取得する 482  
位置変数 390  
一括変更ダイアログ 344  
イベントビューアにメッセージを出力する 514  
イベントログ 360

## う

---

ウォッチウィンドウ 100  
ウォッチへ変数を追加する 128  
うるう年かどうかを調べる 465

## え

---

エディタ 5  
エディタの関連付けダイアログ 250  
エディタの起動と終了 91  
エディタの起動方法 49  
エディタの終了方法 51  
エディタの操作 92  
エディタの動作環境を設定する 111  
エディットコントロールプロパティ ( 一般 )  
ダイアログ 309  
エディットコントロールプロパティ ( キー )  
ダイアログ 313

エディットコントロールプロパティ ( 共通 )  
ダイアログ 311  
エディットコントロールプロパティ ( スタイル )  
ダイアログ 315  
エラーウィンドウ 99  
エラーの有無を返す 632  
エラーメッセージを取得する 575, 624  
演算規則 380  
演算処理コマンド 522  
演算の優先順位 381

## お

---

オプション ( JP1/IM ) ダイアログ 256  
オプション ( 色 ) ダイアログ 264  
オプション ( クラスタ環境 ) ダイアログ 259  
オプション ( 互換性 ) ダイアログ 253, 266  
オプション ( サーバ情報 ) ダイアログ 251  
オプション ( 書式 ) ダイアログ 263  
オプション ( トレース ) ダイアログ 258  
オプション ( 複数起動 ) ダイアログ 255

## か

---

解析トレースファイル 8, 359  
解析トレースファイルの出力形式 638  
外部プログラム呼び出しコマンド 547, 606  
稼働監視プログラムでの JP1/Script の監視 48  
簡易入力 5  
簡易入力機能を使う 105  
簡易入力機能を使ってスクリプトを作成する 65  
簡易入力の操作 138  
簡易入力の操作項目 138  
環境変数を取得する 425  
環境変数を設定する 423

## き

---

キー操作 62, 101, 147, 162, 172  
基本コマンド 413  
基本コマンド一覧 414  
キューサーバにスクリプトを登録する 562

## く

クライアントからサーバ上の JP1/AJS2 の  
キューへのジョブの登録 14  
クラスタ環境で JP1/Script を使用する場  
合のシステム構成 16  
クラスタシステム環境での環境設定 44  
クリアの確認ダイアログ 275  
グリッドの設定ダイアログ 346  
グローバル変数ファイル 9  
グローバル変数を削除する 429, 634  
グローバル変数を取得する 428, 634  
グローバル変数を設定する 426, 633

## け

現在の時刻を返す 453  
現在の日付を返す 453  
検索ダイアログ 268, 276

## こ

更新情報ダイアログ 253  
項目 141, 142  
コーディング規則 381  
コマンド実行時のトラブルと対処方法 354  
コマンドのプロパティを設定する 203  
コマンドプロパティダイアログ 337  
コマンドプロパティの設定ダイアログ 339  
コマンドやステートメントを入力する 139  
コマンドラインに関する規則 389  
コマンドラインの記述規則 391  
コマンドラインの形式 389  
コマンドラインの設定ダイアログ 238  
コマンドラインのパラメタの説明 390  
コマンドラインパラメタの設定ダイアログ  
271  
コメント行を設定・解除する 102  
コントロールの大きさをそろえる 191  
コントロールの属性を一括して変更する 177  
コントロールを一列に並べる 180  
コントロールを均等な間隔で並べる 186  
コントロールをテキストの大きさに合わせる  
195

コントロールをメニューフォームの中央に並  
べる 189  
コンバータが扱うファイル 9  
コンピュータの選択ダイアログ 275  
コンボボックスプロパティ（一般）ダイアロ  
グ 330  
コンボボックスプロパティ（キー）ダイアロ  
グ 334  
コンボボックスプロパティ（共通）ダイアロ  
グ 332  
コンボボックスプロパティ（スタイル）ダイ  
アログ 336

## さ

サーバ環境ファイル 8  
サーバトレースファイル 8, 360  
サーバトレースファイルの出力形式 644  
サービスが存在するかどうかを調べる 589  
サービス情報を設定する 591  
サービス操作コマンド 591  
サービスに制御コマンドを送る 603  
サービスの現在の状態を取得する 602  
サービスの情報を取得する 594  
サービスの設定情報を取得する 602  
サービスの設定情報を変更する 601  
サービス表示名からサービス名を取得する  
604  
サービスを一時停止する 599  
サービスを開始する 598  
サービスを削除する 597  
サービスを停止する 599  
サービスを登録する 596  
削除の確認ダイアログ 274  
サンプルファイル 648

## し

自 PC/ 他 PC 上の実行ファイル呼び出す  
549  
自 PC 上の実行ファイル呼び出す 547  
式の論理否定を求める 536  
実行環境（打ち切り時間）の設定ダイアログ  
242

実行環境 ( 起動日時 ) の設定ダイアログ 240  
実行環境 ( コマンドライン ) の設定ダイアログ 236  
実行環境 ( 作業フォルダ ) の設定ダイアログ 239  
実行環境 ( トレース出力先フォルダ ) の設定ダイアログ 243  
実行環境 ( ユーザトレース情報 ) の設定ダイアログ 245  
実行環境構文ファイル 9  
実行環境構文ファイル ( .SPU ) の概要 216  
実行環境構文ファイル ( .SPU ) の詳細 217  
実行環境の設定 ( 起動情報 ) ダイアログ 229  
実行環境の設定 ( 終了情報 ) ダイアログ 232  
実行環境の設定 ( トレース情報 ) ダイアログ 234  
実行環境ファイル 7  
実行環境ファイルコンバータ 6  
実行環境ファイルコンバータの概要 212  
実行環境ファイルコンバータの実行方法 213  
実行環境ファイルコンバータの操作 212  
実行環境ファイルの変換手順 212  
実行しているスクリプトの強制終了方法 52  
実行ダイアログ 228  
実行中のスクリプトプロセスを一覧表示する 209  
実行トレースファイル 8, 359  
実行トレースファイルの出力形式 640  
実行ファイルの識別子を返す 632  
実行ファイルの実行状態を取得する 555  
実行ファイルの終了コードを返す 631  
実行ファイルの終了待ち / 強制終了を行う 553  
実行ファイルを呼び出す 635  
実行フォルダのパスを取得する 499  
実行フォルダのパスを設定する 499  
指定された時刻に加算減算を行う 462  
指定された時刻の時を返す 456  
指定された時刻の秒を返す 458  
指定された時刻の分を返す 457  
指定された日付に加算減算を行う 458  
指定された日付の月を返す 454  
指定された日付の年を返す 453

指定された日付の日を返す 454  
指定された日付の曜日を返す 455  
指定した文字数分の文字列を返す 437  
指定した連結文字で文字列連結を行う 444  
自動起動情報ファイル 7  
自動起動処理コマンド 559  
自動起動の設定ダイアログ 248  
自動起動の方法 51  
使用するエディタを決める 62  
ショートカットが存在するかどうか調べる 589  
ショートカット操作コマンド 608  
ショートカットを削除する 611  
ショートカットを作成する 609  
初期化ファイルから値を読み込む 467  
初期化ファイルに値を登録する 468  
ジョブの自動運転・実行管理 13  
ジョブの終了を待つ 564  
ジョブをキャンセルする 569  
ジョブを保留 / 解除する 567  
処理詳細 223, 225, 226  
新規作成ダイアログ 222

## す

---

数字の記述規則 380  
数値として評価できるかを調べる 539  
スクリプト OLE コントロール 629  
スクリプト OLE コントロールとは 630  
スクリプト実行時のトラブルと対処方法 353  
スクリプト実行制御 5  
スクリプト実行を強制終了する 625  
スクリプト制御インタフェース 619  
スクリプト制御インタフェース一覧 623  
スクリプト制御インタフェースとは 620  
スクリプト制御インタフェースの記述例 627  
スクリプト制御インタフェースの呼び出し 621  
スクリプト制御マネージャをオープンする 623  
スクリプト制御マネージャをクローズする 624  
スクリプトに関する規則 370

スクリプトの実行環境(各項目)を設定する 80  
 スクリプトの実行環境(すべての項目)を設定する 77  
 スクリプトの実行環境を設定する 129  
 スクリプトの実行を中断する 572  
 スクリプトの自動起動を設定する 84  
 スクリプトの名前を変更する 76  
 スクリプトの文法をチェックする 71  
 スクリプトファイル 7  
 スクリプトファイルの[実行環境の設定]ダイアログボックスでコマンドラインを指定する場合 389  
 スクリプトファイルの自動起動の登録を解除する 561  
 スクリプトファイルの文法をチェックする 113  
 スクリプトファイルを自動起動に登録する 559  
 スクリプトプロセスの表示更新間隔を指定する 210  
 スクリプトを削除する 75  
 スクリプトを作成・保存する 63  
 スクリプトを実行する 86, 123  
 スクリプトを終了し Windows をログオフ / シャットダウンする 617  
 スクリプトを終了する 574  
 スクリプトを追加する 73  
 スクリプトを複写する 72  
 スクリプトを編集する 70  
 スタティックプロパティ(一般)ダイアログ 286  
 スタティックプロパティ(共通)ダイアログ 287  
 スタティックプロパティ(スタイル)ダイアログ 291  
 スタティックプロパティ(背景)ダイアログ 290  
 ステートメント 395  
 ステートメント一覧 396  
 スピーカからビーブ音を鳴らす 573

## せ

---

設定ダイアログ 267  
 全角文字かどうかを調べる 452  
 選択したコントロールを前面に移動する 200  
 選択したコントロールを背面に移動する 202  
 先頭および末尾のスペースを削除した文字列を返す 441  
 先頭のスペースを削除した文字列を返す 439

## そ

---

操作 140, 141, 142  
 その他のコマンド 572, 617  
 その他のファイル 9

## た

---

ターミナルサービス環境で JP1/Script を使用する場合のシステム構成 17  
 ターミナルサービス環境での環境設定 46  
 ダイアログでの操作 223, 225, 226  
 ダイアログの項目 223, 224, 225  
 ダイアログの説明 221  
 ダイアログボックスにメッセージを表示する 511  
 対処の手順 352  
 他のコンピュータとの通信時のアカウント 42  
 タブオーダーの設定ダイアログ 347  
 タブの移動順序を設定する 198

## ち

---

チェック処理コマンド 538, 587  
 直線プロパティ(一般)ダイアログ 317

## て

---

定数 379  
 ディスク空き容量を取得する 502  
 テキストファイルから 1 行分のデータを読み込む 472  
 テキストファイルにデータを書き込む 474  
 テキストファイルの文字列を置き換える 469  
 テキストファイルをオープンする 470

テキストファイルをクローズする 471

## と

---

動作環境情報のバックアップとリカバリー 368

特殊コマンド 577

ドライブの種類を調べる 546

トラブルシューティング 353

トラブル発生時に採取が必要な資料 362

トラブルへの対処方法 351

トレース管理ファイル 9

トレースビューア 5

トレースビューア実行時のトラブルと対処方法 357

トレースビューアの起動と終了 148

トレースビューアの起動方法 50

トレースビューアの終了方法 52

トレースビューアの操作 143

トレースビューアを起動する 148

トレースビューアを終了する 148

トレースファイルの内容を検索する 156

トレースファイルの内容を表示する 150

トレースファイルの表示・非表示を切り替える 149

トレースファイルの表示ウィンドウとメニュー 159

トレースファイルの表示操作 159

トレースファイルを印刷する 155

トレースファイルをクリアする 154

トレースファイルを削除する 152

トレースファイルを別名で保存する 151

## な

---

名前の変更ダイアログ 227

## は

---

バージョン情報 / ファイル日付の新旧を比較する 544

配列変数 376

配列変数の記述規則 377

配列変数のデータ構造例 379

配列変数の要素から文字列を検索し、インデックス番号を返す 433

配列変数の要素数を取得する 430

配列変数を宣言しメモリ領域を割り当てる 422

バックアップ対象ファイル 367

バックアップとリカバリー 367

半角アルファベットの大文字かどうかを調べる 450

半角アルファベットの大文字を小文字に変換する 435

半角アルファベットの小文字かどうかを調べる 450

半角アルファベットの小文字を大文字に変換する 436

半角スペースからなる文字列を返す 438

半角文字かどうかを調べる 451

## ひ

---

比較演算子(=, <>, <, <=, >, >=) 532

左端から指定した文字数分の文字列を返す 436

日付操作コマンド 453

ビットマップを消去する 586

ビットマップを表示する 585

描画・表示処理コマンド 585

標準入力・出力・エラーファイルを解除する 496

標準入力・出力・エラーファイルを設定する 494

## ふ

---

ファイル / ウィンドウにテキストを出力する 507

ファイル・フォルダ操作コマンド 467

ファイルが存在するかどうかを調べる 541

ファイル参照ボタンプロパティ (一般) ダイアログ 300

ファイル参照ボタンプロパティ (キー) ダイアログ 306

ファイル参照ボタンプロパティ (共通) ダイアログ 302

- ファイル参照ボタンプロパティ (スタイル) ダイアログ 307
  - ファイル参照ボタンプロパティ (背景) ダイアログ 305
  - ファイルの上書き確認ダイアログ 226
  - ファイルの削除の確認ダイアログ 227
  - ファイルの種類 7
  - ファイルの追加ダイアログ 225
  - ファイルのバージョン情報を取得する 490
  - ファイルの日付と時刻を取得する 487
  - ファイルの日付と時刻を設定する 486
  - ファイルの複写ダイアログ 224
  - ファイルのプロパティダイアログ 247
  - ファイルの容量 10
  - ファイルの容量を取得する 489
  - ファイルバージョンの指定ダイアログ 270
  - ファイル名の指定ダイアログ 269
  - ファイル名を変更する 480
  - ファイルをコピーする 502
  - ファイルを削除する 479
  - ファイルを統合する 493
  - ファイルを分割する 491
  - ファンクションキープロパティ (一般) ダイアログ 318
  - ファンクションキープロパティ (共通) ダイアログ 319
  - ファンクションキープロパティ (スタイル) ダイアログ 321
  - フォルダ/ファイルの属性を取得する 485
  - フォルダ/ファイルの属性を調べる 543
  - フォルダ/ファイルの属性を設定する 483
  - フォルダが書き込み可能かどうかを調べる 542
  - フォルダが存在するかどうかを調べる 540
  - フォルダの選択ダイアログ 260
  - フォルダの中身が空かどうかを調べる 539
  - フォルダの変更ダイアログ 246
  - フォルダ名の末尾が (¥) かどうかを調べる 545
  - フォルダを削除する 478
  - フォルダを作成する 477
  - 複数パラメタ指定で SPT ファイルを呼び出す 556
  - 二つの式の論理積を求める 534
  - 二つの式の論理和を求める 535
  - 二つの式の和を求める 522
  - 二つの式を比較する 532
  - 二つの時刻の経過時間を取得する 464
  - 二つの時刻を比較する 463
  - 二つの数値の差を求める 524
  - 二つの数値の剰余を求める 525
  - 二つの数値の商を求める 528, 530
  - 二つの数値の積を求める 527
  - 二つの数値のべき乗を求める 531
  - 二つの日付の経過日数を取得する 461
  - 二つの日付を比較する 459
  - フルパスを解析する 497
  - フルパスを作成する 498
  - プログラムグループを削除する 609
  - プログラムグループを作成する 608
  - プログラム内にコメントを記述する 571
  - プログラムのインストール先のフォルダ 32
  - プロセス監視コマンド 613
  - プロセスの起動数を取得する 613
  - プロセスの情報を取得する 614
  - プロセスビューア 5
  - プロセスビューア実行時のトラブルと対処方法 357
  - プロセスビューアの操作 207
  - プロセスビューアの操作項目 207
  - プロセスを強制終了する 615
- 
- へ
- 変数が Empty 値かどうかを調べる 538
  - 変数が定義されているかどうかを調べる 538
  - 変数操作コマンド 422
  - 変数と式の差を変数に代入する 525
  - 変数と式の剰余を変数に代入する 526
  - 変数と式の商を変数に代入する 529, 530
  - 変数と式の積を変数に代入する 527
  - 変数と式のべき乗を変数に代入する 532
  - 変数と式の文字列連結を行う 443
  - 変数と式の和を変数に代入する 523
  - 変数の追加ダイアログ 272
  - 変数の名前付け規則 370
  - 変数名として扱えないキーワード 371

変数を宣言しメモリ領域を割り当てる 422

## ほ

ホスト名, または IP アドレスの変更手順 47

ボタンプロパティ (一般) ダイアログ 292

ボタンプロパティ (キー) ダイアログ 297

ボタンプロパティ (共通) ダイアログ 294

ボタンプロパティ (スタイル) ダイアログ

299

ボタンプロパティ (背景) ダイアログ 296

ボリュームラベルを取得する 501

ボリュームラベルを設定する 500

## ま

マウス操作 61, 100, 147, 162, 171

末尾のスペースを削除した文字列を返す 440

マネージャ 5

マネージャの起動方法 49

マネージャの終了方法 51

マネージャの操作 56

## み

右端から指定した文字数分の文字列を返す

438

## め

メソッドのエラー詳細コードを返す 631

メッセージ出力コマンド 506

メッセージとテキストボックスを表示する

506

メニューエディタ 5

メニューエディタの起動方法 50

メニューエディタの終了方法 52

メニューエディタの操作 163

メニュー情報ファイル 8

メニューフォーム 173

メニューフォーム上にグリッドを表示する

196

メニューフォームの印刷情報ダイアログ 348

メニューフォームの内容を印刷する 205

メニューフォームプロパティ (一般) ダイア  
ログ 279

メニューフォームプロパティ (キー) ダイア  
ログ 282

メニューフォームプロパティ (スタイル) ダイ  
アログ 283

メニューフォームプロパティ (背景) ダイア  
ログ 281

メニューフォームプロパティ (壁紙) ダイア  
ログ 285

メニューフォーム名の指定ダイアログ 344

メニューフォームをコピーする 175

メニューフォームを削除する 176

メニューフォームを作成する 89, 110

メニューフォームをテスト表示する 204

メニューフォームを貼り付ける 175

メニュー呼び出しコマンド 519

## も

文字列操作コマンド 432

文字列の記述規則 380

文字列の文字数を返す 434

文字列分割を行い, 分割文字列の数を返す  
445

文字列分割を行い, 分割文字列を返す 446

文字列連結 (&) 442

文字列連結 (&=) 443

文字列連結 (+) 441

文字列連結を行う 441, 442

文字列を検索し文字位置を返す 432

文字列を検索する 133

文字列を置換する 135

モニタリング実行時の動作環境を設定する  
128

モニタリング実行時のブレークポイントを設  
定・解除する 124

モニタリング情報ファイル 8

モニタリングを実行・中止する 115

## ゆ

ユーザエラーを警告 / 解除する 572

ユーザ定義のメニューを表示する 519



ユーザトレースファイル 360  
ユーザトレースファイル(トレースファイル) 8  
ユーザトレースファイルの出力形式 643  
ユーザプログラムなどから実行形式  
(SPTXE.EXE)でコマンドラインを指定して  
スクリプトファイルを実行する場合 389

## よ

---

読み書き開始位置を先頭または末尾に移動する 475  
読み書きの開始位置を返す 476  
予約語ルールファイル 9  
予約変数 373

## ら

---

ラージファイル 11

## り

---

リストコントロールプロパティ(一般)ダイアログ 323  
リストコントロールプロパティ(キー)ダイアログ 327  
リストコントロールプロパティ(共通)ダイアログ 324  
リストコントロールプロパティ(スタイル)ダイアログ 328  
利用できる変数の個数と容量 371

## れ

---

レジストリから値を読み込む 580  
レジストリサブキーが存在するかどうかを調べる 588  
レジストリサブキーの中身が空かどうかを調べる 587  
レジストリに値を登録する 581  
レジストリのサブキーを削除する 583  
レジストリの値を削除する 583

## ろ

---

ロード方法 621

ログ情報 359  
ログ情報の種類 359  
ログファイルおよびディレクトリ一覧 360

## わ

---

ワークファイル 9



# ソフトウェアマニュアルのサービス ご案内

ソフトウェアマニュアルについて、3種類のサービスをご案内します。ご活用ください。

## 1. マニュアル情報ホームページ

ソフトウェアマニュアルの情報をインターネットで公開しております。

URL <http://www.hitachi.co.jp/soft/manual/>

ホームページのメニューは次のとおりです。

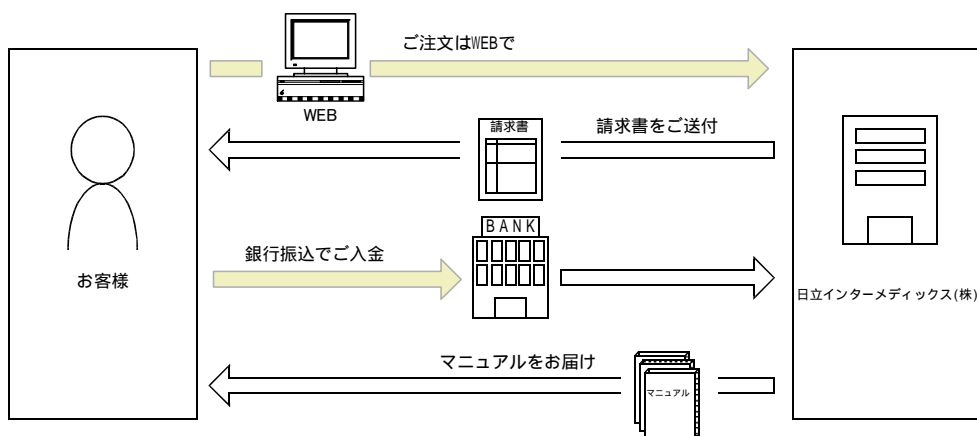
マニュアル一覧	日立コンピュータ製品マニュアルを製品カテゴリ、マニュアル名称、資料番号のいずれかから検索できます。
CD-ROMマニュアル情報	複数マニュアルを格納したCD-ROMマニュアルを提供しています。どの製品に対応したCD-ROMマニュアルがあるか、を参照できます。
マニュアルのご購入	日立インターメディックス(株)の「日立コンピュータ製品マニュアルサイト」からお申し込みできます。 (詳細は「3. マニュアルのご注文」を参照してください。)
Web提供マニュアル一覧	インターネットで参照できるマニュアルの一覧を提供しています。 (詳細は「2. インターネットからのマニュアル参照」を参照してください。)
ご意見・お問い合わせ	マニュアルに関するご意見、ご要望をお寄せください。

## 2. インターネットからのマニュアル参照(ソフトウェアサポートサービス)

ソフトウェアサポートサービスの契約をしていただくと、インターネットでマニュアルを参照できます。本サービスの対象となる契約の種別、及び参照できるマニュアルは、マニュアル情報ホームページでご確認ください。なお、ソフトウェアサポートサービスは、マニュアル参照だけでなく、対象製品に対するご質問への回答、問題解決支援、バージョン更新版の提供など、お客様のシステムの安定的な稼働のためのサービスをご提供しています。まだご契約いただいていない場合は、ぜひご契約いただくことをお勧めします。

## 3. マニュアルのご注文

日立インターメディックス(株)の「日立コンピュータ製品マニュアルサイト」からご注文ください。



下記 URL にアクセスして必要事項を入力してください。

URL [http://www2.himdx.net/manual/privacy.asp?purchase\\_flag=1](http://www2.himdx.net/manual/privacy.asp?purchase_flag=1)

ご注文いただいたマニュアルについて、請求書をお送りします。

請求書の金額を指定銀行へ振り込んでください。なお、送料は弊社で負担します。

入金確認後、7日以内にお届けします。在庫切れの場合は、納期を別途ご案内いたします。