

COBOL

ユーザーズガイド

はしがき

本書は、オペレーティングシステム Windows のもとで使用される COBOL アプリケーションの開発方法についてまとめた説明書であり、最初にお読みください。

本書の構成は次のとおりです。

章	タイトル	内 容
1	COBOL の概要	製品に関する概要説明
2	COBOL アプリケーションカスタマイザの使い方	COBOL アプリケーションカスタマイザの操作法に関する説明
3	COBOL ファイルマッパーの使い方	COBOL ファイルマッパーの操作法に関する説明
4	COBOL アプリケーション	COBOL アプリケーションに関する説明
5	ツール	添付ツールに関する説明

COBOL の説明書として本書以外に次のものがあります。

- ・言語説明書 : 『COBOL 言語説明書』
- ・プログラミングの手引 : 『COBOL プログラミングの手引』
- ・COBOL 開発環境 : 『COBOL 開発環境 利用の手引』

オプション製品および関連製品の説明書として次のものがあります。

- ・索引ファイルの利用方法 : 『ISAM 利用の手引』
- ・COBOL SQL の利用方法 : 『COBOL SQL アクセス 言語説明書』
『COBOL SQL アクセス プログラミングの手引』
『COBOL SQL アクセス ユーザーズガイド』

次は、Standard Edition 版および Enterprise Edition for Windows 版にて利用できます。

- ・整列併合機能の利用方法 : 『SORTKIT/Enterprise 説明書』

次は、Standard Edition 版および Enterprise Edition for Windows 版の SE 動作モードにて利用できます。

- ・COBOL DB アクセスの利用方法 : 『COBOL DB アクセス for Oracle 説明書』
『COBOL DB アクセス for SQL Server 説明書』
- ・COBOL アプリケーション実行環境の利用方法 :
『COBOL アプリケーション実行環境 ジョブ管理機能 説明書』
『COBOL アプリケーション実行環境 プリンタ機能 説明書』
『COBOL WS エミュレータ 説明書』
- ・COBOL 拡張システムサブルーチンの利用方法 :
『COBOL 拡張システムサブルーチン 説明書』

次は、Standard Edition 版にて利用できます。

- ・IFAS FILE : 『IFAS FILE, IFASPRO RDB 利用の手引』
- ・IFASPRO RDB の利用方法 : 『IFAS FILE, IFASPRO RDB 利用の手引』
- ・COBOL GUI の利用方法 : 『COBOL GUI ユーザーズガイド』
『COBOL GUI 画面構築ガイド』

次は、Enterprise Edition 版にて利用できます。

- ・RAF ファイルの利用方法 : 『refam 利用の手引』

2014 年 10 月 初 版
2015 年 4 月 第 2 版
2016 年 4 月 第 3 版
2017 年 4 月 第 4 版
2018 年 10 月 第 5 版
2020 年 4 月 第 6 版
2021 年 4 月 第 7 版
2022 年 4 月 第 8 版

ご注意

- (1) 本書の内容の一部または全部を無断転載することは禁止されています。
- (2) 本書の内容に関しては将来予告なしに変更することがあります。
- (3) 本書は内容について万全を期して作成いたしましたが、万一ご不審な点や誤り、記載もれなどお気付きのことがありましたら、ご連絡ください。
- (4) 運用した結果の影響については、(3)項にかかわらず責任を負いかねますのでご了承ください。

Microsoft, Windows, Windows Server は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Oracle および Java は、オラクルおよびその関連会社の登録商標です。その他の社名、商品名等は各社の商標または登録商標である場合があります。

Linux は Linus Torvalds の米国およびその他の国における商標または登録商標です。

Red Hat, Red Hat Enterprise Linux は米国 Red Hat, Inc. の米国およびその他の国における商標または登録商標です。

Apache, Xerces-C++, Tomcat は、The Apache Software Foundation の米国およびその他の国における商標または登録商標です。

その他、本書に記載の製品名は、各社の商標または登録商標です。

輸出する際の注意事項

本製品（ソフトウェア）は日本国内仕様であり、外国の規格等には準拠していません。
本製品は日本国外で使用された場合、当社は一切責任を負いかねます。また、当社は本製品に関して海外での保守サービスおよび技術サポート等は行っておりません。

備考 (1) 本書は次のオペレーティングシステムに対応しています。

[Enterprise for Windows] Windows Server 2016, Windows Server 2019,
Windows 10

[Standard] Windows 10, Windows Server 2016, Windows Server 2019

[Enterprise for Linux] Red Hat Enterprise Linux 7,
Red Hat Enterprise Linux 8

- (2) 本書に使用しているすべての機能は、プログラムプロダクトであり、次のプロダクト名に対応しています。

COBOL

- (3) 本書では、COBOL および関連製品の名称を次のように表記します。

製品名	マニュアル表記
SORTKIT/Enterprise	SORTKIT
SORTKIT/Enterprise Client	SORTKIT
COBOL ソート・マージ Client	SORTKIT
COBOL DB アクセス for Oracle	COBOL DB アクセス

本書の記述について

本書では、次の記号を用いて、Enterprise Edition for Windows 版の標準動作モード、Enterprise Edition for Windows 版の SE 動作モード、Standard Edition 版、および Enterprise Edition for Linux 版の機能差異を説明します。

標準動作モードは、ACOS-4 COBOL85 や COBOL85 for IPF からの移行に適したモードであり、Enterprise Edition for Windows 版の動作既定値である。

SE 動作モードは、A-VX COBOL85 や COBOL85 Pro からの移行に適したモードであり、Standard Edition 版と動作互換である。

【Enterprise for Windows】記号は、次の機能を表します。

Enterprise Edition for Windows 版の機能

標準動作モードおよび SE 動作モードの機能差異は個別に表します。

【Standard】記号は、次の機能を表します。

Standard Edition 版の機能

【Enterprise for Linux】記号は、次の機能を表します。

Enterprise Edition for Linux 版の機能

【Enterprise】記号は、次の機能を表します。

Enterprise Edition for Windows 版と Enterprise Edition for Linux 版の機能

なお、本書で「シフト JIS」と記述している文字コードは「cp932」です。

目次

第 1 章 COBOL の概要	1-1
1.1 製品概要	1-1
1.2 特長	1-2
1.3 サーバ/クライアント/スタンドアロン・アプリケーションが開発可能	1-2
1.4 対話型/バッチ型・アプリケーションが開発可能	1-3
1.5 基幹システム実行基盤に対応	1-3
1.6 オープン対応機能を提供	1-3
1.7 COBOL 開発環境の提供	1-4
1.8 COBOL アプリケーションカスタマイザの提供	1-4
1.9 COBOL ファイルマッパーの提供	1-6
第 2 章 COBOL アプリケーションカスタマイザの使い方	2-1
2.1 概要	2-1
2.2 アプリケーションカスタマイズの適用範囲	2-2
2.3 カスタマイズ情報の配布	2-2
2.4 コマンドラインからの登録	2-3
2.5 操作環境	2-3
2.5.1 ファイル メニュー	2-3
2.5.2 表示 メニュー	2-6
2.5.3 オプション メニュー	2-7
2.5.4 ヘルプ メニュー	2-8
2.6 基本操作	2-9
2.6.1 アプリケーションをアイコンの状態で行わせるには	2-9
2.6.2 アプリケーションの背景色を白にするには	2-10
2.6.3 アプリケーションの文字色を緑にするには	2-11
2.6.4 ウィンドウを最大化表示するには	2-12
2.6.5 ウィンドウ初期表示位置を指定するには	2-13
2.6.6 タイトルバーを非表示にするには	2-14
2.6.7 コントロールメニューを非表示にするには	2-15
2.6.8 ウィンドウの最小化を不可能にするには	2-16
2.6.9 ウィンドウの移動を不可能にするには	2-17
2.6.10 ウィンドウサイズの変更を可能にするには	2-18
2.6.11 ウィンドウの強制終了を不可能にするには	2-19
2.6.12 タイトルバー文字列を変更するには	2-20
2.6.13 ウィンドウサイズの既定値を変更するには	2-21

2.6.14	起動時のウィンドウサイズを変更するには	2-22
2.6.15	起動時の表示開始位置を変更するには	2-23
2.6.16	先行キー入力を可能にするには	2-24
2.6.17	英小文字を英大文字変換するには	2-25
2.6.18	日本語入力時の半角スペースを全角スペースに変換するには	2-26
2.6.19	混在入力項目で日本語入力システムを起動するには	2-27
2.6.20	STOP 定数 / 実行時エラーを画面の最下行の1行下に表示するには	2-28
2.6.21	キー押下による画面ハードコピーを無効にするには	2-29
2.6.22	画面ハードコピー出力モード	2-30
2.6.23	カナ入力モード	2-31
2.6.24	数字項目入力時の桁数チェック方式を変更するには	2-32
2.6.25	メディア識別名 / キー名を追加するには	2-33
2.6.26	メディア識別名 / キー名を編集するには	2-34
2.6.27	メディア識別名 / キー名を削除するには	2-35
2.6.28	COBOL アプリケーション実行環境 (プリンタ機能) の使用を指定する	2-36
2.6.29	プリンタ機能の PASSTHROUGH モードの使用を指定する	2-37
2.6.30	RDB-STATUS への通知動作を指定する	2-38
2.6.31	ファイル排他制御動作を指定する	2-39
2.6.32	行順ファイル (CSV 形式) に対する READ 文の実行時、対応するセルのない基本項目の値を保持する	2-40
2.6.33	行順ファイル (CSV 形式) の区切り文字を指定する	2-41
2.6.34	行順ファイル (CSV 形式) から外部 10 進項目に文字型データを読み込む	2-42
2.6.35	行順ファイル (CSV 形式) へ文字型データをダブルコーテーションで囲まずに出力する	2-44
2.6.36	カスタマイズするキーを指定する	2-46
2.6.37	キー操作を指定する	2-47
2.6.38	キーの動作を指定する	2-48
2.6.39	値の取り込み方法を指定する	2-50
2.6.40	検査対象を指定する	2-51
2.6.41	END STATUS の値を指定する	2-52
2.6.42	FUNCTION-KEY の値を指定する	2-53
2.6.43	画面表示フォントを変更するには	2-54
2.6.44	画面表示フォントの指定を既定値に戻すには	2-55
2.6.45	画面ハードコピーフォントを変更するには	2-56
2.6.46	画面ハードコピーフォントを指定なしに戻すには	2-57
2.6.47	プリント印字フォントを変更するには	2-58

2.6.48	プリント印字フォントを指定なしに戻すには	2-59
2.6.49	画面表示フォントにあわせるには	2-60
2.6.50	ACCEPT 文の小入力ファイル名を指定するには	2-61
2.6.51	DISPLAY 文の小出力ファイル名を指定するには	2-62
2.6.52	DISPLAY 文で出力する改行位置を指定するには	2-63
2.6.53	DISPLAY 文の出力モードに OUTPUT モードを指定するには	2-64
2.6.54	鳴動ピーブ音を変更するには	2-65
2.6.55	鳴動ピーブ音を BUZZER 句にあわせるには	2-65
2.6.56	鳴動ピーブ音を既定値に戻すには	2-67
2.7	カスタマイズ可能な COBOL アプリケーションとは	2-68
2.8	カスタマイズファイルとは	2-68
第3章	COBOL ファイルマッパーの使い方	3-1
3.1	概要	3-1
3.1.1	ファイルマップ機能の概要	3-1
3.1.2	ファイルマップ指定の適用範囲	3-1
3.1.3	COBOL プログラムの指定とファイルマップ指定	3-2
3.1.4	コマンドラインからの登録	3-4
3.1.5	ファイルマップ指定一覧ファイルの構造	3-4
3.2	操作環境	3-7
3.2.1	ファイル メニュー	3-7
3.2.2	編集 メニュー	3-8
3.2.3	ヘルプ メニュー	3-8
3.2.4	右クリック メニュー	3-9
3.3	基本操作	3-9
3.3.1	登録されているファイルマップ指定一覧を表示する	3-9
3.3.2	新しいファイルマップ指定を追加する	3-10
3.3.3	内容を変更する	3-10
3.3.4	コピーして追加する	3-11
3.3.5	削除する	3-11
3.3.6	ファイルに保存する	3-12
3.3.7	ファイルから追加する	3-12
3.3.8	レジストリへ保存する	3-13
第4章	COBOL アプリケーション	4-1
4.1	概要	4-1
4.2	COBOL ウィンドウ	4-2
4.2.1	COBOL メインウィンドウ	4-2

4.2.2	ガイドディスプレイウィンドウ	4-4
4.2.3	実行時メッセージウィンドウ	4-4
4.3	リソースファイル	4-4
4.4	レジストリ	4-5
4.5	COBOL REST API	4-6
4.5.1	COBOL REST API 生成機能	4-6
4.5.2	COBOL REST API の概要	4-6
4.5.3	COBOL REST API の生成方法	4-7
4.5.4	COBOL REST API の実行方法	4-10
4.5.5	基盤生成ツール	4-11
4.5.6	Java 基盤	4-13
4.5.7	Java 基盤プログラミング	4-15
4.5.8	エラーメッセージ	4-18
4.5.9	制限事項 / 注意事項	4-24
第 5 章	ツール	5-1
5.1	カスタマイズ情報/ファイルマップ情報登録ツール	5-1
5.2	COBOL アプリケーションビューア	5-5
5.3	A-VX モード設定ツール	5-9
5.4	アプリケーション動作モード切替ツール	5-11
付録	A-1
A	COBOL アプリケーションカスタマイザの注意事項	A-1

第1章 COBOL の概要

【Enterprise for Windows】【Standard】Windows 対応の COBOL アプリケーションについて説明します。

1.1 製品概要

本製品は、COBOL 第 4 次規格に対応し、NEC 独自の拡張機能を加え、Windows 10、Windows Server 2016、Windows Server 2019 向けのアプリケーションを開発することができます。

Windows 対応のアプリケーションを開発する場合、C 言語などを使用して、メッセージ処理や API 関数によるインタフェースの作成等が必要となります。しかし、そのような煩雑な処理を本製品のランタイムで提供しており、プログラマは COBOL での業務ロジックに集中して Windows 対応のアプリケーションが作成できます。

本製品で開発したアプリケーションは、サーバ・クライアント環境のサーバ/クライアントのいずれでも、またスタンドアロン環境でもそのまま動作可能です。

本製品はサーバ・アプリケーション、クライアント・アプリケーション、スタンドアロン・アプリケーションともに開発できるスケーラビリティと、COBOL GUI の画面節移行機能、および COBOL SQL の埋め込み SQL 文を COBOL ソース内に記述可能とするなどのオープン性をともに兼ね備えています。

【Standard】Standard Edition 版にて、COBOL GUI は使用できます。

また、A-VX 互換をサポートし、A-VX 上に構築してきたシステムからの移行が可能です。COBOL アプリケーション実行環境などの関連製品と連携し、高い互換性を確保できるとともに、クライアントおよびサーバの資源(ディスクやプリンタ)を有効に利用するシステム移行が行えます。

COBOL 開発環境は、ソースの編集、コンパイル、プロジェクト、実行等効率的な開発が可能です。また、COBOL ファイルマッパーや COBOL アプリケーションカスタマイザにより、各種環境設定やカスタマイズが行えます。

1.2 特長

COBOL は、さまざまな形態の業務システム開発に適した基盤製品であり、以下の特長があります。それぞれの特長を構成する各機能に関しては、本書または『COBOL プログラミングの手引』の該当部分を参照ください。

- サーバ・アプリケーション、クライアント・アプリケーション、スタンドアロン・アプリケーションのいずれも、本製品のみで開発可能。
- 対話型アプリケーションとバッチ型アプリケーションともに開発可能。
- 基幹システム構築向けの実行基盤製品に対応。
- オープン対応機能を提供。
- A-VX COBOL85 および Open COBOL Factory 21/COBOL85 Pro との高い互換性を維持。
- ユーティリティとして、COBOL アプリケーションカスタマイザ、COBOL ファイルマッパーを提供。

1.3 サーバ/クライアント/スタンドアロン・アプリケーションが開発可能

本製品で作成したアプリケーションは、サーバ(Windows Server 2016, Windows Server 2019)上、クライアント(Windows 10)上のいずれでも実行することができ、リモートデスクトップ接続して Windows Server 2016, Windows Server 2019 のサーバアプリケーションを実行することができます。

また、関連製品と組み合わせることで、サーバ集中、クライアント・サーバ、スタンドアロンのいずれのシステム形態でも、再コンパイルすることなく、同じアプリケーションで運用することができます。

1.4 対話型/バッチ型・アプリケーションが開発可能

画面入出力のある対話型アプリケーションと画面入出力のないバッチ型アプリケーションが開発できます。業務形態にあわせ、アプリケーションの形態を選択できます。ただし、バッチ型アプリケーションでは画面入出力関連の機能など一部機能が使用できません。

1.5 基幹システム実行基盤に対応

以下の基幹システム構築向けの実行基盤製品に対応しています。

- WebSAM JobCenter
- 【Standard】COBOL アプリケーション実行環境
- 【Enterprise for Windows】TPBASE

1.6 オープン対応機能を提供

本製品では、以下のオープン対応機能を提供しています。

- ユーザ DLL を作成可能
COBOL アプリケーションで DLL モジュールを作成することができます。
- 64 ビットの呼び出し規約をサポート
_fastcall 呼び出し規約に対応しています。
Visual C++のモジュールとインタフェースを取る場合に、有効です。
- CSV 形式ファイルを入出力可能
COBOL アプリケーションから、表計算ソフトやデータベースソフトでのデータ交換に広く使用されている CSV 形式ファイルに対する入出力が、通常の COBOL ファイル同様に READ/WRITE 文で行えます。
- オプション製品 COBOL GUI 対応
【Standard】COBOL GUI の画面節移行機能を使用して、既存プログラムの画面節を GUI 化することができます。一部のプログラムに関してはソース修正が必要です。
- オプション製品 COBOL SQL 対応
COBOL SQL のブリコンパイル機能を使用して、ODBC 経由でオープン DB(ORACLE, SQL Server, Access など)にアクセスすることができます。

1.7 COBOL 開発環境の提供

本製品は、COBOL 開発環境を提供しています。これを使用することにより、ソースの編集、コンパイル、ソリューション、プロジェクト、資産管理、実行等効率的な開発が可能となります。また、既存 COBOL 資産のプロジェクトファイルをインポートすることもできます。

COBOL 開発環境を用いた開発の流れや開発環境の操作については『COBOL 開発環境 利用の手引』を参照してください。

1.8 COBOL アプリケーションカスタマイズの提供

本製品は、COBOL アプリケーション実行時の動作を、お客様の好みに合う様に変更するツールとして、COBOL アプリケーションカスタマイズを提供しています。

【Standard】Standard Edition 版にて、COBOL アプリケーションカスタマイズは使用できます。

【Enterprise for Windows】SE 動作モードにて、COBOL アプリケーションカスタマイズは使用できません。

カスタマイズできる項目は以下です。

- COBOL アプリケーションのウィンドウに関するカスタマイズ

- 動作に関するカスタマイズ

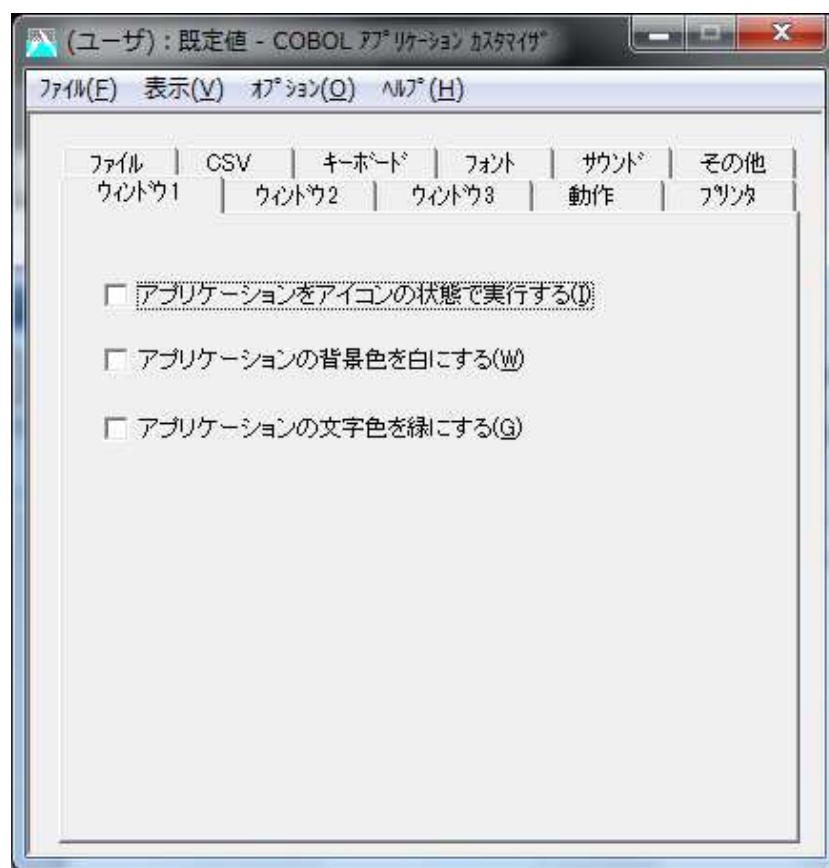
- プリンタ出力に関するカスタマイズ

- キーボードに関するカスタマイズ

- 小入出力に関するカスタマイズ

- データベースに関するカスタマイズ

- 行順ファイル(CSV 形式)に関するカスタマイズ

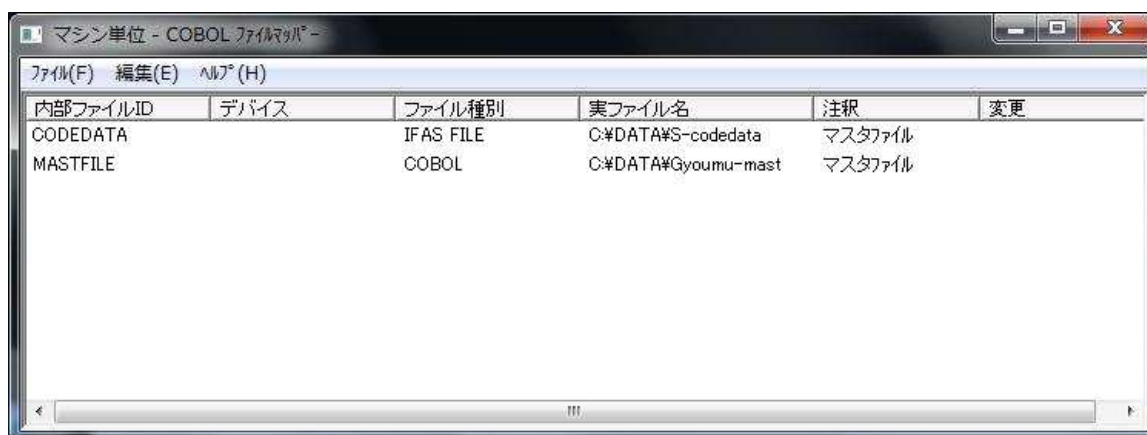


詳しくは、「第2章 COBOL アプリケーションカスタマイザの使い方」を参照してください。

1.9 COBOL ファイルマッパーの提供

本製品は、COBOL アプリケーションが使用するデータファイル(データベースも含む)のファイル種別(COBOL 独自、IFAS FILE、IFASPRO RDB、COBOL DB アクセス等)や実ファイル名を指定する COBOL ファイルマッパーを提供しています。本機能を使用することで、プログラム上の記述とファイルの所在等を分離でき、システムの保守性向上や既存資産の移行性向上にも効果があります。

【Enterprise for Windows】SE 動作モードにて、ファイル種別を変更することができます。ただし、マルチスレッド対応の COBOL アプリケーションでは、ファイルマッパーは使用できません。



詳しくは、「第3章 COBOL ファイルマッパーの使い方」を参照してください。

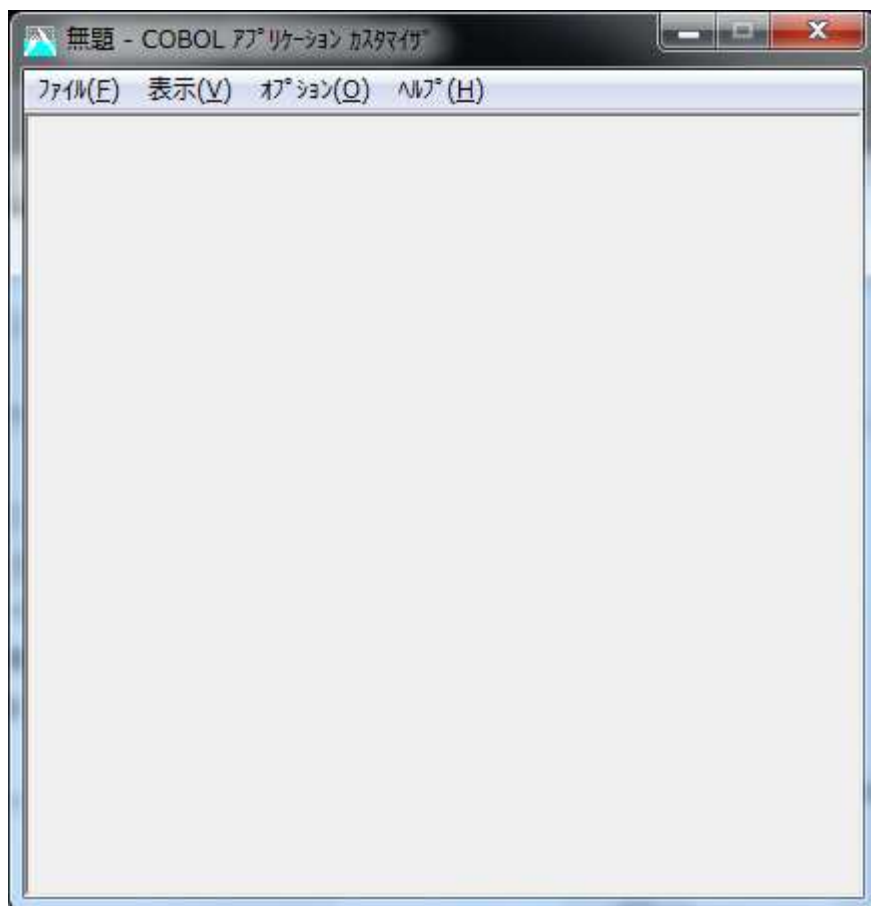
第2章 COBOL アプリケーションカスタマイザの使い方

【Enterprise for Windows】【Standard】COBOL アプリケーションカスタマイザは、Enterprise Edition for Windows 版の SE 動作モード、Standard Edition 版で使用できます。

2.1 概要

COBOL アプリケーションカスタマイザは、COBOL アプリケーション実行時の動作を、COBOL システムがあらかじめ決めたものから、ユーザの好みに合うように変更するためのツールです。

COBOL アプリケーションカスタマイザを起動すると、以下のウィンドウが表示されます。



2.2 アプリケーションカスタマイズの適用範囲

COBOL アプリケーションカスタマイズの指定方法によるカスタマイズ情報の適用範囲を以下に示します。

- (1) ユーザ単位: ユーザ指定レジストリ
COBOL アプリケーションごとに設定したレジストリを使って、現在のユーザ単位でカスタマイズを行います。
同じ COBOL アプリケーションを実行しても、他のユーザではカスタマイズ情報は反映されません。
- (2) マシン単位: ユーザ指定レジストリ
COBOL アプリケーションごとに設定したレジストリを使って、使用するマシン単位でカスタマイズを行います。
同じ COBOL アプリケーションであれば、ユーザが変更されてもカスタマイズは有効です。
- (3) ユーザ単位: 既定値
現在のユーザで実行する COBOL アプリケーションすべてに対して有効になるようにカスタマイズを行います。
このカスタマイズは、他のユーザでは無効ですが、レジストリ指定の有無(-Yr オプション付きコンパイルの有無)に関係なく、すべての COBOL アプリケーションに対して有効となります。
- (4) マシン単位: 既定値
使用するマシン単位で実行する COBOL アプリケーションすべてに対して有効になるようにカスタマイズを行います。
このカスタマイズは、同じマシンであれば、レジストリ指定の有無(-Yr オプション付きコンパイルの有無)に関係なく、すべての COBOL アプリケーションに対して有効となります。

COBOL アプリケーションは、実行時に上記の項番(1) (4)の順にカスタマイズ設定の検索を行い、どれかを発見したところで検索を終了します。

2.3 カスタマイズ情報の配布

カスタマイズした情報を他のマシンに配布する場合には、配布したいカスタマイズ情報をカスタマイズファイルとして保存し、そのカスタマイズファイルを配布し、設定するマシンのコマンドライン上でカスタマイズファイルを入力として COBOL カスタマイズ情報/ファイルマップ情報登録ツールを起動します。

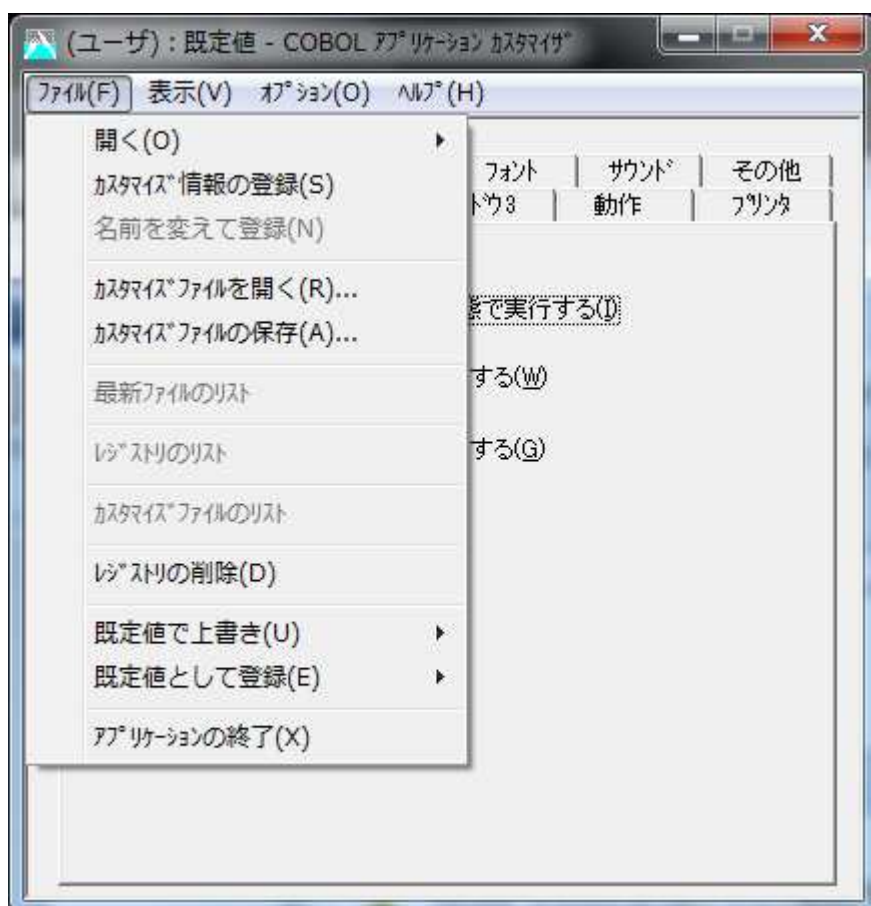
コマンドラインからの登録は「2.4コマンドラインからの登録」を参照ください。

2.4 コマンドラインからの登録

カスタマイズ情報を他マシンに配布する場合、COBOL カスタマイズ情報/ファイルマップ情報登録ツール(CBLCTFM.EXE)を利用し、コマンドライン上から登録を行います。COBOL カスタマイズ情報/ファイルマップ情報登録ツールの使用方法については、「5.1 カスタマイズ情報/ファイルマップ情報登録ツール」を参照してください。

2.5 操作環境

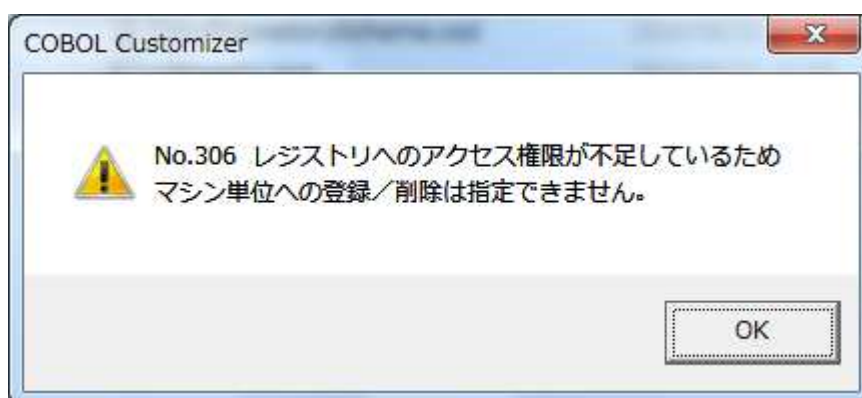
2.5.1 ファイル メニュー



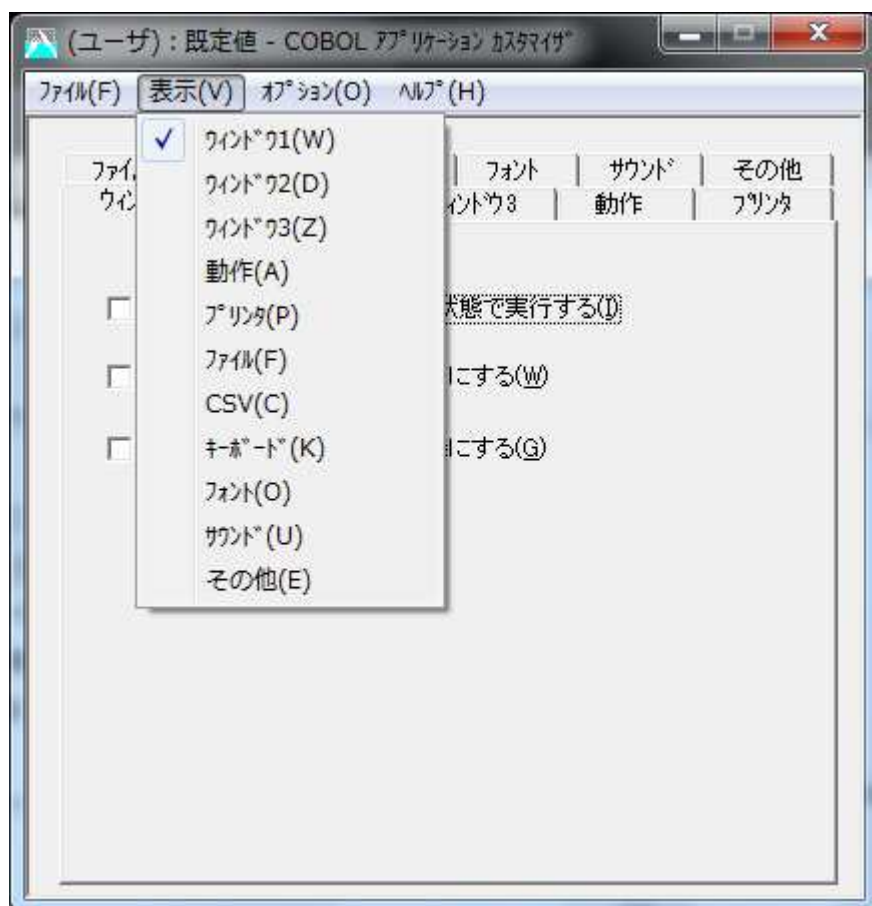
コマンド	ショート カットキー	説 明
アプリケーションを開く(O)		COBOL アプリケーションから編集対象となるレジストリを参照し開きます。 [開く]ダイアログボックスが表示され、COBOL アプリケーション名を入力します。COBOL アプリケーション名を決定すれば、その COBOL アプリケーションに関連付いているレジストリを参照し、ユーザ単位のレジストリとして開きます。指定する COBOL アプリケーションは、COBOL で-Yr オプション付きでコンパイルされている必要があります。
レジストリを開く(R)		編集対象となるレジストリをユーザ単位 / マシン単位で開きます。 [レジストリを開く]ダイアログボックスが表示され、レジストリ名を指定します。 [アプリケーションを開く]コマンドがアプリケーションを指定して(関連付いている)レジストリを設定するのに対し、こちらはレジストリを直接指定します。
既定値を開く(V)		レジストリ設定を行っていない(-Yr オプション付きでコンパイルしていない)または、レジストリ指定では定義されていない場合に、参照する設定(既定値)を任意に変更することができます。 本コマンドを実行することで、ユーザ単位またはマシン単位で現在設定されている既定値が表示されます。必要に応じて、各項目ごとに設定を行います。
カスタマイズ情報の登録(S)		カスタマイズ情報を登録します。
名前を変えて登録(N)		レジストリ名を変更して登録します。 [レジストリを開く]ダイアログボックスが表示され、登録したいレジストリ名を指定します。 このコマンドは、[レジストリを開く]コマンドで開いた場合にのみ有効となります。
カスタマイズファイルを開く(R)		カスタマイズファイル(拡張子 .CCF)を開き、カスタマイズファイルとして保存されていたカスタマイズ情報を読み込みます。 COBOL85 Pro のカスタマイズファイルを開く場合、拡張キーおよび DCOM のカスタマイズ情報はメモ扱いとします。
カスタマイズファイルの保存(A)		[ファイル名を付けて保存]ダイアログボックスが表示され、カスタマイズファイル(拡張子 .CCF)に現在のカスタマイズ情報を保存します。
レジストリの削除(D)		現在開いているレジストリおよびカスタマイズ情報を削除します。 本コマンドを実行すると、確認メッセージ表示後、指定されたレジストリを削除します。 削除した内容を元に戻すことはできませんので、ご注意ください。
既定値で上書き(U)		現在選択されているカスタマイズ情報を既定値(ユーザ単

		位 または マシン単位)で上書きします。 本コマンドを実行すると、確認メッセージ表示後、上書きを行います。
既定値として登録(E)		現在選択されているカスタマイズ情報を既定値(ユーザ単位 または マシン単位)として保存します。 本コマンドを実行すると、確認メッセージ表示後、既定値として保存します。
アプリケーションの終了 (X)		COBOL アプリケーションカスタマイズを終了します。

マシン単位の登録 / 削除を行うには、管理者権限が必要となります。
権限が不足している場合、起動時に以下のメッセージが表示されます。

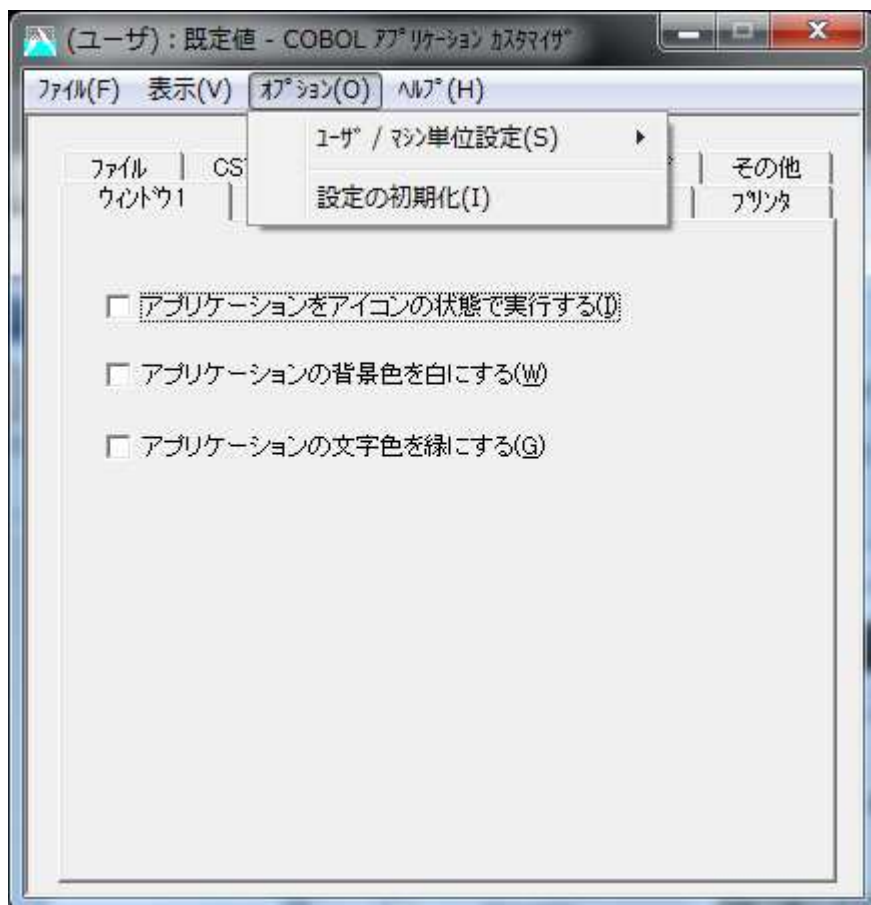


2.5.2 表示 メニュー



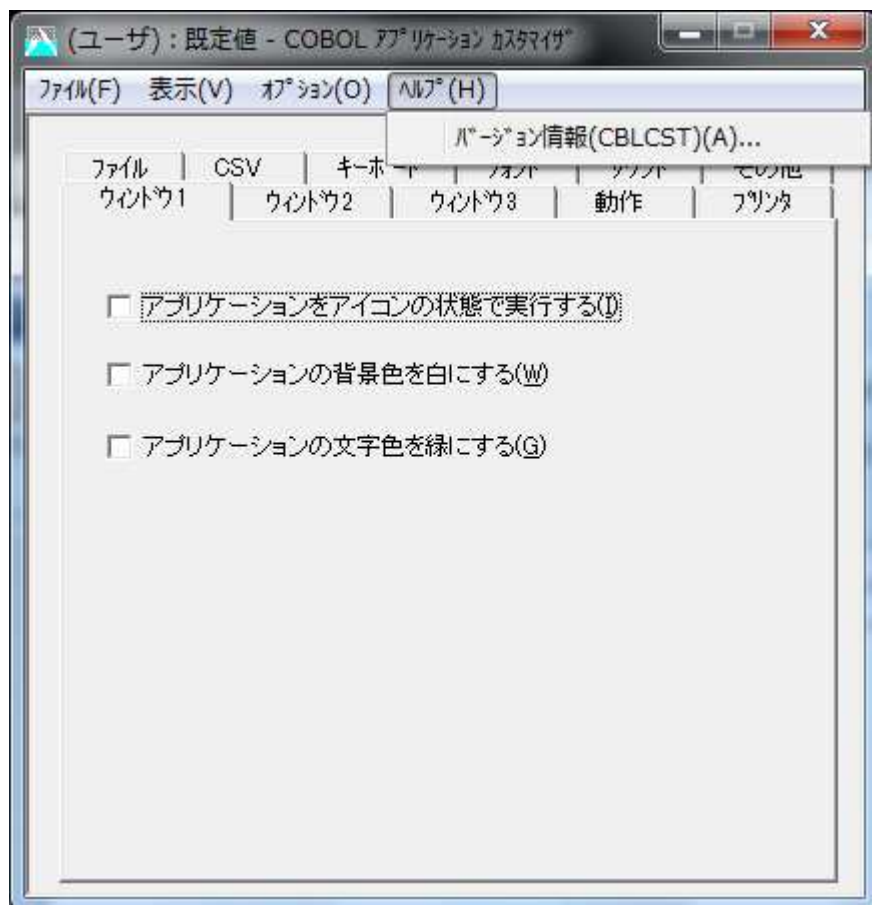
コマンド	ショート カットキー	説 明
ウィンドウ 1 (<u>W</u>)		[ウィンドウ 1] タブに切り替えます。
ウィンドウ 2 (<u>D</u>)		[ウィンドウ 2] タブに切り替えます。
ウィンドウ 3 (<u>Z</u>)		[ウィンドウ 3] タブに切り替えます。
動作 (<u>A</u>)		[動作] タブに切り替えます。
プリンタ (<u>P</u>)		[プリンタ] タブに切り替えます。
ファイル (<u>F</u>)		[ファイル] タブに切り替えます。
CSV (<u>C</u>)		[CSV] タブに切り替えます。
キーボード (<u>K</u>)		[キーボード] タブに切り替えます。
フォント (<u>O</u>)		[フォント] タブに切り替えます。
サウンド (<u>U</u>)		[サウンド] タブに切り替えます。
その他 (<u>E</u>)		[その他] タブに切り替えます。

2.5.3 オプション メニュー



コマンド	ショート カットキー	説 明
ユーザ / マシン単位設定 (S)		COBOL アプリケーションカスタマイザの以下の既定値を指定 します。 ・ [レジストリを開く] ダイアログボックスのユーザ単位 / マシン 単位の選択ラジオボタンの既定値 (ユーザ単位 / マシン 単位を既定値にする)
設定の初期化(I)		現在開いている設定をインストール時の状態に戻します。 確認メッセージ表示後、現在開いている設定を破棄し、イン ストール時の設定に変更します。破棄された情報は元に戻 せませんので、注意してください。

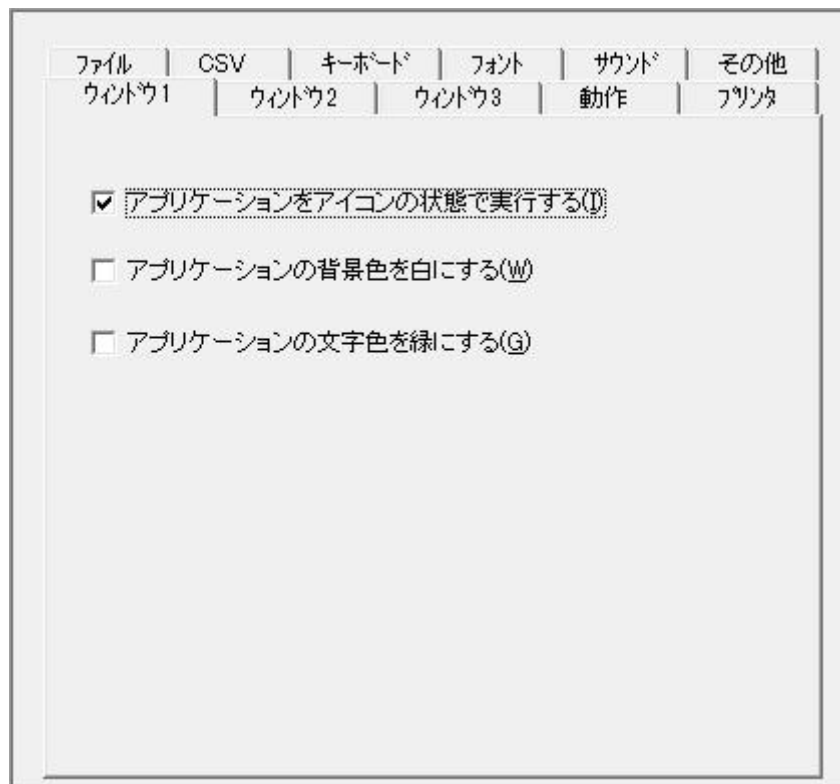
2.5.4 ヘルプ メニュー



コマンド	ショート カットキー	説 明
バ ー ジ ョ ン 情 報 (CBLCST) (A)		COBOL アプリケーションカスタマイザのバージョンや著作権など についての情報を表示します。

2.6 基本操作

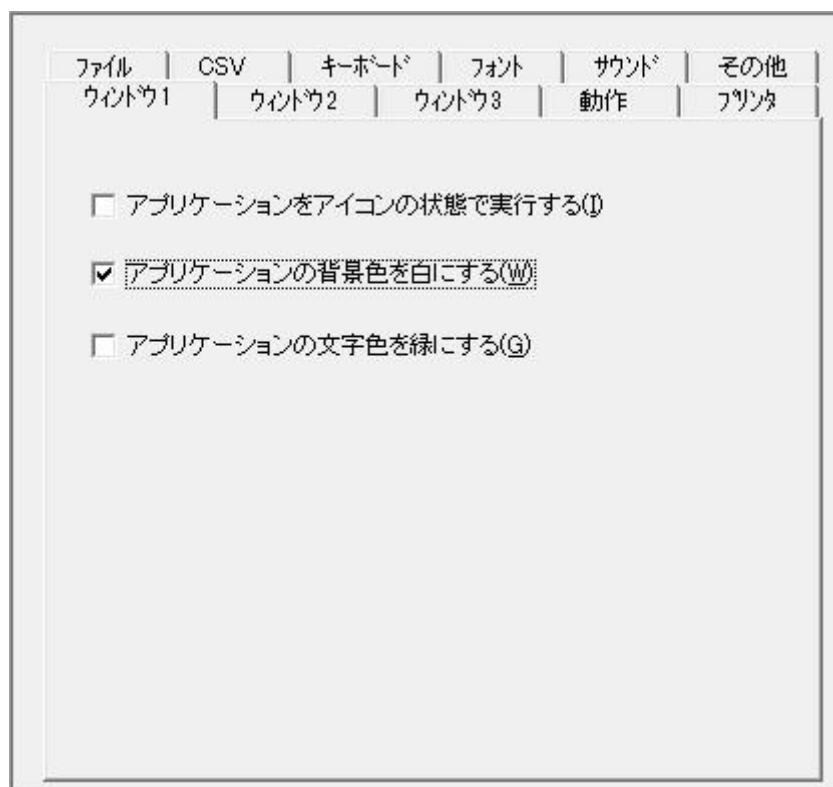
2.6.1 アプリケーションをアイコンの状態で行わせるには



アプリケーションをアイコンの状態で行わせる

- ・ チェックボックスがオンの場合、COBOL アプリケーションはアイコン状態で実行されます。
- ・ チェックボックスがオフの場合、COBOL アプリケーションはウィンドウ状態で実行されます。

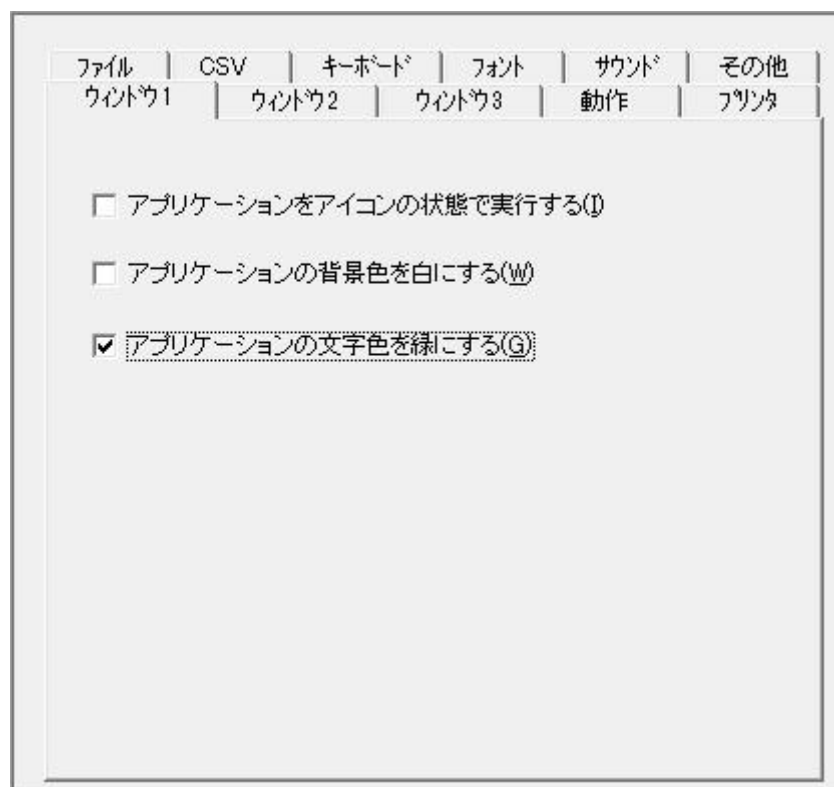
2.6.2 アプリケーションの背景色を白にするには



アプリケーションの背景色を白にする

- ・ チェックボックスがオンの場合、COBOL アプリケーションの背景色は白になります。
- ・ チェックボックスがオフの場合、COBOL アプリケーションの背景色は黒になります。

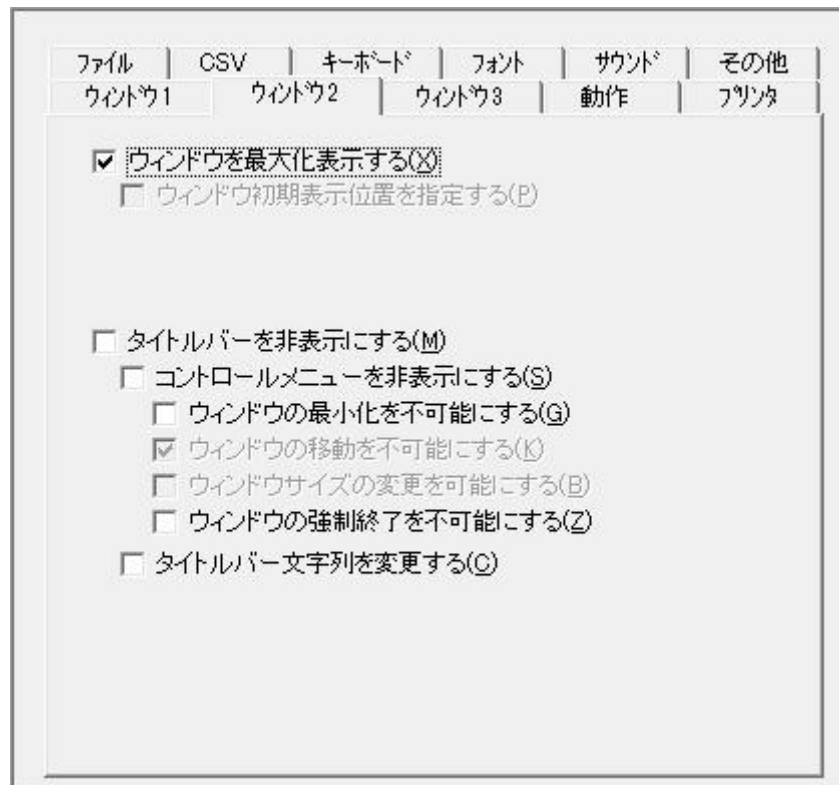
2.6.3 アプリケーションの文字色を緑にするには



アプリケーションの文字色を緑にする

- ・ チェックボックスがオンの場合、罫線および色指定のない文字の色が緑になります。
- ・ チェックボックスがオフの場合、罫線および色指定のない文字の色が白(背景色を白にした場合は黒)になります。

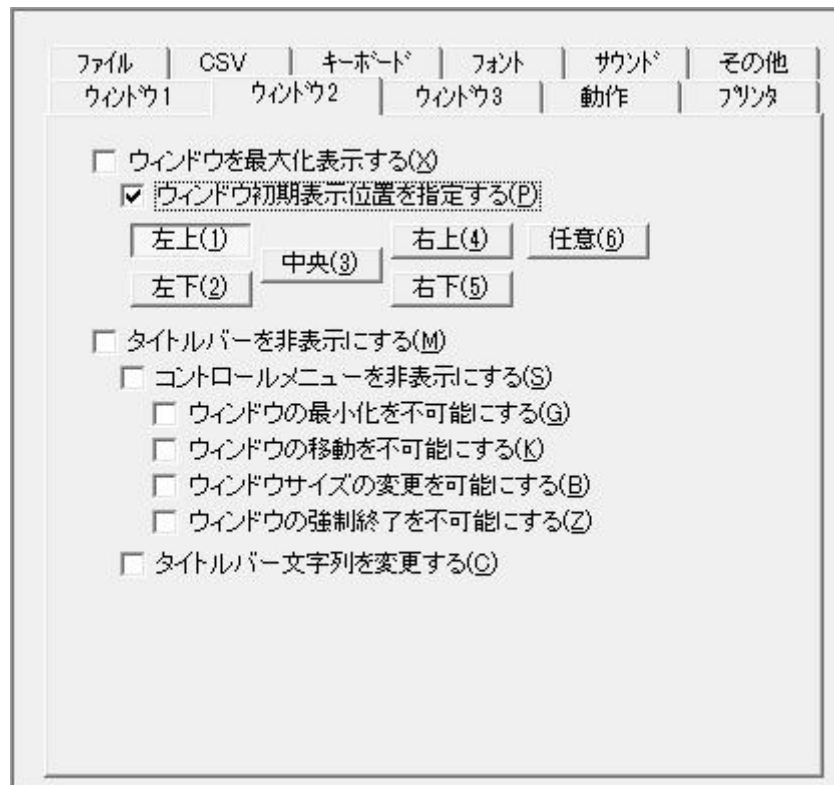
2.6.4 ウィンドウを最大化表示するには



アプリケーション(ウィンドウ)を最大化の状態で行う

- ・ チェックボックスがオンの場合、COBOL アプリケーションを起動した際、COBOL ウィンドウをデスクトップのサイズにあわせて最大化して表示されます。
- ・ チェックボックスがオフの場合、COBOL アプリケーションを起動した際、ウィンドウサイズの既定値(行数および桁数)と使用フォントサイズによって決定されるウィンドウサイズで表示されます。

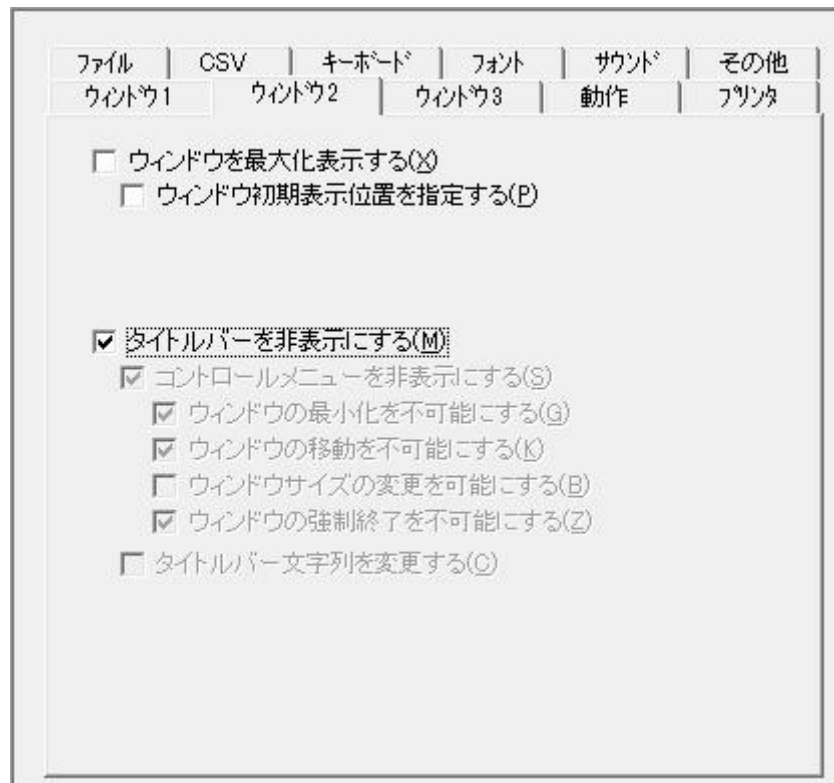
2.6.5 ウィンドウ初期表示位置を指定するには



アプリケーション(ウィンドウ)の初期表示位置を指定する

- ・ チェックボックスがオンの場合、COBOL アプリケーションを起動した際の COBOL ウィンドウの表示位置を指定します。
表示位置は、デスクトップの左上端、左下端、右上端、右下端、中央、または任意(デスクトップの上および左からの位置をピクセルで指定)から選べます。
- ・ チェックボックスがオフの場合、COBOL アプリケーションを起動した際の COBOL ウィンドウはシステムが決める位置に表示されます。

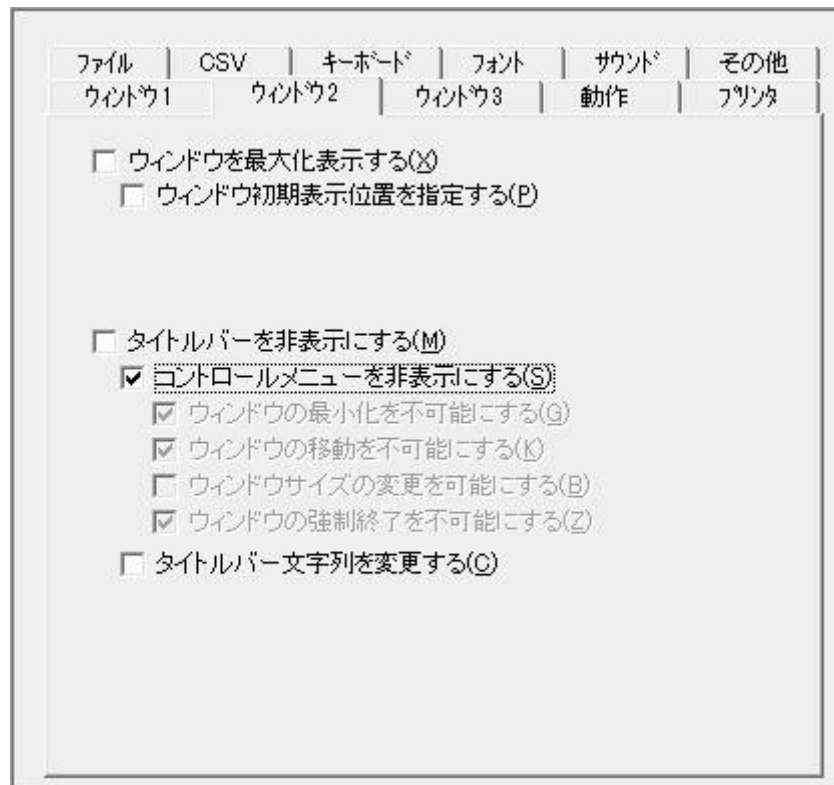
2.6.6 タイトルバーを非表示にするには



タイトルバーを非表示の状態で行う

- ・ チェックボックスがオンの場合、COBOL ウィンドウのタイトルバーは表示されず、タイトルバーアイコンや最小化ボタン等のタイトルバーボタンも表示されません。コントロールメニューの非表示、ウィンドウ表示位置の固定、ウィンドウの最小化ボタンの非表示およびウィンドウの強制終了不可能が暗黙的に指定されたものとみなし、ウィンドウ表示位置の指定がなければ、デスクトップの左上端に COBOL ウィンドウが表示されます。
- ・ チェックボックスがオフの場合、COBOL ウィンドウのタイトルバーを表示します。

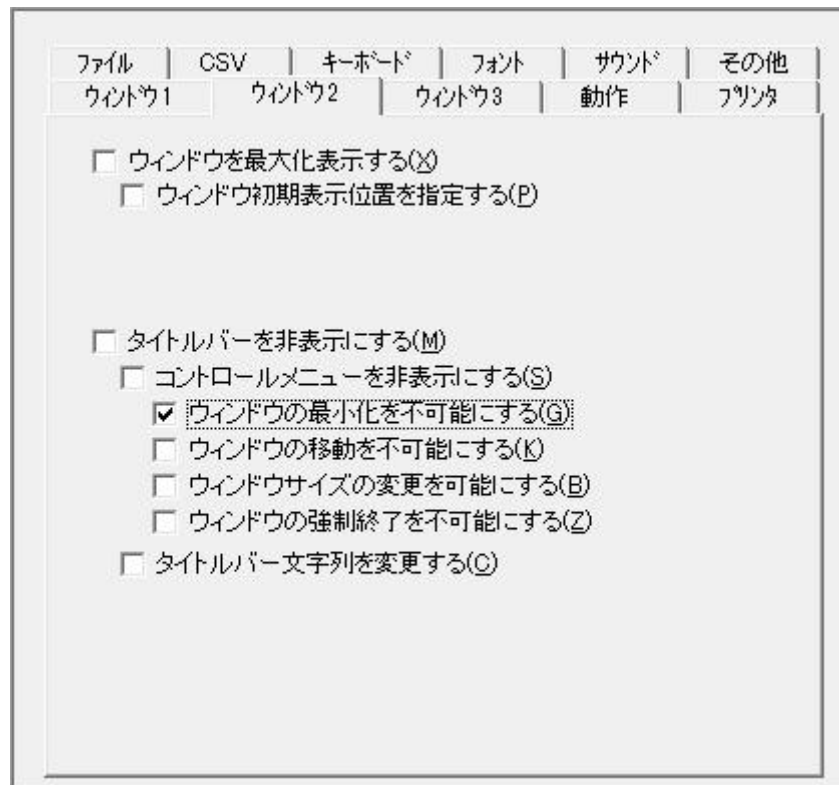
2.6.7 コントロールメニューを非表示にするには



コントロールメニューを非表示にする

- ・ チェックボックスがオンの場合、COBOL ウィンドウのコントロールメニューは表示されず、最小化ボタン等の各種ボタン類も非表示となり、コントロールメニューに含まれる機能(「移動」、「サイズ変更」、「最小化」および「閉じる」)が使えなくなります。ウィンドウの最小化ボタンの非表示、COBOL ウィンドウ表示位置の固定、COBOL ウィンドウの強制終了ボタンの非表示が暗黙的に指定されたものとみなします。
- ・ チェックボックスがオフの場合、COBOL ウィンドウのコントロールメニューを表示します。

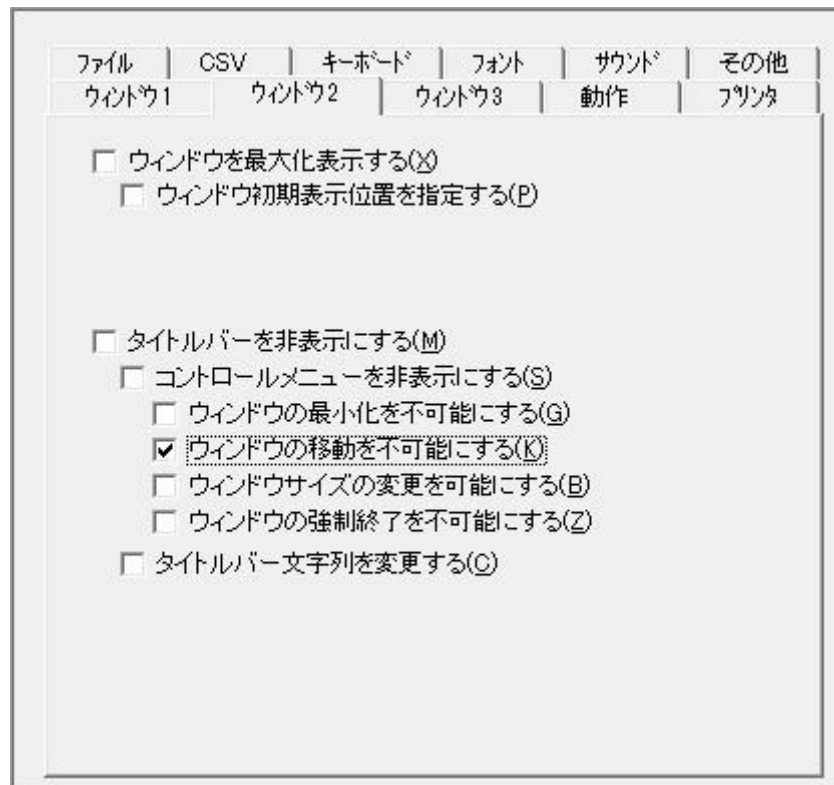
2.6.8 ウィンドウの最小化を不可能にするには



ウィンドウの最小化を不可能な状態で実行する

- ・ チェックボックスがオンの場合、COBOL ウィンドウの最小化ボタンは表示されず、コントロールメニューの「最小化」も使用できません。タイトルバーやコントロールメニューを非表示にした場合には、本設定に関係なく COBOL ウィンドウの最小化が行えません。
- ・ チェックボックスがオフの場合、COBOL ウィンドウの最小化ボタンを表示します。

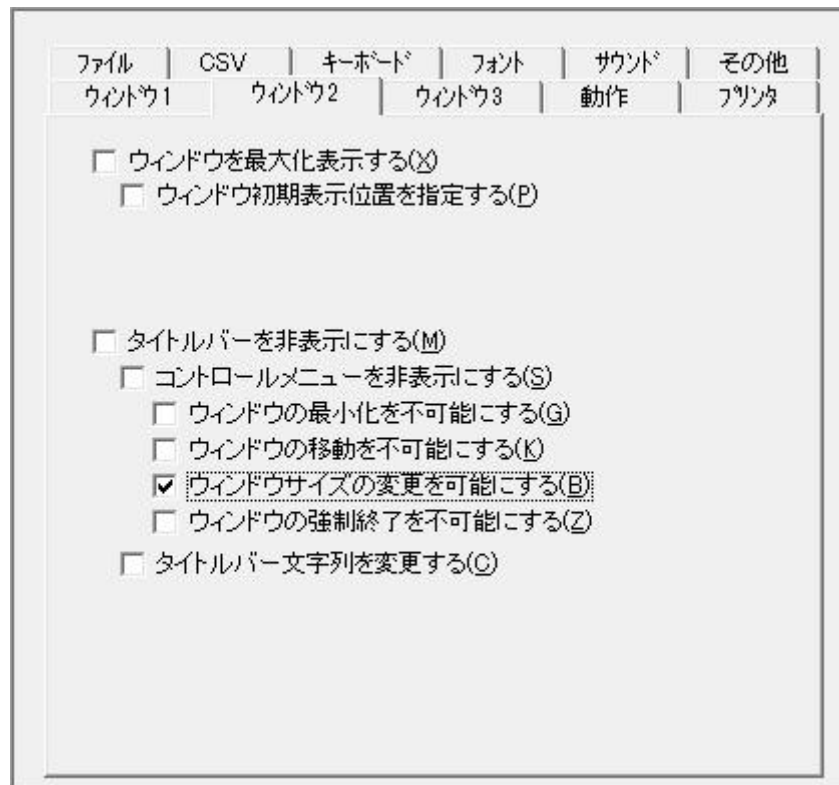
2.6.9 ウィンドウの移動を不可能にするには



ウィンドウの移動を不可能な状態で実行する

- ・ チェックボックスがオンの場合、COBOL ウィンドウの表示位置の変更を不可能(表示位置を固定)にします。固定にすると、COBOL ウィンドウの移動は行えず、コントロールメニューの「移動」も使用できません。また、タイトルバーやコントロールメニューを非表示にした場合や最大化表示を行った場合には、本設定に関係なく COBOL ウィンドウの移動ができません。
- ・ チェックボックスがオフの場合、ウィンドウの移動は可能です。

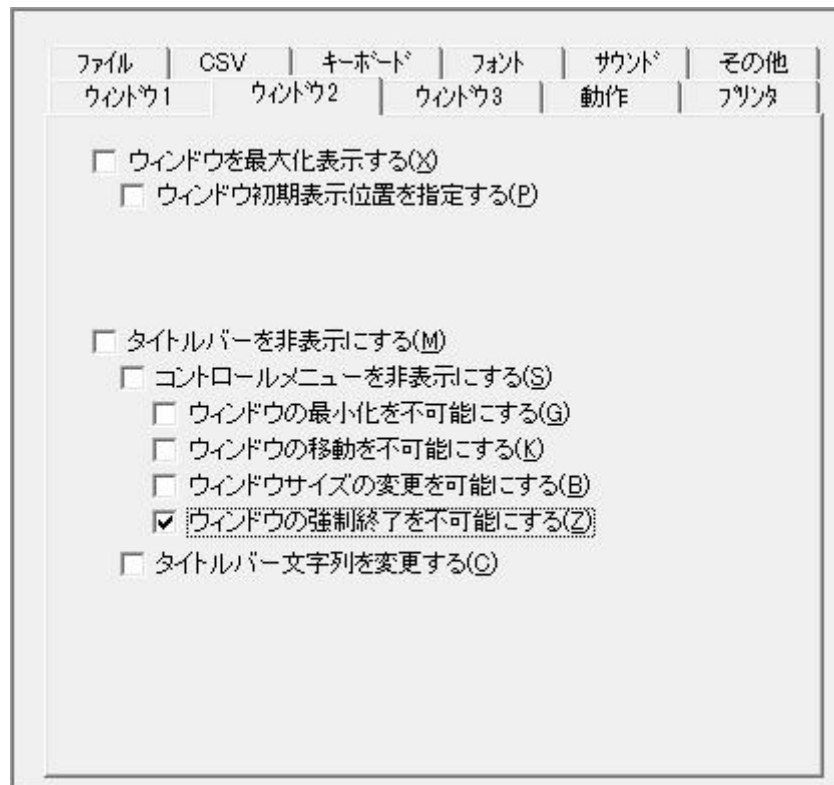
2.6.10 ウィンドウサイズの変更を可能にするには



ウィンドウサイズの変更が可能な状態で実行する

- ・ チェックボックスがオンの場合、COBOL ウィンドウにスクロールバーを表示し、COBOL ウィンドウのサイズ変更を可能にします。タイトルバーやコントロールメニューを非表示にした場合や最大化表示を行った場合には、スクロールバーは表示されず、COBOL ウィンドウのサイズ変更はできません。
- ・ チェックボックスがオフの場合、COBOL ウィンドウにスクロールバーは表示されず、サイズ変更は行えません。

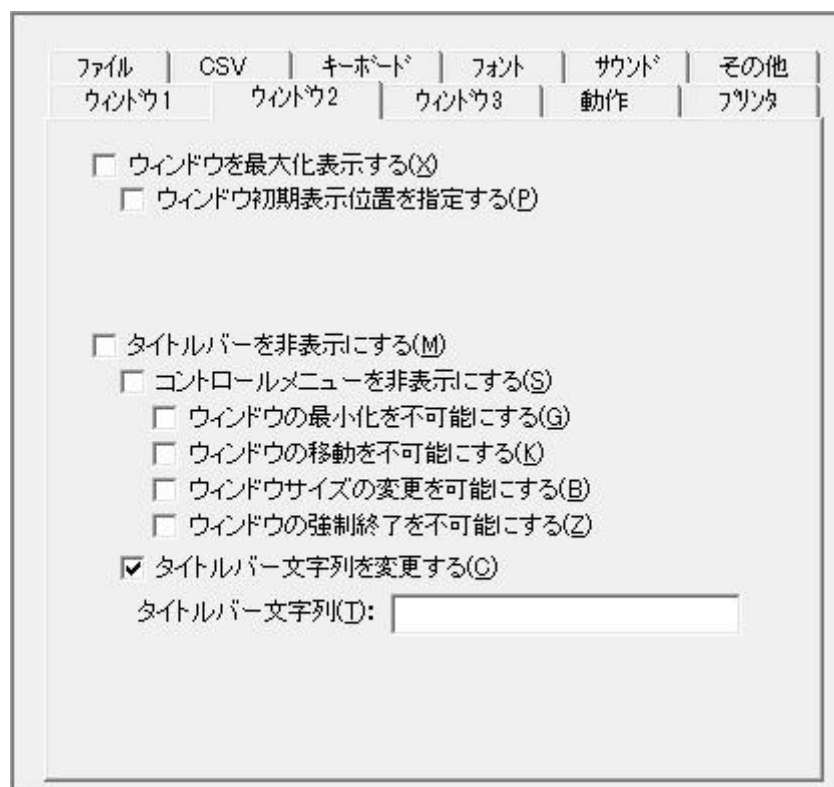
2.6.11 ウィンドウの強制終了を不可能にするには



アプリケーション(ウィンドウ)の強制終了が不可能な状態で実行する

- ・ チェックボックスがオンの場合、COBOL アプリケーション(ウィンドウ)を閉じるボタンは表示されず、強制終了を不可能にします。タイトルバーやコントロールメニューを非表示にした場合には、本設定に関係なく COBOL アプリケーション(ウィンドウ)の強制終了は行えません。
- ・ チェックボックスがオフの場合、COBOL アプリケーション(ウィンドウ)の閉じるボタンを表示します。

2.6.12 タイトルバー文字列を変更するには



タイトルバー文字列を変更する

- ・ チェックボックスがオンの場合、エディットボックスが表示され、タイトルを入力することができます。
- ・ チェックボックスがオフの場合、タイトルは COBOL アプリケーション名になります。

2.6.13 ウィンドウサイズの既定値を変更するには

ファイル	CSV	キーボード	フォント	サウンド	その他
ウィンドウ1	ウィンドウ2	ウィンドウ3	動作	プリンタ	

☒ ウィンドウサイズの既定値を変更する(D)

行数(L) 桁数(C)

☐ 起動時のウィンドウサイズを変更する(S)

☐ 起動時の表示開始位置を変更する(V)

ウィンドウサイズの既定値を変更する

- ・ チェックボックスがオンの場合、エディットボックスが表示され、COBOL ウィンドウのサイズを行数と桁数で指定することができます。
- ・ チェックボックスがオフの場合、COBOL ウィンドウサイズは 25 行 × 80 桁となります。

2.6.14 起動時のウィンドウサイズを変更するには

ファイル	CSV	キーボード	フォント	サウンド	その他
ウィンドウ1	ウィンドウ2	ウィンドウ3	動作	プリンタ	

☐ ウィンドウサイズの既定値を変更する(D)
行数(L) 桁数(O)

☒ 起動時のウィンドウサイズを変更する(S)
行数(I) 桁数(O)

☐ 起動時の表示開始位置を変更する(V)

起動時のウィンドウサイズを変更する

[ウィンドウ 2] タブのウィンドウサイズの変更が可能な状態で実行する のチェックボックスがオンの場合、編集可能となります。

- ・ チェックボックスがオンの場合、エディットボックスが表示され、COBOL アプリケーションを起動した際の COBOL ウィンドウのサイズを行数と桁数で指定します。
- ・ チェックボックスがオフの場合、COBOL アプリケーションを起動した際の COBOL ウィンドウのサイズはウィンドウサイズの既定値を変更する によって決定されるウィンドウサイズで表示されます。

2.6.15 起動時の表示開始位置を変更するには

ファイル	CSV	キーボード	フォント	サウンド	その他
ウィンドウ1	ウィンドウ2	ウィンドウ3	動作	プリンタ	

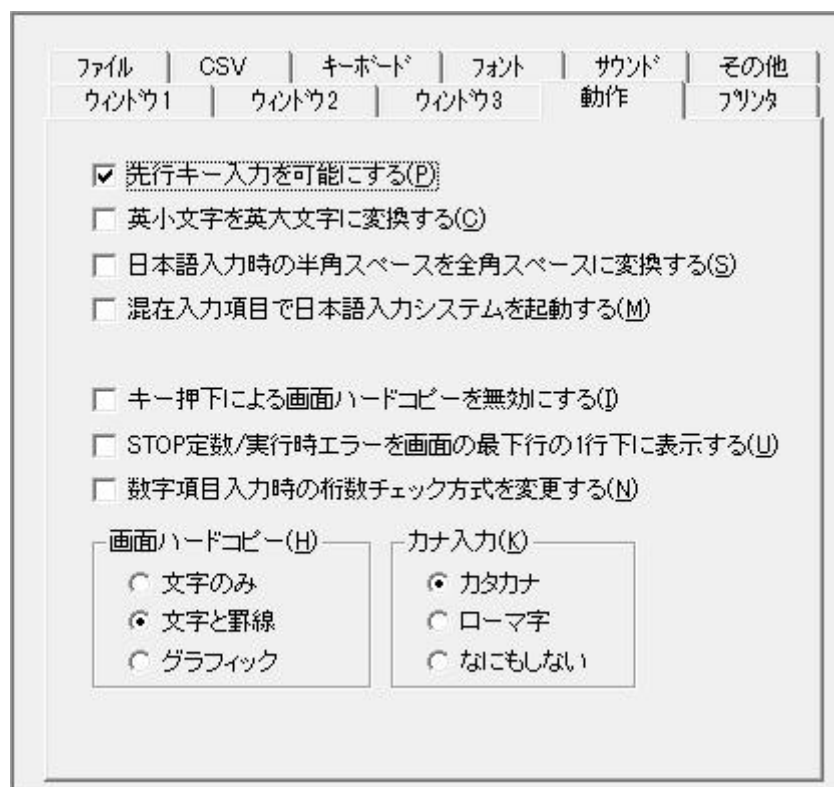
<input type="checkbox"/> ウィンドウサイズの既定値を変更する(D)
行数(L) <input type="text" value="25"/> 桁数(O) <input type="text" value="80"/>
<input checked="" type="checkbox"/> 起動時のウィンドウサイズを変更する(S)
行数(I) <input type="text" value="25"/> 桁数(O) <input type="text" value="80"/>
<input checked="" type="checkbox"/> 起動時の表示開始位置を変更する(V)
行位置(N) <input type="text" value="1"/> 桁位置(U) <input type="text" value="1"/>

起動時の表示開始位置を変更する

ウィンドウサイズの既定値を変更する のチェックボックスがオンの場合、編集可能となります。

- ・ チェックボックスがオンの場合、エディットボックスが表示され、COBOL アプリケーションを起動した際の COBOL ウィンドウの表示開始位置を行数と桁数で指定します。
- ・ チェックボックスがオフの場合、COBOL アプリケーションを起動した際の COBOL ウィンドウの表示開始位置は 1 行 × 1 桁目から表示されます。

2.6.16 先行キー入力を可能にするには



先行キー入力を可能にする

- ・ チェックボックスがオンの場合、ACCEPT 文の実行中だけでなく入力されたキーが有効となります。
- ・ チェックボックスがオフの場合、ACCEPT 文の実行中のみ入力されたキーが有効となります。

2.6.17 英小文字を英大文字変換するには

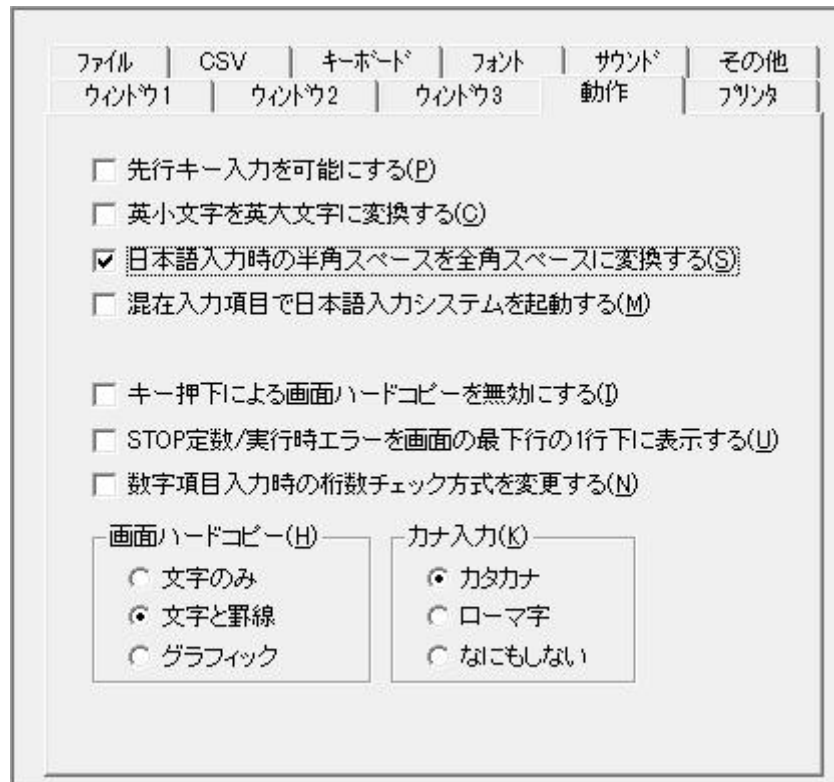
ファイル	CSV	キーボード	フォント	サウンド	その他
ウインドウ1	ウインドウ2	ウインドウ3	動作	プリンタ	

<input type="checkbox"/> 先行キー入力を可能にする(P)	
<input checked="" type="checkbox"/> 英小文字を英大文字に変換する(C)	
<input type="checkbox"/> 日本語入力時の半角スペースを全角スペースに変換する(S)	
<input type="checkbox"/> 混在入力項目で日本語入力システムを起動する(M)	
<input type="checkbox"/> キー押下による画面ハードコピーを無効にする(I)	
<input type="checkbox"/> STOP定数/実行時エラーを画面の最下行の1行下に表示する(U)	
<input type="checkbox"/> 数字項目入力時の桁数チェック方式を変更する(N)	
画面ハードコピー(H)	カナ入力(K)
<input type="radio"/> 文字のみ	<input checked="" type="radio"/> カタカナ
<input checked="" type="radio"/> 文字と罫線	<input type="radio"/> ローマ字
<input type="radio"/> グラフィック	<input type="radio"/> なにもしない

英小文字を英大文字に変換する

- ・ チェックボックスがオンの場合、COBOL アプリケーションは英小文字と英大文字を同一として扱うため、英小文字をキーボードから入力しても英大文字として入力表示されます。
- ・ チェックボックスがオフの場合、COBOL アプリケーションは英小文字と英大文字を区別して扱います。

2.6.18 日本語入力時の半角スペースを全角スペースに変換する には



日本語入力時の半角スペースを全角スペースに変換する

- ・ チェックボックスがオンの場合、COBOL アプリケーションで日本語入力システムを起動している際、「スペース」キーを押したときに入力される半角スペースを全角スペースにします。
- ・ チェックボックスがオフの場合、COBOL アプリケーションで日本語入力システムを起動している際、「スペース」キーを押したときに入力される半角スペースの変換を行いません。

2.6.19 混在入力項目で日本語入力システムを起動するには

ファイル	CSV	キーボード	フォント	サウンド	その他
ウィンドウ1	ウィンドウ2	ウィンドウ3	動作	プリンタ	

☐ 先行キー入力を可能にする(P)

☐ 英小文字を英大文字に変換する(C)

☐ 日本語入力時の半角スペースを全角スペースに変換する(S)

☒ 混在入力項目で日本語入力システムを起動する(M)

☐ キー押下による画面ハードコピーを無効にする(I)

☐ STOP定数/実行時エラーを画面の最下行の1行下に表示する(U)

☐ 数字項目入力時の桁数チェック方式を変更する(N)

画面ハードコピー(H)

☐ 文字のみ

☒ 文字と罫線

☐ グラフィック

カナ入力(K)

☒ カタカナ

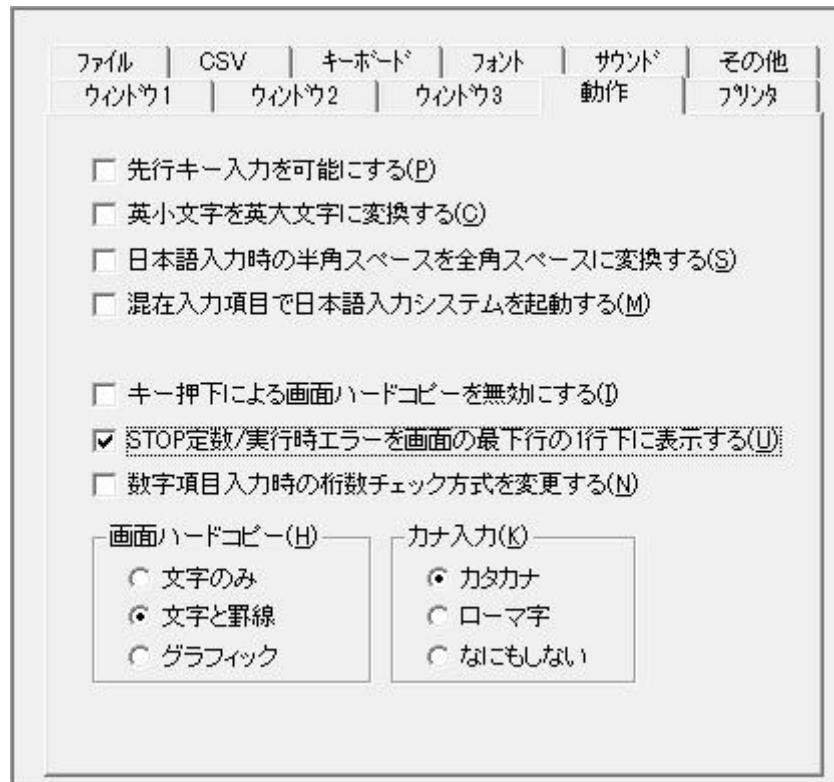
☐ ローマ字

☐ なにもしない

混在入力項目で日本語入力システムを起動する

- ・ チェックボックスがオンの場合、COBOL アプリケーションの混在入力可能な位置にカーレットが位置付いたとき、日本語入力システムが起動されます。
- ・ チェックボックスがオフの場合、COBOL アプリケーションの混在入力可能な位置にカーレットが位置付いたときでも、日本語入力システムは起動されません。

2.6.20 STOP 定数 / 実行時エラーを画面の最下行の 1 行下に表示するには



STOP 定数 / 実行時エラーを画面の最下行の 1 行下に表示する

- ・ チェックボックスがオンの場合、STOP 定数 / 実行時エラーのメッセージ文字列は、画面の最下行の 1 行下に表示され、ESC キー押下待ちとなります。
- ・ チェックボックスがオフの場合、STOP 定数 / 実行時エラーのメッセージ文字列は、ポップアップウインドウに表示され、操作介入待ちとなります。

2.6.21 キー押下による画面ハードコピーを無効にするには

ファイル	CSV	キーボード	フォント	サウンド	その他
ウィンドウ1	ウィンドウ2	ウィンドウ3	動作	プリンタ	

<input type="checkbox"/> 先行キー入力を可能にする(P)
<input type="checkbox"/> 英小文字を英大文字に変換する(C)
<input type="checkbox"/> 日本語入力時の半角スペースを全角スペースに変換する(S)
<input type="checkbox"/> 混在入力項目で日本語入力システムを起動する(M)
<input checked="" type="checkbox"/> キー押下による画面ハードコピーを無効にする(U)
<input type="checkbox"/> STOP定数/実行時エラーを画面の最下行の1行下に表示する(U)
<input type="checkbox"/> 数字項目入力時の桁数チェック方式を変更する(N)

画面ハードコピー(H)	カナ入力(K)
<input type="radio"/> 文字のみ	<input checked="" type="radio"/> カタカナ
<input checked="" type="radio"/> 文字と罫線	<input type="radio"/> ローマ字
<input type="radio"/> グラフィック	<input type="radio"/> なにもしない

キー押下による画面ハードコピーを無効にする

- ・ チェックボックスがオンの場合、COBOL アプリケーションから PrtScr キーを押下しても画面ハードコピーは出力されません。
- ・ チェックボックスがオフの場合、COBOL アプリケーションから PrtScr キーを押すと画面ハードコピーが出力されます。

2.6.22 画面ハードコピー出力モード

ファイル	CSV	キーボード	フォント	サウンド	その他
ウィンドウ1	ウィンドウ2	ウィンドウ3	動作	プリンタ	
<input type="checkbox"/> 先行キー入力を可能にする(P)					
<input type="checkbox"/> 英小文字を英大文字に変換する(C)					
<input type="checkbox"/> 日本語入力時の半角スペースを全角スペースに変換する(S)					
<input type="checkbox"/> 混在入力項目で日本語入力システムを起動する(M)					
<input type="checkbox"/> キー押下による画面ハードコピーを無効にする(I)					
<input type="checkbox"/> STOP定数/実行時エラーを画面の最下行の1行下に表示する(U)					
<input type="checkbox"/> 数字項目入力時の桁数チェック方式を変更する(N)					
画面ハードコピー(H)			カナ入力(K)		
<input type="radio"/> 文字のみ			<input checked="" type="radio"/> カタカナ		
<input checked="" type="radio"/> 文字と罫線			<input type="radio"/> ローマ字		
<input type="radio"/> グラフィック			<input type="radio"/> なにもしない		

画面ハードコピー出力モード

COBOL ウィンドウの画面ハードコピーをとる際の、出力モードを指定します。

[文字のみ]

COBOL ウィンドウの文字のみを、画面ハードコピーとして出力します。

[文字と罫線]

COBOL ウィンドウの文字と罫線を、画面ハードコピーとして出力します。

[グラフィック]

COBOL ウィンドウのビットマップイメージを、画面ハードコピーとして出力します。

【Standard】-CF オプションを指定して作成した COBOL アプリケーションでは、常に[グラフィック]モードでの出力となります。

2.6.23 カナ入力モード

ファイル	CSV	キーボード	フォント	サウンド	その他
ウインドウ1	ウインドウ2	ウインドウ3	動作	プリンタ	

☐ 先行キー入力を可能にする(P)
☐ 英小文字を英大文字に変換する(C)
☐ 日本語入力時の半角スペースを全角スペースに変換する(S)
☐ 混在入力項目で日本語入力システムを起動する(M)

☐ キー押下による画面ハードコピーを無効にする(I)
☐ STOP定数/実行時エラーを画面の最下行の1行下に表示する(U)
☐ 数字項目入力時の桁数チェック方式を変更する(N)

画面ハードコピー(H)

- ☐ 文字のみ
- ☒ 文字と罫線
- ☐ グラフィック

カナ入力(K)

- ☒ カタカナ
- ☐ ローマ字
- ☐ なにもしない

カナ入力モード

- ・ COBOL アプリケーションでの、INPUT MODE IS KANA-SHIFT 句が指定された項目にキャレットが位置付いた際の入力モードを指定します。

[カタカナ]

キーボードの設定をカナ入力の状態にします。

[ローマ字]

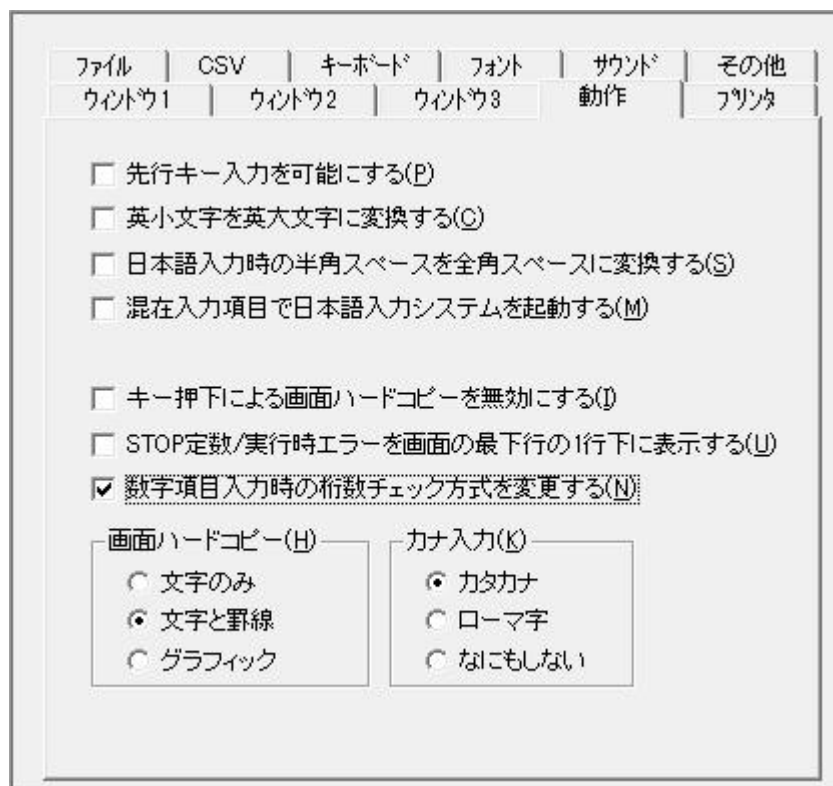
日本語入力システムを起動し、ローマ字入力、半角カタカナ変換の状態にします。

[なにもしない]

キーボードの状態の変更や日本語入力システムの起動を行いません。

- ・ カナ入力での動作指定を[カタカナ]に設定した場合、キーボードの設定がカナ入力モードに変更されますが、この変更は COBOL アプリケーション内でのみ有効となります。
カナインジケータランプのあるキーボードの場合、ランプの状態と実際の入力状態が異なる場合があります。

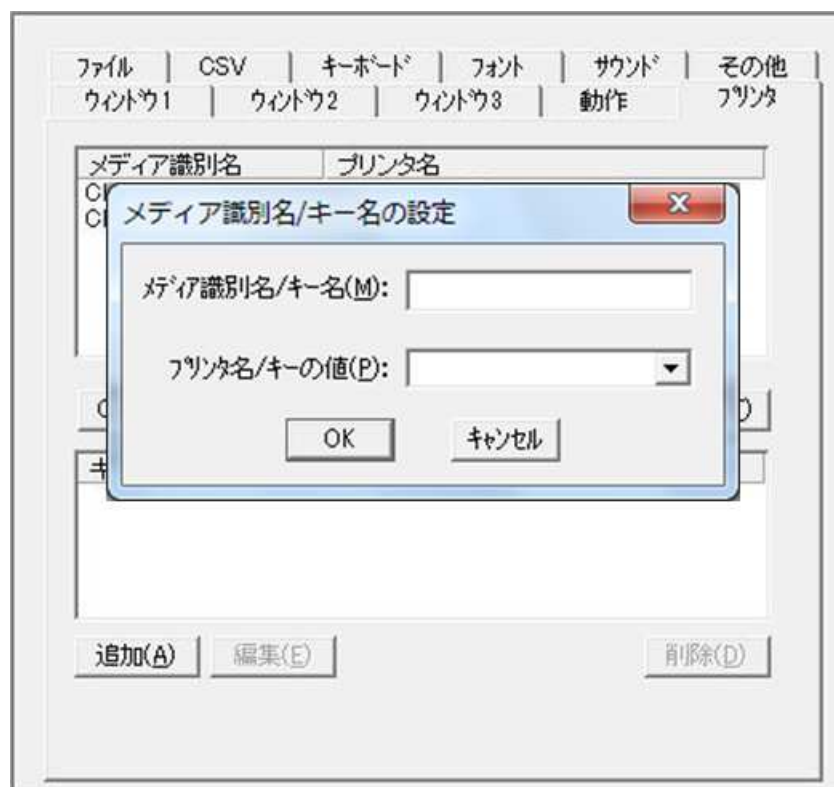
2.6.24 数字項目入力時の桁数チェック方式を変更するには



数字項目入力時の桁数チェック方式を変更する

- ・ チェックボックスがオンの場合、数字項目の桁数チェック方式を A-VX 互換とします。
入力データが不正のとき、ガイド領域に次のエラーを表示します。
Enter キーまたは小数点(.)入力時に整数部の桁数をチェックし、PICTURE 句の整数部の桁数より入力データの整数部の桁数が大きい場合、'?'エラーを表示します。
Enter キー入力時に小数部の桁数をチェックし、PICTURE 句の小数部の桁数より入力データの小数部の桁数が大きい場合、'?'エラーを表示します。
PICTURE 句が整数部のみ(小数部なし)の場合、小数点(.)の入力時に '?'エラーを表示します。
PICTURE 句が符号なしの場合、符号文字 '+' と '-' の入力時に 'NUMERIC' エラーを表示します。
上記 は Enter キーだけでなく値の取込方法が「カーソルの左」のキーが対象になります。値の取込方法については「2.6.39 値の取り込み方法を指定する」を参照してください。
- ・ チェックボックスがオフの場合、数字項目の桁数チェック方式を COBOL85 Pro 互換とします。

2.6.25 メディア識別名 / キー名を追加するには



メディア識別名 / キー名を追加する

- ・ [追加] ボタンをクリックすると、[メディア識別名 / キー名の設定] ダイアログボックスが表示されます。
- ・ メディア識別名またはキー名を入力します。
- ・ プリンタ名を選択するか、値のONかOFFを入力します。
- ・ [OK] ボタンをクリックすると、キー名が指定されていた場合は、キー名のリストボックスに、それ以外はメディア識別名のリストボックスに追加されます。

キー名の追加は[CBLAMRMODE を定義]ボタンおよび[THROUGHMODE を定義]ボタンにて実施することを推奨します。使用方法については、「2.6.28 COBOL アプリケーション実行環境 (プリンタ機能)の使用を指定する」または「2.6.29 プリンタ機能の PASSTHROUGH モードの使用を指定する」を参照してください。

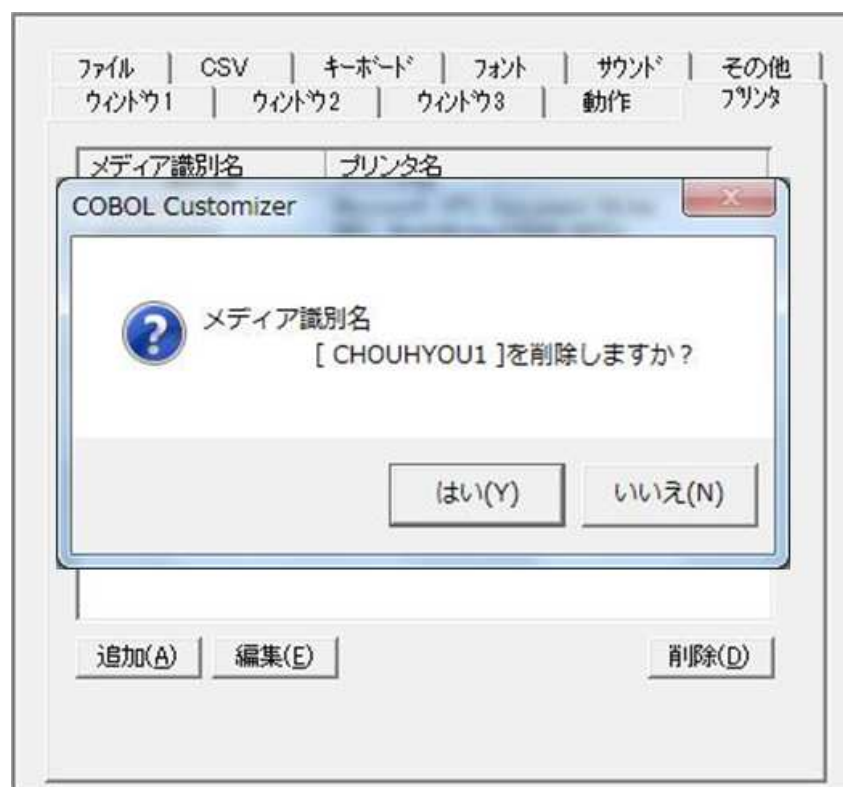
2.6.26 メディア識別名 / キー名を編集するには



メディア識別名 / キー名を編集する

- ・ リストボックスより、編集したいメディア識別名またはキー名を選択します。
- ・ [編集] ボタンをクリックするか、リストボックス上のメディア識別名またはキー名をダブルクリックします。
- ・ [メディア識別名 / キー名の設定] ダイアログボックスが表示されます。
- ・ メディア識別名やプリンタ名またはキー名や値を編集します。
- ・ [OK] ボタンをクリックすると、選択された項目が変更されます。

2.6.27 メディア識別名 / キー名を削除するには



メディア識別名 / キー名を削除する

- ・ リストボックスより、削除したいメディア識別名またはキー名を選択します。
- ・ [削除]ボタンをクリックすると削除を確認するメッセージボックスが表示されます。
- ・ [はい]ボタンをクリックすると、リストボックスからメディア識別名またはキー名が削除されます。

2.6.28 COBOL アプリケーション実行環境（プリンタ機能）の 使用を指定する

The screenshot shows a configuration window with a menu bar at the top containing: ファイル, CSV, キーボード, フォント, サウンド, その他. Below the menu bar are tabs: ウィンドウ1, ウィンドウ2, ウィンドウ3, 動作, and プリンタ. The 'プリンタ' tab is selected. The main area contains a table with two columns: 'メディア識別名' and 'プリンタ名'. The table has two rows: 'CHOUHYOU1' with 'Microsoft XPS Document Writer' and 'CHOUHYOU2' with 'NEC MultiWriter8250N NPDL'. Below the table are two buttons: 'CBLAMRMODEを定義(C)' and 'THROUGHMODEを定義(T)'. Below these buttons is another table with two columns: 'キー名' and '値'. This table has one row: 'CBLAMRMODE' with 'ON'. At the bottom are three buttons: '追加(A)', '編集(E)', and '削除(D)'.

メディア識別名	プリンタ名
CHOUHYOU1	Microsoft XPS Document Writer
CHOUHYOU2	NEC MultiWriter8250N NPDL

キー名	値
CBLAMRMODE	ON

アプリケーション実行環境（プリンタ機能）を使用して印刷するか、プリントマネージャを使用して印刷するかを指定します。

[CBLAMRMODE を定義] ボタンをクリックすると、値が ON の CBLAMRMODE のキー名をリストボックスに追加できます。このキーはアプリケーション実行環境（プリンタ機能）をインストールしている場合のみ有効です。

- ・ 値が ON の場合、アプリケーション実行環境（プリンタ機能）を使用して印刷します。プリンタ名の割り当てがアプリケーション実行環境のツール(AERO)で行われていることが必要です。
- ・ 値が OFF の場合、プリントマネージャを使用して印刷します。

2.6.29 プリンタ機能の PASSTHROUGH モードの使用を指定する



PASSTHROUGH モードを使用してプリンタシーケンスを発行する場合に THROUGHMODE を追加します。

[THROUGHMODE を定義] ボタンをクリックすると、値が ON の THROUGHMODE のキー名をリストボックスに追加できます。

- ・ 値が ON の場合、PASSTHROUGH モードを使用します。
- ・ 値が OFF の場合、PASSTHROUGH モードを使用しません。

2.6.30 RDB-STATUS への通知動作を指定する

ウィンドウ1	ウィンドウ2	ウィンドウ3	動作	プリンタ
ファイル	CSV	キーボード	フォント	サウンド
				その他

データベースの設定

☒ RDB-STATUSへの通知動作を指定する(R)
通知(A)
☐ する
☒ しない

ファイル排他制御の設定

☐ ファイル排他制御動作を指定する(E)

RDB-STATUS への通知動作を指定する

チェックボックスがオンの場合、COMMIT/ROLLBACK 文の実行結果を、特殊レジスタ RDB-STATUS に通知するかどうかを指定するラジオボタンが表示されます。

- ・ する場合、COMMIT/ROLLBACK 文の実行結果は、特殊レジスタ RDB-STATUS に通知されます。
- ・ しないの場合、COMMIT/ROLLBACK 文の実行結果は、特殊レジスタ RDB-STATUS に通知されません。

2.6.31 ファイル排他制御動作を指定する

ウィンドウ1	ウィンドウ2	ウィンドウ3	動作	プリンタ
ファイル	CSV	キーボード	フォント	サウンド その他

データベースの設定

☐ RDB-STATUSへの通知動作を指定する(R)

ファイル排他制御の設定

☒ ファイル排他制御動作を指定する(E)

待ち合わせ(W)

☒ する

☐ しない

ファイル排他制御動作を指定する

- ・ チェックボックスがオンの場合、ファイル排他制御による待ち合わせの動作を指定するラジオボタンが表示されます。
 - ・ するの場合、ファイル排他制御による待ち合わせをします。
 - ・ しないの場合、ファイル排他制御による待ち合わせをしません。
- ・ チェックボックスがオフの場合、ファイル排他制御による待ち合わせの動作は環境変数(COB_FILE_SHARING)またはコンパイラ動作既定値により決定されます。

2.6.32 行順ファイル（CSV 形式）に対する READ 文の実行時， 対応するセルのない基本項目の値を保持する

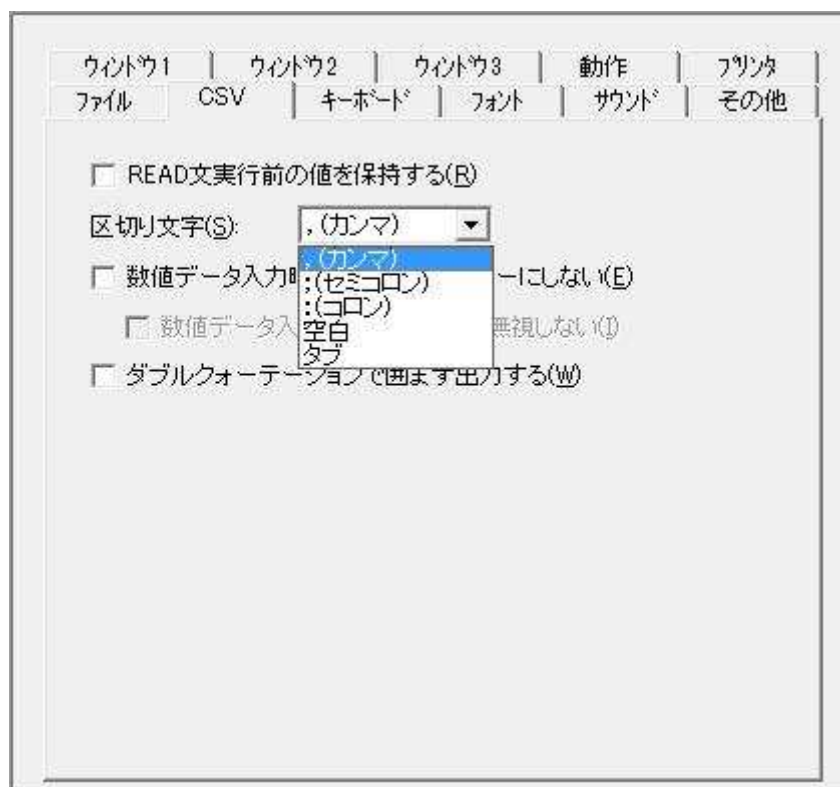
ウィンドウ1	ウィンドウ2	ウィンドウ3	動作	プリント
ファイル	CSV	キーボード	フォント	サウンド
その他				

☒ READ文実行前の値を保持する(B)
区切り文字(S): , (カンマ)
☐ 数値データ入力時の不正文字をエラーにしない(E)
☐ 数値データ入力時の不正文字を無視しない(I)
☐ ダブルクォーテーションで囲まず出力する(W)

行順ファイル(CSV 形式)に対する READ 文の実行時，対応するセルのない基本項目の値を READ 文の実行前の値で保持する。

- ・ チェックボックスがオンの場合，行順ファイル(CSV 形式)を READ したときに，対応するセルのないレコードの基本項目の値は READ 文の実行前から変更されません。
- ・ チェックボックスがオフの場合，行順ファイル(CSV 形式)を READ したときに，対応するセルのないレコードの基本項目の値は，レコードの基本項目の属性に従い，空白またはゼロが埋められます。

2.6.33 行順ファイル (CSV 形式) の区切り文字を指定する



行順ファイル (CSV 形式) の区切り文字を指定する。

- [,] 区切り文字としてカンマを使用します。
- [;] 区切り文字としてセミコロンを使用します。
- [:] 区切り文字としてコロンを使用します。
- [空白] 区切り文字として1バイト空白 (' 20 ' H)を使用します。
- [タブ] 区切り文字としてタブ (' 09 ' H)を使用します。

2.6.34 行順ファイル (CSV 形式) から外部 10 進項目に文字型 データを読み込む

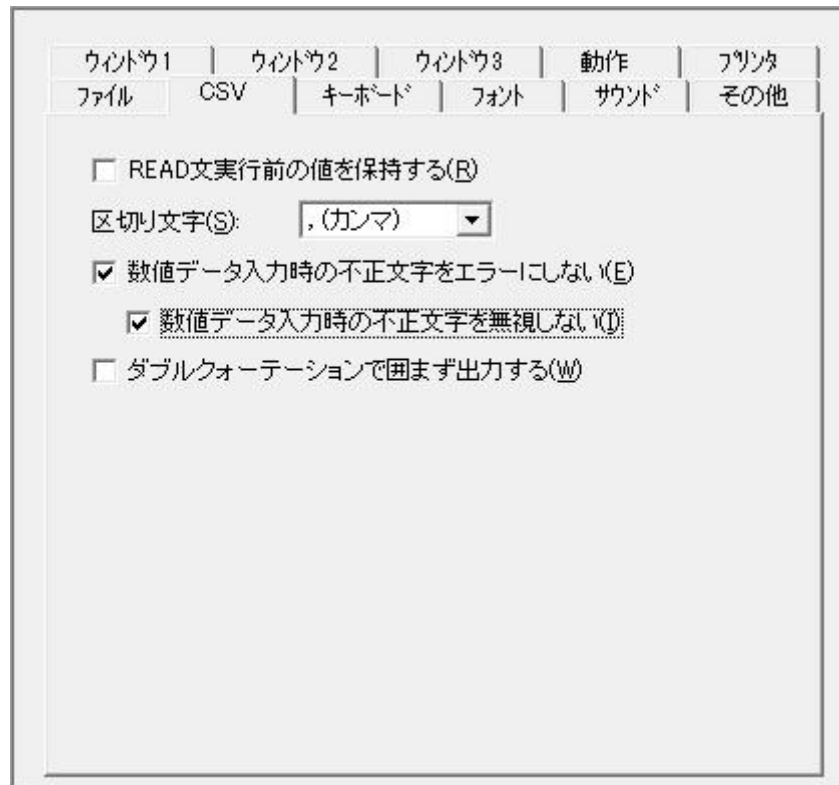
ユニット1	ユニット2	ユニット3	動作	プリンタ
ファイル	CSV	キーボード	フォント	サウンド
その他				

<input type="checkbox"/> READ文実行前の値を保持する(B)
区切り文字(S): ,(カンマ)
<input checked="" type="checkbox"/> 数値データ入力時の不正文字をエラーにしない(E)
<input type="checkbox"/> 数値データ入力時の不正文字を無視しない(I)
<input type="checkbox"/> ダブルクォーテーションで囲まず出力する(W)

行順ファイル (CSV 形式) から数値データ入力時の不正文字をエラーにしない。

- ・ チェックボックスがオンの場合、行順ファイル (CSV 形式) を読み込むときに、外部 10 進項目として定義されている基本項目に数字として有効でない文字が含まれていても、実行時エラーは出力されません。このときの動作は「数値データ入力時の不正文字を無視しない」チェックボックスの指定に従います。
- ・ チェックボックスがオフの場合、行順ファイル (CSV 形式) を読み込むときに、外部 10 進項目として定義されている基本項目に数字として有効でない文字が含まれていると、実行時エラーが出力されます。

「数値データ入力時の不正文字をエラーにしない」チェックボックスの指定



ウィンドウ1	ウィンドウ2	ウィンドウ3	動作	プリンタ
ファイル	CSV	キーボード	フォント	サウンド
その他				

☐ READ文実行前の値を保持する(B)

区切り文字(S): , (カンマ)

☒ 数値データ入力時の不正文字をエラーにしない(E)

☒ 数値データ入力時の不正文字を無視しない(I)

☐ ダブルクォーテーションで囲まず出力する(W)

行順ファイル(CSV 形式)から数値データ入力時の不正文字を無視しない。

- ・ チェックボックスがオンの場合、行順ファイル(CSV 形式)を読み込むときに、外部10進項目として定義されている基本項目に数字として有効でない文字が含まれていると、英数字項目と同様に入力されます。後続の処理において、この項目を基本項目として参照しようとする、実行時エラー(不正十進エラー等)となる可能性があります。
- ・ チェックボックスがオフの場合、行順ファイル(CSV 形式)を読み込むときに、外部10進項目として定義されている基本項目に数字として有効でない文字が含まれていると、その文字を読み飛ばし、数字として有効な文字のみを入力します。「¥1230」というデータを「1230」という数字データとして入力したい場合などに有効です。

2.6.35 行順ファイル (CSV 形式) へ文字型データをダブルコー テーションで囲まらずに出力する

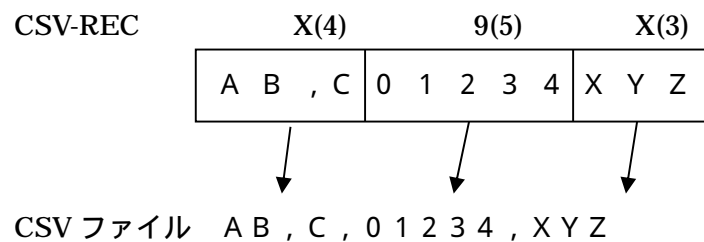
ユニット1	ユニット2	ユニット3	動作	プリンタ
ファイル	CSV	キーボード	フォント	サウンド
その他				

<input type="checkbox"/> READ文実行前の値を保持する(B)
区切り文字(S): , (カンマ)
<input type="checkbox"/> 数値データ入力時の不正文字をエラーにしない(E)
<input type="checkbox"/> 数値データ入力時の不正文字を無視しない(I)
<input checked="" type="checkbox"/> ダブルクォーテーションで囲まらずに出力する(W)

行順ファイル (CSV 形式) への出力で文字型データをダブルコーテーションで囲まない。

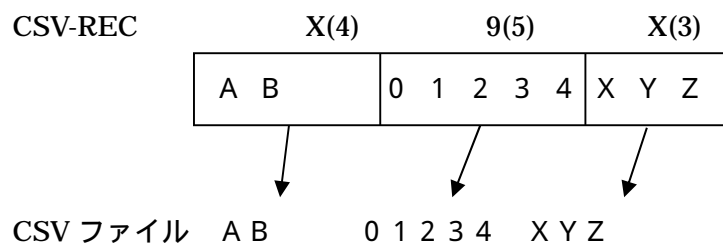
- ・ チェックボックスがオンの場合、行順ファイル (CSV 形式) に書き出すときに文字型データをダブルコーテーションで囲まらずに出力します。文字列中に区切り文字に使われている文字が含まれているデータを書き出すと、区切り文字との区別がつかなくなるので注意が必要です。

例) WRITE CSV-REC (区切り文字にカンマを指定したとき)



“ AB ” の後ろの “ , ” と区切り文字 “ , ” との区別がつかなくなる

例) WRITE CSV-REC (カスタマイザで区切り文字に空白を指定したとき)
 “ ”は空白を表しています。

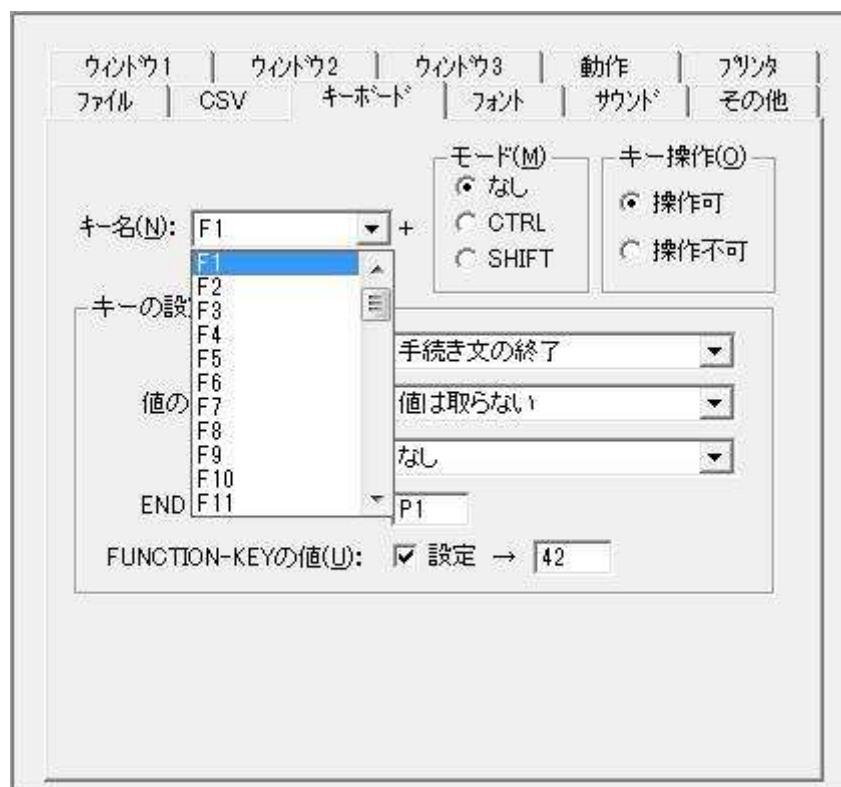


“ AB ” の後ろの “ ” と区切り文字 “ ” との区別がつかなくなる

- ・ チェックボックスがオフの場合, 行順ファイル(CSV 形式)に書き出すときに文字型データをダブルコーテーションで囲んで出力します。

注意 数字型データは常にダブルコーテーションで囲まず出力されます。

2.6.36 カスタマイズするキーを指定する



カスタマイズするキーを指定する

[キー名]

コンボボックスの中からカスタマイズの対象となるキー名を指定します。

[モード]

[キー名]に指定したキーと組み合わせるキーを指定します。

注意 キー名「_」のカスタマイズはできません。

2.6.37 キー操作を指定する

ウィンドウ1	ウィンドウ2	ウィンドウ3	動作	プリンタ
ファイル	CSV	キーボード	フォント	サウンド
			その他	

キー名(N): F1	+	モード(M) <input checked="" type="radio"/> なし <input type="radio"/> CTRL <input type="radio"/> SHIFT	キー操作(O) <input checked="" type="radio"/> 操作可 <input type="radio"/> 操作不可
------------	---	--	---

キーの設定

キーの動作(B):	手続き文の終了
値の取り込み方法(Q):	値は取らない
検査対象(S):	なし
END STATUSの値(E):	P1
FUNCTION-KEYの値(U):	<input checked="" type="checkbox"/> 設定 → 42

キー操作を指定する

[操作可]

選択されたキーセットの動作が設定できるようになります。

[操作不可]

選択されたキーセットの動作は設定できません。

2.6.38 キーの動作を指定する

ウインドウ1	ウインドウ2	ウインドウ3	動作	プリンタ
ファイル	CSV	キーボード	フォント	サウンド
その他				

キー名(N): +

モード(M)

☒ なし
☐ CTRL
☐ SHIFT

キー操作(O)

☒ 操作可
☐ 操作不可

キーの設定

キーの動作(B):

手続き文の終了

モード切り替え

リセット

次のフィールドへ

前のフィールドへ

手続き文の終了

値の取り込み方法(Q):

検査対象(S):

END STATUSの値(E):

P1

FUNCTION-KEYの値(U):

☒ 設定 →

キーの動作を指定する

- ・ ACCEPT 文実行中にキー押下時の動作を指定します。

設定できる動作は以下のとおりです。

- | | |
|------------|---|
| カーソル左 | ACCEPT 文実行時に表示されるカーソルを左に移動します。入力した文字を再入力するためにカーソル位置を変更する場合などに使用します。 |
| カーソル右 | ACCEPT 文実行時に表示されるカーソルを右に移動します。左に移動したカーソルを元の位置に戻す場合などに使用します。 |
| カーソルの左文字削除 | ACCEPT 文実行時に表示されている文字列のうち、カーソルの左側にある文字を削除します。間違った入力を行った時などでその文字を削除する場合などに使用します。 |
| カーソルの文字削除 | ACCEPT 文実行時に表示されている文字列のうち、カーソル位置にある文字を削除します。不要な文字を入力してしまった時などに、その位置までカーソルを移動して削除する場合などに使用します。 |
| クリア | 入力中の文字列を含めてその入力項目の領域に表示されている文字列を消去します。その項目に対する入力をやり直す場合などに使用します。 |
| モード切り替え | 実行中の入力項目への入力モードを変更します。キーを押下する度に<上書>モードと<挿入>モードが切り替わります。 |

リセット エラー解除待ち状態を元に戻します。OVERFLOW エラーなどが発生した場合などにエラー状態を解除するために使用します。

次のフィールドへ 現在の入力項目への入力を終了し、次の入力項目への入力になります。集団項目への入力でない場合や集団項目中の最後の入力項目の場合は ACCEPT 文を終了することになります。

前のフィールドへ 現在の入力項目への入力を終了し、1つ前の入力項目への入力に戻ります。集団項目への入力でない場合や集団項目中の先頭の入力項目の場合は ACCEPT 文を終了することになります。

手続き文の終了 ACCEPT 文を終了します。集団項目への ACCEPT 文実行中でも終了することになります。

- ・ この項目に、
 - ・ 次のフィールドへ
 - ・ 前のフィールドへ
 - ・ 手続き文の終了
 - のいずれかを選択すると、
 - ・ [値の取り込み方法]
 - ・ [検査対象]
 - ・ [END STATUS の値]
 - ・ [FUNCTION-KEY の値]
- が設定できます。

2.6.39 値の取り込み方法を指定する

値の取り込み方法を指定する

- ・ 入力項目への入力を終了する場合に、入力した(または表示されている)文字列をどのように作業領域に取り込むかを指定します。
それぞれの指定による動作は以下のとおりになります。

値はとらない	入力された文字列を一切取り込みません。作業領域の値はACCEPT 文実行前から変更されません
カーソルの左	入力項目への入力終了キーが押下された時点でカーソルがあった位置より左の文字列が作業領域に取り込まれます。カーソル位置およびカーソルより右の文字列は無効となります。
画面表示から	入力項目への入力終了キーが押下された時点でのカーソル位置に関係なく、入力項目の領域に表示されている文字列のすべてを作業領域に取り込みます。
- ・ この項目は、[キーの動作]で、
 - ・ 次のフィールドへ
 - ・ 前のフィールドへ
 - ・ 手続き文の終了
 のいずれかを指定したときに設定できます。

2.6.40 検査対象を指定する

ウィンドウ1	ウィンドウ2	ウィンドウ3	動作	プリンタ
ファイル	CSV	キーボード	フォント	サウンド
その他				

キー名(N): +

モード(M)

☒ なし
☐ CTRL
☐ SHIFT

キー操作(O)

☒ 操作可
☐ 操作不可

キーの設定

キーの動作(B):

手続き文の終了

▼

値の取り込み方法(Q):

値は取らない

▼

検査対象(S):

なし

▼

END STATUSの値(E):

なし

▼

FUNCTION-KEYの値(U):

ADVANCE

BTAB

ADVANCE and BTAB

検査対象を指定する

- キー押下時の動作として CHECK 句の対象とするかどうかを指定します。対象となるのは ADVANCE と BTAB のみです。それぞれの指定が行われた場合の動作は以下のとおりです。

なし	CHECK 句の指定で ADVANCE や BTAB の指定がある入力項目であっても、この指定のあるキーが押下された場合にはそのチェックが行われません。
ADVANCE	CHECK 句の指定で ADVANCE 指定がある入力項目の場合、この指定のあるキーが押下されますと ADVANCE エラーとなります。
BTAB	CHECK 句の指定で BTAB 指定がある入力項目の場合、この指定のあるキーが押下されますと BTAB エラーとなります。
- この項目は、[キーの動作]で、
 - 次のフィールドへ
 - 前のフィールドへ
 - 手続き文の終了
 のいずれかを指定したときに設定できます。

2.6.41 END STATUS の値を指定する

ウィンドウ1	ウィンドウ2	ウィンドウ3	動作	プリンタ
ファイル	CSV	キーボード	フォント	サウンド
その他				

キー名(N): F1	+	モード(M) <input checked="" type="radio"/> なし <input type="radio"/> CTRL <input type="radio"/> SHIFT	キー操作(O) <input checked="" type="radio"/> 操作可 <input type="radio"/> 操作不可
------------	---	--	---

キーの設定

キーの動作(B):	手続き文の終了
値の取り込み方法(Q):	値は取らない
検査対象(S):	なし
END STATUSの値(E):	P1
FUNCTION-KEYの値(U):	<input checked="" type="checkbox"/> 設定 → 42

END STATUS の値を指定する

- ・ 選択されたキーセットに対する入力終了状態(END STATUS)の値を設定できます。
- ・ この項目は、[キーの動作]で、
 - ・ 次のフィールドへ
 - ・ 前のフィールドへ
 - ・ 手続き文の終了

のいずれかを指定したときに設定できます。

2.6.42 FUNCTION-KEY の値を指定する

ウィンドウ1	ウィンドウ2	ウィンドウ3	動作	プリンタ
ファイル	CSV	キーボード	フォント	サウンド
			その他	

キー名(N): +

モード(M)

☒ なし
☐ CTRL
☐ SHIFT

キー操作(O)

☒ 操作可
☐ 操作不可

キーの設定

キーの動作(B):

値の取り込み方法(Q):

検査対象(S):

END STATUSの値(E):

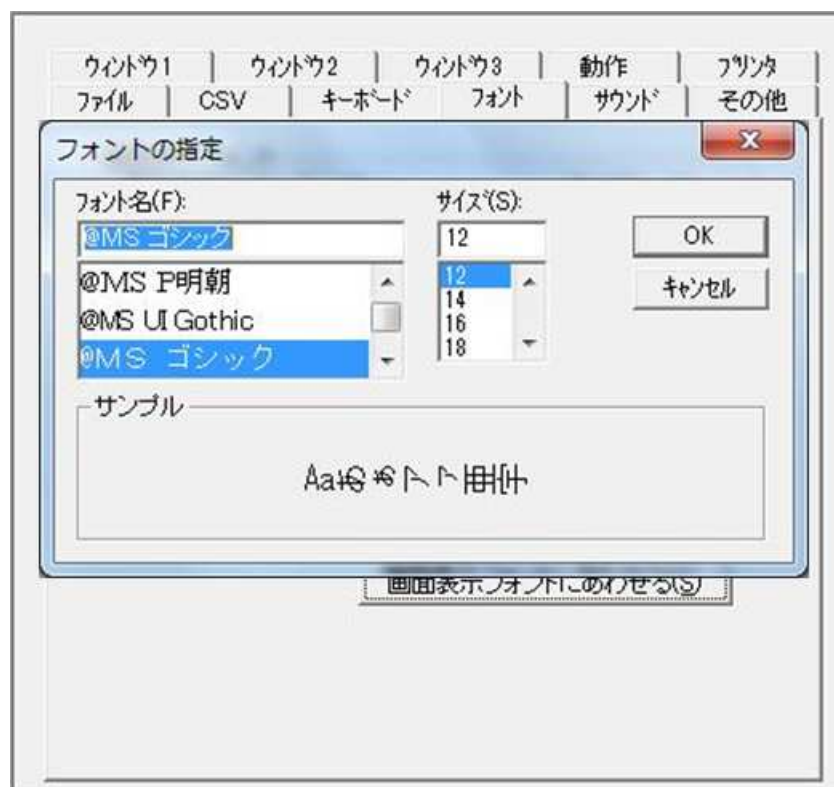
FUNCTION-KEYの値(U):

☒ 設定 →

FUNCTION-KEY の値を指定する

- ・ ACCEPT 文が実行中でない状態で該当キーが押下された場合、これを機能割込みキーとして認識し、特殊レジスタ FUNCTION-KEY に値を設定するかどうかを指定します。"設定"チェックボックスがチェックされている場合、このキーは機能割込みキーとして認識されます。これらのキーが ACCEPT 文が実行中でない場合に押下された場合、特殊レジスタ FUNCTION-KEY に指定された値が設定されます。ACCEPT 文実行中には機能割込みキーとしての動作が行われません。
- ・ 選択されたキーセットに対する機能キー (FUNCTION-KEY) の値を設定できます。値は、[設定]チェックボックスを設定した場合にのみ設定できます。
- ・ この項目は、[キーの動作]で、
 - ・ 次のフィールドへ
 - ・ 前のフィールドへ
 - ・ 手続き文の終了
 のいずれかを指定したときに設定できます。

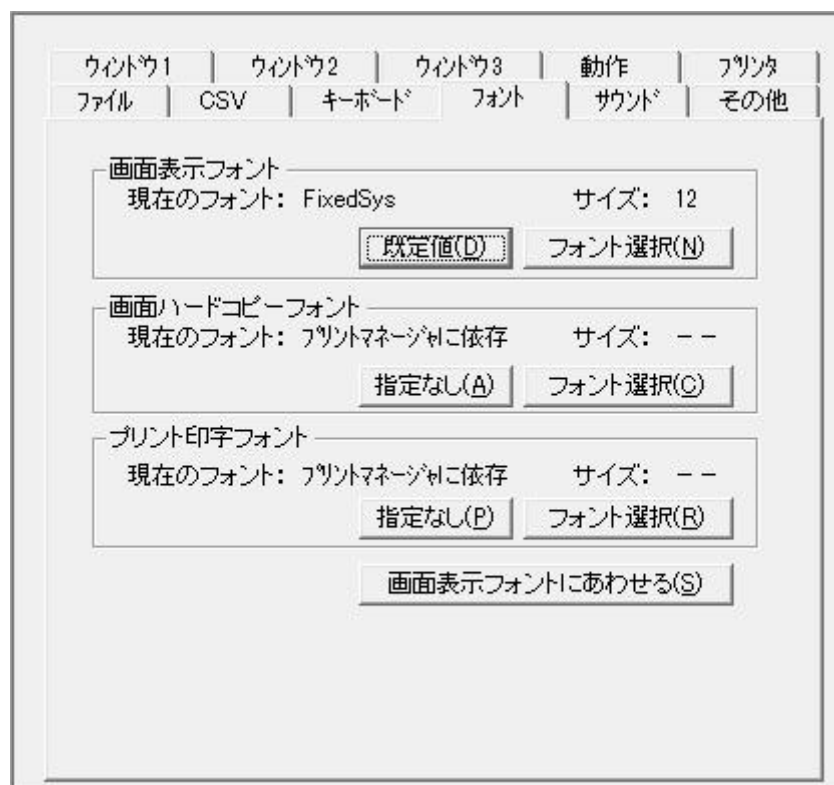
2.6.43 画面表示フォントを変更するには



画面表示フォントを変更する

- ・ 画面表示フォントの[フォント選択]ボタンをクリックします。
- ・ [フォントの指定] ダイアログボックスが表示されます。
- ・ [フォント名]を選択するとフォントが変更できます。
- ・ [サイズ]を選択するとサイズが変更できます。

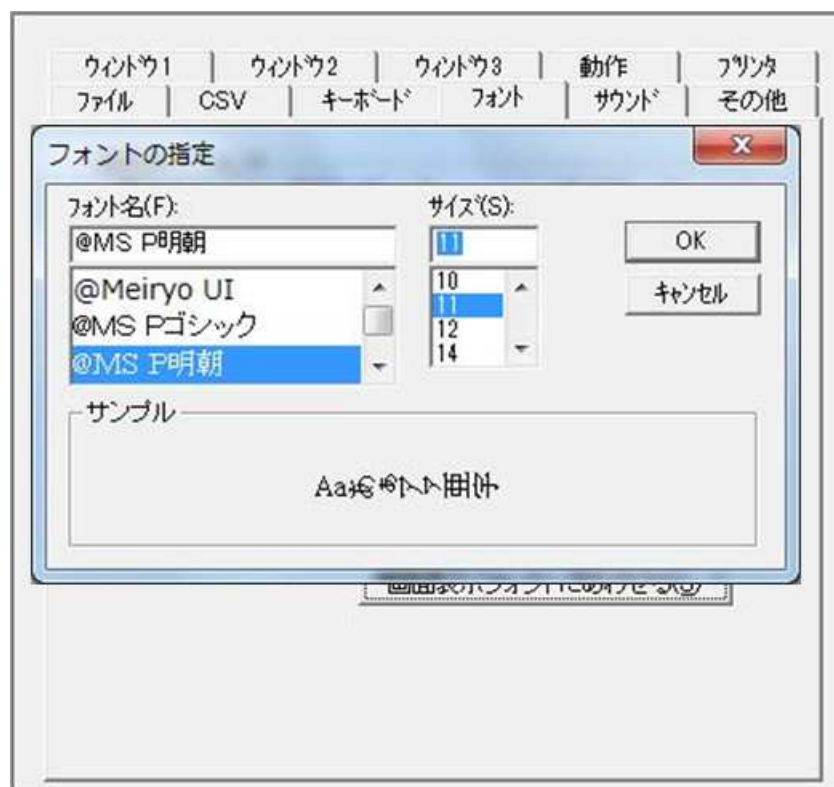
2.6.44 画面表示フォントの指定を既定値に戻すには



画面表示フォントを既定値に戻す

- ・ [既定値] ボタンをクリックするとフォントの指定は FixedSys になります。

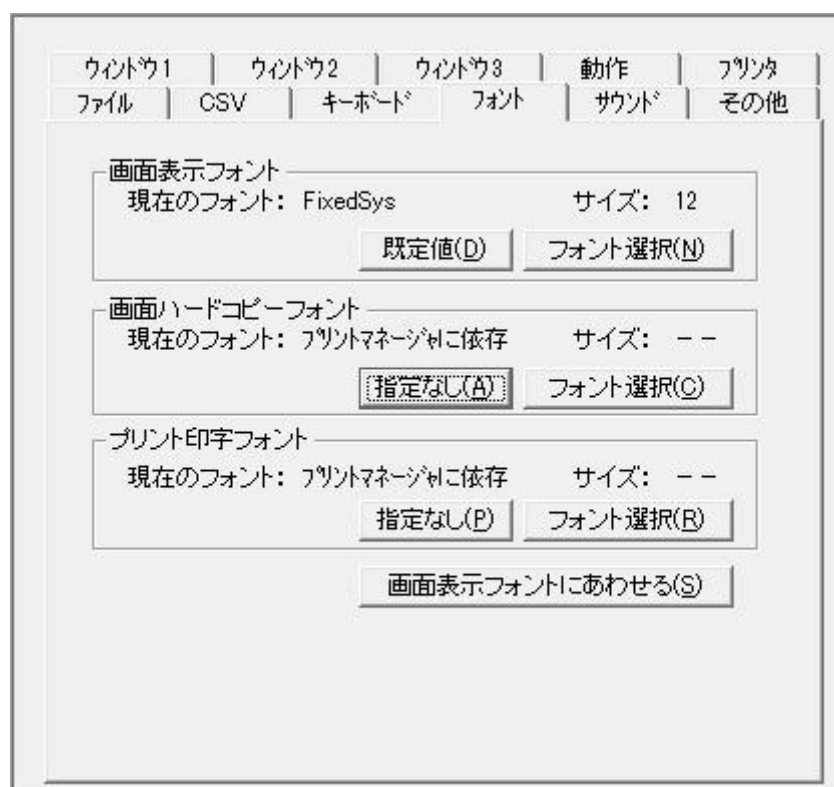
2.6.45 画面ハードコピーフォントを変更するには



画面ハードコピーフォントを変更する

- ・ 画面ハードコピーフォントの[フォント選択]ボタンをクリックします。
- ・ [フォントの指定] ダイアログボックスが表示されます。
- ・ [フォント名]を選択するとフォントが変更できます。
- ・ [サイズ]を選択するとサイズが変更できます。

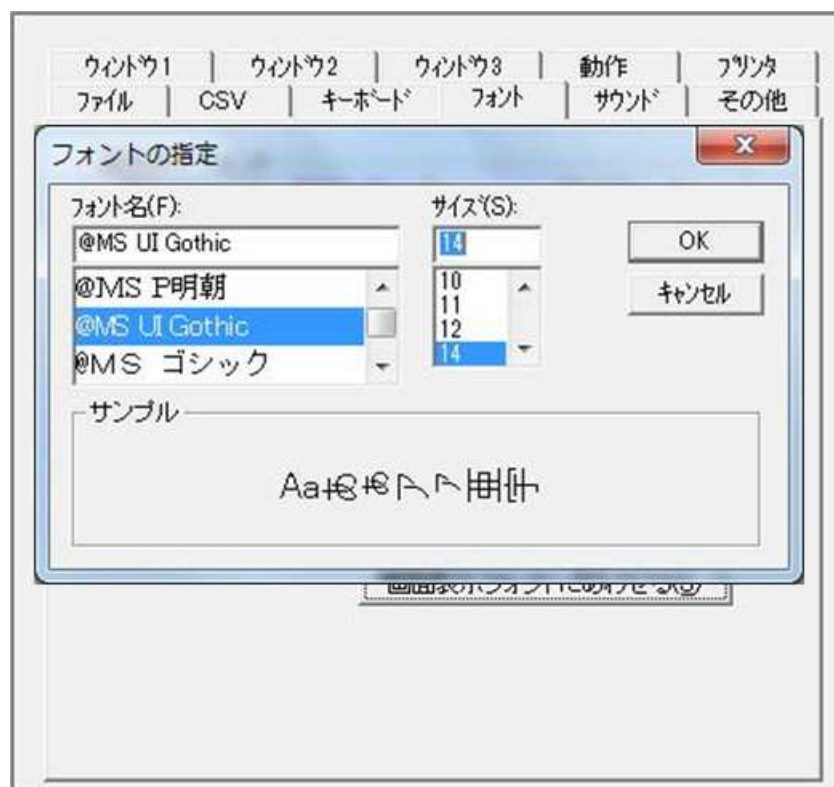
2.6.46 画面ハードコピーフォントを指定なしに戻すには



画面ハードコピーフォントを指定なしに戻す

- ・ 画面ハードコピーフォントの[指定なし]ボタンをクリックするとフォントの指定はなくなり、プリントマネージャに依存されます。

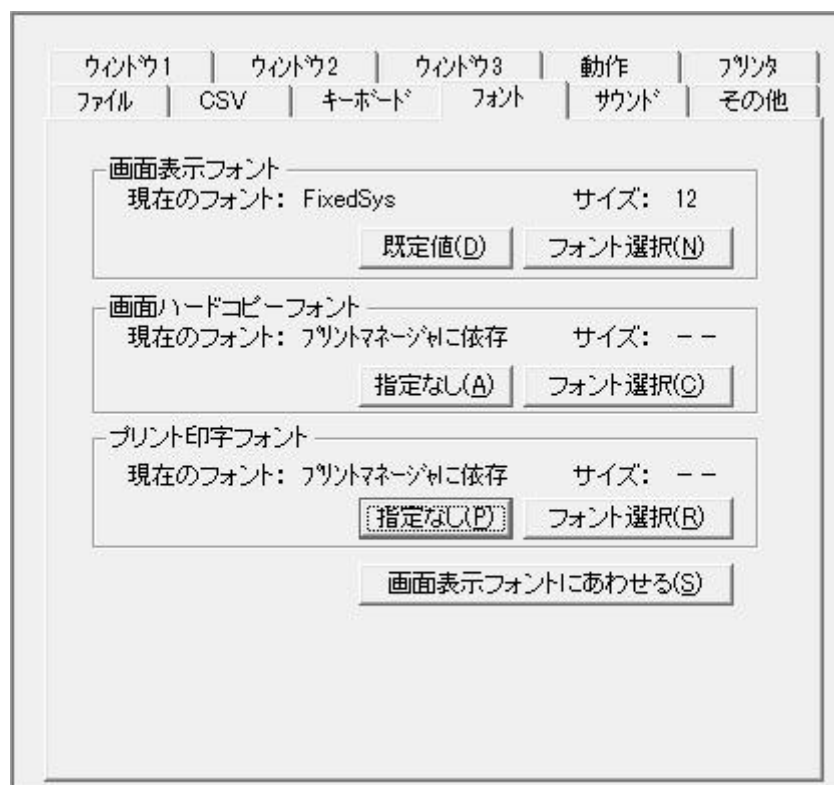
2.6.47 プリント印字フォントを変更するには



プリント印字フォントを変更する

- ・ プリント印字フォントの[フォント選択]ボタンをクリックします。
- ・ [フォントの指定] ダイアログボックスが表示されます。
- ・ [フォント名]を選択するとフォントが変更できます。
- ・ [サイズ]を選択するとサイズが変更できます。

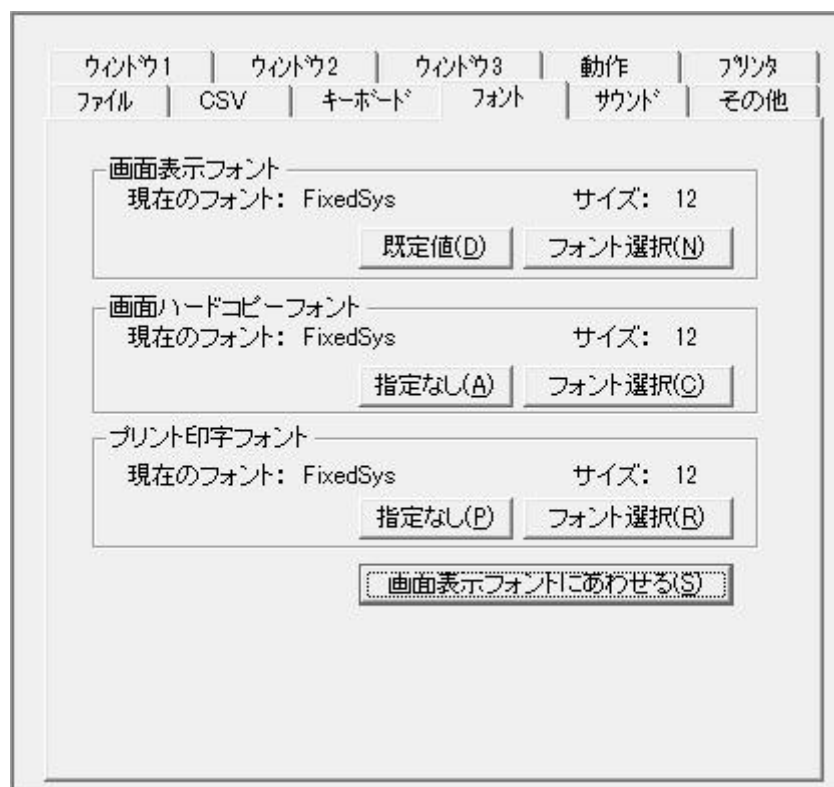
2.6.48 プリント印字フォントを指定なしに戻すには



プリント印字フォントを指定なしに戻す

- ・ プリント印字フォントの[指定なし]ボタンをクリックするとフォントの指定はなくなり、プリントマネージャに依存されます。

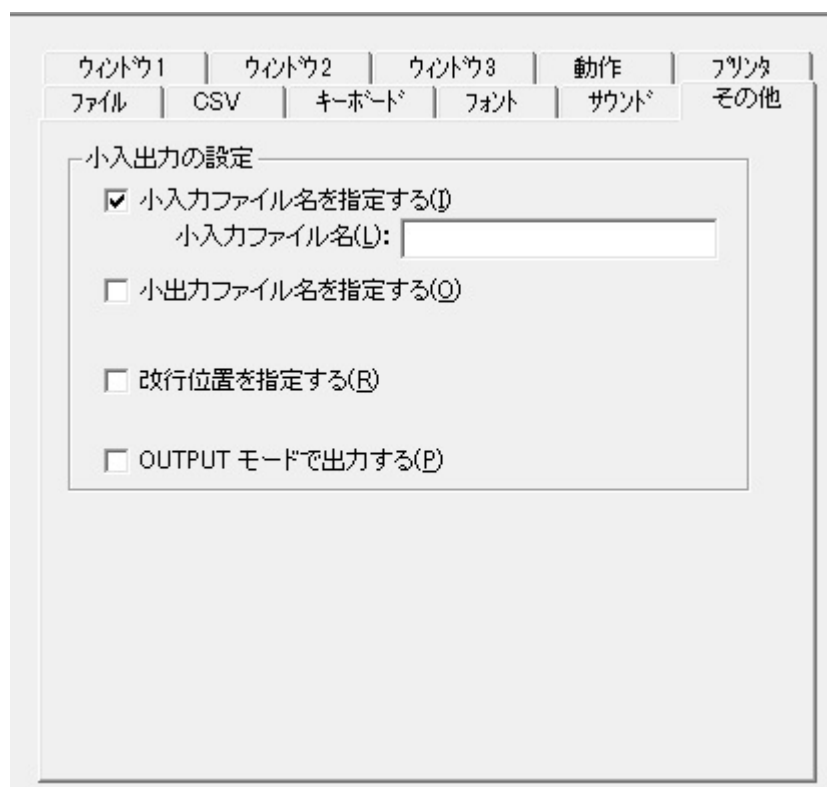
2.6.49 画面表示フォントにあわせるには



画面表示フォントにあわせる

- ・ [画面表示フォントにあわせる] ボタンをクリックすると画面ハードコピーフォントとプリント印字フォントの指定が画面表示フォントと同じ内容になります。

2.6.50 ACCEPT 文の小入力ファイル名を指定するには



小入力ファイル名を指定する

- ・ チェックボックスがオンの場合、エディットボックスが表示され、小入力ファイル名を入力することができます。
- ・ チェックボックスがオフの場合、小入力ファイル名は [プログラム名].in になります。

2.6.51 DISPLAY 文の小出力ファイル名を指定するには

ウィンドウ1	ウィンドウ2	ウィンドウ3	動作	プリント
ファイル	CSV	キーボード	フォント	サウンド
その他				

小入出力の設定

☐ 小入力ファイル名を指定する(I)

☒ 小出力ファイル名を指定する(O)
小出力ファイル名(S):

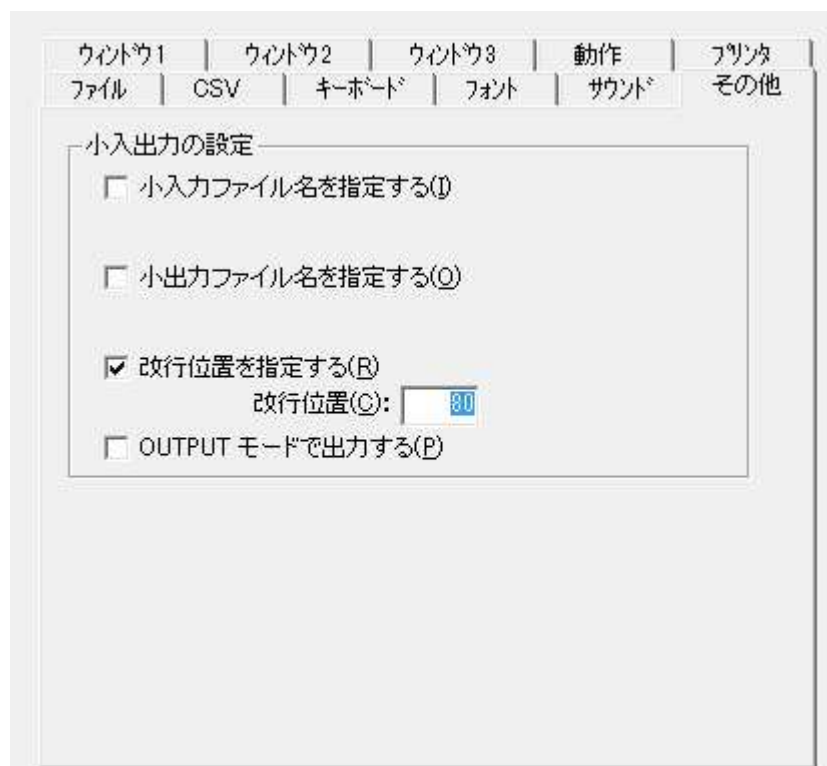
☐ 改行位置を指定する(R)

☐ OUTPUT モードで出力する(P)

小出力ファイル名を指定する

- ・ チェックボックスがオンの場合、エディットボックスが表示され、小出力ファイル名を入力することができます。
- ・ チェックボックスがオフの場合、小出力ファイル名は [プログラム名] + [プロセスID].out になります。

2.6.52 DISPLAY 文で出力する改行位置を指定するには



改行位置を指定する

- ・ チェックボックスがオンの場合、エディットボックスが表示され、改行位置を入力することができます。
- ・ チェックボックスがオフの場合、データ長分出力されます。

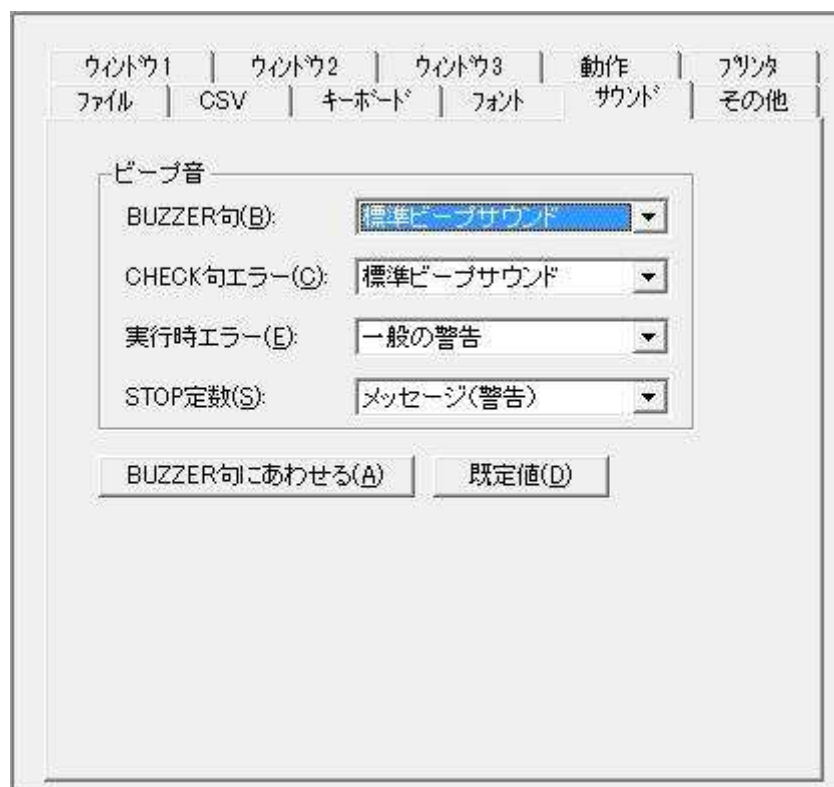
2.6.53 DISPLAY 文の出力モードに OUTPUT モードを指定する には

ウインドウ1	ウインドウ2	ウインドウ3	動作	プリンタ
ファイル	CSV	キーボード	フォント	サウンド
小入出力の設定				
<input type="checkbox"/> 小入力ファイル名を指定する(I)				
<input type="checkbox"/> 小出力ファイル名を指定する(O)				
<input type="checkbox"/> 改行位置を指定する(R)				
<input checked="" type="checkbox"/> OUTPUT モードで出力する(P)				

OUTPUT モードで出力する

- ・ チェックボックスがオンの場合、OUTPUT モードとなり、上書きされます。
- ・ チェックボックスがオフの場合、EXTEND モードとなり、追加書き込みになります。
(COBOL85 Pro 互換とする場合、チェックボックスをオンにしてください。)

2.6.54 鳴動ビープ音を変更するには



おのこののビープ音鳴動時のビープ音に対応するサウンドの種別を変更する

- ・ 変更したい COBOL アプリケーション実行時の動作に対して、鳴動ビープ音のサウンド種別をコンボボックスより選択し、変更します。変更可能なサウンドは以下のとおりです。

- ・ 標準ビープサウンド
- ・ メッセージ(情報)
- ・ メッセージ(警告)
- ・ メッセージ(問い合わせ)
- ・ システムエラー
- 一般の警告音

上記の標準ビープサウンド以外のサウンドは、コントロールパネルのサウンドより、サウンドファイルを変更することによりビープ音に対応するサウンドを変更することができます。

2.6.55 鳴動ビープ音を BUZZER 句にあわせるには

ウインドウ1	ウインドウ2	ウインドウ3	動作	プリンタ
ファイル	CSV	キーボード	サウンド	その他

ビーブ音

BUZZER句(B):

標準ビーブサウンド ▼

CHECK句エラー(C):

標準ビーブサウンド ▼

実行時エラー(E):

標準ビーブサウンド ▼

STOP定数(S):

標準ビーブサウンド ▼

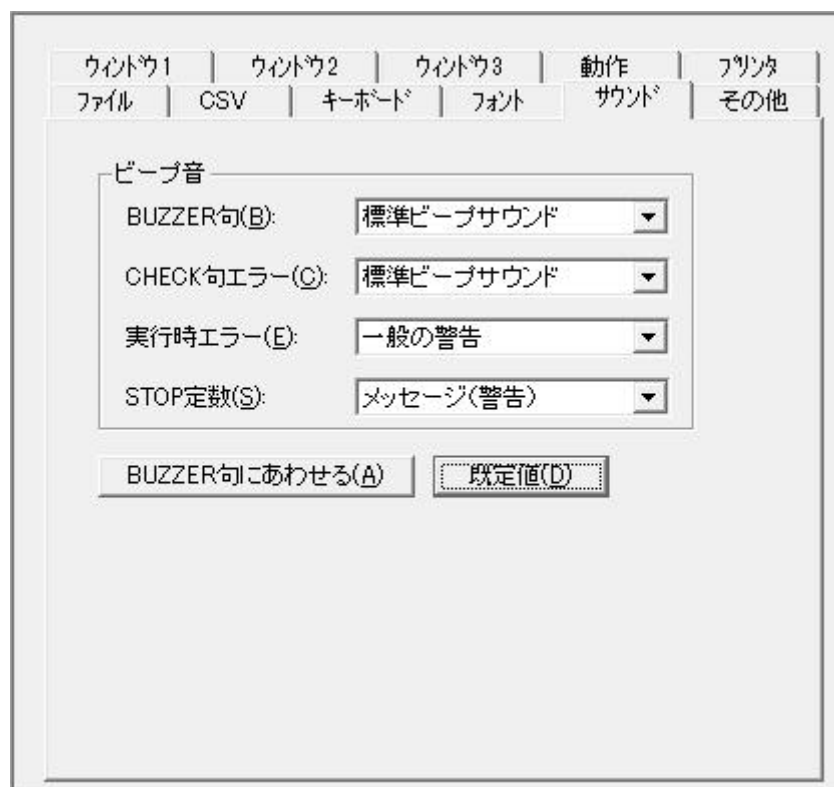
BUZZER句にあわせる(A)

既定値(D)

BUZZER 句にあわせる

- ・ [BUZZER 句にあわせる] ボタンをクリックすると、おのこののビーブ音鳴動時のビーブ音に対応するサウンドの種別の指定が、BUZZER 句と同じ内容になります。

2.6.56 鳴動ビープ音を既定値に戻すには



おのこののビープ音鳴動時のビープ音に対応するサウンドの種別を既定値に戻す

- ・ [既定値]ボタンをクリックすると、おのこののビープ音鳴動時のビープ音に対応するサウンドの種別の指定が、既定値に戻ります。なお、既定値は以下のとおりです。

BUZZER 句	標準ビープサウンド
CHECK 句エラー	標準ビープサウンド
実行時エラー	一般の警告音
STOP 定数	メッセージ(警告)

2.7 カスタマイズ可能な COBOL アプリケーションとは

COBOL アプリケーションがカスタマイズされているか否か、どのカスタマイズ情報を使用するかは、「2.2 アプリケーションカスタマイズの適用範囲」に記載した規則に従って、調べられます。そのうち、ユーザ指定のレジストリを使い、アプリケーション個別にカスタマイズする場合は、メインプログラムの翻訳時にレジストリのキー名指定（-Yr）コンパイルオプションを指定する必要があります。指定方法については「4.4レジストリ」を参照してください。

-Yr コンパイルオプションを指定して翻訳すると、COBOL コンパイラはオブジェクト中にレジストリのキー名を埋め込みます。そのキー名に対して、COBOL アプリケーションカスタマイザを使用してカスタマイズ情報を設定できます。

2.8 カスタマイズファイルとは

COBOL アプリケーションをカスタマイズした情報をファイルに保存したものです。本ファイルを使用することにより、以下のことが可能です。

1. 他のマシンに同じ COBOL アプリケーションのカスタマイズ情報を設定することができます。
2. 他の COBOL アプリケーションに同じカスタマイズ情報を設定することができます。

カスタマイズファイルの構造

カスタマイズファイルは以下の構造です。

一般形式

```
[OWNER]
Unit=[USER/MACHINE]
Owner=[ユーザ指定レジストリ名]
```

Unit : USER または MACHINE を設定します。それぞれ ユーザ単位 / マシン単位を意味します。

存在しない、あるいは誤っている場合は USER が設定されたものとします。

Owner : 設定するレジストリを設定します。 -Yr と同じものか、既定値の場合は「NEC¥CBLAP default」を指定します。

存在しない、あるいは誤っている場合は、設定を行いません。

なお、上記の指定は COBOL アプリケーションカスタマイザから出力することにより、自動的に付加さ

れています。

コマンドラインから登録する場合は、上記の記述がないと、カスタマイズファイルと認められません。

カスタマイズファイルを編集する場合は、上記の記述のみ編集可能です。

他の設定を編集した場合は、読み込めなくなる場合があります。

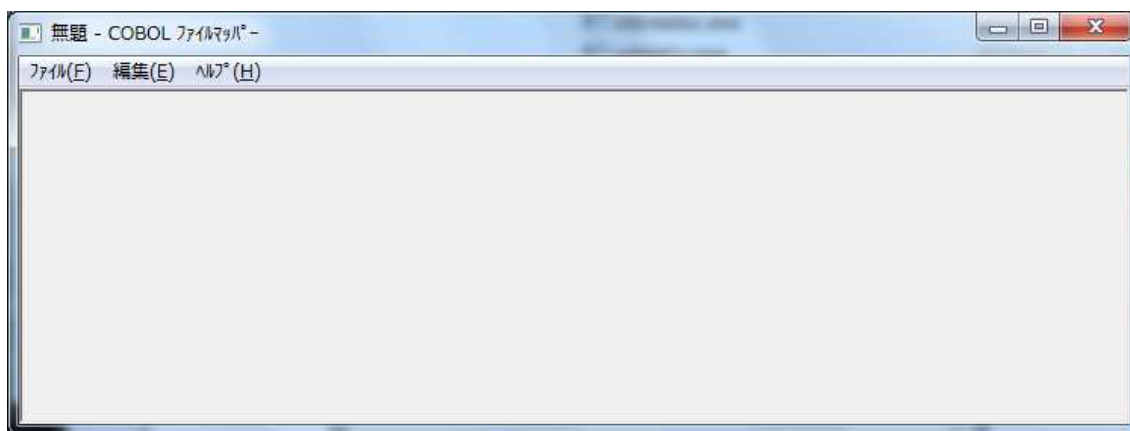
第3章 COBOL ファイルマッパーの使い方

【Enterprise for Windows】【Standard】COBOL ファイルマッパーは、Enterprise Edition for Windows 版、Standard Edition 版で使用できます。ただし、Enterprise Edition for Windows 版の標準動作モードでは、デバイスアドレスはメモ扱いです。

3.1 概要

3.1.1 ファイルマップ機能の概要

ファイルマップ機能とは、COBOL アプリケーションが使用するデータファイル(データベースも含む)のファイル種別(COBOL 独自、IFAS FILE、IFASPRO RDB、COBOL DB アクセス等)や実ファイル名等を指定し、一元管理する機能です。本機能を使用することで、COBOL プログラム上の記述とファイルの所在等を分離でき、システムの保守性向上や A-VX 上からの移行性向上にも効果があります。COBOL ファイルマッパーを起動すると、以下のウィンドウが表示されます。



設定は、[ユーザ単位]または[マシン単位]を切り替えることで、それぞれの単位での設定が行えます。

3.1.2 ファイルマップ指定の適用範囲

COBOL ファイルマッパーの指定方法によるファイルマップ情報の適用範囲を以下に示します。

(1) ユーザ単位

現在のユーザで実行する COBOL アプリケーションすべてに対して有効になるようにファイルマップ指定を行います。この指定は、他のユーザでは無効です。

(2) マシン単位

使用するマシン単位で実行する COBOL アプリケーションすべてに対して有効になるようにファイルマップ指定を行います。

COBOL アプリケーションは、実行時に上記の項番(1) (2)の順にファイルマップ指定の検索を行い、どれかを発見したところで検索を終了します。

注意

COBOL WS エミュレータから、COBOL アプリケーションを起動した場合に有効となるファイルマップ情報は、COBOL WS エミュレータで接続したサーバ上で設定されたファイルマップ情報となります。そのとき、ユーザ単位で設定されたファイルマップ情報が有効となるかあるいはマシン単位で設定されたファイルマップ情報が有効となるかは、サーバ上のサービス「WS-EML_SERVICE」の設定に依存します。「WS-EML_SERVICE」の設定ダイアログで「ログオン」の項目がシステムアカウント(既定値)の場合は、マシン単位のファイルマップ情報のみが有効となります。ユーザ単位のファイルマップ情報を有効としたい場合は、「ログオン」の項目に有効とするユーザのアカウントを設定する必要があります。この場合、そのユーザでのファイルマップ情報の設定が必要です。なお、「WS-EML_SERVICE」の設定に関する詳細は、『COBOL WS エミュレータ 説明書』を参照してください。

3.1.3 COBOL プログラムの指定とファイルマップ指定

COBOL プログラム上で内部ファイルIDとデバイスとして設定されている名称をキーとして、COBOL ファイルマッパーにより、ファイル種別・実ファイル名・レコード長・ファイル編成・注釈を定義する機能です。ファイルマップ指定されていないファイルは、内部ファイルIDを実ファイル名とし、ファイル種別は動作既定値により決定します。

【Standard】Standard Edition 版では、-Cf オプションを指定して、順編成の大容量記憶ファイルのファイル種別を COBOL 独自形式に変更することができます。

【Standard】Standard Edition 版では、-Cg オプションを指定して、相対編成の大容量記憶ファイルのファイル種別を COBOL 独自形式に変更することができます。

【Standard】Standard Edition 版では、-Ch オプションを指定して、索引編成の大容量記憶ファイルのファイル種別を ISAM 形式に変更することができます。

ファイルマップ機能を使用するには、以下の制約がありますので、注意してください。

指定項目	入力可能桁数	入力可能文字
------	--------	--------

内部ファイルID	200 桁	“(ダブルコーテーション),=(イコール)以外の記号文字,英数字,カナ文字,日本語のみ。それ以外の文字が指定された場合の処理は保証しません。
実ファイル名	259 桁	システムのロングファイル名形式に従います。
レコード長		0 ~ 65535 までの数字
注釈	256 桁	特に規定しません。

(日本語文字は,1文字2桁として計算してください)

内部ファイルIDの桁数が 200 桁以上の場合は,ファイルマップ指定がされていないと認識し,内部ファイルIDを実ファイル名とし,ファイル種別は動作既定値により決定します。

COBOL アプリケーション動作時,ファイルマップ情報を使用して,以下の処理を行います。(デバイスタイプが MSD , RDB のみ)

[ファイルマップ指定の検索方法]

- * COBOL プログラム中の VALUE OF IDENTIFICATION 句の値と SELECT 句のデバイス指定をキーに,ファイルマップ指定の適用範囲の規則に従い,ファイルマップ指定を検索します。
- * デバイスアドレスがファイルマップ情報で MSD000 ~ MSD009 または MSD990 ~ MSD998 に関連付けされている際の検索順序は,以下のとおりです。

デバイスアドレス	検索順序	
MSDnnn	(1)	MSDnnn として関連付けされているファイルマップ情報
MSD999	(1)	MSD990 ~ MSD998 として関連付けされているファイルマップ情報を順次
省略	(1)	MSD000 ~ MSD009 として関連付けされているファイルマップ情報を順次
	(2)	MSD990 ~ MSD998 として関連付けされているファイルマップ情報を順次

【Enterprise for Windows】標準動作モードでは,デバイスアドレスはメモ扱いのため,デバイスアドレス「省略」時の検索順序が適用されます。

検索の結果,発見できない場合は,「ファイル未発見」エラーまたは「入出力状態 35」が返却されます。

[ファイルマップ指定との整合性検査]

COBOL プログラム上のデバイスタイプにより, COBOL ファイルマッパーで設定できるファイル種別が限られます。以下の表以外の設定の組み合わせは,実行時エラーとなります。

デバイスタイプ	ファイル種別
---------	--------

MSD	COBOL 独自ファイル, IFAS FILE, ISAM, COBOL DB アクセス(一般の索引編成)
RDB	COBOL DB アクセス(RDB の順編成 / 相対編成 / 索引編成), IFASPRO RDB

[ファイルマップ指定で COBOL アプリケーションが参照しない項目]

以下の指定項目は, COBOL アプリケーションでは参照しません。

指定項目	処 理
レコード長	COBOL プログラム中のファイル定義でのレコード長と一致しなくても, エラーとなりません。COBOL プログラム上の定義で動作します。
ファイル編成	COBOL プログラム中のファイル定義でのファイル編成と一致しなくても, エラーとなりません。COBOL プログラム上の定義で動作します。
注釈	処理されません。

[ファイルマップ指定と IFAS FILE, IFASPRO RDB の環境設定ファイルとの関係]

ファイル種別として, IFAS FILE, IFASPRO RDB が指定された場合, COBOL ファイルマッパーで指定された実ファイル名が IFAS FILE, IFASPRO RDB にそのまま通知されます。よって, IFAS FILE の環境設定ファイル(IFASCLT.INI)の指定は, ファイルマップ指定を評価後, IFAS FILE, IFASPRO RDB にて評価され, 有効となります。詳しくは, 『IFAS FILE, IFASPRO RDB 利用の手引き』を参照してください。

3.1.4 コマンドラインからの登録

ファイルマップ情報を他のマシンに配布する場合, COBOL カスタマイズ情報/ファイルマップ情報登録ツール(CBLCTFM.EXE)を利用し, コマンドライン上から登録を行います。COBOL カスタマイズ情報/ファイルマップ情報登録ツールの使用方法については, 「5.1 カスタマイズ情報/ファイルマップ情報登録ツール」を参照してください。

3.1.5 ファイルマップ指定一覧ファイルの構造

ファイルマップ指定一覧ファイル(拡張子 .CFM)は以下のフォーマットになっていなければいけません。

- (1) ファイル名の拡張子は, .CFM でなければなりません。
- (2) 1 つのファイルに関する形式は, 以下でなければなりません。
["(内部ファイル ID)" {(デバイス)}]

ENGINE=(ファイル種別)

{FILENAME=(実ファイル名)}

{RECORD_LENGTH=(レコード長)}

{ORGANIZATION=(ファイル編成)}

{NOTE=(注釈)}

この形式がファイルの数分繰り返すことで、複数のファイルに関する指定が記述できます。

最終行の末尾も必ず改行を行ってください。

(3) すべての設定は、1桁目から始まります。

(4) (内部ファイル ID)は、200 桁以内の“(ダブルコーテーション),=(イコール)以外の記号文字、英数字、カナ文字、日本語からなる任意の文字列として指定します。

日本語は1文字で2桁と数えます。

(5) (デバイス)は、省略かまたは以下の形式になります。

MSD000 ~ MSD009

MSD990 ~ MSD998

RDB

(デバイス)を省略する場合は、何も記述してはいけません。省略された場合は 一般ファイルならば「指定なし」を、RDB ファイルならば「RDB」を指定したものとみなします。

誤ったデバイスが指定された場合は、一般ファイルならば「指定なし」に RDB ファイルなら「RDB」に強制的に変更します。

(デバイス)と(内部ファイル ID)の組み合わせが同じファイルは登録できません。また、(デバイス)を省略する場合は、同じ(内部ファイル ID)をもつ他の一般ファイルは登録できません。

(6) (内部ファイル ID)と(デバイス)は、= で結び[]で囲みます。

内部ファイルIDの前後には“(ダブルコーテーション)が必須ではありませんが、付加する場合は 必ず 前後両方に付けてください。

(デバイス)を省略する場合は、(内部ファイル ID)のみを []で囲みます。

内部ファイルIDの最後に空白を付加しても、それは無視されます。

(7) (ファイル種別)は、ENGINE= に続けて、以下のいずれかの記述名を指定します。

記述名	ファイルエンジン名
COBOL	COBOL 独自ファイル
COBOL DB アクセス	COBOL DB アクセス(一般ファイル)
COBOL DB アクセス(RDB)	COBOL DB アクセス(RDB ファイル)
IFAS FILE	IFAS FILE(一般ファイル)
IFASPRO RDB	IFASPRO RDB ファイル
ISAM	ISAM

記述名の英数字(空白を含む)は1バイト系、カタカナは2バイト系のみ有効。

なお、空白にタブコードは指定不可。空白文字の複数指定も不可。

- (8) (実ファイル名)は、省略または FILENAME= に続けて、259 桁以内で指定します。実行時はこの設定が 実際のファイル名 (または、表名)として動作します。

省略した場合は[内部ファイル ID]を代用します。

- (9) (レコード長)は、RECORD_LENGTH= に続けて、0～65535 の数値で、そのファイルの最大レコード長を指定します。この指定は省略できます。

省略された場合は 0 が指定されたものとみなします。

- (10) ファイル編成は、省略または ORGANIZATION= に続けて、そのファイルの編成を以下の中から選択します。

順編成

行順編成

行順編成 (CSV 形式)

相対編成

索引編成

この指定は省略できます。省略された場合は 指定なし が指定されたものとみなします。

- (11) (注釈)は省略または NOTE= に続けて 256 桁以内の任意の文字列として指定します。日本語は1文字で2桁と数えます。

また、改行したい場所には「¥n」を指定してください。

- (12) コマンドライン起動から参照された場合は、以下の情報を元に登録先を決定する場合があります。

[OWNER=MODE]

MODE={USER/MACHINE}

USER : 対象先をユーザ単位にします

MACHINE : 対象先をマシン単位にします

本情報は、ファイル出力時に自動的に付加しています。

本情報は、GUI形式から読み込まれた場合は意味を持ちません。

- (13) 例を示します。

[FILE01]

ENGINE=COBOL DB アクセス

[FILE02=MSD001]

ENGINE=COBOL

FILENAME=C:¥AP¥NOUHIN

RECORD_LENGTH=256

ORGANIZATION=順編成

NOTE=納品ファイル

```
[MASTFILE]
ENGINE=IFAS FILE
FILENAME=C:¥DATFILE
```

定義ファイルを出力し(「3.3.6 ファイルに保存する」を参照),それを雛形とされる事をお勧めします。

3.2 操作環境

3.2.1 ファイル メニュー



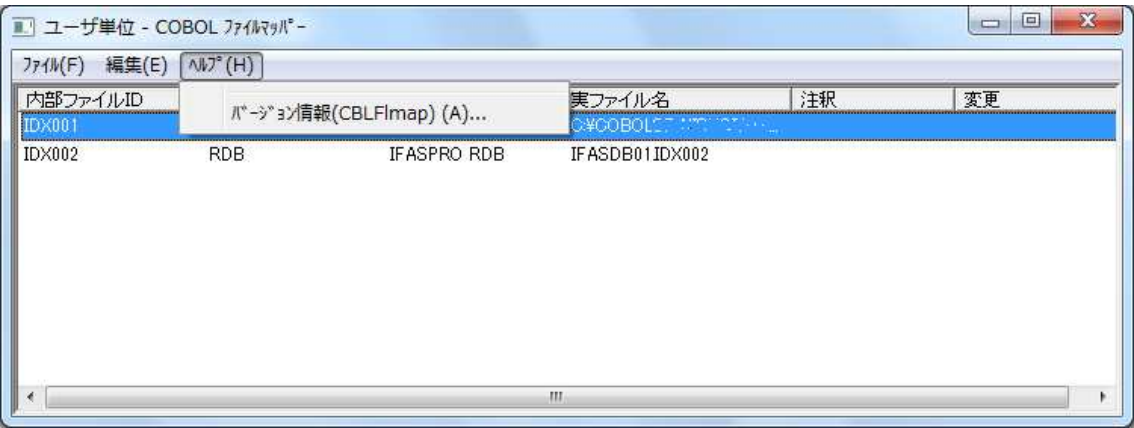
コマンド	ショート カットキー	説 明
開く(O)		編集対象となるファイルマップ情報をユーザ単位 / マシン単位で開きます。
レジストリへ保存(S)	Ctrl+S	変更内容をレジストリへ保存します。
ファイルから追加(F)		保存したファイルマップ指定一覧ファイルからデータを追加します。 COBOL85 Pro のファイルマップデータから追加する場合, デバイスタイプ FDU を MSD に置換します。置換対象の内部ファイル ID の変更欄に を表示します。
ファイルへ保存(A)		設定した内容をファイルマップ指定一覧ファイルへ保存します。
アプリケーションの終了(X)		COBOL ファイルマッパーを終了します。

3.2.2 編集 メニュー



コマンド	ショート カットキー	説 明
新規登録(N)	Ctrl+N	新しいファイルマップ指定を追加します。
内容の編集(E)	Enter	登録されている内容を変更します。
コピー(C)	Ctrl+C	登録されている内容をコピーして追加します。
全選択(A)	Ctrl+A	登録されている内容をすべて選択します。
削除(D)	Delete	リスト中の選択した項目を削除します。

3.2.3 ヘルプ メニュー



コマンド	ショート カットキー	説 明
バージョン情報		COBOL ファイルマッパーのバージョンや著作権などについての

(CBLFImap)(A)	情報を表示します。
---------------	-----------

3.2.4 右クリック メニュー

COBOL ファイルマッパー上でマウスの右ボタンをクリックすることで、以下のコマンドを実行できます。

- [新規登録]
- [内容の編集]
- [コピー]
- [全選択]
- [削除]

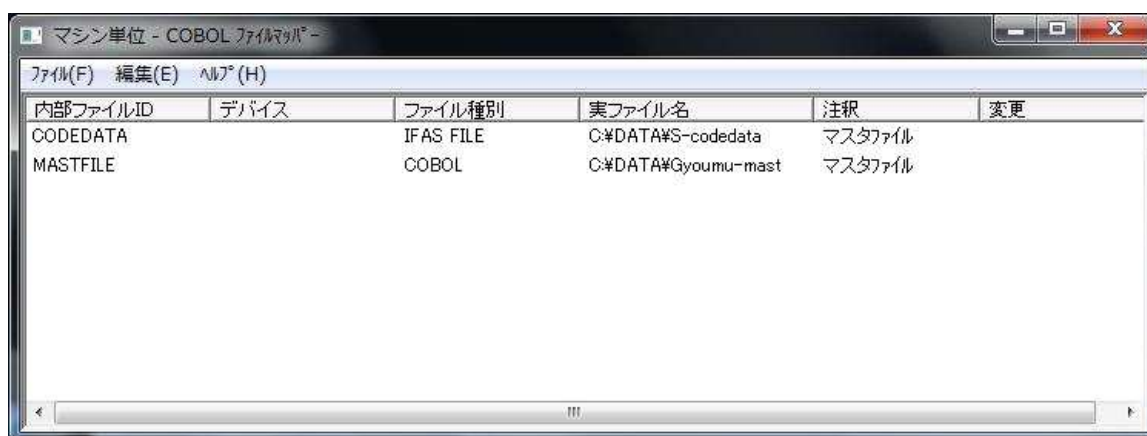
3.3 基本操作

3.3.1 登録されているファイルマップ指定一覧を表示する

登録されているファイルマップ指定一覧を表示するには

COBOL ファイルマッパーを起動します。

表示したいレジストリ種別を選択してください。



3.3.2 新しいファイルマップ指定を追加する

新しいファイルマップ指定を追加するには

1. [編集]メニューの[新規登録]コマンドを選びます。
2. [内部ファイルID]を入力します。すでに登録されている内部ファイルIDとデバイスを指定することはできません(アドレスの異なるデバイスは登録できますが、デバイス 指定なし とアドレス指定のあるデバイスの両方を登録することはできません)。
3. [ファイル種別]をプルダウンメニューから選択します。
4. [実ファイル名]を入力します。この値は省略可能で、省略した場合は内部ファイルIDと同じ値を実ファイル名とみなします。
5. 必要に応じて、[デバイス]、[ファイル編成]、[レコード長]、[注釈]を選択または入力します。
6. [追加]ボタンをクリックします。

[内部ファイルID]には、ファイルの FD 句に設定する VALUE OF IDENTIFICATION 句の値を指定します。

注意

- 追加した指定は、レジストリへ保存しないと有効になりません。
- [終了]ボタンをクリックするまで、続けて追加できます。

3.3.3 内容を変更する

内容を変更するには

1. 登録されている一覧から内容を変更する行を選択します。
2. [編集]メニューの[内容の編集]コマンドを選びます。または、変更したい行をダブルクリックします。
[内容の編集]ダイアログボックスが表示されます。
3. [内部ファイルID]を入力します。このとき、元の内部ファイルIDと異なる内部ファイルIDを指定すると、元の行は消去されます。内部ファイルIDを変更する場合は、すでに登録されている内部ファイルIDとデバイスを指定することはできません(アドレスの異なるデバイスは登録できますが、デバイス 指定なし とアドレス指定のあるデバイスの両方を登録することはできません)。

4. [ファイル種別]をプルダウンメニューから選択します。
5. [実ファイル名]を入力します。
6. 必要に応じて,[デバイス],[ファイル編成],[レコード長],[注釈]を選択または入力します。

[更新]ボタンをクリックします。

注意

- 変更した内容は,レジストリへ保存しないと有効になりません。

3.3.4 コピーして追加する

コピーして追加するには

- 1 登録されている一覧から内容をコピーする行を選択します。
- 2 [編集]メニューの[コピー]コマンドを選びます。
[コピー]ダイアログボックスが表示されます。
- 3 [内部ファイルID]を入力します。このとき,すでに登録されている内部ファイルIDとデバイスを指定することはできません(アドレスの異なるデバイスは登録できますが,デバイス 指定なしとアドレス指定のあるデバイスの両方を登録することはできません)。
- 4 [ファイル種別]をプルダウンメニューから選択します。
- 5 [実ファイル名]を入力します。
- 6 必要に応じて,[デバイス],[ファイル編成],[レコード長],[注釈]を選択または入力します。
- 7 [追加]ボタンをクリックします。

注意

- [終了]ボタンをクリックするまで,続けてコピーできます。
- 追加した指定は,レジストリへ保存しないと有効になりません。

3.3.5 削除する

削除するには

1. 登録されている一覧から削除する行を選択します。
2. [編集]メニューの[削除]コマンドを選びます。
3. メッセージボックスが表示されるので,削除してよければ,[はい]ボタンをクリックします。

注意

- 削除した指定は、レジストリへ保存しないと有効になりません。

3.3.6 ファイルに保存する

ファイルマップ指定一覧ファイルに保存するには

1. 保存するファイル名を入力します。ファイル名の拡張子は、[.CFM]でなければなりません。

3.3.7 ファイルから追加する

ファイルマップ指定一覧ファイルから追加するには

1. [ファイル]メニューの[ファイルから追加]コマンドを選びます。
2. COBOL ファイルマッパーで保存したファイルマップ指定一覧ファイルの名前を入力します。ファイル名の拡張子は、[.CFM]でなければなりません。

注意

- すでに登録されている項目がファイル中に存在した場合、[上書きの確認]ダイアログボックスが表示されます。
- 読み込みが終了すると、[ファイル読み込み結果表示]メッセージボックスが表示されます。
- 追加した指定は、レジストリへ保存しないと有効になりません。

3.3.8 レジストリへ保存する

変更内容をレジストリへ保存するには

1. [ファイル] メニューの[レジストリへ保存]コマンドを選びます。

注意

[レジストリへ保存]コマンドは、設定内容が変更されている状態の時に選択可能になります。

レジストリへ保存しないと、変更内容が有効になりません。

第4章 COBOL アプリケーション

4.1 概要

【Enterprise for Windows】【Standard】Windows 対応の COBOL アプリケーションについて説明します。

ここでは、COBOL アプリケーションの特長やアプリケーション作成時の留意点について説明します。

COBOL アプリケーションの形態として、

対話型アプリケーション : COBOL ウィンドウまたは COBOL WS エミュレータ画面上からのデータ入力が行える形態のアプリケーションです。
(コンパイル時の-Mw オプションを指定することで作成されます。)

バッチ型アプリケーション : COBOL ウィンドウを持たず、コマンドプロンプト上で動作する形態のアプリケーションです。
(コンパイル時の既定値オプションで作成されます。)

の2つがあります。

4.2 COBOL ウィンドウ

[Enterprise for Windows] [Standard] Windows 対応の COBOL アプリケーションについて説明します。

対話型アプリケーション実行時に表示されるウィンドウについて記述します。

COBOL ウィンドウには、COBOL メインウィンドウ、ガイドディスプレイウィンドウ、実行時メッセージウィンドウがあります。

4.2.1 COBOL メインウィンドウ

アプリケーション起動時、以下に示す COBOL メインウィンドウが表示されます。

[illegible]

COBOL の画面に対する入出力文の実行は、このウィンドウに対して行われます。

また、COBOLメインウィンドウが最小化されている場合は、COBOLの画面に対する入出力文の実行は、ウィンドウを復元しないと表示されません。

ここで示した COBOL メインウィンドウは、既定値の画面に対して、COBOL アプリケーションカスタマイザを使用して、「アプリケーションの背景色を白にする」指定を行ったものです。COBOL アプリケーションカスタマイザを使用することで、その他 COBOL メインウィンドウに対する各種カスタマイズが行えます。

注意

COBOL ウィンドウで使用するフォントの既定値は、FixedSys 12pt です。ただし、アプリケーションを

実行する OS によっては、そのフォントがなくディスプレイ解像度が 640*480 では表示しきれない場合があります。その場合にはフォントの既定値を MS ゴシック 12pt に変更して表示します。

4.2.2 ガイドディスプレイウィンドウ

ガイドディスプレイウィンドウは、GDD に関連付けられた呼び名を指定した DISPLAY 文または定数指定の STOP 文による、ガイドディスプレイ領域への表示要求が発生した場合に作成されます。

ガイドディスプレイウィンドウの外形は以下に示すとおりです。



内容を確認後、マウスにより[OK]ボタンをクリックするか、キーボードから[Esc]キーを入力するとウィンドウを閉じます。

また、COBOL メインウィンドウが最小化されている場合でも、ガイドディスプレイウィンドウは表示されます。

4.2.3 実行時メッセージウィンドウ

アプリケーション実行時にエラー等が発生した場合、以下に示すウィンドウ(表示内容はメッセージによって異なります)が表示されます。



内容を確認後、マウスにより[OK]ボタンをクリックするか、キーボードから[Esc]キーを入力するとウィンドウを閉じます。その後の動作は、発生したエラーにより異なりますので、『COBOL プログラミングの手引』「付録 C 実行時メッセージ一覧」を参照ください。

また、COBOL メインウィンドウが最小化されている場合も、実行時メッセージウィンドウは表示されず。

4.3 リソースファイル

【Enterprise for Windows】【Standard】Windows 対応の COBOL アプリケーションについて説明します。

リソースファイルは、COBOL アプリケーションにリンク後のリソースファイルを追加する時に必要になります。アイコンのリソースを変更するには、『COBOL プログラミングの手引』「1.4.2 翻訳用コマンドとその使用法」のコンパイルオプション(-Yi)を参照ください。

4.4 レジストリ

【Enterprise for Windows】【Standard】Windows 対応の COBOL アプリケーションについて説明します。

アプリケーションごとに、または複数のアプリケーション共有で、プログラムの動作などをカスタマイズするには、レジストリを使用します。

メインプログラムの翻訳時にコンパイルオプション-Yr で、使用するレジストリのキー名を指定します。

[オプション形式] -Yrregistrykey

registrykey : レジストリのキー名(<CompanyName>¥<ProductName>)

キー名に対して、COBOL アプリケーションカスタマイザを使用してカスタマイズ情報を設定します。

なお、アプリケーションに対して特別な設定を必要とせず、既定値のまま動作させる場合には、レジストリの指定は必要ありません。

また、COBOL アプリケーション全体に対する各種設定を行う場合にも、レジストリの設定は不要であり、COBOL アプリケーションカスタマイザを使用して、マシンまたはユーザ単位で既定値の変更が行えます。

[レジストリキーの名前について]

- ・ 以下の記述にあるレジストリキーは、コンパイル時に-Yr オプションによってユーザ指定のキー名が指定され、実際にそれらのキーがレジストリ情報として作成されている場合を想定して記述しています。
- ・ 翻訳時の-Yr オプションによるユーザ指定キー名の有無、実行時における定義場所、および既定値情報の有無によりカスタマイズ情報を取得するレジストリキーが異なります。検索時の優先順位は以下のようになります。
 - (1) HKEY_CURRENT_USER¥Software¥ユーザ指定のキー名
 - (2) HKEY_LOCAL_MACHINE¥Software¥ユーザ指定のキー名
 - (3) HKEY_CURRENT_USER¥Software¥NEC¥CBLAP default
 - (4) HKEY_LOCAL_MACHINE¥Software¥NEC¥CBLAP default
- ・ カスタマイズ情報を取得するレジストリキーは次の規則によって決定します。
 - (a) ユーザ指定のキー名が設定されている場合は上記(1)を、指定されていない場合は上記(3)のレジストリキーの存在をチェックします。
 - (b) (a)のキーが存在した場合は、そのキーおよびその配下の情報のみを取得します。その他のキーおよびその配下の情報は取得しません。
 - (c) (a)のキーが存在しない場合は、優先順位に従って次のキーの存在をチェックし、そのキーが存在すれば、そのキーおよびその配下の情報のみを取得する。その他のキーおよびその配下の情報は取得しません。
 - (d) (1)～(4)のキーがすべて存在しない場合は、すべての情報に既定値が設定されているものとします。

設定方法についての詳細は、「第 2 章 COBOL アプリケーションカスタマイザの使い方」を参照してください。

4.5 COBOL REST API

【Enterprise】

本機能は Enterprise Edition 版で使用できます。SE 動作モードでは使用できません。

本機能を使用するために、開発する環境には COBOL Enterprise Edition Developer/Compiler API ライセンス V4.0 のインストールが必要です。実行する環境には COBOL Enterprise Edition Server Runtime API ライセンス V4.0 のインストールが必要です。

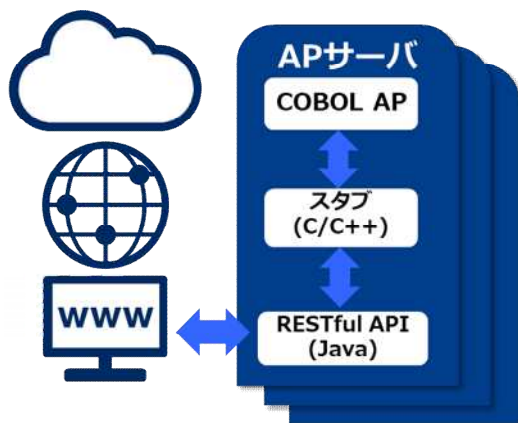
4.5.1 COBOL REST API 生成機能

COBOL REST API 生成機能は、COBOL プログラム(関数)を Java プログラムから呼び出す部品を生成する機能で、COBOL アプリケーションモジュールを Java から呼び出すための COBOL スタブモジュールと、COBOL REST API モジュールで構成されます。

COBOL REST API モジュールと COBOL スタブモジュールは、Java VM やコンテナ内での動作であるため、Java VM やコンテナの性能、信頼性に依存する部分があります。システムとして稼動する前に、評価期間を設けて評価してください。

4.5.2 COBOL REST API の概要

COBOL REST API は、JNI (Java Native Interface) 機能を使用する Java API で、COBOL スタブモジュールを介して Java プログラムから COBOL プログラムの呼び出しを行うことができます。本章では、Restful API を REST API と表します。

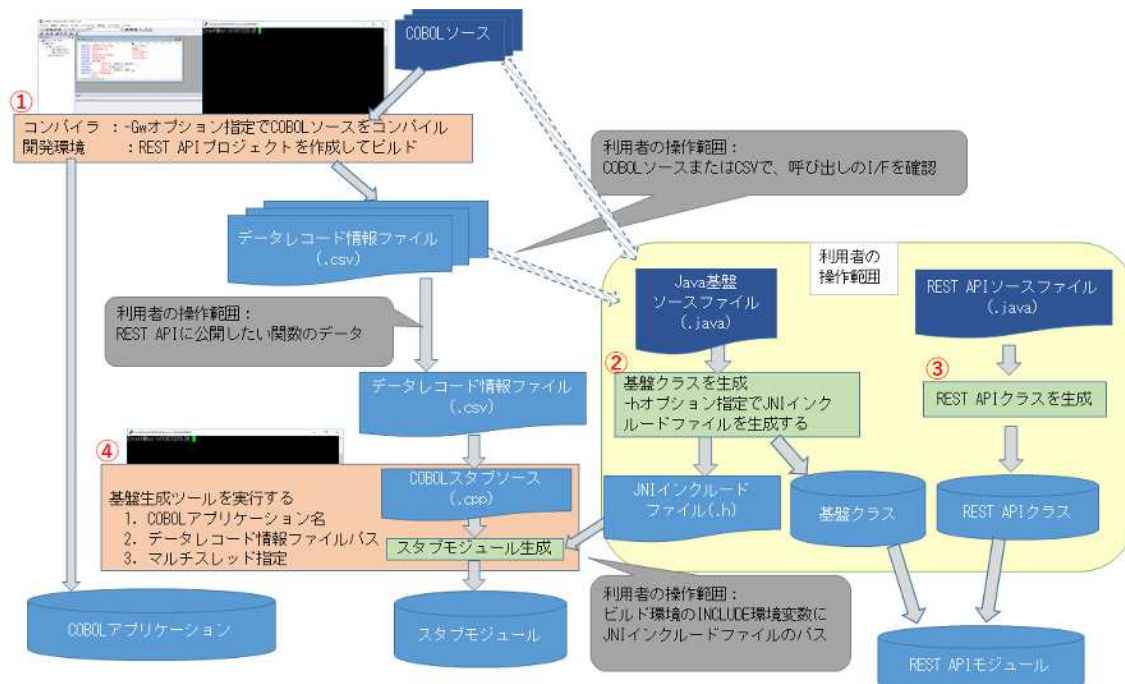


4.5.3 COBOL REST API の生成方法

COBOL REST API モジュールの生成には、COBOL 開発環境を使用する方法と、コマンドラインで COBOL コンパイラを起動する方法があります。COBOL 開発環境を使用するには、COBOL Enterprise Edition の Developer 製品のインストールが必要です。

COBOL REST API 生成手順を説明します。

COBOL REST API モジュールと COBOL スタブモジュールの生成手順には、利用者による Java ソースファイルの作成およびコンパイルと、本製品が提供する基盤生成ツールの起動操作を含みます。



(1) COBOL 開発環境を使用する場合

COBOL アプリケーションモジュールとデータレコード情報ファイルの生成

- ・「プロジェクトの新規作成」の手順で、COBOL バッチ型アプリケーションのプロジェクトを作成し、[オプション設定]画面で成果物形式の[REST API]をチェックします。

オプション設定

転送	ACOS互換製品	コマンドライン
全般	COBOL	リンカ
	Pro*COBOL	C/C++
		オプション製品

初期化

成果物形式

アプリケーション

☐ バッチ型

ライブラリ

☐ 動的 ☒ REST API

☐ 静的

☐ オブジェクトファイル

成果物のファイル名 [-o]

プリコンパイラのビルド設定

全初期化 OK キャンセル

- ・プロジェクトに COBOL REST API を生成する対象の COBOL ソースファイルを登録します。
- ・[オプション設定]画面で、任意の COBOL のコンパイルオプションを設定します。
次のコンパイルオプションの指定は必須です。
「複数スレッドにて動作できるスレッドセーフなオブジェクトを生成する[-MT]」

[Enterprise for Linux]

動的ライブラリ名を lib"COBOL アプリケーション名".so にする必要があります。

開発環境では既定で lib"プロジェクト名".so で作成されますので、成果物のファイル名に指定の必要はありません。

アプリケーション名を SAMPLE にする場合の例)

成果物のファイル名の指定は、SAMPLE
生成されるライブラリ名は、libSAMPLE.so

- ・[ビルド] - [リビルド]コマンドを選択します。
コンパイルが正常終了すると、COBOL アプリケーションモジュールとデータレコード情報ファイルが Release ディレクトリに生成されます。
COBOL REST API 用の COBOL アプリケーションモジュールの生成時に指定できないコンパイルオプションを指定するとコンパイルエラーになります。該当するオプションのチェックを外してリビルドしてください。

[Enterprise for Linux]

COBOL REST API のプロジェクトを開く前に、Linux サーバーにログインしてください。サーバーにログインする前にプロジェクトを開くと、サーバーから COBOL Enterprise Edition Developer/Compiler API ライセンス V4.0 のインストール情報を取得できないため、「COBOL 動的ライブラリ」のプロジェクト状態に変更されます。この場合は、プロジェクトを保存せずにソリューションを閉じて、サーバーにログインしてからソリューションを開きなおすことで、COBOL REST API のプロジェクト状態になります。

基盤クラスファイルの生成

・COBOL ソースファイルまたは で生成したデータレコード情報ファイルを参照して、製品に添付の基盤ソースのサンプルファイルを参考に、基盤ソースファイル(Java)を作成します。基盤ソースファイルのプログラミングについては、「4.5.7 Java 基盤プログラミング」の章で説明します。

- ・作成した基盤ソースファイルを Java コンパイルして、基盤クラスファイルと JNI インクルードファイルを生成します。

コマンド例) javac -h ./COBOLAPBase.java

REST API モジュールの生成

- ・Java 基盤を呼び出す REST API ソースファイルを作成して、REST API クラスを生成します。
- ・REST API の配置方法に則り、基盤クラスファイルと REST API クラスファイルをパッケージし、War ファイル形式といった REST API モジュールを作成します。

COBOL スタブモジュールの生成

基盤生成ツールを起動して、COBOL スタブモジュールを生成します。
基盤生成ツールの使用方法については、「4.5.5 基盤生成ツール」の章で説明します。

(2) COBOL 開発環境を使用しない場合

COBOL アプリケーションモジュールとデータレコード情報ファイルの生成

- ・-Gw オプションと-MT オプションを指定して COBOL のコンパイルを実行します。
コンパイルが正常終了すると、COBOL アプリケーションモジュールとデータレコード情報ファイルがカレントディレクトリに生成されます。
- ・動的ライブラリ名を lib"COBOL アプリケーション名".so にする必要があります。
-o オプションでライブラリ名を指定してください。

コマンド例) `cob -Gw -MT -o"libSAMPLE.so" SAMPLE.cob`

から の手順は、(1)の COBOL 開発環境を使用する場合と同じです。

4.5.4 COBOL REST API の実行方法

COBOL アプリケーションモジュール、COBOL スタブモジュール、COBOL REST API モジュールをサーバー環境にある Web コンテナ (Apache Tomcat など) に配置して、Web システムを構築します。

COBOL アプリケーションモジュールおよび COBOL スタブモジュールは、実行時にパス指定なしで参照可能とするための設定と配置を行ってください。

【Enterprise for Windows】

例) 環境変数 PATH で設定されているディレクトリに配置

【Enterprise for Linux】

例) 環境変数 LD_LIBRARY_PATH で設定されているディレクトリに配置

4.5.5 基盤生成ツール

[機 能]

COBOL コンパイラで生成されたデータレコード情報ファイルを入力し、COBOL スタブモジュールを生成します。

[形 式]

```
cobap4j {-file} "file" {-libname} "libname" [-MT]
```

コマンドライン上で実行してください。

基盤生成ツールの使用にあたっての環境の設定については、「4.5.9 制限事項 / 注意事項」を参照してください。

[Enterprise for Windows]

COBOL コンパイラを起動する Visual C++ のコマンドプロンプトでの実行が可能です。

次のオプションが指定可能です。

-file "file"

"file" には、入力するデータレコード情報ファイルをフルパスで指定します。

短縮形(-f)

-libname "libname"

"libname" には、呼び出す COBOL アプリケーションモジュール名を指定します。

短縮形(-l)

[Enterprise for Windows]

モジュール名(ファイル名)から拡張子を除いた部分

(例: cobolap.dll の場合、libname は cobolap)

[Enterprise for Linux]

モジュール名(ファイル名)から先頭の lib と拡張子を除いた部分

(例: libcobolap.so の場合、libname は cobolap)

-MT

マルチスレッド対応版の COBOL スタブモジュールを生成します。

指定しない場合は、シングルスレッドモジュールを生成します。

本バージョンでは、必ず -MT を指定してください。

[出 力]

カレントディレクトリに COBOL スタブモジュールを生成します。

[Enterprise for Windows]

"COBOL アプリケーション名"BS.dll

(例: cobolap.dll の場合、cobolapBS.dll)

COBOL スタブモジュールが正しく生成できた場合は、「正常終了」を標準出力へ出力します。

異常が発生した場合は、「エラーメッセージ」と「異常終了」を標準出力へ出力します。

エラーメッセージの詳細は、「4.5.8 エラーメッセージ」を参照してください。

[Enterprise for Linux]

lib"COBOL アプリケーション名"BS.so

(例: libcobolap.so の場合, libcobolapBS.so)

ロケールが SJIS の場合は、COBOL スタブモジュールが正しく生成できた場合は、「正常終了」を標準出力へ出力します。異常が発生した場合は、エラーメッセージと「異常終了」を標準出力へ出力します。

それ以外の場合は、COBOL スタブモジュールが正しく生成できた場合は、「success」を標準出力へ出力します。異常が発生した場合は、エラーメッセージと「failure」を標準出力へ出力します。

スタブモジュールが正しく生成できた場合は、終了コード 0 を返却します。

正しく生成できなかった場合は、終了コード 0 以外を返却します。

エラーメッセージの詳細は、「4.5.8 エラーメッセージ」を参照してください。

4.5.6 Java 基盤

Java 基盤について示します。

Java 基盤は、基盤クラスとパラメータ定義 Bean クラスの2つで構成されます。

ここでは、それぞれのクラス名を “ COBOLAPBase ” “ SAMPLEParamRequest ” として記述します。クラス名およびメソッド名は、COBOL のアプリケーション名とプログラム名に置換する必要があります。詳細は、「4.5.7 Java 基盤プログラミング」で説明します。

1. 基盤クラス

基盤クラスは、COBOL アプリケーションの中の COBOL プログラム(関数)を呼び出すための初期処理と終了処理を行います。初期処理では呼び出しを中継する COBOL スタブモジュールをロードし、COBOL スタブモジュールの初期処理を呼び出します。終了処理では COBOL スタブモジュールの終了処理を呼び出します。

基盤クラスは、パッケージ neccobol に属します。

neccobol.COBOLAPBase

基盤クラスのメソッドを次に示します。

Const()

項目	初期化処理
引数	なし
戻り値	なし
使用条件	RestAPI クラスの初期化処理時に実行する必要があります。

Dest()

項目	終了処理
引数	なし
戻り値	なし
使用条件	RestAPI クラスの終了処理時に実行する必要があります。

int SAMPLE (SAMPLEParamRequest paramRequest)

項目	COBOL プログラム(関数)の呼出し
引数	COBOL プログラム(関数)に USING または RETURNING 指定 ある場合: パラメータ定義 Bean クラスのオブジェクト ない場合: なし
戻り値	0 :正常終了。 0 以外 :異常終了。 エラー 番号を返却する。 エラー 番号の詳細は「4.5.8 エラーメッセージ」を参照

2. パラメータ定義 Bean クラス

パラメータ定義 Bean クラスは、パッケージ neccobol に属します。

neccobol.SAMPLEParamRequest

パラメータのデータは、パラメータの順に、プログラム名_PARAMn (n は 1 から昇順)の形式で定義します。

次の表は、プログラム名が 'SAMPLE' で初期値指定する場合の記載例です。

COBOL のデータ型	指定する COBOL の文 (PROCEDURE DIVISION)	Java のデータ定義	記載例
COMP-1	USING 句	short	SAMPLE_PARAM1[] = {0};
COMP-2		int	SAMPLE_PARAM2[] = {0};
PIC X(n)		String	SAMPLE_PARAM3[] = {""};
COMP-2	RETURNING 句	int	SAMPLE_Result[] = {0};

データの設定と受け取りは、各データの setter と getter によって行います。

4.5.7 Java 基盤プログラミング

ここでは、次の COBOL プログラム 'SAMPLE' をメソッドとする Java 基盤クラスとパラメータ定義 Bean クラスの Java のプログラミングについて示します。

'SAMPLE' サブプログラムの PROCEDURE DIVISION の USING 句には 3 つの引数の指定があり、型はそれぞれ COMP-1, COMP-2, PIC X(10) とします。RETRUNING 句の指定があり、データ型は COMP-2 です。

このプログラムから、アプリケーション名 'COBOLAP' の COBOL アプリケーションを作成したものとします。

```
IDENTIFICATION DIVISION.  
PROGRAM-ID.      SAMPLE.  
DATA DIVISION.  
LINKAGE SECTION.  
01 PARAM1 COMP-1.  
01 PARAM2 COMP-2.  
01 PARAM3 PIC X(10).  
01 RET COMP-2.  
PROCEDURE DIVISION USING PARAM1 PARAM2 PARAM3 RETURNING RET.  
SAMPLE-START.  
:  
EXIT PROGRAM.
```

1. 基盤クラスファイル

下記のサンプルソースの 'SAMPLE' と 'COBOLAP' を、REST API 化するプログラム名とアプリケーション名に変更します。

```
// COBOLAPBase.java : Source of COBOL RestAPI  
  
package neccobol;  
import java.io.*;  
  
public class COBOLAPBase implements Serializable{  
    private static final long serialVersionUID = 1L;  
    private String strErrMsg = null;  
    private static int loaderr = 0;  
    static {  
        try{  
            System.loadLibrary("COBOLAPBS");  
            loaderr = 0;  
        } catch (Error e){  
            System.err.println("[COBOLAPBase] 3301:スタブモジュールのロードに  
失敗しました" + "(" + e.toString() + ")");
```

```

        loaderr = 3301;
    }
}

private synchronized native int cSAMPLE(short[] P1,int[] P2,
                                         String[] P3,int[] R);
private synchronized native void ConstCOBOLAPBS();
private synchronized native void DestCOBOLAPBS();

public int SAMPLE (SAMPLEParamRequest paramRequest){
    int result = 0;
    if(loaderr == 0)
    {
        try{
            this.ConstCOBOLAPBS();
            result = this.cSAMPLE(paramRequest.getSAMPLE_PARAM1(),
                                  paramRequest.getSAMPLE_PARAM2(),
                                  paramRequest.getSAMPLE_PARAM3(),
                                  paramRequest.getSAMPLE_RESULT());
            this.DestCOBOLAPBS();
        } catch (Error e){
            strErrMsg = "[COBOLAPBase] 3302:呼び出しに失敗しました";
            strErrMsg += "(" + e.toString() + ")";
            System.err.println(strErrMsg);
            return 3302;
        }
    }else{
        return loaderr;
    }
    return result;
}
}

```

2. パラメータ定義 Bean クラスファイル

```

// SAMPLEParamRequest : Source of COBOL RestAPI ParamRequest

package neccobol;

public class SAMPLEParamRequest {
    private short    SAMPLE_PARAM1[] = {0};
    private int      SAMPLE_PARAM2[] = {0};
    private String   SAMPLE_PARAM3[] = {""};
    private int      SAMPLE_Result[] = {0};

    public short[] getSAMPLE_PARAM1(){
        return SAMPLE_PARAM1;
    }
}

```

```
}
public void set SAMPLE_PARAM1(short[] val){
    SAMPLE_PARAM1 = val;
    return;
}
public int[] getSAMPLE_PARAM2() {
    return SAMPLE_PARAM2;
}
public void setSAMPLE_PARAM2(int[] val){
    SAMPLE_PARAM2 = val;
    return;
}

public String[] getSAMPLE_PARAM3() {
    return SAMPLE_PARAM3;
}
public void setSAMPLE_PARAM3(String[] val){
    SAMPLE_PARAM3 = val;
    return;
}

public int[] getSAMPLE_RESULT() {
    return SAMPLE_Result;
}
public void setSAMPLE_RESULT(int[] val) {
    SAMPLE_Result = val;
}
}
```

4.5.8 エラーメッセージ

1. 基盤生成ツール

パラメータエラー。(errchar)

Illegal parameter.(errchar)

[埋め込み情報] errchar: 不正指定されたパラメータ文字列

[原因] パラメータの指定が誤っています。

[処置] 指定したパラメータに問題ないかを確認してください。

メモリ不足のため処理を続けることができませんでした。

Memory Overflow.

[原因] ツール実行に必要なメモリが不足しています。

[処置] 同時実行するプログラムの数を減らしてください。
利用可能メモリを確保するためにメモリを増設してください。

データレコード情報ファイルのオープンに失敗しました。(errno)

Failed open The data record information file.(errno)

[埋め込み情報] errno: OS が出力するエラーコード

[原因] データレコード情報ファイルのオープンが失敗しました。

[処置] OS の説明書を参照して、エラーコードの対策を実施してください。

データレコード情報ファイルの読み込みに失敗しました。(errno)

Failed read The data record information file.(errno)

[埋め込み情報] errno: OS が出力するエラーコード

[原因] データレコード情報ファイルの読み込みが失敗しました。

[処置] OS の説明書を参照して、エラーコードの対策を実施してください。

プログラム名(関数名)が重複しています。

The program name is duplicated.

[原因] データレコード情報ファイルのプログラム名(関数名)が重複しています。

[処置] データレコード情報ファイルにあるプログラム名定義の重複がないか確認してください。

パラメータ数の指定に誤りがあります。
The number of using items is incorrect.

- [原 因] データレコード情報ファイルのパラメータ数の指定が不正です。
- [処 置] データレコード情報ファイルの該当内容を確認してください。

パラメータ情報の指定に誤りがあります または見つかりません。
The using item is incorrect.

- [原 因] データレコード情報ファイルのパラメータ情報の指定が不正です。
- [処 置] データレコード情報ファイルの該当内容を確認してください。

戻り値情報の指定に誤りがあります または見つかりません。
The returning item is incorrect.

- [原 因] データレコード情報ファイルの戻り値情報の指定が不正です。
- [処 置] データレコード情報ファイルの該当内容を確認してください。

生成ファイルのオープンに失敗しました。(errno)
Failed open the file. (errno)

- [埋め込み情報] errno: OS が出力するエラーコード
- [原 因] スタブソースファイルのオープンが失敗しました。
- [処 置] OS の説明書を参照して、エラーコードの対策を実施してください。

スタブモジュールの生成に失敗しました。(errchar)
Failed make the stub module. (errchar)

- [埋め込み情報] errchar:
 - (1) ECODE: OS が出力するエラーコード
 - (2) ESIG: プロセス終了の原因となるシグナルの番号
 - (3) ECMD: プロセスの終了ステータス
- [原 因] スタブモジュールの生成に失敗しました。
- [処 置] 出力したメッセージの内容を確認してください。

プログラム名の指定に誤りがあります または見つかりません。
The program-ID is incorrect.

- [原 因] データレコード情報ファイルのプログラム名情報の指定が不正です。
- [処 置] データレコード情報ファイルの該当内容を確認してください。

生成ファイルの書き込みに失敗しました。
Failed write the file.

- [原 因] 生成ファイルの書き込みに失敗しました。
- [処 置] ファイルの生成に影響する可能性があるデスク容量などを確認した上、基盤生成ツールを再実行してください。

データレコード情報が不正です。
Invalid the data record information.

- [原 因] データレコード情報ファイルの情報が不正です。
- [処 置] データレコード情報ファイルの内容を確認してください。

2 . Java 基盤

[クラス名]3301: スタブモジュールのロードに失敗しました

- [埋め込み情報] クラス名: 基盤のクラス名
- [処 置] スタブモジュールの配置を確認してください。

[クラス名]3302: 呼び出しに失敗しました(エラーメッセージ)

- [埋め込み情報] (1)クラス名: 基盤のクラス名
(2)エラーメッセージ: 失敗原因を表す文字列
- [処 置] エラーメッセージを元に原因を特定し取り除いてください。

3 . COBOL スタブ

メッセージの前に、次のヘッダ情報を付加します。

[yyyy/mm/dd,HH:MM:SS:MS,PID=プロセス ID,TID=スレッド ID]

yyyy:西暦(4桁)
mm :月(2桁)
dd :日(2桁)
HH :時(24H表記2桁)
MM:分(2桁)
SS :秒(2桁)
MS :ミリ秒(4桁)
PID :プロセス ID(10進)
TID :スレッド ID(10進)

日時情報はログ出力時点のローカル時間

【Enterprise for Windows】

【クラス名】 3303: COBOL DLL ロードエラー(ライブラリ名)(エラーコード)

【埋め込み情報】 (1)クラス名:呼び出し元のクラス名
(2)ライブラリ名:ロードに失敗したライブラリ名
(3)エラーコード:失敗原因を表すコード(GetLastError())の10進値)
【処 置】 エラーコードを元に原因を特定し取り除いてください。

【クラス名】 3304: 関数アドレスの取得に失敗しました(関数名)(エラーコード)

【埋め込み情報】 (1)クラス名:呼び出し元のクラス名
(2)関数名:関数アドレスの取得に失敗した関数名または COBOL プログラム名
(3)エラーコード:失敗原因を表すコード(GetLastError())の10進値)
【処 置】 エラーコードを元に原因を特定し取り除いてください。

【クラス名】 3305: COBOL プログラムの初期化に失敗しました

【埋め込み情報】 クラス名:呼び出し元のクラス名
【処 置】 同時実行するプログラムの数を減らしてください。
利用可能メモリを確保するためにメモリを増設してください。

【クラス名】 3306: COBOL ランタイムロードエラー(ライブラリ名)(エラーコード)

【埋め込み情報】 (1)クラス名:呼び出し元のクラス名
(2)ライブラリ名:ロードに失敗した COBOL ランタイムライブラリ名
(3)エラーコード:失敗原因を表すコード(GetLastError())の10進値)
【処 置】 エラーコードを元に原因を特定し取り除いてください。
COBOL コンパイル時と基盤生成ツールでスタブモジュール作成時の
-MT オプション指定有無を一致させてください。

【クラス名】 3307: データの変換に失敗しました(関数名_PARAMj)(B/A)(エラーコード)

【埋め込み情報】 (1)クラス名:呼び出し元のクラス名
(2)関数名:データ変換に失敗した COBOL プログラム名
(3)PARAMj:データ変換に失敗した引数(j 番目の引数)
(4)B/A:変換エラーの発生した時点
B:COBOL プログラム呼び出し前の COBOL データ型への変換
A:COBOL プログラム呼び出し後の Java データ型への変換
(5)エラーコード:
7:メモリ不足
9:データ中に不正な文字が存在
10:予期せぬエラー
【処 置】 エラーメッセージを元に原因を特定し取り除いてください。
7: 同時実行するプログラムの数を減らしてください。
利用可能メモリを確保するためにメモリを増設してください。

9: データ中に不正な文字が存在しないようにしてください。

10: 弊社担当者にお問い合わせください。

[クラス名] 3308: 予期せぬエラーが発生しました

[埋め込み情報] クラス名: 呼び出し元のクラス名

[処 置] 弊社担当者にお問い合わせください。

[クラス名] COBxxx [program-id/line]

[埋め込み情報] クラス名: 呼び出し元のクラス名

xxx : COBOL 実行時エラーのエラーメッセージ番号

program-id : COBOL プログラム名

line : エラー発生行番号

[処 置] 『COBOL プログラミングの手引』『付録 C 実行時メッセージ一覧』を参照してください。

[Enterprise for Linux]

[クラス名] 3303: COBOL 共有ライブラリロードエラー(ライブラリ名)(エラー文字列)

COBOL Shared Library Load Error(ライブラリ名)(エラー文字列)

[埋め込み情報] (1)クラス名: 呼び出し元のクラス名

(2)ライブラリ名: ロードに失敗したライブラリ名

(3)エラー文字列: 失敗原因を表す文字列

[処 置] エラー文字列を元に原因を特定し取り除いてください。

[クラス名] 3304: 関数アドレスの取得に失敗しました(関数名)(エラー文字列)

Failed to get function address(関数名)(エラー文字列)

[埋め込み情報] (1)クラス名: 呼び出し元のクラス名

(2)関数名: 関数アドレスの取得に失敗した関数名または COBOL プログラム名

(3)エラー文字列: 失敗原因を表す文字列

[処 置] エラー文字列を元に原因を特定し取り除いてください。

[クラス名] 3305: COBOL プログラムの初期化に失敗しました

COBOL program initialization failed

[埋め込み情報] クラス名: 呼び出し元のクラス名

[処 置] 同時実行するプログラムの数を減らしてください。
利用可能メモリを確保するためにメモリを増設してください。

[クラス名] 3306: COBOL ランタイムロードエラー(ライブラリ名)(エラー文字列)

COBOL runtime load error(ライブラリ名)(エラー文字列)

[埋め込み情報] (1)クラス名: 呼び出し元のクラス名

(2)ライブラリ名:ロードに失敗した COBOL ランタイムライブラリ名
 (3)エラー文字列:失敗原因を表す文字列
 [処 置] エラー文字列を元に原因を特定し取り除いてください。
 COBOL コンパイル時と基盤生成ツールでスタブモジュール作成時の
 -MT オプション指定有無を一致させてください。

[クラス名] 3307: データの変換に失敗しました(関数名_PARAMj)(B/A)(エラーコード)
 Data conversion failed(関数名_PARAMj)(B/A)(エラーコード)

[埋め込み情報] (1)クラス名:呼び出し元のクラス名
 (2)関数名:データ変換に失敗した COBOL プログラム名
 (3)PARAMj:データ変換に失敗した引数(j 番目の引数)
 (4)B/A:変換エラーの発生した時点
 B:COBOL プログラム呼び出し前の COBOL データ型への変換
 A:COBOL プログラム呼び出し後の Java データ型への変換
 (5)エラーコード:
 7:メモリ不足
 9:データ中に不正な文字が存在
 10:予期せぬエラー
 [処 置] エラーメッセージを元に原因を特定し取り除いてください。
 7: 同時実行するプログラムの数を減らしてください。
 利用可能メモリを確保するためにメモリを増設してください。
 9: データ中に不正な文字が存在しないようにしてください。
 10:弊社担当者にお問い合わせください。

[クラス名] 3308: 予期せぬエラーが発生しました
 An unexpected error occurred

[埋め込み情報] クラス名:呼び出し元のクラス名
 [処 置] 弊社担当者にお問い合わせください。

[クラス名] COBxxx [program-id/line]

[埋め込み情報] xxx :COBOL 実行時エラーのエラーメッセージ番号
 program-id :COBOL プログラム名
 line :エラー発生行番号
 [処 置] 『COBOL プログラミングの手引』『付録 C 実行時メッセージ一覧』を参照
 してください。

4.5.9 制限事項 / 注意事項

1. 制限事項

COBOL REST API 機能を使用するにあたっての制限事項を示します。

- (1) スレッドセーフ対応の COBOL スタブモジュールと COBOL アプリケーションモジュールのみ対応します。
- (2) 1回の REST API 呼び出しでの COBOL プログラム(関数)の実行は1回のみ可能です。
- (3) REST API で公開する COBOL プログラム(関数)の USING 句に指定できるデータ型は COMP-1, COMP-2 および英数字です。RETURNING 句に指定できるデータ型は COMP-2 のみです。
- (4) 基盤生成ツールでパラメータに指定できるデータレコード情報ファイルは1つのみです。
- (5) Java 基盤とスタブのインタフェースのパラメータ(USING/RETURNING 句の指定データ)の定義はすべて配列とし、該当データは配列の最初の要素に設定する必要があります。
- (6) COMPLETION-CODE の値は取得できません。

2. 注意事項

COBOL REST API 機能を使用するにあたっての注意事項を示します。

(1) 基盤生成ツールの環境設定

基盤生成ツールを使用するには、環境変数にインクルードファイルのパスを設定する必要があります。

[対象のインクルードファイルのパス]

・基盤ソースファイルをコンパイルすると生成される JNI インクルードファイルを格納するディレクトリ

・%JAVA_HOME%\include

・%JAVA_HOME%\include\win32 (Windows のみ)

・%JAVA_HOME%\include\linux (Linux のみ)

[環境変数]

INCLUDE (Windows)

CPATH (Linux)

(2) COBOL アプリケーション名

COBOL REST API で使用する COBOL アプリケーション名は半角英字から始まる英数字(英字と数字のみ、記号を除く)の名前で生成してください。

[Enterprise for Linux]

コマンドラインで COBOL コンパイラを起動して COBOL アプリケーションを生成する場合、-o オプションを指定してください。-o オプションには「lib"アプリケーション名".so」を指定します。

オプション例) -o"libSAMPLE.so"

(3) Java 基盤とスタブのインタフェースのパラメータ数

Java 基盤とスタブのインタフェースのパラメータ(COBOL REST API から呼び出す COBOL アプリケーションの LINKAGE SECTION に指定するデータ)の数は最大 254 個です。

(4) USING と RETURNING の PIC X 項目

COBOL REST API から呼び出す COBOL アプリケーションが下記条件すべて満たした場合、PIC X 項目が正しく表示されません。

- ・COBOL アプリケーション作成時に、-CU もしくは -Cj オプション指定があります。
- ・USING 句の PIC X 項目に 2 バイト文字(ASCII 範囲外文字)が存在します。

(5) COBOL アプリケーションモジュールの Web コンテナ配置

Web コンテナ(Apache Tomcat など)に配置可能な COBOL REST API モジュール、COBOL スタブモジュール、COBOL アプリケーションモジュールは 1 つのみです。

同時に複数の COBOL REST API モジュール、COBOL スタブモジュール、COBOL アプリケーションモジュールを Web コンテナに配置した場合、COBOL アプリケーションの終了処理が正しく実行されない可能性がありますので、ご注意ください。

第5章 ツール

5.1 カスタマイズ情報/ファイルマップ情報登録ツール

【Enterprise for Windows】【Standard】カスタマイズ情報/ファイルマップ情報登録ツールは、Enterprise Edition for Windows 版、および Standard Edition 版において使用できます。ただし、Enterprise Edition for Windows 版の標準動作モードでは、ファイルマップ情報のみ使用できます。

[機 能]

カスタマイズ(CCF)ファイル/ファイルマップデータ(CFM)の自動登録を行う、両ファイルに対応したコマンドラインツールです。

[形 式]

```
CBLCTFM [ /help ]
CBLCTFM /mapdelete [ /user | /machine ]
CBLCTFM { /add | /file | /update } "file" [ /user | /machine ] [ [ /entry ]
[ /ignore ] ] | /check ]
CBLCTFM /remove "file" [ /user | /machine ]
```

コマンドライン上で実行してください。

以下のオプションが指定可能です。

/file "CCF/CFM ファイル名"

CCF/CFM ファイルから情報を上書登録します。

登録先レジストリはファイル内を参照します。

短縮形(/f)

/add "CFM ファイル名"

CFM ファイルから情報を追加登録します。

登録先レジストリはファイル内を参照します。

短縮形(/a)

/update "CCF/CFM ファイル名"

CCF/CFM ファイルから情報を上書登録します。

登録先レジストリはファイル内を参照します。

短縮形(/p)

/remove "CCF/CFM ファイル名"

CCF/CFM ファイルの情報を登録先レジストリから削除します。

短縮形(/r)

/user

登録先レジストリをユーザ単位にします。

CCF/CFM ファイル内の情報よりも優先されます。

短縮形(/u)

/machine

登録先レジストリをマシン単位にします。

CCF/CFM ファイル内の情報よりも優先されます。

短縮形(/m)
 /entry
 コンソール上に、登録内容やエラーなどの詳細を表示します。
 短縮形(/e)
 /mapdelete
 ファイルマップ情報を登録先レジストリから削除します。
 登録先レジストリの指定を省略した場合、マシン単位を削除します。
 短縮形(/d)
 /ignore
 CFM ファイル情報の不正部分を無視して登録します。
 短縮形(/i)
 /check
 チェックのみを行い、レジストリへ登録しない。
 短縮形(/c)
 /help
 コマンドラインオプションのヘルプを表示します。
 短縮形(/? /h)

[メッセージ一覧]

<<CBLCTFM.exe からの出力内容>>

メッセージ内容

- ・ 正常終了(DONE)
 [内容] 処理はすべて正常に終了しました。
- ・ 正常終了(WARNING)
 [内容] (a) /ignore オプションが指定されたためエラーを無視しました。
 (b) FDU MSD のデバイスタイプの置換が発生しました。
- ・ 旧製品の CCF/CFM ファイルです
 [内容] 入力ファイルは COBOL カスタマイザ/ファイルマッパー以外の手段で作成されたものです。
- ・ 異常終了
 [内容] 一部の処理に異常がありました。
- ・ 登録先アクセス拒否
 [内容] 登録先レジストリへのアクセスが拒否されました。
 指定されたレジストリへの書き込み権がないユーザのため、
 情報の登録 / 削除を行うことができません。
- ・ パラメータが多すぎます
 [内容] パラメータの指定が多すぎます。
- ・ パラメータに誤りがあります
 [内容] パラメータの指定が誤っています。
- ・ ファイル名不正
 [内容] 指定されたファイル名の拡張子が誤っています。
 本ツールは .CCF / .CFM のファイルしか扱えません。
- ・ パラメータエラー
 [内容] パラメータの指定が誤っています。
- ・ ファイル未発見
 [内容] 指定されたファイルが見つかりませんでした。
- ・ 登録先情報未発見

- [内容] ファイル内に登録先レジストリ情報が発見できませんでした。
- ・ メモリ不足のため登録内容の表示が出来ません
 - [内容] /entry 指定ですが、メモリ不足で表示処理が出来ませんでした。
- ・ 登録内容の表示に失敗しました
 - [内容] /entry 指定ですが、テンポラリファイルが作成できなかったため表示処理が出来ませんでした。
- ・ デバイスタイプ置換(FDU MSD)が n 箇所発生しています
 - [内容] 扱えないデバイスタイプが指定されたため、置換を行いました(n は置換箇所の数を表します)。
 入力された CFM ファイルが旧製品で作成された場合に発生する可能性があります。
 /entry オプションを同時指定すると、置換後の「[(内部ファイルID):(デバイス)]」の変更欄に を表示します。

<</entry 指定時出力内容>>

[CCF ファイル読み込み時]

メッセージ内容

- ・ (*1)に失敗しました。いくつかのデータは設定できません。
- ・ (*1)に失敗しました。いくつかのデータは不正な可能性があります。
- ・ (*2)を設定します。
- ・ (*2)の処理中に何らかのエラーが発生したので、既定値で設定します。
- ・ (*2)は設定されていないので、既定値で設定します。
- ・ (*2)の処理中に無効なデータを検出したので、既定値で設定します。
- ・ (*2)が限界値を超えているので、既定値で設定します。
- ・ (*2)が限界値を超えているので一部を切り捨てました。
- ・ (*2)の処理中にレジストリへの書き込みエラーが発生しました。
- ・ (*2)はサポート対象外です。

*n の内容

- ・ *1
 - [内容] 「ファイルのオープン」
 「レジストリのオープン」
 「ファイルのクローズ」
 「レジストリのクローズ」
- ・ *2
 - [内容] 「アイコン状態で実行」
 「タイトルバーを変更」
 「背景色を白にする」
 「文字色を緑にする」
 「最大化表示する」
 「タイトルバーを非表示」
 「コントロールメニューを非表示」
 「移動不可能」
 「最小化不可能」
 「サイズ変更可能」
 「初期表示位置固定」
 「フォント変更」

「フォント変更(サイズ)」
 「キーカスタマイズ」
 「プリンタカスタマイズ(メディア識別名)」
 「プリンタカスタマイズ(個数)」
 「プリンタカスタマイズ(プリンタ名)」
 「小入力ファイル名指定」
 「小出力ファイル名指定」
 「改行位置指定」
 「OUTPUT モード出力」
 「RDB-STATUS 通知」
 「強制終了不可能」
 「ウィンドウサイズ既定値」
 「起動時のウィンドウサイズ」
 「起動時の表示開始位置」
 「先行キー入力」
 「英小文字変換」
 「半角スペース変換」
 「混在入力項目 IME 起動」
 「カンマ変換」
 「画面ハードコピー出力モード」
 「カナ入力モード」
 「ファイル排他制御」

[CFM ファイル読み込み時]

メッセージ内容

- ・ x 件目: [(内部ファイル ID):(デバイス)]を登録しました。以下詳細情報
 ファイル種別: [(ファイル種別)]
 実ファイル名: [(実ファイル名)]
 ファイル編成: [(ファイル編成)], レコード長: [(レコード長)]
 注釈: [(注釈)]
- ・ x 件目: [(内部ファイル ID)]ファイル種別が不明です。
- ・ x 件目: 内部ファイルIDなどのフォーマットが誤っています。
- ・ x 件目: 内部ファイルIDがありません。
- ・ x 件目: 内部ファイルIDは、200バイト以下でなければなりません。
- ・ x 件目: 内部ファイルID領域の不正な位置に " や = があります。
- ・ x 件目: レコード長が0～65535までの数字でなければなりません。
- ・ x 件目: ファイル編成が誤っています。
- ・ x 件目: 実ファイル名が259バイト以下でなければなりません。
- ・ x 件目: すでにデバイス情報が同一内部ファイルID名で存在するので、登録しませんでした。
- ・ x 件目: すでに同一内部ファイルID名が存在するので、登録しませんでした。
- ・ x 件目: 注釈が256桁以下でなければなりません。
- ・ x 件目: すでに複数のデバイス情報が同一内部ファイルID名で存在するので、登録しません。
- ・ x 件目: 不明な情報が指定されています。

注意

- Windows 10, Windows Server 2016, Windows Server 2019 において、マシン単位の設定を行うには、管理者権限が必要です。(OS の設定によっては、明示的に管理者として実行する必要があります。)

5.2 COBOL アプリケーションビューア

【Enterprise for Windows】【Standard】COBOL アプリケーションビューアは、Enterprise Edition for Windows 版、および Standard Edition 版において使用できます。

【機能】

COBOL で作成されたアプリケーションの実行環境の情報を表示するツールです。

【形式】

【Enterprise for Windows】

```
CBLAPINF [/R] [/S] [/C] [/G] [/E] [/P] [/F] [/M] [/ALL] [/?] "filename"
"filename" :COBOL アプリケーション{.exe | .dll}
```

【Standard】

```
CBLAPINF [/R] [/S] [/C] [/G] [/E] [/P] [/F] [/ALL] [/?] "filename"
"filename" :COBOL アプリケーション{.exe | .dll}
```

【パラメータ】

COBOL アプリケーションが ” dll ” ファイルの場合は/E および/F オプションのみが使用できません。

/R

指定された COBOL アプリケーションを実行するとロードされる COBOL ランタイムライブラリのフルパス、COBOL ランタイムライブラリ名、製品名、ファイルバージョン、タイムスタンプ情報を取得します。以下に表示される情報の形式の表示例を示します。

【COBOL ランタイム】

```
C:¥WINDOWS¥SYSTEM32¥NECCOB.DLL
製品名: COBOL SE
ファイルバージョン: 1.02.000
タイムスタンプ: Sat Oct 18 01:00:00 2014
```

/S

指定された COBOL アプリケーションを実行するとロードされる COBOL アプリケーション使用システムライブラリのフルパス、システムライブラリ名、ファイルバージョン、タイムスタンプ情報を取得します。以下に表示される情報の形式の表示例を示します。

【ライブラリ】

C:¥WINDOWS¥SYSTEM32¥KERNEL32.DLL

製品名: Microsoft® Windows® Operating System

ファイルバージョン: 6.1.7600.16385

タイムスタンプ: Tue Jul 14 10:41:13 2009

/C

指定された COBOL アプリケーションを実行するとロードされる COBOL アプリケーション
関連コンポーネントライブラリのフルパス、関連コンポーネントライブラリ名、ファイル
バージョン、タイムスタンプ情報を取得します。以下に表示される情報の形式の表
示例を示します。

[関連コンポーネント]

IFAS FILE:

C:¥Program Files (x86)¥NEC¥IFAS FILE¥BIN¥IFASBASE.DLL

製品名: IFAS FILE

ファイルバージョン: 6.00

タイムスタンプ: Sat Oct 18 01:00:00 2014

/G

指定された COBOL アプリケーションを実行すると有効となるカスタマイズレジストリキ
ーおよびアプリケーションカスタマイズの各タブと対応したカスタマイズ情報を取得
します。以下に表示される情報の形式の表示例を示します。

[カスタマイズキー]

有効レジストリキー = ¥HKEY_CURRENT_USER¥Software¥NEC¥

CBLAP default

[カスタマイズ(ウィンドウ1)]

アプリケーションをアイコンの状態で行う = 有効

アプリケーションの背景色を白にする = 無効

アプリケーションの文字色を緑にする = 無効

アプリケーションカスタマイズの各タブ毎にカスタマイズ項目の
現在の状態が表示される

/E

指定された COBOL アプリケーションを実行すると有効となる環境変数情報を取得し
ます。以下に表示される情報の形式の表示例を示します。

[環境変数]

COB_AMRMODE = ON

/P

指定された COBOL アプリケーションを実行すると有効となるプリンタの動作モード情
報を取得します。以下に表示される情報の形式の表示例を示します。

[プリンタ動作モード]

プリンタ動作モード = COBOL 独自 (プリントマネージャー経由)

/F

指定された COBOL アプリケーションを実行すると有効となるシステムに登録されているユーザ単位 / マシン単位のファイルマップ情報を取得します。以下に表示される情報の形式の表示例を示します。

[ファイルマップ(ユーザ単位)]

内部ファイルID (URIAGE)
デバイス = MSD000
ファイル種別 = IFAS FILE
実ファイル名 = URIAGE_KOBE
注釈 = 神戸支店の売り上げです。
ファイル編成 = 順編成
レコード長 = 80

[ファイルマップ(マシン単位)]

内部ファイルID (RIEKI)
デバイス = MSD001
ファイル種別 = IFASPRO RDB
実ファイル名 = RIEKI_KOBE
注釈 = 神戸支店の利益です。
ファイル編成 = 順編成
レコード長 = 80

[Enterprise for Windows]/M

現在の環境の動作モード情報を取得します。
以下に表示される情報の形式の表示例を示します。

[動作モード]

アプリケーション: 標準動作モード

/ALL

上記, すべての実行環境情報を取得します。

/? (/HELP)

アプリケーションビューアの使用方法を表示します。

オペレーティングシステム情報およびコンパイル情報は既定値で必ず取得します。以下に表示される情報の形式の表示例を示します。

実行形式ファイルが複数の COBOL オブジェクトからなる場合, コンパイル情報はオブジェクトの個数分出力します。

[オペレーティングシステム]

OS 名: Microsoft Windows 10 (x64)
ユーザ名: TEST

[コンパイル]

ソースファイル名 (test.cob)

コンパイル日時:20-01-15 13:39:39
コンパイラバージョン:COBOL(SE) 3.1
オペレーティングシステム:Microsoft Windows 10 (x64)
コンパイルオプション : -M -01

また、カレントディレクトリ情報および環境変数(PATH)情報は、指定された COBOL アプリケーションが ".exe" ファイルの場合に必ず取得します。以下に表示される情報の形式の表示例を示します。

[カレントディレクトリ]

D:\TEST

[環境変数(PATH)]

C:\WINDOWS;C:\WINDOWS\COMMAND

[エラーメッセージ]

メッセージ内容

- ・ No.1 オプションは無効です。(埋め込み情報)
 - [原因] 認識できないオプションが指定されています。
 - [処置] プログラムはオプションを無視して継続されます。
 - [埋め込み情報] 無効なオプション名
- ・ No.2 対象となるファイル名が指定されていません。
 - [原因] 対象となるファイル名が指定されていません。
 - [処置] プログラムはヘルプを表示後、異常終了します。
- ・ No.3 対象となるファイル名は1つだけ指定できます。
 - [原因] 対象となるファイル名が複数指定されています。
 - [処置] プログラムはヘルプを表示後、異常終了します。
- ・ No.4 対象となるファイルが存在しません。
 - [原因] 対象となるファイルが存在しません。
 - [処置] プログラムはヘルプを表示後、異常終了します。
- ・ No.5 対象となるファイルの種類が誤っています。
 - [原因] 対象となるファイルに拡張子が ".exe" ".dll" ".obj" 以外のファイルが指定されています。
 - [処置] プログラムはヘルプを表示後、異常終了します。
- ・ No.6 メモリの確保に失敗しました。
 - [原因] アプリケーションビューアを実行するためのメモリの確保に失敗しました。
 - [処置] プログラムは異常終了します。他のアプリケーションを終了させて、使用可能なメモリ容量を増やしてください。

5.3 A-VX モード設定ツール

【Enterprise for Windows】【Standard】A-VX モード設定ツールは、Enterprise Edition for Windows 版の SE 動作モード、および Standard Edition 版において使用できます。

[機 能]

関連製品の実行時動作を A-VX5 または A-VX 互換に設定するツールです。
設定を行うには管理者権限が必要です。

本ツールの設定が有効となる機能や関連製品は以下のとおりです。

- ・IFAS FILE
- ・IFASPRO RDB
- ・COBOL DB アクセス
- ・COBOL アプリケーション実行環境 (PRNAMR 利用時)

[形 式]

CBLOSSET [/AVX2 | /AVX5]

[パラメータ]

/AVX2

A-VX 互換モードを A-VX 互換に設定します。

短縮形 (/2)

設定完了時の表示例を示します。

A-VX 互換モードを A-VX 互換に設定しました。

/AVX5

A-VX 互換モードを A-VX5 互換に設定します。

短縮形 (/5)

設定完了時の表示例を示します。

A-VX 互換モードを A-VX5 互換に設定しました。

パラメータ省略

現在設定されている A-VX 互換モードおよび本ツールの使用方法を表示します。
表示例を示します。

現在の A-VX 互換モードは A-VX 互換です。

[エラーメッセージ]

メッセージ内容

- ・ No.1 パラメータは無効です。(埋め込み情報)

- [原因] 認識できないパラメータが指定されています。
- [処置] プログラムは異常終了します。パラメータを修正して再度実行してください。
- [埋め込み情報] 無効パラメータ名
- ・ No.2 A-VX 互換モードを設定する権限が不足しています。

[原因] 管理者として実行されていません。

[処置] プログラムは異常終了します。管理者として実行してください。
 - ・ No.3 A-VX 互換モードが正しく設定されていません。

[原因] 本ツールを使用して A-VX 互換モードを一度も設定していないか、関連製品のインストール / アンインストールによって A-VX 互換モードの設定が不正になっています。

[処置] プログラムは異常終了します。本ツールを使用して A-VX 互換モードを設定し直してください。
 - ・ No.4 メモリの確保に失敗しました。

[原因] 本ツールを実行するためのメモリの確保に失敗しました。

[処置] プログラムは異常終了します。他のアプリケーションを終了させ、使用可能なメモリ容量を増やしてください。

5.4 アプリケーション動作モード切替ツール

【Enterprise for Windows】アプリケーション動作モード切替ツールは、Enterprise Edition for Windows 版において使用できます。

[機能]

コンパイル時および実行時のアプリケーション動作モード(標準動作モード/SE 動作モード)を切り替えるツールです。

設定を行うには管理者権限が必要です。

注意

本ツールは作成したアプリケーションの動作を動的に切り替えることを目的としたものではありません。

アプリケーション作成時(コンパイル時)とアプリケーション実行時のアプリケーション動作モードは同じにしなければなりません。

[形式]

CBLAPMODE [/STD | /SE]

[パラメータ]

/STD

アプリケーション動作モードを標準動作モードに設定します。
設定完了時の表示例を示します。

アプリケーション動作モードを標準動作モードに設定しました。

/SE

アプリケーション動作モードを SE 動作モードに設定します。
設定完了時の表示例を示します。

アプリケーション動作モードを SE 動作モードに設定しました。

パラメータ省略

現在設定されているアプリケーション動作モードおよび本ツールの使用方法を表示します。
表示例を示します。

現在のアプリケーション動作モードは標準動作モードです。

[エラーメッセージ]

メッセージ内容

- ・ No.1 パラメータは無効です。(埋め込み情報)
 - [原因] 認識できないパラメータが指定されています。
 - [処置] プログラムは異常終了します。パラメータを修正して再度実行してください。
 - [埋め込み情報] 無効パラメータ名
- ・ No.2 アプリケーション動作モードを設定する権限が不足しています。
 - [原因] 管理者として実行されていません。
 - [処置] プログラムは異常終了します。管理者として実行してください。
- ・ No.3 フォルダまたはファイルが見つかりません。
 - 埋め込み情報
 - [原因] 指定したアプリケーション動作モードのランタイムを配置する際に対象となるフォルダまたはファイルが見つかりませんでした。
 - [処置] プログラムは異常終了します。埋め込み情報に表示されたフォルダまたはファイルが存在することを確認してください。
 - [埋め込み情報] 対象フォルダまたはファイル名
- ・ No.4 フォルダまたはファイルのアクセス権がありません。
 - 埋め込み情報
 - [原因] 指定したアプリケーション動作モードのランタイムを配置する際に対象となるフォルダまたはファイルへアクセスする権限がありませんでした。
 - [処置] プログラムは異常終了します。埋め込み情報に表示されたフォルダまたはファイルのアクセス権を確認し、本ツールを起動する管理者ユーザにフルコントロールのアクセス権を設定してください。
 - [埋め込み情報] 対象フォルダまたはファイル名
- ・ No.5 メモリの確保に失敗しました。
 - [原因] 本ツールを実行するためのメモリの確保に失敗しました。
 - [処置] プログラムは異常終了します。他のアプリケーションを終了させ、使用可能なメモリ容量を増やしてください。

付録

A COBOL アプリケーションカスタマイズの注意事項

COBOL アプリケーションカスタマイズが有効となるアプリケーションの実行形態は以下のとおりです
(有効な組み合わせが)。

機能	COBOL WS エミュレータ配下でない		COBOL WS エミュレータ配下	
	バッチ型アプリケーション	対話型アプリケーション	バッチ型アプリケーション	対話型アプリケーション
ウィンドウ1				
アプリケーションをアイコンの状態で行する	×		×	×
アプリケーションの背景色を白にする	×		×	×
アプリケーションの文字色を緑にする	×		×	×
ウィンドウ2				
ウィンドウを最大化表示する	×		×	×
ウィンドウ初期表示位置を指定する	×		×	×
タイトルバーを非表示にする	×		×	×
コントロールメニューを非表示にする	×		×	×
ウィンドウの最小化を不可能にする	×		×	×
ウィンドウの移動を不可能にする	×		×	×
ウィンドウサイズの変更を可能にする	×		×	×
ウィンドウの強制終了を不可能にする	×		×	×
タイトルバー文字列を変更する	×		×	×
ウィンドウ3				
ウィンドウサイズの既定値を変更する	×		×	×
起動時のウィンドウサイズを変更する	×		×	×
起動時の表示開始位置を変更する	×		×	×
動作				
先行キー入力を可能にする	×		×	×
英小文字を英大文字に変換	×		×	×

する				
日本語入力時の半角スペースを全角スペースに変換する	×		×	×
混在入力項目で日本語入力システムを起動する	×		×	×
キー押下による画面ハードコピーを無効にする	×		×	×
STOP 定数 / 実行時エラーを画面の最下行の1行下に表示する	×		×	×
数字項目入力時の桁数チェック方式を変更する	×		×	×
画面ハードコピー	×		×	×
カナ入力	×		×	×
プリンタ				
プリンタ定義				
ファイル				
RDB-STATUS への通知動作を指定する				
ファイル排他制御の指定				
CSV				
READ 文実行前の値を保持する				
区切り文字の指定				
数値データ入力時の不正文字をエラーにしない				
ダブルクォーテーションで囲まず出力する				
キーボード				
キー定義	×		×	×
フォント				
画面表示フォント	×		×	×
画面ハードコピーフォント	×		×	×
プリント印字フォント	×		×	×
サウンド				
鳴動ピープ音を変更する	×		×	×
その他				
小入力ファイル名を指定する				
小出力ファイル名を指定する				
改行位置を指定する				
OUTPUT モードで出力する				

COBOL
ユーザーズガイド

2014 年 10 月 初 版

2022 年 4 月 第 8 版

日本電気株式会社

東京都港区芝五丁目 7 番 1 号

TEL (03) 3454-1111 (大代表)

(C)NEC Corporation 2014, 2022

日本電気株式会社の許可なく複製・改変などを行うことはできません。

本書の内容に関しては将来予告なしに変更することがあります。