

# PowerBuilder マイグレーション手順書

最新バージョンへのマイグレーション

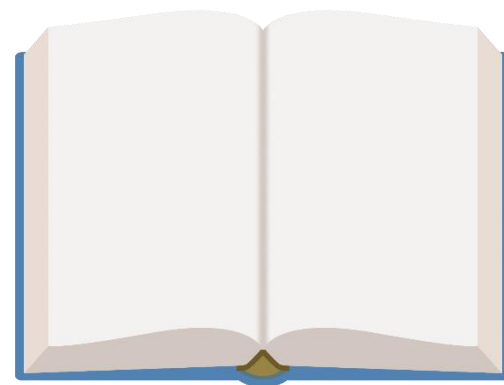


日本コンピュータシステム株式会社  
プラットフォームビジネス事業部

2020年4月30日

# 目次

- **はじめに**
  - 本資料について
  - マイグレーション作業の事前準備
  - マイグレーション作業の注意点
- **各バージョン毎の変更点**
  - PowerBuilder 7 以前からの変更点
  - PowerBuilder 8 ～ 10 の変更点
  - PowerBuilder 11 ～ 12.6 の変更点
  - PowerBuilder 2017 の変更点
- **マイグレーション手順**
  - 最新版 PowerBuilder へのマイグレーション
- **補足内容**
  - マイグレーション作業
  - マイグレーション後の作業



# はじめに

**nCS**  
日本コンピュータシステム株式会社



# 本資料について

- **本資料の目的**

- 本資料は PowerBuilder 日本語版を対象に、旧バージョンから最新バージョン<sup>※</sup>へマイグレーションを行う手順を紹介することを目的としております。

※ 2020 年 4 月末現在、日本語版の最新は PowerBuilder 2017 R3 となります。

- **本資料の内容**

- 本資料はマイグレーション手順を紹介するもので、本資料内に記載されている情報が全てではございません。詳細については、Appeon 社ウェブサイトに掲載されている Migrating PowerBuilder Applications をはじめとした製品マニュアルをご確認ください。

# マイグレーション作業の事前準備

- **Migrating PowerBuilder Applications の確認**

- Migrating PowerBuilder Applications には各バージョンで追加、変更された機能を始め、様々な情報が記載されておりますので、事前に一読されることを推奨しております。

[https://docs.appeon.com/appeon\\_online\\_help/pb2017r3/migrating\\_pb\\_apps/index.html](https://docs.appeon.com/appeon_online_help/pb2017r3/migrating_pb_apps/index.html)

- **サポート OS, DB の確認**

- マイグレーションに併せて利用する OS や DB 環境も変更される場合は事前に PowerBuilder でサポートされている OS や DB をご確認くださいことを推奨しております。

<https://japan.appeon.com/archive/document/powerbuilder-r3-support/>

※ サポート OS や DB については Appeon 社として動作確認を行っておりますが、動作保証を謳うものではなく、同サポート対象環境で利用する上で**不具合などの問題や要望があった場合に Appeon 社として対応することが可能な環境**となります。

# マイグレーション作業の注意点

- **バックアップ**
  - マイグレーションするアプリケーションは、必ずバックアップを取ってからマイグレーション作業を行ってください。
- **PowerBuilder .NET IDE からのマイグレーション**
  - PowerBuilder 12 より新たに追加された .NET IDE は、PowerBuilder 2017 で廃止されました。  
PowerBuilder 2017 をご利用の場合は PowerBuilder .NET にマイグレーションする前のソースからマイグレーション作業を行ってください。

# 各バージョン毎の変更点



# PowerBuilder 7 以前からの変更点

- **データウィンドウオブジェクトのネスティッドレポートへの名前の割り当て (PowerBuilder 7 以降の変更)**
  - データウィンドウ内の全オブジェクトに名前が必要となります。名前のないオブジェクトは、マイグレーション中にユーザー定義可能な接頭辞の設定 (通常は dw\_1, dw\_2 など) に基づいて名前が付与されます。名前は順番に付与されるため、すでに別のオブジェクトで使用されている場合は、予期せぬ現象が発生することがあります。この問題を回避するには、マイグレーション前に [デザイン | オプション] を選択して [接頭辞] タブからデータウィンドウの接頭辞を変更し、マイグレーション後に接頭辞を元に戻します。
- **ウィンドウアイコンの割り当て (PowerBuilder 7 以降の変更)**
  - ウィンドウはそのウィンドウを含むアプリケーションのアイコンを継承しません。アプリケーションのアイコンを使用するには、マイグレーション後に明示的にウィンドウにアイコンを割り当てる必要があります。



# PowerBuilder 8 ～ 10 の変更点

- **ワークスペースの追加 (PowerBuilder 8 以降の変更)**
  - PowerBuilder 8 からワークスペースという概念が追加され、編集対象となるアプリケーションオブジェクトを切り替えることなく複数アプリケーションのオブジェクトを同時に編集することが可能になりました。
- **予約語の追加 (PowerBuilder 8 以降の変更)**
  - 例外処理をサポートするために新しい予約語が追加されました。追加された予約語は TRY, CATCH, FINALLY, THROW および THROWS となり、いずれかを使用している場合には競合しない名前に変更する必要があります。
- **OpenSheet 関数の動作変更 (PowerBuilder 8 以降の変更)**
  - OpenSheet 関数と OpenSheetWithParm 関数は処理失敗時に -1 を返す代わりに実行時エラーが発生するように変更されました。
- **IsValid 関数の変更 (PowerBuilder 8 以降の変更)**
  - 以前のバージョンでは PowerObject に変換できない Any 型の引数が渡された場合にシステムエラーが発生しましたが、変更後は「false」を返します。

# PowerBuilder 8 ～ 10 の変更点

- **SystemError イベントの変更 (PowerBuilder 8 以降の変更)**
  - PowerBuilder 7 以前では、Error イベントや ExternalException イベントで捕捉できないエラーが発生した場合、SystemError イベントがただちに起動し、処理終了後エラー発生箇所に戻り、続きの処理が実行されていました。しかし、変更後はスクリプトが終了し、コールスタックが解放される動作に変更され、SystemError イベントが実行された後はエラー発生箇所の続きの処理が実行されません。
- **ScrollNextRow と ScrollPriorRow の動作変更 (PowerBuilder 8 以降の変更)**
  - ScrollNextRow と ScrollPriorRow は、下記の順でイベントが発生するようになりました。
    - RowFocusChanging
    - RowFocusChanged
    - ItemFocusChanged
    - ScrollVertical

# PowerBuilder 8 ～ 10 の変更点

- **ソースコード管理の変更 (PowerBuilder 8, 9 以降の変更)**

- PowerBuilder 8 で外部 SCC 準拠のソース管理システムへの直接的な接続が提供され、PowerBuilder 9 でさらに変更が行われました。

PowerBuilder 8 以前のソース管理プロジェクトを PowerBuilder 9 以降に移行する前に「ユーザズガイド」マニュアルの「ソース管理の使い方」をご確認下さい。

- **ScrollToRow の動作変更 (PowerBuilder 9 以降の変更)**

- ScrollToRow は RowFocusChanging イベントと RowFocusChanged イベントを発生させます。PowerBuilder 7 ではフォーカスが新しい行に移動後、両イベントが発生していましたが、PowerBuilder 9 から RowFocusChanging イベントでスクロールを取り消せるようになりました。その結果、RowFocusChanging イベントはスクロール前に発生し、RowFocusChanged イベントはスクロール後に発生するように変更されました。

# PowerBuilder 8 ～ 10 の変更点

- **Unicode への変更 (PowerBuilder 10 以降の変更)**
  - PowerBuilder で使用する文字コードが SJIS から Unicode へ変更されました。この変更によって、これまでバイト数単位で処理されていたものが文字数単位に変更されます。
- **文字列操作関数処理の変更 (PowerBuilder 10 以降の変更)**
  - これまで Len, Mid および Pos などの DBCS 文字列操作関数は戻り値としてバイト数を返したり、引数としてバイト数を使用していました。これが、Unicode 対応により戻り値を文字数で返し、引数として文字数を扱うように変更されました。
- **Date 関数の動作変更 (PowerBuilder 10 以降の変更)**
  - Date 関数に文字列を引数として指定すると PowerBuilder は PC の地域設定の日付書式に一致させようとしています。PowerBuilder 10 から完全一致が見つからない場合には部分一致を試みるように動作が変更されました。

# PowerBuilder 8 ～ 10 の変更点

- **ImportFile のサイズ変更 (PowerBuilder 10 以降の変更)**
  - Unicode 対応により ImportFile を使用して、非常に大きなテキストファイル (約 839,000 行) をデータウィンドウやデータストアにインポートする場合、エラーコード -15 が返されます。
    - ※ ファイルのサイズ制限は、列数と行数によっても異なります。
- **OpenTab 関数の動作変更 (PowerBuilder 10 以降の変更)**
  - OpenTab 関数と OpenTabWithParm 関数について、以前のバージョンでは Visible プロパティを「false」にしてもタブページが表示されていましたが、変更後は Visible プロパティを「true」にしないと表示されないように変更されました。

# PowerBuilder 11 ～ 12.6 の変更点

- **Windows Vista サポート (PowerBuilder 11 以降の変更)**
- **アイコンの変更 (PowerBuilder 11 以降の変更)**
  - ユーザーインターフェイスで使用されているアイコンの多くが変更されました。  
PowerBuilder 11 以降へアプリケーションをマイグレーションすると自動的に更新が行われます。
- **ツールバーの変更 (PowerBuilder 11 以降の変更)**
  - メニューペインタで、MDI フレームウィンドウと同様に独立したメインウィンドウにツールバーが追加できるようになりました。
- **Microsoft Office 2007 Excel フォーマットをサポート (PowerBuilder 11.5 以降の変更)**
  - PowerBuilder 11.5 よりデータウィンドウを .XLSX と .XLSB (Excel12) で保存できるようになりました。
- **Windows 7 サポート (PowerBuilder 11.5, 12 以降の変更)**

# PowerBuilder 11 ～ 12.6 の変更点

- **PowerBuilder Classic と PowerBuilder .NET (PowerBuilder 12 以降の変更)**
  - PowerBuilder 12 から 2 つの IDE が提供されました。これまでの PowerBuilder IDE は PowerBuilder Classic となり以前と同じ基本的な機能を保持しています。新しい IDE は PowerBuilder .NET と呼ばれ、Visual Studio Isolated Shell をホストし、.NET の共通言語仕様に準拠するように設計されています。
- **OriginalSize プロパティの追加 (PowerBuilder 12 以降の変更)**
  - コントロールの幅と高さを元の値に指定できます。
- **ランタイムパッケージの機能強化 (PowerBuilder 12 以降の変更)**
  - ランタイムパッケージで Microsoft Merge Module (MSM file) も作成できるようになりました。
- **FontHeight 関数の追加 (PowerBuilder 12 以降の変更)**
  - この関数は Classic IDE のみの機能となります。
  - カラムや計算フィールドの高さを調べたり、設定したりできます。

# PowerBuilder 11 ～ 12.6 の変更点

- **HasMinHeight プロパティの追加 (PowerBuilder 12 以降の変更)**
  - このプロパティは Classic IDE のみの機能となります。
  - 高さの自動調節プロパティを有効にした状態でデータがないとカラムは非表示になっていましたが、このプロパティで最小の高さを維持できます。
- **ウィンドウコントロールの透明色と「透明度」プロパティ (PowerBuilder 12.5 以降の変更)**
  - このプロパティは Classic IDE のみの機能となります。
  - ウィンドウコントロールの BackColor の透明色、ウィンドウコントロール用の「透明度」プロパティが新たに追加されました。
- **ASE 15.5 の新しいデータ型のサポート (PowerBuilder 12.5 以降の変更)**
  - ASE 15.5 の新しいデータ型となる BIGTIME, BIGDATETIME 型をサポートします。
- **Width.AutoSize プロパティの追加 (PowerBuilder 12.5 以降の変更)**
  - グリッド形式のデータウィンドウで Width.AutoSize プロパティを利用することでカラム内のデータに合わせたカラムの幅調節が可能です。



# PowerBuilder 11 ～ 12.6 の変更点

- 「タブ順序」, 「有効」および「フォーカス時の枠表示」プロパティのサポート (PowerBuilder 12.5 以降の変更)
  - PowerBuilder .NET IDE の WPF データウィンドウでは「タブ順序」と「有効」プロパティのみ利用できます。  
以下のデータウィンドウコントロールに追加されています。
    - ボタン (Show Focus Rectangle は適用されない)
    - 計算フィールド
    - グラフ
    - 画像
    - TableBlob
    - テキスト
- データウィンドウのユーザー描画コントロール (PowerBuilder 12.5 以降の変更)
  - このプロパティは Classic IDE のみの機能となります。
  - Paint 関数を利用してデータウィンドウの中で多角形, 矢印のヒント, 円グラフの扇形などのオブジェクトを描画できます。

# PowerBuilder 11 ～ 12.6 の変更点

- Windows 8 / 8.1 と Windows Server 2012 R2 サポート (PowerBuilder 12.6 以降の変更)
- JRE 並びに JDK の廃止 (PowerBuilder 12.6 以降の変更)
  - PowerBuilder で Java ランタイム環境 (JRE) または Java Development Kit (JDK) は使用せず、代わりに SAP JVM を使用しています。
- 新しい SoapConnection 関数の追加 (PowerBuilder 12.6 以降の変更)
  - 新たに EnablePreAuthentication と SetKeepAlive 関数が追加されました。
- PowerBuilder .NET IDE で EAServer ターゲットの未サポート (PowerBuilder 12.6 以降の変更)

# PowerBuilder 11 ～ 12.6 の変更点

- **OData データソースのサポート (PowerBuilder 12.6 以降の変更)**
  - DB プロファイルで OData インターフェイスを利用して OData サービスへのアクセスが定義できます。
  - データウィンドウのウィザードで OData データソースを選択できます。
- **64-bit アプリケーションの配布サポート (PowerBuilder 12.6 以降の変更)**
  - PowerBuilder Classic IDE で 64-bit アプリケーション配布がサポートされました。
  - Environment オブジェクトに新たに ProcessBitness プロパティが追加されました。
  - 新しいデータ型の longptr 型が新たに追加されました。
- **ドッキングウィンドウ機能の追加 (PowerBuilder 12.6 以降の変更)**
  - Classic IDE のみの機能となります。
  - MDI (Multiple Document Interface) フレームウィンドウ内で開くシートのドッキング動作を設定できます。

# PowerBuilder 2017 の変更点

- **Windows 10 と Windows Server 2016 サポート (PowerBuilder 2017 の変更)**
- **SAP JVM からの変更 (PowerBuilder 2017 の変更)**
  - PowerBuilder 2017 以降で SAP JVM は使用せず、代わりに Oracle JDK を使用しています。
- **リッチテキストエディットコントロールとリッチテキストデータウィンドウの変更 (PowerBuilder 2017 の変更)**
  - PowerBuilder 2017 以降では、リッチエディットコントロールとして新しい組み込みリッチテキストエディタが追加されています。
- **PB .NET IDE および WPF プロジェクトの廃止 (PowerBuilder 2017 の変更)**
  - PowerBuilder 2017 以降では、PowerBuilder .NET IDE および WPFプロジェクトが廃止されました。
- **Web DataWindow の廃止 (PowerBuilder 2017 の変更)**
  - PowerBuilder 2017 以降では、Web DataWindow (ActiveX) に関連するプロジェクト / ターゲット / ツールが削除されました。

# PowerBuilder 2017 の変更点

- **DBCS 文字列操作関数の変更 (PowerBuilder 2017 の変更)**

- PowerBuilder 2017 日本語版は、英語版に UI の日本語リソースと日本語版特有機能を追加した製品となるため一部の DBCS 文字列操作関数について変更があり、実行結果が PowerBuilder 12.6 以前の日本語版と異なります。

- **大文字 / 小文字の取り扱いの変更 (PowerBuilder 2017 の変更)**

- PowerBuilder 2017 日本語版は、アルファベットに代表される大文字 / 小文字が存在する文字を識別子や文字列操作関数等で使用した場合の動作が半角のみでなく全角も対象となる仕様に変更されています。

- **FillC / LeftC / MidC / RightC の廃止 (PowerBuilder 2017 の変更)**

- 将来廃止予定とされていた FillC, LeftC, MidC, RightC 関数は削除されました。

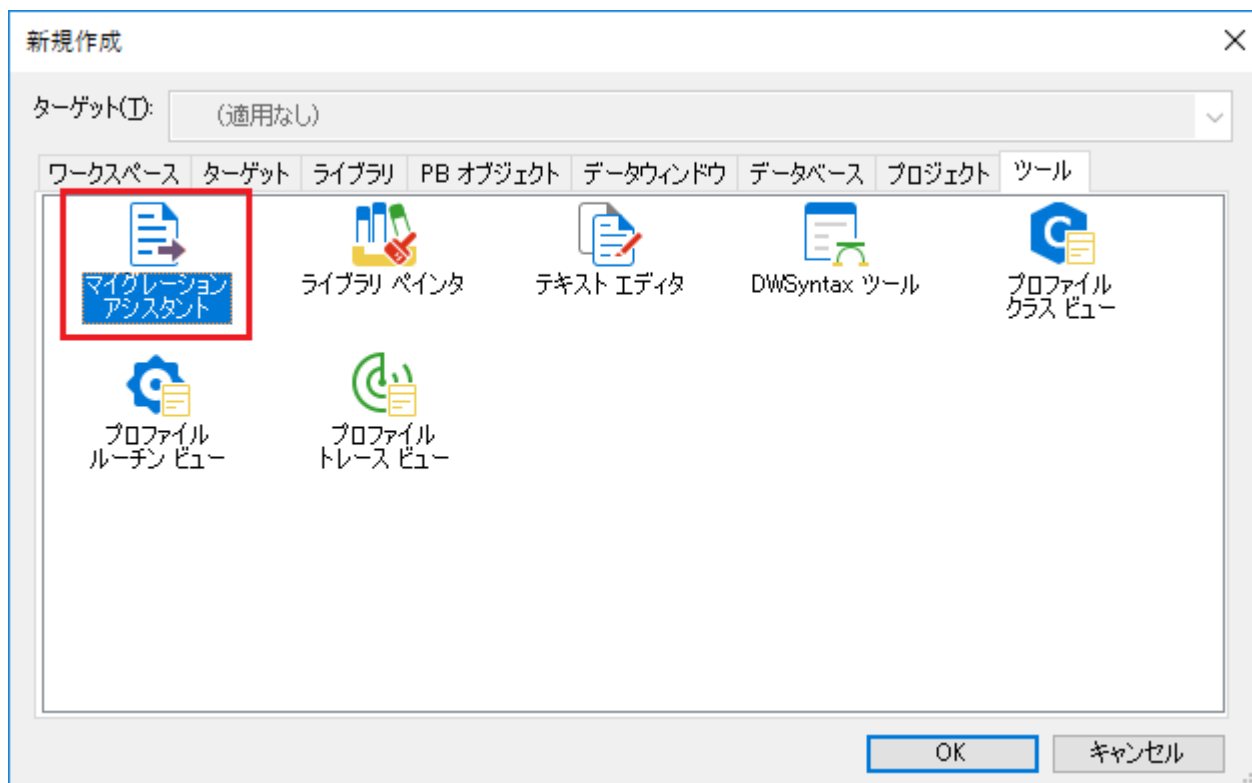
※ PowerBuilder 2017 に関するより詳細な変更点の情報につきましては  
「[SAP PowerBuilder、InfoMaker 12.6 \(日本語版\) からの変更点](#)」資料を確認してください。

# マイグレーション手順



# 最新版 PowerBuilder へのマイグレーション

- マイグレーションアシスタントの処理手順
  - 新規作成ウィザード画面を表示し、「ツール」タブから「マイグレーションアシスタント」を選択します。



新規作成ウィザード画面

# 最新版 PowerBuilder へのマイグレーション

- 図 1 の画面で検索対象を選択し、「次へ」ボタンを押下します。
- 図 2 の画面でマイグレーション対象となるアプリケーションのバージョンを選択し「次へ」ボタンを押下します。

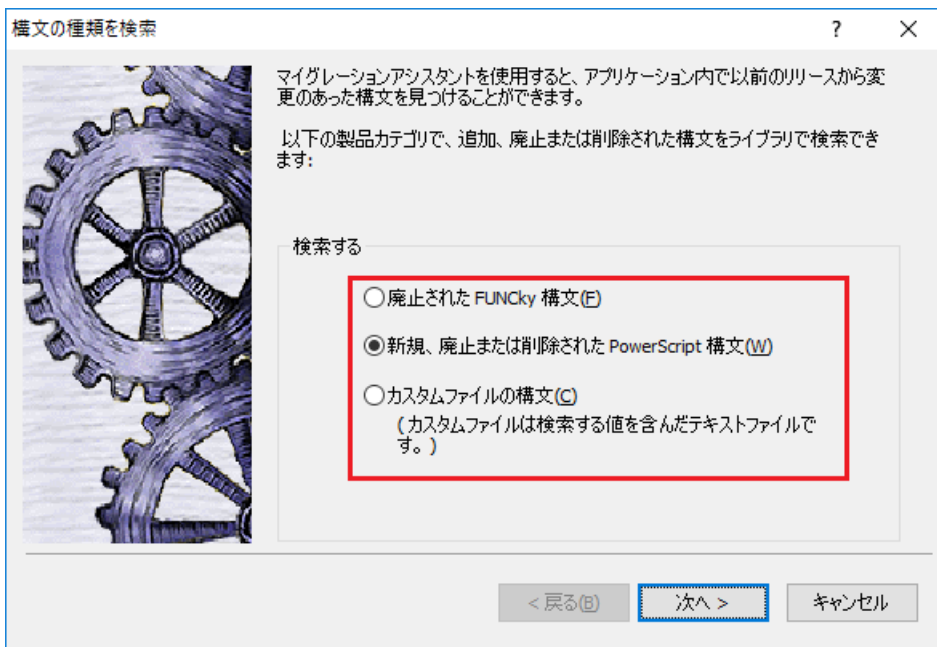


図 1 : 構文の種類を検索画面

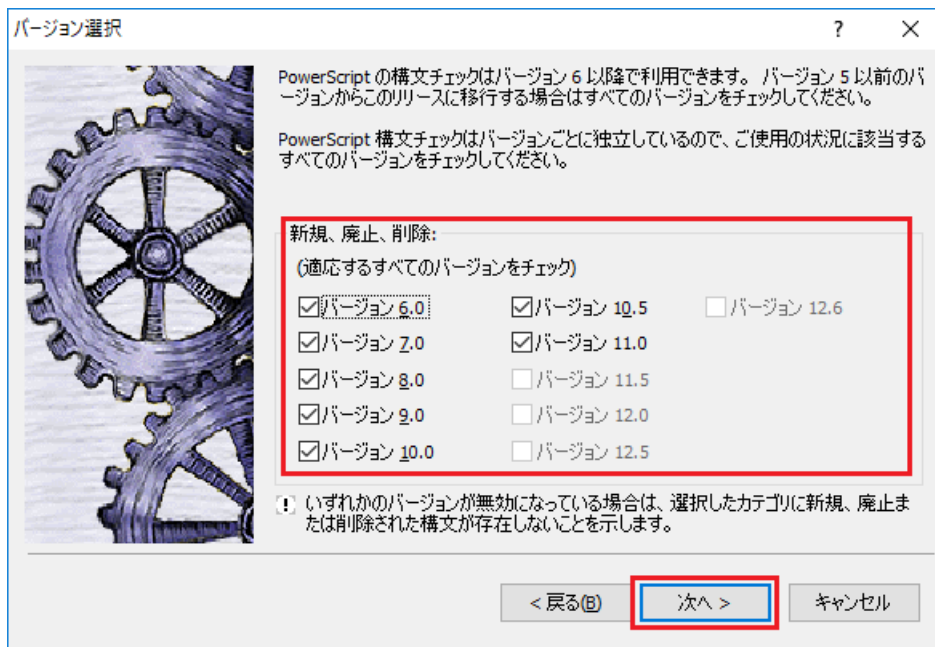


図 2 : バージョン選択画面



# 最新版 PowerBuilder へのマイグレーション

- 図 3 の画面でマイグレーションを行うライブラリを選択し、「次へ」ボタンを押下します。
- 図 4 の画面で内容を確認し、問題がなければ「完了」ボタンを押下します。

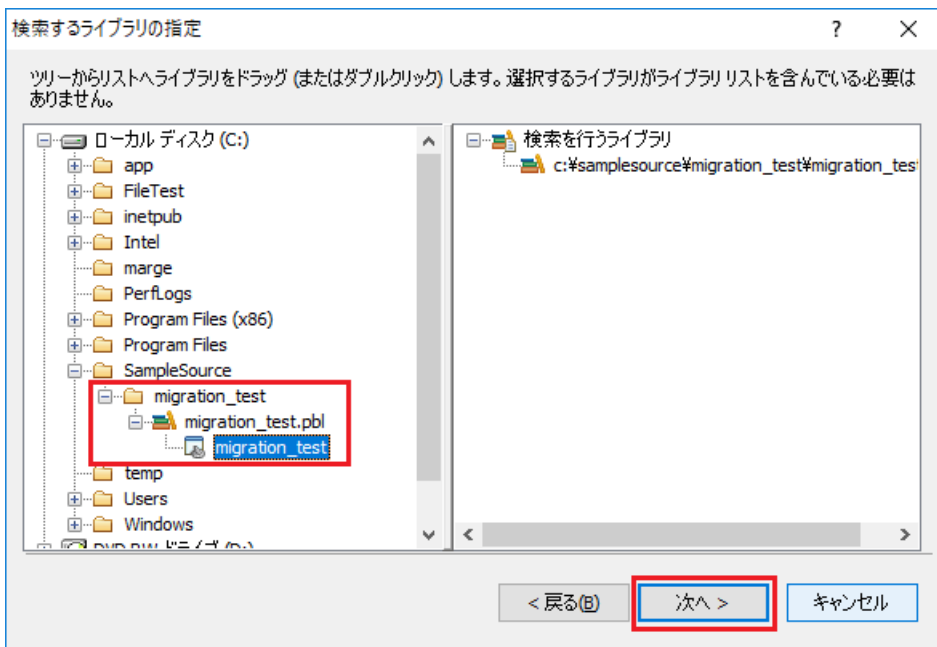


図 3 : 検索するライブラリの指定画面

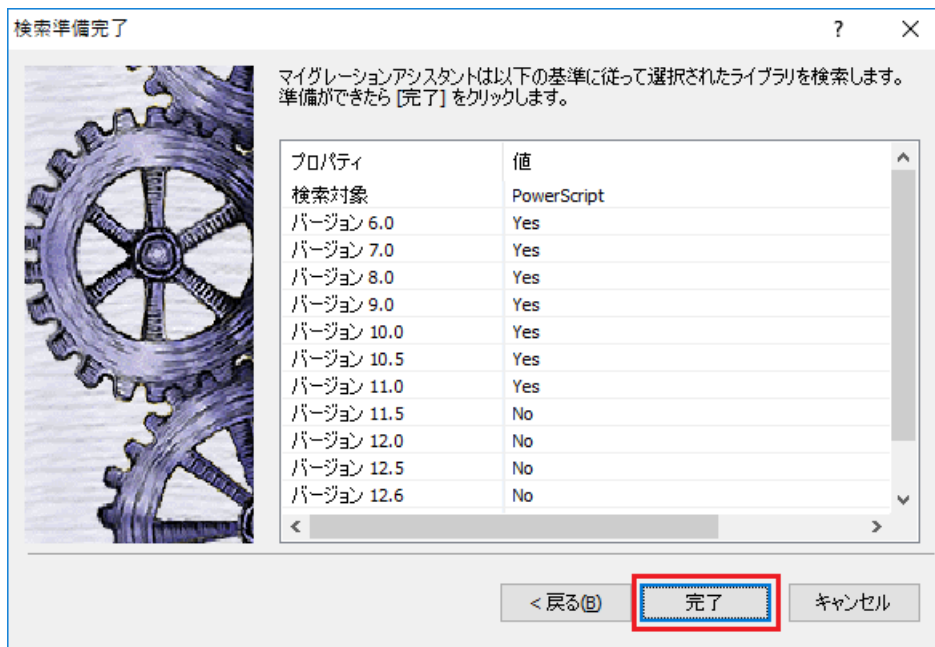


図 4 : 検索準備完了画面

# 最新版 PowerBuilder へのマイグレーション

- 最新バージョンで新規追加、廃止または削除の情報が出力されるので、マイグレーション作業時に発生したコンパイルエラーや警告を修復する際の参考情報として利用します。

マイグレーションアシスタント 検索結果

作成日: 2019/07/31 18:55:38 ページ 1 / 1

**PowerBuilder** マイグレーションアシスタント  
**PowerScript** 構文 (バージョン 6.0, 7.0, 8.0, 9.0, 10.0, 10.5, 11.0で、新規、廃止または削除)

ライブラリ	オブジェクト名	オブジェクトタイプ	見つかったコンテキスト	見つかった行	検索した値	置換すべき値
c:\samplesource\migration_test\migraticgf_migration_test		Function	Function Prototypes	s_sqlerrtext = leftw(s_errtext, l_cnt)	LeftW	Left ( string, n )
			Function Prototypes	if midw("", 1, 5) = "" then	MidW	Mid ( string, start {, length } )

マイグレーションアシスタント 検索結果画面

# 最新版 PowerBuilder へのマイグレーション

- ワークスペース、ターゲットを追加します。PowerBuilder 8 以降のバージョンからマイグレーションするお客様には影響ありませんので、次ページをご確認ください。
- PowerBuilder 7 以前のバージョンからマイグレーションする際には、ワークスペースに pbl を追加する必要があります。
- 図 5 の画面で「ワークスペース」を選択し、新しいワークスペースを作成します。
- 図 6 の画面で「既存のアプリケーション」を押下後、表示された画面でマイグレーション対象となる pbl を選択すると次ページの画面が表示されます。

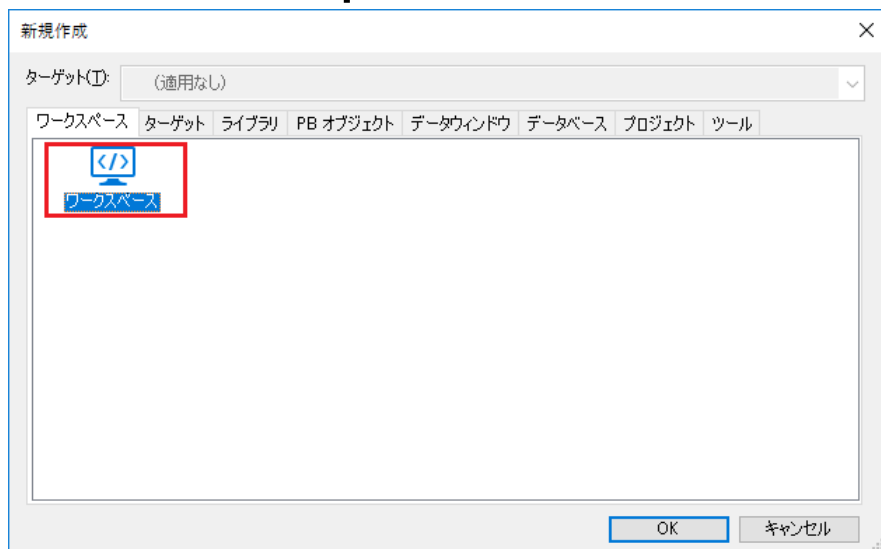


図 5：新規作成 (ワークスペース) 画面

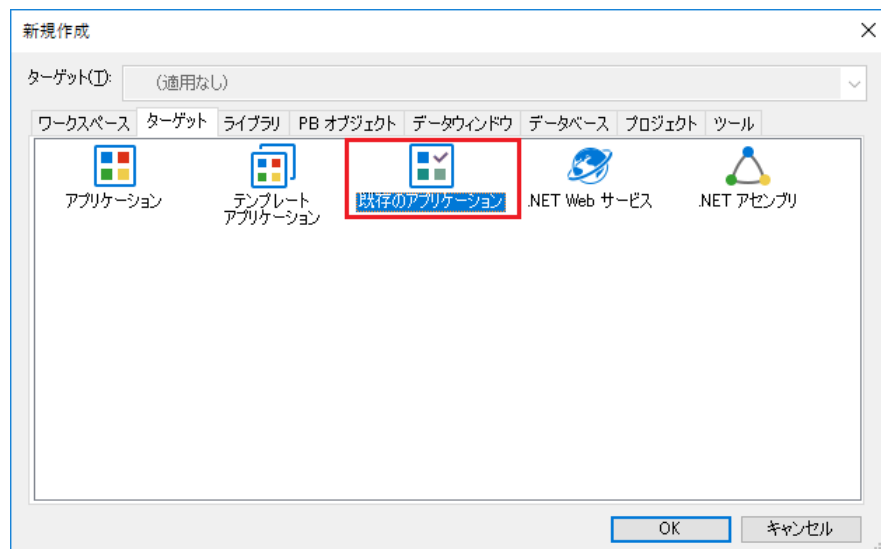


図 6：新規作成 (ターゲット) 画面

# 最新版 PowerBuilder へのマイグレーション

- 最新バージョンへマイグレーションするワークスペース (.pbw) ファイルを選択します。
- 図 7 の画面が表示されるので、「OK」ボタンを押下します。
- 図 8 の画面が表示されます。DBCS 文字列操作関数を自動変換する場合は、「DBCS 文字列を操作する関数を自動的に変換」をチェックします。自動変換しない場合は、チェックせずに「OK」ボタンを押下します。

※ データウィンドウ内の DBCS 文字列操作関数は自動変換されないため手動で修正する必要があります。

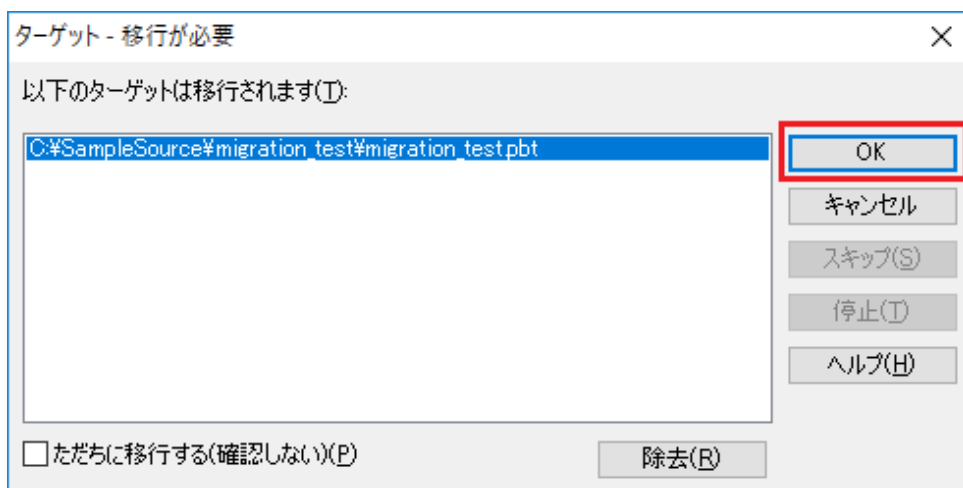


図 7 : ターゲット - 移行が必要画面

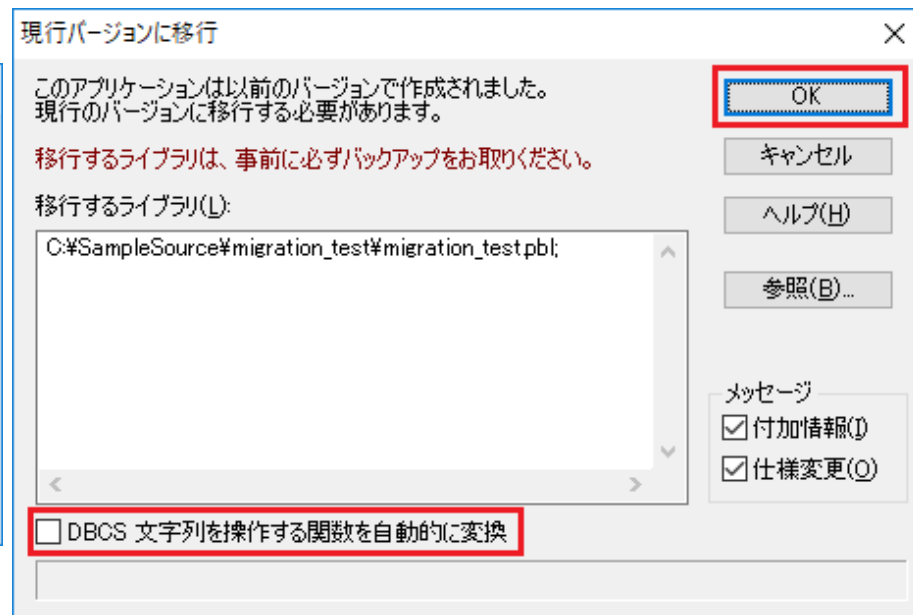


図 8 : 現行バージョンに移行画面

# 補足内容

**nCS**  
日本コンピュータシステム株式会社



# マイグレーション作業

- DBCS 文字列関数の自動変換

- 関数名の変換

- 関数名が x x x W → x x x
- 関数名が x x x → x x x A

(注) xxxC 関数は、PowerBuilder 2017 で廃止されました。その結果、xxxC 関数を含むオブジェクトはエラーが発生しマイグレーションされないため、マイグレーション前に修正を行うことを推奨します。

- グローバル外部関数の宣言で ANSI として取り扱うことを明示する構文に自動変換

- ALIAS FOR (“functionname;ansi”) 句が追加

- string データ型, char データ型, 構造体データ型を返す関数
- 引数として string 値, char 値, structure 値を使用する関数

```
migration_test.pbl(mig000).7: 情報 C0207: 以前のバージョンとの互換性を保つため、ALIAS FOR 句が'migration ncode' 関数に追加されました。
migration_test.pbl(mig000).8: 情報 C0207: 以前のバージョンとの互換性を保つため、ALIAS FOR 句が'lcmigration' 関数に追加されました。
migration_test.pbl(mig000).10: 情報 C0207: 以前のバージョンとの互換性を保つため、ALIAS FOR 句が'migration te128' 関数に追加されました。
migration_com.pbl(f_com_migration_comm).26: 仕様変更 C0185: 'LenW' は、以前のバージョンでサポートされていた関数です。将来のバージョンでの動作は保証されていません。
migration_com.pbl(f_com_migration_comm).28: 仕様変更 C0185: 'PosW' は、以前のバージョンでサポートされていた関数です。将来のバージョンでの動作は保証されていません。
migration_com.pbl(f_com_migration_comm).34: 仕様変更 C0185: 'MidW' は、以前のバージョンでサポートされていた関数です。将来のバージョンでの動作は保証されていません。
migration_com.pbl(f_com_migration_comm).38: 仕様変更 C0185: 'PosW' は、以前のバージョンでサポートされていた関数です。将来のバージョンでの動作は保証されていません。
migration_com.pbl(f_com_migration_comm).42: 仕様変更 C0185: 'MidW' は、以前のバージョンでサポートされていた関数です。将来のバージョンでの動作は保証されていません。
migration_com.pbl(f_com_del_migrm_space).32: 仕様変更 C0185: 'LenW' は、以前のバージョンでサポートされていた関数です。将来のバージョンでの動作は保証されていません。
migration_com.pbl(f_com_del_migrm_space).33: 仕様変更 C0185: 'MidW' は、以前のバージョンでサポートされていた関数です。将来のバージョンでの動作は保証されていません。
migration_com.pbl(f_com_del_migrm_space).37: 仕様変更 C0185: 'ReplaceW' は、以前のバージョンでサポートされていた関数です。将来のバージョンでの動作は保証されていません。
migration_com.pbl(f_com_del_migrm_space).41: 仕様変更 C0185: 'LenW' は、以前のバージョンでサポートされていた関数です。将来のバージョンでの動作は保証されていません。
```

# マイグレーション作業

## 出力結果の利用

- エラー内容に基づき、マイグレーションしたソースコードを修正します。
- エラーによって開けないオブジェクトは「ソースの編集」でスクリプトを開いて直接編集を行う必要があります。

```
デフォルト 警告 エラー
----- コンパイラ: 警告 (17:36:43)
migration_test.pbl(mig000).7: 情報      C0207: 以前のバージョンとの互換性を保つため、ALIAS FOR 句が'migration ncode' 関数に追加されました。
migration_test.pbl(mig000).8: 情報      C0207: 以前のバージョンとの互換性を保つため、ALIAS FOR 句が'lcmigration' 関数に追加されました。
migration_test.pbl(mig000).10: 情報     C0207: 以前のバージョンとの互換性を保つため、ALIAS FOR 句が'migration tel28' 関数に追加されました。
migration_com.pbl(f_com_migration_comm).26: 仕様変更 C0185: 'LenW' は、以前のバージョンでサポートされていた関数です。将来のバージョンでの動作は保証されていません。
migration_com.pbl(f_com_migration_comm).28: 仕様変更 C0185: 'PosW' は、以前のバージョンでサポートされていた関数です。将来のバージョンでの動作は保証されていません。
migration_com.pbl(f_com_migration_comm).34: 仕様変更 C0185: 'MidW' は、以前のバージョンでサポートされていた関数です。将来のバージョンでの動作は保証されていません。
migration_com.pbl(f_com_migration_comm).38: 仕様変更 C0185: 'PosW' は、以前のバージョンでサポートされていた関数です。将来のバージョンでの動作は保証されていません。
migration_com.pbl(f_com_migration_comm).42: 仕様変更 C0185: 'MidW' は、以前のバージョンでサポートされていた関数です。将来のバージョンでの動作は保証されていません。
migration_com.pbl(f_com_del_migrm_space).32: 仕様変更 C0185: 'LenW' は、以前のバージョンでサポートされていた関数です。将来のバージョンでの動作は保証されていません。
migration_com.pbl(f_com_del_migrm_space).33: 仕様変更 C0185: 'MidW' は、以前のバージョンでサポートされていた関数です。将来のバージョンでの動作は保証されていません。
migration_com.pbl(f_com_del_migrm_space).37: 仕様変更 C0185: 'ReplaceW' は、以前のバージョンでサポートされていた関数です。将来のバージョンでの動作は保証されていません。
```

```
デフォルト 警告 エラー
----- コンパイラ: エラー (17:36:43)
migration_com.pbl(f_c00_migration_test).62: エラー C0051: 関数 leftc は未定義です。
migration_com.pbl(f_c00_migration_test).74: エラー C0020: 式の中で戻り値を返さない関数を使用されています。
migration_com.pbl(f_c00_migration_test).84: エラー C0020: 式の中で戻り値を返さない関数を使用されています。
----- 終了: エラー (17:36:43)
```

# マイグレーション作業

- PowerBuilder バージョン間の非互換項目への対応

- DBCS 関数の処置

- グローバル外部関数として DBCS 関数を宣言している場合は、手作業で DBCS 関数名の修正を行う必要があります。
- DW オブジェクトの計算フィールド, 入力条件則, プロパティ式で使用されている関数は自動変換されないため、手作業で修正を行う必要があります。
- xxxC 関数を使用している場合は、PowerBuilder 2017 以降で廃止されているため手作業で修正を行う必要があります。
- PowerBuilder 2017 以降で MidA, RightA, PosA, FillA 関数を使用してマルチバイト文字を扱う場合、マイグレーション前後で一部動作が変更されています。そのため文字をバイト数で扱わないよう修正するか、マイグレーション前と同等の処理を行う関数を作成し手作業で関数を置き換える修正を行う必要があります。

関数の一例については下記をご参照ください。

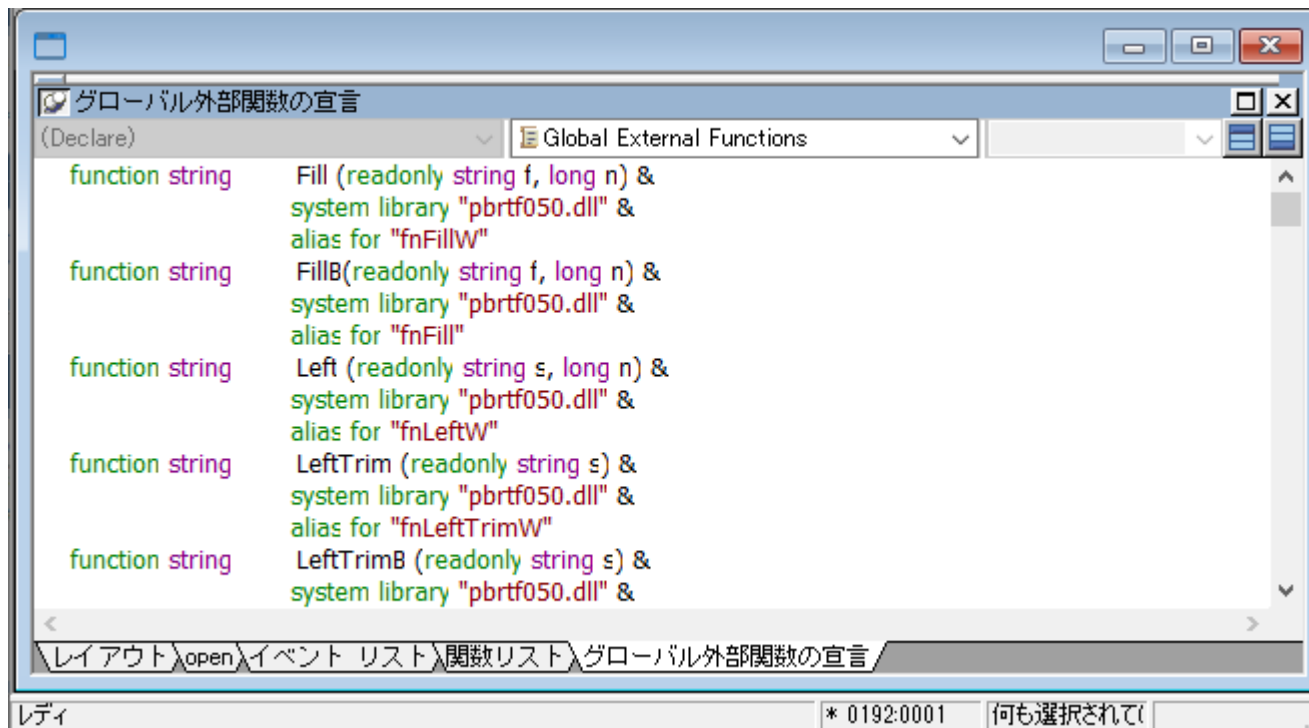
<https://japan.appeon.com/technical/techblog/technicalblog007/>



# マイグレーション作業

- DBCS 外部関数定義について

- グローバル外部関数定義で pbrtf050.dll, pbvm60j.dll が library に指定されていないかを確認します。
- 指定されていた場合はマイグレーション後に上記 dll が指定されている外部関数の定義を削除しアプリケーションの再構築を行う必要があります。



# マイグレーション作業

- DB に関する Unicode 対応について

- DB を Unicode 対応にする場合

- Unicode に対応したカラムでスキーマを定義し、再構築を行う必要があります。
- 他ホストへ SJIS コードデータを転送する機能がある場合には別途 Unicode から SJIS への変換機能を実装する必要があります。

- DB を Unicode 対応にしない場合

- DB のコードセットを SJIS として、クライアントにインストールされているミドルウェアが入力された Unicode を SJIS に変換します。
- Unicode 特有文字が入力された場合には SJIS に変換が出来ないため、文字化けが発生します。
- Unicode は 2 バイト文字のため、DB で定義されているバイト数より入力が多くなる場合があるので、入力可能文字数を判定する必要があります。

# マイグレーション作業

- SJIS で使用していない文字が入力可能になる問題

- PowerBuilder 9 以前ではシフト JIS に存在する文字のみ入力できましたが PowerBuilder 10 以降では Unicode 特有文字も入力可能です。

※ Unicode サロゲートペアは未サポートです。

- DB 側が Unicode に対応していない場合、例えば Unicode 文字を含む“鷗外”を入力して DB に登録しようとする、Unicode 文字は“?”で置き換えられます。

入力			DB更新結果
入力値	シフト JIS	Unicode	
鷗外	×	○	?外
鷗外	○	○	鷗外

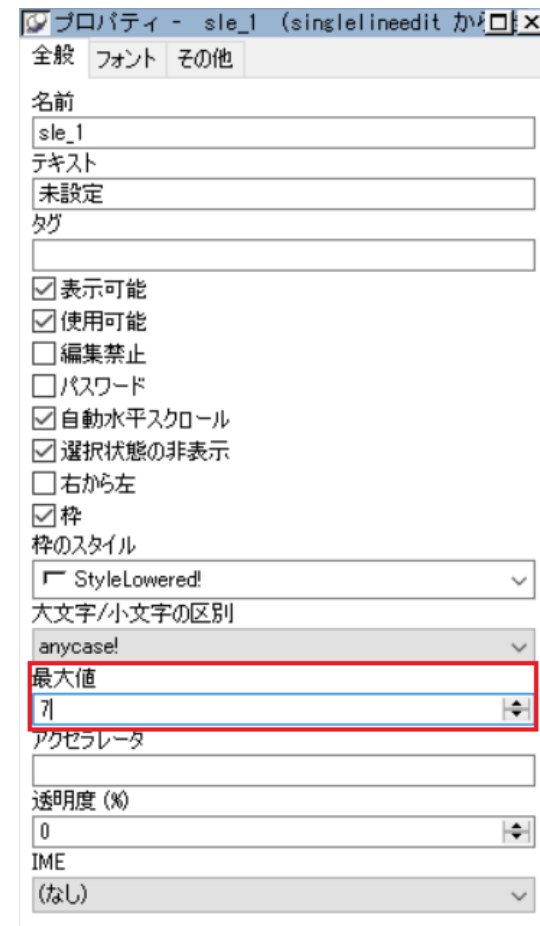
# マイグレーション作業

## • Unicode の影響

- PowerBuilder 10 以降は、シングルラインエディットなどの入力用のコントロールで入力可能な最大値の考え方が変更されます。
- プロパティでの指定は「バイト数」から「文字数」に変更されます。

### 文字列の長さの例

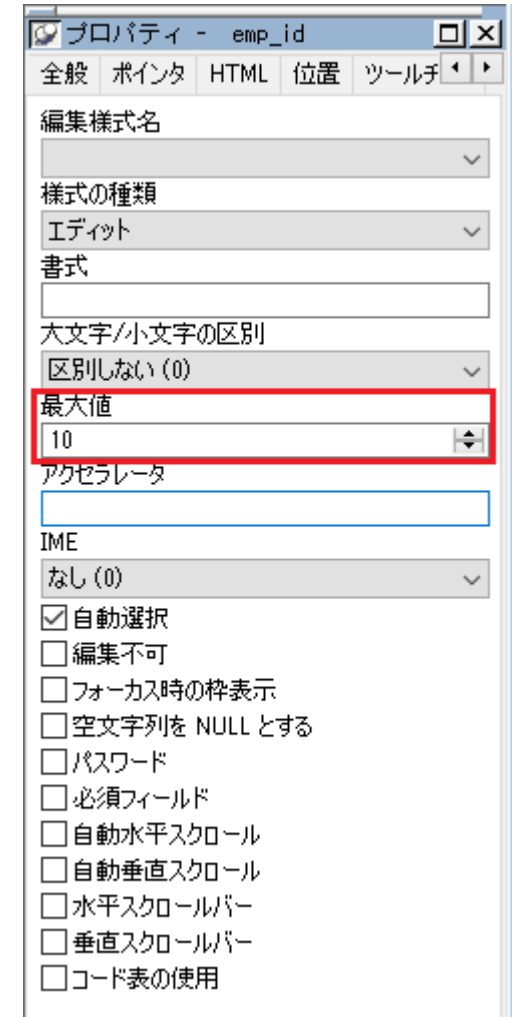
入力された文字	Ver.9 以前 (バイト数)		Ver.10 以降 (文字数)	
DATA	4	可	4	可
入力データ	10	不可	5	可
入力データ	8	不可	6	可
入力DATA	8	不可	6	可



# マイグレーション作業

## • Unicode の影響

- PowerBuilder 10 以降は、これまでバイト数で扱われていた部分が文字数で扱われるように変更されます。
- これは DataWindow の入力カラムについても同様で最大値プロパティが、バイト数から文字数への扱いに変更されます。
- この結果、全角 10 文字を入力すると 20 バイトとなり CHAR(10) で定義されたカラムに値を代入し DB を更新するとエラーが発生します。
- 対策として LenA 関数等を使用する方法が考えられます。(P.38 ~ 39 参照)



# マイグレーション作業

- **Unicode の影響による LenA 関数を使用した回避例**

- LenA 関数は引数の文字列を Unicode から SJIS に一度変換し文字列のバイト数を返す関数です。Unicode から SJIS へ変換する際に、SJIS に変換できない Unicode 特有文字がある場合にはエラーが返される (戻り値が -1) ため、Unicode 特有文字の判別にも利用できます。

- ※ 末尾に 'A' の付く DBCS 関数 (LenA, LeftA など) はすべて内部的に Unicode の文字列を SJIS に変換した文字列でバイト単位での処理が行われます。

# マイグレーション作業

- 回避策の例

- データウィンドウの入力条件則に、LenA 関数のチェックを設定する

- 例 (40 バイト以上の入力を NG とする場合)

LenA( GetText() ) <= 40 Or IsNull( GetText() )

	カラム名	データ型	条件指定	初期値	入力条件則	入力条件則エラー メッセージ	DB 識別子
1	entry_long	long	<input type="checkbox"/>				sample.entry_long
2	entry_string	char(40)	<input type="checkbox"/>		LenA( GetText() ) <= 40 Or IsNull( GetText() )	'入力可能な文字数は全角 20 文字までです.'	sample.entry_string

- editchanged イベントなどにチェック処理の Script を記述する

- 例 (40 バイト以上の入力を NG とする場合)

IF LenA( data ) >= 40 then

    Undo( )

End IF

# マイグレーション作業

- **String / Char 型について**

- PowerBuilder 10 以降で格納できるのは Unicode に存在する文字となります。
- PowerBuilder 9 以前は、Char 型は 1 バイトのみ格納できましたが PowerBuilder 10 以降は、1 文字を格納できるように変更されました。

- **Asc / Char 関数について**

- PowerBuilder 10 以降 Asc 関数は、文字列の最初の文字を Unicode コードポイント (文字に指定されている整数値) に変換するように変更されました。
- PowerBuilder 10 以降 Char 関数は、指定した文字列の先頭の Unicode 文字を返す、または Integer 型を Char 型に変換するように変更されました。
- PowerBuilder 10 以降 AscA / CharA 関数は、PowerBuilder 9 以前の Asc や Char 関数と同じ性質を持っています。



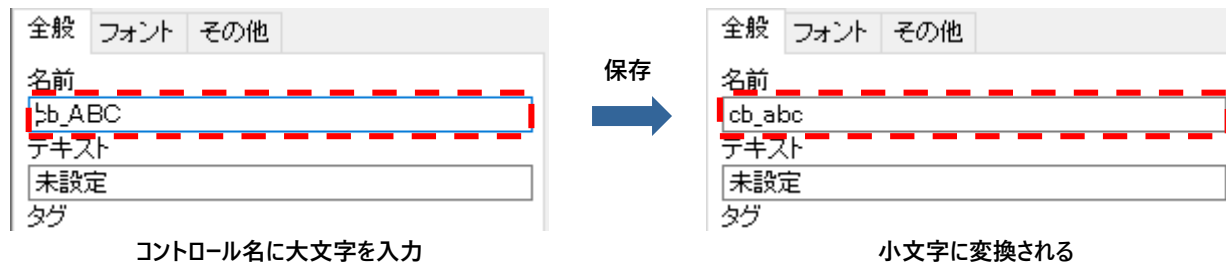
# マイグレーション作業

- **全角大文字 / 小文字の取り扱い**

- PowerBuilder 2017 日本語版は、アルファベットに代表される大文字 / 小文字が存在する文字を、識別子や文字列操作関数等で使用した場合の動作が、半角のみでなく全角も対象となる仕様となっています。
- コントロール等に指定する識別子の内部的な取り扱いや Upper / Lower / WordCap 関数の動作、Oracle インターフェースの「大文字 / 小文字の区別」オプション指定時の動作について、PowerBuilder 12.6 以前の日本語版と一部異なる箇所があります。

# マイグレーション作業

- 識別子 (コントロール, イベント, 引数などを参照するための名称) について
  - 半角は大文字と小文字を区別せず全角は区別していましたが、PowerBuilder 2017 日本語版では全角も大文字 / 小文字は区別されず内部的にすべて小文字で取り扱われます。



- ClassName 関数や DWObject の name プロパティで取得した識別子を条件式等で文字列として比較している場合、対象文字列を修正する必要があります。

```
if dwo.name = "b_BUTTON" then  
    this.SetItem(row, "QUANTITY", 0)  
    cb_UPDATE.enabled = false  
end if
```

- } Clicked イベントや ItemChanged イベント等で使用される dwo.name の値は小文字であるため、この判定は False となります。
- } SetItem 等でカラムを指定する場合や、コントロールを指定してメソッドの呼び出し、プロパティへのアクセスでは大文字でも影響はありません。

# マイグレーション作業

- Upper / Lower / WordCap 関数について

- PowerBuilder 12.6 以前の日本語版は、半角英字のみ大文字または小文字に変換する仕様でしたが、PowerBuilder 2017 では大文字と小文字の区別がある文字 (英字およびギリシア文字等) はすべて変換する仕様となっています。

引数に "A B C a b c XYZ xyz あいう ΔΠΛ ωσφ" を与えた場合の戻り値

Upper	PowerBuilder 12.6 日本語版	A B C a b c XYZ XYZ あいう ΔΠΛ ωσφ
	PowerBuilder 2017 日本語版	A B C A B C XYZ XYZ あいう ΔΠΛ ΩΣΦ
Lower	PowerBuilder 12.6 日本語版	A B C a b c xyz xyz あいう ΔΠΛ ωσφ
	PowerBuilder 2017 日本語版	a b c a b c xyz xyz あいう δπλ ωσφ
WordCap	PowerBuilder 12.6 日本語版	A B C a b c Xyz Xyz あいう ΔΠΛ ωσφ
	PowerBuilder 2017 日本語版	A b c A b c Xyz Xyz あいう Δπλ Ωσφ

英字(全角)      英字(半角)      日本語      ギリシア文字

- 同等の処理を行う場合は、半角英字のみ変換する代替関数を作成して置き換える修正を行う必要があります。

# マイグレーション作業

- Oracle DB インターフェイスについて

- 「大文字/小文字を区別」パラメーターがオフの場合、PowerBuilder 2017 日本語版は全角文字が大文字に変換されるため、二重引用符で囲まれたテーブル名 / カラム名は Oracle 側でエラーになる可能性があります。

テーブル名：TABLE カラム名：項目 N o 大文字/小文字の区別が**オフ**に対して下記SQLを実行した場合

**PowerBuilder 12.6 日本語版**  
SELECT "TABLE"."項目 N o" FROM "TABLE" 全角は記述されたまま送信するため、正常終了

SELECT "table"."項目 N o" FROM "table" →

**PowerBuilder 2017 日本語版**  
SELECT "TABLE"."項目 N O" FROM "TABLE" 全角も大文字へ変換され送信するため、カラム名不一致でエラー

- 「大文字/小文字を区別」パラメーターをオンにすることで PowerBuilder側では変換せずに Oracle 側へ送信できますが、SQL のテーブル名 / カラム名に二重引用符で囲まれた半角英小文字を指定していた場合、Oracle 側で大文字 / 小文字が区別されるため、エラーとなる可能性があります。この場合は、SQL に記述しているテーブル名 / カラム名を大文字に変更してください。

テーブル名：TABLE カラム名：項目 N o 大文字/小文字の区別が**オン**に対して下記SQLを実行した場合

SELECT "table"."項目 N o" FROM "table" 大文字へ変更する →

SELECT "table"."項目 N O" FROM "table" Oracle 側で大文字/小文字が判別されるためテーブル名不一致でエラー

# マイグレーション作業

- OS の変更に関して

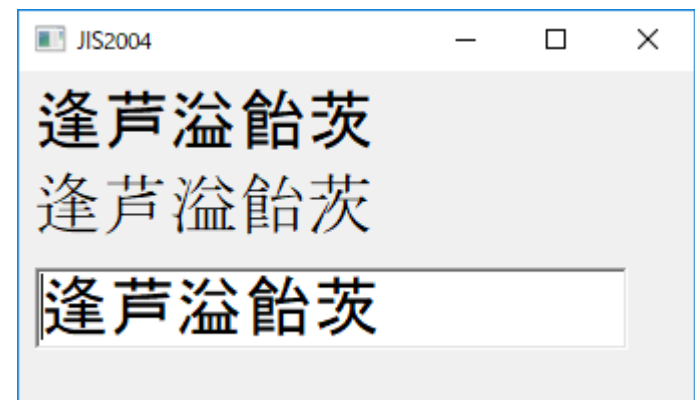
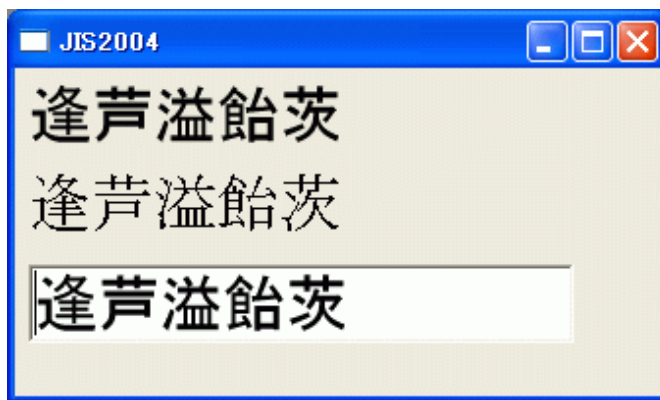
- Windows XP

- Unicode の領域は確保されていたものの、標準搭載されたフォントが非対応であったため、Unicode は Windows の内部処理だけで扱われています。
- 標準の文字コード : JIS X 0208 : 1990 (通称 JIS90)

- Windows Vista, 7, 8, 10

- フォントを含めすべて文字を Unicode で扱うことが可能になっています。Unicode のエンコーディング方式は "UTF-16" が使用されています。
- 標準の文字コード : JIS X 0213 : 2004 (通称 JIS2004)

※ OS の文字コードの差異により表示形式が異なる漢字が存在します。



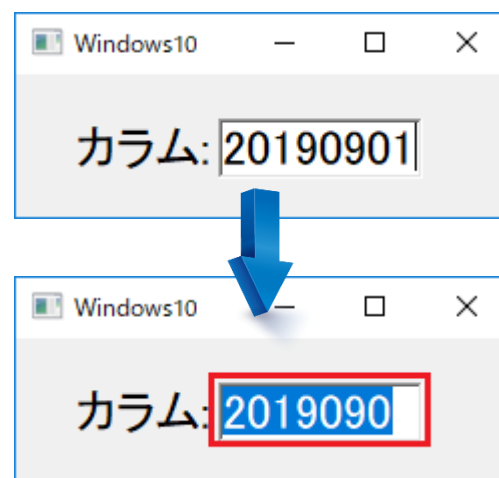
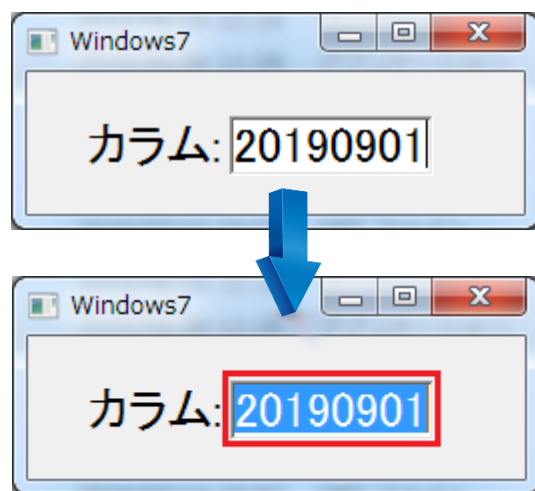
# マイグレーション作業

- OS の変更に関して

- PowerBuilder のマイグレーションと併せてアプリケーションを動作させる OS も変更する場合、OS のテーマの違いによってフォントやコントロールの大きさに差異が発生することがあります。その結果、変更前と比較してデータウィンドウのカラムやシングルラインエディットなどで領域内の表示可能な文字数が少なくなり、文字が切れて表示されることがあります。

この場合はコントロールの大きさを変更する、またはフォントサイズを小さくするなどの対応を行う必要があります。

**※ PowerBuilder アプリケーションを動作させる OS のみを変更した場合も発生することがあります。**



# マイグレーション後の作業

- **動作確認ポイント (仕様変更の可能性)**

- 旧バージョンからのマイグレーションや Maintenance Release (MR), Emergency Bug Fixes (EBF) を適用することで、マイグレーションや MR, EBF 適用前のアプリケーションと比較して関数などの**動作が変更されている可能性**があります。

- グローバル外部関数 (Windows API など) を使用している場合に関数の仕様が変更されている可能性があります。

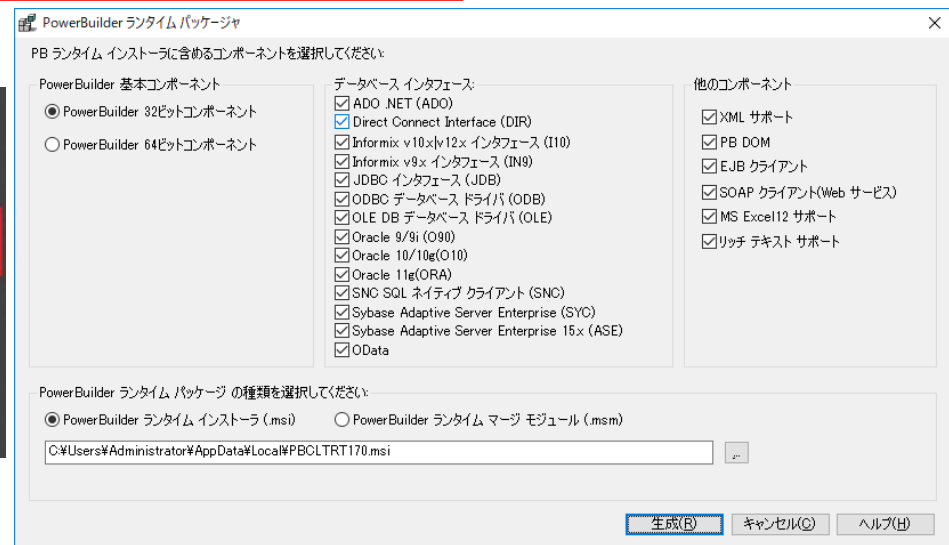
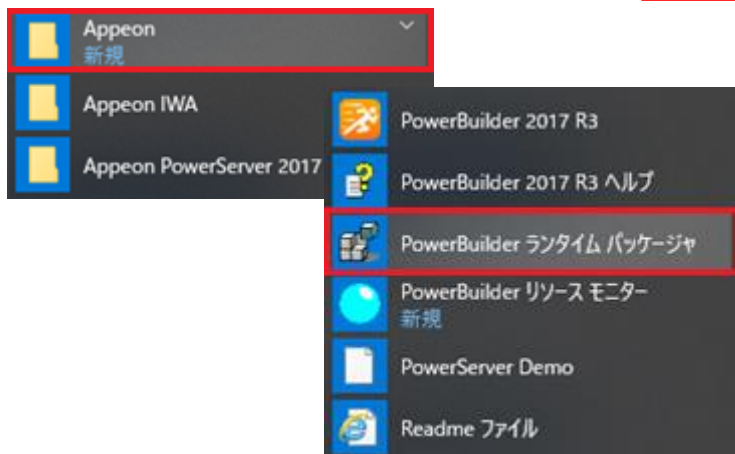
※ グローバル外部関数の仕様などについては、他社製品のため弊社で情報は持ち合わせておりません。各製品の提供元へお問い合わせ下さい。

上記の点から弊社ではマイグレーションや MR, EBF 適用後には現新システムの動作や表示に関する比較テストを行い**実行結果の確認**を実施すること推奨しております。

# マイグレーション後の作業

## ランタイムの配布

- PowerBuilder ランタイム パッケージを使用して PowerBuilder アプリケーションの実行時に必要な PowerBuilder ランタイムファイルをパッケージ化できます。
- PowerBuilder ランタイム パッケージで生成したインストール モジュールを実行する前に **atl100.dll**, **msvcr100.dll** および **msvcpr100.dll** が対象 PC 内に**インストールされている**ことを確認してください。
- マイグレーションや MR, EBF 適用後のアプリケーション配布時には、実行ファイルの再配布に併せ PowerBuilder **ランタイムファイルの再配布**も行ってください。





ご不明な点がございましたら、  
こちらまでお問い合わせ下さい。

 [powerbuilder-info@ncsx.co.jp](mailto:powerbuilder-info@ncsx.co.jp)

**ncs**

日本コンピュータシステム株式会社  
プラットフォームビジネス事業部