RAD Studioアプリケーションとバックエンドシステムを接続する

第32回 エンバカデロ・デベロッパーキャンプ

株式会社エンハンサー
代表取締役　藤田 和宏

embarcadero DEVELOPER CAMP

アジェンダ

- バックエンドシステムの状況
- ポストモダンERP
- DEMO(ソースコード解説と、実機と接続してのデモを行います)
  - DEMO概要
  - SAPシステムとの接続
  - データの抽出
  - データ登録
  - SAPジョブ実行
  - BWデータロード実行
  - BWデータロードモニタ
- 当社製品のご紹介　(時間が残れば)

バックエンドシステムの状況

バックエンドシステムの状況

- システムの乱立
- プロセスの分断
- 過度のカスタマイズによるレガシー化
- 機能の陳腐化

- ポストモダンERP

embarcadero DEVELOPER CAMP

ポストモダンERP

- Postmodern ERP
- Postmodern ERP is a technology strategy that automates and links administrative and operational business capabilities (such as finance, HR, purchasing, manufacturing and distribution) with appropriate levels of integration that balance the benefits of vendor-delivered integration against business flexibility and agility. This definition highlights that there are two categories of ERP strategy: administrative and operational.
- Administrative ERP Strategy. This focuses on the administrative aspects of ERP, primarily financials, human capital management and indirect procurement. Some industries don't need operational capabilities, such as manufacturing or distribution, so they focus their ERP strategy on administrative functions, perhaps augmented by some industry-specific functionality (such as grant management in the higher education and public sectors, or project resourcing, billing and costing in professional services). These industries are generally characterized as service-centric industries.
- Operational ERP Strategy. Organizations in manufacturing, distribution, retail, etc. (sometimes referred to as product-centric industries) are likely to extend their ERP strategy beyond administrative functions into operational areas, such as order management, manufacturing and supply chain, to maximize operational efficiencies. Also, asset-intensive organizations, such as utilities and mining, may include operations and maintenance of assets in their ERP strategy. These organizations can realize benefits from the integration between administrative and operational capabilities, for example, where operational transactions that have a financial impact are reflected directly in the financial modules.

出典：ガートナー「**Gartner IT Glossary > Postmodern ERP**」
http://www.gartner.com/it-glossary/postmodern-erp/

ガートナー社プレスリリースより引用

ガートナー　プレスリリース「ガートナー、2018年末までに、ポストモダン・アプリケーション統合戦略を持たない企業は90%に上るとの見解を発表」2016年3月9日 https://www.gartner.co.jp/press/html/pr20160309-01.html
ガートナーのリサーチを基にエンハンサーにて図表を作成

- 2018年末までに、ポストモダン・アプリケーション統合戦略を持たない企業は90%に上る

ERPアプリケーションのポートフォリオが複雑化するのに伴い、ポストモダン・アプリケーション統合戦略へのニーズが高まる

2017年までに、IT部門の75%は、バイモーダルな対応力を備えるようになるが、そのうち半数は作業に混乱を来し、ERPソリューションをリスクにさらす

2018年まで、企業の80%はポストモダンERP戦略を順調に実行する能力を欠く

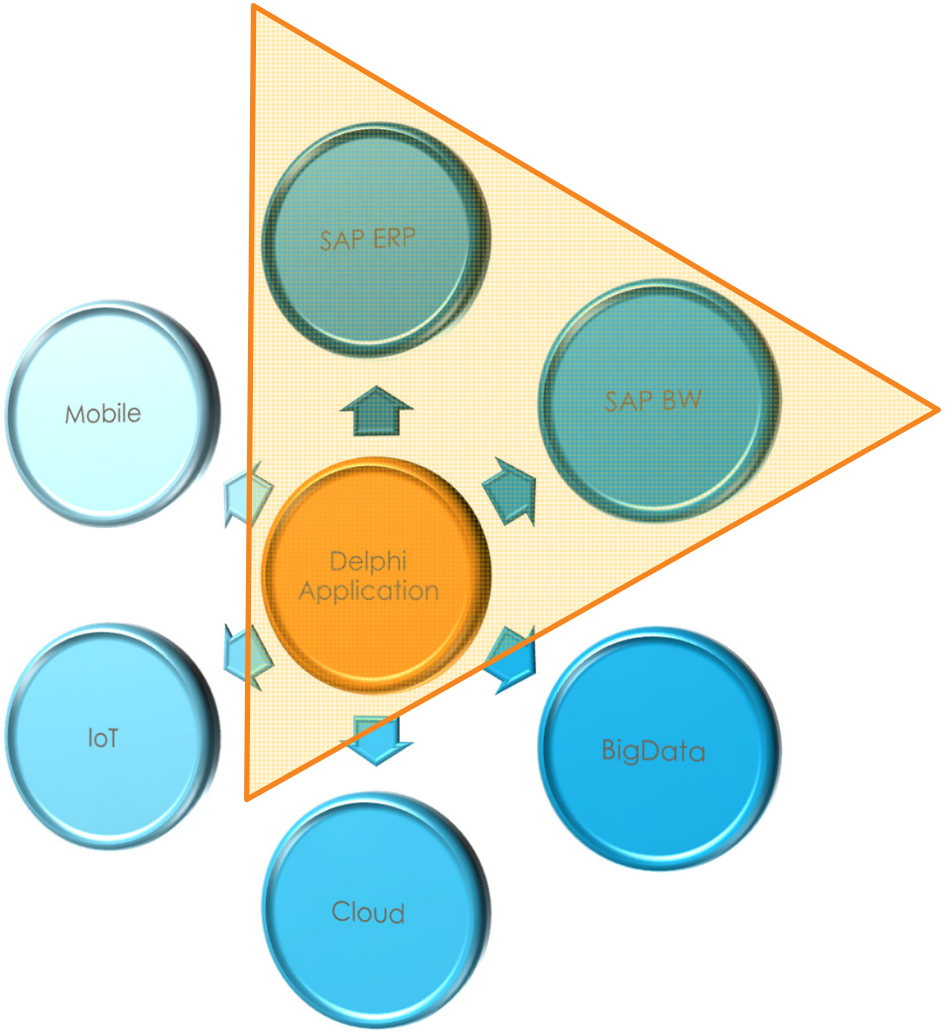
2018年までに、企業は、実証された価値を2年以内に提示するポストモダンERPプロジェクトの展開を求めるようになる

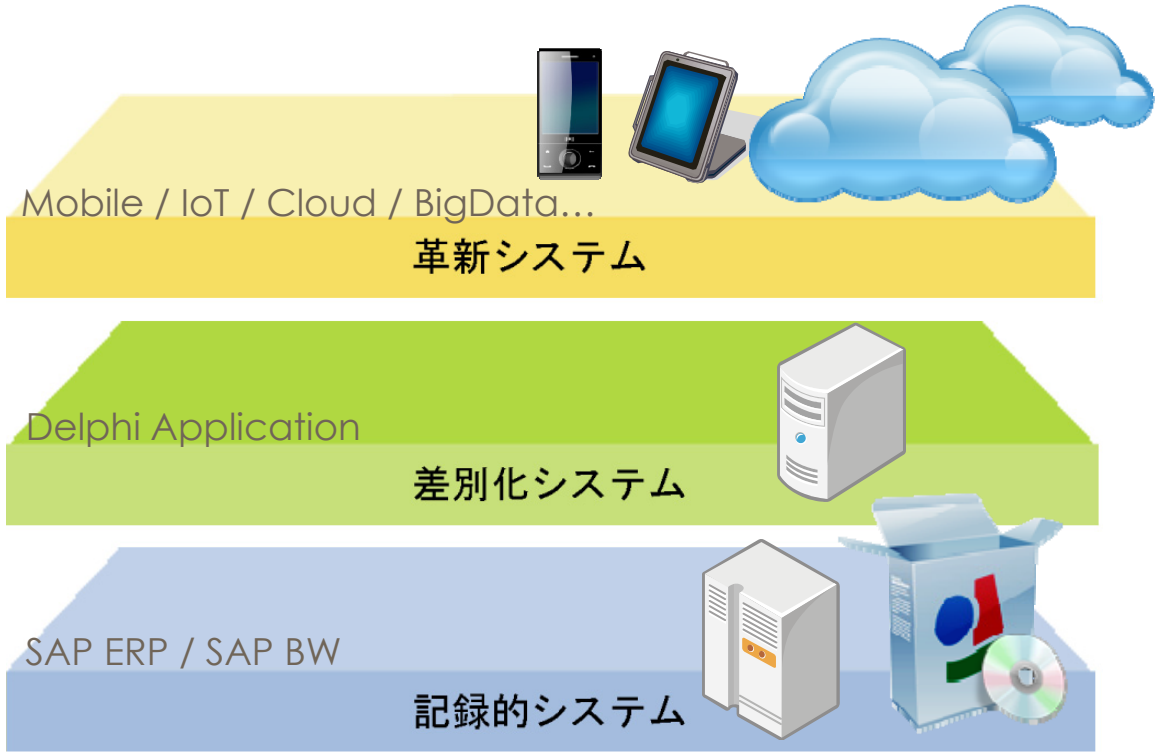
DEMO概要

- DelphiでSAPシステムと連携する方法

embarcadero DEVELOPER CAMP

DEMOで想定するシステム構成





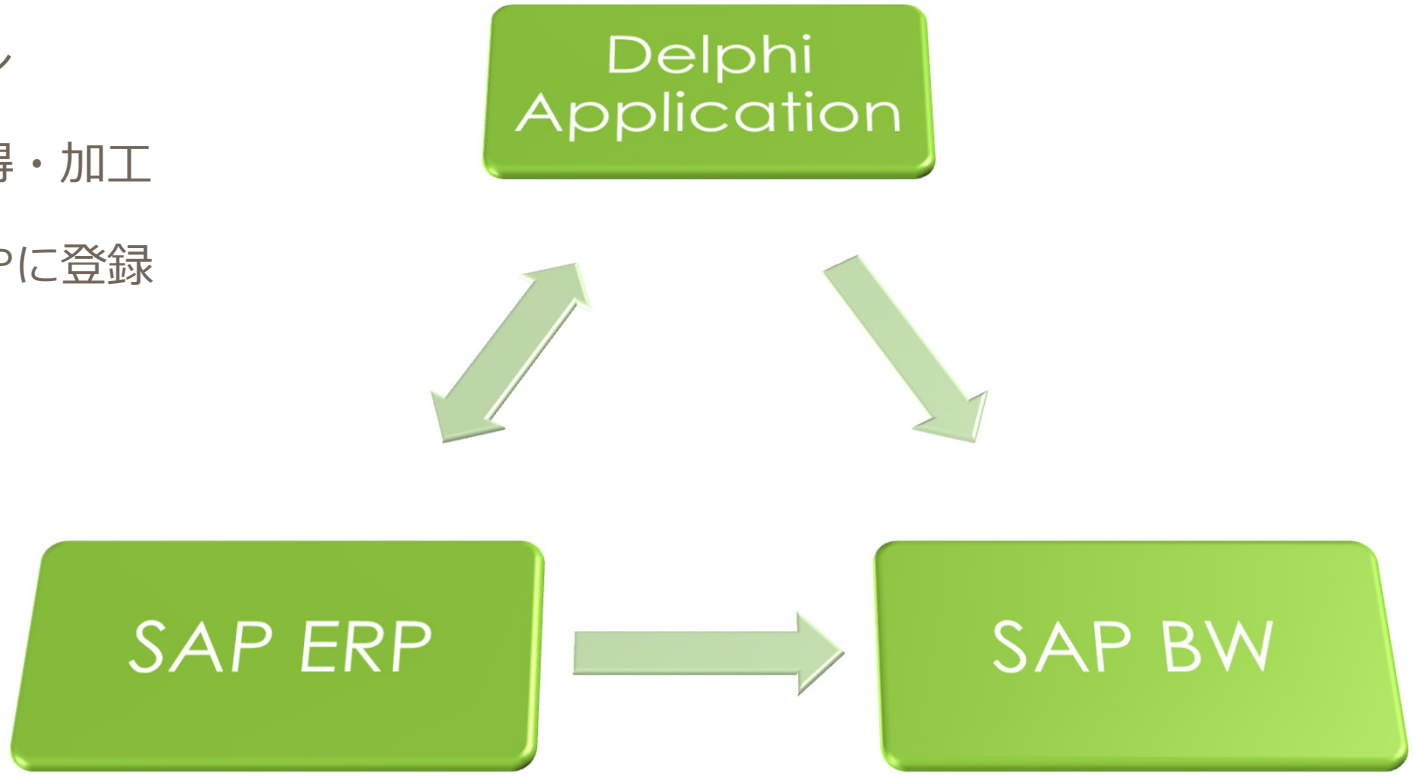
DEMOで想定する通信

DEMOで使用するポート

C:¥Windows¥System32¥drivers¥etc¥services

DEMOの範囲

- デモシナリオ
  - SAPシステムへのログオン
    - DEMO SAPシステムとの接続 1, 2
  - SAP ERPからデータを取得・加工
    - DEMO データ抽出 1, 2
  - 加工したデータをSAP ERPに登録
    - DEMO データ登録
  - SAP ERPのジョブを実行
    - DEMO SAPジョブ実行
  - SAP BWにデータをロード
    - DEMO BWプロセスチェーン実行
  - ロード状況を監視
    - DEMO BWデータロードモニタ



- DEMO環境準備

embarcadero DEVELOPER CAMP

DEMO環境準備

DEMO環境準備

- SAP サーバ インストール

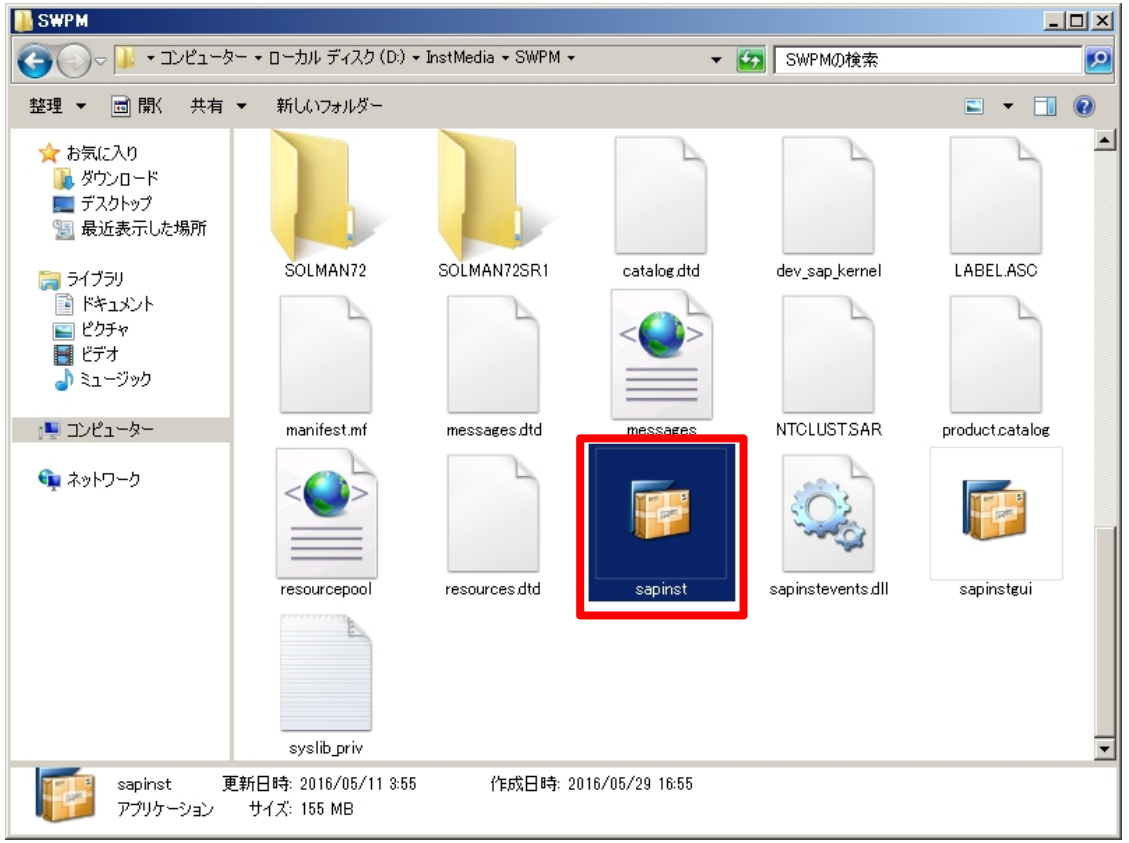
embarcadero DEVELOPER CAMP

SAP サーバ インストール

- ライセンスをお持ちであることが前提です
  - Partner Edge
  - SAP Open Ecosystem Program
  - Developer & Trial Editions
  - ・・・
- インストールメディアはSAP Support Portalから
  ダウンロードしてください

SAP サーバ インストール

- sapinstを実行し、少しパラメータを入力して待てばインストール完了
  - Installするだけなら難易度は低い
  - 本稼動に耐えられる設定は難しい
  - インストールには12～24時間ほどかかる (16core 32GB)
  - お試し環境であれば、IDES版がお勧め (組織や品目等の設定がされている)



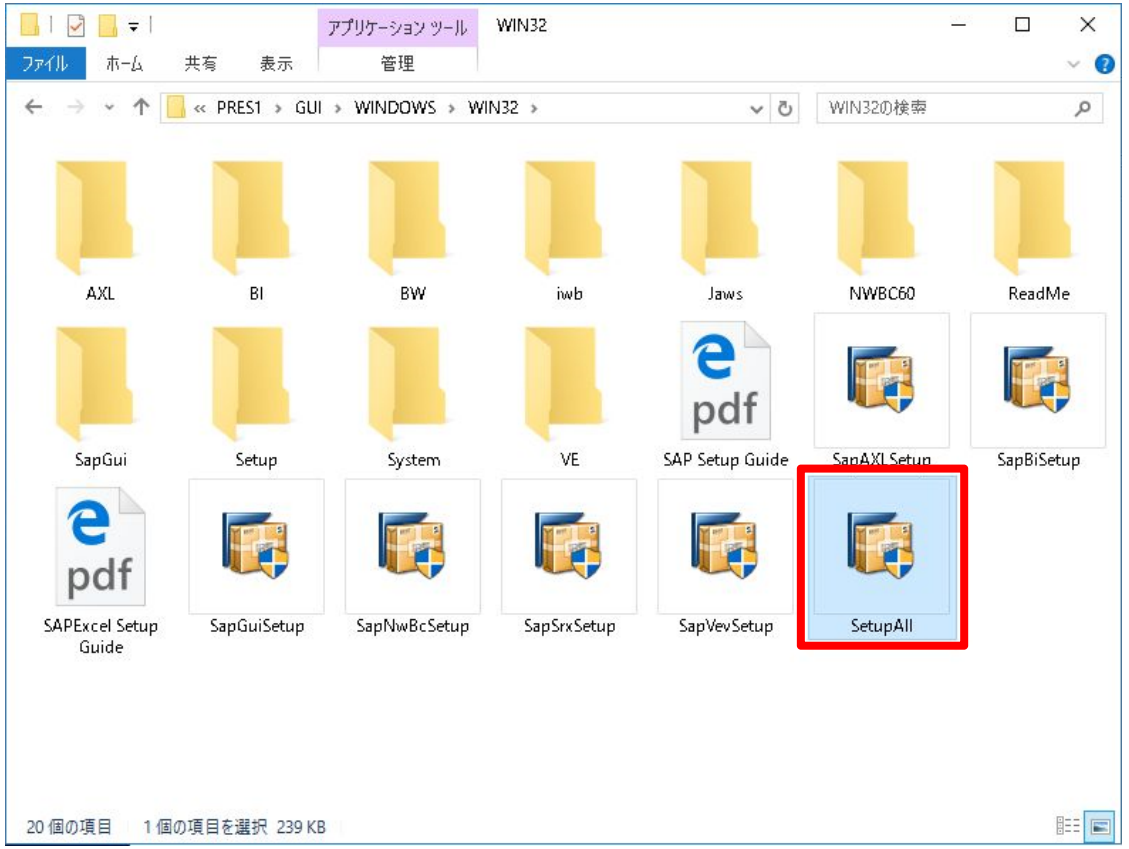
DEMO環境準備

- SAP Front Endインストール

embarcadero DEVELOPER CAMP

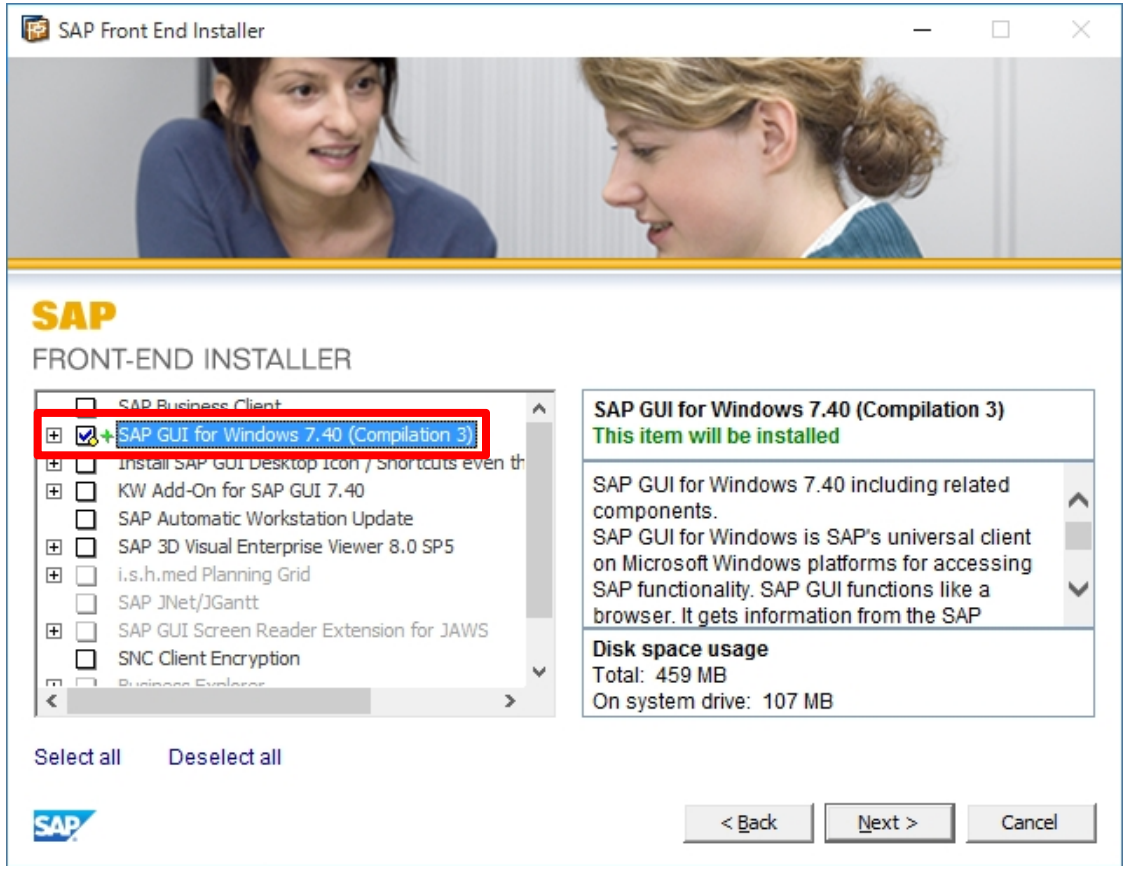
SAP Front End インストール

- Setupを実行するだけ…
  - 通常はBASISチームから指示された場所にあるSetupを実行するだけだと思います。



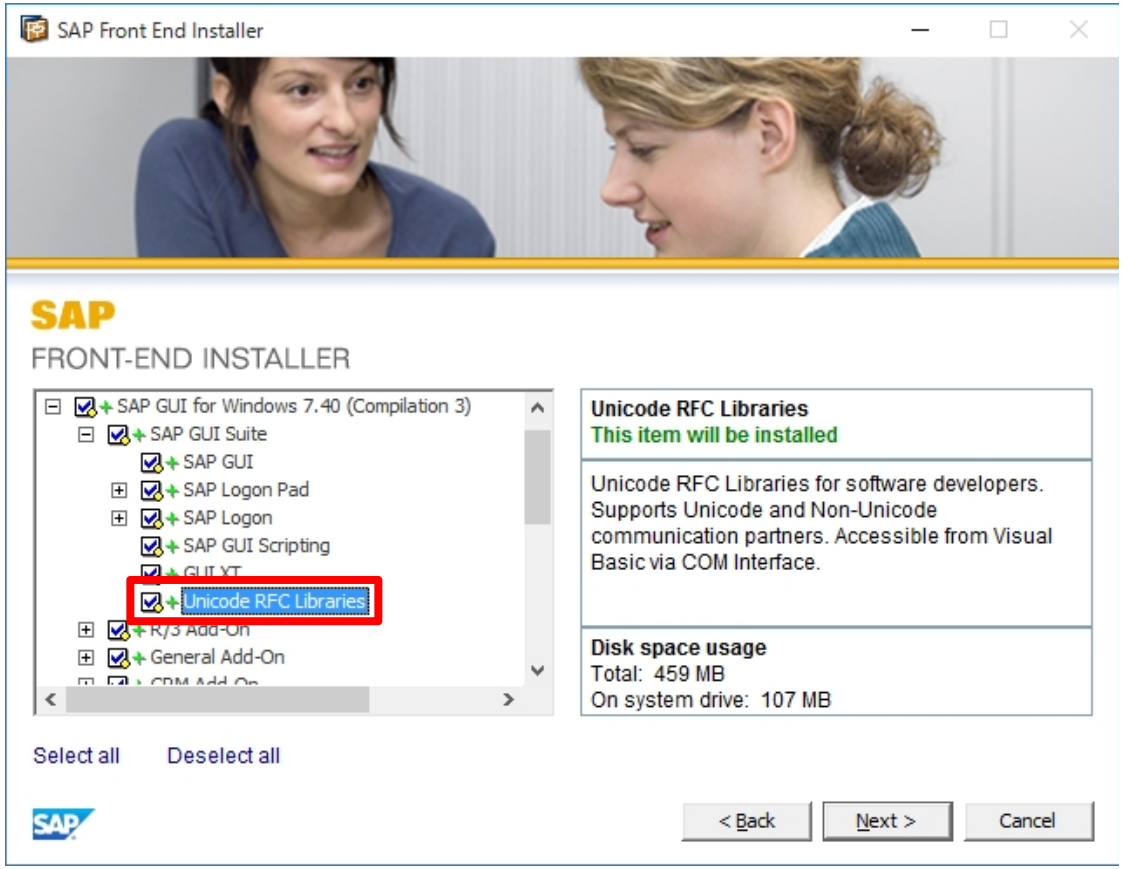
SAP Front End インストール

- SAP Front Endインストール時の注意点
  - 多くの場合、「SAP GUI for Windows 7.40」を選択しインストールしていると思います
  - 実はこの階層の下には…



SAP Front End インストール

- SAP Front Endインストール時の注意点
  - 「Unicode RFC Libraries」というものが含まれています
  - 今回説明する方法では、このライブラリを使用します



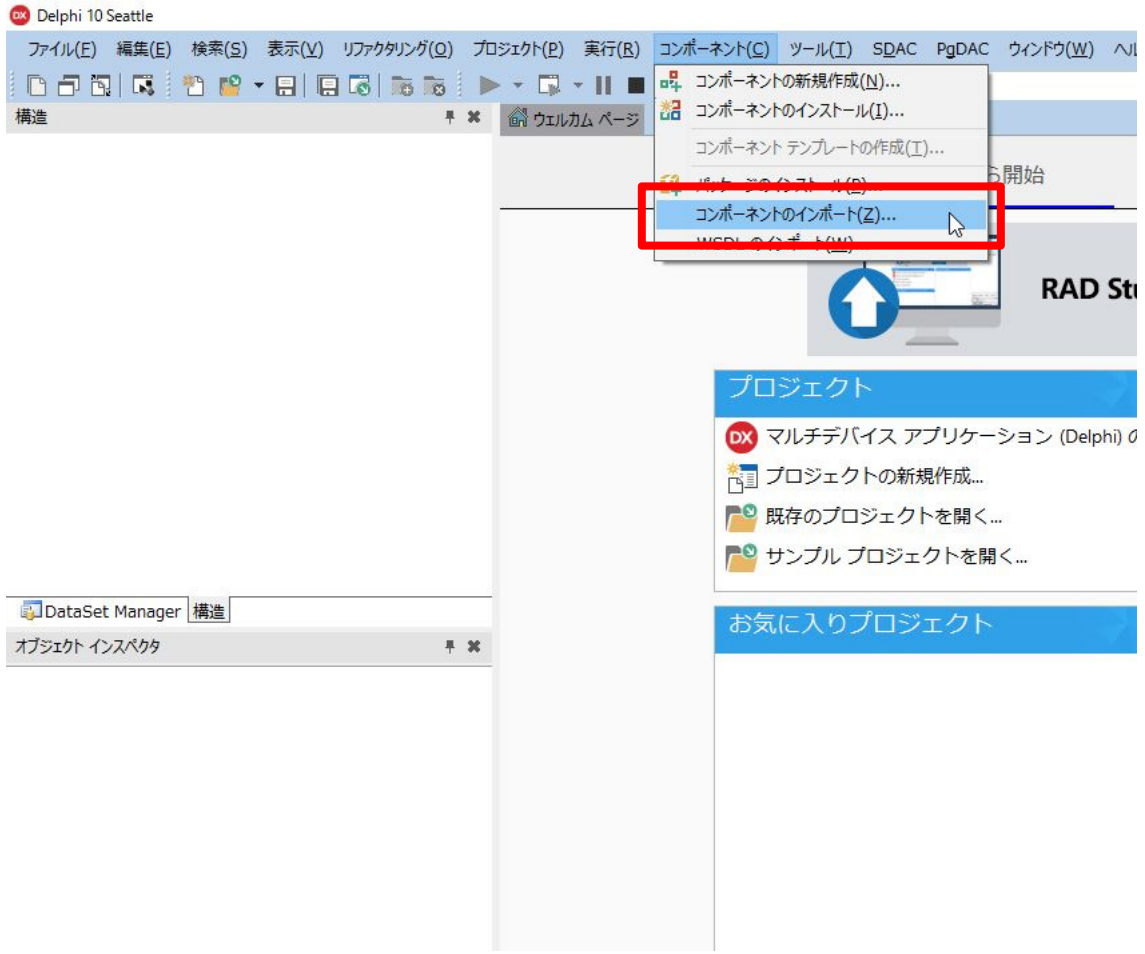
DEMO環境準備

- RAD Studio 設定

embarcadero DEVELOPER CAMP

RAD Studio 設定

- 「Unicode RFC Libraries」を使用するため、「コンポーネントのインポート」を行います



RAD Studio 設定

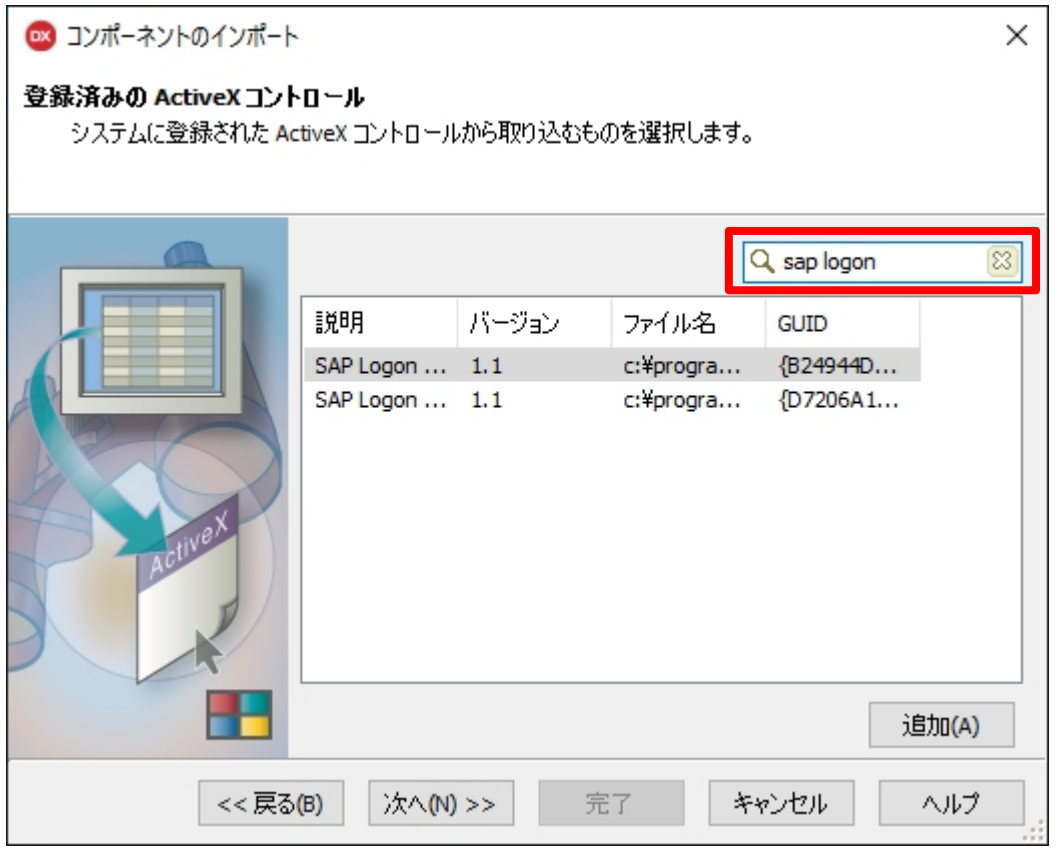
- 今回の説明は「Active Xコントロールの取り込み」で進めます

RAD Studio 設定

- 大量のコントロールから選択することになるのですが…
- 右上の検索で絞込みができます

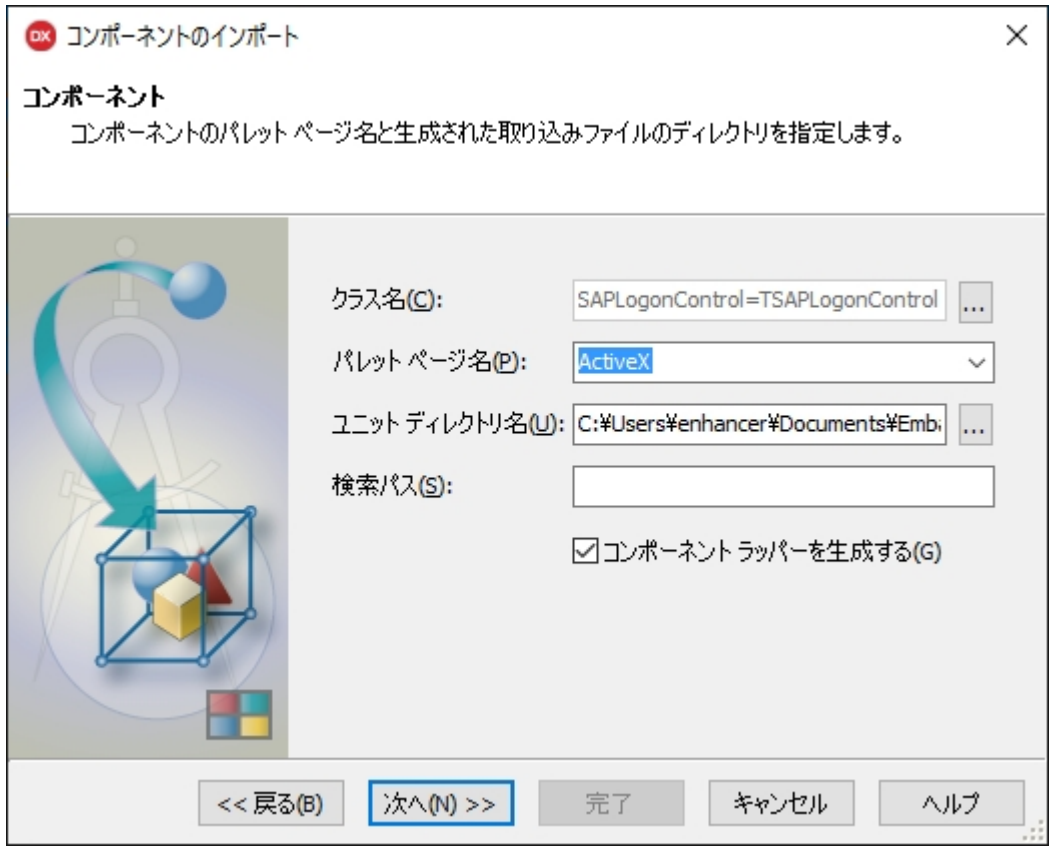
RAD Studio 設定

- 検索に　sap logon　と入力します
- 以下に絞り込まれます
  - SAP Logon Control
  - SAP Logon Unicode Control
- ご使用の環境に合わせて、Unicode版、Non Unicode版を選択してください



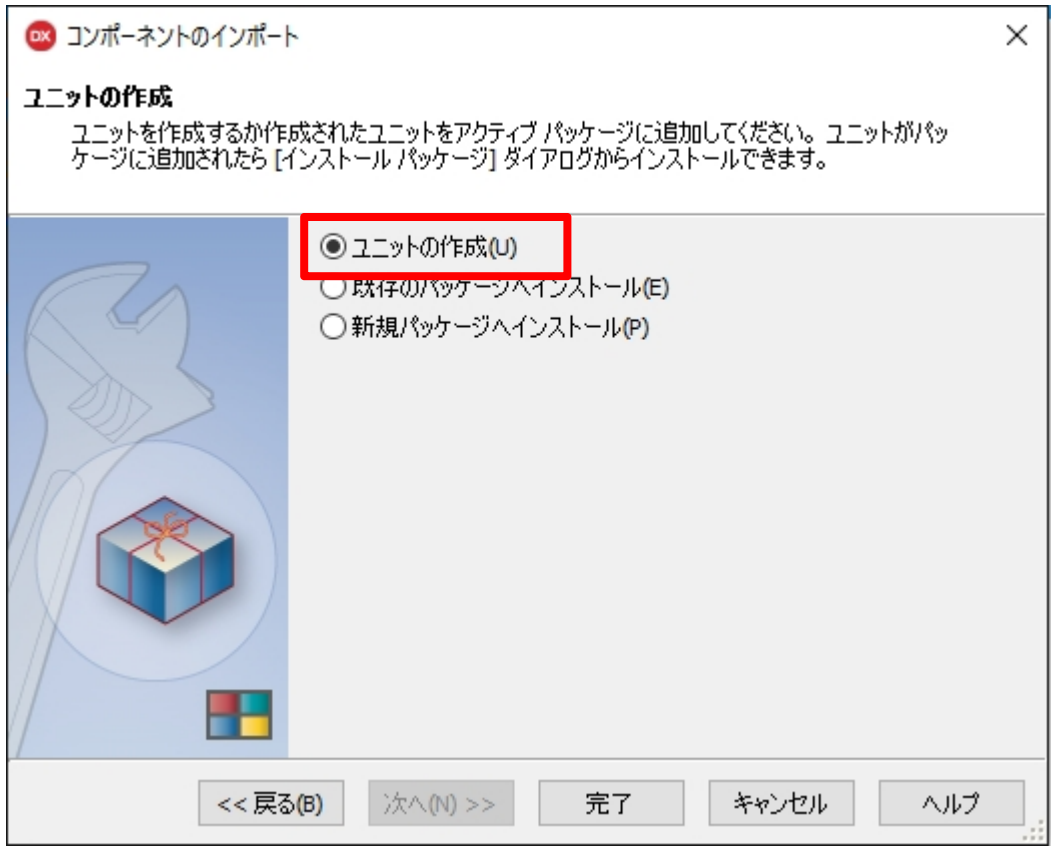
RAD Studio 設定

- パッケージページ名、ユニットディレクトリ名等を指定します。
- 今回の説明では初期値のまま進めます



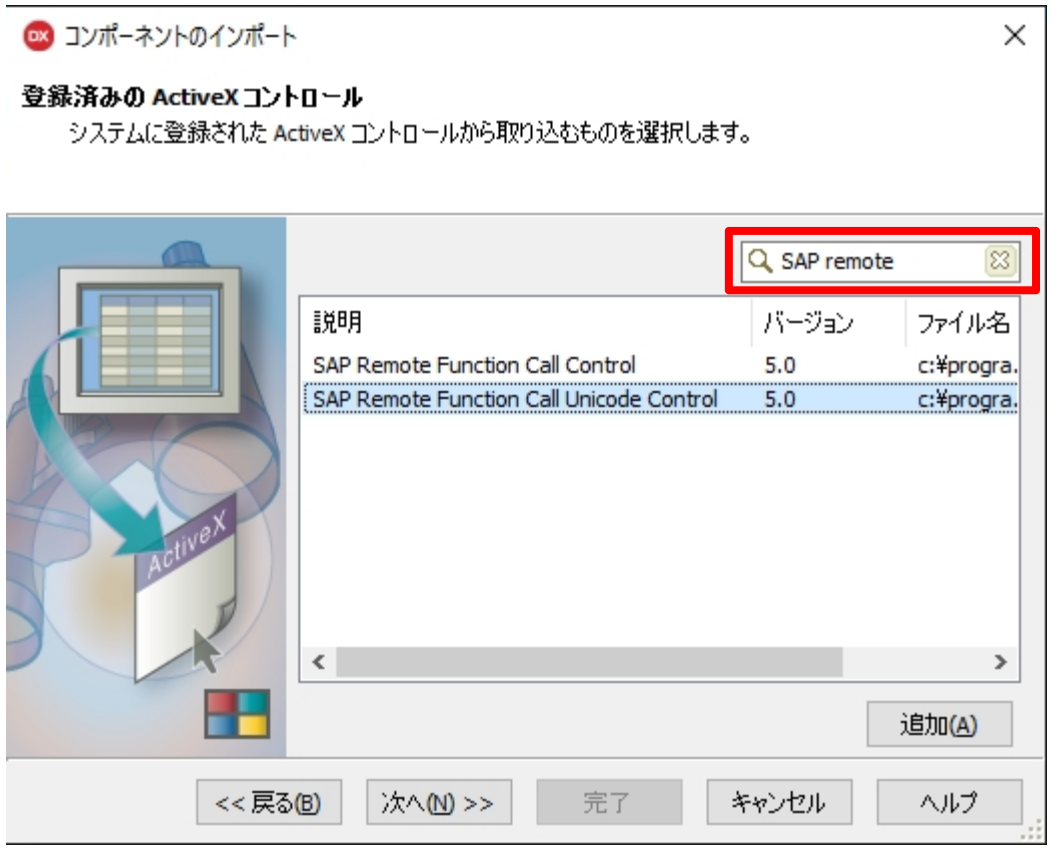
RAD Studio 設定

- ユニットをどのように作成するか指定します。
- 今回の説明では「ユニットの作成」を指定します。



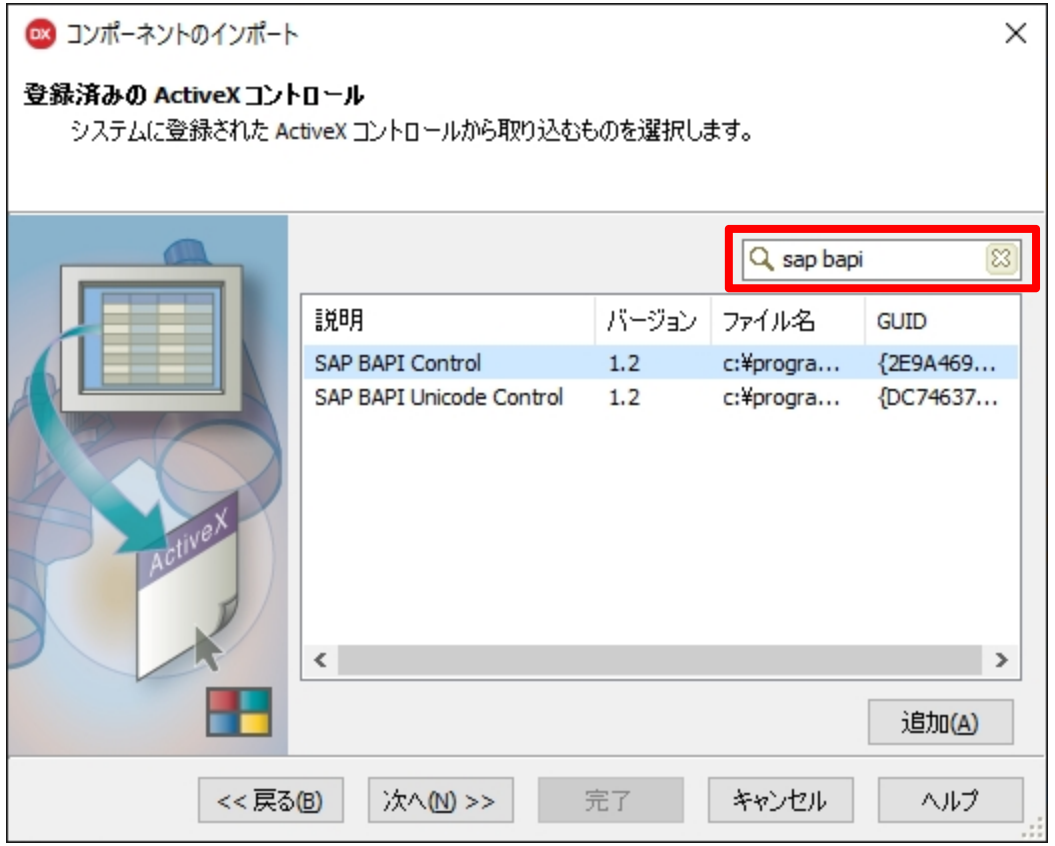
RAD Studio 設定

- コンポーネントの登録を再度実行します
- 検索に　sap remote　と入力します
- 以下に絞り込まれます
  - SAP Remote Function Call Control
  - SAP Remote Function Call Unicode Control
- ご使用の環境に合わせて、Unicode版、Non Unicode版を選択してください



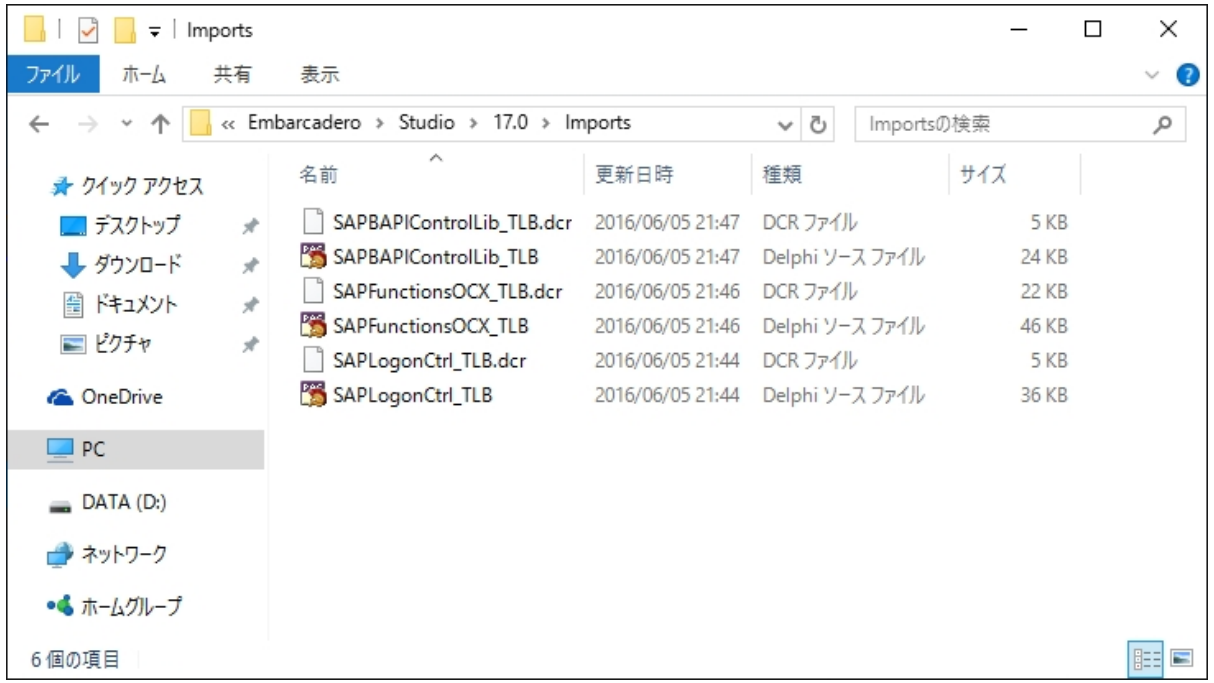
RAD Studio 設定

- コンポーネントの登録を再度実行します
- 検索に　sap bapi　と入力します
- 以下に絞り込まれます
  - SAP BAPI Control
  - SAP BAPI Unicode Control
- ご使用の環境に合わせて、Unicode版、Non Unicode版を選択してください



RAD Studio 設定

- 作成されたUnitは、プロジェクトに追加して使用します



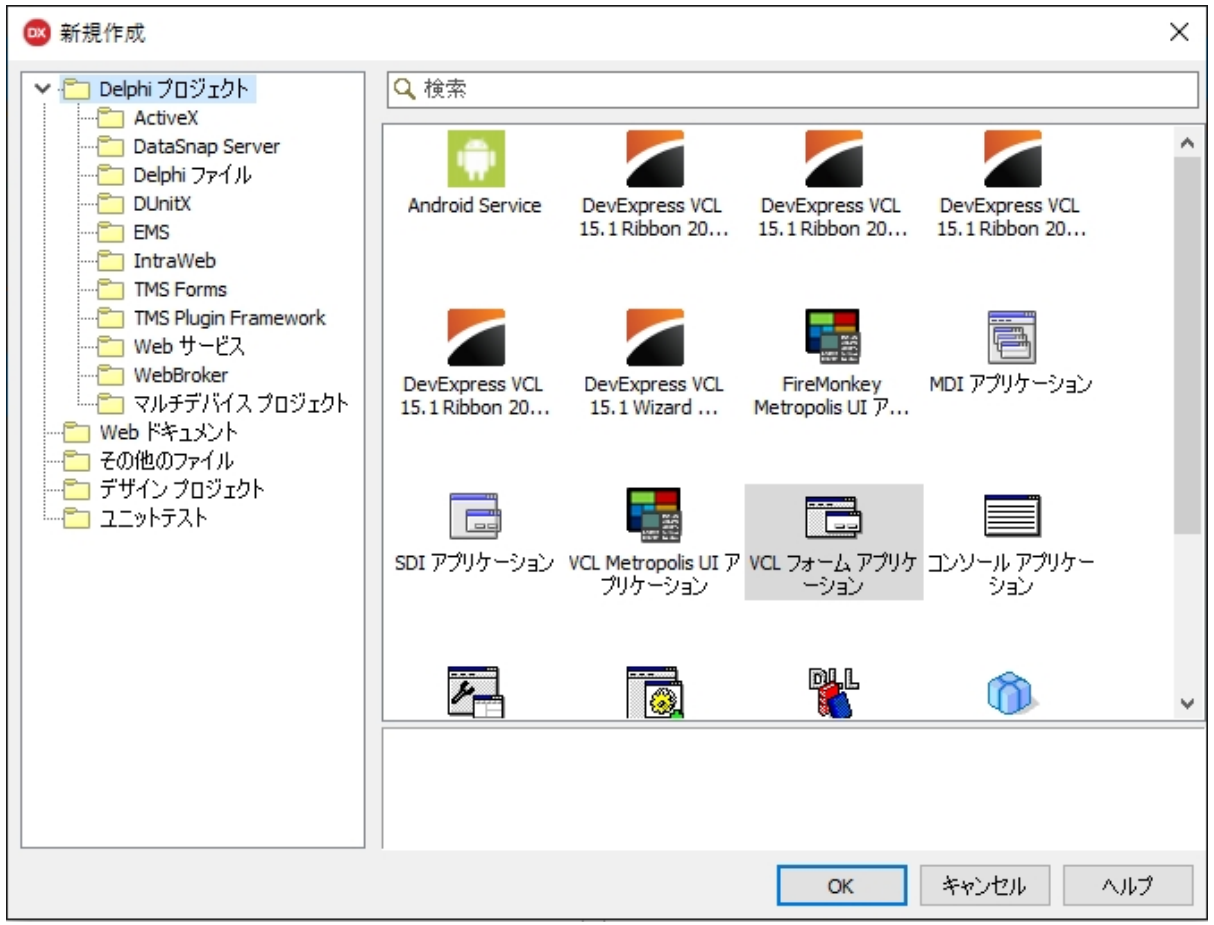
DEMO環境準備

- Application 設定

embarcadero DEVELOPER CAMP

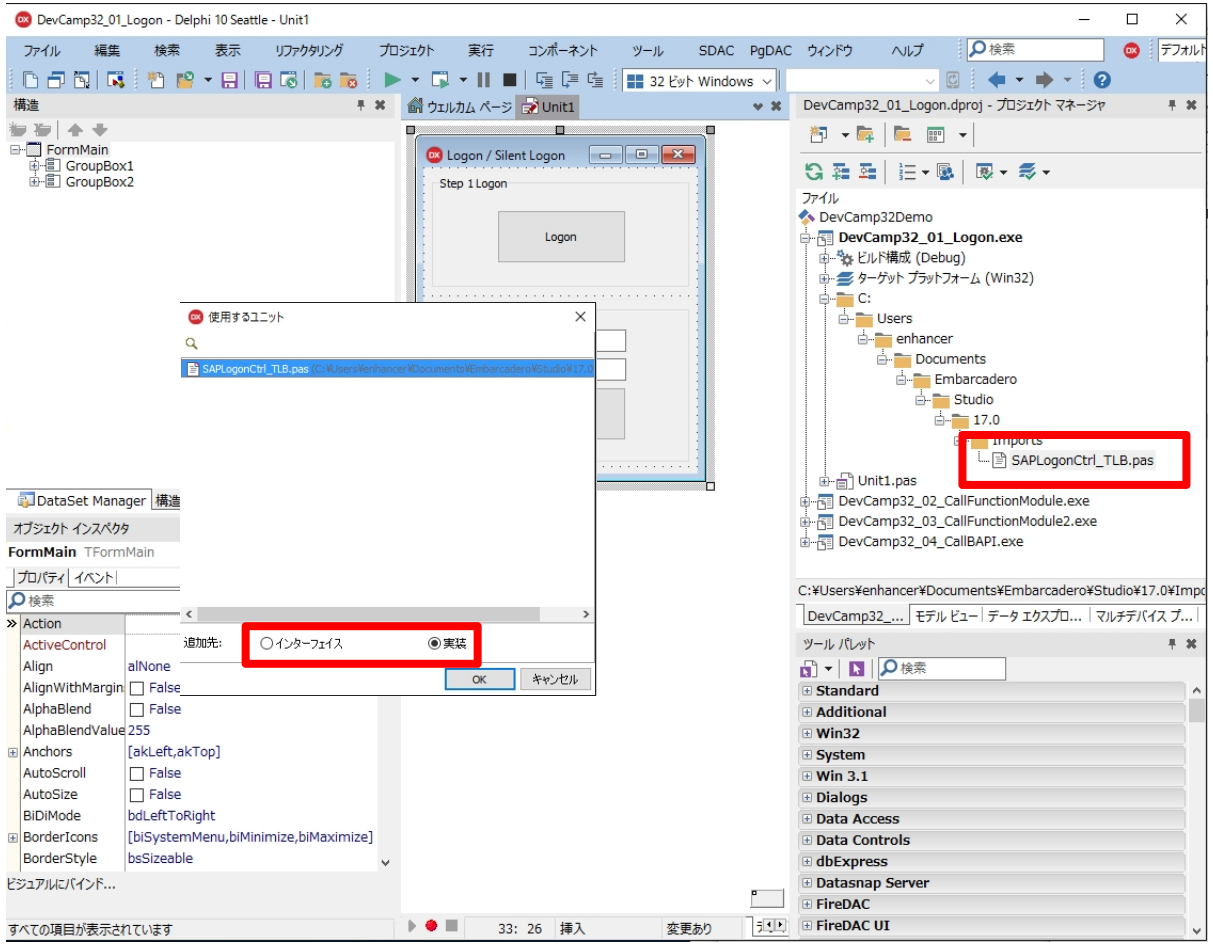
Application 設定

- 参考（使い方）
  - 新規のDelphiプロジェクト作成
  - VCLフォームアプリケーション



Application 設定

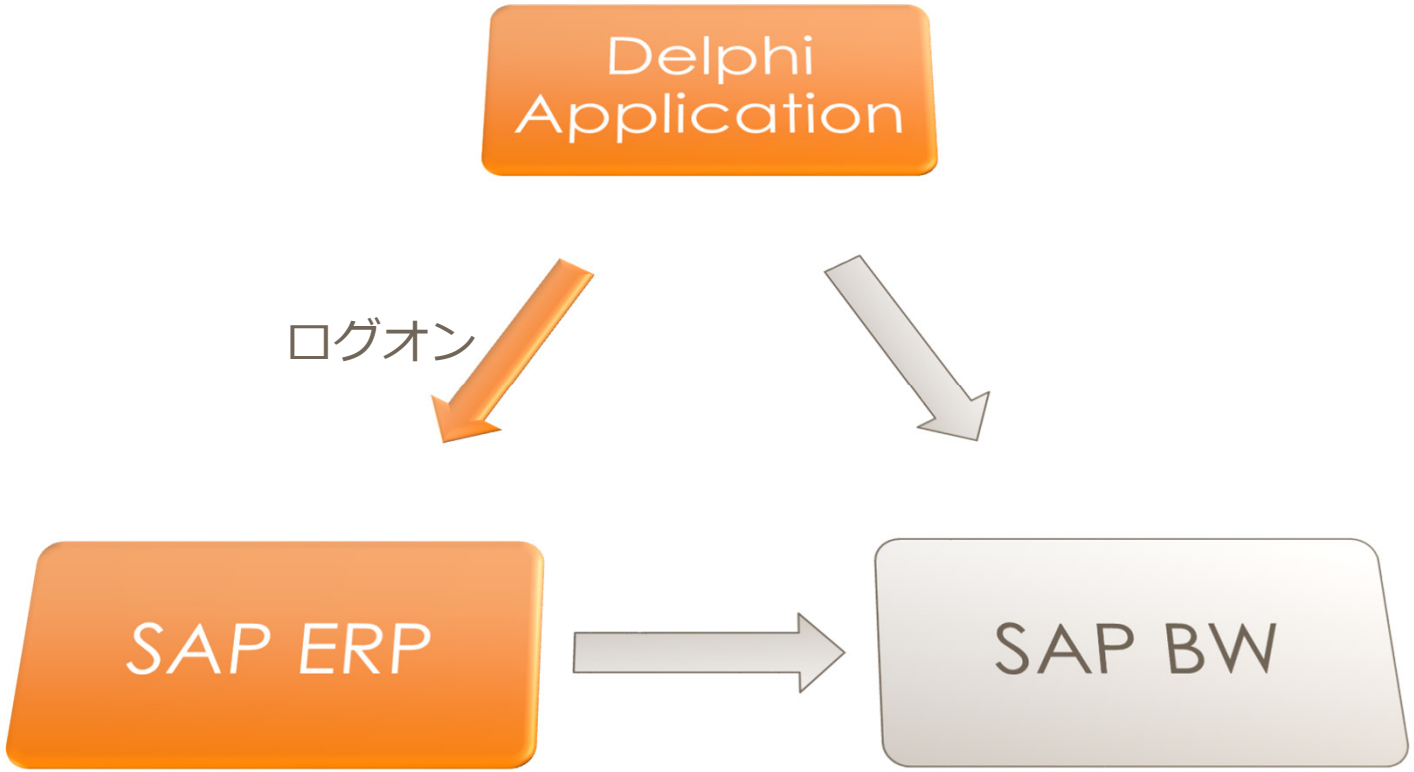
- 参考（使い方）
  - 作成されたUnitをプロジェクトに追加
  - 適当なコンポーネントを配置
  - 使用するUnitを追加



DEMO

- SAPシステムとの接続 1

DEMO　[SAPシステムとの接続 1]



DEMO　[SAPシステムとの接続 1]

TSAPLogonControl

- ERPやCRM、BW等にログオンするオブジェクト

Connection

- ERPやCRM、BW等との接続状況とパラメータを保持するオブジェクト
- 接続後はすべてのパラメータが読み取り専用変更されます

接続／切断

- 接続
  - Connection.Logon(0, false)
- 切断
  - Connection.LogOff;

DEMO　[SAPシステムとの接続 1]

```
var
  SAPLogOnCtrl: TSAPLogonControl;
  Connection: Variant;

// 接続の設定
Connection := SAPLogOnCtrl.NewConnection;

// ログオン
if Connection.Logon(0, false) = true then
begin … end;

// ログオフ
Connection.LogOff;
```

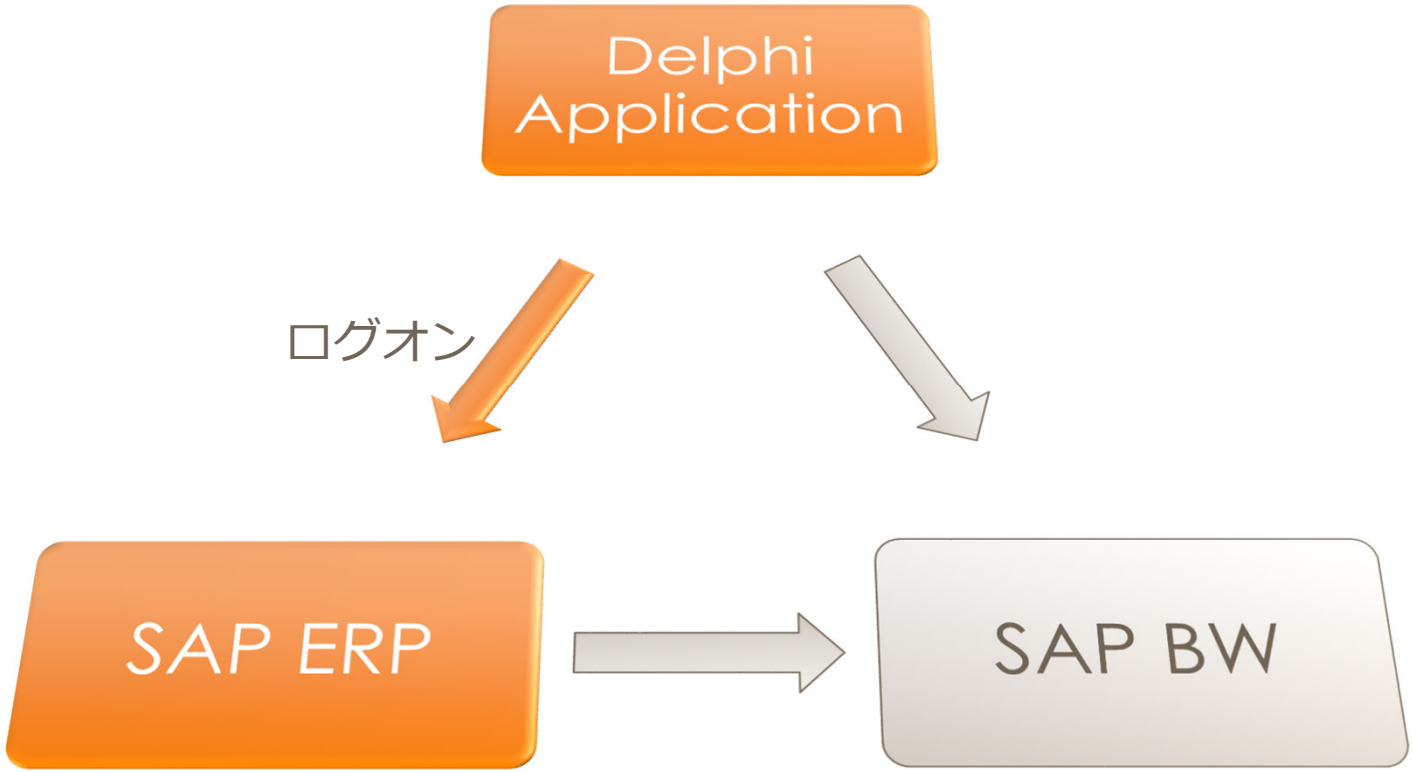
DEMO

DEMO

- SAPシステムとの接続 2

embarcadero DEVELOPER CAMP

DEMO　[SAPシステムとの接続 2]



DEMO　[SAPシステムとの接続 2]

Connection

- 接続先の情報を設定します
  - ApplicationName
  - System
  - ApplicationServer
  - SystemNumber
  - MessageServer
  - GroupName
  - TraceLevel
  - RFCWithDialog
  - Client
  - User
  - Password
  - Language
- 接続状態確認
  - IsConnected

IsConnected

- tloRfcNotConnected
  - 未接続の状態
- tloRfcConnected
  - 接続状態
- TloRfcConnectCancel
  - ログインダイアログでCancel
- tloRfcConnectParameterMissing
  - サイレントログオンのパラメータ誤り
- TloRfcConnectFailed
  - 接続失敗。LastErrorで追加情報を参照。

Connection.Logon(0, true)

- サイレントログオン、ダイアログ表示を切り替えます

DEMO　[SAPシステムとの接続 2]

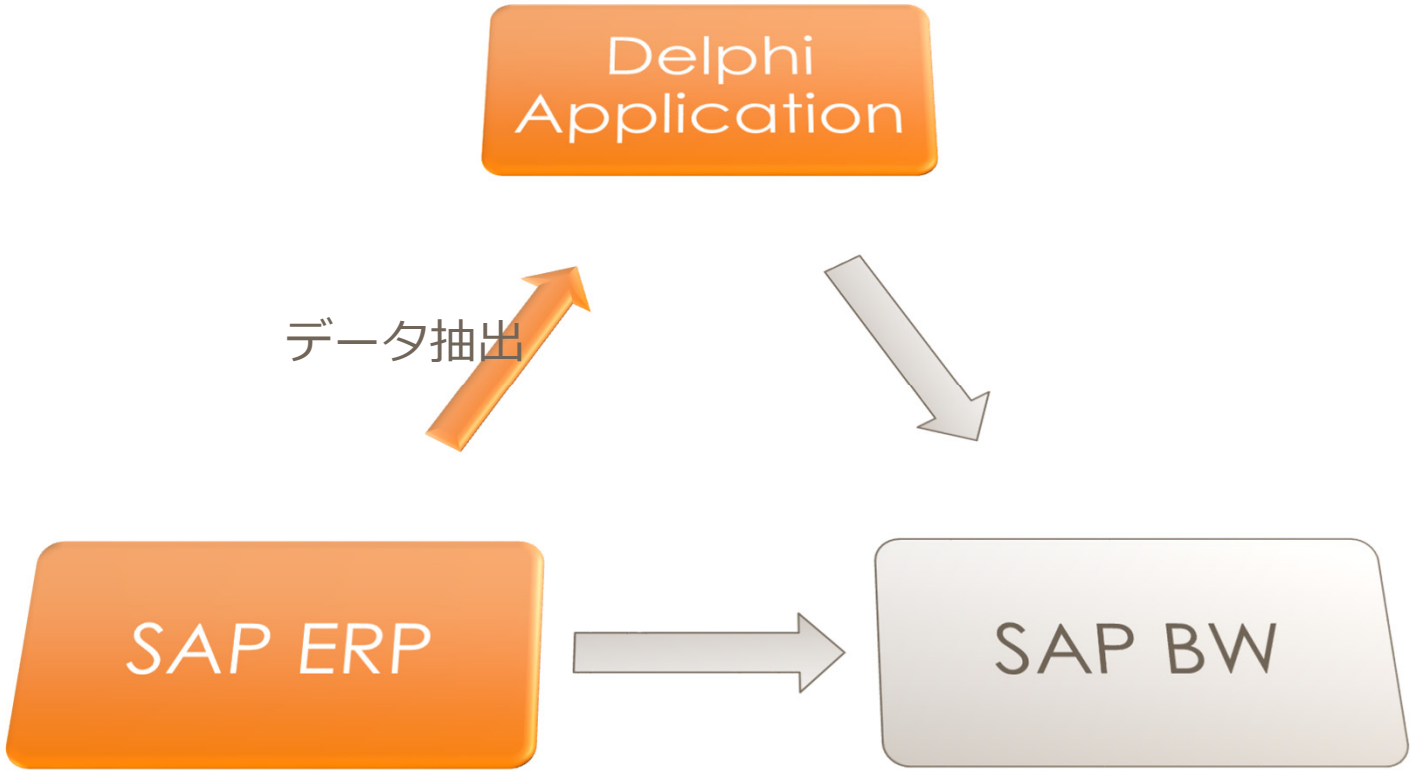
```
// 接続の設定
Connection := SAPLogOnCtrl.NewConnection;

// ログオン情報設定
Connection.ApplicationServer := 'APSERVER';
Connection.SystemNumber      := '00';
Connection.Client            := '100';
Connection.System            := 'SID';
Connection.User              := 'USERNAME';
Connection.Password          := 'PASSWORD';
Connection.Language          := 'EN';

// サイレントログオン
if Connection.Logon(0, true) = true then
begin … end;

// ログオフ
Connection.LogOff;
```

DEMO

DEMO

- データ抽出1

DEMO　[データ抽出 1]



DEMO　[データ抽出 1]

TSAPFunctions

- ERPやCRM、BW等のリモート実行可能汎用モジュールを実行します。
- T-code:SM37
  「属性」タブの処理タイプが「リモート可能モジュール」となっている汎用モジュールが使用可能

RFC_READ_TABLE

- 任意のテーブルのデータを取得する汎用モジュールです。
- 取得する項目、行数などを簡単に設定できます。
- 取得できるレコード長、型の制限などがあるため使用には注意が必要です。
  詳しくはNote:382318を参照。

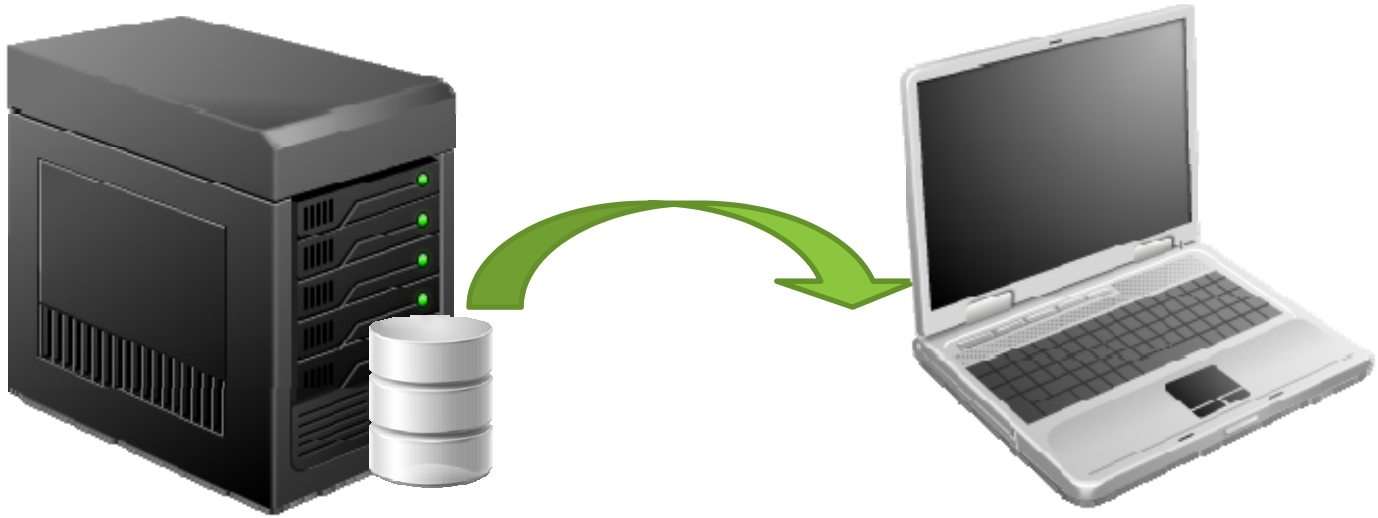
パラメータ設定

- Funct.exports('QUERY_TABLE').value := 'T001';
- ここではRFC_READ_TABLEの取得対象のテーブルを指定するパラメータを設定します

DEMO　[データ抽出 1]

```
// 使用する接続の設定
SAPFunct.Connection := Connection;

// 汎用モジュールを設定
Funct := SAPFunct.add('RFC_READ_TABLE');

// パラメータ設定
Funct.exports('QUERY_TABLE').value := ‘T001’;

// 汎用モジュール実行
Funct.call

// 結果取得
Table := Funct.tables.item(‘DATA’);
for Row := 1 to Table.RowCount do
begin
   … = Table.value(Row,1);
end;
```

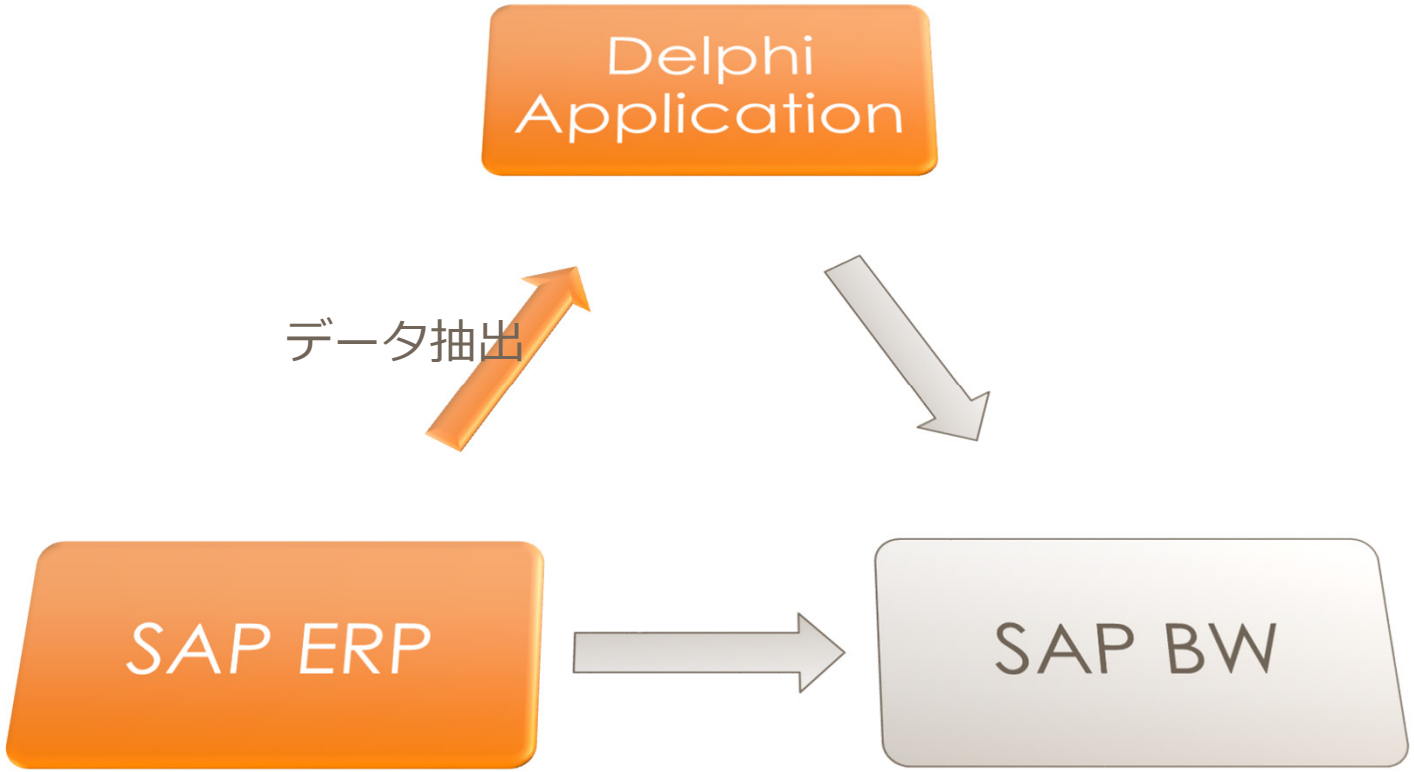
DEMO



DEMO

- データ抽出2

DEMO　[データ抽出 2]



DEMO　[データ抽出 2]

テーブル型パラメータの使用

- Options := Funct.tables.item('OPTIONS');
- バリアント型に代入

テーブル型への値設定

- Options.Rows.add;
- Options.value(Row, 'TEXT') := '値'

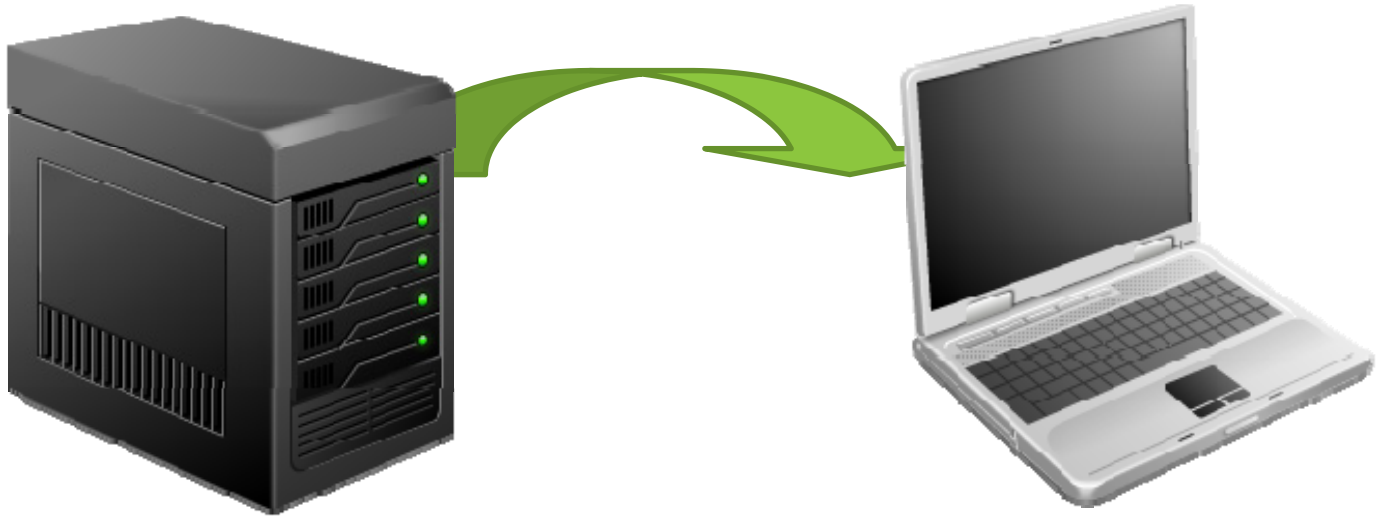
テーブル型からの値取得

- Table := Funct.tables.item('DATA');
- バリアント型に代入
- for Row := 1 to Table.RowCount do
  begin
  … = Table.value(Row,1);
  end;
- 行番号と列を指定してデータを抽出します。列は名称でも指定可能です。
- 行番号は「1」からです

DEMO　[データ抽出 2]

```
// 使用する接続の設定
SAPFunct.Connection := Connection;

// 汎用モジュールを設定
Funct := SAPFunct.add('RFC_READ_TABLE');

// パラメータ設定
Funct.exports('QUERY_TABLE').value := ‘T001’;

// テーブル型へのパラメータ設定
Options := Funct.tables.item('OPTIONS');
for Row := 1 to MemoOptions.Lines.Count do
begin
   Options.Rows.add;
   Options.value(Row, 'TEXT') := MemoOptions.Lines[Row - 1];
end;

// 汎用モジュール実行
Funct.call
```

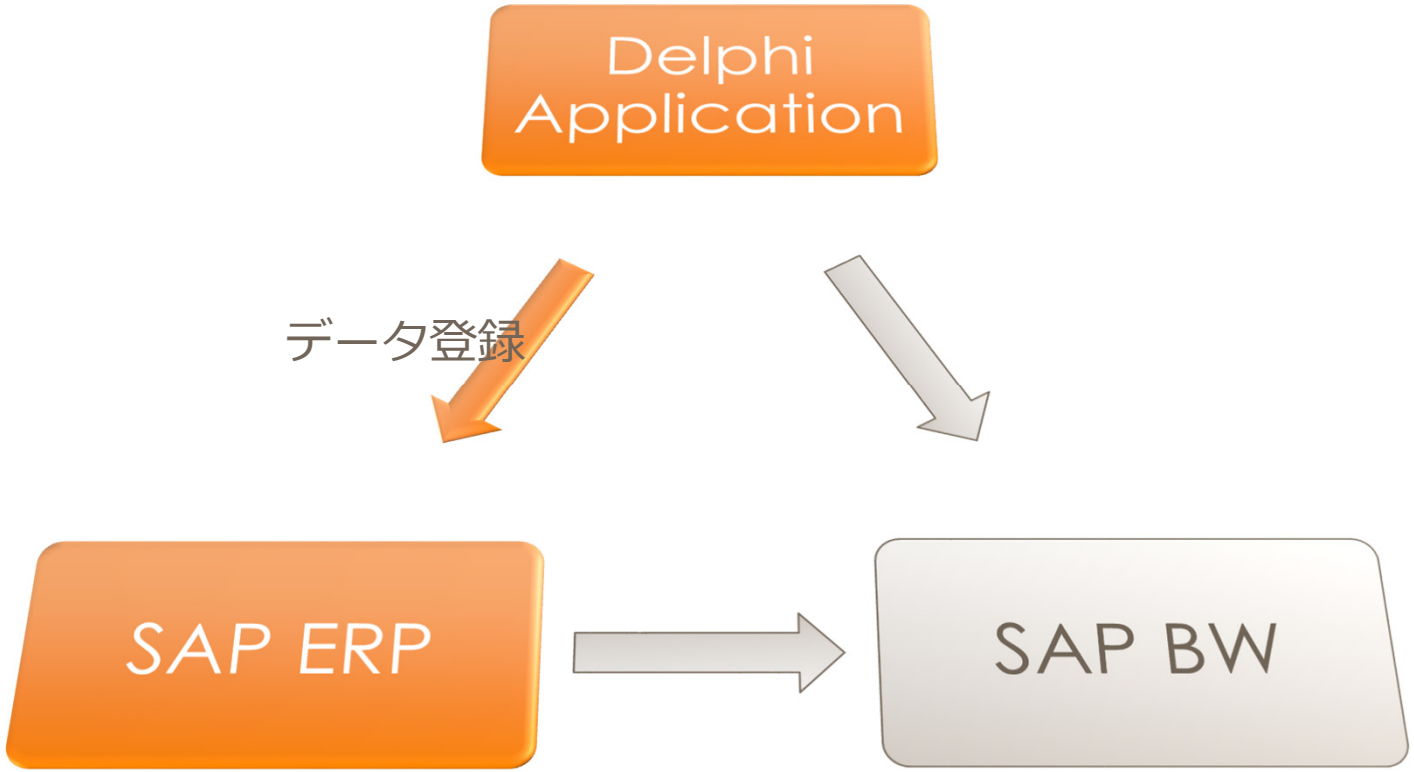
DEMO



DEMO

- データ登録

DEMO　[データ登録]



DEMO　[データ登録]

BAPI_MATERIAL_SAVEDATA

- 品目を登録する汎用モジュールです。

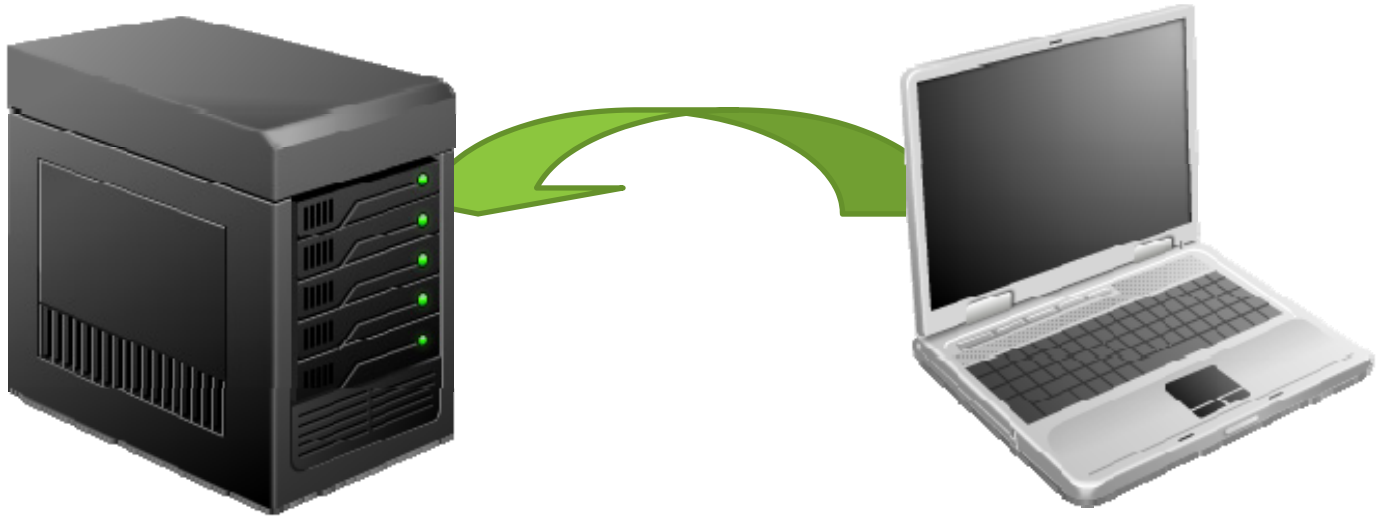
パラメータ設定

- 今回は以下4つのパラメータを使用します。
  - Header
  - ClientData
  - ClientDataX
  - MaterialDescription

DEMO　[データ登録]

```
Begin
  // 汎用モジュールを設定
  Funct := SAPFunct.add('BAPI_MATERIAL_SAVEDATA');
  // パラメータ設定
  Header := Funct.exports('HEADDATA');
  ClientData := Funct.exports('CLIENTDATA');
  ClientDataX := Funct.exports('CLIENTDATAX');
  MaterialDescription := Funct.tables.item('MATERIALDESCRIPTION');

  Header.Value['MATERIAL'] := ‘Material Number’
  ...
  ClientData.Value['MATL_GROUP'] := ‘Material Group’
  ClientDataX.Value['MATL_GROUP'] := 'X';    // Material Group

  MaterialDescription.Rows.add;
  MaterialDescription.value(1, 'LANGU') := 'EN';
  MaterialDescription.value(1, 'LANGU_ISO') := 'EN';
  MaterialDescription.value(1, 'MATL_DESC') := ‘Description’;
end;
```

DEMO

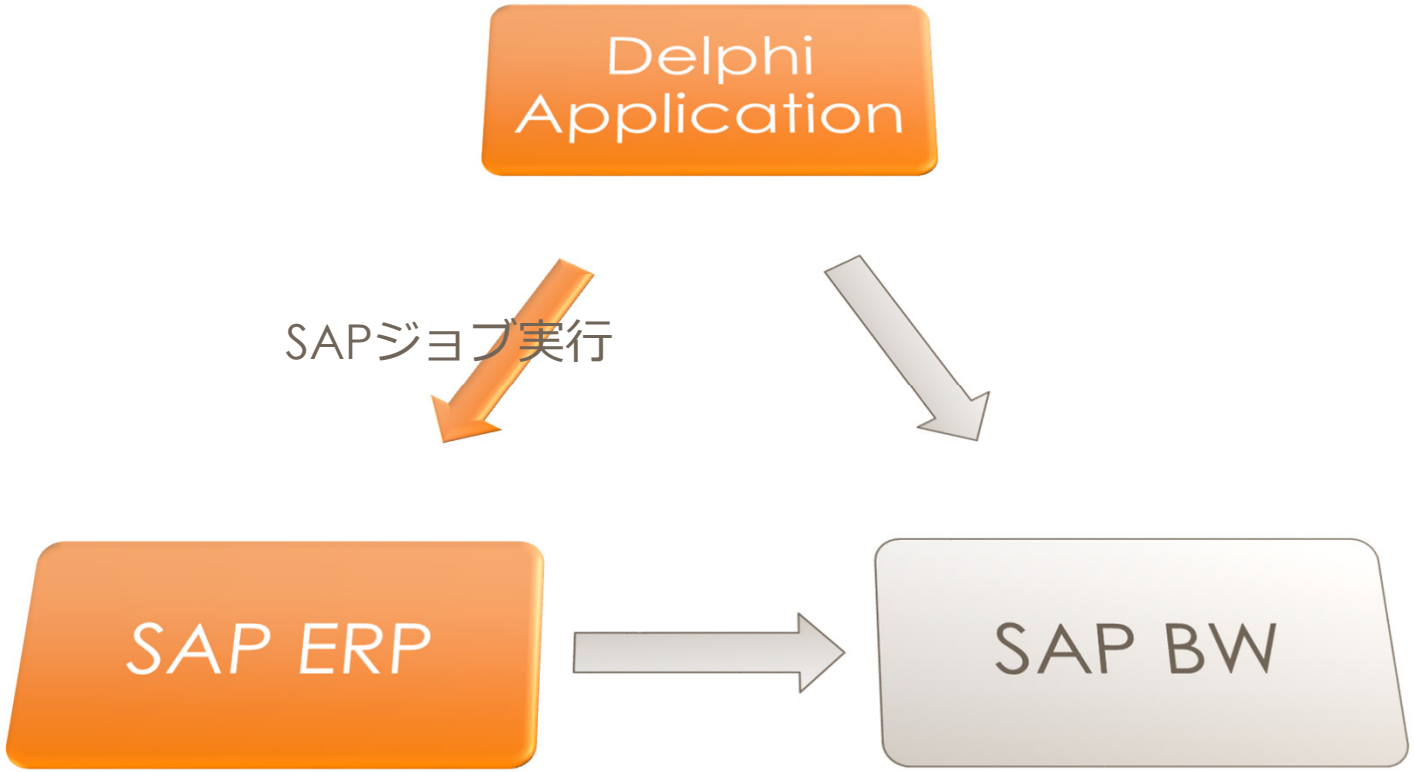


DEMO

- SAPジョブ実行

embarcadero DEVELOPER CAMP

DEMO　[SAPジョブ実行]



DEMO　[SAPジョブ実行]

登録済みのジョブを実行

- 事前にSM36でジョブを定義します。
- プログラムでジョブをコピーし、実行します。
- 新規にジョブを登録することもできます。

XBPにログオンして実行します

- 汎用モジュールを単純に事項するだけでは動きません。
- 以下エラーが発生したときは、ログオン処理を確認してください。
  Not logged on in interface XBP

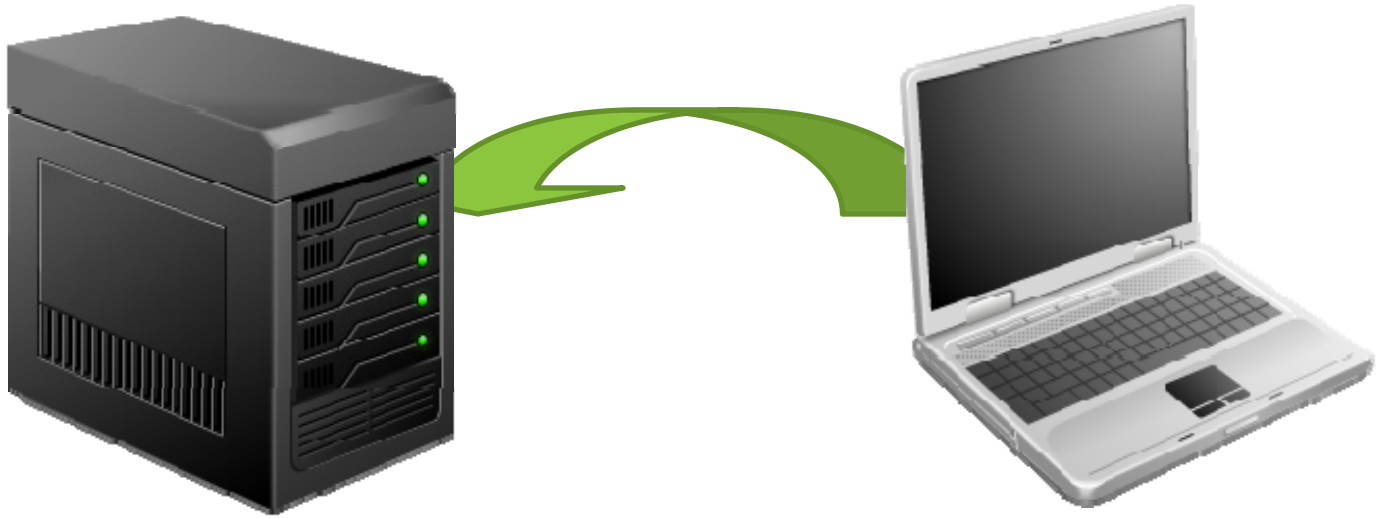
使用する汎用モジュール

- BAPI_XMI_LOGON
- BAPI_XBP_JOB_COPY
- BAPI_XBP_JOB_START_IMMEDIATELY
- BAPI_XMI_LOGOFF

DEMO　[SAPジョブ実行]

```
begin
  Funct := SAPFunct.add('BAPI_XMI_LOGON');
  Funct.exports('EXTCOMPANY').value := 'ENHANCER';
  Funct.exports('EXTPRODUCT').value := 'DEVCAMPDEMO';
  Funct.exports('INTERFACE').value := 'XBP';
  Funct.exports('VERSION').value := '1.0';
  Funct.call;

  Funct := SAPFunct.add('BAPI_XBP_JOB_COPY');
  …

  Funct := SAPFunct.add('BAPI_XBP_JOB_START_IMMEDIATELY');
  …

  Funct := SAPFunct.add('BAPI_XMI_LOGOFF');
  Funct.exports('INTERFACE').value := 'XBP';
  Funct.call;
end;
```

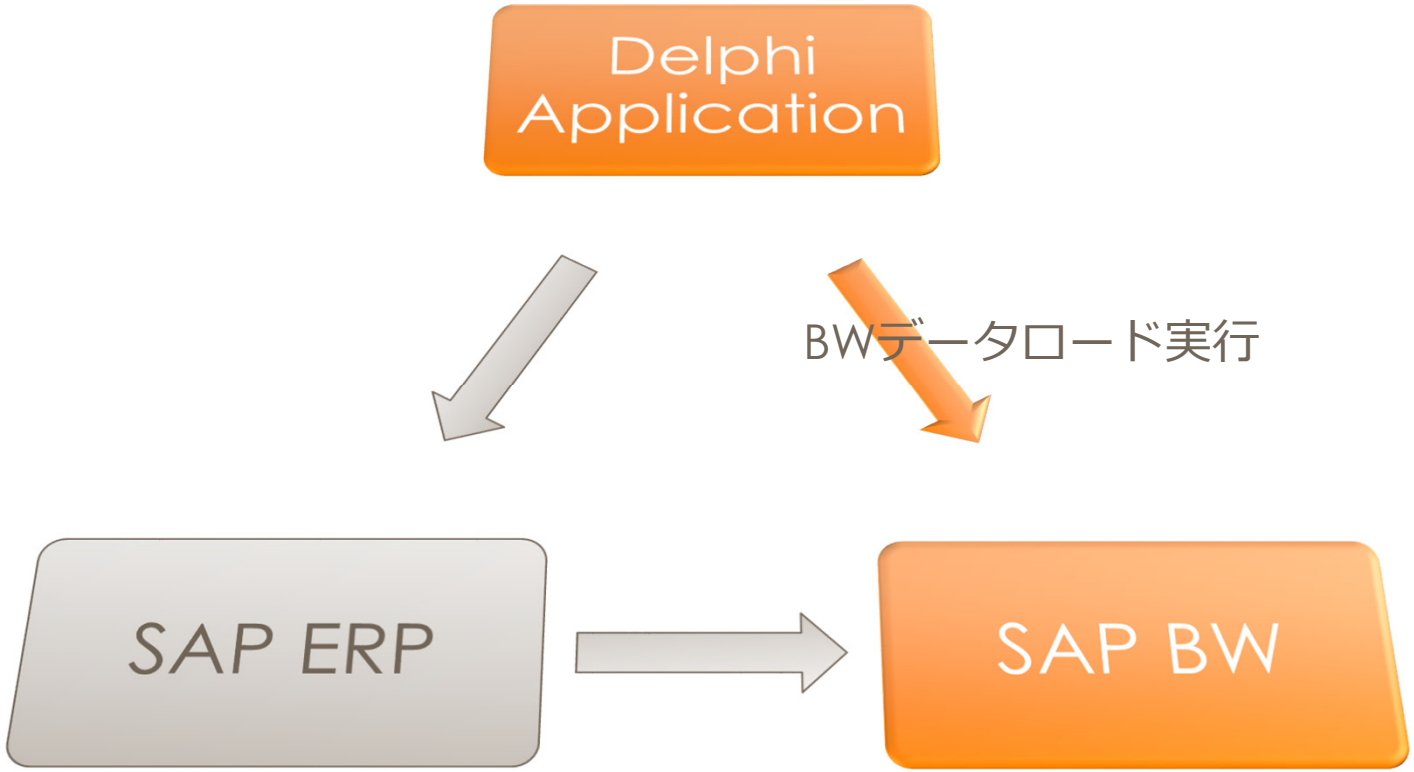
DEMO



DEMO

- BWデータロード実行

DEMO　[BW データロード実行]



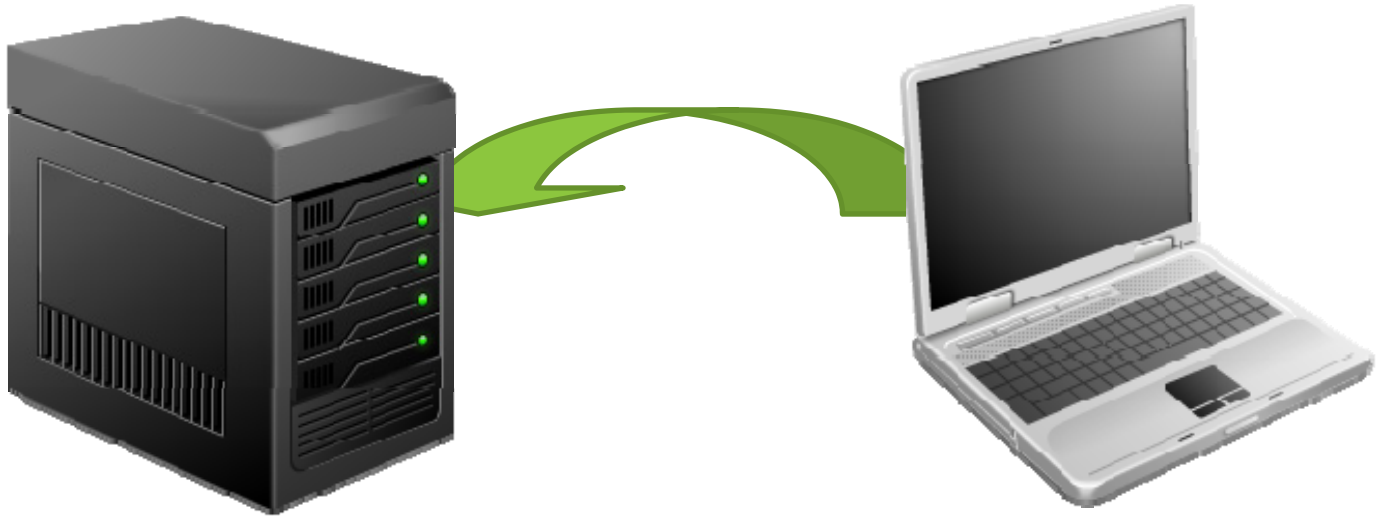
DEMO　[BW データロード実行]

登録済みのプロセスチェーンを実行	使用する汎用モジュール	
• 事前にRSPCでプロセスチェーンを定義します。 • バックグラウンドで実行されます。	• RSPC_API_CHAIN_START	

DEMO　[BW データロード実行]

```
begin

  …

  // BW Process Chain Start
  Funct := SAPFunct.add('RSPC_API_CHAIN_START');
  Funct.exports('I_CHAIN').value := 'YPC0002';
  Funct.exports('I_SYNCHRONOUS').value := 'X';
  if not Funct.call then
  begin
    ShowMessage(Funct.exception)
  end
end;
```

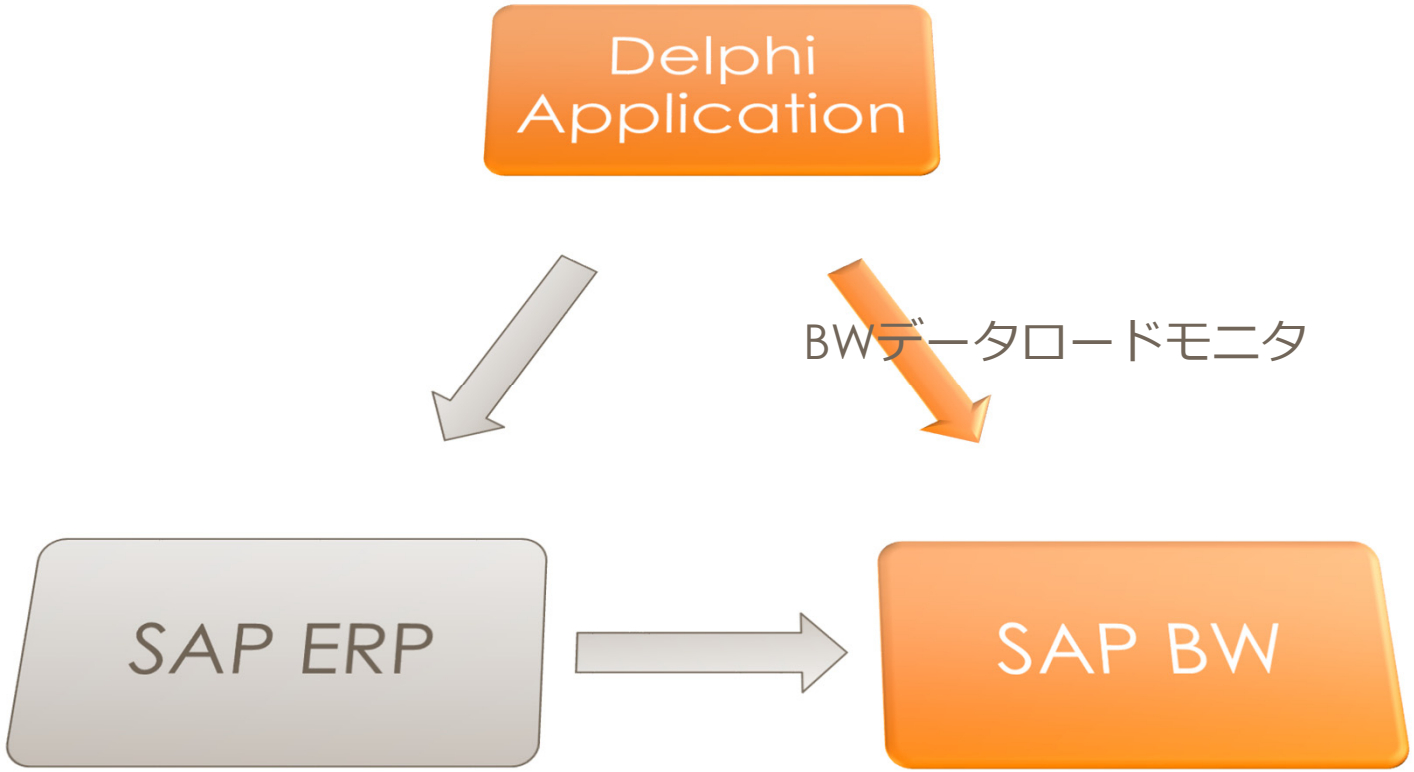
DEMO



DEMO

- BWデータロードモニタ

DEMO　[BW データーロードモニタ]



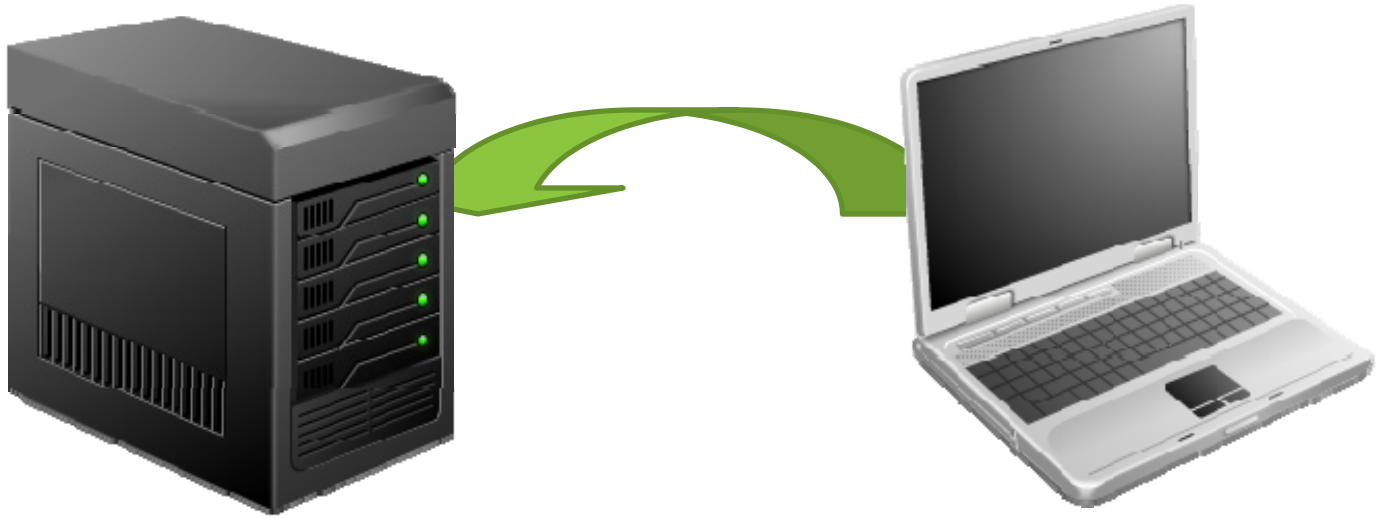
DEMO　[BW データロードモニタ]

登録済みのプロセスチェーンを実行	使用する汎用モジュール	
•事前にRSPCでプロセスチェーンを定義し、実行します。	•RSPC_API_CHAIN_GET_STATUS	

DEMO　[BW データロード モニタ]

```
begin
  // BW Process Chain Start
  Funct := SAPFunct.add('RSPC_API_CHAIN_GET_STATUS');
  Funct.exports('I_CHAIN').value := 'YPC0002';
  Funct.exports('I_LOGID').value := EditLogId.Text;
  if not Funct.call then
  begin
    ShowMessage(Funct.exception)
  end;
  EditResult.Text := Funct.imports('E_STATUS').value;
end;
```

DEMO



まとめ

- バックエンドシステムの状況
- システム連携の重要性
- Delphiで簡単にSAP連携が可能

弊社製品ご紹介

embarcadero DEVELOPER CAMP

弊社製品ご紹介

- もとは、自社業務の効率化目的で開発。
- 詳細設計書を自動生成します。
- 他分野ではよくあるツールですが。。。

- 導入事例

(事例紹介許可ありのみ)

- 株式会社日立システムズ様
- プロジェクト名：　株式会社日立製作所様　hi-fronts BW 国内
- プロジェクト期間：　2015/12
- 概要：　グループ会社向け BW開発、仕様書の可視化

- 株式会社日立システムズ様
- プロジェクト名：　株式会社日立製作所様　hi-fronts BW 海外
- プロジェクト期間：　2015/07 ～ 2015/11
- 概要：　グループ会社向け BW開発、仕様書の可視化

- 株式会社日立システムズ様
- プロジェクト名：　株式会社ドーム様SAPシステム運用保守支援
- プロジェクト期間：　2013/07 ～ 2013/09
- 概要：　SAPシステム運用保守支援。仕様書の可視化、改善・追加構築対応

- 株式会社日立システムズ様
- プロジェクト名：　旭硝子株式会社様DWHシステム構築
- プロジェクト期間：　2011/05 ～ 2012/03
- 概要：　バックエンド SAP BW、フロントエンドSAP BOによるBIシステム構築

自己紹介

会社

- 名前　株式会社エンハンサー
- URL　http://www.enhancer.co.jp
- 主な事業　SAP社パッケージ導入支援
  SAP社パッケージ連携システム開発
  アミューズメント業界向けパッケージ開発
  臨床検査関連システム開発

私

- 名前　藤田 和宏
- 役職　代表取締役
- Email　kazuhiro.fujita@enhancer.co.jp
- Facebook　https://www.facebook.com/kazuhiro.fujita.92

- 略歴
  1994年　パソコン通信にはまる
  Turbo Pascalに出会い初の業務アプリ開発
  1996年　Delphiに出会い、開発の容易さに驚愕
  1997年　C++Builderに出会い、さらに驚愕
  2002年　DWH導入支援
  2006年　自社パッケージリリース
  2010年　法人成り

THANKS!

www.embarcadero.com/jp

第32回 エンバカデロ・デベロッパーキャンプ