Getting Started for RIA

Magic xpa

本書に記載の内容は、将来予告なしに変更することがあります。これらの情報について MSE（Magic Software Enterprises Ltd.）および MSJ（Magic Software Japan K.K.）は、いかなる責任も負いません。

本書の内容につきましては、万全を期して作成していますが、万一誤りや不正確な記述があったとしても、MSE および MSJ はいかなる責任、債務も負いません。

MSE および MSJ は、この製品の商業価値や特定の用途に対する適合性の保証を含め、この製品に関する明示的、あるいは黙示的な保証は一切していません。

本書に記載のソフトウェアは、製品の使用許諾契約書に記載の条件に同意をされたライセンス所有者に対してのみ供給されるものです。同ライセンスの許可する条件のもとでのみ、使用または複製することが許されます。当該ライセンスが特に許可している場合を除いては、いかなる媒体へも複製することはできません。

ライセンス所有者自身の個人使用目的で行う場合を除き、MSE または MSJ の書面による事前の許可なしでは、いかなる条件下でも、本書のいかなる部分も、電子的、機械的、撮影、録音、その他のいかなる手段によっても、コピー、検索システムへの記憶、電送を行うことはできません。

サードパーティ各社商標の引用は、MSE および MSJ の製品に対する互換性に関しての情報提供のみを目的としてなされるものです。

本書において、説明のためにサンプルとして引用されている会社名、製品名、住所、人物は、特に断り書きのないかぎり、すべて架空のものであり、実在のものについて言及するものではありません。

一般に、会社名、製品名は各社の商標または登録商標です。

MSE および MSJ は、本製品の使用またはその使用によってもたらされる結果に関する保証や告知は一切していません。この製品のもたらす結果およびパフォーマンスに関する危険性は、すべてユーザが責任を負うものとします。

この製品を使用した結果、または使用不可能な結果生じた間接的、偶発的、副次的な損害（営利損失、業務中断、業務情報の損失などの損害も含む）に関し、事前に損害の可能性が勧告されていた場合であっても、MSE および MSJ、その管理者、役員、従業員、代理人は、いかなる場合にも一切責任を負いません。

第 1 版 2013 年 10 月 30 日

Copyright 2013 Magic Software Enterprises Ltd.and Magic Software Japan K.K.　All rights reserved.

目次

第 1 章： Magic xpa RIA 入門コースについて

入門コースで取扱う内容 1
入門コースの構成 1
入門コースの前提条件 1
本コーステキスト表記上の約束 3

第 2 章： Magic xpa Enterprise Studio について

はじめに 5
新しいプロジェクトの作成 5
新規プロジェクトの詳細 6
［ナビゲータ］ペイン 6
特性シート 8
［チェック結果］ペイン 8
［コメント］ペイン 9
ステータス行 10
要約 10

第 3 章： プログラムの作成

はじめに 12
Magic xpa RIA 入門プロジェクトの開始 12
プログラムの新規作成 15
タスクのインタフェース 20
［データビュー］エディタ 21
タスクのフォーム 22
フォームへの変数項目の配置 27
プログラムの保存 28
プログラムの実行 29
ビルトインタスクの動作 29
Magic xpa の型 30
Magic xpa の書式 31
変数項目の追加 33
［フォーム自動作成］ウィザード 34
ウィンドウタイプの指定 36
要約 37

第 4 章：データ操作

はじめに 39
数値データの操作 39
プログラムの実行 41
計算の条件的実行 41
IF 関数の利用（ゴールド会員の例） 43
IF 関数を使用したプログラムの実行 44
文字列データの取り扱い 45
文字列関数を使用したプログラムの実行 48
練習問題 48
要約 49

第 5 章：データチェック

はじめに 51
メニューから Magic xpa 内部に組み込まれたデータ妥当性検証機能 51
データ妥当性チェック機能 52
妥当性検証ロジックの開発 55

データの整合性 59
データ整合性を維持するための方法について－要約 63
パーク条件 64
練習問題 66
要約 66

第 6 章：データ項目初期値の設定

はじめに 68
［タスク前］での項目更新 68
項目の代入を使用する 72
［コントロール前］での項目更新 74
要約 78

第 7 章：フォームの設計

はじめに 80
基本色定義ファイルへの色の追加 80
コントロール色の変更 82
フォント定義ファイルへのフォント追加 83
コントロールのフォント変更 85
コントロールの移動 87
背景 89
練習問題 93
要約 94

第 8 章：データソースの内容参照

［データベース］テーブル 96
データソース定義 99
＜顧客＞プログラムの生成 102
＜リッチクライアント－顧客＞プログラムの実行 106
顧客データプログラムの手動作成 110
ここまでの要約 114
複数レコードの表示 114
＜照会－顧客ラインモード＞プログラムの実行 116
＜顧客－ラインモード＞プログラムの手作業による作成 120
顧客ラインモードプログラムの実行 126
データの検索 (位置付) 127
データのフィルタリング（範囲） 128
フォーム上のコントロールの整理 130
Tab 順序 131
練習問題 133
要約 137

第 9 章：アプリケーションエンジンのコンセプト

はじめに 139
イベントドリブンの考え方 139
タスク 139
実行ルール 143
要約 145

第 10 章：イベント

はじめに 147
イベントの作成（実習） 149
イベントの発動 150
結果の確認 154
要約 154

第 11 章：ハンドラ

はじめに 156
同一イベントに対応する複数のハンドラの処理 157
＜ゴールド会員＞イベントのハンドラ 157
＜日付と時刻の設定＞イベントのハンドラ 158
動作結果の確認 159
Magic エンジンによるイベントの監視 160
要約 165

第 12 章：［ブロック］処理コマンド

はじめに 167
［ブロック］処理コマンドの構造 167
［ブロック If］処理コマンドの実習 168
［ブロック Else］処理 コマンドの実習 172
［ブロック While］処理コマンド 177
練習問題 177
要約 178

第 13 章：リポジトリの操作

はじめに 180
クイックアクセス 180
リポジトリのエントリ登録操作 183
移動登録 184
上書登録 185
マルチマーキング 186
練習問題 187
要約 188

第 14 章：クロスリファレンス

はじめに 190
クロスリファレンスの使用例 191
クロスリファレンス結果の保存 193
練習問題 194
要約 194

第 15 章：１対１のデータリレーション

はじめに 196
'１対１'と'１対多'のデータリレーション比較 196
他のデータソースへのリンク 196
受注管理プログラムの実習 197
これまでの要約 210
［リンク式の再計算］のメカニズム 210
リンクの戻り値 211
練習問題 214
要約 219

第 16 章：一覧からのデータ選択

はじめに 222
選択テーブルプログラム 222
［選択テーブルプログラム］の呼び出し 231
データコントロールと選択テーブルプログラムの比較 235
練習問題 240
要約 244

第 17 章：１対多のデータリレーション

はじめに 246

１対多のデータリレーションの前提 246
商品選択プログラムの作成 247
受注明細行管理タスクの作成 254
練習問題 282
要約 289

第 18 章：非インタラクティブタスクのアプリケーションエンジン

はじめに 291
インタラクティブタスクと非インタラクティブタスクの特徴比較 292
非インタラクティブタスクのエンジン処理フロー 292
非インタラクティブタスクの動作 293
非インタラクティブタスクによる削除 294
削除を行なう非インタラクティブタスクの例 294
要約 298

第 19 章：帳票印刷

はじめに 301
帳票出力の環境設定 301
帳票の例　---　APG を使用して作成する場合 302
手作業による帳票出力プログラムの作成 305
帳票デザインの例 313
練習問題 327
要約 335

第 20 章：複雑な帳票

はじめに 337
複雑な帳票の概念 337
＜受注書印刷＞プログラムの作成 337
受注書表示タスクの作成 360
練習問題 363
要約 368

第 21 章：オブジェクト定義の一元管理

はじめに 370
モデル定義の効果：データの一貫性と妥当性 370
＜顧客コード＞モデルの定義 371
オブジェクトへのモデル割当て 376
モデル特性の継承 377
フォームモデルを使ったプログラムの自動作成 379
練習問題 380
要約 384

第 22 章：メニュー

はじめに 386
［メニュー］リポジトリ 386
デフォルトのプルダウンメニュー 387
デフォルトプルダウンメニューへのエントリ追加 388
コンテキストメニュー 389
メニュー特性 396
練習問題 398
要約 399

第 23 章：リッチクライアントに関する考慮事項

はじめに 403
RIA の運用環境について 403
遅延トランザクションについて 404
処理の最適化 408

要約 410

第 24 章：アプリケーションの公開

はじめに 412
Web サーバの設定 412
Magic xpa の動作環境 413
アプリケーションの公開準備 414
クライアント側でのインストール処理 417
要約 420

第 1 章 Magic xpa RIA 入門コースについて

この入門コースは、Magic xpa をはじめて使用する方を対象にまとめられています。通常のトレーニングセミナーに参加できない場合でも、この入門コース内容を一通り学ぶことで、基本的なリッチインターネットアプリケーション（RIA）の作成ができるようになります。

この入門コースでは、Magic xpa の基本的なコンセプト、および Magic xpa を利用して RIA プログラムを作成するためのさまざまなツール機能について学習します。このコースを一通り学ぶことで、Magic xpa で RIA を開発する基本的なスキルを身につけることができます。

入門コースで取扱う内容

本コースでは、次のことを学びます。

- Magic xpa の基本的事項
- Magic xpa Enterprise Studio のインタフェース
- ウィザードとユーティリティの使い方
- 製品コンセプト
- Magic エンジンの標準的な動作ルール
- 次のような特徴を持つ基本的アプリケーションの作成
  - RIA インタフェース
  - SQL データベースを利用する
  - 一対一、または一対多のリレーションを持つデータを取扱う
  - レポート（帳票印刷）出力を行なう

入門コースの構成

本コースの主目的は、学習者が基本的アプリケーションを開発できるようになることです。コースを一通り終えると、課題となるアプリケーションが一つ完成します。この実習課題となるアプリケーションの作成過程で、Magic xpa の動作ルールと、プログラミング方法の理解を深めることができます。

本コースは全部で 24 の章があります。これらの章は基本的な内容のものから次第に高度な内容となるように順序づけられています。したがって、各章についてはスキップせずに、最初から順を追って学習されることをおすすめします。

本コースは実際に作業しながら学習を進めてください。各章には実践的例題と練習問題があり、それらを実際にやってみることで課題アプリケーションが完成してゆきます。問題の指示に従って注意深くプログラミングすることが大切です。本コースではスクリーンショットを豊富に用意していますので、参考にしてください。

入門コースの前提条件

本コースは、RIA を前提としたプログラミングの自習書ですが、プログラミングの基本的なスタイルは、クライアント／サーバの場合と同じです。このため、Magic xpa でクライアント／サーバ用プログラムの経験者であれば、前半の説明はほぼ同じと感じると思います。まず、Magic xpa による RIA のプログラミングを行っていただきます。その後で、RIA としての注意点を説明していきます。

本コースを学ぶ方は、次のようなことに関する基本的知識を持っているか、あるいは慣れ親しんでいることを想定していますが、必ずしも必須というわけではありません。

PC 関連知識

1. 日本語入力／表示が基本的にできる
2. Windows のエクスプローラ、マイコンピュータなどを利用して、
   a. ファイルやフォルダを検索できる
   b. ファイルのコピー、移動、削除ができる
3. テキストファイルや HTML ファイルのブラウジングができる
4. プリンタ／印刷ダイアログボックスが使える
5. ［名前をつけて保存］ダイアログボックスが使える

6. インストールプログラムを利用してプログラムをインストールしたりアンインストールできる
7. PC 関連の基本的知識を持っている
8. Windows 7 の基本的知識を持っている
9. IIS（Internet Information Service）の動作環境を設定できる

開発に関する知識

1. フローチャートの意味を理解できる
2. フローチャートを使用して論理的な問題を解くことができる
3. システム解析の用語を知っている
4. プロシージャ（手続き）について知っている
5. 手続き型と非手続き型処理の違いが分かる
6. 関数について知っている
7. イベントについて知っている
8. ループの意味を知っている
9. IF-THEN-ELSE の構文を知っている
10. フォームについて知っている
11. I/O について知っている
12. 変数について知っている
13. レポート（帳票）について知っている
14. 以下の少なくとも一つについて使用経験がある。

C, C++	Fortran	Pascal
ADA	VisualBasic	Java
COBOL	Access	RPG
Prolog	Gupta	Python
Power Builder	Perl	Delphi

データベース知識

1. 次のデータベース用語を知っている
   - データベース
   - テーブル
   - 列／レコード
   - フィールド
   - インデックス
   - セグメント
2. 上記オブジェクトの関係について理解している
3. 次の意味が分かる
   - テーブルの登録
   - テーブルの修正
   - テーブルの削除
   - テーブルへのレコード登録
   - テーブルのレコード修正
   - テーブルからのレコード削除
4. 次の用語の意味が分かる
   - ソート
   - 位置付け
   - 範囲

PC 動作環境のセットアップ

本コースを始める前に、お使いの PC に Magic xpa が動作する環境を作成しておく必要があります。必要に応じて以下のセットアップを行ってください。

- Windows で IIS（Internet Information Service）の有効化
  Magic xpa をインストールする前に、以下のオプションを有効化する必要があります。
  - Web 管理ツール
    - IIS 管理コンソール
    - IIS6 管理互換（OS が 64 ビット版の場合に必要です）
  - World Wide Web サービス
    - HTTP 共通機能
      - 規定のドキュメント
      - 静的コンテンツ
    - アプリケーション開発機能
      - .NET 拡張機能
      - ASP .NET
      - CGI
      - ISAPI フィルタ
      - ISAPI 拡張機能
  - セキュリティ
    - 要求のフィルタリング
- Magic xpa Enterprise Studio（または体験版）
- MS SQL Server（Express）
- .Net Frame Work Ver3.5SP1（Windows のバージョンによっては、デフォルトで有効になっている場合があります）

本コーステキスト表記上の約束

本コーステキストでは、特定文字列の表現において、下表のように取り決めています。

表現	意味
＜文字列＞	本コースの説明にしたがって学習者がキーボードから入力する文字列です。必ずしも本テキストと同じでなくともよいです。
＜文字列＞	学習者がすでに入力した文字列で、上記の内容に対応するものであることを意味します。実際に入力された値が本コーステキストと異なる場合は、読み換えてください。
「文字列」	学習者の入力した文字列を用いて、システムが別の箇所で引用表示する文字列であることを示します。実際に入力された文字列が本コーステキストと異なる場合、読み換えてください。
［文字列］	Magic xpa の操作画面上に表示される文字列で、システムにより既定のものであることを意味しています。
文字列／文字列	メニューやヘルプなどのように、階層構造を持つものは、／で上位と下位を区切って表示します。

第 2 章 Magic xpa Enterprise Studio について

Magic xpa は Windows XP, Vista, 7 などに対応したグラフィカルインタフェースを持っています。このインタフェースを Magic xpa Enterprise Studio と呼びますが、主に［ナビゲータ］ペイン、ワークスペース、ステータス行の 3 つで構成されます。

本章では、Magic xpa Enterprise Studio で表示される各項目の操作方法について学習します。

キーワード

- プロジェクト
- ［ナビゲータ］ペイン
- 特性シート
- ［コメント］ペイン
- チェッカー
- ステータス行

学習目標

本章を学ぶことで、次の内容に対する理解を深めましょう。

- 新しいプロジェクトの作り方
- Magic xpa Enterprise Studio のインタフェース
- リポジトリ間をナビゲートする方法
- オブジェクトのコメントを参照する方法
- チェッカーの結果を参照する方法
- ステータス行の表示内容の意味

参照

Magic ヘルプの［Magic xpa リファレンス／ナビゲータとワークスペース］

同様に、［Magic xpa リファレンス／プロジェクトとアプリケーション］

本書では、Magic xpa 2.3 Enterprise Studio の RIA Edition を使用して開発しています。バージョン等で表示画面が変更される場合があります。

１．はじめに

Magic xpa Enterprise Studio は主に次の要素によって構成されています。

- ［ナビゲータ］ペイン
- ワークスペース
- ステータス行

［ナビゲータ］ペインでは、［データ］リポジトリや［プログラム］リポジトリなど、Magic xpa 内部の各リポジトリを選択します。［ナビゲータ］ペインで選択されたものが、ワークスペースに内容表示されます。

Magic xpa Enterprise Studio で重要なものに特性シートがあります。ほとんどの Magic xpa オブジェクトは、動的な特性を持っており、特性シート上で設定することができます。特性シートはフローティングウィンドウ、固定ウィンドウ、または［ナビゲータ］ペインのタブとしてのいずれかの方法で表示させます。

プロジェクトオブジェクトのコメントは［コメント］ペインで、チェッカーの結果はチェッカーペインでそれぞれ表示させます。

Magic xpa Enterprise Studio の最下行はステータス行で、ここには編集の状態や、ユーザ名などの情報が表示されます。

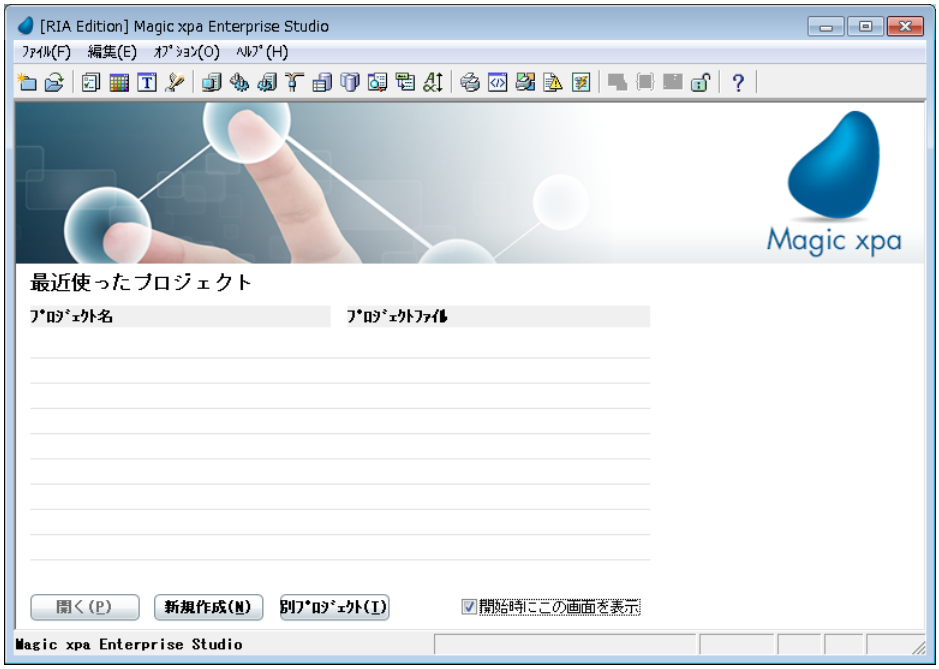
２．新しいプロジェクトの作成

Magic xpa でプログラムを作成するには、最初にまずプロジェクトを登録します。プロジェクトはユニットの集まりで、各ユニットには特定の設定に関連した情報がすべて含まれています。このなかには、メインコード、関連するコンポーネント、他のプロジェクトへの参照情報などが含まれます。メインコードは基本的に、モデル、データソース、プログラム、およびメニューによって構成されます。

本章では、Magic xpa RIA プロジェクトをまず作成します。モデル、データソース、プログラム、メニューについては別の章で説明します。

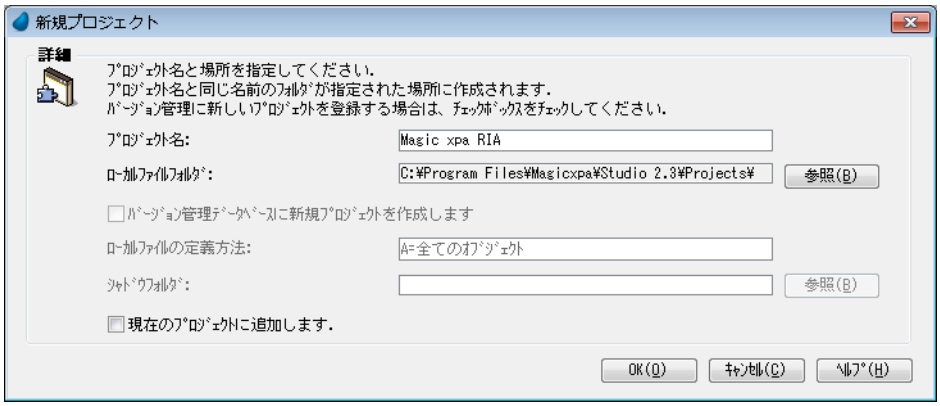
入門プロジェクトの作成

1. Magic xpa Enterprise Studio を開きます。



2. 開始ウィンドウで、［新規作成］ボタンをクリックし、［新規プロジェクト］ダイアログを開きます。
3. プロジェクト名として、＜ Magic xpa RIA ＞と入力します。

4. 位置のデフォルトは「Magic xpa をインストールしたフォルダ ¥」Projects¥ ですが、必要に応じて［参照］ボタンをクリックするか、直接入力などにより、変更することができます。



5. ［OK］ボタンをクリックして確定します。

プロジェクトを登録すると、プロジェクトファイルが作成され、新規プロジェクトがオープンします。

インターネット アプリケーション を開発する場合は、プロジェクト 名に日本語を使用しないようにしてください。

３．新規プロジェクトの詳細

［新規プロジェクト］ダイアログボックスは、プロジェクト名と場所を指定することにより、生成されるプロジェクトファイルを自動的に格納します。

特性	説明
プロジェクト名	新しいプロジェクト名を入力します。 そのプロジェクト名を持つフォルダとファイルが作成されます。 (プロジェクト名には次の文字は使用できません：/?:&\=<>|#%*) プロジェクト名の先頭には $ を使用することはできません。
位置	参照ボタンを押下して、新規プロジェクトを作成したいフォルダの場所を検索します。 デフォルトは、「Magic xpa をインストールしたフォルダ ¥」Projects¥ です。 指定した場所にすでにプロジェクトフォルダが存在するとき、次のエラーがステータス行に表示されます。 ［指定された場所にはすでにプロジェクトファイルが存在します。］ 指定したフォルダが存在しないとき、そのフォルダを作成してよいかどうかの確認を求められます。
バージョン管理データベースに新規プロジェクトを作成します。	Magic xpa ではプロジェクトをバージョン管理データベースに登録し、 バージョン管理のすぐれた機能を利用できます。 このチェックボックスをオンにすることで、バージョン管理データベースにそのプロジェクトが登録されます。

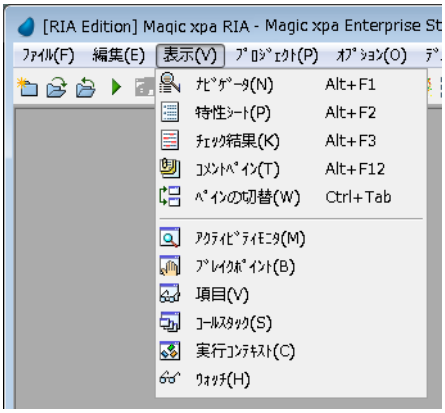
バージョ ン管理の機能については、この入門コースでは取り扱いません。

４．［ナビゲータ］ペイン

［ナビゲータ］ペインによって、Magic xpa Enterprise Studio で取り扱うプロジェクトオブジェクトにすばやくアクセスできます。

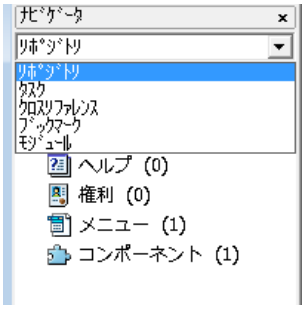
Magic xpa Enterprise Studio では、［ナビゲータ］ペインの表示／非表示を選択できます。通常、［ナビゲータ］ペインは Magic xpa Enterprise Studio の左側に配置して使用します。

1. ［表示］メニューから［ナビゲータ］（Alt+F1）を選択することで、［ナビゲータ］ペインの表示／非表示を切り替えます。



［ナビゲータ］ペインの最初のエントリーは選択ボックスで、5 つのカテゴリから一つ選ぶようになっています。

2. ドロップダウンリストを開いてカテゴリを表示します。
3. リポジトリを選択します



［ナビゲータ］ペインの選択オプション

オプション	説明
リポジトリ	このオプションを選択することで、プロジェクトの次のリポジトリ内容を表示します。 • モデル • データ • プログラム • ヘルプ • 権利 • メニュー • コンポーネント
タスク	選択したプログラムを編集するとき、このオプションはプログラムのタスクツリー（メインタスクとそのサブタスク）を表示します。
クロスリファレンス	検索ユーティリティを使用することで、対象のオブジェクトがどこで使われているかを調べることができます。 クロスリファレンスオプションを選択することで、その結果が表示されます。
ブックマーク	オブジェクトにブックマークを定義できます。 ブックマークオプションを選択することで、ブックマークを持つオブジェクトが表示されます。
モジュール	既存のプロジェクトにプロジェクトをモジュールとして追加できます。 モジュールオプションを選択することで、現在のプロジェクトにモジュールとして登録されているプロジェクトが表示されます。

リポジトリは Magic xpa のリポジトリと、各リポジトリに含まれる要素の数を一覧表示します。

Magic xpa には次のようなリポジトリがあります。

- モデル

- データ
- プログラム
- ヘルプ
- 権利
- メニュー
- コンポーネント

マウスの左ボタンクリックによりリポジトリを選択すると、その内容がワークスペースに表示されます。たとえば、［プログラム］リポジトリをクリックすると、そのプロジェクトの［プログラム］リポジトリ内容が一覧表示されます。

５．特性シート

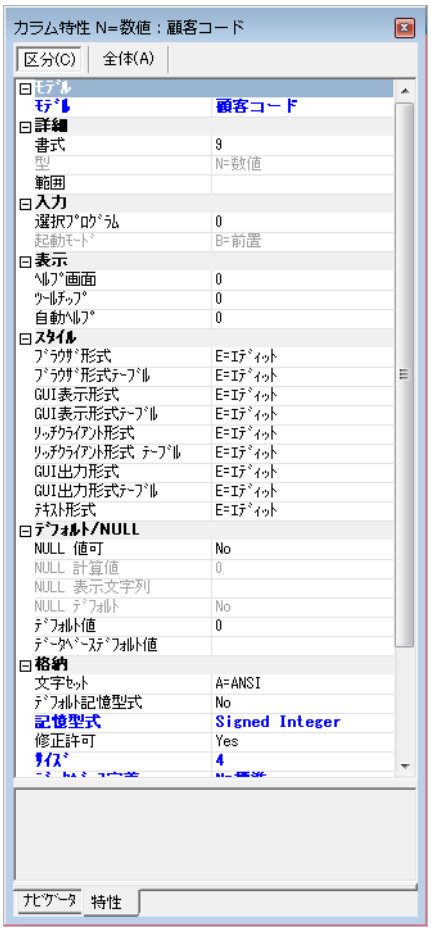
特性シートには、選択されたオブジェクトの特性が一覧表示されます。特性シートを表示するには、［表示］メニューから［特性］を選択します。ここに示しているのは、本コースで作成してゆくプログラムのなかで定義する、あるデータ項目の特性です。

特性シートは、次のオブジェクトに対応しています。

- モデル
- 変数およびカラム
- フォーム
- コントロール
- ヘルプ画面

別のオブジェクトが選択されると、特性シートの表示内容も対応して変わります。

- 特性の表示順序をアルファベット順（全体）またはカテゴリ別（区分）の二通りから選択できます。
- 変更された特性は異なる色（青）により識別されます。

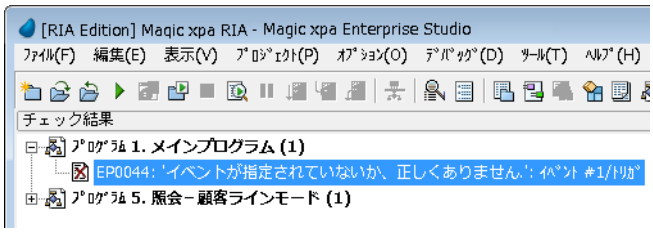


６．［チェック結果］ペイン

Magic xpa にはオブジェクトの定義エラーをチェックする機能があります。最新のチェック内容は［チェック結果］ペインに表示されます。

［チェック結果］ペインを表示するに、［表示］メニューから［チェック結果］を選択します。

ここに例示しているのは、本コースで作成してゆくプログラムのなかで定義する、あるプログラムについてチェックした結果です。



［チェック結果］ペインは、デフォルトではワーク領域の下に表示されます。

チェック結果はツリー構造で表示され、ツリーのレベルは、オブジェクト、メッセージタイプ、またはその両方をもとにソートすることができます。

7 .［コメント］ペイン

Magic xpa はプロジェクトの各オブジェクトにコメントをつけることができます。

具体的には、次のオブジェクトが対象になります。

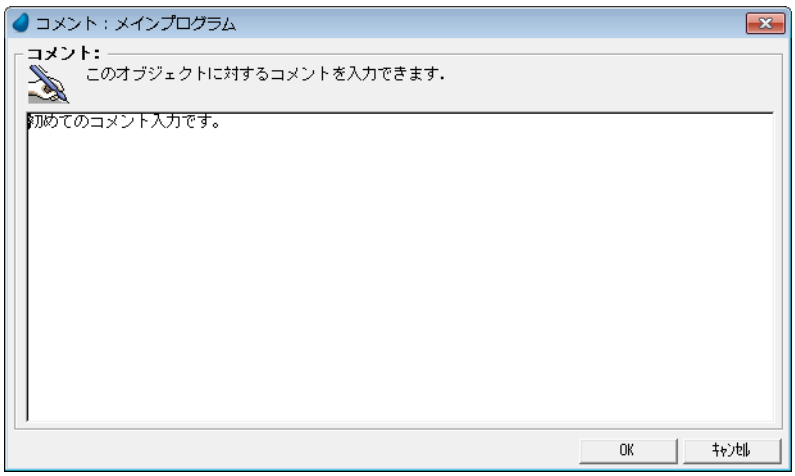
モデル、カラム、変数、データソース、インデックス、外部キー、プログラム、タスク、コンポーネント、ダイレクト SQL、イベント、ロジックユニット、ヘルプ、範囲、権利、入出力ファイル

オブジェクトへのコメント作成

たとえばメインプログラムにコメントをつける場合の手順は次のようになります。

1. ［プログラム］リポジトリを開きます。
2. メインプログラムを選択します。
3. ［オプション］メニューから［コメント］（F12）を選択します。

4. ［コメント］ボックスが表示されるので、コメント文を入力します。
5. ［OK］ボタンをクリックして［コメント］ボックスを閉じます。

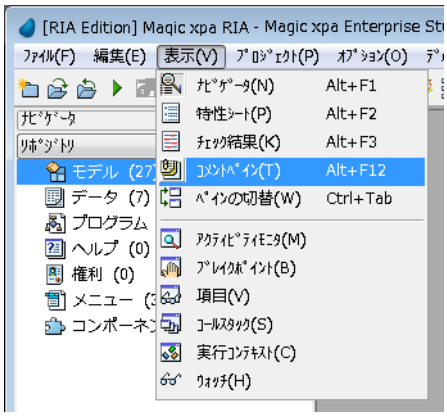


［コメント］ペインでコメントを見る

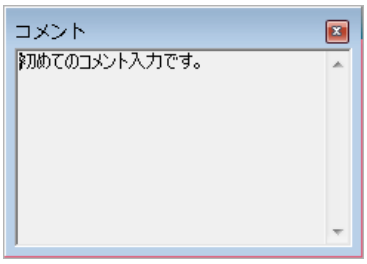
［コメント］ペインには、選択されたオブジェクトのコメントが表示されます。

オブジェクトのコメントを見るためには次のようにします。

1. ［表示］メニューから［コメントペイン］（Alt+F12 キー）を選択します。



2. 上記の例の場合、［プログラム］リポジトリのメインプログラムを選択します。プログラムにつけられているコメントの例を次に示します。



８．ステータス行

ステータス行は、Magic xpa Enterprise Studio または実行版のウィンドウの最下行に表示されます。

Studio のステータス行はいくつかの領域に分けられます。

1. ログイン後には、ロギングボックスにログインしたユーザ名が表示されます。
2. ステータスボックスには、選択されているオブジェクトがズーム対象かどうかが表示されます。
3. 広域／通常ボックスには、選択されている［エディット］コントロールが広域編集可能な場合、［広域］と表示されます。（このボックスには現在の状態が表示されるため、すでに広域編集状態になっている場合は、［通常］と表示されます。）
4. 挿入／上書ボックスでは、現在の編集モードが［挿入］か［上書］かが表示されます。

5. 実行版では、自動ヘルプや入力チェックの警告の表示のみ行われます。

９．要約

本章では Magic xpa Enterprise Studio のユーザインタフェースについて紹介し、次の内容について学びました。

- ［ナビゲータ］ペインの表示オプションと各リポジトリを選択表示させる方法
- 選択したオブジェクトの特性シートを表示させる方法
- ［コメント］ペインにオブジェクトのコメントを表示させる方法
- 最新のチェック結果を表示させる方法
- ステータス行に表示される内容

第 3 章 プログラムの作成

Magic xpa ではプロジェクトベースの開発を行います。本章ではプロジェクトの概念について学び、プログラム作成を行います。

キーワード

- ［プログラム］リポジトリ
- プログラム
- フォーム
- 変数項目
- 型
- 書式

学習目標

本章を学ぶことで、次の内容に対する理解を深めましょう。

- プロジェクトの概念
- 簡単なプログラムの作成
- 変数項目の定義
- フォームに変数項目を追加
- プログラムの実行

参照

Magic ヘルプの［Magic xpa リファレンス／プロジェクトとアプリケーション］

同様に、［Magic xpa リファレンス／プログラム］

１．はじめに

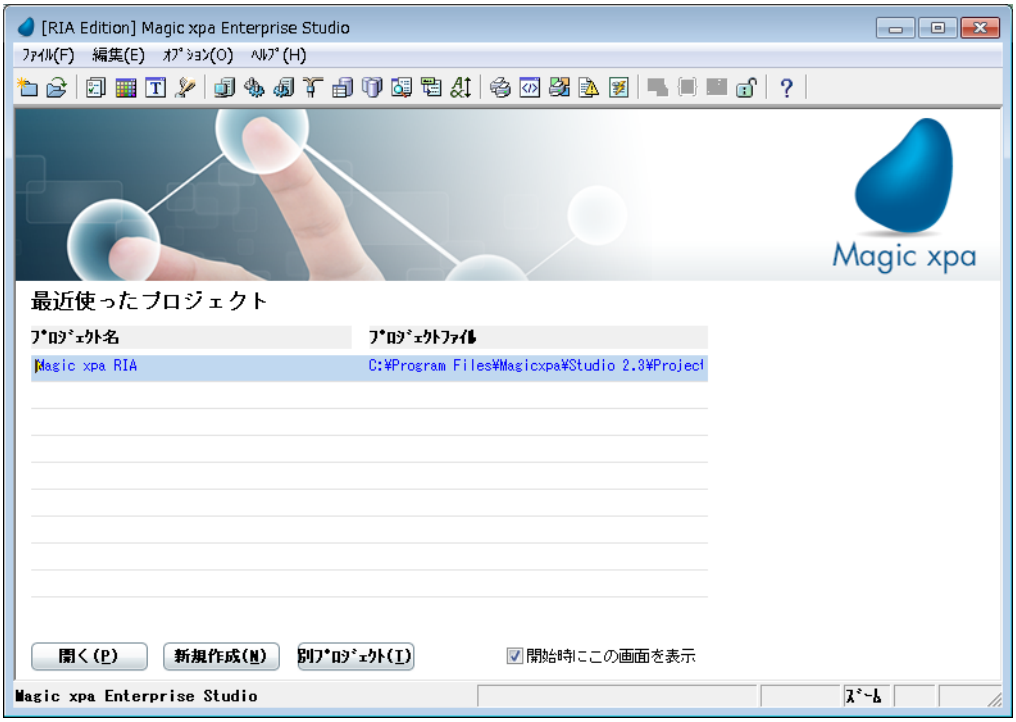
プログラムはプロジェクトの基本的要素です。開発ツールや開発言語の基本は、プログラムを書くことです。

本章では、Magic xpa を使用して、はじめてのプログラム作成を行います。そのプログラムを動作させ、結果を確認してみましょう。

２．Magic xpa RIA 入門プロジェクトの開始

前章で作成した< Magic xpa RIA >プロジェクトを開き、プログラムを作成しましょう。

1. 開始画面で［最近使ったプロジェクト］一覧から< Magic xpa RIA >プロジェクトを選択し、［開く］ボタンをクリックします。



プロジェクトのソースファイル

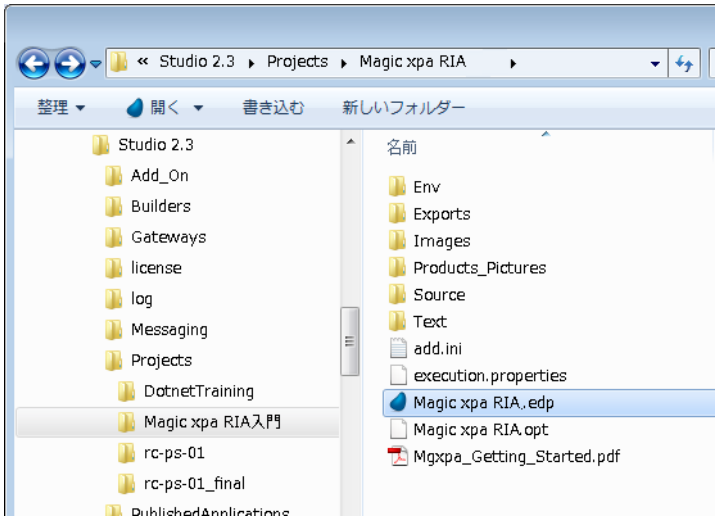
プロジェクトを新規作成するとき、Magic エンジンは［位置］で指定したフォルダの中に、そのプロジェクト名のフォルダを作成します。すべてのプロジェクトファイルはこのフォルダ内に保存されます。

プロジェクトには、次のものが含まれています。

- メインプロジェクトファイル …… プロジェクト名 .edp の形式のファイル
- ［Source］サブフォルダ …… プロジェクトのすべてのソースファイルが保存されるサブフォルダ
- ［Exports］サブフォルダ …… エクスポートファイルを保存するためのサブフォルダ

プロジェクトソースファイルの参照

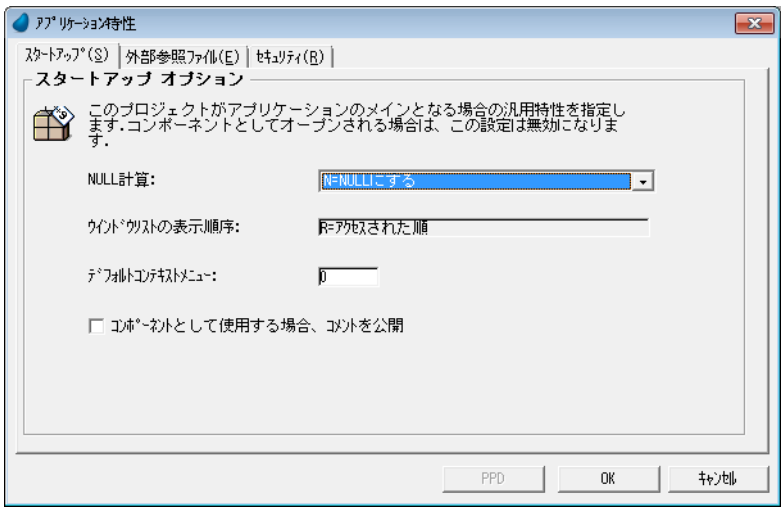
プロジェクトのフォルダ（デフォルトでは、C:¥Program Files¥Magicxpa¥Studio 2.3¥Projects¥Magic xpa RIA）を開いて、実際に内容を確かめてみましょう。



アプリケーション特性

プロジェクトは特性を持っています。現在開いているプロジェクトの特性は、プロジェクトソースファイルの一部として保存されています。

1. ［ファイル］メニューから［アプリケーション特性］(Ctrl+Shift+P) を選択します。



アプリケーション特性は三つのカテゴリに分類されます。

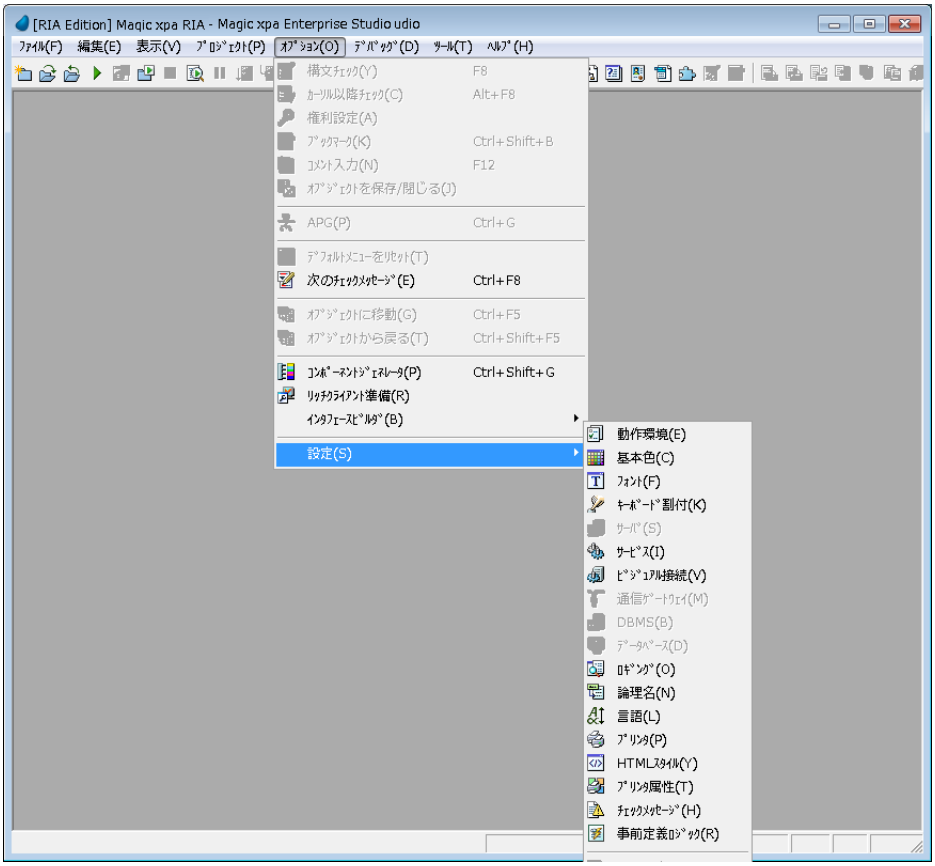
- スタートアップ
- 外部参照ファイル
- セキュリティ

［スタートアップ］と［セキュリティ］については、入門コースでは扱いません。これらについての詳細な説明は、Magic ヘルプの［Magic xpa リファレンス／プロジェクトとアプリケーション／プロジェクト開発／アプリケーション特性］のセクションを参照してください。

Magic xpa の設定

Magic xpa ではすべてのプロジェクトに適用される一般的設定と、各プロジェクトに対する個別の設定を定義します。

1. ［オプション］メニューから［設定］を選択し、［設定］サブメニューを表示させます。



［設定］サブメニューにある各オプションの設定内容を変更することにより、すべてのプロジェクトに対するデフォルト設定が変更されます。

たとえば、［基本色］オプションを選択し、基本色テーブルにエントリを追加すれば、他のプロジェクトからもそのエントリが使えるようになります。

また、次のオプションは、各プロジェクトごとに固有の設定を行うこともできます。

- 基本色
- フォント
- キーボード割付
- プリンタ属性
- HTML スタイル

これらのオプションについては［アプリケーション特性］の［外部参照ファイル］で定義できます。

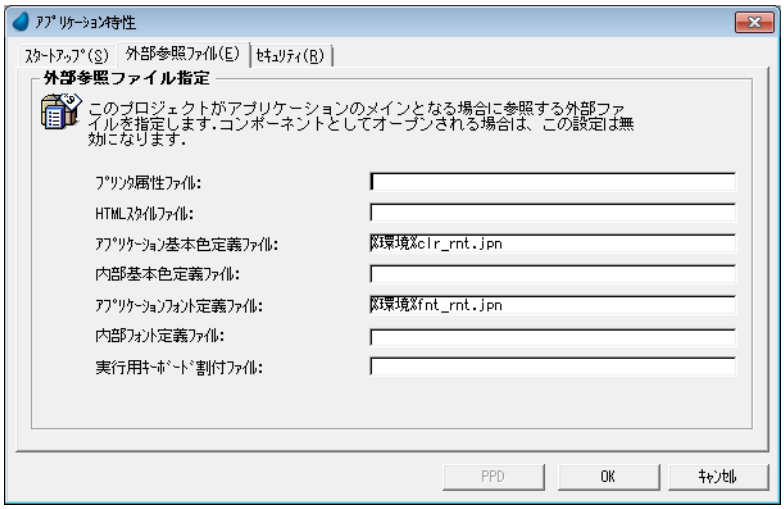
外部参照ファイル

外部参照ファイルを定義することにより、デフォルトの特性設定ファイルを置き換えます。

これらを定義することにより、各プロジェクトにユニークな特性を持たせることができます。

外部参照ファイルを定義するには、次のようにします。

1. ［アプリケーション特性］ダイアログボックスを開き、［外部参照ファイル］タブをクリックします。
2. 設定する特性ファイルのフォルダ位置情報とファイル名を指定します。



ここで参照するファイルは、サーバ上に保存されています。

３．プログラムの新規作成

プログラムを新規に作成するときは、次の手順で行います。

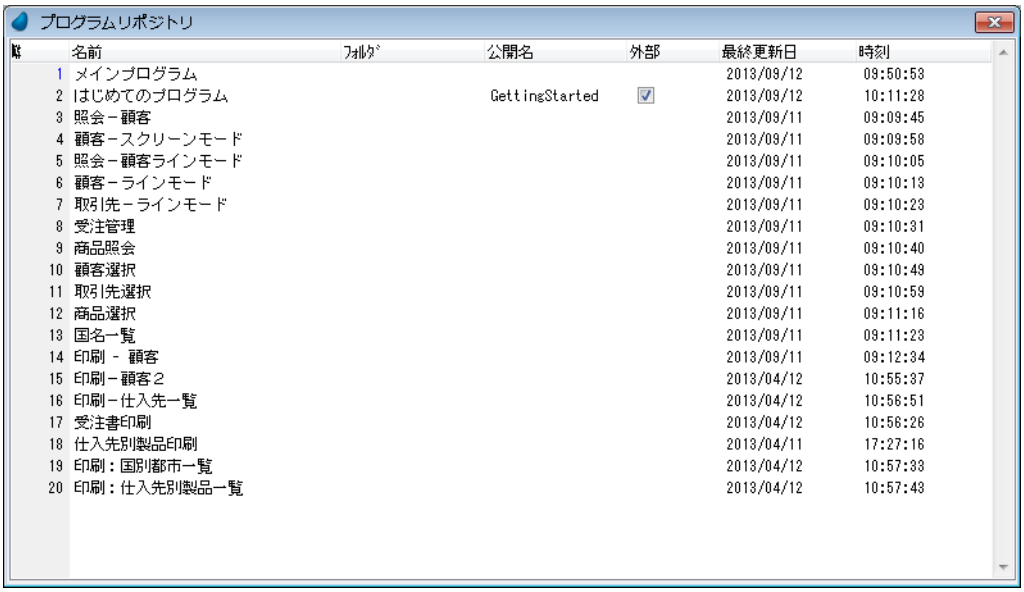
- ［プログラム］リポジトリにエントリを一つ追加します。
- タスク特性を定義します。
- タスクデータビューを定義します。
- タスクロジックを定義します。
- タスクフォームを定義します。

プログラムを作成したら、それを実行し、結果を確認します。

［プログラム］リポジトリ

［プログラム］リポジトリはプロジェクト内のプログラムエントリの一覧であり、先頭には常にメインプログラムが定義されています。

［プログラム］リポジトリの例を示します。



#	名前	フォルダ	公開名	外部	最終更新日	時刻
1	メインプログラム				2013/09/12	09:50:53
2	はじめてのプログラム		GettingStarted	☑	2013/09/12	10:11:28
3	照会－顧客				2013/09/11	09:09:45
4	顧客－スクリーンモード				2013/09/11	09:09:58
5	照会－顧客ラインモード				2013/09/11	09:10:05
6	顧客－ラインモード				2013/09/11	09:10:13
7	取引先－ラインモード				2013/09/11	09:10:23
8	受注管理				2013/09/11	09:10:31
9	商品照会				2013/09/11	09:10:40
10	顧客選択				2013/09/11	09:10:49
11	取引先選択				2013/09/11	09:10:59
12	商品選択				2013/09/11	09:11:16
13	国名一覧				2013/09/11	09:11:23
14	印刷 - 顧客				2013/09/11	09:12:34
15	印刷－顧客２				2013/04/12	10:55:37
16	印刷－仕入先一覧				2013/04/12	10:56:51
17	受注書印刷				2013/04/12	10:56:26
18	仕入先別製品印刷				2013/04/11	17:27:16
19	印刷：国別都市一覧				2013/04/12	10:57:33
20	印刷：仕入先別製品一覧				2013/04/12	10:57:43

［プログラム］リポジトリの各カラム

カラム	説明
#	プログラムの識別番号で、Magic xpa により自動的につけられます。 ［プログラム］リポジトリにプログラムの追加／削除／移動などの操作が行われると、Magic xpa は自動的に番号を振りなおします。 この番号は手動で変更することはできません。
名前	この欄にはプログラムの名前を定義します。 名前は必ずしもユニークである必要はなく、英数字や記号を含めることもできます。
フォルダ	フォルダを定義することで、プログラムをグループ化できます。 このカラムでは定義されたフォルダからプログラムの所属するフォルダを選択します。 このカラムはフォルダが定義されているときのみ有効です。
公開名	このカラムはプログラムの公開名を定義します。 公開名はプロジェクト内ではユニークでなければなりません。公開名は RIA をインターネットで公開する場合の起動プログラムで必要になります。
外部	このカラムは外部からの呼び出しを可能にするかどうかを指定します。 このカラムは公開名が定義されているときのみ有効です。
最終更新日	プログラム内容が更新されたとき、Magic xpa が自動的に日付を更新します。
時刻	更新されたときの時刻が自動的に記録されます。

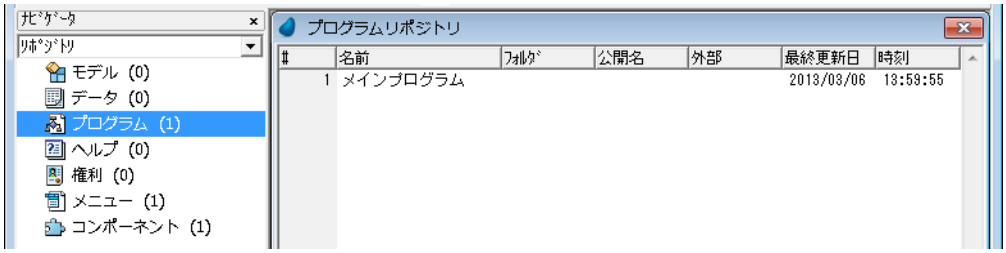
MDI フォームの定義

プログラムを作成する前に、アプリケーションの MDI フォームを定義します。MDI（Multi Document Interface）フォーム上を定義することで、実行時に 1 つの親ウィンドウ内に複数のウィンドウを表示させることができます。メニューやステータス行も MDI 上に表示させることができるため、管理しやすくなります。また、このフォームのタイトルがアプリケーション名として常に表示されるようになります。

MDI フォームは必須ではありません。RIA では、MDI を使用しない表示がデフォルトになりますが、本書の説明では、MDI にプログラムのウィンドウを表示させることを前提に作成します。

Magic xpa では、メインプログラムで定義されたリッチクライアント表示フォームが、Magic アプリケーションの MDI フォームとなります。

1. ［プロジェクト］メニューから［プログラム］（Shift+F3）を選択します。
2. 最初の行（メインプログラム）にカーソルをパークさせます。
3. ［編集］メニューから［ズーム］（F5）を選択して、メインプログラムを開きます。

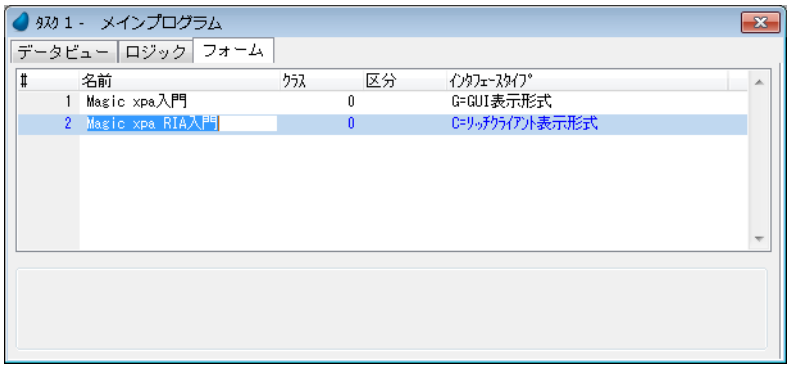


メインプログラムについて

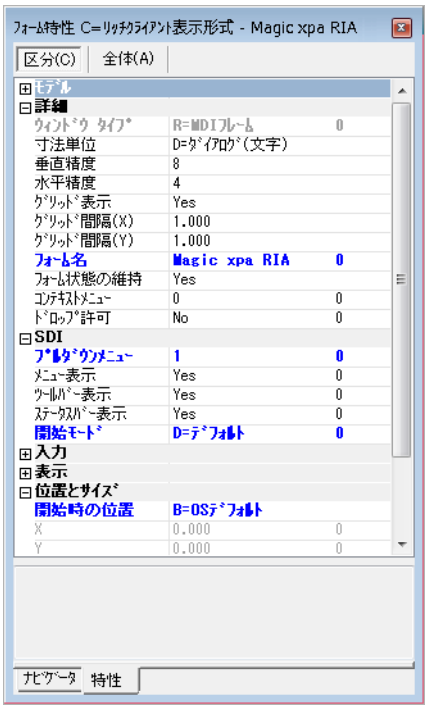
メインプログラムは、Magic アプリケーションの最も上位に位置するプログラムです。アプリケーションの起動時に最初に実行され、終了時には最後に実行されます。

ここに定義された変数項目やフォーム、イベントは、同一プロジェクト内の全てのプログラムで利用することができます。

1. メインプログラムを開いた状態で、Ctrl+3 を押下して［フォーム］タブに切り替えます。
2. 1 行にカーソルを置いた状態で F4 キーを押下して一行追加します。［名前］欄に「Magic xpa RIA 入門」と入力します。
3. ［インタフェース］欄で「C= リッチクライアント表示」を選択します。



4. Alt+Enter を押下して、［フォーム特性］ペインを開きます。以下の特性を変更します。
   - 開始時の位置 …… B=OS デフォルト
   - 開始モード …… D= デフォルト



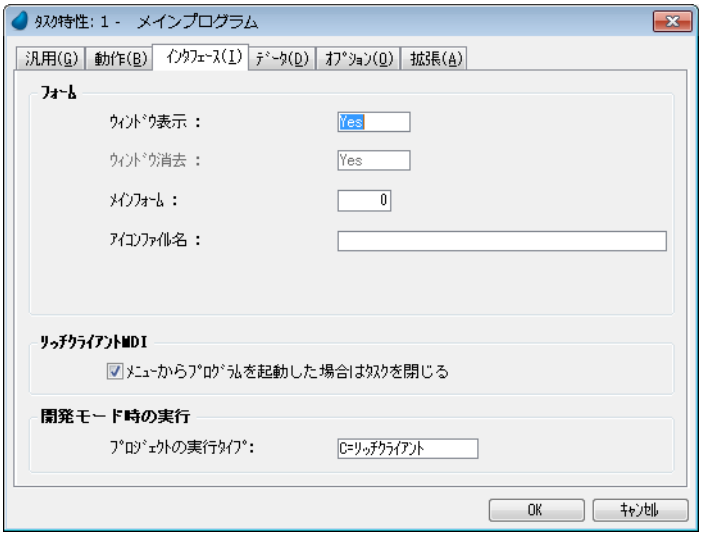
［開始時の位置］特性によって MDI のデフォルト の表示位置は以下のようになります。

- **C= カスタム …… ［X］／［Y］特性で指定された位置に表示されます。**
- **B=OS デフォルト…… 位置とサイズは OS に依存します。**
- **P= 親ウィンドウの中央…… デスクトップの中央に表示されます。**
- **M=MDI の中央…… デスクトップの中央に表示されます。**
- **D= デスクトップの中央…… デスクトップの中央に表示されます。**
- **L=OS デフォルト位置…… 位置は OS に依存します。**

タスク特性の設定

メインプログラムの［タスク特性］でリッチクライアント環境の設定を行います。

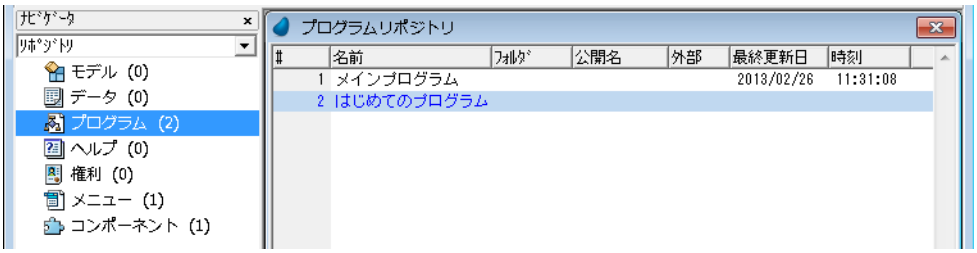
1. Ctrl+P を押下してタスク特性を開きます。
2. ［インタフェース］タブをクリックして切り替えます。
3. 以下の特性を設定します。この設定は、第 24 章 「アプリケーションの公開」でプロジェクトとして実行するときに関係します。
   - メニューからプログラムを起動した場合はタスクを閉じる …… チェック
   - プロジェクトの実行タイプ …… C= リッチクライアント



［プログラム］リポジトリにプログラムを作成

1. ［プロジェクト］メニューから［プログラム］（Shift+F3）を選択します。
2. 最初の行（メインプログラム）にカーソルをパークさせます。
3. ［編集］メニューから［行作成］（F4）を選択します。

4. 作成された行の［名前］カラムに、**＜はじめてのプログラム＞**と入力してください。



タスクについて

Magic xpa では［プログラム］と［タスク］という用語を、いずれもオペレーションコードの意味で用いています。

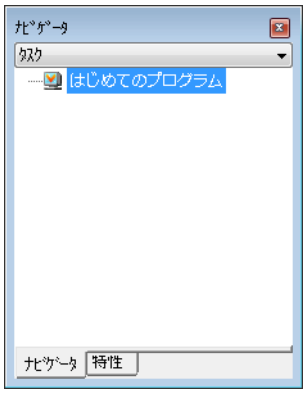
Magic xpa では、［タスク］は［プログラム］を構成する基本単位を意味します。一つの［プログラム］は通常、［メインタスク］と複数の［サブタスク］によって構成されています。

もし［プログラム］が一つの［メインタスク］だけで構成されているとすれば、それは［タスク］と呼ぶこともできます。

1. ＜はじめてのプログラム＞にカーソルをパークしてください。
2. F5 キーを押下するか、マウスのダブルクリックにより、プログラム内部にズームしましょう。

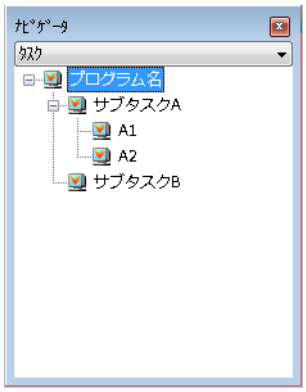
［ナビゲータ］ペインにはタスクツリーが表示されます。

この場合、＜はじめてのプログラム＞は一つのメインタスクだけがある状態です。



次の画像では、複数のタスクで構成されるプログラムの例を示しています。

- タスク「プログラム例」がメインタスクです。
- 「サブタスク A」と「サブタスク B」は「プログラム例」のサブタスクです。
- 「A1」と「A2」は「サブタスク A」のサブタスクです。



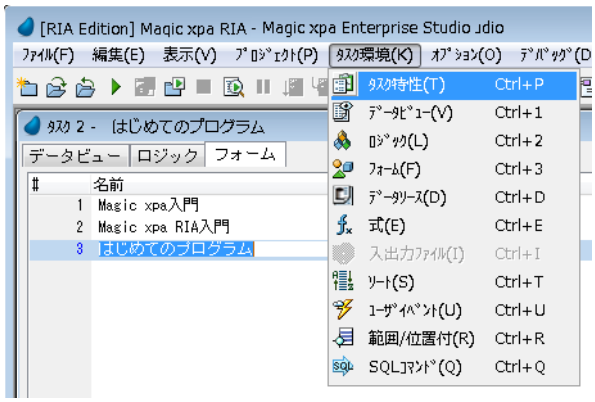
リッチクライアント タスク内には、リッチクライント タスクかバッチタスクのみ定義できます。

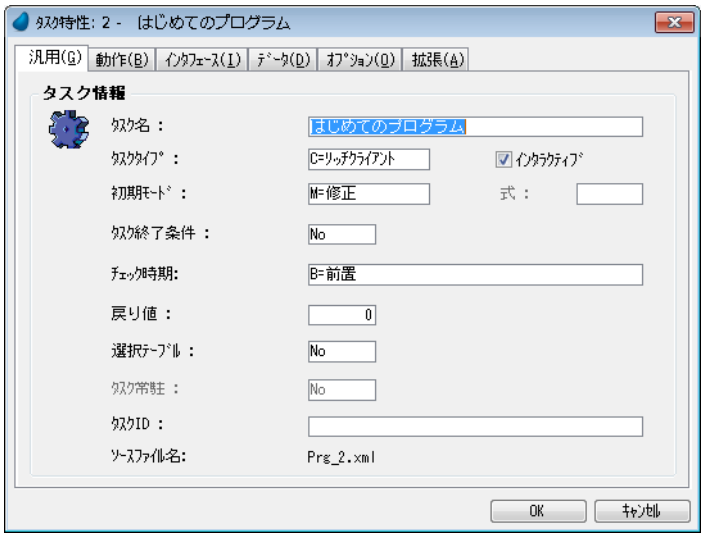
タスク特性

各タスクにはそれぞれ固有の特性を定義します。

タスク特性を定義することで、タスクのタイプ、動作モード、データ、ユーザインタフェースや、その他いろいろな特性を決めることができます。

1. ［タスク環境］メニューから［タスク特性］（Ctrl+P）を選択して［タスク特性］ダイアログを表示させましょう。





タスク特性の各項目

項目	説明
タスク名	タスクの名称を入力します。
タスクタイプ	タスクのタイプを定義します。タスクのタイプには次の三つがあります。 • オンライン …… 対話型のタスクです（本書では扱いません）。 • バッチ …… バッチタスクです。 • リッチクライアント …… リッチクライアントのインターネットプログラムを動作させるタスクです。
インタラクティブ	［タスクタイプ］特性が「リッチクライアント」の場合のみ有効です。ユーザ操作を必要とするタスクを定義する場合は、チェックします。 チェックを外すと、バッチタスクと同じような動作になります。
初期モード	タスクの動作開始時点の動作モード（照会、修正、登録、削除など）を定義します。

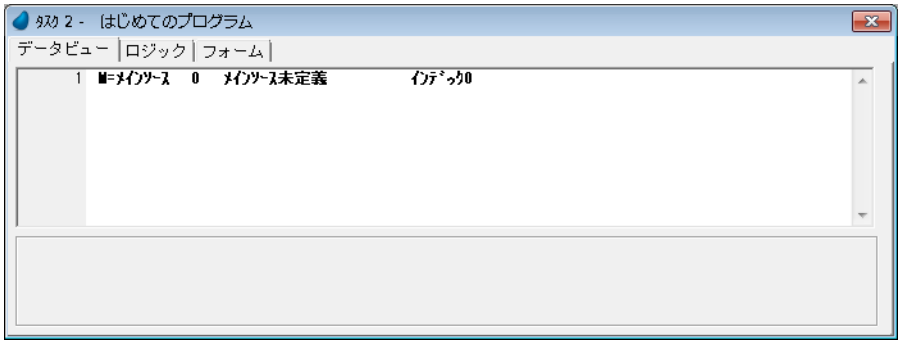
- **ここで説明した特性は、本コースに関連しているものに限定しています。**
- **タスクタイプのデフォルト値は、条件により異なります。RIA Edition を使用した場合は、「リッチクライアント」になります。**

2. ［OK］ボタンをクリックして［タスク特性］ダイアログを閉じ、設定した内容を保存します。

４．タスクのインタフェース

タスクにズームするとタスクインタフェースが表示されます。タスクインタフェースは三つのエディタで構成されています。

- データビュー
- ロジック
- フォーム



各エディタの切り替えは次のように行ないます。

- Ctrl+Tab キーを押下することでエディタが順に切り替わります。
- Ctrl+1 キーを押下するか、［データビュー］タブをクリックすると、［データビュー］エディタが選択されます。
- Ctrl+2 キーを押下するか、［ロジック］タブをクリックすると、［ロジック］エディタが選択されます。
- Ctrl+3 キーを押下するか、［フォーム］タブをクリックすると、［フォーム］エディタが選択されます。

５．［データビュー］エディタ

［データビュー］エディタではタスクのデータソースと変数項目などを定義します。

［データビュー］エディタでは次のことを行います。

- タスクのメインデータソース、またはダイレクト SQL ステートメントを定義します。
- 他のデータソースにリンクします。
- データソースのカラム、変数項目、パラメータを定義します。
- 追加するデータソースを宣言します。

データビュー行の種類

［データビュー］エディタでは、二つのタイプの行を作成します。

- ヘッダ行 …… ヘッダ行では、タスクのデータソースタイプと特性を定義します。特に最初のヘッダ行では、タスクがメインデータソースを使用するタスクか、SQL コマンドタスクにするかを指定します。
- 詳細行 …… ヘッダ行の下に作成する詳細行では、タスクで使用するデータソースのカラム、変数項目、パラメータなどを定義します。

本章では、変数項目の定義についてのみ説明します。

データビューの定義

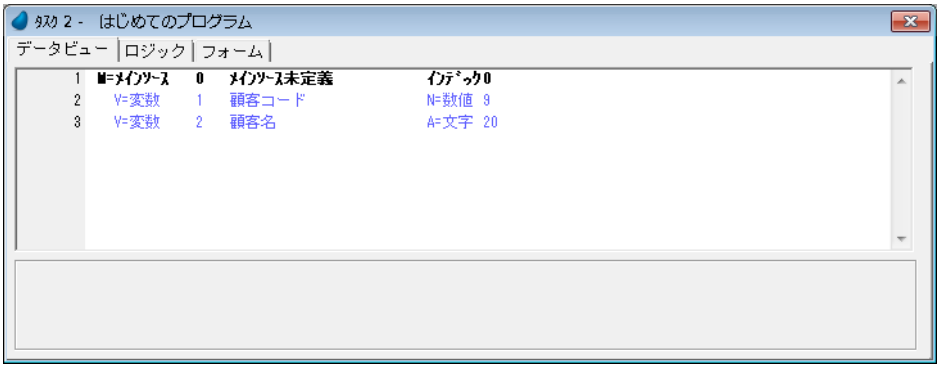
変数項目を使用して、このプログラムのデータビューを定義してみましょう。

変数項目はローカルに定義され、プログラム実行中に一時的に値を保持するために使われます。変数項目の値はそのプログラム内でのみ保持され、プログラムの実行が終われば無効になります。再びプログラム実行を開始するとき、変数項目の値はデフォルト値に初期化されます。

変数項目の作成

1. ［データビュー］エディタを選択します。
2. ［編集］メニューから［行作成］を選択し、2 行追加します。
3. 下の図の 2 行目、3 行目に示すように行を定義します。

項目タイプ	項目番号	項目名	モデル	型	書式	範囲	終了	代入
V= 変数	1	顧客コード	0	N= 数値	9	0	0	0
V= 変数	2	顧客名	0	A= 文字	20	0	0	0



変数項目行の定義項目

特性	説明
項目タイプ	V= 変数、P= パラメータ から選択します。
項目番号	Magic xpa が自動的に割り付けます。
項目名	変数項目の名称を入力します。
モデル	変数項目をモデルを使用して定義する場合、そのモデル番号を入力します。 ズームで［モデル］リポジトリが表示されます。モデルについては、第 21 章 「オブジェクト定義の一元管理」で学習します。
型	変数項目の型を選択します。モデルが使用される場合は無効になります。
範囲、終了	範囲の最小値と最大値を式で定義し、ここにはその式番号を指定します。 ズームで式テーブルが表示されます。
代入	変数項目の初期化に用いるデフォルト値を式で定義し、ここにはその式番号を指定します。 ズームで式テーブルが表示されます。

６．タスクのフォーム

多くのプロジェクトではユーザとの対話操作が行われます。フォームはその操作を行うためのインタフェースを提供します。タスクが実行されるとき、フォームは画面に表示されたり、あるいは印刷帳票としてプリンタ出力やファイル出力などに用いられます。フォーム上には、必要な情報を表示するためのコントロールや、ユーザがデータを入力したり選択入力するためのコントロールを配置します。

各タスクにはデフォルトで一つのメインフォームがありますが、必要に応じて追加することができます。

リッチクライアントタスクでは、タスクのデータビューを表示し、エンドユーザの対話操作ができるようにします。

バッチタスクでは、タスクが実行されるときの処理の進捗度を表示させるようにできます。

タスクのメインフォームの作成

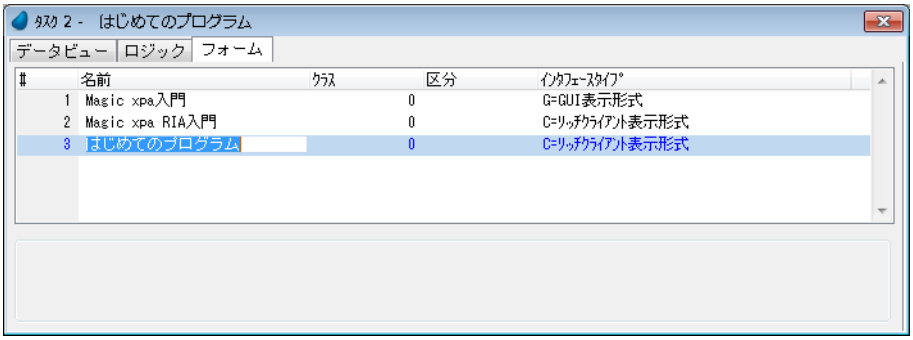
［フォーム］エディタを選択するとタスクのフォーム定義一覧が表示されます。

新しくタスクを作成したとき、Magic xpa は自動的に［フォーム］エディタにエントリを一つ追加し、それがそのタスクのメインフォームとなります。

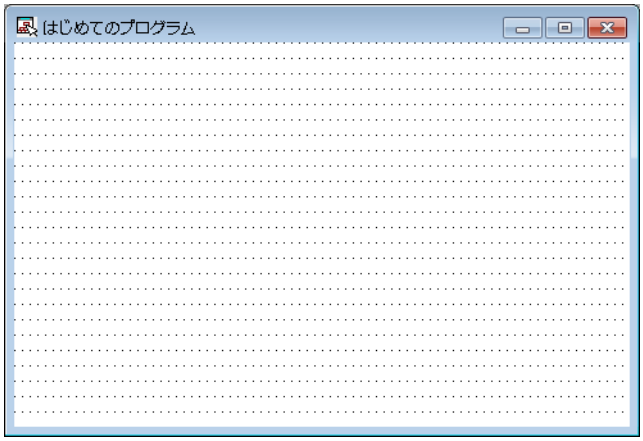
タスクのメインフォームは、削除したり、リポジトリ内での位置を移動させることはできません。

ここで、タスクのメインフォームを定義してみましょう。

1. ［フォーム］エディタを選択します。
2. ＜はじめてのプログラム＞（#2）を選択し、ズーム（F5）します。



［フォーム］エディタの特性



特性	説明
#	フォームの番号。自動的につけられるため、変更することはできません。
フォーム名	デフォルトはタスク名称ですが、自由に変更できます。 フォームを追加した場合はその都度入力が必要です。 フォームが画面表示されるとき、フォーム名がウィンドウタイトルとして表示されます。
クラス	Magic xpa はメインフォームに対してはクラス 0 が自動的に割り付けられます。 クラス 0 のフォームは自由に追加できます。 クラス 0 のフォームは対話操作用のフォームとして用いられます。 また、クラス＞ 0 のフォームを定義できますが、それらのフォームは帳票やレコードの入出力様式の定義、HTML 出力に用いるためのものです。
区分	クラス 0 のフォームでは無効です。 帳票用のフォームでは、D= 明細、H= ヘッダ、F= フッタ、P= ページヘッダ、G= ページフッタから選択指定します。 レコードの入出力様式を定義するフォームの場合、この設定は意味がありません。 これらの詳細については、第 19 章 「帳票印刷」で学習します。
インタフェースタイプ	クラス >0 のフォームで選択指定できます。 • T= テキスト形式 …… テキスト帳票出力のとき指定します。 • G=GUI 出力形式 …… GUI 帳票出力のとき指定します。 • F= フレームセット形式 …… 複数のフレームで構成される HTML ファイルを作る場合に指定します。 • M= マージ形式 …… 既存のテンプレートファイルを利用して動的にデータをマージし、ファイルを作る場合に指定します。

［フォーム］エディタでは上位のタスクのフォームが階層的に表示されます。したがって［メインプログラム］フォームが常に最初の行に表示されています。

［フォーム］エディタのパレット

［フォーム］エディタにはパレットと呼ぶツールがあり、このパレットを利用してフォームを作成します。

- ［フォーム］エディタパレットは［フォーム］エディタの一覧からフォームを選択してズームし、フォームを開くと表示されます。
- フォームが開いているときに、［表示］メニューから［フォームエディタパレット］を選択することで、その表示の ON/OFF を切り替えることができます。

［コントロール］パレット

フォーム上に配置するコントロールを選択します。フォームのタイプ（クラス 0 またはクラス >0）に基づいた、配置可能なコントロールのアイコンが表示されています。

Magic xpa には二つのタイプのコントロールがあります。

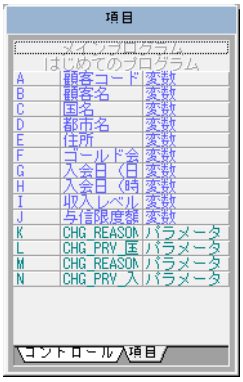
- スタティックコントロール …… ［ラベル］コントロールなど、ユーザが変更できない固定情報を表示します。
- ダイナミックコントロール …… ［エディット］コントロールなど、プログラム実行中にユーザが変更できる動的な情報を表示します。

［コントロール］パレットアイコンの詳細

アイコン	コントロール名	説明
	ラベル	フォーム上にスタティックなテキストを表示します。
	エディット	データ項目や式の値を表示します。
	ラジオボタン	複数の選択肢を持つデータ項目に使用します。このコントロールは選択可能な値を表示し、ユーザはその中から一つを選択します。
	テーブル	テーブル形式のデータを表示します。
	タブ	コントロールをカテゴリ別に分類して表示させたいときに使用します。コントロールはそれぞれ対応するタブに関連づけて配置します。
	プッシュボタン	エンドユーザがクリックすることでアクションのトリガとする場合に使用します。
	コンボボックス	複数の値を持つデータ項目を表示するときに使用しますが、通常は最初の値だけが表示されています。ユーザはコンボボックスをクリックして開くことで、値を選択できるようになります。
	チェックボックス	論理値を持つデータ項目に使用します。ユーザはチェックを入れたり、クリアしたりできます。
	リストボックス	リストとして表示される複数の値を持つデータ項目に使用します。ユーザはリストの中から一つを選択できます。

アイコン	コントロール名	説明
	ツリー	異なるレベルのデータを表す親子ノードを持ったツリー構造のテーブルをメインソースとして使用したタスクにおいて、階層構造を持つデータビューを表示するときに使用します。
	サブフォーム	選択されているフォーム内に他のフォームを表示させる場合に使用します。
	グループ	複数のコントロールをまとめたいときに使用します。グループコントロールにはラベル（名前）をつけることができます。
	イメージ	グラフィックイメージをフォームに配置したいときに使用します。
	ライン	スタティックなラインを表示します。
	リッチテキスト	開発時の編集中にフォントや色などを修正できるスタティックなテキストを表示します。
	リッチエディット	データ項目や式を表示させるときに使用し、リッチテキストと同様に開発時にフォントや色などを修正できます。またこのコントロールには BLOB 型のデータ項目を割り当てることができます。
	.NET	ユーザインタフェースの一部として .NET コントロールを直接配置することができます。
	ブラウザ	HTML ファイルを表示する［ブラウザ］コントロールを挿入します。［リッチクライアント形式］フォームでのみ有効です。

項目パレット



フォーム上に配置する項目を選択します。そのタスクおよび上位タスクで定義しているすべてのデータ項目が表示されます。

項目パレットには項目の文字 ID、名前、データソース名または変数項目、パラメータの項目種類が表示されます。複数のタスクの項目が表示される場合には、タスクの境界にはセパレータとしてタスク名が表示されます。

項目パレットの使用方法

1. フォームに配置したいデータ項目をクリックします（このときマウスボタンを押下し続ける必要はありません）。
2. フォーム上でクリックすると、名前と［エディット］コントロールがペアになって配置されます。

Esc キーを押下するか、セパレータをクリックすることで、選択がキャンセルされます。

［コマンド］パレット

フォームの編集コマンドを選択します。フォーム上のコントロールのを整列、サイズ変更、親子リンクの設定、グループ設定などを行います。

［コマンド］パレットのアイコン

アイコン	名称	説明
	選択左寄せ	選択したコントロールを、選択フレームの左端に揃えます。
	選択右寄せ	選択したコントロールを、選択フレームの右端に揃えます。
	垂直センタリング	選択したコントロールを、フォームの中央に縦に並べます。
	選択垂直センタリング	選択したコントロールを、選択フレームの中央に縦に並べます。
	垂直均等揃え	選択したコントロールを、選択フレーム内に均等間隔で縦に並べます。
	水平均等揃え	選択したコントロールを、選択フレーム内に均等間隔で横に並べます。
	選択上寄せ	選択したコントロールを、選択フレームの上部に揃えます。
	選択下寄せ	選択したコントロールを、選択フレームの下部に揃えます。
	選択水平センタリング	選択したコントロールを、選択フレームの中央に横に並べます。
	水平センタリング	選択したコントロールを、フォームの中央に横に並べます。
	幅最大化	選択したコントロールを、最も幅の広いコントロールに合わせて拡張します。
	幅最小化	選択したコントロールを、最も幅の狭いコントロールに合わせて縮小します。
	高さ最大化	選択したコントロールを、最も縦に長いコントロールに合わせて拡張します。
	高さ最小化	選択したコントロールを、最も縦に短いコントロールに合わせて縮小します。
	前面移動	選択したコントロールを、フォームの 1 つ手前のレイヤに移動します。
	背面移動	選択したコントロールを、フォームの 1 つ後ろのレイヤに移動します。
	最前面移動	選択されているコントロールのレイヤを一番先頭にします。
	最背面移動	選択されているコントロールのレイヤを一番後ろにします。

アイコン	名称	説明
	Z オーダ表示	コントロールの Z オーダをフォームに表示します。
	自動 Z オーダ	フォーム内のコントロールの自動 Z オーダを有効にします。
	タブ順序表示	タブ順序を表示します。
	自動タブ順序	自動タブ順序のを有効／無効を切り替えます。
	区切りなし表示	フォーム内のブレイクとブレイクの間の仕切を除去します。
	元に戻す	最後に行った［フォーム］エディタでの処理を取り消します。
	やり直し	最後に戻した処理を実行します。
	親子リンクの作成	コントロール間の親子リンクを確立します。
	親子リンクの切断	コントロール間の親子リンクを切断します。
	テーブルへの配置	［テーブル］コントロールにコントロールを配置します。
	親の表示	親コントロールを表示します。
	サイズ調整	コントロールのフレームをコントロールの書式サイズに合わせます。

色パレット

コントロールに指定する色を選択します。各色はボタンで表示されています。

7．フォームへの変数項目の配置

ここでは＜顧客コード＞と＜顧客名＞の変数項目をメインフォームに配置してみましょう。

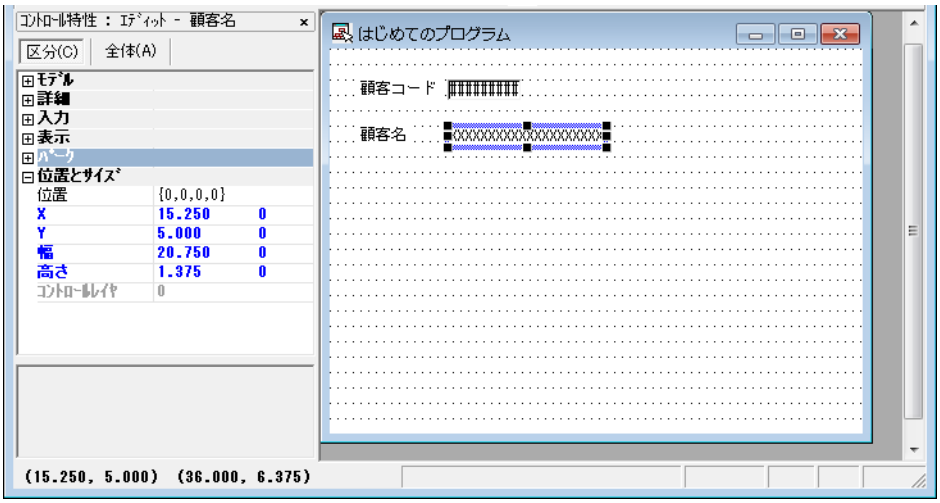
1. 項目パレットを選択します。
2. 項目リストから＜顧客コード＞を選択し、図に示すような位置に配置します。
3. ＜顧客名＞についても同様にして配置します。

コントロールのサイズや位置の変更

コントロールのサイズ（幅と高さ）や位置（左辺、上辺の位置座標）を変更できます。

変更するには次のようにします。

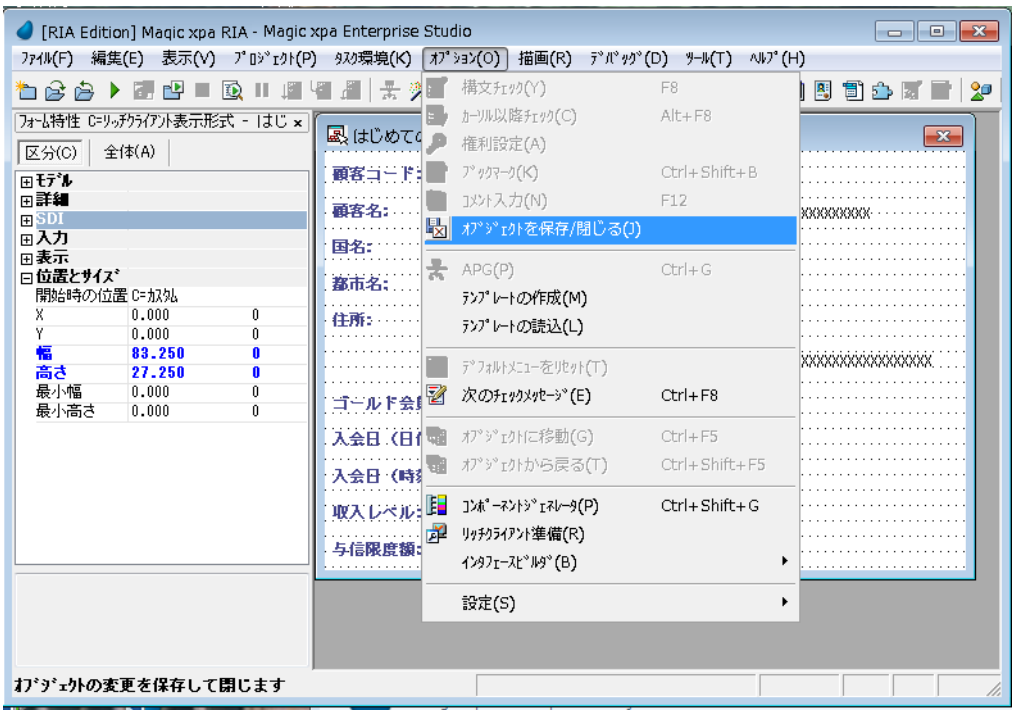
1. コントロールを選択します。
2. マウスによりサイズを変更します。またはコントロール特性の左辺位置、上辺位置、幅、高さなどに直接数値を入力して変更します。



８．プログラムの保存

プログラムを作成した後は、その内容を保存し終了します。

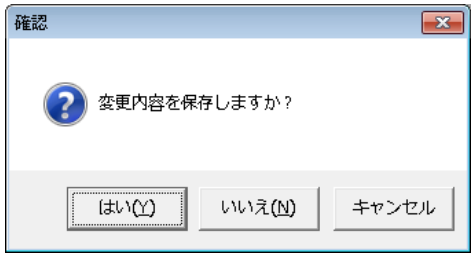
- ［オプション］メニューから［オブジェクトを保存／閉じる］を選択し、内容を保存して終了します。
- ［オプション］メニューから［プログラムの保存］を選択すると、内容を保存しますが、タスクはそのままの状態で表示されます。



Esc キーを押下してプログラムを終了させようとした場合は、変更を保存するかまたは破棄するかの確認を求めるダイアログボックスが表示されます。

このときの動作は次のようになります。

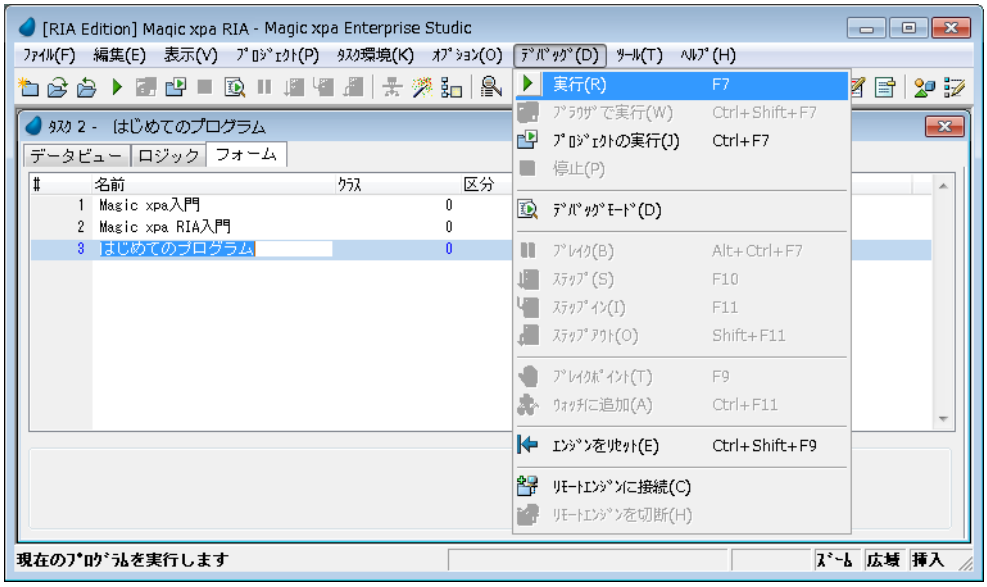
- ［はい］をクリックすると、変更内容を保存し、タスクを終了します。
- ［いいえ］をクリックすると、変更内容を破棄し、タスクを終了します。
- ［キャンセル］をクリックすると、何もせずにダイアログを閉じます。



９．プログラムの実行

ここまでの作業で、Magic xpa Enterprise Studio 環境でプログラムを動かしてみる準備が整いました。

1. ［プログラム］リポジトリで、＜はじめてのプログラム＞を選択します。
2. ［デバッグ］メニューから［実行］（F7）を選択します。



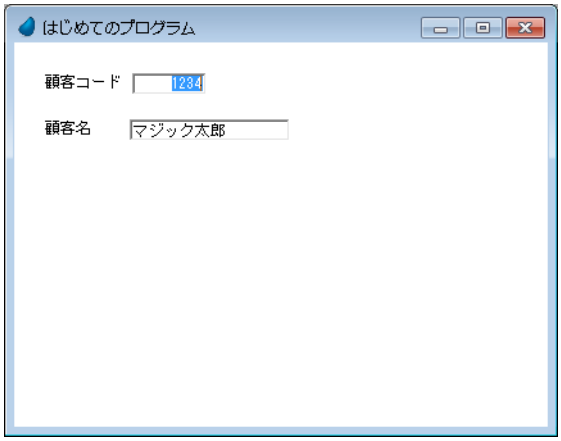
Magic xpa の実行ウィンドウが開き、プログラムのメインフォームが表示されます。

変数項目が表示され、データの入力ができることを確認しましょう。

3. 顧客コードに、たとえば＜ 1234 ＞ と入力してください。
4. Tab キーを押下して、顧客名にカーソルを移動してください。
5. 顧客名に、たとえば<マジック太郎 ＞と入力してください。

このプログラムでは変数項目に値を入力するようになっていますが、変数項目はプログラム実行中のみ値を保持し、プログラムが終了するとその値は破棄されます。

6. プログラムを終了し再度実行させると、変数項目はクリアされて初期状態に戻っています。



１０.ビルトインタスクの動作

ここでは、Magic xpa のビルトインタスクの動きについて二つの例を使用して説明します。この内部の動きについては、後ほどさらに詳しく解説します。

入力データチェック ー 数値項目の場合

Magic xpa では項目の型にしたがって、データ入力時のチェックを行います。

＜顧客コード＞は数値型の変数項目として定義したので、Magic xpa はこの変数項目にデータが入力されるとき、数値のみを受け付けます。

たとえば、＜顧客コード＞に 数字以外の文字を入力しようとしてください。するとステータス行に［数字を入力してください］というメッセージが表示され、入力は受け付けられません。

書式による入力値制限

Magic xpa はデータ項目の書式に基づいて、入力値のチェックを自動的に行います。

＜はじめてのプログラム＞で定義した＜顧客コード＞では、書式を 9 桁としています。したがってこの変数項目には 9 桁以下の数字だけが入力できます。

＜顧客名＞では書式を 20 文字として定義しているので、この変数項目に入力できるのは 20 文字以下ということになります。

実行例：

1. ＜顧客コード＞に 10 桁の数字を入力してみましょう。Magic xpa は 9 文字目までを受け付け、10 文字目は無視します。
2. ＜顧客名＞に 20 バイト（漢字などの 2 バイトコードでは 10 文字に相当）以上の文字を入力してみましょう。Magic xpa は最初の 20 バイトだけを受け付け、その後のキー入力を無視します。

１１ .Magic xpa の型

データ項目にはいろいろな特性があるので、これらの内容についてよく把握しておく必要があります。

重要な特性の一つにデータ項目の［型］があります。

［型］はデータ項目の性質を決定します。

Magic xpa がサポートしている主な［型］

型	説明
文字	英数字、記号、漢字、ひらがな、カタカナなどの任意の文字列です。 文字型はデフォルトの型です。全角文字のみ 2 バイトデータとして処理されます。
Unicode	（文字型と同様に）英数字、記号、漢字、ひらがな、カタカナなどの任意の文字列です。全て 2 バイトデータとして処理されます。
数値	整数または小数です。18 桁までの数値を扱うことができます。
論理	真、または偽の値を持ち、内部的に 1 バイトのデータとして格納されます。［True/False］、［はい / いいえ］などの値の組み合わせを格納する場合、論理型を使用します。
日付	日付を表すデータで、年、月、日からなります。 内部的には数値データで、西暦 1 年 1 月 1 日を基点として数えた日数が格納されています。 日付型データは、画面に表示される場合のみ日付の表示形式に変換され、Magic xpa 内部での計算は、全て数値データを用いて処理されます。
時刻	内部的に秒数のカウンタとして格納されます。 時刻型を使用して、時間の長さ、または時刻を表します。 日付型と同様、時刻型は数値であるため、加減算できます。 時刻型は日付型と同様に、画面に表示される場合にのみ時刻の表示形式に変換されます。
BLOB	バイナリデータがそのまま格納されます。
OLE	Magic xpa のフォーム外で実行される OLE COM オブジェクトのインスタンスを作成するために使用される BLOB 項目を格納します。
ベクトル	指定されたセルインデックスを使用してデータを格納したり、検索することのできる配列です。 ベクトル型は、追加のセルモデル特性を持つ BLOB 型に基づいています。
.NET	.NET アセンブリは、Microsoft .NET によって開発されたコンポーネントの用の標準オブジェクトで、最新の Windows アーキテクチャに基づいた COM コンポーネントに相当するものです。

１２ .Magic xpa の書式

書式では基本的にデータ項目の長さを定義します。

［書式］特性では、そのデータ項目に入力できる文字数（または、バイト数）を指定します。文字数の設定は、数値またはマスク文字（機能指示記号として扱われる特殊な予約文字）によって行われます。

- 数値データ項目の書式で数字の 6 を設定すれば、6 桁の数値であることを意味します。
- 数字の 6 の代わりに、数字を意味するマスク文字 # を用いて ###### と設定しても、6 桁の数値であることを意味します。

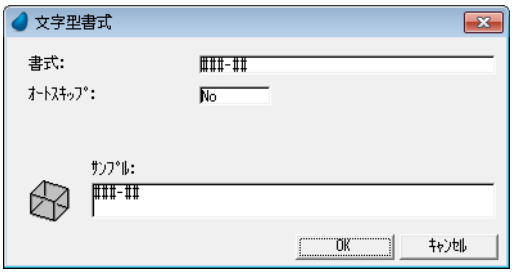
Magic xpa の書式設定では、文字数以外にも型によっていろいろな設定を行うことができます。

書式を詳細に設定する場合は、［書式］特性からズームして［書式］ダイアログを表示させて行います。

［書式］ダイアログで詳細に定義した内容は、［書式］特性では指示記号に変換されて表示されます。

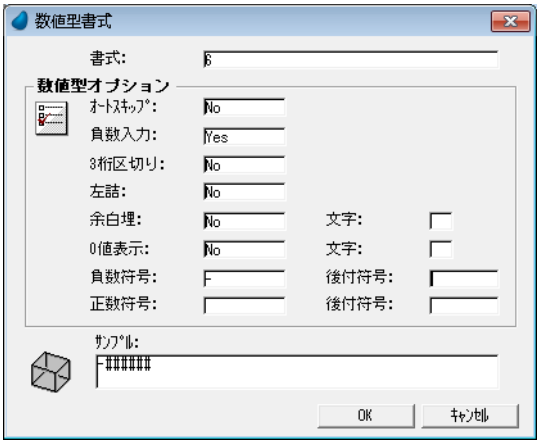
文字／ Unicode 型の書式設定

指示記号	説明
X	文字型の場合は任意の 1 バイト、Unicode 型の場合は任意の 1 文字を意味します。
U	入力した英文字を大文字に変換します。文字型の場合は、半角のみ変換されます。
L	入力した英文字を小文字に変換します。文字型の場合は、半角のみ変換されます。
#	１桁の数字（0-9）を意味します。この記号のおかれている位置に数字以外の文字が入力されると、エラーとなり受け付けません。
A	データ入力時に、書式で指定された文字数のデータが入力されたら、自動的に次の項目にカーソルを移動させます。



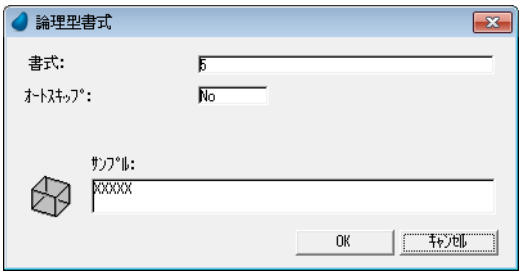
数値型の書式設定

指示記号	説明
#	１桁の数値を意味します。
.	小数点の挿入位置を指示します。
A	オートスキップ：データ入力時に、書式で指定された文字数のデータが入力されたら、自動的に次の項目にカーソルを移動させます。
N	負数入力：データ入力時に、負数が入力できるようになります。
C	3 桁区切：データを表示するとき、3 桁ごとにカンマで区切られるようになります。
L	左詰：データを表示するとき、左寄せで表示するようになります。（デフォルトは右寄せ）
Pc	余白埋、文字：データの数字が含まれていない部分に［文字］欄にて指定した文字が表示されるようになります。c は文字（例：0）です。
Zc	0 値表示、文字：データがゼロの場合、全ての桁に［文字］欄にて指定した文字が表示されるようになります。
-c{,S};	負数符号、後付符号：負の値の前に通常のマイナス記号（－）以外の文字列を表示する場合、［負数符号］欄に表示したい文字を指定します。負の値の末尾に文字列を表示する場合、［後付符号］欄に表示したい文字を指定します。
+c{,S};	正数符号、後付符号：正の値の前に文字列を表示する場合、［正数符号］欄に表示したい文字を指定します。正の値の末尾に文字列を表示する場合、［後付符号］欄に表示したい文字を指定します。



論理型の書式設定

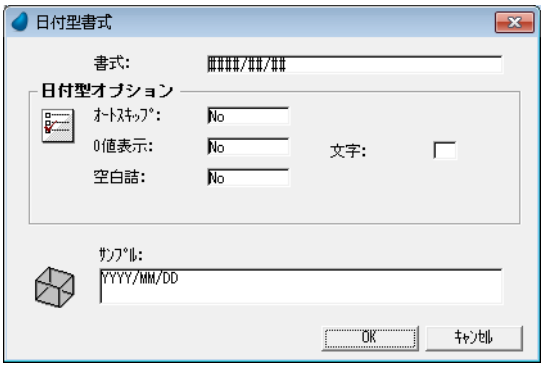
指示記号	説明
X	内部値を表示値または入力値に変換する際に使用される文字です。 1 は［True］を、0 は［False］に対応します。 ［範囲］に他の文字表現を設定することができ、たとえば［True］を＜○＞、［False］を＜×＞などとします。
A	オートスキップ：データ入力時に、書式で指定された文字数のデータが入力されたら、自動的に次の項目にカーソルを移動させます。



日付型の書式設定

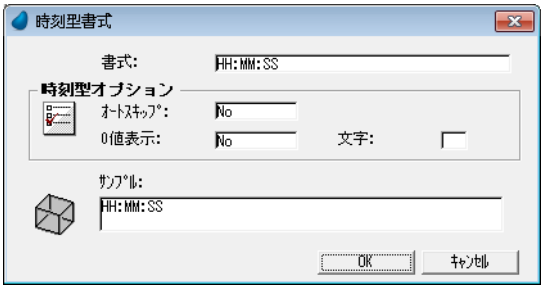
指示記号	説明
DD	月間の日付を示す数字；1-31
DDD	年間の日付を示す数字；1-366
DDDD	月間の日付の英語表記；1st, 2nd, 3rd, 4th, etc.
MM	月を示す数字；1-12
MMM...	月の英語表記（［M］は 3 ～ 10 桁使用可能であり、月の名前が［M］の数より少ない場合、残りは空白を表示）；January、February など
YY	年を示す数字の末尾 2 桁；0-99
YYYY	年を示す数字 4 桁；0-9999
W	週間の曜日を示す数字；1-7
WWW...	曜日の英語表記（［W］は 3 ～ 10 桁使用可能であり、曜日の名前が［W］の数より少ない場合、残りは空白を表示）；Sunday、Monday など
SS...	曜日の日本語表記；S が 2 つで曜日の日本語の表記 1 文字に相当します。（例えば、書式を［SS］とした場合、水曜日であれば、［水］と表記されます。）；日曜日など
J	元号の（簡易）英語表記；M……明治、T……大正、S……昭和、H……平成
JJ...	元号の日本語表記

指示記号	説明
/	日付区切文字の挿入位置を指定します。挿入される文字は［動作環境］ダイアログの［日付区切文字］欄にて指定します。
##/##/##	［動作環境］ダイアログの［国別設定］タブにある［日付タイプ］欄の設定により、日付を DD/MM/YY、MM/DD/YY、または YY/MM/DD の型式で表示します。
A	オートスキップ：データ入力時に、書式で指定された文字数のデータが入力されたら、自動的に次の項目にカーソルを移動させます。
Z	0 値表示、文字：データがゼロの場合、全ての桁に［文字］欄にて指定した文字が表示されるようになります。
T	空白詰：位置指示記号の［WWW...（曜日）］［MMM...（月）］［DDDD（日付の英語表記）］により挿入された空白を削除して表示するようになります。これらの位置指示記号は、使用可能な最大の桁数で書式文字列に指定されるため、指定した長さより短い名前には不要な空白が作成されることがあります。この指示記号は、これらの空白を全て削除します。



時刻型の書式設定

指示記号	説明
HH	時間；00-23
MM	分；00-59
SS	秒；00-59
:	時刻区切文字の挿入位置を指定します。挿入される文字は［動作環境］ダイアログの［時刻区切文字］欄にて指定します。
MMM...	月の英語表記（[M］は 3 ～ 10 桁使用可能であり、月の名前が［M］の数より少ない場合、残りは空白を表示）；January、February など
PM	am または pm を表示し、時間を 12 時間制（1 ～ 12）にて表示；am, pm
A	オートスキップ：データ入力時に、書式で指定された文字数のデータが入力されたら、自動的に次の項目にカーソルを移動させます。
Z	0 値表示、文字：データがゼロの場合、全ての桁に［文字］欄にて指定した文字が表示されるようになります。

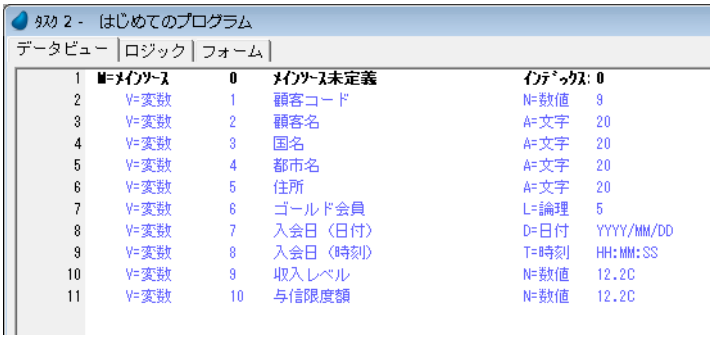


１３ .変数項目の追加

Magic xpa の書式について一通り学んだので、ここでプログラムに変数項目を追加しましょう。

＜はじめてのプログラム＞を開き、［データビュー］エディタで以下の項目番号 3 以後の変数項目を追加してください。

項目タイプ	項目番号	項目名	モデル	型	書式	範囲	終了	代入
V= 変数	1	顧客コード	0	N= 数値	9	0	0	0
V= 変数	2	顧客名	0	A= 文字	20	0	0	0
V= 変数	3	国名	0	A= 文字	20	0	0	0
V= 変数	4	都市名	0	A= 文字	20	0	0	0
V= 変数	5	住所	0	A= 文字	20	0	0	0
V= 変数	6	ゴールド会員	0	L= 論理	5	0	0	0
V= 変数	7	入会日（日付）	0	D= 日付	YYYY/MM/DD	0	0	0
V= 変数	8	入会日（時刻）	0	T= 時刻	HH:MM:SS	0	0	0
V= 変数	9	収入レベル	0	N= 数値	12.2C	0	0	0
V= 変数	10	与信限度額	0	N= 数値	12.2C	0	0	0



１４.［フォーム自動作成］ウィザード

ここまでの学習で、項目パレットから選択してフォームにコントロールを配置することを実習しました。

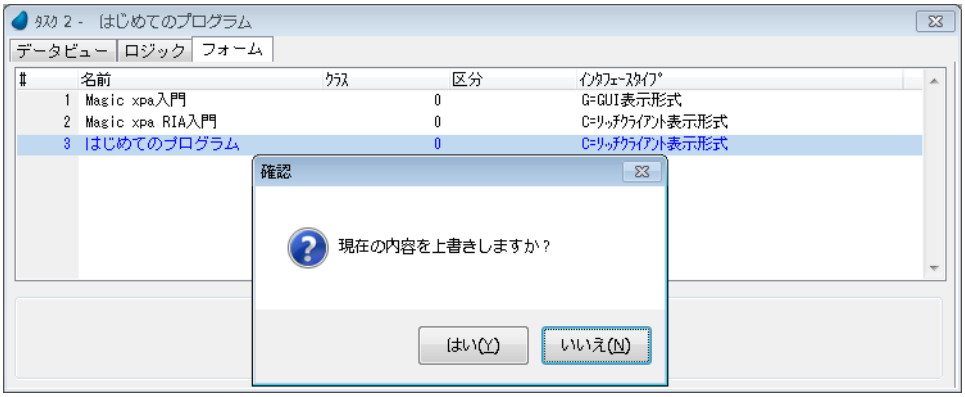
ここではウィザードを使用してコントロールを配置し、フォームを再作成してみましょう。

［フォーム自動作成］ウィザードは、指定されたすべての項目をフォーム上に自動的に配置します。

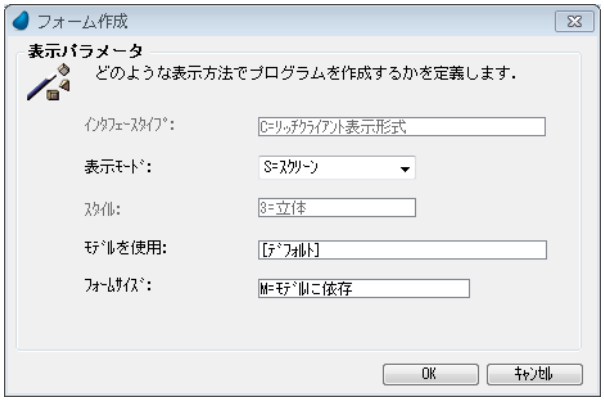
実際に＜はじめてのプログラム＞を使用して、データビューに定義したすべての変数項目を［フォーム自動作成］ウィザードを使用してフォームに配置してみましょう。

ウィザードの実行手順

1. ［プログラム］リポジトリを開き、<はじめてのプログラム>を開きます。
2. ［フォーム］タブを選択し、二行目の<はじめてのプログラム>フォームを選択します。
3. ［オプション］メニューから［フォーム自動作成］（Ctrl+G）を選択します。このとき、既存のフォームを上書きするかどうかを確認するダイアログが表示されます。



4. ここでは上書きするために［はい］をクリックします。［フォーム作成］ダイアログが表示されます。
5. ［表示モード］で［S= スクリーン］を選択します。
6. ［OK］ボタンをクリックして、ダイアログを閉じます。
7. ズーム（F5）してフォームを開きます。
8. 自動作成されたフォームを確認してみましょう。

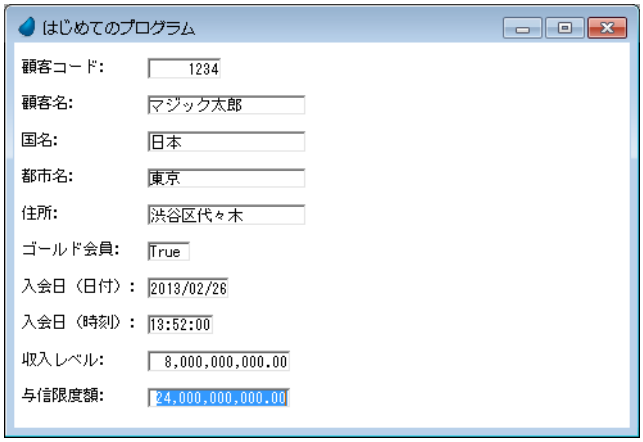


結果の保存と実行

変更内容を保存してプログラム編集を終了し、結果を確認するために、プログラムを実行してみましょう。

1. ［オプション］メニューから［オブジェクトを保存／閉じる］を選択します。
2. ［プログラム］リポジトリから<はじめてのプログラム>を実行します（F7）。Magic xpa の実行ウィンドウが開き、プログラムのメインフォームが表示されます。
3. 図に示すように、各項目に適当な値を入力してください。

このとき項目間のカーソル移動は、Tab キーを押下することで行われます。



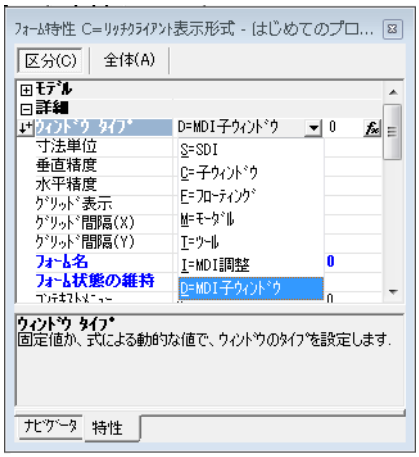
１５.ウィンドウタイプの指定

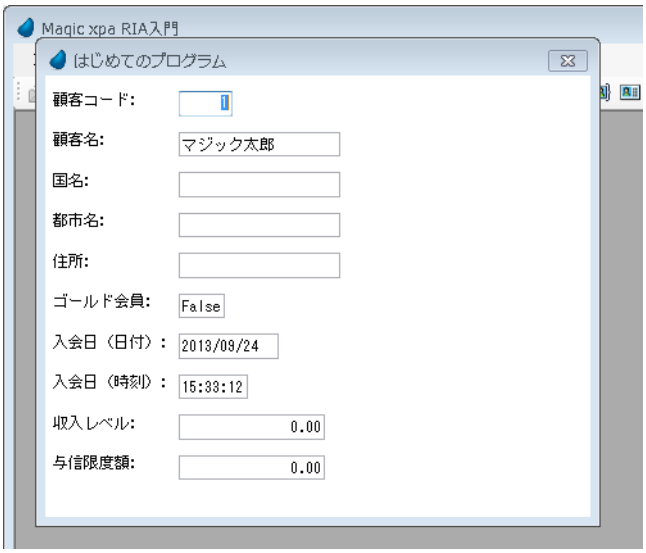
リッチクライアントタスクを実行した場合のウィンドウの表示方法は、［フォーム特性］の［ウィンドウタイプ］特性で決まります。ここでは、ウィンドウタイプ毎の表示の違いについて確認してみます。リッチクライアントフォームの［ウィンドウタイプ］特性には、以下のオプションがあります。

- S=SDI …… SDI ウィンドウという独立したコンテキストのウィンドウで表示されます。
- C= 子ウィンドウ …… 親タスクのウィンドウ内の範囲で表示されます。
- F= フローティング …… フローティングウィンドウで表示されます。（MDI フォームを含めた）親タスクのウィンドウの外に表示させることができます。
- M= モーダル（デフォルト）…… モーダルウィンドウで表示されます。別のウィンドウにフォーカスを移動させることができません。
- T= ツール …… ツールウィンドウで表示されます。ツールウィンドウは、フローティング表示でアイコンが表示されません。
- I=MDI 調整 …… MDI フォームに収まるようにウィンドウサイズが調整されて表示されます。
- D=MDI 子ウィンドウ …… MDI フォームにウィンドウが表示されますが、［フォーム特性］で定義されたサイズで表示されます。

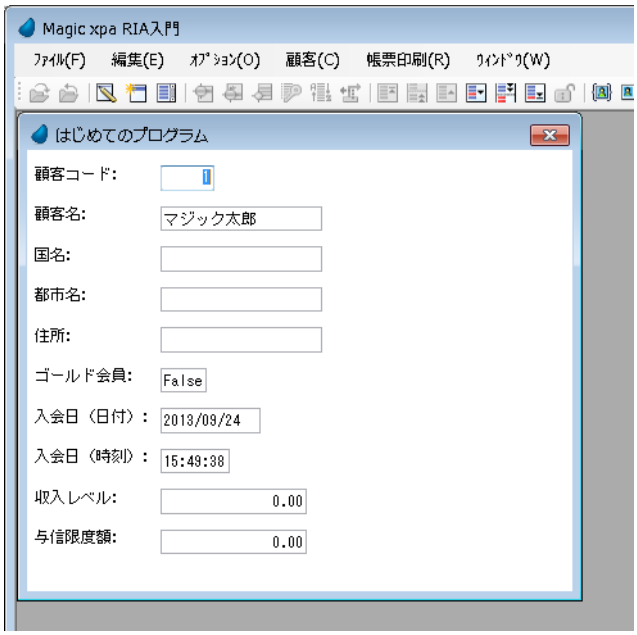
「モーダルウィンドウ」と「MDI 子ウィンドウ」の違いを確認してみましょう。

1. ［プログラム］リポジトリから＜はじめてのプログラム＞を開きます（F5）。
2. ［フォーム］エディタに切り替え（Ctrl+3）、メインフォームの［ウィンドウタイプ］特性を確認してください。デフォルトの「M= モーダル」になっているはずです。
3. Enter キーを押下して［プログラム］リポジトリに戻り、プログラムを実行します（F7）。プログラムのウィンドウが MDI フォームの外に移動することができます。
4. プログラムをを終了し、再度、＜はじめてのプログラム＞を開きます（F5）。
5. ［フォーム］エディタに切り替え（Ctrl+3）、メインフォームの［ウィンドウタイプ］特性を「D=MDI 子ウィンドウ」に変更します。
6. Enter キーを押下して［プログラム］リポジトリに戻り、プログラムを実行します（F7）。プログラムのウィンドウは MDI フォーム内で表示されるようになります。





モーダルウィンドウ



MDI 子ウィンドウ

次章からのプログラム実行例の図は、すべて「MDI 子ウィンドウ」で実行させたものです。タスクタイプの設定は任意に変更して、動作を確認してみてください。

１６.要約

本章では、Magic xpa で、はじめてのプロジェクトとプログラムを作成しました。

実習を通して次のようなことについて理解を深めました。

- プロジェクトの概念
- ［プログラム］リポジトリと、リポジトリへのプログラム新規作成方法
- プログラムの定義といくつかの特性の設定方法
- ローカル変数項目を用いたデータビューの定義方法
- Magic xpa で用いる［型］と［書式］の詳細
- プログラムの作成内容を保存し、実行させて結果を確認する方法
- ウィンドウタイプの定義と実行時の表示

第 4 章 データ操作

この章では、プログラムでのデータ操作方法について学び、数値の演算と文字列の取り扱いについて実習します。

キーワード

- 条件
- 数値データ
- 文字データ
- 連結文字列
- コントロール後処理

学習目標

本章を学ぶことで、次の内容に対する理解を深めましょう。

- 数値データ項目の演算を行う
- 文字列データに簡単な操作を行う
- 処理コマンドを条件的に実行する

参照

Magic ヘルプの［Magic xpa リファレンス／［式］エディタ／関数ディレクトリ／演算子］

同様に、［Magic xpa リファレンス／［式］エディタ／関数ディレクトリ／文字列操作関数］

１．はじめに

本章では、数値データと文字データの取り扱いについて学習します。

最初に Magic xpa で簡単な四則演算を行う演算子について学びます。

次に演算処理を条件的に行う方法に関連して、［IF］関数を学習します。

そのあと、文字列の連結や、余分なスペースを取り除くための関数などについて学習します。

２．数値データの操作

Magic xpa では基本的な四則演算以外にも多くの数値関数を使用することができます。

ここでは基本的な演算子をとりあげ、表にまとめました。

基本的演算子

演算子	説明
＋	加算演算子
-	減算演算子
*	乗算演算子
/	除算演算子　；　整数を除算した結果は、最も近い整数に丸められます。
^	指数演算子　；　A^B は、A を B 乗して戻します。
MOD	剰余演算子　；　整数を除算した時の余りを戻します。例：17 MOD 10 は、7 を戻します。

このセクションでは、データ項目に簡単な演算結果を反映させる方法を学びます。

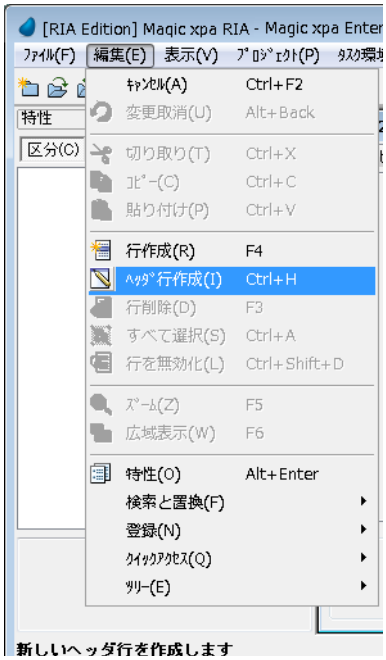
与信限度額の計算

以下の手順で、収入レベルに基づいて、与信限度額を計算する方法について説明します。

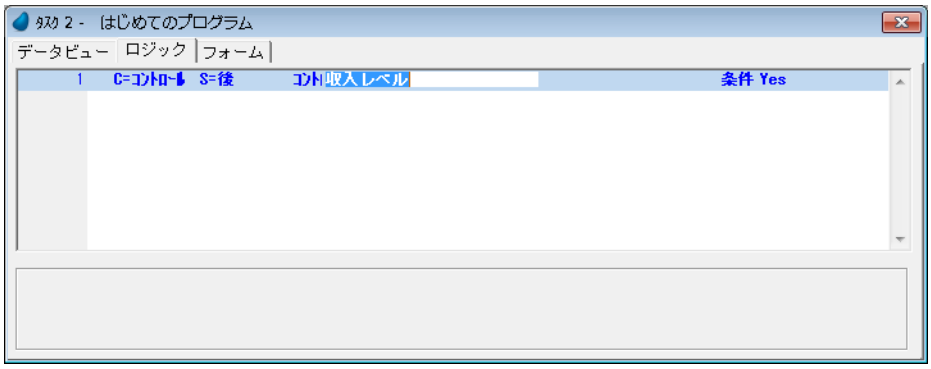
1. ＜はじめてのプログラム＞にズーム（F5）してプログラムを開きます。
2. ［ロジック］エディタを選択します。
3. ［編集］メニューから［ヘッダ行作成］（Ctrl+H）を選択します。

4. 作成されたヘッダ行のパラメータを次のように設定します。

ロジックユニットタイプ	イベント	コントロール名	条件
C= コントロール	S= 後	収入レベル	Yes



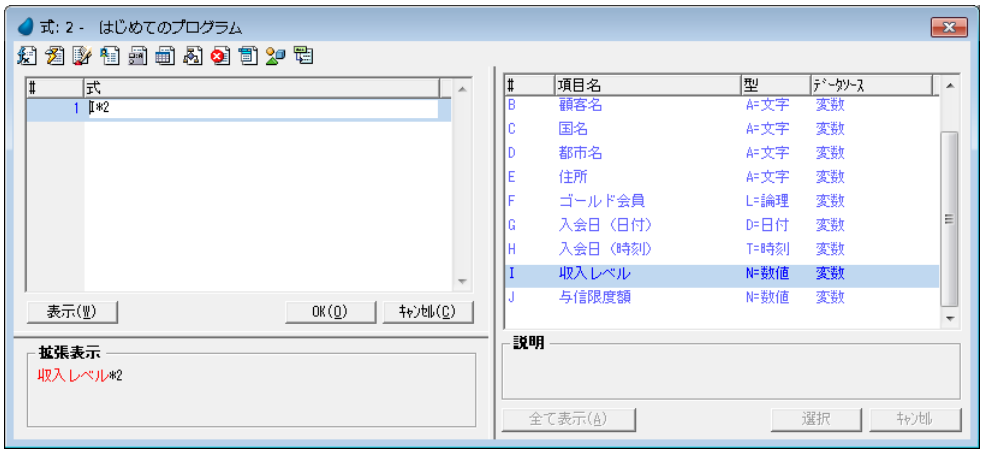
5. コントロール名は、カラムからズームして表示されるコントロールリストから＜収入レベル＞を選択します。



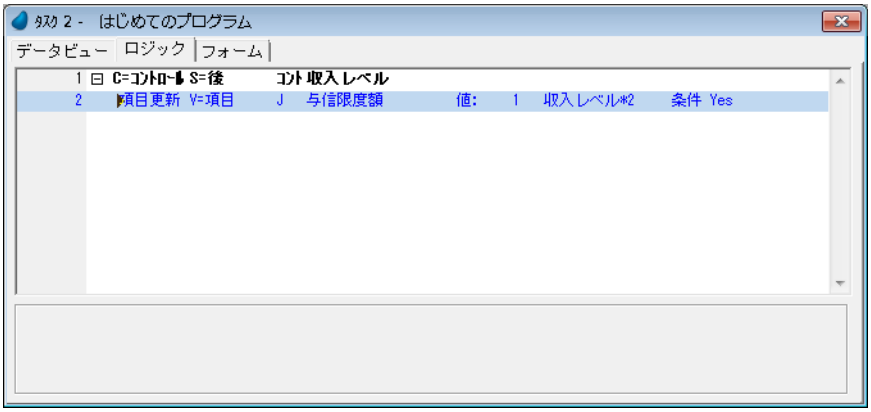
6. 作成したヘッダ行を選択した状態で、F4 キーを押下して、一行作成します。
7. 以下のようにパラメータを設定します。

コマンド	項目	値	条件
項目更新	J （与信限度額）	値： 1 収入レベル *2	Yes

8. ［値］カラムで F5 キーを押下してズームし、［式］エディタを開きます。
9. F4 キーを押下して一行作成します。
10. F5 キーでズームし、データ項目リストから＜収入レベル＞を選択します。
11. ［式］欄に戻り、＜ *2 ＞と追加入力します。ここで式表現は、＜ I*2 ＞となっており、［値］欄には、収入レベル *2 と表示されています。
12. ［OK］ボタンをクリックして［式］エディタを終了します。



［値：］欄には式の［値］と［式番号］が表示されます。



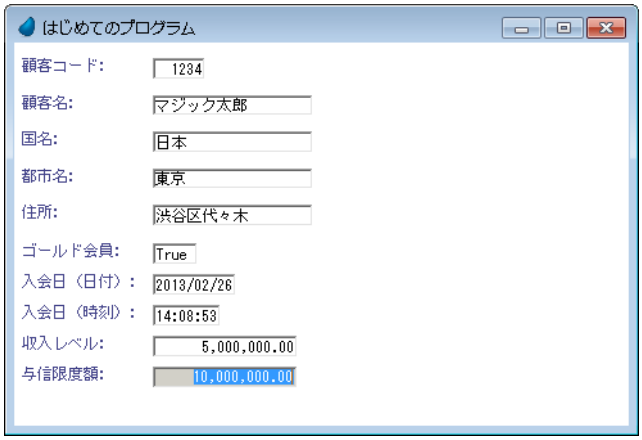
13. プログラムを保存して閉じてください。

３．プログラムの実行

プログラムを実行させ、実行結果を確認してみましょう。

1. F7 キーを押下して＜はじめてのプログラム＞を実行させます。
2. Tab キーを押下して、カーソルを＜収入レベル＞に移動させます。（ここでは途中の項目には必ずしも値を入れる必要はありません。）

3. ＜収入レベル＞に、＜ 5000000 ＞と入力し、Tab キーを押下して次の項目に移動します。
4. ＜与信限度額＞に自動的に、＜ 10000000 ＞が入力されたことを確認してください。



＜収入レベル＞に値を入力した後、他の項目に移動すると、＜与信限度額＞は自動的に＜収入レベル＞ *2 の値に更新されることが分かります。

結果の説明

この例では、［コントロール後］のロジックユニットが＜収入レベル＞の［エディット］コントロールに定義されました。［コントロール後］ロジックユニットは、エンドユーザのオペレーションによってカーソルがそのコントロールから他に移動するときに実行されます。（典型的なオペレーションとしては、Tab キーを押下することで次のコントロールに移るとき。）

ロジックユニットの内部では、今回の例では［項目更新］処理コマンドが使われており、項目の値が式によって定義された値によって更新されました。

この例では、算術演算子（*）を用いて＜収入レベル＞を 2 倍にするように式を定義していました。

同じようにして、すでに説明した他の算術演算子を使用することができます。

４．計算の条件的実行

Magic xpa ではコマンドの実行条件を設定することができます。実行条件は True または False を返す論理式で定義されます。

ここで、先ほどの［項目更新］処理コマンドに条件を設定し、＜収入レベル＞の値が＜与信限度額＞よりも大きい場合のみ＜収入レベル＞ *2 の値で＜与信限度額＞を更新するようにしてみましょう。

1. ＜はじめてのプログラム＞にズームします。
2. ［ロジック］エディタを選択します。
3. ［項目更新］処理コマンドの行を選択します。

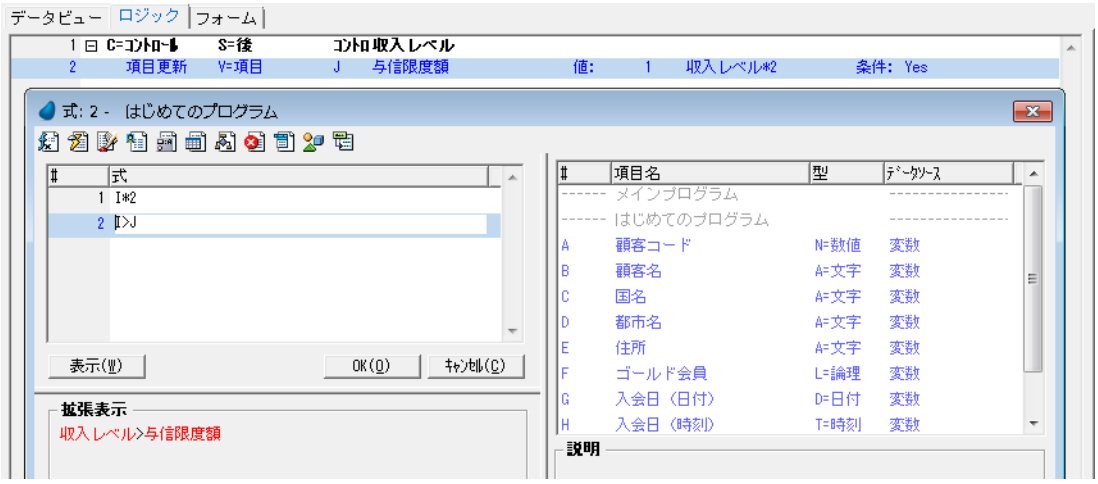
4. 右端の［条件：］欄でズームし、［式］エディタを開きます。
5. F4 キーを押下して新しく一行作成します。
6. 次の式を入力します：＜ I>J ＞

ここで、I は＜収入レベル＞、J は＜与信限度額＞の各項目です。

［式］エディタからズームして項目リストに移動し、そこから選択することもできます。

7. ［OK］ボタンをクリックして［式］エディタを終了します。

8. オブジェクトを保存し、プログラムを閉じます。

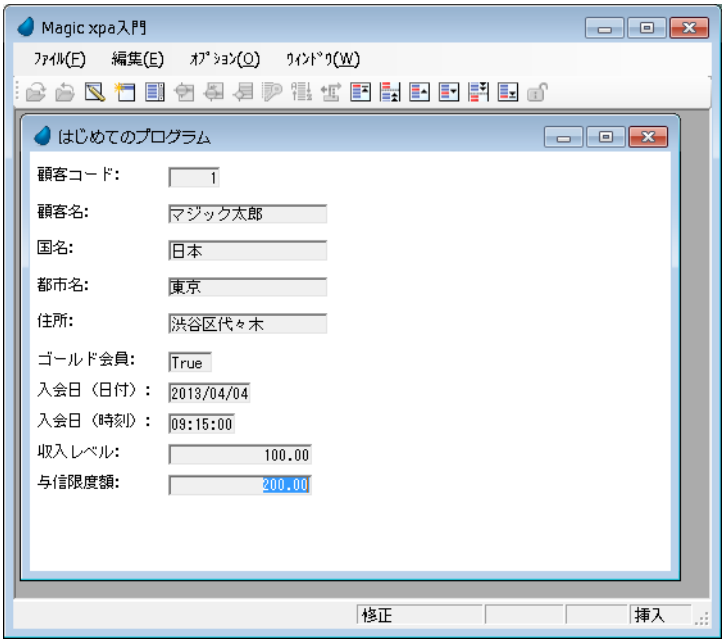


条件付きプログラムの実行 1

1. ＜はじめてのプログラム＞を実行します。
2. ＜収入レベル＞に＜ 100 ＞と入力しましょう。
3. Tab キーを押下してください。

実行結果として、この場合は条件が True と評価されるため、＜与信限度額＞に＜ 200 ＞が設定されます。

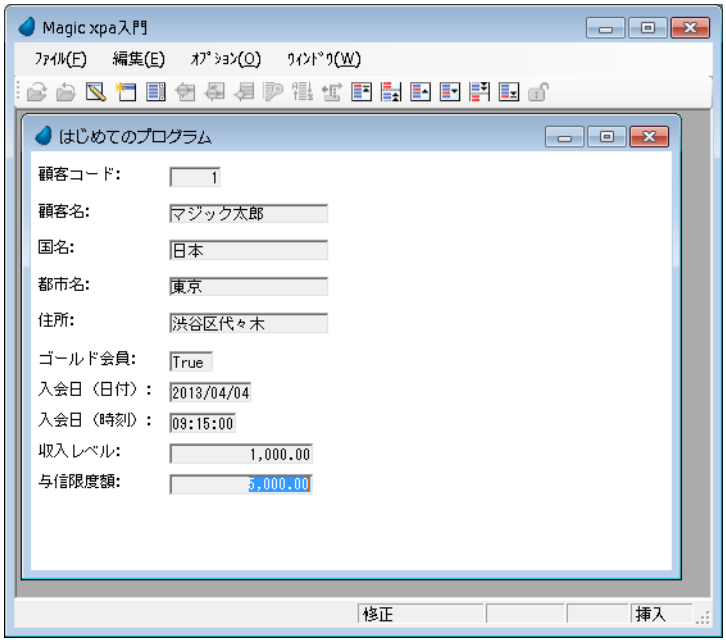
つまり、この場合は＜収入レベル＞（100）が＜与信限度額＞（0）よりも大きいので条件は True です。



つぎに条件を変更して実行してみましょう。

4. ＜与信限度額＞に＜ 5000 ＞を入力してください。
5. ＜収入レベル＞に＜ 1000 ＞を入力し、Tab キーを押下してください。

この場合は、＜収入レベル＞（1000）が＜与信限度額＞（5000）よりも小さいため、条件式は False となり、項目更新は実行されません。



今までのところで条件的な処理コマンドの実行について学びました。実際にロジックを組むときには、処理コマンドの実行だけでなく、戻り値を条件によって切りかえて利用する場合があります。

ここでは条件を適用して結果を得る場合について、［IF］関数を使用する方法を学習します。

５．IF 関数の利用（ゴールド会員の例）

次に示す例では［IF］関数を利用して、乗算の方法を切り替え、ゴールド会員の場合には通常よりも与信限度額を高く（通常は 2 倍、ゴールド会員の場合は 3 倍）設定しています。

1. ＜はじめてのプログラム＞にズームします。
2. ［ロジック］エディタを選択します。
3. ［項目更新］処理コマンドの行（2 行目）を選択します。

4. ［値：］欄でズームし、［式］エディタを開きます。
5. 式の内容を次のように変更します：＜ I*2 ＞　→　＜ IF(F, I*3, I*2) ＞

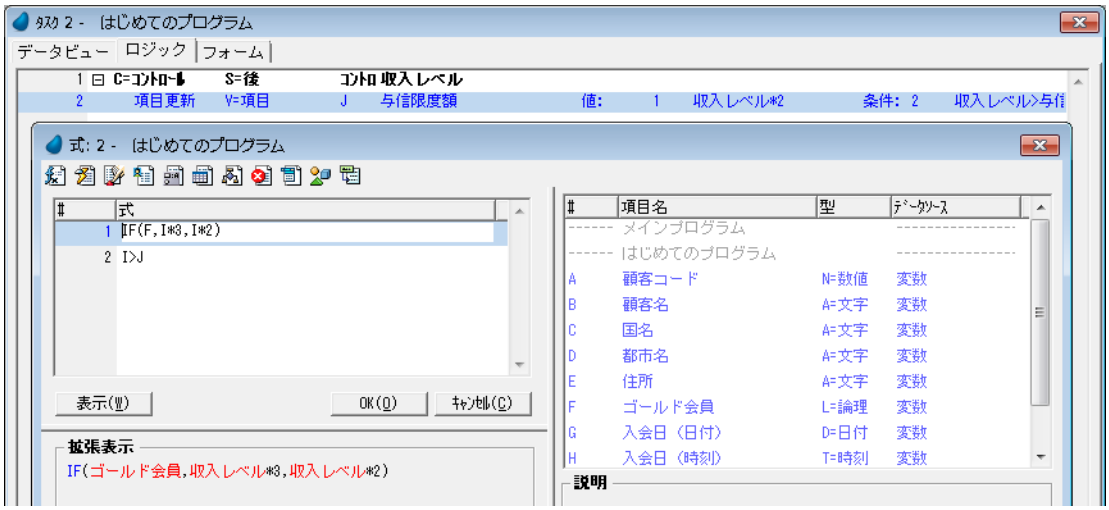
ここで、

F は＜ゴールド会員＞ I は＜収入レベル＞の各項目です。

式からズームして項目リストに移動し、そこから項目を選択することもできます。

6. ［OK］ボタンをクリックして［式］エディタを閉じます。

7. オブジェクトを保存し、プログラムを閉じます。



IF 関数

条件を指定し、その条件が評価された結果の論理値（[True] または [False]）に応じた戻り値を返します。

構文：	IF(論理式 , 値 1, 値 2)
パラメータ：	論理式 …論理値が返される式、または論理型項目
	値 1 …… True が返された時の戻り値として使用される値
	値 2 …… False が返された時の戻り値として使用される値
戻り値：	値 1（True のとき）または値 2（False のとき）
例：	IF(A>100,10,5) では、A=101 の場合、10 が戻り値として返ります。A=99 なら、5 が返ります。

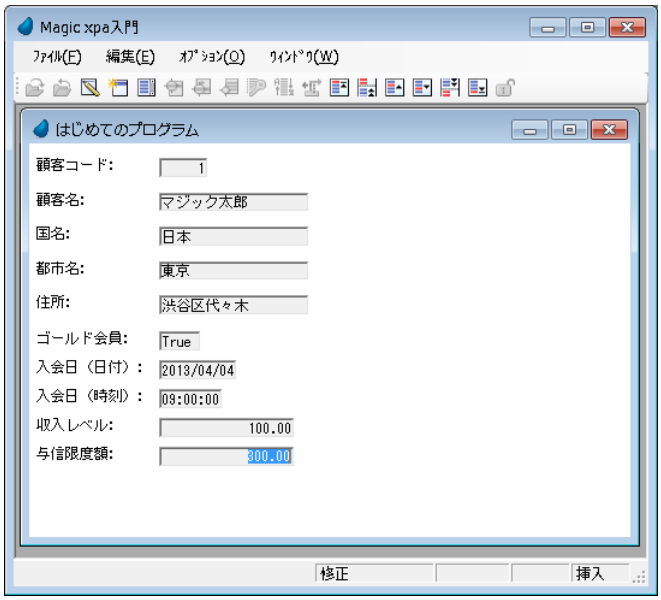
６．IF 関数を使用したプログラムの実行

1. F7 キーを押下して、<はじめてのプログラム>を実行します。
2. ゴールド会員の欄に［True］と入力してください。

True のときは＜ T ＞、False のときは＜ F ＞のみを入力すれば、Magic xpa は自動的に True/False と補って入力します。

3. ＜収入レベレ＞に＜ 100 ＞を入力し、Tab キーを押下してください。

＜与信限度額＞に 3 倍の＜ 300 ＞が自動的に設定されました。



4. 次にゴールド会員欄を＜ False ＞にします。
5. ＜収入レベル＞に＜ 2000 ＞を入力し、Tab キーを押下します。

今度は＜与信限度額＞が 2 倍の＜ 4000 ＞になりました。

［項目更新］処理コマンドの実行条件式（収入レベル＞与信限度額）があることを忘れないでください。上記のコマンドを実行するとき、この条件が満たされていることに留意しましょう。

７．文字列データの取り扱い

これまでのところで数値に関する取り扱いを学びました。ここからは、Magic xpa の文字列データの取り扱いについて学びましょう。

このセクションではまず複数の文字列を連結させるための演算子について学びます。

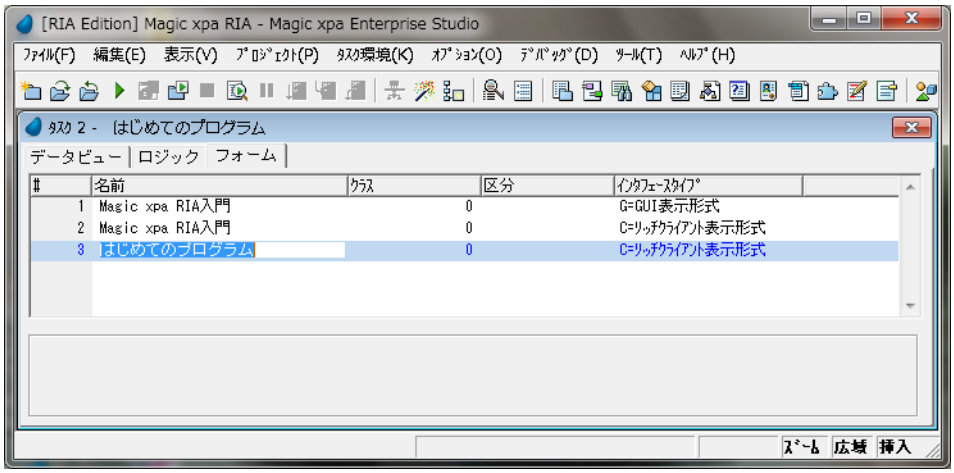
Magic xpa では、&（アンパーサンド）が文字列連結演算子として使用されます。

このセクションでは、この & の使い方と、さらに Trim 関数について学びます。その他の文字列関数については、本コースの後半で学びます。

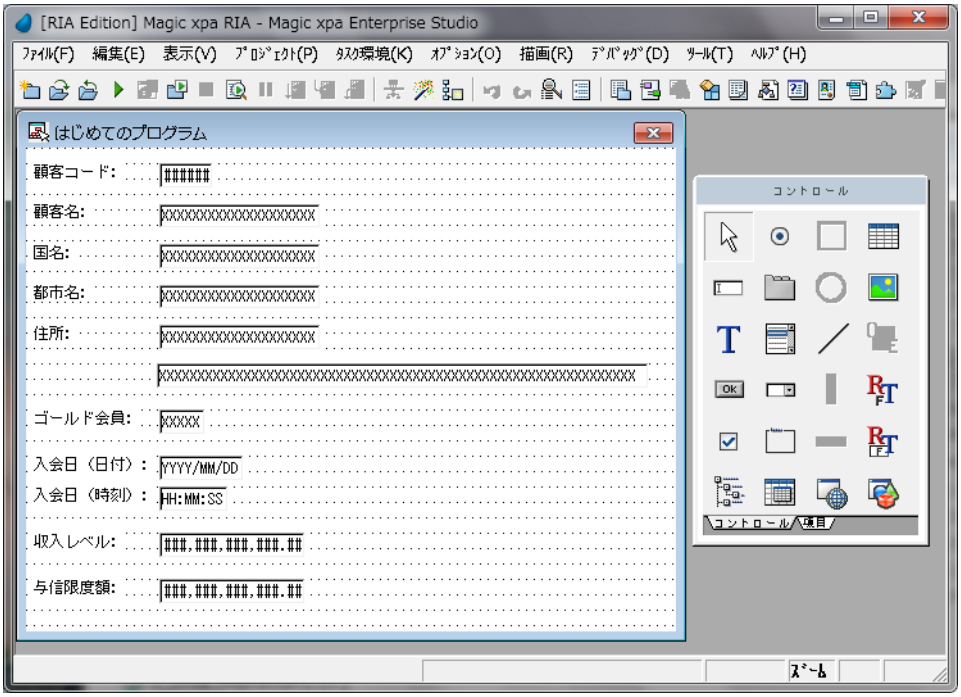
顧客住所の例

次の例で、顧客の＜国名＞、＜都市名＞、＜住所＞などの文字データ項目の連結を行ってみましょう。

1. ＜はじめてのプログラム＞でズームします。
2. ［フォーム］エディタを選択します。
3. ＜はじめてのプログラム＞フォームにパークし、ズームします。



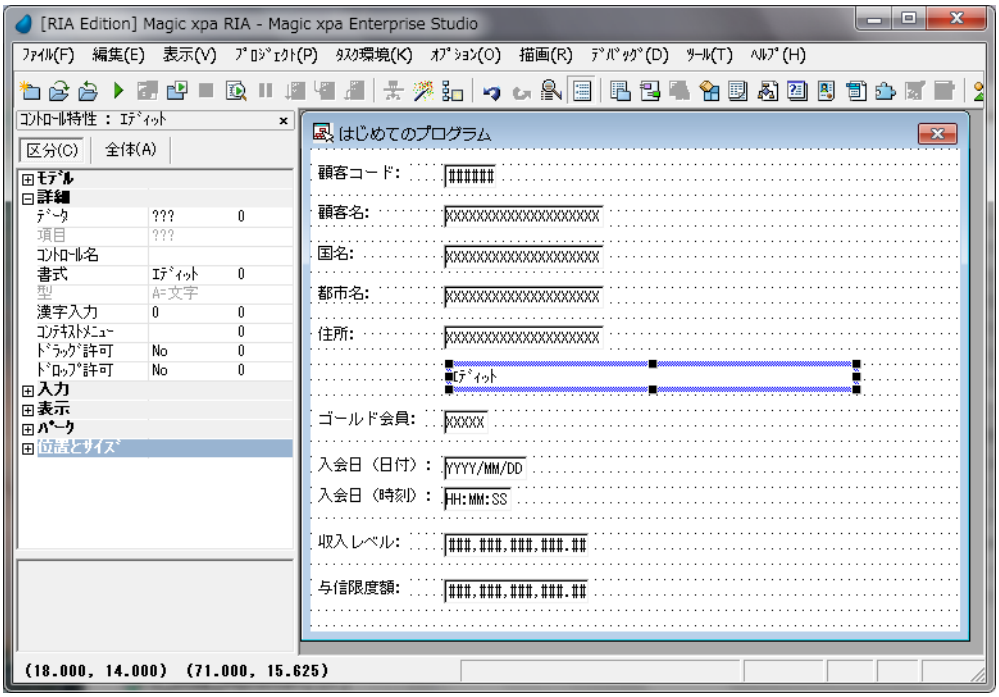
フォーム上に連結した住所データを表示するためのコントロールを追加します。



4. ［コントロール］パレットで［エディット］コントロールのアイコンを選択し、前ページのイメージに示すようにフォーム上に配置します。
5. 新しいコントロールが選択された状態で Alt+Enter キーを押下して、コントロール特性を表示させます。
6. ［データ］特性の［式］エディタ［fx］ボタンをクリックし、このコントロールに式を割り付けます。

［エディット］コントロールは［データ項目］または［式］に割り付けることができます。

前のセクションでは、［データ項目］に割り付けましたが、ここでは［式］に割り付けます。



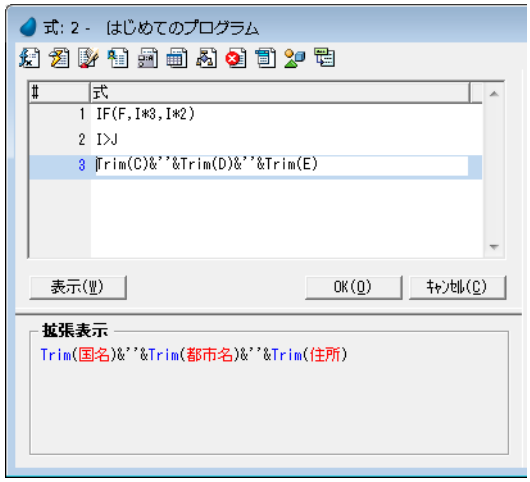
［式］エディタが開いたら次の操作を行ってください。

7. F4 キーを押下して一行作成し、次の式を入力してください。

   ＜ Trim(C)&' '&Trim(D)&' '&Trim(E) ＞

   ここで、C：国名、D：都市名、E：住所　の各項目です。

8. ［OK］ボタンをクリックして、この式をデータの特性として選択します。



Trim 関数

文字列の左右にある空白を削除します。

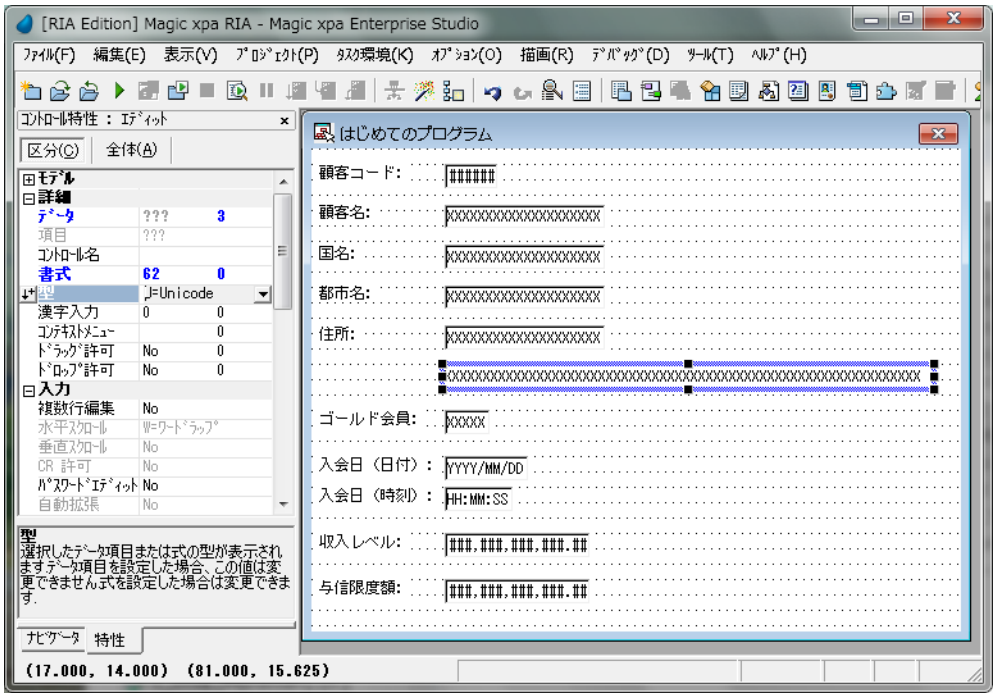
構文：	Trim(文字列)
パラメータ：	文字列 …処理対象となる文字列
	項目のシンボル名または文字列を '（アポストロフィ）で囲った表記で指示します。
戻り値：	文字列 …空白削除後の文字列
例：	Trim('　日本　')　　前後の空白が取り除かれ、＜日本＞が返ります。

フォームに追加した［エディット］コントロールには 3 つのデータ項目の情報が含まれています。

各項目の書式は 20 に定義されていたので、このデータ項目の書式としては、これらの総和と 2 つの半角スペースを追加して、少なくとも 62 以上にする必要があります。

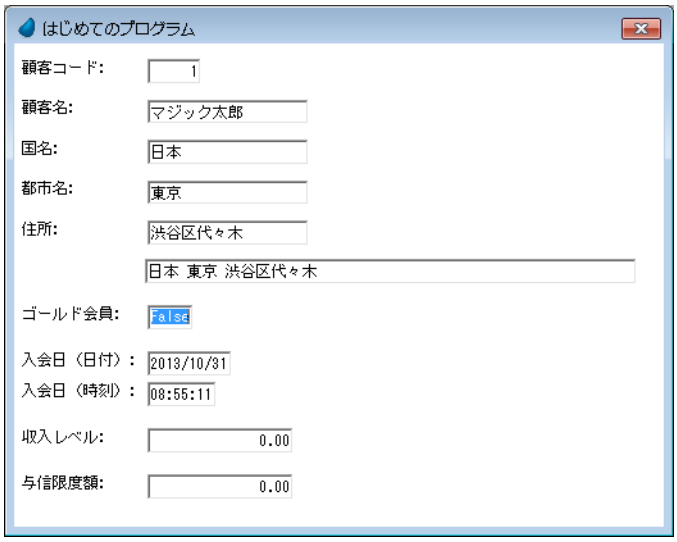
9. ［書式］特性に、＜ 62 ＞ を設定します。

10. プログラムを閉じて変更内容を保存します。



８．文字列関数を使用したプログラムの実行

1. F7 キーを押下して、＜はじめてのプログラム＞を実行させてください。
2. イメージに示すようにデータを入力してみましょう。
   さきほど追加した［エディット］コントロールに、文字列が連結されて表示されることが分かります。



９．練習問題

- 日本の顧客利益向上のために、日本の各顧客の与信限度額を通常の額に対して上乗せすることを決定したとします。上乗せ額は各顧客の収入レベルの 10% とします。
  - この計算は、エンドユーザが収入レベルを入力し終えて次の項目に移動するタイミングで行います。(本章ですでに作成した方法です。)
  - ＜与信限度額＞を項目更新する式を次のように変更します。
    - If 関数を使用して、顧客の国名が日本かどうかを判定します。

      ＜（Trim(C)=’日本’、C は国名の項目シンボル名）＞
    - 与信限度額の計算式で、既定義の内容に次を追加する：＜+IF(Trim(C)=’日本’,0.1*I,0)＞

（ここで、I は＜収入レベル＞の項目シンボル名）

- このプロジェクトをよりフレンドリーにするために、次のようなメッセージをフォームに追加しましょう。
  - 顧客名の［エディット］コントロールの右側に、［エディット］コントロールを一つ追加します。
  - この［エディット］コントロールに式を割り付け、**＜ようこそ＞**という文字列と＜顧客名＞を連結させます。
  - またこのとき＜ようこそ＞と＜顧客名＞の間には半角スペースを一つ入れてください。

１０.要約

本章では、Magic xpa でよく使用する操作方法を実習しました。たとえばプログラムへのズーム、タスクエディタの切り替え、［式］エディタの使用などです。

本章の主目的は、数値や文字列のデータを取扱うための基本的な処理コマンドや関数を使用し、具体的なプログラミングのために理解を深めることでした。

本章の前半では、数値の基本的演算子と、計算式での使用法を学びました。そして、［If］関数の使用法について実習しました。

後半では、文字列を取扱うための方法を学び、文字列の結合を行なう場合や、文字列に含まれる余分な空白を取り除くための［Trim］関数について実習しました。

第 5 章 データチェック

本章では内部のデータチェックの仕組みについて学習します。

またエンドユーザのデータ入力時の妥当性チェックとデータ整合性維持の方法について学びます。

キーワード

- データ妥当性
- データ整合性
- 型
- 書式
- ［エラー］処理コマンド
- ［項目変更］ロジックユニット
- 項目更新
- パーク

学習目標

本章を学ぶことで、次の内容に対する理解を深めましょう。

- Magic xpa 内部で行われるデータ妥当性チェック
- ［エラー］処理コマンドの使用方法
- エンドユーザのデータ入力の検証方法
- ［項目変更］ロジックユニットの使用方法
- ［パーク許可］特性
- ［パーク許可］特性の条件的使用方法

参照

Magic ヘルプの［Magic xpa リファレンス／［ロジック］エディタ／処理コマンド］

１．はじめに

オンラインプログラムでは、エンドユーザの入力したデータが妥当なものかどうかを検証する仕組みが必要です。Magic xpa では二通りの方法をサポートしています。

- Magic xpa に組み込まれているロジックで妥当性検証を行う方法
- アプリケーション開発者の作成するプログラムロジックによって妥当性検証を行う方法

この章では Magic xpa に組み込まれている検証の仕組みについて学び、その利点を考察します。

また、アプリケーション開発者がどのようなロジックを作成すればユーザデータの妥当性を検証し、データの整合性を維持することができるかについて学びます。

２．メニューから Magic xpa 内部に組み込まれたデータ妥当性検証機能

第 3 章ですでに Magic xpa に組み込まれたデータ検証の仕組みについて、2 つの例をあげて説明しました。

型によるデータ検証

たとえば、エンドユーザは数値型のデータ項目に対し、文字データを入力することはできませんでした。

Magic xpa ではこのように項目の型（上記の場合は数値型）を定義することにより、それに該当しないデータの入力を防いでいます。

書式によるデータ検証

データ項目の［書式］は、さらに詳細にデータの型とフォーマットを指定することで、この機能をさらに強力なものとしています。

たとえば、文字型のデータ項目において書式の定義を組合わせることで、通常の文字と数値文字からなる組み合わせ文字列を厳密に指定できます。

つまり、［書式］特性を使用することで、データ項目内で通常の文字と、数値を表す文字の正確な位置を指定できるのです。

同様に、日付を表すデータでは、［日］、［月］、［年］を意味する文字列位置を厳密に指定し、その定義に違反するようなデータ入力をはじくことができます。

次ページにいろいろな書式の定義例を示します。

データ項目の書式定義（例）

型	書式	説明
数値	5	5 桁の数値（0 から +99999 まで）。 この書式ではプラスの数値のみ入力できます。
数値	N5	5 桁の数値（-99999 から +99999）。 N はマイナスの値を持つことを許可します。
文字	10	10 バイトの任意の文字列。 漢字などの 2 バイトコードでは 5 文字になります。
文字	##XXXX	6 バイトの文字列。 最初の 2 バイトは数字で、残りの 4 バイトは任意の文字になります。 # は数字であることを意味しています。
文字	###-####	郵便番号などを定義するとき、3 桁の数字と、セパレータとしての '-'（固定表示）、4 桁の数字として定義します。
日付	YYYY/MM/DD	日付データで、最初の 4 桁が年、次の 2 桁が月、残りの 2 桁が日を意味するように定義した例です。 YYYY には 4 桁で年を入力します。 MM には 01 から 12 までの数字が入ります。 DD は 01 以上で、その月の最終の日付けまでを入力できます。

型	書式	説明
日付	YY/MM/DD	この例では、最初に 2 桁で年を指定し、次に月と日をそれぞれ 2 桁で入力します。 この場合、YY には 00 から 99 が入力できます。 Magic xpa は 2 桁の年データを 4 桁に変換する仕組みを持っています。
論理	5	5 桁の文字列で、この場合は True か False のいずれか一つです。
時刻	HH:MM:SS	時刻データ項目の書式です。 HH は時間を意味し、00 から 24 までが入力できます。 MM は分を意味し、00 から 59 までを入力できます。 SS は秒を意味し、00 から 59 までを入力できます。

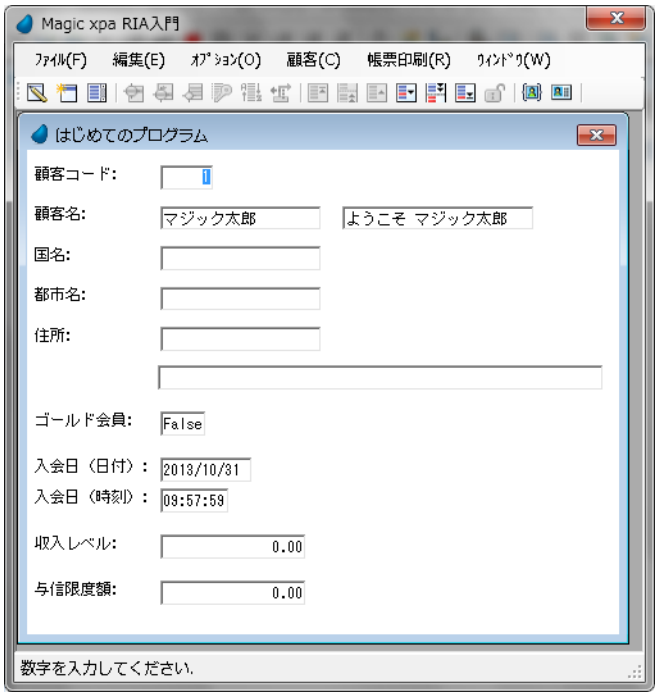
３．データ妥当性チェック機能

このセクションでは、Magic xpa に組み込まれたデータ妥当性チェック機能の例について、いくつか確認してみましょう。

数値型の妥当性チェック

1. F7 キーを押下して＜はじめてのプログラム＞を実行します。
2. ＜顧客コード＞欄で、例えば＜ a ＞のような文字を入力してください。

Magic xpa は数値型項目には文字データの入力を許可しません。上記の操作をすると、ステータス行には［数字を入力してください.］というエラーメッセージが表示されます。

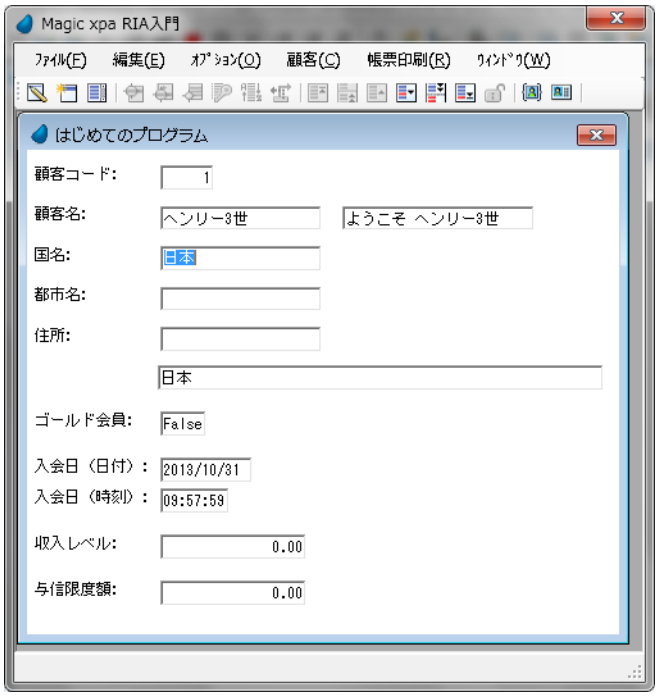


文字型の妥当性チェック

＜顧客名＞欄に＜ヘンリー 3 世＞と入力してみましょう。

Magic xpa では、文字型データには任意の文字を入力できます。

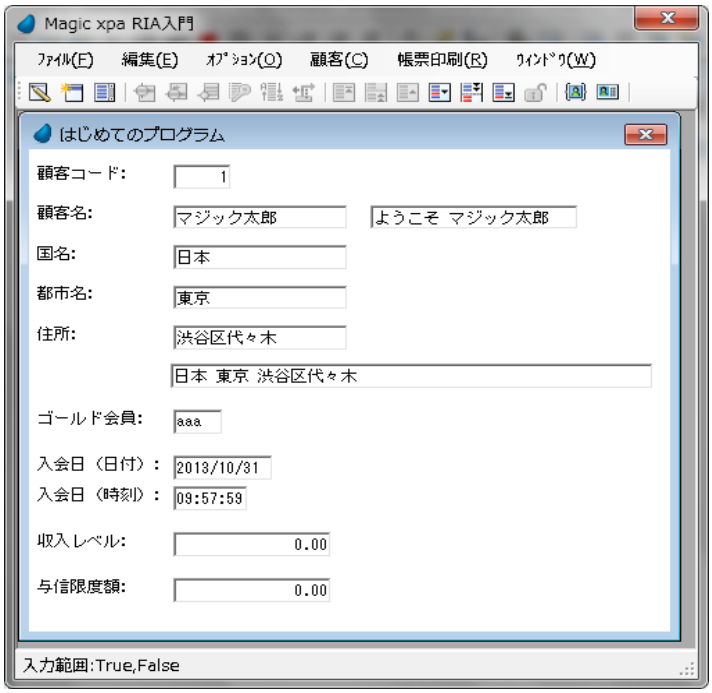
データ項目の書式では、データの長さも定義します。この場合、顧客名には最大 20 バイトまでの文字を入力でき、それ以上は受け付けません。



論理型の妥当性チェック

1. ＜ゴールド会員＞欄に＜ aaa ＞と入力してみましょう。
2. Tab キーを押下します。

Magic xpa は論理型項目には True または False の入力のみ受け付けます。それ以外の入力をしようとした場合、ステータス行には、［入力範囲：True,False］というエラーメッセージが表示され、別の入力項目には移動できなくなります。



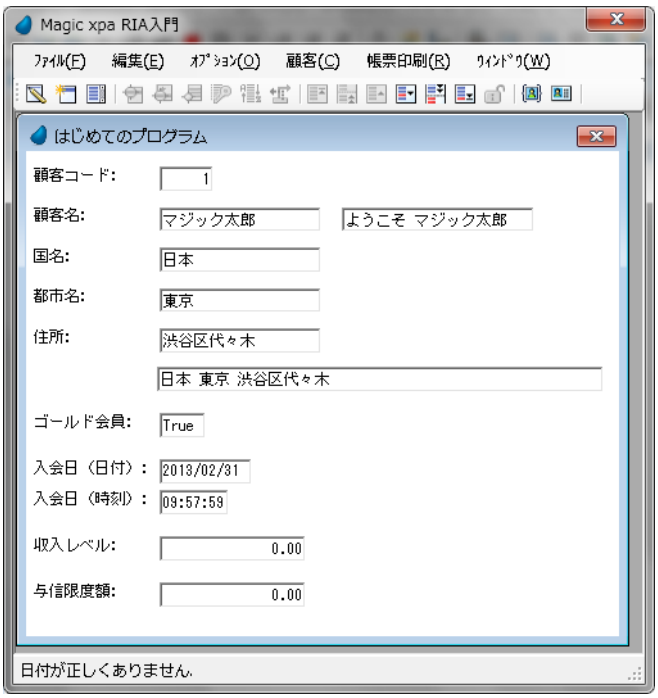
日付型の妥当性チェック

下記の操作をする前に、＜はじめてのプログラム＞を開き、［データビュー］エディタで、 ＜入会日（日付）＞項目の特性を開き、書式を YYYY/MM/DD にしておきましょう。

1. ＜はじめてのプログラム＞を実行します。
2. ＜入会日（日付）＞欄に、＜ 2013/02/31 ＞と入力してみましょう。
3. Tab キーを押下します。

Magic xpa はこの入力を受け付けません。

書式定義が、YYYY/MM/DD となっていて、MM の位置には 01 から 12 の範囲の値しか入力することができないにもかかわらず、この例では 31 を入力したため、［日付が正しくありません.］というエラーメッセージがステータス行に表示されます。

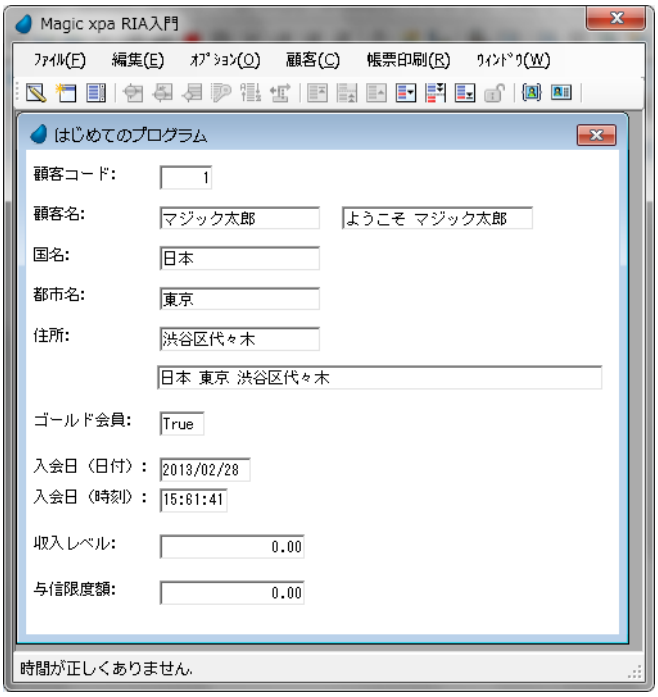


時刻型の妥当性チェック

1. ＜入会日（時刻）＞欄に、＜ 15:61:41 ＞と入力してみましょう。
2. Tab キーを押下します。

Magic xpa はこのような入力を受け付けません。

書式定義が HH:MM:SS になっていて、MM の位置には 00 から 59 までの値しか入力することができないにもかかわらず、この例では 61 を入力したため、［時間が正しくありません.］というエラーメッセージがステータス行に表示されます。



４．妥当性検証ロジックの開発

ここまでは、Magic xpa に組み込まれた妥当性チェックの仕組みについて学んできました。

このセクションでは、エンドユーザのデータ入力チェックをプログラムで実現するための開発手法について学習します。

日付データチェックの例

この例では、エンドユーザは現在日付よりも未来の日付入力をできないようにします。

1. ＜はじめてのプログラム＞にズームします。
2. ［ロジック］エディタを選択します。
3. 2 行目にパークします。

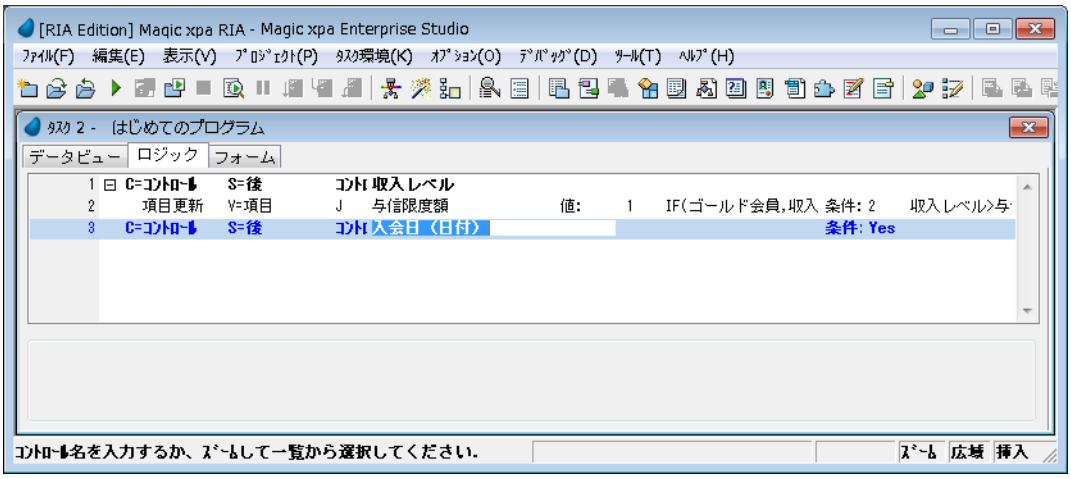
4. ［編集］メニューから［ヘッダ行作成］（Ctrl+H）を選択します。
5. ヘッダ行に次のようにパラメータを設定します。

ロジックユニットタイプ	イベント	コントロール名	条件
C= コントロール	S= 後	入会日（日付）	Yes

コントロールの選択は次のように行います。

6. ［コントロール］欄でズームし、コントロール一覧を表示させます。
7. コントロール名＜入会日（日付）＞にパークします。

8. ［選択］ボタンをクリックします。



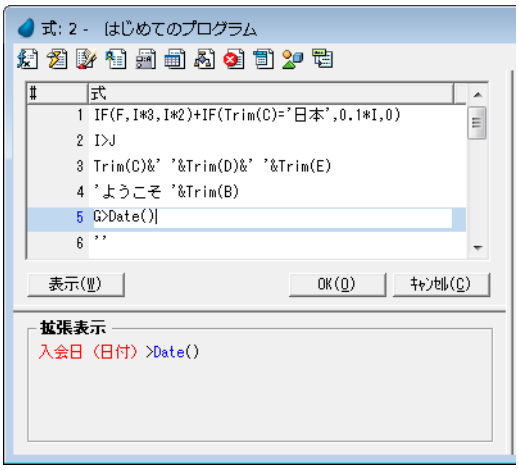
9. 作成したヘッダ行にパークし、［編集］メニューから［行作成］を選択するか、F4 キーを押下して一行作成します。
10. 次のパラメータ設定を行います。

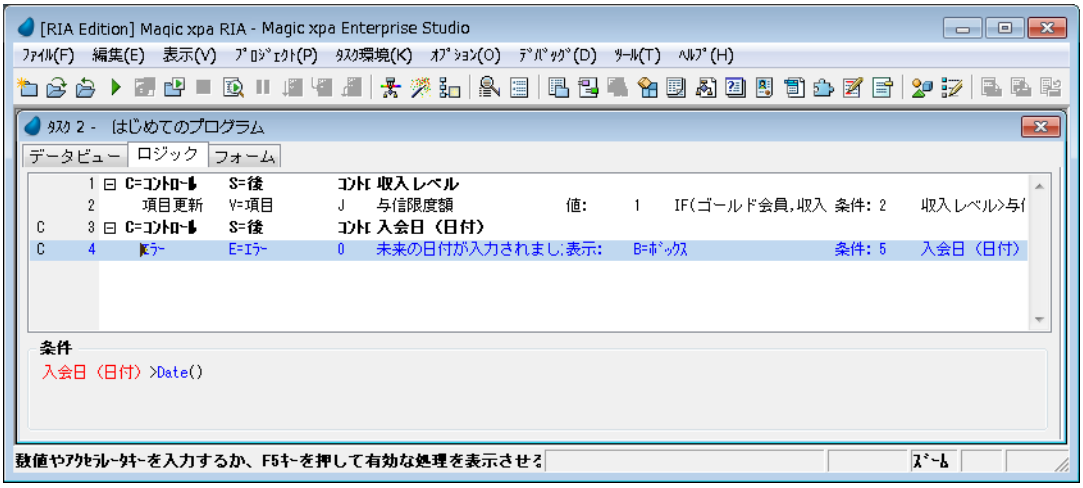
コマンド	モード	式	テキスト	表示	条件
エラー	E= エラー	0	未来の日付が入力されました	B= ボックス	5

条件欄の式の設定は次の手順で行います。

11. ［条件］欄でズームして［式］エディタを開きます。
12. 最後の行にパークします。
13. F4 キーを押下して一行作成します。
14. F5 キーを押下して、項目リストに移動します。
15. ＜入会日（日付）＞にパークし、［選択］ボタンをクリックします。（G が選択されます）
16. 式を次のように定義します：＜ G>Date() ＞
17. ［OK］ボタンをクリックして式定義を終了します。

18. オブジェクトを保存し、プログラムを閉じます。





［ロジック］エディタの行カラムに表示されている「C」は、クライアント側で実行されることを意味しています。

Date() 関数

システム日付が返ります。

構文：　Date()
パラメータ：　なし
戻り値：　日付
例：　システム日付が 07/03/01 の場合、Date() では 07/03/01 が返ります。

［エラー］処理コマンド

［エラー］処理コマンドは条件が［True］と評価されたときに実行され、エラーまたは警告のメッセージを表示します。メッセージはウィンドウのステータス行、またはダイアログボックスとして表示させることができます。

［エラー］処理コマンドの詳細

特性	説明
モード	• E= エラー……ビープ音が鳴ると同時に指定したメッセージが表示され、タスクの実行が中断します。その後次の規則にしたがって動作が実行されます。 ［エラー］処理コマンドがオンラインタスクのコントロールレベルで使用されている場合 1. 処理が一時中断します。 2. カーソルが対応するコントロールに止まります。 3. エンドユーザがエラーを訂正するまで待機状態に入ります。 • W= 警告……ビープ音が鳴ると同時に指定したメッセージが表示されますが、タスクの実行は停止しません。メッセージをダイアログボックスで表示させた場合は、［OK］ボタンをクリックすることでメッセージボックスが閉じます。 • R= 復帰……エラーメッセージが表示され、ハンドラ内をフローが逆方向に移動し、ロジックユニット内の先頭のコマンド行に戻ります。
式	［式］エディタにズームしてメッセージテキストが返る式を記述します。
テキスト	テキストを入力します。 入力した文字列がテキスト入力欄の幅より大きい場合、自動的に広域ウィンドウが表示されます。 注：　テキストは［式］欄が 0 になっている場合のみ入力することができます。
表示	［表示］特性では、［エラー］処理コマンドで表示されるメッセージの形式を指定できます。 指定できるオプションは、以下の通りです。 • B= ボックス …… メッセージがダイアログボックスで表示されます。Magic xpa はこのダイアログが閉じられるまで次の処理に移行しません。［モード］特性が［E= エラー］の場合、［エラー］ダイアログが表示され、［W= 警告］の場合、［警告］ダイアログが表示されます。 • S= ライン …… ウィンドウのステータス行にメッセージが表示されます。［モード］特性が［W= 警告］の場合、Magic エンジンは直ちに次の処理に移行します。
条件	この特性には、論理条件を指定します。 その論理条件から返る値を使用して、処理コマンドを実行するかどうか制御できます。 指定できる値は以下の通りです。 • Yes …… 条件が常に［True］であることを示しています。 • No …… 条件が常に［False］であることを示しています。処理コマンドは実行されません。 • 式番号 …… 処理コマンドの実行条件を動的に制御する場合に指定します。ここから［式］エディタにズームして実行条件の論理式を定義します。

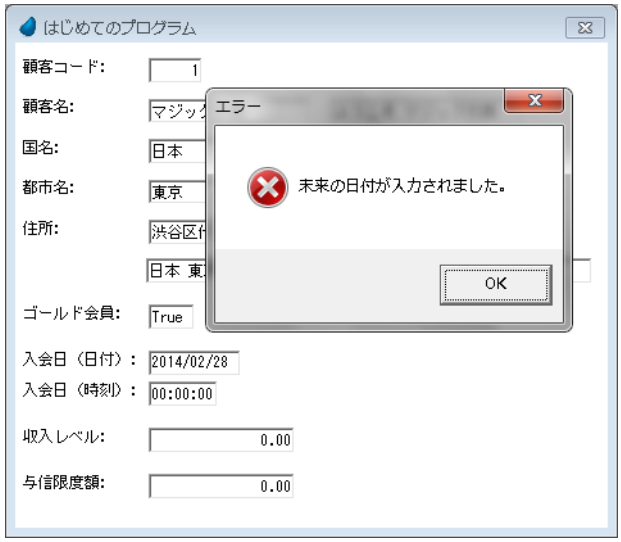
エラーチェックロジックの実行確認

プログラムを実行して、結果をみてみましょう。

1. F7 キーを押下して＜はじめてのプログラム＞を実行します。
2. 各項目にたとえば下記のように適当にデータを入力してください。（全部を入力する必要はありません。）

＜入会日（日付）＞項目に未来の日付、たとえば＜ 14/02/28 ＞を入力し、Tab キーで次の項目に移動しようとすると、エラーメッセージのダイアログボックスが表示され、［OK］ボタンを押下すると再び＜入会日（日付）＞項目にカーソルがパークし、正しい入力が行われるよう促します。

プログラムを続行するか中断するためには、エラーにならないデータ入力を行う必要があります。



５．データの整合性

プロジェクトを開発するときに、データの整合性を維持するためのルールとして考慮すべき一般的なシナリオがあります。

たとえば本コースで作成しているものを例にとってみると、エンドユーザが住所データの詳細を入力するときにデータ整合性を考慮しなければなりません。

まず＜国名＞を入力し、続いてその国にある＜都市名＞を入力する場合を考えてみます。もしエンドユーザが＜国名＞を変更した場合には、＜都市名＞はクリアしなければなりません。通常、都市は国に属しているので、国が変わることにより、すでに設定されていた都市名は意味が無くなるからです。

＜都市名＞のデータをクリアするプログラム例

このセクションでは、［項目変更］ロジックユニットを利用して、住所データの整合性を維持するための方法について説明します。

1. ＜はじめてのプログラム＞でズームします。
2. ［ロジック］エディタを選択します。
3. 新しくヘッダ行（Ctrl+H）を作成します。

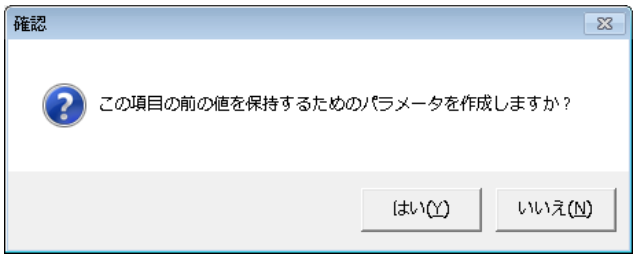
4. ヘッダ行に次のようにパラメータを設定します。

ロジックユニットタイプ	イベント	コントロール名	条件
V= 項目	C= 変更	C　国名	Yes

国名を選択するには、次のようにします。

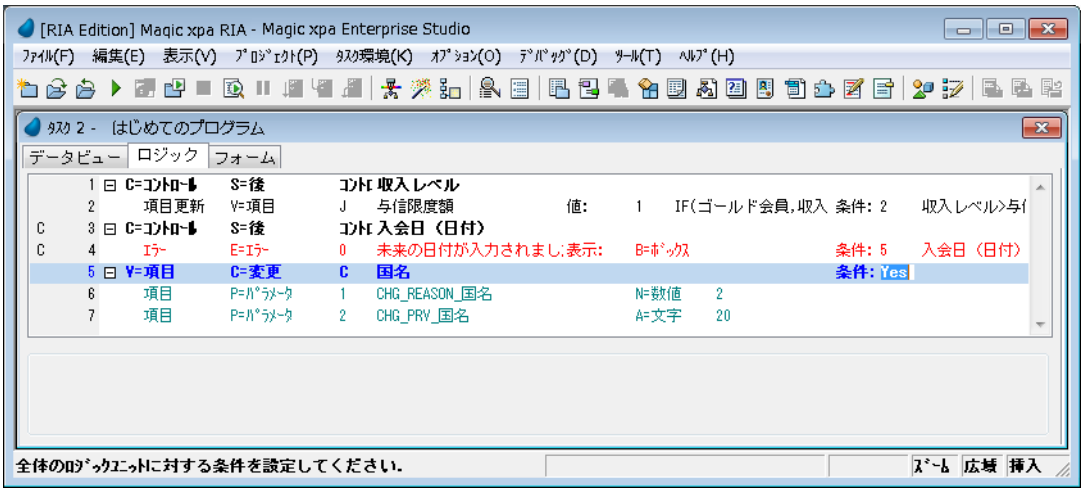
5. データ項目の［シンボル名］欄からズームし、項目一覧を表示させます。
6. ＜国名＞の項目にパークします。
7. ［選択］ボタンをクリックします。

前の値を保持するためのパラメータを作成するかどうかを聞いてくる［確認］ダイアログが表示されます。



8. ［はい］ボタンをクリックします。

ここまでの作業で、プログラムは次のイメージのようになっていることを確認してください。



9. ［項目変更］ロジックユニットのヘッダ行にパークし、F4 キーを押下して一行作成します。
10. 作成した行に次のようにパラメータを設定します。

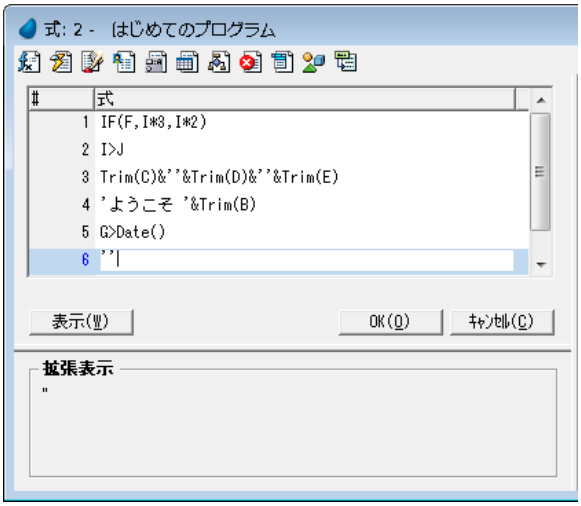
コマンド	項目	値	条件
項目更新	D （都市名）	6 （"）	7 （CHG_PRV_ 国名 <>"）

データ項目は［シンボル名］欄からズームして項目一覧を表示させ、選択します。

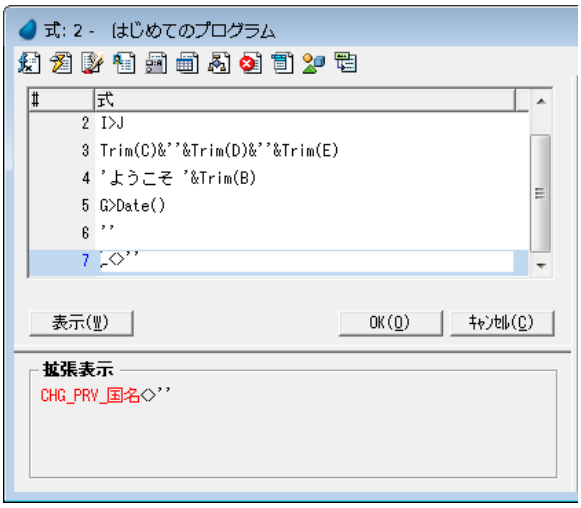
値の設定は次のようにします。

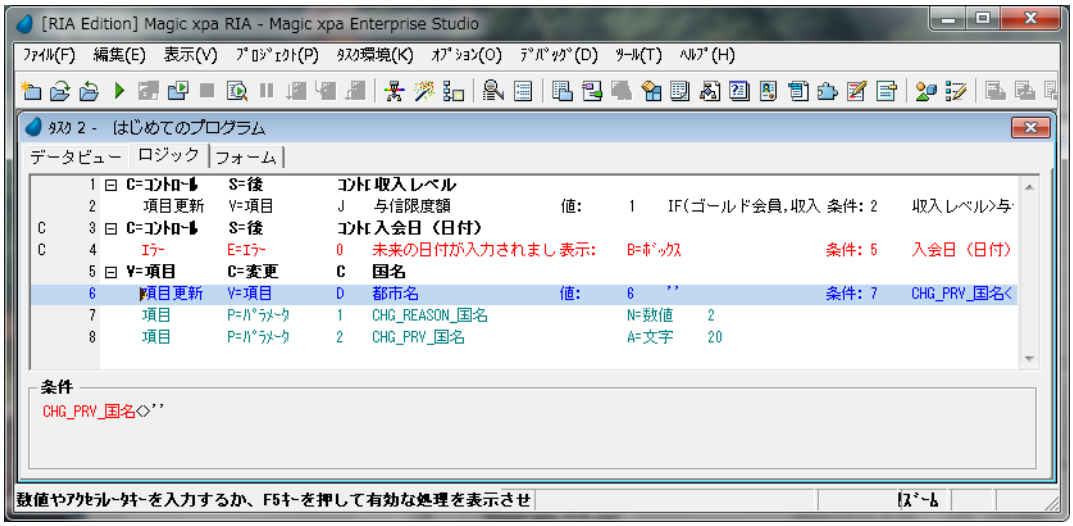
11. 値の［式］欄からズームし、［式］エディタを開きます。
12. 最終行にパークします。
13. F4 キーを押下して一行作成します。
14. 式として、＜’’＞（アポストロフィを２つ）を入力します。
15. ［OK］ボタンをクリックして［式］エディタを閉じます。

条件の設定は次のようにします。



16. 条件の［式］欄からズームして［式］エディタを開きます。
17. 最終行にパークします。
18. F4 キーを押下して一行作成します。
19. F5 キーを押下してズームし、項目リストから、＜ L（CHG_PRV_ 国名）＞を選択します。
20. 式を定義します：＜ L<>’’＞
21. ［OK］ボタンをクリックして［式］エディタを閉じます。



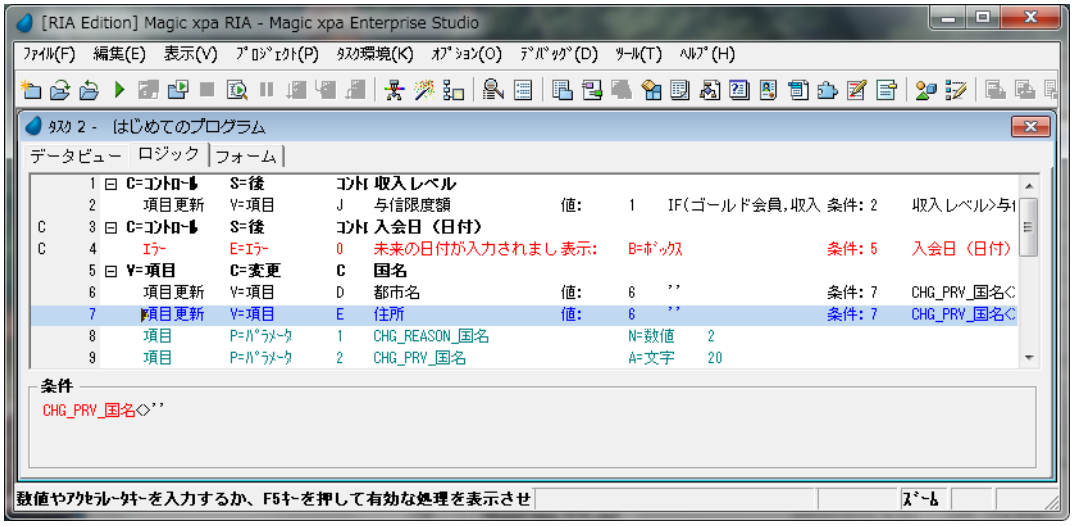


住所データ項目のクリア

都市名をクリアした方法と同様のやり方で住所データをクリアすることができます。

1. ［項目変更］ロジックユニット内に F4 キーを押下して一行作成します。
2. 次のようにパラメータを設定します。

コマンド	項目	値	条件
項目更新	E （住所）	6 （'')	7 （CHG_PRV_ 国名 <>'')



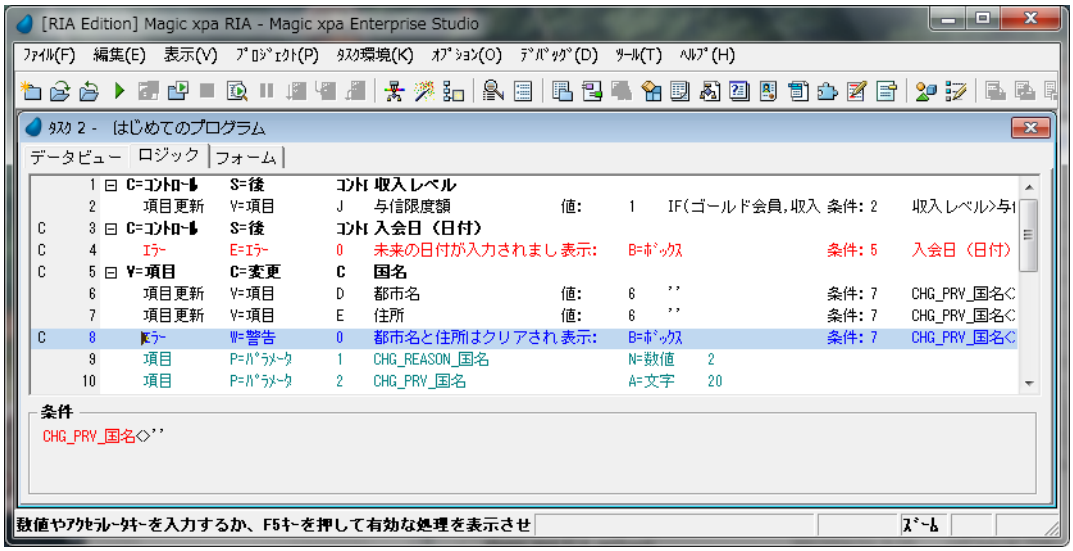
エンドユーザへの警告

次に、都市名と住所がクリアされたことをエンドユーザに知らせる［エラー］処理コマンドの使い方を見てみましょう。

1. ［項目変更］ロジックユニット内に F4 キーを押下して一行作成します。
2. 次のようにパラメータを設定します。

コマンド	モード	式	テキスト	表示	条件
エラー	W= 警告	0	都市名と住所はクリアされました！	B= ボックス	7 CHG_PRV_ 国名 <>''

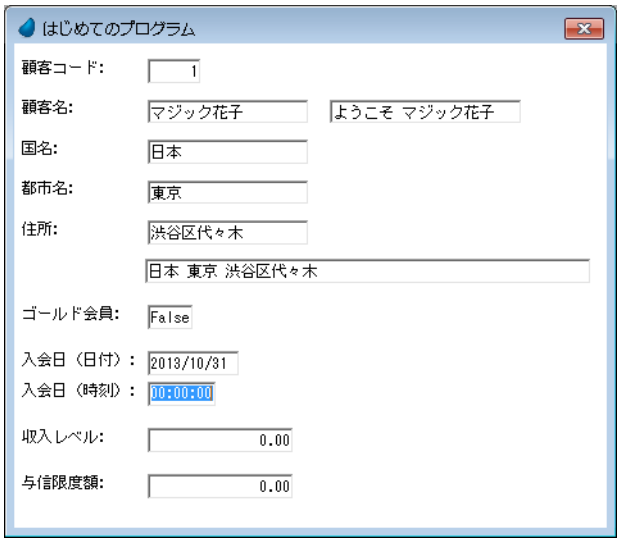
3. プログラムを閉じ、変更内容を保存します。



データ整合性を維持するロジックの実行確認

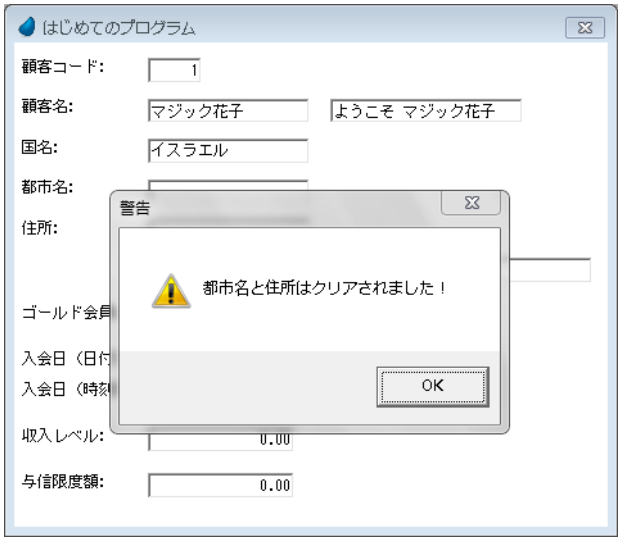
プログラムを実行させて、結果を確認しましょう。

1. F7 キーを押下して＜はじめてのプログラム＞を起動します。
2. 顧客詳細データを次の例のように入力します。



3. ＜国名＞欄を＜日本＞から**＜イスラエル＞**に変更してみましょう。
4. Tab キーを押下してください。

国名を変更した後そのコントロールから移動すると、都市名と住所がクリアされ、警告のメッセージが表示されました。



６．データ整合性を維持するための方法について－要約

今までのセクションで、どのようにデータ整合性を維持するかについて学びました。

［項目変更］ロジックユニットを作成し、国名の変更に対する処理を定義しました。

ロジックユニット内では、［項目更新］処理コマンドを用いて、都市名と住所の各データをクリアし、［エラー］処理コマンドを用いてエンドユーザへの警告メッセージを表示させました。

［項目変更］ロジックユニット内では次の条件式を定義し、使用しました：＜ CHG_PRV_ 国名 <>" ＞

この条件式は、国名が設定されているスペース以外のある値から別の値に変更されたときのみ True となり、データクリアの実行と警告メッセージの表示が行われます。

また国名が未設定の状態から最初に値を入力するときには、この条件式は False となり、データクリアや警告メッセージ表示は行われません。

7．パーク条件

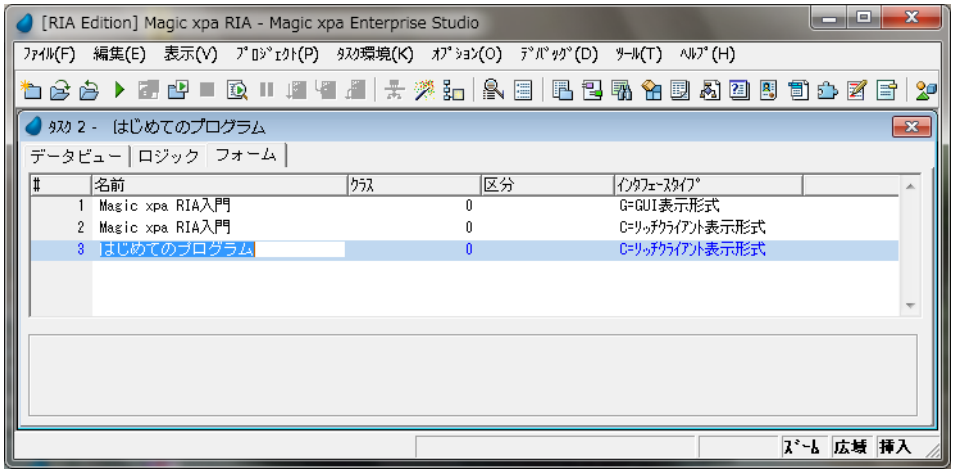
前のセクションでは、データ整合性維持に関し、ある項目値の変更がトリガとなって他の項目値に対する処理が行われる方法を学びました。

ここでは、データ整合性維持に関する別の方法、すなわち、ある項目の値によって、コントロールにパークできるかどうかを制御することで整合性を維持する方法について学習します。

パーク条件の記述例

ここでは、＜国名＞が入力されているときのみ＜都市名＞のデータ項目にパークできるという例について学びます。また、＜住所＞については、＜国名＞と＜都市名＞の両方が入力されているときのみパークできるようにします。

1. ＜はじめてのプログラム＞を選択し、F5 キーでズームします。
2. ［フォーム］エディタを選択します。
3. ＜はじめてのプログラム＞フォームを選択し、F5 キーでズームします。

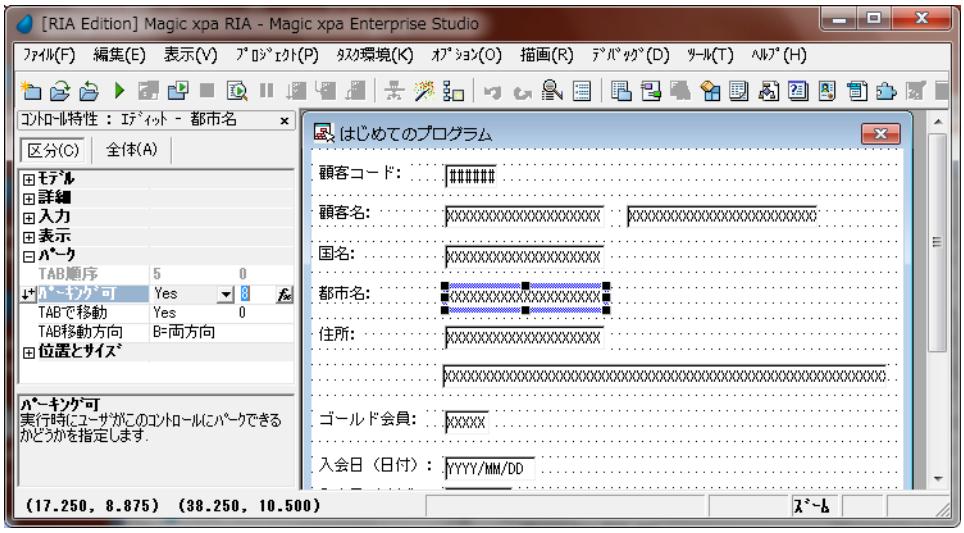


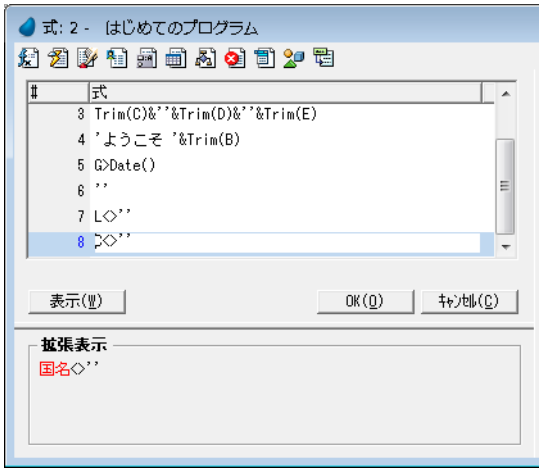
＜都市名＞のパーク条件

＜国名＞に値が設定されている場合のみ、＜都市名＞にパーク可能となる条件式を定義してみましょう。

4. フォーム上の＜都市名＞の［エディット］コントロールを選択します。
5. コントロール特性（Alt+Enter）を表示させます。
6. ［パーキング可］特性を選択します。
7. ［fx］をクリックして［式］エディタを開きます。
8. F4 キーを押下して一行作成します。（この例では最後の行の後ろに作成）

9. 項目リストから＜国名＞を選択し、次の式を定義します：＜ C<>’’（国名 <>’’）＞



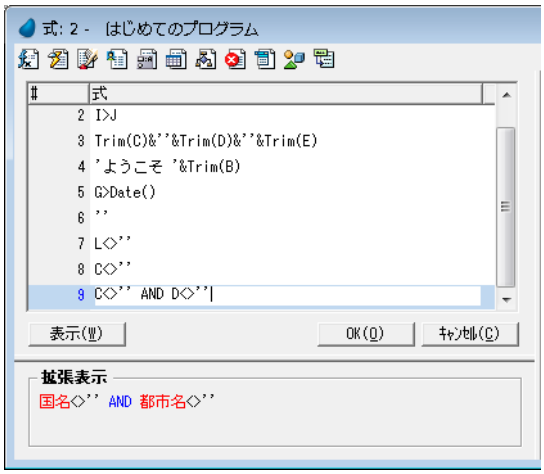


＜住所＞のパーク条件

次に＜住所＞項目への条件式を定義し、＜国名＞と＜都市名＞に値が入力されているときのみパーク可能となるようにしましょう。

1. ＜住所＞の［エディット］コントロールを選択します。
2. コントロール特性を表示させます。
3. ［パーキング可］特性を選択します。

4. ［fx］をクリックして［式］エディタを開きます。
5. F4 キーを押下して一行作成します。（この例では最後の行の後ろに作成）
6. 次の式を定義します：＜ C<>’’ AND D<>’’ （国名 <>’’ AND 都市名 <>’’）＞
7. プログラムを閉じ、変更内容を保存します。



［パーキング可］特性

［パーキング可］特性は、カーソルがそのコントロールにパークできるかどうかを制御します。

この特性が［Yes］であれば、カーソルは次のような手順でそのコントロールにパークできます。

- Tab キーを使用した項目間のカーソル移動
- マウスのクリックによるコントロールの選択

この特性が［No］のときは、カーソルはそのコントロールにパークできません。

条件式が定義されているときは、その条件式が［True］の場合はパークでき、［False］となる場合はパークできません。

この特性は、［データ特性］に項目が割り当てられているときのみ有効です。

［パーク］特性を設定した項目の動作確認

1. F7 キーを押下して＜はじめてのプログラム＞を起動しましょう。
2. ＜顧客コード＞と＜顧客名＞を入力してください。
3. ＜国名＞には入力せずに、Tab キーを押下してみましょう。

この状態では、＜都市名＞と＜住所＞にはパークできないことが分かります。

カーソルはこれらの項目をスキップし、＜ゴールド会員＞の項目に移動します。

4. 次に＜国名＞を入力してみましょう。＜都市名＞にパークするようになることが分かります。
5. さらに＜都市名＞を入力すれば、＜住所＞にパークできることが分かります。

＜都市名＞と＜住所＞の両方にデータが入力されているとき、＜都市名＞をクリアしても住所データの値には影響がないことに留意してください。

８．練習問題

本章で学んだことを応用して、以下の練習問題を解いてみましょう。

- ＜はじめてのプログラム＞において、＜顧客コード＞を入力後でなければ次のコントロールに移動できないようにしてください。
  - ［コントロール後］ロジックユニットを＜顧客コード＞コントロールに対して定義します。
  - ［エラー］処理コマンドをエラーモードで定義し、＜顧客コード＞ =0 のときエンドユーザに警告します。
- 入会日（日付）と入会日（時刻）のデータ整合性を維持するため、日付がクリアされたら時刻もクリアされるようにしてください。
  - ［項目変更］ロジックユニットを＜入会日（日付）＞に対して定義します。
  - ［項目更新］処理コマンドを定義し、＜入会日（時刻）＞をクリアするようにします。

９．要約

本章では Magic xpa に組み込まれたデータ検証メカニズムについて学習しました。

プログラムを実行させ、いろいろな入力を行い、データ項目の型によって定められた内部の検証メカニズムを実習しました。

本章の後半では、プログラムのロジックによってエンドユーザのデータ入力を検証し、［エラー］処理コマンドを利用してデータ整合性を維持する方法について学習しました。

また、［項目変更］ロジックユニットの具体例を実習することで、プロジェクト内でどのように定義したらよいのかについて学びました。

第 6 章 データ項目初期値の設定

Magic xpa では、タスクのいろいろな場所でデータ項目に初期値の設定を行うことができます。

この章では初期値設定について、いくつかの方法を学習します。

キーワード

- ［代入］特性
- タスクロジックユニット
- コントロールロジックユニット
- ［項目更新］処理コマンド
- 日付関数
- 時刻関数

学習目標

本章を学ぶことで、次の内容に対する理解を深めましょう。

- ［代入］特性に値を設定し、変数項目の初期値設定を行う方法
- ［タスク前］、［タスク後］のロジックユニット
- ［コントロール前］、［コントロール後］のロジックユニット
- タスクレベルによる［項目更新］処理コマンドの動作

参照

Magic ヘルプの［Magic xpa リファレンス／［データビュー］エディタ／詳細行／変数項目特性］の［代入］の内容

同様に、［Magic xpa リファレンス／ Magic エンジン／エンジン実行レベル］

１．はじめに

データ項目にデフォルト値として初期値を設定することは、エンドユーザの入力の手間を省くことができるため、よく行われていることです。

Magic xpa では次のような方法で項目の初期値設定を行うことができます。

- ［項目更新］処理コマンドの利用
- 変数項目の［代入］特性

本章ではまずタスクの各レベルにおける［項目更新］処理コマンドの機能について学び、次に［代入］特性と初期化のメカニズムがどのように動作するかについて学びます。

２．［タスク前］での項目更新

データ項目に初期値を設定する一つの方法は、［項目更新］処理コマンドを使用することです。

［項目更新］処理コマンドはタスクレベルのロジックユニットで利用することができます。

- タスク前 …… このロジックユニットはプログラムの最初に実行されます。このロジックユニット内のコマンドは、初期化手続きのために使用され、たとえば変数項目の初期化を行います。
- タスク後 …… このロジックユニットはプログラムの最後に実行されます。このロジックユニット内のコマンドは、終了手続きのために使用され、たとえばパラメータ項目に戻り値を設定する処理を行います。

このセクションでは、まず［タスク前］ロジックユニットで［項目更新］処理コマンドを使用し、変数項目に値を設定する方法を学習します。

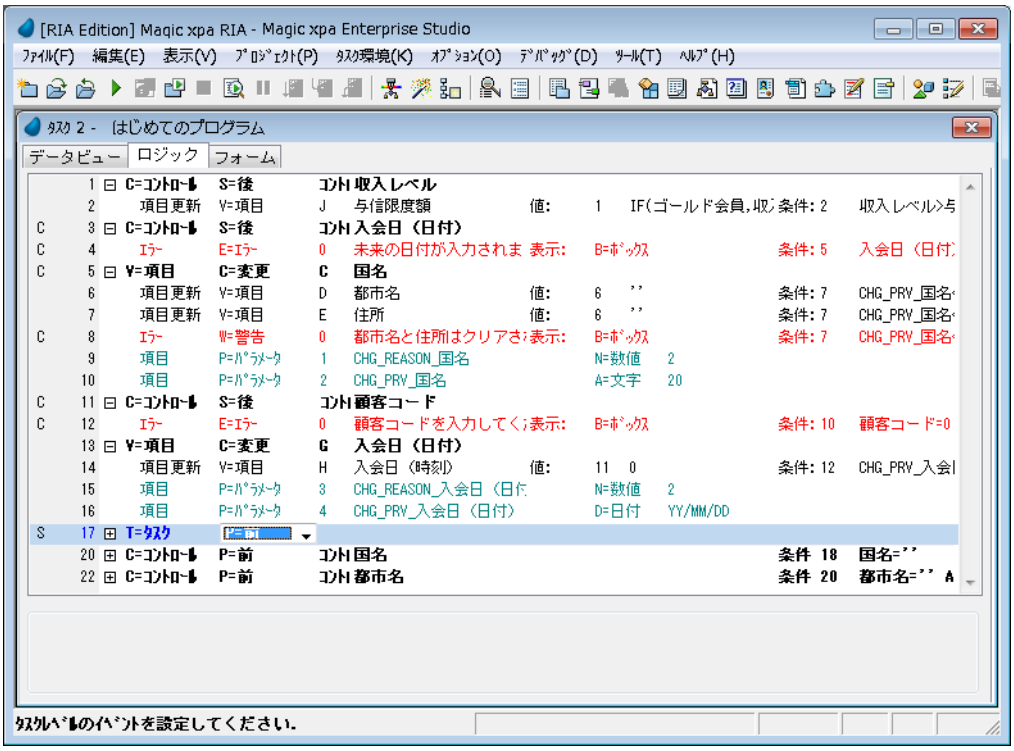
＜入会日（日付）＞にそのときの日付を設定する

Date() 関数を利用して、＜入会日（日付）＞項目を設定してみましょう。この Date() 関数はシステム日付（そのときの日付）を返す関数です。

［項目更新］処理コマンドを［タスク前］のロジックユニットで使用することにより、項目の更新はタスク実行の最初に行われます。

1. ＜はじめてのプログラム＞にズームします。
2. ［ロジック］エディタを選択します。
3. ヘッダラインを一行作成（Ctrl+H）します。
4. 次のようにパラメータ設定します。

ロジックユニットタイプ	イベント
T= タスク	P= 前



5. ［タスク前］のヘッダ行にパークした状態で、F4 キーを押下して一行作成します。
6. 次のようにパラメータを設定します。

コマンド	項目	値	条件
項目更新	G　（入会日（日付））	14　(Date())	Yes

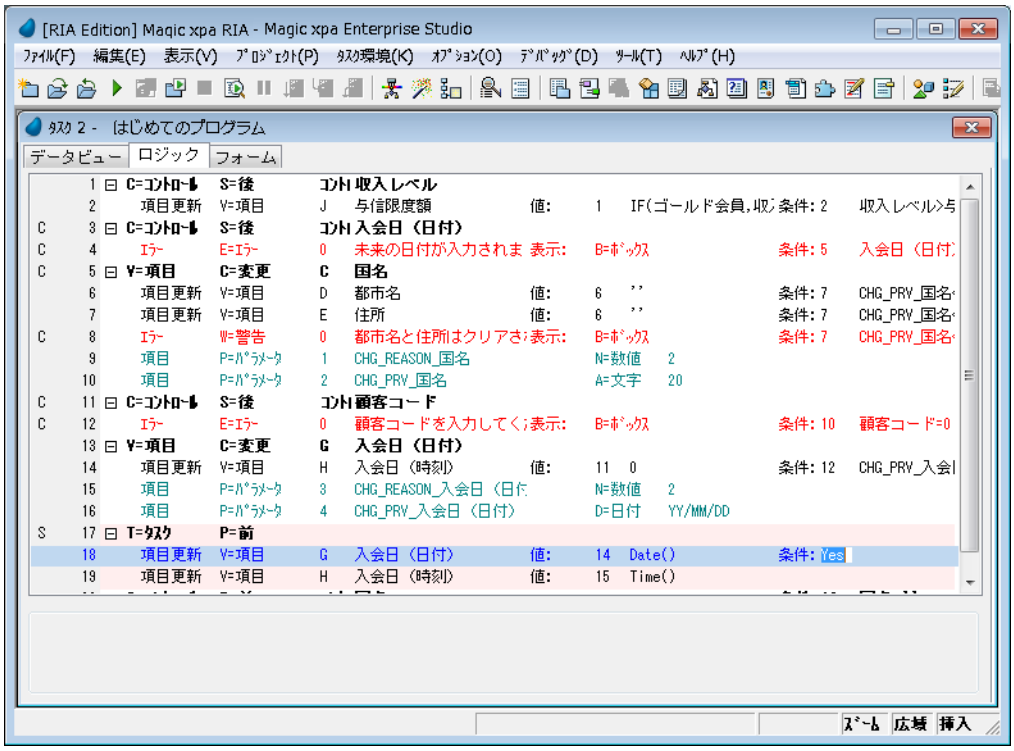
［値］欄の設定

7. ［値］欄からズームして［式］エディタを開きます。
8. 最後の行にパークします。
9. F4 キーを押下して一行作成します。
10. 作成した行に、次の式を入力してください：＜ Date() ＞
11. ［OK］ボタンをクリックして［式］エディタを終了します。

Date() 関数

システムの日付が返ります。

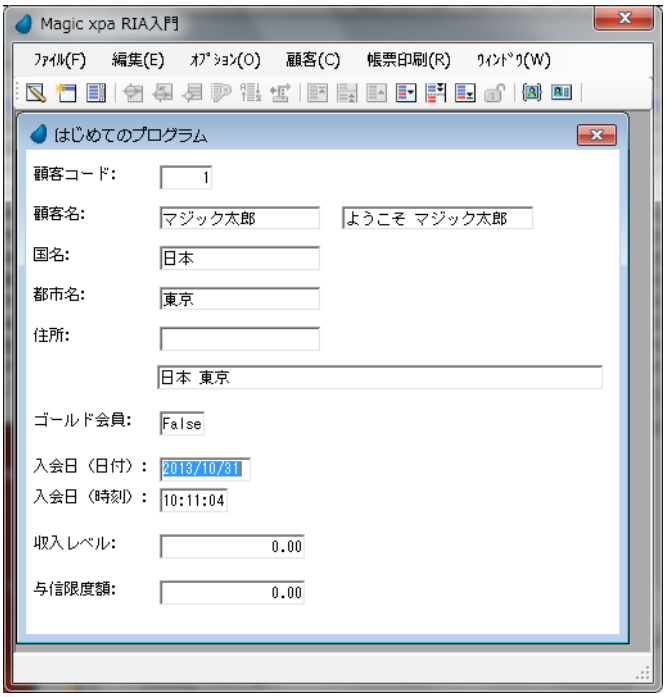
構文：　Date()
パラメータ：　なし
戻り値：　日付
例：　システム日付が 07/04/01 の場合、Date()　では 07/04/01 が返ります。
また、Date()　+5 では、07/04/06 が返ります。



［ロジック］エディタの行カラムに表示されている「C」は、クライアント側で処理が実行されることを意味しています。また、「S」はサーバ側で実行されることを意味しており、行の表示色も変わります。

12. 変更内容を保存し、プログラムを閉じます。
13. F7 キーを押下してプログラムを実行させます。

入会日（日付）としてシステム日付が設定されていることが分かります。



本コースの第 9 章では、タスクのアプリケーションエンジンについて学習し、そのなかで［タスク前］の処理コマンドが実行されるタイミングについて学びます。

＜入会日（時刻）＞をそのときの時刻に設定する

Time() 関数を利用して、＜入会日（時刻）＞項目を設定してみましょう。この Time() 関数はシステム時刻（そのときの時刻）を返す関数です。

入会日（日付）を設定した方法と同様のやり方を適用します。

1. ＜はじめてのプログラム＞にズームします。
2. ［ロジック］エディタを選択します。
3. ［タスク前］のロジックユニットにパークし、F4 キーを押下して一行作成します。

4. 次のようにパラメータ設定します。

コマンド	項目	値	条件
項目更新	H （入会日 （時刻））	15 （Time()）	Yes

［値］欄の設定

5. ［値］欄からズームして［式］エディタを開きます。
6. 最後の行にパークします。
7. F4 キーで一行作成します。
8. 作成した行に、次の式を入力してください：＜ Time() ＞
9. ［OK］ボタンをクリックします。

Time() 関数

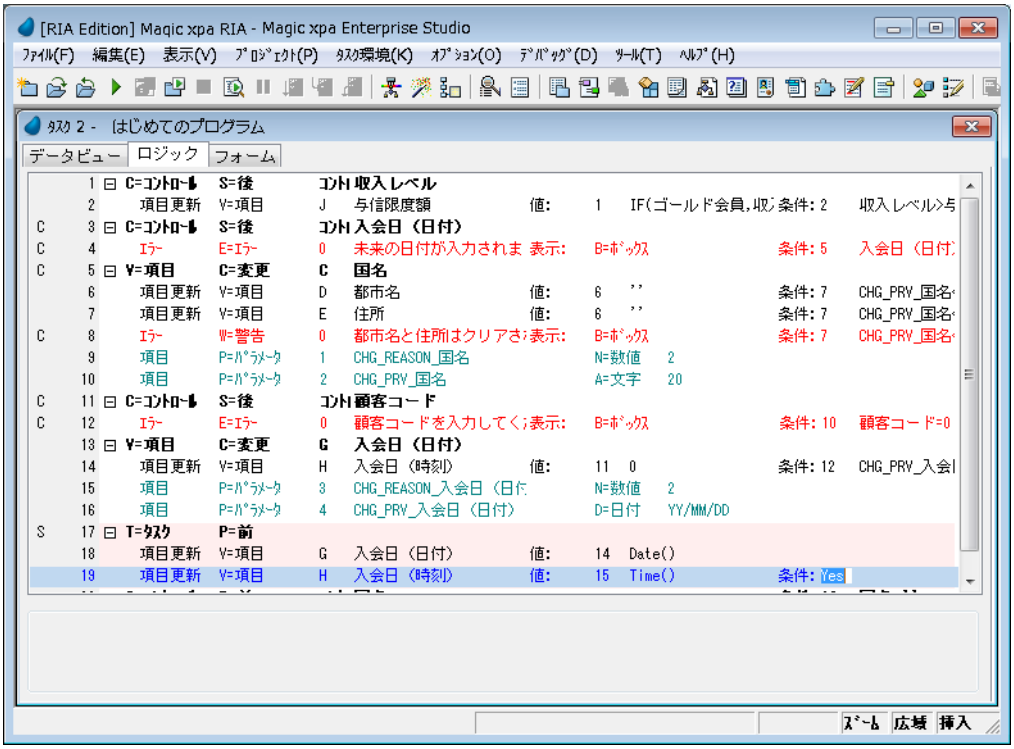
システムの時刻が返ります。

構文：　Time()

パラメータ：　なし

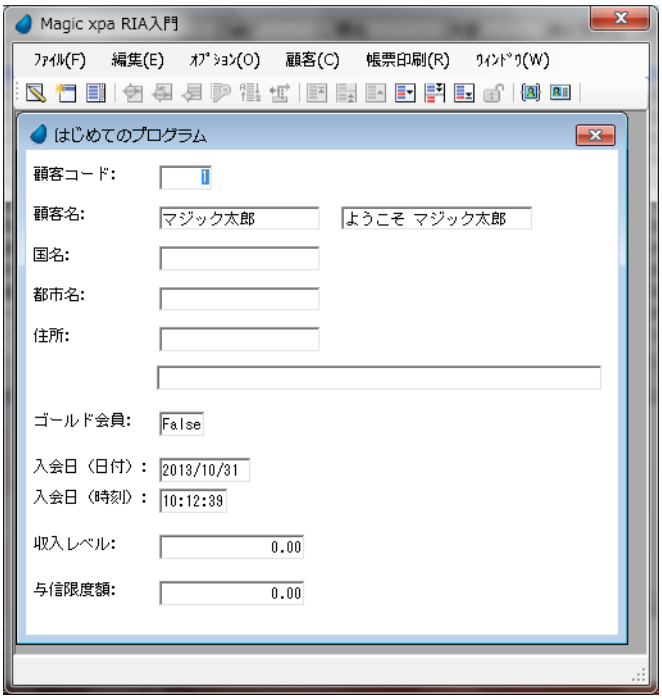
戻り値：　時刻値 … 現在の時刻（デフォルトの書式は、'HH:MM:SS'）

例：　Time() で、17:08:42 が返る場合、Time()+5 では、システムの時刻に 5 秒が加えられ、17:08:47 が返ります。



10. 変更内容を保存し、プログラムを閉じます。

11. F7 キーでプログラムを実行させます。



これまでのプログラミング作業で、入会日（日付）と入会日（時刻）にプログラム動作時点のシステム日付とシステム時刻が入力されるようになっていることが分かります。

３．項目の代入を使用する

前のセクションでは［項目更新］処理コマンドを用いて項目に値を設定する方法を学びました。

項目には、特定の値を［代入］特性を用いて初期化する必要のある場合があります。

［項目更新］処理コマンドと、項目の［代入］特性を用いて初期化を行うことの間には、相違点があります。

［項目更新］処理コマンドは手続き的なオペレーションであり、Magic xpa 内部の制御フローがその処理コマンドに到達するたびに実行されます。

項目の［代入］特性による初期化は 2 つの主要フェーズに分かれます。

すなわち、手続き的フェーズと非手続き的フェーズです。

- 手続き的フェーズは、新たにレコードが作成されるときに実行されます。このときの実行内容は、項目が変数項目か実項目かによって異なります。
  - 変数項目の時には、登録および修正モードにおいてデータは初期化されます。
  - 実項目の時には、登録モードのときのみ初期化されます。
- 非手続き的フェーズは、［代入］特性の式定義に他のデータ項目が含まれている場合に実行されます。
  - 変数項目および実項目の［代入］特性の式定義に含まれる、どれか一つの項目の内容が変更されたとき、直ちに再計算されます。

- **実項目とタスクのモードについては、この後の章において学習します。**
- **再計算のメカニズムは、項目の［代入］式定義で引用している項目がデータビューで前もって定義されている場合に有効です。（すなわち、B 項目の代入式で項目 A が使われているとき、データビュー内で A が B よりも前に定義してある場合のみ、A の内容変更にともない B が再計算されます。）**

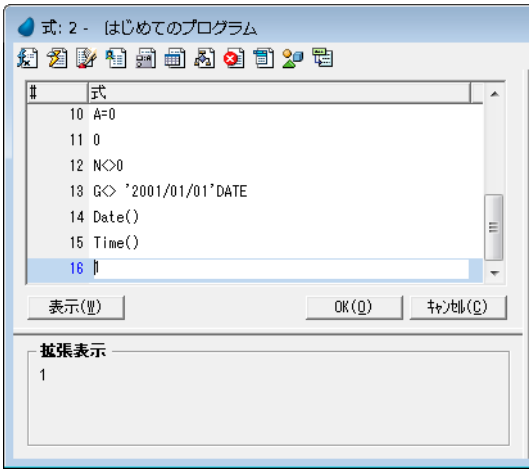
このセクションでは、変数項目の［代入］特性を用いて初期化する場合について学びます。

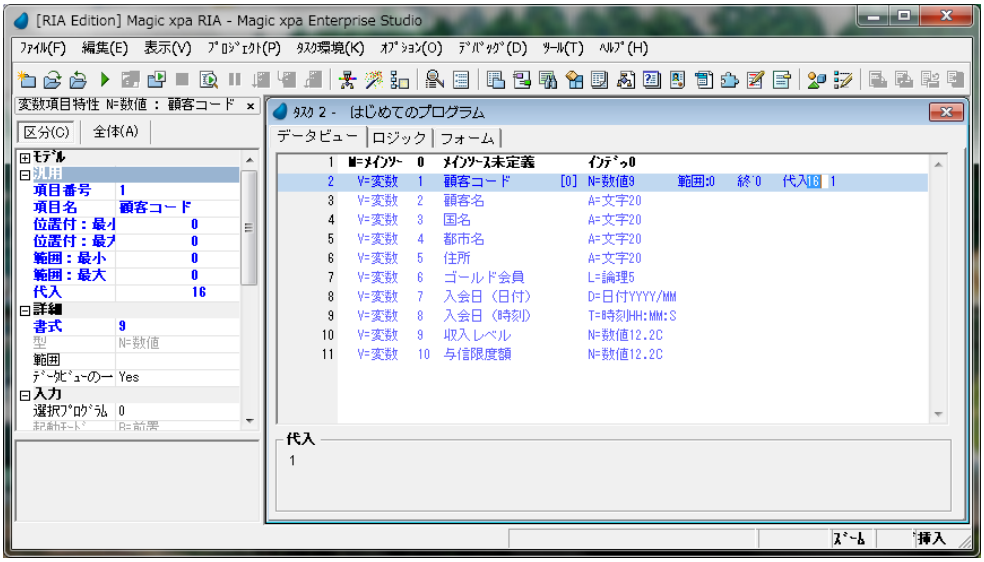
＜顧客コード＞および＜顧客名＞の［代入］による初期化

このセクションでは、＜顧客コード＞に＜１＞を、＜顧客名＞に＜マジック太郎＞を、［代入］により初期化してみましょう。

［代入］特性を用いて初期化を行うとき、フォームが表示される前に項目値の初期化が行われます。

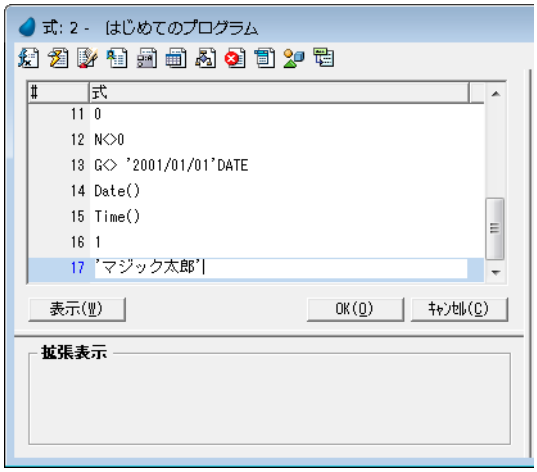
1. ＜はじめてのプログラム＞にズームします。
2. データビューを選択します。
3. ＜顧客コード＞項目の定義行にパークします。
4. ［代入］特性からズームして［式］エディタを開きます。
5. 最後の行にパークします。
6. 一行追加します。
7. 次の式を入力します：＜ 1 ＞
8. ［OK］ボタンをクリックします。



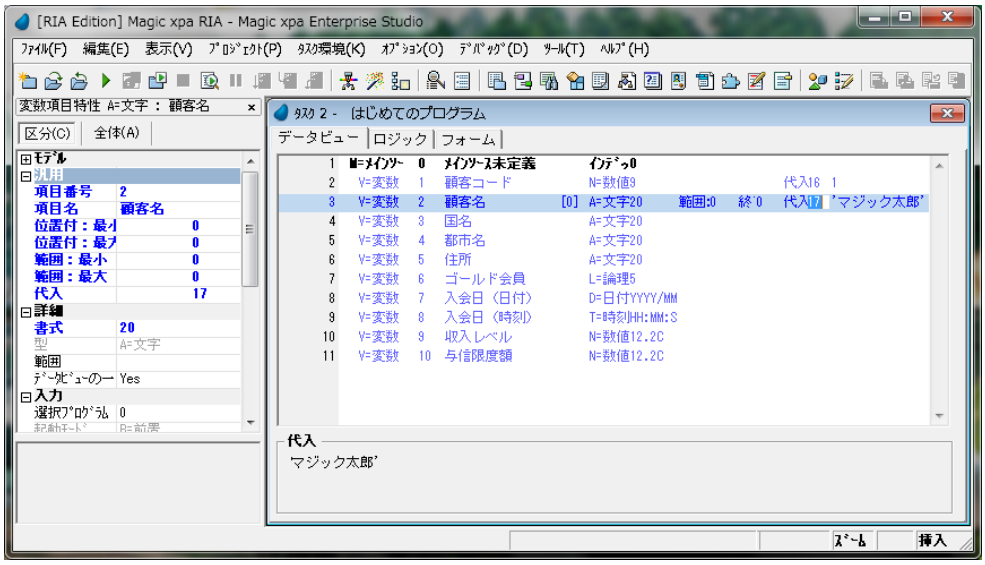


＜顧客コード＞の［代入］特性の値（この場合 16）は、［式］エディタでの番号で、式の値ではないことに留意してください。

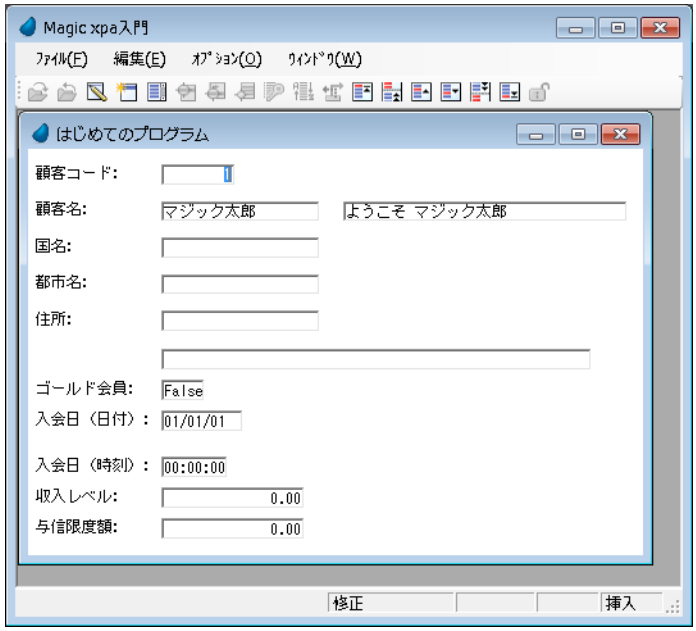
9. ＜顧客名＞項目の定義行にパークします。
10. ［代入］特性からズームして［式］エディタを開きます。
11. 最後の行にパークし、一行作成します。
12. 次の入力をします：＜’マジック太郎’＞
13. ［OK］ボタンをクリックします。



式 17 の定義で、文字列は ' （アポスト ロフィ ） で囲うことに留意してく ださい。（' マジック太郎 ' ）



14. 変更内容を保存し、プログラムを閉じてください。
15. F7 キーを押下して、<はじめてのプログラム>を実行しましょう。



プログラムの実行画面で、<顧客コード>と<顧客名>がそれぞれ式で定義した値で初期化されていることが分かります。

４．［コントロール前］での項目更新

今までのプログラム例では、［タスク前］ロジックユニットで［項目更新］処理コマンドを使用しましたが、［コントロール前］ロジックユニットで使用することもできます。

コントロールロジックユニットは、特定のコントロールの挙動を決めることがきます。

コントロールロジックユニットはいろいろな目的のために利用し、たとえばデータ項目へ値を設定するとか、他のコントロールに移る前にデータチェックを行わせたりします。

［コントロール前］ロジックユニットの処理コマンドは、カーソルがそのコントロールに移ってきたときに実行されます。対照的に、［コントロール後］ロジックユニットの処理コマンドは、カーソルがそのコントロールを離れ、他のコントロールに移るときに実行されます。

したがって、［コントロール前］ロジックユニット内に［項目更新］処理コマンドを使用すれば、カーソルがそのコントロールに移ってくるタイミングで項目のデータ更新が実行されます。

このセクションでは実際に［コントロール前］ロジックユニットを使用したプログラムを組んで項目更新の様子を確認してみましょう。

以下の例では、＜国名＞の項目には＜日本＞、＜都市名＞の項目には＜東京＞の値で項目更新を行うようにします。

＜国名＞の更新

1. ＜はじめてのプログラム＞にズームします。
2. データビューを選択します。
3. ＜顧客コード＞項目の定義行にパークします。

1. ＜はじめてのプログラム＞にズームします。
2. ［ロジック］エディタを選択します。
3. ヘッダ行を作成（Ctrl+H）します。
4. ヘッダ行を次のように設定します。

ロジックユニットタイプ	レベル	コントロール	条件
C= コントロール	P= 前	国名	18　国名 ="

［コントロール］欄の設定

5. ［コントロール］欄からズームし、コントロール一覧から選択します。

［条件］欄の設定

6. ［条件］欄からズームし、［式］エディタを開きます。
7. 最後の行にパークします。
8. 一行作成します。
9. 次の式を入力します：＜ C=’’ ＞　（国名 =" ）
10. ［OK］をクリックし、［式］エディタを閉じます。
11. ［コントロール前］ロジックユニットのヘッダ行にパークしている状態で、一行作成（F4）します。
12. 次のコマンドを定義します。

コマンド	項目	値	条件
項目更新	C　（国名）	19　（' 日本 ')	Yes

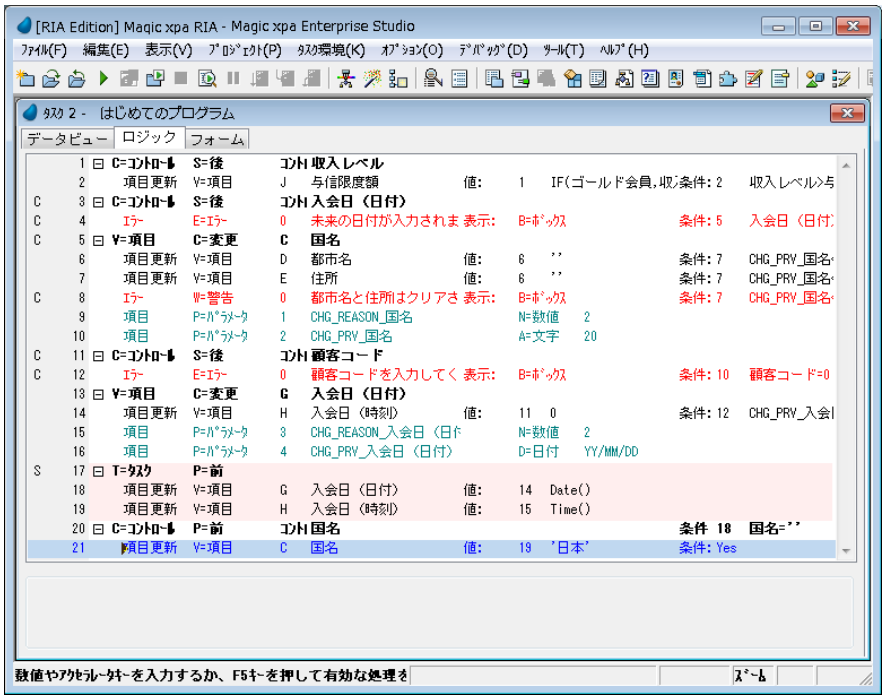
データ項目の選択

13. ［項目記号］欄からズームし、項目一覧を表示させて選択します。

［値］欄の設定

14. ［値：］欄からズームし、［式］エディタを開きます。
15. 最後の行にパークし、一行作成します。
16. 次のテキストを入力します：＜’日本’＞

17. ［OK］ボタンをクリックします。



＜都市名＞の更新

1. ＜はじめてのプログラム＞にズームします。
2. ［ロジック］エディタを選択します。
3. Ctrl+H を押下して、ヘッダ行を作成します。

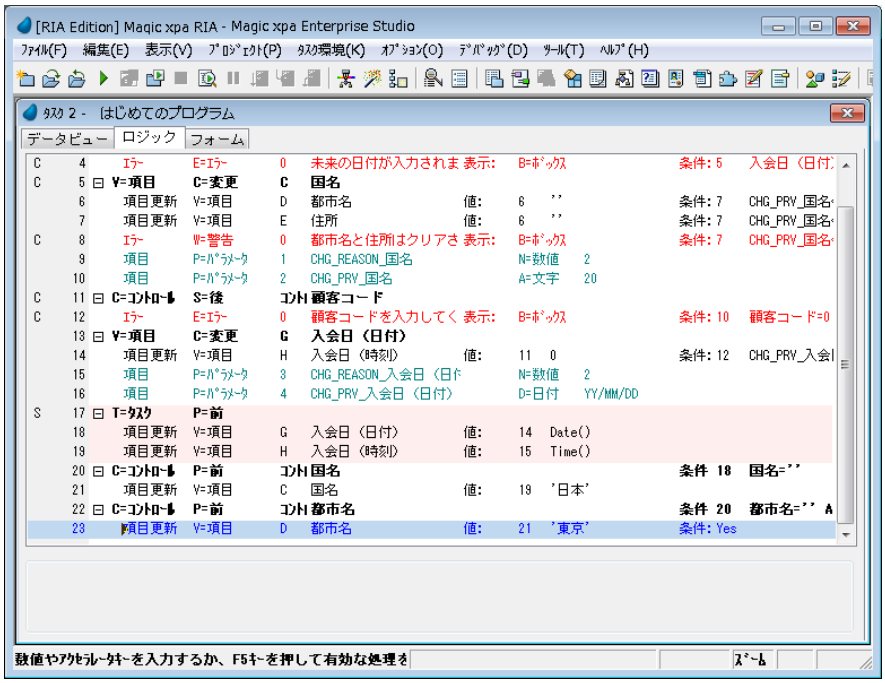
4. ヘッダ行を次のように設定します。

ロジックユニットタイプ	レベル	コントロール	条件
C= コントロール	P= 前	都市名	20　都市名 =""

5. ［コントロール前］ロジックユニットのヘッダ行にパークしている状態で、一行作成（F4）します。
6. 次の処理コマンドを定義します。

コマンド	項目	値	条件
項目更新	D　（都市名）	21　（' 東京 '）	Yes

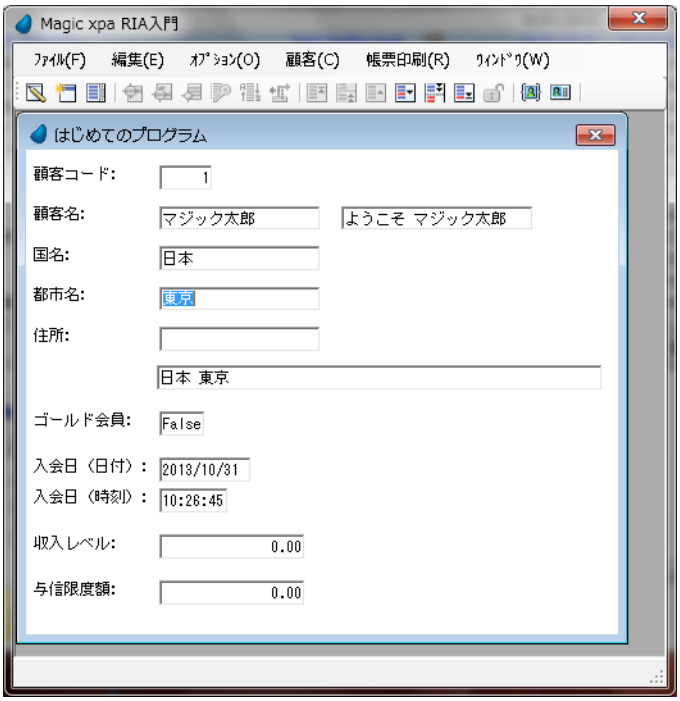
7. 変更内容を保存し、プログラムを閉じます。



結果の確認

1. F7 キーを押下して、＜はじめてのプログラム＞を実行させます。
2. カーソルを＜国名＞欄に移動してみましょう。
3. カーソルを＜都市名＞欄に移動してみましょう。

- ＜入会日（日付）＞と＜入会日（時刻）＞は、プログラムを起動したときに項目更新されています。
- ＜顧客コード＞と＜顧客名＞は、プログラムを起動した時に代入されています。
- ＜国名＞は、＜国名＞欄にカーソルをパークさせたときにのみ項目更新されます。
- ＜都市名＞は、＜都市名＞欄にカーソルをパークさせたときにのみ項目更新されます。



５．要約

この章では、項目に値を初期設定する方法について、いくつかの例を通して学びました。

具体的な例として、次の 2 つの方法がありました。

- ロジックユニット内で［項目更新］処理コマンドを使用する方法
- 項目の［代入］特性を使用する方法

また、上記方法における挙動の相違について学びました。

本章では、タスクロジックユニットとコントロールロジックユニットについて解説しました。

第 7 章 フォームの設計

Magic xpa はフォームの外形を整えるためのいろいろなツールを提供しています。フォームの色、壁紙、コントロールの色、フォント、スタイル、サイズ、また位置などを設定できます。

本章ではフォーム、およびコントロールの特性を設定する方法について学習します。

キーワード

- フォーム
- コントロール
- 壁紙
- スタイル
- 基本色テーブル
- フォントテーブル
- 論理名

学習目標

本章を学ぶことで、次の内容に対する理解を深めましょう。

- フォーム外観の設定
- コントロールの色、フォント、スタイルの変更
- コントロールのサイズと位置の変更
- フォームの色と壁紙の設定タスクレベルによる［項目更新］処理コマンドの動作

参照

Magic ヘルプの［Magic xpa リファレンス／フォームエディタ］

１．はじめに

第 3 章で［フォーム］エディタについて解説し、メインフォームについて学び、タスクの項目を表示するための基本的フォームを作成しました。

本章では、Magic xpa でのフォーム設計について、いろいろな方法とツールについてさらに知識を深め、フォームの外見を見映えよくしてみましょう。

まずフォーム上に配置された各コントロールの外見を設定します。コントロールのサイズ、位置、また色やフォント、スタイルなどを変更してみましょう。

また、Magic xpa に組み込まれたツールにより、フォームの色を変更したり、壁紙を設定したり、フォームのサイズや画面での表示位置を変更してみましょう。

本章では、＜はじめてのプログラム＞フォームと、その上に配置されたコントロールの外見を修正します。

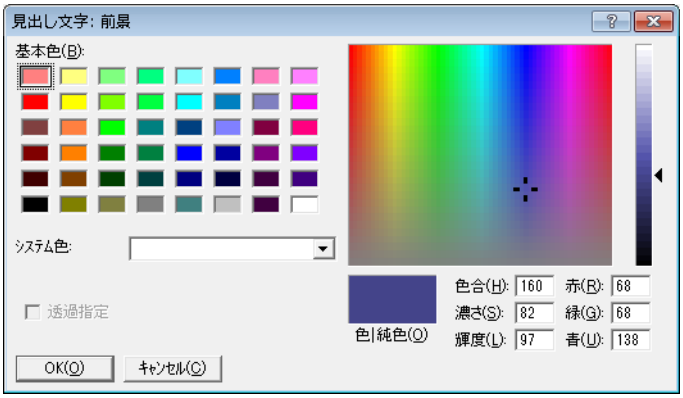
２．基本色定義ファイルへの色の追加

Magic xpa は動作環境に応じた固有の色定義ファイルを用意しています。すなわち Magic xpa Enterprise Studio 用には［clr_std.jpn］ファイル、Magic xpa 実行版および内部画面用には［clr_int.jpn］ファイル、そしてアプリケーション用としては［clr_rnt.jpn］ファイルがそれぞれ対応しています。各ファイルには色情報が定義されており、それぞれ前景色、背景色で構成されています。開発者はこれらのファイルに色定義を追加したり、既存色情報の削除、編集を行うことができます。

＜見出し文字＞色の追加

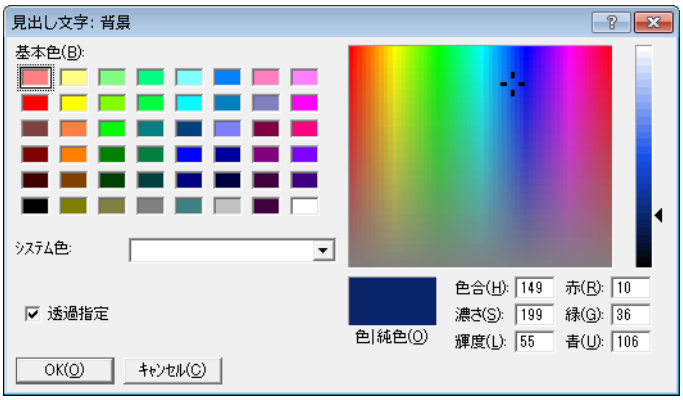
ここでは、基本色テーブルに対し、2 つの色定義を追加してみましょう。

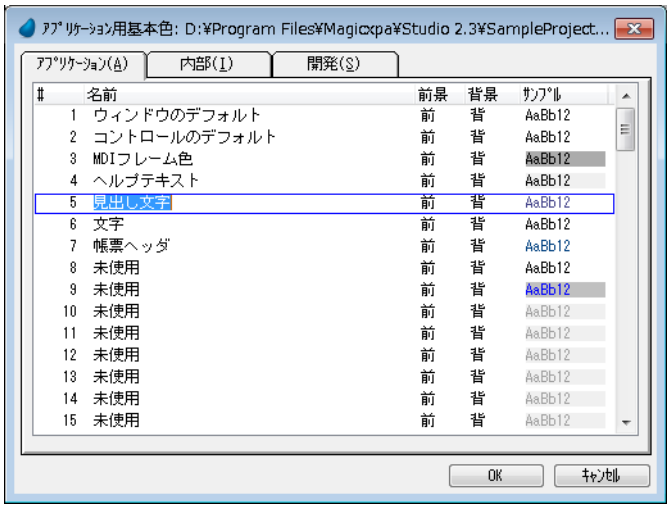
1. Magic xpa Enterprise Studio のエントリー画面で、［オプション］メニューから［設定］を選択し、さらに［基本色］を選択します。
2. ［アプリケーション］タブを選択し、アプリケーション用基本色テーブルを表示させます。
3. 5 行目の［未使用］を選択します。
4. ［未使用］の文字列を修正し、＜見出し文字＞と入力します。
5. ［前景］カラムからズームします。
   a. ［システム色］のドロップダウンリストで先頭（空行）を選択します。
   b. 赤：＜ 68 ＞、緑：＜ 68 ＞、青：＜ 138 ＞の各値を直接設定します。
   c. ［OK］ボタンをクリックして前景色の設定を終了します。



6. ［背景］カラムからズームします。
   a. ［システム色］のドロップダウンリストで先頭（空行）を選択します。
   b. ［透過指定］チェックボックスにチェックします。

c. ［OK］ボタンをクリックして背景色の設定を終了します。





基本色テーブルの詳細

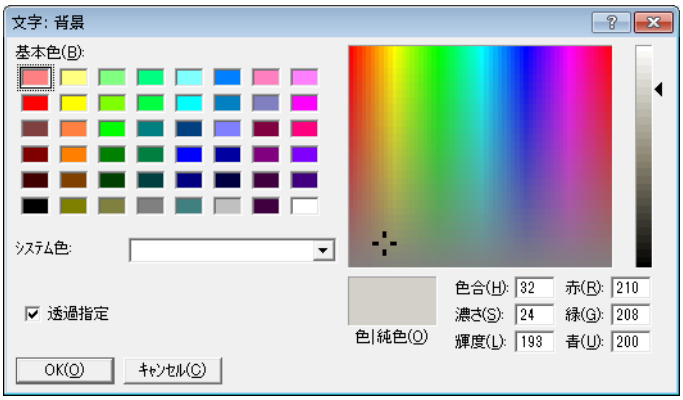
カラム	説明
#	色の識別番号。Magic xpa が自動的に生成します。
名前	色の名前。意味の分かりやすい名称をつけてください。 プロジェクトの開発期間中は首尾一貫して同じ名前を用いてください。
前景	前景色を定義します。変更するときは、この欄から［基本色定義］パレットにズームします。
背景	背景色を定義します。変更するときは、この欄から［基本色定義］パレットにズームします。背景色では［透過指定］が選択できます。
サンプル	定義した色の表示サンプルを示します。

＜文字＞色の追加

1. 基本色定義テーブルの 6 行目［未使用］を選択します。
2. ［名前］カラムの［未使用］の文字列を修正し、＜文字＞と入力します。

3. ［背景］カラムからズームします。
   a. ［システム色］のドロップダウンリストで先頭（空行）を選択します。
   b. ［透過指定］チェックボックスにチェックします。

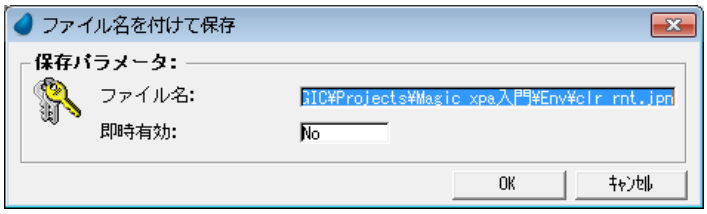
c. ［OK］ボタンをクリックします。



変更内容の保存

1. 基本色定義テーブルの［OK］ボタンをクリックします。［ファイル名を付けて保存］ダイアログが表示されます。
2. ［即時有効］パラメータを［Yes］にします。

3. ［OK］ボタンをクリックします。



３．コントロール色の変更

Magic xpa では、フォーム上に配置した各コントロールの外見をデザインすることができます。

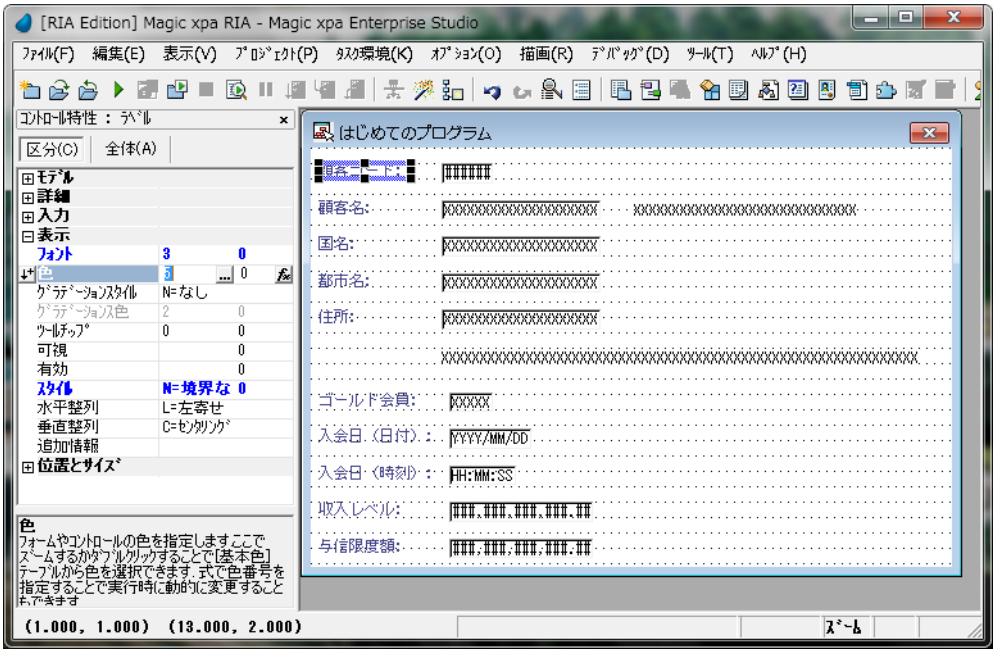
このセクションでは、コントロールの色を変更する方法を学びます。これまでの作業で基本色テーブルに、すでに 2 つの色を定義しました。#5 の＜見出し文字＞と #6 の＜文字＞の名前を持つ色です。

ここでは、［ラベル］コントロールの色を＜見出し文字＞色に変えてみましょう。

1. ＜はじめてのプログラム＞にズームします。
2. ［フォーム］エディタを選択します。
3. ＜はじめてのプログラム＞フォームにズームします。

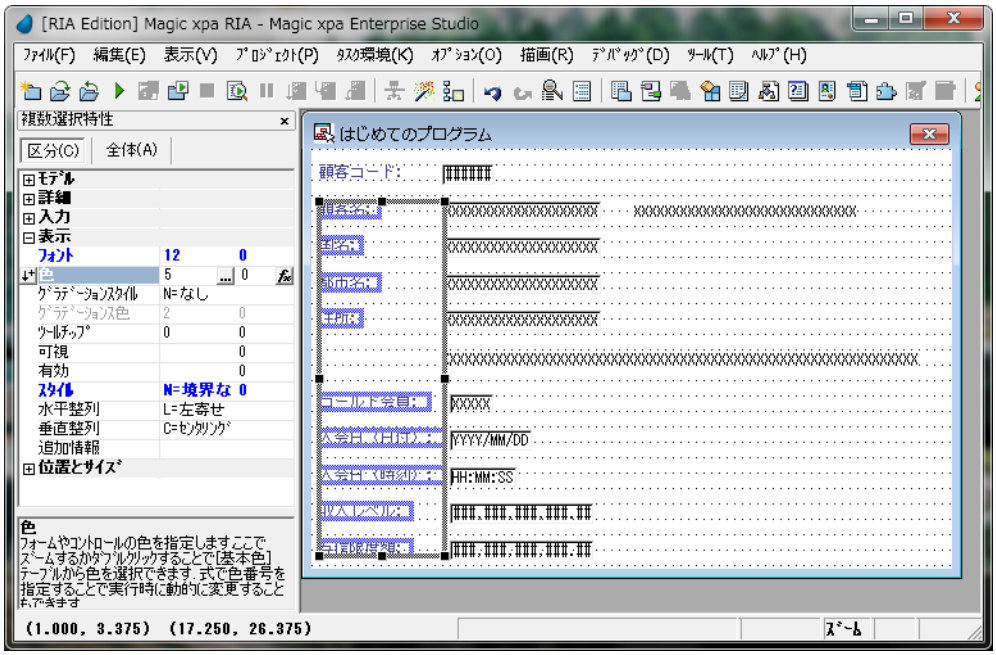
4. 「顧客コード：」［ラベル］コントロールをクリックして選択します。
5. ［編集］メニューから［特性］（Alt+Enter）を選択し、コントロール特性を開きます。

6. ［表示］の［色］特性からズームし、［アプリケーション用基本色］テーブルを開き、＜見出し文字＞を選択します。



＜顧客コード＞の文字の色が青色に変わったことが分かります。

7. ４－６の手順を、他の「顧客名：」、「国名：」、「都市名：」、「住所：」、「ゴールド会員：」、「入会日（日付）：」、「入会日（時刻）：」、「収入レベル：」、「与信限度額：」にも適用してください。



8. 変更内容を保存し、プログラムを閉じます。

４．フォント定義ファイルへのフォント追加

Magic xpa は、各動作環境に対応するフォント定義ファイルを用意しています。

- アプリケーション用（fnt_rnt.jpn）のフォント定義ファイルとして、10 のデフォルト定義を含んでいます。
- 内部用（fnt_int.jpn）のフォント定義ファイルとして、100 の定義を含んでいます。
- 開発用（fnt_std.jpn）のフォント定義ファイルとして、100 の定義を含んでいます。

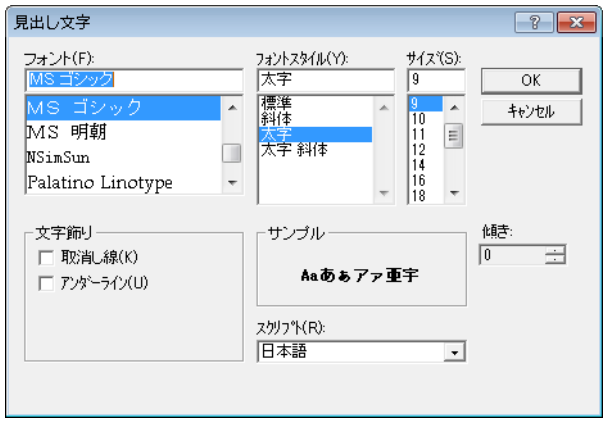
これらのファイルが含むフォント定義はツール内部で使われています。またこれらのファイルに開発者が自由にフォント定義を追加し、タスクのフォームやコントロールのデザインを行うことができます。

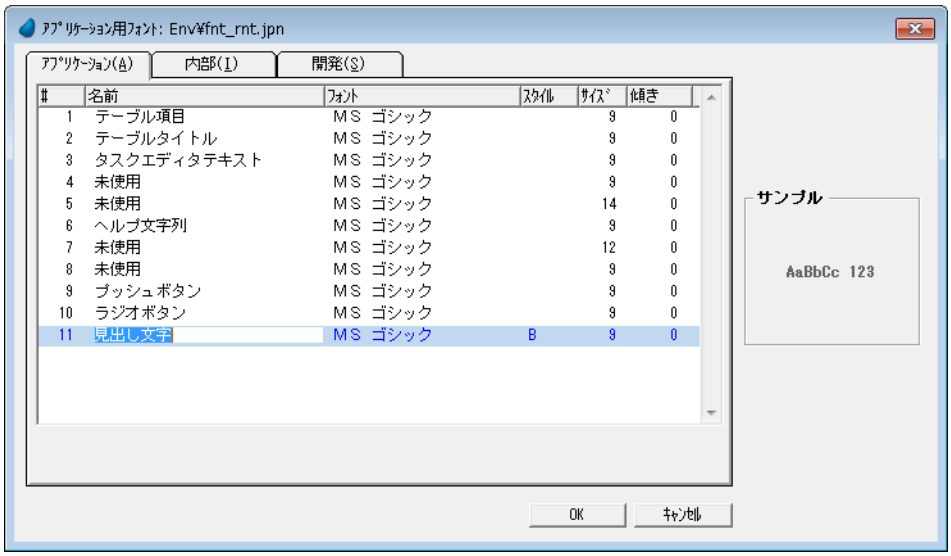
ここでは新たに 2 つのフォント定義を追加してみましょう。

＜見出し文字＞フォントの追加

1. プロジェクト一覧の画面で、［オプション］メニューから［設定／フォント］を選択します。
2. アプリケーションタブをクリックします。
3. 最終行（この場合は #10 の行）にパークします。

4. F4 キーを押下して、一行作成します。
5. ［名前］カラムに次のように入力します：**＜見出し文字＞**
6. ［フォント］カラムからズームし、＜見出し文字＞の定義ダイアログを開きます。
   a. ［フォント］として、［ＭＳ ゴシック］を選択します。
   b. ［フォントスタイル］として［太字］を選択します。
   c. ［サイズ］は＜９＞を選択します。
   d. ［OK］ボタンをクリックして、ダイアログを閉じます。





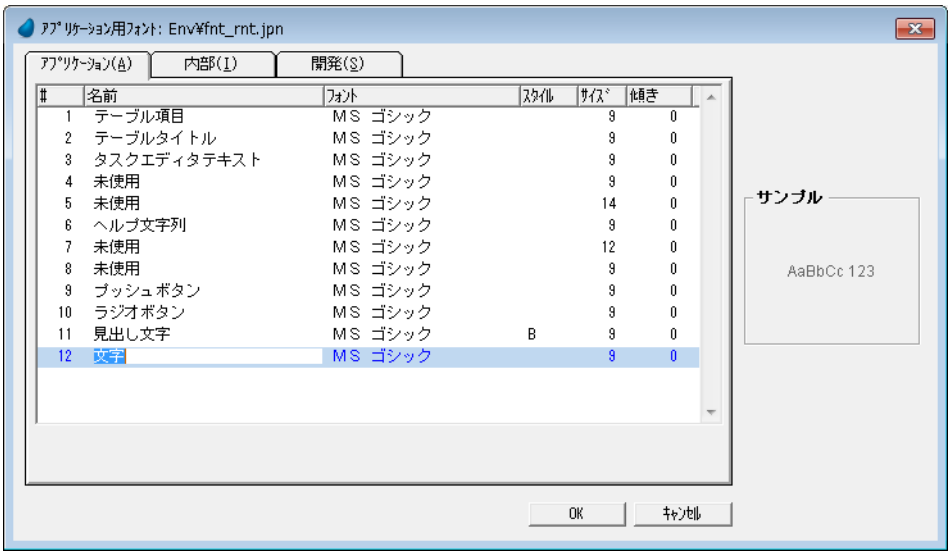
フォントテーブルの詳細

カラム	説明
#	フォント識別番号。Magic xpa が自動的に割り当てます。
名前	フォントを説明する名前を入力します。プロジェクトを保守しやすくするため、できるだけ分かりやすい表現にしてください。
フォント	ここからズームして［フォント定義］ダイアログを開き、［フォント］、［フォントスタイル］、［サイズ］、［文字飾り］、［傾き］などを指定します。
スタイル	指定されているフォントスタイルです。
サイズ	指定されているフォントの大きさです。
傾き	指定されている文字の傾き（単位：度）です。
サンプル	フォントのサンプルを表示します。

＜文字＞フォントの追加

1. ［アプリケーション用フォント］テーブルで、最終行（この場合 #11）にパークします。
2. F4 キーで一行作成します。
3. ［名前］カラムに次のように入力します：**＜文字＞**

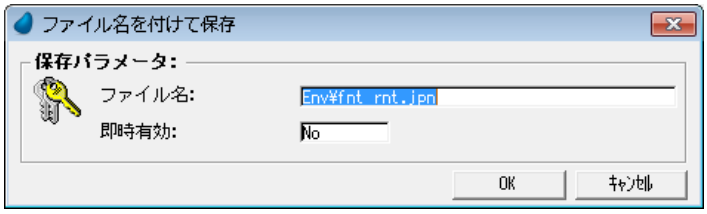
4. ［フォント］として［ＭＳ ゴシック］を選択します。



変更内容の保存

1. フォントテーブルで、内容設定後［OK］ボタンをクリックします。
2. ［ファイル名を付けて保存］ダイアログが表示されます。
3. ［即時有効］パラメータを［Yes］にします。

4. ［OK］ボタンをクリックして保存し、設定を終了します。



［ファイル名をつけて保存］ダイアログ

フォントを追加したり編集した後には、［ファイル名を付けて保存］ダイアログが表示されます。

変更内容は、このダイアログで指定するファイルに書き出されます。

［ファイル名］欄には、そのとき使用しているフォントファイル名が表示されていますが、必要に応じて別のファイル名を指定することができます。

［即時有効］欄は、変更結果を即時に反映させる（Yes の場合）か、次回 Magic xpa を起動したときに反映させる（No の場合）かを指定します。

５．コントロールのフォント変更

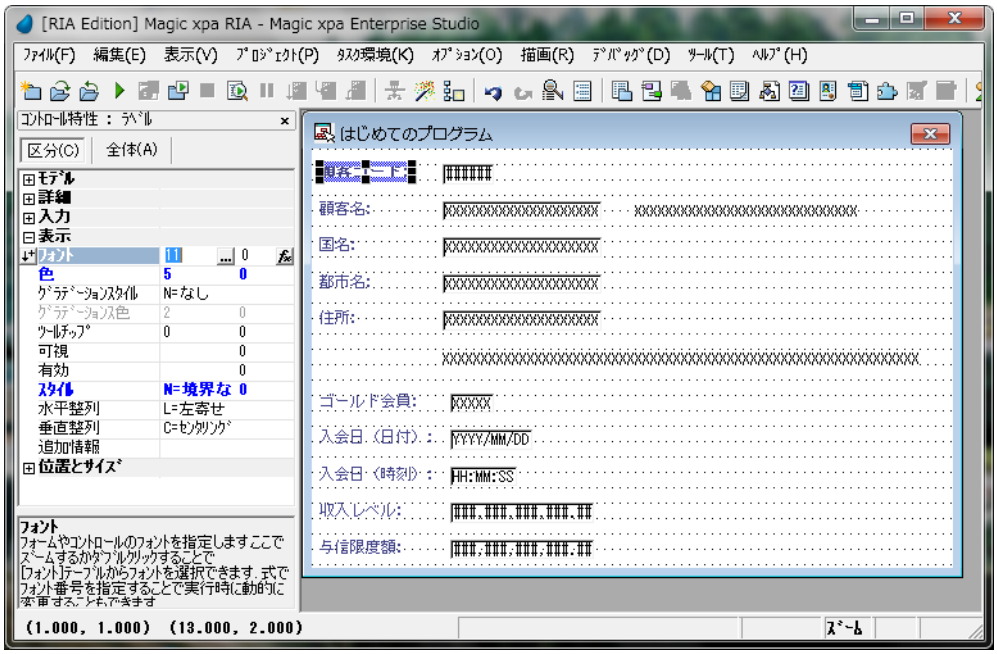
Magic xpa はフォームに配置された各コントロールの外観をデザインすることができます。

このセクションでは、コントロールのフォント変更について学びます。ここまでの作業でフォントテーブルに＜見出し文字＞（#11）と＜文字＞（#12）の 2 つの新しいフォントを追加しました。

つぎに、［ラベル］コントロールのフォントを＜見出し文字＞に変更してみましょう。

1. ＜はじめてのプログラム＞にズームします。
2. ［フォーム］エディタを選択します。
3. ＜はじめてのプログラム＞フォームにズームします。

4. ＜顧客コード：＞をクリックして、［ラベル］コントロールを選択します。
5. 特性シートが表示されないときは、Alt+Enter キーを押下して表示させます。
6. ［フォント］特性からズームし、フォントテーブルを開いて＜見出し文字＞（#11）を選択し［OK］ボタンをクリックします。



コントロールサイズの調整

選択したフォントのサイズが大きいときは、変更後はテキストの一部しか表示されません。

すべて表示させるために、右中のハンドルを右方向にドラッグしてサイズ調整することができます。

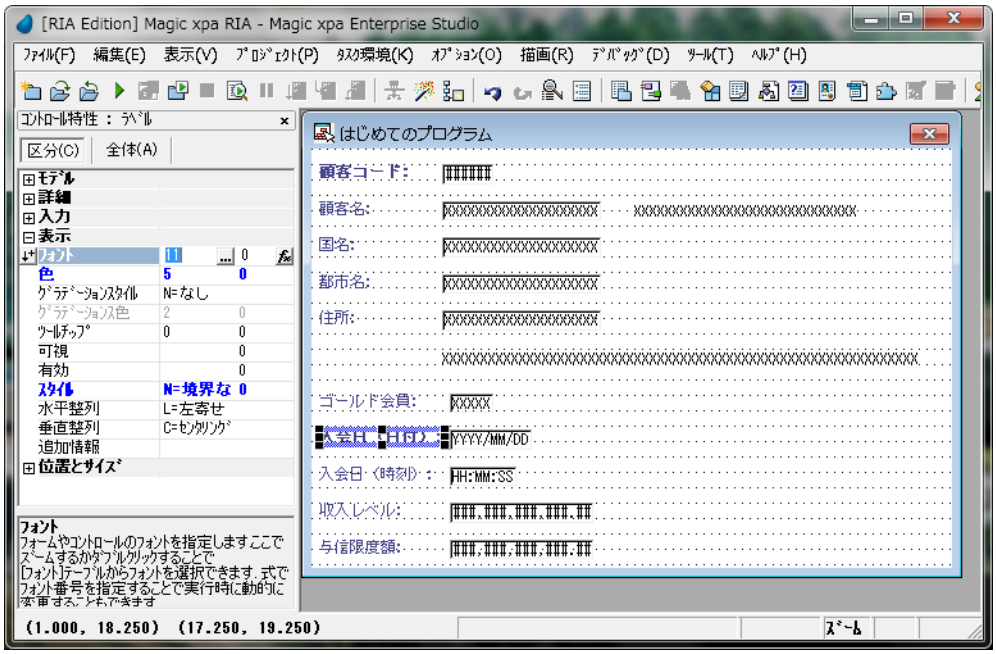
あるいは、Magic xpa の［コマンド］パレットにある［サイズ調整］機能を使用して、自動的に調整させることもできます。

1. ＜顧客コード：＞の［ラベル］コントロールを選択します。
2. ［コマンド］パレットの［サイズ調整］アイコンをクリックします。

上記操作により、「顧客コード：」の［ラベル］コントロールのサイズが変更されることが分かります。

同様の操作により、＜入会日（日付）：＞の［ラベル］コントロールのフォントを＜見出し文字＞に変更してください。

この場合、コントロールのサイズ調整により、文字の一部が右にある日付の［エディット］コントロールと重なり、正しく表示されないようになります。この状態は、Z オーダによってもたらされます。



次のセクションでは、［エディット］コントロールを移動して、［ラベル］コントロールと重ならないようにします。

Z オーダについて

フォーム中のコントロールの Z オーダは、コントロール表示上の上下関係を表します。Z オーダは、複数のコントロールの位置を重ねて定義する時に重要な意味を持ちます。

コントロールは、Z オーダに従って 2 つのグループに分類されます。

第 1 グループ	第 2 グループ
プッシュボタン	イメージ
チェックボックス	タブ
ラジオボタン	エディット
コンボボックス	ライン
リストボックス	グループ
RTF	テーブル
ツリー	
サブフォーム	

Z オーダは、同じグループ内のコントロール間で変更できますが、別のグループ間ではできません。（第 2 グループのコントロールを、第 1 グループのコントロールの上に置くことはできません。）

第 1 グループと第 2 グループのコントロール間の Z オーダの配列は、自動的に行われます。

例えば、エディットをプッシュボタンの上に重ねると、Z オーダによってコントロールが配列されて、プッシュボタンが上に配置されるようになります。

コントロールの Z オーダをフォームに表示するには、［描画］メニューから［順序／Z オーダ表示］を選択するか［コマンド］パレットの［Z オーダ表示］アイコンを押下します。また、変更するには、変更しようとするコントロールを選択し、コンテキストメニューの［並べ換え］または［コマンド］パレットの Z オーダ変更のためのアイコンを使用します。

６．コントロールの移動

フォームデザインをするなかで、フォームの見映えをよくするためにコントロールの位置を調整することがあります。

コントロールは、マウスまたはキーボードを使用して移動させることができます。

- マウスの場合 …… コントロールをクリックし、ドラッグして所定の位置まで移動します。

- キーボードの場合 …… コントロールを選択し、矢印キーを押下して所定の位置まで移動します。

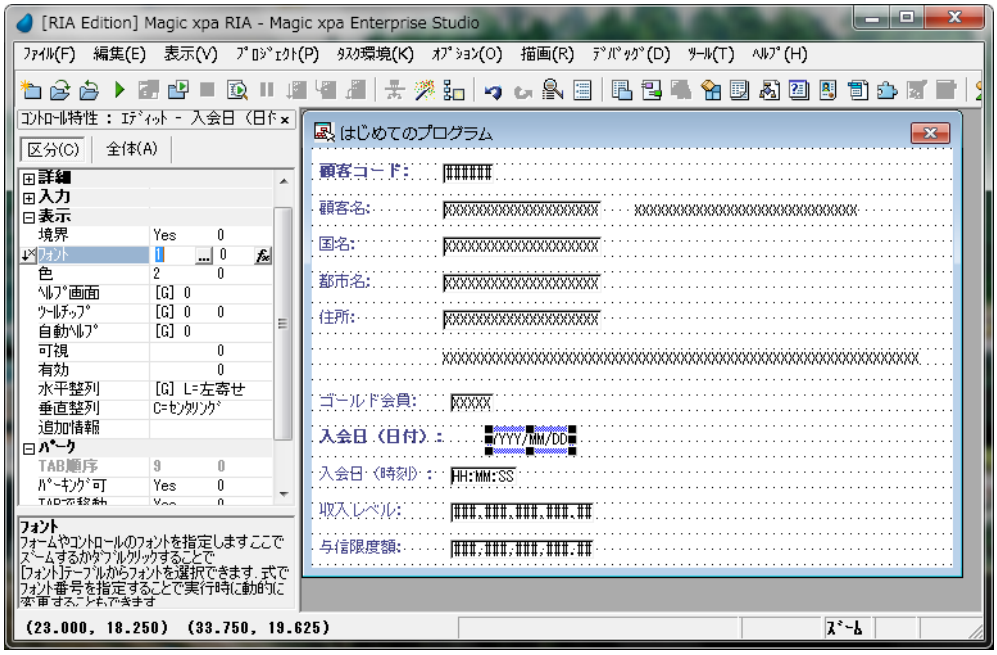
次の操作により、＜入会日（日付）＞［エディット］コントロールを移動させて、見出しの［ラベル］コントロールと重ならないようにしてみましょう。

マウスを使用して

1. ＜入会日（日付）＞の［エディット］コントロールをドラッグして右に移動させます。
2. ＜入会日（日付）＞の［ラベル］コントロールがすべて表示される位置まで移動させましょう。

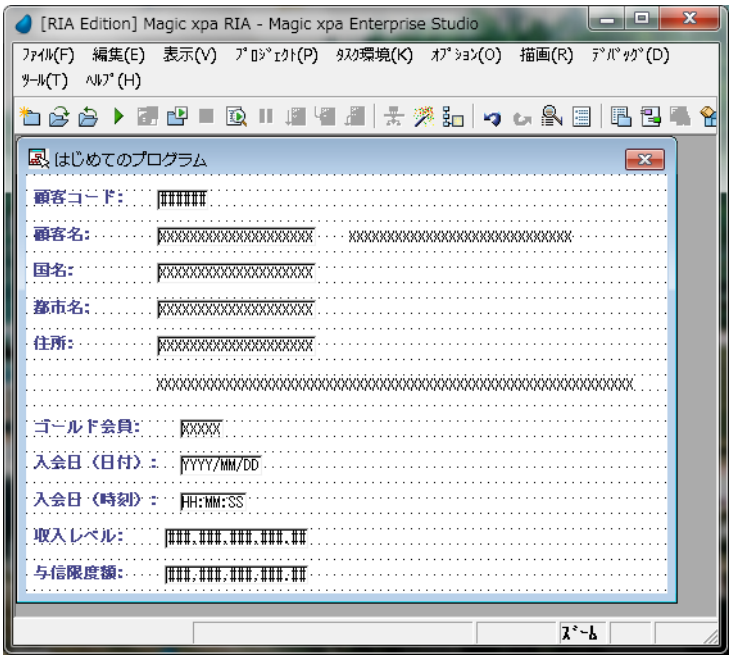
キーボードを使用して

1. ＜入会日（日付）＞の［エディット］コントロールをクリックして選択します。
2. →キーを押下して、コントロールを右に移動させます。
3. ＜入会日（日付）＞の［ラベル］コントロールがすべて表示される位置まで移動させましょう。



1. ＜顧客コード＞［ラベル］コントロールのフォントを変更したのと同じ手順で、他の［ラベル］コントロールをすべて＜見出し文字＞（#11）のフォントに変更してください。
2. すべての［ラベル］コントロールで、［サイズの調整］を行ってください。

3. すべてのコントロールの重複がないように、必要に応じて位置の調整を行ってください。



7．背景

Magic xpa のフォームに［背景］を設定することができます。

［背景］はウィンドウの背景表示として使用するイメージファイルです。

［背景］は［フォーム表示特性］の一つとして指定します。

このセクションでは、［背景］として使用するファイルを指定し、その特性について学んでみましょう。

ファイルのコピー

本コースのためにいくつかのフォルダとデータを用意しています。

本コースのデータフォルダから、次のフォルダ内容をプロジェクトのフォルダにコピーしてください。

- Exports
- Images
- Products_Pictures
- Text
- PDF

フォームの［背景］特性設定

1. ＜はじめてのプログラム＞にズームします。
2. ［フォーム］エディタを選択します。
3. ＜はじめてのプログラム＞フォーム行にパークします。
4. ［編集］メニューから［特性］（ALt+Enter）を選択します。
5. ［背景］特性からズームし、［ファイルを開く］ダイアログでファイルを参照します。
6. GS_bj.jpg ファイルを選択します。（～ ¥Studio 2.3¥Projects¥Magic xpa RIA¥Images フォルダにあります）

［ファイルを開く］ダイアログでのデフォルトは、BMP、GIF、JPG、PNG の各ファイルになっています。他のファイルを参照するときには、全てのファイル（*.*）を指定してください。

フォームの［表示］特性（リッチクライアント表示フォーム特性の［表示］セクション）

特性	説明
背景	タスクウィンドウの背景となるイメージファイル名を指定します。 ［式］エディタで式を定義して指定することもできます。
背景スタイル	背景をフォームに合わせる方法を指定します。
フォント	フォームのフォントを指定します。 ［フォントテーブル］にズームしてフォントを選択できます。式で動的に指定することもできます。 コントロールのフォントは、［コントロール特性］シートで定義します。 コントロールのフォントが指定されていない場合、フォームのフォントが使用されます。
色	フォームの色を指定します。［基本色テーブル］にズームして色を選択できます。 ［式］エディタにズームし、色番号を表す式を入力することによって動的に指定することもできます。
グラデーションスタイル	グラデーションの表示スタイルを定義します。［グラデーションスタイル］ダイアログボックスを開き、スタイルを選択できます。
グラデーション色	グラデーション表示の色を定義します。
ヘルプ画面	フォームと関連付けるヘルプ画面を指定します。［ヘルプ一覧］にズームしてヘルプを選択できます。リッチクライアントの場合、URL ヘルプのみ利用できます。
境界スタイル	フォームの境界線のスタイルを指定します。I= 細線、H= 太線 、N= なし

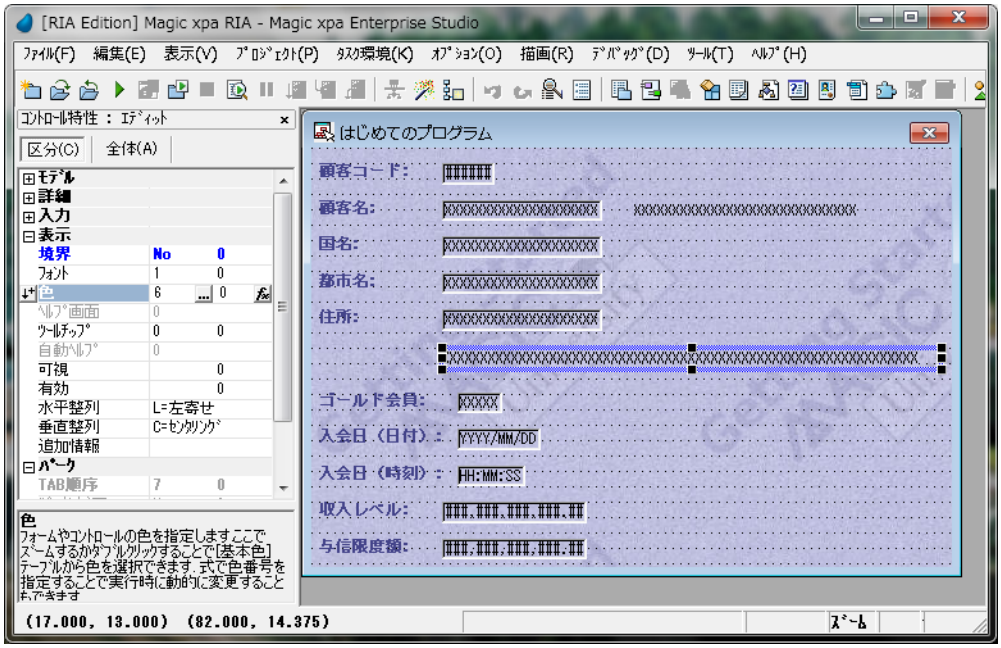
透過色の設定

背景を設定したとき、コントロールの色を透過色に設定し、背景との親和性を高めたい場合があります。透過色に設定する場合は、その色特性を透過色特性のものにします。

次の例では、＜全住所＞の［エディット］コントロールに透過色を設定してみます。

1. ＜はじめてのプログラム＞フォームにズームします。
2. ＜全住所＞の［エディット］コントロールにパークします。
3. コントロール特性（Alt+Enter）を表示します。

4. ［色］特性で、＜文字＞（#6）を選択します。
5. 変更内容を保存し、プログラムを終了します。



ここまでの作業で、フォームにブルーの背景が設定され、また＜全住所＞のコントロールを背景色に対して透過にしました。

動的なファイルの設定

前の例では背景のファイルとして、具体的なパスとファイル名を指定しました。

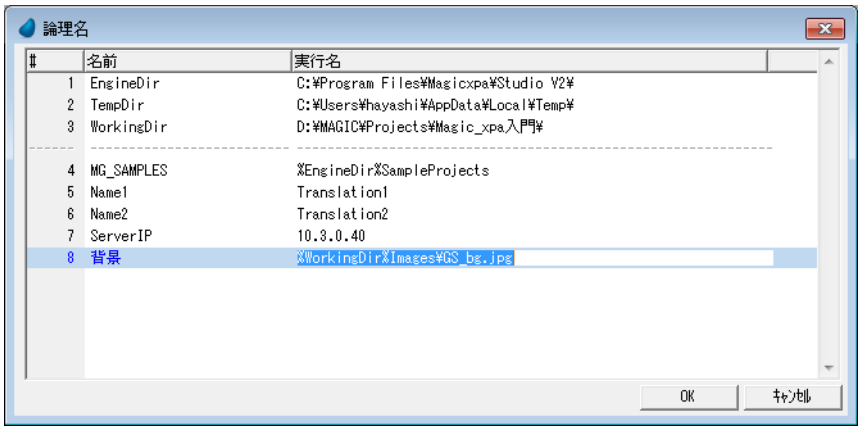
しかし、このように具体的な名前や固定したパスを記述することは、プロジェクトの管理上あまり得策ではありません。たとえば< GS_bg.jpg >を複数のプログラムで背景用のファイルとして使用する場合、もしフォルダ名を変更したり、ファイルを別のフォルダに移動したりすると、すべてのプログラムについて修正をしなければなりません。これでは実際には管理が難しくなってしまいます。また、同じプロジェクトを、異なるフォルダ構造を持つ他のコンピュータでも動作させる場合、その環境に合わせたファイルパス設定などの修正が必要になってしまいます。この問題を解決するために、Magic xpa では［論理名］を用意しています。

［論理名］はプロジェクトの保守性を高めるための、とても重要な機能です。［論理名］を利用することで、物理ファイルに対するアクセスパスを具体的に記述しなくてもよくなります。ファイル名を記述する必要があるときは、いつでも論理名を使用することができます。Magic xpa は実行時に［論理名］テーブルを参照し、実際の物理パスを判断します。

論理名の設定

ここで背景のファイルを指定するのに［論理名］を使用してみましょう。

1. ［オプション］メニューから［設定／論理名］を選択し、［論理名］テーブルを開きます。
2. 最後の行にパークし、F4 キーを押下して一行作成します。
3. ［名前］カラムに、**<背景>**と入力してください。
4. ［実行名］カラムに、< %WorkingDir%images¥GS_bg.jpg >と入力してください。
5. ［OK］ボタンをクリックして［論理名］テーブルを閉じます。



論理名テーブルについて

Magic xpa には論理名機能があり、この機能を使用することで、実際の物理ファイルパスや OS のネーミング規則から離れて、アプリケーションを開発できます。Magic xpa は論理名テーブルを実行時に参照し、実行環境の物理的情報に変換するため、アプリケーションの移植が容易になります。

カラム	説明
#	プロジェクト内で使用される論理名の ID。Magic によって自動的に設定されます。
名前	プロジェクトで使用する論理名を指定します。この論理名は、実行時に［実行名］欄で指定された値に置き換えられます。論理名は、同じプロジェクトであれば、プラットフォームや OS が変わっても同じです。
実行名	ここには、実際の名前を指定します。ファイルへのアクセスの際には、論理名に代わって、この名前が使われます。この名前は、Magic xpa のインストール環境に応じて変更してください。また、論理名の前後に［%］を付加することで実行名の中に別の論理名を使用することができます。

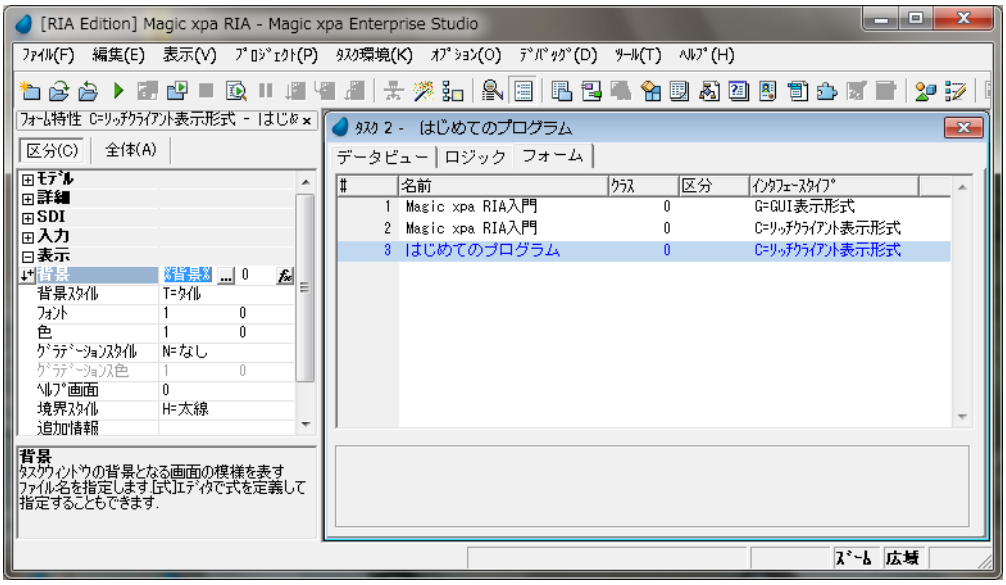
論理名には、以下の文字が使用できません。

- **0x1F 以下の制御コード**
- **?, *, <, >**

背景ファイルを論理名で指定

1. ＜はじめてのプログラム＞にズームします。
2. ［フォーム］エディタを選択します。
3. ＜はじめてのプログラム＞フォームにパークします。

4. フォーム特性（Alt+Enter）を開きます。
5. ［背景］特性で、**＜％背景％＞**と入力してください。



これで、背景のファイルを論理名＜背景＞で指定したことになります。

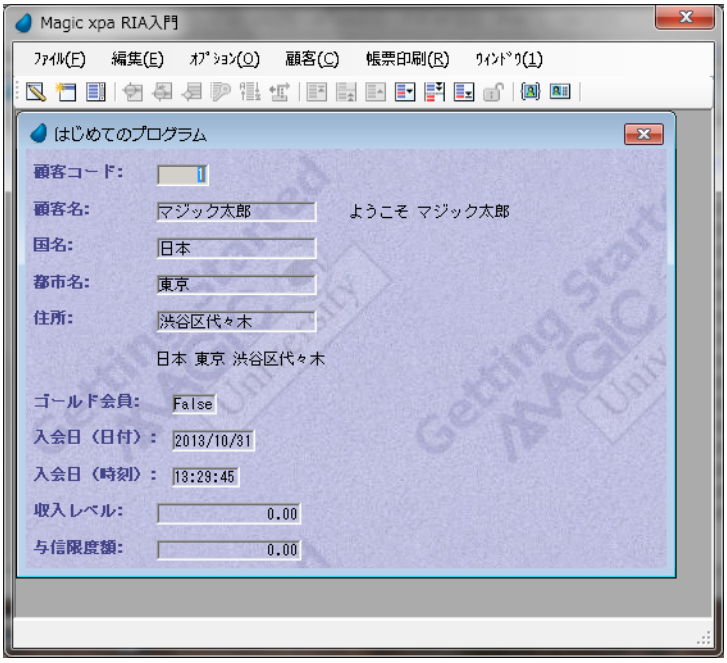
論理名は環境に依存する設定ですので、実際にプロジェクトがインストールされるコンピュータに合わせて変更されます。

Magic xpa は論理名＜背景＞を実行時に実際の物理情報に変換し、動作します。

プログラムの実行と確認

これまでの作業でフォームのデザインは終了です。プログラムを動かして表示させてみましょう。

1. F7 キーを押下して＜はじめてのプログラム＞を起動させます。
2. 顧客情報を適当に入力してみてください。



８．練習問題

本章で学んだことを応用してみましょう。

- フォームの＜ようこそ＞［エディット］コントロールを透過にしてください。
  - ［色］特性を＜文字＞にします。
  - ［フォント］特性を＜文字＞にします。
- 以下の［エディット］コントロールについても、フォントを＜文字＞、色を＜文字＞の各特性にしてください。
  - 顧客コード
  - 顧客名
  - 国名
  - 都市名
  - 住所
  - ゴールド会員
  - 入会日（日付）
  - 入会日（時刻）
  - 収入レベル
  - 与信限度額

次に、色とフォントの定義ファイルをこのプロジェクト固有のものにしましょう。

標準の設定ファイルをプロジェクトのフォルダにコピーし、論理名を使用してアプリケーション特性に定義しましょう。

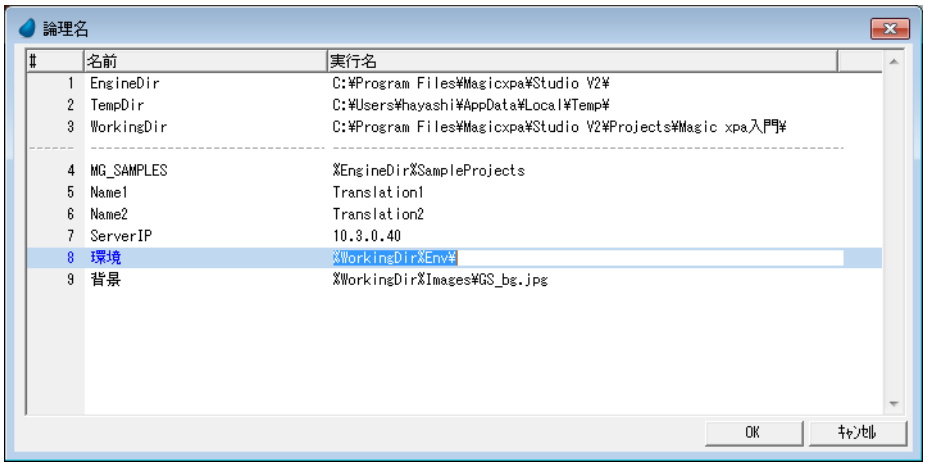
色およびフォントの定義ファイルをコピー

Windows の Explorer を利用します。

1. 本コースのプロジェクトフォルダ（C:¥Program Files¥Magicxpa¥Studio 2.3¥Projects¥Magic xpa RIA）に新しいフォルダ 'Env' を作成します。
2. Magic xpa の Support フォルダ（C:¥Program Files¥Magicxpa¥Studio 2.3¥Support）から、clrr_rnt.jpn と fnt_rnt.jpn をコピーし、Env フォルダに貼り付けます。

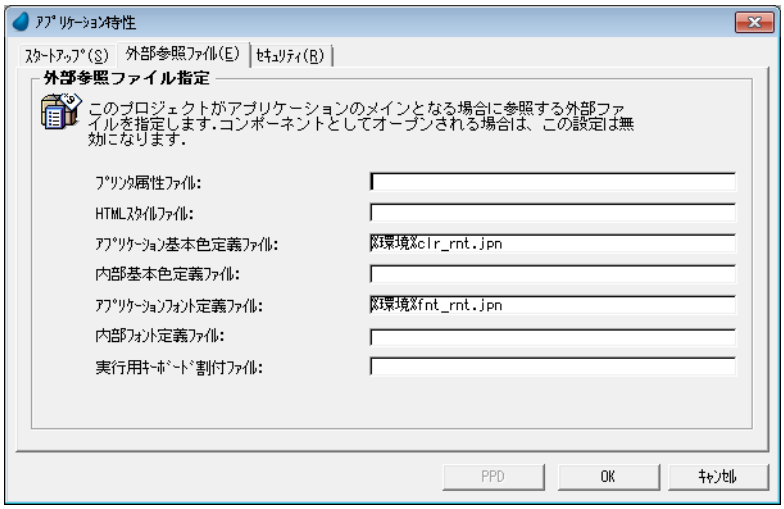
論理名の作成

1. ［オプション］メニューから［設定／論理名］を選択し、論理名テーブルを表示します。
2. 一行作成します。
3. ［名前］カラムに＜環境＞と入力します。
4. ［実行名］カラムに、＜ %WorkingDir%Env¥ ＞と入力します。
5. ［OK］ボタンをクリックし、論理名の設定を終了します。



色およびフォント定義ファイルをアプリケーション特性に登録

1. ＜ Magic xpa RIA ＞プロジェクトを開いた状態にします。
2. ファイル］メニューから［アプリケーション特性］（Ctrl+Shift+P）を選択します。
3. ［外部参照ファイル］タブをクリックします。
4. ［アプリケーション基本色定義ファイル：］欄に、＜％環境％clr_rnt.jpn ＞と入力します。
5. ［アプリケーションフォント定義ファイル：］欄に、＜％環境％fnt_rnt.jpn ＞と入力します。
6. ［OK］をクリックし、アプリケーション特性の設定を終了します。



これらの操作では特にプロジェクトに変化はありませんが、今後基本色およびフォントの各テーブルに加える変更はこのプロジェクトの定義ファイルにのみ反映され、［Support］フォルダにある標準の定義ファイルとは関係がなくなります。

９．要約

本章では、以下のことを学びました。

- Magic xpa のフォームデザインの方法
- プロジェクトで使用する基本色、フォントの定義方法
- コントロールの色、フォント、サイズ、位置、スタイルなどの変更方法
- フォームの背景設定やコントロールの外観に関する特性
- プロジェクトの管理に有効な論理名の定義や使い方

第 8 章 データソースの内容参照

本章では、Magic xpa でのデータソース定義と、単純なプログラムを使用してその内容を表示する方法を学習します。

キーワード

- データベース
- データソース
- カラム
- インデックス
- レコード
- APG
- スクリーンモード
- ラインモード
- 位置付
- 範囲
- ソート
- ［コマンド］パレット
- タブ順序

学習目標

本章を学ぶことで、次の内容に対する理解を深めましょう。

- Magic xpa でデータソースを定義する方法
- Magic xpa の APG（自動プログラム作成機能）の使用方法
- 実行時のデータ操作方法
- スクリーンモードとラインモードのフォーム表示について
- ［テーブル］コントロールと、テーブルへの項目追加方法
- 実行時のデータ位置付について
- データビューへの範囲設定について
- 実行時のデータソートについて
- ［コマンド］パレットのアイコンについて
- タブ順序の設定について

参照

Magic ヘルプの［Magic xpa リファレンス／データソース，［データビュー］エディタ，エンドユーザ機能，［フォーム］エディタ，表示フォームとコントロール］の各セクション

１.［データベース］テーブル

データソース定義の最初のステップは、データソースの配置されるデータベースを定義することです。

Magic xpa は、Pervasive(Btrieve)、MSSQL、Oracle などの複数の DBMS と接続することができます。

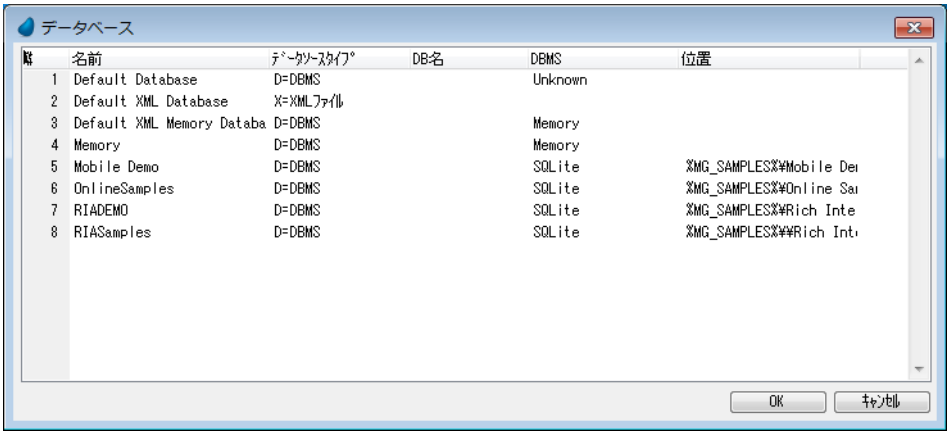
SQLite との接続環境も定義されていますが、デモアプリケーション用のためサポート 対象外になっています。

［データベース］テーブルには、現在の Magic xpa インストレーション環境で使用できる物理的なデータベース情報の詳細が登録されています。

［データベース］テーブルには、異なるデータベースあるいは同じデータベースの複数のエントリを定義することができます。各エントリには、DBMS 情報、データベース名、ユーザ名、パスワード、その他の追加情報など、Magic xpa がそのデータベースに実際に接続するときに必要となる情報が定義されます。

デフォルトの状態として、以下のようなデータベースが登録されています。

- Default Database …… デフォルトで選択されるデータベースです。新しくデータソースを作成したときに、特にデータベースを指定しないと、これが選択されます。
- Default XML Database …… 新しく XML データソースを定義したときにデフォルトで選択されます。
- Memory …… メモリーテーブルが必要な場合、開発者または Magic xpa が一時的に保持する内部使用目的で選択します。
- Default XML Memory Database …… Default XML Database の一時データベースとして使用されるメモリデータベース。



＜ Magic_xpa 入門＞データベースの定義

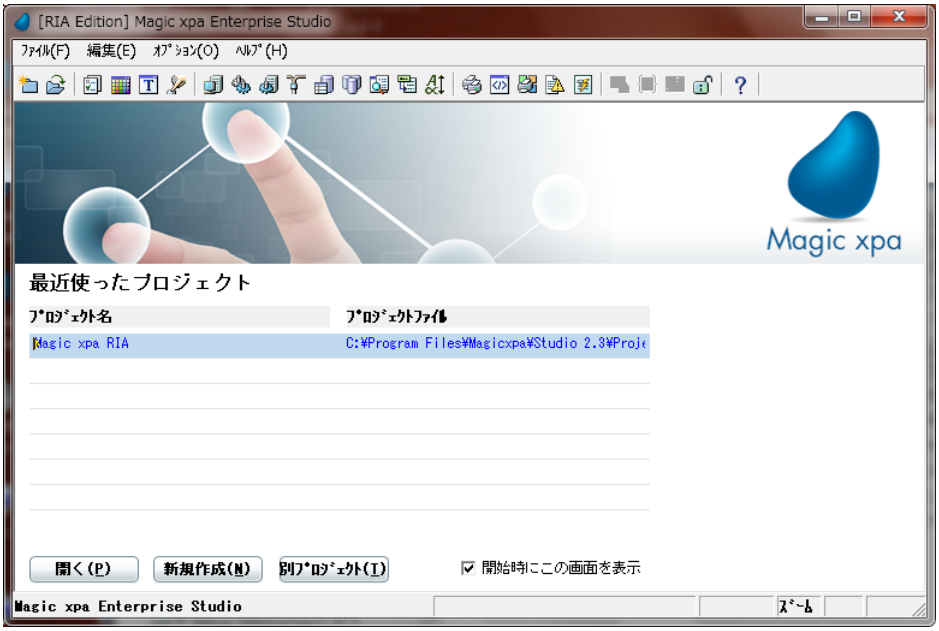
本コースのプロジェクトは、MSSQL（MicrosoftSQL Server）DBMS を使用して開発します。

本コースをインストールすると、＜ Magic_xpa 入門＞データベースと、コースに関連するテーブルが MSSQL に作成されることを前提としています。

それらのテーブルにアクセスするには、予め［データベース］テーブルで MSSQL に接続するための定義をしなければなりません。

［データベース］テーブルを開くには、次のようにします。

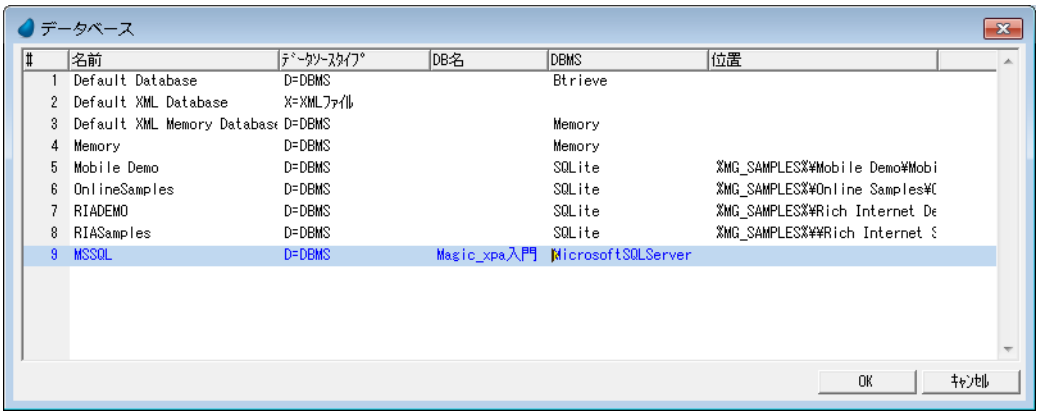
1. すべてのプロジェクトを閉じた状態にします。プロジェクトが開いているときは、［ファイル］メニューから［プロジェクトを閉じる］を選択します。
2. ［オプション］メニューから［設定／データベース］を選択し、［データベース］テーブルを開きます。



3. 最後の行にパークします。
4. F4 キーで一行作成します。
5. 次のようにパラメータを設定してください。

#	名前	データソースタイプ	DB 名	DBMS	位置
9	MSSQL	D=DBMS	Magic_xpa 入門	MicrosoftSQLServer	

DBMS の選択は、［DBMS］カラムからズームして DBMS 一覧を表示し、そのなかから選択します。



［データベース］テーブルについて

カラム	説明
#	エントリーの識別番号。Magic xpa によって自動的に割り当てられます。
名前	物理データベースの名前
データソースタイプ	利用するデータソースのタイプを次のなかから指定します。 • D=DBMS … データベース用 • X=XML ファイル … XML データソース用
DB 名	接続する DB 名を設定します。

カラム	説明
DBMS	利用するデータベースを指定します。 この欄からズームして DBMS 一覧を表示させ、選択します。
位置	この欄はオプションで、ISAM データベースまたは XML ファイルの場合のみ有効です。実際の物理的位置情報を記述します。

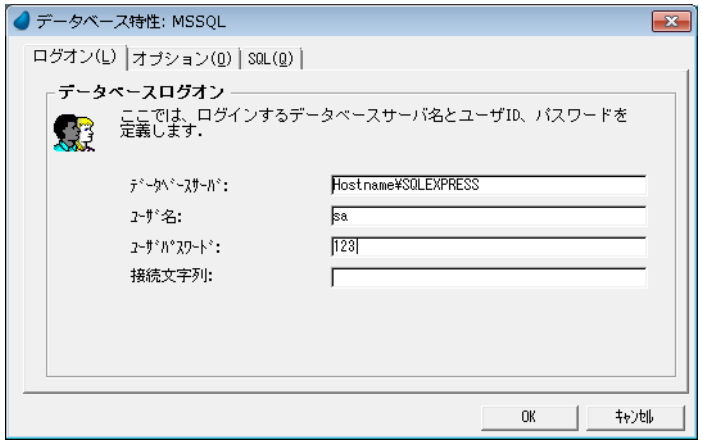
データベース特性の設定

［データベース特性］ダイアログには 3 つのタブがあります。

- ［ログオン］タブ …… データベースにログオンするときの情報（たとえばユーザ名、パスワード）を設定します。
- ［オプション］タブ …… ロック処理に関する［データベース］テーブルの処理や、テーブル構成について設定します。
- ［SQL］タブ …… SQL データベースに接続するための追加情報を設定します。

最低限設定が必要なものは、データベースに接続するときの［ユーザ名］と［ユーザパスワード］です。

6. ＜ MSSQL ＞の［データベース特性］ダイアログ（Alt+Enter）を開きます。
7. ［データベースサーバ］を入力します。（たとえば、＜ HostName¥SQLEXPRESS ＞）
8. ［ユーザ名］を入力します。（たとえば、＜ sa ＞）
9. ［ユーザパスワード］を入力します。（たとえば、＜ 123 ＞）



データベースサーバのインストールを行うときに、ユーザ 'sa' に対するパスワードの設定を行ってください。詳細についてはデータベースアドミニストレータの方などにご確認ください。

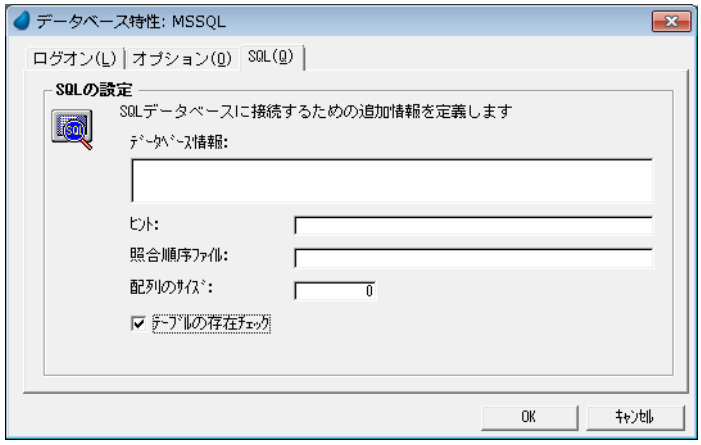
テーブルの存在チェック

データベースのデータを表示させるプログラムを実行するとき、Magic xpa は RDBMS に対して要求されたデータを返すようにリクエストを送ります。もし RDBMS に該当するテーブルが存在しないとき（空のテーブルという意味ではありません）、RDBMS からはエラーメッセージが返されます。

このような場合、Magic xpa が［データ］リポジトリの情報に基づいてテーブルを自動的に作成するようにできます。Magic xpa はテーブルの存在をチェックし、必要であればそのテーブルをデータベースに作成します。（テーブルの存在チェックとテーブルの自動作成は、プログラムの実行時に一回だけ起こります。）

［データベース特性］ダイアログの SQL タブにある、［テーブルの存在チェック］チェックボックスにチェックすると、この機能が有効になります。

10. ［SQL］タブをクリックします。
11. ［テーブルの存在チェック］チェックボックスにチェックします。
12. ［OK］ボタンをクリックし、［データベース特性］ダイアログを終了します。



２．データソース定義

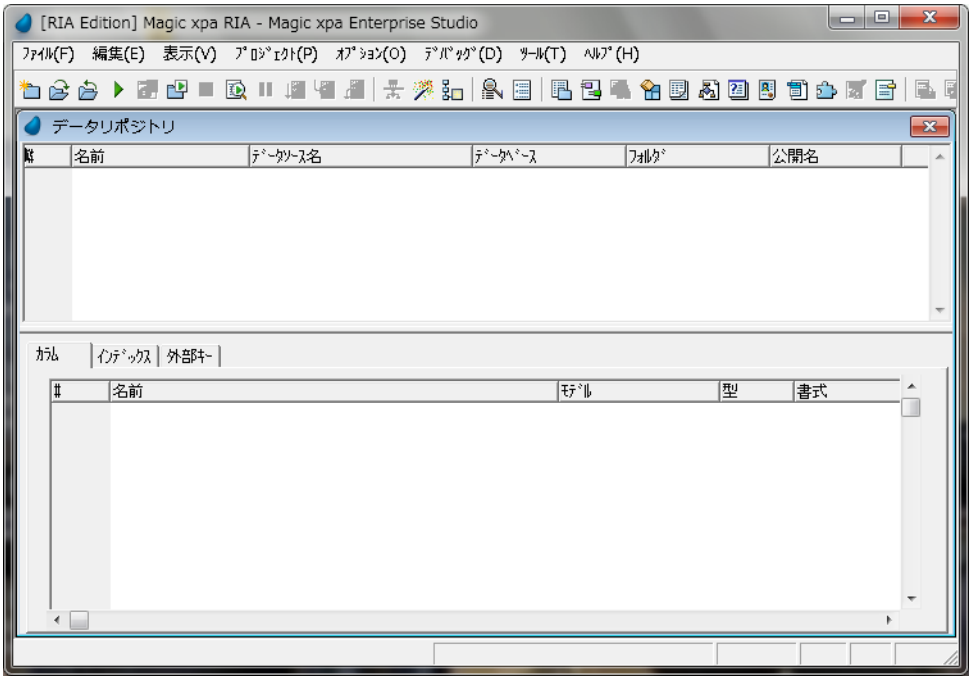
次のステップは、データソースの定義です。データソース定義は次のような手順で行います。

- データソースヘッダの定義
- カラムの定義
- インデックスの定義
- 外部キーの定義

本コースでは［外部キー］は取り扱いません。

［データ］リポジトリは 2 つのセクションに分かれています。

- 上部ペインには、データソースの基本パラメータとして、名前、データソース名、データベース、フォルダ、公開名が定義してあります。
- 下部ペインには、3　つのリポジトリ、すなわちカラム、インデックス、外部キーを定義します。下部ペインの表示内容は、上部ペインで選択されるデータソースに応じて変わります。

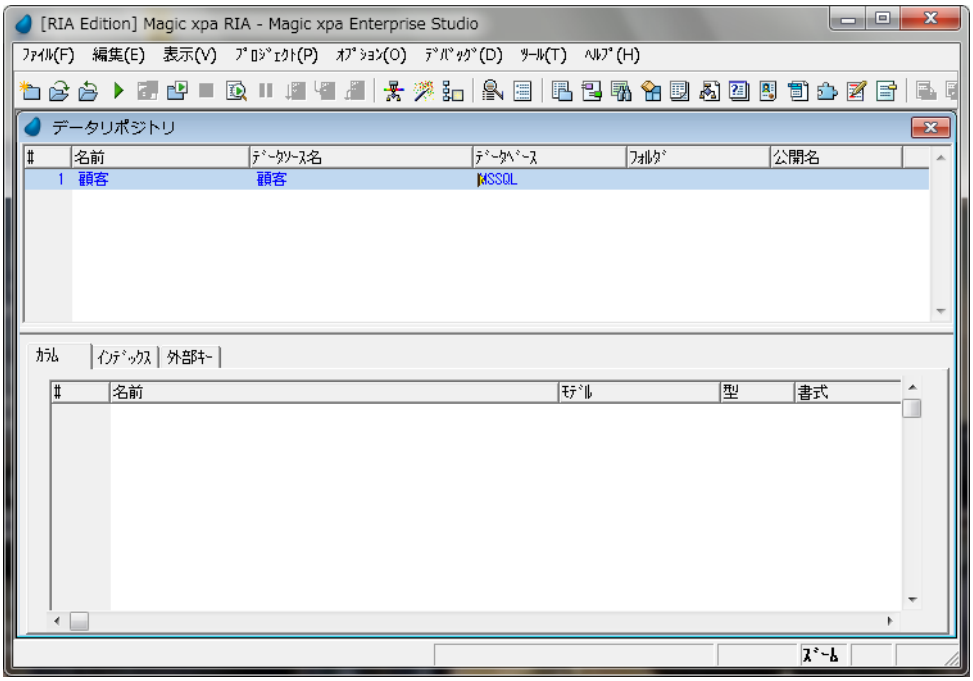


＜顧客＞データソースの定義

それでは、これから＜顧客＞データソースを定義しましょう。

1. ［プロジェクト］メニューから［データ］（Shift+F2）を選択し、［データ］リポジトリを開きます。
2. F4 キーで一行作成します。
3. 次のようにパラメータを設定します。

#	名前	データソース名	データベース	フォルダ	公開名
1	顧客	顧客	MSSQL		



［データ］リポジトリについて

［データ］リポジトリには、プロジェクトで使用する全てのデータソースが定義されています。

各データソースは、関連するカラム、インデックス、および外部キーによって定義されます。

［データ］リポジトリのカラム

カラム	説明
#	各エントリのユニークな識別番号で、Magic xpa が自動的に生成します。
名前	データソースの名前を入力します。この名前は Magic xpa 内部で使用され、たとえばエラーメッセージなどの表示で使用します。
データソース名	データベース内のデータソースの名前を入力します。
データベース	［データベース］テーブルに定義されたデータベースを指定します。［データベース一覧］にズームして選択することもできます。　データベースを選択することにより、データソースのある場所と、ログイン方法、データソースへの追加情報などを決定することになります。
フォルダ	フォルダを定義しておくことで、所属するフォルダを選択できるようになります。フォルダを利用し、データソースをグループ化できます。
公開名	データソースを外部から利用する場合の公開名を入力します。公開名はプロジェクトファイル内でユニークでなければなりません。

カラムの定義

［カラム］テーブルでは、データソースのカラムに関する定義を行います。各データソースはそれぞれに対応する［カラム］テーブルを持っています。データソースの各カラムについては、プログラムのデータビューで変数項目の定義を行ったのと同様の方法で定義します。各カラムにおいて、名前、モデル、型、書式を設定します。

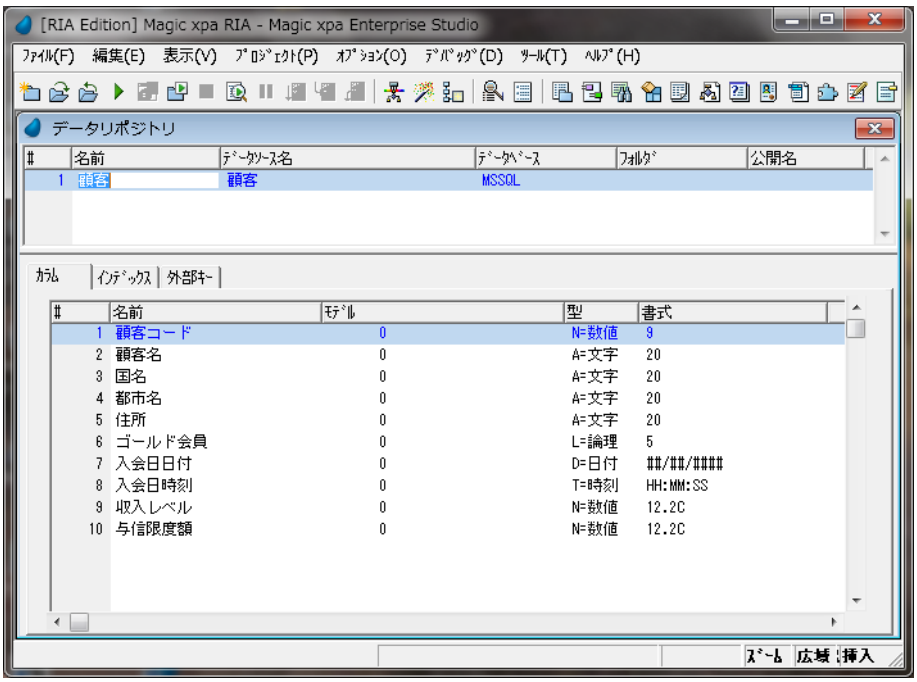
4. 上部ペインの＜顧客＞データソースにパークします。
5. 下部ペインの［カラム］タブをクリックします。

［カラム］テーブルで、以下の手順を繰り返し、エントリを追加してください。

6. F4 キーで一行作成します。
7. カラムを次の表に示すように設定してください。

#	名前	モデル	型	書式
1	顧客コード	0	N= 数値	9

#	名前	モデル	型	書式
2	顧客名	0	A= 文字	20
3	国名	0	A= 文字	20
4	都市名	0	A= 文字	20
5	住所	0	A= 文字	20
6	ゴールド会員	0	L= 論理	5
7	入会日日付	0	D= 日付	####/##/##
8	入会日時刻	0	T= 時刻	HH:MM:SS
9	収入レベル	0	N= 数値	12.2C
10	与信限度額	0	N= 数値	12.2C



インデックスの定義

次に、＜顧客＞データソースにユニークなインデックスを定義してみましょう。

8. 下部ペインの［インデックス］タブをクリックします。
9. F4 キーで一行作成します。
10. 次のようにインデックス名とタイプを定義します。（次のセクションでインデックスセグメントを定義します。）

#	名前	タイプ	プライマリキー
1	顧客コード	U= 重複不可	

本コースでは［プライマリキー］は取り扱いません。

［インデックス］テーブルについて

［インデックス］テーブルでは、重複不可または重複可のインデックスを定義することができます。

カラム	説明
#	各エントリの識別番号で、Magic xpa が自動的に作成します。
名前	インデックスの名前を入力します。

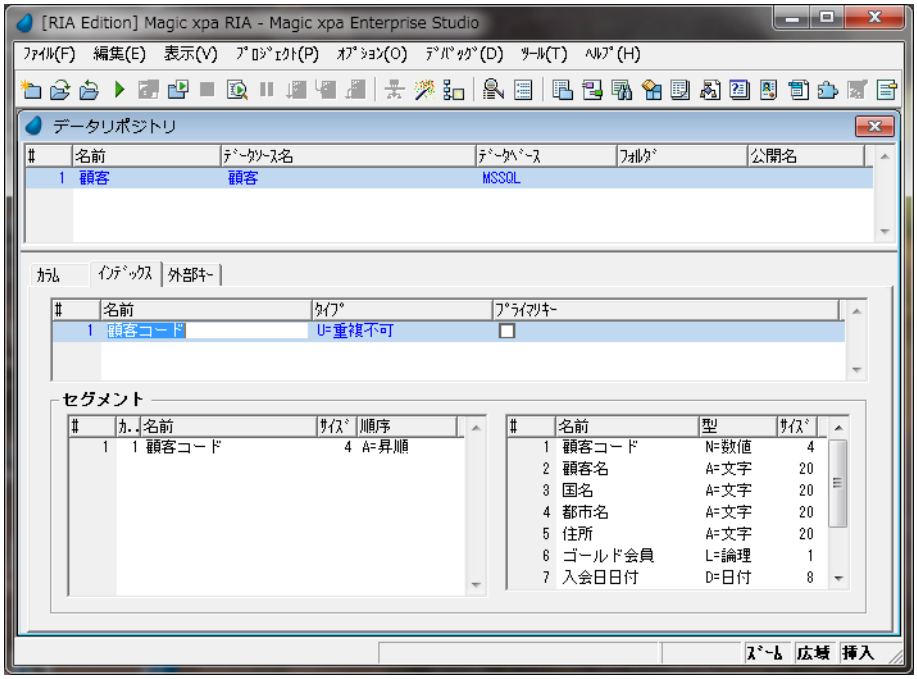
カラム	説明
タイプ	以下のどちらかを指定します。 • U= 重複不可……ユニークなインデックス値のみ入力できます。SQL ゲートウェイの場合、少なくとも 1 つの重複不可インデックスが必要な場合があります。 • N= 重複可……重複した値をインデックス値として入力できます。
プライマリキー	どのインデックスがプライマリキーかを指定します。 ユニークなインデックスのみがプライマリキーとなることができます。

インデックス定義を完了するためには、インデックスセグメントを定義しなければなりません。

- **定義されたテーブルがデータベース内に存在しない場合、Magic xpa は［データ］リポジトリの定義内容に基づいて、そのテーブルとインデックスをデータベース内に作成します。**
- **データベース内にすでに存在しているテーブルが定義された場合、そのテーブル定義はインデックス定義も含めて既存のものの定義内容と一致しなければなりません。もし間違ったインデックス定義でそのテーブルにアクセスすると、レスポンスが非常に悪くなる場合があるので、このことは特に重要です。**

インデックスセグメントの定義

11. インデックス名＜顧客コード＞から F5 キーで［セグメント］テーブルにズームします。
12. ［セグメント］テーブルに F4 キーで一行作成します。
13. ［カラム］カラムからズームし、カラム一覧に移動します。
14. ＜顧客コード＞エントリにパークします。
15. Enter キーを押下して＜顧客コード＞カラムを選択します。（＜顧客コード＞が［セグメント］テーブルにインデックスセグメントとして表示されます。）
16. Enter キーを 2 回押下することで、上部ペインに戻ります。



これで＜顧客＞データソースの定義は完了です。

３．＜顧客＞プログラムの生成

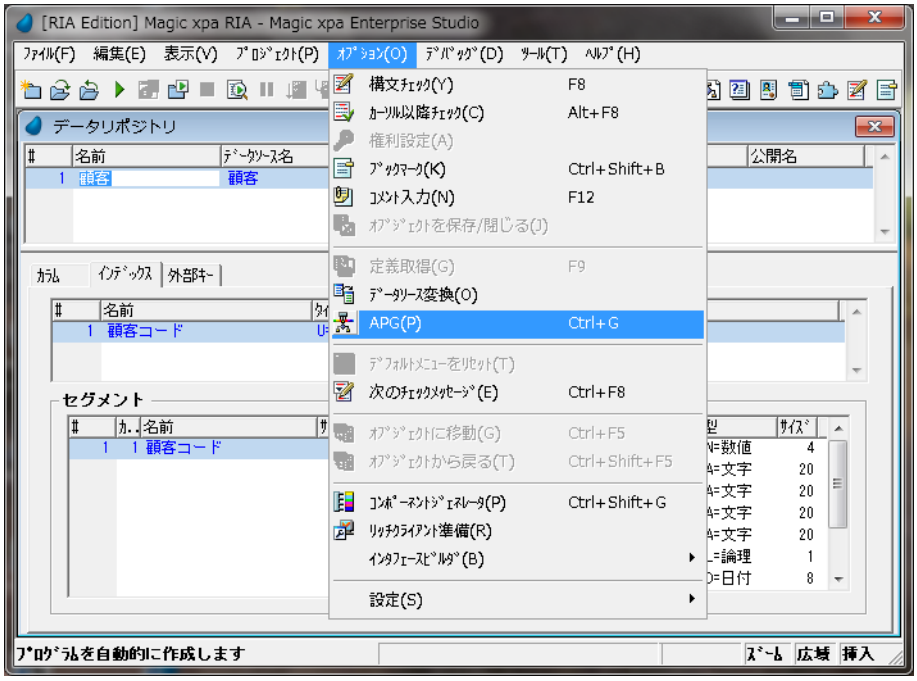
テーブルデータ内容の表示と操作（登録、修正、削除など）は、プログラムにより行われます。

本コースのこれまでの学習では、変数項目の表示と操作を行うプログラムを作成しました。

このセクションでは、APG 機能を使用して、＜顧客＞テーブル内容を表示し操作するプログラムを作成してみましょう。

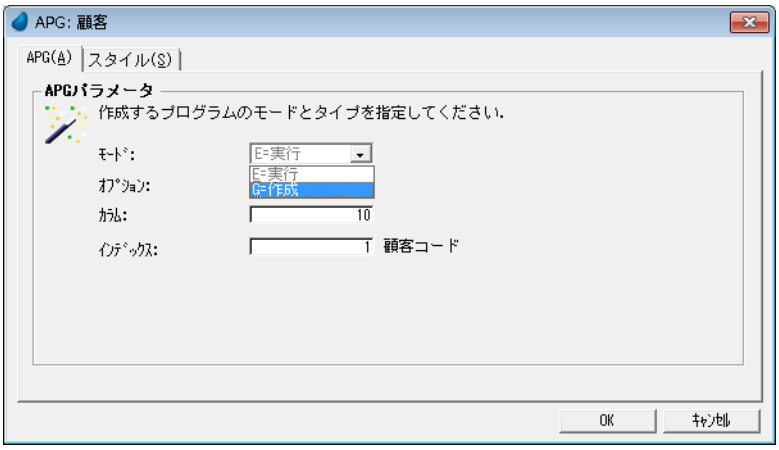
［データ］リポジトリにおいて、

1. ＜顧客＞データソースにパークします。
2. ［オプション］メニューから［APG］（Ctrl+G）を選択します。



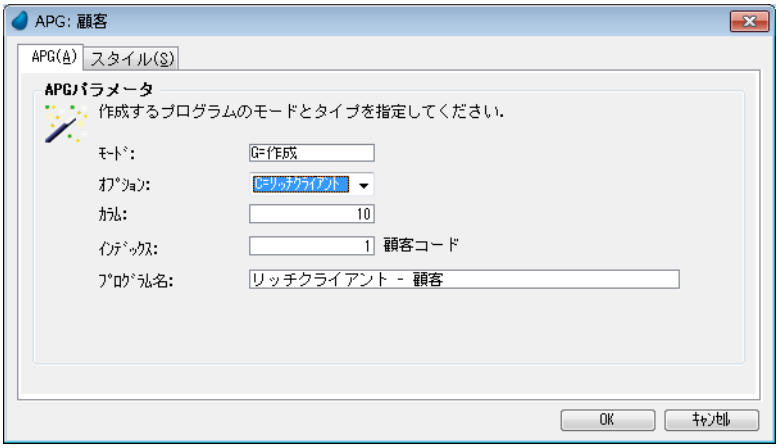
［APG］ダイアログの APG パラメータのうち、

3. ［モード］欄で［G= 作成］を選択します。

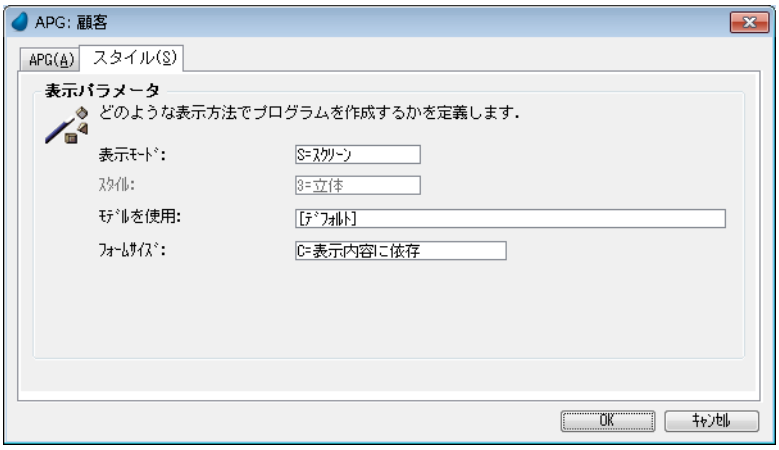


APG パラメータ

パラメータ	説明
モード	結果プログラムが一時的なものか、保存されるものかを指定します。 オプションとしては、次の二つがあります。 • E= 実行 …… 選択されているデータソースに対し、一時的なプログラムを生成すると同時に実行します。 • G= 作成 …… 選択されているデータソースに対するプログラムを生成し、［プログラム］リポジトリに追加します。プログラムは自動的には実行されません。
オプション	プログラムの機能を指定します。 モードが G= 作成のとき指定できる機能には次のものがあります。 • B ＝照会 …… オンラインのデータ入力／保守プログラムを実行（実行モード）したり、作成（作成モード）します。RIA Edition では選択できません。 • E ＝出力 …… 現在、選択しているデータテーブルのデータを出力するプログラムを作成、実行します。データは、テキストファイルで出力されます。 • I ＝入力 …… テキストファイルからデータを取り込むためのプログラムを作成、実行します。読み込むファイル構造は、［E= 出力］で出力したファイル構造と同じにしなければなりません。 • P ＝印刷 …… データテーブルの内容を印刷するプログラムを作成、実行します。 • C ＝リッチクライアント …… リッチクライアントベースでのプログラムを作成します。このオプションは、［データ］リポジトリから実行した場合は、無効です。
カラム	現在選択しているデータソースのカラム数が表示されます。 ［カラム選択］テーブルの 3 番目の［カラム］カラムの順番を変更することができます。［0］を入力することで表示させないようにできます。
インデックス	プログラムのメインソースで使用するインデックスを指定します。
プログラム名	［プログラム］リポジトリに登録するプログラム名を入力します。



4. ［スタイル］タブをクリックします。
5. ［表示モード］パラメータで、［S= スクリーン］を選択します。
6. ［OK］をクリックしてプログラムを生成します。

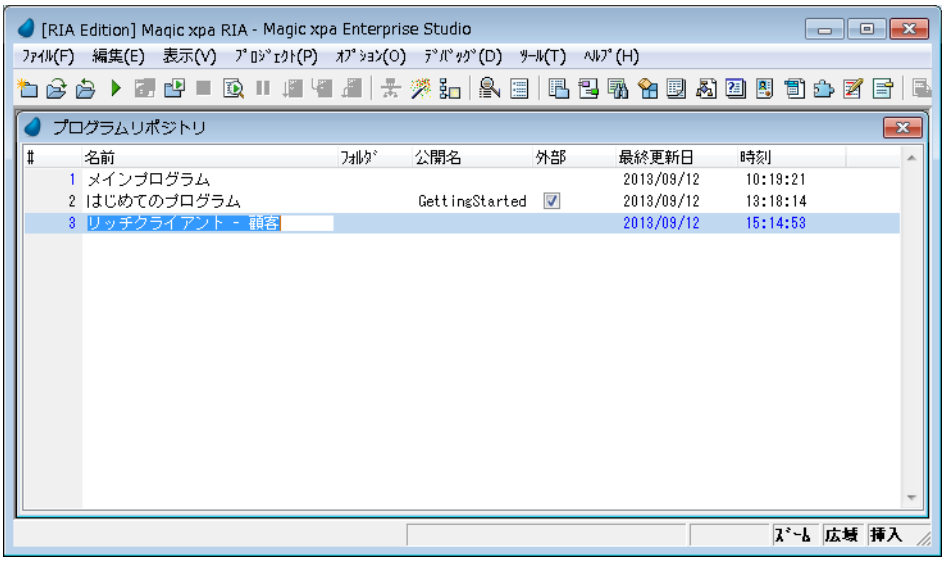


スタイルパラメータ

パラメータ	説明
表示モード	プログラムのオンライン画面へのレコード表示形式で、2 つのオプションがあります。 • L= ライン …… レコードが行として表示されます。各行は 1 レコードに相当します。 • S= スクリーン …… 画面ごとに 1 つのレコードが表示されます。
スタイル	このパラメータは設定できません。
モデルを使用	既存のモデルを割り当てることができます。
フォームサイズ	APG によって生成される RIA フォームのサイズを指定し、次のオプションがあります。 • M ＝モデルに依存 …… フォームの幅と高さは指定したモデルで定義されます。 • C ＝表示内容に依存 …… フォームの幅と高さは画面の表示内容に依存します。
タイトル	フォーム上にタイトルを表示するかどうかを指定します。

APG は＜顧客＞データソースを基に新しくプログラムを生成し、［プログラム］リポジトリに追加します。

7. ［プログラム］リポジトリ（Shift+F3）を開きます。
8. プログラム＜リッチクライアント－顧客＞が追加されていることを確認してください。



タスクモード

実行時のタスクモードはデータに対して行うことのできる操作を制限します。

タスクモード

実行モード	可能な処理
C= 登録	新しいレコード（行）の作成。登録モードでは、そのときに作成されたレコードの情報だけが表示されます。他のレコード情報を見るためには照会または修正モードに切り替える必要があります。
M= 修正	既存のレコードの内容を変更します。修正モードでは新たにレコードを追加することも可能です。これを［修正モードでの登録］と呼びます。
Q= 照会	修正や削除を行わずに、レコード全体をスキャンして表示します。
D= 削除	レコードの削除。修正モードの機能の一部です。このモードはステータス行には表示されません。

開発者はタスクの初期モードと、エンドユーザが使用できるモードを指定することができます。

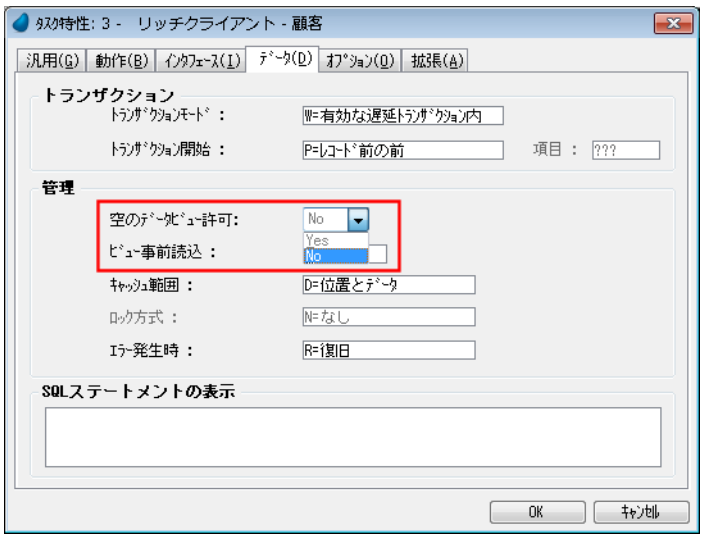
エンドユーザは許可されたタスクモードを実行時に切り替えて使用することができます。

たとえば、初期モードを照会に設定した場合、プログラムの起動時には照会モードとなりますが、修正モードや登録モードが許可されているときには、エンドユーザはそれらのモードに実行中に切り替えることができます。

タスク特性の確認

ここで、APG で作成したプログラムのタスク特性を確認しておいてください。

1. 今回作成した＜照会 － 顧客＞プログラムにカーソルを置きます。
2. F5 キーを押下してください。プログラムが開きます。
3. Ctrl+P キーを押下してください。［タスク特性］ダイアログが開きます。
4. ［データ（D)］タブをクリックしてください。



5. ここで、「空のデータビュー許可」というパラメータが＜ No ＞になっていることを確認してください。もし ＜ Yes ＞になっていたら、＜ No ＞に直してください。
6. ［Enter］キーを 2 回押下して、［プログラム］リポジトリまで戻ります。

このパラメータの設定は、テーブルに対象となるレコードが 1 件もなかった場合の動作に影響を与えます。詳細は F1 キーを押下して、リファレンスヘルプを参照してください。

４．＜リッチクライアント－顧客＞プログラムの実行

このセクションでは、生成されたプログラムを動かし、データの操作方法を学びます。

1. ［プログラム］リポジトリで＜リッチクライアント－顧客＞プログラムの行にパークし、F7 キーを押下して実行させます。

顧客データの登録

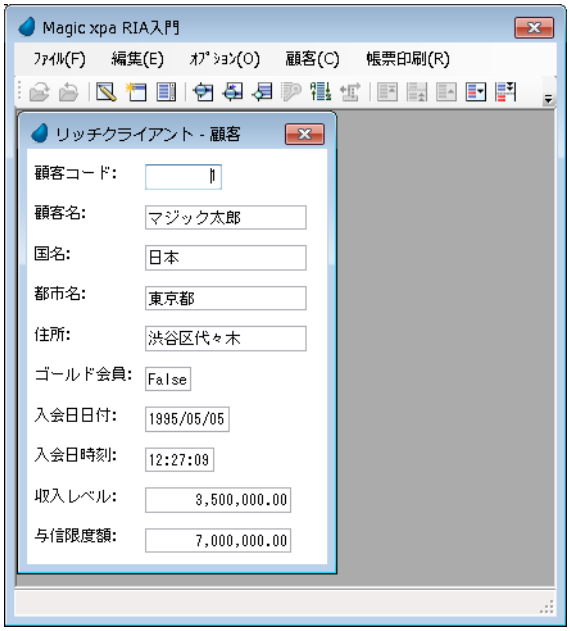
＜リッチクライアント－顧客＞プログラムは、＜顧客＞データソースの内容を表示します。

このプログラムの初期モードは［Q= 照会］ですので、もしデータソースに一つでもレコードがあれば、その最初のレコードが表示されます。

しかし＜顧客＞データソースには最初はデータがないので、Magic xpa はタスクモードを［C= 登録］に自動的に変更し、エンドユーザのデータ登録を促します。

＜リッチクライアント－顧客＞プログラムは一つのレコード情報のみを表示しています。Magic xpa ではこの表示モードを［スクリーンモード］と呼びます。

2. 顧客データを適当に入力しましょう。（イメージは参考入力です。）



このスクリーンモードでのデータナビゲーションについて次に説明します。

項目間のカーソル移動

Magic xpa では次のようにしてフォーム上の項目間のカーソル移動を行います。

スクリーンモードでは下記の方法が利用できます。

- 特定の項目をクリック
- ［Tab］キーを押下して、次項目に移動
- ［Shift ＋ Tab］キーを押下して、前項目に移動
- ［↓］キーを押下して、次項目に移動
- ［↑］キーを押下して、前項目に移動

顧客データの追加登録

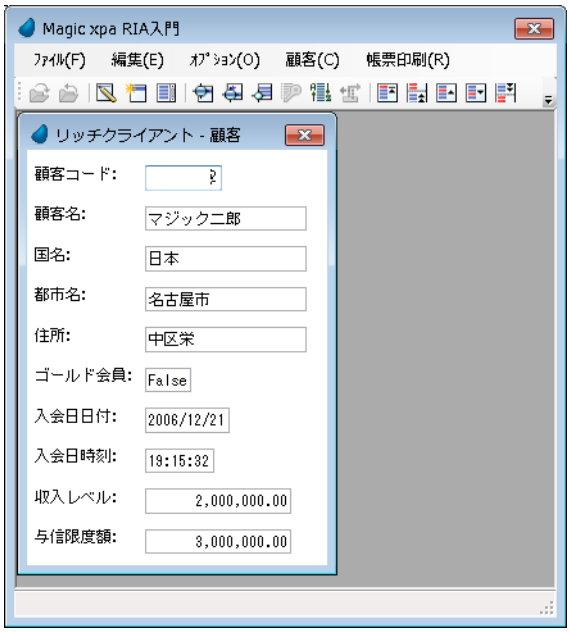
顧客レコードを一つ入力しましたが、まだデータベースには保存されていません。

顧客データの保存は、もう一つ別の顧客データを追加するなど、現在のレコードから他へ移動するアクションを実行したときに行われます。

では次のようにして顧客データをさらに追加してみましょう。

3. PageDown を押下します。

4. 顧客データを適当に追加しましょう。



この状態では、追加登録したデータはまだデータベースに保存されません。このレコードから移動するアクションが実行されていないからです。なお、プログラムを終了すれば、データは保存されます。

レコードを新規に追加登録するとき、ユニークなインデックスキーに重複する値を入力しないよう注意しなければなりません。重複があった場合、[インデックスが重複しています。] というエラーがステータス行に表示され、操作を続けることができなくなります。上記の例では、<顧客コード>が重複不可のインデックスなので、'1' を再び使用することはできません。

顧客データの照会

照会モードでは、レコードの内容を変更することなく、スキャンを行うことができます。

先ほどの操作で登録した顧客データを、照会モードに切り替えて、表示させてみましょう。

照会モードに切り替えることにより、Magic xpa はレコードデータを保存し、同じレコードを照会モードで表示します。

1. Ctrl+Q を押下します。
2. 顧客データ間を次のキー操作を行って移動してみましょう。

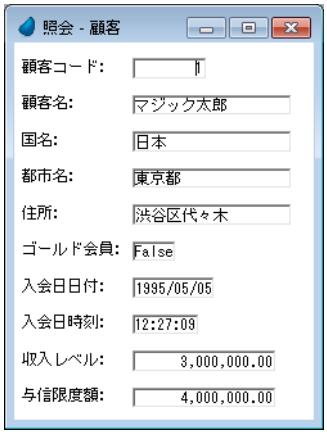
移動キー	説明	[オプション] メニュー
PageDown	次レコードへ移動	次画面
PageUp	前レコードへ移動	前画面
↑	前の項目へ移動	
↓	次の項目へ移動	
Ctrl+Home	最初のレコードへ移動	テーブル先頭
Ctrl+End	最後のレコードへ移動	テーブル末尾

モーダルウィンドウで作成されたプログラムでは、MDI に表示されている [オプション] などのプルダウンメニューにアクセスできません。この場合の操作は、キーボードで行ってください。

顧客データの修正

このセクションでは、修正モードでの操作について学びます。

修正モードではデータの編集と削除を行うことができます。



1. 最初のレコード（Ctrl+Home）へ移動します。
2. Ctrl+M を押下します。
3. ＜入会日日付＞項目にパークします。
4. 日付を＜ 1975/01/31 ＞に変更してください。

この例では＜入会日日付＞を変更しましたが、同様に他の項目も変更することができます。

コントロールの内容編集について

コントロールの内容を編集する方法は、以下のように二通りあります。

- コントロールの内容がマークされている場合

  これはデフォルトの方法です。コントロールにカーソルがパークするとき、コントロールの内容はマークされています（青色で）。

  この場合、入力を開始すると、既存の内容は削除され、新しくデータを入れ直すことになります。

- コントロールの内容がマークされていない場合

  コントロールにパークしているけれども、内容がマークされていない場合の処理です。

  この場合は、既存の内容に対して、書式を超えない範囲で文字を追加したり、既存のデータの一部（または全部）を削除してから追加したりできます。

修正内容のキャンセル

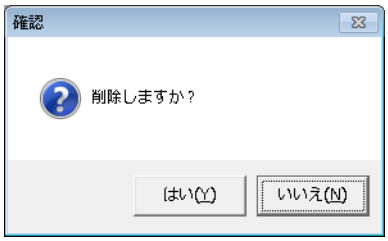
Ctrl+F2 を押下することで、レコードに加えた編集内容をキャンセルすることができます。

顧客データの削除

［削除］は修正モードの機能の一部として実行されます。F3 キーを押下して、［確認］ダイアログで［はい］を選択することにより実行されます。

ここでレコードを一つ削除してみましょう。

1. 最初のレコードに移動します。
2. タスクモードが［修正］であることを確認しましょう。
3. F3 を押下します。
4. ［確認］ダイアログで［はい］を選択し、削除を実行します。



この操作で最初の顧客データが［データベース］テーブルから削除されました。

これでデータ操作についての一通りの説明は終了です。

次のセクションからは、手動で同じプログラムを作成していきます。そのために現在実行中のプログラムを終了します。実行プログラムを終了すると、Magic xpa の実行エンジンが終了することに留意してください。

実行プログラムの終了方法

- Esc キーを押下する
- プログラムを［閉じる］アイコンをクリックする（×アイコン）

プログラムの開発者は、プログラム終了用の＜終了＞ボタンをプログラムのフォームに定義することもできます。

５．顧客データプログラムの手動作成

前のセクションでは、スクリーンモードのプログラムを APG を使用して自動生成しました。ここでは同じプログラムを手動で作成してみましょう。

通常の場合、Magic xpa のプログラムは三つのステップで作成されます。すなわち、

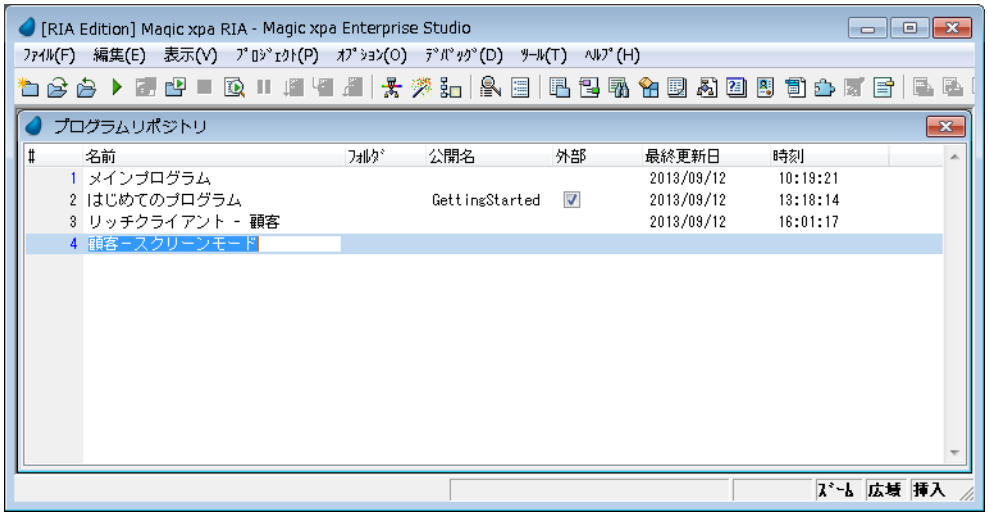
- データビューの定義
- ロジックの定義
- フォームの定義

です。

これから作成する例では、データビューの定義と、フォームの定義だけを行います。Magic xpa は基本的なデータソース内容照会のためのロジックをデフォルトで提供しているため、この例のように単純なプログラムでは、ロジック定義は必要ないのです。

1. ［プログラム］リポジトリ（Shift+F3）を開きます。
2. 最後の行にパークします。
3. 一行作成（F4）します。

4. ［名前］カラムに、**＜顧客ースクリーンモード＞**と入力しましょう。



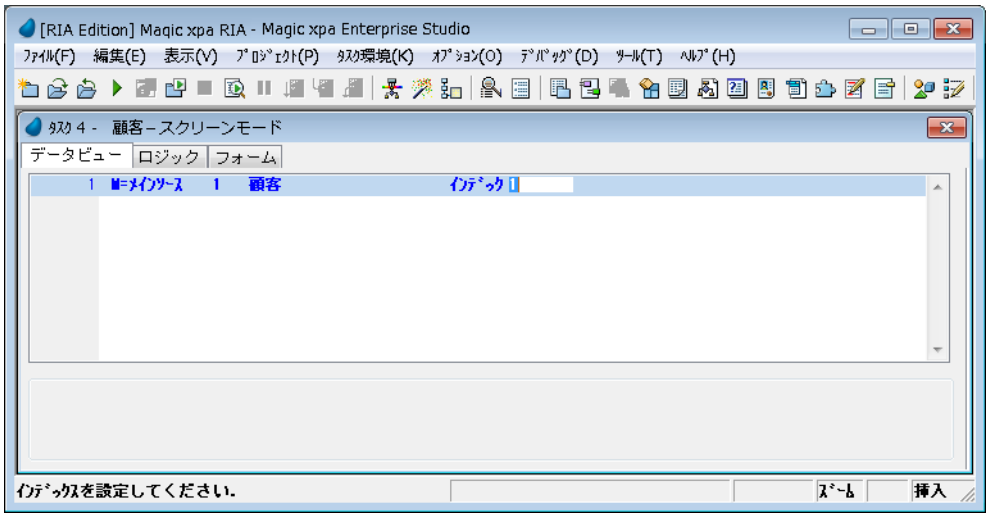
メインソースの定義

［プログラム］リポジトリの［名前］カラムから、

1. ズーム（F5）して、＜顧客ースクリーンモード＞のタスク特性を開きます。
2. タスク特性では、［タスクタイプ］特性を「C= リッチクライアント」に設定して、［OK］ボタンをクリックするか、そのまま Enter キーを押下してダイアログを閉じます。

3. タスクエディタにおいて、［データビュー］エディタを選択します。
4. 最初の行であるメインソース定義ヘッダ行にパークし、次のようにパラメータを設定します。

#	データソース番号	データソースの説明	インデックス番号
1　M= メインソース	1　（注１）	顧客	1　（注２）

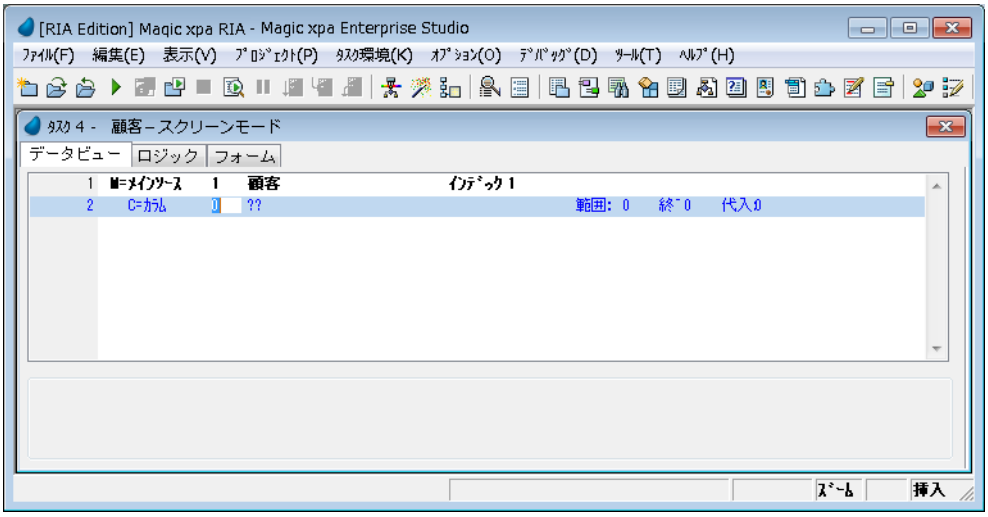


- （注１）：［データソース番号］欄からズームしてデータソース一覧を表示し、最初のエントリ＜顧客＞を選択します。
- （注２）：［インデックス］欄からズームしてインデックス一覧を表示し、最初のインデックス＜顧客コード＞を選択します。

カラムの定義

次に、＜顧客＞データソースからカラムを選択する方法について説明します。

5. 一行作成（F4）します。
6. ［カラム番号］欄からズーム（F5）して、［カラム選択］テーブルを開きます。



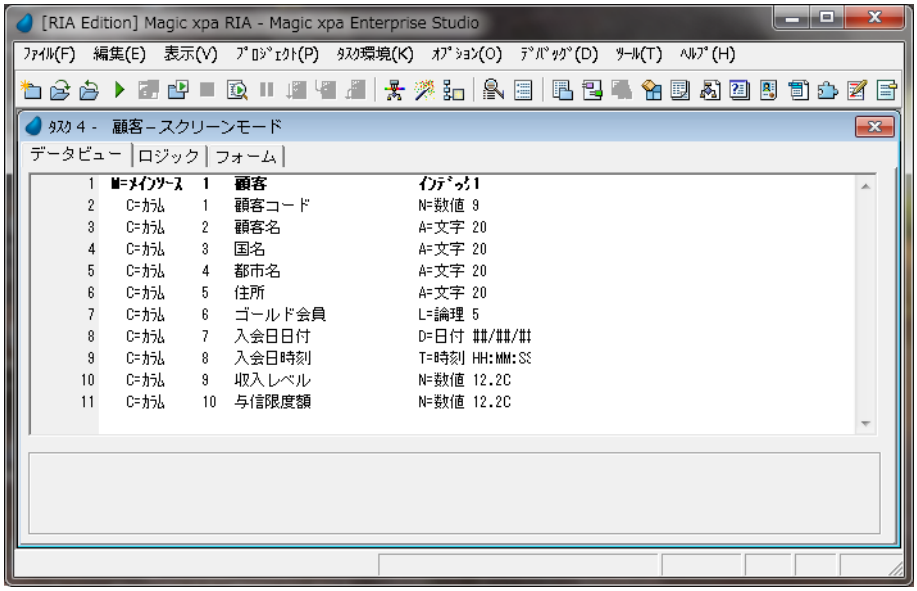
［カラム選択］テーブルで、

7. 次のような方法で、すべてのカラムを選択します。
   a. 最初の行にパークします。
   b. Shift キーを押下した状態にします。

c. 最後の行の # の部分をマウスでクリックするか、'↓' キーを押下し続けて 10 個のカラムをマークします。

8. ［選択］ボタンを押下するか、Enter キー入力します。

すべての選択されたカラムがデータビューに定義されました。



フォームの定義

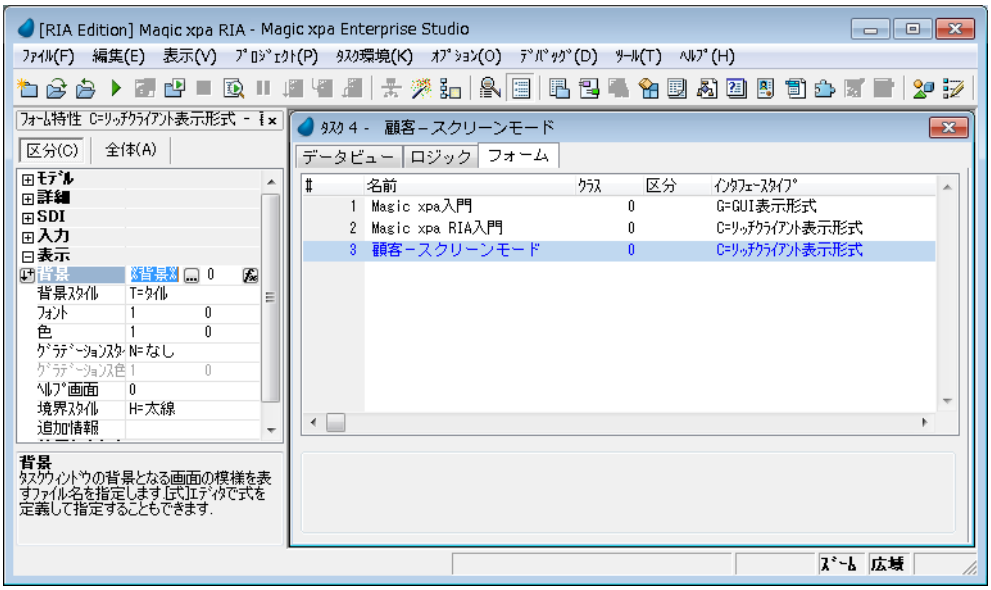
データビューの定義が終わったので、次にフォームのデザインを行いましょう。

最初に、フォームに［背景］を設定します。

1. ［フォーム］エディタを開きます。
2. 2 行目＜顧客－スクリーンモード＞にパークしていることを確認してください。
3. フォーム特性（Alt+Enter) を開きます。

4. ［背景］特性に、**＜％背景％ ＞**と入力してください。

5. いったんフォーム特性を閉じてください。



- **% 背景 % は論理名であることを示しています。**
- **論理名でアルファベットを使用する場合には、大文字と小文字を区別します。２バイト文字を使用する場合は、大文字小文字も区別します。**

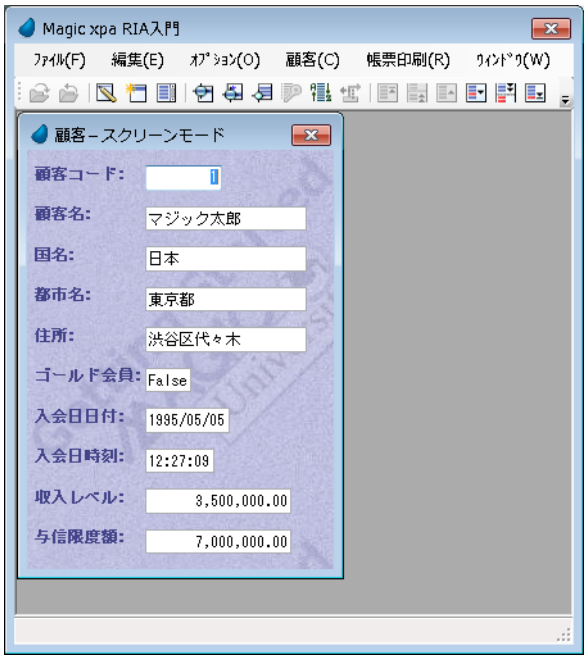
［フォーム］エディタ上で、以下の操作を行います。

6. ＜顧客－スクリーンモード＞のフォームにズーム（F5）します。
7. ［コントロール］パレットの［項目］タブをクリックし、項目パレットを表示させます。
8. 項目パレットから＜顧客コード＞を選択し、フォーム上に配置します。
9. 残りの項目についても同じことを繰り返し、すべてフォームに配置します。
10. ［ラベル］コントロールのフォントをすべて＜見出し文字＞（#11）にします。
11. ［ラベル］コントロールの色をすべて＜見出し文字＞（#5）にします。
12. ［コマンド］パレットの［サイズ調整］アイコンを利用して、＜見出し文字＞（［ラベル］コントロール）のサイズ調整をします。
13. 必要に応じて、表示の重複がないように［エディット］コントロールの位置を調整します。
14. 変更を保存し、プログラムを閉じます。

＜顧客－スクリーンモード＞プログラムの実行

F7 キーを押下して、作成したプログラムを実行しましょう。

今回作成したプログラムの機能は、前のセクションで APG を利用して作成したプログラムと同じであることが分かります。



６．ここまでの要約

ここまでの実習では、［データ］リポジトリにデータソース（テーブル）を一つ作成し、そのテーブル内容を参照するプログラムを二つ作成しました。

最初のプログラムは［APG］機能を利用して自動的に作成し、二つ目は手作業により作成しました。

データソースの作成において、［データ］リポジトリについて学び、二つのペインがあることが分かりました。

- 上部ペイン …… ここではデータソース名と、利用するデータベース情報を定義します。
- 下部ペイン …… ここでは 3 種類の定義を行います。
  - カラム …… データソースのカラム定義、たとえば名前、型や書式などを定義します。
  - インデックス …… データソースのインデックスとそのセグメントを定義します。
  - 外部キー …… 定義テーブルの特定カラムに関連する外部の（テーブルの）インデックスを定義します。これはデータの整合性を維持するために用いられますが、本コースでは取り扱いません。

［APG］を利用してプログラムを自動作成する方法を学びました。

プログラムを実行し、タスクモードの実際について学びました。（登録、修正、照会、削除）

スクリーンモードのプログラムで、レコード間、項目間を移動する操作について学びました。

手作業によるプログラム作成で、メインソースとそのカラムを定義し、タスクフォームに配置しました。

次のセクションでは、1 レコード以上のデータをフォームに表示させ、またそのときのレコード間を移動する方法について学びます。

７．複数レコードの表示

フォーム上にデータを表示する方法を、プログラムの目的に応じて選択しましょう。

- スクリーンモード …… 特定のオブジェクトに注目し、詳細な情報を画面上に最大限表示させたいような場合に使用します。このモードでは画面に表示できるのは 1 レコードだけです。
- ラインモード …… レコードが複数ある場合、レコード検索するためのキーとなるような最低限の情報を高速に表示させたいような場合に使用します。このモードでは画面上に複数レコードを同時表示できます。

前のセクションでは、顧客の詳細データ表示を行うスクリーンモードのプログラムを作成しました。

このセクションでは、ラインモードのプログラムを二通りの方法で作成します。

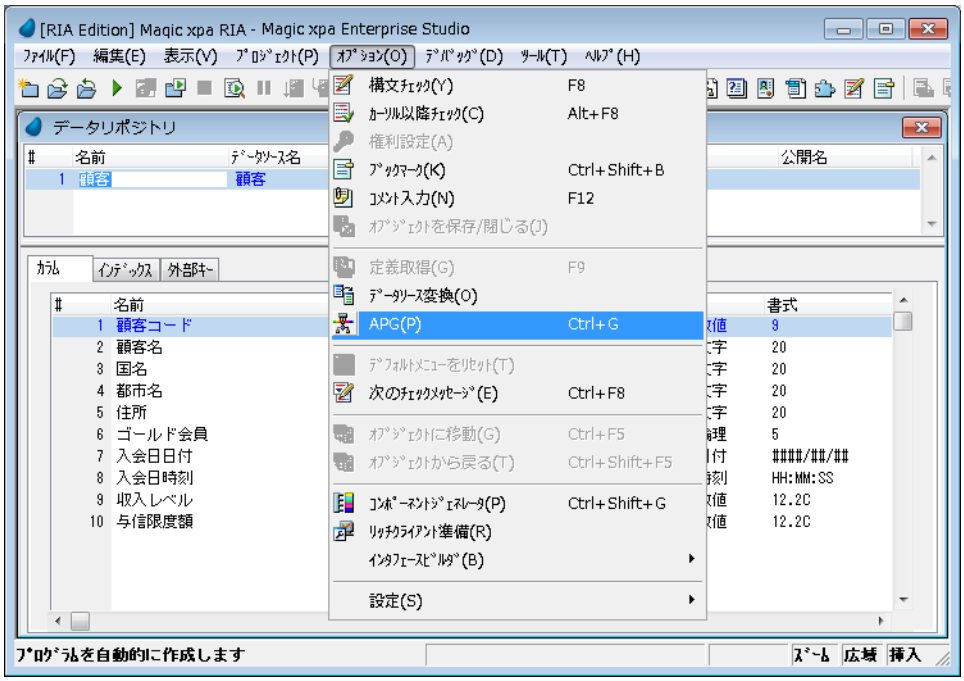
1. APG を利用して、基本的なラインモードプログラムを作成します。
2. ラインモード表示において詳細情報を表示させるソリューションを手動で作成します。

ラインモード表示においては、情報量と表示スペースのバランスを取ることが重要です。多くの場合、一行中に多くの情報を表示させる余裕はありません。その場合、フォームサイズを大きくするか、表示する情報量を減らすのが基本です。上記の２．の実習例においては、参考となるアプローチを紹介します。

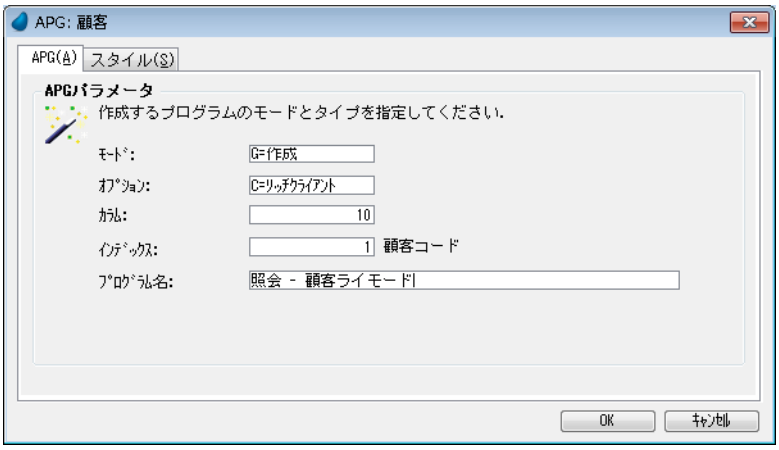
顧客照会ラインモードプログラムの生成

このセクションでは、＜照会－顧客ラインモード＞プログラムを、APG 機能を利用して作成します。

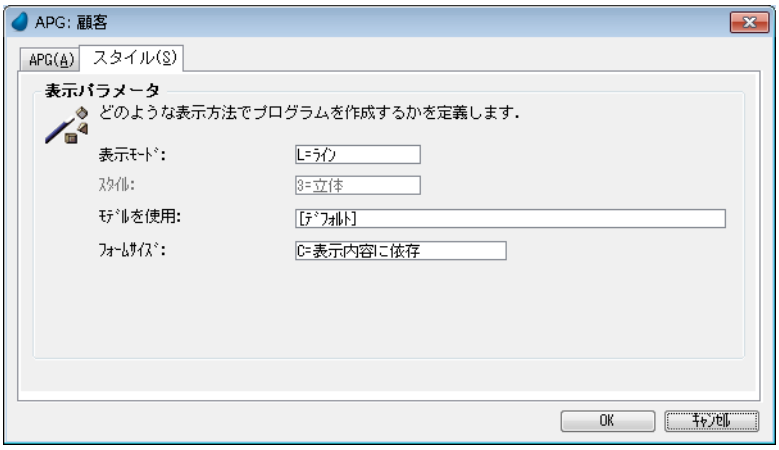
1. ［データ］リポジトリ（Shift+F2）を開き、＜顧客＞データソースにパークします。
2. ［オプション］メニューから［APG］（Ctrl+G）を選択します。



3. モードパラメータを［G= 作成］とします。
4. プログラム名を**＜照会－顧客ラインモード＞**とします。



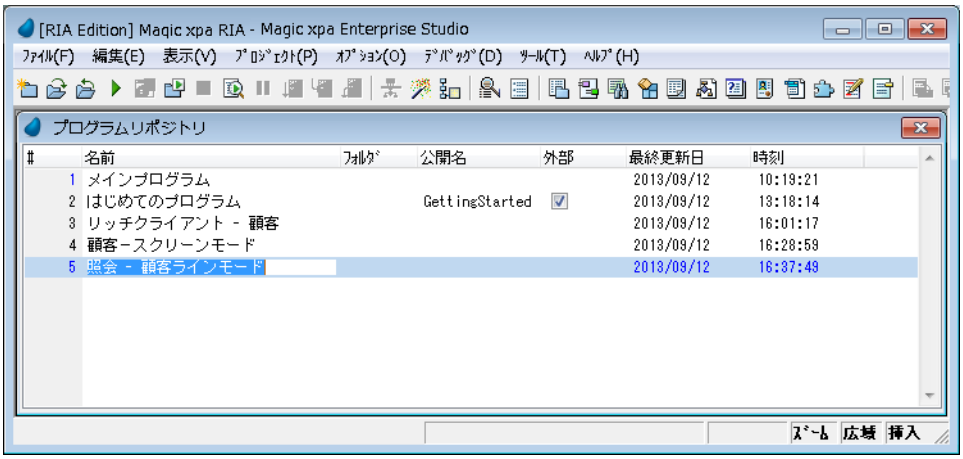
5. ［スタイル］タブをクリックします。（ラインモードがデフォルトで選択されています。）
6. ［OK］をクリックし、プログラムを自動生成します。



APG が＜顧客＞データソースを基にして、新しいプログラムを生成しました。

7. ［プログラム］リポジトリ（Shift+F3）を開きます。

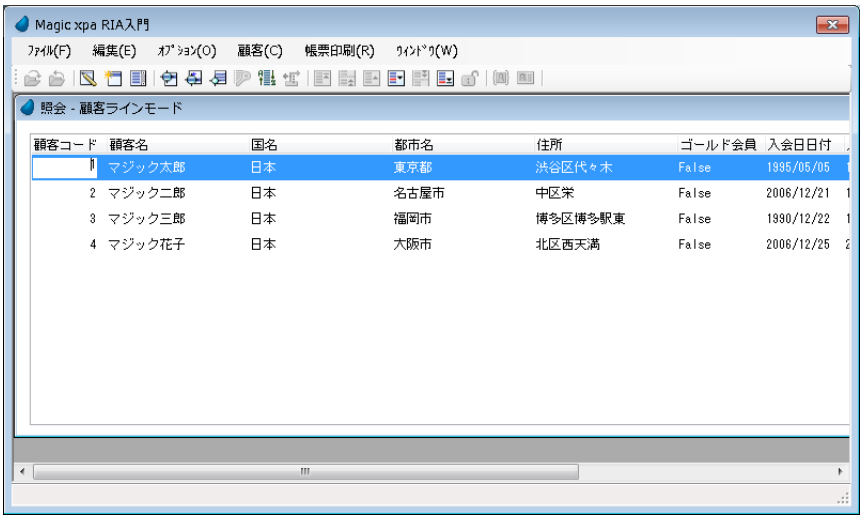
5 行目に、＜照会－顧客ラインモード＞プログラムが生成されています。



８．＜照会－顧客ラインモード＞プログラムの実行

このセクションでは、APG で作成したプログラムを実行させ、ラインモードでのレコード間の移動や、データ操作方法について学びます。

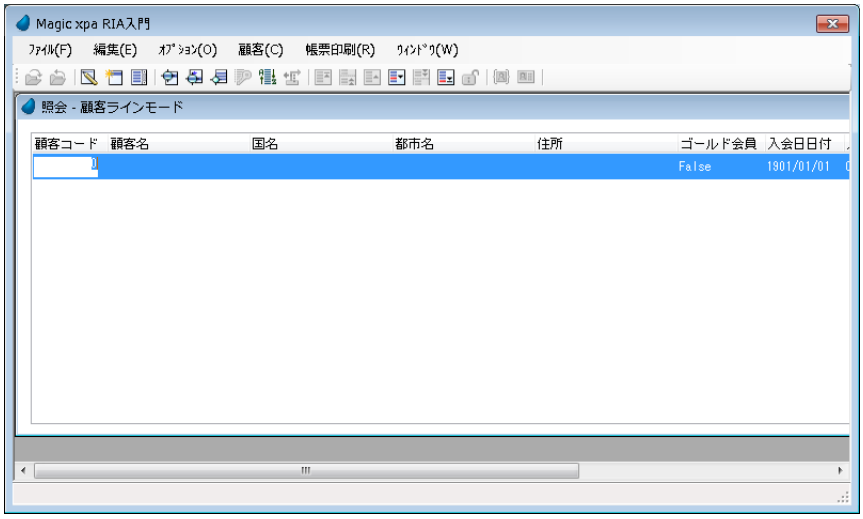
F7 キーを押下して、＜照会－顧客ラインモード＞プログラムを実行します。



レコードがテーブル表示されていることが分かります。1 レコードが一行に対応しています。この例では、レコードはたった一つしかないので、一行が表示されているだけです。テーブルのサイズが大きいので、ウィンドウに入りきらず、スクロールバーが表示される場合があります。

顧客レコードの追加

1. Ctrl+E を押下してください。
2. 新しく顧客を追加し、データを適当に入力しましょう。



カラム幅のサイズ調整

カラム幅は APG 実行時にサイズ調整されて、テーブル内容がすべて隠れることなく表示されるようになっています。

カラムのセパレータにカーソルをパークさせ、ドラッグすることで表示幅を調整することはできますが、その幅の情報はプログラム終了時に保存されません。

照会モードでのレコード間移動

登録モードから照会モードに切り替えることで、入力したデータを保存し、同じレコードを照会モードで表示するようになります。

1. Ctrl+Q を押下します。
2. 下記に示すいろいろな方法で、2 つの顧客データ間を移動してみましょう。

ラインモードにおけるレコード間移動キー操作

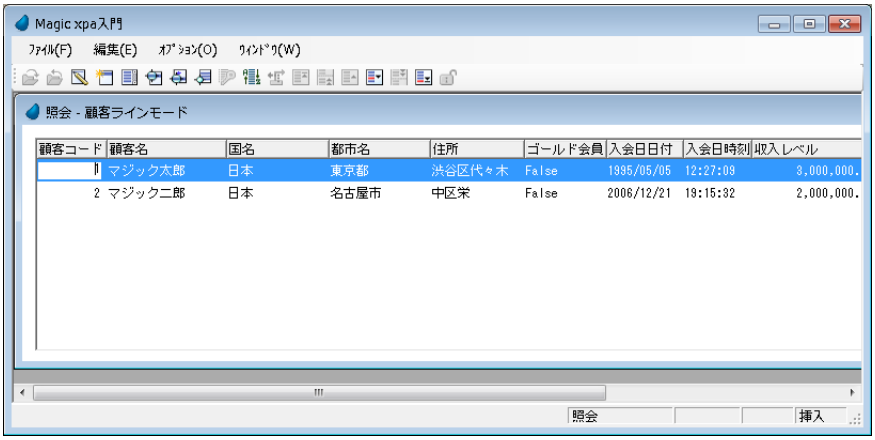
移動キー	説明	オプションメニュー
PageDown	次画面の表示	次画面
PageUp	前画面の表示	前画面
↑	前のレコードへ移動	
↓	次のレコードへ移動	
Ctrl+Home	最初のレコードへ移動	テーブル先頭
Ctrl+End	最後のレコードへ移動	テーブル末尾

顧客データの修正

スクリーンモードと異なり、ラインモードでは同時に複数のレコード情報を見ながら内容を修正できます。

このモードはレコード情報を更新するときに、他のレコード情報が必要となる場合、たいへん有効です。

1. Ctrl+M を押下します。
2. 最初の顧客レコードにパークします。
3. ＜ゴールド会員＞欄を＜ False ／ True ＞にします（設定されている値と異なる値にしてください）。



論理項目の変更

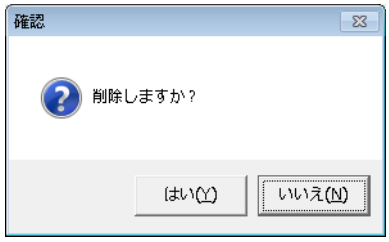
論理値のデフォルトは［False］で、Magic xpa では［True］または［False］のいずれかを設定できます。

論理項目の値を変更するとき、最初の文字（［True］のときは半角の＜ T ＞または＜ t ＞、［False］のときは半角の＜ F ＞または＜ f ＞）を入力するだけでよく、項目を移動すると Magic xpa が自動的に値を設定します。

顧客データの削除

テーブルから顧客データを一つ削除してみましょう。

1. 修正モードであることを確認します。
2. 2 行目の顧客レコードにパークします。
3. F3 を押下します。
4. 削除を実行するかどうかの［確認］ダイアログが表示されるので、［はい］ボタンをクリックするとレコードが削除されます。

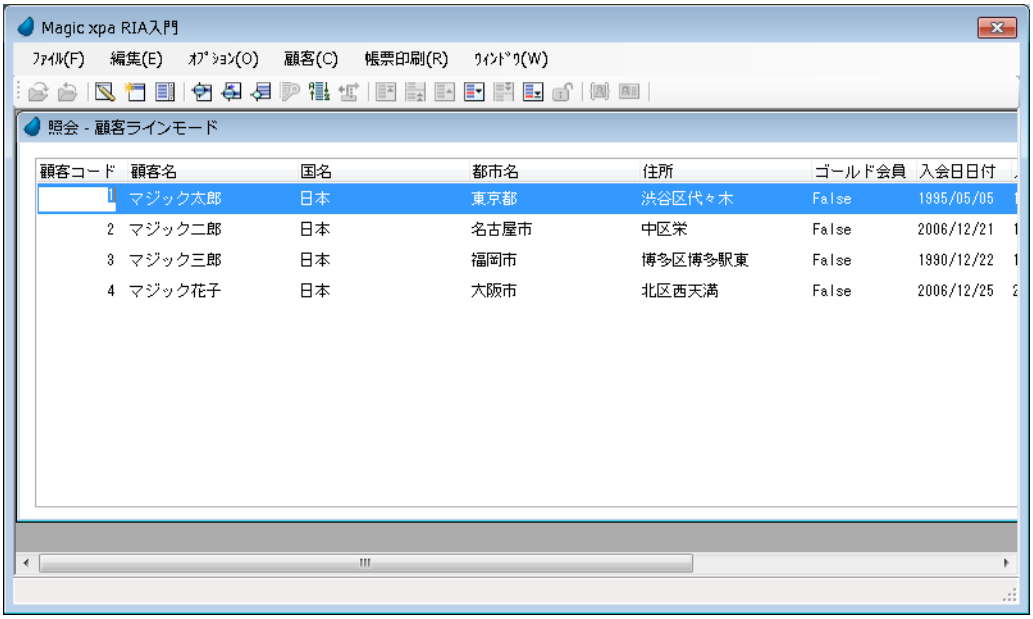


修正モードでのデータ登録

この例では、修正モードでの登録を行うことが可能で、修正モードのまま顧客データソースにレコードを追加できます。

1. 修正モードであることを確認します。もし他のモードのときは修正モードにします。
2. F4 を押下します。
3. タスクモードが［登録］に変わりますが、他のレコード情報はそのまま表示されています。
4. 顧客情報を下図のように入力します。
5. 同様にしてさらに二つのレコードを追加してください。

6. 4 番目のレコードを入力した後、最初の顧客レコードに移動してください。タスクモードは［修正］に変わっています。



データのソート

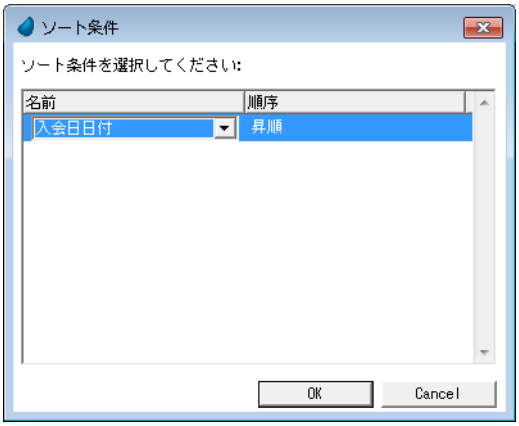
データソースの内容は、［データビュー］エディタで選択したインデックスにしたがってソートされています。Magic xpa は実行の途中で、エンドユーザがデータをソートし直す機能を提供しています。

方法は以下の通りです。

- ［ソート条件］ダイアログに一時的なソート条件を設定してソートします。

1. Ctrl+T を押下します。
2. ［ソート条件］ダイアログの［名前］カラムで「入会日日付」を選択します。
3. ［順序］カラムで昇順か降順かを選択します。

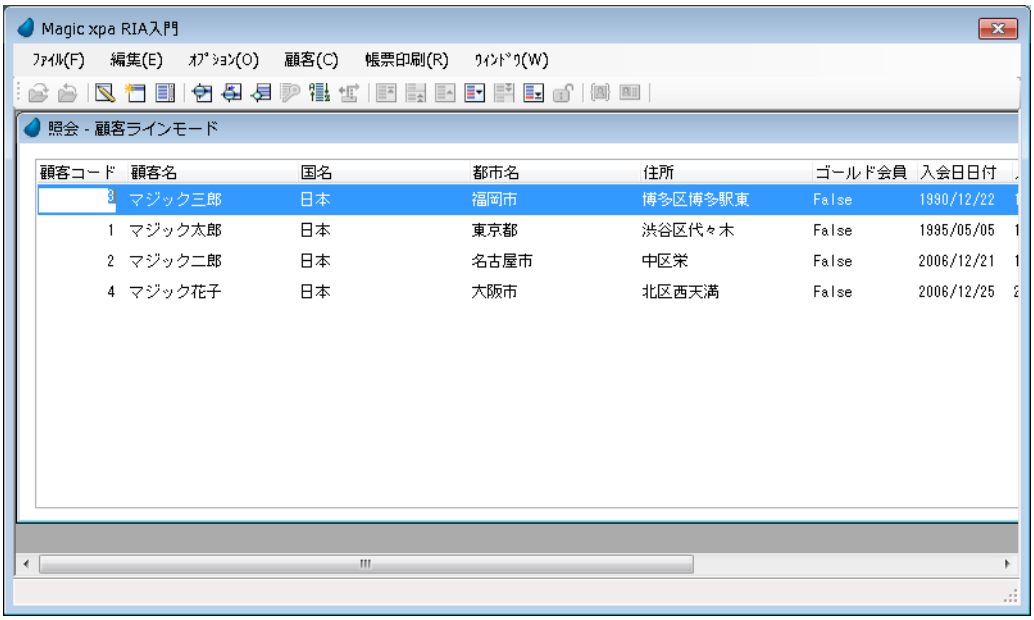
4. ［OK］ボタンをクリックします。



［ソート条件］ダイアログについて

［ソート条件］ダイアログでは、データビューの項目を使用して、一時的なインデックスを定義することができます。現在パークしているカラムがデフォルトの［ソート条件］カラムとなりますが、コンボボックスから変更できます。また、F4 を押下することで条件となるカラムを追加することができます。

テーブルは＜入会日日付＞によってソートし直されました。



5. Esc キーを押下してプログラムの実行を終了します。

９．＜顧客－ラインモード＞プログラムの手作業による作成

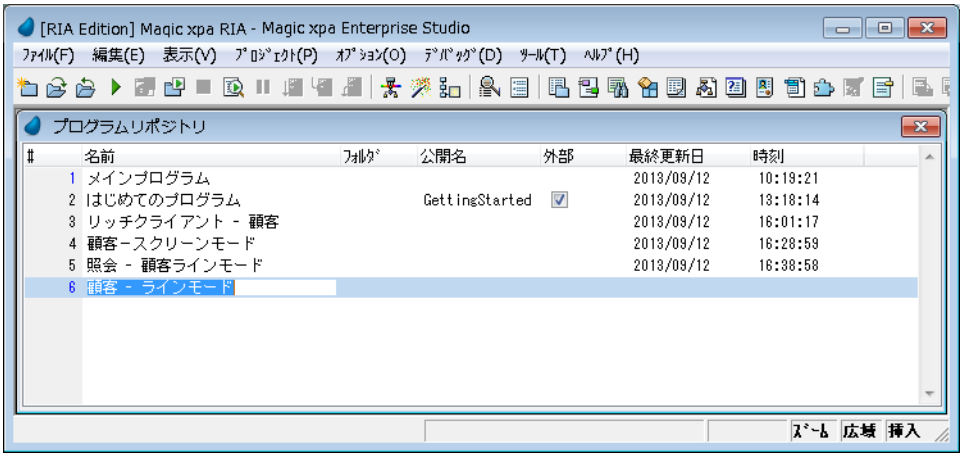
このセクションでは二つのことを学びます。

まず最初に、前のセクションで APG を利用して自動的に作成したプログラムと同等のラインモードプログラムを、手作業により作成します。

次に、ラインモードではカラムの数が多いとき幅の広いフォームが必要になり、エンドユーザが横スクロールしなければすべてのデータを見ることができないため、この問題への対処法を検討します。

1. ［プログラム］リポジトリ（Shift+F3）を開きます。
2. 最後の行にパークします。
3. 一行作成（F4）します。

4. ［名前］カラムで、**＜顧客－ラインモード＞**と入力します。



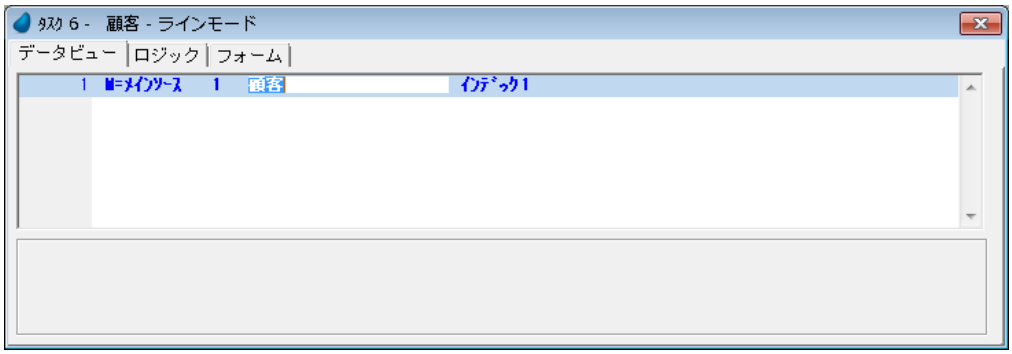
メインソースの定義

1. ＜顧客－ラインモード＞プログラムにズーム（F5）します。
2. ［タスク特性］ダイアログが開きます。［タスクタイプ］特性を「C= リッチクライアント」に設定し、［OK］ボタンをクリックします。［データビュー］エディタがデフォルトで開きます。

3. メインソース定義で次のように設定します。

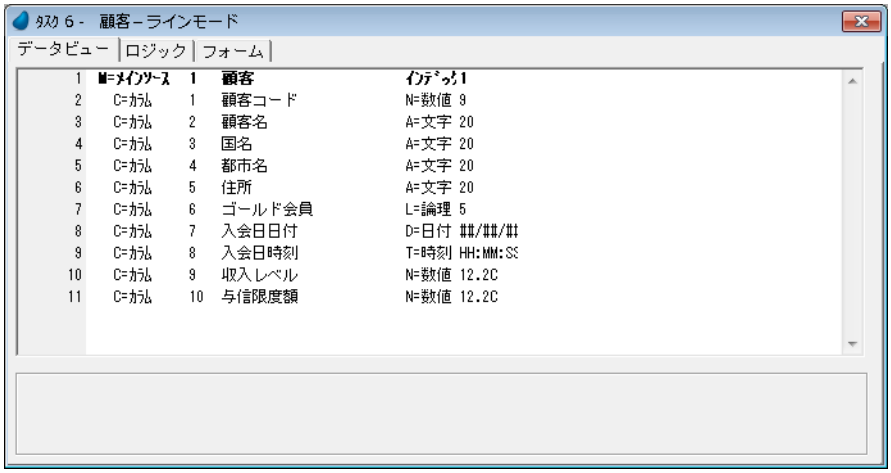
#	データソース番号	データソースの説明	インデックス番号
1　M= メインソース	1　（注１）	顧客	1　（注２）

- 注１：［データソース番号］欄からズームしてデータソース一覧を表示し、最初の＜顧客＞を選択します。
- 注２：［インデックス番号］欄からズームしてインデックス一覧を表示し、最初のインデックスを選択します。



カラムの定義

4. 一行作成（F4）し、カラムを定義します。
5. ［カラム番号］欄でズームし（F5）、［カラム選択］テーブルを開きます。
6. すべてのカラムを選択します。



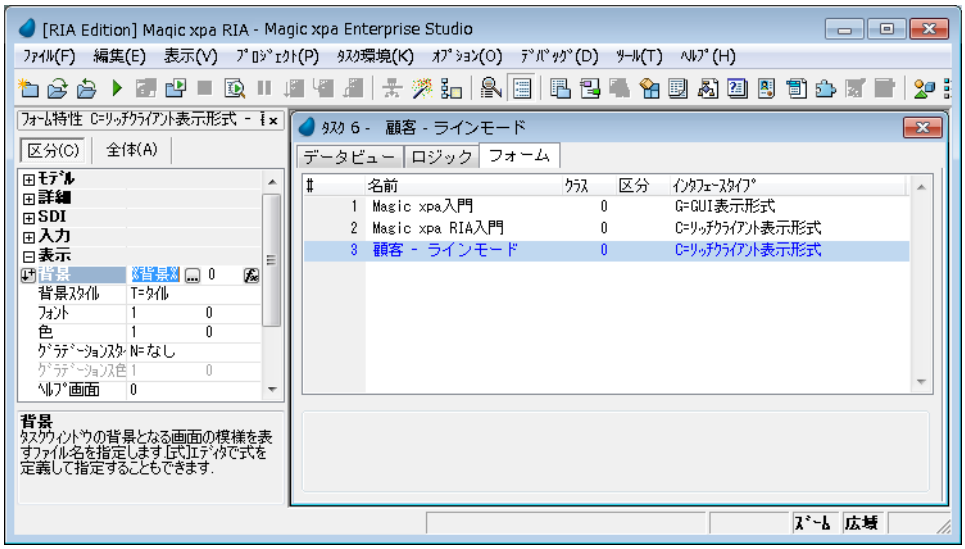
フォームのデザイン

これまでの作業でデータビュー定義は終了したので、次にフォームをデザインしましょう。

最初に、背景を定義します。

1. ［フォーム］タブをクリックして［フォーム］エディタを開きます。
2. ＜顧客－ラインモード＞行にパークします。
3. ［フォーム特性］シート（Alt+Enter）を開きます。

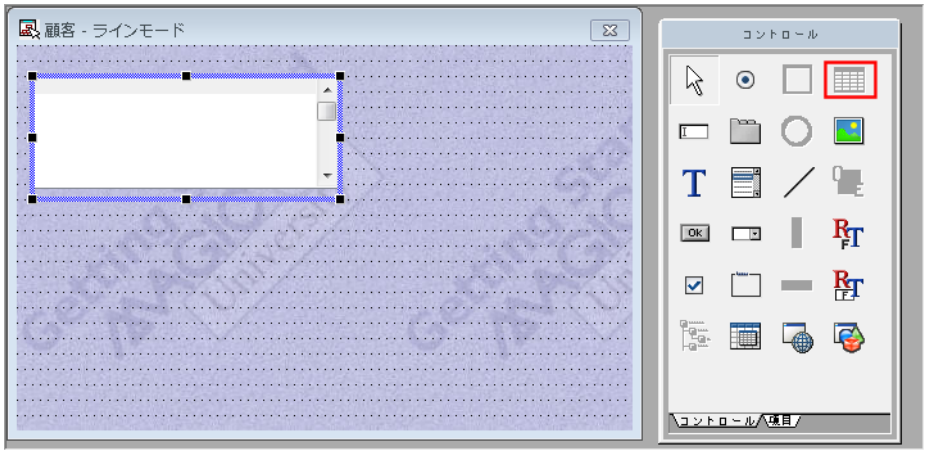
4. ［表示／背景］特性で、＜％背景％＞と入力します。



- **％背景％は論理名であることを示しています。**
- **論理名でアルファベットを使用する場合には、大文字と小文字を区別します。**

［テーブル］コントロールをフォーム上へ配置

5. ＜顧客－ラインモード＞行からズームしてフォームを開きます。
6. ［コントロール］パレットで、［テーブル］コントロールをクリックします。
7. 図に示すような位置に［テーブル］コントロールを配置します。



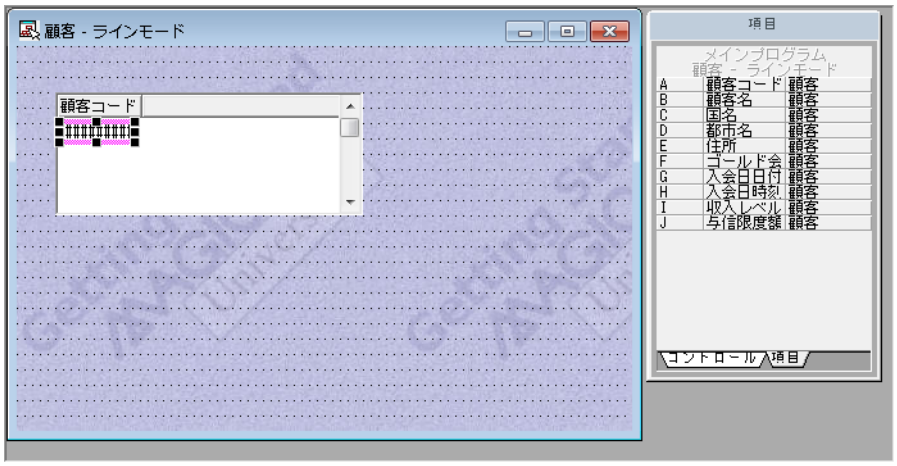
Magic xpa ではフォームにコントロールを配置する場合は、'ドラッグ＆ドロップ' ではなく、'クリック＆プレース' で行います。

フォームに配置できる［テーブル］コントロールは一つだけです。したがって一つを配置した後は、［コントロール］パレットのテーブルアイコンは無効となります。

［テーブル］コントロールへのデータ項目割り当て

8. ［コントロール］パレットで［項目］タブをクリックし、項目パレットを選択します。
9. ＜顧客コード＞をクリックします。

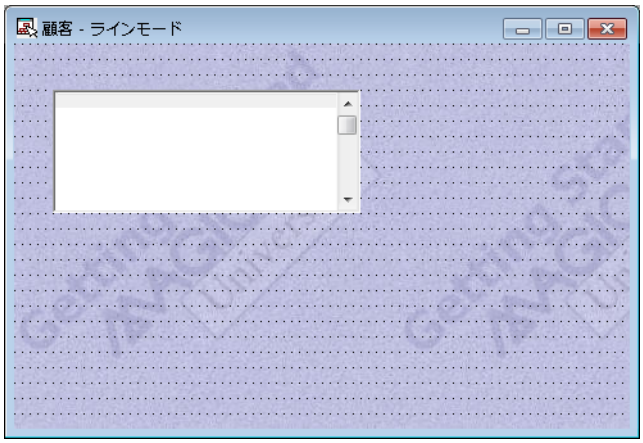
10. ＜顧客コード＞を［テーブル］コントロールに配置します。



［テーブル］コントロールに項目を配置する方法の詳細について

ここでは［テーブル］コントロールに項目を配置する方法を順を追って説明します。

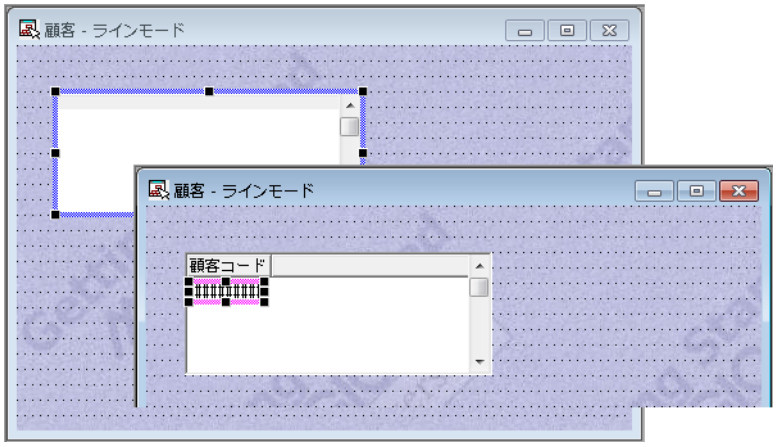
下図はフォームに［テーブル］コントロールが配置されているだけの初期状態を示しています。



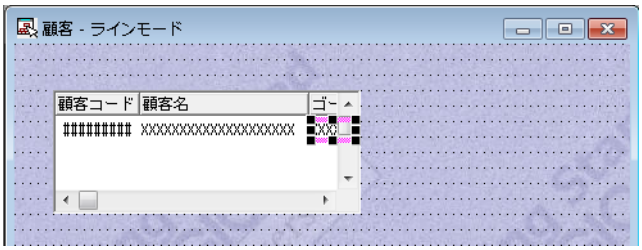
a. 項目を選択するために［コントロール］パレットの［項目］タブをクリックし、項目パレットに切り替えます。

b. 項目パレットの＜顧客コード＞をクリックして選択します。（ドラッグする必要はありません）

c. カーソルを［テーブル］コントロールの白い領域に持って行き、クリックして配置します。
この項目が［テーブル］コントロールの一つのカラムになります。



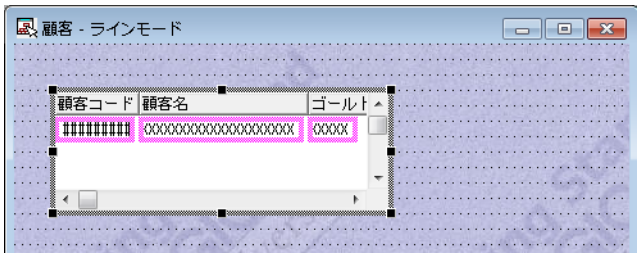
11. 同様にして、次の項目をテーブルに配置します。
- ＜顧客名＞
- ＜ゴールド会員＞



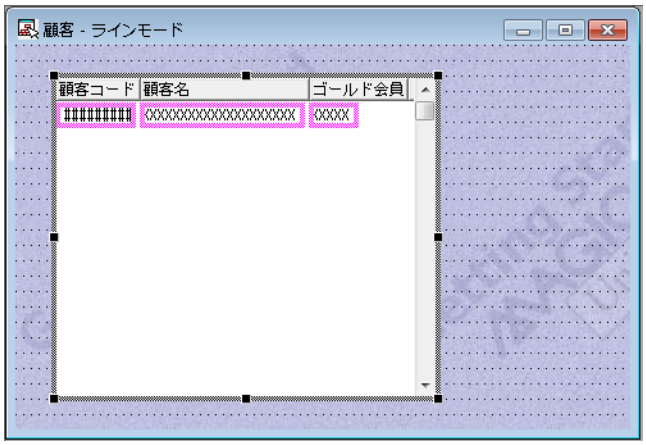
テーブルサイズの調整

コントロールをテーブルに追加するとき、テーブルの幅サイズは自動的には調整されません。
テーブルの高さは表示できるレコード数を制限しますが、これも自動的には変わりません。
テーブルの幅と高さは次のように手動で調整します。

1. テーブルの白い領域をクリックして、テーブル全体を選択します。
2. 右の中央のハンドルをドラッグして、横方向のサイズ調整をします。



3. 下の中央のハンドルをドラッグして、縦方向のサイズ調整をします。



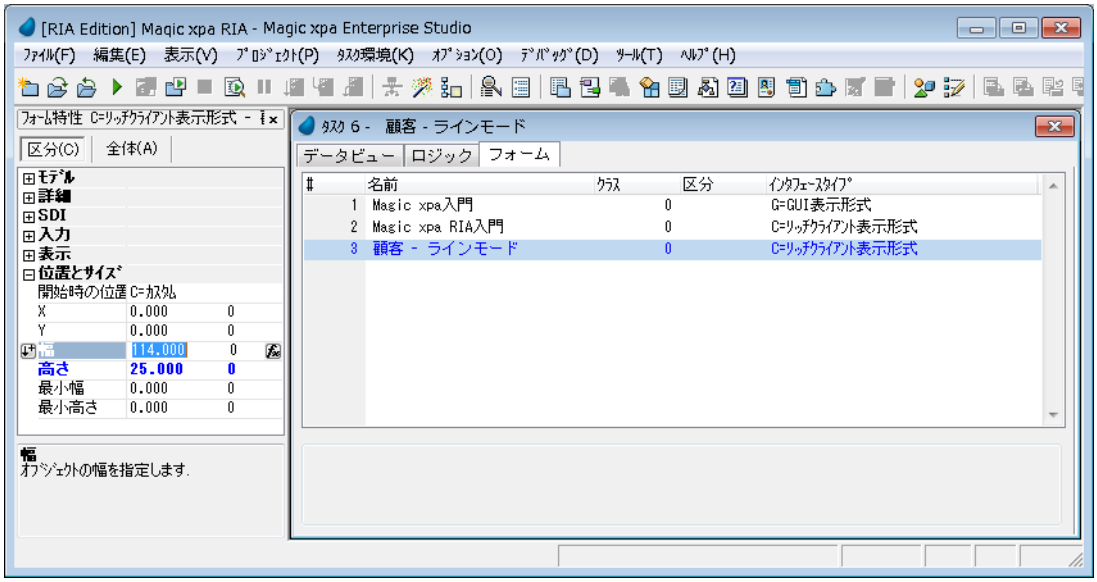
フォームサイズの調整

APG で作成したラインモードプログラムでは、すべての項目がテーブルのカラムとして表示されました。しかし項目の数が非常に多い場合などは、テーブルの幅が広がりすぎて、すべての項目をテーブルに並べるのは現実的ではありません。

ここでは、そのような場合の一つのソリューションとして、一部の項目についてテーブルに登録し、残りの項目についてはテーブルの外に配置する方法を紹介します。

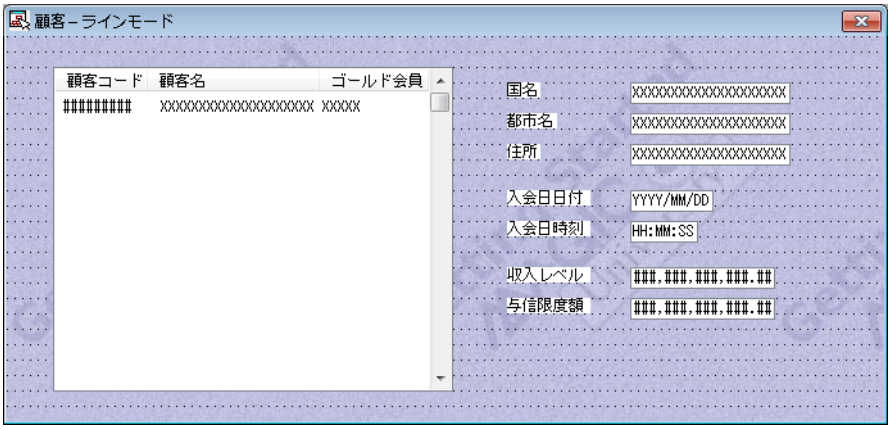
まず最初にフォームサイズを拡げ、テーブルの右に他の項目を配置する領域を確保しましょう。

1. Esc キーを押下します。
2. ［変更確認］ダイアログで［はい］を押下して、変更内容を保存します。
3. ［フォーム］エディタで、＜顧客－ラインモード＞の行にパークしていることを確認し、フォーム特性（Alt+Enter）を開きます。
4. ［位置とサイズ／幅］特性に＜ 114 ＞を設定します。
5. ［フォーム］エディタから＜顧客－ラインモード＞フォームを開き、フォームの幅が拡大されたことを確認してください。

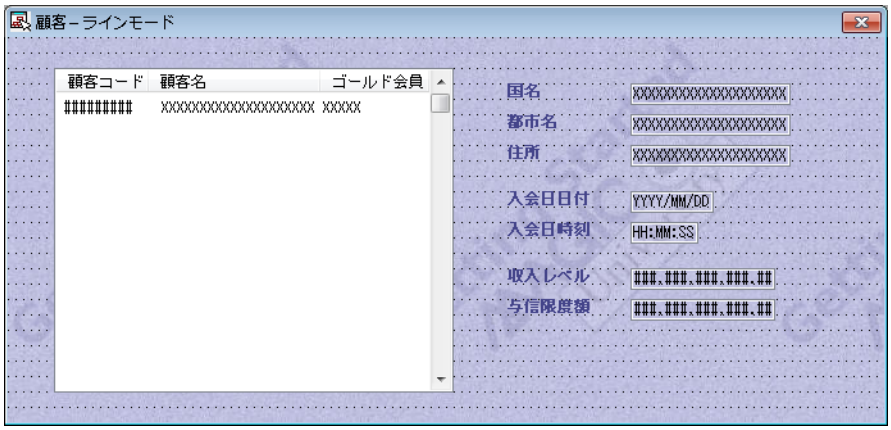


その他のコントロールの配置

1. 残りのコントロールを図に示すように配置してください。
＜国名＞　＜都市名＞　＜住所＞　＜入会日日付＞　＜入会日時刻＞
＜収入レベル＞　＜与信限度額＞



2. ［ラベル］コントロールの色を＜見出し文字＞（#5）にします。（透過色になります）
3. ［ラベル］コントロールのフォントを＜見出し文字＞（#11）にします。（太字になります）
4. ［ラベル］コントロールのサイズ調整をして、すべてが表示されるようにします。
5. ［エディット］コントロールの位置のバランスが良くなるように移動します。
6. ［エディット］コントロールの色を＜文字＞（#6）にします。
7. ［エディット］コントロールのフォントを＜文字＞（#12）にします。
8. プログラムを閉じて変更を保存します。



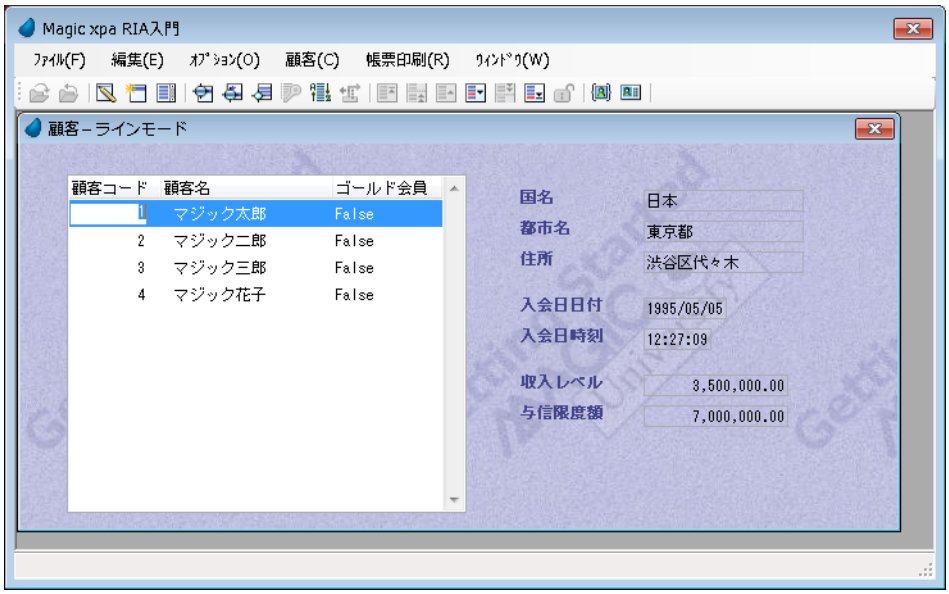
１０.顧客ラインモードプログラムの実行

［プログラム］リポジトリで＜顧客－ラインモード＞プログラムの行（#6）にパークし、F7 キーを押下して実行してみましょう。

カーソルがテーブルのいずれかの項目にパークしているとき、'↑'、'↓'のキーを押下することで顧客レコードを移動することができます。

テーブルの外に配置してある項目データの表示内容は、そのときテーブルでカーソルがパークしている顧客レコードの内容に対応しています。

前のセクションで行ったのと同様の方法で、レコードの追加、内容の編集、削除ができます。



１１.データの検索 (位置付)

データ検索は、データベースを取り扱うシステムにおいては必須と言っていい機能の一つです。

Magic xpa には、この検索機能があらかじめ組み込まれていて、実行モードで利用することができます。

Magic xpa では、［位置付］の機能を用いて検索を行います。

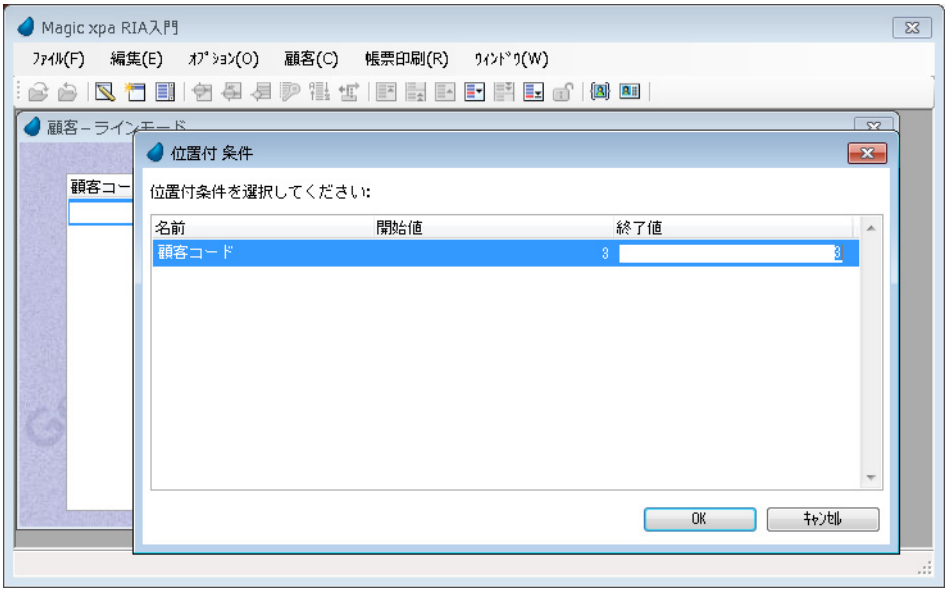
このセクションでは、この［位置付］を用いて、顧客コード３のレコードを検索してみましょう。

実行時の位置付、範囲指定、ソート の各処理は、専用の Magic プログラムを呼び出すことで実現していますが、ここでは、このプログラムの説明は行いません。

＜顧客－ラインモード＞のプログラムが実行しているとき、

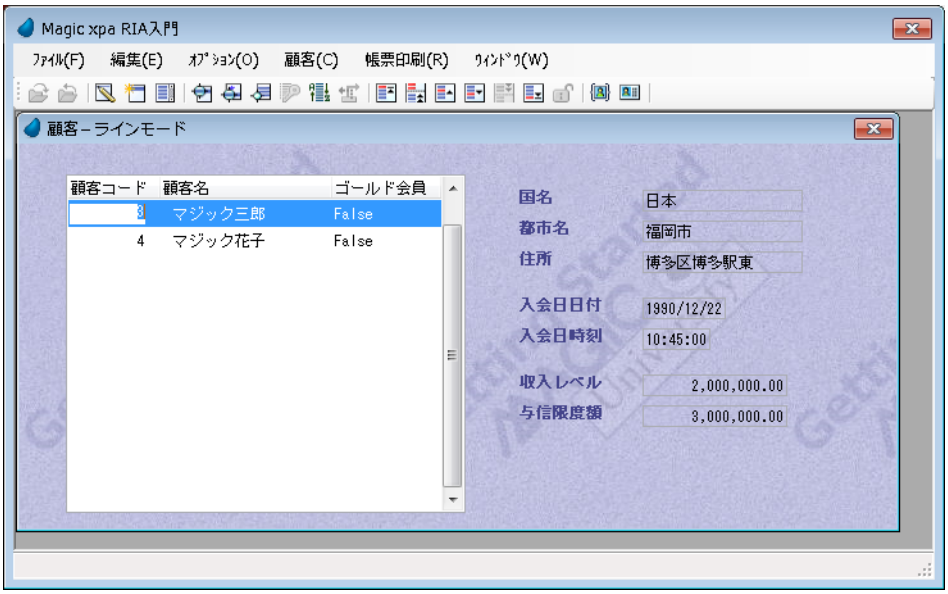
1. Ctrl+L を押下します。［位置付条件］ダイアログが表示され、ここで位置付条件を設定します。位置付条件］ダイアログでは各項目に検索したい値を入力することができます。
2. ＜開始値＞カラムに、＜３＞と入力してみましょう。

3. ［OK］をクリックすると、位置付けた結果が表示されます。



- カラムの型が ' 文字 ' の場合、入力は一文字以上であればよく、ワイルドカード表現が使えます。
- '*' は任意の文字列、'?' は任意の 1 バイト文字コードを意味しますので、たとえば＜ A* ＞と入力すると、先頭文字が 'A' ではじまる文字列を持つデータを検索します。
- 位置付の機能では、英文字は大小を区別します。

カーソルは検索内容の一致した最初のレコードにパークします。この例では＜顧客コード＞が 3 の顧客レコードにパークしています。



検索内容が一致した最初のレコードが最上行に表示されますが、それ以降はインデックス順にしたがう表示となります。このとき、通常のレコード移動のためのキー操作を行うことができます。（矢印キー、Page Up/Down キーなど）

１２ .データのフィルタリング（範囲）

エンドユーザは画面に表示させるデータを絞り込みたいことがあります。

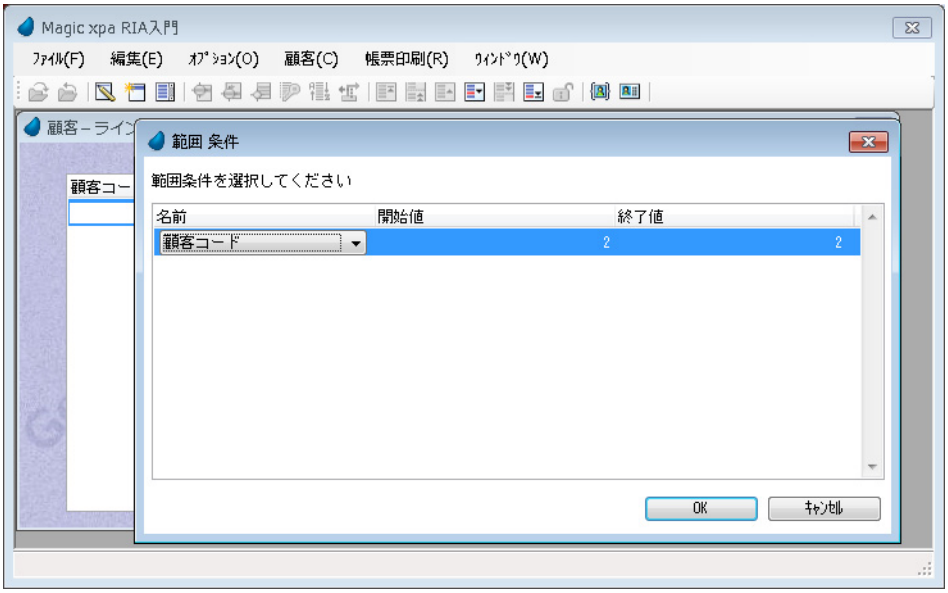
たとえば、次のような場合です。

- 特定の都市の顧客だけを表示させたい。
- ある期間内に登録された顧客だけを表示させたい。
- 収入レベルが一定以上ある顧客だけを表示させたい。

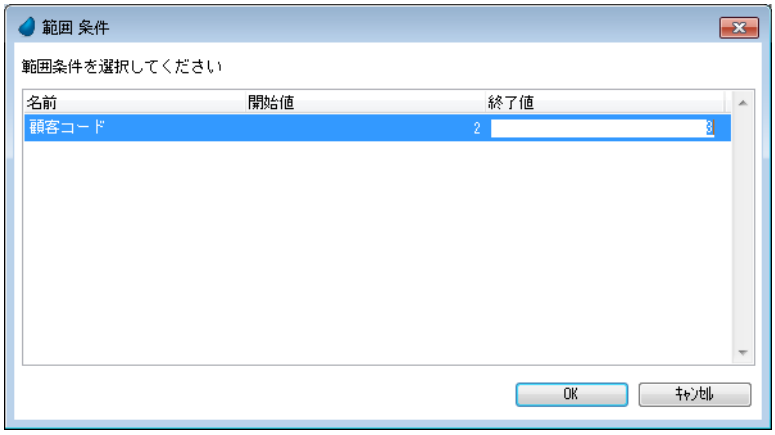
Magic xpa は、この目的のため、実行時に表示データを絞り込むための範囲設定機能を提供します。

ここでは例として、＜顧客コード＞で範囲の設定をしてみましょう。

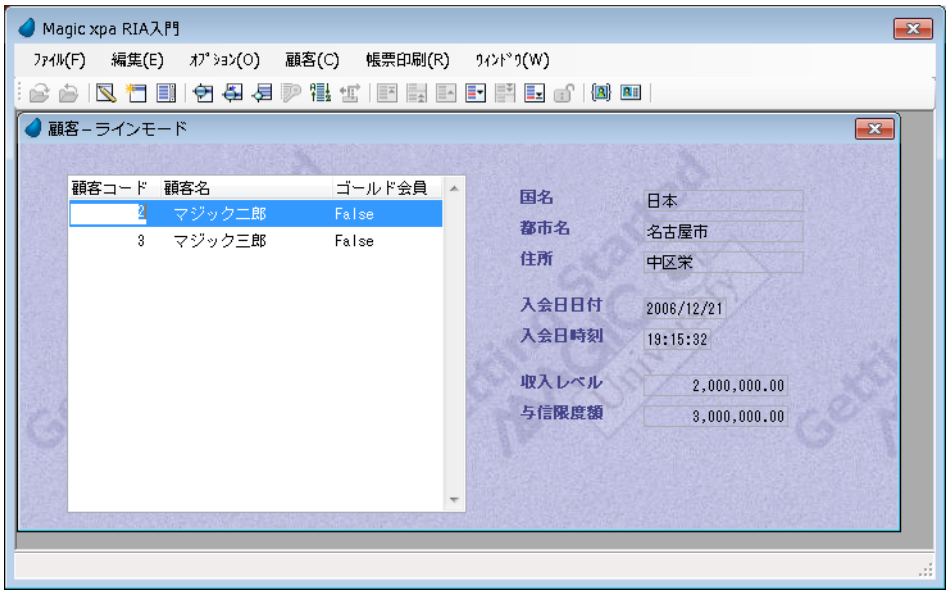
1. Ctrl+R を押下します。［範囲条件］ダイアログが表示され、ここで範囲条件を設定します。
2. ＜開始値＞カラムに＜ 2 ＞を入力します。



3. ＜終了値＞カラム＜ 3 ＞を入力します。



4. Enter キーを押下して結果を表示します。この場合は＜顧客コード＞に設定した範囲（2,3）の 2 レコードのみが表示されているのが分かります。位置付の場合と異なり、範囲指定の場合はその条件に一致するデータレコードのみが表示されます。



範囲設定条件のキャンセル

5. Ctrl+R を押下します。
6. ［範囲条件］ダイアログに表示されている条件項目を［行削除］（F3）で全て削除します。
7. OK をクリックすると、通常の表示画面になります。

範囲設定実習

＜顧客＞データを用いて、いくつかの範囲設定をしてみましょう。

また、範囲指定実行前に必要に応じて現在のデータ項目の内容を修正してください。

1. ゴールド会員だけを表示する。（＜ゴールド会員＞ =True）
2. 10 年以上会員として登録している顧客だけ表示する。（たとえば＜入会日日付＞が 1901/01/01 から 1997/01/01 の範囲）

3. ゴールド会員でかつ与信限度額が 3,500,000 以上で 10,000,000 以下の顧客だけが表示されます。それぞれ実行後に正しくデータが絞り込まれていることを確認しましょう。
4. Esc キーを押下してプログラムを終了します。

１３ .フォーム上のコントロールの整理

フォーム上のコントロールの配置を整理して、もっと見映えよくしたい場合があります。

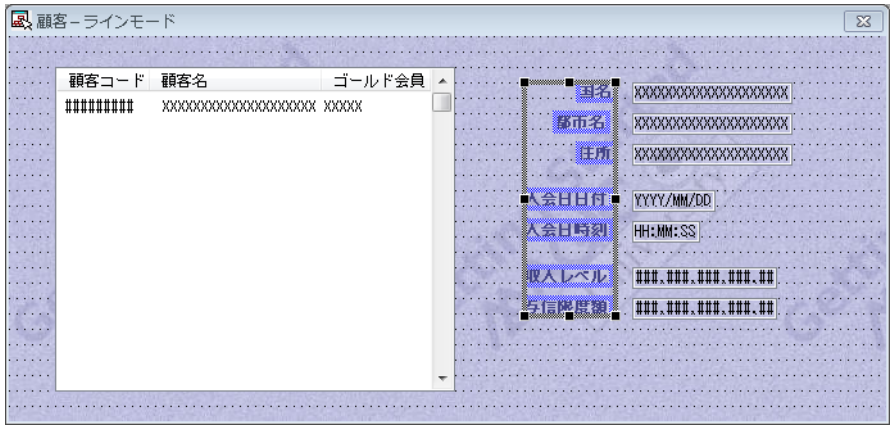
［コマンド］パレットの上二行には、そのようなときのためにコントロールを整列させる機能を持つアイコンがあります。

1. ［プログラム］リポジトリで、＜顧客－ラインモード＞プログラムにズームします。
2. ［フォーム］エディタを選択します。

3. Ctrl キーを押下しながら次の［[ラベル］コントロール］を一つ一つ順にクリックします。
   - ＜国名＞
   - ＜都市名＞
   - ＜住所＞
   - ＜入会日日付＞
   - ＜入会日時刻＞
   - ＜収入レベル＞

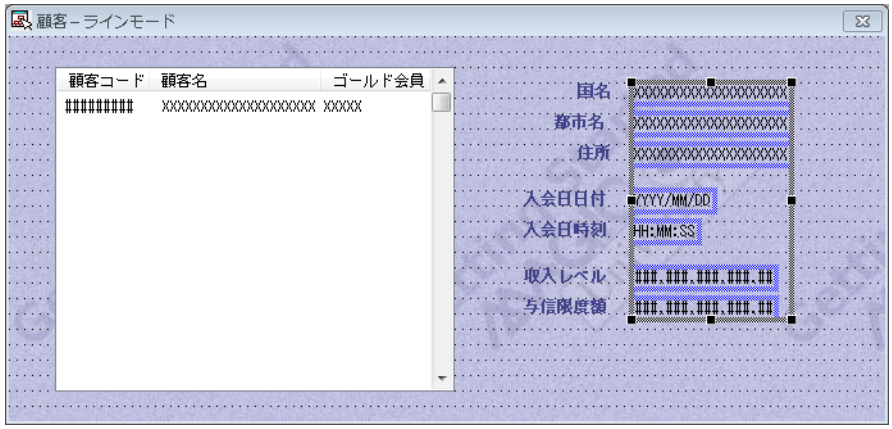
- ＜与信限度額＞

4. コントロールがすべて図のように選択されていることを確認しましょう。
5. ［コマンド］パレットで［選択右寄せ］のアイコンをクリックしてください。この操作で、見出しの［ラベル］コントロールが右寄せに整列されました。



6. Space キーを押下してください。この操作はコントロール選択を解除します。
7. 同様にして、＜国名＞から＜与信限度額＞までの［エディット］コントロールを選択してください。

この状態で、見出し文字が［エディット］コントロールに重ならないよう注意しながら、'→'または'←'キーを押下して選択されたコントロールの位置を調整しましょう。



１４ .Tab 順序

Tab 順序は、エンドユーザが［Tab］キーを押下したときにコントロール間をカーソルが移動する順番を定義するものです。

各コントロールは Tab 順序番号を持っており、それによって Tab キーを押下したときの動作が決まります。

Tab 順序１番のコントロールは、プログラムが起動したときにカーソルが最初にパークするコントロールであることを意味しています。

Tab 順序２番のコントロールは、Tab 順序１番のコントロールにカーソルがパークしているときに Tab キーが押されたときの移動先コントロールであることを意味します。

Magic xpa はフォーム上のすべてのコントロールの Tab 順序を制御します。開発者は自動的に決定される Tab 順序を採用するか、または独自に設定するかを選択することができます。

自動 Tab 順序が有効になっている場合、Magic xpa はフォームへのコントロールの追加や削除に応じて自動的に Tab 順序が振り直され、開発者が変更することはできません。

自動 Tab 順序が無効になっているときは、開発者がコントロールの Tab 順序を変更することができます。

このセクションでは、［Tab 順序］特性について実習を通して学んで行きます。

自動 Tab 順序

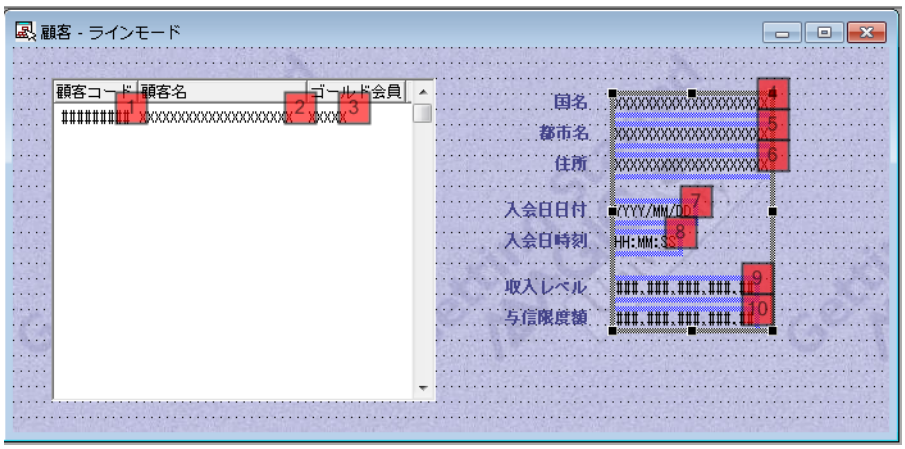
1. ［描画］メニューから［順序／ TAB 順序表示］を選択します。

Tab 順序が赤色で表示されます。自動 Tab 順序モードが有効なとき、Magic xpa はフォーム上のダイナミックコントロール（たとえば［エディット］コントロール）すべてに Tab 順序の識別番号を生成します。

Tab 順序の識別番号は 1 から始まり、最後のコントロールまで連番で付けられます。

自動 Tab 順序は、フォーム上のコントロールの位置が上から下、左から右という順にしたがって番号が割り振られます。

フォームにコントロールが追加されたり、位置が変更になったりした場合、Tab 順序は自動的に再計算されます。



手動 Tab 順序

自動 Tab 順序では望むような順序とならない場合、手動で設定することができます。

手動で Tab 順序を設定するときは次のようにな手順で行います。

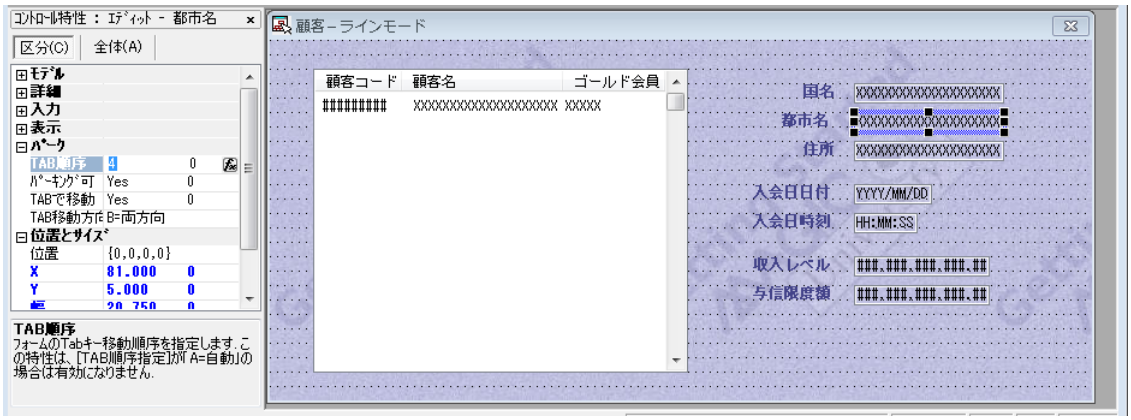
1. 自動 Tab 順序を無効にします。
2. ［Tab 順序］特性を各コントロール毎に設定します。

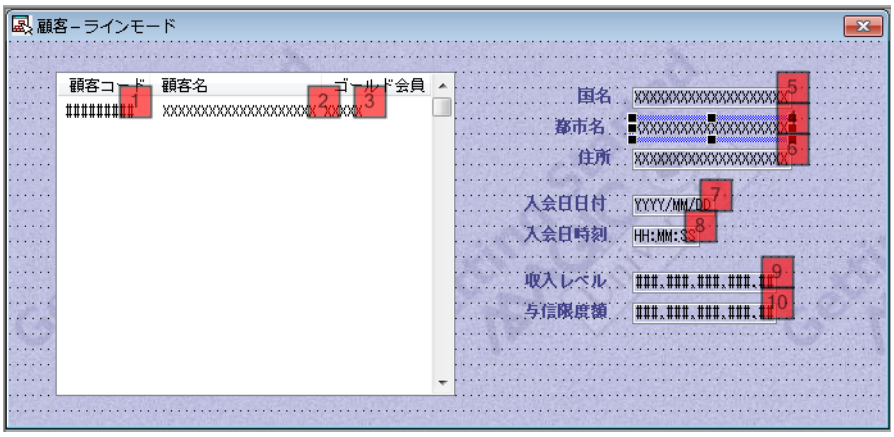
Tab 順序の設定

自動 Tab 順序の例では、「国名」の［エディット］コントロールの Tab 番号は＜ 4 ＞で、「都市名」は＜ 5 ＞でした。ここではそれを変更して、「都市名」を＜ 4 ＞にしてみましょう。

1. ［描画］メニューから［順序／自動タブ順序］を選択して、自動 Tab 順序を無効に切り替えます。無効の場合、［順序／自動タブ順序］のアイコンはフラットな表示となります。
2. 「都市名」の［エディット］コントロールを選択し、コントロール特性（Alt+Enter）を開きます。
3. ［パーク／ TAB 順序］特性に＜ 4 ＞を設定します。

4. プログラムを閉じ、変更を保存します。





手動でコントロールの Tab 順序を変更したとき、他のコントロールの Tab 順序は再計算されます。この例では「都市名」を＜4＞に設定しましたが、再計算の結果「国名」が＜5＞になっています。

プログラムの実行

1. F7 キーを押下して＜顧客－ラインモード＞プログラムを実行します。
2. Tab キーを押下してコントロール間をカーソル移動させてください。

ゴールド会員→都市名→国名の順でカーソルが移動することに留意してください。これは手動で Tab 順序を変更したからに他なりません。

自動 Tab 順序が有効な場合は、ゴールド会員→国名→都市名の順になります。

１５ .練習問題

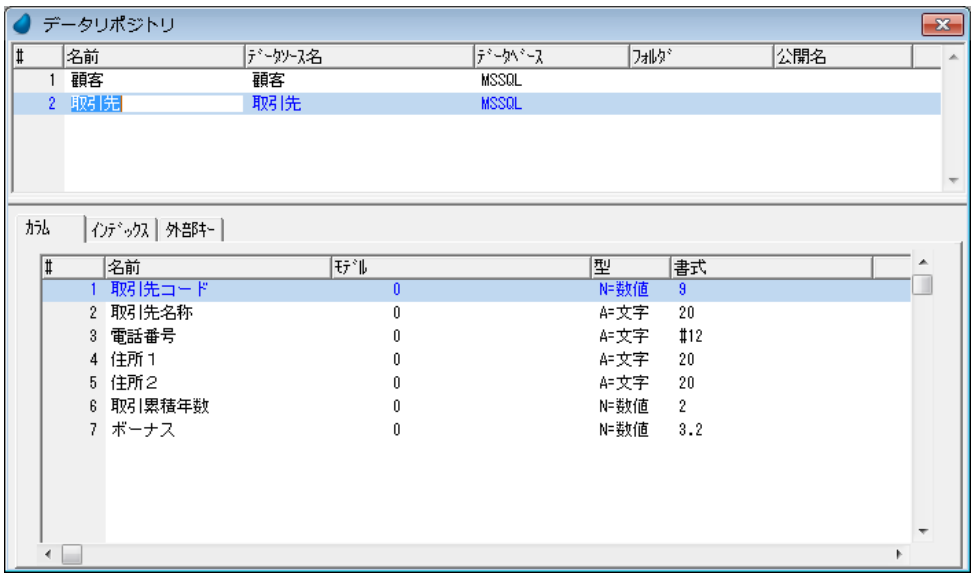
1. 新しいデータソース＜取引先＞を［データ］リポジトリに定義してください。

#	名前	データソース名	データベース	フォルダ	公開名
2	取引先	取引先	MSSQL		

2. カラムを定義してください。

#	名前	モデル	型	書式
1	取引先コード	0	N= 数値	9
2	取引先名称	0	A= 文字	20
3	電話番号	0	A= 文字	#12
4	住所 1	0	A= 文字	20
5	住所 2	0	A= 文字	20

#	名前	モデル	型	書式
6	取引累積年数	0	N= 数値	2
7	ボーナス	0	N= 数値	3.2

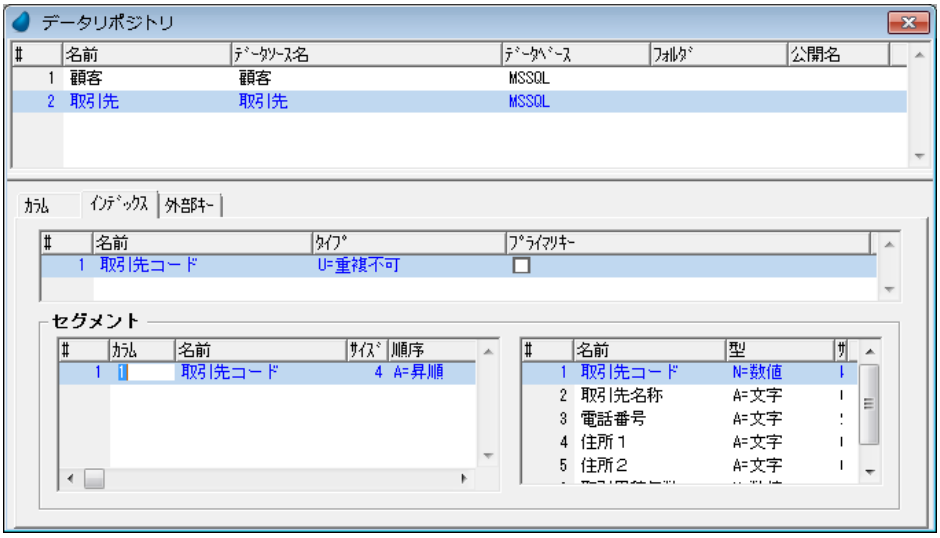


3. インデックスを定義してください。

#	名前	タイプ	プライマリキー
1	取引先コード	U= 重複不可	

4. セグメントを定義してください。

#	カラム	名前	サイズ	順序
1	1	取引先コード	4	A= 昇順



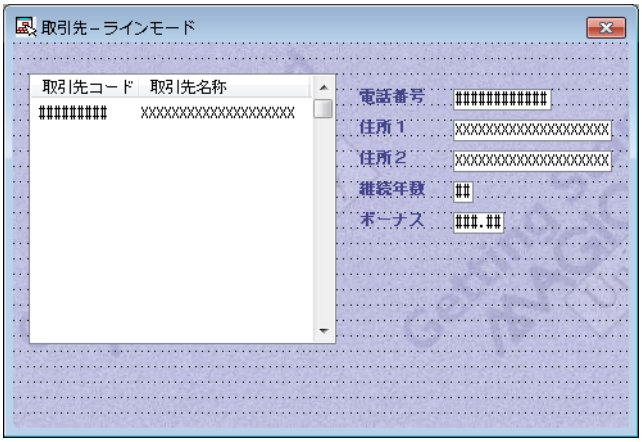
＜取引先－ラインモード＞プログラムのデータビューを以下の手順で作成しましょう。

5. 新しくプログラムを登録し、名前を＜取引先－ラインモード＞としてください。
6. ［データビュー］エディタで、＜取引先＞データソースをプログラムのメインソースとしてください。

7. ＜取引先＞データソースのすべてのカラムを選択します。

＜取引先－ラインモード＞プログラムのフォームを以下の手順で作成しましょう。

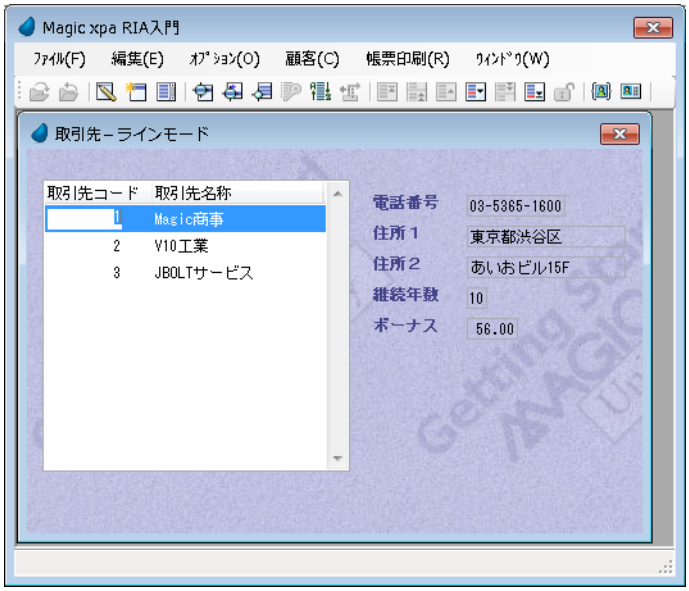
8. フォーム特性（Alt+Enter）を開き、［背景］特性に**＜％背景％＞**を設定してください。
9. ［フォーム］エディタを開きます。
10. ［テーブル］コントロールをフォームに配置してください。
11. ＜取引先コード＞と＜取引先名称＞をテーブルのカラムとして配置してください。
12. ＜電話番号＞，＜住所１＞，＜住所２＞，＜取引累積年数＞，＜ボーナス＞の各項目をテーブルの外に配置してください。
13. ＜電話番号＞，＜住所１＞，＜住所２＞，＜取引累積年数＞，＜ボーナス＞の各［ラベル］コントロールのフォントと色の特性を設定してください。
    - フォント …… ＜見出し文字＞（#11）
    - 色 …… ＜見出し文字＞（#5）
14. ＜取引累積年数＞のテキストを**＜継続年数＞**に変更してください。
15. ＜電話番号＞，＜住所１＞，＜住所２＞，＜継続年数＞，＜ボーナス＞の［ラベル］コントロールのサイズ調整をしましょう。
16. ＜電話番号＞，＜住所１＞，＜住所２＞，＜取引累積年数＞，＜ボーナス＞の各［エディット］コントロールのフォントと色の特性を設定してください。
    - フォント …… ＜文字＞（#12）
    - 色 …… ＜文字＞（#6）
17. ＜電話番号＞，＜住所１＞，＜住所２＞，＜取引累積年数＞，＜ボーナス＞の各［エディット］コントロールのサイズ調整、位置調整を行ってください。
18. ＜電話番号＞，＜住所１＞，＜住所２＞，＜継続年数＞，＜ボーナス＞の各［ラベル］コントロールを［選択左寄せ］で位置調整してください。



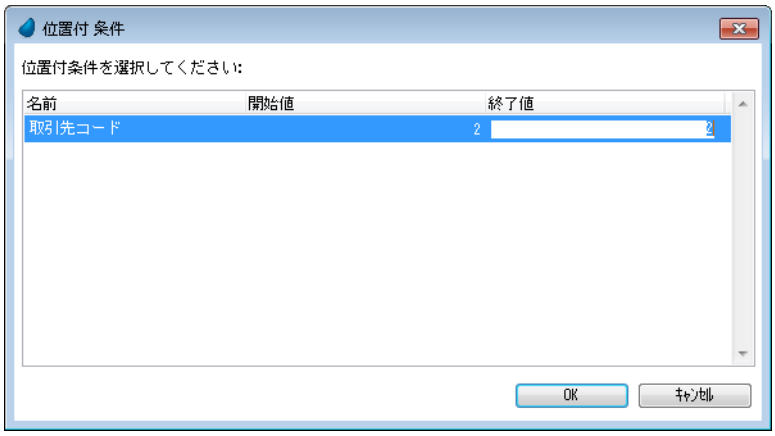
19. ＜取引先－ラインモード＞プログラムを実行してみましょう。
20. 次のような取引先データを登録しましょう。

取引先コード	取引先名称	電話番号	住所１	住所２	継続年数	ボーナス
1	Magic 商事	03-5365-1600	東京都渋谷区	あいおビル１５Ｆ	10	56.00

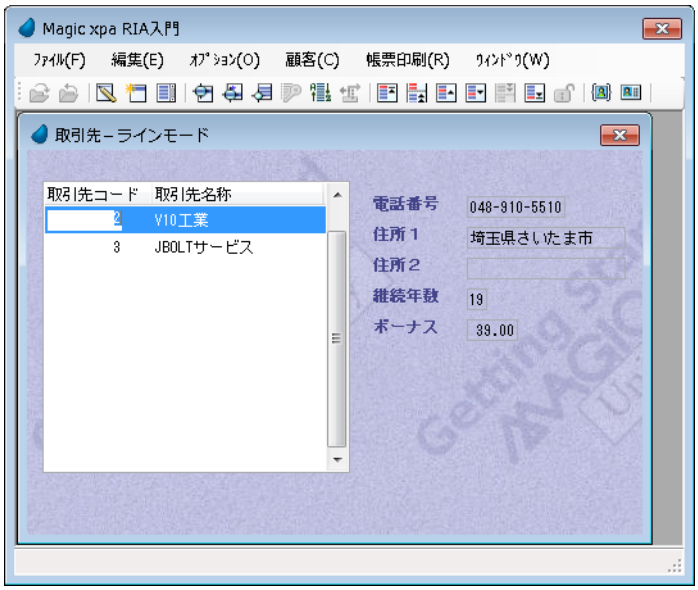
取引先コード	取引先名称	電話番号	住所 1	住所 2	継続年数	ボーナス
2	V10 工業	048-910-5510	埼玉県さいたま市		19	39.00
3	JBOLT サービス	06-8139-3939	大阪府大阪市	JBOLT ビル 12F	16	78.00



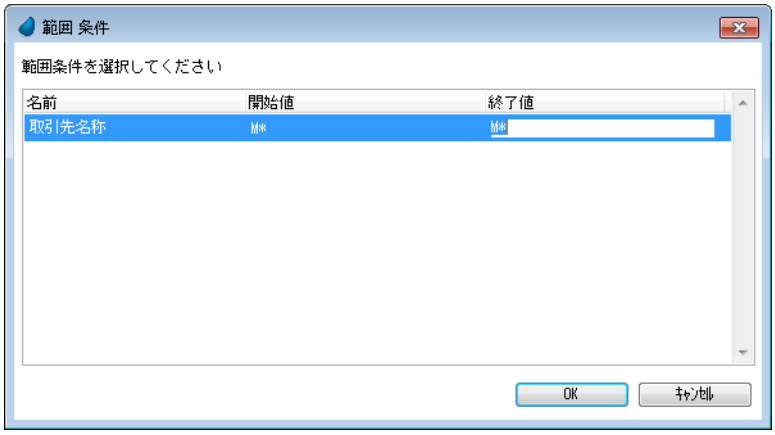
21. 取引先コード #2 に位置付けしてみましょう。



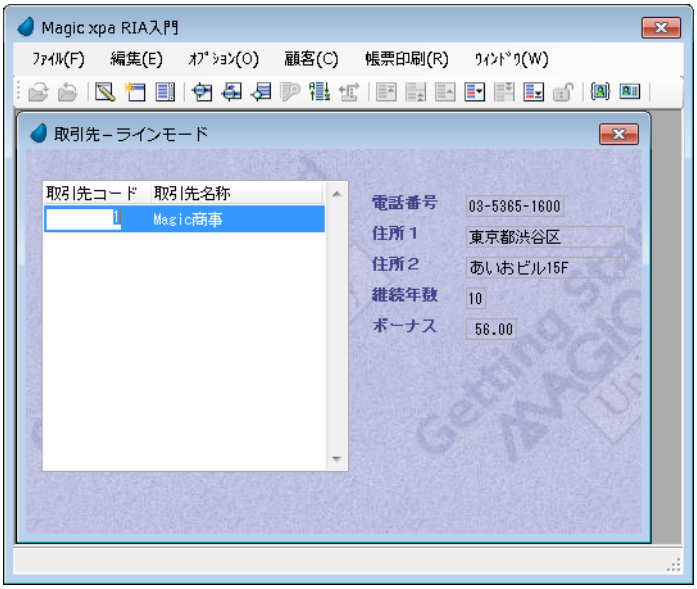
結果として、#2 の< V10 工業>の行にパークし、そのレコードが最上行に表示されます。



22. 取引先名称が 'M’ の文字で始まる会社だけを表示するようにしてください。ワイルドカード表現 '*' を使用することに留意してください。



結果は次のようになります。



１６ .要約

本章では、次のことを学びました。

- データソースを定義し、データソースの内容を表示し、操作しました。
- ［データベース］テーブルに MSSQL を使用するエントリを作成しました。
- ［データベース］テーブルに＜顧客＞データソースを定義しました。
- ［APG］機能を使い、スクリーンモードとラインモードのプログラムを自動作成しました。
- ［APG］で作成したプログラムと同等のものを手作業により作成しました。
- プログラムで［テーブル］コントロールの扱い方について実習しました。
- データの操作を実習しました。
  - レコードの登録
  - レコードの修正
  - レコードの削除
- Tab 順序について実習しました。
  - 自動 Tab 順序
  - 手動設定
- ［コマンド］パレットの整列アイコンを利用してコントロール位置を調整したり、移動、サイズ調整の実習を行いました。

次章では、アプリケーションエンジンの仕組みや考え方について学習します。

第 9 章 アプリケーションエンジンのコンセプト

本章では、Magic xpa の実行エンジン（Magic エンジン）の動作する仕組み、すなわちアプリケーションのデータ処理とプログラムの実行を管理する仕組みについて学習します。

キーワード

- イベントドリブン
- イベント
- トリガー
- ハンドラ
- ロジックユニット
- 実行ルール

学習目標

本章を学ぶことで、次の内容に対する理解を深めましょう。

- Magic エンジンの仕組み
- イベントドリブンの方法
- タスク実行のロジックユニット
- タスク実行のルール

参照

Magic ヘルプの［Magic xpa リファレンス／ Magic エンジン］

１．はじめに

Magic エンジンは、開発者とエンドユーザの双方に使いやすいデータ操作環境を実現する Magic xpa の中心的な処理実行モジュールです。

本章では、この Magic エンジンがどのような手順で処理を実行するのかについて学ぶことで、より良いプロジェクトプログラムを開発できるようになることを目標としています。

Magic エンジンでは、他のアプリケーション開発ツールと異なり、ファイルをオープンし、データソースからレコードを読み込み、ソートして、さらに必要なアクションを実行するなど、プログラム実行に必要な多くの処理が自動的に実行されます。このようなプログラム実行に必要な機能が Magic エンジン内にあらかじめ豊富に組込まれているために、アプリケーション開発者は開発時間を大きく削減することができます。

Magic xpa では上記のような機能をロジックユニットと呼ばれる実行単位として定義し、その中でコマンドを複数組合わせることで処理を実現します。開発者は実行するコマンドの種類と実行タイミング等を指定できますが、実行に際しては Magic エンジンがどのロジックユニットが実行されるべきかを判断し、コマンドの実行を制御します。Magic エンジンによって実行が管理される Magic xpa の組込みロジックユニットの多くは、開発者やエンドユーザがその実行を制御できないため、これらの組込みロジックユニットの挙動を理解することは大変重要で、その動作の仕組みを踏まえてプログラム開発を行うことにより、さらに効率のよい開発作業を実現できるようになります。

ここでは、Magic エンジンのコンセプト全般について説明しています。

２．イベントドリブンの考え方

Magic エンジンの本質はイベントドリブン制御です。つまり、Magic エンジンはタスク実行中に発動されるさまざまなイベントに呼応して、対応する処理を行います。イベントドリブン型のプログラミングでは、ある事象を想定し、その発生によりオンデマンドで実行される処理コードを、事象ごとに記述して行きます。プログラム開発者はある処理が実行された後、次にどの処理が実行されるかについては知ることがありません。

イベントドリブンの考え方には三つの要素があります。すなわち、イベント, イベントトリガ, そしてイベントハンドラです。

イベント

イベントはある出来事の論理的定義です。別の言い方をすれば、イベントはフラグであり、Magic xpa に対して何か処理をしなければならない状態にあることを告げるものです。イベントは Magic xpa に組込みのものであったり、開発者によって定義されるユーザイベントであったりします。

イベントトリガ

イベントの発動に関する手続き的な処理というものはないため、イベントが発動されたことを判断するための条件が必要になり、それをトリガと呼びます。トリガには、たとえば Magic xpa に内部定義されているものや、開発者が定義するもの、エンドユーザアクションによるものなどがあります。

イベントハンドラ

イベントに対応する実際の処理は、Magic xpa ではハンドラとして定義されます。イベントのトリガが成立したときに処理が実行されるためには、イベントに対応するハンドラが定義されていなければなりません。ハンドラは通常複数のコマンドの組合わせで定義されます。ハンドラには、Magic xpa 組込みのものと開発者によって定義されるものとがあります。

３．タスク

Magic xpa の基本的オブジェクトは［タスク］です。一つ以上のタスクの集まりがプログラムを構成します。タスクは必ず明確な始まりと終わりがあるデータ処理の手続きであり、エンドユーザとの対話の有無を含め、一連のコマンドの処理によって機能を実現しています。このタスクの実行プロセスはタスクのメインソースとタスクタイプによって基本的な定義が行われます。

したがってプログラム作成においては、まずはじめにタスクのメインソースとタスクタイプの定義を行います。

メインソース

多くの場合、Magic エンジンはタスクメインソースのレコードをループして処理を行います。

タスクタイプ

タスクには次の三つのタイプがあります：［O= オンライン］、［B= バッチ］、［C= リッチクライアント］

「オンライン」と「リッチクライアント」のタスクは、タスクの実行中にエンドユーザとの対話処理を行います。「オンライン」と「リッチクライアント」のタスクにおいては、メインソースが定義されている場合、エンドユーザの操作によってカーソルがレコード間を移動することによりメインソース内を参照してゆきます。そしてエンドユーザによって参照されているレコードだけが Magic エンジンの処理対象となります。

本書では、オンラインタスクについて扱いません。

バッチタスクでは、データの集計計算やレポート作成などのためにデータソース全体をスキャンする場合など、エンドユーザとの対話はなく、自動的に処理を実行します。バッチタスクでメインソースが定義されている場合は、与えられた範囲の最初のレコードから最後のレコードまで、Magic エンジンはメインソース内を自動的にループします。

- **全てのタスクタイプにおいて範囲を設定することができます。**
- **Magic エンジンはその設定されたレコード範囲内でループします。**

タスクの実行ステージ

メインソースとタスクタイプを定義したら、次にそのタスクの目的を実現するためのカスタマイズを行います。具体的には実行するコマンドの選択と、そのコマンドの実行条件を設定すること、およびタスクのどの実行ステージにおいてそれらのコマンドを実行するかを定義することです。

標準的なタスクはいくつかのステージで実行されます。これらのステージはタスクタイプや設定されるロジックユニットにより構成が変わります。

たとえば単純なオンラインタスクでは次のような実行ステージがあります。

1. タスクデータビューで定義されているデータベースのデータソースがオープンされます。（メインソースおよびリンクされたデータソースを含みます。）
2. Magic エンジンはタスクの実行を開始し、変数項目の初期値設定など必要な初期化を実行します。
3. メインソースの最初のレコードが読み込まれ、レコードに対する処理が開始されます。
4. 設定されているコマンドにしたがって、レコードに対する処理が行われます。たとえば画面へのデータ表示やエンドユーザによるデータ選択、データ更新やデータベースへのレコード内容の保存などが実行されます。
5. Magic エンジンは他のレコードを読み込んで処理を続行するか、タスクを終了するかの判定を行います。タスクの終了はエンドユーザの終了リクエストによるか、あるいはタスク終了条件が［True］かどうかで判断されます。処理を続行する場合は、次のレコードが読み込まれ、ステージ 4 に戻ります。
6. Magic エンジンはデータベースのデータソースをクローズするなどの処理を実行し、タスクを終了します。

タスク実行ロジックユニット

Magic xpa のタスク実行中に発動される特定のイベントに対して、多くのロジックユニットがあらかじめ組込まれています。

ロジックユニット内部では、開発者は実行する一連のコマンドを定義し、イベントのトリガ条件が成立したとき、これらのコマンドが実行されることになります。

Magic xpa のロジックユニットは、以下に示すような異なるレベルを取り扱えるように設計されています。

- タスクレベル
  タスクの開始時と終了時に実行させる処理コマンドは、タスクレベルのロジックユニットに定義します。
- レコードレベル
  メインソースのレコード処理を行うための処理コマンドは、レコードレベルのロジックユニットに定義します。
- コントロールレベル
  エンドユーザが特定のコントロールにアクセスしたときに実行する処理コマンドは、コントロールレベルのロジックユニットに定義します。

- 項目レベル
  項目の値が変更されたときに実行する処理コマンドは、項目レベルのロジックユニットに定義します。
- イベントレベル
  イベントのトリガが成立したときに実行する処理コマンドは、イベントレベルのロジックユニットに定義します。

これらのロジックユニットは開発者によって定義することができます。Magic エンジン内部には、これらとは別に内部のルールによって定められたロジックユニットが定義されており、それについては開発者が変更することはできません。

［タスク前］と［タスク後］

タスクレベルのロジックユニットは、Magic xpa の基本的なロジックユニットであり、タスクの開始時と終了時に実行する処理コマンドを定義します。このレベルは二つのロジックユニットがあります。

［タスク前］

［タスク前］のロジックユニットはタスク開始時に一度だけ実行されます。

タスクの開始がトリガとなります。

このロジックユニットには、初期化ステージに一度だけ実行される処理コマンドを定義します。（このステージではデータソース情報はまだ利用できません。）

［タスク後］

［タスク後］のロジックユニットはタスク終了時に一度だけ実行されます。

タスクの終了がトリガとなります。

このロジックユニットには、終了ステージに一度だけ実行される処理コマンドを定義します。（データソース情報はこのステージでは操作すべきではありません。）

［レコード前］と［レコード後］

レコードレベルのロジックユニットは、Magic xpa の基本的なロジックユニットであり、レコードが読込まれるときやレコードのデータ操作が終了するときに、レコードごとに実行される処理コマンドを定義します。このレベルには二つのロジックユニットがあります。

［レコード前］

［レコード前］ロジックユニットは、レコードがデータソースから読込まれ、その内容が表示される前のタイミングで実行されます。レコードの読み込みがトリガとなります。

［レコード前］ロジックユニットでは、レコードに対する初期化ステージとして、新たに読込んだレコードに対して一度だけ実行される処理コマンドを定義します。

［レコード後］

［レコード後］ロジックユニットはバッチタスクの場合はすべてのレコードに対し実行され、オンラインタスクとリッチクライアントタスクの場合には、参照され内容が変更されたレコードに対してのみ実行されます。レコードに対する処理の終了がトリガとなります。

［レコード後］ロジックユニットでは、レコードの内容がデータベースに保存される前に、レコードに対する終了ステージの処理として、一度だけ実行される処理コマンドを定義します。

［レコード後］ロジックユニットは、エンドユーザがそのレコードに対してどのように操作を加えるかにより、一度とは限らず、何度も処理が起動されることがあります。

レコードレベルのロジックユニットは周期的であり、タスク実行中に、通常は何度も実行されます。

［コントロール前］、［コントロール後］、［コントロール検証］

コントロールレベルのロジックユニットは、Magic xpa の基本的なロジックユニットであり、オンラインタスクとリッチクライアントタスクにおいて利用します。

コントロールレベルのロジックユニットには、フォーム上の特定のコントロールに対するアクセスがあったときに実行される処理コマンドを定義します。

［コントロール前］

［コントロール前］ロジックユニットは、該当コントロールにカーソルが移ってきたときに実行されます。そのコントロールにカーソルをパークさせるアクションがトリガとなります。

［コントロール前］のロジックユニットには、エンドユーザがコントロールの内容に対して変更を加える前に実行される処理コマンドを定義します。

［コントロール検証］

［コントロール検証］ロジックユニットは、該当コントロールからカーソルが他に移るときに、［コントロール後］ロジックユニットの実行に先立って必ず実行されます。

このロジックユニットは、該当コントロールにカーソルがパークすることなく通過するときにも実行されます。（これは特別な［高速］モードの時に実行されます。）

［コントロール検証］ロジックユニットには、エンドユーザがその内容を編集したか否かにかかわらず、その内容を検証するための処理コマンドを定義します。

［コントロール後］

［コントロール後］ロジックユニットは、該当コントロールからカーソルが他に移るときに実行されます。コントロールからのカーソル移動がトリガとなります。

［コントロール後］ロジックユニットには、コントロールの内容を検証したり、内容が変更された場合の対応アクションを起こすための処理コマンドを定義します。

コントロールレベルのロジックユニットは、エンドユーザの操作によりコントロールにカーソルが移る都度、何度も実行されます。

［項目変更］

項目レベルのロジックユニットは、Magic xpa の基本的なロジックユニットであり、項目値の変化を検出して実行される処理コマンドを定義します。

［項目変更］ロジックユニットはエンドユーザが項目の値を変更する（［エディット］コントロールの編集）か、内部処理（［項目更新］処理コマンドの実行などによる）の結果として変更された場合に実行されます。つまり項目値の変更がトリガになります。

［項目変更］ロジックユニットは、項目値の変更により実行されるため、変更の都度、何度も実行されます。

［イベント］

これまで説明してきたロジックユニットは、タスクの実行開始、レコードの読み込み、項目の変更検出などの Magic xpa の内部トリガによって起動されるものでした。

［イベント］ロジックユニットは、これに対して開発者が定義し、顧客の新規登録、帳票の印刷、集計計算などのアプリケーションロジックを構成することができます。

イベントについては、第 10 章で詳細に説明します。

これまでの要約

次の表は、Magic xpa の基本的ロジックユニットとそのトリガをまとめたものです。

ロジックユニット名	ロジックユニットトリガ	実行の回数
タスク前	タスク処理の開始	1 回
タスク後	タスク処理の終了	1 回
レコード前	レコードの読み込み	0 回以上
レコード後	レコード操作の終了	0 回以上
コントロール前	コントロールへのパーク	0 回以上／レコード毎
コントロール検証	コントロール移動前	0 回以上／レコード毎
コントロール後	コントロール移動時	0 回以上／レコード毎
項目変更	項目値の変更	0 回以上／レコード毎

４．実行ルール

このセクションではタスクの開始から終了までの一連の処理プロセスについて、より詳細に解説します。

タスクの初期化

データビューの準備

- ［データビュー］エディタで定義されたすべてのデータソースをオープンします。
- Magic xpa はタスクデータビューのコマンドを解析し、次の情報に基づいてタスクの論理レコードを作成します。
  - メインソースおよびリンクされたデータソースのカラム定義
  - 変数およびパラメータ
- 他のタスクから呼び出されている場合は、パラメータの値を受け取ります。
- 変数項目については、その型のデフォルト値が設定されます。（数値型は 0、文字型はブランクなど）
- タスクで扱うデータ範囲が［範囲］ウィンドウに定義された範囲基準にしたがって設定されます。
- ［ソート］テーブルに指定があるとき、それに従ってデータビューをソートします。

それ以外の準備事項

次に実行されるのはタスクを初期化する手続きです。

- ［入出力ファイル］テーブルに定義があるときは、その I/O デバイスをオープンします。
- ［タスク前］ロジックユニットの処理コマンドが実行されます。

ここまでのステージでは、メインソースおよびリンクデータソースからのデータはまだ利用できません。

レコードの処理

［タスク前］が終了した後、以下の手順でデータビューのレコードが処理されます。

［レコード前］の処理

- データビューの最初のレコードが位置付けされ、読込まれてきます。

オンラインタスクやリッチクライアントタスクでは、データソースにデータが存在しないとき、タスクは［登録］モードで実行が開始されます。ただし、これはタスク特性で登録モードを許可する設定になっている場合であり、許可しない設定の場合、タスクは何もせずに終了します。また、［空のデータビュー許可］特性が＜ Yes ＞の場合は、データが存在しなくてもタスクを起動させることができます。

- すべての変数の［代入］式が評価され値が設定されます。実項目に定義された［代入］式はタスクが［登録］モードのときのみ有効です。
- 最初に読込まれたレコードから他のデータソースへのすべてのリンクが実行されます。
- タスク特性の［チェック時期］が［B= 前置］のとき、Magic エンジンは［タスク終了条件］を評価します。
- レコード処理のループが開始されます。したがって、この処理からデータソースのカラムに操作を行うことができます。
- ［レコード前］ロジックユニットの処理コマンドが実行されます。

コントロール処理

各レコードでは、オンラインタスクやリッチクライアントタスクの場合、アクセスされたコントロールに定義されたロジックユニットが実行されます。エンドユーザはキーボード（Tab または Shift+Tab）またはマウスのクリックによりコントロールにアクセスします。

- コントロール前 …… カーソルがそのコントロールに移動したとき、該当する［コントロール前］ロジックユニットの処理コマンドが実行されます。
- コントロール検証 …… カーソルがそのコントロールから他に移動しようとするとき、［コントロール検証］ロジックユニットが定義されていればそのコマンドが実行されます。また、（高速モードで）カーソルがそのコントロールを通過した場合も同様です。

- コントロール後 …… カーソルがそのコントロールから離れるとき、［コントロール後］ロジックユニットの処理コマンドが実行されます。これは他のコントロールにカーソルが移動したり、他のレコードに移動したり、タスクを終了する場合などがあります。

［項目変更］の処理

オンラインタスクやリッチクライアントタスクでは、項目の値が変更される都度、［項目変更］ロジックユニットが実行されます。

［レコード後］の処理

オンラインタスクやリッチクライアントタスクでは、［レコード後］ロジックユニットは変更されたレコードに対してのみ実行されます。

一方、バッチタスクでは［レコード後］ロジックユニットはすべてのレコードに対して実行されます。

オンラインタスクやリッチクライアントタスクでは、［レコード後］ロジックユニットはエンドユーザがレコードに対する操作を終えたときに起動されます。具体的には、次のような場合になります。

- キーボードまたはマウスの操作により、他のレコードに移動したとき
- タスクモード（修正、登録、照会）が変更されたとき
- レコードの削除が行われたとき
- タスクの実行が終了するとき、すなわちエンドユーザの操作により終了するか、［タスク終了条件］が 'True' となったとき

Magic エンジンがタスクの実行を終了するときは、レコードレベルの処理がまず最初に終了します。

レコードレベル処理の終了

オンラインタスクやリッチクライアントタスクではレコードの変更があった場合にのみ、［レコード後］ロジックユニットが実行され、レコード内容が保存されます。レコードの変更がない場合には何も実行しません。

タスクの終了

タスクを終了させる方法には、エンドユーザの操作により終了させる場合、［タスク終了条件］が［True］となる場合、あるいは［終了］イベントが発生する場合などがあります。

タスクが終了するときには次の処理が内部で行われます。

- ［タスク後］ロジックユニットの処理コマンドが実行されます。
- I/O デバイスがクローズされます。
- タスクのデータソースがクローズされます。

［タスク後］ロジックユニットはレコードレベルの処理が終了した後に実行されます。したがってその実行時にはすでにデータ項目の値は信頼できるものではなくなっていることに留意してください。

以下に示す表は、Magic xpa のロジックユニットトリガと実行ルール、およびロジックユニットの実行順についてまとめたものです。

ロジックユニット	トリガ	次のロジックユニット	次のロジックユニットの実行条件（IF）	（Else）実行
タスク前	タスクの実行開始	レコード前	データビューに最低一つのレコードがある	タスク後
レコード前	レコードの読込み	コントロール前	オンラインタスク／リッチクライアントタスクである	レコード後
コントロール前	コントロールにパーク	コントロール検証	（特になし）	
コントロール検証	コントロール編集が終了またはカーソルがコントロールを通過	コントロール後		（特になし）
コントロール後	コントロール編集が終了	項目変更	項目値が変更された	レコード後
項目変更	項目値が変更された	レコード後	オンラインタスク／リッチクライアントタスクでレコードが保存される	タスク後
タスク後	タスクが終了			

- **前の表は各ロジックユニットの１実行サイクルを示しています。もしレコードまたはコントロールのサイクルが終了しない場合は、その［前］処理から［後］処理までを反復することになります。**
- **コントロールロジックユニットはレコードロジックユニットサイクルの内部で実行され、同様にレコードロジックユニットは、タスクロジックユニットの内部で実行されるようになっています。**
- **親となるロジックユニットは、その下位にあるサブロジックユニットが終了しない限り、終了することができません。**

５．要約

本章では次のような Magic エンジンの仕組みについて学びました。

- イベントドリブンの方法と、3 つの要素すなわちイベント、トリガ、ハンドラについて
- タスクのメインソースとタスクタイプ、およびタスクの実行プロセスについて
- タスクレベル、レコードレベル、コントロールレベル、項目レベルなどのタスクの主なレベルについて
- タスクの各レベルが異なるロジックユニットにより実行制御されていること
- ロジックユニットとそのトリガおよび役割について
- Magic エンジンにはいろいろな場合に対応して実行される豊富な組込み処理があること

たとえばオンラインタスクでメインデータソースのカラムを選択している場合、タスクの実行にともなって、データソースをオープンし、データソースからそのカラムデータを読込み、データを表示し、エンドユーザが内容を変更したときにはデータソースにその変更を保存するといった一連の処理が Magic エンジンには組込まれていて、自動的にこれらの処理の実行を制御しているのです。

本章の結論として、次のことを覚えておいてください。すなわち、Magic xpa では、［どのようにプログラムを構成しなければならないかではなく、（すでにある枠組みの中で）そこで何がなされるべきか］を明確にすればよいということです。

さて、本章で Magic エンジンの仕組みについて多く学んだので、タスク実行エンジンの処理ルールを利用した、より効率的なプログラムが作成できるようになりました。

第 10 章 イベント

本章と次章では、Magic xpa のイベントとハンドラの考え方と仕組みについて説明します。

本章ではイベントの定義と発動について、次章でイベントの処理について説明します。

キーワード

- イベントドリブン
- トリガ
- イベント
- ハンドラ
- ［イベント実行］処理コマンド
- ［イベント実行］特性

学習目標

本章を学ぶことで、次の内容に対する理解を深めましょう。

- Magic xpa における［トリガ］
- Magic xpa における［イベント］
- イベントの定義方法
- イベントの発動方法
- プロジェクトでのイベントドリブンロジック活用法

参照

Magic ヘルプの［Magic xpa リファレンス／ Magic エンジン／ユーザ定義イベント］

１．はじめに

イベントドリブンプログラミングと手続き型プログラミング

イベントドリブンプログラミングは、ある事象の発生を想定し、それに対応するプログラムを事象単位に記述し、それぞれがオンデマンドで実行されるものです。

手続き型プログラミングは、あらかじめ定義された手順に従い、コマンドを一つ一つ順に処理するようにプログラムを構成するものです。

イベントドリブン処理についての必要知識

イベントドリブンプログラムを実際に作成するとき、トリガ、イベント、ハンドラについて理解しておくことが必要です。すなわち、Magic xpa でのイベント、それに対するトリガの設定、対応するハンドラの作成方法を理解してください。

本章では Magic xpa で扱うさまざまなタイプのイベントと、そのイベントをプロジェクト全体に適用したり、あるいは特定のタスクのみに適用する方法を学習します。

またトリガのいろいろな定義方法についても学習します。なお、イベントを処理するハンドラの詳細については次章で扱います。

イベントとハンドラのコンセプト

イベントドリブンプログラミングは 3 つの要素で構成されます。すなわち、イベント、トリガ、ハンドラです。

- イベントは事象を論理的に定義したものです。
- トリガはイベントを成立させるアクションであり、エンドユーザによって外部で実行されるものと、Magic xpa 内部のコードまたはプログラムコードによって内部で実行されるものがあります。
- ハンドラはイベントのトリガが実行されたとき、それに呼応して実行されるプログラムコードの単位です。

あるイベントには一つ以上のトリガまた一つ以上のハンドラを対応づけることができます。それぞれのトリガは異なるシナリオでイベントを発動させ、各シナリオに対して、一つ以上のハンドラがイベントの対応処理を行います。

Magic xpa のイベントは、メインプログラムに定義することでプロジェクト全体に適用されるようになります。このようなイベントはプロジェクト内のどのプログラムを実行しているかに関わらず、いつでもトリガをかけることができます。またイベントは特定のタスクに関連づけることができ、そのイベントを定義したタスクのみ、あるいはそのタスクとサブタスクに適用範囲を限定することができます。

［イベント］テーブルについて

Magic xpa のユーザ定義イベントは［イベント］テーブルに定義します。

その特性について次表にまとめます。

特性	説明
#	自動的に生成されるイベント番号
説明	イベントの名前を指定します。ハンドラを定義する際にこの名前でイベントを選択します。
トリガタイプ	イベントのためのトリガとして使用するイベントのタイプを指定します。 以下のタイプが指定できます。 • S= システム…… キーボード操作の組み合わせ • I= 内部……Magic エンジンに組み込まれているイベント • T= タイマ……定義された間隔ごとに発生するイベント • E= 式……式が［True］と評価された場合に発生するイベント • N= なし……トリガの指定は不要です。［ユーザイベント］と同じように開発者が任意に定義することのできるイベントです。

特性	説明
トリガ	イベントを発生させるためのトリガを指定します。 ［トリガタイプ］で［N= なし］と設定された場合以外は、トリガを指定する必要があります。 ここからズームすると［トリガタイプ］の指定に基づいて以下のトリガを指定します。 • S= システム……［キー設定］ダイアログが表示され、そのイベントをトリガするキーの組み合わせを入力できます。 • I= 内部……［イベント一覧］が表示され、ここから内部イベントを選択します。 • T= タイマ……イベントの発生間隔を指定します。 • E= 式……［式］エディタが表示されます。ここに条件式を定義します。
パラメータ	プロジェクトとユーザ定義イベントの間でデータを受け渡しをする場合に使用するパラメータを定義することができます。 ここからズームすると［パラメータ］テーブルが開きます。
強制終了	Magic エンジンが［イベント］ロジックユニットとして定義されている一連の処理コマンドを実行する前に終了するレベルを指定します。 次のようなオプションがあります。 • N= なし……どのレベルに対しても終了処理を行わず、イベントが発生したコントロール の編集モード内でハンドラ処理を実行します。 • E= 編集…… イベントを実行する前に既存のコントロールの編集モードを終了します。このとき、Magic エンジンは、以下の順番に処理を実行します。 1. 編集モードを終了します。 2. 編集項目の内容を更新します。 3. 更新された項目に関係する全ての値を再計算します。 4. データが変更された場合、［項目変更］ロジックユニットを実行します。 5. ［イベント］ロジックユニットを実行します。 6. コントロールの編集モードに戻ります。 実行されたハンドラ内でコントロール項目を参照する場合、その項目は編集された値として参照されます。 このオプションは［エディット］、［リッチエディット］、および複数選択の［リストボックス］コントロールで有効で、編集状態の時に項目の内容と異なる値の場合に更新されます。 • C= コントロール……イベントを実行する前に現在のコントロールレベルを終了します。このとき、Magic エンジンは、以下の順番に処理を実行します。 1. 編集モードを終了します。 2. 編集項目の内容を更新します。 3. 更新された項目に関係する全ての値を再計算します。 4. ［項目変更］ロジックユニットを実行します。 5. ［コントロール検証］と［コントロール後］ロジックユニットを実行します。 6. ［イベント］ロジックユニットを実行します。 7. ［コントロール前］ロジックユニットを実行します。 8. コントロールの編集モードに戻ります。 • R= レコード更新前……イベントを実行する前に現在のレコードレベルを終了します。このとき、Magic エンジンは、以下の順番に処理を実行します。 1. 編集モードを終了します。 2. 編集項目の内容を更新します。 3. 更新された項目に関係する全ての値を再計算します。 4. ［項目変更］ロジックユニットを実行します。 5. ［コントロール検証］と［コントロール後］ロジックユニットを実行します。 6. レコードが変更された場合、［レコード後］ロジックユニットを実行します。 7. レコードが変更された場合、テーブルのレコードを更新します。 8. ［イベント］ロジックユニットを実行します。 9. レコードが更新された場合、テーブルを更新します。 10. ［イベント］ロジックユニットを実行します。 11. レコードが更新された場合、テーブルを更新します。

特性	説明
	［イベント］ロジックユニットが実行されるとき、レコードはまだ更新されていないため、［イベント］ロジックユニットからレコードを参照したときレコードが更新されていないものを見ることになります。 • P= レコード更新後……Magic エンジンは、以下の順番に処理を実行します。 1. ［コントロール検証］ロジックユニットを実行します。 2. ［コントロール後］ロジックユニットを実行します。 3. ［レコード後］ロジックユニットを実行します。 4. レコードを更新します。 5. ［レコード前］ロジックユニットを実行します。 6. ［イベント］ロジックユニットを実行します。 7. ［コントロール前］ロジックユニットを実行します。 ［イベント］ロジックユニットが実行されるとき、レコードはすでに更新されているので、［イベント］ロジックユニットからレコードを参照すると、更新された内容を参照できます。

２．イベントの作成（実習）

このセクションでは、＜ゴールド会員＞および＜日付と時刻の設定＞イベントを定義してみましょう。

＜ゴールド会員＞イベントはトリガを定義しません。したがって、このイベントは［イベント実行］処理コマンドにより発動させます。

＜日付と時刻の設定＞イベントはキーボードの 'Ctrl+A' の組合せ入力をトリガとすることにします。

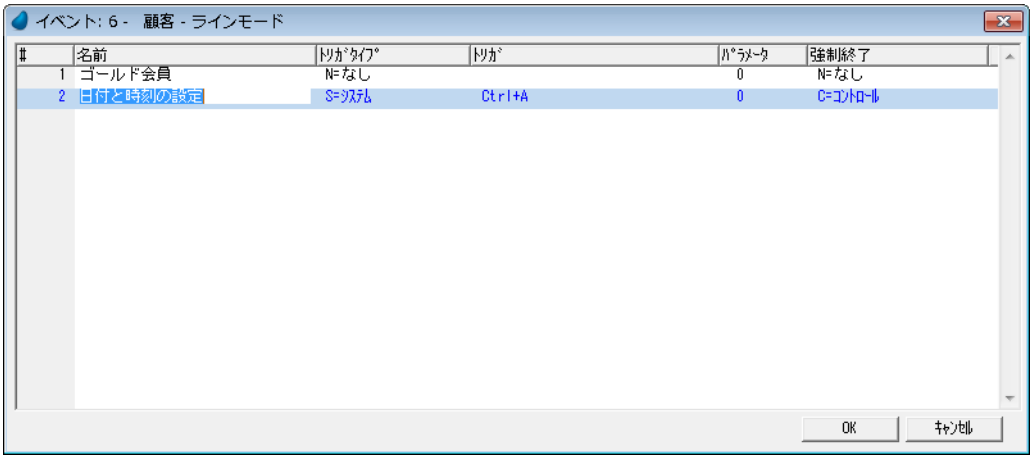
1. ［プログラム］リポジトリ（Shift+F3）を開きます。
2. ＜顧客－ラインモード＞プログラムにズーム（F5）します。
3. ［タスク環境］メニューから［ユーザイベント］（Ctrl+U）を選択します。

4. ［イベント］テーブルに一行作成（F4）します。
5. パラメータを下表のように設定し、＜ゴールド会員＞イベントを定義します。
6. 二つ目のイベントを４－５のステップを繰り返して作成します。

二つ目のイベントの［トリガ］は次のように設定します。

7. ［トリガ］欄からズーム（F5）して［キー設定］ダイアログを開きます。
8. Ctrl キーと A キーを同時に押下します。
9. ［OK］ボタンをクリックします。

#	説明	トリガタイプ	トリガ	パラメータ	強制終了
1	ゴールド会員	N= なし		0	N= なし
2	日付と時刻の設定	S= システム	Ctrl+A	0	C= コントロール



３．イベントの発動

イベントテーブルで定義する［トリガタイプ］と［トリガ］によってイベントが発動されます。

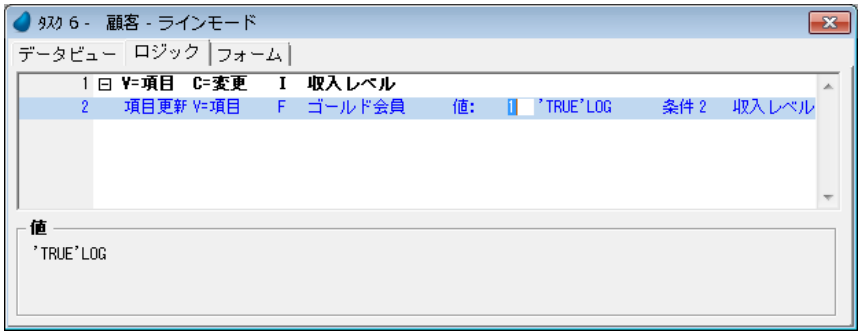
しかし［トリガタイプ］と［トリガ］が指定されている場合であっても、それに関係なくアプリケーションロジックによりイベントを実行させたいときがあります。

そのようなときは、［イベント実行］処理コマンドを使用します。

［イベント実行］コマンドによるイベントの発動

＜収入レベル＞が 4,000,000 を超える場合は、＜ゴールド会員＞の値を［True］にする処理の作成を考えてみましょう。

この場合、もっとも基本的な方法は、［項目更新］処理コマンドを使用して、＜ゴールド会員＞の値を［True］にすることです。下図はこのような例を示しています。



しかし顧客のステータスを ' ゴールド会員 ' に変更することは、通常、単に＜ゴールド会員＞の値を［True］にすることだけではありません。（詳細は略します。）

また、顧客のステータスの変更は、＜収入レベル＞以外の要素により行われることもあります。

イベントドリブンの方法を用いれば、顧客のステータスを変更する処理は一箇所（イベントハンドラ）で定義し、変更理由（トリガ）については、複数の場所で定義することが可能になります。

各プロセスに名前を付けることで、トリガとハンドラでどのような参照が行われたかを知ることができます。

［イベント実行］コマンドを使ったイベント発動

ここでは、＜収入レベル＞の変更後に＜ゴールド会員＞イベントを実行するようなロジックユニットを作成してみましょう。

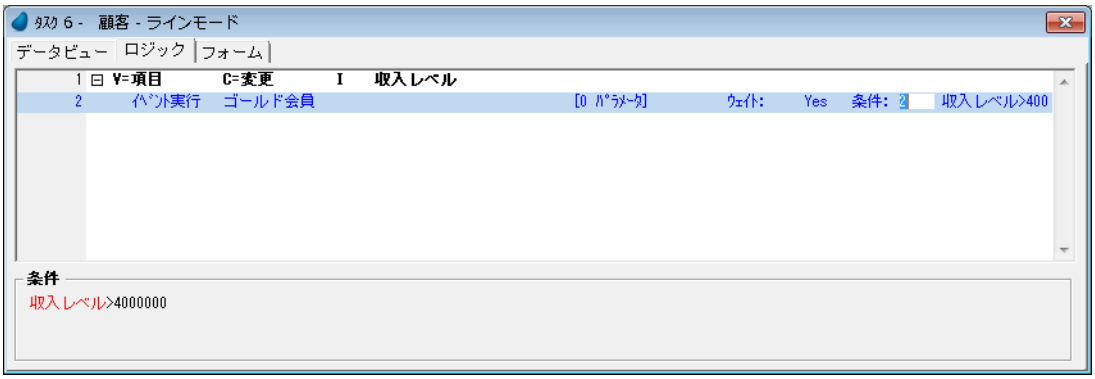
1. ＜顧客－ラインモード＞プログラム（#6）を開きます。
2. ［ロジック］エディタを選択します。
3. 新しいヘッダ行（Ctrl+H）を作成します。
4. ［項目変更］ロジックユニットを選択し、項目として＜収入レベル＞を指定します。
5. 新しく一行作成（F4）します。
6. ［イベント実行］処理コマンドを選択します。
7. パラメータを下表のように設定します。

コマンド	イベント	パラメータ	ウェイト	条件
イベント実行	ゴールド会員		Yes	収入レベル >4,000,000

イベントの設定は次のようにします。

8. ［イベント設定］欄からズームして［イベント］ダイアログを開きます。
9. ［イベントタイプ］として［U= ユーザ］を選択します。
10. ［イベント］欄からズームして、イベント一覧を開きます。
    a. ＜ゴールド会員＞イベントにパークします。
    b. ［選択］ボタンをクリックします。

11. ［OK］ボタンをクリックして［イベント］ダイアログを終了します。



この例では、＜収入レベル＞の値が変更され、かつその値が 4,000,000 よりも大きくなったときに［イベント実行］処理コマンドが有効になり、＜ゴールド会員＞イベントが発動されます。

つまり、タスクに組込まれたロジックにしたがい、［イベント実行］処理コマンドによってイベントが発動されるのです。

イベント が発動されただけでは、まだ Magic xpa は イベント に対応する処理を行いません。イベント に対応するハンドラが存在する場合に、はじめて処理が行われます。

プッシュボタンによるイベントの発動

［イベント実行］処理コマンドを利用してイベントを発動したように、タスクフォーム上のプッシュボタンを利用してイベントを発動することができます。

ただし、プッシュボタンを利用する場合には、次のことができません。

- ［イベント］ロジックユニットに対してパラメータを渡すこと。
- イベント処理が終了するまでタスクの実行を待機すること。

［イベント実行］処理コマンドやプッシュボタンを利用してイベント発動を行うことができるイベントのタイプは、次のようなものです。

- U= ユーザ …… ［イベント］テーブルに登録されたすべてのイベントが該当します。
- I= 内部 …… Magic xpa の内部イベント
- S= システム …… 特定のキーボード入力の組合せに対応付けられた Magic xpa の内部イベント

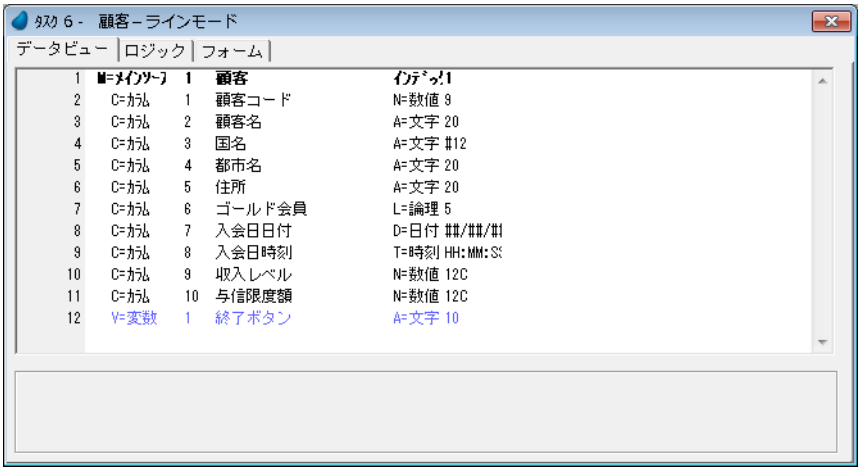
ここでは次のようにして［終了］プッシュボタンをタスクフォームに追加します。

［終了］イベントは、Magic xpa の内部イベントであり、現在のタスクを終了するように Magic xpa を制御します。

1. ＜顧客ーラインモード＞プログラムにズーム（F5）します。
2. ［データビュー］エディタを選択します。
3. 最後の行にパークし、一行作成（F4）します。

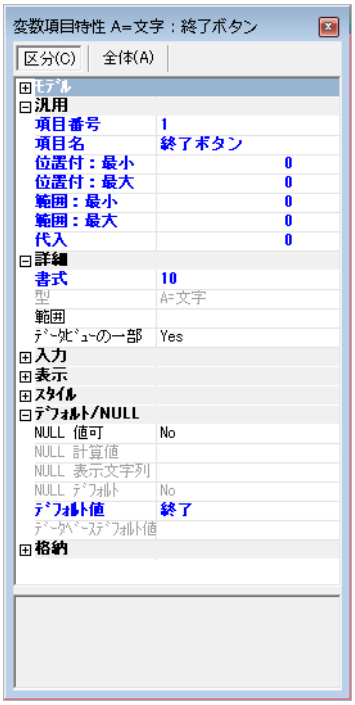
4. 次のようにパラメータを設定します。

項目タイプ	項目番号	項目名	モデル	型	書式	範囲	終了	代入
V= 変数	1	終了ボタン	0	A= 文字	10	0	0	0

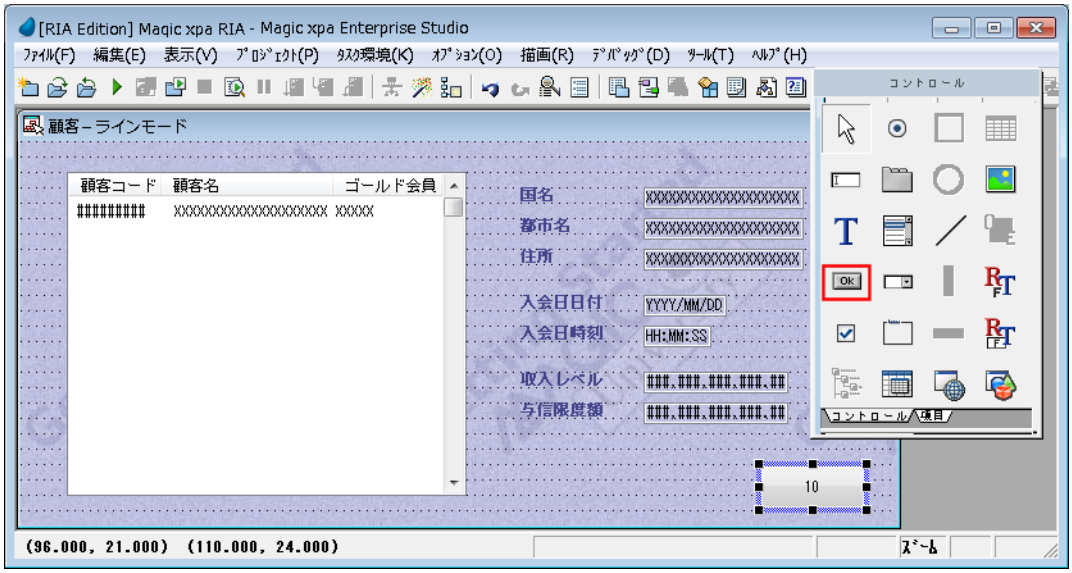


5. ［変数項目］特性（Alt+Enter）を開きます。
6. デフォルト値として**＜終了＞**と入力します。

［デフォルト値］特性には（変数）項目に設定するデフォルト値を指定できます。

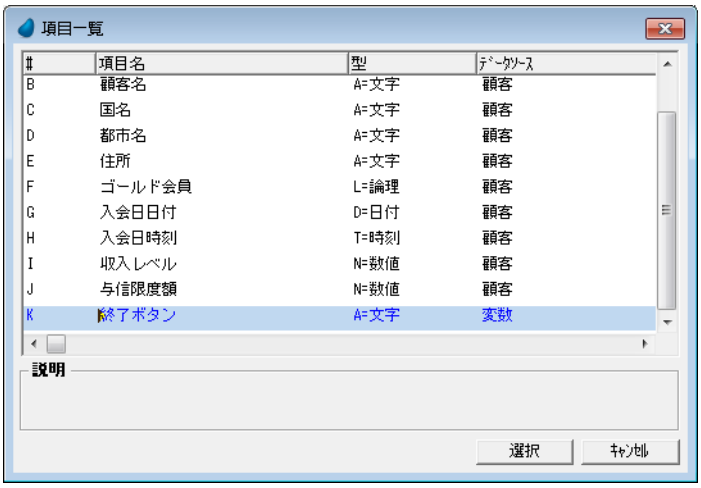


7. ［フォーム］エディタを開きます。
8. ＜顧客－ラインモード＞フォームにズームします。
9. ［コントロール］パレットで［プッシュボタン］コントロールを選択し、フォームの適当な位置に追加します。



10. ［プッシュボタン］コントロールの特性を開きます。

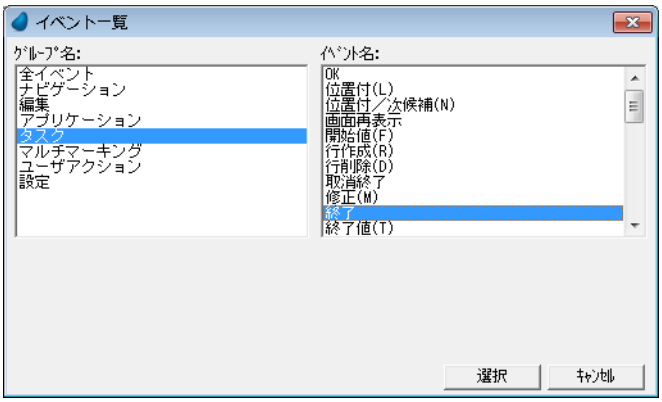
11. ［データ］特性から項目一覧にズームし、＜終了ボタン＞を選択します。

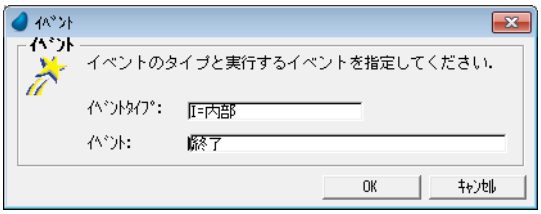


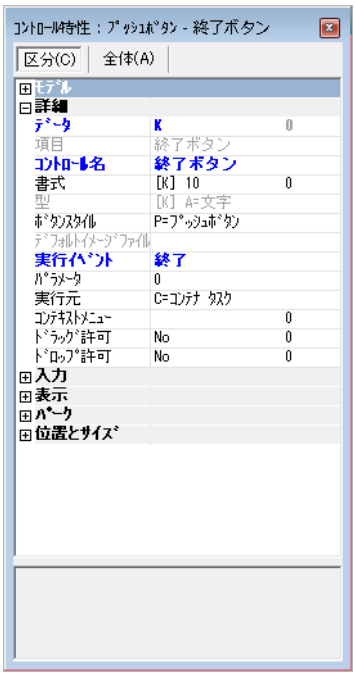
12. ［実行イベント］特性から［イベント］ダイアログにズームします。

a. ［イベントタイプ］として、［I= 内部］を選択します。

b. ［イベント］欄からイベント一覧にズームして、［終了］イベントを選択します。







13.変更内容を保存し、プログラムを閉じます。

これでプッシュボタンにイベント発動を行わせることができます。つまり、エンドユーザがこのプッシュボタンをクリックすることで、［終了］イベントが発動されるようになりました。

４．結果の確認

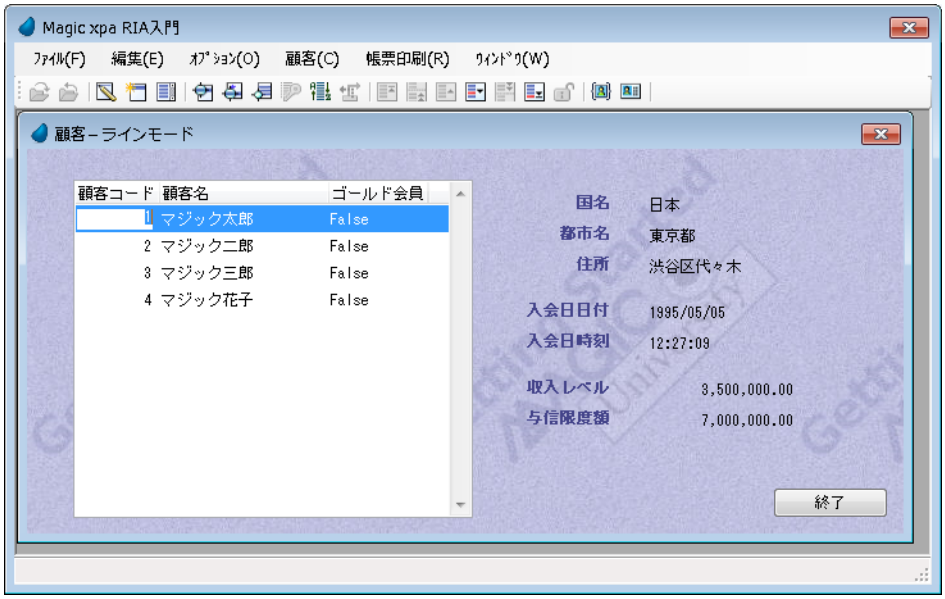
ここまでの作業で、＜顧客－ラインモード＞プログラムに三つのイベントを定義しました。

- 二つのユーザイベントをタスクに定義しました。
  - ＜ゴールド会員＞イベント …… タスクロジックによって、［イベント実行］処理コマンドにより発動されます。
  - ＜日付と時刻の設定＞イベント …… キー入力の組合わせ（Ctrl+A）によって発動されます。

  これらはイベントの定義だけを行いました。（対応ハンドラについては次章で扱います。）
- 三つ目は Magic xpa の内部イベントである［終了］イベントです。このイベントはタスクの実行を終了させます。

ここまでの段階では、［終了］イベントだけについて、その動作を確認できます。すなわち Magic xpa の内部イベントはハンドラも内部にあらかじめ組込まれているからです。

1. ＜顧客－ラインモード＞プログラムを実行してください。（F7）
2. ＜終了＞ボタンをクリックして、タスクが終了することを確認しましょう。



５．要約

本章ではイベントドリブンの仕組みについて学習しました。

イベントドリブンプログラミングの三要素は、イベント、イベントを発動させるトリガ、具体的な処理を行うハンドラですが、本章ではイベントとその発動のさせ方を学習しました。

- ユーザ定義イベントは次のようなタイプに分類されるトリガによって発動されます。
  - S= システム …… ［キー設定］ダイアログが表示され、そのイベントをトリガするキーの組み合わせを入力して設定します。
  - I= 内部 …… ［イベント一覧］が表示され、ここから内部イベントを選択します。
  - T= タイマ …… イベントの発生間隔を指定します。
  - E= 式 …… ［式］エディタが表示されます。ここに条件式を定義します。

ユーザ定義イベントは、タスクのイベントテーブルにて定義します。

- その他、［イベント実行］処理コマンドを使ったり、［プッシュボタン］コントロールを使用する方法もあります。

次章で、イベントハンドラについて説明し、実習します。

第 11 章 ハンドラ

本章では［イベント］ロジックユニットについて学び、10 章で作成したイベントの対応ハンドラを作成します。

キーワード

- イベントドリブン
- イベント
- トリガ
- ハンドラ
- ［イベント］ロジックユニット
- ウェイト
- 伝播

学習目標

本章を学ぶことで、次の内容に対する理解を深めましょう。

- Magic xpa でのイベント処理方法
- Magic xpa がイベント実行条件をチェックする方法
- イベント伝播の仕組み
- イベントの同期
- Magic xpa 内部イベントハンドラへのロジック追加と書き換え

参照

Magic ヘルプの［Magic xpa リファレンス／ Magic エンジン／エンジン実行レベル／イベントレベル］

１．はじめに

Magic xpa の［イベント］ロジックユニットは、特定のイベントが発動されたときに実行される一連のコマンドによって構成されます。イベントの発生によって［イベント］ロジックユニットが処理される内部機構をイベントハンドラと定義しています。

Magic xpa には次のようなハンドラがあります。

- 内部イベント用ハンドラ …… Magic エンジンに組込まれたハンドラで、Magic xpa 内部イベントを処理します。たとえば、内部イベント［次行］のハンドラは、現在のレコードから次のレコードにカーソルを移動させます。
- 組込みイベント用ハンドラ …… Magic エンジンに組込まれたハンドラで、第 9 章で学んだ異なるタスクレベルに適用される Magic エンジンの実行ルールを実行します。
- ユーザ定義イベント用ハンドラ …… 開発者が作成するハンドラで、［イベント］ロジックユニットとして定義します。

本章では［イベント］ロジックユニットによるユーザ定義イベントの処理と、［イベント］ロジックユニットの実行がどのように行われるかを学びます。

［イベント］ロジックユニット

［イベント］ロジックユニットは、イベントに対応した実際の処理を行なう、ハンドラを定義します。

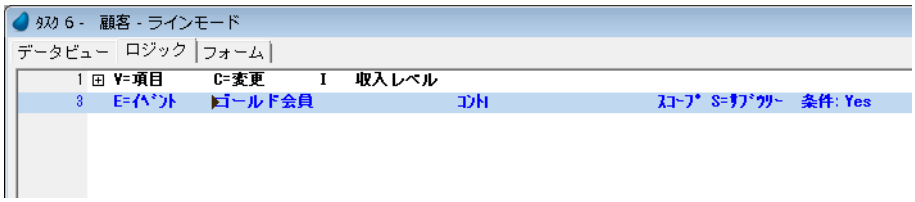
［イベント］ロジックユニットは、プロジェクト実行中に対応イベントが発行されたときにのみ実行されます。

［イベント］ロジックユニットは異なるタイプのイベントに対応しています。

- ユーザ定義イベント …… 開発者によってユーザ定義イベントテーブルに登録されたイベント。
- Magic xpa 内部イベント …… 通常は Magic xpa が管理するイベントに対しても、開発者は新規に、または追加のハンドラを作成できます。
- システムイベント …… キー入力の組合わせによって発動されるイベント。
- タイマーイベント …… 定義された時間間隔で発動されるイベント。
- 式イベント …… 定義された式が 'True' と評価された時に発動されるイベント。
- エラーイベント …… データベースに関連したエラーが発生したときに発動されるイベント。
- .Net イベント …… .Net オブジェクトから発動されるイベントです。本コースでは扱いません。

イベントロジックユニットに設定するパラメータ

タイプ	イベント名	コントロール名	スコープ	条件
E= イベント	処理するイベント名	イベントを割当てるコントロール名	イベントのスコープ	実行条件（式）



イベント名は次のようにして取得します。

1. ここからズームして［イベント］ダイアログを開きます。
2. ［イベントタイプ］をドロップダウンリストから選択します。
3. ［イベント］欄からズームしてイベント一覧から、該当するイベント名を選択します。
4. コントロール名は以下のどちらかの方法で設定します。
   - 直接コントロール名を入力します。
   - ここからズームしてコントロール一覧を開き、選択します。
5. イベントのスコープは以下のどちらかを指定できます。
   - タスク …… イベントがそのタスク実行中に発動された場合のみハンドラが実行されます。
   - サブツリー …… イベントがそのタスクか、またはサブタスク実行中に発動された場合にハンドラが実行されます。

２．同一イベントに対応する複数のハンドラの処理

場合によって、一つのイベントに対応する複数の［イベント］ロジックユニットを定義することがあります。

Magic エンジンのデフォルトの動作では、発動されたイベントは（Magic エンジンのルールに従って）最初に対応づけられた［イベント］ロジックユニットの処理を行い、その後そのイベントはクリアされます。したがって、もし［イベント］ロジックユニットが他に定義されていたとしてもその処理は行われません。

たとえば、＜印刷＞というユーザ定義イベントに対して、二つの［イベント］ロジックユニットが定義されており、一つは顧客からの注文書を印刷し、他方はその注文に対する顧客への請求書を印刷するとします。このとき、Magic エンジンの実行ルールにより、どちらかのロジックユニットのみが＜印刷＞イベントの発動に対応して処理を行います。

どのハンドラがイベントを処理するのでしょうか？

Magic エンジンの実行ルールにより、実行ツリー上の最下位にあるハンドラがイベントを処理します。たとえば上記の例において、開発者が注文書を印刷するロジックユニットを請求書印刷のためものよりも（［ロジック］エディタ内の位置で）後に定義した場合、＜印刷＞イベントを処理するのは注文書を印刷するロジックユニットになります。

両方のハンドラに処理を行わせたい場合はどうしたらよいのでしょうか？

上記の例において、注文書を印刷する［イベント］ロジックユニットの［伝播］特性を［Yes］に設定します。この設定により、Magic エンジンは注文書印刷のハンドラ実行後も＜印刷＞イベントを発動された状態に維持します。

内部ハンドラに処理の追加を行いたい場合はどうしたらよいのでしょうか？

Magic エンジン内部のハンドラを直接変更することはできません。しかし内部のハンドラが内部イベントを処理する前に、［イベント］ロジックユニットに定義するハンドラが先に処理を行うように設定することができます。Magic エンジンに組込まれている内部ハンドラは、常に実行ツリー上の最上位に位置しているためです。

分かりやすい例として、［終了］イベントについて操作してみましょう。

プログラムの終了前に、顧客からの注文書を印刷するかどうかをエンドユーザに確認したいような場合は、［終了］イベントに対するハンドラ処理に手を加える必要があります。このようなときには、［終了］イベントに対応して［イベント］ロジックユニットに新しく定義した注文書印刷の確認と実行を行うハンドラの［伝播］特性を［Yes］とすることで、引き続き内部の［終了］ハンドラが実行され、印刷処理後にプログラムが終了するようになります。

［伝播］特性

［伝播］特性による動作の違いは次の通りです。

- Yes …… Magic エンジンは、実行ツリー上のより上位に定義されているハンドラを探し、実行しようとします。（定義されていないときは何も実行されません。）
- No …… Magic エンジンはイベントの発動状態をクリアするため、上位のハンドラは実行されません。

３．＜ゴールド会員＞イベントのハンドラ

前の章で＜ゴールド会員＞イベントを定義し、［イベント実行］処理コマンドを使用してそのイベントを発動しました。

ここでは、そのイベントに対応する［イベント］ロジックユニットを作成しましょう。

この例では、＜収入レベル＞が 4,000,000 よりも大きい場合には＜ゴールド会員＞の値を 'True' にするような＜ゴールド会員＞ハンドラを作成します。

以下の手順で［イベント］ロジックユニットに処理を定義します。

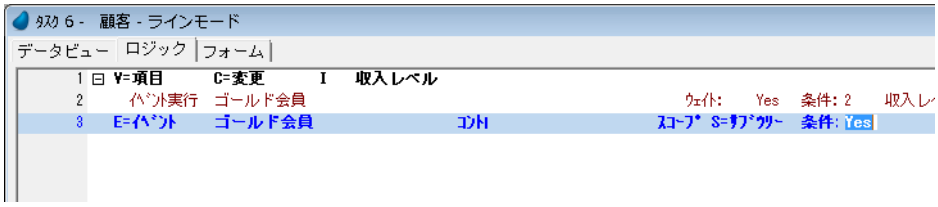
1. ＜顧客ーラインモード＞プログラムにズーム（F5）します。
2. ［ロジック］エディタを選択します。
3. ［ロジック］エディタの最終行にパークします。
4. ヘッダ行（Ctrl+H）を一行作成します。

5. 次のようにパラメータを設定します。

タイプ	イベント名	コントロール名	スコープ	条件
E= イベント	ゴールド会員		S= サブツリー	Yes

イベントの設定は以下の手順で行います。

6. ［イベント］欄からズームして［イベント］ダイアログを開きます。
   a. ［イベントタイプ］として、［U= ユーザ］を選択します。
   b. ［イベント］欄からズームして［イベント一覧］ダイアログを開きます。
   c. ＜ゴールド会員＞にパークし、［選択］ボタンをクリックします。
7. ［イベント］ダイアログの［OK］ボタンをクリックします。



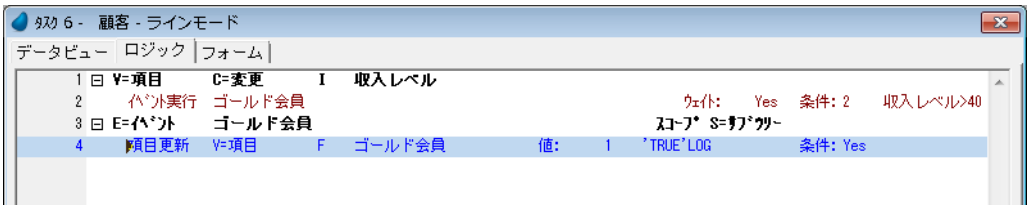
8. ［イベント］ロジックユニットのヘッダ行にパークし、一行作成（F4）してください。
9. 次のようにパラメータを設定しましょう。

コマンド	項目	値	条件
項目更新	F（ゴールド会員）	1（'TRUE'LOG）	2 （収入レベル >4,000,000）

項目の選択は、項目記号欄から項目一覧にズームして選択します。

値欄の設定は次のようにします。

10. ［値］欄からズームして［式］エディタを開きます。
11. 一行作成し、＜ ’TRUE’LOG ＞ と入力します。
12. ［OK］ボタンをクリックします。



これで＜ゴールド会員＞ハンドラが作成されました。

４．＜日付と時刻の設定＞イベントのハンドラ

前章で、プログラム実行中に Ctrl+A キー入力を行うことで発動される、＜日付と時刻の設定＞イベントを作成しました。

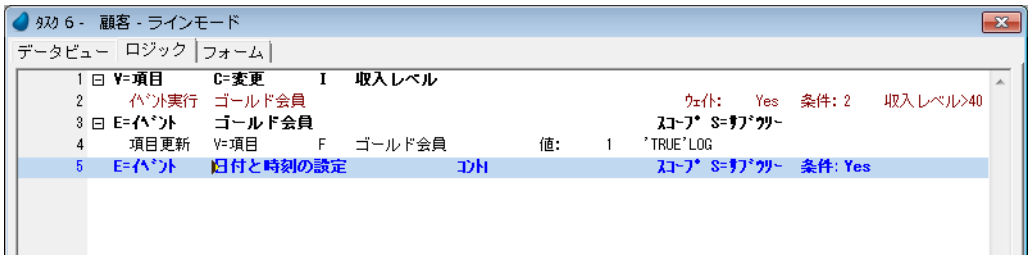
ここでは、上記イベントが発動されたときに、＜入会日日付＞と＜入会日時刻＞の値が、そのときのシステム日付と時刻になるようなロジックを作成します。

以下の手順で＜日付と時刻の設定＞の［イベント］ロジックユニットを作成しましょう。

1. ＜顧客ーラインモード＞プログラムにズーム（F5）します。
2. ［ロジック］エディタを選択します。
3. ［ロジック］エディタの最終行にパークします。

4. ヘッダ行（Ctrl+H）を一行作成します。
5. 次のようにパラメータを設定します。

タイプ	イベント名	コントロール名	スコープ	条件
E= イベント	日付と時刻の設定		S= サブツリー	Yes



6. ヘッダ行にパークしていることを確認します。
7. 新しく二行作成し、下表のようにパラメータを設定します。

コマンド	項目	値	条件
項目更新	G　（入会日日付）	3　（Date()）	Yes
項目更新	H　（入会日時刻）	4　（Time()）	Yes

項目と値の設定方法はすでに前のセクション等で説明してあるので、ここでは省略します。

これで＜日付と時刻の設定＞ハンドラが作成されました。

8. 内容を保存し、プログラムを閉じます。

Date() 関数はシステムの日付、Time() 関数はシステムの時刻をそれぞれ返します。

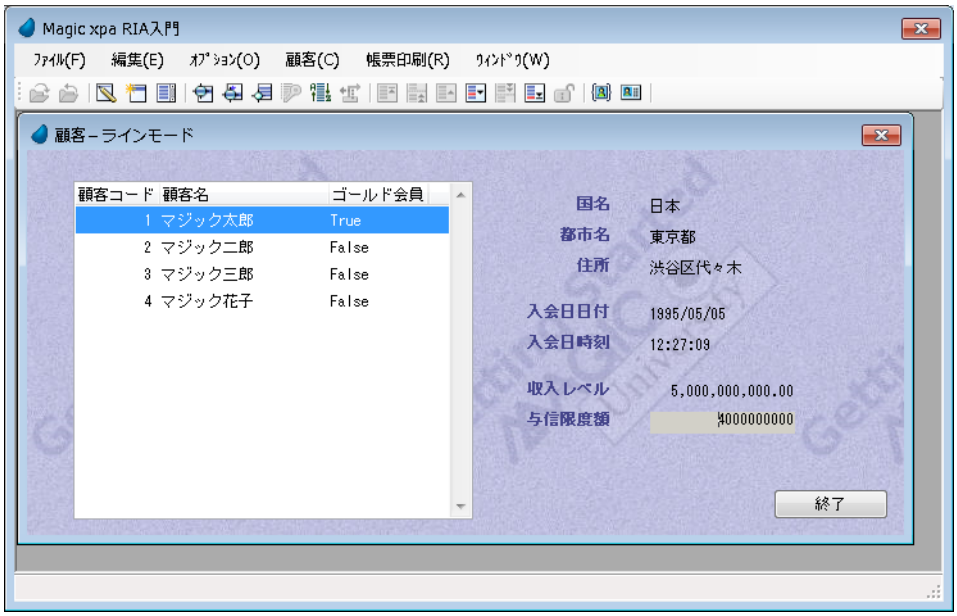
５．動作結果の確認

＜ゴールド会員＞イベントの発動とハンドラの実行結果

1. ＜顧客－ラインモード＞プログラムを実行（F7）します。
2. 最初の顧客データにパークします。
3. ＜ゴールド会員＞のカラムに移動します。

4. 値が［True］になっているときは、［False］にします。
5. ＜収入レベル＞に移動します。
6. ＜収入レベル＞の値を＜ 5000000 ＞に変更してください。

7. Tab キーで次の項目に移動します。



このとき、＜ゴールド会員＞の値が［True］に変わっていることを確認してください。

＜収入レベル＞の値が変更され、4,000,000 よりも大きな値になったため、＜ゴールド会員＞イベントが発動され、ハンドラにより［True］が設定されたことが分かります。

＜日付と時刻の設定＞イベントの発動とハンドラの実行結果

1. 最初の顧客にパークします。
2. Ctrl+A のキー入力をしてください。

＜入会日日付＞と＜入会日時刻＞の値が、システム日付とシステム時刻になることが分かります。

このとき、＜入会日日付＞と＜入会日時刻＞の値は、カーソルがどこにパークしているかに関係なく更新されます。つまり値の更新を行うときに、カーソルをどこにパークさせておいてもよいことになります。このことはイベントドリブンによるプログラミングの特徴をよく示しています。

＜終了＞イベントの発動とハンドラの実行

3. ＜終了＞ボタンをクリックしてみましょう。

内部の［終了］イベントが発動され、ハンドラが実行され、その結果としてタスクの実行が終了します。

６．Magic エンジンによるイベントの監視

Magic エンジンはタスクタイプにより決まる方法によりタスクイベントの発動を監視しています。

オンラインタスクおよびリッチクライアントタスクの場合

- エンドユーザがキーを押下したとき。
- タスクが入力待ちとなりアイドル状態になったとき。これはタスクがエンドユーザのキー入力待ちとなり、［動作環境／動作設定／キーボード休止秒数］に設定された時間のタイムアウトが発生した状態です。

バッチタスクの場合

- ［動作環境／システム／バッチイベント間隔］に設定された時間のタイムアウトが発生したとき。
- バッチタスクの［タスク特性／動作／レコードイベント間隔］に設定された数のレコードが処理されたとき。

タスクタイプに関係なく［イベント実行］処理コマンドが使用されたとき

Magic エンジンは［イベント実行］処理コマンドの実行条件を評価します。［イベント実行］処理コマンドで［ウェイト］パラメータを［Yes］に設定したときには、Magic エンジンはイベントの対応ハンドラを検索しすぐに実行します。［ウェイト］パラメータを［No］に設定したときには、イベントはイベントキューに登録され、タスクタイプに従う上記の方法に従ってその発動が監視され、順に処理されてゆきます。

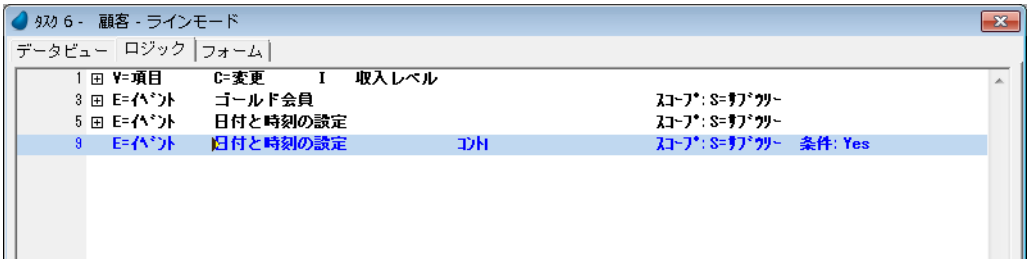
例題 1＜日付と時刻の設定＞イベントへのハンドラ追加

＜日付と時刻の設定＞イベントに対し、［イベント］ロジックユニットを追加してみましょう。この作業を通してイベントを監視するメカニズムに対する理解が深まります。

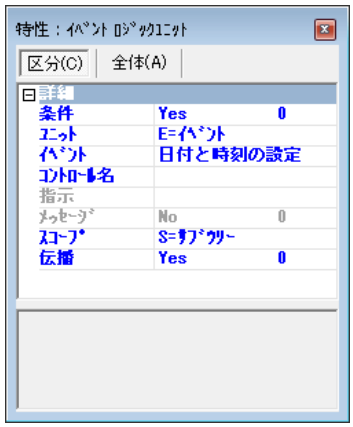
1. ＜顧客ーラインモード＞プログラムにズーム（F5）します。
2. ［ロジック］エディタを選択します。
3. 最終行にパークします。

4. ヘッダ行（Ctrl+H）を作成します。
5. ヘッダラインのパラメータを次のように設定します。

タイプ	イベント名	コントロール名	スコープ	条件
E= イベント	日付と時刻の設定		S= サブツリー	Yes



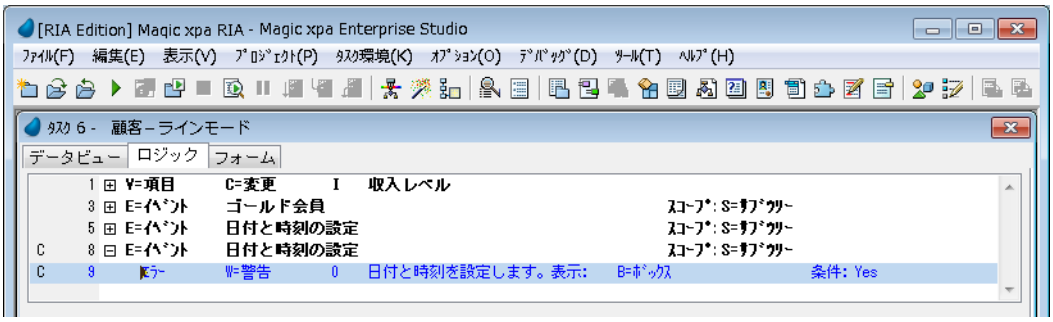
6. ［イベント］ロジックユニットの行（#8）にパークしロジックユニットの特性（Alt+Enter）を開きます。
7. ［伝播］特性を［Yes］に設定します。
8. ［イベント］ロジックユニットの行（#8）にパークします。
9. 一行作成（F4）します。
10. 作成した行のパラメータを次のように設定します。



タイプ	モード	テキスト	表示	条件
E= エラー	W= 警告	日付と時刻を設定します。	B= ボックス	Yes

11. 変更を保存し、プログラムを閉じます。

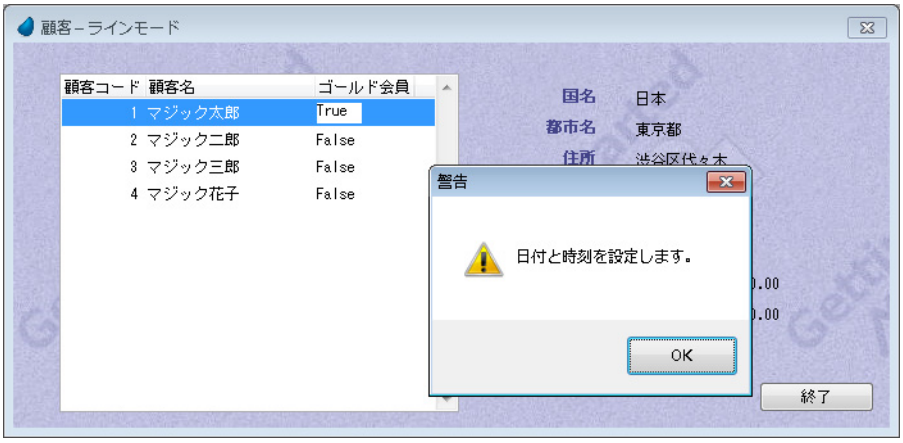
これで、＜日付と時刻の設定＞イベントのハンドラを一つ追加しました。



例題１の確認

1. ＜顧客－ラインモード＞プログラムを実行（F7）してください。
2. Ctrl+A のキーを押下してください。警告ボックスが表示されるはずです。
3. ［OK］ボタンをクリックします。

日付と時刻のデータが、それぞれシステムの日付と時刻になることを確認してください。



例題１の実行結果について

イベントが発動されたとき、Magic エンジンは対応して定義されているロジックユニットを探し、見つかればそれを実行します。

ロジックユニットの実行が終了したとき、その［伝播］特性が［Yes］に設定されているときのみ、Magic エンジンは他にロジックユニットが定義されていないか探します。（複数のロジックユニットが一つのイベントに対して定義されているときは、実行ツリーで最下位にあるロジックユニットが最初に実行されます。）［伝播］特性が 'No' のときは、それ以上ロジックユニットを探すことはせず、イベント処理の終了となります。

今回の例では、イベント＜日付と時刻の設定＞に対して二つのロジックユニットを定義しました。イベントが発動されたとき、最初のロジックユニットがまず実行された結果、＜日付と時刻が設定されます。＞という警告が表示され、さらにその［伝播］特性が［Yes］に設定されているので、二つ目のロジックユニットが実行され、＜入会日日付＞と＜入会日時刻＞のデータが、そのときのシステム日付とシステム時刻になりました。

例題２　内部イベントへのハンドラ追加

ここでは Magic エンジンの内部イベントである［終了］イベントに対応するロジックユニットを追加してみましょう。

1. ＜顧客－ラインモード＞プログラムにズーム（F5）します。
2. ［ロジック］エディタを選択します。
3. 最終行にパークします。

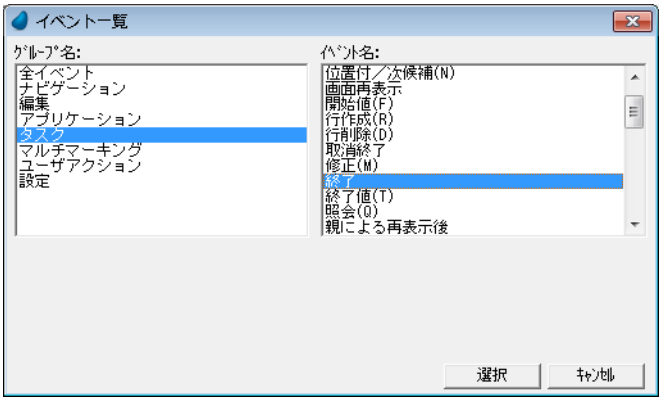
4. ヘッダ行（Ctrl+H）を作成します。

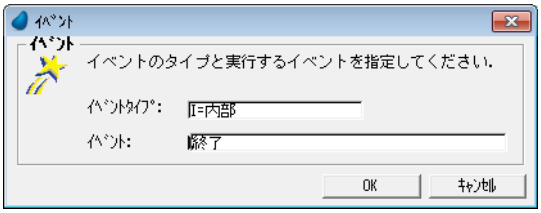
5. ヘッダラインのパラメータを次のように設定します。

タイプ	イベント名	コントロール名	スコープ	条件
E= イベント	終了		S= サブツリー	5

［イベント］欄への［終了］イベント設定は次のようにします。

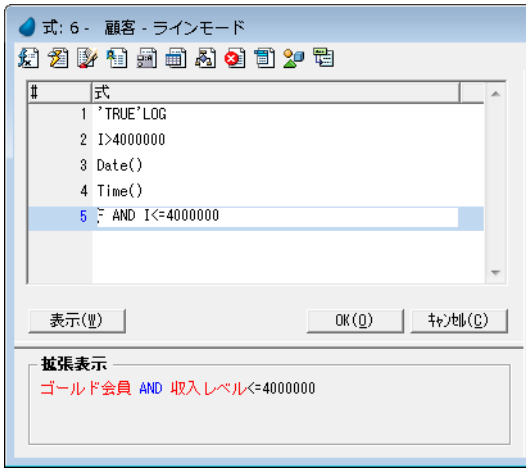
6. ズームして［イベント］ダイアログを表示します。
7. ［イベントタイプ］として［I= 内部］を選択します。
8. ［イベント］欄からズームして、イベント一覧を表意します。
9. グループ名で［タスク］を選択し、イベント名で［終了］を選択します。
10. ［選択］ボタンをクリックして［イベント］ダイアログに戻り、［OK］ボタンをクリックします。

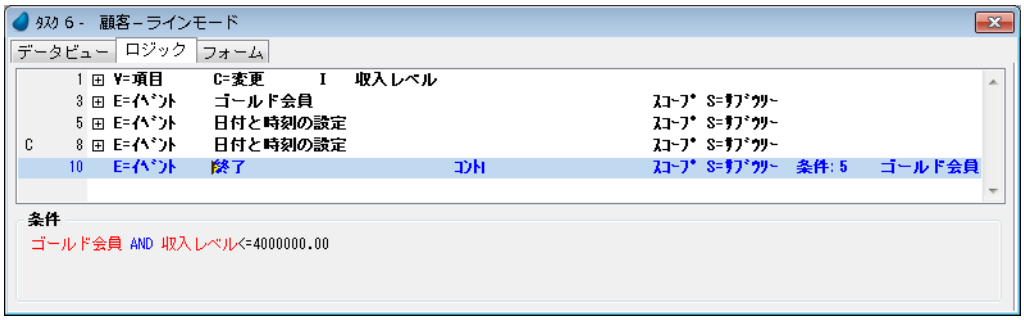




［条件］欄は次のように設定します。

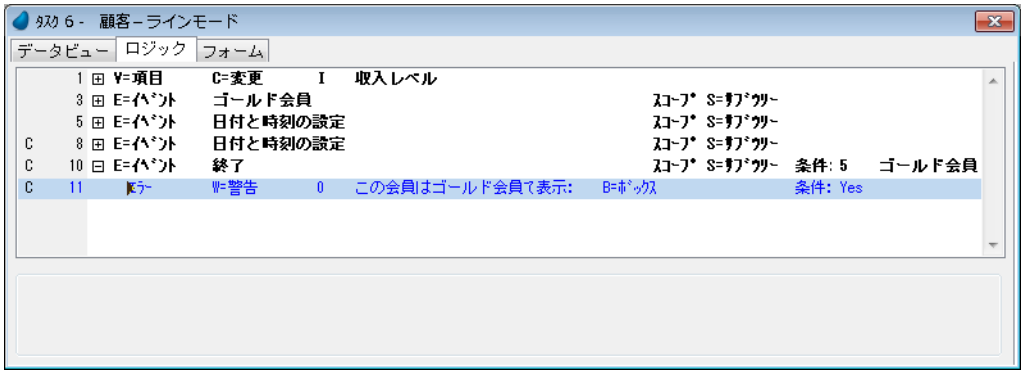
11. ズームして［式］エディタを開きます。
12. 最後の行にパークします。
13. 一行作成します。
14. 式として、＜ F AND I<=4000000 ＞と入力します。
    ここで、 F : ゴールド会員、 I : 収入レベル
15. ［OK］をクリックして式の設定を終了します。





16. 今回作成した［ロジック］エディタの［イベント］ロジックユニットヘッダ行にパークします。
17. 一行作成します。
18. 次のようにパラメータを設定します。

タイプ	モード	テキスト	表示	条件
E= エラー	W= 警告	この会員はゴールド会員ではなくなります。	B= ボックス	Yes



19. 変更内容を保存し、プログラムを閉じてください。
これで、［終了］イベントに追加のハンドラを作成したことになります。

例題２の確認

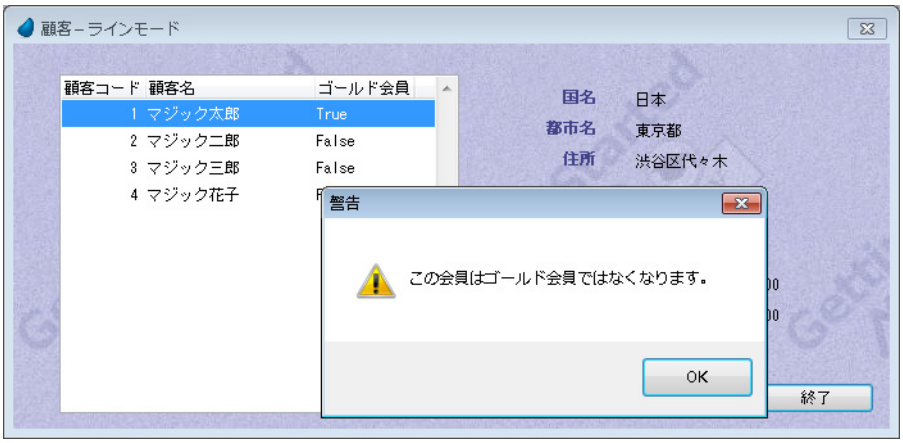
1. ＜顧客－ラインモード＞プログラムを実行します。
2. 最初の顧客にパークします。
3. ＜ゴールド会員＞欄が［True］になっていることを確認してください。

4. ＜収入レベル＞欄に＜ 3000000 ＞と入力してください。
5. ［終了］ボタンをクリックしてみましょう。

＜この会員はゴールド会員ではなくなります。＞という［警告］ダイアログが表示されます。

6. ［OK］をクリックしてみましょう。

［OK］ボタンをクリックすると［警告］ダイアログの表示は消えますが、プログラムは終了しません。プログラムを終了させるためには、＜ゴールド会員＞欄を［False］にするか、あるいは＜収入レベル＞を '4000000' より大きな値にしてから［終了］ボタンを押下することが必要です。



例題２の実行結果について

イベントが発動されたとき、Magic エンジンは対応して定義されている実行可能なハンドラを探し、見つかればそれを実行します。

ハンドラの実行が終了したとき、そのハンドラの［伝播］特性が［Yes］に設定されているときのみ、Magic エンジンは他にハンドラが定義されていないか探します。（複数のハンドラが一つのイベントに対して定義されているときは、実行ツリーで最下位にあるハンドラが最初に実行されます。）［伝播］特性が［No］のときは、それ以上ハンドラを探すことはせず、イベント処理の終了となります。

今回の例では内部イベントである［終了］に対してハンドラを追加しました。イベントが発動されたとき、すなわち［終了］ボタンがクリックされたとき、今回作成されたハンドラがまず実行されました。これはこのイベントハンドラの実行条件である＜ゴールド会員＞が［True］かつ＜収入レベル＞が＜ 4000000 ＞以下が成立していたからです。このとき、＜この会員はゴールド会員ではなくなります。＞という警告メッセージが表示されました。また、このハンドラの［伝播］特性はデフォルトの［No］のままだったので、このハンドラが終了すると Magic エンジンはそれ以上ハンドラを探すのをやめてしまい、Magic エンジン内部の［終了］ハンドラは実行されず、その結果、タスクも終了しないままになります。

次に、＜ゴールド会員＞欄を［False］にするか、あるいは＜収入レベル＞を＜ 4000000 ＞より大きな値にしてから［終了］ボタンをクリックすると、今度は今回作成したハンドラの実行条件が成立しないため、Magic エンジンは次のハンドラを探し、結果として内部の［終了］ハンドラが実行され、タスクは終了することになります。

７．要約

イベントドリブンによるプログラミングは、タスク内に非手続き的処理を組込むことを可能にします。イベントハンドラはその実行部を定義するものです。

イベントハンドラはタスクの［ロジック］エディタのなかで［イベント］ロジックユニットとして定義され、イベントが発動されたときにそのコマンドが実行されます。

一つのイベントに対して複数のハンドラを定義することができ、［伝播］特性を使用する使用することで、どのハンドラを実行させるかを制御することができます。

また Magic エンジンに組込まれている内部イベントに対しても、ハンドラを定義することができます。

リッチクライアントタスクにおいては、タスクがアイドル状態のときか、エンドユーザがイベントを発動したとき、Magic エンジンはイベント実行を監視し制御します。［イベント実行］処理コマンドを使用するときには、開発者は［ウェイト］特性を［Yes］とするか［No］とするかにより、イベントの発動を［同期］させたり［非同期］にすることができます。

第 12 章 ［ブロック］処理コマンド

本章では、条件にしたがって、複数のコマンドグループの実行を制御する、［ブロック］処理コマンドについて学びます。

キーワード

- ブロック If
- ブロック Else
- ブロック While
- 実行条件

学習目標

本章を学ぶことで、次の内容に対する理解を深めましょう。

- ［ブロック IF］処理コマンドと［ブロック Else］処理コマンドの使い方
- 条件に対応する複数コマンドのグループ化
- ［ブロック While］処理コマンドの使い方

参照

Magic ヘルプの［Magic xpa リファレンス／［ロジック］エディタ／処理コマンド／ブロック］

１．はじめに

プログラムの一部は条件的に実行を制御することが必要になる場合があります。多くの場合、それぞれ異なる一まとまりの内容を、条件によって選択的に実行します。

そのような場合、［ブロック］処理コマンドにより、構造的でかつ効率的なプログラムを作成することができます。

［ブロック］処理コマンドを使用する利点は二つあります。

- 複数コマンドの条件的実行
  複数の処理コマンドのまとまりを、ある一つの実行条件でくくることができます。
  この場合は三つの処理コマンドを利用します。
  - ブロック If
  - ブロック Else
  - ブロック End
- 複数コマンドの反復（ループ）実行
  複数の処理コマンドのまとまりを、ある一つの実行条件が成立している間、繰り返し実行することができます。
  この場合は二つの処理コマンドを利用します。
  - ブロック While
  - ブロック End

本章ではこれらの処理コマンドをプロジェクトの中で実際に使用してみることにより、理解を深めます。

２．［ブロック］処理コマンドの構造

［ブロック］処理コマンドのいろいろな使い方を最初にまとめておきます。

例１．一つの実行条件で、コマンドグループの実効制御を行う場合

ブロック If　実行条件
　処理コマンド
　・・・
　処理コマンド
ブロック End

Magic エンジンが［ブロック If］処理コマンドを実行するときには、まずその実行条件を評価します。

- 結果が 'True' の場合、If ブロックの処理コマンドが実行されます。
- 結果が 'False' の場合、If ブロックの処理コマンドを実行せず、スキップします。

例２．実行条件が成立するときは If ブロック、成立しない場合は Else ブロックの処理コマンドを実行させる場合

ブロック If　実行条件
　処理コマンド
　・・・
　処理コマンド
ブロック Else
　処理コマンド
　・・・
　処理コマンド
ブロック End

例３．例２と似ているが、ブロック Else にも条件を追加して、さらに Else ブロックを追加する場合

ブロック If　実行条件１

処理コマンド

・・・

処理コマンド

ブロック Else　実行条件２

処理コマンド

・・・

処理コマンド

ブロック Else

処理コマンド

・・・

処理コマンド

ブロック End

［ブロック Else］処理コマンドが定義されているとき、その実行条件を評価します。

- 結果が 'True' の場合、Else ブロックの処理コマンドが実行されます。
- 結果が 'False' の場合、Else ブロックの処理コマンドを実行せずスキップし、次の［ブロック Else］処理コマンドがあれば同じことを繰り返します。次の［ブロック Else］処理コマンドがない場合は、［ブロック End］処理コマンドの次の処理コマンドに移動し処理が続けられます。

［ブロック Else］処理コマンドを使用する回数には制限はありません。

例４．実行条件が成立しているときはコマンドブロックを繰り返し処理する場合

ブロック While　実行条件

処理コマンド

・・・

処理コマンド

ブロック End

［ブロック While］処理コマンドは実行条件が 'True' と評価されている間は、ブロック内の処理コマンドを繰り返し処理します。評価が 'False' となった場合、［ブロック End］処理コマンドの次の処理コマンドに移動し処理が続けられます。

３．［ブロック If］処理コマンドの実習

＜顧客－ラインモード＞プログラムで＜国名＞に値がすでに入っていて、その値が変更されたとき、＜都市名＞と＜住所＞の値をクリアするプログラムを作ってみましょう。

［項目変更］ロジックユニットの定義

1. ＜顧客－ラインモード＞プログラムにズームします。
2. ［ロジック］エディタを選択します。
3. 最後の行にパークします。

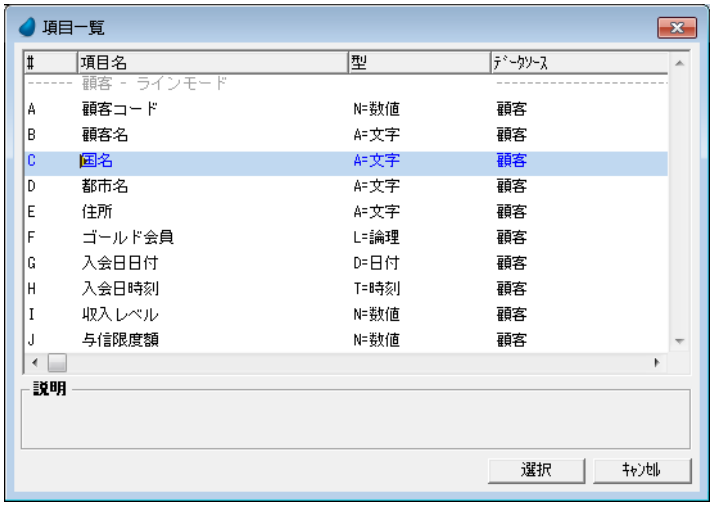
4. ヘッダ行（Ctrl+H）を作成します。
5. 次のようにパラメータ設定します。

タイプ	レベル	項目	条件
V= 項目	C= 変更	国名	Yes

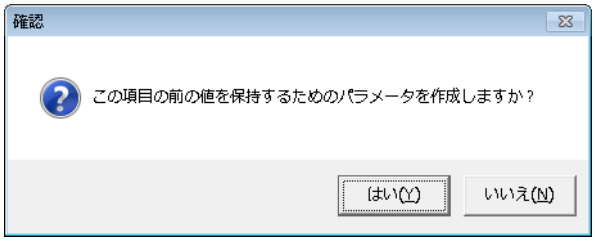
項目の選択は次のようにします。

6. 項目一覧にズームします。

7. ＜国名＞にパークします。
8. ［選択］ボタンをクリックします。



項目の前回値を保持するパラメータ作成の［確認］ダイアログが表示されます。



9. ［はい］をクリックします。

［ブロック］処理コマンドの定義

1. ［項目変更］ロジックユニットの最後の行にパークし、一行作成します。
2. 次のようにパラメータを設定します。

コマンド	項目	条件
ブロック	I=If	6　{CHG_PRV_ 国名 <>"
ブロック	N=End	}

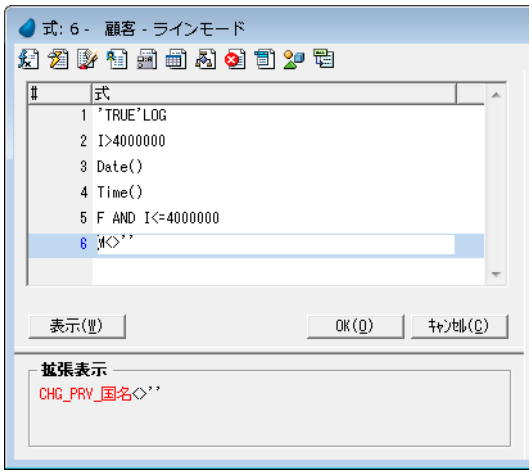
条件は次のように設定します。

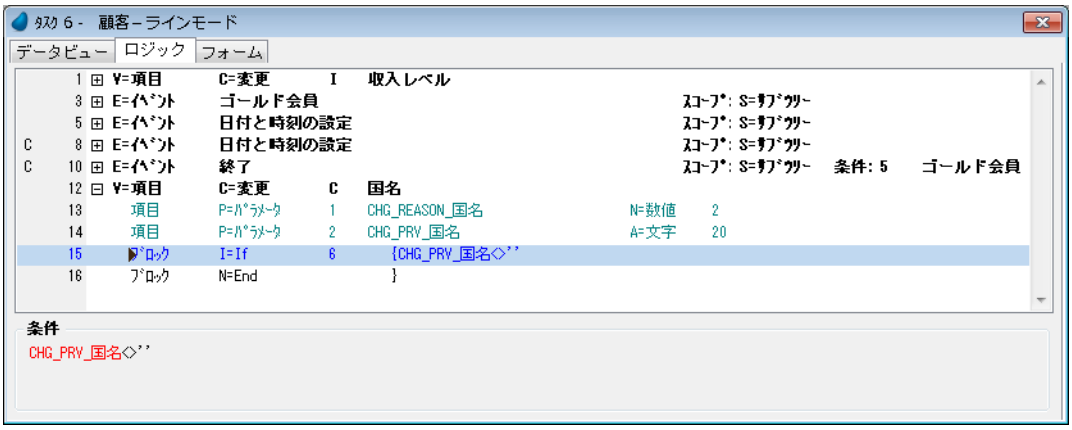
3. ［式］欄からズームして［式］エディタを開きます。
4. 最後の行にパークします。
5. 一行作成（F4）します。
6. ズームして項目リストに移動します。
7. ＜ CHG_PRV_ 国名＞項目を選択します。

＜ CHG_PRV_ 国名＞が表示されていない場合、［全て表示］ボタンをクリックすることですべての項目が表示されるようになります。

8. 式を次のように入力します：＜ M<>’’ ＞　（M は＜ CHG_PRV_ 国名＞を示すとします。）

9. ［OK］ボタンをクリックします。





［ブロック End］処理コマンドは、［ブロック If］処理コマンドを定義すると自動的に作成されます。

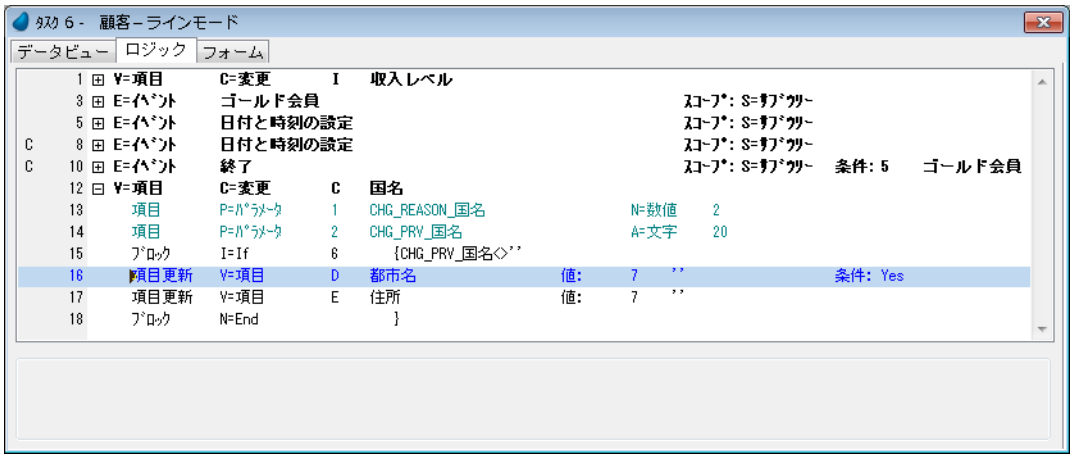
＜都市名＞と＜住所＞のクリア

1. ［ブロック If］処理コマンド行にパークし、二行作成します。
2. 次のようにパラメータを設定します。

コマンド	項目	値	条件
項目更新	D （都市名）	7 （"）	Yes
項目更新	E （住所）	7 （"）	Yes

各項目は［項目一覧］にズームして選択してください。

値の設定は、［式］エディタにズームして一行作成し、式を定義（この場合、＜’’＞）して選択してください。



3. 変更内容を保存し、プログラムを閉じます。

プログラムの実行

1. ＜顧客－ラインモード＞プログラムを実行（F7）します。
2. 最初の顧客にパークします。
3. ＜国名＞を変更してみましょう：**＜米国＞**

4. 次のコントロールに移動します。

このとき、＜都市名＞と＜住所＞のデータがクリアされることが分かります。

5. Esc キーを押下して終了しましょう。

［ブロック］処理コマンドを使用する利点

［ブロック］処理コマンドは、一連のコマンドをグループ化し、一つの論理ブロックとすることができるため、ブロック内のコマンドは一つの条件（［ブロック］処理コマンドの条件）にのみ依存します。

このような［ブロック］処理コマンドを使用する利点を列挙します。

- 開発者の効率向上

  処理コマンドごとに条件を設定しなくてもよいため、効率の良い開発作業になります。

  また保守を考えた場合も一つの条件のみを修正すればよいため、楽になります。

  つまりブロック実行条件だけに着目すればよいのです。

  先ほどの例では、＜都市名＞と＜住所＞の更新に一つの条件を使用しました。

- Magic エンジンの効率向上

  条件判定の回数が減らせるため、実行効率がよくなります。

  ［ブロック］処理コマンドを使用しているロジックでは、コマンド実行の都度条件を確認する必要がなく、一度だけで済むからです。

- 記述の単純化（理解のしやすさ向上）

  ［ブロック］処理コマンドは、条件の記述を単純化することができるという利点もあります。複雑な条件式になるような場合、その条件をいくつかの論理に分割し、ブロックごとに割当てることで、一つ一つの条件が単純化されます。

  たとえば、'国名＝イスラエル　AND　都市名＝エルサレム' というような条件がある場合、次のように記述することもできます。

  ブロック　If　（条件：　国名＝イスラエル）
  　　ブロック　If　（条件：　都市名＝エルサレム）
  　　（実行コマンド）
  　　ブロック　End
  ブロック　End

  ［ブロック］処理コマンドはネストさせることができます。コマンドをある条件下で実行させたい場合で、さらに他のコマンドをもう一つ別の条件下で実行させたい場合、［ブロック］処理コマンドをネストすることにより実現します。この場合、あとのコマンドについては、両方の条件が成立した場合にのみ実行されることになります。

  上記の例では、（実行コマンド）は、国名 ='イスラエル' でかつ都市名 ='エルサレム' の場合にのみ実行されます。

４．［ブロック Else］処理 コマンドの実習

［ブロック Else］処理コマンドは、［ブロック If］処理コマンドの拡張ということができ、いくつかの条件を同一のブロック内に設定することができます。

［ブロック If］処理コマンドと対になる［ブロック End］処理コマンドで囲まれる論理ブロックの中に、いくつでも［ブロック Else］処理コマンドを定義することができます。

［ブロック If］処理コマンドの条件が［False］と評価されたときには、Magic エンジンは［ブロック Else］処理コマンドの行に制御を移します。そしてもし［ブロック Else］の条件が［True］と評価された場合には、［ブロック Else］内（次の［ブロック Else］行または、［ブロック End］行まで）の処理コマンドが実行されます。［ブロック Else］の条件が［False］の場合は、Magic エンジンは次の［ブロック Else］（または［ブロック End］）行に制御を移します。以下、［ブロック End］に到達するまで同様の処理が行われます。

例題

このセクションでは、［ブロック Else］処理コマンドを使用して、＜与信限度額＞を更新する次のようなロジックを作ってみましょう。

- ＜ゴールド会員＞となっている顧客は、＜与信限度額＞を＜収入レベル＞の 3 倍にします。
- ＜ゴールド会員＞以外の顧客は、＜収入レベル＞の 2 倍にします。

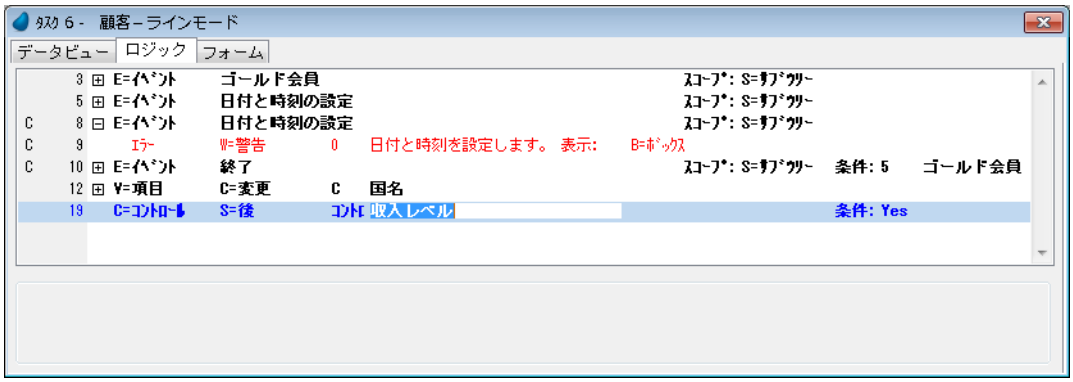
［コントロール後］ロジックユニットの定義

＜与信限度額＞は、＜収入レベル＞のコントロールからカーソルが他に移動するときに更新するようにします。この場合、この処理は［＜収入レベル＞コントロール後］ロジックユニットに定義し、その中で［ブロック］処理コマンドを使用するようにします。

1. ＜顧客－ラインモード＞プログラムにズーム（F5）します。
2. ［ロジック］エディタを選択します。
3. 最後の行にパークします。

4. ヘッダ行（Ctrl+H）を一行作成します。
5. ヘッダ行のパラメータを次のようにします。

タイプ	レベル	コントロール	条件
C= コントロール	S= 後	収入レベル	Yes



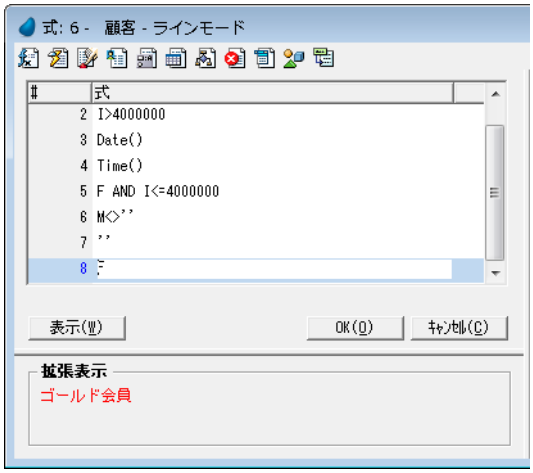
［ブロック］処理コマンドの定義

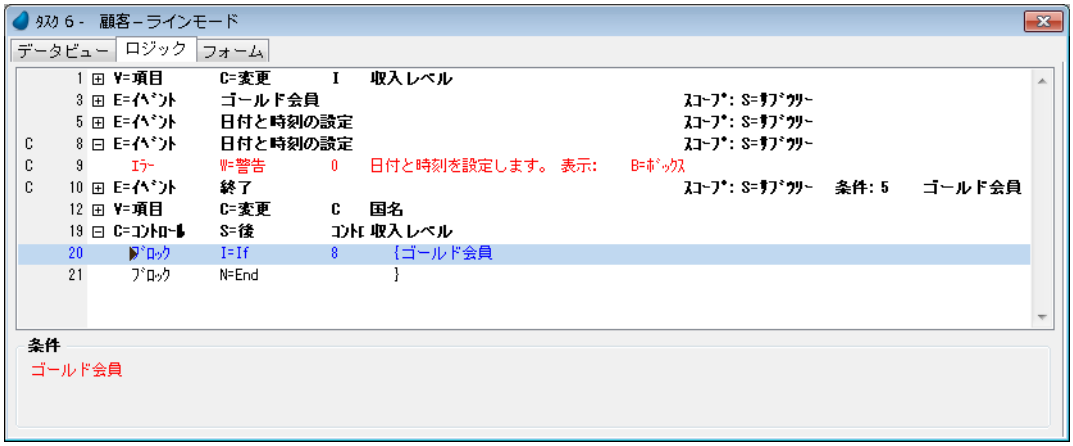
1. ［コントロール後］ロジックユニットのヘッダ行にパークし、一行作成します。
2. パラメータを次のように設定します。

コマンド	条件
ブロック If	8　（ゴールド会員）

条件の設定は次のように行います。

3. ［式］エディタにズームします。
4. 最後の行にパークします。
5. 一行作成（F4）します。
6. 項目リストに移動し、＜ゴールド会員＞を選択します。（または、項目記号＜ F ＞を式として入力します。）
7. ［OK］ボタンをクリックします。





ゴールド会員の場合の与信限度額更新

1. ［コントロール後］ロジックユニットのヘッダ行にパークし、一行作成します。
2. パラメータを次のように設定します。

コマンド	条件
ブロック If	8　（ゴールド会員）

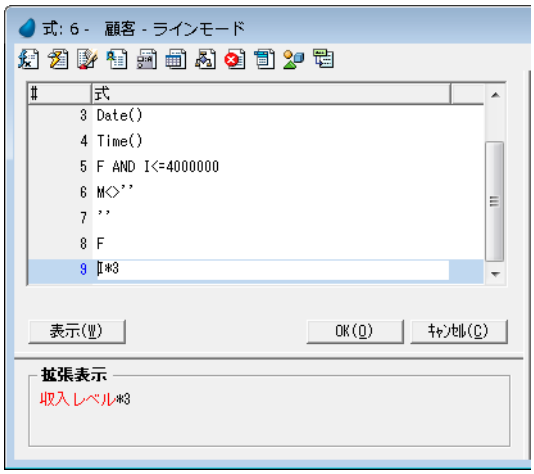
3. ［ブロック If］処理コマンド行にパークし、一行作成（F4）します。

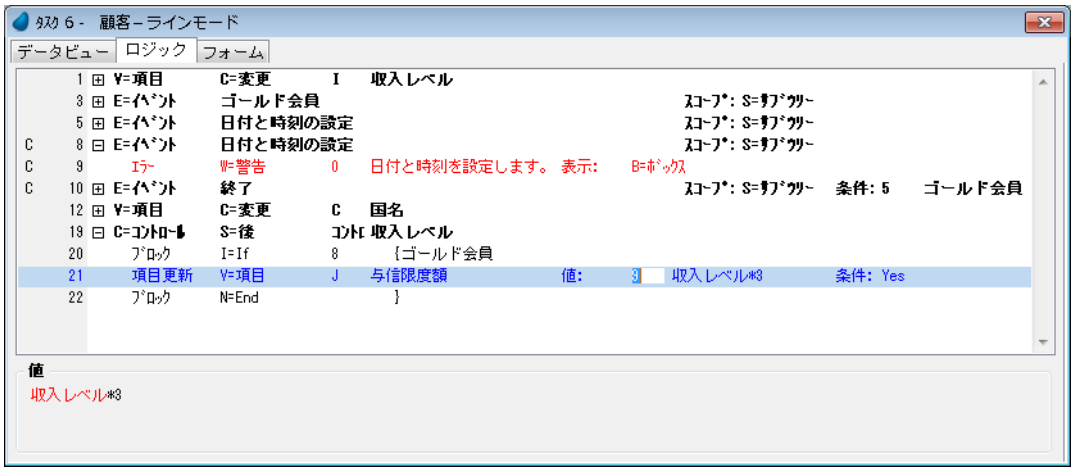
4. 次のようにパラメータを設定します。

コマンド	項目	値	条件
項目更新	J （与信限度額）	9 （収入レベル *3）	Yes

項目は、項目一覧にズームして選択するようにします。

値は［式］エディタにズームし、最終行の次に一行作成し、＜ I*3 ＞（I は収入レベルの項目記号）と入力します。

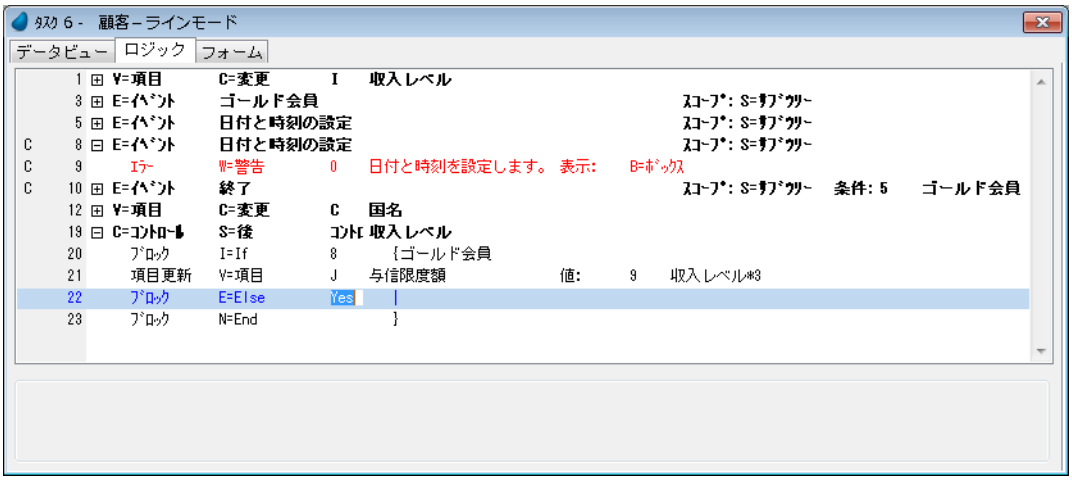




［ブロック Else］処理コマンドの定義

1. ［項目更新］処理コマンドの行にパークし、一行作成します。
2. 次のパラメータを設定します。

コマンド	条件
ブロック Else	Yes



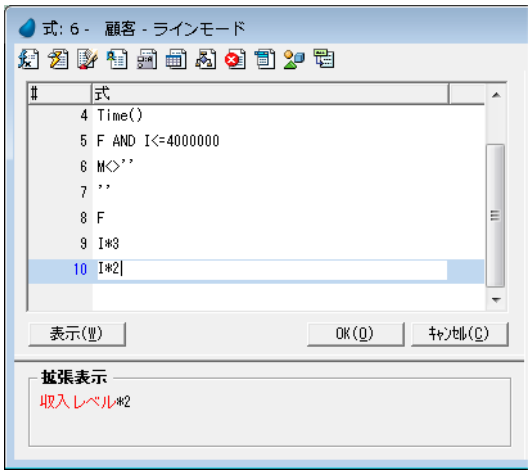
ゴールド会員ではない場合の与信限度額更新

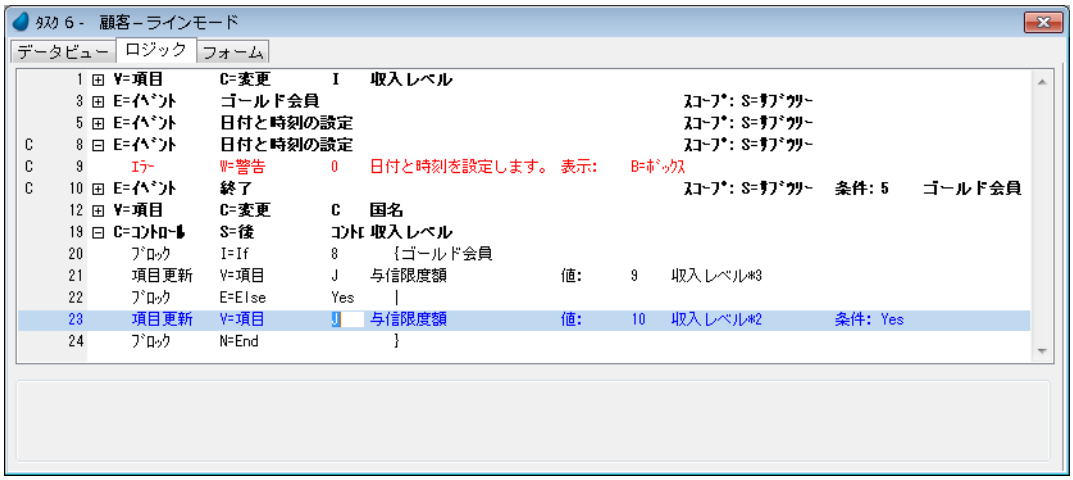
1. ［ブロック Else］処理コマンド行にパークし、一行作成します。
2. パラメータを次のように設定します。

コマンド	項目	値	条件
項目更新	J　（与信限度額）	10　（収入レベル *2）	Yes

項目は、項目一覧にズームして選択するようにします。

値は［式］エディタにズームし、最終行の次に一行作成し＜ I*2 ＞（I は収入レベルの項目記号）と入力します。



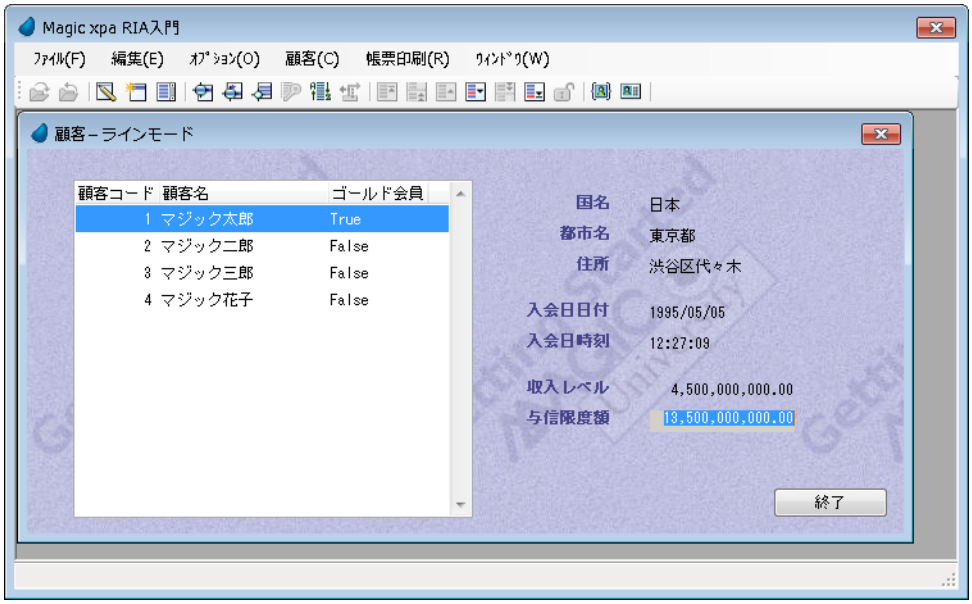


3. 変更内容を保存し、プログラムを閉じます。

プログラムの実行

1. F7 キーを押下して＜顧客－ラインモード＞プログラムを起動します。
2. 三番目の顧客にパークしてください。
3. ＜ゴールド会員＞欄を［True］にしてください。

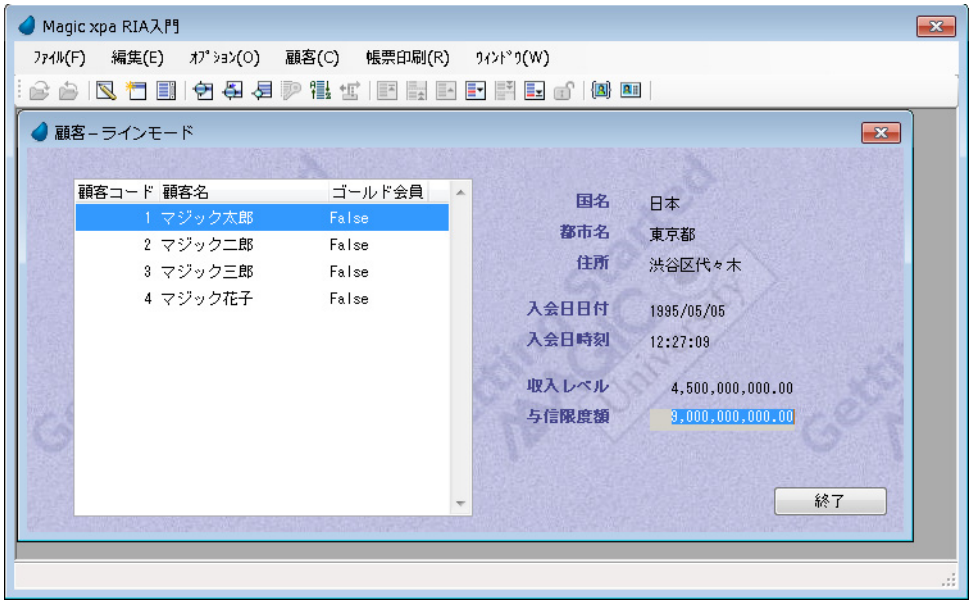
4. ＜収入レベル＞に Tab キーを使用して移動します。
5. ＜収入レベル＞として、＜ 4500000 ＞と入力してください。
6. Tab キーで次のコントロールに移動してください。



＜収入レベル＞から＜与信限度額＞に移動したときに、その値が＜収入レベルの3倍の値、すなわち13,500,000（=4,500,000*3）になっていることが分かります。この顧客がゴールド会員だからです。

7. 続いて、二番目の顧客にパークしてください。
8. ＜ゴールド会員＞欄が［False］になっていることを確認してください。（［True］になっている場合、［False］としてください。）
9. ＜収入レベル＞にパークします。
10. ＜収入レベル＞の値に、＜ 3500000 ＞と入力してみましょう。

11. 次のコントロールに移動してください。



こんどは＜与信限度額＞が＜収入レベル＞の 2 倍の値になっていることが分かります。この顧客がゴールド会員ではないからです。

12. Esc キーを押下してプログラムを終了してください。

５．［ブロック While］処理コマンド

［ブロック While］処理コマンドは、ブロック実行条件の評価結果が 'True' の間、ブロック内の処理コマンドを反復して実行するコマンドです。

Magic エンジンは［ブロック While］処理コマンドの実行条件を評価し、［True］の場合、ブロック内のコマンドを一つ一つ順に［ブロック End］処理コマンド行まで処理を行います。その後再び条件を評価し、評価結果が［False］となるまで、ブロック内コマンドの実行を繰返します。［False］が検出された場合は、ブロック内コマンドをスキップし、［ブロック End］処理コマンド行の次のコマンドから処理を続行します。

６．練習問題

第 8 章で、＜取引先＞データソースを定義し、＜取引先－ラインモード＞プログラムを作成しました。ここではそのプログラムを修正し、取引先ごとにボーナスを計算する機能を追加してみましょう。

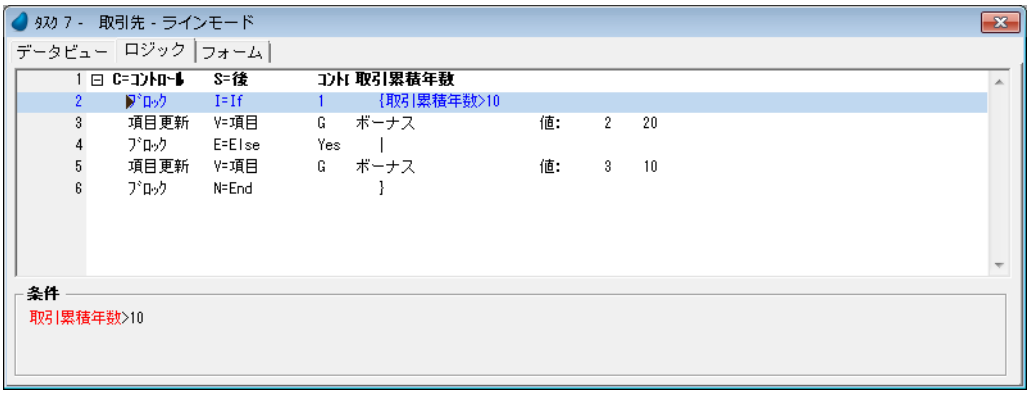
取引先のボーナスは会社との取引関係が始まってからの継続年数によって計算されるものとし、具体的には次の規則を適用します。

- 10 年以上の取引実績がある場合、取引額の 20% をボーナスとします。
- 10 年未満の場合は 10% とします。

次のようにして、このロジックを組込んでみましょう。

1. ＜取引先－ラインモード＞プログラムにズームします。
2. ［ロジック］エディタで＜取引累積年数＞コントロールに［コントロール後］ロジックユニットを定義してください。
3. ロジックユニットで［ブロック If］処理コマンドを定義し、実行条件に＜取引累積年数＞=10 ＞を設定します。
4. ［ブロック If］のブロック内に、＜ボーナス＞の値を＜ 20 ＞とする［項目更新］処理コマンドを定義します。
5. ［項目更新］処理コマンドの次に、［ブロック Else］処理コマンドを追加します。
6. もう一つ［項目更新］処理コマンドを定義し、こちらは＜ボーナス＞の値を＜ 10 ＞にします。
7. 変更内容を保存し、プログラムを閉じます。

8. プログラムを実行し、結果を確認しましょう。



7．要約

本章では、［ブロック］処理コマンドについて学習しました。

［ブロック］処理コマンドは式の定義を単純化し、同じ式をあちこちに定義する手間を省いて開発効率を高めるだけでなく、プログラムの保守性を高め、Magic エンジンの実行速度を上げ、また条件定義のネスト構造を可能とすることで、見やすいプログラムを実現します。

［ブロック］処理コマンドを使用する場合としては、主に二つあります。

1. コマンドグループを条件に基づいて実行管理する
2. 条件が成立している間、コマンドグループを繰返し実行する

#1 の場合には次の処理コマンドを利用します。

- ブロック If
- ブロック Else
- ブロック End

#2 の場合には次の処理コマンドを利用します。

- ブロック While
- ブロック End

第 13 章 リポジトリの操作

本章ではリポジトリエントリの操作、すなわちリポジトリの各エントリについて複写、移動、上書き、検索などを行う方法について説明します。

学習目標

本章を学ぶことで、次の内容に対する理解を深めましょう。

- リポジトリ内の特定の行の検索
- リポジトリ内の特定の行へジャンプする
- リポジトリのエントリを他の場所に移動する
- リポジトリのエントリを複写する
- 既存エントリを他のエントリで上書きする
- リポジトリ内の複数のエントリへの操作を実行する

参照

Magic ヘルプの［Magic xpa リファレンス／ナビゲータとワークスペース／リポジトリの操作］

１．はじめに

Magic xpa には次のようなリポジトリがあります。（製品の用語としては、必ずしも［リポジトリ］という用語を使用していない場合もあります。）

- モデル
- データ
  - カラム
  - インデックス
  - 外部キー
- プログラム
  - データビュー
  - ロジック
  - フォーム
  - 式
  - イベント
- ヘルプ
- 権利
- メニュー
- コンポーネント

Magic xpa にはリポジトリを操作するいろいろな機能があります。これらの機能を利用してリポジトリ内の検索、複写、移動、上書き（置き換え）などを行うことができます。

さらに次のような機能もあります。

- 位置付
- 行ジャンプ
- フォルダ指定

これらの機能は、［編集］メニューから［クイックアクセス］にあるサブメニューを選択することで使用することができます。

２．クイックアクセス

クイックアクセスメニューはエントリを見つけやすくする次の機能があります。

- 位置付
- 行ジャンプ
- フォルダ指定

このセクションでは、［位置付］と［行ジャンプ］の使い方について実習を通して学びます。

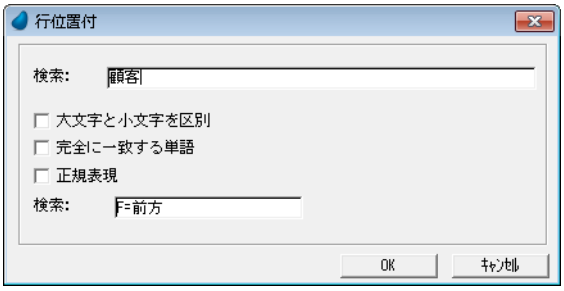
位置付

位置付機能は、入力した条件に一致する最初の行を見つける機能です。

ここでは［プログラム］リポジトリでの例を見ることにしましょう。

1. ［プログラム］リポジトリ（Shift+F3）を開きます。
2. 最初のプログラム（メインプログラム）の行にパークします。
3. ［編集］メニューから［クイックアクセス／位置付］（Ctrl+L）を選択します。

［行位置付］ダイアログが開きます。



4. ［検索］欄に**＜顧客＞**と入力します。
5. ［OK］ボタンをクリックします。

これでプログラム名の中に＜顧客＞という文字を含むものを検索し、見つかった場合その行に位置付けることになります。

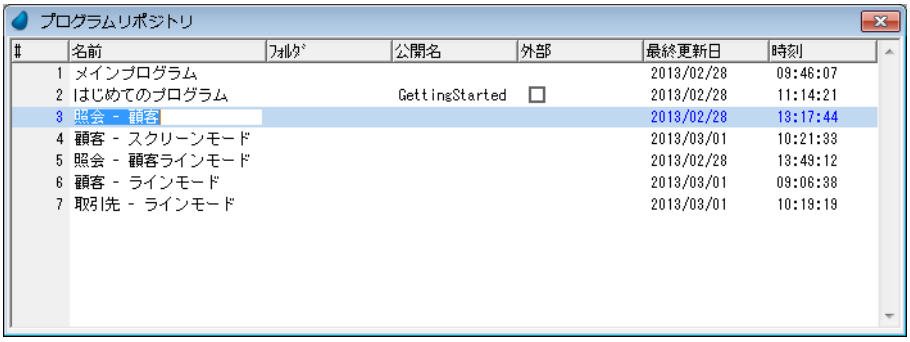
［行位置付］ダイアログについて

以下のような検索条件を設定することができます。

設定項目	説明
検索	検索する文字列を入力する。
大文字と小文字を区別	英文字の小文字と大文字を区別するかどうかの指定を行う。
完全に一致する単語	すべての文字が一致するものを検索するかどうかを指定する。
正規表現	正規表現を使用するかどうかを指定する。（正規表現については Magic ヘルプを参照、たとえば、＊は任意の文字列、$ または？は任意の一文字を表します。）
検索	検索する方向を指定する。 • F= 前方 …… 現在カーソルのある行から前方に検索を行う。 • B= 後方 …… 現在カーソルのある行から後方に検索を行う。 • T= フォーム先頭 …… カーソルの現在位置に関係なく検索をリポジトリの先頭行から始める。

位置付けが実行された結果、カーソルは #3 の＜照会－顧客＞プログラムの行にパークします。

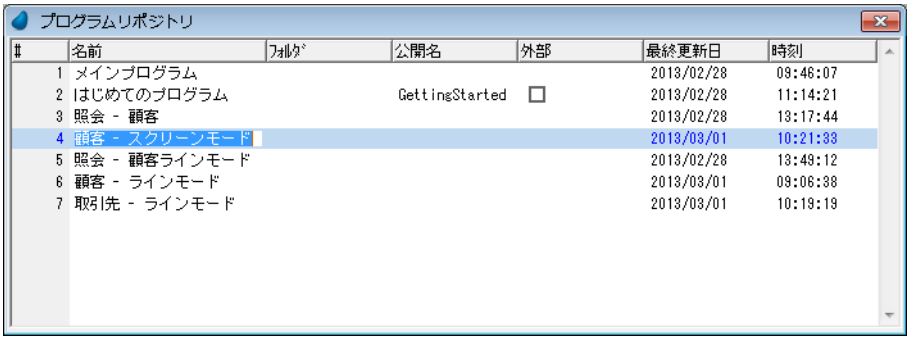
このプログラムが検索条件である＜顧客＞という文字を含む最初のプログラムだからです。



位置付／次候補

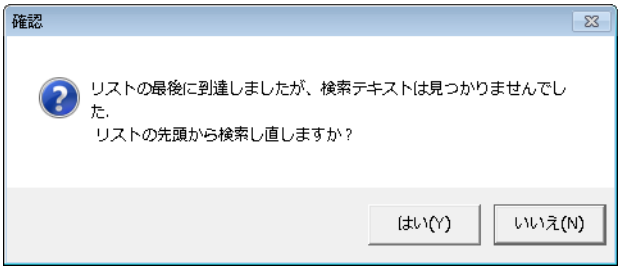
［位置付］の実行後、［位置付 / 次候補］を選択すると、同じ条件で次の候補を検索します。

1. ［編集］メニューから、［クイックアクセス／位置付 / 次候補］（Ctrl+N）を選択します。



カーソルが次に＜顧客＞という文字を含むプログラム、この場合 #4 の＜顧客ースクリーンモード＞にパークします。

2. もう一度［編集］メニューから、［クイックアクセス／位置付 / 次候補］（Ctrl+N）を選択します。カーソルは #5 のプログラム行にパークします。
3. さらにもう一度［編集］メニューから、［クイックアクセス／位置付 / 次候補］（Ctrl+N）を選択します。カーソルは #6 のプログラム行にパークします。
4. 続けてもう一度［編集］メニューから、［クイックアクセス／位置付 / 次候補］（Ctrl+N）を選択します。今度はもうこれ以上＜顧客＞という文字をプログラム名に含むプログラムがなく、リポジトリの最終行に到達したため、もう一度リポジトリの先頭から検索を行うかどうかの［確認］ダイアログが表示されます。



- ［はい］をクリックすると、カーソルが先頭に戻り、検索を続行します。
- ［いいえ］をクリックすると、検索を終了します。

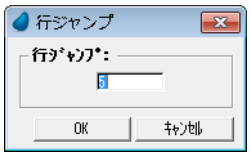
ここでは、［いいえ］をクリックしましょう。

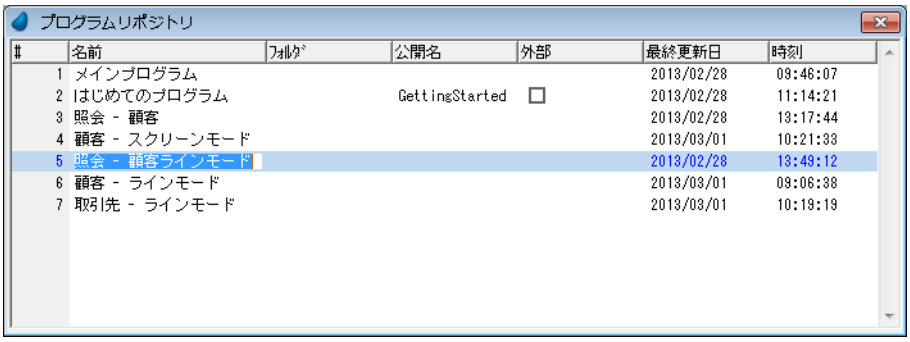
行ジャンプ

Magic xpa はリポジトリのエントリに対し識別するための行番号をつけています。行ジャンプ機能は、指定した行番号までジャンプする機能です。

このセクションでは行ジャンプ機能を使用して、［プログラム］リポジトリの指定行までジャンプしてみます。

1. ［プログラム］リポジトリ（Shift+F3）を開きます。
2. 最初のメインプログラムにパークします。
3. ［編集］メニューから［クイックアクセス／行ジャンプ］（Ctrl+J）を選択します。［行ジャンプ］ダイアログが表示されます。
4. ［行ジャンプ］欄に＜ 5 ＞と入力します。
5. ［OK］ボタンをクリックします。





カーソルが #5 のプログラムまでジャンプして移動しました。

３．リポジトリのエントリ登録操作

［編集］メニューから［登録］にはリポジトリのエントリを操作する次の三つの機能があります。

- 複写登録
- 移動登録
- 上書登録

具体的にどのようなことができるのか順に見てゆきましょう。

複写登録

［複写登録］は、一つ以上のエントリをコピーし、リポジトリ内の別の場所に貼り付ける機能です。

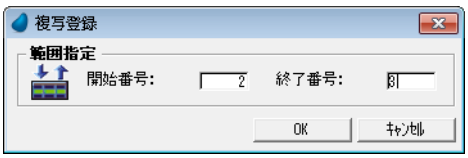
Magic xpa は貼り付けた後のエントリ番号を自動的に振り直します。

エントリは次のような手順でコピーし貼り付けます。

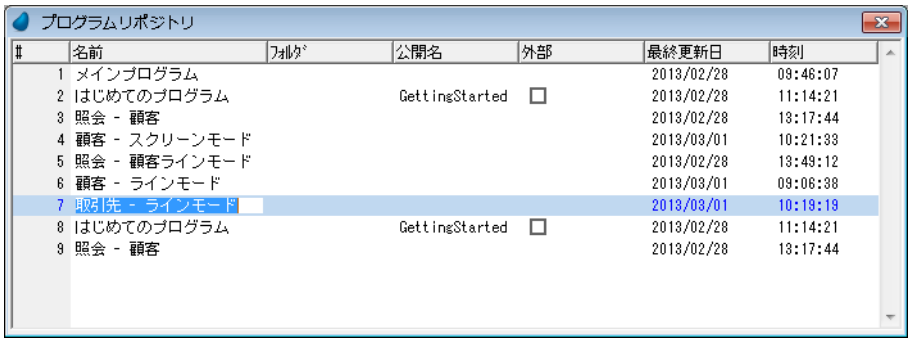
- 貼り付けたい位置の上のエントリ行にパークします。
- ［編集］メニューから［登録／複写登録］を選択します。
- コピーするエントリを範囲指定します。
- パークしている行の下の位置に、指定したエントリがコピーされ、挿入貼り付けされます。

次の実習例では、［プログラム］リポジトリのエントリ #2 と #3 を最後の位置に［複写登録］してみましょう。

1. ［プログラム］リポジトリを開きます。
2. 最後のエントリ行（#7）にパークします。
3. ［編集］メニューから［登録／複写登録］（Ctrl+Shift+R）を選択します。
4. ［複写登録］ダイアログが開くので、［開始番号］欄からズームしてプログラム一覧を開き、＜プログラム #2 ＞を選択します。（または＜ 2 ＞を入力します。）
5. 同じようにして、［終了番号］欄に＜プログラム #3 ＞を選択します。（または＜ 3 ＞を入力します。）
6. ［OK］ボタンをクリックします。



プログラム #2 と #3 が、プログラム #7 の後に複写登録されました。

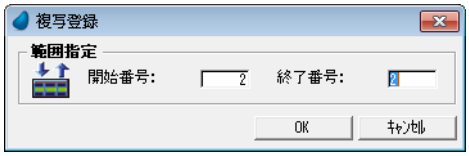


この実習例では、プログラム #2 と #3 を最後の位置に複写登録しましたが、実際には任意の位置に複写登録することができます。既存のエントリの間に複写登録するような場合には、エントリ番号は自動的に振り直されます。

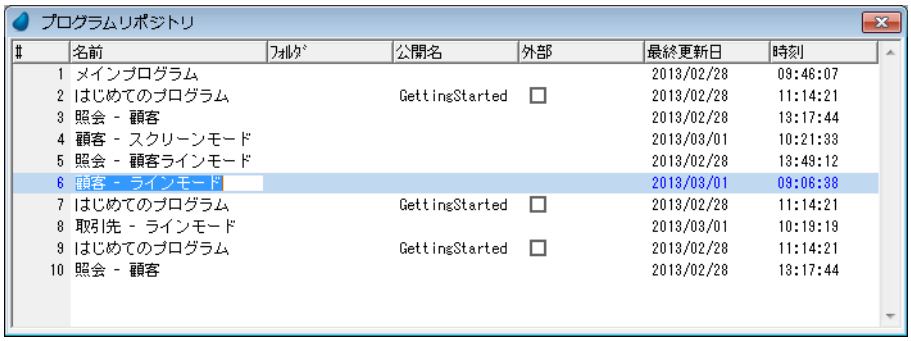
プロジェクト内に同じ公開名を定義することはできません。上記の操作で「GettingStarted」が重複した状態になるため、通常はどちらかの公開名を変更する必要があります。

今度はプログラム #2 をプログラム #6 の後に複写登録してみましょう。

1. ［プログラム］リポジトリを開きます。
2. プログラム # 6 の行にパークします。
3. ［編集］メニューから［登録／複写登録］（Ctrl+Shift+R）を選択します。
4. ［複写登録］ダイアログで、［開始番号］欄からズームしてプログラム一覧を開き、＜プログラム #2 ＞を選択します（または＜ 2 ＞を入力します。）
5. 同じようにして、［終了番号］欄にも＜プログラム #2 ＞を選択します（または ＜ 2 ＞ を入力します。）
6. ［OK］ボタンをクリックします。



プログラム #2 がプログラム #6 の後ろに複写登録されてプログラム #7 となり、これ以降のプログラムのエントリ番号は自動的にそれぞれ +1 されていることが分かります。



４．移動登録

［移動登録］は、リポジトリ内でのエントリの登録位置を変更する機能です。エントリを移動したとき、エントリ番号は振り直されますが、内部で参照しているリンク関係は維持されます。

［移動登録］は次のような手順で行います。

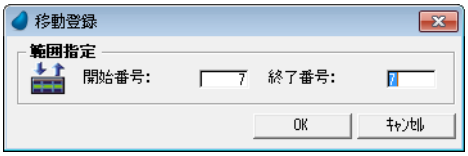
- 移動したい位置の上のエントリ行にパークします。
- ［編集］メニューから［登録／移動登録］を選択します。

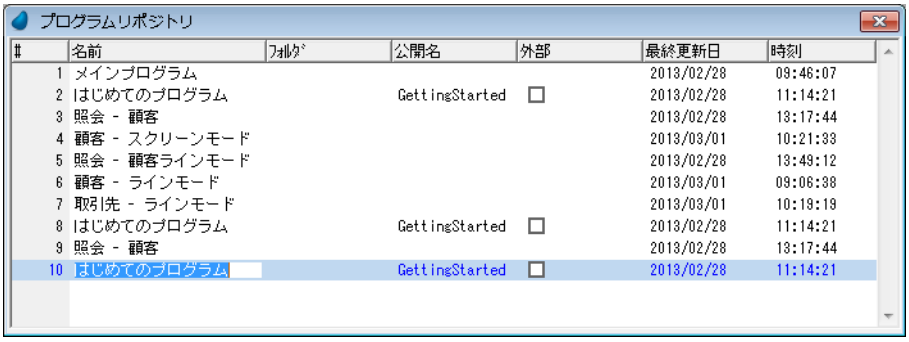
- 移動するエントリを範囲指定します。
- パークしている行の下の位置に、指定したエントリが移動（挿入）されます。
- エントリ番号が振り直されます。

ここでは、エントリ #7 にある＜はじめてのプログラム＞をリポジトリの最後に移動してみましょう。

1. ［プログラム］リポジトリを開きます。
2. プログラム # 10 の行にパークします。
3. ［編集］メニューから［登録／移動登録］（Ctrl+Shift+M）を選択します。

4. ［移動登録］ダイアログで、［開始番号］欄からズームしてプログラム一覧を開き、＜プログラム #7 ＞を選択します（または＜ 7 ＞を入力します）。
5. 同じようにして、［終了番号］欄にも＜プログラム #7 ＞を選択します（または ＜ 7 ＞ を入力します）。
6. ［OK］ボタンをクリックします。





プログラム #7 が最後の行の後ろに移動登録され、各エントリ行の番号が振り直されたことが分かります。

５．上書登録

［上書登録］は、リポジトリ内の既存エントリをコピー元のエントリで上書きする機能です。

この機能を使用した後、元のエントリ内容は失われ、コピーされたエントリ内容に置き換わります。コピー元のエントリの内部にあるリンク関連情報は維持されます。

［上書登録］は次のような手順で行います。

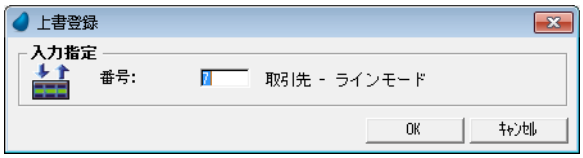
1. 上書きしたい位置のエントリ行にパークします。
2. ［編集］メニューから［登録／上書登録］を選択します。
3. 移動するエントリを範囲指定します。
4. パークしている行のエントリが、指定したエントリからコピーした内容によって上書きされます。

ここでは、プログラム #10 の＜はじめてのプログラム＞を 7 番の＜取引先－ラインモード＞で上書きしてみましょう。

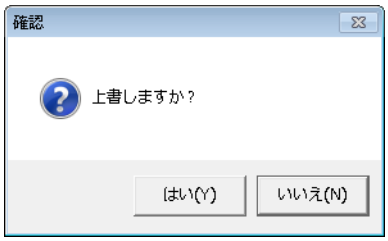
1. ［プログラム］リポジトリを開きます。
2. プログラム # 10 の行にパークします。
3. ［編集］メニューから［登録／上書登録］（Ctrl+Shift+O）を選択します。

4. ［上書登録］ダイアログで、［番号］欄からズームしてプログラム一覧を開き、＜プログラム #7 ＞を選択します（または ＜ 7 ＞を入力します）。

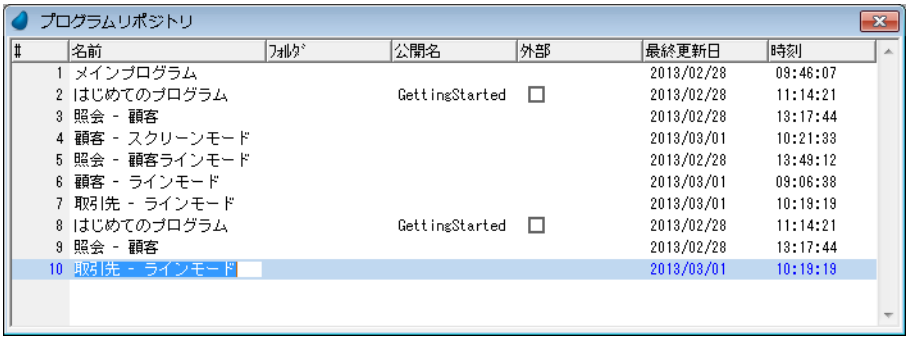
5. ［OK］ボタンをクリックします。



6. 上書きするかどうかの確認を求めるダイアログが表示されるので、［はい］をクリックします。



プログラム #10 ＜はじめてのプログラム＞が #7 の＜取引先－ラインモード＞プログラムによって上書きされました。



６．マルチマーキング

マルチマーキング機能を使用することで、一度に複数のエントリを選択できます。マルチマーキングによる複数エントリの選択は、一括削除や一括チェックなどを行うときにも利用できます。

マーク方法

次のような方法により複数エントリの選択を行います。

キーボードを利用した連続範囲選択

1. 最初のエントリ行にパークします。
2. Shift キーを押下します。
3. ↓キーを押下して所定の範囲を選択します。

キーボードとマウスを利用した連続範囲選択

1. 最初のエントリ行にパークします。
2. Shift キーを押下します。
3. 選択範囲最後のエントリ行番号カラムをマウスでクリックします。

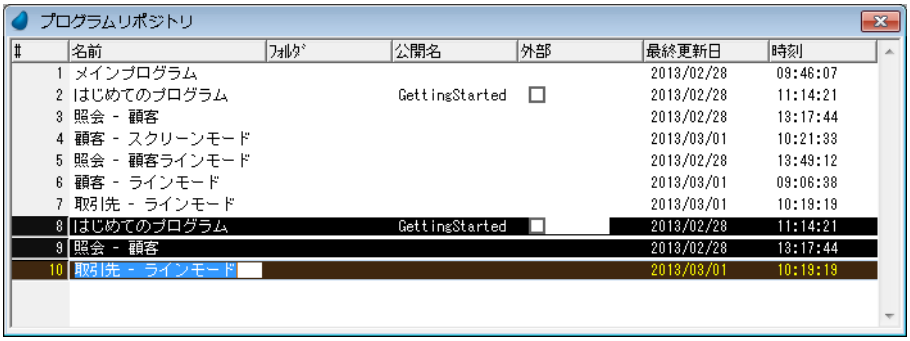
マウスを利用した、非連続的選択

1. Ctrl キーを押下します。
2. 選択したいエントリ行の行番号カラムをクリックしてゆきます。

マルチマーキングの実習

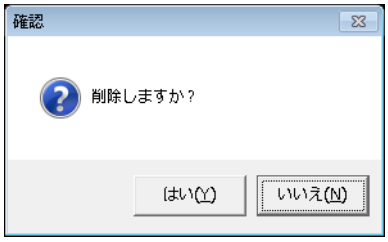
ここでは［プログラム］リポジトリの不要なエントリを、マルチマーキングを利用して一括削除してみましょう。

1. ［プログラム］リポジトリを開きます。
2. エントリ行 #8 にパークします。
3. #8 ～ #10 までを選択します。



4. ［編集］メニューから［行削除］（F3）を選択します。

［確認］ダイアログが表示されますので、［はい］をクリックしましょう。



７．練習問題

［位置付／次候補］機能

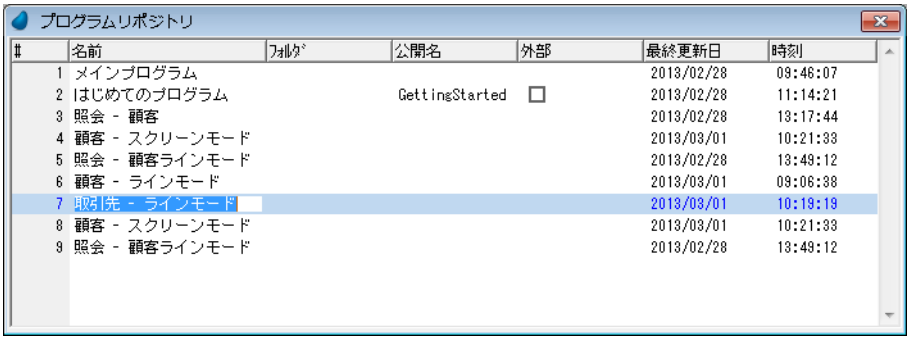
［プログラム］リポジトリにおいて、プログラム名のなかに ' ライン ' という文字列を含むものをすべて位置付けてください。

結果として、#5，#6，#7 のプログラムにカーソルがパークすれば OK です。

［複写登録］機能

エントリ行 #4，#5 のプログラムをリポジトリの最後に複写登録してください。

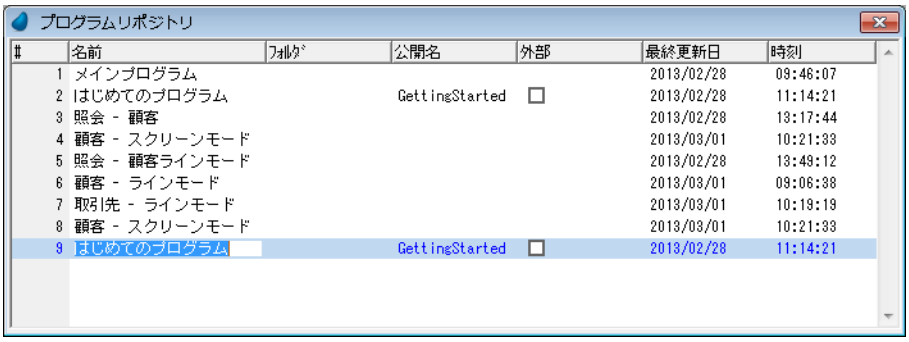
実行結果が次のようになっていれば OK です。



［上書登録］機能

エントリ行 #9 のプログラムを #2 のプログラムで上書きしてください。

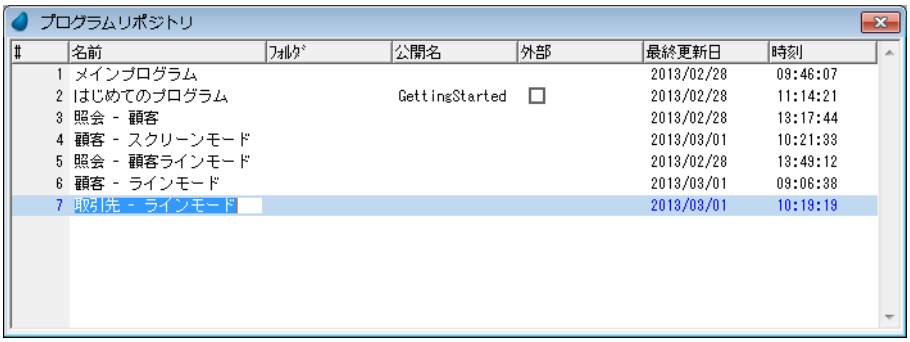
実行結果は次のようになります。



［マルチマーキング］機能

1. エントリ行 #8，#9 をマルチマークしてください。
2. 一括削除してください。

すべての練習課題を終えると、次のように元の状態に戻っています。



８．要約

本章ではリポジトリのエントリを操作する機能について学びました。

これらの機能は、［編集］メニューにあり、次のようなものがあります。

- クイックアクセス
  - 位置付
  - 位置付／次候補
  - 行ジャンプ
- 登録
  - 複写登録
  - 移動登録
  - 上書登録
- ［位置付］および［位置付／次候補］は、検索条件にしたがってリポジトリ内を検索し、位置付ける機能です。
- ［複写登録］は、リポジトリ内の別の場所に、あるエントリを複写してから（挿入）登録する機能です。
- ［移動登録］は、リポジトリ内の別の場所に、あるエントリを移動させる機能です。
- ［上書登録］は、リポジトリ内の既存のエントリを、別の指定したエントリで上書きする機能です。
- マルチマーキング機能は、一括削除を行う場合などに、複数のエントリを同時に選択する機能です。

これらの機能をうまく使用して、リポジトリを効率よく管理しましょう。

第 14 章 クロスリファレンス

クロスリファレンス機能は、モデル、データソース、プログラムなどのオブジェクトの、プロジェクトで使われている場所を検索します。

キーワード

- クロスリファレンス機能
- リポジトリ
- フォルダ
- ［ナビゲータ］ペイン

学習目標

本章を学ぶことで、次の内容に対する理解を深めましょう。

- クロスリファレンス機能を使用して、プログラム内でのデータソースの使われている場所を検索する方法
- クロスリファレンス機能を使用して、プログラム内でインデックスの使われている場所を検索する方法
- 検索結果の表示
- 検索結果のファイルへの保存
- 検索結果の印刷

参照

Magic ヘルプの［Magic xpa リファレンス／ユーティリティ／クロスリファレンス］

１．はじめに

Magic xpa のプログラミングの過程ではオブジェクトを定義して行きますが、それをまた別のオブジェクト内から参照するようになっています。たとえば、データソースをオブジェクトとして定義し、それをプログラムから参照して使用します。Magic xpa はオブジェクト間の参照関係を管理しているので、いつでもその参照関係情報を検索することができます。

これを行うのがクロスリファレンス機能です。

この機能をうまく使用することによって、あるオブジェクトを変更するときの結果を予想することができます。たとえば、あるデータソースをリポジトリから削除しようとするときに、このクロスリファレンス機能を利用してこのデータソースへの参照関係をチェックし、プログラムや他のオブジェクトから参照のないことを確認することができます。

本章ではクロスリファレンス機能の使い方と、その結果の表示、結果の保存、また結果の印刷などについて実習します。

オブジェクトと参照関係のあるリポジトリ

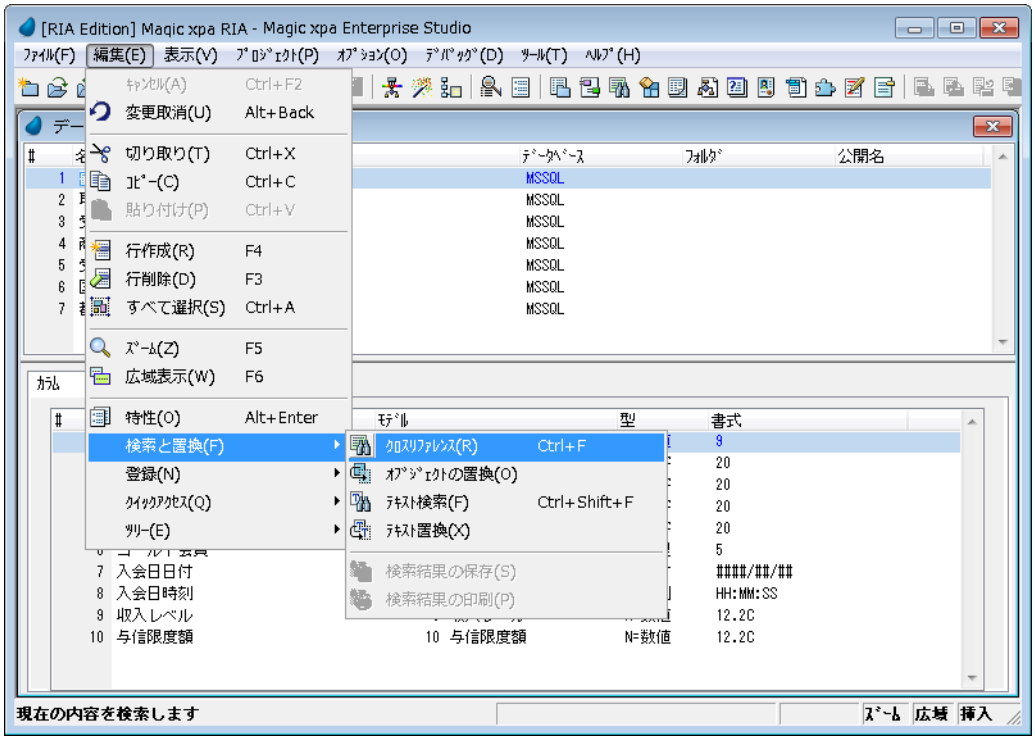
オブジェクト	参照リポジトリ
モデル	・モデル ・データ ・プログラム
データソース	・モデル ・データ ・プログラム
データソースカラム	・モデル ・データ ・プログラム
データソースインデックス	・モデル ・データ ・プログラム
プログラム	・モデル ・データ ・プログラム ・メニュー
プログラム変数項目	・プログラム
プログラム式	・プログラム
プログラムフォーム	・プログラム
プログラムイベント	・プログラム
プログラム入出力ファイル	・プログラム
ヘルプ	・モデル ・データ ・プログラム ・メニュー
メニュー	・モデル ・データ ・プログラム・アプリケーション特性

２．クロスリファレンスの使用例

＜顧客＞データソースの参照関係

ここでは例として＜顧客＞データソースの使われているオブジェクトを、クロスリファレンス機能を使用して調べてみることにします。

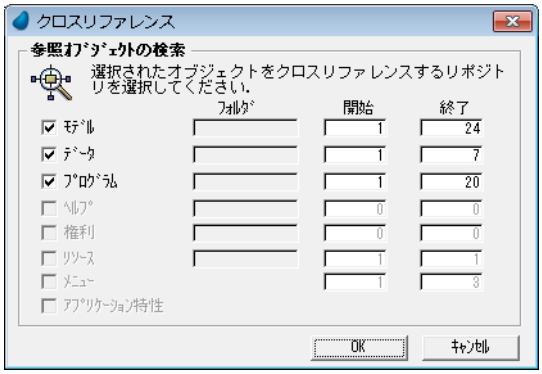
1. ［データ］リポジトリ（Shift+F2）を開きます。
2. 調査対象となる＜顧客＞データソースの行にパークします。
3. ［編集］メニューから［検索と置換／クロスリファレンス］（Ctrl+F）を選択します。



［クロスリファレンス］ダイアログが表示されます。

このダイアログでは検索対象とするリポジトリを選択することができます。

調査対象となるオブジェクトにより、選択できる検索対象のリポジトリが変わります。

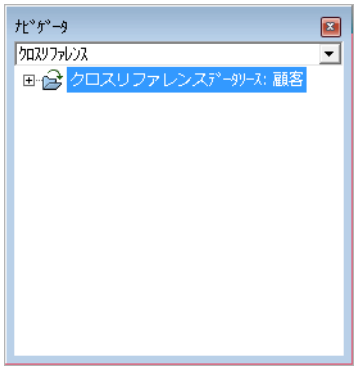


4. ［OK］ボタンをクリックします。

結果の表示

クロスリファレンスの検索結果は、［ナビゲータ］ペイン（「クロスリファレンス」のカテゴリ）にツリー形式で表示されます。

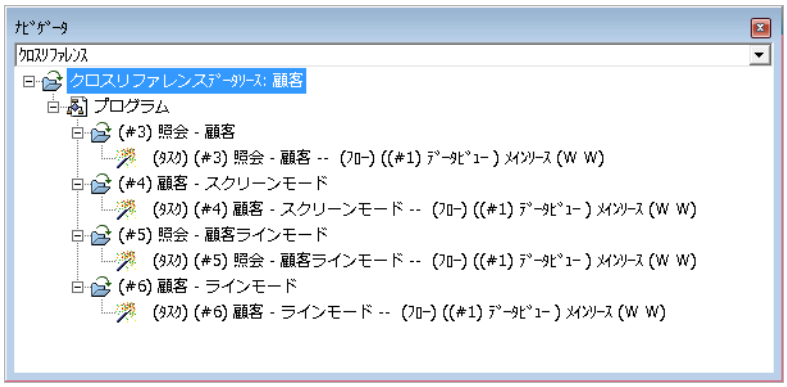
各ノードはクロスリファレンスを意味しています。



- **［ナビゲータ］ペインが表示されていないときは、［表示］メニューから［ナビゲータ］を選択します。**
- **クロスリファレンスは、［ナビゲータ］ペインのドロップダウンリストで［クロスリファレンス］を選択すると表示されます。**

今回の結果を展開して、検索結果を表示してみましょう。

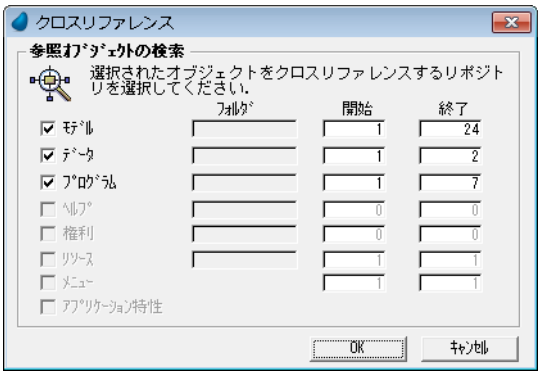
1. ［クロスリファレンスデータソース：顧客］にパークします。
2. ［編集］メニューから、［ツリー／サブツリー展開］を選択します。あるいは、キーボードで '+' のキーを押下します。
3. ［プログラム］ノードが展開されない場合、［プログラム］行にパークし、［編集／ツリー／サブツリー展開］を選択します。
4. 同様にして、全てのサブツリーを展開します。（プログラム #3、#4、#5、#6 の各ノード）



＜顧客コード＞インデックスを参照しているオブジェクト

1. ［データ］リポジトリ（Shift+F2）を開きます。
2. ＜顧客＞データソースにパークします。
3. 下部ペインの［インデックス］タブをクリックします。
4. ＜顧客コード＞インデックスにパークします。
5. ［編集］メニューから、［検索と置換／クロスリファレンス］（Ctrl+F）を選択します。

［クロスリファレンス］ダイアログが開きます。



6. ［OK］ボタンをクリックします。

結果の表示

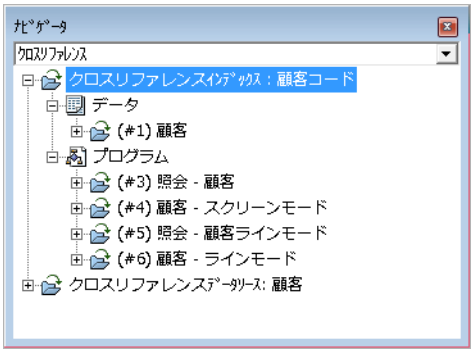
最初の例と同じように、クロスリファレンスの検索結果が［ナビゲータ］ペインに表示されます。

一番上のノードが最新の検索結果になります。

最初の検索結果を縮小させ、今回の結果を展開させてみましょう。

1. ［クロスリファレンスデータソース：顧客］の行にパークします。
2. ［編集］メニューから、［ツリー／サブツリー縮小］を選択します。あるいは、行の左にある、'-' のノードをクリックしてください。

3. ［クロスリファレンスインデックス：顧客コード］の行にパークします。
4. ［編集］メニューから、［ツリー／サブツリー展開］を選択します。あるいは、行の左にある、'+' のノードをクリックしてください。
5. ［データ］および［プログラム］ノードが展開されるまで、上記操作を繰返します。



表示された検索結果を見れば、<顧客コード>インデックスを使用しているデータソースは<顧客>ただ一つであり、プログラムでは 4 つのプログラムでメインソースのインデックスとして参照していることが分かります。

３．クロスリファレンス結果の保存

Magic xpa ではクロスリファレンスの検索結果をテキストファイルとして保存することができます。

1. ［ナビゲータ］ペインでクロスリファレンスが表示されている状態にします。（ドロップダウンリストから「クロスリファレンス」を選択）
2. ［クロスリファレンスデータソース：顧客］の行にパークしてください。

3. ［編集］メニューから、［検索と置換／検索結果の保存］を選択します。［名前を付けて保存］ダイアログが表示されます。
4. ファイルを保存するフォルダを指定します。
5. **<顧客 _ 検索 .txt >**というファイル名を入力してください。
6. ［保存］をクリックします。

保存したテキストファイルを開いて内容を確認してみましょう。

クロスリファレンス結果の印刷

Magic xpa ではクロスリファレンスの検索結果を印刷することができます。

以下の手順はプリンタが接続された環境で行ってください。

1. ［ナビゲータ］ペインでクロスリファレンスが表示されている状態にします。（ドロップダウンリストから［クロスリファレンス］を選択）
2. ［クロスリファレンスデータソース：顧客コード］の行にパークしてください。

3. ［編集］メニューから、［検索と置換／検索結果の印刷］を選択します。

［印刷］ダイアログが表示されるので、出力したいプリンタを選択肢、［OK］ボタンをクリックします。

クロスリファレンス結果の削除

クロスリファレンスの検索結果は削除することができます。これまでの二つの結果を削除してみましょう。

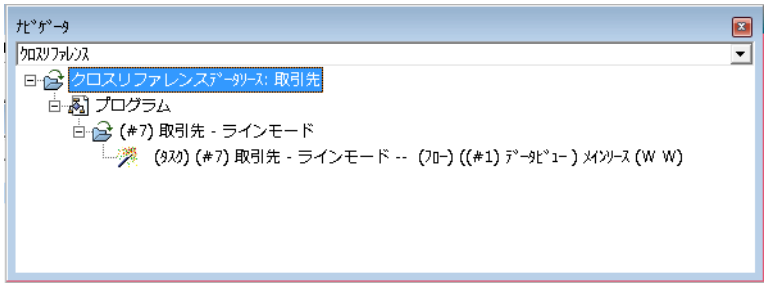
1. ［ナビゲータ］ペインで［クロスリファレンス］を表示させます。
2. ［クロスリファレンスインデックス：顧客コード］行にパークします。
3. ［編集］メニューから、［行削除］（F3）を選択します。

4. ［確認］ダイアログが表示されるので、［はい］をクリックします。
5. ［クロスリファレンスデータソース：顧客］行にパークします。
6. ［編集］メニューから、［行削除］（F3）を選択します。
7. ［確認］ダイアログが表示されるので、［はい］をクリックします。

これでクロスリファレンスがすべて削除されました。

４．練習問題

1. ＜取引先＞データソースを参照しているすべてのオブジェクトを検索してください。結果は次のようになります。



2. ＜取引先コード＞インデックスを参照しているすべてのオブジェクトを検索してください。
3. ＜取引先＞データソースの検索結果を＜取引先 _XREF.txt ＞というファイルに保存してください。（フォルダは任意です）

５．要約

本章ではクロスリファレンスの機能について学びました。

- クロスリファレンス機能を利用して、各オブジェクト（データソース、カラム、インデックス、プログラム、etc.）が他のオブジェクト内で参照されていることを検索することができます。
- ［ナビゲータ］ペインに表示される結果ツリーを展開したり縮小して、内容の確認を行いました。
- クロスリファレンスの検索結果をテキストファイルに保存したり、印刷できることを確認しました。
- クロスリファレンス機能はプロジェクト内容を変更するときに非常に役に立ちます。たとえば、あるオブジェクトを削除するときに、そのオブジェクトがどこからも参照されていないことを確認できます。

第 15 章 １対１のデータリレーション

本章では、１対１のデータリレーションについて学び、Magic xpa で作成する方法を実習します。

キーワード

- 照会リンク
- 登録リンク
- 書出リンク
- メインソース
- 方向
- 戻り値
- リンク評価
- リンク式の再計算

学習目標

本章を学ぶことで、次の内容に対する理解を深めましょう。

- １対１のデータリレーションとは何か
- Magic xpa で１対１のデータリレーションを実現する方法
- 異なるリンクの種類とそれぞれの特徴
- リンク式の再計算メカニズムと１対１のデータリレーション維持に果たす役割
- リンク特性への戻り値項目設定とリンク成功／失敗の判断

参照

Magic ヘルプの［Magic xpa リファレンス／［データビュー］エディタ／リンク定義］

１．はじめに

データソースは特定の対象に関するデータを保持するように設計されています。たとえば、

- ＜顧客＞データソース …… 顧客に関する情報：会社名、担当者、住所、電話番号など
- ＜商品＞データソース …… 商品に関する情報：商品名、価格、数量、仕入先／製造元など
- ＜受注＞データソース …… 受注に関する情報：顧客名、受注商品名、個数、合計価格、割引率など

多くの場合、これらのデータソースには相互に関連があります。たとえば、＜受注＞データソースにはその注文をした＜顧客名＞項目がありますが、この項目を入力するとき、この顧客が既登録顧客であるかどうかを確認したり、さらに詳細情報を知りたいような場合、＜顧客＞データソースに接続し、＜顧客名＞を使用して検索をすることでしょう。

Magic xpa では、メインソースの各レコードについて他のデータソースの対応する特定の１レコードを関連付けることを、１対１のデータリレーションを構築すると言い、具体的にはタスクの［データビュー］エディタ上の［リンク］コマンドによって実現します。

本章では、この'１対１のデータリレーション'について、'１対多のデータリレーション'との対比を行いつつ説明し、［リンク］コマンドの実習を通して、１対１のデータリレーションをプログラムで実現する方法を学びます。

２．'１対１'と'１対多'のデータリレーション比較

Magic xpa ではデータソース間に次の二つの関係を構築することができます。

- １対１のデータリレーション
- １対多のデータリレーション（この詳細については、第 17 章で学習します。）

二つのリレーションの対比

１対１のデータリレーション	１対多のデータリレーション
共通の項目がリンクを保持するために用いられる	共通の項目がリンクを保持するために用いられる
メインソースとリンク先データソースは同一タスクに定義されている	各データソースは、異なるタスクでそれぞれメインソースとして定義されている
リンク先データソースから１レコードのみが返される	一般に複数のレコードが返される
Magic xpa のリンク式再計算メカニズムがリンクを維持する	Magic xpa の［サブフォーム］コントロールのメカニズムが関係を維持する

３．他のデータソースへのリンク

Magic xpa では、［データビュー］エディタにリンクヘッダ行を定義することでデータソースを接続し、１対１データリレーションを確立します。

２つのデータソースを接続するためには、ともに同じ項目を含んでいることが必要です。通常の場合、共通となる項目はリンク先となるデータソースのインデックスセグメントになっています。

タスク実行の間、Magic xpa はメインソースの各レコードに対し、リンク先のデータソースから１レコードずつ読み込みます。

リンクヘッダ行

他のデータソースへのリンクは、Magic xpa のデータビュー定義で行います。つまり［リンク］コマンドの定義は［データビュー］エディタで行います。

リンクの定義には次のような要素を含んでいます。

- リンクヘッダ行では次のような内容を定義します。
  - リンク先データソース
  - リンクの種類
  - リンクに使用するインデックス
- リンクセクション内にリンク先データソースのインデックスのカラム定義を行う詳細行が作られます。

- ［位置付］条件をリンク先データソースの共通項目に対し定義します。
- リンクセクションを閉じるための［リンク終了］コマンド行。
- ［リンク］から［リンク終了］までのリンクセクション内のリンク先データソースのカラム定義行。

［リンク］コマンドの使用法

［リンク］コマンドは次のような場合に使用します。

- レコードのデータビューを拡張するとき
- 所定のレコードがリンク先データソースにあるかどうかの確認をするとき（データの存在確認）

レコードデータビューの拡張

メインソースからの情報を拡張するときに、他のデータソースからの情報を読込んで追加したい場合があります。そのようなときに［リンク］コマンドを使用します。

例

次に示すようなデータソースがプロジェクトに定義してある場合を考えます。

- ＜受注＞…… 受注番号、受注日付、顧客コード、受注数量、消費税額、支払い方法などが含まれます。
- ＜顧客＞…… 顧客コード、顧客名、国名、都市名、住所、ゴールド会員などが含まれます。

＜受注＞データソースのレコードを表示するプログラムは今までの実習の範囲で作成できます。そのプログラムでは、受注に対応する顧客コードを表示することはできます。しかしこのとき、同じプログラムで＜顧客名＞を表示するためには、プログラムのデータビューを拡張しなければなりません。つまり、＜顧客＞データソースにリンクし、＜顧客コード＞によって対応する顧客レコードに位置付け、＜顧客名＞を表示させるようにする必要があるのです。

データの存在確認

［リンク］コマンドのもう一つの用途は、リンク先データソースに該当するデータが存在するかどうかの確認を行うことです。

先ほどの例において、＜受注＞データソースには＜顧客コード＞があり、＜顧客＞データソースには顧客に関する全ての詳細情報を持っていました。

新しく受注データを登録するとき、エンドユーザは受注データの一部として＜顧客コード＞の入力が求められます。

このとき、エンドユーザが入力する情報は、すでにシステムが持っている情報と照合されなければなりません。こういう場合に［リンク］コマンドを利用し、エンドユーザが入力する＜顧客コード＞が、＜顧客＞データソースに存在するかどうかを確認することができるのです。

リンクの種類

Magic xpa の［リンク］コマンドには 5 種類あり、それぞれ動作が異なります。

- L= 照会リンク …… リンク先データソースの既存レコードへの接続と読込みを行いたい時に使用します。対応する既存レコードが存在しないときは、データビューには何も読込まれません。
- W= 書出リンク …… 照会リンクと似ていますが、既存レコードが存在している時には読込みを、存在しない時にはリンク先のデータソースにレコードを作成したい場合に使用します。
- C= 登録リンク …… リンク先データソースに新規レコードを作成したい場合に使用します。

以下の 2 つは、SQL 系データベースで使用できます。本コースでは取り扱いません。

- I= 結合リンク …… プログラムのデータビューには、リンク先データソースに対応する既存レコードが存在するメインソースのレコードのみが読込まれます。
- O= 外部リンク …… このコマンドは、［照会リンク］と同じ動作をしますが、データビューの扱いが異なります。プログラムのデータビューには、リンク先データソース内に既存レコードが存在するか否かにかかわらず、メインソースの全てのレコードが読込まれます。

４．受注管理プログラムの実習

このセクションでは、＜受注管理＞プログラムを作成し、＜受注＞データソースのデータと＜顧客＞データソースから［リンク］コマンドによって取得した追加データとを表示させてみましょう。

この実習は 3 つのステップで構成されます。

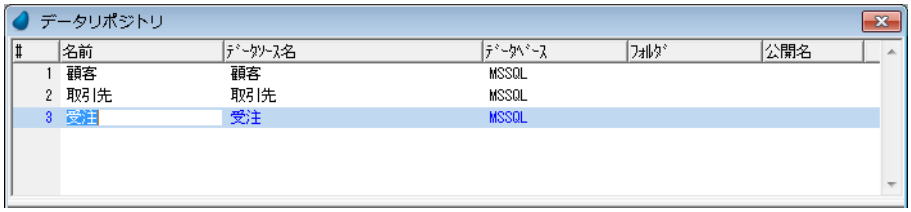
- ＜受注＞データソースの定義
- ＜受注管理＞プログラムの作成
- ＜受注管理＞プログラムのデータビューを＜顧客＞データソースへのリンクを使用して拡張

＜受注＞データソースの定義

1. ［プロジェクト］メニューから［データ］（Shift+F2）を選択し、［データ］リポジトリを開きます。
2. 最後の行（#2）にパークします。
3. 一行作成（F4）します。

4. 次のようにパラメータ設定します。

#	名前	データソース名	データベース	フォルダ	公開名
3	受注	受注	MSSQL		



カラムの定義

カラムを次のように定義してください。

1. ＜受注＞データソースにパークし、下部ペインで［カラム］タブを選択します。
2. 新規に 6 行作成し、以下のように定義してください。

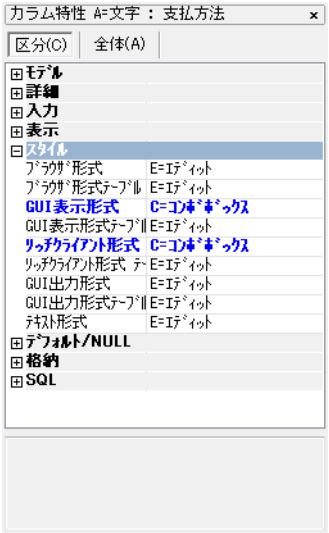
#	名前	モデル	型	書式
1	受注番号	0	N= 数値	6
2	受注日	0	D= 日付	#####/##/##
3	顧客コード	0	N= 数値	9
4	合計金額	0	N= 数値	8
5	消費税	0	N= 数値	8
6	支払方法	0	A= 文字	30

リッチクライアント コントロール設定

Magic xpa では各カラムのリッチクライアント表示フォーム上でのデフォルト表示形式を指定できます。これは各カラムの［スタイル］特性で指定します。

この実習例では、＜支払方法＞カラムをリッチクライアント表示フォームに配置するとき、［コンボボックス］コントロール形式で表示させるようにしてみましょう。

1. ＜支払方法＞カラムにパークします。
2. ［カラム］特性（Alt+Enter）を開きます。
3. ［スタイル／リッチクライアント形式］で［C= コンボボックス］を選択します。



リッチクライアント形式を［コンボボックス］に設定した場合、選択肢は一組のリストになります（たとえば、性別を選択するコンボボックスのように）。このような場合、リストはデータソースにカラムを定義するときに作成します。

さらに、Magic xpa では選択肢のリストにおいて、エンドユーザが見る値（［選択表示リスト］）と、データソースのカラムに保存される値（［選択項目リスト］）を異なる値にすることができます。

この実習例では、＜支払方法＞として次の 4 つの選択肢があるものとしましょう。

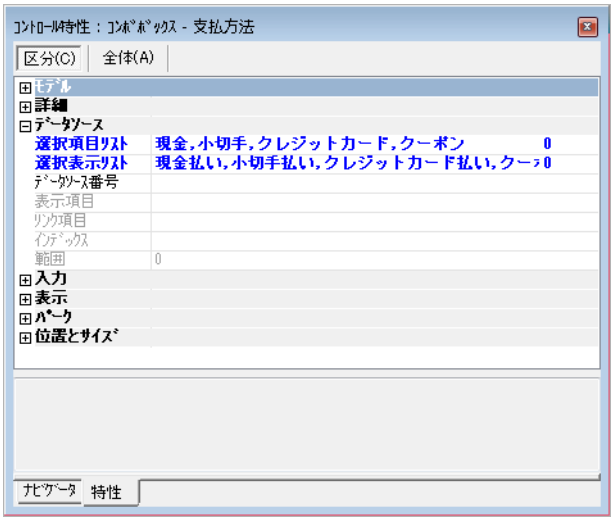
- 現金
- 小切手
- クレジットカード
- クーポン

4. ［リッチクライアント形式］特性からズーム（F5）して［コンボボックス特性］ダイアログを開きます。
5. ［選択項目リスト］特性に値を ',' で区切り順に入力します：**＜現金，小切手，クレジットカード，クーポン＞**
6. ［選択表示リスト］特性にはたとえば次のように入力してください：**＜現金払い，小切手払い，クレジットカード払い，クーポン券で払う＞**
7. ［コンボボックス特性］ダイアログを閉じます。

- ダイアログ右上角にある×アイコンをクリックするか、Esc キーを押下します。（Magic xpa は設定を保存します。）

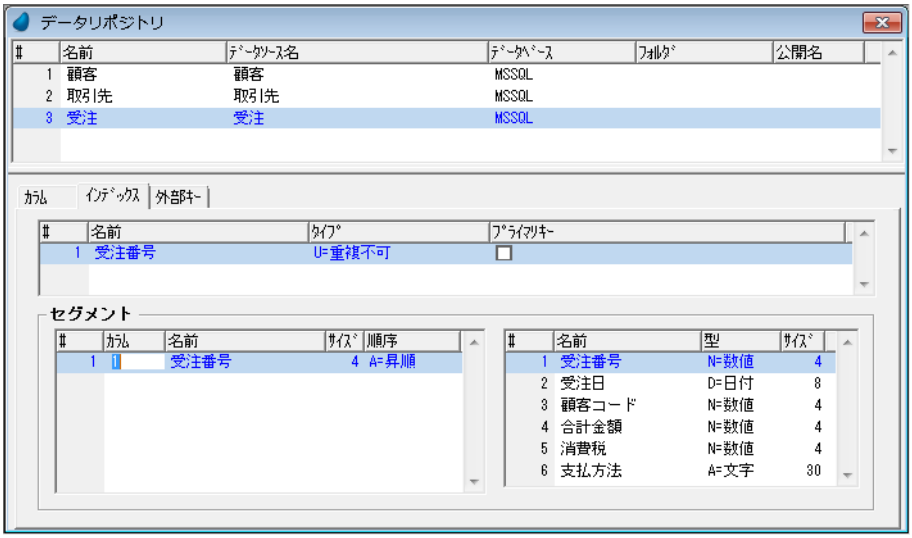
［選択項目リスト ］と［選択表示リスト ］について

- **［選択項目リスト］は、チョイスコントロールで戻り値となる選択肢の文字列を設定します。各戻り値はカンマ（半角）で区切ります。**
- **［選択表示リスト］は、チョイスコントロールの戻り値に対応する表示文字列を設定します。表示文字列の数は、［選択項目リスト］の戻り値の数に一致していなければなりません。**



インデックス＜受注番号＞の定義

1. 下部ペインの［インデックス］タブを選択します。
2. 一行作成（F4）します。
3. 次のインデックスを定義します。
   - 名前 …… ＜受注番号＞
   - タイプ …… ［U= 重複不可］
4. ［セグメント］テーブルにズーム（F5）します。
5. 一行作成（F4）します。
6. ［カラム］欄からズームして右のペイン（項目一覧）に移動します。
7. ＜受注番号＞を選択します。（Enter キー押下）
8. 左のペインの［名前］カラムに＜受注番号＞と表示されます。
9. Enter キーを押下して、［インデックス］ペインに戻ります。

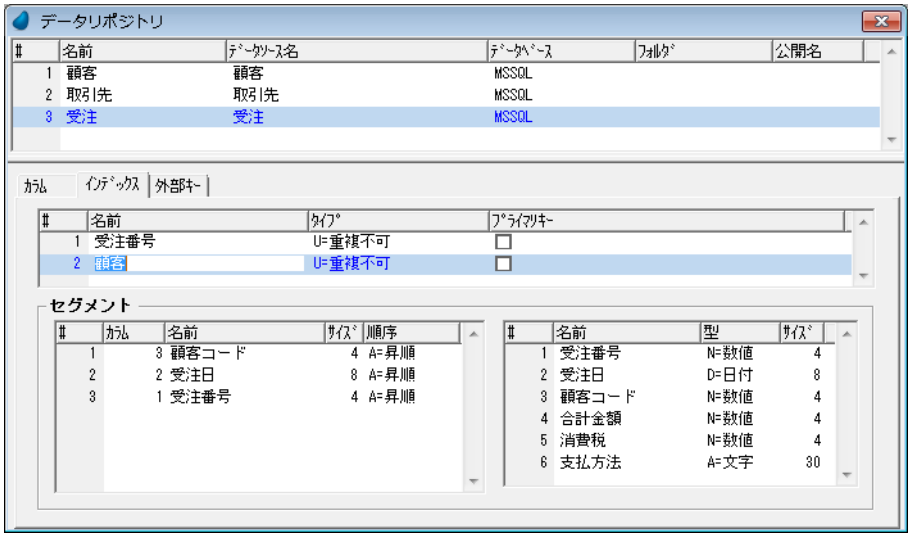


インデックス＜顧客＞の定義

インデックスをもう一つ定義しましょう。

1. ［インデックス］ペインで一行作成します。
2. 次のインデックスを定義します。
   - 名前 …… **＜顧客＞**
   - タイプ …… ［U= 重複不可］

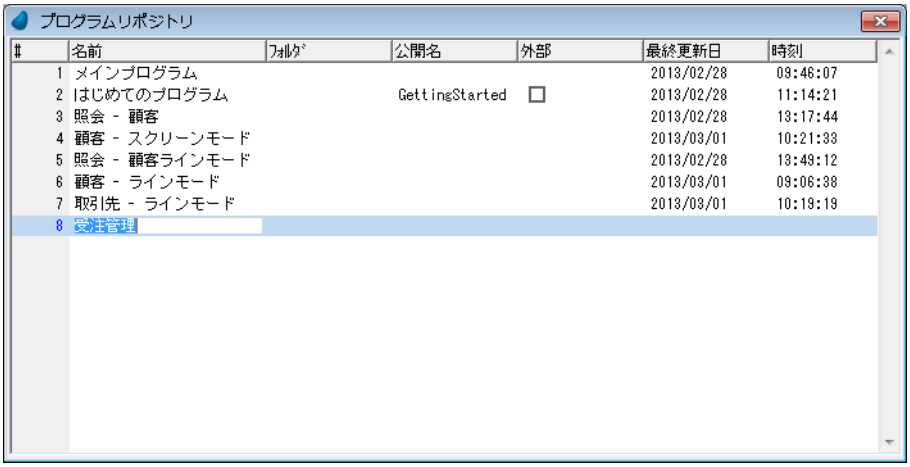
3. ［セグメント］テーブルにズームします。
4. 一行作成（F4）します。
5. ［カラム］カラムから右の項目一覧にズームします。
6. ＜顧客コード＞を選択します。（Enter キー押下）
7. ４－６の操作を繰返し、セグメントとして＜受注日＞、＜受注番号＞を追加します。
8. Enter キーを 2 回押下して、上部ペインに戻ります。



これで、＜受注＞データソースの定義が終了しました。

＜受注管理＞プログラムの作成

1. ［プログラム］リポジトリ（Shift+F3）を開きます。
2. 最後の行にパークし、一行作成（F4）します。
3. ［名前］カラムに**＜受注管理＞**と入力してください。



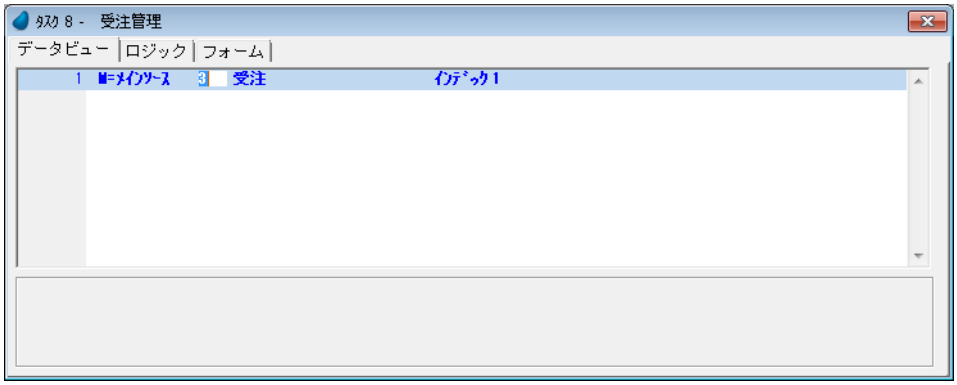
4. ズームして＜受注管理＞プログラムを開きます。

5. ［タスク特性］ダイアログが開きます。［タスクタイプ］特性を「C= リッチクライアント」に設定して［OK］ボタンをクリックします。

メインソースの定義

1. ［データビュー］エディタを開きます。
2. メインソース定義行のパラメータを次のように設定します。

#	データソース番号	データソース名	インデックス
1　M= メインソース	3	受注	1

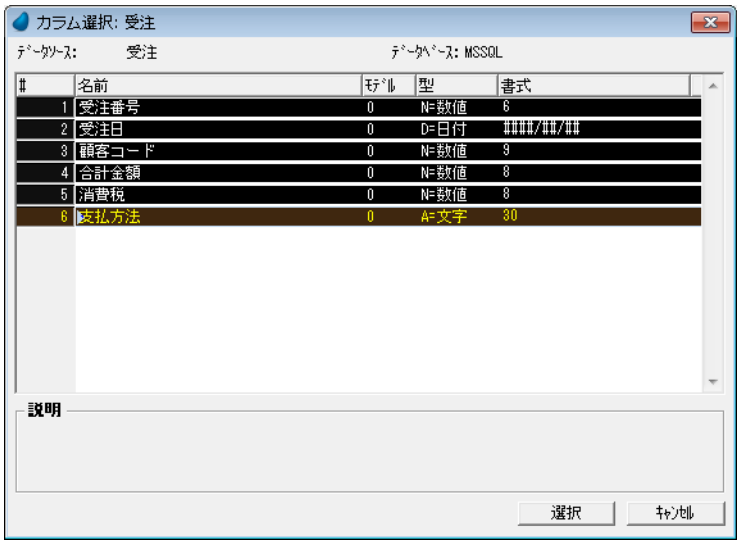


メインソースカラムの定義

1. 一行作成（F4）します。
2. ［カラム番号］欄からズームして［カラム選択］テーブルを開きます。
3. カラム #1 から #6 までをすべて選択します。

4. ［選択］ボタンをクリックします。

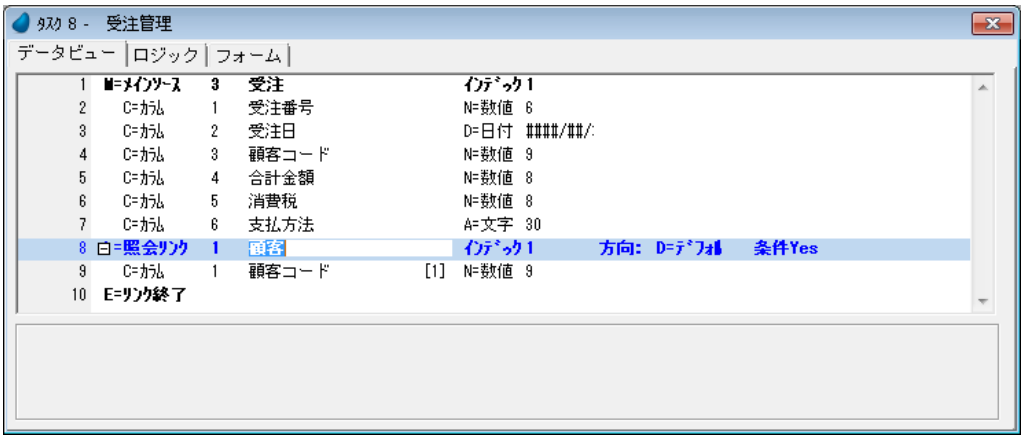
5. ＜受注＞データソースのすべてのカラムがメインソースカラムとして登録されたことを確認してください。



顧客データソースへのリンク

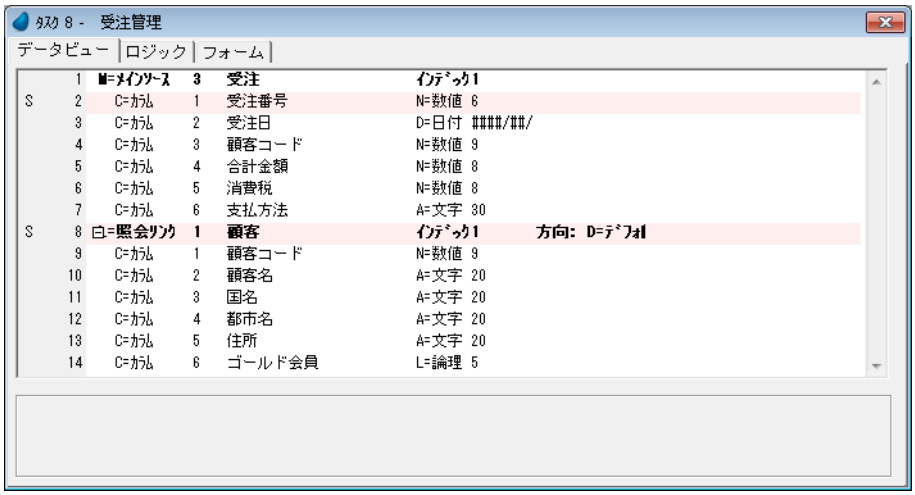
1. ［データビュー］エディタの最終行にパークします。
2. ヘッダ行（Ctrl+H）を一行作成します。
3. 次のようにパラメータを設定します。

#	データソース番号	データソース名	インデックス	方向	条件
8　L= 照会リンク	1	顧客	1	D= デフォルト	Yes



- ［E= リンク終了］コマンド行は、［リンク］コマンドを定義すると自動的に作成されます。
- ［リンク］コマンドを定義し、さらにインデックスを設定すると、インデックスセグメントになっているカラムがリンクセクション内に自動的に登録されます。

4. ＜顧客コード＞カラム行（#9）にパークします。
5. 一行作成（F4）します。
6. ［カラム番号］欄からズームして［カラム選択］テーブルを開きます。
7. ＜顧客名＞の行（#2）にパークします。
8. Shift キーを押下しながら #6 の番号をクリックし、#2 ～ #6 までの行を選択します。
9. ［選択］ボタンをクリックします。
10. ＜顧客＞データソースのカラムがリンクセクション内に定義されました。



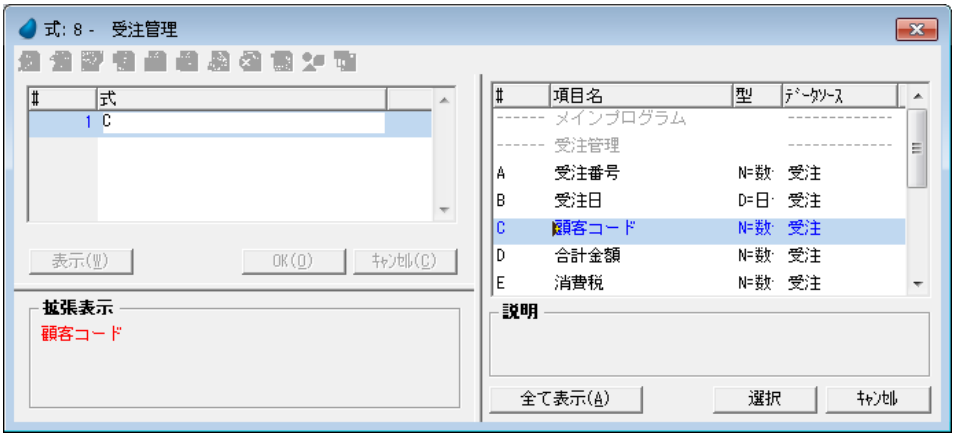
［位置付］パラメータの設定

この例では、＜顧客コード＞カラムが 2 つのデータソースの共通項目になっています。

＜受注＞データソースの＜顧客コード＞を、＜顧客＞データソースの＜顧客コード＞カラム行の［位置付］パラメータとして設定します。

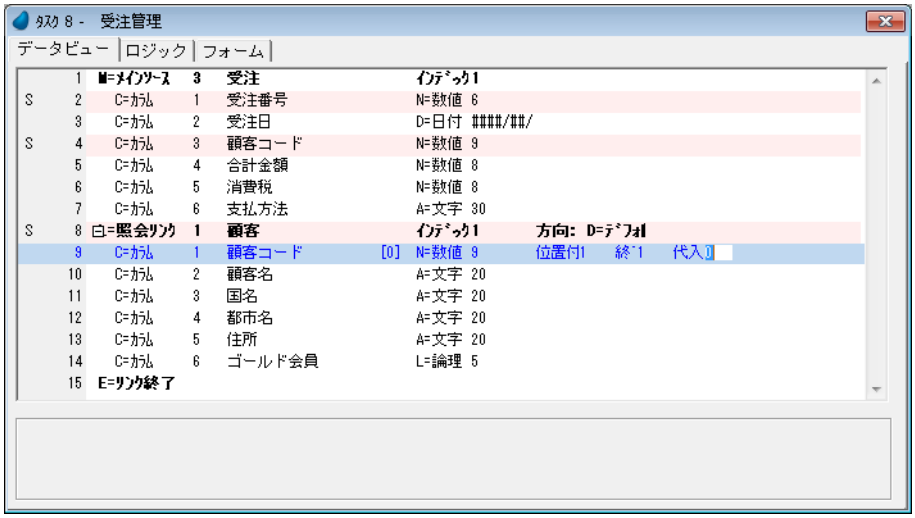
1. ＜顧客＞データソースの＜顧客コード＞カラム行（#9）にパークします。
2. ［位置付］欄からズーム（F5）し、［式］エディタを開きます。
3. 一行作成（F4）します。

4. ズーム（F5）して項目リストに移動し、＜受注＞データソースの＜顧客コード＞を選択します。



5. ［式］エディタに ＜ C ＞ と表示されることを確認してください。
6. ［OK］ボタンをクリックします。

7. ［終了］欄に、＜ 1 ＞と入力します。（式番号 1 を指定したことになり、これは［位置付］欄と同じ式を指定していることになります。）



これで［リンク］コマンドの定義が終わりました。

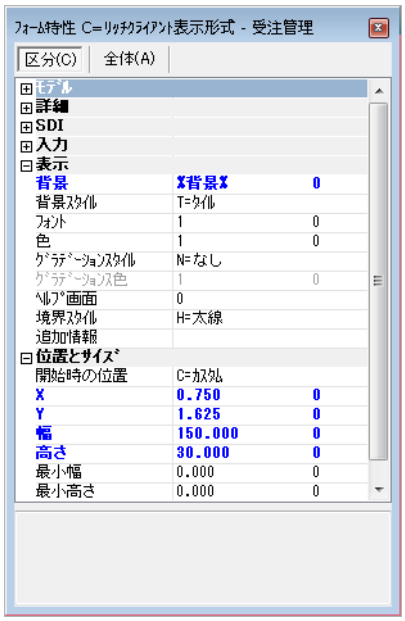
［位置付］および［終了］の設定を行うとき、項目としてリンク先データソース（この場合は＜顧客＞）の項目を間違えて指定しないように留意しましょう。

＜受注管理＞フォームの作成

このセクションでは、＜受注管理＞プログラムのリッチクライアントフォームを作成します。フォームには＜受注＞データの詳細と、＜顧客＞データの一部を表示させることにします。

1. ＜受注管理＞プログラムを開き、［フォーム］エディタを選択します。
2. 2 行目（#2）の＜受注管理＞フォームにパークします。
3. フォーム特性（Alt+Enter）を開きます。

4. ［表示／背景］特性に、＜ % 背景 % ＞と入力してください。
5. ［位置とサイズ／幅］特性に、＜ 150 ＞と入力してください。
6. ［位置とサイズ／高さ］特性に、＜ 30 ＞と入力してください。
7. ［フォーム］エディタの #2 からズームして＜受注管理＞フォームを開きます。



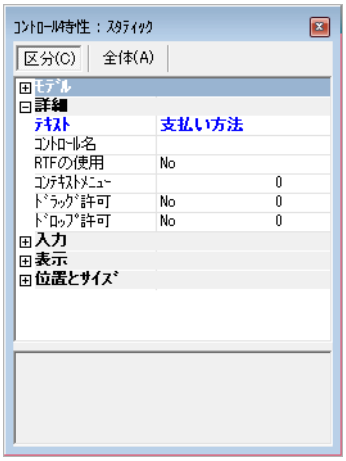
これからコントロールをフォーム上に配置しましょう。フォーム上の位置はイメージを参考にしてください。

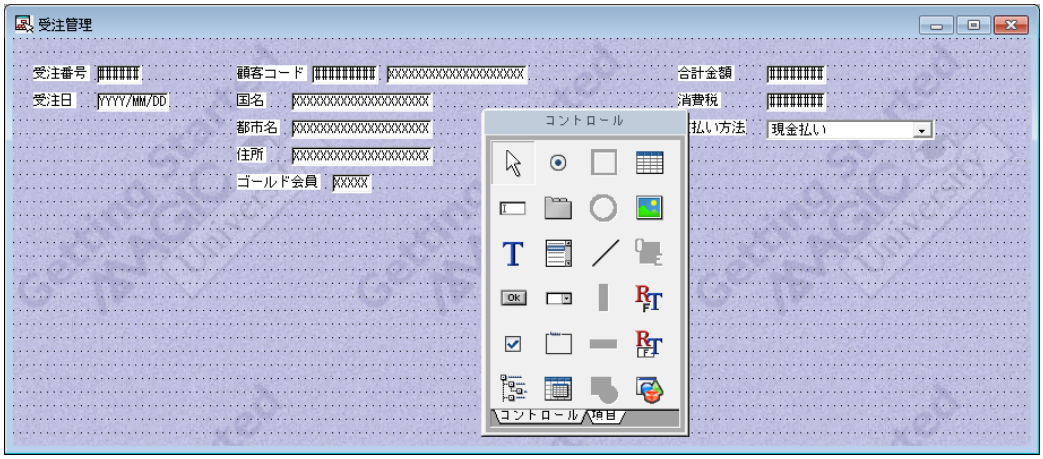
8. ［コントロール］パレットで［項目］タブをクリックします。
9. ［項目］タブで＜ A: 受注番号＞をクリックします。
10. フォーム上の左上の場所をクリックして、＜受注番号＞の見出し文字と、［エディット］コントロールを配置します。
11. 上記の 9,10 のステップを繰り返し、＜受注＞データソースの各項目をフォームに配置してください。
    ＜受注日＞，＜顧客コード＞，＜合計金額＞，＜消費税＞，＜支払い方法＞
12. 同様にして、＜顧客＞データソースの次の項目もフォームに配置しましょう。
    ＜顧客名＞，＜国名＞，＜都市名＞，＜住所＞，＜ゴールド会員＞
13. ＜顧客名＞の見出し文字の［ラベル］コントロールを削除してください。

＜支払方法＞のコンボボックスで、

14.［コントロール］パレットで［ラベル］コントロールを選択し、＜支払方法＞コンボボックスの左に配置します。

15.［ラベル］コントロールの特性で、［テキスト］特性に **＜支払い方法＞** と入力してください。





見出し文字の編集

16.［ラベル］コントロールをすべて選択します。（Ctrl キーを押下しながら、一つ一つ順にクリックして選択します。）

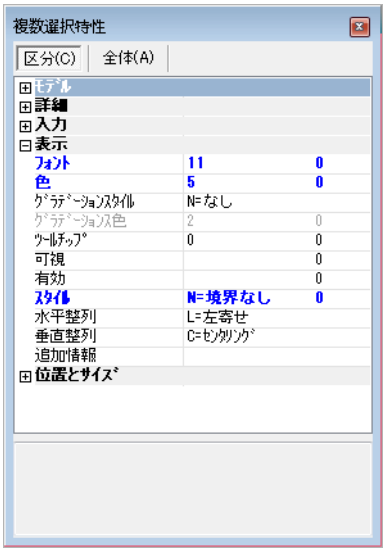
17.特性シートを開きます。（Alt+Enter キーを押下すると、［複数選択特性］シートが開きます。）

18.［表示／色］特性からズームし、基本色テーブルの＜見出し文字＞（#5）を選択します。

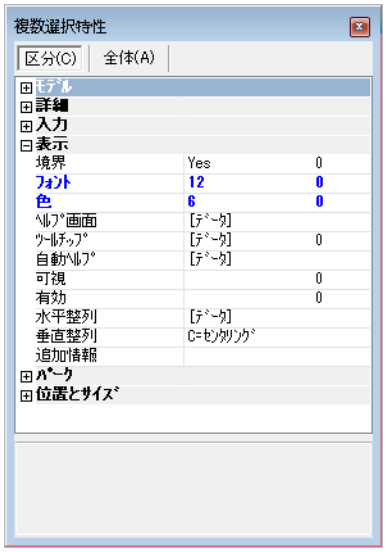
19.［表示／フォント］特性からズームし、フォントテーブルから＜見出し文字＞（#11）を選択します。

20.すべての［ラベル］コントロールが選択された状態で、［コマンド］パレットの［サイズ調整］アイコンをクリックします。

21.Space キーを押下して、コントロール選択を解除します。



［エディット］コントロールの編集



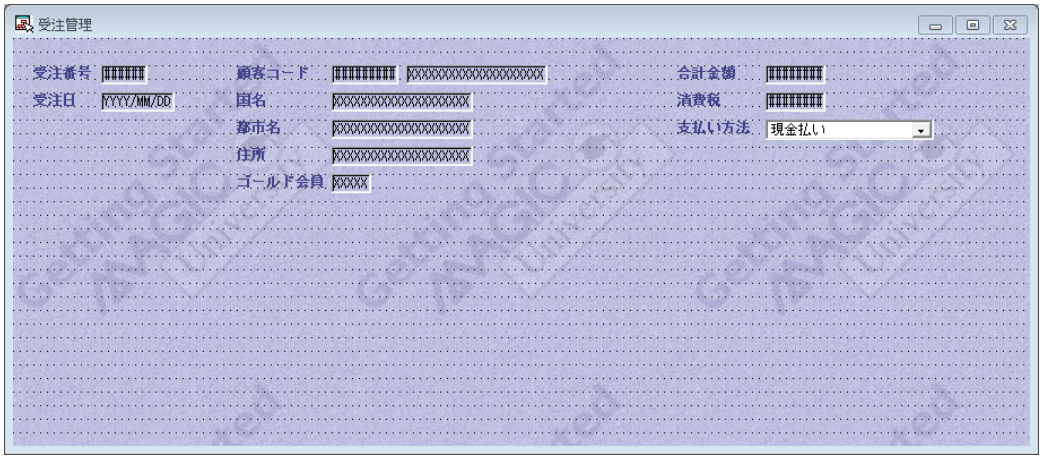
22.すべての［エディット］コントロールを選択してください。

23.［複数選択特性］シート（Alt+Enter）を開いてください。

24.［表示／フォント］特性に＜ 12 ＞を入力してください（フォントテーブル #12 ：文字）。

25.［表示／色］特性に＜ 6 ＞を入力してください（基本色テーブル #6 ：文字）。

26.［エディット］コントロールおよびコンボボックスの位置を調整し、整列機能を使用してイメージのようにしてください。



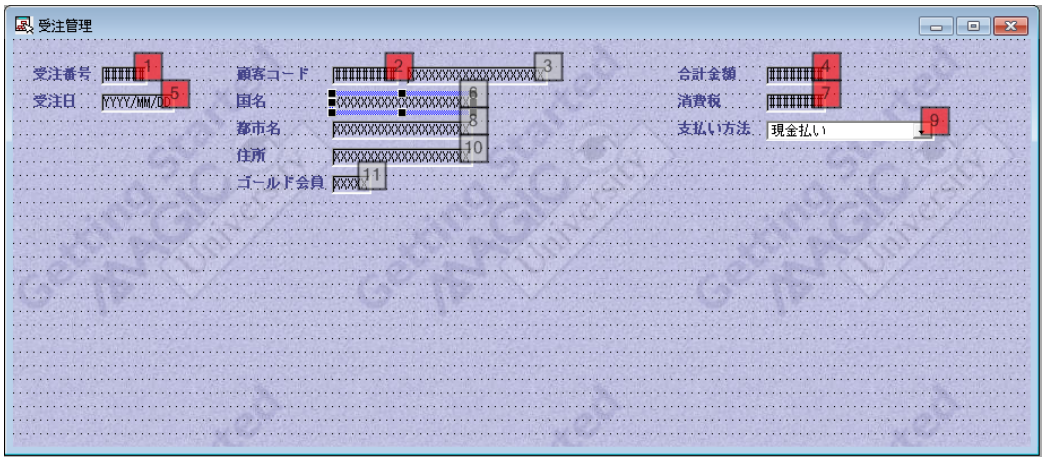
［フォーム］エディタの使い方が分からないときは、「第 7 章：フォームの設計」（79 ページ）を復習しましょう。

このプログラムでは、顧客コード以外の顧客情報については編集できないようにする必要があります。

TAB 順序の設定

次に［TAB 順序］を設定し、エンドユーザが Tab キーでカーソル操作するときに、秩序ある動きとなるようにしましょう。

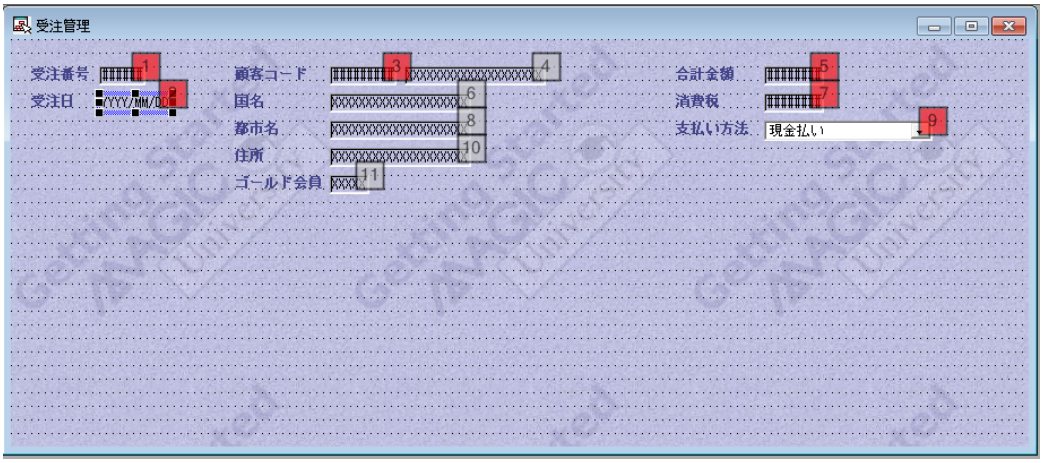
自動的に設定された TAB 順序は必ずしも使いやすくないことがあり、この例でもカーソルが水平方向に行ったり来たりします。



このようなときは、［自動 TAB 順序］を無効にし、手動でタブ順序を設定しましょう。

1. ［描画］メニューから、［順序／自動 TAB 順序］を選択し、手動設定できるように切り替えます。（アイコンが平面になるようにします。）
2. ＜受注日＞［エディット］コントロールを選択し、コントロール特性を開きます。

3. ［パーク／ TAB 順序］特性を＜ 2 ＞とします。
4. ［コマンド］パレットの［TAB 順序表示］アイコンをクリックしてタブ順序を表示させます。
5. 手動で変更した結果を反映して次ページのイメージのように順序が変わりました。
6. 変更内容を保存し、プログラムを閉じてください。



＜受注管理＞プログラムの実行

1. F7 キーを押下して＜受注管理＞プログラムを実行しましょう。
2. ＜受注番号＞欄に＜ 1 ＞と入力してください。
3. ＜受注日＞欄に、例えば今日の日付を入れてください。（例えば、＜ 2007/05/01 ＞）
4. ＜顧客コード＞欄に、＜ 2 ＞と入れてください。
5. Tab キーを押下して、次のコントロールに移動します。

カーソルが＜合計金額＞に移動するとともに、入力した顧客コードの顧客詳細情報が表示されます。

6. ＜顧客コード＞の値を 3，4，5 などと変えてみて、詳細情報が変化することを確認してください。このとき、存在しない顧客コード（たとえば＜ 10 ＞）を入力した場合、詳細情報がクリアされることに留意してください。
7. ＜顧客コード＞の値を＜ 2 ＞に戻しておいてください。（マジック花子）
8. Esc キーを押下してプログラムを終了し、データをデータベースに保存します。

５．これまでの要約

作成した＜受注管理＞プログラムのデータビューには 2 つのデータソースのカラムが定義してあります。

- ＜受注＞データソースのカラム
- ＜顧客＞データソースのカラム

この 2 つのデータソースは、共通の値を持つカラム、すなわち＜顧客コード＞を含んでいます。

これらのデータソースのデータ関係は、＜顧客コード＞が＜顧客＞データソースではユニークな項目で、値が重複しないため、１対１のリレーションであるといえます。

プログラムでは、照会リンクを用いて顧客詳細情報を読み込みました。（ただし、対応する顧客データが存在する場合のみ）

＜受注管理＞プログラムのフォームでは＜受注＞情報詳細と、＜顧客＞情報の一部を表示するようになっています。

［リンク］コマンドの詳細について
次の表に示す特性は、［リンク］コマンド行に定義するものの一部です。

特性	説明
リンク種類	使用するリンクの種類を選択します。 • L= 照会リンク（デフォルト） • W= 書出リンク • C= 登録リンク • I= 結合リンク • O= 外部リンク
データソース番号	リンク先データソースの識別番号です。 ズームすることで、データソース一覧から選択できます。
インデックス	リンク先データソースのインデックス番号を指定します。 メインデータソースとリンク先データソースの間のリンクは、リンク先データソースのインデックスセグメントに基づいて実行されます。 インデックスを適切に設定することで、レコードの検索時間を短縮することができます。 ズームして［インデックス一覧］を開き、リンクで使用するインデックスを選択できます。
方向	ここでは、レコードが検索される方向を指定します。 検索はリンクに指定されているインデックスを基に実行されます。 以下のオプションが指定できます。 • D= デフォルト …… インデックスの定義内容に基づいて検索されます。 • R= 逆方向 …… インデックスの定義内容とは反対の方向に検索されます。つまり、［リンク］コマンドと［リンク終了］の間に定義されているデータ項目の［位置付］特性や［範囲］特性は全て、逆方向に評価されます。

条件	この特性は、リンク条件を定義します。
	この特性に［Yes］、［No］、［式］のいずれかを入力することで、リンク条件を定義できます。式の評価対象は論理値でなければなりません。
	• Yes（式の評価結果が［True］）…… 必要なレコードがそのリンク条件によって取り出されます。
	• No（式の評価結果が［False］）…… レコードの取り出しは実行されません。このレコードはリンクに失敗したものとして動作し、以下の処理が行われます。
	  • 定義されたリンク項目の全ての値がデフォルトに戻されます。
	  • リンクの［戻り値］特性は［False］となります。

６．［リンク式の再計算］のメカニズム

＜受注管理＞プログラムを実行させる実習で、＜顧客コード＞を途中で変えてみることを試みました。

このとき、＜顧客コード＞を変更する都度、顧客の詳細情報も変わりました。

このことは、Magic xpa が＜顧客コード＞の内容が変わる都度、［リンク］コマンドを再実行し、新しい＜顧客コード＞に対応する詳細情報を読込んでいることを示しています。

このような Magic xpa の動作を、［リンク式の再計算］と呼んでいます。

Magic エンジンが管理する処理のなかで、［リンク］コマンドは非手続き型の処理になり、以下の条件がすべて成立したときに実行されます。

- 項目が［位置付］式で使用される項目の一部である。
- 項目が［データビュー］エディタでリンクセクションより上部で定義されている。
- その項目の値が変更された。

７．リンクの戻り値

リンクが成功したことを示す必要性

＜受注管理＞プログラムを実行させる実習で、＜顧客コード＞に＜ 5 ＞を入力してみました。このときは詳細情報がクリアされました。なぜなら、5 という値の顧客コードのレコードが＜顧客＞データソースに存在しなかったからです。

エンドユーザは顧客詳細情報が空欄となったとき、このままではそれがなぜかを正しく判断できません。つまり、次の２つの場合が考えられるからです。

- 顧客が存在しない
- 顧客は存在するけれど、詳細情報はクリアされてしまっている

エンドユーザが正しく判断できるためには、リンクが成功したのか失敗したのかを示してあげることが必要です。

リンク［戻り値］の特性

［リンク］コマンドの特性の一つに［戻り値］があります。

これはリンクの結果を次のように論理値または数値で返します。

- True または 0(Zero) …… リンクが成功（リンク先に対応するレコードがある）
- False または 1(One) …… リンクに失敗

この戻り値を受け取るためには、この［戻り値］特性欄に論理型または数値型の項目（通常は変数項目）を設定する必要があります。

この戻り値を使用することで、エンドユーザに警告することが可能になります。

［戻り値］の使用例

ここでは＜受注管理＞プログラムを改良し、＜顧客＞データソースに該当する＜顧客コード＞のレコードが見つからなかった場合には警告を表示するようにしてみましょう。

戻り値項目の追加

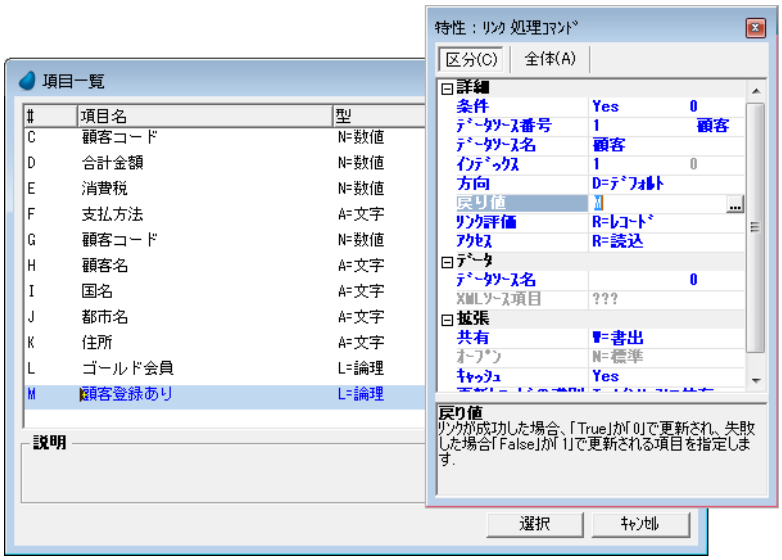
1. ＜受注管理＞プログラムにズームします。
2. ［データビュー］エディタを選択します。
3. #15［ リンク終了］コマンド行にパークし、一行作成します。
4. 次の変数項目を定義しましょう。

項目タイプ	項目番号	項目名	モデル	型	書式	範囲	終了	代入
V= 変数	1	顧客登録あり	0	L= 論理	5	0	0	0

［戻り値］特性の設定

1. #8［照会リンク］コマンド行にパークし、特性シートを開きます。
2. ［戻り値］特性からズームして項目一覧を開きます。
3. ＜顧客登録あり＞を選択します。

4. ［選択］ボタンをクリックします。



リンクに失敗したときの警告メッセージ

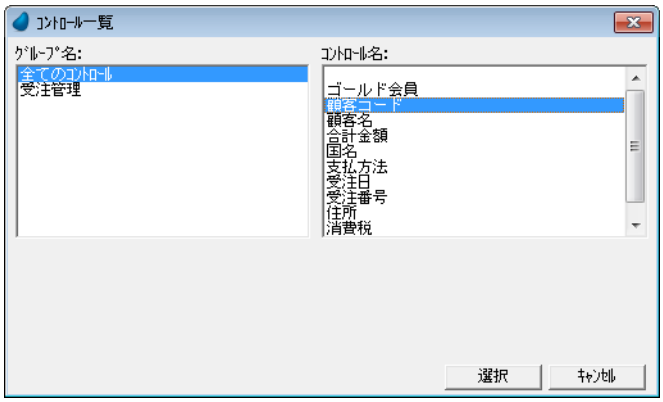
1. ［ロジック］エディタを選択します。
2. ［編集］メニューから［ヘッダ行作成］（Ctrl+H）を選択し、ヘッダ行を作成します。
3. 次のパラメータを設定します。

ロジックユニットタイプ	レベル	項目	条件
C= コントロール	S= 後	顧客コード	Yes

項目コントロールは、次のように選択します。

4. ［コントロール］欄からズームして［コントロール一覧］を表示します。
5. ＜顧客コード＞にパークします。

6. ［選択］ボタンをクリックします。



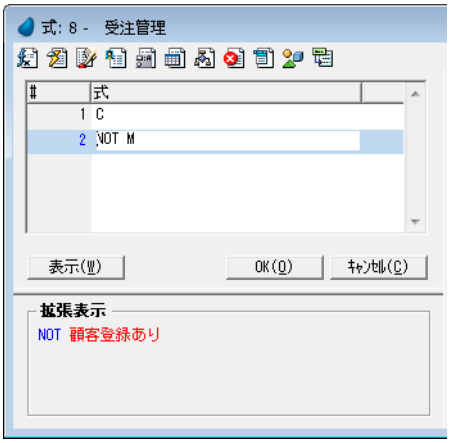
7. ［コントロール後］ロジックユニット行（#1 ）にパークした状態で F4 キーを押下して、一行作成します。
8. 次のようにパラメータを作成します。

コマンドタイプ	モード	式	テキスト	表示	条件
E= エラー	E= エラー	0	顧客が未登録です！	B= ボックス	2

［条件］欄の式設定について

a. ［条件］欄からズームして［式］エディタ［リンク］コマンドを開きます。
b. 最後の行（この場合は #1）にパークします。
c. 一行作成し、＜ NOT ＞と入力します。

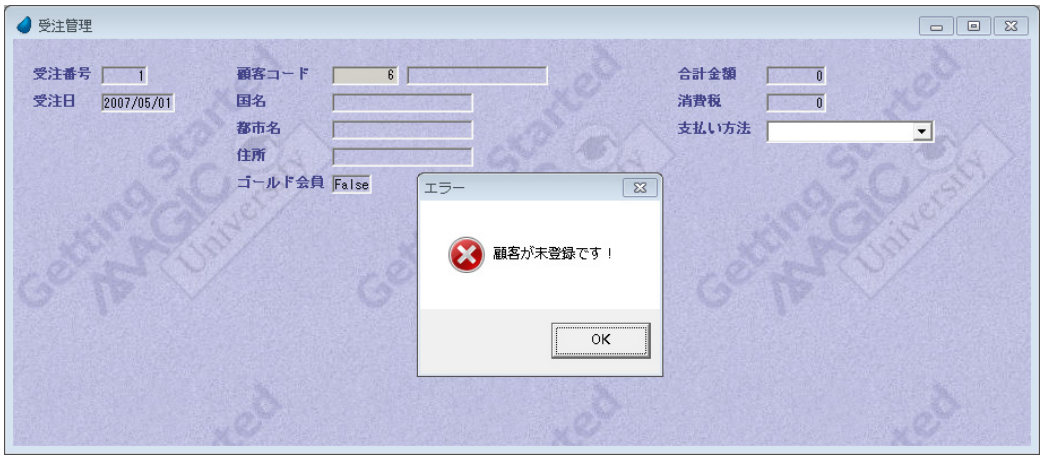
d. 半角スペースを一つ入力します。
e. ズームして項目一覧を表示します。
f. 項目＜顧客登録あり＞を選択します。
g. ［OK］ボタンをクリックします。
9. 変更内容を保存し、プログラムを閉じます。



プログラムの実行

1. ＜受注管理＞プログラムを実行（F7）してください。
2. ＜顧客コード＞を＜ 5 ＞に変更してください。
3. Tab キーを押下すると、エラーメッセージが表示されます。
4. ［OK］ボタンをクリックします。

プログラムは＜顧客コード＞に登録済みのコードを入力しないと終了できません。（エラーモードのため）



5. ＜顧客コード＞に ＜ 2 ＞ を入力します。
6. Esc キーを押下してプログラムを終了します。

ここまでのまとめ

［リンク］コマンドを利用して、データの存在の有無を判定できることが分かりました。このために［戻り値］特性を利用しました。

まず、論理型の変数項目を定義し、この変数項目を［リンク］コマンドの［戻り値］特性に設定しました。

次に、コントロール＜顧客コード＞に対する［コントロール後］ロジックユニットを定義し、［リンク］コマンドが失敗したときに警告表示を行わせるようにしました。

プログラム実行では、まだ未登録の顧客コードを入力したとき、［顧客が未登録です！］というエラーを表示し、登録されているコードの入力を促しました。

８．練習問題

この練習問題では、まず最初に＜商品＞データソースを定義します。

次に商品データを表示させるための＜商品照会＞プログラムを作成し､［リンク］コマンドを利用して＜取引先＞の詳細データをデータビューに読込み、表示させるようにします。

その際、取引先が存在するかどうかの確認を［リンク］コマンドの［戻り値］特性を使用して行なうようにします。

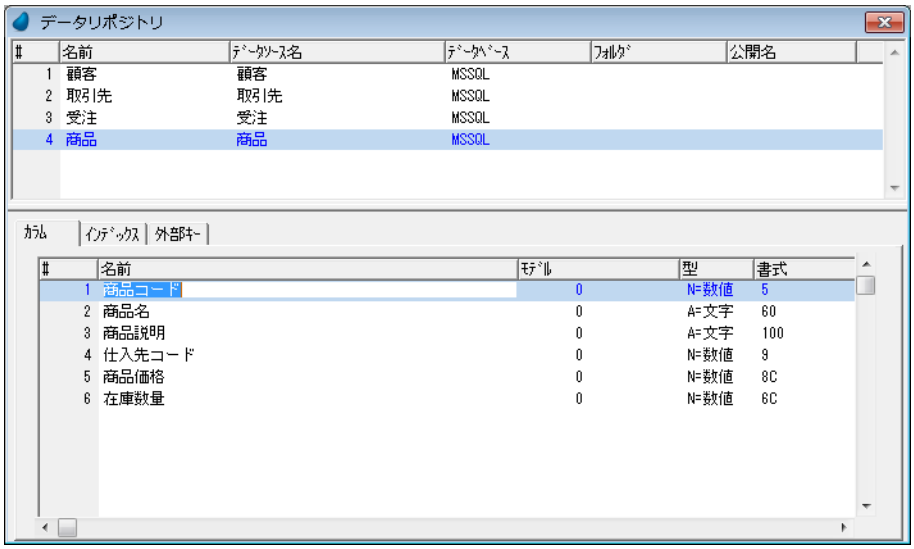
＜商品＞データソースの定義

1. ［データ］リポジトリに＜商品＞データソースを次のように定義してください。

#	名前	データソース名	データベース	フォルダ	公開名
1	商品	商品	MSSQL		

2. カラムを定義してください。

#	名前	モデル	型	書式
1	商品コード	0	N= 数値	5
2	商品名	0	A= 文字	60
3	商品説明	0	A= 文字	100
4	仕入先コード	0	N= 数値	9
5	商品価格	0	N= 数値	8C
6	在庫数量	0	N= 数値	6C

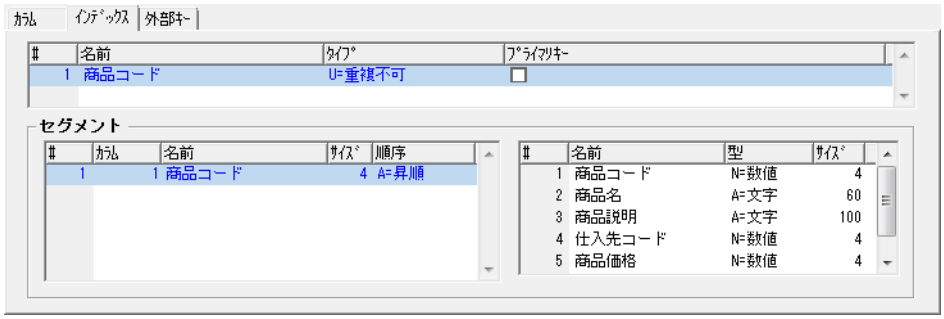


3. インデックスを定義してください。

#	名前	タイプ	プライマリキー
1	商品コード	U= 重複不可	

4. インデックスセグメントを次のように定義してください。

#	カラム	名前	サイズ	順序
1	1	商品コード	4	A= 昇順

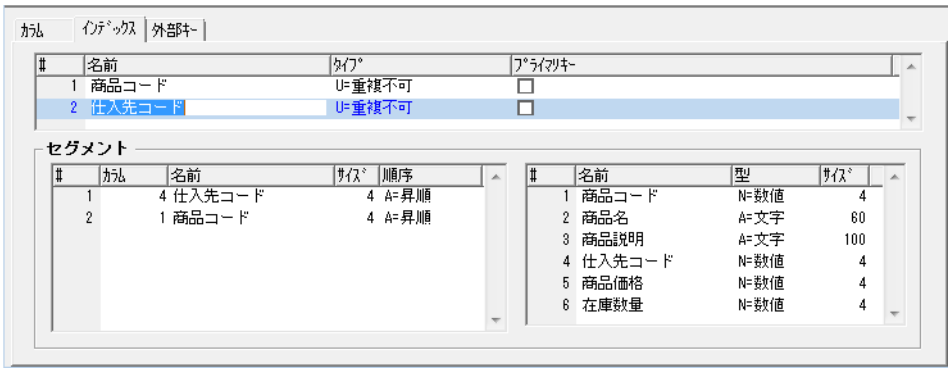


5. ［インデックス］テーブルに一行追加してください。
6. 次のインデックスを定義してください。

#	名前	タイプ	プライマリキー
2	仕入先コード	U= 重複不可	

7. インデックスセグメントを次のように定義してください。

#	カラム	名前	サイズ	順序
1	4	仕入先コード	4	A= 昇順
2	1	商品コード	4	A= 昇順

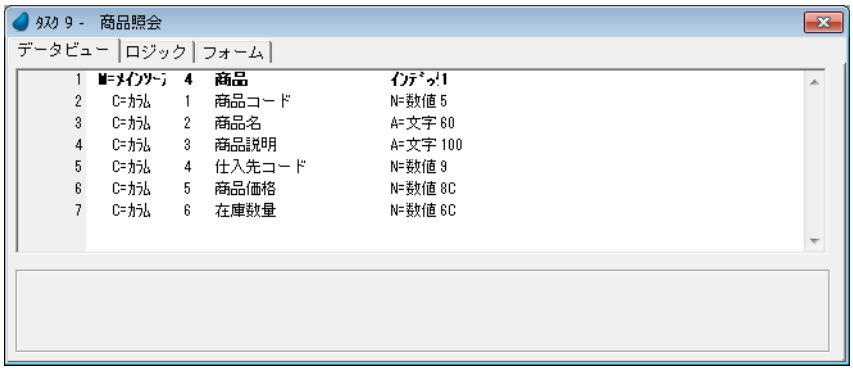


＜商品照会＞プログラムの作成

以下の手順にしたがって、＜商品照会＞プログラムを作成してください。

1. ［プログラム］リポジトリを開きます。
2. 最終行に新しくプログラムを作成します。
3. プログラム名を**＜商品照会＞**としてください。

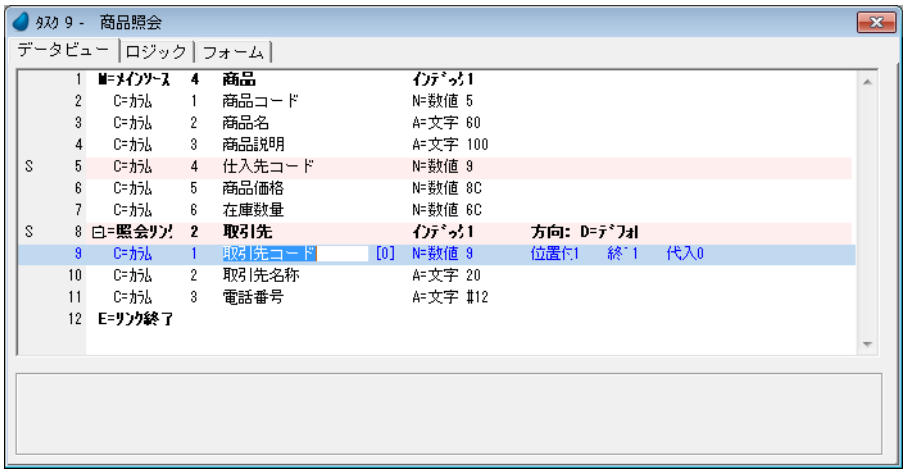
4. ズームしてプログラムを開きます。
5. タスク特性はデフォルトのままで［OK］ボタンをクリックします。
6. ［データビュー］エディタを開きます。
7. ＜商品＞データソースをメインソースとして定義してください。
8. ＜商品＞データソースのすべてのカラムを選択してください。



［リンク］コマンドの定義

1. ＜取引先＞データソースへの［照会リンク］［リンク］コマンドコマンドを定義してください。インデックスは＜1＞とします。
2. 次のカラムを選択します。＜取引先コード＞、＜取引先名称＞、＜電話番号＞

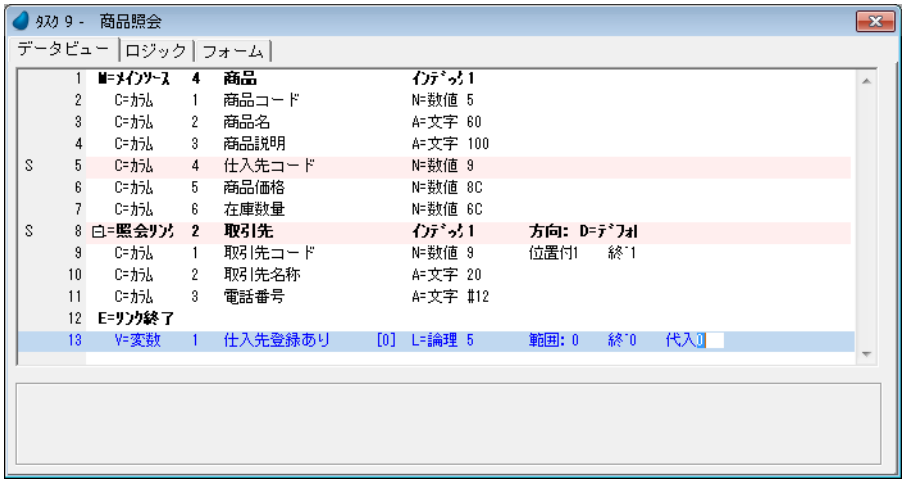
3. ＜商品＞データソースの＜仕入先コード＞を、＜取引先コード＞の［位置付］式と［終了］式に設定します。



戻り値用変数＜仕入先登録あり＞の定義

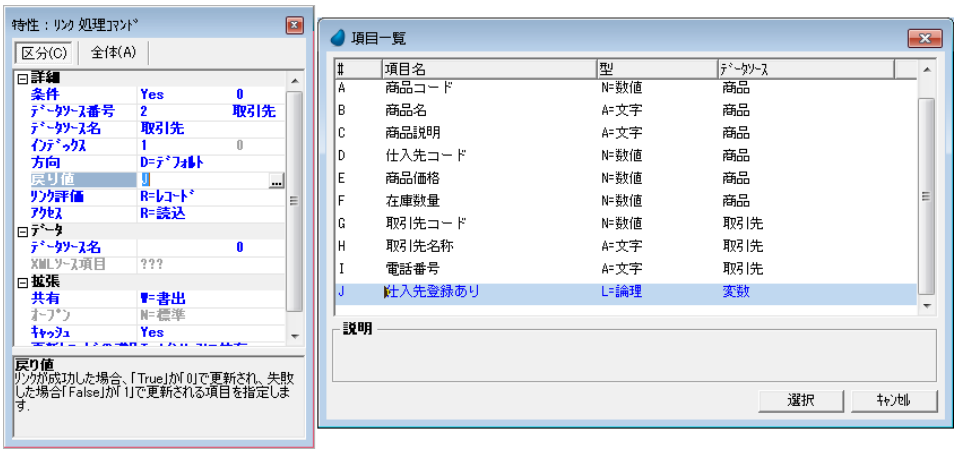
［データビュー］エディタで、［リンク］コマンドの戻り値用変数として、＜仕入先登録あり＞を定義してください。

項目タイプ	項目番号	項目名	モデル	型	書式	範囲	終了	代入
V= 変数	1	仕入先登録あり	0	L= 論理	5	0	0	0



リンク特性の［戻り値］設定

［リンク］コマンドの［戻り値］特性に、変数＜仕入先登録あり＞を定義してください。

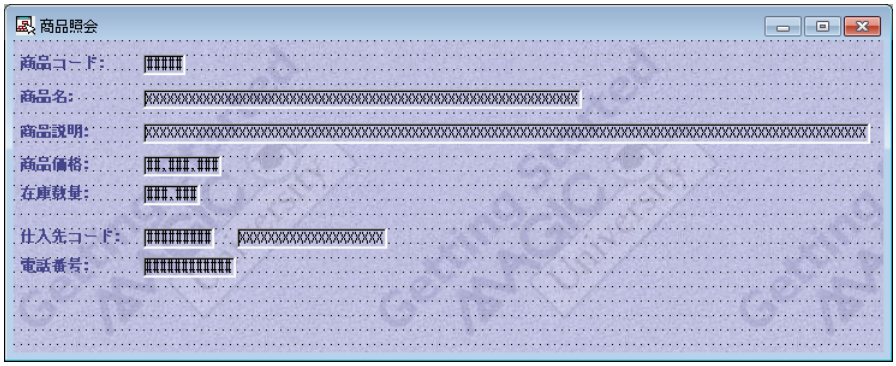


＜商品照会＞フォームの作成

［フォーム］エディタを選択し、以下のようにフォームを作成してください。

1. フォームに背景を設定してください。（**＜％背景％＞**を定義しましょう。）
2. ＜商品＞データソースのすべてのカラムをフォームに配置します。また＜取引先＞データソースの＜取引先名称＞と＜電話番号＞も配置します。それぞれの位置は、下のイメージを参考にしてください。

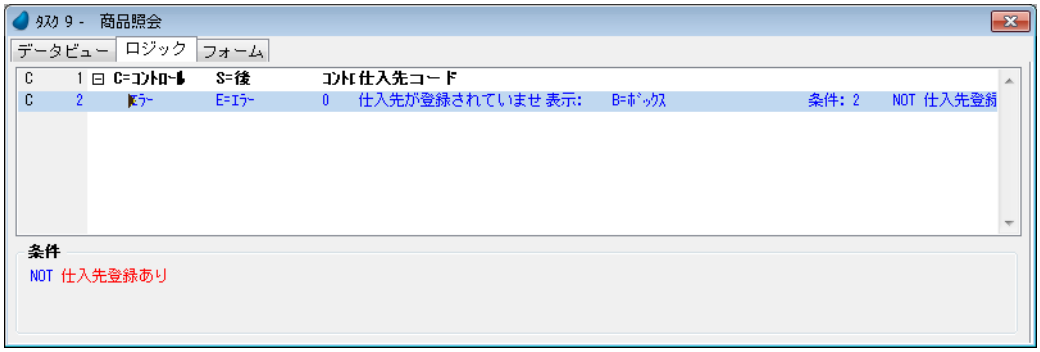
3. ［ラベル］コントロールのフォントをすべて＜見出し文字＞（#11）にします。
4. ［ラベル］コントロールの基本色をすべて＜見出し文字＞（#5）にします。
5. ［ラベル］コントロールのサイズ調整をして、すべての文字が読み取れるようにします。
6. ［エディット］コントロールの位置を調整して、［ラベル］コントロールと重ならないようにします。
7. ［エディット］コントロールのフォントをすべて＜文字＞（#12）にします。
8. ［エディット］コントロールの基本色をすべて＜文字＞（#6）にします。



データの存在確認ロジック作成

1. ［ロジック］エディタで、＜仕入先コード＞コントロールの［コントロール後］ロジックユニットを定義してください。
2. リンクが失敗したときに、［エラー］処理コマンドによって、エラーメッセージを表示させるようにしてください。

3. 変更内容を保存し、プログラムを閉じてください。



プログラムの実行

1. F7 キーを押下して＜商品照会＞プログラムを実行させてください。
2. タスクの初期モードは［修正モード］ですが、最初は商品データが存在しないので、［登録モード］でプログラムが動作しています。

3. たとえば、次のようにデータを登録してゆきましょう。（必ずしも実際の商品ではありません）

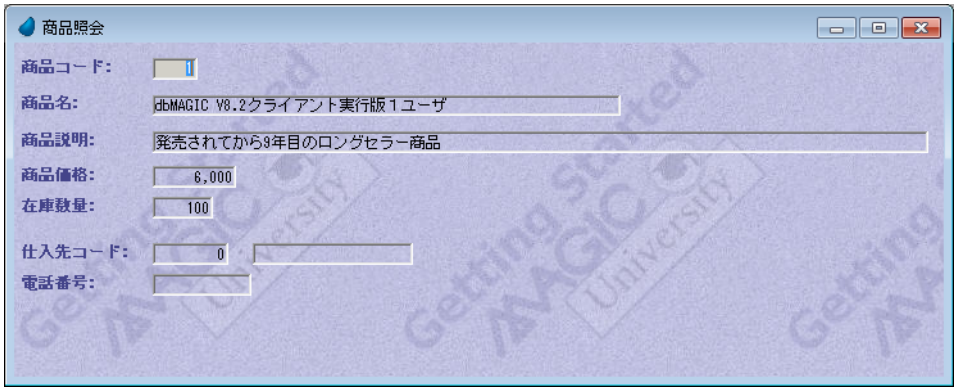
商品コード	商品名	商品説明	商品価格	在庫数量	仕入先コード
1	dbMAGIC V8.2 クライアント実行版 1 ユーザ	発売されてから 9 年目のロングセラー商品	60,000	1,000	1
2	Magic eDeveloper V9 開発版	2002 年発売開始しました	600,000	100	1
3	Magic eDeveloper V9 Plus	2004 年発売開始しました	600,000	500	1
4	uniPaaS Studio V10	2006 年末デビューしました	600,000	1,000	2
5	Magic jBOLT V2.5	SAP Certified Integration	2,000,000	20	3

4. Page Down/Page Up キーを押下して、レコードを移動してみましょう。

このとき、＜仕入先名＞と＜電話番号＞が一緒に変更されることに留意してください。

＜仕入先コード＞を変更して、［リンク式の再計算］が行われることを確認してみましょう。

＜仕入先コード＞を存在しない値（この場合は 4 以上の値）にして、エラーとなることを確認してください。



９．要約

本章では、［リンク］コマンドについて学び、１対１のデータリレーションの実習プログラムを作成しました。

このタイプのリレーションは、プログラムのメインソースのレコードと、他のデータソースの特定のデータとの関連付けを行うために使用されます。

［リンク］コマンドは、次のような目的のために使用されます。

- リンク先データソースから項目を読込むことにより、データビューを拡張する。
- 入力されたデータが他のデータソースですでに登録されているものかどうかの確認と、リンク先データソースでの特定データの存在確認を行う。
- リンク条件に従い、リンク先データソースもレコードの修正または登録を行う。

次のような［リンク］コマンドの種類について学習しました。

- 照会リンク
- 書出リンク
- 登録リンク
- 結合リンク
- 外部リンク

本章では照会リンクを使用して受注管理プログラムのデータビューを拡張し、戻り値により＜顧客コード＞の登録済確認を行いました。

練習問題では＜商品照会＞プログラムを作成し、商品の詳細データと、［リンク］コマンドを使用することで、仕入先データの一部を取得し、表示するようにしました。また戻り値を利用して、＜仕入先コード＞がすでに登録済かどうかの確認処理を追加しました。

第 16 章 一覧からのデータ選択

本章では、選択肢の一覧からエンドユーザが値を選択できるようにする、選択テーブルプログラムの作成と利用方法について学習します。

またデータコントロールについて学びますが、これは［コンボボックス］コントロールを利用して、データソースの値から選択できるようにするものです。

キーワード

- ［選択テーブル］特性
- 照会モード
- パラメータ
- ［位置付：最小］特性
- プッシュボタン
- ［選択］イベント
- ［イベン］トロジックユニット
- ［コール］処理プログラム
- 引数
- データコントロール

学習目標

本章を学ぶことで、次の内容に対する理解を深めましょう。

- 選択テーブルプログラムの作成方法
- 選択テーブルプログラムでの Magic 実行エンジンの動作
- コールされたプログラム側でのパラメータ受け取り
- 呼出し元プログラムへのパラメータの戻り値設定
- プログラムコール
- プログラムへの引数の渡し方
- データコントロールの定義

参照

Magic ヘルプの［Magic xpa リファレンス／プログラム／タスク特性］

同様に、［Magic xpa リファレンス／表示フォームとコントロール／表示コントロール］

１．はじめに

前章で作成した＜受注管理＞プログラムでは、エンドユーザはまず＜顧客コード＞をタイプ入力しなければならず、その後はじめて顧客詳細データが表示されるようになっていました。

しかし、顧客の名前が分からないままで＜顧客コード＞を入力させるというのは、実際のプログラムとしては良い仕様とは言えません。エンドユーザが実際に入力するためには、コード（数字）だけではなく、もっと具体的なデータを見る必要があります。

多くの場合、＜顧客コード＞自体には特別な意味は何もなく、＜顧客名＞の方にこそ意味があります。

また、コードと顧客との対応関係をエンドユーザがすべて記憶するということには無理があります。

このような場合のソリューションとしては、エンドユーザに候補一覧から値を選択させるという方法が適当と考えられます。

利用可能な情報（たとえば＜顧客コード＞と＜顧客名＞の一覧）のすべての組合わせを見ることができるようになれば、エンドユーザは入力値が適正なものかどうか確信がもてないというストレスにさらされることなく、確実に存在する候補から値を選択することができるようになります。

Magic xpa ではそのような［選択テーブルプログラム］を簡単に作成するための方法があり、本章ではそれについて学習します。

［選択テーブルプログラム］は、一覧からデータを選択するための包括的なソリューションですが、場合によっては、特に一覧となるデータの数が少ない場合などに、データコントロールを使用することがより効率的なソリューションになります。

データコントロールはコンボボックスを使用し、１つのデータソースから、あるカラムの内容を表示させ、選択させるようにするものです。

データコントロールの制限事項としては、データの数が少ないものであること、およびデータソースの１カラムだけしか表示できないことです。

本章では、［選択テーブルプログラム］と［データコントロール］について実習を含めて説明し、データコントロールを使用すると良い場合について検討します。

２．選択テーブルプログラム

［選択テーブルプログラム］はリッチクライアント形式のプログラムで、データを一覧表形式で表示し、エンドユーザが値を選択できるようにしたものです。

［選択テーブルプログラム］を使用するときは、次の二点に留意してください。

- まず［選択テーブルプログラム］を作成する。
- 作成した［選択テーブルプログラム］をホストプログラムから［コール］処理コマンドを使用して呼出す。

顧客選択テーブルプログラムの作成

このセクションでは、＜顧客＞データソースの内容（＜顧客コード＞と＜顧客名＞）を一覧表示する選択テーブルプログラムを作成します。

1. ［プログラム］リポジトリ（Shift+F3）を開きます。
2. 最後の行にパークします。
3. 一行作成（F4）します。
4. ［名前］カラムに**＜顧客選択＞**と入力してください。
5. ＜顧客選択＞プログラムにズーム（F5）します。

［タスク特性］ダイアログが開いたら、次の操作をしてください。

6. ［タスクタイプ］特性を「C= リッチクライアント」としてください。
7. ［初期モード］特性を［Q= 照会］としてください。
8. ［選択テーブル］特性を［Yes］としてください。
9. ［OK］ボタンをクリックします。

［初期モード］特性を［Q= 照会］に設定することの意味

［選択テーブルプログラム］はデータの一覧から値を選択する目的で使用します。つまり、データを修正したり、新規に登録したりすることは行いません。したがって、タスクの［初期モード］特性は［Q= 照会］（エンドユーザはデータの更新、作成ができません）にしておくのがよいのです。

また、照会モードにすることによってインクリメンタルサーチ機能が使えるようになるという利点があります。これは、参照したいデータの値を、そのカラムに入力してゆくことで自動的に位置付ける機能です。

プログラム内でエンドユーザがデータの更新・登録・削除などを行えなくする方法はこの他にもありますが、ここではこれ以上の説明は略します。

［選択テーブル］特性を［Yes］にすることの意味

［選択テーブル］特性を［Yes］にすることにより、Magic xpa はエンドユーザが［Enter］キーを入力すると、［レコード後］ロジックユニットを実行し、直ちにプログラムの実行を終了するようになります。

この場合の Magic エンジンの動きは次のようになります。

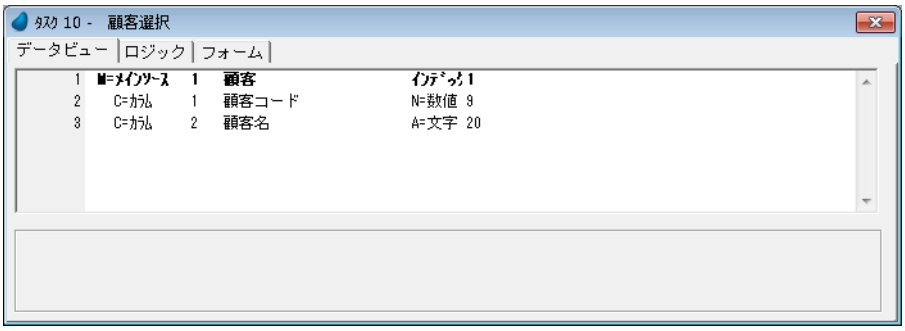
a. ［レコード後］ロジックユニットを実行します（このロジックユニット内で、選択された値をパラメータに戻り値として設定します）。
b. ［選択テーブルプログラム］を終了します。
c. 呼出したプログラムに制御を戻します。

データビューの定義

1. ［データビュー］エディタを開きます。
2. メインソースとして、#1 ＜顧客＞を選択し、インデックスを＜ 1 ＞ とします。
3. 一行作成（F4）します。

4. ［カラム番号］欄からズーム（F5）して、［カラム選択］ウィンドウを開きます。

5. 最初の２つのカラム（＜顧客コード＞と＜顧客名＞）を選択します。



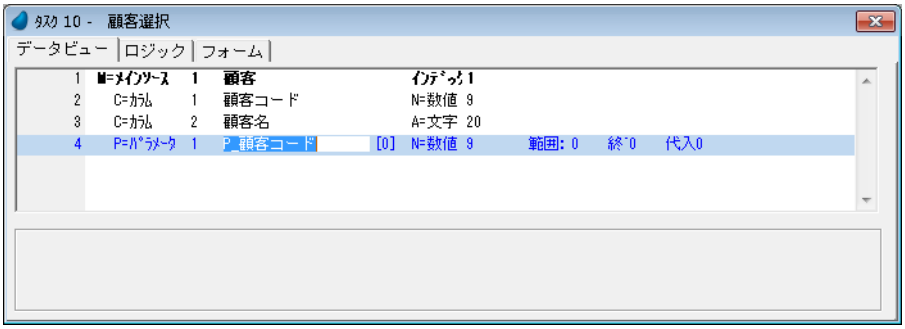
プログラム［パラメータ］の設定

プログラムの［パラメータ］はローカル変数であり、呼出し元のプログラムから渡された値を一時的に保持するものです。

このプログラム例では、＜顧客コード＞に対する［パラメータ］を設定します。この［パラメータ］は、渡された値を用いて＜顧客＞データに位置付けるだけでなく、より重要なこととして、選択された＜顧客コード＞を、呼出し元プログラムに返す役目があります。

1. 最後の行（#3 ＜顧客名＞の行）にパークします。
2. 一行作成（F4）します。
3. パラメータを定義し、各設定を次のようにします。

項目タイプ	番号	名前	モデル	型	書式	範囲	終了	代入
P= パラメータ	1	P_ 顧客コード	0	N= 数値	9	0	0	0



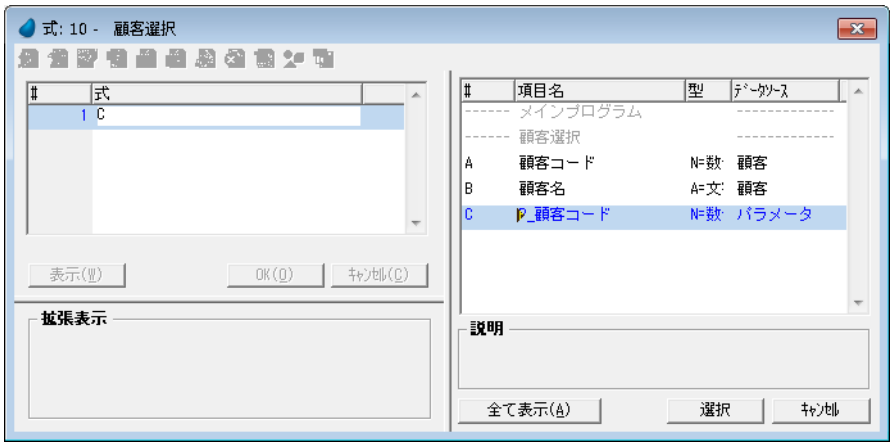
・パラメータの項目名は任意ですが、例のように頭に＜ P_ ＞などをつけることで、分かり易くなります。
・パラメータの型と書式は、渡される項目（この例の場合、＜顧客コード＞）と同一でなければなりません。
この型と書式管理を容易にするための機能として［モデル］がありますが、これについては第 21 章で扱います。

［位置付］条件の設定

［位置付］条件を設定することにより、プログラムが起動したときに特定のレコードを検索し、一致するレコードにカーソルをパークさせることができます。このプログラム例では、＜ P_ 顧客コード＞を＜顧客コード＞の［位置付］条件として設定することにより、呼出し元のプログラムが＜ P_ 顧客コード＞に値を渡してきたとき、選択テーブルが表示されると同時に、対応する顧客レコードにカーソルがパークするようになります。

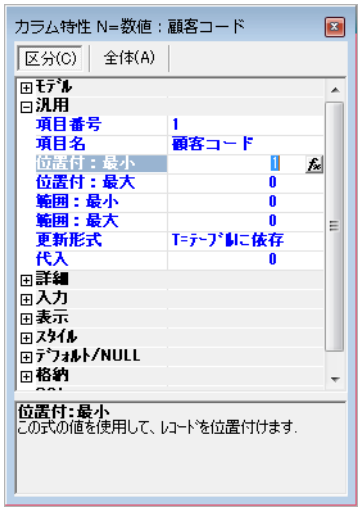
1. #2 の＜顧客コード＞カラム行にパークします。
2. ［カラム］特性（Alt+Enter）を開きます。
3. ［位置付：最小］特性からズーム（F5）するか、［fx］ボタンをクリックし、［式］エディタを開きます。
4. ［式］エディタで一行作成（F4）します。
5. ズーム（F5）して、項目リストに移動します。

6. 項目リストから＜ P_ 顧客コード＞を選択します。



7. ［OK］ボタンをクリックします。

ここまでの作業で、式番号 1 が［位置付 : 最小］特性に表示されるようになりました。



選択された値を戻り値として設定

すでに説明したように、［タスク特性］ダイアログの［選択テーブル］特性を［Yes］に設定した場合、Magic エンジンはエンドユーザが値を選択して Enter キーを押下すると、直ちに［レコード後］ロジックユニットの処理を行います。

このことを利用して、パラメータ＜ P_ 顧客コード＞を、エンドユーザの選択した＜顧客コード＞の値で更新しましょう。

パラメータの値は更新の有無に関わらず、呼出し元のプログラムに返されます。

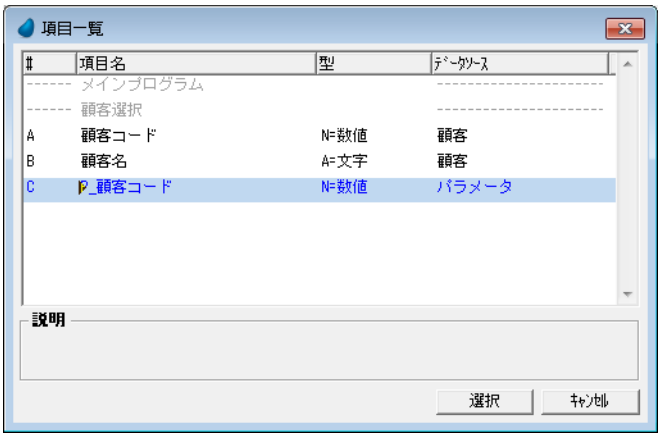
1. ［ロジック］エディタを開きます。
2. ヘッダ行（Ctrl+H）を一行作成します。
3. ロジックユニットタイプとして、＜ R= レコード＞を選択します。

4. レベルとして、［S= 後］を選択します。
5. 一行作成（F4）します。
6. 次のようにコマンドパラメータを設定します。

コマンドタイプ	項目	値	条件
項目更新	C　（P_ 顧客コード）	2　（顧客コード）	Yes

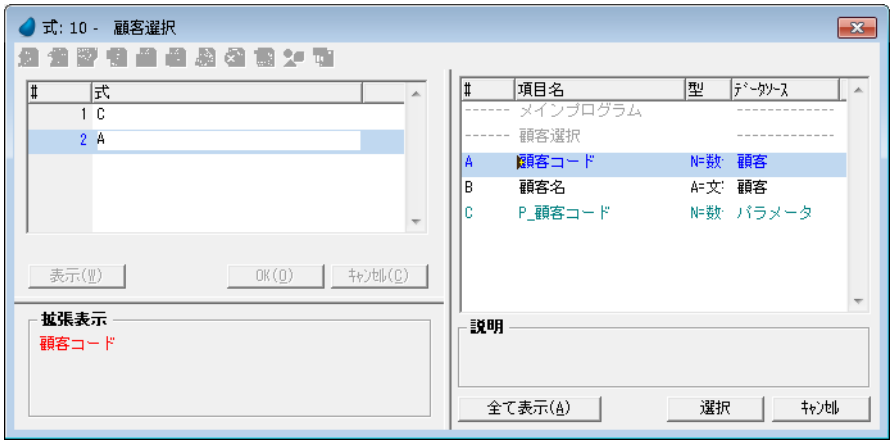
［項目］欄の設定

7. ［項目］欄からズームして［項目一覧］を開き、＜ P_ 顧客コード＞を選択します。

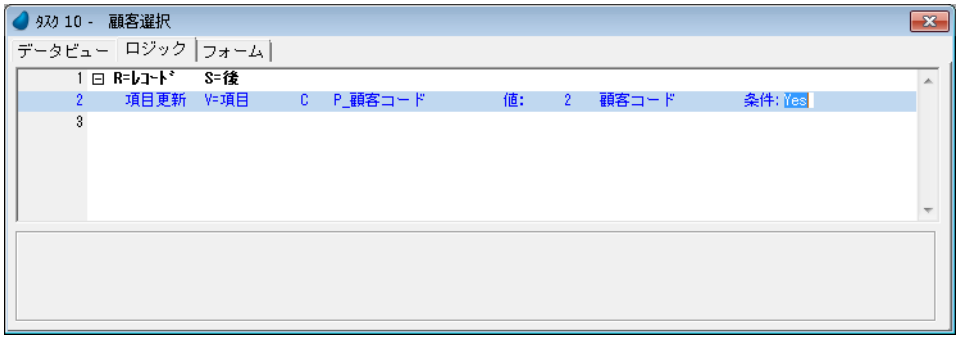


［値］欄の設定

8. ［式］欄からズームして［式］エディタを開きます。
9. 最終行にパークします。
10. 一行作成します。
11. ズームして項目リストへ移動し、＜顧客コード＞を選択します。



12. ［OK］ボタンをクリックします。



［選択テーブルプログラム］のフォーム

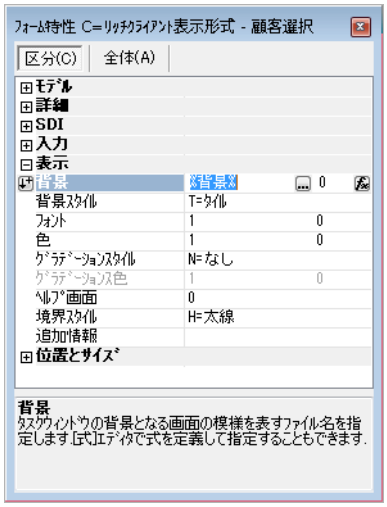
［選択テーブルプログラム］のフォームは単純です。

この例では、2 つのカラム（＜顧客コード＞と＜顧客名＞）からなるテーブルと、＜選択＞と＜終了＞の 2 つのプッシュボタンで構成されます。

ここでは上記の方針にしたがって、＜顧客選択＞フォームを作成してみましょう。

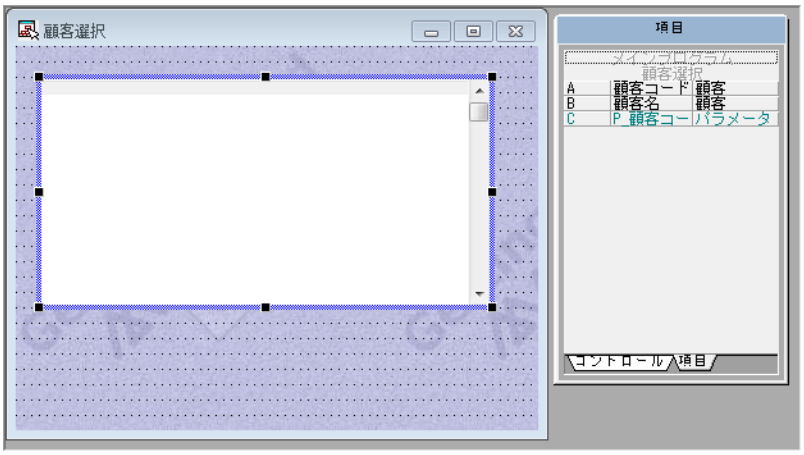
フォーム特性の設定

1. ［フォーム］エディタを開きます。
2. ＜顧客選択＞フォームにパークします。
3. フォーム特性（Alt+Enter）を開きます。
4. ［背景］特性に **＜％背景％＞** と入力してください。



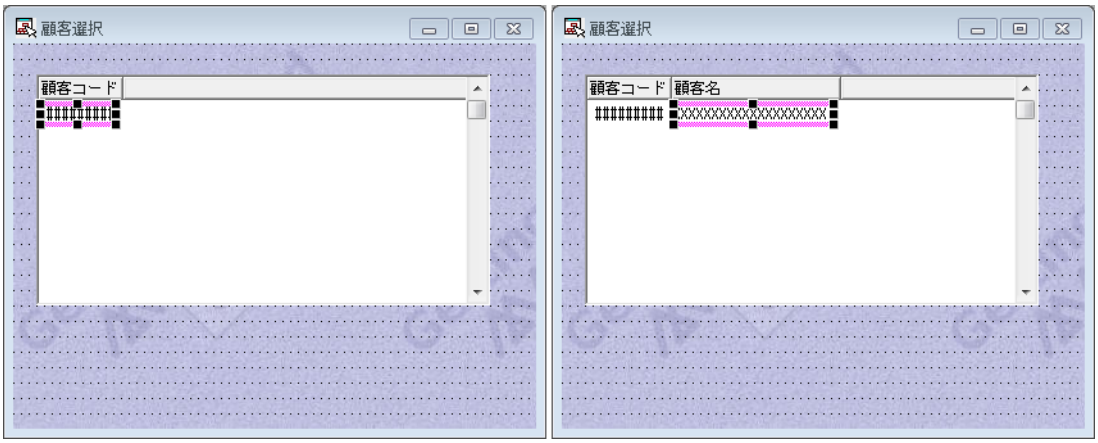
フォームのデザイン

1. ズームして＜顧客選択＞フォームを開きます。
2. ［コントロール］パレットで［テーブル］コントロールを選択し、フォームに配置します。
3. ［コントロール］パレットで［項目］タブをクリックし、項目テーブルを表示します。
4. ＜顧客コード＞（A）をクリックします。
5. カーソルをフォーム上の［テーブル］コントロールの白い領域内に持って行きます。このとき［テーブル］コントロールの境界線が緑色に変わります。



6. ［テーブル］コントロール上でクリックします。＜顧客コード＞カラムが［テーブル］コントロールに追加されます。

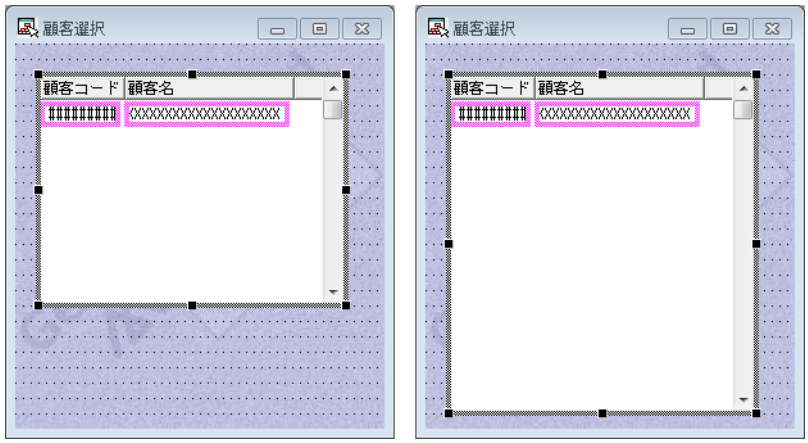
7. ４－６と同様の手順で、＜顧客名＞をテーブルに追加してください。



［テーブル］コントロールのサイズ調整

1. ［テーブル］コントロールの白い領域をクリックします。
2. ［テーブル］コントロールの下部にスライダ表示がある場合は、右中央部のハンドルをドラッグして、スライダ表示がなくなるまで右方向に拡張します。

3. Shift キーを押下しながら、↓キーを 1、2 回押下します。
4. Shift キーを離し、Enter キーを押下することで、サイズ調整を確定させます。



今までの操作において、［テーブル］コントロールのサイズ調整を 2 通りの方法で行いました。

- マウスを使用して行う方法 …… 上記２の方法
- キーボードを使用して行う方法 …… 上記３－４の方法

フォームのサイズ調整

上記と同様の方法でフォーのサイズを調整することができます。

フォームのサイズ調整を行う場合は、コントロールの選択がない状態で行います。

［選択］ボタンの追加

［選択］ボタンは必ずしも［選択テーブルプログラム］に必須なものではありませんが、ここでは追加することにします。

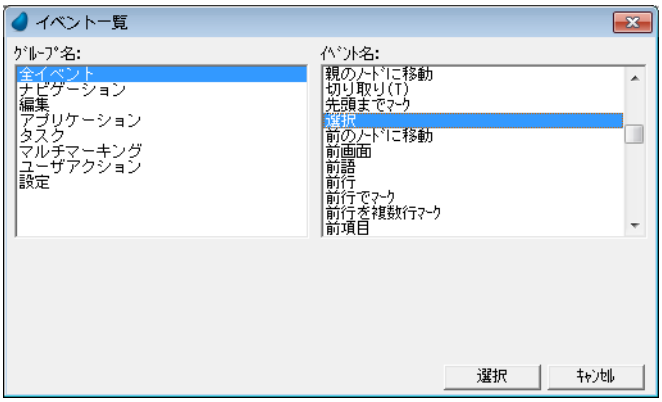
すでに何度も説明したように、［選択テーブルプログラム］の特徴は、エンドユーザが Enter キーを押下したときの処理手続きにあります。

Magic エンジンの内部では、エンドユーザが Enter キーを押下することにより、［選択］イベントが発動されます。

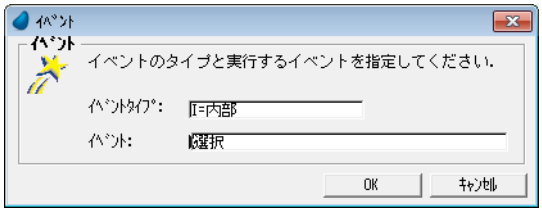
このことを踏まえて、プッシュボタンに［選択］イベントを発動するように定義することで、同様の処理を行わせることができます。

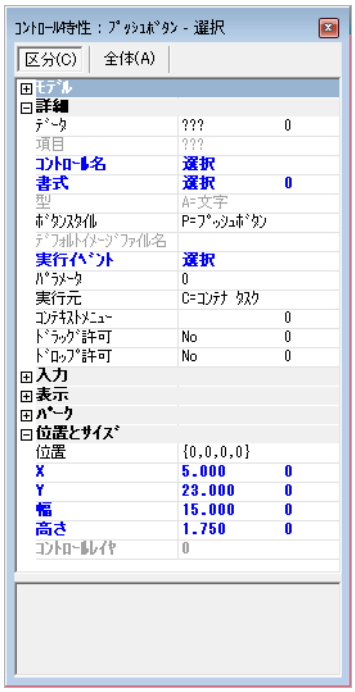
1. ［コントロール］パレットで［プッシュボタン］を選択し、フォームに配置します。
2. ［プッシュボタン］コントロールの［特性］シート（Alt+Enter）を開きます。
3. ［コントロール名］特性に＜選択＞と入力します。

4. ［書式］特性にも同様に、＜選択＞と入力します。
5. ［実行イベント］特性からズームして［イベント］ダイアログを開きます。
   a. ［イベントタイプ］欄として、［I= 内部］を選択します。
   b. ［イベント］欄からズームして、イベント一覧を表示し、［選択］イベントを選択します。
   c. ［選択］をクリックします。



   d. ［OK］ボタンをクリックします。



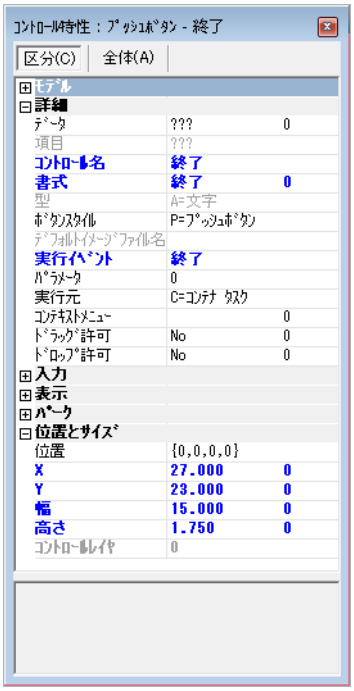


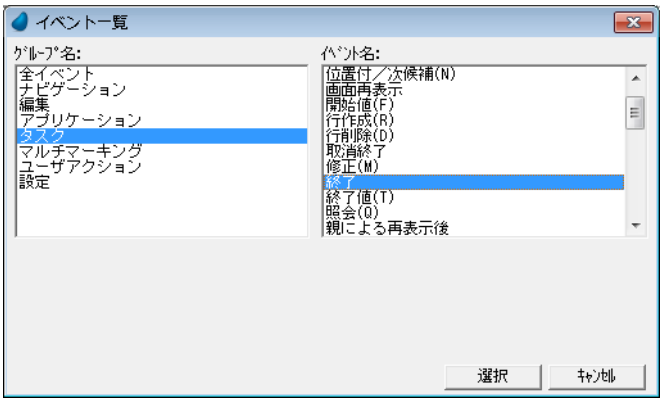
6. ［幅］特性を＜ 15.0 ＞とします。

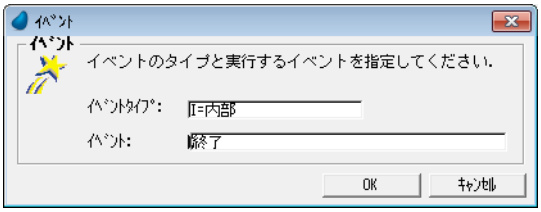
- プッシュボタンに項目が割当てられない場合、［書式］特性に入力された内容が、プッシュボタン上に表示されます。
- ［書式］特性に英字を入力する場合、最初に '&' を付けると、'&' の次の英字と 'Alt' キーを同時に押下することでショートカットキーになります。

［終了］ボタンの追加

1. ［コントロール］パレットで、［プッシュボタン］を選択し、フォームに配置します。
2. ［プッシュボタン］コントロールの［特性］シート（Alt+Enter）を開きます。
3. ［コントロール名］特性に＜終了＞と入力します。
4. ［書式］特性にも同様に＜終了＞と入力します。
5. ［実行イベント］特性からズームして［イベント］ダイアログを開きます。
   a. ［イベントタイプ］欄として、［I= 内部］を選択します。
   b. ［イベント］欄からズームして、イベント一覧を表示し、［終了］イベントを選択します。
   c. ［終了］をクリックします。
   d. ［OK］をクリックします。
6. ［幅］特性を＜ 15.0 ＞とします。
7. 変更内容を保存し、プログラムを閉じます。







ここまでの要約

今までの作業で＜顧客選択＞プログラムを作成しました。

このプログラムは＜顧客コード＞をパラメータとして呼出し元のプログラムから受け取ります。Magic エンジンは受け取ったパラメータにしたがって、一致するレコードを検索し位置付けます。

エンドユーザが値を選択すると、Magic エンジンが［レコード後］ロジックユニットを実行し、そこで＜顧客コード＞パラメータを選択された値に更新します。

［テーブル］コントロールをフォームに配置し、2 つのカラムすなわち＜顧客コード＞、＜顧客名＞を定義しました。また［テーブル］コントロールの幅や高さの 2 通りのサイズ調整方法、すなわちマウスを使用する場合と、キーボードを使用する場合について実習しました。

2 つのボタンをフォームに配置しました。

- ［選択］プッシュボタンには［選択］イベントを定義し、Enter キーを押下した場合と同じ選択動作をするようにしました。
- ［終了］プッシュボタンには［終了］イベントを定義し、値の選択なしにプログラムを終了するようにしました。

３．［選択テーブルプログラム］の呼び出し

［選択テーブルプログラム］を作成するだけではまだ十分ではありません。

次に行うのは、＜受注管理＞プログラムにイベントを設定することです。

そのイベントハンドラは＜顧客選択＞プログラムを呼び出し、受注データの＜顧客コード＞をパラメータとして渡します。

顧客選択イベントの作成

1. ［プログラム］リポジトリを開きます。
2. ＜受注管理＞プログラムを開きます。
3. ［タスク環境］メニューから、［ユーザイベント］（Ctrl+U）を選択します。

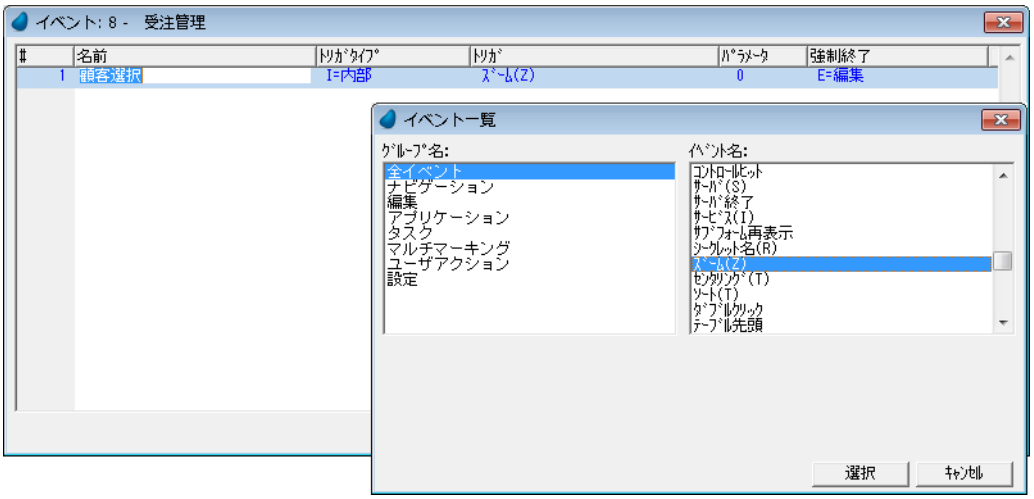
4. 一行作成（F4）します。
5. 次のように設定します。

#	名前	トリガタイプ	トリガ	パラメータ	強制終了
1	顧客選択	I= 内部	ズーム (Z)	0	E= 編集

トリガの設定

6. ［トリガ］欄からズームして［イベント一覧］を表示します。

7. ズーム（Z）を選択します。



［強制終了］欄を［E= 編集］とすることにより、選択テーブルプログラムが閉じて制御が戻ったときに、選択テーブルプログラムで選択された新しい＜顧客コード＞が表示されるようになります。

顧客選択イベントハンドラの作成

ここでは［イベント］ロジックユニットを定義し、顧客選択テーブルプログラムを呼び出して＜顧客コード＞を引数として渡すようにします。

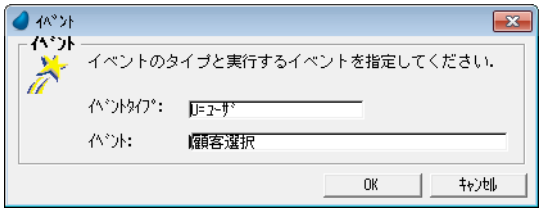
1. ［ロジック］エディタを選択します。
2. 最終行にパークします。
3. ヘッダ行（Ctrl+H）を一行作成します。

4. 次のように設定します。

#	ロジックユニット	イベント	コントロール	スコープ	条件
3	E= イベント	顧客選択（注 1）	顧客コード（注 2）	S= サブツリー	Yes

イベントの設定

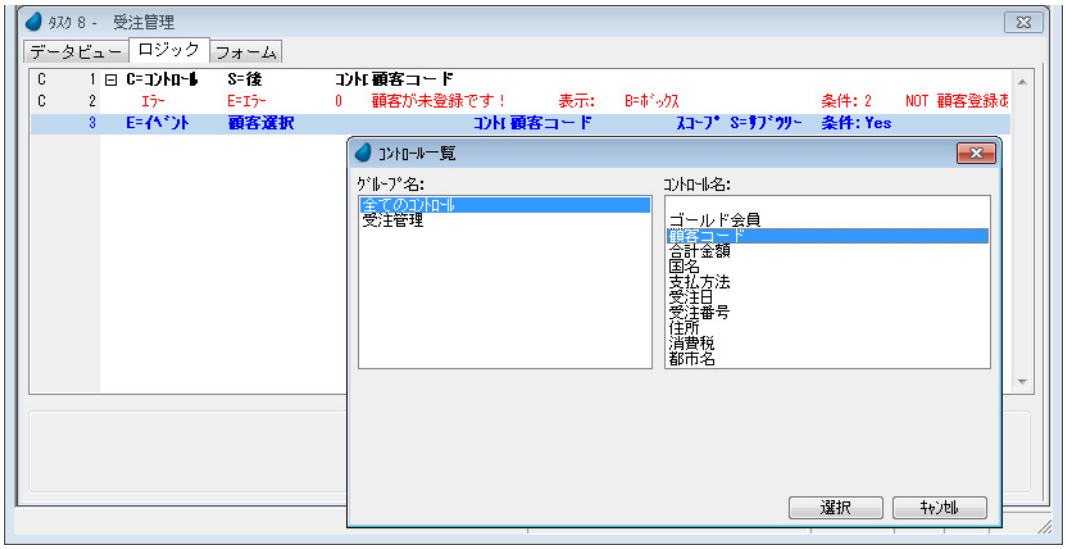
5. ズームして［イベント］ダイアログを開きます。
6. ［イベントタイプ］欄として［U= ユーザ］を選択します。
7. ［イベント］欄からズームして［イベント一覧］を表示します。
8. ＜顧客選択＞を選択します。
9. ［OK］ボタンをクリックします。



コントロールは次のようにして設定します。

10. ズームしてコントロール一覧を表示します。
11. ＜顧客コード＞コントロールを選択します。

12.［選択］をクリックします。



13.イベントヘッダ行（#3）にパークしている状態で、一行作成（F4）します。

14.次のように設定します。

#	コマンド	コールタイプ	プログラム番号	パラメータ	結果	条件
4	コール	P= プログラム	10　（顧客選択）	（顧客コード）		Yes

プログラム番号は次のように設定します。

15.プログラム一覧にズームします。

16.＜顧客選択＞プログラムを選択します。

［パラメータ］欄は次のように設定します。

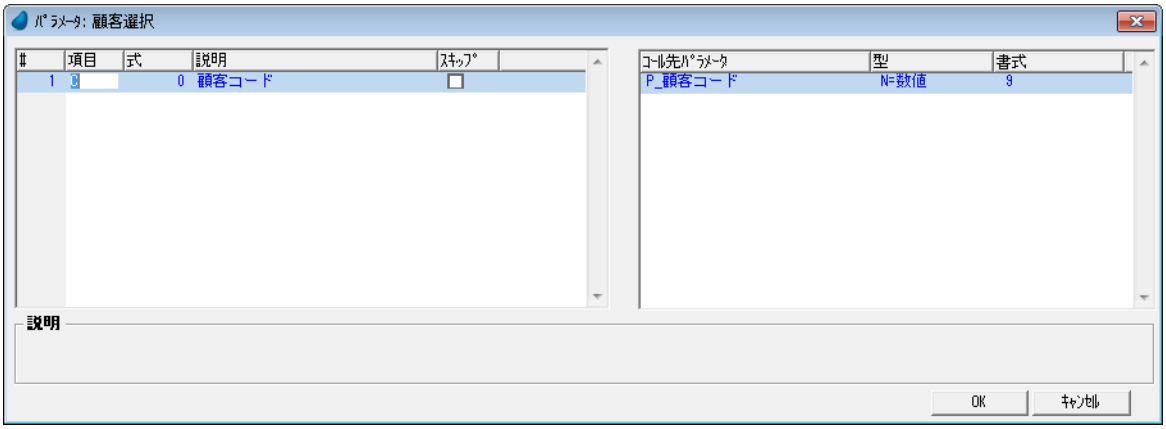
17.ズームして［パラメータ］テーブルを開きます。

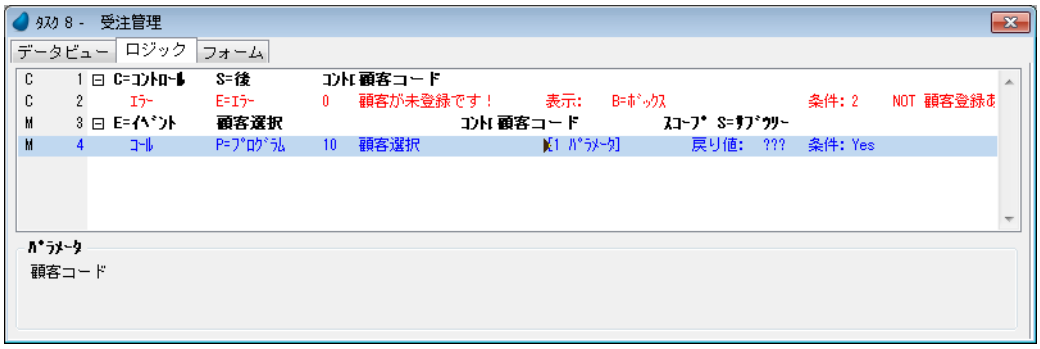
18.一行作成（F4）します。

19.［項目］欄からズームして項目一覧を開きます。

20.＜顧客コード＞（C）を選択します。

21.［OK］ボタンをクリックします。



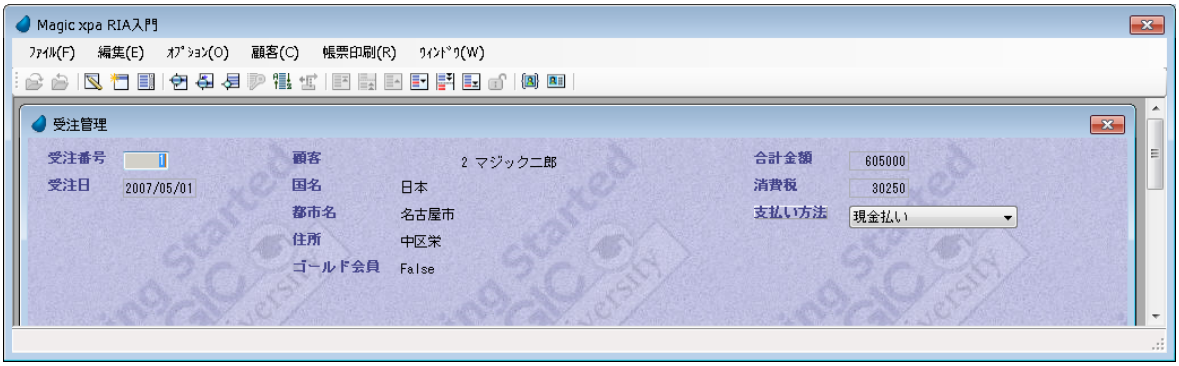


［選択テーブルプログラム］のテスト

ここでは＜受注管理＞プログラムを実行させ、作成した＜顧客選択＞プログラムを起動させてみましょう。

1. 変更内容を保存して、＜受注管理＞プログラムを閉じます。
2. F7 キーを押下して、＜受注管理＞プログラムを実行します。
3. Tab キーを押下して、＜顧客コード＞コントロールにパークします。

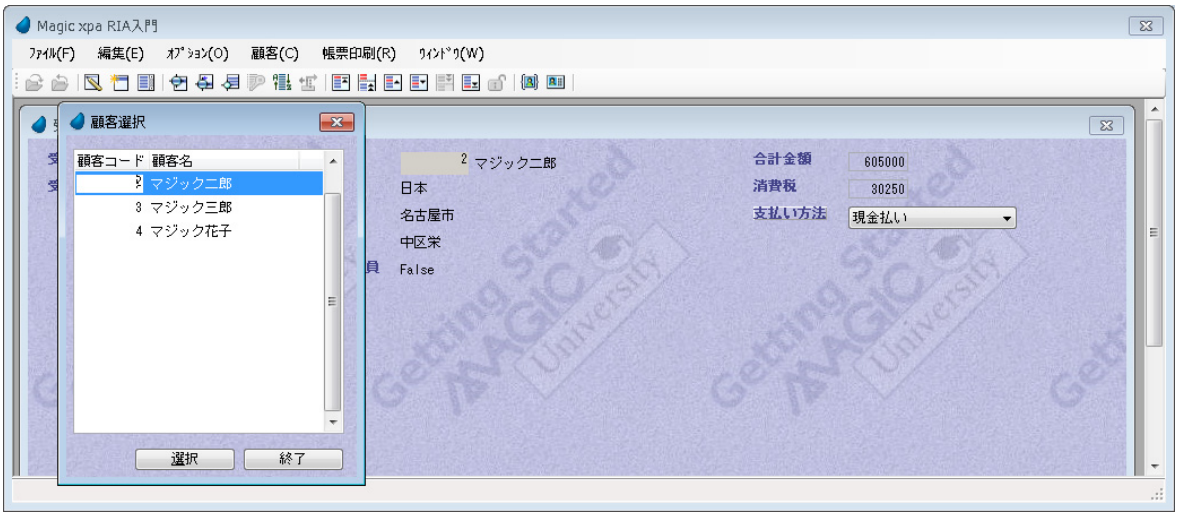
このプログラムは、＜顧客コード＞コントロールでズームすることができます。



ズームの方法には２つあります。

- **F5 キーを押下する。**
- **パークしている場所でダブルクリックする（この例では、＜顧客コード＞欄）**

4. ＜顧客コード＞からズームします。＜顧客選択＞ウィンドウが表示され、顧客一覧から選択できるようになります。



ズームしたとき、2 番目の顧客が選択されていることに留意してください。これは＜受注管理＞プログラムで＜顧客コード＞として＜ 2 ＞が設定されており、引数として渡されたからです。

このとき、垂直スクロールバーを動かせば、1 番の顧客コードのレコードも見ることができます。

5. ↓キーを押下して、顧客コード＜ 3 ＞の顧客にパークしてください。
6. Enter キーを押下して、顧客コード 3 を選択してみましょう。

［選択テーブルプログラム］から結果を得る方法は３つあります。
- **Enter キーを押下する。**
- **［選択］プッシュボタンをクリックする。**
- **選択したい行でダブルクリックする。**

＜顧客選択＞プログラムが閉じると、＜受注管理＞プログラム画面の＜顧客コード＞欄は＜ 3 ＞に更新され、他の詳細情報が読込まれていることが分かります。

次に、値の選択をせずに＜顧客選択＞プログラムを終了してみましょう。

1. ＜顧客コード＞欄からズームして＜顧客選択＞プログラムを起動します。
2. #4 の行にパークします。
3. Esc キーを押下または［終了］ボタンをクリックし、＜顧客選択＞プログラムを終了します。

＜受注管理＞プログラム画面の＜顧客コード＞は＜ 3 ＞のままで変化しないことが分かります。

4. ＜顧客コード＞の値に＜ 2 ＞を入力してください。
5. ＜受注管理＞プログラムを終了してください。（Esc キーを押下）

［選択テーブルプログラム］を選択せずに終了する方法は３つあります。
- **Esc キーを押下する。**
- **［終了］プッシュボタンをクリックする。**
- **ウィンドウ右上部角にある、システムの X アイコンをクリックする。**

４．データコントロールと選択テーブルプログラムの比較

選択テーブルプログラムと同じように、データコントロールを利用することでエンドユーザは候補一覧から選択できるようになります。

データコントロールは、データに関連づけられたチョイスコントロールであり、その選択肢となる値がデータソースから読込まれてくるもののことです。

Magic xpa はコントロール特性の設定にしたがって、実行時にコントロールの選択値一覧を作成します。

選択テーブルプログラムとは異なり、データコントロールはプログラム内部で定義されるものであり、個別に独立したものではありません。

データコントロールの特徴

- 別途［選択テーブルプログラム］を作成する必要がなく、またそのプログラムをコールする処理も必要がない。
- エンドユーザの選択値の妥当性を確認しなくてもよい。(既存のものから選択させるので自明)
- エンドユーザはキーボードから入力する代わりに、値を選択するだけでよい。
- プログラム内のコードを外部に表示させることなく、説明文や名称を使用して選択させることができる。
- 選択テーブルプログラムと異なり、ただ一つのカラムしか表示に用いることはできない。

データコントロールを使用できるのは、選択値となるデータ量がごく少ない場合に限定されます。

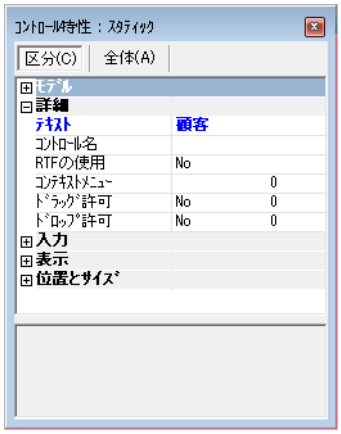
顧客データコントロールの定義例

データコントロールを利用して顧客を選択するのはよい仕様とは言えません。なぜなら顧客データソースは、通常、きわめて大きなものになるからです。

しかし本コースでは、<顧客>データソースは少しのデータしかもたないことと、実習であるという観点から、このデータソースをデータコントロールの適用対象として用いることにします。

このセクションでは、選択テーブルプログラムをデータコントロールで置き換えてみることにします。

<顧客コード>の見出しの変更



1. ［プログラム］リポジトリ（Shift+F3）を開きます。
2. <受注管理>プログラムにズームします。
3. ［フォーム］エディタを選択します。
4. <受注管理>フォームにズームします。
5. <顧客コード>［ラベル］コントロールを選択します。
6. コントロール特性を開きます。
7. ［テキスト］特性にパークし、**<顧客>**に変更します。

コントロールの削除

8. <顧客コード>［エディット］コントロールを選択します。
9. ［編集］メニューから［削除］を選択し、コントロールを削除します。
10. <顧客名>［エディット］コントロールを選択します。
11. ［編集］メニューから［削除］を選択し、コントロールを削除します。

データコントロールの追加

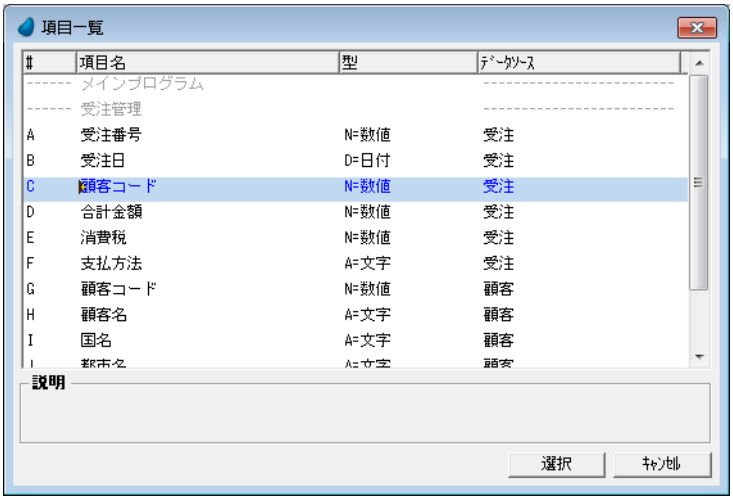
12. ［コントロール］パレットで、［コンボボックス］コントロールをクリックします。

13.［コンボボックス］コントロールを＜顧客＞［ラベル］コントロールの右に配置します。

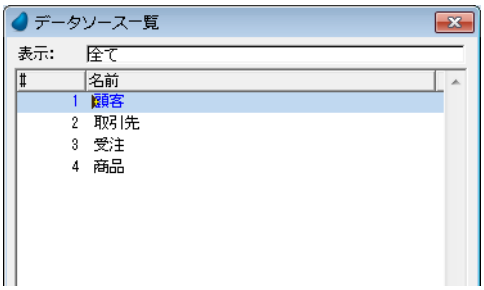
データコントロールの設定

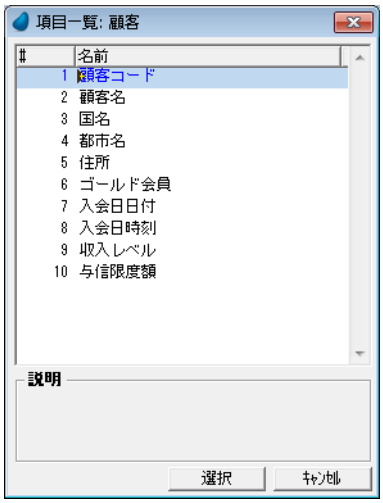
14.［コンボボックス］コントロールの［特性］シート（Alt+Enter）を開きます。

15.［データ］特性でズーム（F5）し、項目一覧から＜顧客コード＞（C）（＜受注＞データソースの項目）を選択します。



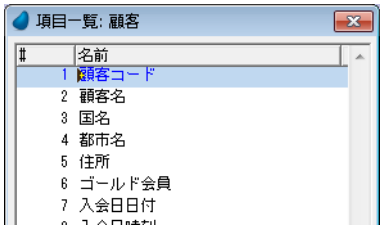
16.［データソース番号］特性からズームし、データソース一覧から＜顧客＞データソースを選択します。



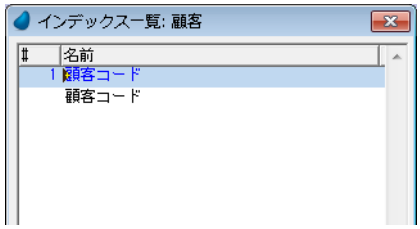


17.［表示項目］特性からズームし、項目一覧から＜顧客名＞を選択します。

18.［リンク項目］特性からズームし、＜顧客コード＞を選択します。



19.［インデックス］特性からズームし、インデックス一覧から＜顧客コード＞インデックスを選択します。



20.［TAB 順序］特性を＜ 3 ＞ にします。

21.［幅］特性に＜ 30 ＞ を入力します。

データコントロール特性について

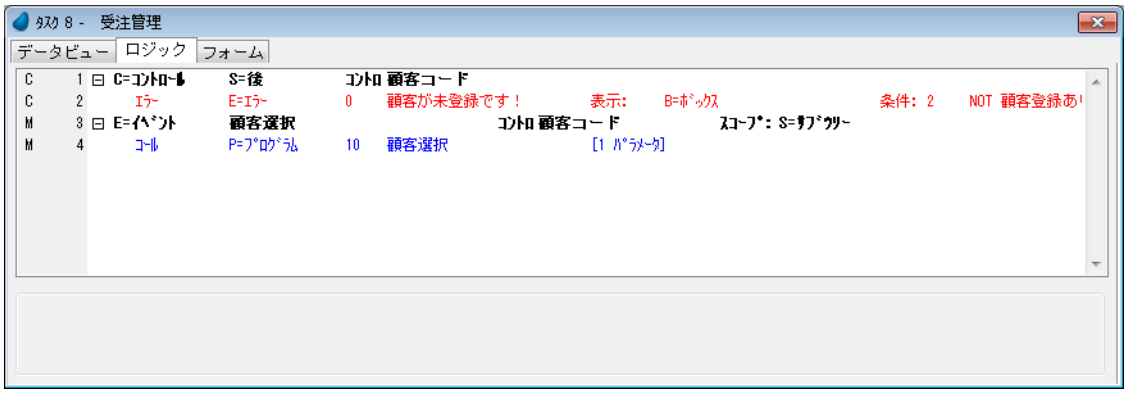
特性	ソーステーブル
データソース番号	チョイスコントロールの選択肢となるデータを読込むためのリンク先データソース • 本コースの例題では＜顧客＞データソースを選択しています。
表示項目	選択肢として画面に表示される項目 • 本コースの例題では＜顧客名＞（＜顧客＞データソース）を選択したので、データコントロールは顧客名の一覧を表示します。
リンク項目	リンク先データソースの共通項目（選択された場合に値を返す項目） • 本コースの例題では＜顧客コード＞（＜顧客＞データソース）を選択したので、２つのデータソースのリンクは＜顧客コード＞に基づいて行われます。
インデックス	テーブルの表示順を決めるインデックス • 本コースの例題では＜顧客コード＞インデックスを指定したので、＜顧客コード＞順に表示されます。
範囲項目	データソース内で選択肢の範囲を制限する場合に指定します。

既存の［選択テーブルプログラム］関連ロジックユニットの修正

データコントロールを使用することにより、エンドユーザは表示される一覧から値を選択するだけになります。

したがってデータ検証のためのロジックは［コントロール後］ロジックユニットには不要になり、また＜顧客選択＞［イベント］ロジックユニットから［選択テーブルプログラム］を呼び出す必要もなくなります。

［ロジック］エディタから、これらのロジックユニットを削除してもよいし、そのまま残しておいてもかまいません。残しておく場合は、これらのロジックユニットは（コントロール名によって対応づけられるため）新しい＜顧客コード＞データコントロールに適用されます。



22.変更内容を保存し、プログラムを閉じます。

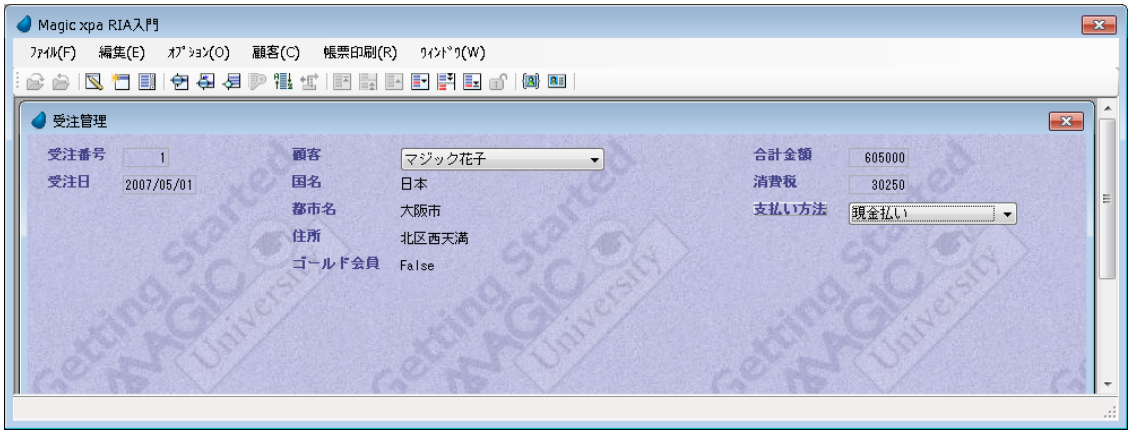
＜受注管理＞プログラムの実行と動作確認

＜受注管理＞プログラムを実行し、結果を確認しましょう。

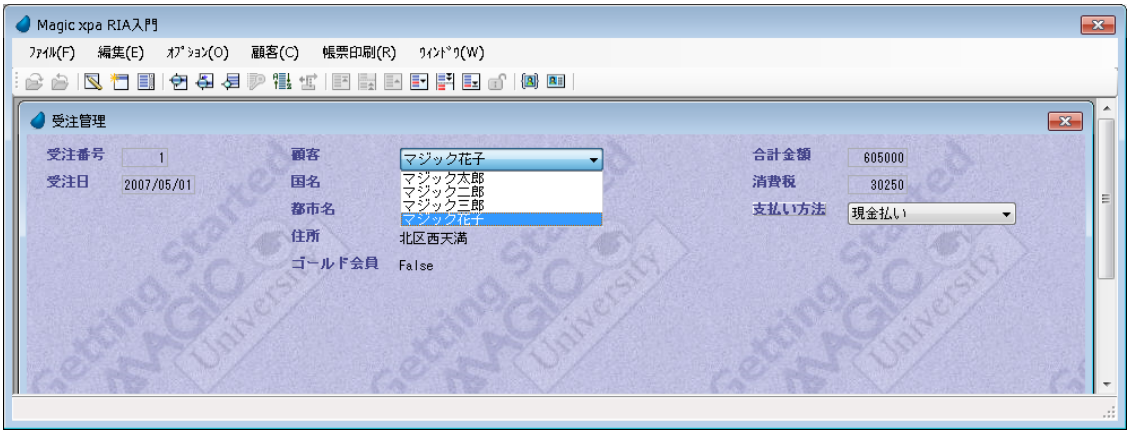
1. F7 キーを押下して＜受注管理＞プログラムを実行します。

データコントロールに顧客名が表示されていることが分かります。［表示項目］として＜顧客名＞を設定しているので、顧客コードは表示されません。

ただし、プログラム内部では、顧客名ではなく、顧客コードが保存されていることに留意してください。



ドロップダウンリストを開いて、他の値を選択することができます。



2. Esc キーを押下してプログラムを終了しましょう。

５．練習問題

この練習問題は、本章で学んだことを要約しています。

選択テーブルプログラムを取引先選択のために作成し、＜商品照会＞プログラムで表示させます。

次に、選択テーブルプログラムで作成した内容を、データコントロールに置き換えます。

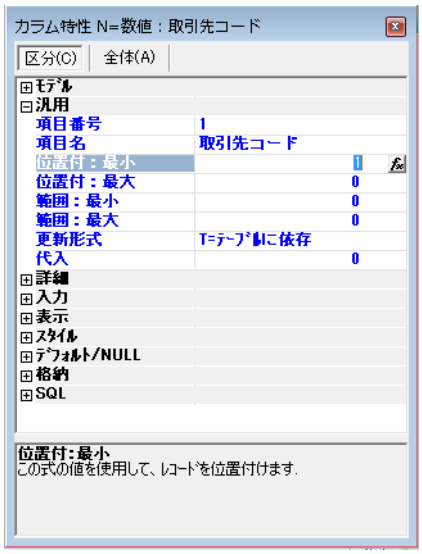
取引先選択テーブルプログラムの作成

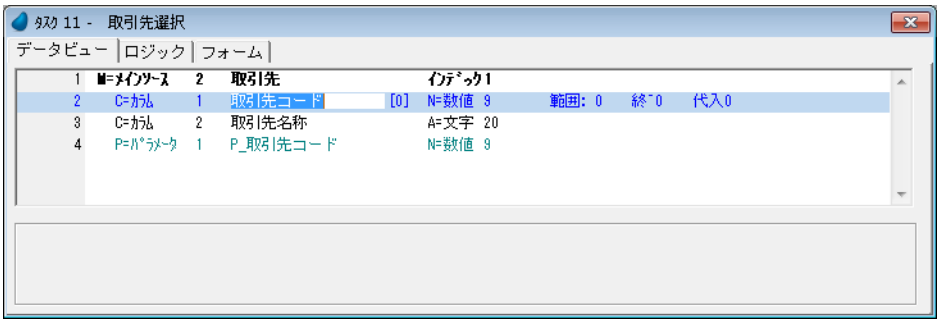
1. ［プログラム］リポジトリを開きます。
2. 新しくプログラムを作成し、＜取引先選択＞と名前をつけてください。

3. ［タスク特性］ダイアログ（Ctrl+P）を開きます。
4. ［タスクタイプ］を［C= リッチクライアント］としてください。
5. ［初期モード］を［Q= 照会］としてください。
6. ［選択テーブル］特性を［Yes］としてください。ダイアログを閉じます。
7. ［データビュー］エディタ（Ctrl+1）を開きます。
8. メインソースを＜取引先＞とします。
9. ＜取引先＞データソースから最初の２つのカラム（＜取引先コード＞と＜取引先名＞）を選択します。
10. ２つのカラム定義のあとにパラメータ＜ P_ 取引先コード＞を追加します。

位置付け条件の設定

11. ＜取引先＞データソースの＜取引先コード＞の行にパークし、特性を開きます。
12. ［位置付：最小］特性の［式］欄でズームし、［式］エディタにて一行作成し、項目＜ P_ 取引先コード＞（項目記号：C）を設定します。



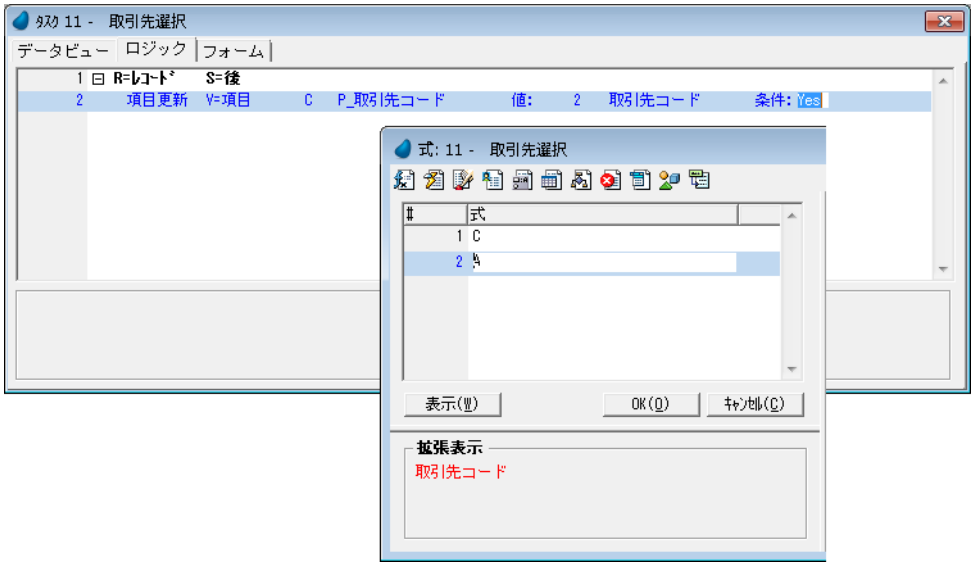


パラメータの更新

13.［ロジック］エディタを選択します。

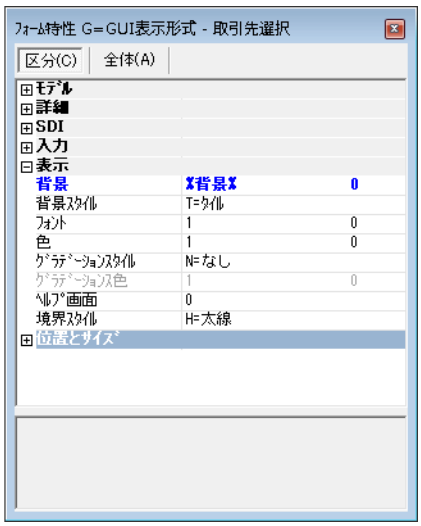
14.［レコード後］ロジックユニットを作成します。

15.＜ P_ 取引先コード＞（項目記号：C）を＜取引先コード＞（項目記号：A）で更新します。



＜取引先選択＞フォームのデザイン

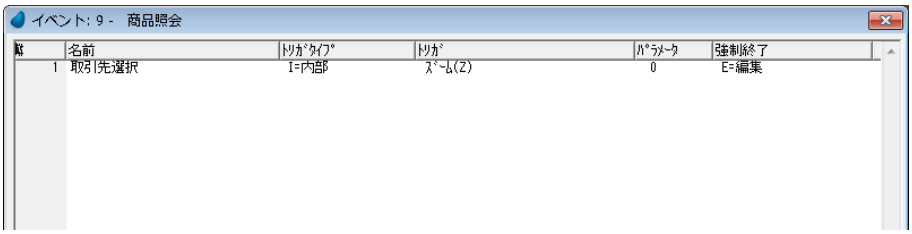
16.＜取引先選択＞フォームを下のイメージのようにデザインしてください。

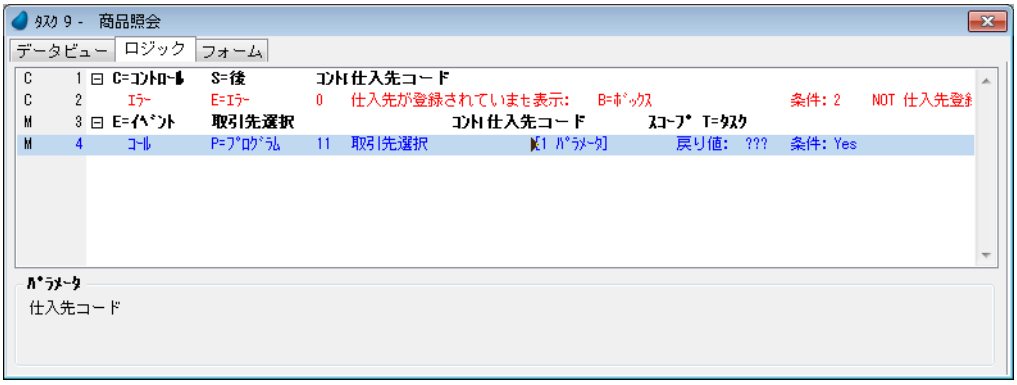


＜取引先選択＞プログラムのコール

＜商品照会＞プログラムから、＜取引先選択＞プログラムをコールするように設定しましょう。

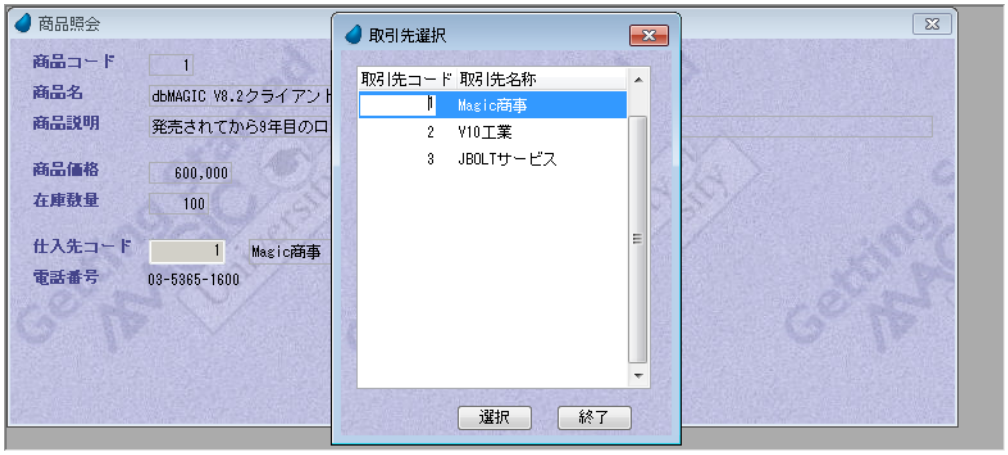
1. ＜商品照会＞プログラムを開きます。
2. ＜取引先選択＞イベントを設定します。
3. ［ロジック］エディタで新しいイベントロジックを定義し、＜仕入先コード＞コントロールに対応づけます。
4. ＜取引先選択＞プログラムをコールし、＜取引先コード＞を引数として渡すよう設定します。
5. 変更内容を保存しプログラムを閉じます。





プログラムの実行と確認

1. F7 キーを押下して、＜商品照会＞プログラムを実行させ、動作を確認しましょう。

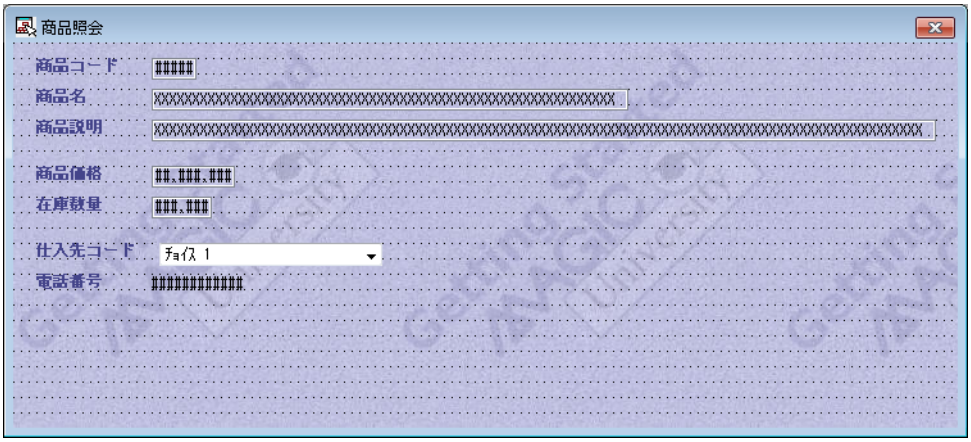


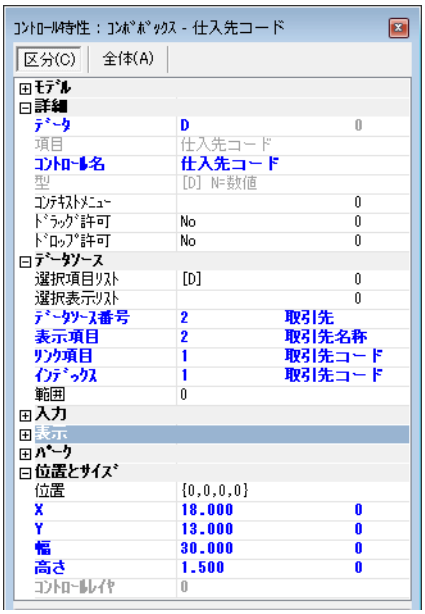
＜仕入先コード＞データコントロール

次の課題として、＜商品照会＞プログラムの＜仕入先コード＞コントロールと＜取引先名称＞コントロールを＜仕入先コード＞データコントロールに置き換えてください。

1. ＜商品照会＞プログラムを開きます。
2. ［フォーム］エディタを選択します。
3. ＜仕入先コード＞［ラベル］コントロールのテキストを＜仕入先＞に変更します。

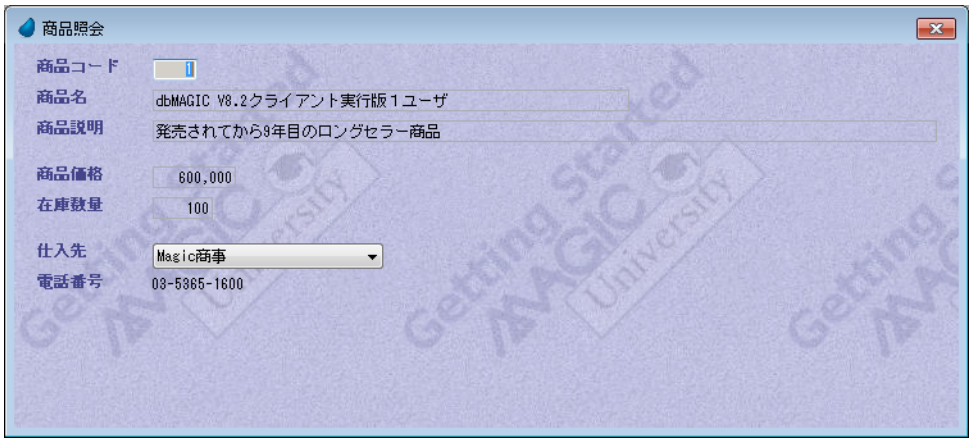
4. ＜仕入先コード＞および＜取引先名称＞の各［エディット］コントロールを削除します。
5. ［コンボボックス］コントロールを追加し、イメージで示すように特性を設定してください。





データコントロールの動作確認

6. F7 キーを押下して、＜商品照会＞プログラムを起動し、＜仕入先コード＞データコントロールの動作を確認しましょう。



６．要約

本章では、選択テーブルプログラムとデータコントロールについて学習しました。

選択テーブルプログラム

選択テーブルプログラムは Magic xpa のリッチクライアントプログラムで、エンドユーザが選択肢の一覧から値を選択できるようにするものです。

選択テーブルプログラムを作成する主な手順は次のようになります。

- プログラムの動作モードは［照会］にします。
- タスク特性の［選択テーブル］特性を［Yes］にします。
- メインソースを定義します。
- 選択値を返すためのパラメータを定義します。
- 位置付け条件を設定します。
- プログラムフォームを作成し、［選択］ボタン、［終了］ボタンを配置し、それぞれ内部イベントを定義します。

エンドユーザが選択テーブルプログラムで値を選択したときの Magic xpa の動作は次のようになります。

- ［レコード後］ロジックユニット、および［タスク後］ロジックユニットを実行します。
- プログラムの実行が終了します。
- 呼出し元プログラムにパラメータの値が返されます。

データコントロール

データコントロールはチョイスコントロールで、エンドユーザが一覧から値を選択できるようになります。

データコントロールはデータソースから読込んだレコードの値を選択肢として表示します。

データコントロールは、プログラム開発時間の短縮だけでなく、エンドユーザ操作をより容易なものとするという効果があります。

データコントロールは選択テーブルプログラムに比べ制限事項があります。

データソースのレコードで表示できるのは１カラムだけであり、また選択肢となるデータ量はごく少ない場合にのみ利用することができます。

第 17 章 １対多のデータリレーション

本章では１対多のデータリレーションについて学び、Magic xpa でのプログラミング方法を実習します。

キーワード

- １対多のデータリレーション
- タスクとプログラム
- 親タスク
- サブタスク
- 範囲
- 代入式
- ［サブフォーム］コントロール
- パラメータの受け渡し

学習目標

本章を学ぶことで、次の内容に対する理解を深めましょう。

- １対多のデータリレーションとは何か
- １対多のリレーションを持つ２つのデータソースを処理するプログラムの構成法
- サブタスクの作成法
- ［サブフォーム］コントロールの使い方

参照

Magic ヘルプの［Magic xpa リファレンス／表示フォームとコントロール／表示コントロール／サブフォームコントロール］

１．はじめに

第 15 章では１対１のデータリレーションについて学びました。

もう一つの一般的なデータリレーションとして、１対多のデータリレーションがあります。

１対多のデータリレーションは、あるデータソースの１レコードに対し、別のデータソースの複数レコードが対応関係にある状態を言います。

Magic xpa では、１対多のデータリレーションは、2 つのタスクまたはプログラムによって構成します。

この方法では、最初のタスク（通常、親タスクと呼びます）がメインとなるレコードを表示し、2 番目以降のタスク（通常、サブタスクまたは子タスクと呼びますが、本コースではサブタスクで統一表記します）が、対応する［多くの］レコードを表示します。

Magic xpa では［サブフォーム］コントロールを使用することで、親タスクのフォームのなかにサブタスクのフォームを表示させ、親タスク側で共通項目が変わるたびにサブタスク側の表示を再描画させることができるようになります。

本章ではこの１対多のデータリレーションの例として、＜受注管理＞プログラムに受注明細行を表示するサブタスクを追加してみましょう。

２．１対多のデータリレーションの前提

１対多のデータリレーションは、主となるデータソース（以後、主データソース）および従となるデータソース（以後、従データソース）との間において、主データソースの各レコードが従データソースに複数の対応レコードを持つとき成立します。

ここでは＜受注＞および＜受注明細＞データソースを用いて１対多のデータリレーションを説明します。

主データソース

以下の例においては、＜受注＞データソースが主データソースになります。

受注レコードは受注に関する次のような基本的項目を含んでいます。

- 受注番号
- 受注日
- 顧客コード
- 合計金額
- 支払方法

＜受注番号＞はデータソースのユニークな（重複不可の）インデックスです。

従データソース

＜受注明細＞データソースは従データソースであり、次のような項目を含んでいます。

- 受注番号
- 受注明細行
- 商品
- 受注数量
- 価格

＜受注番号＞と＜受注明細行＞の複合インデックスはデータソースのユニークなインデックスになります。つまり同じ＜受注番号＞が繰り返し現れたとしても、それぞれ異なる＜受注明細行＞と組み合わさることで、ユニークさを維持していることになります。

データソースの関係

＜受注＞データソースの各レコードに対応し、＜受注明細＞データソースに複数のレコードが存在します。

この関係がまさしく１対多のデータリレーションになります。

たとえば次のような受注データのサンプルを考えてみてみましょう。

＜受注＞データソースのサンプル

受注番号	受注日	顧客コード	合計金額	支払い方法
1	2007/03/01	1111	7000	現金

＜受注明細＞データソースのサンプル

受注番号	明細行	商品	数量	価格
1	1	レーザマウス	3	1500
1	2	キーボード	4	4000
1	3	USB メモリ	1	1500

＜受注番号＞の項目が両方のデータソースにあり、２つのデータソースの接続関係を成立させる共通要素になっています。

上記の例で、＜受注番号＞＝１のレコードは、＜受注＞データソースに１つ、＜受注明細＞データソースには３つあります。

１対多のデータリレーションによるメリット

もし１対多のデータリレーションを知らないでいると、受注データを１つのデータソースで実現しようとするかも知れません。先ほどの例を考えてみると、１つで実現した場合のデータソースは次のようになるでしょう。

受注番号	受注日	顧客コード	合計金額	支払方法	明細行	商品	数量	価格
1	2007/03/01	1111	7000	現金	1	レーザマウス	3	1500
1	2007/03/01	1111	7000	現金	2	キーボード	4	4000
1	2007/03/01	1111	7000	現金	3	USB メモリ	1	1500

この例では、最初の５カラムが同じデータの繰り返しになっていることが分かります。

もしこのようにデータソースを定義してしまうと、エンドユーザが同じ情報を何度も入力しなければならないだけでなく、レコードサイズが大きくなってしまうという問題が生じます。その結果データソース自体も巨大なものとなり、ディスクを不必要に占有し、またエンジンの実行速度にも悪影響を及ぼすことになってしまいます。

このような問題を避けるためには、2 つのデータソースを次の規則にしたがって定義します。

- 繰り返し出てくるデータカラムは、主データソースで定義します。
- ユニークなカラムについては、従データソースにて定義します。
- 2 つのデータソースは共通カラムによって関連付けられます。
- 共通カラムだけは従データソースで繰り返し現れても構いません。

＜受注＞データソースおよび＜受注明細＞データソースの例のように、データソースを 2 つに分けることによって、エンドユーザのタイプ入力の手間を減らすだけでなく、各データソースのレコードサイズも小さくできるというメリットが生じます。

３．商品選択プログラムの作成

１対多のデータリレーションの実習に取り組む前に、まず＜商品＞データソースのために選択プログラムを一つ作成しましょう。

商品選択プログラムは、前の章で実習した方法と同じように作成します。

写真の論理名を定義

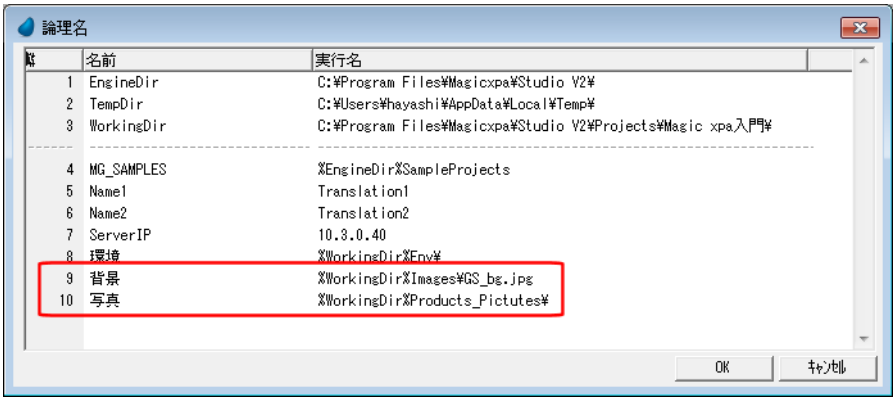
本コースのデータフォルダ内には商品のイメージデータファイルの入ったサブフォルダ（Products_Pictures）があります。

各イメージファイルの名前は、商品名と同じになっています。

商品のイメージを表示させるために、論理名を一つ定義し、イメージファイルの入っているフォルダの場所を指すようにします。

1. ［オプション］メニューから［設定／論理名］を選択します。
2. 最終行にパークします。
3. 一行作成（F4）します。

4. ［名前］カラムに＜写真＞と入力します。
5. ［実行名］カラムに＜ %WorkingDir%Products_Pictures¥ ＞と入力します。
6. ［OK］ボタンをクリックします。



7. ［プログラム］リポジトリ（Shift+F3）を開きます。
8. 最終行にパークし、一行作成（F4）します。
9. ［名前］カラムに＜商品選択＞と入力します。
10. ズーム（F5）して、＜商品選択＞プログラムを開きます。

タスク特性の設定

1. ［タスクタイプ］を「C= リッチクライアント」とします。
2. ［初期モード］を［Q= 照会］とします。
3. ［選択テーブル］特性を［Yes］にします。

タスクデータビューの定義

1. ［データビュー］エディタを選択します。
2. メインソースとして #4 ＜商品＞ を選択します。
3. インデックスは＜ 1 ＞を選択します。
4. 一行作成（F4）します。
5. ズームして［カラム選択］ダイアログを表示させます。
6. ＜商品コード＞と＜商品名＞を選択します。

7. パラメータを追加し、名前を＜P_ 商品コード＞と入力し、型と書式を＜商品コード＞と同じにします。

8. ＜商品コード＞行（#2）にパークします。
9. ＜商品コード＞の［カラム］特性で、［位置付：最小］に＜P_ 商品コード＞を設定します。

［レコード後］ロジックユニットの定義

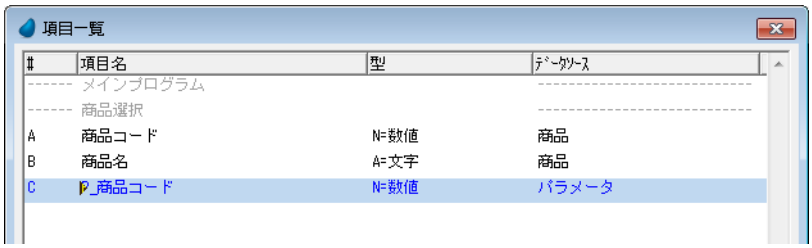
1. ［ロジック］エディタを選択します。
2. ヘッダ行（Ctrl+H）を一行作成します。
3. ロジックユニットタイプを［レコード］にします。

4. レベルを［S= 後］にします。
5. 一行作成（F4）します。
6. ［項目更新］処理コマンドを下記のように定義します。

コマンド	項目	値	条件
項目更新	C　（P_ 商品コード）	1　（商品コード）	Yes

［項目］欄の設定

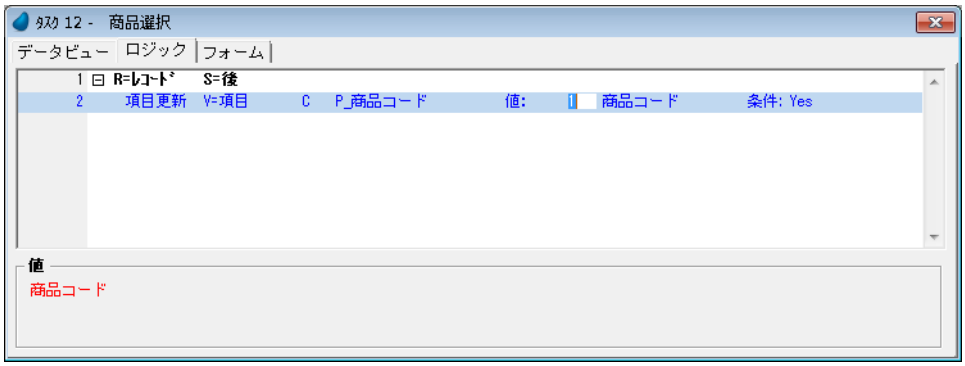
7. ［項目］欄からズームして［項目一覧］を開き、＜P_ 商品コード＞（#C）を選択します。



［値］欄の設定

8. ［式］欄からズームして［式］エディタを開きます。
9. 一行作成します。
10. ズームして項目リストに移動します。
11. ＜商品コード＞（#A）を選択します。

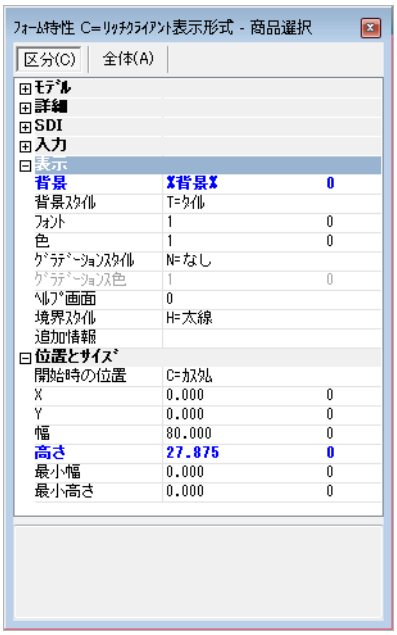
12.［OK］ボタンをクリックします。



フォームのデザイン

フォーム特性の設定

背景とウィンドウ高さを設定します。



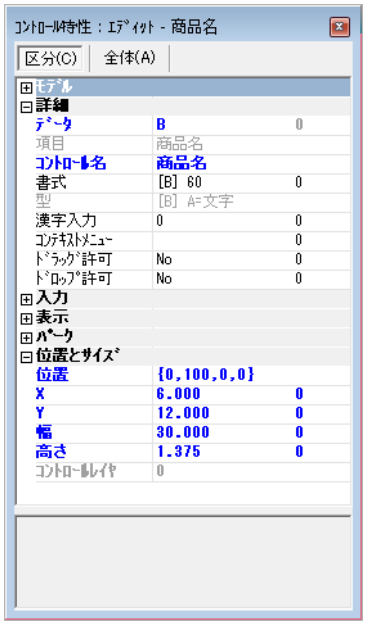
1. ［フォーム］エディタを選択します。
2. ＜商品選択＞フォーム行にパークします。
3. フォーム特性（Alt+Enter）を開きます。

4. ［背景］特性に、＜％背景％＞と入力します。
5. ［高さ］特性に、＜27.875＞と入力します。

フォームへのコントロールの配置

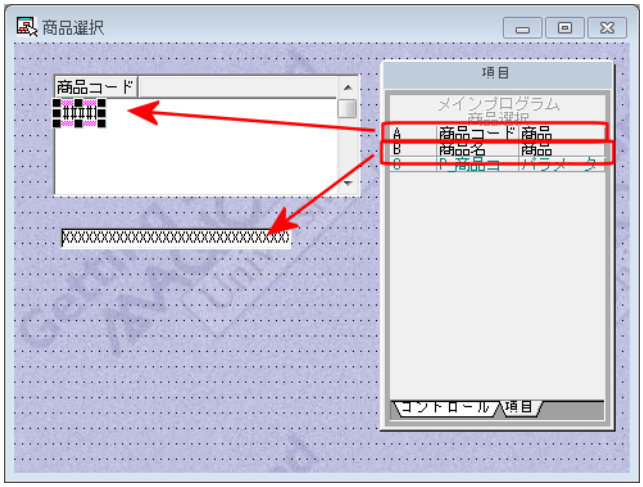
1. ［フォーム］エディタからズームして、＜商品選択＞フォームを開きます。
2. ［コントロール］パレットで［テーブル］コントロールを選択し、フォームに配置します。
3. ［エディット］コントロールを選択し、フォームに配置します。

4. この［エディット］コントロール特性（Alt+Enter）を開き、次のように設定します。
   a. ［データ］特性からズーム（F5）し、項目一覧を開きます。
   b. ＜商品名＞（#B）を選択します。
   c. ［幅］特性に、＜ 30 ＞と入力します。

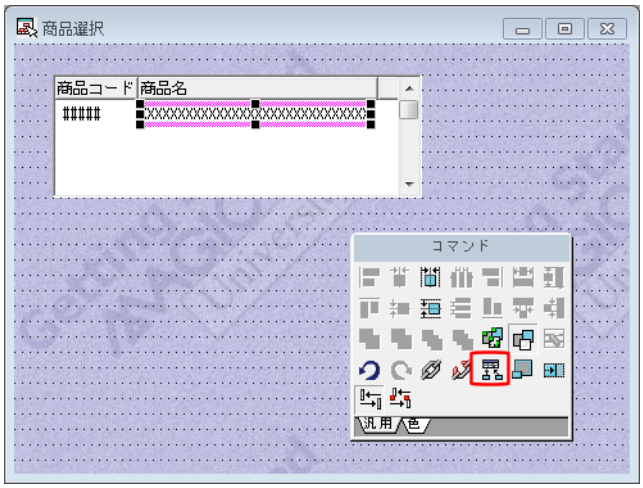


5. ［コントロール］パレットの［項目］タブを選択します。
6. ＜商品コード＞（#A）をクリックします。
7. フォーム上の［テーブル］コントロール内でクリックし、＜商品コード＞カラムを配置します。

8. ＜商品名＞の［エディット］コントロールを選択します。



9. ［コマンド］パレットで、［テーブルへの配置］アイコンをクリックします。（＜商品名＞の［エディット］コントロールがデータカラムとして追加されます。）



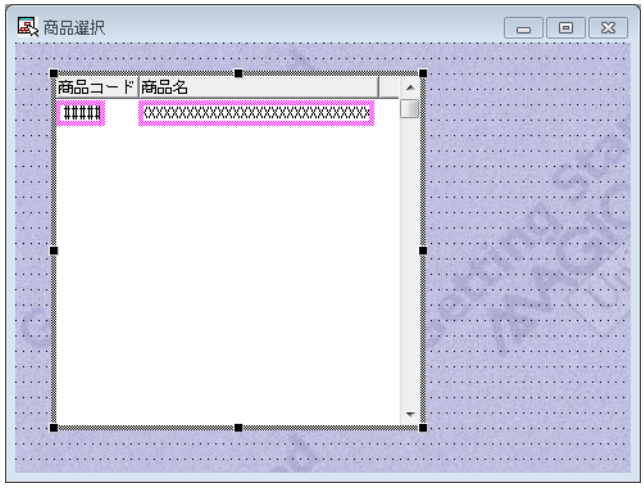
［テーブル］コントロールでの＜商品名＞カラムの作成について

＜商品名＞のカラムをテーブルに追加する場合、＜商品コード＞と同じように、直接テーブルに配置することもできるのに、なぜ上記（8， 9）のような方法で配置したのでしょう。

この場合、商品名の書式が 60 ありますが、直接テーブル内に配置してしまうと、［エディット］コントロールのサイズ調整が難しいため、まず［エディット］コントロール単体でフォームに配置し、サイズ調整を行なった後に［テーブルへの配置］アイコンの機能を使用してテーブルに配置すると、作業が容易になるからなのです。

［テーブル］コントロールのサイズ調整

1. ［テーブル］コントロール内の白い部分をクリックします。
2. 右中央部のハンドルを右方向にドラッグして調整してください。
3. 下中央部のハンドルを下方向にドラッグして調整してください。

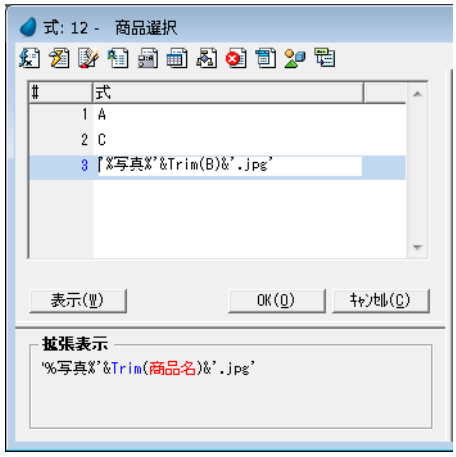


［イメージ］コントロールの追加

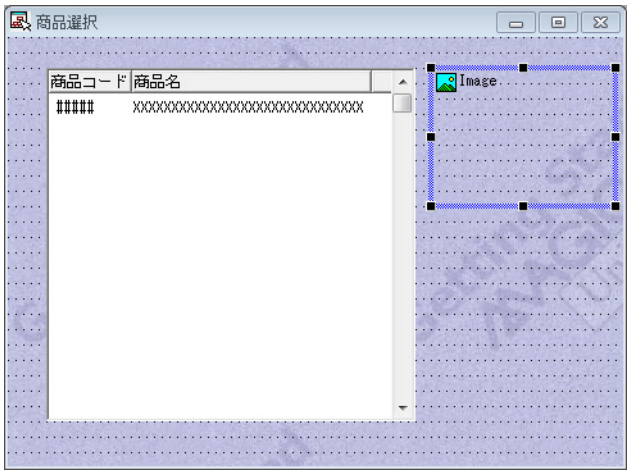
ここでは［イメージ］コントロールを使用して、商品に対応する画像をフォームに配置しましょう。

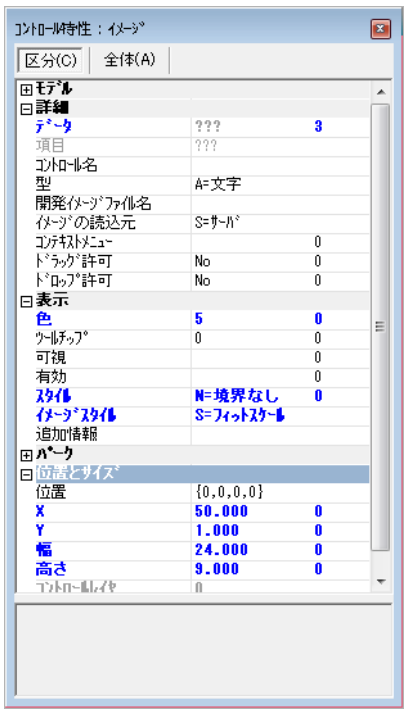
1. ［コントロール］パレットで［イメージ］コントロールアイコンをクリックします。
2. ［テーブル］コントロールの右の位置に［イメージ］コントロールを配置します。このとき［ファイルを開く］ダイアログが自動的に開きますが、［キャンセル］をクリックして閉じてください。

3. ［イメージ］コントロール特性（Alt+Enter）を開きます。
4. ［データ］特性の［fx］ボタンをクリックし、［式］エディタを開きます。
5. 一行作成し、次のように入力します。＜'%写真%'&TRIM(B)&'.jpg'＞



6. ［スタイル］特性は、［N= 境界なし］を選択します。
7. ［色］特性は、＜ 5 ＞とします。
8. ［イメージスタイル］特性は、［S= フィットスケール］とします。
9. ［幅］特性は＜ 24 ＞とします。
10. ［高さ］特性は＜ 9 ＞とします。





［イメージ］コントロールについて

［イメージ］コントロールは画像データを表示するときに使用します。

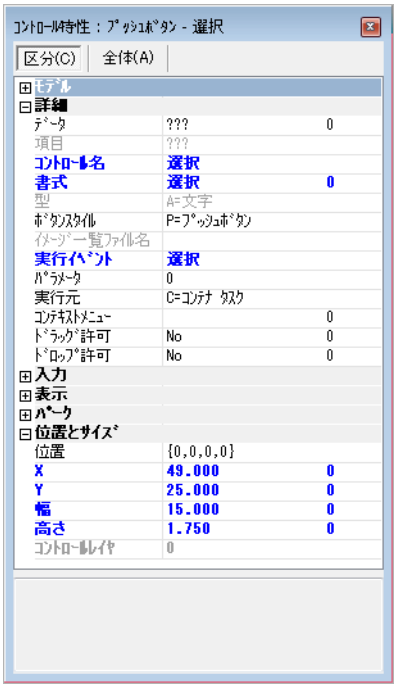
画像データを表示するための特性には以下のものがあります。

［イメージ］コントロール設定に関連するイメージ特性

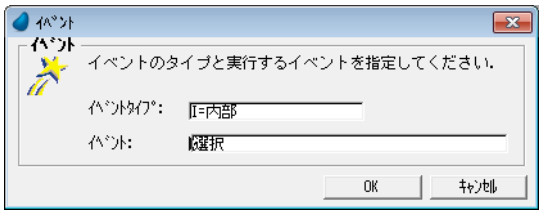
特性	説明
データ	画像ファイルのパスとファイル名を内容するデータ項目あるいは式を指定します。
開発イメージファイル名	ここからズームして［ファイルを開く］ダイアログを開き、［フォーム］エディタ上に表示する画像ファイルを選択します。
イメージの読込元	画像ファイルをサーバ／クライアントのどちらから取得するかを指定します。「サーバ」を指定した場合、［データ］特性にはサーバ上のパスを指定します。画像ファイルは、実行時に自動的にクライアントに転送されます。
イメージスタイル	表示されるイメージを［イメージ］コントロールに合わせる方法を指定します。有効な値は以下のとおりです。 • T= タイル …… 領域全体が埋まるまで、画像を繰り返しコントロールにコピーします。 • C= コピー …… 画像をそのままコントロールにコピーします。コントロールより大きいイメージはトリミングされます。イメージがコントロールより小さい場合、コントロールの一部がカバーされないままになることがあります。 • S= フィットスケール …… 画像がコントロールの高さまたは幅に収まるようにサイズが調整されてコピーされます。この場合、画像の縦横比は維持されます。画像の縦横比とコントロールの縦横比が同じでない場合、コントロールの一部がカバーされないままになることがあります。 • F= フィルスケール …… 画像がコントロールの領域を全て埋めるようにコピーされます。イメージの縦横比は維持されます。画像の縦横比とコントロールの縦横比が同じでない場合、画像の一部だけがコントロールに表示されることがあります。 • D= コントロールスケール …… 画像の縦横比が調整され、［イメージ］コントロールの領域を全て埋めるようにコピーされます。したがって、元の画像は縦長または横長になります。画像の縦横比とコントロールの縦横比が同じでない場合、縮小または拡大された画像は歪んで表示されます。

［選択］と［終了］ボタンの追加

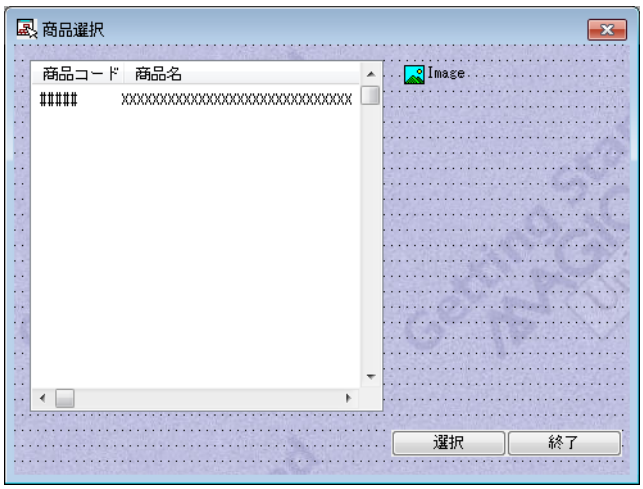
前章で作成した選択プログラムと同じように、このプログラムでも［選択］と［終了］のプッシュボタンを作成しましょう。ここでは［選択］プッシュボタンについて追加しますが、［終了］ボタンについては説明を省きますので、同じ作業を行ない、追加するようにしてください。



1. ［プッシュボタン］コントロールをフォームに配置します。
2. ［プッシュボタン］コントロール特性を開きます。
3. ［コントロール名］に＜選択＞と入力します。
4. ［書式］にも＜選択＞と入力します。
5. ［実行イベント］からズームして［イベント］ダイアログを開きます。



   a. ［イベントタイプ］は［I= 内部］を選択します。
   b. ［イベント］からズームしてイベント一覧を開き、［選択］を選択します。
   c. ［OK］ボタンをクリックして［イベント］ダイアログを閉じます。
6. ［幅］特性に＜ 15 ＞と入力します。
7. 上記までの操作を［終了］ボタンについても実行してください。（5 の b. のところでは［終了］イベントを選択）
8. 変更内容を保存し、プログラムを終了します。



ここまでの作業で、＜商品選択＞プログラムが完成しました。あとで＜受注管理＞プログラムからこの選択プログラムを呼び出すようになります。

４．受注明細行管理タスクの作成

このセクションでは、＜受注管理＞プログラムにサブタスクを追加してみましょう。

サブタスクとして＜受注明細行管理＞タスクを作成することで、＜受注＞データソースと＜受注明細＞データソースとの間に 1 対多のデータリレーションを構築します。

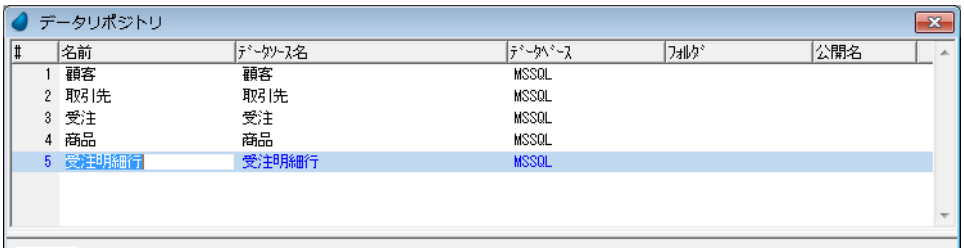
まず＜受注明細行＞データソースの定義をしましょう。

受注明細行データソースの定義

1. ［データ］リポジトリ（Shift+F2）を開きます。
2. 最終行（#4 ＜商品＞）にパークします。
3. 一行作成します。

4. 次のように設定します。

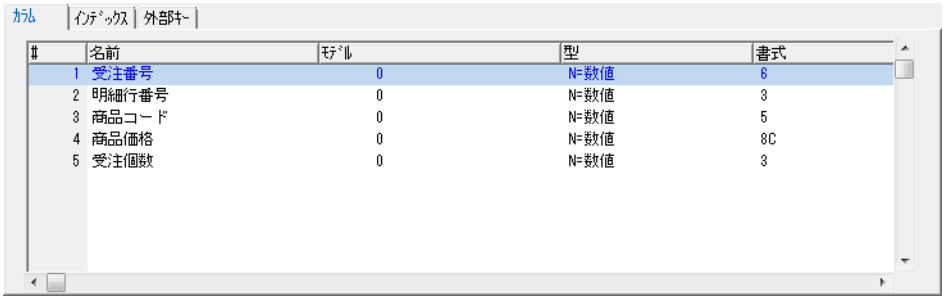
#	名前	データソース名	データベース	フォルダ	公開名
5	受注明細行	受注明細行	MSSQL	Yes	



＜受注明細行＞カラムの定義

5. 下部ペインの［カラム］タブを選択します。
6. 五行作成します。（F4 キーを 5 回押下する）
7. 次のようにカラムを 5 つ定義します。

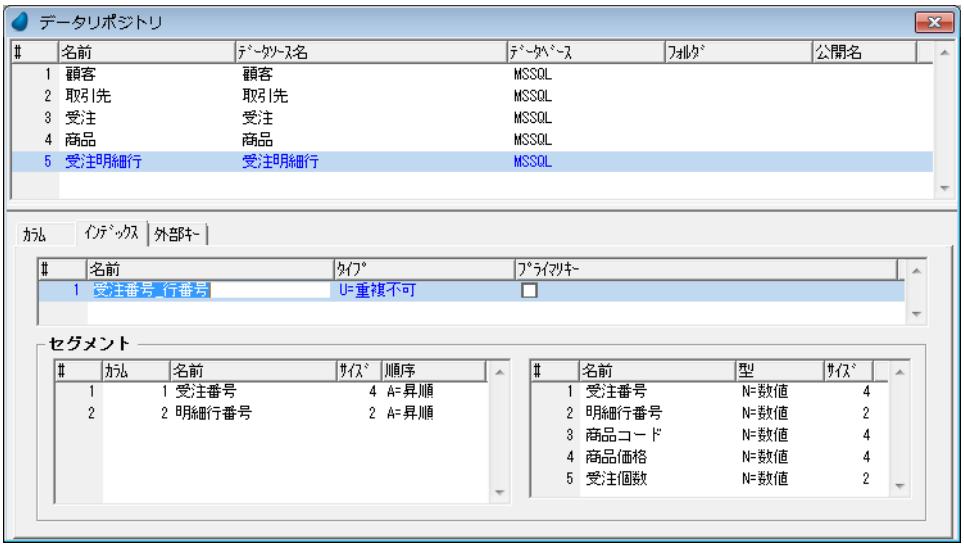
#	名前	モデル	型	書式
1	受注番号	0	N= 数値	6
2	明細行番号	0	N= 数値	3
3	商品コード	0	N= 数値	5
4	商品価格	0	N= 数値	8C
5	受注個数	0	N= 数値	3



＜受注明細行＞インデックスの定義

8. ［インデックス］タブを選択します。
9. 一行作成します。
10. 次のようなインデックスを作成します。
    - 名前 …… ＜受注番号 _ 行番号＞
    - タイプ …… ［U= 重複不可］
11. ズームして［セグメント］テーブルに移動します。（またはセグメントテーブルをクリックします）
12. 二行作成します。
13. ［カラム］欄に右の項目一覧の番号を参照し、#1：受注番号と #2：明細行番号の番号を順に設定します。（1 と 2）

14.Enter キーを 2 回押下して、一番上のペインに戻ります。



（サブ）タスクの作成

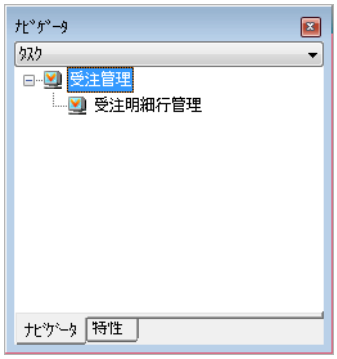
1 対多のデータリレーションを構築するために、一つのプログラム内に二つのタスクを作成します。一つが親タスクとなり、二つめがそのサブタスクになります。

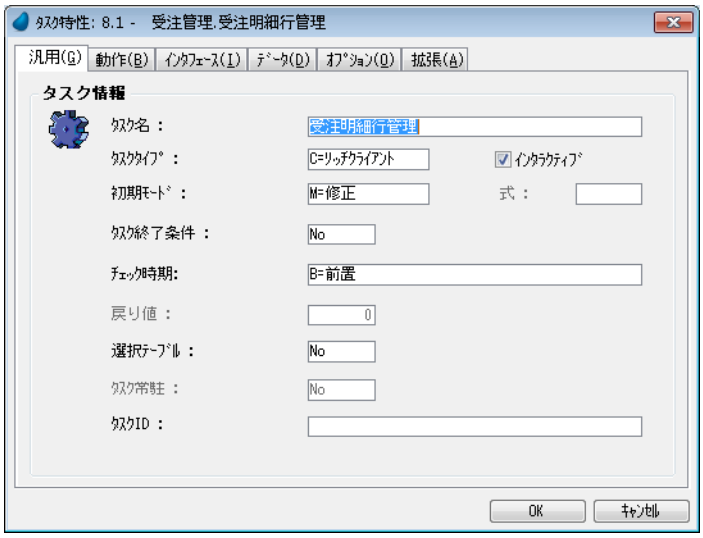
ここでは＜受注管理＞タスクにサブタスクとして＜受注明細行管理＞を追加するようにします。タスクについて記憶があいまいになっている場合は、第 3 章をもう一度復習しておきましょう。

タスク特性の設定

1. ［プログラム］リポジトリ（Shift+F3）を開きます。
2. ＜受注管理＞プログラムにズームします。
3. ［表示］メニューから［ナビゲータ］を選択し、［ナビゲータ］ペインを表示させます。

4. ［ナビゲータ］ペインで、＜受注管理＞エントリをクリックしてパークします。
5. 一行作成（F4）します。新しいエントリ（この場合、サブタスク）がタスクツリーに作成され、その［タスク特性］ダイアログが自動的に開きます。
6. ［タスク名］に**＜受注明細行管理＞**と入力します。
7. ［OK］ボタンをクリックします。

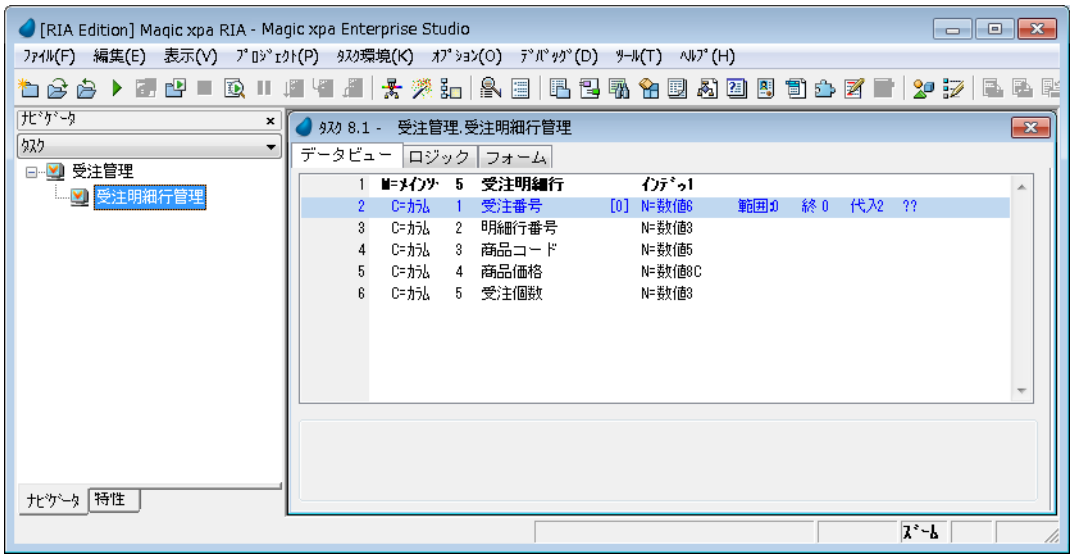




サブタスクのデータビュー定義

1. ［データビュー］エディタを選択します。
2. メインソースとして、＜受注明細行＞（#5）　を選択します。
3. インデックスは＜ 1 ＞とします。

4. 一行作成（F4）します。
5. カラム番号欄からズームして項目一覧を表示します。
6. 項目一覧にあるすべてのカラムを選択します。
7. ［選択］ボタンをクリックします。（または Enter キーを押下します。）



リンクの追加

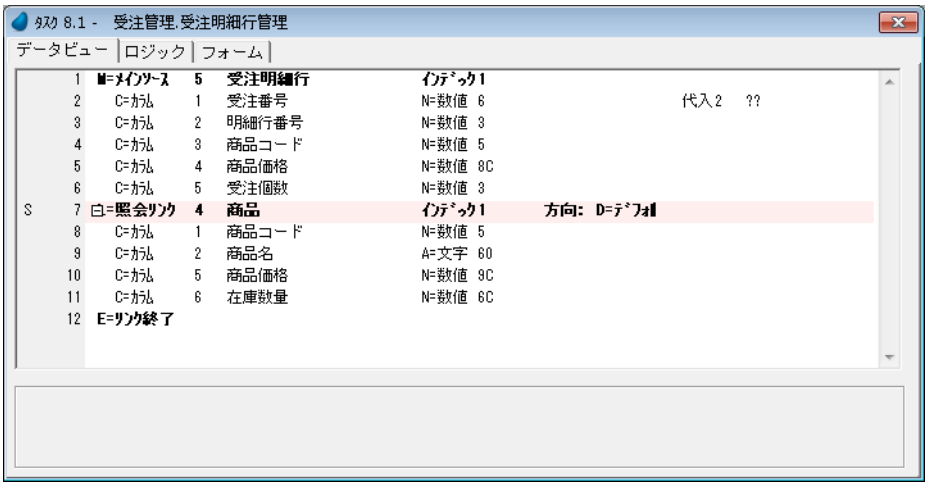
＜受注明細行＞データソースには、＜商品＞データソースと 1 対 1 のデータリレーションを持つカラムがあります。すなわち＜商品番号＞です。受注商品の詳細情報を得るために、＜商品＞データソースへのリンクを追加しましょう。

8. 最終行にパークします。
9. ヘッダ行（Ctrl+H）を作成します。
10. ［L= 照会リンク］を選択し、＜商品＞データソース（#4）を選択します。

    インデックスは＜ 1 ＞（商品コード）を指定します。

    ＜商品コード＞カラム行が自動的に一行追加されます。

11. 一行作成（F4）します。
12. 作成した行のカラム選択欄からズームして、［カラム選択］テーブルを開きます。
13. #2，#5，#6 の各行を選択します。（Ctrl キーを押下しながら、クリックして選択します）



リンク先データソースに渡す位置付用パラメータの設定

データリレーションを維持するために、メインソースにある共通カラムを使用して、リンク先共通項目の位置付パラメータを設定しましょう。

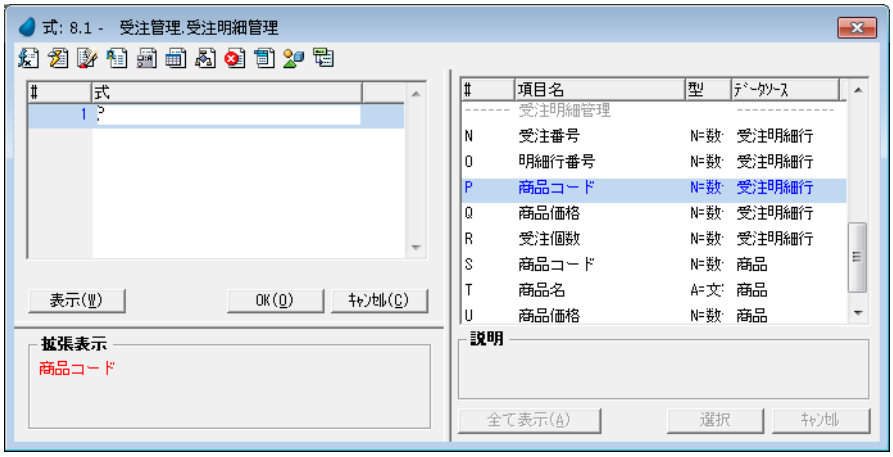
このプログラムの場合、2 つのデータソースにある＜商品コード＞が共通カラムになります。したがってここではメインソースの＜商品コード＞をリンク先データソースでの位置付パラメータとして使用します。

14.＜商品コード＞の行（#8）にパークします。（リンク定義内にあります）

15.［位置付］の式欄からズームして、［式］エディタを開きます。

16.［式］エディタで一行作成します。

17.ズームして項目リストに移動し、＜受注明細行＞データソースの＜商品コード＞を選択します。



18.［OK］ボタンをクリックします。

19.Tab キーを押下して、［終了］の式欄に移動します。

20.［終了］の式欄からズームして［式］エディタを開きます。

21.［位置付］の式欄に選択した同じ式を選択します。（#1 の式）

22.［OK］をクリックします。

2 つの式欄には同じ値（この場合＜１＞）が設定されています。

データ整合性の維持

＜受注明細行＞データソースは、＜受注＞データソースからみて、１対多のリレーションがありました。

サブタスク＜受注明細行管理＞が、親タスク＜受注管理＞の受注レコードに関連したレコードだけを処理するためには、次のようになっていることが必要です。

- ［範囲］条件として、共通項目（この場合、＜受注番号＞）を設定すること。
- サブタスクの共通項目＜受注番号＞は、親タスクの共通項目＜受注番号＞を初期値とすること。
- エンドユーザはサブタスク側の共通項目＜受注番号＞を見ることができないか、または更新できないようにすること。

サブタスクの内容は［サブフォーム］コントロールを利用して、親タスクのフォーム上に表示させるようにします。

［サブフォーム］コントロールは、親タスクからサブタスクに渡されるパラメータの値に応じて、サブタスク側の表示内容を自動的に更新します。ただし、［サブフォーム］コントロールの自動更新機能を使用するためには、サブタスク側でパラメータ項目を定義しなければなりません。

＜受注番号＞パラメータの設定

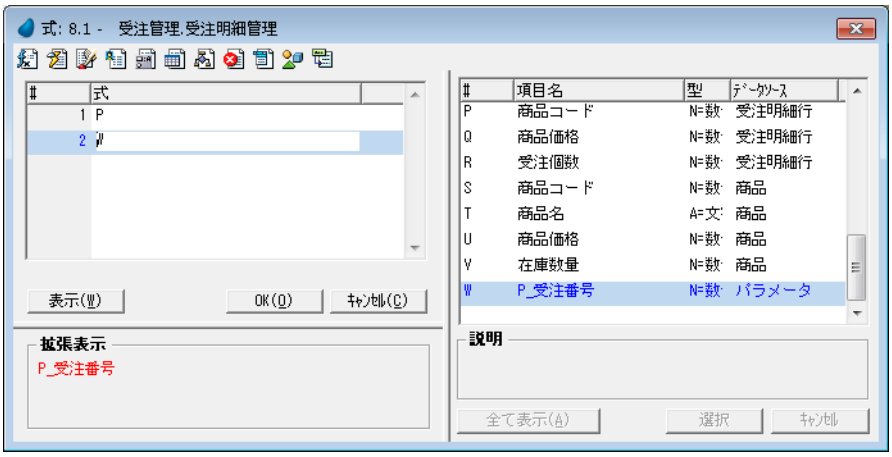
サブタスク＜受注明細行管理＞の［データビュー］エディタで次のようにします。

1. 最終行（#12［E= リンク終了］）にパークします。
2. 一行作成（F4）します。
3. ［P= パラメータ］を選択し、名前として＜ P_ 受注番号＞、型を［N= 数値］、書式を＜ 6 ＞と定義します。

タスク範囲の定義

4. メインソースの＜受注番号＞行（#2）にパークします。
5. ［範囲］の式番号欄からズームし、［式］エディタを開きます。

6. ［式］エディタで最終行にパークし、一行作成します。
7. ズームして項目リストに移動し、＜ P_ 受注番号＞（#W）を選択します。



8. ［OK］ボタンをクリックします。
9. Tab キーを押下して、［終了］の式番号欄にパークします。
10.［終了］の式番号欄からズームし、［式］エディタを表示します。
11.［範囲］の式番号欄に設定した式と同じものを選択します。
12.［OK］ボタンをクリックします。

範囲条件設定を行う両方の式番号欄には、同じ式番号が入っています。（この場合、＜ 2 ＞）

受注番号の初期化

1. ＜受注番号＞（#2）カラム行で、［代入］の式番号欄にパークします。
2. ズームして［式］エディタを開きます。
3. 式番号 #2 を選択します。（＜ P_ 受注番号＞）

4. ［OK］ボタンをクリックします。

新しいレコードがサブタスクで作成される都度、Magic xpa は親タスクからパラメータで渡される＜受注番号＞の値で、メインソース＜受注明細行＞の＜受注番号＞カラムを初期化します。

商品選択プログラムの呼び出し

ここまで、サブタスクのデータビューとデータの整合性を維持するためのパラメータ設定を行いました。

ここからはサブタスクにロジックを追加します。最初に商品の選択を行なうことができるようにしましょう。

すでに学んだように、エンドユーザが一覧から選択する方法は２つあります。

- 選択プログラム
- データコントロール

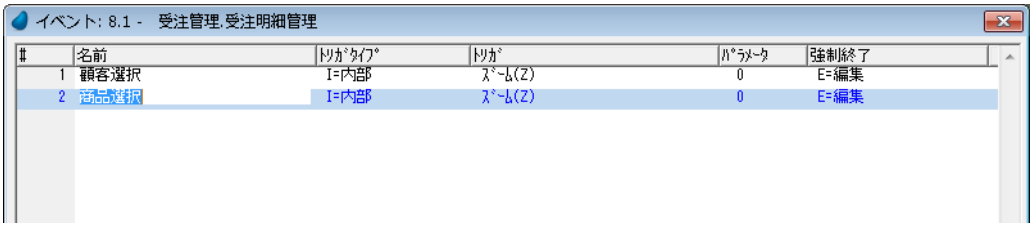
本章では選択プログラムを利用しますが、これは商品イメージを表示することで、エンドユーザにより視覚的な情報を提供し、正しい商品を選択しやすくしています。データコントロールの場合、商品名のみしか表示できないため、エンドユーザは判断しづらいことでしょう。

商品選択イベントの追加

［ユーザイベント］テーブルに＜商品選択＞イベントを追加しましょう。

1. ［タスク環境］メニューから［ユーザイベント］を選択します。
2. ＜顧客選択＞行（#1）にパークし、一行作成します。
3. 次のように設定します。

#	説明	トリガタイプ	トリガ	パラメータ	強制終了
5	商品選択	I= 内部	ズーム (Z)	0	E= 編集



トリガの設定

4. ズームして、イベント一覧を表示します。

5. ［ズーム（Z)］を選択します。

［強制終了］を ［E= 編集］に設定しているため、商品選択プログラムを終了したときに、選択された値が＜商品コード＞［エディット］コントロールに表示されるようになります。

商品選択ハンドラの作成

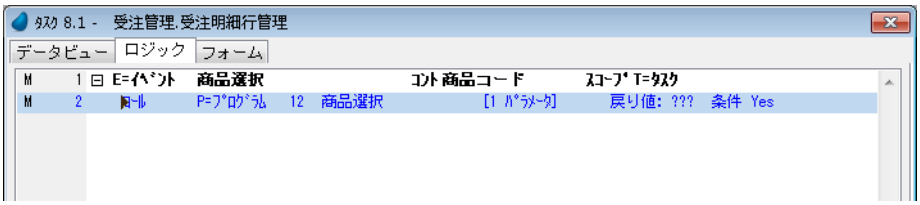
ここでは［イベント］ロジックユニットを作成し、商品選択プログラムを呼び出すように定義します。

1. ［ロジック］エディタを開きます。
2. ヘッダ行を一行作成します。
3. ［イベント］ロジックユニットを選択し、［イベント］ダイアログにて、［イベントタイプ］を［U= ユーザ］、［イベント］欄からズームして＜商品選択＞を選択します。
4. ［コントロール］欄には、直接＜商品コード＞と入力します。

サブタスク＜受注明細行管理＞のフォームはまだ定義していないため、［コントロール］欄からズームしてコントロール一覧を表示しても何も表示されず、選択することはできません。
このような場合、直接タイプ入力して、後からフォームのコントロール名を一致させるようにします。

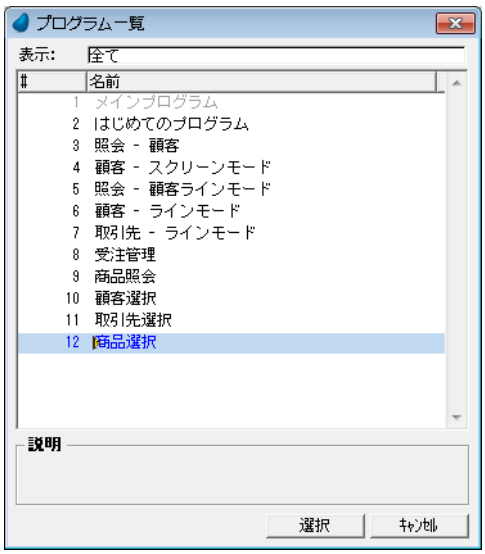
5. スコープは［T= タスク］とします。
6. 一行作成します。
7. 次のように設定します。

コマンド	プログラム ID	パラメータ	戻り値	条件
コール　P= プログラム	12　（商品選択）	1　（商品コード）	0	Yes



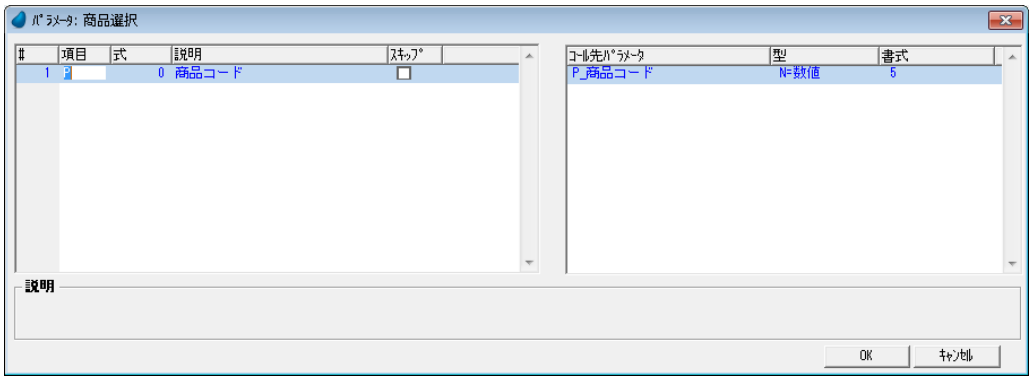
プログラム ID の設定

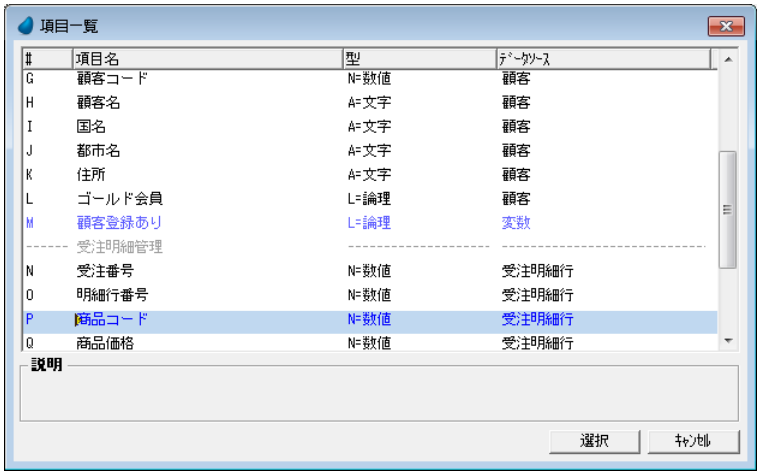
8. ズームしてプログラム一覧を表示します。
9. ＜商品選択＞プログラム（#12）を選択します。



パラメータの設定

10. ズームして［パラメータ］テーブルを開きます。
11. 一行作成します。
12. ［項目］欄からズームして項目一覧を表示し、＜受注明細行＞データソースの＜商品コード＞を選択します。
13. ［OK］ボタンをクリックします。





商品価格の更新

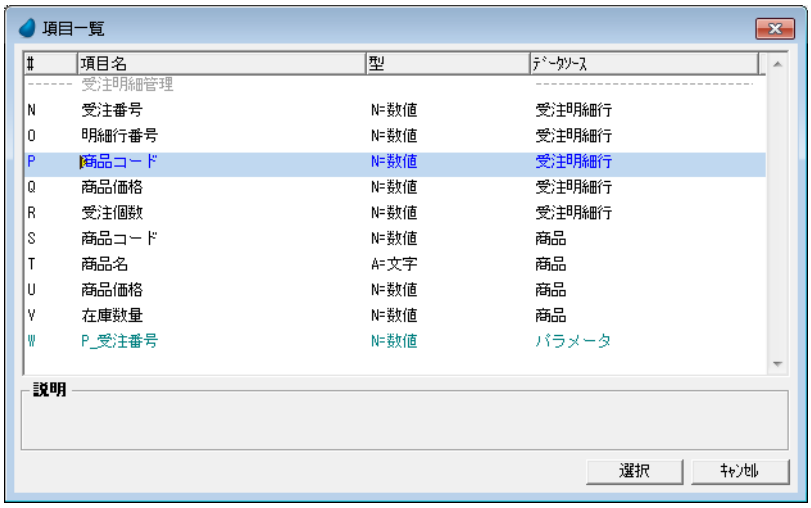
各商品はデフォルトの価格を持っており、＜商品＞データソースの＜商品価格＞カラムに保持しています。

新しく受注行を作成し、商品コードを登録したとき、デフォルトの価格がエンドユーザに対し示されるようにします。また、エンドユーザはこの価格をそれぞれの受注行ごとに個別に変更できるようにします。このために＜受注明細行＞データソースのレコードには＜商品価格＞カラムが定義してあります。

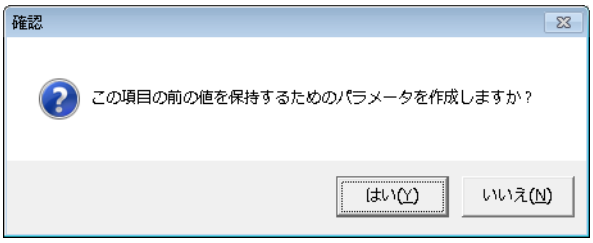
ここではこのロジックを、［項目変更］ロジックユニットと［項目更新］処理コマンドを使用して実現します。

1. ［ロジック］エディタを開きます。
2. 最終行（#2）にパークします。
3. ヘッダ行（Ctrl+H）を一行作成します。

4. ［項目変更］ロジックユニットを、＜商品コード＞コントロールに対して定義します。（ズームして項目一覧から選択します。



5. 前の値を保持するパラメータを作成するかどうかの［確認］ダイアログが表示されますので、［はい］を選択します。



2 行の変数項目定義行が［項目変更］ロジックユニットのヘッダ行の下に自動的に作成されます。

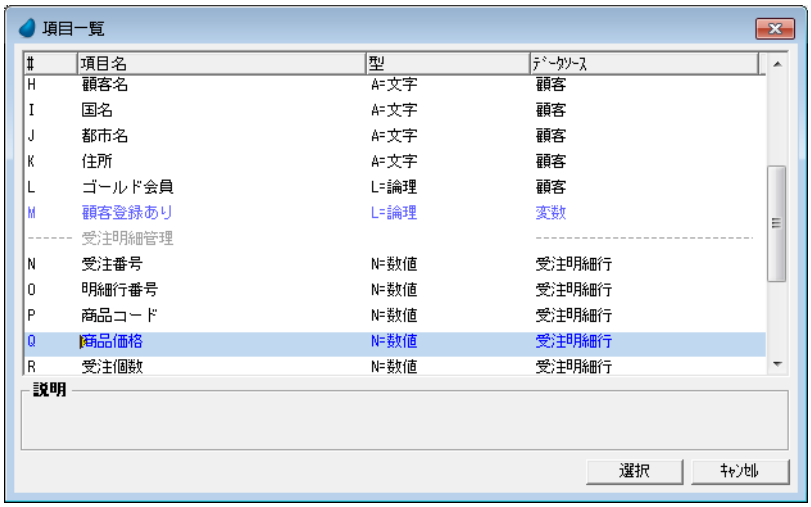
6. 最終行（#5）にパークします。
7. 一行作成（F4）します。
8. 次のように設定します。

コマンド	項目	値	条件
項目更新	Q （商品価格）	3 （商品価格）	4 （CHG_PRV_ 商品コード <> 商品コード）

項目の設定

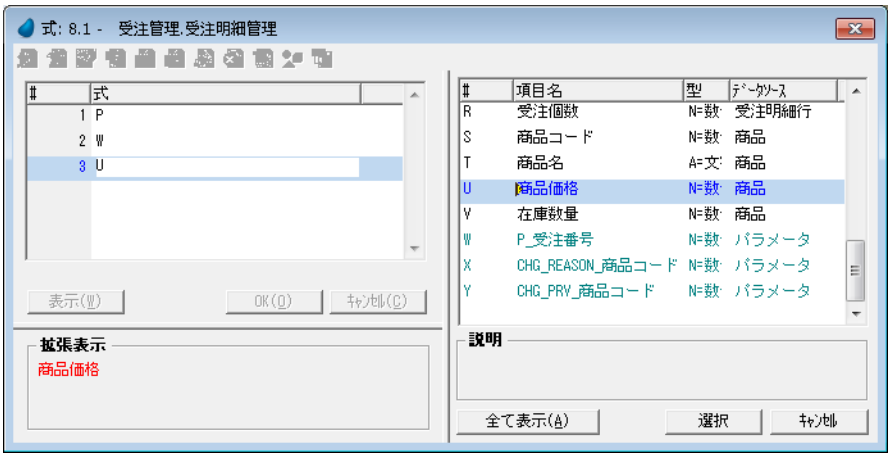
9. ズームして項目一覧を表示します。

10.＜受注明細行＞データソースの＜商品価格＞（#Q）を選択します。



値の設定

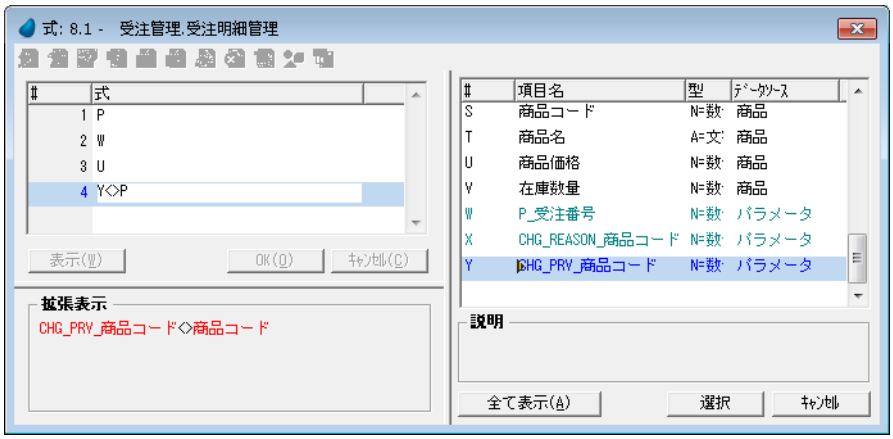
11.[式] 欄からズームして [式] エディタを開きます。
12.最終行にパークします。
13.一行作成（F4）します。
14.ズームして項目リストに移動します。
15.＜商品＞データソースの＜商品価格＞（#U）を選択します。
16.[式] エディタの [OK] ボタンをクリックします。



条件の設定

17.[式] 欄からズームして [式] エディタを開きます。
18.最終行にパークします。
19.一行作成（F4）します。
20.ズームして項目リストに移動します。
21.変数項目＜ CHG_PRV_ 商品コード＞（#Y）を選択します。
22.<> とタイプ入力します。
23.もう一度項目リストに移動します。
24.＜受注明細行＞データソースの＜商品コード＞（#P）を選択します。

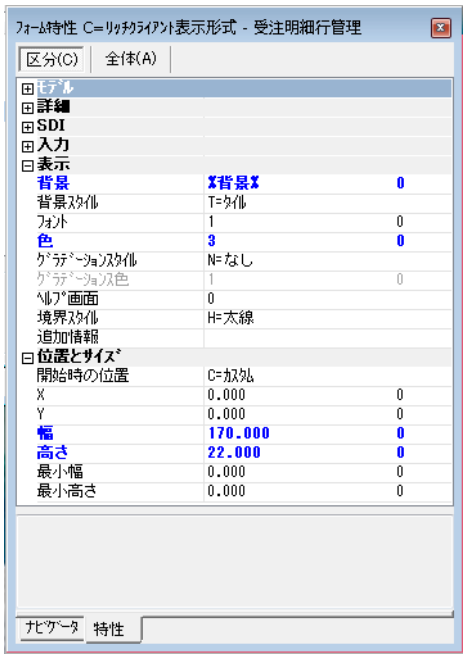
25.［式］エディタの［OK］ボタンをクリックします。

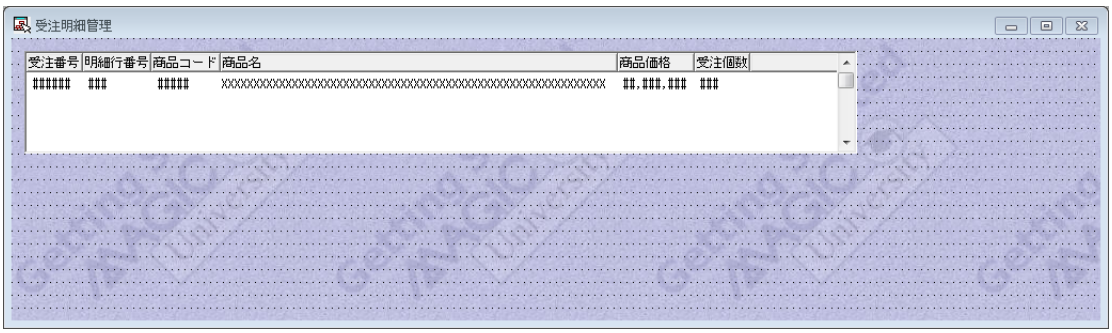


ズームして選択するかわりに、項目記号が分かっている場合は、直接キーボード入力しても同じです。

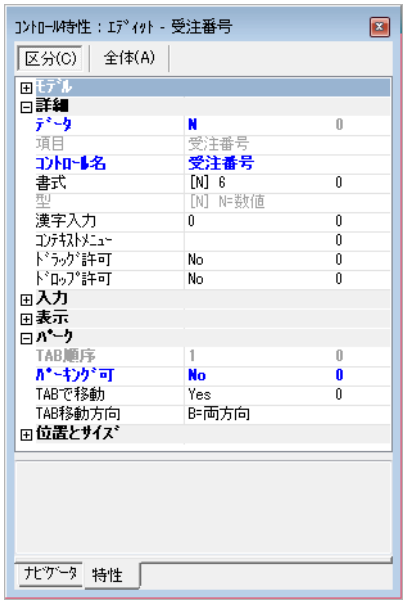
サブタスクの受注明細行フォームデザイン

1. ［フォーム］エディタを開きます。
2. ＜受注明細行管理＞フォームの特性を開きます。
3. ［背景］特性に＜％背景％＞を入力します。
4. ［幅］特性に＜170＞を設定します。
5. ［高さ］特性に＜22＞を設定します。
6. ズームして＜受注明細行管理＞フォームを開きます。
7. ［テーブル］コントロールをフォーム左上部に配置します。
8. テーブルに次の項目を配置します。
   - ＜受注番号＞（＜受注明細行＞データソース）
   - ＜明細行番号＞（＜受注明細行＞データソース）
   - ＜商品コード＞（＜受注明細行＞データソース）
   - ＜商品名＞（＜商品＞データソース）
   - ＜商品価格＞（＜受注明細行＞データソース）
   - ＜受注個数＞（＜受注明細行＞データソース

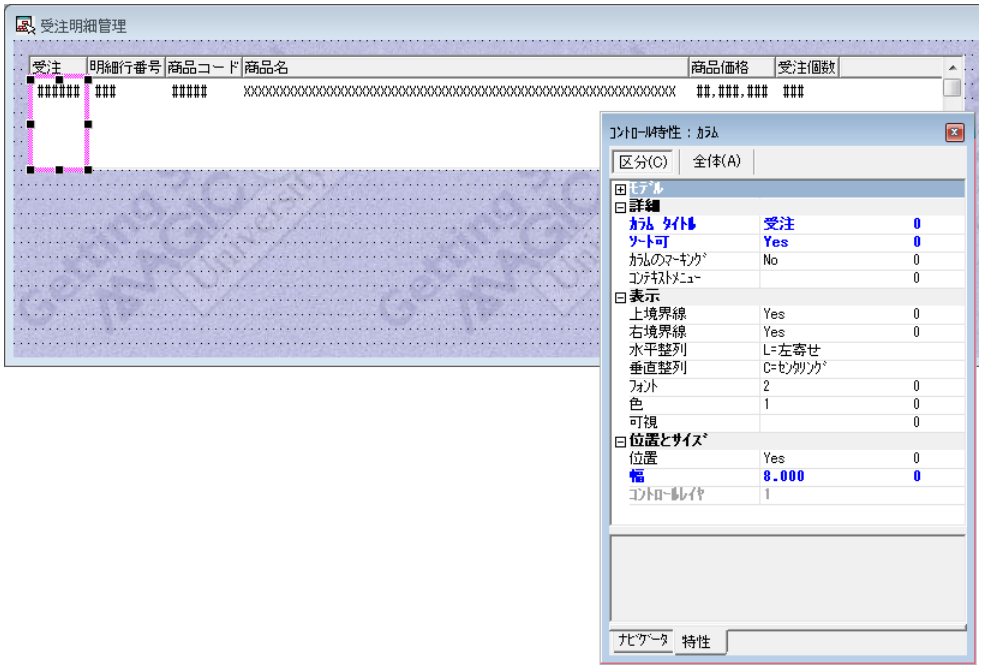




＜受注番号＞カラムの特性設定



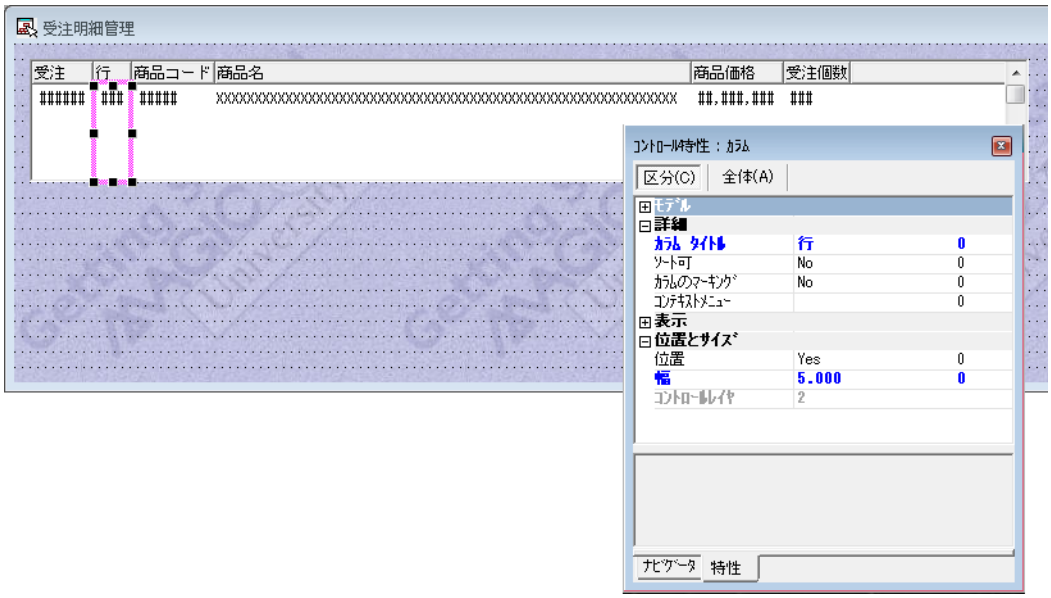
1. ＜受注番号＞［エディット］コントロールの特性を開きます。
2. ［パーキング可］特性を［No］にします。
3. ＜受注番号＞［エディット］コントロールの特性を閉じます。
4. Alt キーを押下しながら＜受注番号＞カラムのテーブル内の白い領域をクリックします。
5. ［カラム特性］（Alt+Enter）を開きます。
6. ［カラムタイトル］特性を**＜受注＞**に修正します。
7. ［幅］特性を ＜ 8 ＞ に設定します。



＜明細行番号＞カラムの特性設定

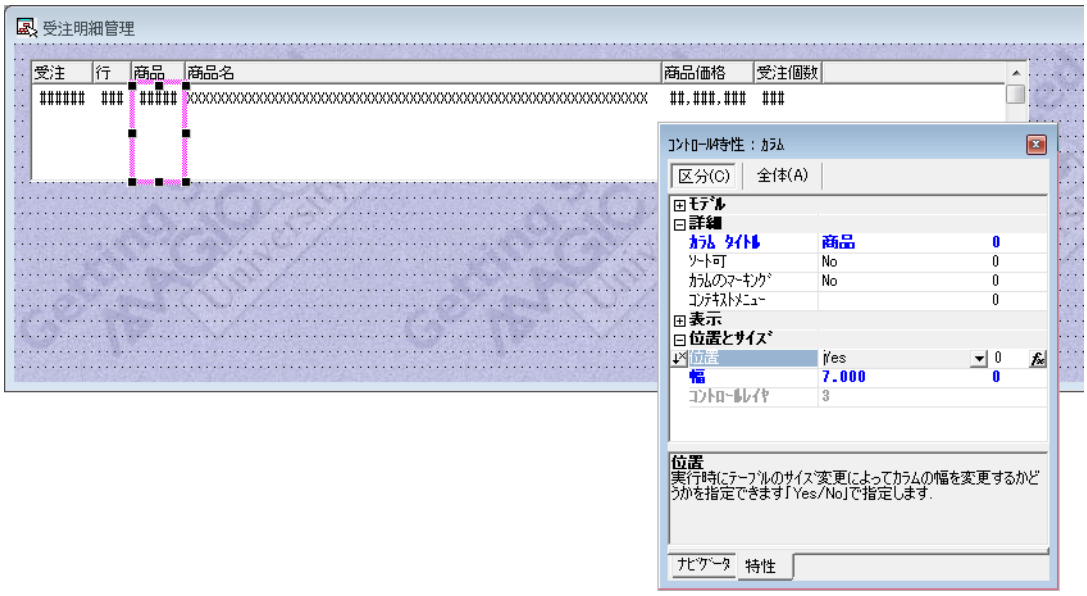
1. Alt キーを押下します。
2. ＜明細行番号＞カラムのテーブル内の白い領域をクリックします。
3. ［カラム特性］（Alt+Enter）を開きます。
4. ［カラムタイトル］特性を**＜行＞**に修正します。

5. ［幅］特性を＜ 5 ＞ に設定します。



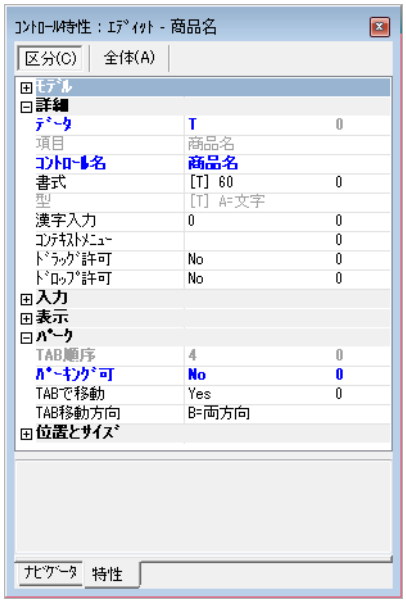
＜商品コード＞カラムの特性設定

1. Alt キーを押下します。
2. ＜商品コード＞カラムのテーブル内の白い領域をクリックします。
3. ［カラム特性］（Alt+Enter）を開きます。
4. ［カラムタイトル］特性を＜商品＞に修正します。
5. ［幅］特性を ＜ 7 ＞ に設定します。



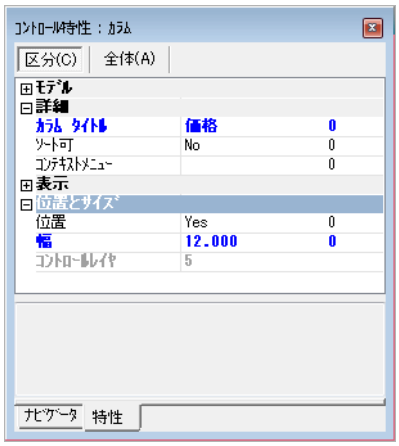
＜商品名＞カラムの特性確認

1. ＜商品名＞［エディット］コントロールの特性（Alt+Enter）を開きます。
2. ［パーキング可］特性が［No］であることを確認してください。
3. ＜商品名＞［エディット］コントロールの特性を閉じます。



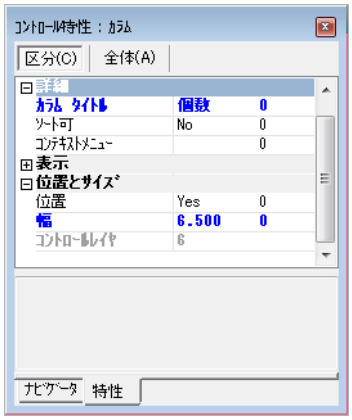
＜商品価格＞カラムの特性設定

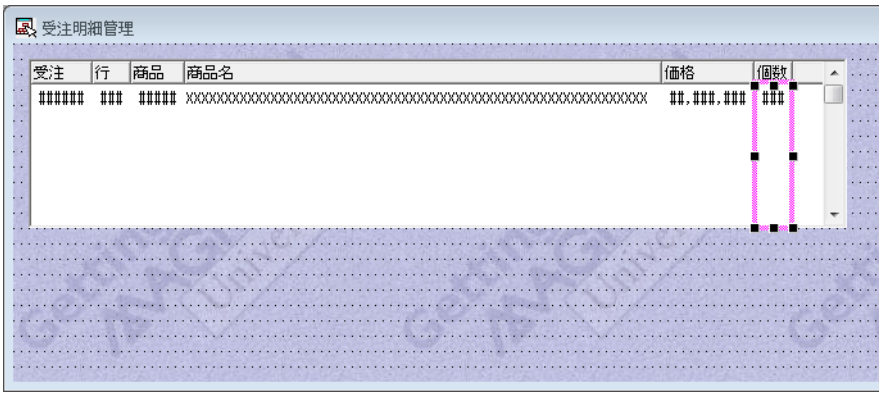
1. Alt キーを押下します。
2. ＜商品価格＞カラムのテーブル内の白い領域をクリックします。
3. ［カラム特性］（Alt+Enter）を開きます。
4. ［カラムタイトル］特性を**＜価格＞**に修正します。



＜受注個数＞カラムの特性設定

1. Alt キーを押下しながら＜受注個数＞カラムのテーブル内の白い領域をクリックします。
2. ［カラム特性］（Alt+Enter）を開きます。
3. ［カラムタイトル］特性を**＜個数＞**に修正します。
4. ［幅］特性を＜ 6.5 ＞に設定します。





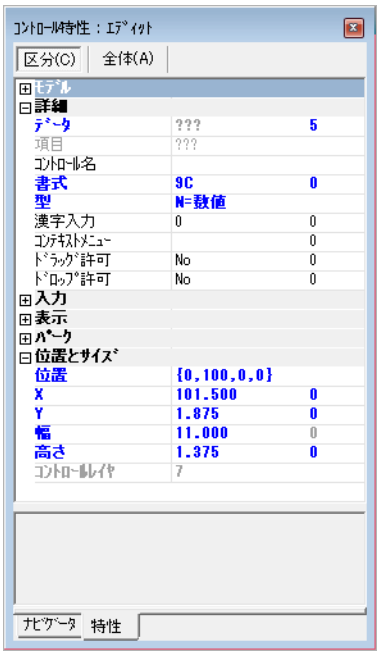
明細行小計カラムの追加

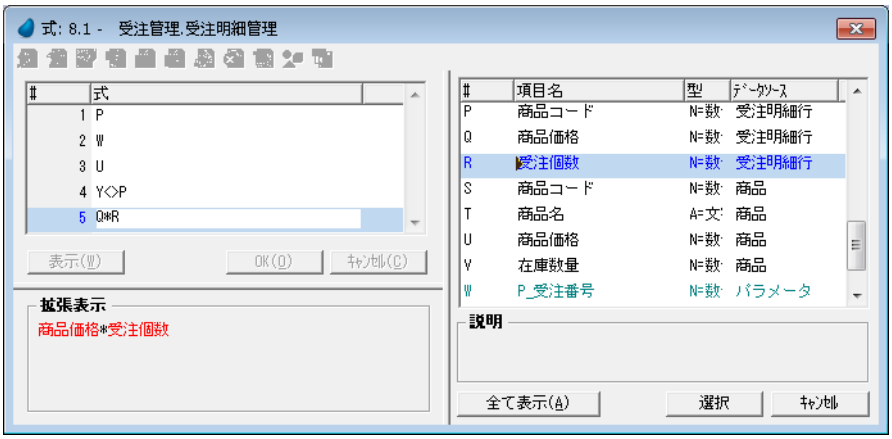
テーブル定義で、各明細行は価格と個数が入力されるようになっていますので、行ごとの小計金額を計算して表示するようにしましょう。

まず、明細行小計を表示するための［エディット］コントロールを追加します。

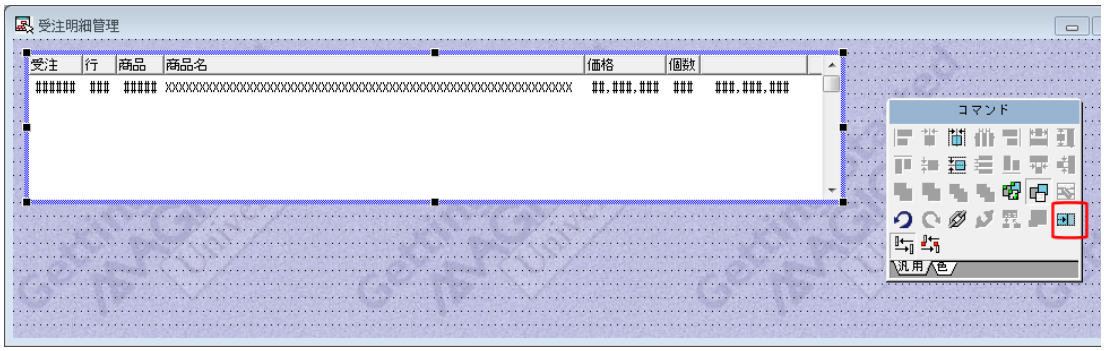
1. ［コントロール］パレットから［エディット］コントロールを選択します。
2. ［テーブル］コントロールに配置します。
3. ［エディット］コントロールの特性（Alt+Enter）を開きます。

4. ［データ］特性の［fx］ボタンを押下して、［式］エディタを開きます。
5. 最終行の次に一行作成します。
6. 式＜ Q*R ＞（商品価格＊受注個数）を設定します。この設定は次のように行います。
   a. ズームして項目リストに移動します。
   b. ＜受注明細行＞データソースの＜商品価格＞（#Q）を選択します。
   c. ＜＊＞(アスタリスク）をタイプ入力します。
   d. もう一度項目リストに移動します。
   e. ＜受注明細行＞データソースの＜受注個数＞（#R）を選択します。
7. ［書式］特性を＜ 9C ＞とします。
8. ［幅］特性を ＜ 11 ＞ に設定します。

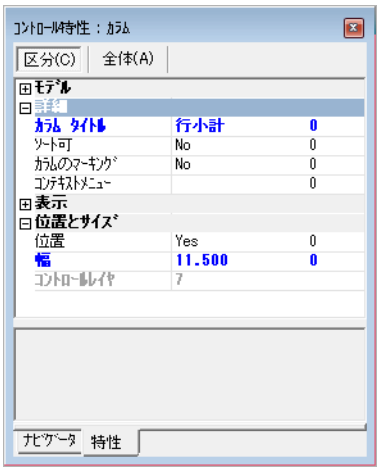


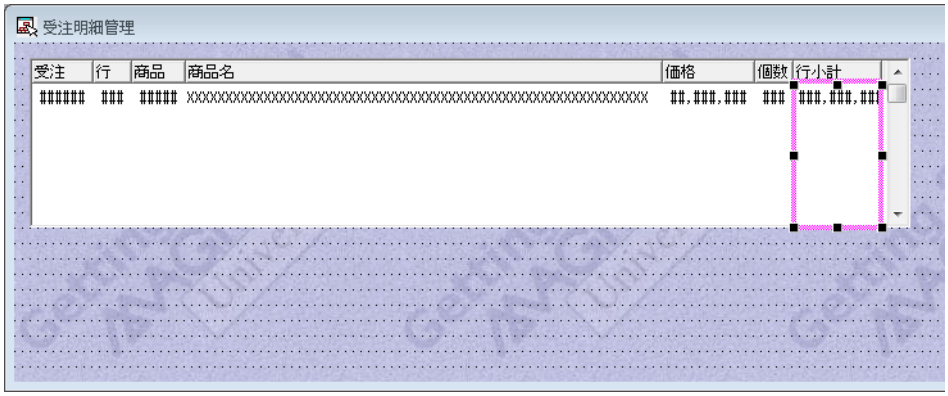


9. ［エディット］コントロール特性を閉じます。
10.Ctrl キーを押下して［テーブル］コントロールの白い領域をクリックし、［テーブル］コントロールを選択します。
11.［コマンド］パレットの［サイズ調整］アイコンをクリックし、テーブルサイズを調整します。



12.Alt キーを押下しながら、先ほど追加した［エディット］コントロールカラムの白い領域をクリックします。
13.［カラム特性］（Alt+Enter）を開きます。
14.［カラムタイトル］特性を<行小計>にします。
15.［幅］特性を< 11. 5 >に設定します。





16.Space キーを押下するか、フォーム上の何もない場所をクリックして、コントロールの選択をクリアします。
17.Ctrl キーを押下しながら、［テーブル］コントロールの白い領域をクリックし、［テーブル］コントロールを選択します。
18.下部中央のハンドルをクリックし、ドラッグしてフォームの下辺までサイズを調整します。

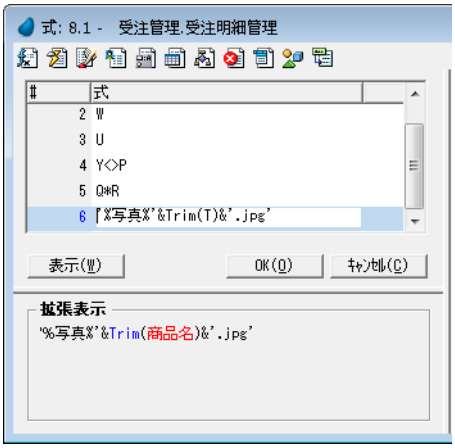
商品イメージの追加

明細行のフォームに商品イメージを表示すれば、エンドユーザは視覚的により分かりやすくなることでしょう。

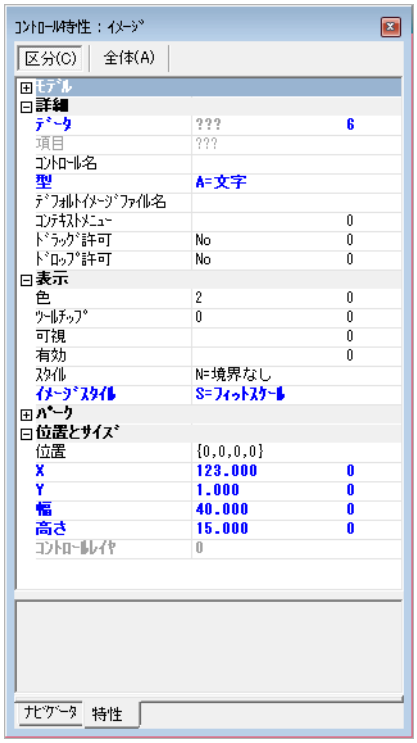
前に＜商品選択＞プログラムを作成したときと同様の方法で、このフォームにもイメージを追加しましょう。

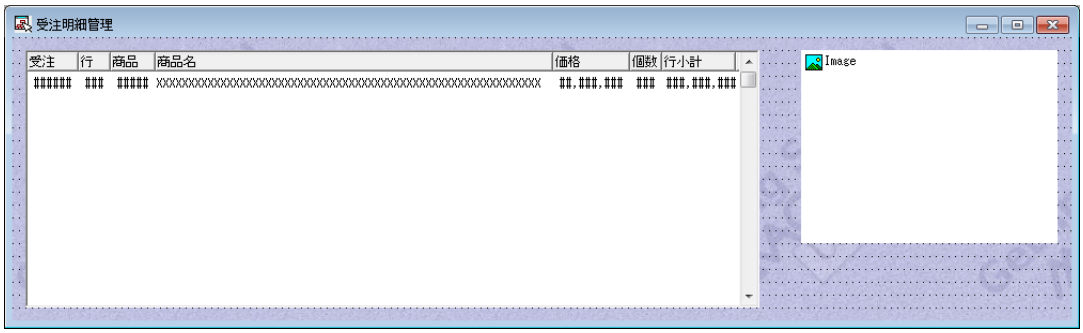
1. ［コントロール］パレットから［イメージ］コントロールを選択し、［テーブル］コントロールの右側に配置します。［ファイルを開く］ダイアログが開きますが、［キャンセル］をクリックして閉じます。
2. ［イメージ］コントロールの特性を開きます。

3. ［データ］特性の［fx］ボタンをクリックし、［式］エディタを開きます。
4. ［式］エディタで最終行にパークし、一行作成します。
5. 式として、＜'%写真%'&Trim(T)&'.jpg'＞を入力します。次のような手順で入力します。
   a. 文字列、＜'%写真%'&＞を入力します。
   b. 関数名＜Trim(＞　までを入力します。
   c. ズームして項目リストに移動します。
   d. ＜商品名＞（#T）を選択します。
   e. 関数を閉じる＜) ＞をタイプします。
   f. 文字列＜ &'.jpg' ＞を入力します。
6. ［OK］ボタンをクリックします。



7. ［スタイル］特性を ［N= 境界なし］とします。
8. ［イメージスタイル］特性を ［S= フィットスケール］とします。
9. ［幅］特性を ＜ 40 ＞とします。
10. ［高さ］特性を ＜ 15 ＞とします。
11. ［イメージ］コントロールの特性を閉じます。
12. フォームの右下角をクリックし、ドラッグしてフォームサイズを調整しましょう。





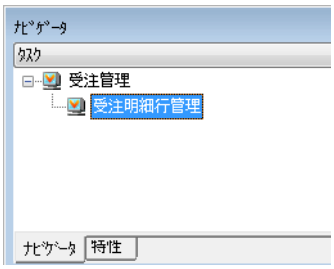
＜受注管理＞フォームへのサブフォーム追加

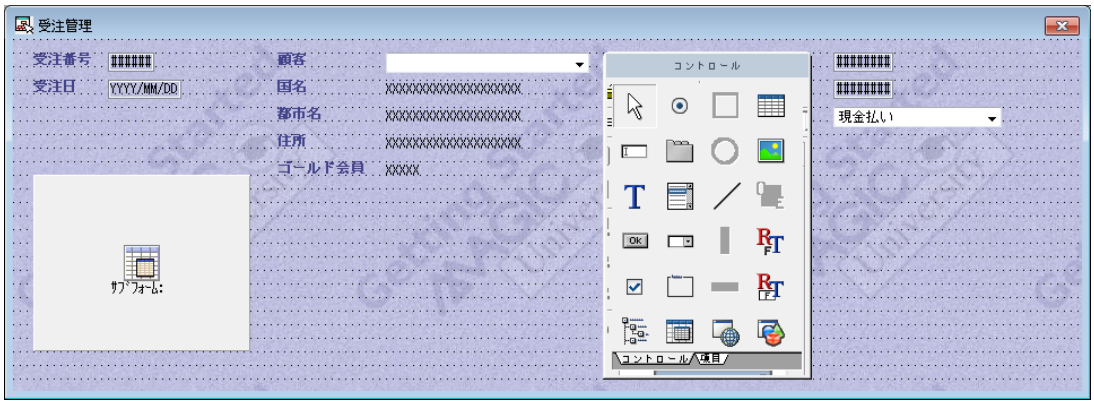
ここまででサブタスクの作成は終わったので、次に親タスクからこれをコールし、そのフォームを親タスク＜受注管理＞のフォーム内に表示させましょう。

Magic xpa では［サブフォーム］コントロールを利用して、サブタスクをコールし、内容の表示と更新を行なうことができます。

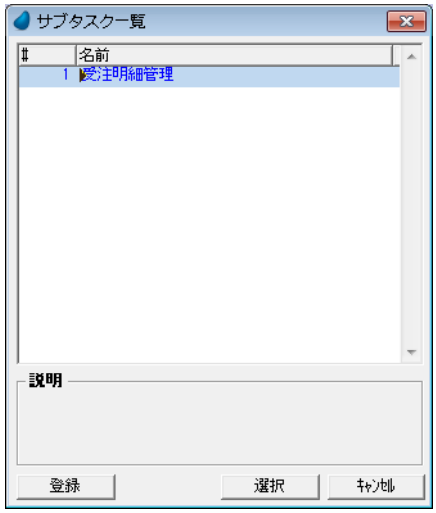
以下のようにして、＜受注管理＞フォームに［サブフォーム］コントロールを配置します。

1. ［ナビゲータ］ペインで＜受注管理＞タスクをクリックします。
2. ［フォーム］エディタで、＜受注管理＞フォームにズームして開きます。
3. ［コントロール］パレットで［サブフォーム］コントロールアイコンをクリックし、フォームに配置します。

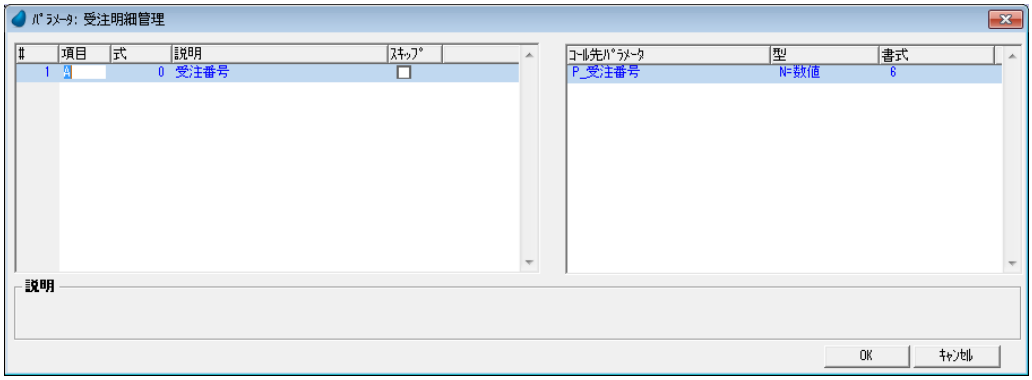


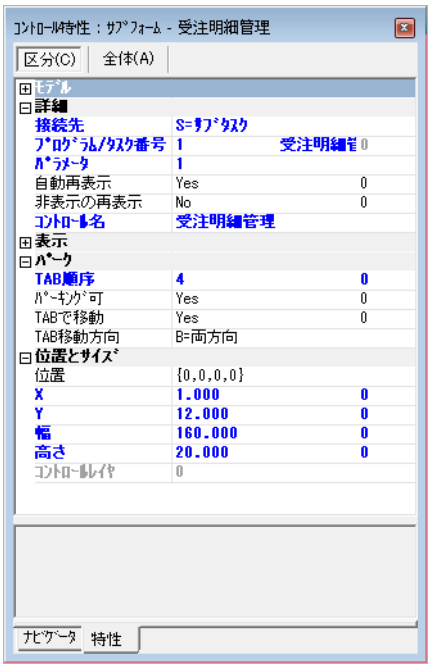


4. ［サブフォーム］コントロールの特性（Alt+Enter）を開きます。
5. ［接続先］特性で［S= サブタスク］を選択します。
6. ［プログラム／タスク番号］特性でズームしサブタスク一覧を表示します。
7. ＜受注明細行管理＞（#1）を選択します。



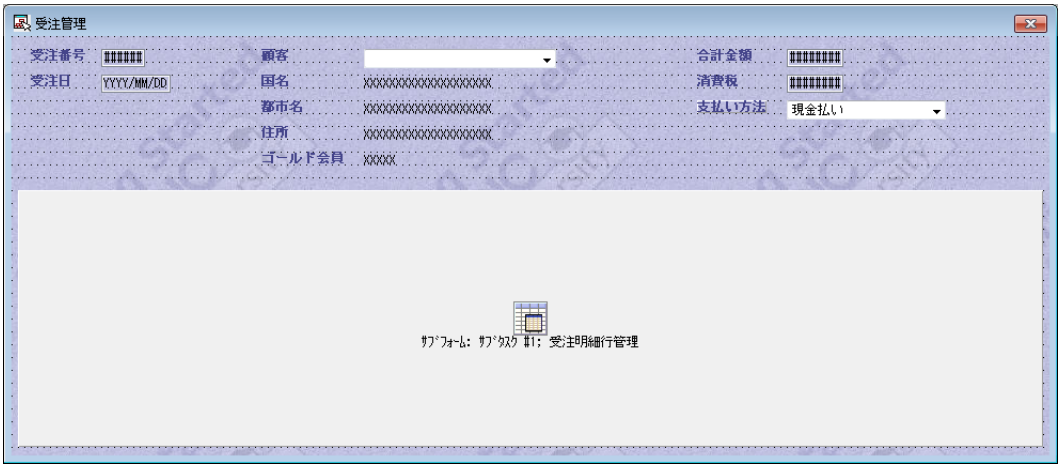
8. ［パラメータ］特性でズームし、［パラメータ］テーブルを開きます。
   a. 一行作成（F4）します。
   b. 項目欄からズームし、項目一覧を表示して＜受注番号＞（#A）を選択します。
9. ［TAB 順序］特性で＜ 4 ＞を設定します。
10. ［幅］特性で＜ 160 ＞を設定します。
11. ［高さ］特性として＜ 20 ＞を設定します。





12.＜受注管理＞フォームの右下ハンドルをクリックし、ドラッグして［サブフォーム］コントロールが全て表示されるようにサイズ調整しましょう。

13.変更内容を保存し、プログラムを閉じます。



［サブフォーム］コントロールについて

［サブフォーム］コントロールを使用することで、他のタスクのフォーム上にタスクのフォームを容易に組み込むことができます。

2 つのタスクが親子関係にある場合などが典型的な適用例になります。

この場合、サブフォームのタスクのロジックやレコードサイクルは親タスクから独立していますが、親タスクのデータビューの値範囲などによる制限は受けます。

サブフォームの利点

- 親タスクのコントロール上にパークした状態で、サブタスクで詳細データを表示する 1 対多の関係のタスクを作成することができます。
- 親タスクからサブタスクへの制御の移動は Magic xpa が自動的に管理します。
- 親タスクからサブタスクに渡すパラメータの内容が変わると、Magic xpa はサブフォームの表示を自動的にリフレッシュします。
- カーソルがサブフォームに出たり入ったりする場合、Magic xpa はその時点でのサブフォーム内の最終位置を記憶します。
- 呼出す親タスクのタブ順序サイクルに［サブフォーム］コントロールを組込むことができます。

- 親タスクからサブフォームへのイベント発動を行なうことができます。

- **［サブフォーム］コントロールは、インタラクティブなリッチクライアントプログラムの表示フォームでのみ使用することができます。**
- **［サブフォーム］コントロールは、バッチタスクやメインプログラムでは使用できません。**

＜受注管理＞プログラムの実行

ここまでの作業で、＜受注明細行管理＞サブタスクを作成し、そのフォームを＜受注管理＞フォーム内に組込みました。

これで 1 対多のデータリレーションを取り扱うプログラムが完成したことになります。

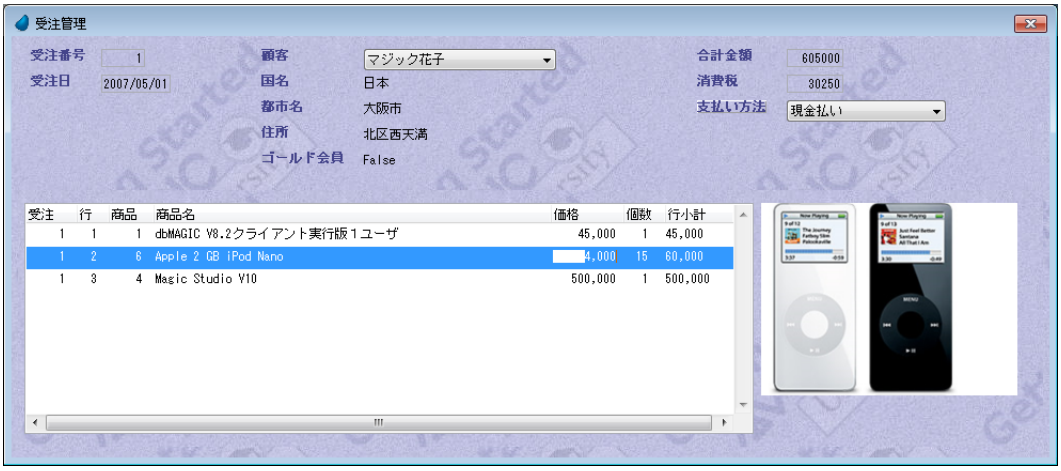
これから＜受注管理＞プログラムを動かして、1 対多のデータリレーションがどのようなものか確認しましょう。

1. ［プログラム］リポジトリで＜受注管理＞プログラム（#8）にパークし、F7 キーを押下してプログラムを実行しましょう。
2. Tab キーを押下して、［サブフォーム］コントロール内にカーソルを移動させます。
3. 下記のように商品を追加してみましょう。

受注	行	商品	商品名	価格	個数	行小計
1	1	1	dbMAGIC V8.2 クライアント実行版 1 ユーザ	45,000	1	(45,000)
1	2	6	Apple 2 GB iPod Nano	4,000	15	(60,000)

- 商品の選択は、＜商品＞欄からズームして商品選択プログラムを起動して行います。
- ＜価格＞はデフォルトで表示された値を修正してください。
- ＜受注＞カラム（受注番号）はパーク不可の設定です。
- サブフォームに表示される＜受注＞カラムによって、データの整合が取れていることが分かります。
- 明細行レコードが作成される都度、受注番号データは自動的に設定されます。
- 親タスクのレコードの受注番号と同じ受注番号を持つ受注明細行レコードのみがサブフォームに表示されています。

4. ＜受注管理＞フォーム上で、＜合計金額＞コントロールをクリックしてください。
5. 値として、＜ 105000 ＞を入力してください。
6. Tab キーを押下して＜消費税＞コントロールに移動してください。
7. 値として、＜ 5250 ＞を入力してください。（合計金額の 5% と仮定しています。）
8. Tab キーを押下して＜支払い方法＞コントロールに移動してください。
9. ＜現金払い＞を選択しましょう。

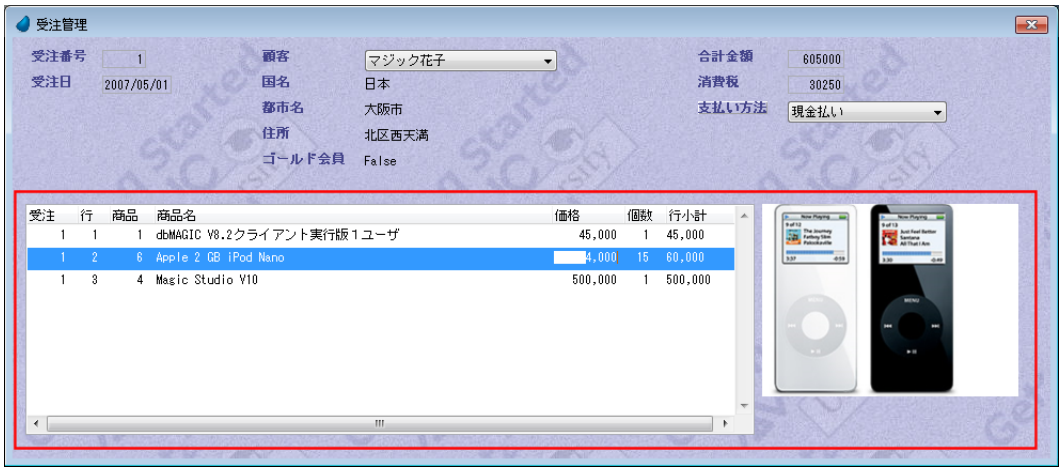


いくつか改良点が明らかになってきました。以後、順に検討してゆきます。

サブタスクフォームの透過性

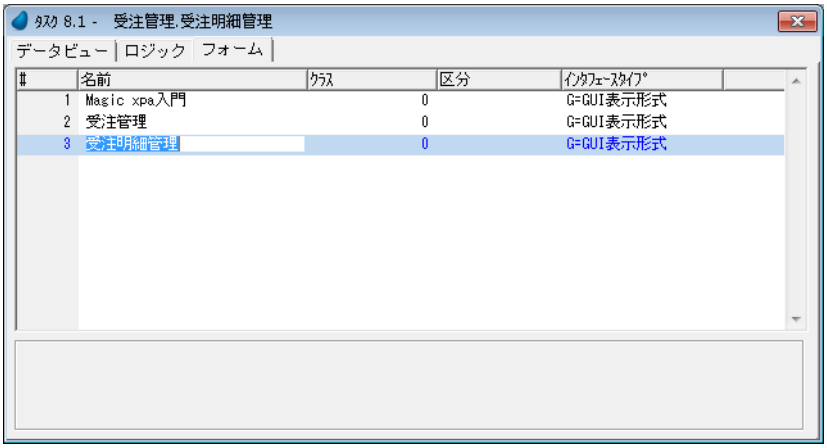
できあがったフォームをよく見てみましょう。

このフォームは実際には二つの背景が不連続に組み合わさっています。つまり＜受注管理＞と＜受注明細行管理＞の各フォーム背景です。もしプレーンな背景であれば気がつきませんが、今回はコースで用意したロゴ文字の入っている背景のため、よく見ると分かります。

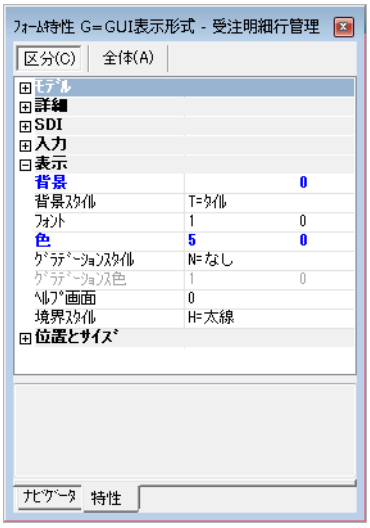


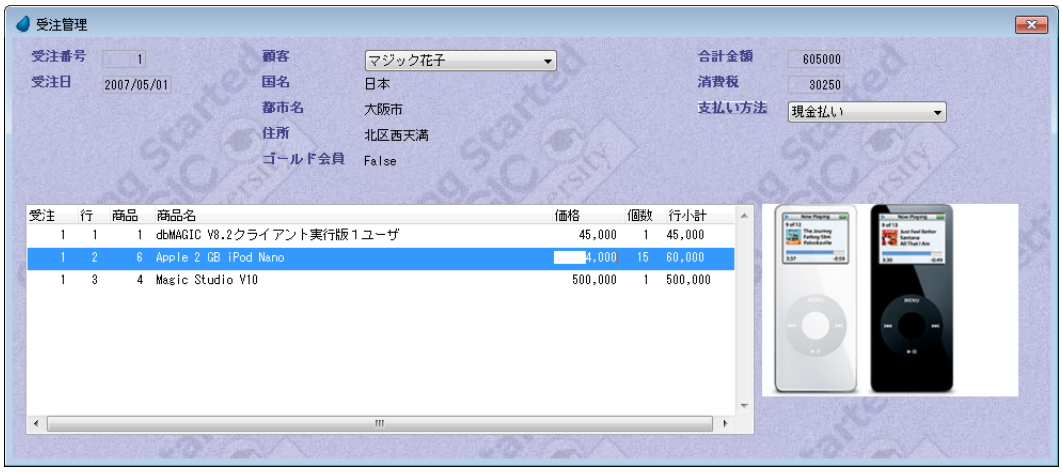
この問題を解決するためには、サブタスクのフォームに透過色を設定し、［背景］特性をクリアします。

1. 実行中のプログラムを終了してください。
2. ［プログラム］リポジトリで、＜受注管理＞プログラム（#8）にズームします。
3. ［ナビゲータ］ペインで、＜受注明細行管理＞を選択します。
4. ［フォーム］エディタを選択し、＜受注明細行管理＞フォーム行（#3）にパークします。



5. フォーム特性（Alt+Enter）を開きます。
6. ［背景］特性をクリアします。
7. ［色］特性を＜ 5 ＞に設定します。
8. 変更内容を保存し、プログラムを閉じます。
9. ＜受注管理＞プログラムを実行しましょう。
10. 結果を確認してください。





合計金額の計算

加算更新

最初の受注登録では、合計金額を手入力しなければなりませんでした。

これは、実際の運用プログラムとしては不十分な仕様と言わざるを得ません。

Magic xpa はこのような計算をするために、［加算更新］という便利な機能を持っています。

このセクションでは、この［加算更新］の機能について説明します。

Magic xpa では項目の値をインクリメンタルに更新することができます。つまり更新式の内容が評価され、次のルールにしたがって更新項目から加算あるいは減算されるようになります。

- ［登録］モードで新しくレコードが追加されたとき、更新式の値が更新項目に加算されます。
- ［削除］モードでレコードが削除されたとき、更新式の値が更新項目から減算されます。
- ［修正］モードでレコードが修正されたとき、更新項目から前の更新式の値を減算した後、新しい更新式の値を加算します。

［加算更新］の機能は、インタラクティブなタスクでのみ使用することができます。

次のセクションでは＜合計金額＞を自動的に計算するようにプログラムを改良し、＜合計金額＞［エディット］コントロールにはパークできないように変更することで、エンドユーザの手入力による設定ができないようにします。

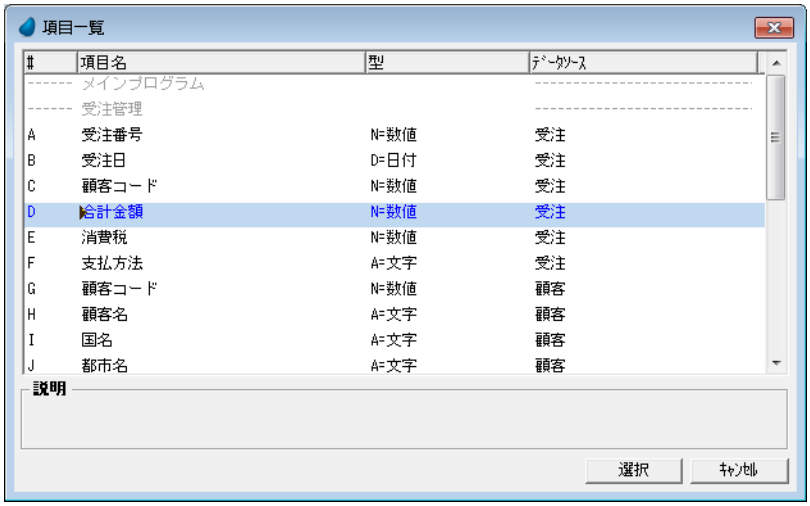
合計金額の項目更新

1. 実行中のプログラムを終了してください。
2. ［プログラム］リポジトリで、＜受注管理＞プログラム（#8）にズームします。
3. ［ナビゲータ］ペインで、＜受注明細行管理＞を選択します。
4. ［ロジック］エディタを選択します。
5. 最後の行にパークします。
6. ヘッダ行（Ctrl+H）を一行作成します。
7. ［レコード後］ロジックユニットを設定します。
8. 一行作成（F4）します。
9. 次のように設定します。

コマンド	項目	代入	条件
項目更新	D （合計金額）	5 （商品価格＊受注個数）	Yes

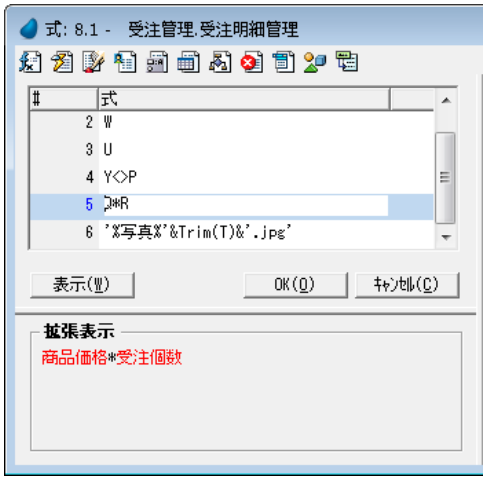
項目の設定

10.ズームして項目一覧を表示します。
11.＜合計金額＞（#D）を選択します。

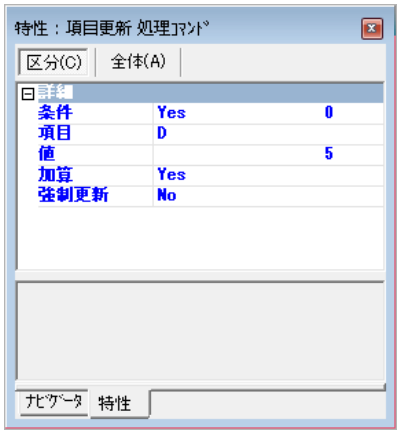


代入式の定義

12.ズームして［式］エディタを表示します。
13.式番号＜ 5 ＞を選択します。
14.［OK］ボタンをクリックします。



15.［項目更新］処理コマンドの特性（Alt+Enter）を開きます。
16.［加算］特性を［Yes］にします。

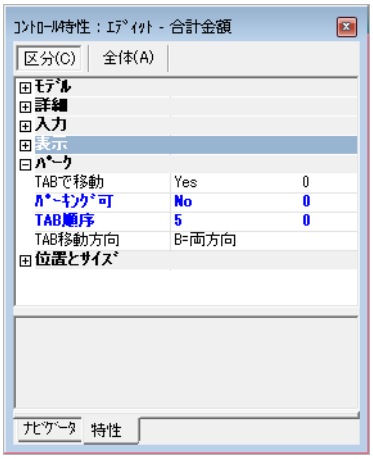


＜合計金額＞［エディット］コントロールへのパーク不可設定

＜合計金額＞は Magic xpa により自動的に計算されるように設定したので、エンドユーザが手入力で変更できないようにしましょう。このためには、＜合計金額＞［エディット］コントロール特性の［パーキング可］特性を［No］に設定します。

1. ［ナビゲータ］ペインで、＜受注管理＞タスクをクリックします。
2. ［フォーム］エディタを選択します。
3. ＜受注管理＞フォームにズームします。

4. ＜合計金額＞［エディット］コントロールをクリックして選択します。
5. コントロール特性（Alt+Enter）を開きます。
6. ［パーキング可］特性を［No］にします。
7. 変更内容を保存し、プログラムを閉じます。



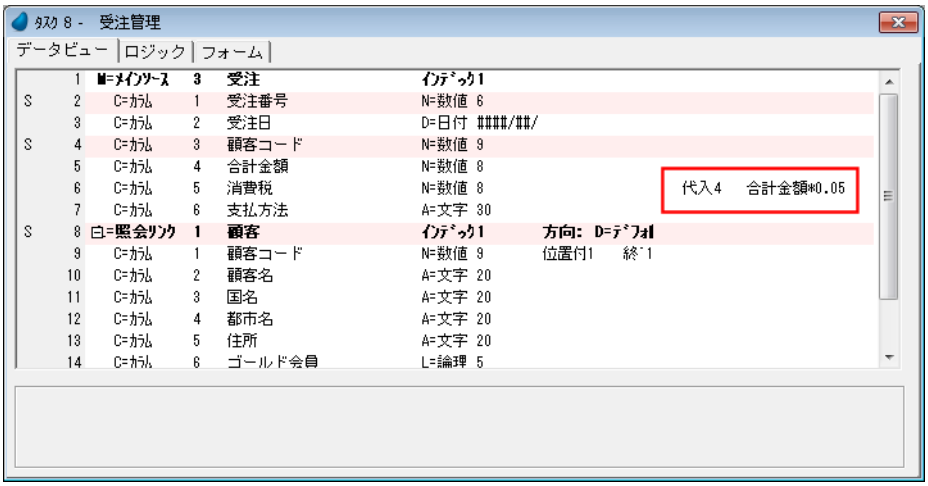
＜消費税＞ 項目の自動再計算

＜合計金額＞項目が変更されたとき、＜消費税＞項目も自動的に更新されると便利です。ここでは代入式を設定して、＜消費税＞項目が自動再計算されるようにしましょう。

1. ［ナビゲータ］ペインで、＜受注管理＞タスクをクリックします。
2. ［データビュー］タブをクリックします。
3. ＜消費税＞を定義している［C= カラム］コマンドの行にカーソルを移動します。

4. ［代入：］欄からズームし、式テーブルを開きます。
5. 最終行で F4 キーを押下して、新規行を作成します。
6. 次のように入力します。：＜ D * 0.05 ＞
7. ［OK］キーを押下して、式テーブルを閉じます。

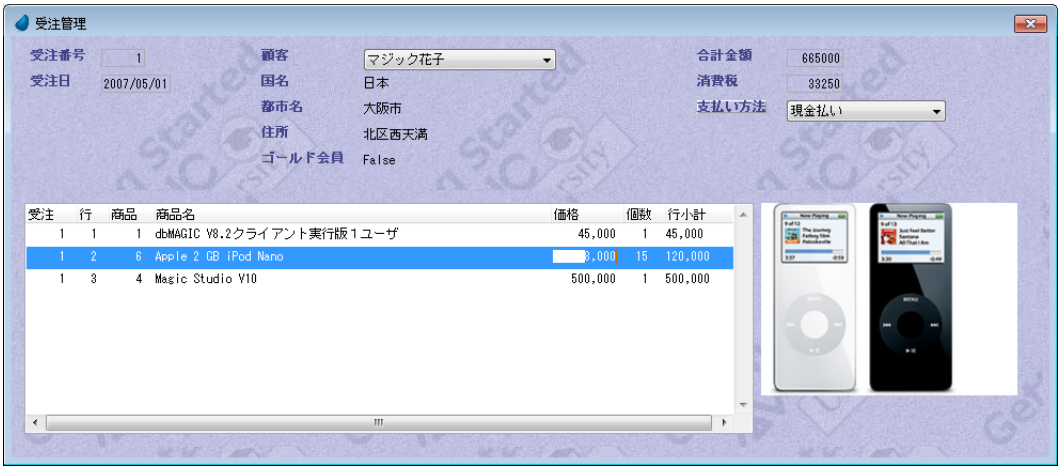
データビューは次のようになります。



＜受注管理＞プログラムの動作確認

1. ＜受注管理＞プログラムを起動（F7）しましょう。
2. Tab キーを押下して、受注明細行のサブフォームにカーソルを移動してください。
3. 最後の行にパークし、一行受注明細行を追加（F4）します。

4. ＜行＞カラムに＜ 3 ＞と入力します。
5. ＜商品＞カラムからズームして、＜ Magic Studio V10 ＞を選択します。（商品コード 4 番）
6. ＜個数＞カラムは＜ 1 ＞とします。
7. ↑キーを押下して、2 行目に移動ください。
8. ＜合計金額＞が＜ 705,000 ＞に更新されたことを確認してください。これはそれまでの合計金額 105,000 に、追加された 3 行目の価格 600,000 が加算された結果です。
9. もう一度 3 行目にパークしてください。
10. ＜価格＞を＜ 500, 000 ＞に変更してみましょう。40
11. ↑キーを押下して、2 行目に移動ください。
12. ＜合計金額＞が更新されて、＜ 605,000 ＞になりました。これはそれまでの合計金額 705,000 に、3 行目の価格変更分（500,000-600,000=-100,000）が加算された結果です。
13. もう一度 3 行目にパークしましょう。
14. F3 キーを押下するか［編集］メニューから［行削除］を選択して、3 行目を削除します。
15. ［確認］ダイアログで、［はい］を選択します。
16. ＜合計金額＞を確認し、＜ 105,000 ＞となっていることを確認しましょう。これは、それまでの合計金額 705,000 から 3 行目の価格 500,000 を差し引いた結果です。
17. プログラムを終了しましょう。



５．練習問題

国名と都市名とは 1 対多のデータリレーションがあります。各国には通常多くの都市があるからです。

練習問題として、これまで学習したことを応用し、国名とその国の都市名とを一覧表示させるプログラムを作成しましょう。

国名データソースの定義

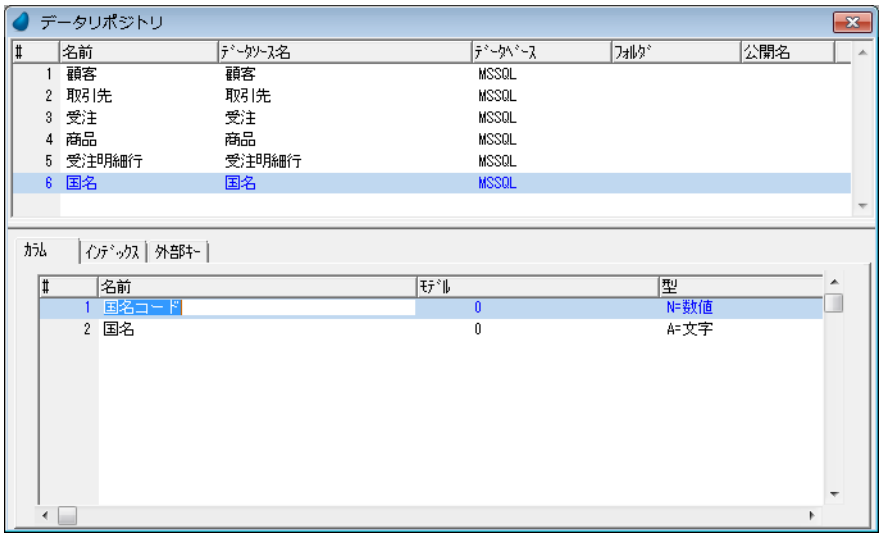
1. ［データ］リポジトリ（Shift+F2）を開きます。
2. 最後の行にパークし、F4 キーを押下して一行作成します。
3. ［名前］として＜国名＞と入力してください。

4. ［データソース名］として、同様に＜国名＞としてください。
5. ［データベース］からズームし、［MSSQL］を選択してください。

カラムの定義

次のようにカラムを定義してください。

#	名前	モデル	型	書式
1	国名コード	0	N= 数値	6
2	国名	0	A= 文字	30

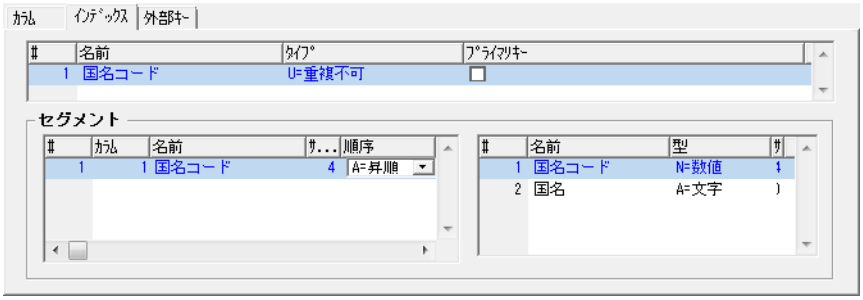


インデックスの定義

1. 下部ペインの［インデックス］タブをクリックします。
2. F4 キーを押下して一行作成してください。
3. 次のインデックスを定義します。
   - ［名前］…… ＜国名コード＞
   - ［タイプ］…… ［U= 重複不可］

4. ズーム（F5）して［セグメント］テーブルに移動します。
5. F4 キーを押下して一行作成します。
6. ［カラム］からズームして右側の項目一覧テーブルに移動します。
7. ＜国名コード＞（#1）を選択します。（Enter キーで選択）

8. Enter キーを 2 回押下して、上部ペインに戻りましょう。



都市名データソースの定義

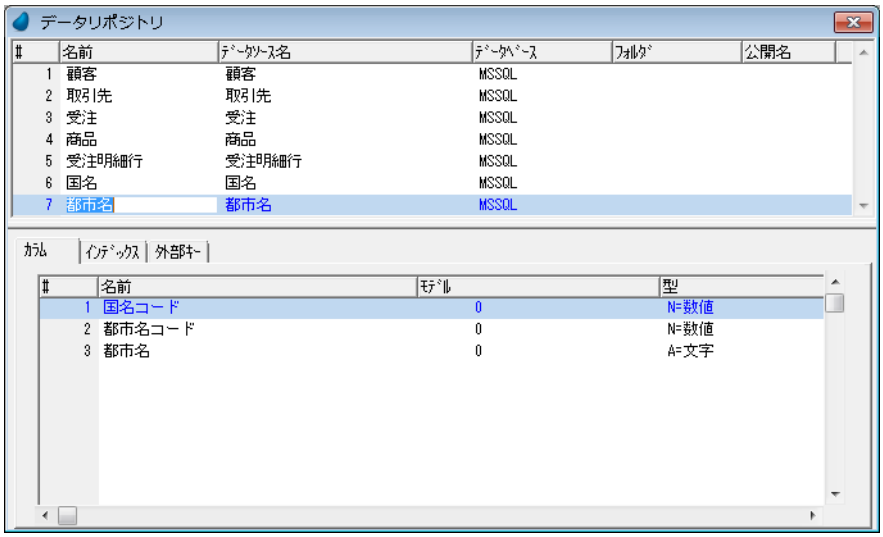
1. ［データ］リポジトリ（Shift+F2）を開きます。
2. 最後の行にパークし、F4 キーを押下して一行作成します。
3. ［名前］として**＜都市名＞**と入力してください。

4. ［データソース名］として、同様に**＜都市名＞**としてください。
5. ［データベース］からズームし、［MSSQL］を選択してください。

カラムの定義

次のようにカラムを定義してください。

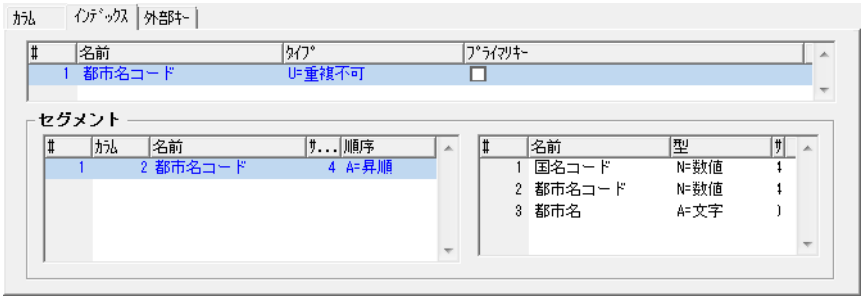
#	名前	モデル	型	書式
1	国名コード		N= 数値	6
2	都市名コード	0	N= 数値	6
3	都市名	0	A= 文字	30



インデックスの定義

1. 下部ペインの［インデックス］タブをクリックします。
2. F4 キーを押下して一行作成してください。
3. 次のインデックスを定義します。
   - ［名前］…… ＜都市名コード＞
   - ［タイプ］…… ［U= 重複不可］

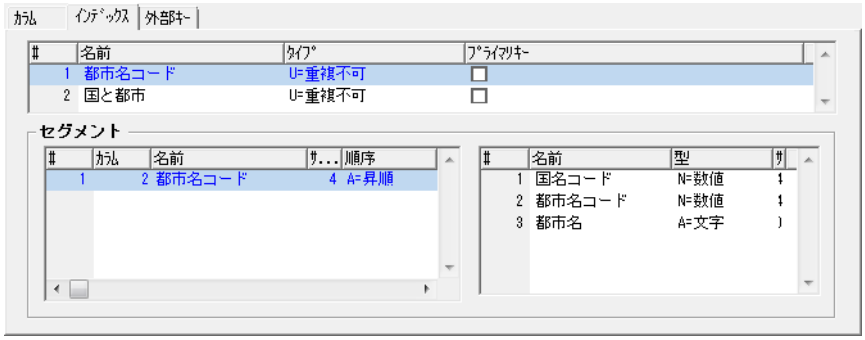
4. ズーム（F5）してセグメントテーブルに移動します。
5. F4 キーを押下して一行作成します。
6. ［カラム］からズームして右側の項目一覧テーブルに移動します。
7. ＜都市名コード＞（#2）を選択します。（Enter キーで選択）
8. Enter キーを 2 回押下して、上部ペインに戻りましょう。



2 番目のインデックス定義

1. 下部ペインの［インデックス］タブをクリックします。
2. #1 行めにパークし、F4 キーを押下して新たに一行作成してください。
3. 次のインデックスを定義します。
   - ［名前］…… ＜国と都市＞
   - ［タイプ］…… ［U= 重複不可］

4. ズーム（F5）して［セグメント］テーブルに移動します。
5. F4 キーを押下して一行作成します。
6. ［カラム］からズームして右側の項目一覧テーブルに移動します。
7. ＜国名コード＞（#1）を選択します。（Enter キーで選択）
8. F4 キーを押下してもう一行作成します。
9. ［カラム］からズームして右側の項目一覧テーブルに移動します。
10. ＜都市名コード＞（#2）を選択します。（Enter キーで選択）
11. Enter キーを 2 回押下して、上部ペインに戻りましょう。



＜国名一覧＞プログラムの作成

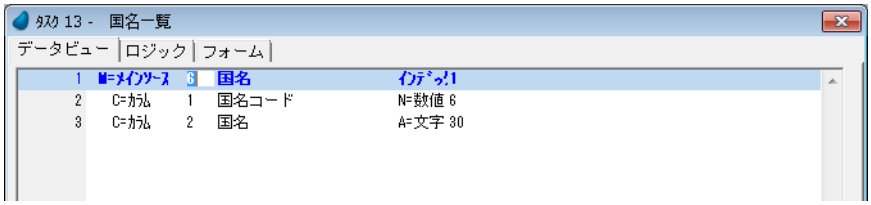
データビューの設定

1. ［プログラム］リポジトリを開きます。
2. 最終行にパークし、F4 キーを押下して新たに一行作成します。
3. ［名前］として、**＜国名一覧＞**とします。

4. ズーム（F5）してプログラムを開きます。
5. タスク特性は［タスクタイプ］を「C= リッチクライアント」に設定し、［OK］ボタンをクリックします。

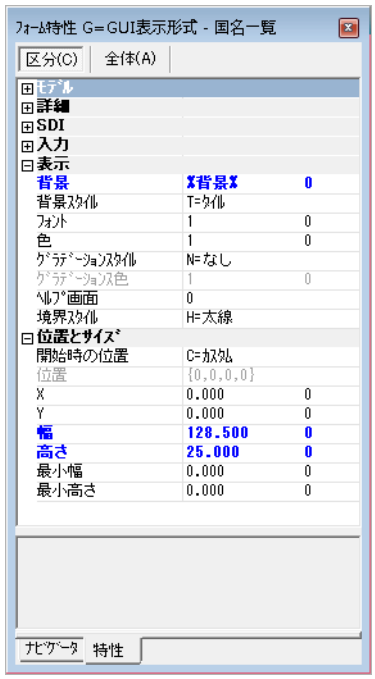
［データビュー］エディタで次の作業をしましょう。

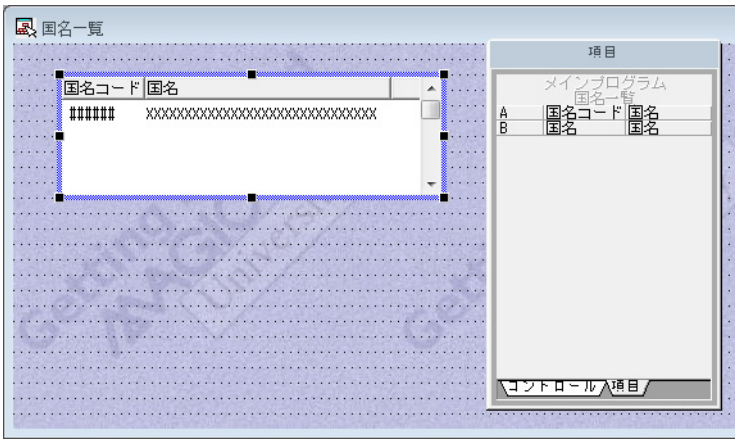
6. メインソースとして、＜国名＞（#6）を選択します。
7. インデックスは＜国名コード＞（#1）になります。
8. 2 つのカラムを定義してください。



フォームのデザイン

1. ［フォーム］エディタを選択します。
2. ＜国名一覧＞（#2）行にパークします。
3. フォーム特性を開きます。
4. ［背景］特性として **＜％背景％＞** と入力してください。
5. ［幅］特性として **＜ 128.5 ＞** と入力してください。
6. ［高さ］特性として**＜ 25 ＞** と入力してください。
7. ＜国名一覧＞フォームにズームします。
8. ［テーブル］コントロールをフォームに配置します。
9. ＜国名コード＞と＜国名＞項目をデータカラムとして定義します。下のイメージを参考にしてください。
10. ［フォーム］エディタを閉じましょう。

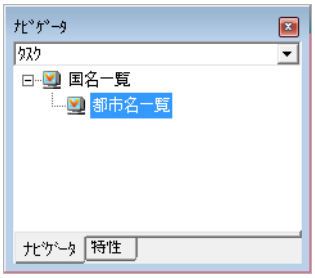




サブタスク＜都市名一覧＞の作成

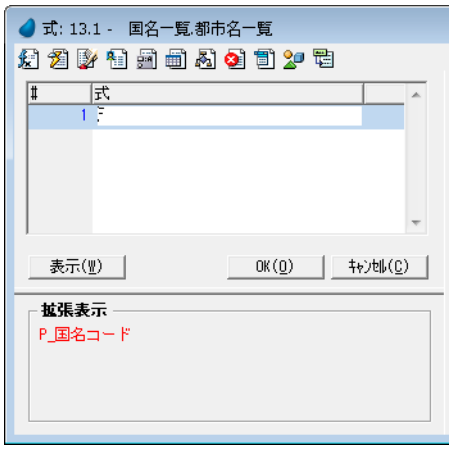
1. ［表示］メニューから［ナビゲータ］を選択し、［ナビゲータ］ペインを表示させます。
2. ［ナビゲータ］ペインで＜国名一覧＞タスクアイコンをクリックして、パークします。
3. F4 キーを押下してサブタスクを作成します。タスク特性が開くので、次の設定をします。

4. ［タスク名］として、**＜都市名一覧＞** と入力します。
5. ［OK］ボタンをクリックします。



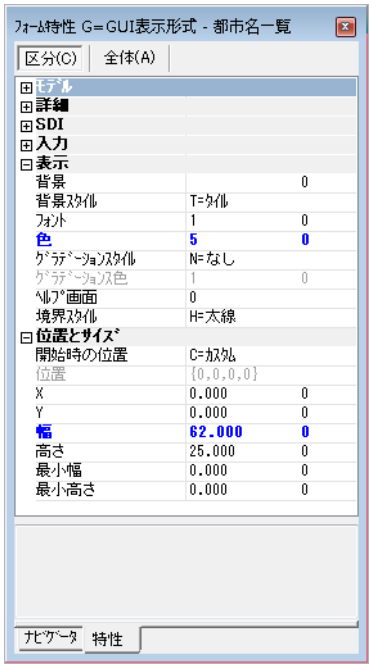
サブタスクのデータビュー設定

6. メインソースとして＜都市名＞（#7）を選択します。
7. インデックスとして、＜国と都市＞（#2）を選択します。
8. 3 つのカラムをすべて定義します。
9. 最終行（#4）にパークし、F4 キーを押下して新たに一行作成します。
10. パラメータ項目を定義し、名前を**＜ P_ 国名コード＞**、型を［N= 数値］、書式を**＜ 6 ＞**とします。
11. ＜国名コード＞（#2）行にパークします。
12. ［範囲］の式番号欄からズームし［式］エディタを開いて一行作成し、＜ P_ 国名コード＞を指定するようにします。［終了］欄にも同じ式番号を設定します。
13. ［代入］欄にも同じ式番号を設定します。

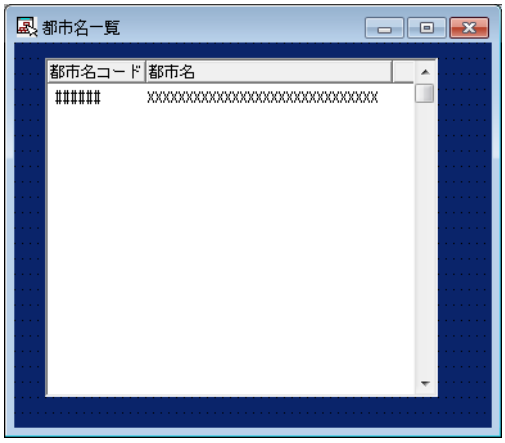


サブタスクのフォームデザイン

14.［フォーム］エディタを選択します。
15.＜都市名一覧＞（#3）フォームにパークします。
16.フォーム特性（Alt+Enter）を開きます。
17.［色］特性を＜ 5 ＞に設定します。この設定は背景が透過色設定になっています。
18.［幅］特性を ＜ 62 ＞ とします。
19.［高さ］特性を ＜ 25 ＞ とします。



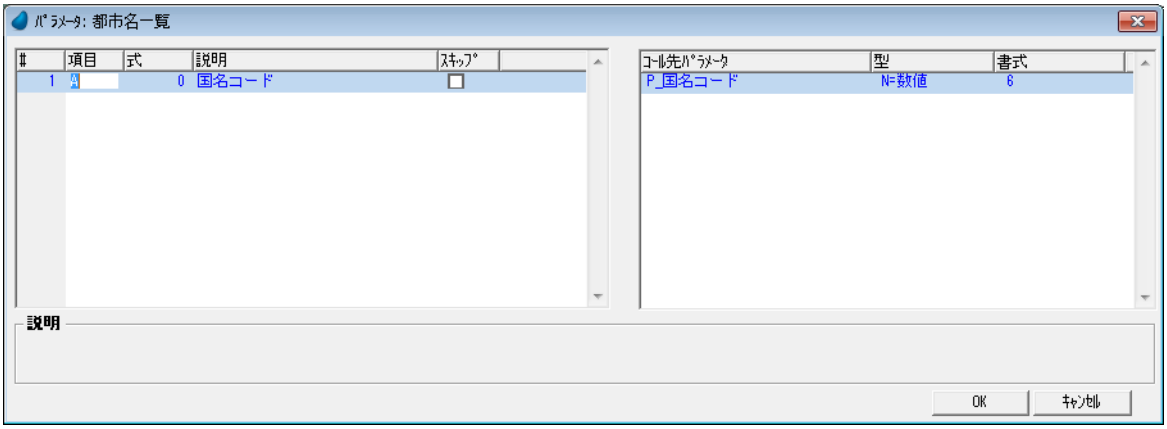
20.＜都市名一覧＞フォームにズームします。
21.［テーブル］コントロールをフォームに配置します。
22.＜都市名コード＞と＜都市名＞の各カラムをテーブルに定義します。
23.テーブルとフォームのサイズ調整を行ない、イメージのように見えるようにしてください。



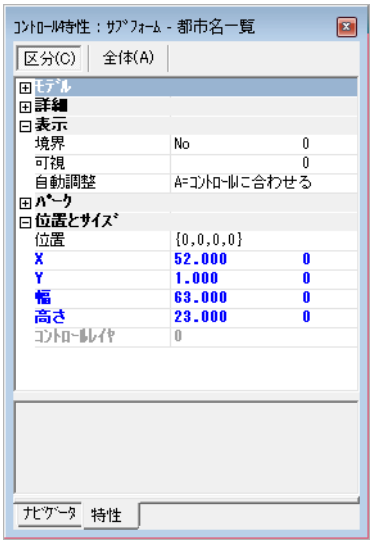
親タスク＜国名一覧＞のフォームデザイン

1. ［ナビゲータ］ペインで＜国名一覧＞アイコンを選択します。
2. ［フォーム］エディタを選択し、＜国名一覧＞フォームにズームします。
3. ［サブフォーム］コントロールを配置します。

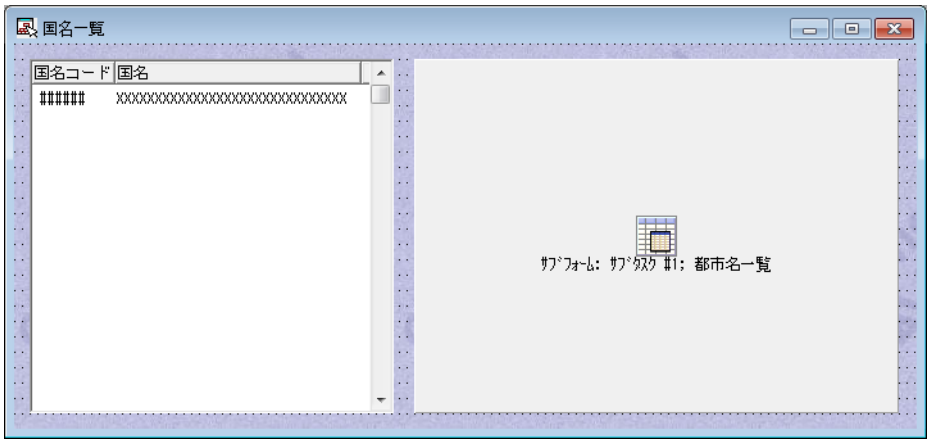
4. ［サブフォーム］コントロール特性を開きます。
5. ［接続先］特性で［S= サブタスク］を指定します。
6. ［プログラム／タスク番号］特性でズームし、サブタスク＜都市名一覧＞を選択します。
7. ［パラメータ］特性でズームし、［式］エディタで次のように設定します。
   a. 一行作成します。
   b. ズームして項目リストに移動します。
   c. ＜国名コード＞を選択します。



8. ［幅］特性に＜ 63 ＞ を入力します。
9. ［高さ］特性に＜ 25 ＞ を設定します。
10. ［テーブル］コントロールおよびフォームのサイズ調整をしてください。
11. 変更内容を保存し、プログラムを閉じます。



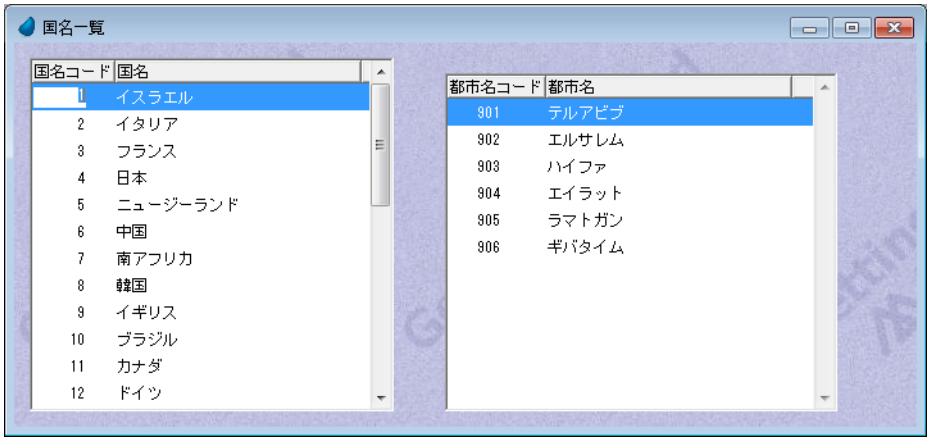
フォームがイメージのように見えるようにしましょう。



＜国名一覧＞プログラムの実行

1. ［プログラム］リポジトリで＜国名一覧＞（#13）にパークし、F7 キーを押下して実行しましょう。
2. 適当に国と都市とを登録してください。

3. カーソルを動かして、国名と都市名が連動して表示されることを確認しましょう。



６．要約

本章では、１対多のデータリレーションについて学習しました。

１対多のデータリレーションを利用して、データソース定義の効率や保守性を高めることができることが分かりました。

また実際のプログラムを作成し、親タスクとサブタスクとで１対多のデータリレーションを実現しました。

例として作成したプログラムは、次のような特徴があります。

- 親タスクとサブタスクを定義しました。
- データの整合性を維持するために、［範囲］と［代入］の各特性を設定しました。
- サブタスクのフォームを親タスクのフォームに統合するために［サブフォーム］コントロールを使用しました。
- 画面の表示更新は、［サブフォーム］コントロールで接続しているサブタスクへパラメータを渡すことで行なわれるようになっています。

本章でインタラクティブなリッチクライアントタスクに関する説明は終わります。

次章からは非インタラクティブタスク（バッチタスク）について学びます。

第 18 章 非インタラクティブタスクのアプリケーションエンジン

本章では、非インタラクティブタスクにおいて Magic エンジンがどのように動作し、どのような利点と使用法があるかについて学習します。

キーワード

- タスクタイプ
- バッチ
- タスク終了条件
- メインソース
- タスクモード
- データ整合性

学習目標

本章を学ぶことで、次の内容に対する理解を深めましょう。

- 非インタラクティブタスクでの Magic エンジン動作
- インタラクティブタスクと非インタラクティブタスクの違い
- どのような場合に非インタラクティブタスクを使ったらよいか
- Magic エンジンはどのような手順で非インタラクティブタスクを処理するか
- 非インタラクティブタスクを使用してデータ整合性を維持する方法

参照

Magic ヘルプの［Magic xpa リファレンス／ Magic エンジン］

１．はじめに

これまでの章ではインタラクティブ（対話型）なプログラムについて学びました。一方、実際の運用を考えたときに、ユーザの介入なしで処理を実行するプログラムも必要になります。

Magic xpa ではこのようなプログラムをバッチタスク（プログラム）または、非インタラクティブタスク（プログラム）と呼んでいます。

これからの章では、この非インタラクティブタスクの概念や動作について学んで行きます。また実習を通して、この非インタラクティブタスクの作成法を学びます。

非インタラクティブタスクはデータソースから読込んだ一まとまりのレコードを、順にまとめて処理するときに使用します。

また非インタラクティブタスクは特定のデータソースを定義しないで、所定のプロセスを処理する目的で使用することがあります。

非インタラクティブタスクを使用する例としては、帳票印刷、レコード一括更新、一括削除、集計計算などがあります。

非インタラクティブタスクでもフォームを定義し、実行時に表示させることができますが、多くの場合エンドユーザに対する表示なしで処理を実行します。

本章では非インタラクティブタスクの実行エンジン動作について学びます。また非インタラクティブタスクをどういう場合に使用するべきかを検討し、簡単な非インタラクティブタスクを作成してみます。

注意

RIA における非インタラクティブタスクには、以下の２種類があります。

- バッチタスク …… サーバで実行されます。
- 非インタラクティブなリッチクライアントタスク …… クライアントで実行されます。

本書では、まとめて「非インタラクティブタスク」として説明しています。

２．インタラクティブタスクと非インタラクティブタスクの特徴比較

インタラクティブタスク	非インタラクティブタスク
エンドユーザとの対話処理	対話処理はない
エンドユーザはメインソースのレコードを参照できる	範囲条件内のメインソースレコードを参照できる
スクロールできるレコードだけが参照できる	メインソースが定義されていないか、タスクが登録モードの場合、タスク終了条件が成立するまでタスクはループする
コントロールロジックユニットを利用できる	コントロールロジックユニットは利用できない
カーソルのコントロール間移動によりロジックが処理される	エンドユーザの対話処理がなく、カーソルはコントロールにパークしない
グループロジックユニットは利用できない	バッチタスクはグループロジックユニットが利用でき、データをグループ単位で取り扱うことができる (非インタラクティブなリッチクライアントタスクは、利用できません)
タスクフォームはプログラム作成と実行において不可欠な要素	タスクフォームは不可欠ではなく、多くの場合使用されない
データが変更されたとき、［レコード後］ロジックユニットが起動され、データは保存される	レコードは変更の有無に関係なく常に保存され、［レコード後］ロジックユニットは常に起動される
イベントはタスクがアイドル状態のとき処理される	イベントは一定周期ごと、または処理されたレコードが一定数になる都度、処理される

３．非インタラクティブタスクのエンジン処理フロー

エンジンの処理フローは非インタラクティブタスクとインタラクティブタスクで異なります。

インタラクティブタスクのエンジン実行サイクル	非インタラクティブタスクのエンジン実行サイクル
• タスク前 　• レコード前 　　コントロール前 　　コントロール検証 　　コントロール後 　　項目変更 　• レコード後 • タスク後	• タスク前 　• グループ前 　　レコード前 　　レコード後 　• グループ後 • タスク後

非インタラクティブタスクでは、コントロールに関連するロジックユニットがありませんが、レコードのグループを処理するためのロジックユニットがあります。

このセクションでは、非インタラクティブタスクの起動から終了までの一連の動作について説明します。

非インタラクティブタスクの初期化

非インタラクティブタスクの初期化プロセスは、インタラクティブタスクと同様です。

グループ処理

- 最初のレコードがメモリーに読込まれます。
- タスク特性の［チェック時期］特性が［B= 前置］に設定されているとき、［タスク終了条件］特性がチェックされます。
- ［終了］アクションがチェックされます。これはエンドユーザによる Esc キー押下の有無に関わりなくチェックされます。
- ［グループ前］ロジックユニットが最初のタスクサイクルとして実行されます。［グループ前］ロジックユニットは、現在レコードのグループ項目値と前レコードのグループ項目値を比較し、内容が異なっているときに起動される処理です。

レコード処理

- ［レコード前］ロジックユニットが実行されます。
- ［レコード後］ロジックユニットが実行されます。
- レコードの値が更新保存されます。タスクモードが［削除］の場合、そのレコードが削除されます。

グループ終了処理

レコード処理が終了し次のレコードが読込まれてきたときに、新しいレコードのグループ項目値と前レコードのグループ項目値を比較し、値が異なるとき、［グループ後］ロジックユニットを前レコードに対して実行し、続いて［グループ前］ロジックユニットを新しいレコードに対して実行します。

タスクの終了

非インタラクティブタスクの終了手順はインタラクティブタスクと同様です。

ただし、インタラクティブタスクがエンドユーザの操作によって終了するのと異なり、非インタラクティブタスクは次のような特性の設定やアクションにより終了します。

［タスク終了条件］特性

この特性は、次のような設定値を持ち、タスク終了の方法を決定します。

- Yes
- No
- 式定義

［チェック時期］特性

この特性は、［タスク終了条件］特性をどの時点で評価するかを指定します。

- B= 前置 …… レコード処理の実行前に評価します。
- A= 後置 …… レコード処理を実行し、レコードデータが更新された後に評価します。
- I= 即時 …… レコードが変更されると同時に評価されます (そのときのロジックユニットの終了を待たずに評価します)。

エンドユーザの介入

非インタラクティブタスクの実行中、Esc キーを押下することで実行を終了させることができます。

４．非インタラクティブタスクの動作

このセクションでは非インタラクティブタスク動作の特徴について説明します。

メインソースのある非インタラクティブタスクの動作

非インタラクティブタスクはメインソースレコードの（範囲設定されている場合はその範囲内の）はじめから最後までをループ処理します。読込まれたすべてのレコードに対して、［レコード前］および［レコード後］のロジックユニット処理が実行されます。

次の条件のどれかが成立したときにタスクは終了します。

- ［タスク終了条件］が［True］と評価されたとき
- 範囲内の最後のレコードを処理し終わったとき

メインソースのない非インタラクティブタスクの動作

メインソースが定義されていないときは、［タスク終了条件］特性が［True］と評価されるまで実行を続けます。

たとえば変数項目のみを処理するようなプログラムではレコードループが無限に続くため、［タスク終了条件］が成立するまで実行は終わりません。

メインソースがある登録モードの非インタラクティブタスク

メインソースを定義している非インタラクティブタスクでタスク初期モードを［C= 登録］にしている場合、タスクは［メインソースのない非インタラクティブタスク］と同様の動作をします。

たとえば、データソースに対してレコードを追加する処理だけを行なうプログラムの場合、レコード範囲の指定ができないためにレコードループは終了を判断できず、［タスク終了条件］が成立するまで実行は終わりません。

非インタラクティブタスクの用途

非インタラクティブタスクはいろいろな用途に利用できます。多くの場合、タスクモードとその用途には明確な関連があります。

次の表に、非インタラクティブタスクのタスクモードと用途に関する一般的な内容をまとめてあります。

タスクモード	用途
照会	メインソースをスキャンし、帳票印刷や特定項目の最大値や最小値を求めるなどの計算処理を行ないます。
登録	データを I/O デバイスから入力したり、他のデータソースからコピー入力します。 (I/O デバイスの利用は、サーバ側のバッチタスクでのみ有効です)
修正	データソースのレコードを自動更新したり、集計計算などを行ないます。
削除	データの整合性を維持するためなど、所定のシナリオにしたがってデータソースレコードの自動削除を行ないます。

５．非インタラクティブタスクによる削除

タスクの［初期モード］を「D= 削除」に設定した非インタラクティブタスクは、データの参照整合性を維持するために利用します。

前章で１対多のデータリレーションについて＜受注管理＞プログラムの実習を通して学びました。

このプログラムでは、エンドユーザはまず最初のタスクで主データソース＜受注＞に１レコード登録し、次に従データソース＜受注明細行＞へ複数のレコードを登録して行きます。

もし主データソースで受注レコードが削除されると、そのままでは従データソースの対応レコードが迷子のまま残されることになってしまいます。

このような事態を避けるために、主データソースでレコードが削除されるときには、非インタラクティブタスクを利用して従データソースの関連レコードをすべて削除するように構成します。

次ページ以降で、この＜受注管理＞プログラムを改良し、迷子のレコードを処理するためのバッチプログラムを作成することにしましょう。

Magic xpa Ver2.3 では、非インタラクティブなリッチクライアントタスクで［初期モード］を「D= 削除」に設定することはできません。本書では、バッチタスクで作成しています。

６．削除を行なう非インタラクティブタスクの例

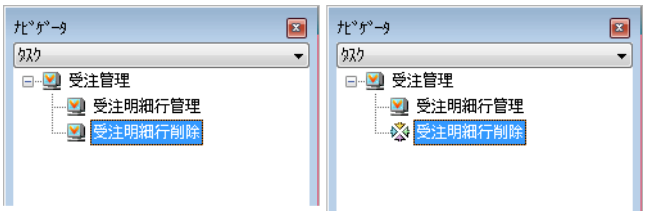
ここでは＜受注管理＞タスクにサブタスクを追加し、エンドユーザが＜受注管理＞タスクでレコードを削除するたびに呼び出されるようにします。

そのサブタスクはバッチタスクで、＜受注管理＞タスクで削除した＜受注＞レコードの受注番号と同じ受注番号を持つ＜受注明細行＞レコードを削除するようにします。

受注明細行レコードを削除するサブタスクの作成

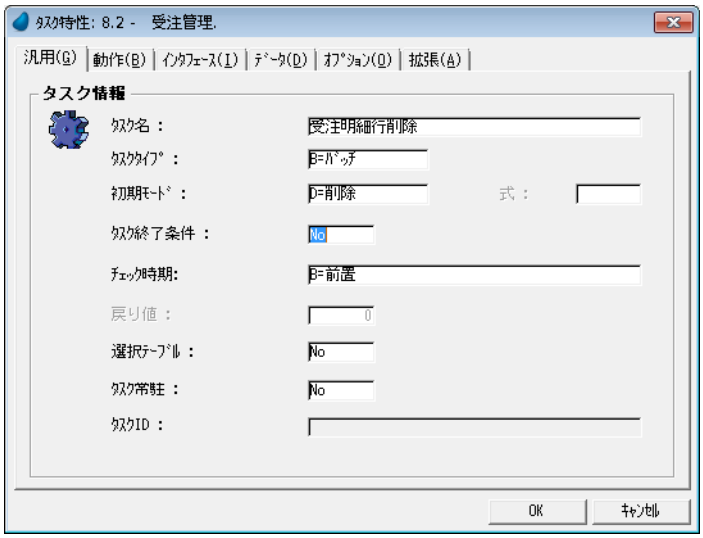
1. ［プログラム］リポジトリ（Shift+F3）を開きます。
2. ＜受注管理＞（#8）プログラムにズームします。
3. ［ナビゲータ］ペインを開き、＜受注管理＞ノードにパークします。

4. F4 キーを押下して一行作成します。



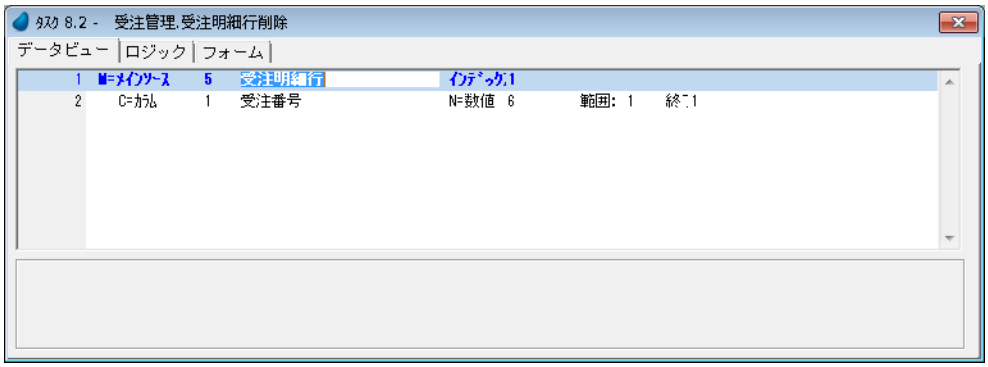
タスク特性が開きますので、次のように設定します。

5. ［タスク名］に、**＜受注明細行削除＞**と入力します。
6. ［タスクタイプ］を、［B= バッチ］とします。
7. ［初期モード］を、［D= 削除］とします。
8. ［OK］ボタンをクリックします。



［データビュー］エディタで次の設定を行います。

9. メインソースとして＜受注明細行＞（#5）を設定します。
10. インデックスは＜受注番号 & 行番号＞（#1）を設定します。
11. F4 キーを押下して、一行作成します。
12. ＜受注明細行＞データソースの最初のカラム＜受注番号＞を選択します。



サブタスクはこの＜受注番号＞値が親タスクレコードの＜受注番号＞値と同じレコードだけを削除しなければなりません。

したがって＜受注番号＞カラムには［範囲］の設定を行います。

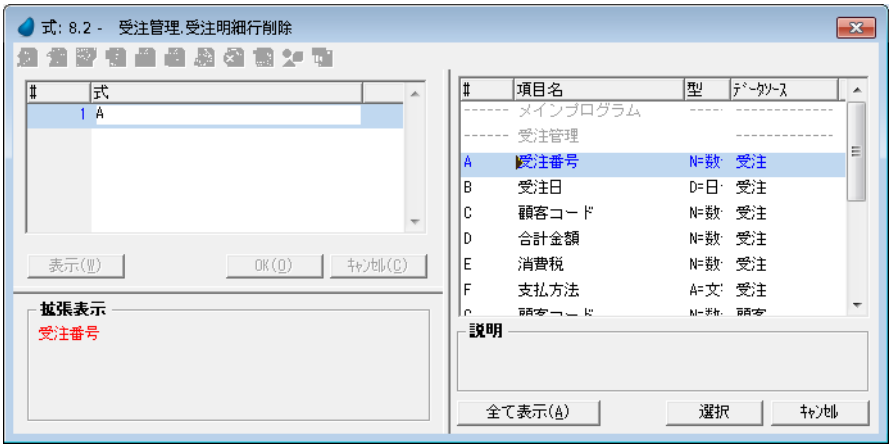
13.［範囲］の式番号欄からズームして［式］エディタを開きます。

14.［式］エディタで一行作成します。

15.ズームして項目リストに移動します。

16.＜受注＞データソースの＜受注番号＞（#A）を選択します。

17.［終了］の式番号欄に同じ式番号＜ #1 ＞を設定しましょう。

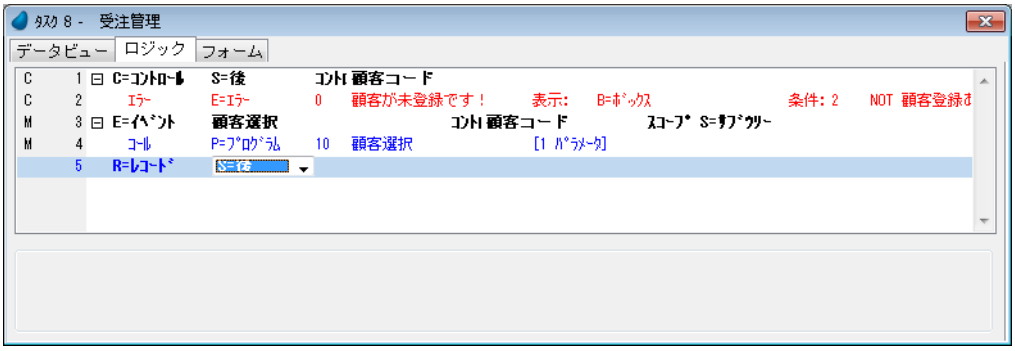


＜受注明細行削除＞サブタスクのコール

＜受注明細行削除＞サブタスクは、エンドユーザが受注レコードを削除するときに＜受注管理＞タスクからコールされなければなりません。

1. ［ナビゲータ］ペインを開き（Alt+F1）、タスクツリーで＜受注管理＞ノードにパークします。
2. ［ロジック］エディタを選択します。
3. 最終行にパークし、新たにヘッダ行（Ctrl+H）を一行作成します。

4. ［レコード後］ロジックユニットを作成します。
5. F4 キーを押下してコマンド行を一行作成します。

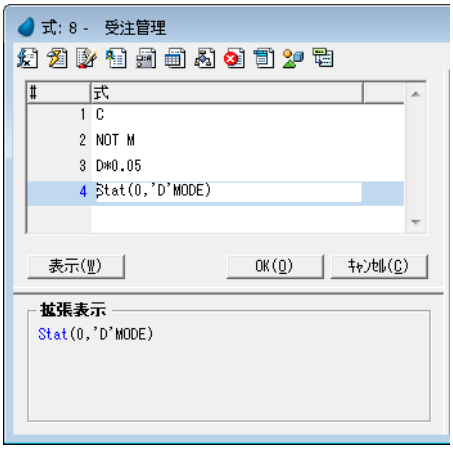


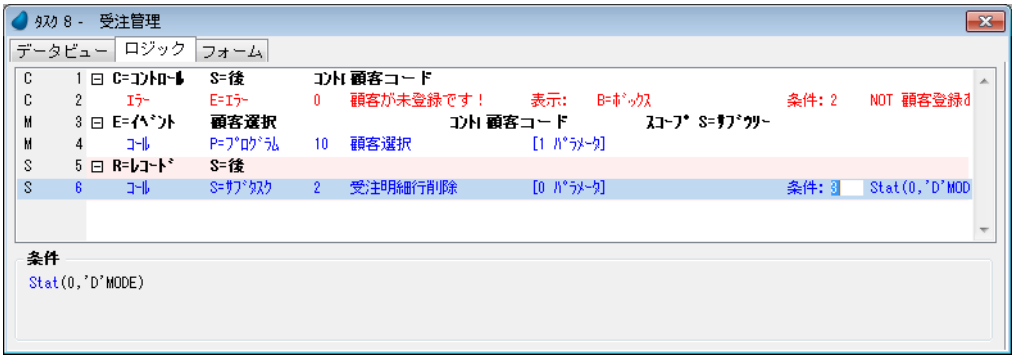
6. 次の設定をします。

コマンド	サブタスク番号	サブタスク名	パラメータ	条件
コール　S= サブタスク	2	受注明細行削除	0	3　(Stat(0,'D'MODE))

条件設定は次のようにします。

7. ［式］エディタにズームします。
8. 最終行にパークし、一行作成します。
9. 文字列＜ Stat(0,'D'MODE) ＞を入力します。
10. ［OK］ボタンをクリックします。





Stat 関数について

タスクモード（照会，修正，登録，削除）を確認します。

構文：　Stat(世代番号 , モード)

パラメータ：

- 世代番号 …… タスクの階層位置を表す番号。カレントタスクが 0、親タスクが 1、その親タスクが 2 などとなります。
- モード …… 確認するタスクモードの頭文字を表す文字。MODE リテラルを指定してください。
  - 'C'MODE … 登録モード
  - 'M'MODE … 修正モード
  - 'Q'MODE … 照会モード
  - 'D'MODE … 削除モード

戻り値：　論理値

- True …… 指定した処理モードと一致する場合
- False …… 指定した処理モードとは異なる場合

プログラムの動作確認

変更内容を保存し、プログラムを閉じたら、動作の確認をしましょう。

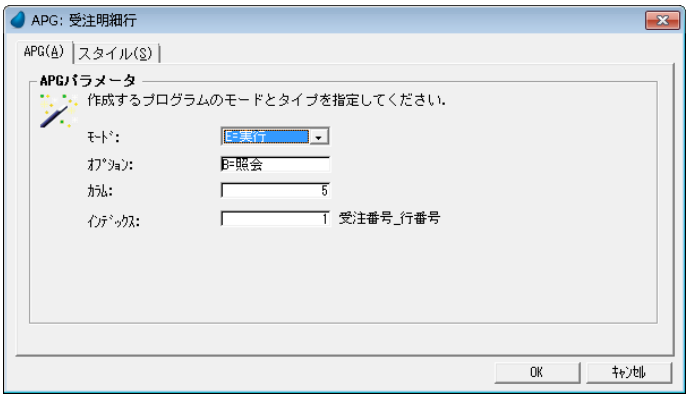
1. F7 キーを押下して＜受注管理＞プログラムを起動しましょう。
2. ＜受注番号＞項目にパークしてください。
3. モードが［登録］に切り替わるまで PageDown キーを押下してください。（修正モードから登録モードへの一時的な切替え）
4. 下のイメージのような受注データを作成してください。
5. ＜受注番号＞欄にパークし、PageUp キーを押下してください。（作成したレコードが保存されます。）
6. PageDown キーを押下して、登録したレコードに戻ります。

7. F3 キーを押下してレコードを削除してみましょう。
8. ［確認］ダイアログで［はい］をクリックします。
9. ＜受注管理＞プログラムを終了します。

画面では非インタラクティブタスクの動いている様子は分かりませんが、実際には非インタラクティブタスクが起動され、関連する明細行レコードは削除されています。

関連する受注明細行レコードがすべて削除されていることを確認してみましょう。

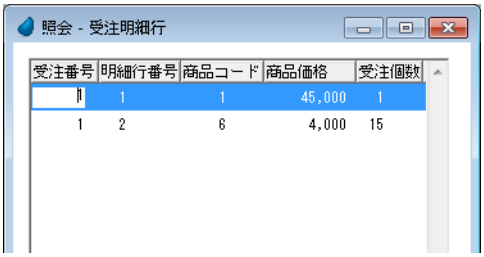
1. ［データ］リポジトリを開きます。
2. ＜受注明細行＞データソース行にパークします。
3. ［オプション］メニューから［APG］を選択します。



［APG］ダイアログが開きますので、そのまま［OK］ボタンをクリックします。

イメージに示すように、先ほど登録し、その後削除した受注番号（この実習では #2）の明細行データは存在しないことが分かります。

4. Esc キーを押下して APG プログラムを終了します。



７．要約

本章では非インタラクティブタスクについて説明し、次のことを学びました。

- ユーザとの対話なく非インタラクティブタスクが動作すること
- インタラクティブタスクと非インタラクティブタスクの違い
- インタラクティブタスクの実行サイクルと非インタラクティブタスクの実行サイクルとの相違点
- 非インタラクティブタスクの実行フロー
- メインソースがある場合とない場合の非インタラクティブタスク動作の違い

また、非インタラクティブタスクは次のような用途に使われることを学びました。

- 帳票の印刷
- レコードグループ単位でのデータ処理
- データの一括更新
- 計算処理
- データソース間のデータコピー
- 外部ファイルからのデータ入力

実習として受注明細行レコードを削除する非インタラクティブタスクを作成しました。このプログラムは＜受注管理＞プログラムにおいて、データの参照整合性を維持するためのものでした。

第 19 章 帳票印刷

本章では、単純な帳票の作成方法について学習します。

キーワード

- 非インタラクティブタスク
- PDF 出力
- フォーム
- 入出力ファイル
- クラス
- 領域
- ［フォーム出力］処理コマンド
- ［ブラウザ］コントロール
- ページヘッダ
- ページフッタ

学習目標

本章を学ぶことで、次の内容に対する理解を深めましょう。

- 単純な帳票を作成するときの APG の利用方法
- 手作業による単純な帳票作成
- ヘッダとフッタを含む帳票のデザイン
- ［ブラウザ］コントロールを使用した PDF の表示

参照

Magic ヘルプの［Magic xpa リファレンス／［フォーム］エディタ／［フォーム］エディタのカラム］

同様に、［Magic xpa リファレンス／［ロジック］エディタ／処理コマンド／フォーム／フォーム出力］

注意

印刷処理は、サーバ側で実行されるバッチタスクでのみ実現可能です。このため、説明では、バッチタスクとして説明しています。RIA で利用する場合、実際の印刷結果はサーバに接続されたプリンタに出力されるか、PDF (Portable Document Format) としてサーバ上に出力し［ブラウザ］コントロール上で表示させる場合が一般的です。

ここでは、PDF を出力する方法を紹介しています。

１．はじめに

帳票の作成はアプリケーション開発における重要な作業の一つです。

帳票によりアプリケーションデータを印刷表示します。

本章では単純な帳票の作成方法について学びます。

Magic xpa にはデータソースから単純な帳票を自動的に作成する APG 機能があるので、まずこの APG を利用した単純な帳票の作成実習を行ないます。

最初に、帳票結果を PDF として出力するための環境設定について紹介します。

２．帳票出力の環境設定

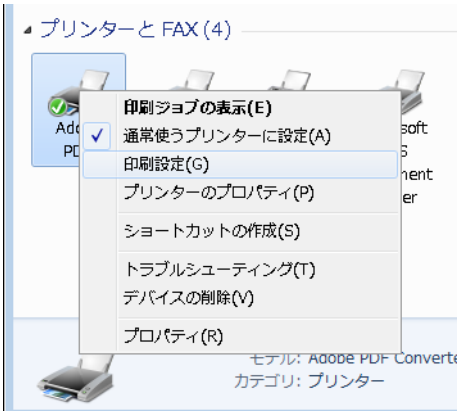
帳票結果を PDF として出力するには、専用の仮想ドライバソフトをサーバ側にインストールする必要があります。以下の機能を持つソフトを推奨します。

- 通常のプリンタドライバと同じように Windows に登録される
- PDF の出力先を指定できる
- （ファイルの上書きや、書込先指定の）ダイアログを表示させないように設定できる
  ダイアログがサーバ上で表示されると処理が停止してしまうため

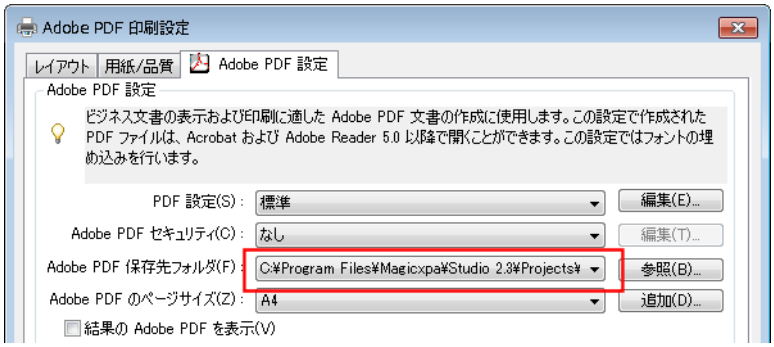
本書では、有償の Adobe Acorbat によってインストールされる Adobe PDF を使用した手順を紹介していますが、無償のソフト（例：Bulzip PDF Printer）を利用することも可能です。

プリンタデバイスの確認

1. Adobe Acorbat をインストールすると Windows の［デバイスとプリンタ］には、「Adobe PDF」というデバイス名が登録されます。
2. このアイコン上で右クリックして「印刷設定」を選択します。



3. ［印刷設定］ダイアログで［Adobe PDF 保存先フォルダ］を指定しておきます。この例では、＜ C:¥Program Files¥Magicxpa¥Studio 2.3¥Projects¥Magic xpa RIA¥PDF ＞フォルダになっています（設定方法は、ソフトによって異なります）。
4. “結果として出力される PDF の自動表示”や”ファイルの上書き確認のオプション”は無効に設定してください（オプション名は、ソフトによって異なります）。

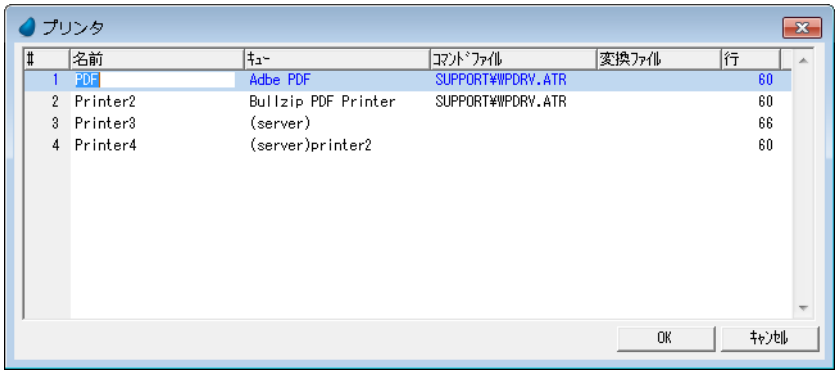


［OK］ボタンをクリックして、設定を保存します。次に、Magic xpa 側の設定を行います。

1. ［オプション］メニューから［設定／プリンタ］を選択して［プリンタ］テーブルを開きます。
2. 一行目を以下のように変更します。

名前	キュー	コマンドファイル	変換ファイル	行
PDF	Adobe PDF	SUPPORT¥WPDRV.ATR		60

キュー名は、Wndow に定義されているプリンタ名です。Adobe PDF 以外を使用する場合は、その名前を指定してください。



3. これで設定は終わりました。［OK］ボタンをクリックしてテーブルを閉じます。

これから作成するプログラムは、ここで定義されたプリンタに出力することになります。

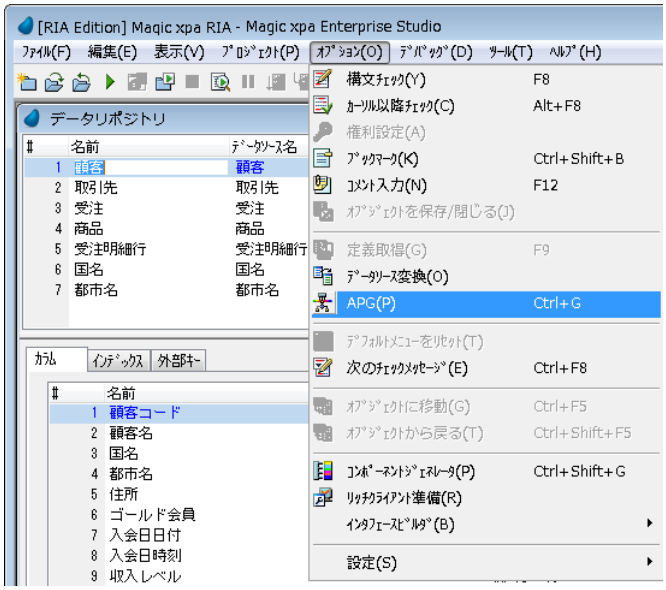
次に APG で印刷プログラムを作成する方法について学んでゆきます。

３．帳票の例　---　APG を使用して作成する場合

＜印刷－顧客＞プログラムの作成

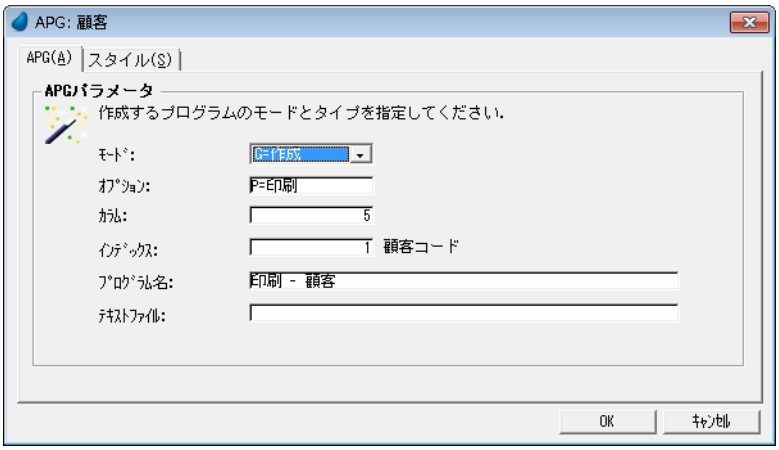
本セクションでは、＜顧客＞データソースの帳票出力を行なうプログラムを作成します。

1. ［データ］リポジトリ（Shift+F2）を開きます。
2. ＜顧客＞（#1）データソースの行にパークします。
3. ［オプション］メニューから［APG］を選択します。



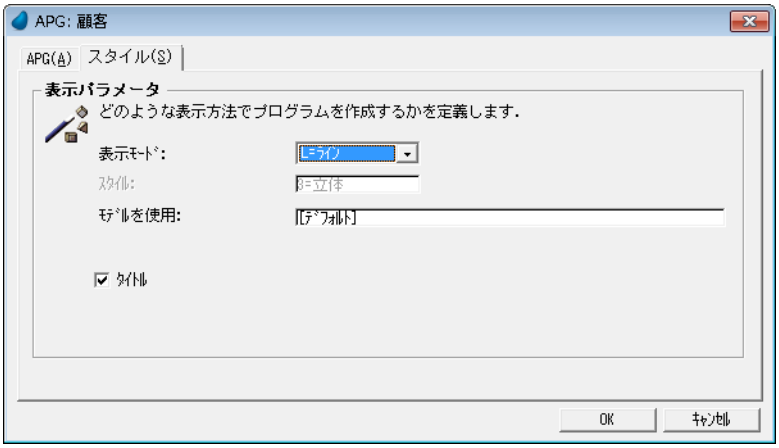
［APG］ダイアログが開きます。

4. ［モード］欄で、［G= 作成］を選択します。
5. ［オプション］欄で、［P= 印刷］を選択します。
6. ［カラム］欄にパークします。
7. ズームして［カラム選択］ウィンドウを開きます。
   a. ＜ゴールド会員＞（#6）以降の［カラム］欄に＜ 0 ＞を入力します。
   b. ［選択］をクリックします。
8. ［インデックス］欄にパークして利用するインデックス番号を指定します。



［カラム選択］ウィンドウの［カラム］欄では、帳票にカラムを表示させる順序を決定します。この欄を＜ 0 ＞に設定した項目は表示対象となりません。

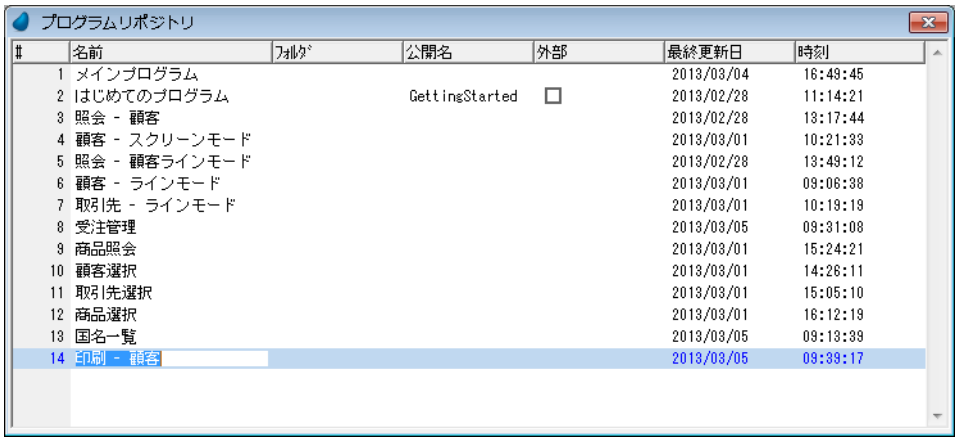
9. ［スタイル］タブをクリックします。
10. ［タイトル］チェックボックスにチェックを入れます。
11. ［OK］ボタンをクリックします。



自動作成されたプログラムの参照

1. ［プログラム］リポジトリ（Shift+F3）を開きます。

＜印刷 - 顧客＞（#14）プログラムが［プログラム］リポジトリ最終行に作成されています。

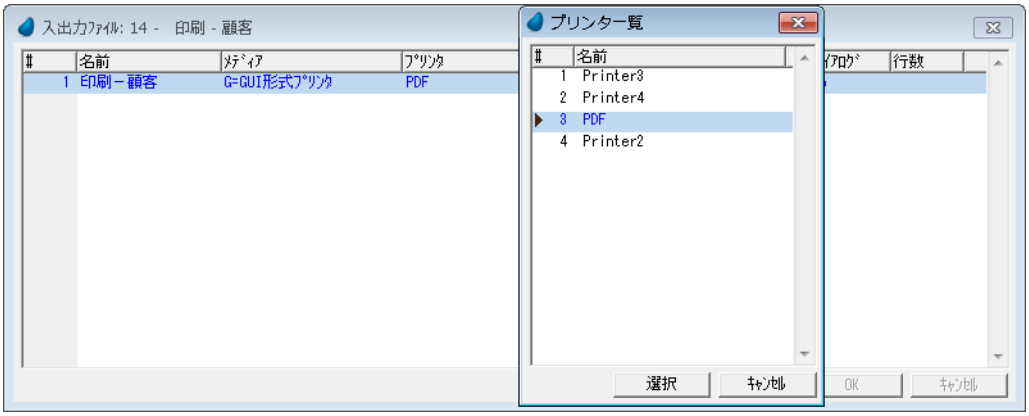


プリンタの設定

帳票の出力先を前述の［プリンタ］テーブルで定義した PDF ドライバに設定することで、印刷結果が PDF で保存されます。この設定は［入出力ファイル］テーブルで行います。

印刷プログラムの［入出力ファイル］テーブルを変更してみましょう。

1. ＜印刷 - 顧客＞プログラムにズームします。
2. ［タスク環境］メニューから［入出力ファイル］（Ctrl+I）を選択します。
3. ［プリンタ］欄からズーム（F5）して［プリンター覧］を開き、＜ PDF ＞を選択します。



ここで選択する「PDF」は、前述で［プリンタ］テーブルに定義した名前です。別の名前で定義している場合は、その名前を選択してください。

［プリンタ］テーブルは自動的にソートされます。このため、入力した時と表示順が異なる場合があります。

4. ［OK］ボタンをクリックします。
5. 変更内容を保存し、プログラムを閉じます。

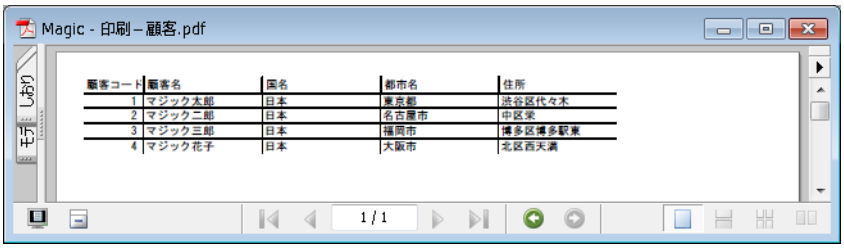
＜印刷 - 顧客＞プログラムの実行

1. F7 キーを押下して＜印刷 - 顧客＞プログラムを実行しましょう。

Magic xpa の出力結果は、＜ C:¥Program Files¥Magicxpa¥Studio 2.3¥Projects¥Magic xpa RIA¥PDF ＞フォルダに作成されます。ファイル名は、上記の例では「Magic - 印刷－顧客 .pdf」になります。ここで、「Magic -」は固定値、「印刷－顧客」は［プリンタ］テーブルの［名前］カラムに定義された文字列です。

「Magic -」を出力しない設定もありますが、ここでは省略します。

2. PDF ビューワで出力結果を確認してください。



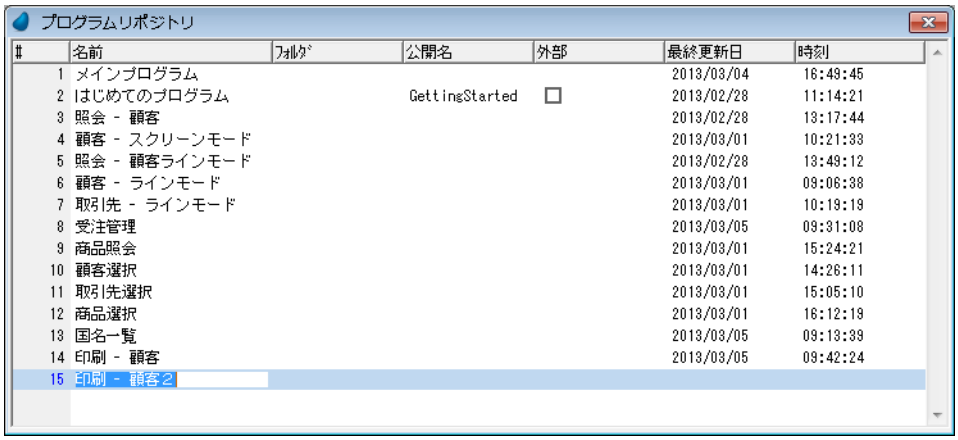
４．手作業による帳票出力プログラムの作成

前セクションで＜顧客＞データソースの内容を印刷出力するプログラムを、APG 機能を使用して自動作成する方法を学びました。

ここでは同じ機能を持つプログラムを手作業で作成してみましょう。

1. ［プログラム］リポジトリを開きます。
2. 最終行にパークします。
3. 一行作成（F4）します。

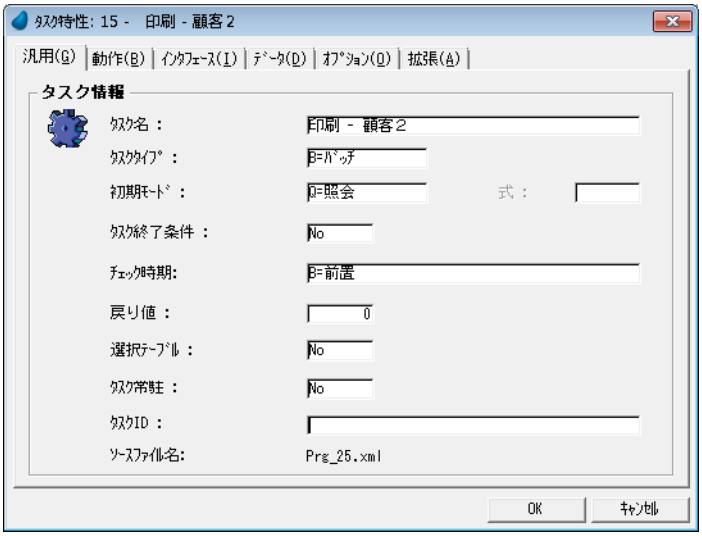
4. ［名前］カラムに **＜印刷－顧客２＞** と入力します。
5. ズームしてプログラムを開きます。



タスク特性の設定

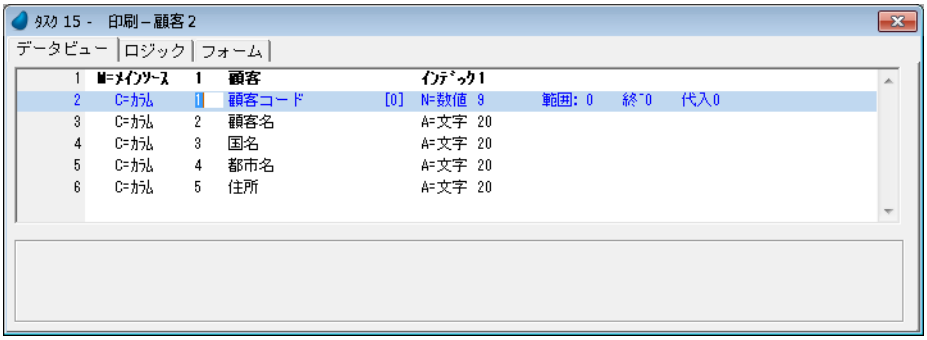
タスク特性が開くので、次のように設定します。

1. ［タスクタイプ］特性は ［B= バッチ］を選択します。
2. ［初期モード］特性は ［Q= 照会］を選択します。
3. ［OK］ボタンをクリックします。



タスクデータビューの設定

1. ［データビュー］エディタを選択します。
2. メインソースとして＜顧客＞（#1）を選択します。
3. インデックスは＜顧客コード＞（#1）を選択します。
4. 一行作成します。
5. ［カラム番号］欄からズームして、［カラム選択］ウィンドウを開きます。
6. 最初の５つのカラムを選択します。
7. ［選択］をクリックします。



入出力ファイルの定義

Magic xpa では入出力ファイルを定義し、次の目的に使用します。

- 外部ファイルからのデータ入力
- グラフィックプリンタやファイルへのデータ出力

ここで［入出力ファイル］テーブルにエントリを追加し、帳票をグラフィックプリンタ（またはグラフィックプリンタの印刷プレビュー）に出力できるようにしましょう。

1. ［タスク環境］メニューから［入出力ファイル］（Ctrl+I）を選択します。
2. ［入出力ファイル］テーブルに一行作成（F4）します。
3. 次のように設定します。

#	名前	メディア	プリンタ	アクセス	書式	式／項目	ダイアログ	行数
1	印刷－顧客	G=GUI 形式プリンタ	PDF	W= 書出	P= ページ	0	No	

［入出力ファイル］テーブルについて

カラム	説明
名前	デバイスの名前。ここで指定した名前は、プリンタのキューにドキュメント名として表示されます。
メディア	ここでは出力先、または入力元のデバイスを以下のなかから選択します。 • G=GUI 形式プリンタ …… 出力はグラフィックデータとして作成され、OS に定義されているグラフィックプリンタドライバに出力されます。 • P= テキスト形式プリンタ …… キャラクタベースの出力が作成され、テキスト印刷用ドライバを通して Magic xpa の論理プリンタに出力されます。 • F= ファイル …… 入力、出力いずれかのディスクファイルを定義します。ファイルのパスと名前は［式］カラムで指定します。 • R= リクエスタ …… 出力結果は、呼び出し元のタスクまたは Web ブラウザに送り返されます。Web 関連で使用します。 • D=XML ダイレクトアクセス …… XML ファイルへのアクセスを行なう場合に使用します。 • V= 項目 …… 文字項目の送受信を行う場合に使用します。
プリンタ	［メディア］カラムの値として［GUI 形式プリント］または［テキスト形式プリント］を指定した場合にのみ有効で、プリンタテーブルに定義されている論理プリンタのいずれかを指定します。
アクセス	ファイルを開く時のモードを定義します。指定できる値は以下のとおりです。 • R= 読込 …… ファイルを読込モードで開きます。［データ入力］処理コマンドをロジックユニットで定義する場合にこのモードを指定します。 • W= 書出 …… ファイルを書出モードで開きます。［データ出力］処理コマンドをロジックユニットで定義する場合にこのモードを指定します。 • A= 追加 …… ファイルを書出モードで開きます。［データ出力］処理コマンドをロジックユニットで定義する場合にこのモードを指定しますが、同名のファイルがすでに存在する場合、そのファイルの末尾に追加されます。
フォーマット	出力ファイルへ自動的に書き出す制御文字、あるいは入力ファイル中で予期される制御文字を定義します。設定できる値は以下のとおりです。 • P= ページ …… 改ページ制御文字がページの終わりに、また改行制御文字が行の終わりに付加されます。 • L= ライン …… 改行制御文字が行の終わりに付加されます。改ページ制御文字は付加されません。 • N= なし …… 改行，改ページ制御文字は付加されません。
式／項目	［式］エディタ内の式の番号を定義します。この式には、OS でのファイル名を指定します。 ［メディア］カラムが［項目］の場合は、項目一覧にズームして項目記号を選択します。BLOB 項目を指定できます。

カラム	説明
ダイアログ	このカラムの有効な値は、［Yes］、［No］、または［式］のいずれかです。［ダイアログ］カラムは、［メディア］カラムが［GUI 形式プリント］の場合にのみ有効です。 • ［Yes］に設定された場合、タスクが入出力ファイルを開いた時に、Windows の［印刷］ダイアログを表示します。ここで、エンドユーザは印刷特性を変更できます。 • ［No］を指定すると、Windows の印刷ダイアログは表示されず、印刷特性はエンドユーザの操作なしでプリンタに直接送られます。 • 式を使用することで、Yes/No の設定を動的に選択できます。
行	［フォーマット］カラムの設定が［P= ページ］の時で、［メディア］カラムの設定が［P= テキスト形式プリント］または［C= コンソール］の場合のみ有効になります。 このカラムには、出力ファイルの 1 ページあたりの行数、つまり 2 つの連続する改ページと改ページの間の行数を指定します。 デフォルト値は［0］で、この場合、出力ページは 60 行になります。これには、ヘッダ領域（3 行）とフッタ領域（3 行）が含まれます。

4. ［OK］ボタンをクリックします。

［入出力特性］ダイアログについて

特性	説明
用紙サイズ	［メディア］カラムで［GUI 形式プリント］を選択した場合に指定できます。出力する用紙のサイズをコンボボックスから選択します。
ページヘッダフォーム	［フォーム］エディタに定義されたフォームからページヘッダとして使用するフォームを選択します。このフォームは改ページするたびに自動的に出力されます。
ページフッタフォーム	［フォーム］エディタに定義されたフォームのうちページフッタとして使用するフォームを選択します。このフォームは改ページするたびに自動的に出力されます。
部数	［メディア］カラムで［GUI 形式プリント］が選択されている場合のみ有効です。印刷される帳票の部数を指定します。
式	グラフィックプリンタで印刷される帳票の部数を式で動的に指定することができます。［式］エディタに定義されている式の番号が入ります。
式／項目	［式］エディタ内の式の番号を定義します。この式には、OS でのファイル名を指定します。 ［メディア］カラムが［項目］の場合は、項目一覧にズームして項目記号を選択します。BLOB 項目を指定できます。
印刷方向	グラフィックプリンタから出力する帳票の印刷方向を指定します。縦方向か横方向かのどちらかを指定します。
印刷プレビュー	出力先を Magic xpa の印刷プレビューシステムに変更します。有効な値は［Yes］，［No］，または論理結果を返す［式］も使用できます。

＜印刷－顧客＞フォームの作成

今までの実習ではクラス =0 のフォームだけを扱ってきました。

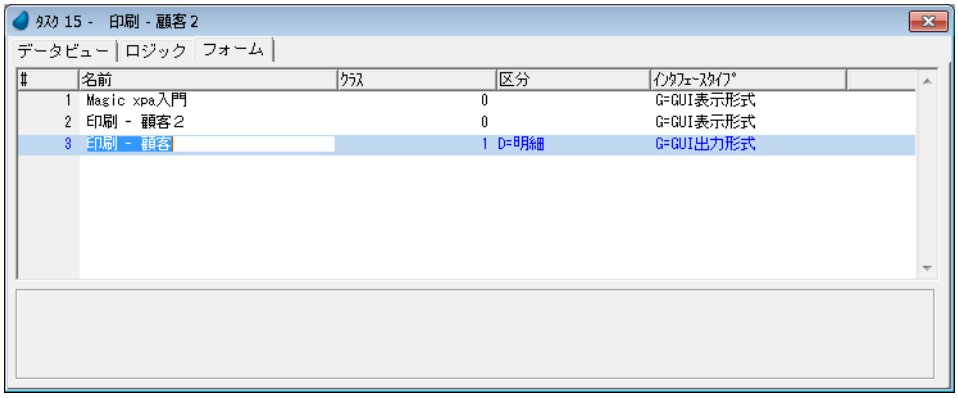
Magic xpa ではフォームは大きく 2 つのクラスに分類されます。

- クラス =0 …… GUI 表示でユーザとの対話用フォームです。
- クラス >0 …… 入出力用フォームです。

クラス =0 のフォームは、インタラクティブなタスク用に使用されますが、RIA ではバッチタスクのフォームは表示されません。このため、このフォームをプログラム内で使用することはありません。

ここではクラス >0 のフォームを使用します。このフォームは［入出力ファイル］テーブルに定義したグラフィックプリンタ（ここでは、PDF ドライバ）に出力されます。

1. ［フォーム］エディタを選択します。
2. 一行作成します。
3. ［名前］カラムに **<印刷－顧客>** と入力します。



クラスカラムが < 1 > となっていることに留意してください。

［フォーム］エディタのカラムについて

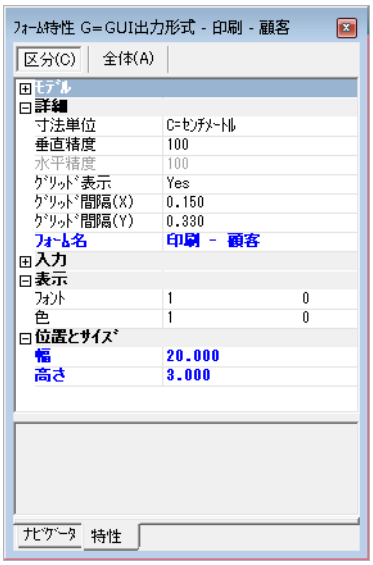
カラム	説明
#	Magic xpa が自動的につける連番です。
名前	フォームの名前です。編集することができます。クラス =0 の場合、表示ウィンドウのタイトルバーにこの名前が表示されます。
クラス	Magic xpa は 2 つのタイプのフォームを定義できます。 • クラス =0 …… エンドユーザとの対話用フォーム • クラス >0 …… 入出力デバイスに対する、入出力用フォーム 同じクラス値（クラス値> 0）のグループのフォームに対し、寸法単位，垂直精度，水平精度の特性が自動的に調整され、同じ値になります。
区分	クラス >0 のフォームでのみ有効です。 • D= 明細 …… 帳票の中央部を定義します。通常はレコードの一覧などを含んでいます。 • H= ヘッダ …… 帳票のヘッダ部を定義します。通常は帳票の見出しなどを出力します。 • F= フッタ ……帳票のフッタ部を定義します。帳票全体のまとめなどを出力します。 • P= ページヘッダ …… 帳票のページヘッダを定義します。入出力特性のなかで指定されます。改ページごとに出力され、会社ロゴや会社情報などを記述したりします。 • G= ページフッタ …… 帳票のページフッタを定義します。入出力特性のなかで指定されます。
インタフェースタイプ	フォームの表示方法を指定します。 • G=GUI 表示形式 …… バッチタスクのクラス =0 のフォームでは常にこの形式が使用されます。 • C= リッチクライアント表示形式 …… 非インタラクティブなリッチクライアントのクラス =0 のフォームでは常にこの形式が使用されます。 • T= テキスト形式 …… テキスト形式プリンタ，ファイル，コンソールなどへの出力に使用します。 • G=GUI 出力形式 …… GUI 形式プリンタ、コンソールなどへの出力に使用します。 • F= フレームセット形式 …… ブラウザウィンドウが複数のフレームに分割される HTML ドキュメントを出力します。出力先メディアとして［R= リクエスタ］が指定されている入出力ファイルに対して使用します。 • M= マージ形式 …… テンプレートファイルを使用してドキュメントのマージ出力を行ないます。

フォーム特性の設定

用紙が A4 の場合、寸法は 210 × 297mm です。（または、8.3 × 11.7 インチ）

フォームの寸法単位をセンチメートルとし、幅を 20cm とすれば、これから作成するフォームが A4 用紙を対象としたものであることになります。

4. フォーム特性を開きます。
5. ［寸法単位］特性が ［C= センチメートル］になっていることを確認します。
6. ［幅］特性を ＜ 20 ＞ とします。これは 1cm のマージンを想定しています。
7. ［高さ］特性を＜ 3 ＞ とします。



フォームデザイン

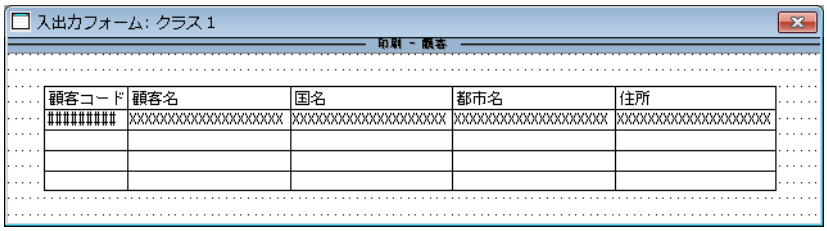
前のセクションで APG を使用して作成したプログラムでは、顧客を表形式で一覧出力していました。

Magic xpa のクラス >0 のフォームでは、区分で［D= 明細］と指定したフォームは繰返し印刷出力されます。クラス =0 の［G=GUI 表示形式］のフォームでは、出力行数は固定で、エンドユーザは必要に応じてスクロールなどをする必要があります。

さて、ここでは［テーブル］コントロールをフォームに配置し、顧客データ詳細を出力するようにしましょう。

8. ズームして＜印刷－顧客＞フォームを開きます。
9. ［コントロール］パレットで［テーブル］コントロールを選択し、フォームに配置します。
10. ［項目］タブから順に一つずつ選択して、5 つの項目すべてテーブルに配置します。

フォームのイメージは下に示すようになるはずです。



11. Esc キーを押下して［フォーム］エディタを終了します。
12. ［確認］ダイアログが表示されますので、［はい］を押下して変更を保存します。

［テーブル］コントロール特性について

ここでは、帳票印刷時に関係する特性についてまとめておきます。

特性	説明
固定サイズテーブル	有効な値は［Yes］または［No］です。［No］がデフォルトです。 • No …… テーブルの実際のサイズは、ページに印刷されるレコードの数によって異なります。 • Yes …… テーブルはフォームに定義されているサイズで印刷されます。
全ページタイトル	有効な値は［Yes］または［No］です。［Yes］がデフォルトです。 • No …… テーブルヘッダはテーブルの先頭だけに印刷されます。 • Yes …… 出力ページ全てにテーブルヘッダが印刷されます。

［フォーム出力］処理コマンドの使用

一つのタスクは、クラス >0 のフォームをいくつも持つことができます。しかし Magic xpa は［フォーム出力］処理コマンドが出力する順序や、何回フォームが使用されるのかということをフォーム定義だけでは知ることができません。

［フォーム出力］処理コマンドは、どのフォームを使用し、どのタイミングで出力デバイスに送るかということを Magic xpa に伝えます。

ここでは［レコード後］ロジックユニットに［フォーム出力］処理コマンドを定義し、＜顧客＞データソースの顧客データをすべて印刷してみることにしましょう。

クラス >0 のフォームに定義された［テーブル］コントロールは、クラス =0 のフォームに定義されているものと機能仕様が異なります。つまりテーブルの明細行部はデータがある限り繰返し出力されます。この仕様により、テーブルの縦方向サイズの検討をすることなく、テーブル形式のデータ出力を実行できるのです。

1. ［ロジック］エディタを開きます。
2. ヘッダ行（Ctrl+H）を一行作成します。
3. ［レコード後］ロジックユニットを定義します。

4. コマンド行を一行作成（F4）します。
5. 次のように処理コマンドを定義します。

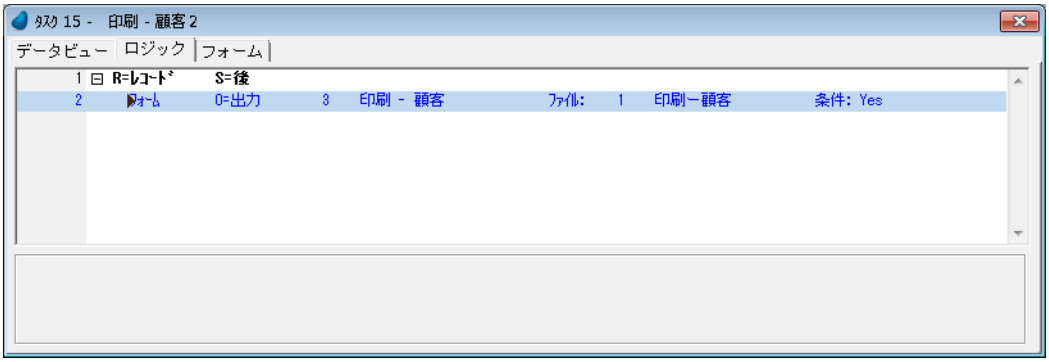
コマンド	入／出力	フォーム	ファイル	条件
フォーム	O= 出力	3　（印刷－顧客）	1　（印刷－顧客）	Yes

［フォーム］欄の設定

6. ［式］欄からズームして［フォーム一覧］を開きます。
7. ＜印刷－顧客＞（#3）を選択します。

［ファイル］欄の設定

8. ［式］欄からズームして［入出力ファイル一覧］を開きます。
9. ＜印刷－顧客＞（#1）を選択します。



10.変更内容を保存し、プログラムを閉じます。

これで単純な＜顧客＞データソースのデータを出力する帳票印刷プログラムが完成です。

［フォーム出力］処理コマンド特性について

特性	説明
フォーム（番号）	出力処理で使用するフォーム番号を指定します。ここからズームし、フォーム一覧からフォーム（クラス >0）を選択します。
入出力ファイル（番号）	フォームを出力する入出力ファイル番号を指定します。ここからズームし、入出力ファイル一覧から選択します。
ページ	「［ページ］特性について」（312 ページ）を参照してください。
条件	Yes , No または式で指定します。

［ページ］特性について

この特性を使用して、ページ終了（EOP）状態が発生した時の動作を指定できます。ページ終了（EOP）状態は、出力対象のフォームが現在のページの残りのスペースより大きい場合に発生します。

ページ終了（EOP）状態の識別にはページサイズ（行数）が使われますが、このページサイズは論理ページサイズで、プリンタの物理ページサイズとは無関係です。Magic xpa の論理ページサイズとプリンタの物理ページサイズは、プリンタテーブルで関係を指定できます。Magic xpa の論理ページサイズは、次のどちらかの値により計算されます。

- ［フォーム出力］処理コマンドで指定されている入出力ファイルの［行］カラムの値。
- ［行］カラムに値が設定されていない時には、［フォーム出力］処理コマンドで指定されている入出力ファイルが出力されるプリンタの［行］カラムの値。

［ページ］特性に指定できる値は、次の通りです。

- S ＝スキップ …… この設定にしておくと、フォームが出力される前に、現在のページにフォームを出力するスペースが十分あるかどうかがチェックされます。スペースがある場合、フォームが出力されます。スペースがない時には、そのフォームはスキップされ、出力先への書き込みは行われません。その場合、ページ終了（EOP）状態がオンになります。このオンとなったページ終了（EOP）状態は、EOP 関数を使用してチェックできます。
- A ＝自動 …… この設定では、ページ終了（EOP）状態が検出されると、改ページが実行されます。また、現在の［フォーム出力］処理コマンドによって出力されるフォームと同じクラス番号を持った全てのヘッダフォームが、もう一度出力されます。この場合、出力されるヘッダフォームは、フォームテーブルで、現在の［フォーム出力］処理コマンドによって出力されるフォームより前に定義されているものに限られます。したがって、フォームテーブルで、現在の［フォーム出力］処理コマンドにより出力されるフォームより後に置かれているヘッダフォームは出力されません。帳票を出力する場合、通常はこの設定にしておきます。
- T ＝先頭 …… この設定にしておくと、プリンタに対して改ページコードが送られ、その後フォームデータがプリンタに送られます。この場合、次のページの先頭から印刷が始まります。つまり、強制改ページが実行されます。この処理では、ヘッダがもう一度出力されることはありません。

プログラムの実行

1. F7 キーを押下して＜印刷－顧客２＞（#15）プログラムを実行します。
2. ＜ C:¥Program Files¥Magicxpa¥Studio 2.3¥Projects¥Magic xpa RIA¥PDF ＞フォルダの「Magic - 印刷－顧客 .pdf」ファイルが更新されます。
3. PDF ビューワーで出力されたファイルを表示してみます。

５．帳票デザインの例

これまで作成したプログラムの帳票は、必要な要素だけを印刷するもっとも単純なデザインのものでした。

これからのセクションでは、どのように帳票を改良して行くかについて学びます。

まず最初に先ほど使用した［テーブル］コントロールの改良を行ないます。

さらに、次のフォームを追加することで、帳票がより洗練された総合的なものとなるようにします。

- ページヘッダ
- ヘッダ
- フッタ
- ページフッタ

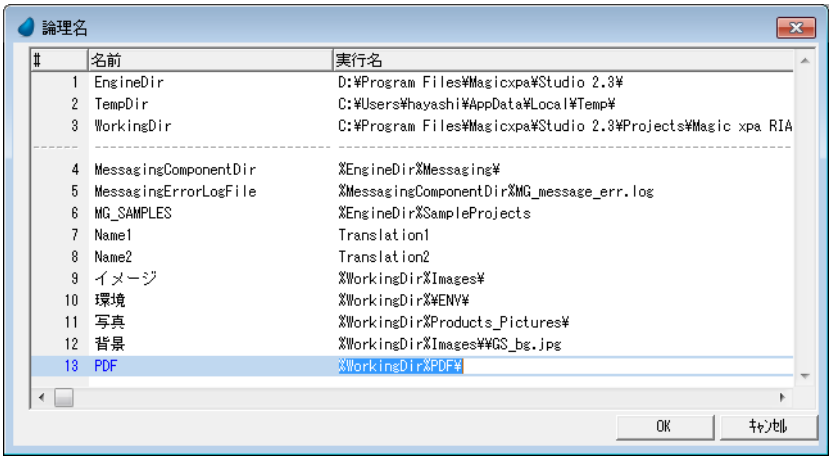
帳票の見映えをよくするため、次の要素を追加しましょう。

- 新しい論理名
- 新しい色
- 新しいフォント

論理名の追加

画像と PDF の格納フォルダ名を論理名で定義しましょう。

1. ［オプション］メニューから、［設定／論理名］を選択し、［論理名］テーブルを開きます。
2. 最終行にパークし、一行作成（F4）します。
3. ［名前］カラムに＜イメージ＞と入力します。
4. ［実行名］カラムに、＜ %WorkingDir%Images¥ ＞と入力します。
5. さらに一行作成（F4）します。
6. ［名前］カラムに＜ PDF ＞と入力します。
7. ［実行名］カラムに、＜ %WorkingDir%PDF¥ ＞と入力します。
8. ［OK］ボタンをクリックします。



Images フォルダには、本コース添付の各種イメージファイルが入っています。

色の追加

これから作成する例では帳票にヘッダを追加します。

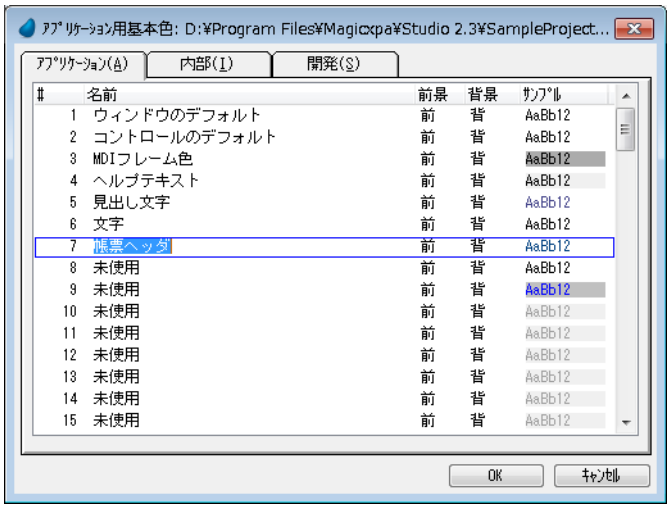
追加するヘッダは、所定の色とフォントを持つテキストで構成することとし、そのための色を基本色テーブルに追加します。

1. ［ファイル］メニューから、［アプリケーション特性］を選択します。
2. ［アプリケーション特性］ダイアログで、［外部参照ファイル］タブを選択します。
3. ［アプリケーション基本色定義ファイル］にパークします。

4. ズームして［アプリケーション用基本色］テーブルを開きます。
5. #7（7 行目）にパークします。
6. ［名前］カラムに**＜帳票ヘッダ＞**と入力します。
7. Tab キーを押下して［前景］カラムに移動します。
8. ズームして［帳票ヘッダ：前景］ダイアログを開きます。
9. ［システム色］欄はブランクを選択します。
10. 基本色領域で、濃い青（RGB=0,64,128）を選択します。
11. ［OK］ボタンをクリックします。
12. もう一度［OK］を押下して［アプリケーション特性］ダイアログに戻ります。

［ファイル名を付けて保存］ダイアログが表示されます。

13. ［即時有効］欄を ［Yes］にしてください。
14. ［OK］ボタンをクリックします。



フォントの追加

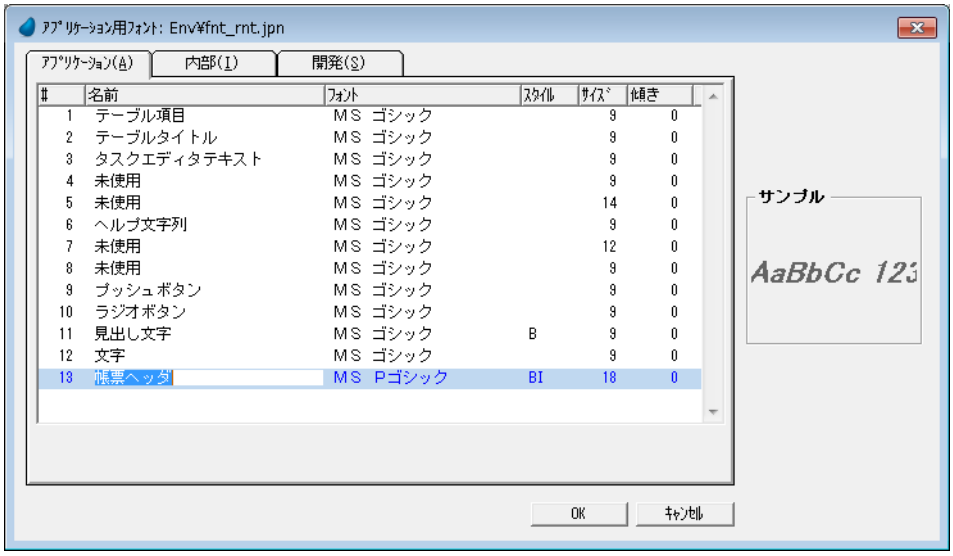
1. ［アプリケーション特性］ダイアログで、［アプリケーションフォント定義ファイル］欄にパークします。
2. ズームして［アプリケーション用フォント］テーブルを開きます。

3. 最終行（#12）にパークします。
4. 一行作成します。
5. ［名前］カラムに**＜帳票ヘッダ＞**と入力します。
6. Tab キーを押下して［フォント］欄に移動します。
7. ズームして［帳票ヘッダ］ダイアログを開きます。
8. ［フォント］欄で、［ＭＳ Ｐ ゴシック］を選択します。

9. ［フォントスタイル］欄で、［太字斜体］を選択します。
10.［サイズ］欄で、［18］を選択します。
11.［OK］ボタンをクリックします。
12.もう一度［OK］を押下して［アプリケーション特性］ダイアログに戻ります。

［ファイル名を付けて保存］ダイアログが表示されます。

13.［即時有効］欄を ［Yes］にしてください。
14.［OK］ボタンをクリックします。
15.［OK］をクリックして［アプリケーション特性］ダイアログを閉じましょう。

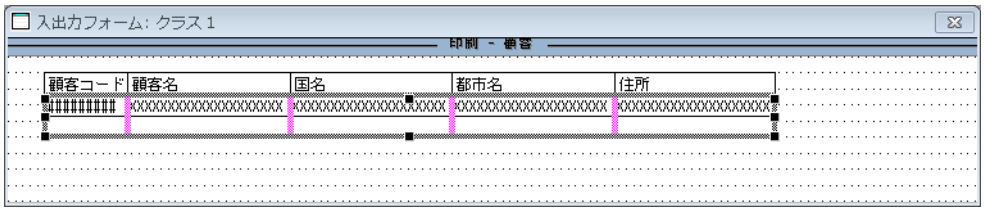


テーブルのカラムタイトルの変更

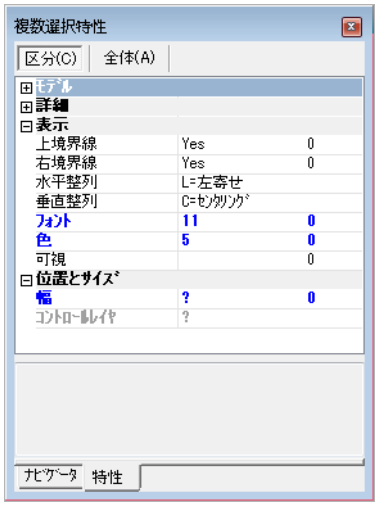
ここでは、テーブルのカラムタイトルの色とフォントを変更してみましょう。

1. ［プログラム］リポジトリを開きます。
2. ズームして＜印刷－顧客２＞（#15）プログラムを開きます。
3. ［フォーム］エディタを選択します。

4. ＜印刷－顧客＞（#3）エントリにパークします。
5. ズームしてフォームを開きます。
6. Ctrl キーを押下したまま、テーブルのタイトル領域をクリックし、テーブルを選択します。
7. テーブルの高さを調節して、明細行が 2 行表示されるようにしてください。
8. テーブル以外をクリックして［テーブル］コントロールのマークをはずします。
9. Alt キーと Ctrl キーを同時に押下したままにします。
10.テーブルの明細行 2 行目のデータカラムを順にクリックし、すべてのカラムが選択された状態となるようにします。

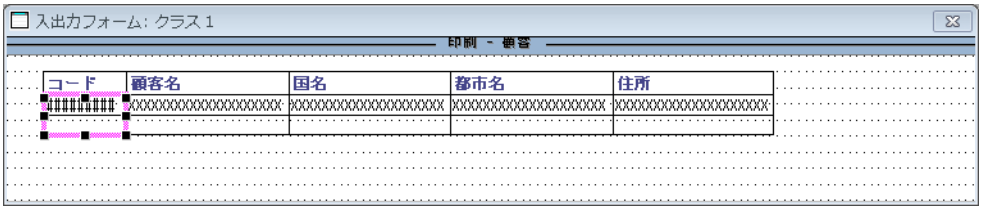


11.Alt+Enter キーを押下して、［複数選択特性］シートを開きます。
12.［フォント］特性を ＜ 11 ＞ にします。
13.［色］特性を＜ 5 ＞ にします。



このままでは＜顧客コード＞がカラム幅から桁あふれするため、表現を修正しましょう。

14.＜顧客コード＞カラムだけを選択します。（Alt キーを押下して＜顧客コード＞の明細行 2 行目カラムをクリックします）
15.カラム特性で、［カラムタイトル］を ＜コード＞ に変更します。

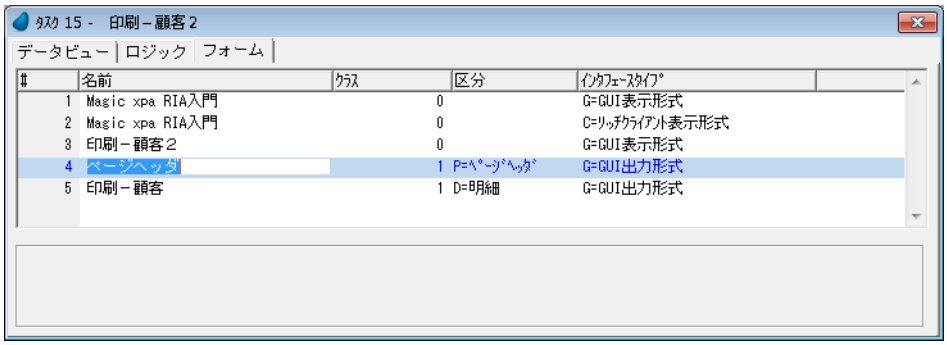


16.変更内容を保存し、［フォーム］エディタに戻ります。

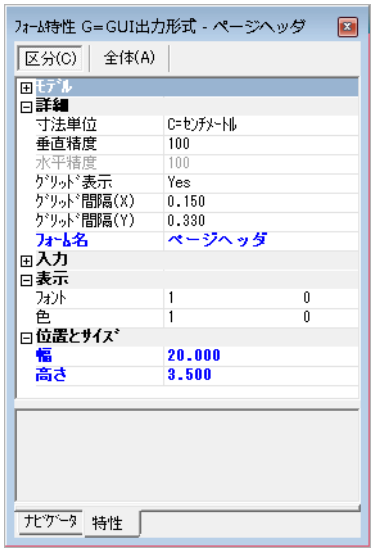
ページヘッダの追加

1. ［フォーム］エディタで、＜印刷－顧客２＞（#3）フォーム行にパークします。
2. 一行作成します。
3. 次のフォームを追加します。

#	名前	クラス	区分	インタフェースタイプ
4	ページヘッダ	1	P= ページヘッダ	G=GUI 出力形式



4. フォーム特性（Alt+Enter）を開きます。
5. ［寸法単位］特性が ［C= センチメートル］であることを確認します。
6. ［幅］特性を ＜ 20.0 ＞ にします。
7. ［高さ］特性を ＜ 3.5 ＞ にします。

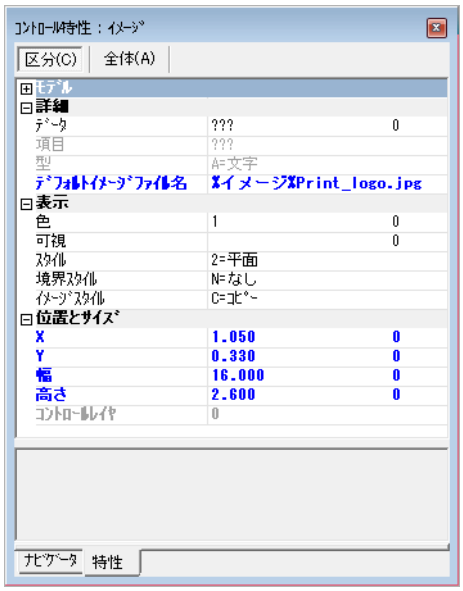


ページヘッダ（または単にヘッダと呼ぶこともあります）は、帳票印刷ページの最上部に出力され、中間の帳票主要部とは分離されるものです。

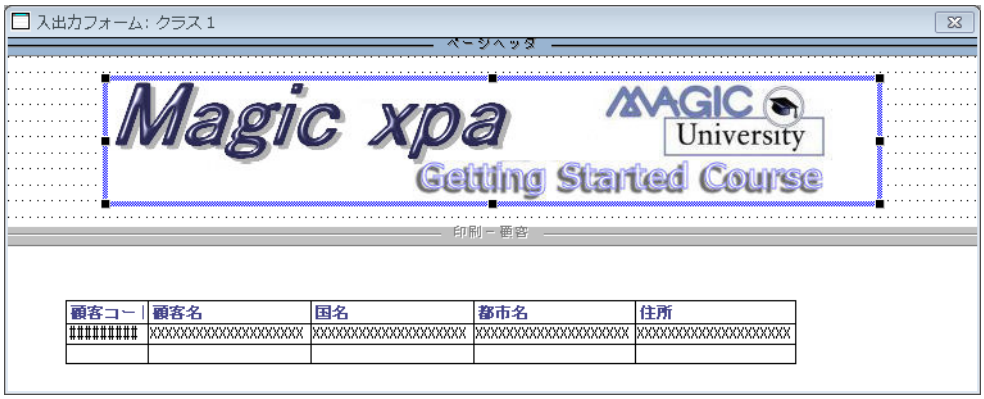
本コースの添付データには、帳票のページヘッダとページフッタ用のイメージを含んだフォルダがあります。

論理名＜イメージ＞でそのフォルダパスを定義しています。

8. ズームして＜ページヘッダ＞フォームを開きます。
9. ［コントロール］パレットで［イメージ］コントロールを選択し、フォームに配置します。
10. ［ファイルを開く］ウィンドウが開きますが、［キャンセル］をクリックして閉じます。
11. ［イメージ］コントロールの特性（Alt+Enter）を開きます。
12. ［デフォルトイメージファイル名］特性に、＜％イメージ %Print_Logo.jpg ＞と入力します。
13. ［幅］特性に ＜ 16 ＞ と入力します。
14. ［高さ］特性に ＜ 2.6 ＞ と入力します。



15. ［イメージ］コントロールの特性を閉じます。
16. ［イメージ］コントロールがマークされています。
17. ［コマンド］パレットで、［水平センタリング］および［垂直センタリング］アイコンをクリックします。



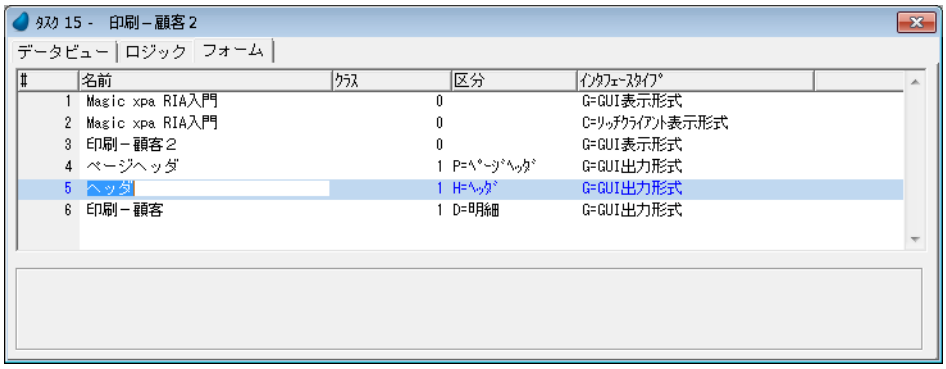
ヘッダの追加

帳票のヘッダは、通常帳票の見出しとして作成されます。

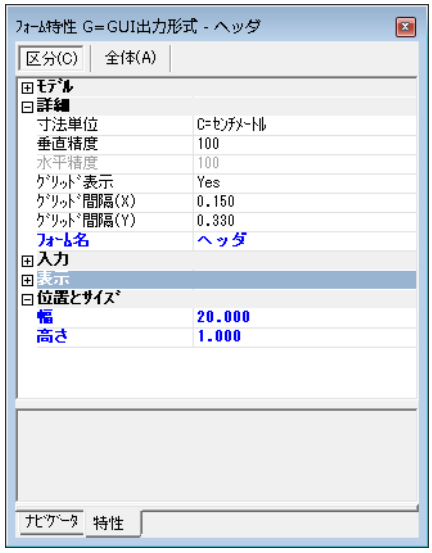
ここからはヘッダを作成しましょう。

1. ［フォーム］エディタに戻ります。
2. ＜ページヘッダ＞（#4）行にパークしています。
3. 一行作成します。

4. ［名前］カラムに **＜ヘッダ＞** と入力します。
5. ［区分］カラムを ［H= ヘッダ］とします。

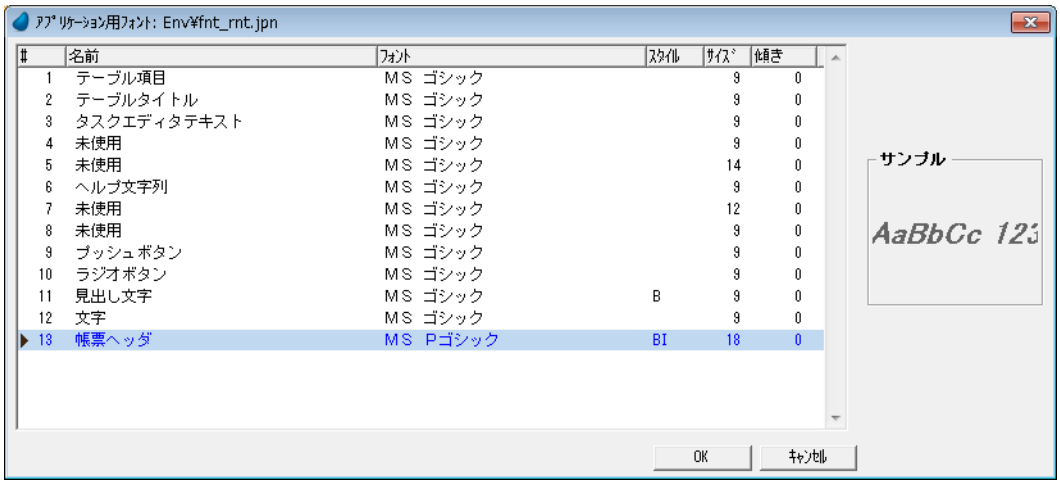


6. フォーム特性（Alt+Enter）を開きます。
7. ［寸法単位］特性が ［C= センチメートル］であることを確認しましょう。
8. ［幅］特性を ＜ 20 ＞ にします。
9. ［高さ］特性を ＜ 1 ＞ にします。



10. ズームして＜ヘッダ＞フォームを開きます。
11. ［コントロール］パレットで［ラベル］コントロールを選択し、フォームに配置します。
12. ［ラベル］コントロールの特性（Alt+Enter）を開きます。
13. ［テキスト］特性に **＜顧客一覧＞** と入力します。

14.［フォント］特性は＜ 13 ＞を設定します。



15.［色］特性は＜ 7 ＞を設定します。

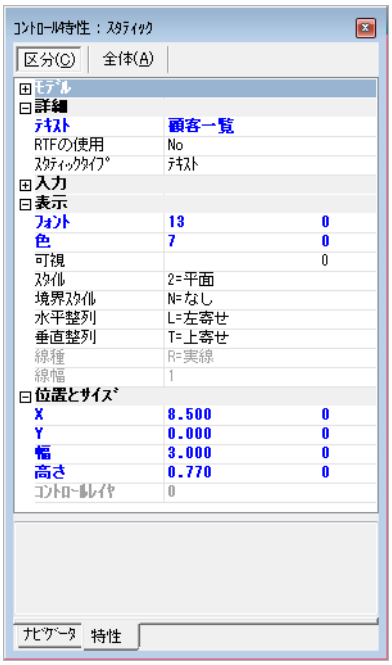
16.［幅］特性は、＜ 3 ＞を設定します。

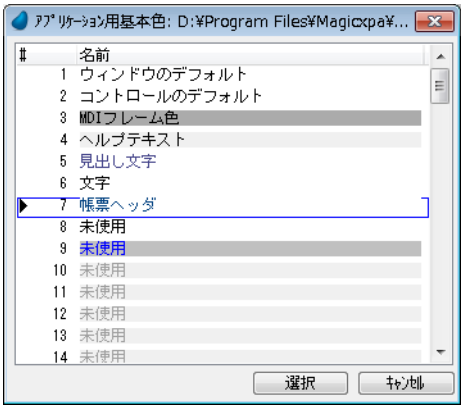
17.［高さ］特性は、＜ 0.77 ＞を設定します。

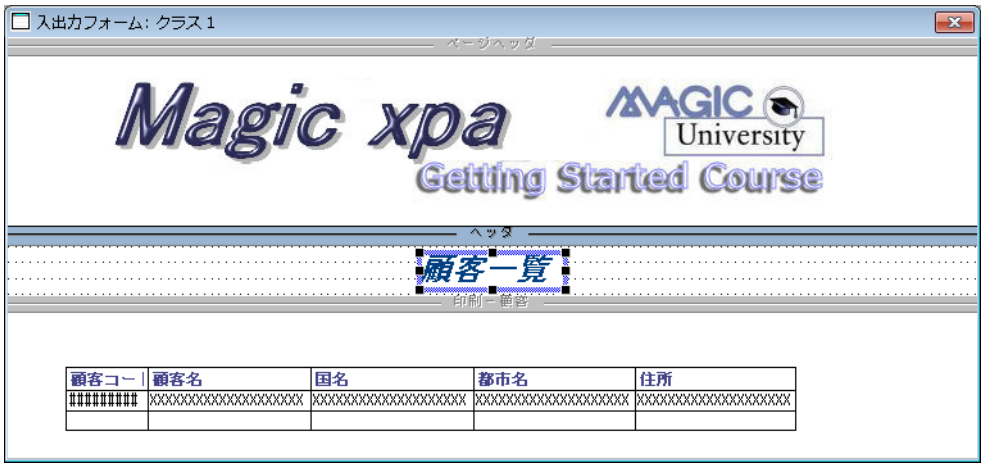
18.［ラベル］コントロール特性を閉じます。

19.［ラベル］コントロールがマークされた状態です。

20.［コマンド］パレットで、［水平センタリング］アイコンをクリックします。



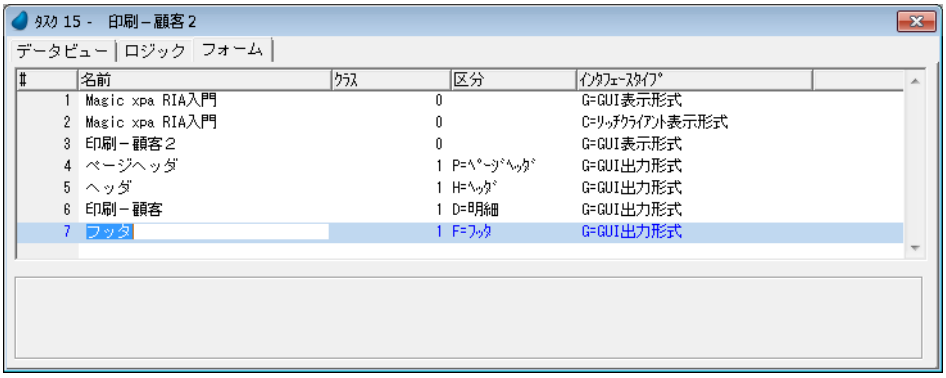




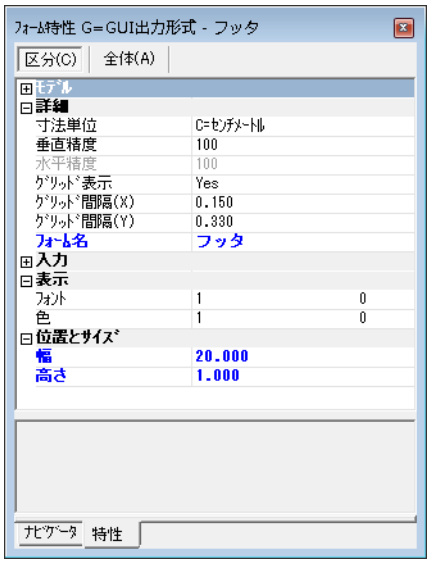
フッタの追加

1. ［フォーム］エディタに戻ります。
2. ＜印刷－顧客＞フォーム（#6）行にパークします。
3. 一行作成（F4）します。

4. ［名前］カラムに **＜フッタ＞** と入力します。
5. ［区分］カラムを［F= フッタ］とします。



6. フォーム特性を開きます。
7. ［寸法単位］特性が ［C= センチメートル］であることを確認します。
8. ［幅］特性を＜ 20 ＞とします。
9. ［高さ］特性を＜ 1 ＞ とします。



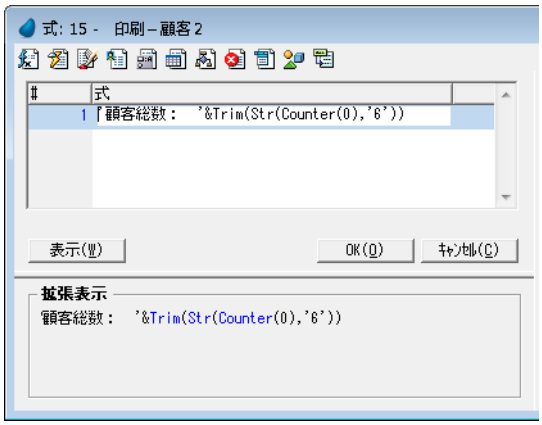
フッタには通常帳票のまとめ的な内容を出力します。この例では、印刷された顧客の総数を出力するようにしてみましょう。

Counter(0) 関数はカレントタスクで処理されたレコード数を返します。

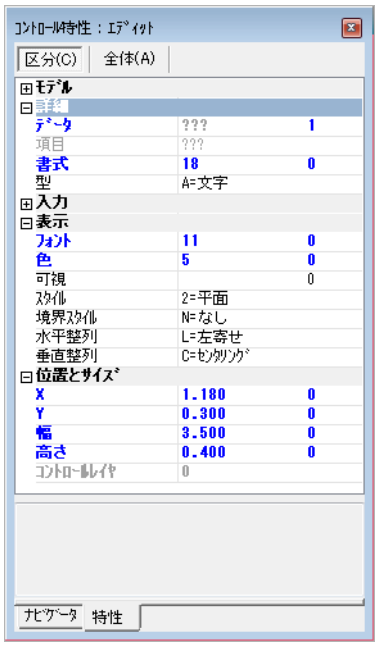
この関数を使用して、帳票に出力された顧客のレコード数を知ることができます。

また、Str 関数を使用して、数値を文字列に変換することができます。

10. ズームして＜フッタ＞フォームを開きます。
11. ［コントロール］パレットで［エディット］コントロールを選択し、フォームに配置します。
12. ［エディット］コントロール特性を開きます。
13. ［データ］特性の［fx］ボタンをクリックして［式］エディタを開きます。
    a. 一行作成します。
    b. ＜’顧客総数： ’&Trim(Str(Counter(0),’6’))＞と入力します。
    c. ［OK］ボタンをクリックします。
14. ［書式］特性は＜ 18 ＞とします。
15. ［フォント］特性は ＜ 11 ＞ とします。
16. ［色］特性は ＜ 5 ＞とします。
17. ［幅］特性 ＜ 3.5 ＞ とします。
18. ［高さ］特性は＜ 0.4 ＞ とします。



19.［エディット］コントロール特性を閉じます。
20.［エディット］コントロールがマークされた状態です。
21.Ctrl キーを押下して＜印刷－顧客＞フォームの［テーブル］コントロールのタイトル行をクリックし、マークします。
22.［コマンド］パレットで［選択左寄せ］アイコンをクリックし、2 つのコントロールの位置を調整します。
23.＜印刷－顧客＞フォームの［テーブル］コントロール以外の場所をクリックして、［テーブル］コントロールのマークをはずします。
24.＜フッタ＞フォームで［エディット］コントロールをクリックし、マークします。
25.［コマンド］パレットで、［垂直センタリング］アイコンをクリックします。



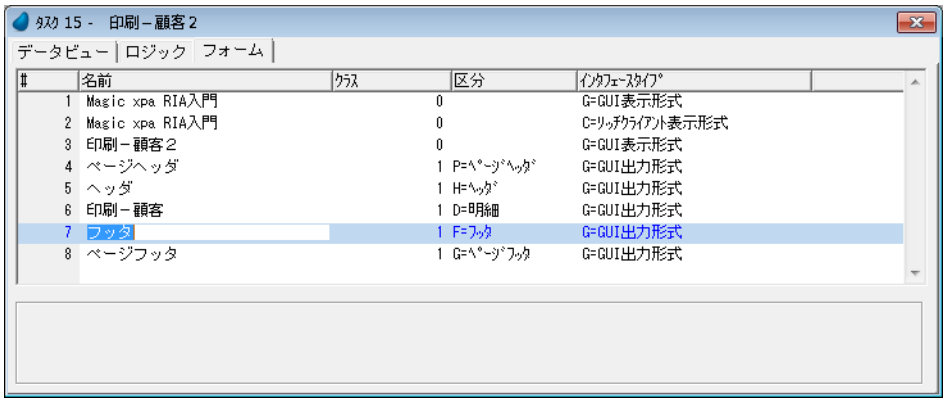
ページフッタの追加

ページフッタには会社に関連した情報を出力したりします。

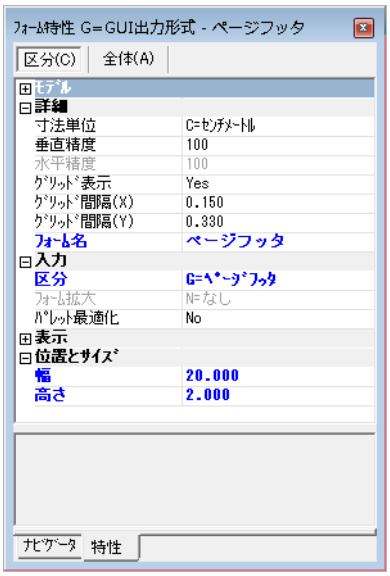
実習例では、イメージデータを出力させましょう。

1. ［フォーム］エディタに戻ります。
2. ＜フッタ＞フォーム（#7）行にパークし、一行作成します。
3. ［名前］特性に＜ ページフッタ＞ と入力します。

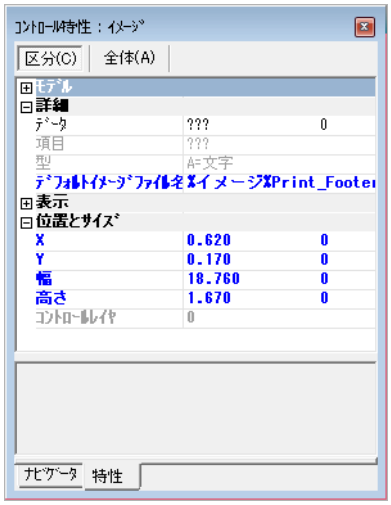
4. ［区分］特性を ［G= ページフッタ］とします。

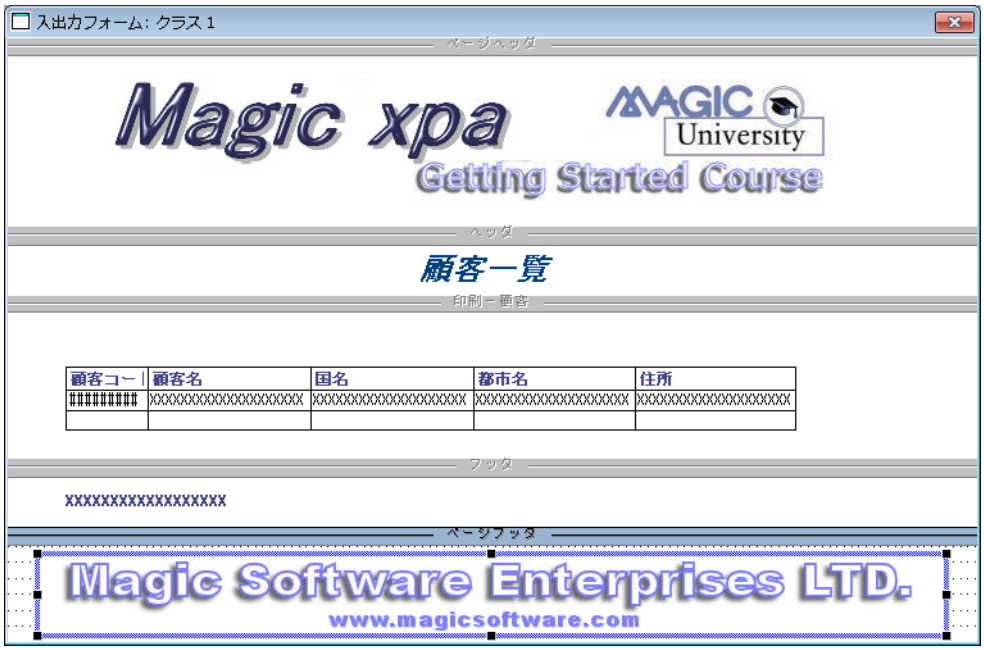


5. フォーム特性（Alt+Enter）を開きます。
6. ［寸法単位］特性が ［C= センチメートル］であることを確認します。
7. ［幅］特性を＜ 20 ＞とします。
8. ［高さ］特性を ＜ 2 ＞とします。



9. ズームして＜ページフッタ＞フォームを開きます。
10. ［コントロール］パレットから［イメージ］コントロールを選択し、フォームに配置します。［ファイルを開く］ウィンドウが開きますが、［キャンセル］を押下して閉じましょう。
11. ［イメージ］コントロールの特性を開きます。
12. ［デフォルトイメージファイル名］欄で、＜ %イメージ%Print_Footer.jpg ＞と入力してください。
13. ［幅］特性は ＜ 18.76 ＞ としてください。
14. ［高さ］特性は ＜ 1.67 ＞ としてください。
15. ［イメージ］コントロールの特性を閉じます。
16. ［メージ］コントロールがマークされている状態です。
17. ［コマンド］パレットで［水平センタリング］および［垂直センタリング］のアイコンをクリックしてください。
18. ［フォーム］エディタを終了します。





ヘッダの出力

ヘッダフォーム（[区分］カラムの設定が［H= ヘッダ］のフォーム）は特徴的な動作をします。

帳票の 2 ページ目以降、同じクラスの帳票が印刷されるときには、毎ページ自動的に出力されます。

しかし最初のページは自動出力ではないため、プログラムで設定しなければなりません。

［タスク前］ロジックユニットはプログラム開始時に一度だけ実行される処理のために利用されます。

ここでは、ヘッダフォームを印刷する［フォーム出力］処理コマンドを追加してみましょう。

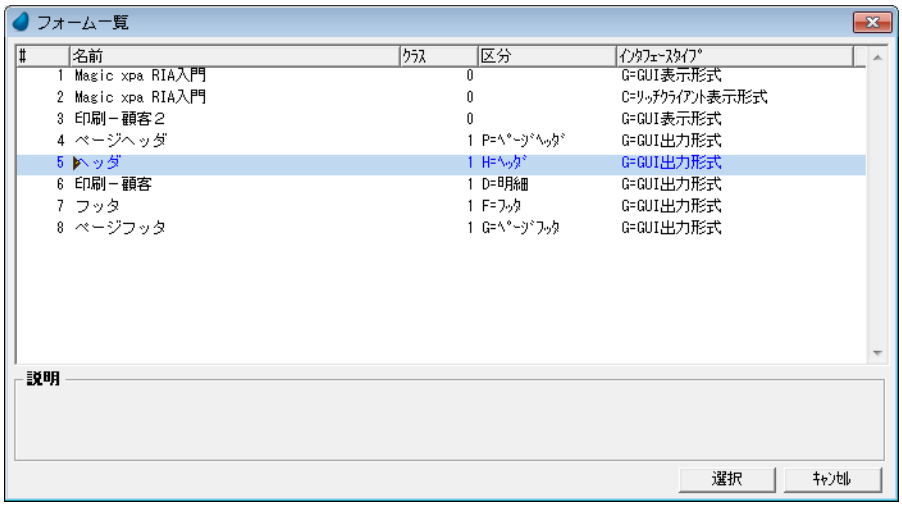
1. ［ロジック］エディタを開きます。
2. ［レコード後］ロジックユニットの行（#1）にパークし、↑ キーを 1 回押下します。
3. ヘッダ行（Ctrl+H）を一行作成します。

4. ［タスク前］ロジックユニットを定義します。
5. コマンド行を一行作成（F4）します。
6. 次のように設定します。

コマンド	入／出力	フォーム	ファイル	条件
フォーム	O= 出力	5 （ヘッダ）	1 （印刷－顧客）	Yes

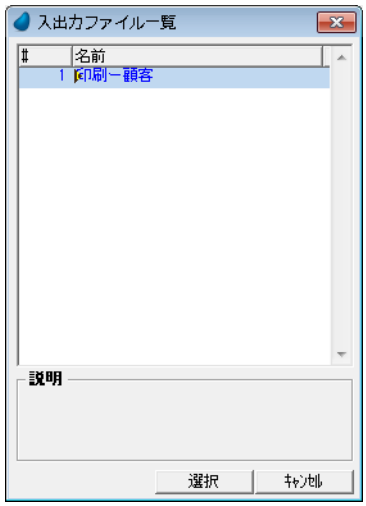
フォームの設定

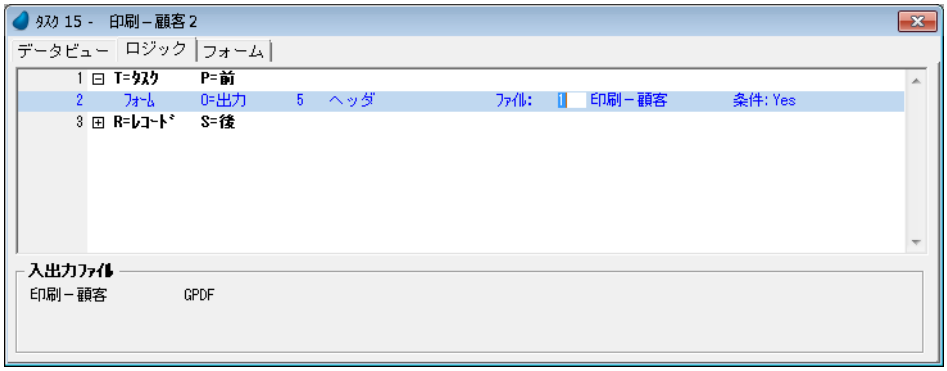
ズームしてフォーム一覧から＜ヘッダ＞（#5）を選択します。



ファイルの設定

ズームして＜印刷－顧客＞（#1）を選択します。





フッタの出力

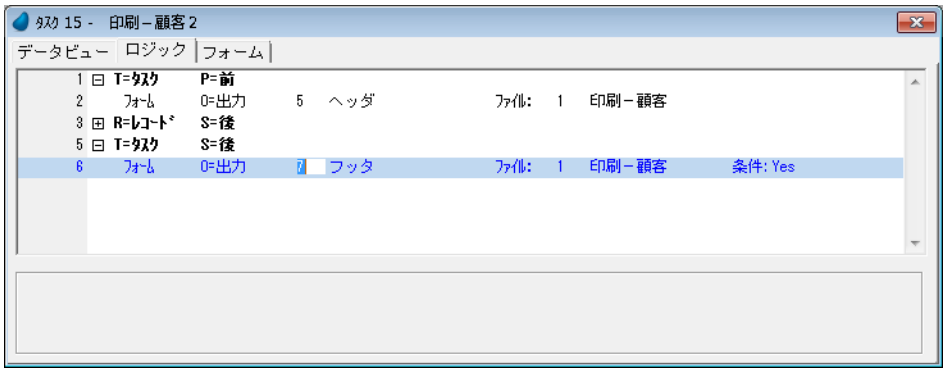
フッタは帳票の最後に一度だけ印刷されるものです。したがって、プログラムの終了時に実行される［タスク後］ロジックユニットで出力するようにします。

1. ［ロジック］エディタで最終行にパークします。
2. ヘッダ行（Ctrl+H）を一行作成します。
3. ［タスク後］ロジックユニットを定義します。

4. コマンド行を一行作成します。(F4)
5. 次のように設定します。

コマンド	入／出力	フォーム	ファイル	条件
フォーム	O= 出力	7　(フッタ)	1　(印刷－顧客)	Yes

フォーム，ファイルの設定はヘッダ出力と同様の手順で選択します。



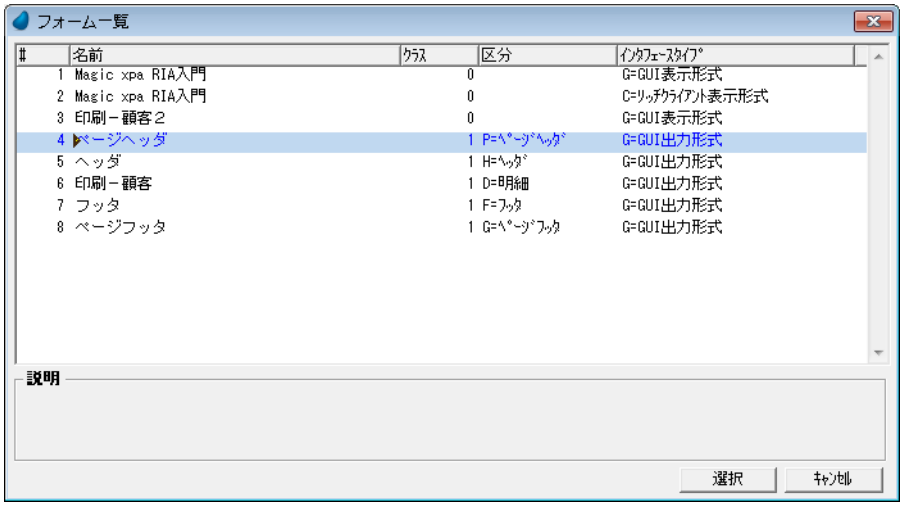
ページヘッダとページフッタの設定

ページヘッダとページフッタは、ページの印刷内容に関係なく、常に出力されることを想定するものです。したがって、これらのフォームについては入出力ファイルの特性として定義を行ないます。この定義を行なうことで、Magic xpa がページ印刷を行なうときには常に出力されるようになります。

では、入出力ファイル特性を設定してみましょう。

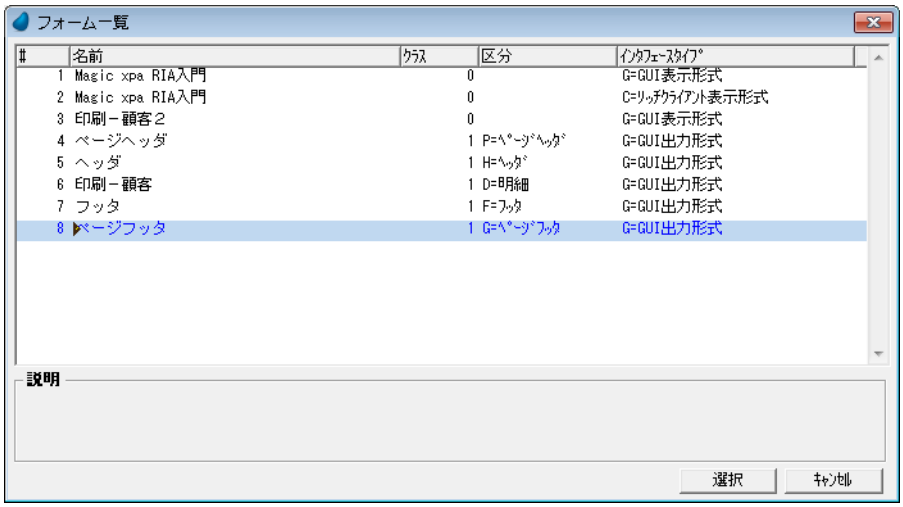
1. ［タスク環境］メニューから［入出力ファイル］を選択します。
2. ＜印刷－顧客＞（#1）の行にパークします。
3. フォーム特性（Alt+Enter）を開きます。

4. ［ページヘッダフォーム］からズームし、フォーム一覧を開きます。
5. ＜ページヘッダ＞（#4）を選択します。

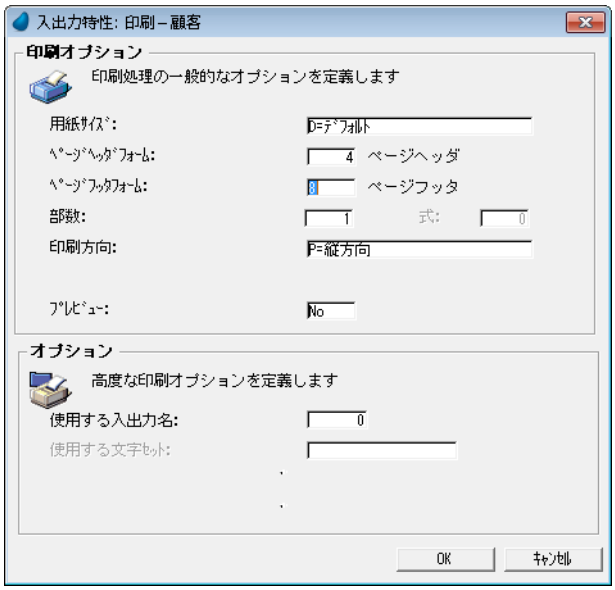


6. ［ページフッタフォーム］からズームし、フォーム一覧を開きます。

7. ＜ページフッタ＞（#8）を選択します。



8. ［OK］ボタンをクリックします。
9. 変更内容を保存し、プログラムを閉じます。



＜印刷－顧客２＞プログラムの実行

1. F7 キーを押下してプログラムを実行しましょう。

これまでに作成した要素をすべて含んだ出力を確認してください。

６．練習問題

これまでに学んだことを応用して、仕入先一覧を出力するプログラムを作成してください。

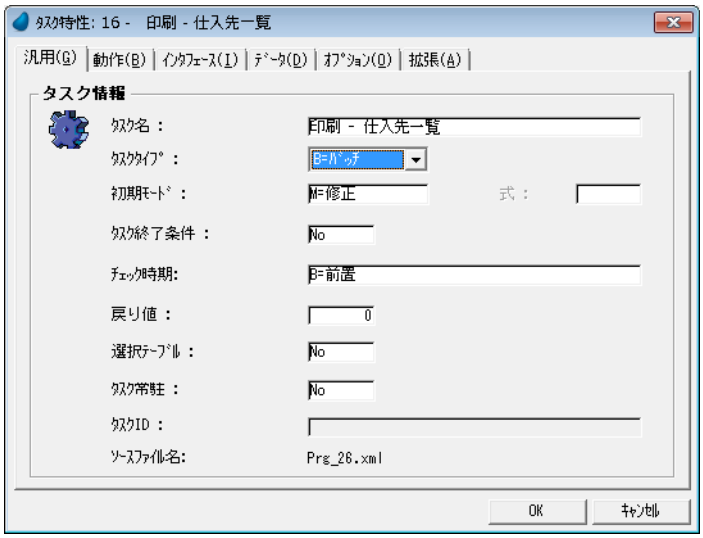
- 実習で取り組んだ<顧客一覧>の帳票形式と同じ形式にしてください。

最初はまず一人で何も見ないで作成してみましょう。
ガイドが必要な場合は、以下に説明がありますので、参考にしてください。

練習問題の解説

［プログラム］リポジトリで、
1. 最後の行で新しく一行作成し、**<印刷－仕入先一覧>**と名前を入力します。
2. ズームしてプログラムを開きます。

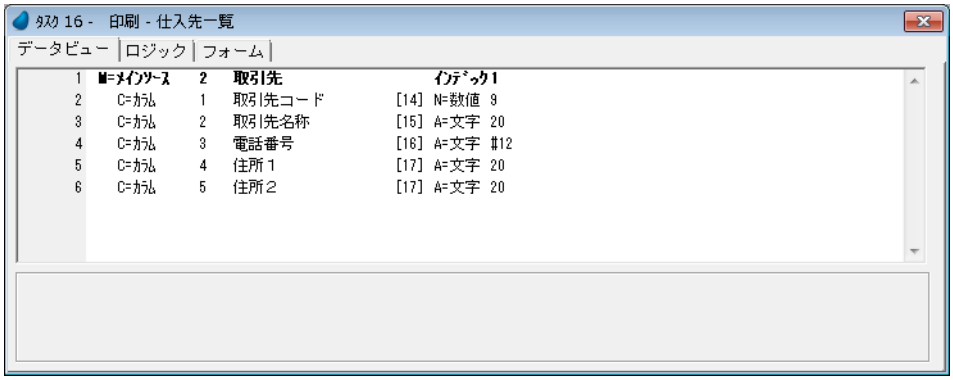
タスク特性が表示されるので、
3. ［タスクタイプ］を ［B= バッチ］とします。
4. ［OK］をクリックして特性を閉じます。



タスクデータビューの定義

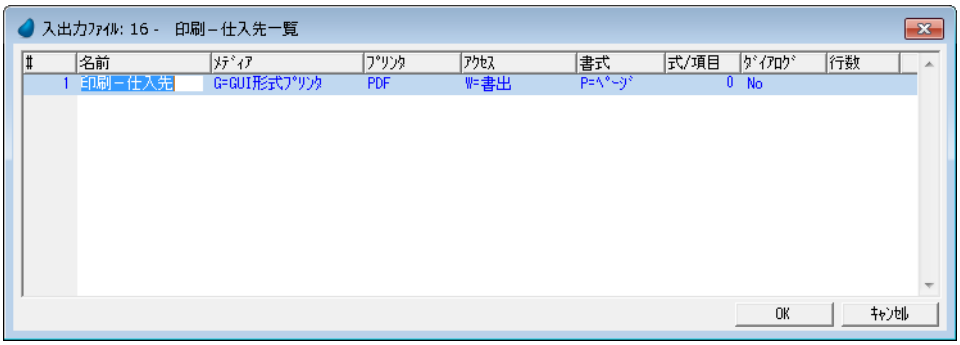
［データビュー］エディタで、

1. メインソースとして、＜取引先＞（#2）データソースを選択します。
2. カラムは最初の 5 つを定義しましょう。



入出力ファイルの定義

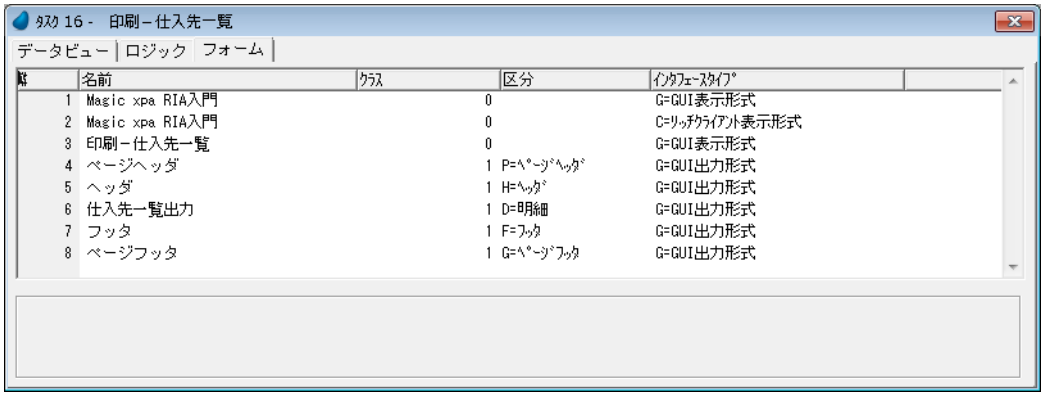
1. ［タスク環境］メニューから［入出力ファイル］（Ctrl+I）を選択します。
2. 一行作成（F4）します。
3. 名前を＜印刷－仕入先＞と入力します。
4. プリンタを＜ **PDF** ＞に変更します。



フォームの作成

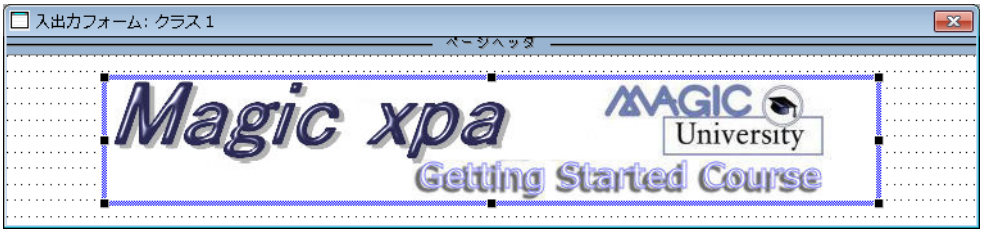
1. ［フォーム］エディタを選択します。
2. 次のフォームを順に作成します。

#	名前	クラス	区分	インタフェースタイプ
4	ページヘッダ	1	P= ページヘッダ	G=GUI 出力
5	ヘッダ	1	H= ヘッダ	G=GUI 出力
6	仕入先一覧出力	1	D= 明細	G=GUI 出力
7	フッタ	1	F= フッタ	G=GUI 出力
8	ページフッタ	1	G= ページフッタ	G=GUI 出力

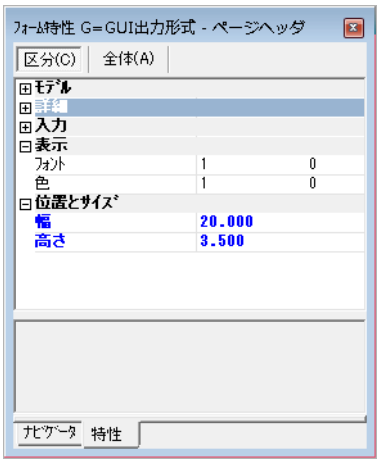


＜ページヘッダ＞フォームの設定

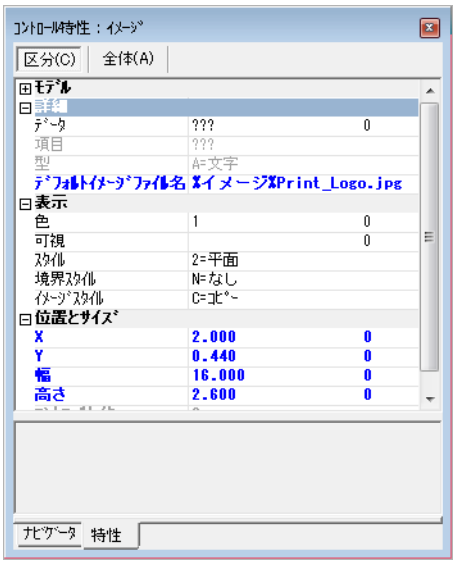
＜ページヘッダ＞フォーム：



フォーム特性：



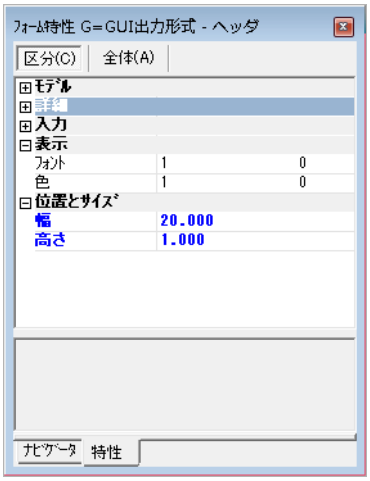
［イメージ］コントロール特性：



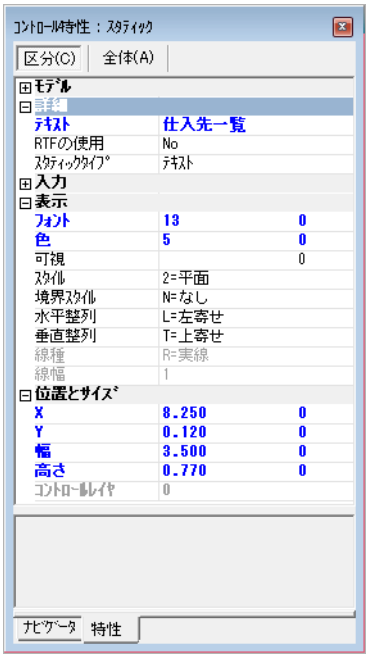
＜ヘッダ＞フォームの設定

＜ヘッダ＞フォーム：

フォーム特性：



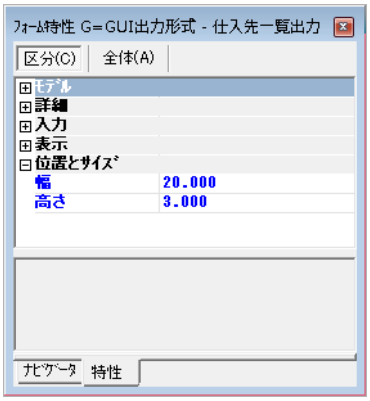
［ラベル］コントロール特性：



＜仕入先一覧出力＞フォームの設定

＜仕入先一覧出力＞フォーム：

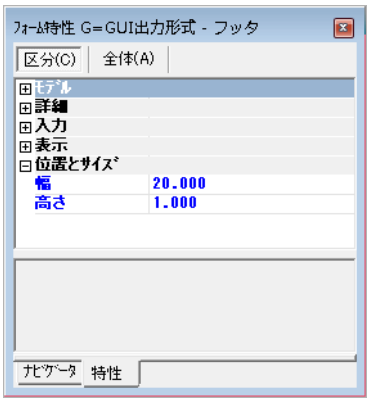
フォーム特性：



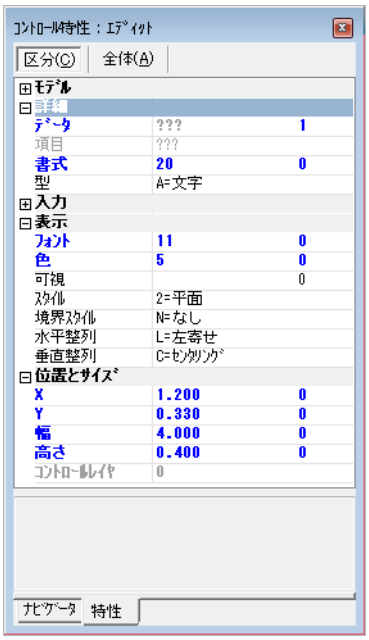
＜フッタ＞フォームの設定

＜フッタ＞フォーム：

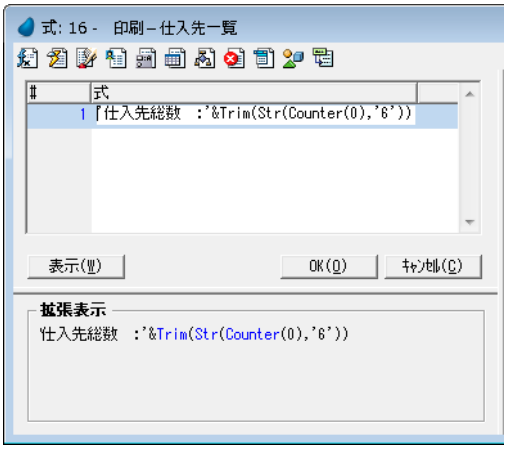
フォーム特性：



［エディット］コントロール特性：

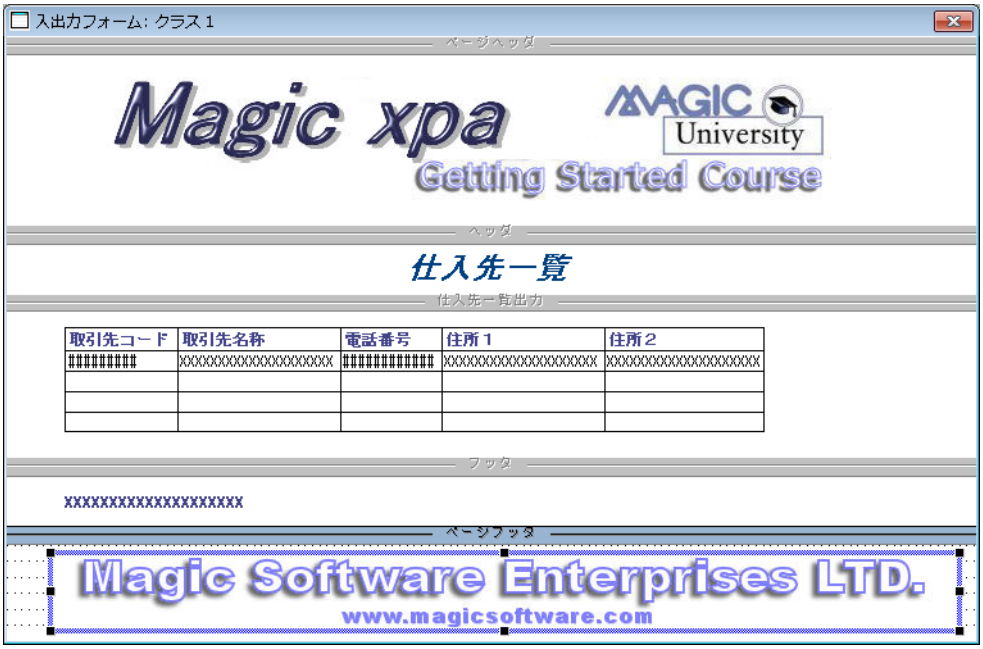


［式］エディタ：

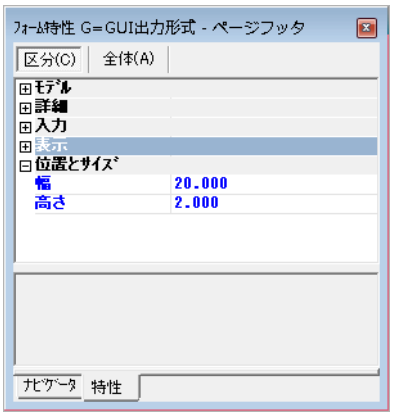


＜ページフッタ＞フォームの設定

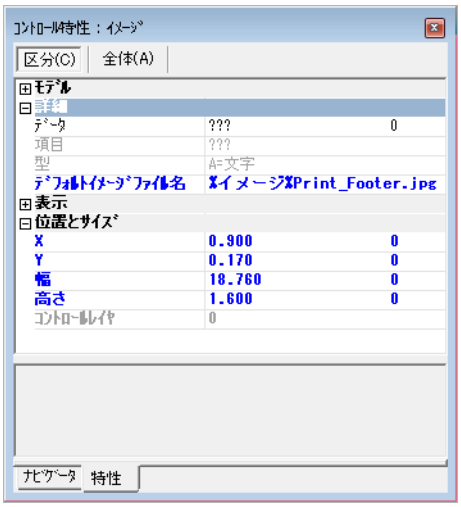
＜ページフッタ＞フォーム：



フォーム特性：



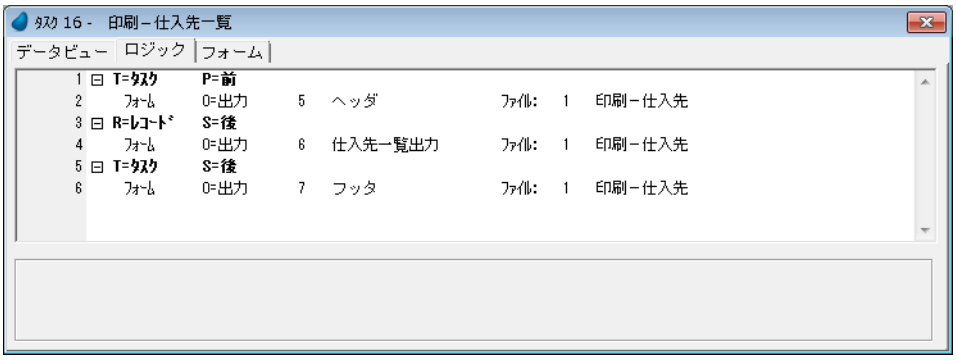
［イメージ］コントロール特性：



ロジックユニットの作成

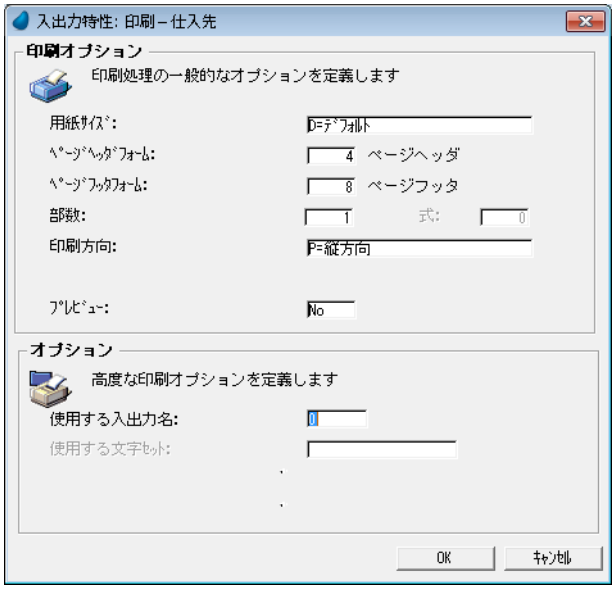
［ロジック］エディタで 3 つのロジックユニットを作成します。

1. ［タスク前］ロジックユニットから、＜ヘッダ＞フォームを出力します。
2. ［レコード後］ロジックユニットから、＜仕入先一覧出力＞フォームを出力します。
3. ［タスク後］ロジックユニットから、＜フッタ＞フォームを出力します。



入出力ファイル特性の設定

1. ［タスク環境］メニューから［入出力ファイル］を選択します。
2. ＜印刷－仕入先＞の特性（Alt+Enter）を開きます。
3. ［ページヘッダフォーム］特性に ＜ 4 ＞ を設定します。
4. ［ページフッタフォーム］特性に ＜ 8 ＞ を設定します。
5. ［OK］ボタンをクリックします。
6. 変更内容を保存し、プログラムを閉じます。



＜印刷－仕入先一覧＞プログラムの実行

1. F7 キーを押下して、＜印刷－仕入先一覧＞プログラムを実行させてください。

7．要約

本章では、基本的な帳票印刷プログラムの作成について学びました。

具体的には、次のことを実習を通して学びました。

- PDF として出力させるための環境設定
- APG 機能を利用した自動的作成
- 同等のプログラムの手作業による作成
- ［入出力ファイル］テーブルについて
- フォームのうち、帳票用に使われる、クラス >0 で、区分を定義するフォーム
- ［フォーム出力］処理コマンドの使い方
- 次の区分のフォームを使った帳票の拡張
  - ページヘッダ
  - ヘッダ
  - フッタ
  - ページフッタ
- 上記各フォームの役割
- ヘッダは 2 ページ目以降は自動的に出力されること
- 最初のヘッダとフッタを出力するロジックユニット
- ページヘッダフォームとページフッタフォームの出力設定を行う場所

次章では複雑な帳票出力プログラムについて学習します。

第 20 章 複雑な帳票

本章ではいくつかのデータレイヤで構成される複雑な帳票を扱います。

キーワード

- バッチタスク
- 入出力ファイル
- フォーム
- ［フォーム出力］処理コマンド
- 範囲条件
- イベントハンドリング
- プッシュボタン
- ［ブラウザ］コントロール

学習目標

本章を学ぶことで、次の内容に対する理解を深めましょう。

- 複雑な帳票印刷プログラムの作成方法
- 親タスクとサブタスクの入出力ファイル共有
- フォーム区分（ヘッダ、明細、フッタ）別の印刷出力用ロジックユニット
- ページヘッダ、ページフッタの設定
- バッチタスクでのサブタスクの使い方
- リッチクライアントプログラムからの帳票印刷プログラムの呼出し方
- リッチクライアントプログラムで PDF を表示する方法

参照

Magic ヘルプの［Magic xpa リファレンス／［フォーム］エディタ／［フォーム］エディタのカラム］

同様に、［Magic xpa リファレンス／出力フォーム］

同様に、［Magic xpa リファレンス／［ロジック］エディタ／処理コマンド］

１．はじめに

前章では単純な帳票印刷プログラムを作成しました。

本章では少なくとも２つ以上のデータレイヤからなる複雑な帳票印刷プログラムの作成方法について学びます。

複雑な帳票の一般的な例としては、受注データと受注明細行データを出力する帳票が考えられます。つまり受注データを出力するヘッダ部と、受注明細行を出力する明細部とで構成される帳票です。

本章では＜受注書印刷＞プログラムを実習により作成してゆきます。

２．複雑な帳票の概念

帳票を見たとき、単調なデータ出力に見えることが多いのですが、しかしよく見ると様々なフォーマットで編集されている場合があることに気づきます。

実際、一つの帳票内にいくつかのフォーマットを混在させることができるのです。これらのフォーマットのことをレイヤと呼ぶことにします。

たとえば、受注書の印刷出力を考えた場合、次のようなレイヤに分けて考えることができます。

- 一般情報 …… 顧客名や住所を含み、ページの最初に出力される
- 詳細情報 …… 受注した商品の一覧などで、ページの中央部に出力される
- 要約情報 …… 受注金額合計など、ページの最後に出力される

各レイヤは、それぞれの情報がもっとも見やすくなるよう工夫した構成とします。

各情報は、一つまたは複数のデータソースから読込まれ、そして一つまたは複数のタスクによって処理されます。

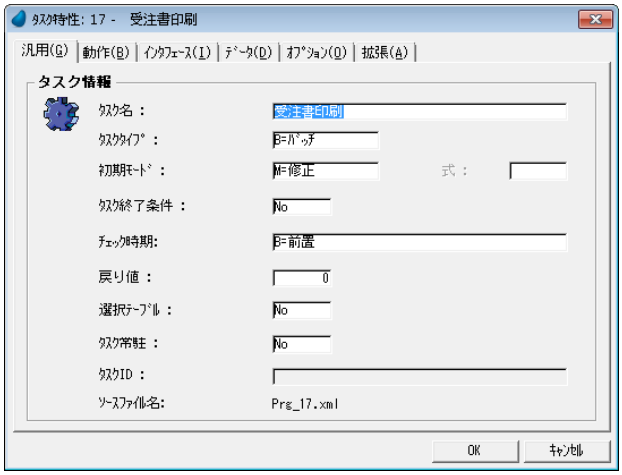
本章で作成する＜受注書印刷＞プログラムでは、＜受注＞、＜受注明細行＞、＜顧客＞、＜商品＞、＜取引先＞の 5 つのデータソースにアクセスします。

３．＜受注書印刷＞プログラムの作成

1. ［プログラム］リポジトリ（Shift+F3）を開きます。
2. 最終行（#16）にパークして F4 キーを押下して、一行作成します。
3. ［名前］カラムに **＜受注書印刷＞** と入力します。

4. F5 キーを押下してズームし、＜受注書印刷＞プログラムを開きます。

［タスク特性］ダイアログ（Alt+Enter）が開きます。

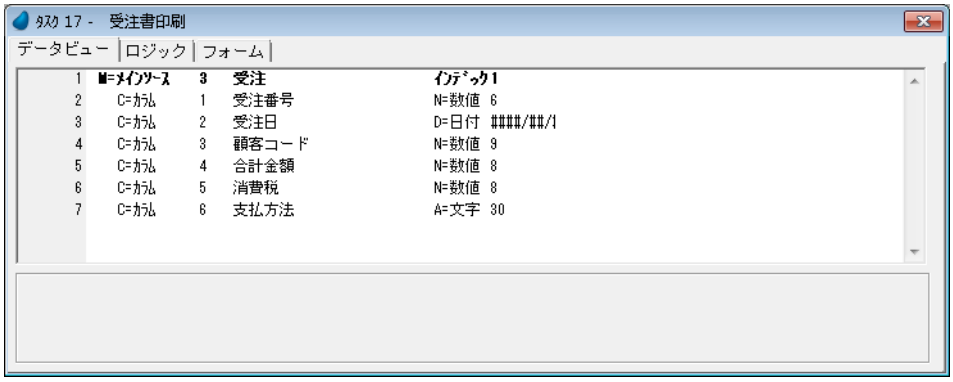


5. ［タスクタイプ］特性を［B= バッチ］とします。
6. ［OK］ボタンをクリックします。

タスクデータビューの定義

1. ［データビュー］エディタを開きます。
2. メインソースとして、＜受注＞（#3）データソースを定義します。
3. インデックスは **＜ 1 ＞**（受注番号）を指定します。

4. F4 キーを押下して一行作成します。
5. ［カラム番号］欄からズームして、［カラム選択］テーブルを開きます。
6. すべてのカラムを選択します。
7. ［選択］ボタンをクリックします。



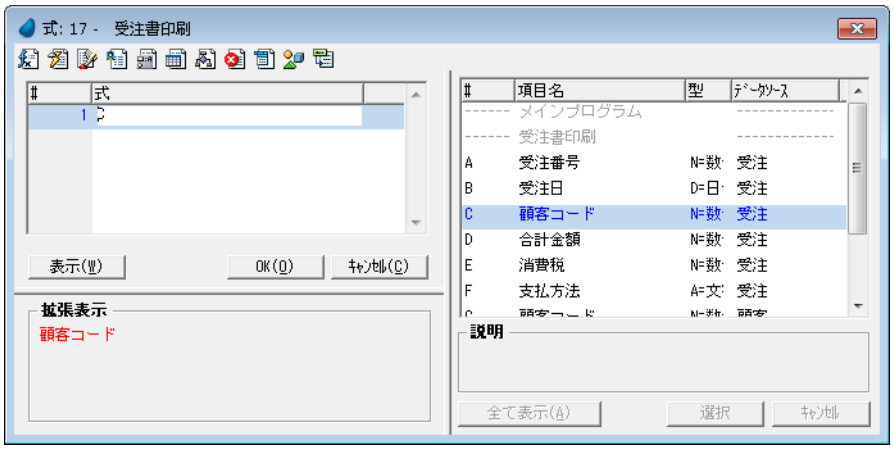
このセクションでは、顧客情報とリンクしてデータビューを拡張します。

8. ［データビュー］エディタの最終行（#7）にパークします。
9. ヘッダ行（Ctrl+H）を作成します。
10. ［L= 照会リンク］を選択し、リンク先として＜顧客＞（#1）データソースを設定します。
11. ［インデックス番号］欄からズームし、インデックス一覧を開き、＜顧客コード＞（#1）を選択します。

＜顧客コード＞（#9）カラム行が自動的に追加されます。

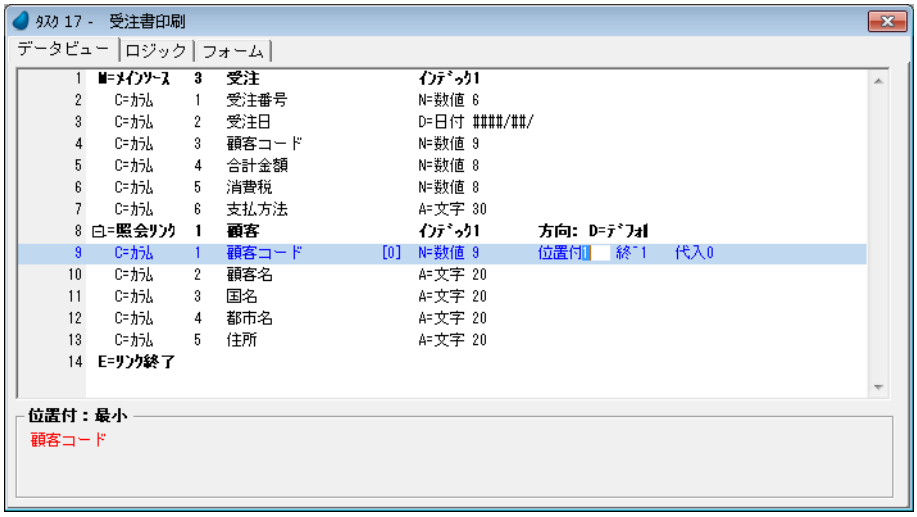
12. ＜顧客コード＞（#9）カラム行にパークし、F4 キーを押下して一行作成します。
13. 新しく作成された行の［カラム番号］欄からズームして、［カラム選択］テーブルを開きます。
14. カラム #2 から #5 までを選択します。
15. ［選択］ボタンをクリックします。
16. ＜顧客コード＞（#9）カラム行の［位置付］式欄にパークします。
17. ズームして［式］エディタを開きます。
18. F4 キーを押下して一行作成します。
19. ズームして項目リストに移動します。

20.受注データソースの＜顧客コード＞（#C）を選択します。



21.［OK］ボタンをクリックします。

22.［終了］式欄にも同じ式番号＜ 1 ＞を入力します。



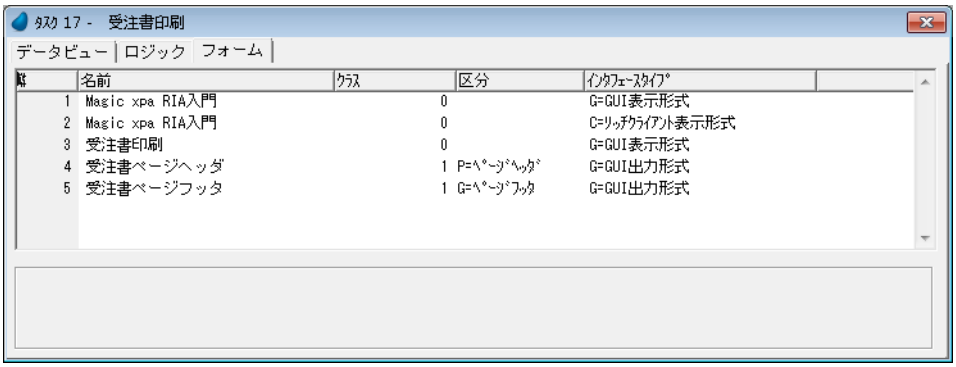
ページヘッダとページフッタの作成

ここで、ページヘッダとページフッタを定義しておきます。

1. ［フォーム］エディタを選択します。
2. 最終行にパークします。
3. 一行作成（F4）します。

4. ［名前］カラムに **＜受注書ページヘッダ＞** と入力してください。
5. 前章の内容と同様にして、ページヘッダフォームを作成してください。
6. もう一行作成（F4）します。
7. ［名前］カラムに **＜受注書ページフッタ＞** と入力してください。

8. 前章の内容と同様にして、ページフッタフォームを作成してください。

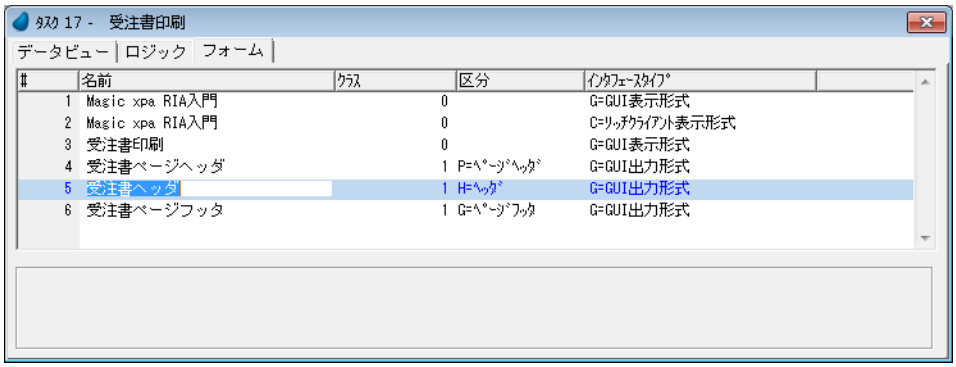


受注書ヘッダの作成

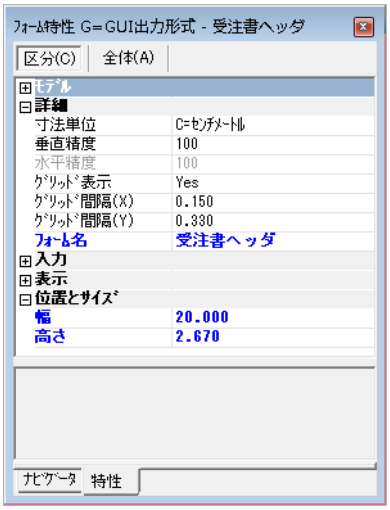
ここでは、ヘッダフォームを作成します。帳票名と受注タイトルを出力するものとします。

1. ［フォーム］エディタ
2. ＜受注書ページヘッダ＞（#4）の行にパークします。
3. 一行作成（F4）します。

4. ［名前］カラムに**＜受注書ヘッダ＞**と入力します。
5. ［区分］カラムで、［H= ヘッダ］を選択します。



6. フォーム特性（Alt+Enter）を開きます。
7. ［寸法単位］特性が ［C= センチメートル］であることを確認します。
8. ［幅］特性を＜ 20 ＞ に設定します。
9. ［高さ］特性を ＜ 2.67 ＞ に設定します。

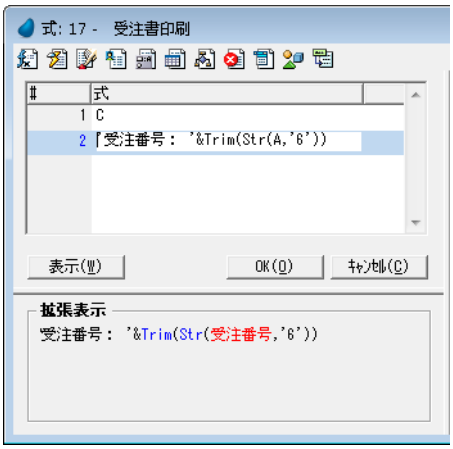


10. ズームしてヘッダフォームを開きます。
11. ［コントロール］パレットで［エディット］コントロールを選択し、フォームに配置します。

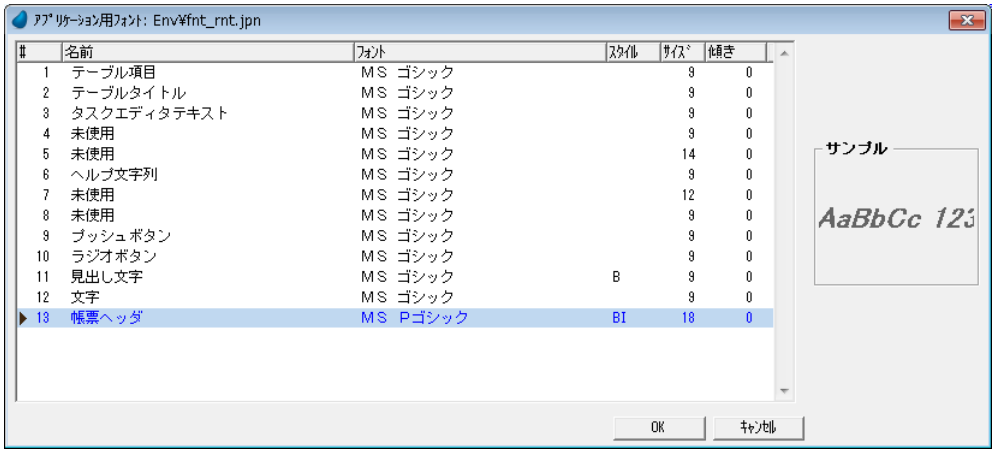
12.［エディット］コントロールの特性（Alt+Enter）を開きます。
13.［データ］特性で ［fx］ボタンを押下して、［式］エディタを開きます。
14.［式］エディタの最終行（#1）にパークします。
15.一行作成（F4）します。
16.次の文字列を入力します。

＜’受注番号： ’&TRIM(STR(A,’6’))＞

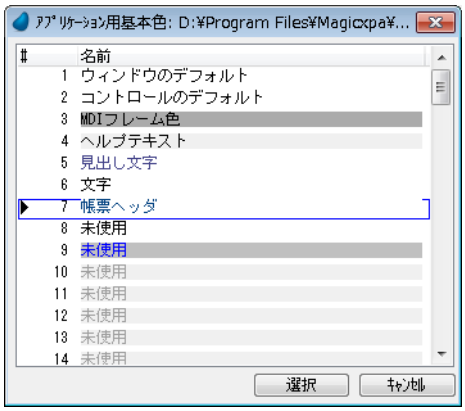
A は項目記号です。（ズームして項目リストから選択できます）



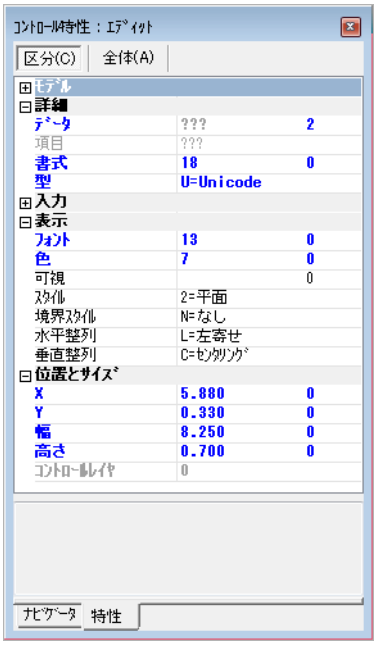
17.［OK］をクリックし、［式］エディタを閉じます。
18.［書式］特性を＜18＞に設定します。（18 バイトの文字列定義）
19.［フォント］特性に＜13＞を設定します。

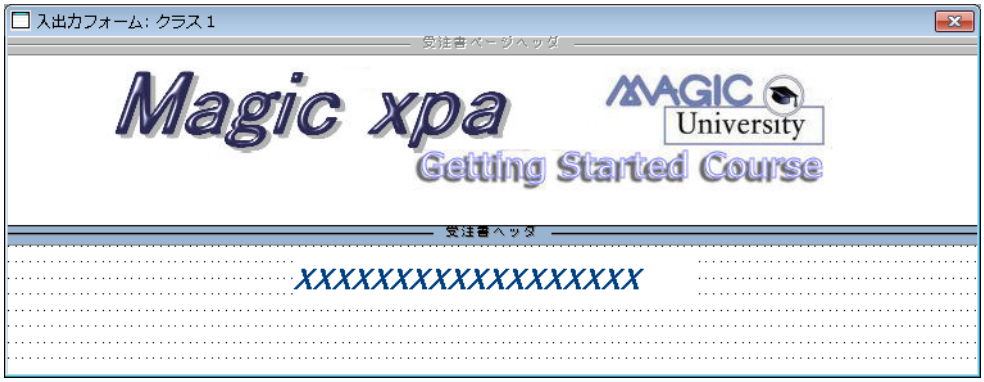


20.［色］特性に＜ 7 ＞を設定します。



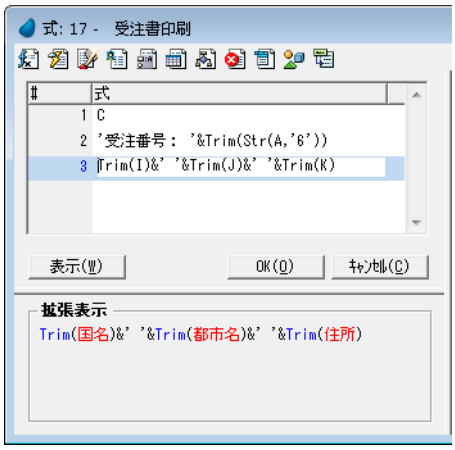
21.［幅］特性を＜8.25＞に設定します。
22.［高さ］特性を＜0.7＞に設定します。
23.［エディット］コントロール特性を閉じます。
24.［エディット］コントロールがマークされています。
25.［コマンド］パレットで、［水平センタリング］アイコンをクリックします。





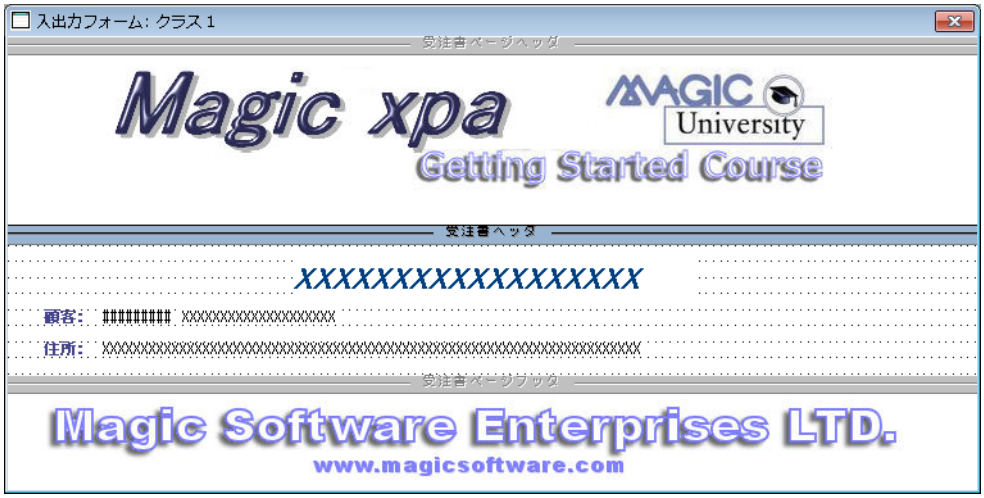
今回作成する例には、単に帳票名を出力するだけでなく、顧客の詳細情報も出力するようにします。

1. ［項目］タブをクリックし、項目一覧から、＜受注＞データソースの＜顧客コード＞（#C）を選択し、フォームに配置します。（345 ページのイメージ参照）
2. 次に、＜顧客＞データソースの＜顧客名＞（#H）を選択し、＜顧客コード＞の右に配置します。
3. ＜顧客名＞の見出し（［ラベル］コントロール）のみを選択します。
4. Delete キーを押下して、＜顧客名＞［ラベル］コントロールを削除します。
5. ＜顧客コード＞の見出し文字（［ラベル］コントロール）を選択し、特性を開きます。
6. ［テキスト］特性を編集し、**＜顧客：＞** とします。
7. ［フォント］特性を＜ 11 ＞（見出し文字）にします。
8. ［色］特性を＜ 5 ＞（見出し文字）にします。
9. 項目パレットから、＜住所＞（#K）を選択し、フォームの＜顧客：＞［ラベル］コントロールの下に配置します。
10. ＜住所＞の［エディット］コントロールを選択します。



11.［エディット］コントロールの［データ］特性で、設定されている値＜K＞を削除し、Tab キーを押下します（［fx］ボタンが表示されるようになります）。
12.［fx］ボタンを押下して［式］エディタを開き、最終行（#2）にパークします。
13.一行作成（F4）します。
14.次の文字列を入力します。
＜ Trim(I)&' '&Trim(J)&' '&Trim(K) ＞
ここで、各項目記号は、I : 国名、J : 都市名、K : 住所　に対応しています。

15.［書式］特性を＜ 70 ＞にします。
16.［フォント］特性を＜ 12 ＞（文字）にします。
17.［色］特性を＜ 6 ＞（文字）にします。
18.［コマンド］パレットで、［サイズ調整］アイコンをクリックし、書式の変更を反映させます。
19.＜住所＞の見出し文字（［ラベル］コントロール）を選択し、特性を表示します。
20.［テキスト］特性を編集し、＜住所：＞とします。
21.［フォント］特性を＜ 11 ＞（見出し文字）にします。
22.［色］特性を＜ 5 ＞（見出し文字）にします。
23.［コマンド］パレットで、［サイズ調整］アイコンをクリックし、変更を反映させます。
24.追加した項目の［エディット］コントロールの位置を調節し、イメージと同様になるよう整えてください。



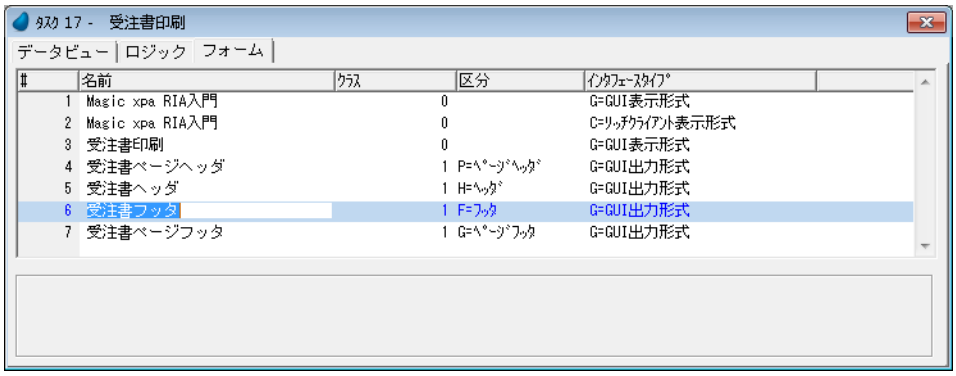
受注書フッタの作成

この例では、フッタに受注の合計金額と支払方法を出力するようにします。

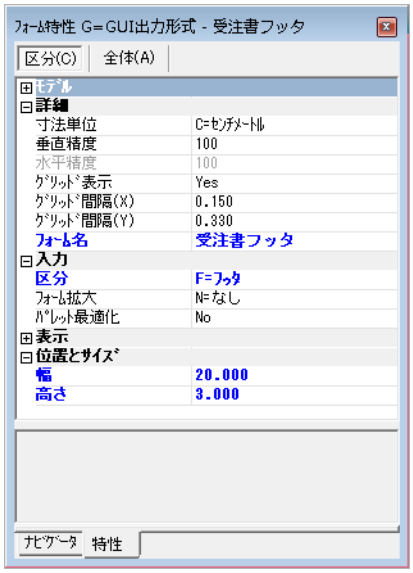
1. ［フォーム］エディタに戻ります。
2. ＜受注書＞（#5）ヘッダフォームにパークしています。
3. 一行作成（F4）します。

4. ［名前］カラムに＜受注書フッタ＞と入力します。

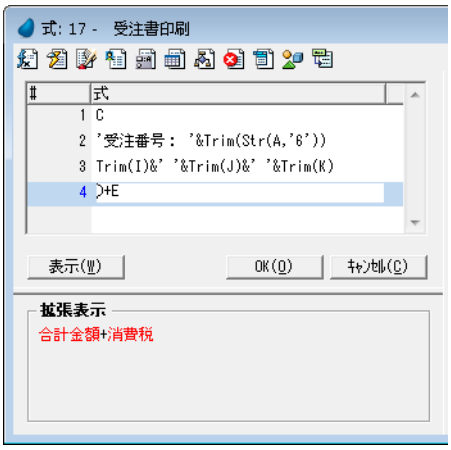
5. ［区分］カラムで、［F= フッタ］を選択します。



6. フォーム特性（Alt+Enter）を開きます。
7. ［寸法単位］特性が ［C= センチメートル］であることを確認します。
8. ［幅］特性を＜ 20 ＞に設定します。
9. ［高さ］特性を＜ 3 ＞に設定します。



10. ズームして＜受注書フッタ＞フォームを開きます。
11. 項目パレットから、＜合計金額＞（#D）と＜消費税＞（#E）の 2 つを順に選択・配置します。
12. ＜合計金額＞と＜消費税＞の各［エディット］コントロールを選択し、特性を開きます。
13. 複数選択特性の［書式］特性に＜ 8C ＞と入力し、［コマンド］パレットの［サイズ調整］アイコンをクリックします。
14. ［コントロール］パレットで［ラベル］コントロールを選択し、＜消費税＞のテキストの下に配置します。
15. ［ラベル］コントロールの特性を開きます。
16. ［テキスト］特性に＜総合計：＞と入力します。
17. ［コントロール］パレットで［エディット］コントロールを選択し、フォームの＜総合計：＞テキストの右に配置します。
18. ［エディット］コントロールの特性を開きます。
19. ［データ］特性で、［fx］ボタンをクリックし、［式］エディタを開きます。
20. 最終行（#3）にパークし、一行作成します。
21. 次の式を入力します：＜ D+E ＞（ここで、D: 合計金額、E: 消費税　です。）

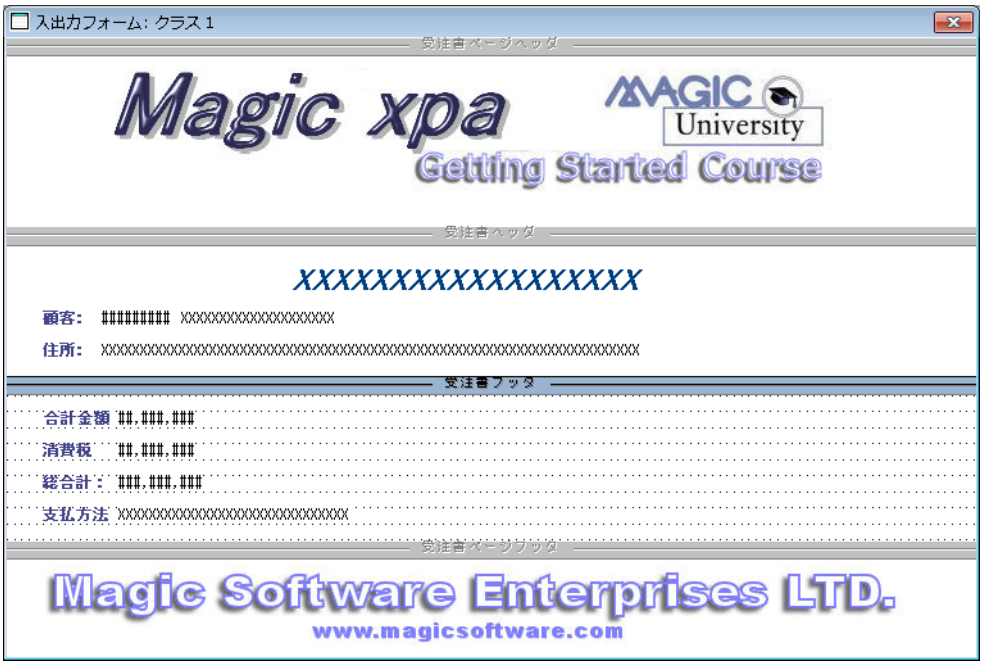


22. ［OK］をクリックし、［式］エディタを閉じます。
23. ［書式］特性で＜ 9C ＞と入力し、［コマンド］パレットの［サイズ調整］アイコンをクリックします。
24. ［水平整列］特性で［ R= 右寄せ］を選択します。
25. 項目パレットから＜支払方法＞（#F）を選択し、＜総合計：＞の［ラベル］コントロールの下に配置します。
26. すべての［ラベル］コントロールを選択します。
27. ［複数選択］特性が開いています。
28. ［フォント］特性を＜ 11 ＞（見出し文字）にします。

29.［色］特性を＜ 5 ＞（見出し文字）にします。
30.［コマンド］パレットの［サイズ調整］アイコンをクリックして、すべての文字が表示されるようにします。
31.すべての［エディット］コントロールを選択します。
32.［コマンド］パレットで、［選択左寄せ］アイコンをクリックし、左端をそろえます。
33.全体を右に移動し（→キーの押下またはマウスドラッグ）、位置を整えます。

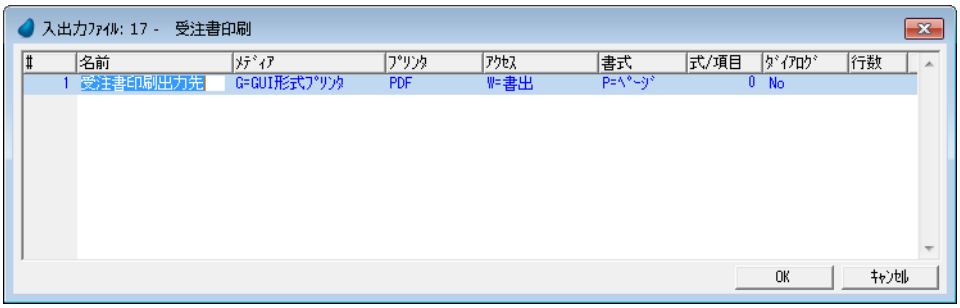
34.＜合計金額＞，＜消費税＞，＜総合計：＞の［エディット］コントロールを選択し、［選択右寄せ］アイコンをクリックします。
35.フォームの編集を終了し、変更内容を保存します。

完成したフォーム



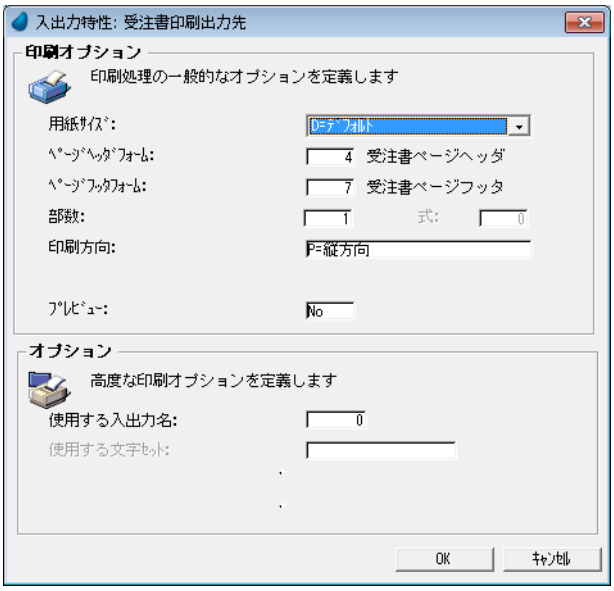
入出力ファイルの定義

1. ［タスク環境］メニューから［入出力ファイル］（Ctrl+I）を選択します。
2. ［入出力ファイル］テーブルで、F4 キーを押下して一行作成します。
3. ［名前］カラムに ＜受注書印刷出力先＞ と入力します。
4. ［プリンタ］カラムは＜ PDF ＞を選択します。



5. 入出力ファイル特性（Alt+Enter）を開きます。

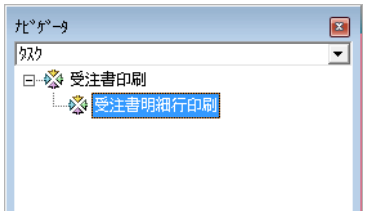
6. ［ページヘッダフォーム］特性に＜ 4 ＞を設定します。（＜受注書ページヘッダ＞を指定）
7. ［ページフッタフォーム］特性に＜ 7 ＞を設定します。（＜受注書ページフッタ＞を指定）
8. ［OK］をクリックし特性を閉じます。
9. ［入出力ファイル］テーブルで［OK］をクリックして閉じます。



受注書明細フォーム作成

受注書の明細フォームに出力するデータは、＜受注明細行＞データソースから読込まれるものです。したがって、ここからは受注詳細を印刷するためのサブタスクを作成します。

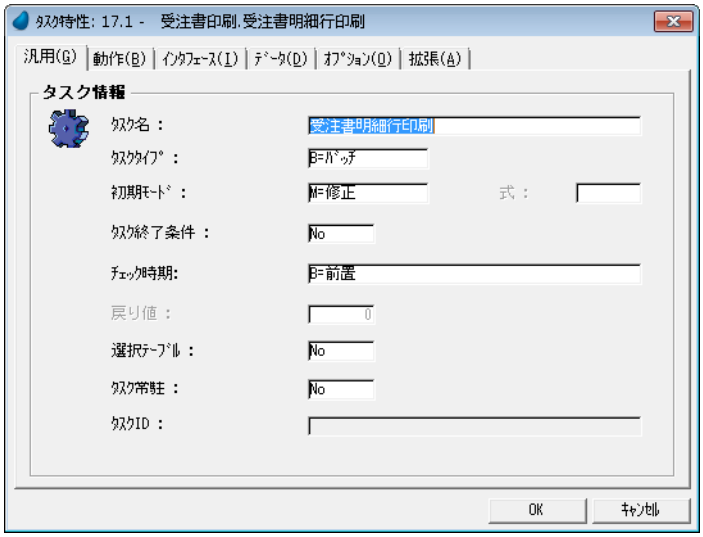
1. ［ナビゲータ］ペイン（タスク）で＜受注書印刷＞エントリをクリックし、パークします。
2. F4 キーを押下して新しいエントリ（サブタスク）を作成します。



［タスク特性］ダイアログが開くので、次のように設定します。

3. ［名前］特性に ＜受注書明細行印刷＞ と入力します。
4. ［タスクタイプ］特性で［B= バッチ］を選択します。

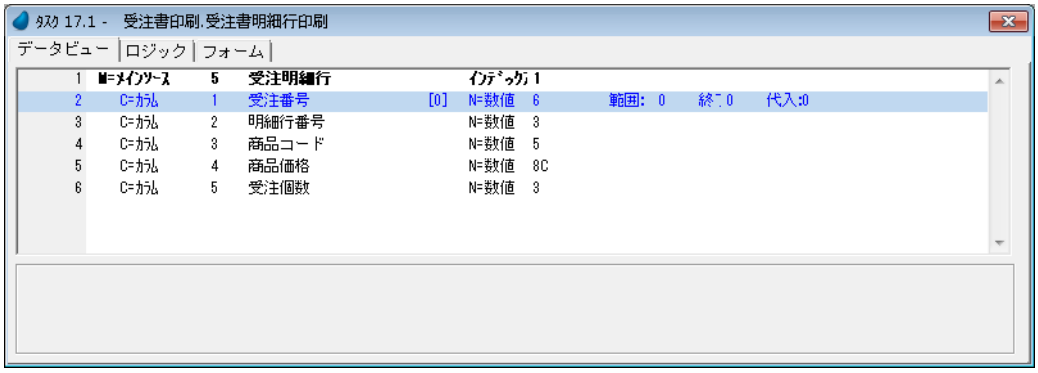
5. ［OK］ボタンをクリックします。



サブタスク＜受注書明細行印刷＞のデータビュー定義

1. ［データビュー］エディタを開きます。
2. メインソースとして、＜受注明細行＞（#5）データソースを選択します。
3. インデックスは、＜受注番号 & 行番号＞（#1）です。

4. F4 キーを押下して、カラム行を一行作成します。
5. ［カラム番号］欄からズームして［カラム選択］テーブルを開きます。
6. 全てのカラムを選択します。
7. ［選択］ボタンをクリックします。



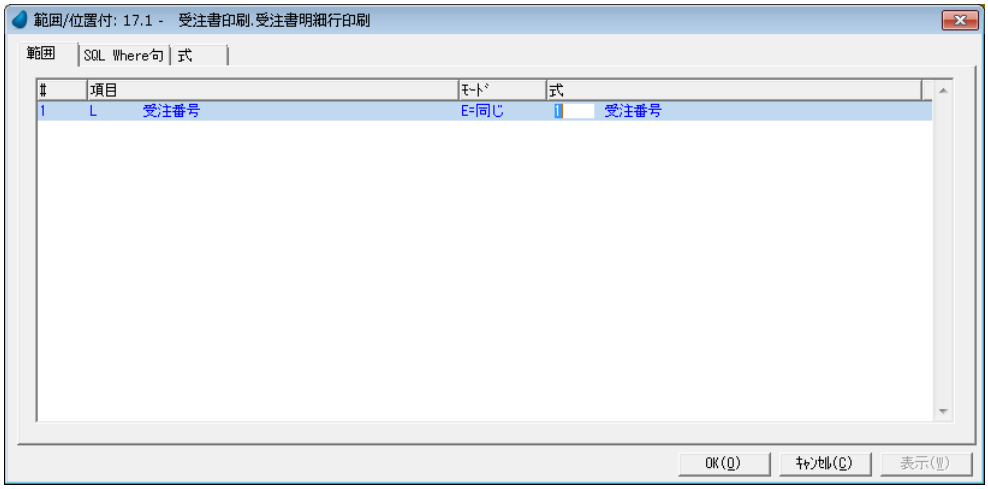
データビュー範囲の設定

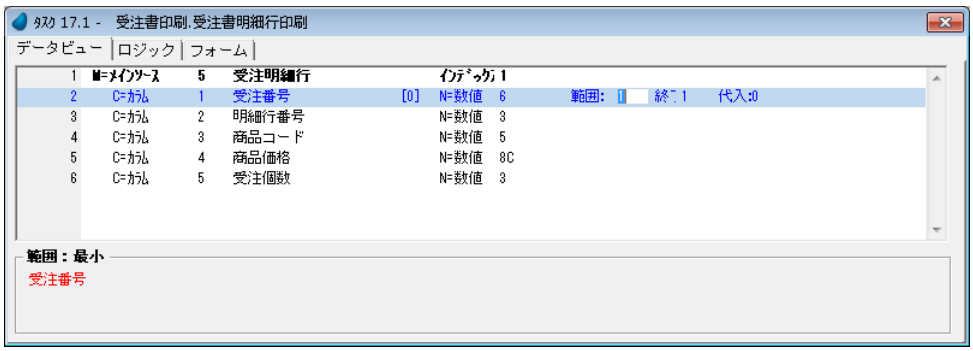
1. ＜受注番号＞（#2）カラム行にパークします。
2. ［範囲］の式番号欄からズームし、［式］エディタを開きます。
3. ［式］エディタで一行作成します。

4. ズームして項目リストから、＜受注＞データソースの＜受注番号＞（#A）を選択します。

5. ［OK］をクリックして［式］エディタを終了します。
6. ［終了］の式番号欄に、［範囲］欄と同じ式番号＜ 1 ＞を入力します。

7. ［タスク環境］メニューから［範囲 / 位置付］を選択（Ctrl+R）すると、［範囲 / 位置付］ウインドウが開くので、設定内容を確認できます。



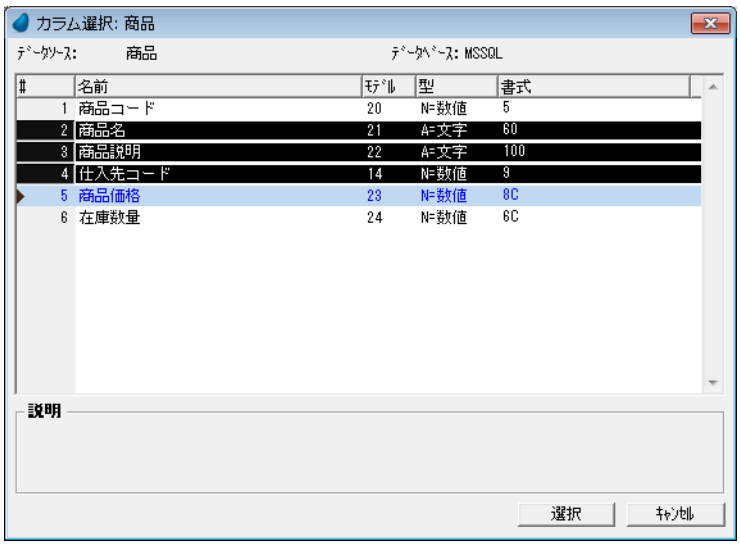


ここで＜商品＞データソースにリンクし、＜商品名＞と＜仕入先コード＞を取得するようにします。

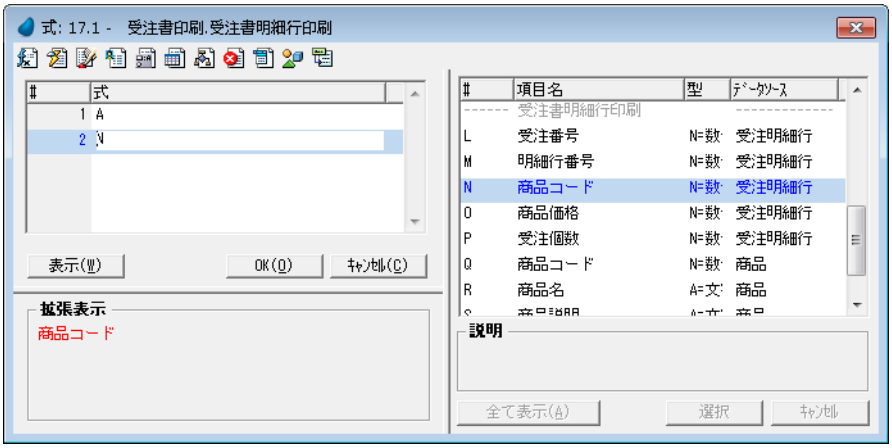
仕入先コードから仕入先名を取得する手続きは次ページで作成します。

1. ［データビュー］エディタの最終行にパークします。
2. ヘッダ行を一行作成（Ctrl+H）します。
3. ［L= 照会リンク］を定義し、リンク先データソースとして、＜商品＞（#4）を設定します。

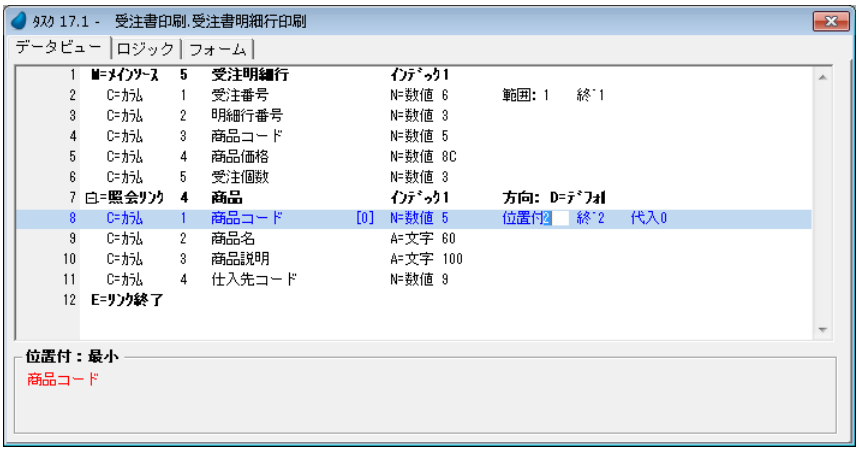
4. ［インデックス］欄からズームしてインデックス一覧を開き、＜商品コード＞（#1）を選択します。＜商品コード＞カラムが自動的に登録されます。
5. ＜商品コード＞（#8）カラム行にパークし、F4 キーを押下して一行作成します。
6. ［カラム番号］欄からズームして［カラム選択］テーブルを開きます。
7. カラム番号 #2 から #4 までを選択します。



8. ［選択］ボタンをクリックします。
9. ＜商品コード＞（#8）カラムの行の［位置付］式番号欄にパークします。
10.ズームして［式］エディタを開きます。
11.#1 行目にパークして、一行作成（F4）します。
12.ズームして項目リストに移動します。
13.＜受注明細行＞データソースの＜商品コード＞（#N）を選択します。

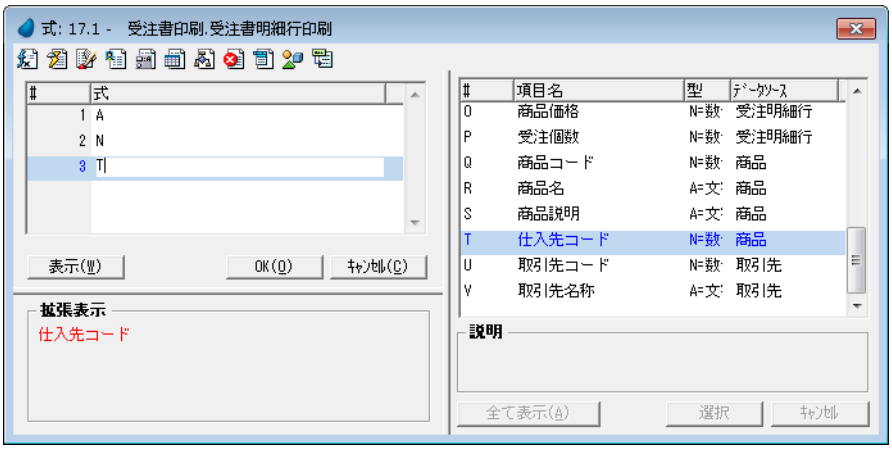


14.［OK］ボタンをクリックします。
15.［終了］の式番号欄にも、［位置付］と同じ＜ 2 ＞を設定します。



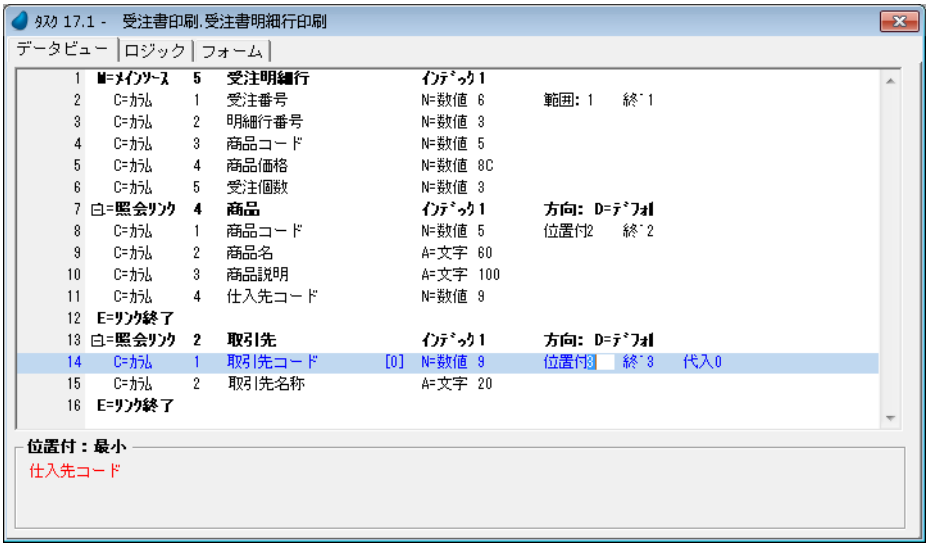
16.［データビュー］エディタの最終行（#12）にパークします。
17.ヘッダ行（Ctrl+H）を一行作成します。
18.［L= 照会リンク］を定義し、リンク先データソースとして、＜取引先＞（#2）を設定します。
19.［インデックス］欄からズームしてインデックス一覧を開き、＜取引先コード＞（#1）を選択します。＜取引先コード＞カラムが自動的に登録されます。
20.＜取引先コード＞（#14）カラム行にパークし、F4 キーを押下して一行作成します。
21.［カラム番号］欄からズームして［カラム選択］テーブルを開きます。
22.カラム番号＜取引先名称＞（#2）を選択します。
23.［選択］ボタンをクリックします。
24.＜取引先コード＞（#14）カラムの行の［位置付］式番号欄にパークします。
25.ズームして［式］エディタを開きます。
26.#2 行目にパークして、一行作成（F4）します。
27.ズーム（F5）して項目リストに移動します。
28.＜商品＞データソースの＜仕入先コード＞（#T）を選択します。

29.［OK］ボタンをクリックします。



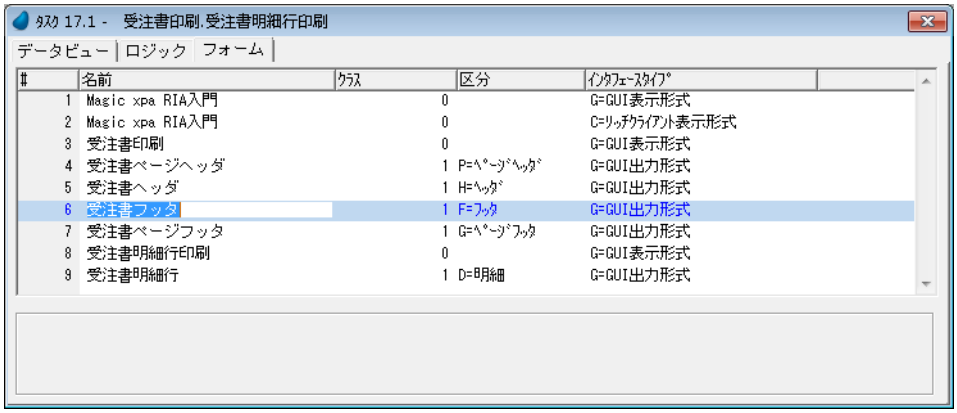
30.［終了］の式番号欄にも、［位置付］と同じ＜ 3 ＞を設定します。

31.変更内容を保存しましょう。

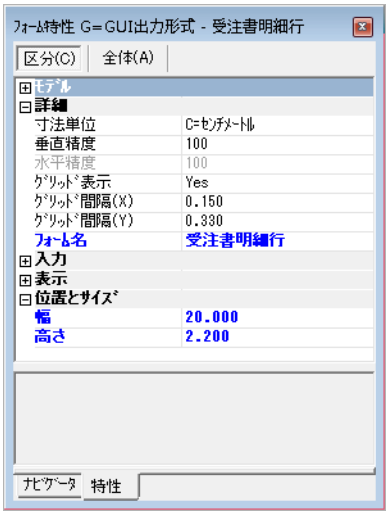


受注書明細行フォームの作成

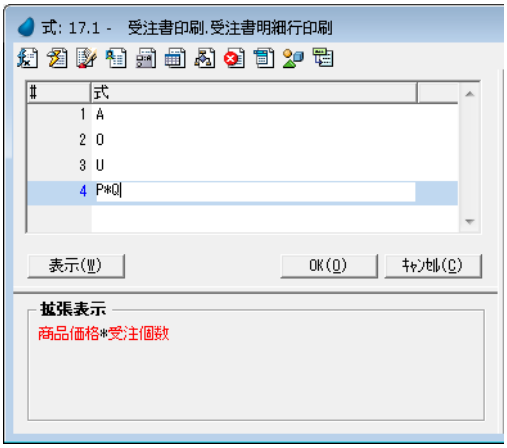
1. ［フォーム］エディタを開きます。
2. 最終行にパークし、一行作成（F4）します。
3. ［名前］カラムに **＜受注書明細行＞** と入力します。



4. フォーム特性（Alt+Enter）を開きます。
5. ［寸法単位］特性が ［C= センチメートル］であることを確認します。
6. ［幅］特性を＜ 20 ＞に設定します。
7. ［高さ］特性を＜ 2.2 ＞に設定します。



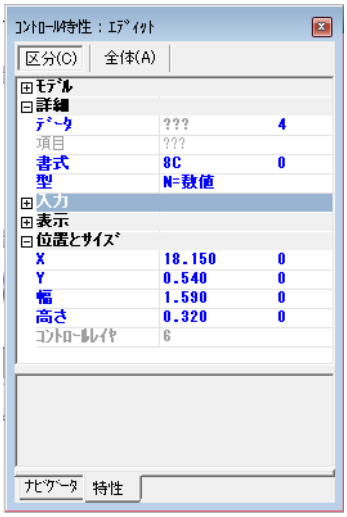
8. ズームして＜受注明細行＞フォームを開きます。
9. ［テーブル］コントロールをフォームに配置します。
10. 次の項目をテーブルに配置します。
    a. ＜明細行番号＞（#M）
    b. ＜商品名＞（#R）
    c. ＜取引先名称＞（#V）
    d. ＜受注個数＞（#P）
    e. ＜商品価格＞（#O）
11. ［エディット］コントロールを一つテーブルに追加します。
12. 追加した［エディット］コントロールをマークします。
13. ［エディット］コントロールの特性（Alt+Enter）を開きます。
14. ［データ］特性の［fx］ボタンをクリックして、［式］エディタを開きます。
15. ［式］エディタで最終行にパークして一行作成します。
16. 次の式を入力します：＜ P*0 ＞（ここで、P：商品価格，O：受注個数）



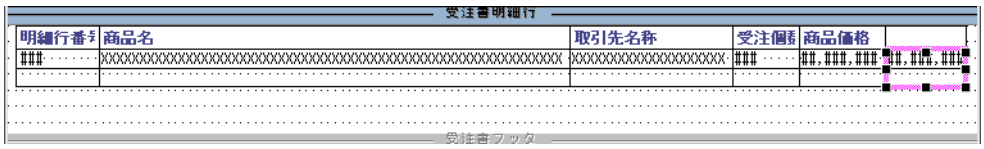
17.[OK］をクリックします。

18.［書式］特性に＜ 8C ＞と設定します。

19.［コマンド］パレットの［サイズ調整］アイコンをクリックし、［幅］特性が＜ 1.590 ＞となったことを確認します。



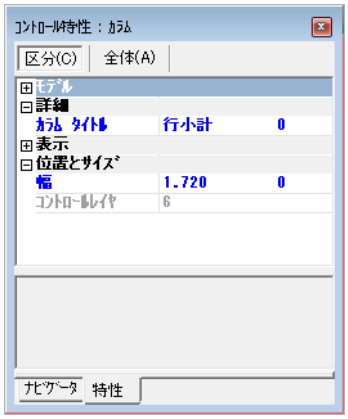
20.［エディット］コントロールを配置したカラムを選択します。（Alt キーを押下しながら、［エディット］コントロールの下のカラムをクリックします。）



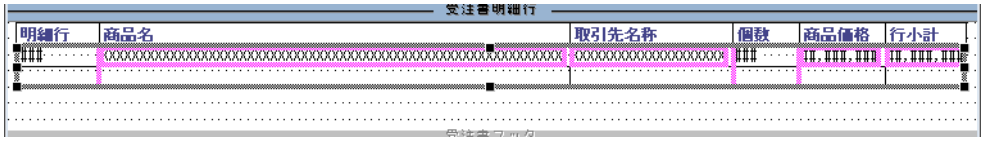
21.カラム特性（Alt+Enter）を開きます。

22.［カラムタイトル］特性に＜行小計＞と入力します。

23.［幅］特性に＜ 1.72 ＞と入力します。



24.すべてのデータルカラムを選択します。



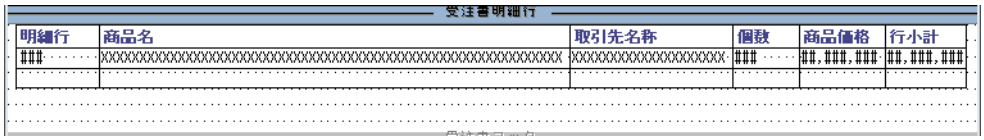
25.［複数選択］特性（Alt+Enter）を開きます。

26.［フォント］特性に＜ 11 ＞（見出し文字）を設定します。

27.［色］特性に＜ 5 ＞（見出し文字）を設定します。

28.カラムタイトル＜明細行番号＞と＜受注個数＞がカラムの幅からあふれるので、それぞれ＜明細行＞，＜個数＞に変更しましょう。

29.変更内容を保存し、［フォーム］エディタを終了します。



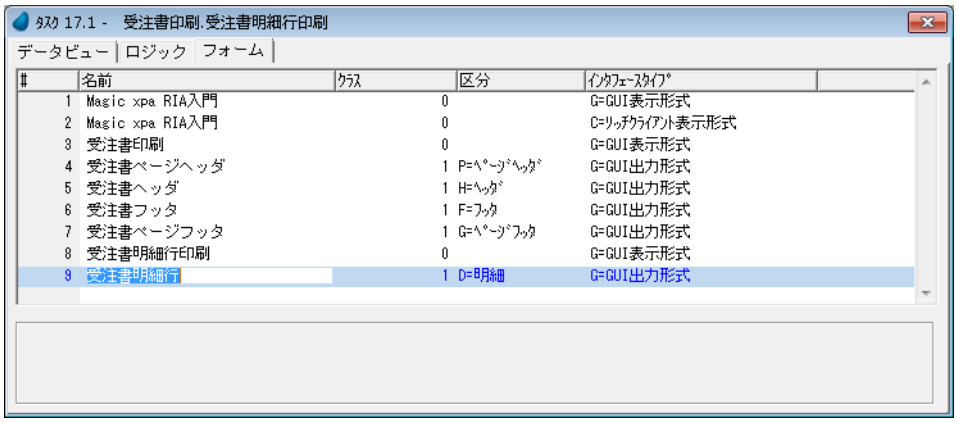
受注書明細行フォームの出力処理

1. ［ロジック］エディタを選択します。
2. ヘッダ行を一行作成します。
3. ［レコード後］ロジックユニットを定義します。

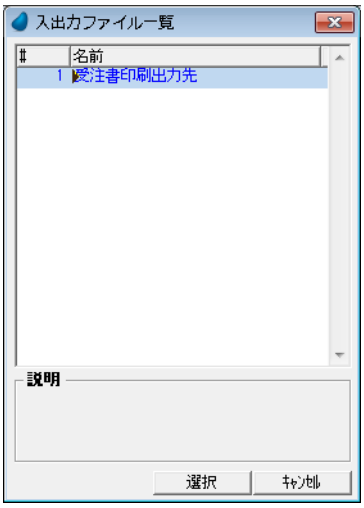
4. F4 キーを押下して、コマンド行を一行作成します。
5. 次のように設定します。

コマンド	入／出力	フォーム	ファイル	条件
フォーム	O= 出力	9（受注書明細行）	1（受注書印刷出力先）	Yes

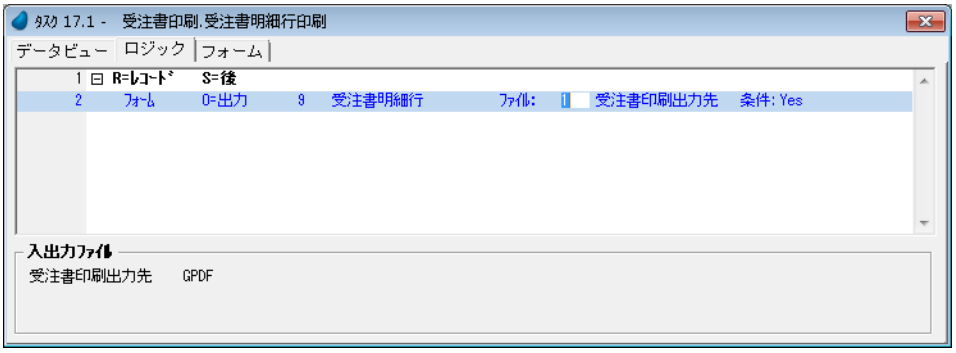
［フォーム］欄からズームしてフォーム一覧から選択する場合のイメージ：



［ファイル］欄からズームして［入出力ファイル］テーブルから選択する場合のイメージ：



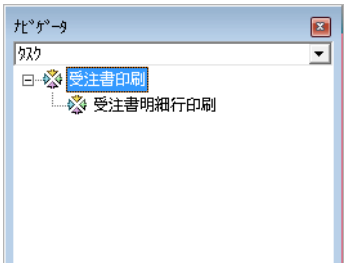
［ロジック］エディタ

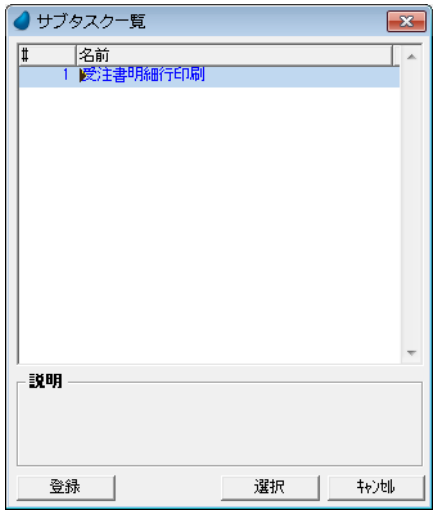


受注書ヘッダと受注書フッタの出力処理

1. ［ナビゲータ］ペインで、＜受注書印刷＞タスクエントリをクリックします。
2. ［ロジック］エディタを選択します。
3. ヘッダ行（Ctrl+H）を一行作成します。

4. ［レコード後］ロジックユニットを作成します。
5. F4 キーを 3 回押下してコマンド行を三行作成します。
6. それぞれの行で、次のように設定します。





コマンド	タイプ			条件
フォーム	O= 出力	5（受注書ヘッダ）	1（受注書印刷出力先）	Yes
コール	S= サブタスク	1（受注書明細行印刷）	［0 パラメータ］	Yes
フォーム	O= 出力	6（受注書フッタ）	1（受注書印刷出力先）	Yes

ここまでの要約

ここまでの実習によって、複雑な帳票をどのような手順で作成するかを学びました。

この例では、帳票のヘッダとフッタは親タスクで作成し、帳票の明細をサブタスクで作成しました。

いずれのフォームも親タスクで定義した入出力ファイルに出力を行なうようにしました。

前章の単純な帳票出力のプログラムでは、ヘッダとフッタはそれぞれ［タスク前］ロジックユニットと［タスク後］ロジックユニットで出力しましたが、この例ではいずれも親タスクの［レコード後］ロジックユニットで出力しています。

今回の例では、＜受注＞データソースから取得するレコード毎の情報をヘッダとフッタで使用しているためです。

明細行のデータはサブタスクで処理されるので、ヘッダ出力を行なった後、フッタ出力を行なう前のところでサブタスクをコールしました。

サブタスクでは、＜商品＞データソースと＜取引先＞データソースにリンクしてデータビューを拡張しました。

すでに気づいているかもしれませんが、データソース（この例では＜取引先＞）にリンクするとき別のリンク先データソース（この例では＜商品＞）により取得した情報を元にして接続することができます。

出力される帳票は、すべての受注情報とその明細を含んだ、複雑なものとなっています。

帳票印刷プログラムの呼び出し

帳票印刷プログラムを呼び出す方法は基本的に２通りあります。

- 静的な条件下（追加情報を必要としない）で実行される場合で、メニューから起動される
- 追加の条件の指定を受けて実行される場合で、リッチクライアントプログラムから起動される

前のセクションで作成した＜受注書印刷＞プログラムは、ある受注番号に対して帳票出力を行なうようにしたいので、ここでは＜受注管理＞プログラムから呼び出すようにします。

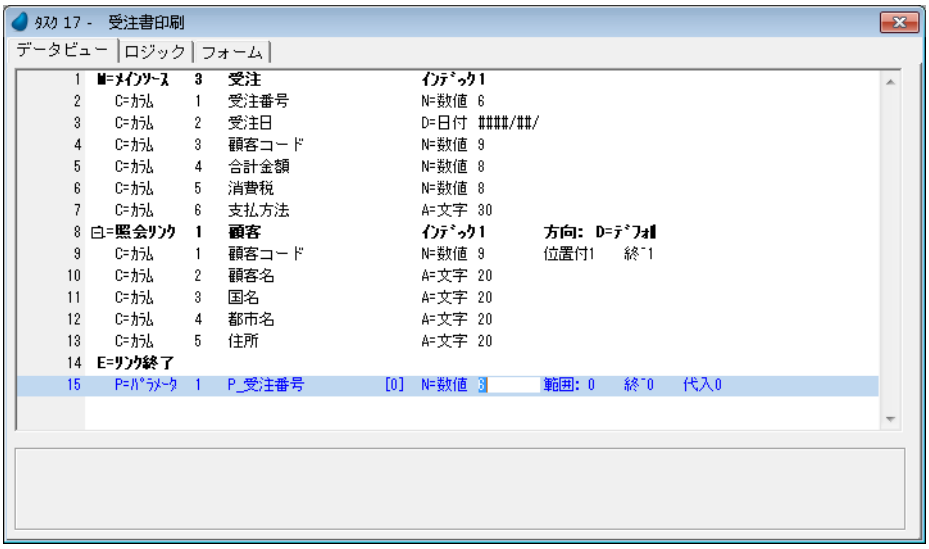
そのとき、＜受注番号＞をパラメータとして＜受注書印刷＞プログラムに渡すようにします。

＜受注番号＞パラメータの追加

1. ＜受注書印刷＞（#17）プログラムにズームして開きます。
2. ［データビュー］エディタを選択します。
3. 最終行にパークし、一行作成（F4）します。

4. 次のように設定します。

#	項目タイプ	項目番号	名前	モデル	型	書式	範囲	代入
15	P= パラメータ	15	P_ 受注番号	0	N= 数値	6	0	0

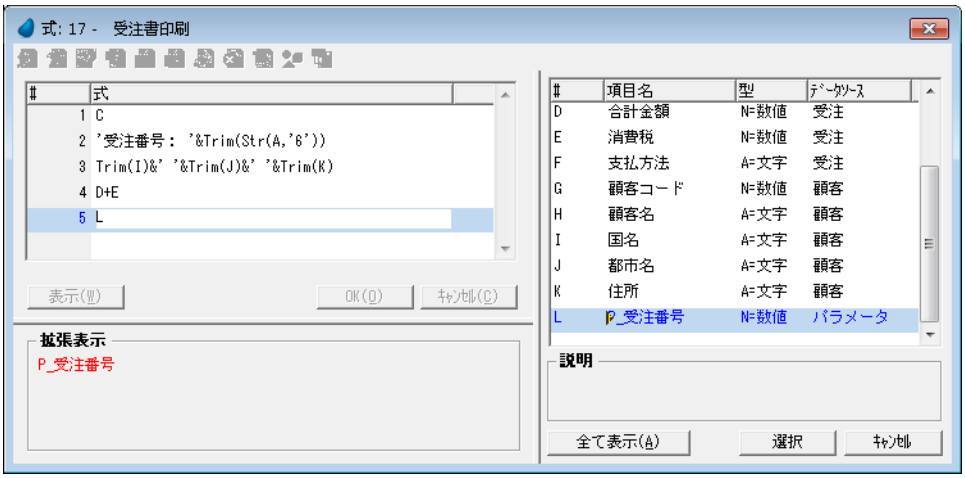


タスクデータビュー範囲の設定

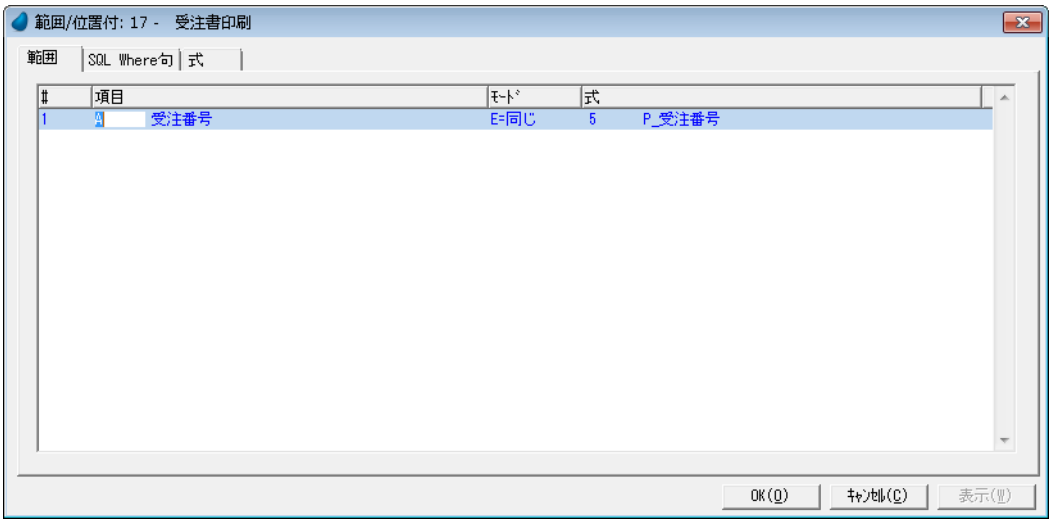
1. ＜受注番号＞（#2）のカラム行にパークします。
2. ［範囲］の式番号欄からズームして［式］エディタを開きます。
3. 最終行（#4）にパークし、F4 キーを押下して一行作成します。

4. ズームして項目リストに移動します。

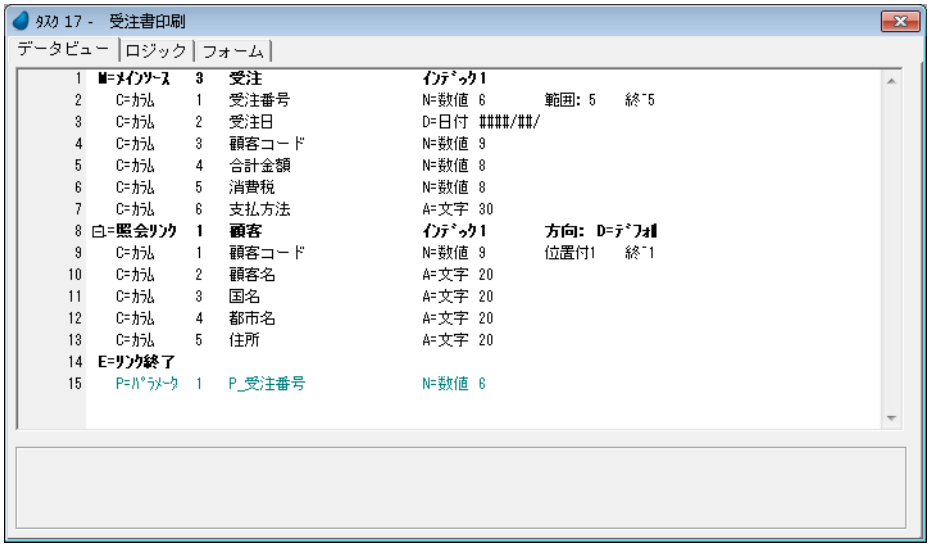
5. パラメータ＜ P_ 受注番号＞（#L）を選択します。



6. ［OK］ボタンをクリックします。
7. ［終了］の式番号欄にも［範囲］と同じ＜ 5 ＞を設定します。
8. Ctrl+R キーを押下して、［範囲 / 位置付］テーブルを表示し、条件の確認ができます。



9. 変更内容を保存し、プログラムを閉じます。



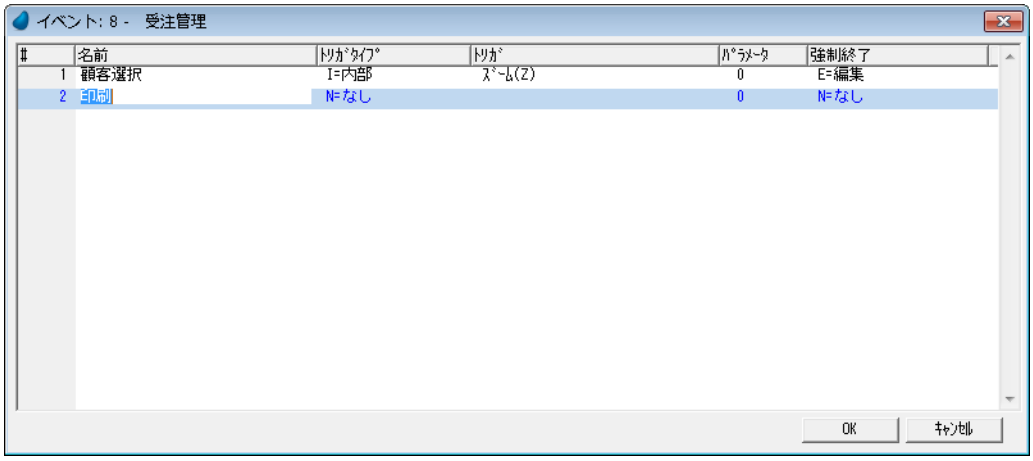
印刷イベントの作成

＜受注管理＞プログラムに＜印刷＞イベントを作成します。この＜印刷＞イベントのハンドラが＜受注書印刷＞プログラムを呼び出し、そのとき＜受注番号＞をパラメータとして渡します。

プッシュボタンをこの＜印刷＞イベントのトリガとして利用します。

1. ＜受注管理＞（#8）プログラムにズームします。
2. ［タスク環境］メニューから［ユーザイベント］（Ctrl+U）を選択し、［ユーザイベント］テーブルを開きます。

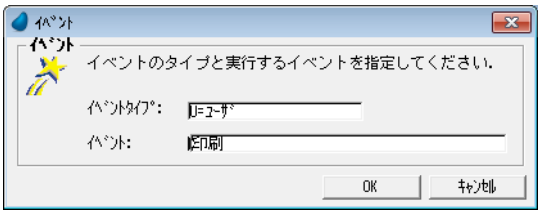
3. 最終行のあとに一行作成します。
4. ［名前］カラムに**＜印刷＞**と入力します。
5. ［トリガタイプ］カラムでは［N= なし］を選択します。
6. ［OK］をクリックして［イベント］テーブルを閉じます。

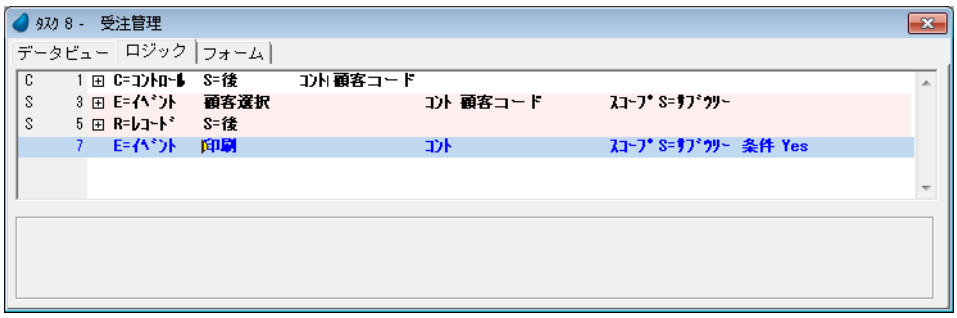


印刷イベントのためのイベントロジックユニット定義

1. ［ロジック］エディタを選択します。
2. 最終行（#6）にパークします。
3. ヘッダ行を一行作成（Ctrl+H）します。

4. 印刷用の［イベント］ロジックユニットを定義します。
   a. ズームして［イベント］ダイアログを開きます。
   b. ［イベントタイプ］として［U= ユーザ］を選択します。
   c. ［イベント］欄からズームして［ユーザイベント］テーブルを開き、＜印刷＞（#2）を選択します。



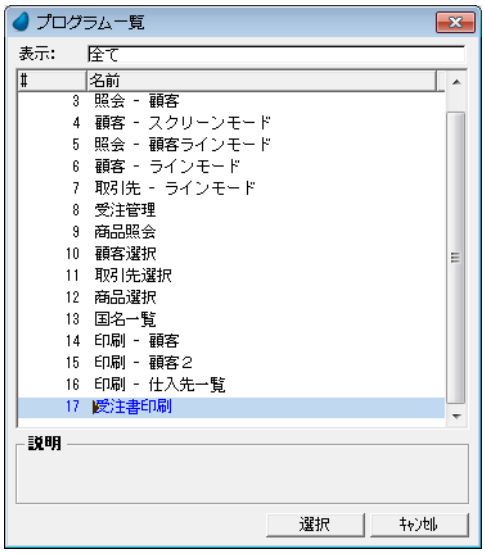


5. F4 キーを押下してコマンド行を一行作成します。
6. 次のように設定します。

コマンド	プログラム番号	パラメータ	戻り値	条件
コール P= プログラム	17 （受注書印刷）	1 （#A 受注番号）	???	Yes

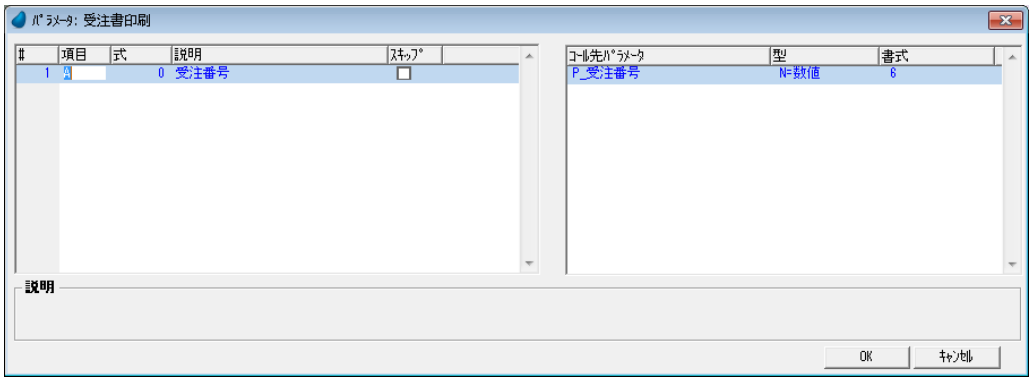
プログラム番号の設定

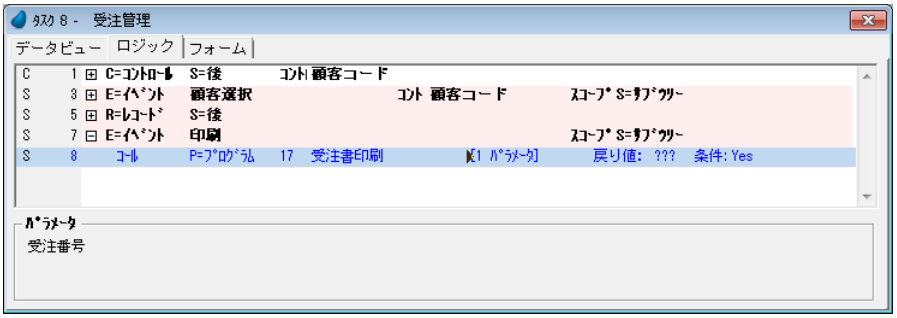
7. ［番号］欄からズームしてプログラム一覧を開きます。
8. ＜受注書印刷＞（#17）プログラムを選択します。



パラメータの設定

9. ズームして［パラメータ］テーブルを開きます。
10. F4 キーを押下してパラメータエントリを一行作成します。
11. ズームして項目一覧を開き、＜受注番号＞（#A）を選択します。

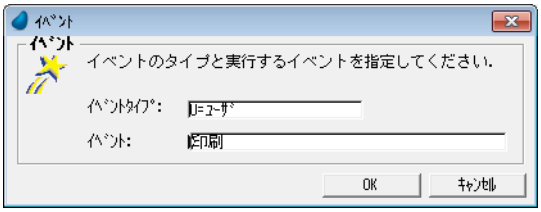


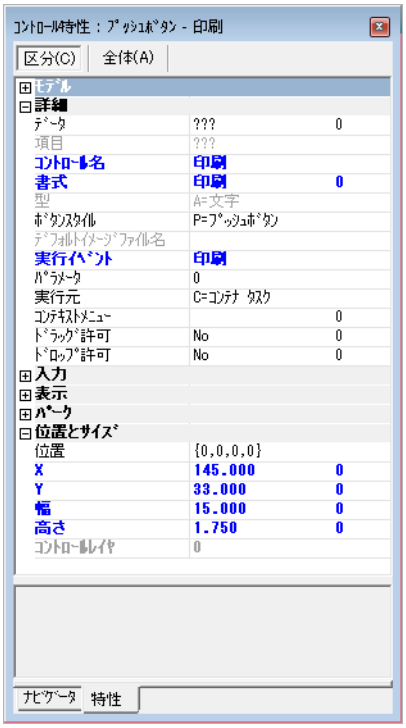


プッシュボタンの追加

1. ［フォーム］エディタを開きます。
2. ＜受注管理＞フォームにズームします。
3. ［コントロール］パレットで［プッシュボタン］コントロールを選択し、フォームの右下に配置します。

4. ［プッシュボタン］コントロールの特性を開きます。
5. ［書式］特性に**＜印刷＞**と入力します。
6. ［実行イベント］特性でズームし、［イベント］ダイアログで［イベントタイプ］で［U= ユーザ］を選択します。
7. ［イベント］特性でズームし、イベント一覧から＜印刷＞（#2）を選択します。





8. ［幅］特性に＜ 15 ＞を設定します。
9. ［高さ］特性に＜ 1.75 ＞を設定します。

プログラム実行と結果確認

1. ＜受注管理＞プログラムを実行します。
2. 受注番号 1 番のデータにパークします。
3. ＜印刷＞ボタンをクリックします。

＜印刷＞ボタンをクリックすることによって、＜ C:¥Program Files¥Magicxpa¥Studio 2.3¥Projects¥Magic xpa RIA¥PDF ＞フォルダに「Magic - 受注書印刷出力先 .pdf」というファイルが作成されます。PDF ビューワーで表示させると以下のように出力されたことが確認できます。

これで受注書の出力プログラムが作成されました。しかし、PDF はサーバ側に作成されるため、クライアントから出力内容を確認することができません。作成された PDF を表示させるリッチクライアントタスクを作成してみましょう。

４．受注書表示タスクの作成

リッチクライアントタスクで PDF を表示させるには、次の２つの機能を持つプログラムを作成します。

- ［ブラウザ］コントロールの使用。これは、IE（Internet Explorer）と同じような機能を持つコントロールです。
- サーバ側のファイルをクライアントにコピーする。これを実現するために、Magic xpa には ServerFileToClient() というサーバ側のファイルをクライアントにコピーする関数が用意されています。

［ブラウザ］コントロールと ServerFileToClient() 関数を組み合わせることでサーバ上の PDF をクライアントにコピーして表示させることができます。

ここでは、＜受注管理＞プログラムのサブタスクとして定義し、＜受注書印刷＞プログラムを呼び出した後で実行するようにしましょう。

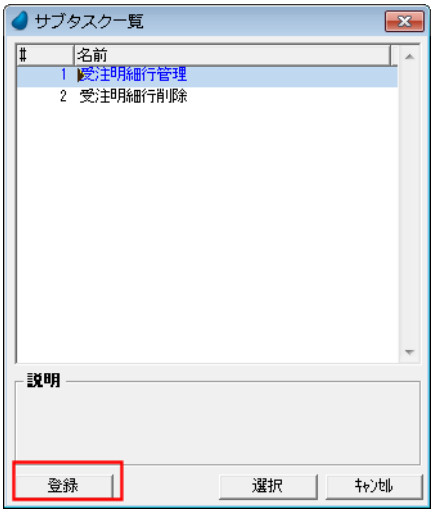
サブタスクの追加

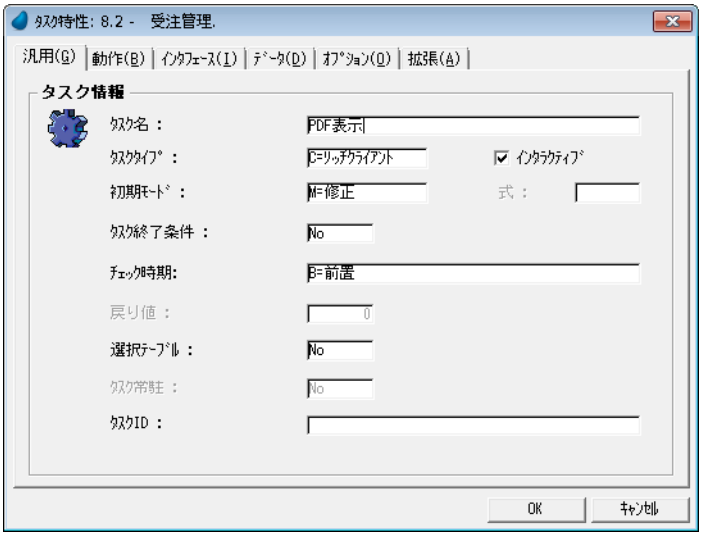
1. ＜受注管理＞プログラムを開きます。
2. ［ロジック］エディタを選択します。
3. 最終行（#8）にパークします。
4. 一行作成（F4）します。

5. 以下のように設定して［コールサブタスク］処理コマンドを定義します。

コマンド	プログラム番号	パラメータ	戻り値	条件
コール S= サブタスク			???	Yes

6. ［プログラム番号］でズーム（F5）して［サブタスク一覧］を表示させます。
7. ［登録］ボタンをクリックしてタスクを追加します。新規のタスク特性が開きます。
8. ［タスク名］には、**＜ PDF 表示＞**と入力します。リッチクライアントタスクの子タスクのため、自動的にリッチクライアントタスクになっています。

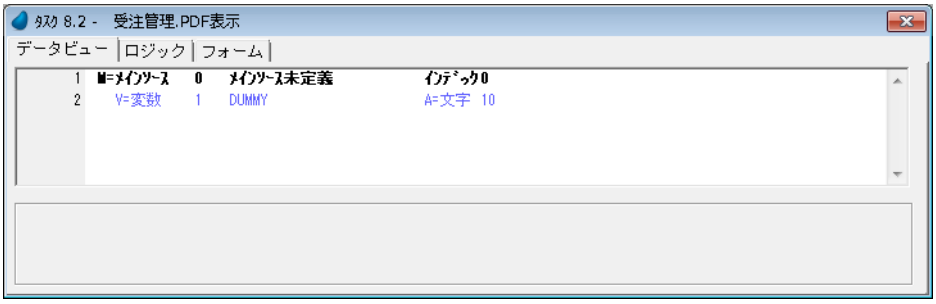




9. ［OK］をクリックしてタスク特性を閉じると［データビュー］エディタが表示されます。
10. 一行追加（F4）して以下のような変数項目を追加します。

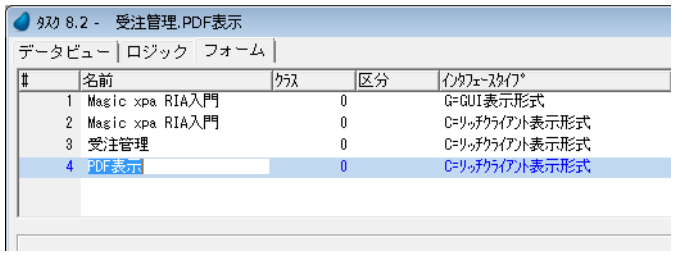
コマンド	名前	モデル	型	書式	代入
V= 変数	DUMMY	0	A= 文字	10	

リッチクライアントタスクには、1 つ以上のデータビューが必要になるため、この変数項目を定義しています。タスク内で使用していません。

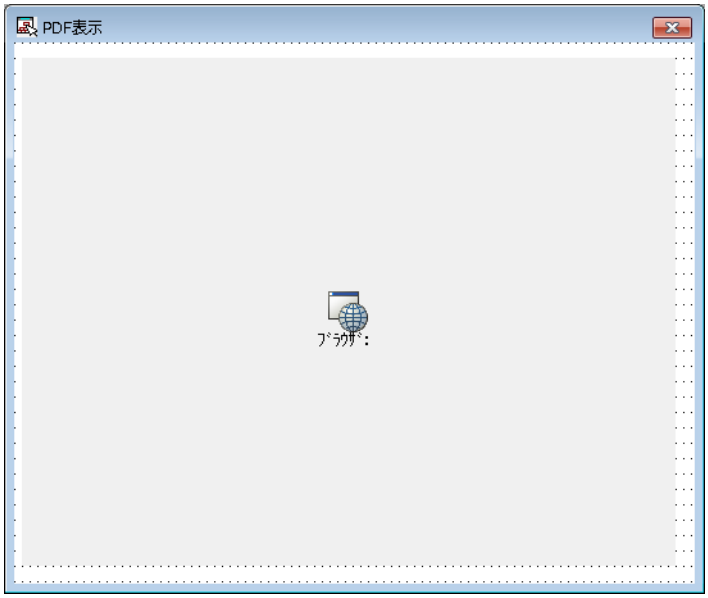


フォームの定義

11. ［フォーム］エディタ（Ctrl+3）に切り替えます。
12. ［PDF 表示］という名前でメインフォームが定義されています。ここからズーム（F5）してフォームを開きます。



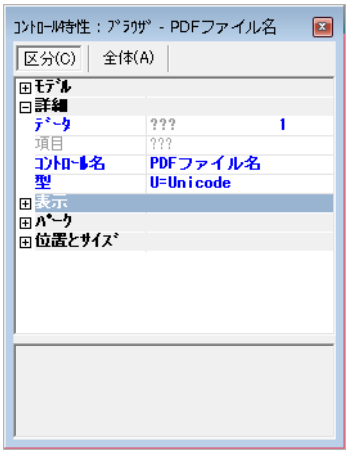
13.フォームに［ブラウザ］コントロールを配置します。コントロールパネルで、 をクリックし、フォームにドロップすることで配置されます。



14.［ブラウザ］コントロールのコントロール特性（Alt+Enter）を開きます。［データ］特性の式からズーム（F5）して、［式］エディタを開きます。

15.一行追加（F4）して、以下の式を定義します。

ServerFileToClient('%PDF%Magic - 受注書印刷出力先.pdf')



コントロール特性

［式］エディタ

フォームのサイズやコントロールのサイズは、好みで設定してください。

ここでは PDF ファイル名を固定値（'Magic - 受注書印刷出力先.pdf'）で指定していますが、汎用性を考慮する場合は、変数項目を利用して変更できるようにするなど工夫することもできます。

プログラム実行と結果確認

1. ＜受注管理＞プログラムを実行します。
2. 受注番号 1 番のデータにパークします。
3. ＜印刷＞ボタンをクリックします。

＜印刷＞ボタンをクリックすると、別のウィンドウが開き、PDF が表示されます。

クライアント で PDF を表示させるためには、クライアント 側にも PDF ビューワソフト がインストールされていなければなりません。

５．練習問題

仕入先と、その仕入製品明細とを出力する帳票印刷プログラムを作りましょう。

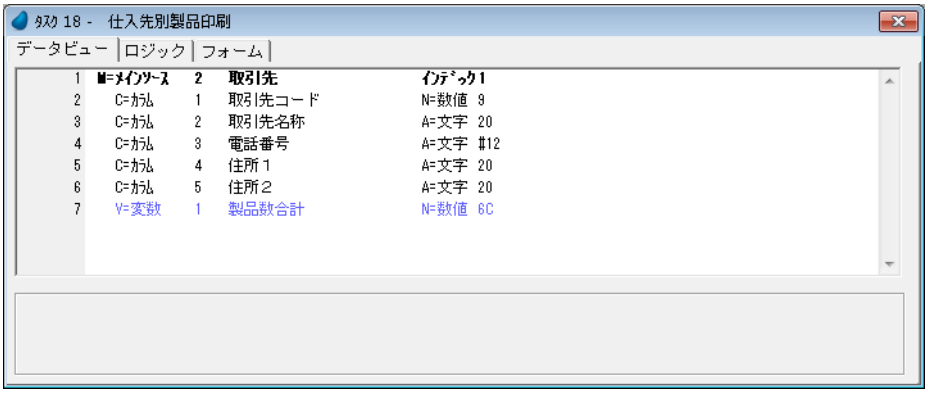
これまでの学習内容を参考にして、以下の内容を見ないで作成することにチャレンジしてください。

仕入先別製品印刷プログラムの作成

1. 新しく＜仕入先別製品印刷＞プログラムを［プログラム］リポジトリに登録します。
2. タスク特性の［タスクタイプ］特性で［B= バッチ］を選択します。
3. 同様に、［初期モード］で［Q= 照会］を選択します。

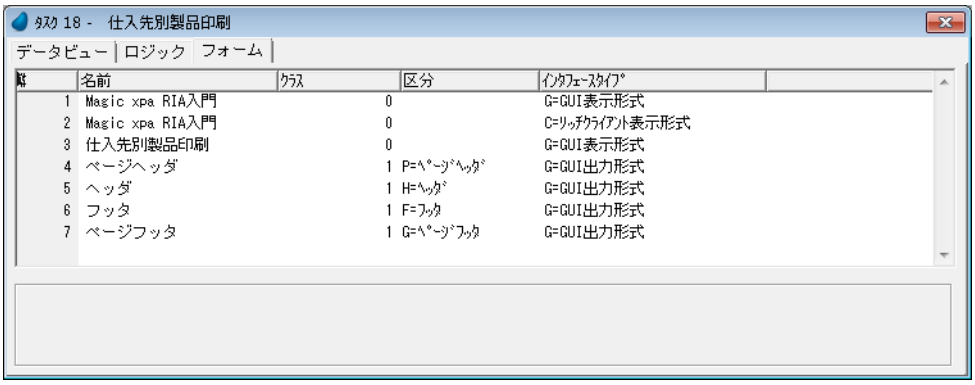
タスクデータビューの定義

次のイメージのように定義します。

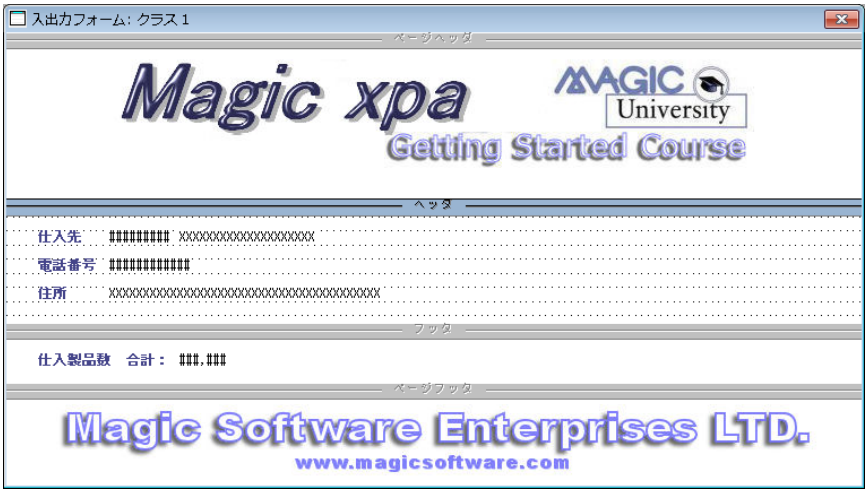


タスクフォームの作成

ページヘッダ、ページフッタ、ヘッダ、フッタをイメージに示すように定義します。

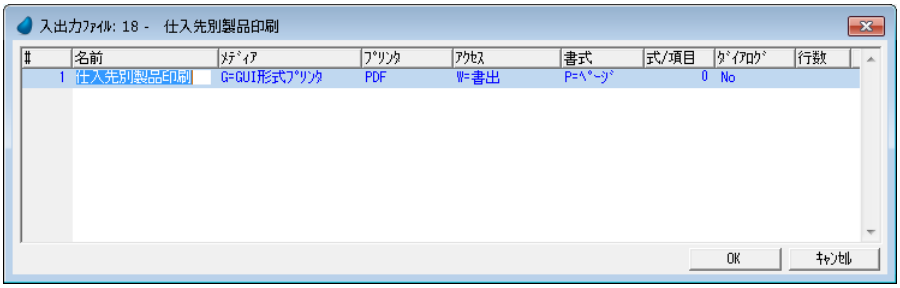


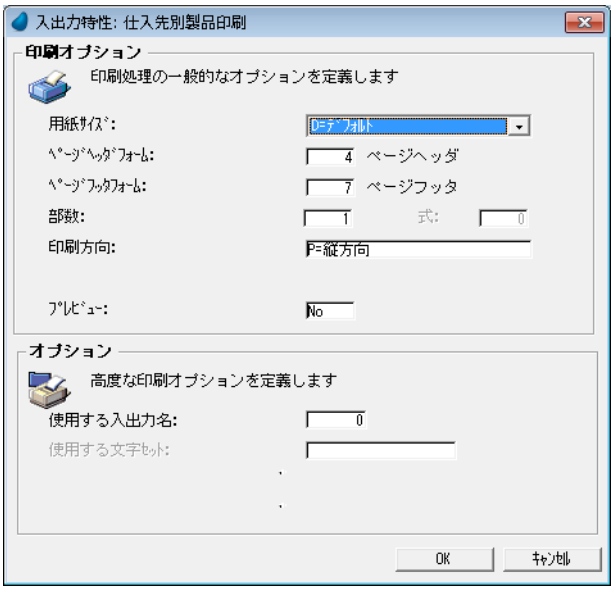
フォームの内容は一つの例として、下図に示すようになります。



入出力ファイルの定義

イメージのように定義してください。



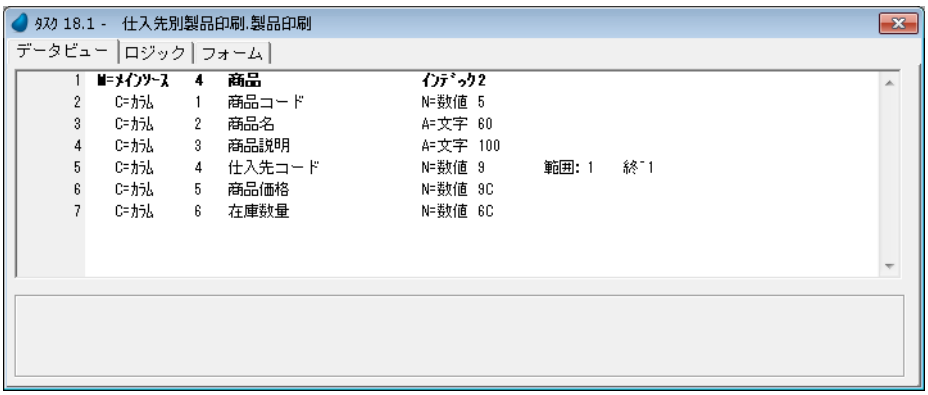


＜製品印刷＞サブタスクの作成

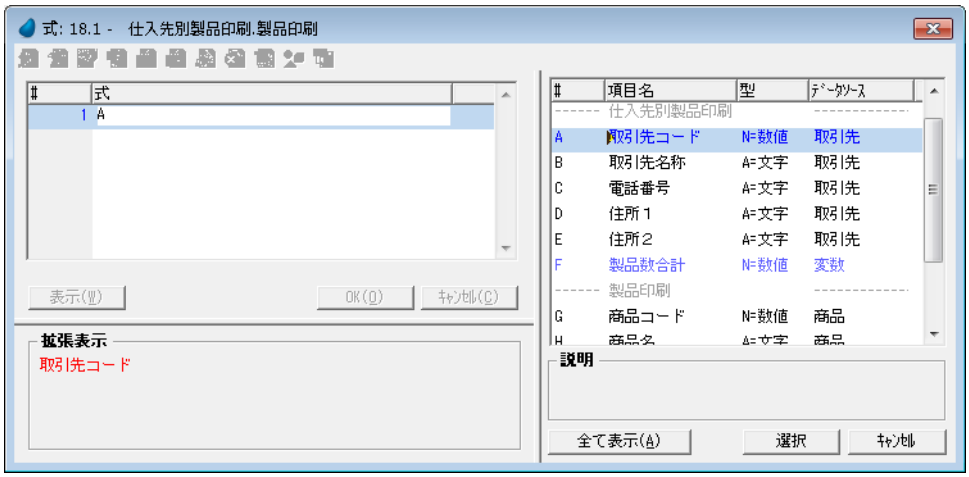
1. ［ナビゲータ］ペインで＜製品印刷＞のサブタスクを作成します。
2. タスクの［タスクタイプ］特性で［B= バッチ］を選択します。
3. 同様に、［初期モード］特性で ［Q= 照会］を選択します。

タスクデータビューの定義

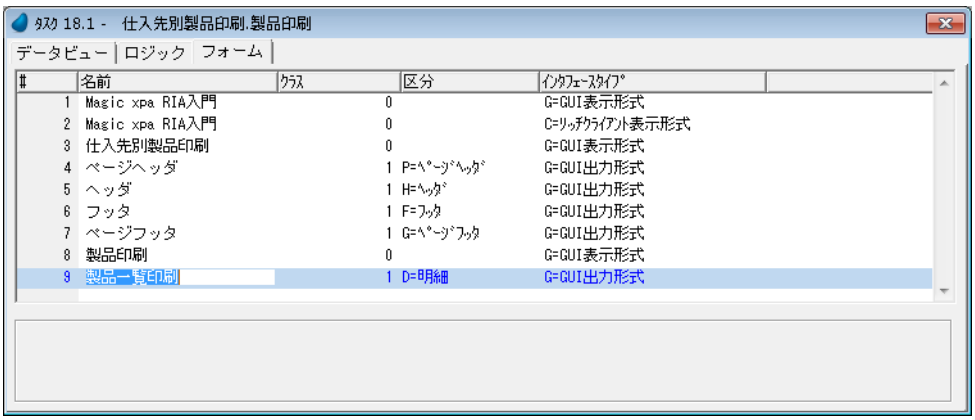
4. イメージに示すように定義します。

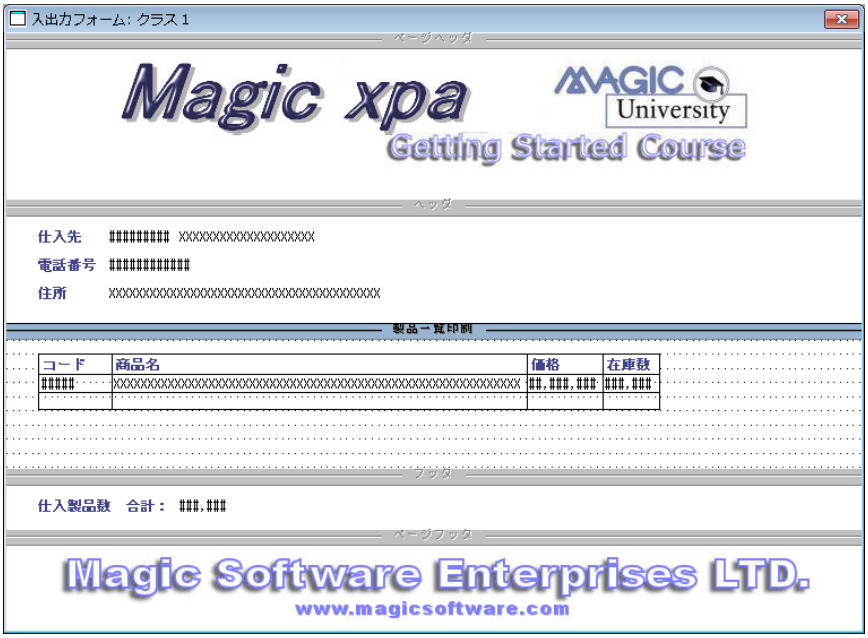


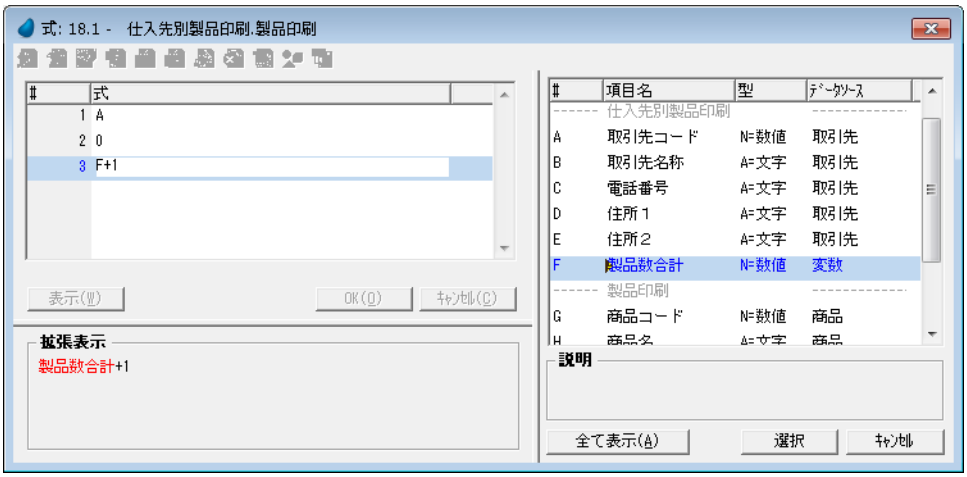
5. ＜仕入先コード＞（#5）について［範囲］設定します。



サブタスクのフォーム作成

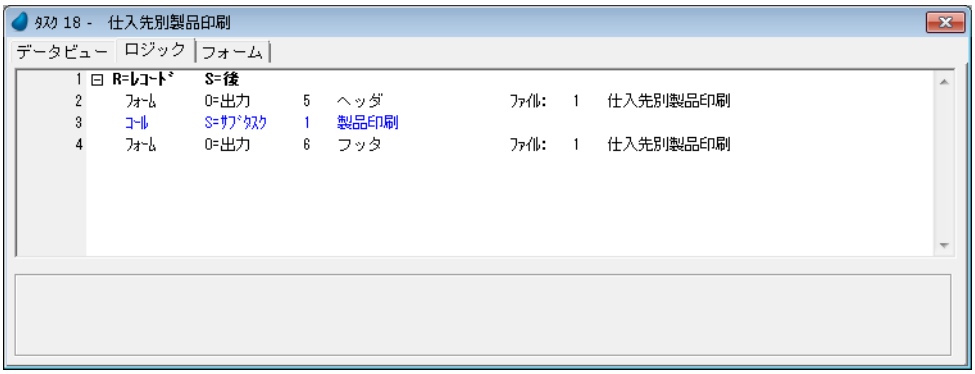



入出力フォーム: クラス 1
Magic xpa
MAGIC University
Getting Started Course
仕入先 ######### XXXXXXXXXXXXXXXXXXX
電話番号 ##########
住所 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
製品一覧印刷
コード 商品名 価格 在庫数
XXXXXXXXXXXXXXXXXXXX



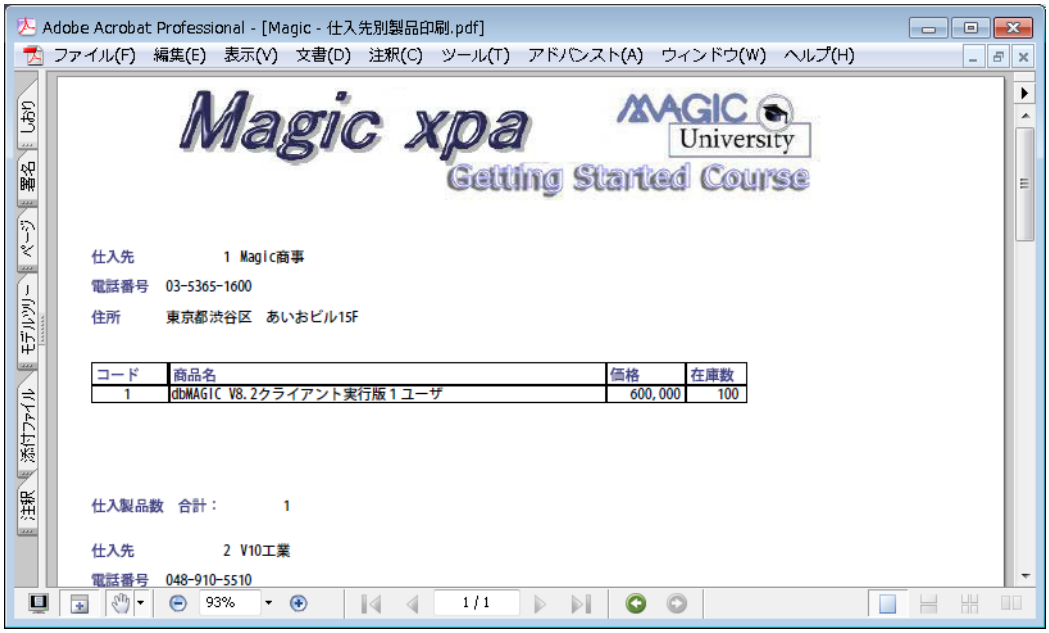
親タスクフォームの出力とサブタスクのコール

1. ＜仕入先別製品印刷＞タスクを開きます。
2. ［ロジック］エディタで下図のように設定します。



プログラム実行と結果確認

1. 変更内容を保存し、プログラムを閉じます。
2. ＜仕入先別製品印刷＞プログラムを実行しましょう。
3. ＜ C:¥Program Files¥Magicxpa¥Studio 2.3¥Projects¥Magic xpa RIA¥PDF¥Magic - 仕入先別製品印刷.pdf ＞というファイルが出力されます。PDF ビューワで開いて確認してください |



６．要約

本章では複雑な帳票の作成方法について説明しました。

このような帳票は、１対多のデータリレーションを持つインタラクティブなプログラムを作成する場合と似ています。

プログラムで使用する基本的なコマンドは同じですが、使用目的が異なります。

１対１のデータリレーションはデータの登録や修正のために使用しますが、一方、複雑な帳票出力では所定のフォーマットで該当するデータを印刷します。

複数レイヤを持つ帳票の出力プログラムは、たとえば 2 つのタスクで構成され、それぞれがメインソースを持っています。

最初のタスクがヘッダフォームとフッタフォームを印刷し、サブタスクが明細フォームを印刷するような構成をとります。

複雑な帳票出力プログラムを作成する場合の留意点をまとめます。

- 入出力ファイルは親タスクだけで定義し、親タスクとサブタスクで共有します。
- 親タスクのフォームはサブタスクの［フォーム］エディタからアクセスすることができます。
- 親タスクとサブタスクのフォームは、同じクラス（クラス >0）値と同じインタフェースタイプであり、また同じ［幅］を持つ必要があります。

本章で本コースにおける非インタラクティブタスクとその作成方法に関する説明を終わります。

次章では、モデルを用いたオブジェクト定義の一元管理について学習します。

第 21 章 オブジェクト定義の一元管理

本章では Magic xpa の［モデル］リポジトリについて説明します。［モデル］リポジトリにオブジェクトのモデルを定義し、プロジェクトの開発を進める際に利用します。

キーワード

- モデル
- 項目モデル
- リッチクライアント表示形式フォームモデル
- リッチクライアント表示スタティックモデル
- リッチクライアント表示エディットモデル
- 継承のしくみ

学習目標

本章を学ぶことで、次の内容に対する理解を深めましょう。

- ［モデル］リポジトリの考え方
- モデルのいろいろなタイプ
- 項目モデルの定義方法
- リッチクライアント表示形式フォームモデルの定義方法
- リッチクライアント表示形式スタティックモデルの定義方法
- リッチクライアント表示形式エディットモデルの定義方法
- モデルをオブジェクトに割当てる方法
- 特性が引き継がれる仕組み

参照

Magic ヘルプの［Magic xpa リファレンス／モデル］

１．はじめに

Magic xpa のモデルは、オブジェクトのクラスを定義する一組の特性であると言うことができます。

モデルの使用は必須ではありませんが、使用することによってプロジェクトの開発や保守においてたいへん役に立つものです。

使用することによる効果の例をいくつか挙げてみます。

- 開発時間の短縮が可能になる
- プロジェクト全体にわたって、オブジェクト定義の一貫性を維持できる
- 保守の容易性が高まる

今までの実習内容では、上記のことを考慮しないまま、プロジェクトを作成してきました。

本章では Magic xpa の［モデル］リポジトリについて学び、モデルの作成方法とオブジェクトへの適用について理解を深めます。

本章で学ぶことを活用することで、より効率的で体系化されたプロジェクト開発ができるようになります。

モデルとは何か？

オブジェクトに継承される’特性のまとまり’をモデルと言います。

モデルが割当てられているオブジェクトにおいて、特に個別に定義された特性項目以外は、モデルからその値を継承します。

個々のオブジェクトにおいて、特性値が定義された項目については［継承］関係が切れているとみなされ、その定義値によってモデルの特性値が上書きされます。

モデルの特性が更新されたとき、そのモデルが割当てられているすべてのオブジェクトにその変更内容が自動的に反映されます。

２．モデル定義の効果：データの一貫性と妥当性

特定のプロジェクト用にオブジェクト要素を検討するとき、共通の属性に基づいて、それらの要素をグループ化できないか考えてみましょう。

［モデル］リポジトリでは、個々の属性のまとまりに対してモデルを定義することにより、オブジェクト定義をグループ化することができます。つまり、そのグループで共通の属性を持つプロトタイプクラスを定義できるのです。

モデル定義を行なうことは必須ではありません。しかしそれを行なうことで、プロジェクト開発および保守のライフサイクル全体にわたって、大きな恩恵を受けることができます。

［モデル］リポジトリ活用による主な効果を列挙します。

- プロジェクト開発期間の短縮。オブジェクトモデルを一度定義してしまえば、他の同様のオブジェクトに対してその都度特性値を設定しなくてもよくなります。
- 保守が容易。モデルの特性設定値を定義しておくことで、モデルのその値が変更されるとき、関連するすべてのオブジェクトにその新しい特性値が継承されます。
- 異なるデータソースにおいて同一項目モデルの適用によりカラム特性の一致を保証。リンクをしたり、パラメータを渡したりする場合には、関連項目の特性は厳密に一致する必要があります。これは、同じモデルを用いて項目を定義することで容易に実現できます。

例

項目モデル

＜顧客コード＞オブジェクトがプロジェクトの随所で使用されているものとします。このような場合、型として［N= 数値］、書式として＜ 9 ＞を設定したモデルを定義し、オブジェクト定義に使用することができます。もし書式を＜ 9 ＞から＜ 10 ＞に修正したい場合、［モデル］リポジトリに登録されているこのモデルだけを修正すればよいのです。Magic xpa が対応するオブジェクトの特性を自動的に更新します。

フォームモデル

プロジェクト内ですべてのフォームが同じ［背景］を持つようにする場合を考えます。その場合、フォームを定義するごとに繰返しフォーム特性を開いて［背景］を設定するとしたら、たいへん面倒な作業になります。モデルを使用することで、その都度設定することなく、あるいは背景については意識することなく、またたとえ設定法さえ忘れたとしても、その特性を全フォームに継承させることができます。さらにそのモデルの特性値の変更内容は、すべての関連するフォームに直ちに反映されるのです。

継承の仕組み

Magic xpa では、モデル特性値のオブジェクトへの継承は次のように行なわれます。

- 新しく定義するオブジェクトに対してモデルが割当てられたとき、そのオブジェクトの特性値はすべてモデルから継承します。
- ある特性値を新たに設定し直した場合、その特性値に対する継承関係は解除されたものと見なされます。
- 既存のオブジェクトに対してモデルが割当てられるときは、そのオブジェクトを定義したときに個別設定した特性項目以外のものをすべてモデルから継承します。
- オブジェクトに割当てられたモデルを他のモデルに変更するとき、それまでモデルから継承していた特性が新しいモデルからの値に置き換わります。
- オブジェクトへのモデルの割当てをなくすときは、そのオブジェクトの特性値はオブジェクトデフォルト値になります。
- 各特性値へのモデルからの継承について、各特性項目欄にある［トグル］ボタンにより制御することができます。
- ［トグル］ボタンは、そのとき設定されている状態（継承する状態か、または継承しない状態）の反対の状態を表示します。つまり継承関係が解除されている状態のときには、継承状態のアイコンが表示されます。

ボタン	機能	説明
↓ X	継承解除	このボタンは継承関係を解除するときにクリックします。
↓ +	継承再設定	このボタンは継承関係を再度設定するときにクリックします。

個別に設定され、継承関係の解除された特性値は、青色ボールドの書体で表示されます。

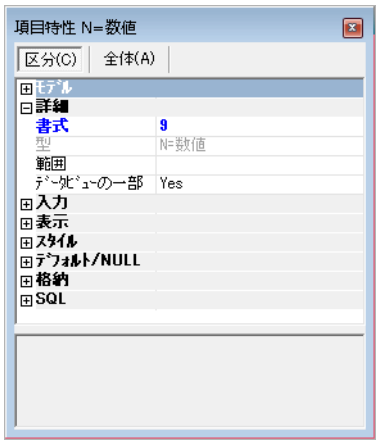
３．<顧客コード>モデルの定義

本セクションでは、項目モデルの定義について説明します。

1. ［プロジェクト］メニューから［モデル］（Shift+F1）を選択します。
2. 一行作成（F4）します。
3. 次のように設定します。

#	名前	クラス	型	フォルダ	公開名
1	顧客コード	F= 項目	N= 数値		

4. ＜顧客コード＞モデルの特性（Alt+Enter）を開きます。
5. ［書式］特性に＜ 9 ＞ と入力します。



これで＜顧客コード＞モデルの定義は完了です。

［モデル］リポジトリについて

［モデル］リポジトリには、異なるオブジェクトクラスに対するモデルが定義されています。同じオブジェクトクラスに対して、複数のモデルを定義することができます。

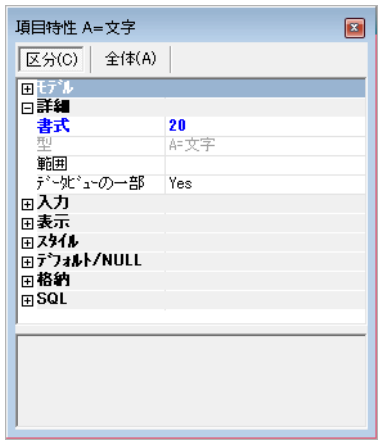
特性	説明
#	Magic xpa が付加するモデルの識別用連番です。
名前	モデルの名前を定義します。モデル名の長さは最大 30 桁です。
クラス	モデルはクラスタイプ別に分類して定義します。クラスタイプは、モデルを割当てることができる、Magic xpa のオブジェクトです。次のようなクラスタイプがあります。 • ヘルプ …… プロジェクトのヘルプ画面です。 • 項目 …… Magic xpa の項目。項目モデルはカラム，変数，パラメータに割当てることができます。 • ブラウザ形式 …… Web のオブジェクトで、フォーム，［エディット］コントロール，［プッシュボタン］コントロールなどです。 • GUI 表示形式 …… GUI 表示フォームとそのフォーム用コントロールです。 • GUI 出力形式 …… GUI 出力フォームとそのフォーム用コントロールです。 • テキスト形式 …… テキスト出力／入力フォームとそのフォーム用コントロールです。 • フレームセット形式 …… ブラウザウィンドウを複数のフレームに分割するフレームセットです。 • マージ形式 …… Magic xpa のデータを、テンプレートファイルとマージするためのマージフォームです。 • リッチクライアント表示形式……リッチクライアント表示フォームとそのフォーム用コントロールです。 • リッチクライアントフレーム形式……リッチクライアントフレームフォームとそのフォーム用コントロールです。 • GUI フレーム形式……GUI フレームフォームとそのフォーム用コントロールです。

＜顧客名＞モデルの定義

1. ［モデル］リポジトリに一行追加（F4）します。
2. 次のように設定します。

#	名前	クラス	型
2	顧客名	F= 項目	A= 文字

3. 特性を開き、［書式］特性に ＜ 20 ＞ と入力します。

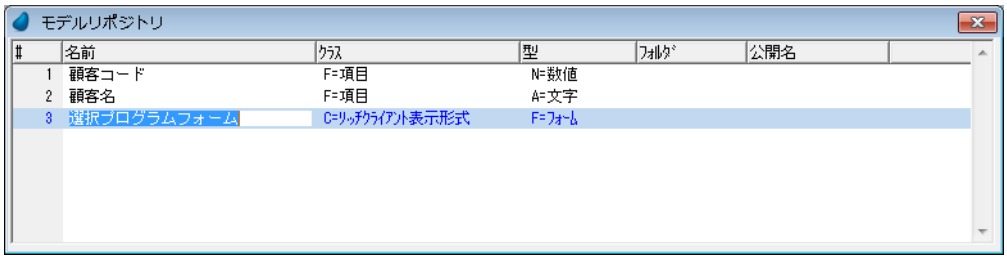


選択プログラム用リッチクライアント表示形式フォーム型モデルの定義

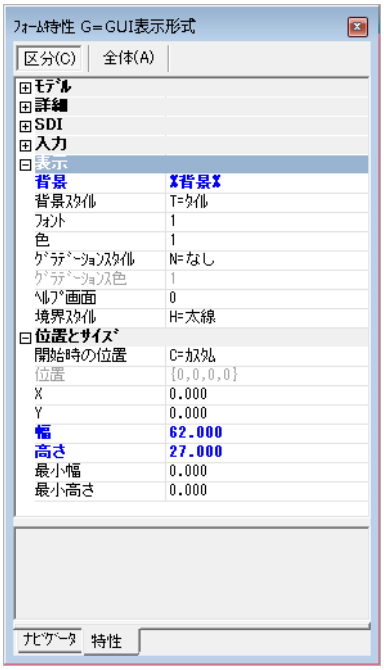
本コースでは選択プログラムをいくつか作成しました。各プログラムを作成するときには、その都度フォーム特性でサイズや背景の設定を行いましたが、ほとんどの選択プログラムは、同じような設定値になっています。そこで、リッチクライアント表示形式クラスのモデルを使用することで、統一感のある同一の外観を持ったフォームを選択プログラムに定義することができます。

1. ［モデル］リポジトリに新たに一行作成（F4）します。
2. 次のように設定します。

#	名前	クラス	型
2	顧客名	F= 項目	A= 文字
3	選択プログラムフォーム	C= リッチクライアント表示形式	F= フォーム



3. モデル特性（Alt+Enter）を開きます。
4. ［背景］特性に ＜％背景％ ＞と入力します。
5. ［幅］特性に＜ 62 ＞ と入力します。
6. ［高さ］特性に＜ 27 ＞ と入力します。



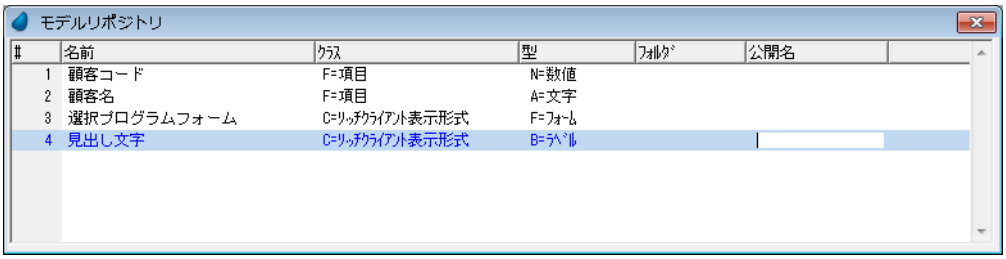
見出し文字用リッチクライアント表示形式スタティック型モデルの定義

このセクションでは、リッチクライアント表示形式クラスのスタティックコントロール用モデルを定義します。本コースのプロジェクトで共通して使われるスタティックコントロールは見出し文字用に使われる［ラベル］コントロールです。

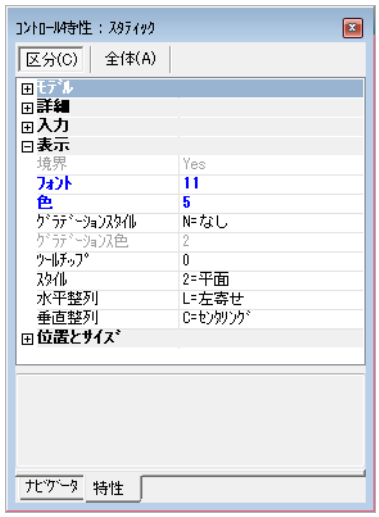
そこで、［ラベル］コントロール用のモデルを定義し、フォント、色、スタイルを設定します。後でこのモデルをフォーム上の文字見出しに適用してみることにしましょう。

1. ［モデル］リポジトリに新たに一行作成（F4）します。
2. 次のように設定します。

#	名前	クラス	型	フォルダ	公開名
4	見出し文字	C= リッチクライアント表示形式	B= ラベル		



3. モデル特性（Alt+Enter）を開きます。
4. ［フォント］特性に＜ 11 ＞を入力します。
5. ［色］特性に＜ 5 ＞を入力します。



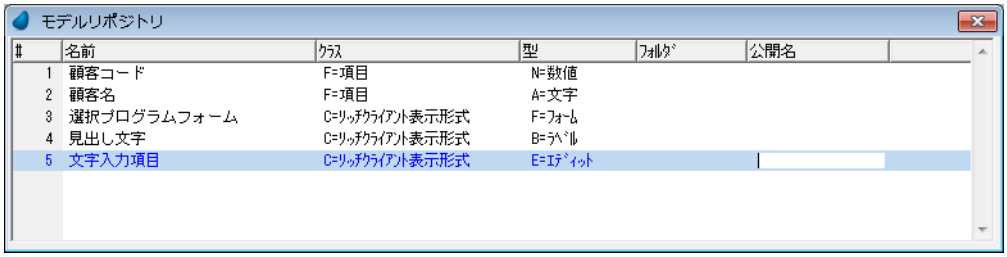
［エディット］コントロール用リッチクライアント表示形式エディット型モデルの定義

このセクションでは、リッチクライアント表示形式クラスのダイナミックコントロール用モデルを定義します。本コースのプロジェクトで共通して使われるダイナミックコントロールは、文字入力項目として使われる［エディット］コントロールです。

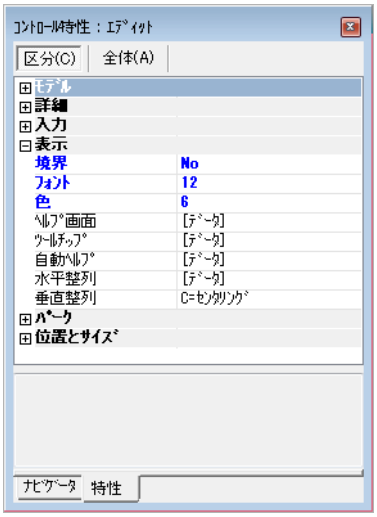
そこで、［エディット］コントロール用のモデルを定義し、フォント、色、スタイルを設定します。後でこのモデルをフォーム上のテキスト入力項目に適用してみることにしましょう。

1. ［モデル］リポジトリに新たに一行作成（F4）します。
2. 次のように設定します。

#	名前	クラス	型	フォルダ	公開名
5	文字入力項目	C= リッチクライアント表示形式	E= エディット		



3. モデル特性（Alt+Enter）を開きます。
4. ［フォント］特性に＜ 12 ＞を入力します。
5. ［色］特性に＜ 6 ＞を入力します。



同じ名前で GUI 表示形式モデルを作成してください。このモデルは後で参照用に APG を実行する際で使用します。

#	名前	クラス	型	フォルダ	公開名
6	文字入力項目	D=GUI 表示形式	E= エディット		

４．オブジェクトへのモデル割当て

通常の場合、オブジェクトの作成（データソースでのカラム定義や、タスクでの変数定義など）に先立ってモデルを定義します。オブジェクトを作成するときには該当するモデルを割当てるだけで、オブジェクトの特性はすべてモデルから継承されるようになります。

本コースでは、オブジェクトはすでに作成されています。したがって既存のオブジェクトにモデルを割当てる作業が必要になります。そして継承が解除されている特性項目の設定を変更し、モデルから値を継承するようにします。

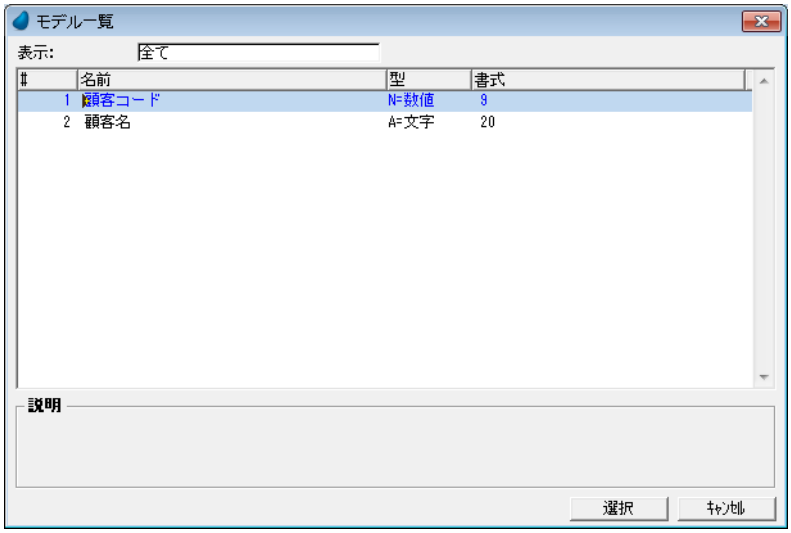
いくつかのタイプの異なるモデルのオブジェクトへの適用例を以下に示します。

項目モデルのカラムへの割当て

この例では、＜顧客コード＞モデルを＜顧客＞データソースの＜顧客コード＞カラムに割当てます。

1. ［データ］リポジトリ（Shift+F2）を開きます。
2. ＜顧客＞データソースにパークします。
3. ［カラム］テーブルにズームします。

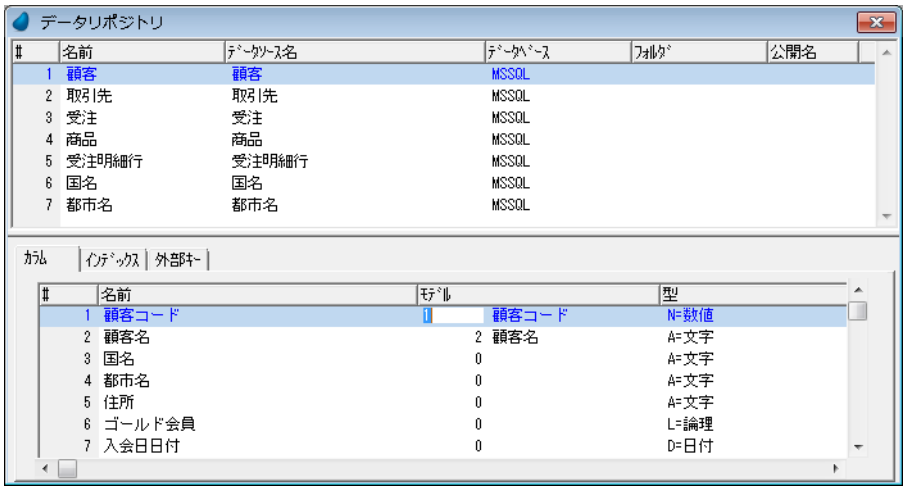
4. ＜顧客コード＞カラムにパークします。
5. ［モデル］特性からズームして、モデル一覧を開きます。



モデル一覧には、そのオブジェクトに適用可能なモデルのみ表示されます。今回の場合、［F= 項目］クラスのモデルのみ表示されます。

1. ＜顧客コード＞モデルを選択します。
2. ＜顧客名＞カラムにパークします。
3. ［モデル］特性からズームして、モデル一覧を開きます。

4. ＜顧客名＞を選択します。



５．モデル特性の継承

オブジェクトにモデルを割当てるだけでは値はまだ継承されません。

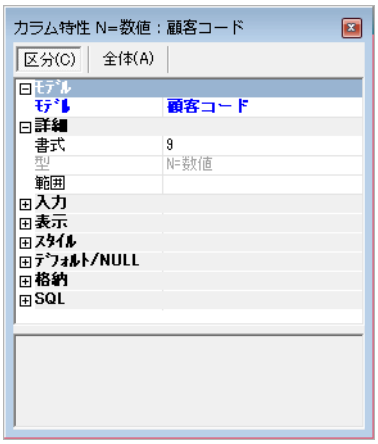
モデルをオブジェクトに割当てたとき、Magic xpa は継承が解除されている特性はその値を維持します。

特性値をモデルから継承させたい場合には、継承解除の状態となっているものを継承設定状態に変更しなければなりません。

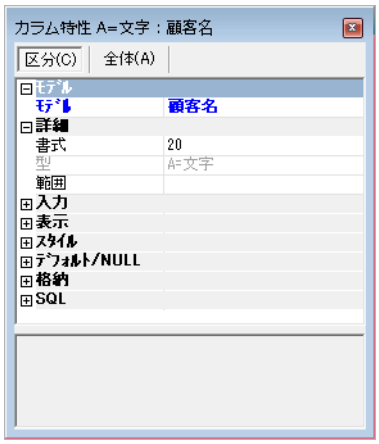
モデルから［書式］特性を継承させたい場合には、次のようにします。

1. ＜顧客コード＞カラムにパークします。
2. ［カラム特性］（Alt+Enter）を開きます。
3. ［書式］特性で、［↓ +］ボタンをクリックします。ボタンの表示が、［↓ X］に変わることを確認してください。

［モデル］特性が ＜顧客コード＞となっていることに留意してください。



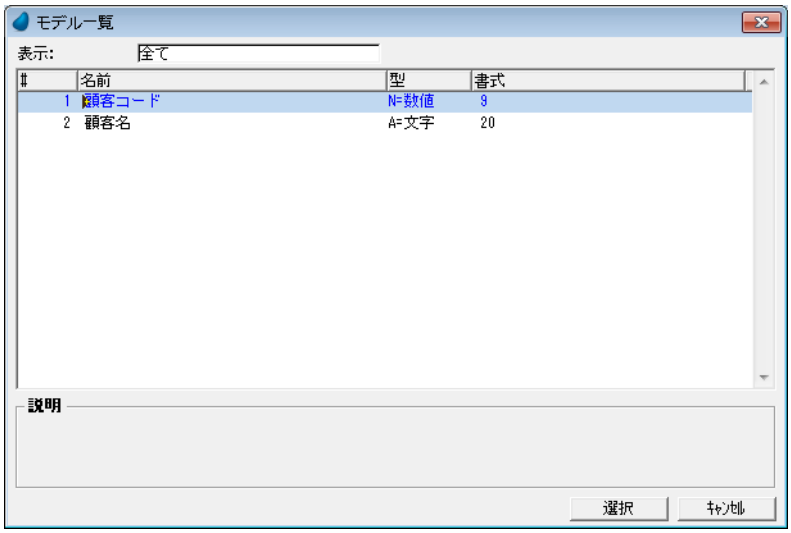
4. ＜顧客名＞カラムにパークします。
5. ［カラム特性］（Alt+Enter）を開きます。
6. ［書式］特性で ［↓ +］ボタンをクリックします。ボタンの表示が、［↓ X］に変わることを確認してください。
7. Esc キーを押下して、＜顧客＞データソースのカラムテーブル編集を終了します。
8. ［確認］ダイアログでは［はい］をクリックして変更内容を保存します。



項目モデルのタスクパラメータへの割当て

1. ［プログラム］リポジトリ（Shift+F3）を開きます。
2. ＜顧客選択＞（#10）にズームしてプログラムを開きます。
3. ［データビュー］エディタを選択します。

4. ＜ P_ 顧客コード＞（#4）行のモデル欄（［0］のところ）にパークします。
5. ズームしてモデル一覧を開きます。



6. ＜顧客コード＞モデルを選択します。

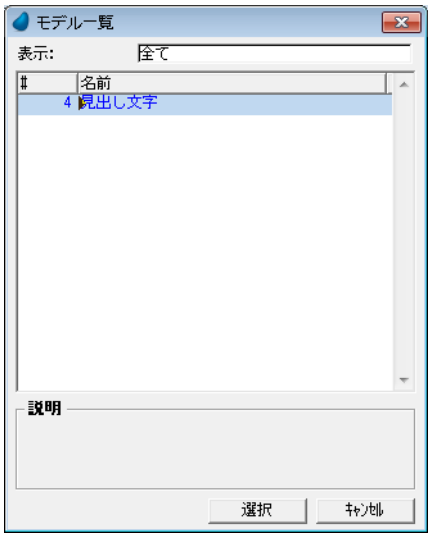
モデル特性の継承

1. ＜ P_ 顧客コード＞（#4）行にパークします。
2. ［カラム特性］（Alt+Enter）を開きます。
3. ［書式］特性で ［↓ +］ボタンをクリックします。ボタンの表示が、［↓ X］に変わることを確認してください。

［モデル］特性が＜顧客コード＞となっていることに留意してください。

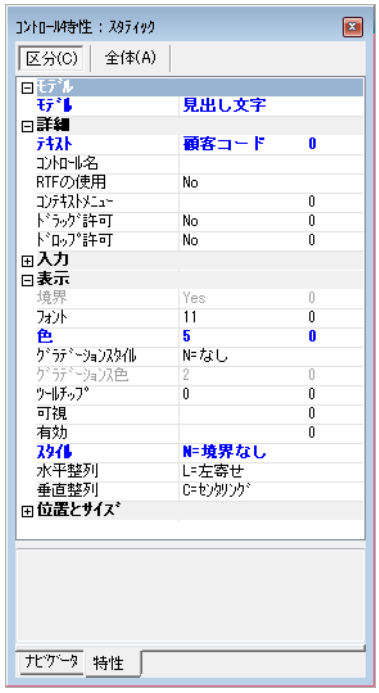
モデルのコントロールへの割当て

1. ［プログラム］リポジトリ（Shift+F3）を開きます。
2. ＜顧客ースクリーンモード＞（#4）にズームしてプログラムを開きます。
3. タスクのフォーム（Ctrl+M）を開きます。
4. 見出しの＜顧客コード＞［ラベル］コントロールの特性を開きます。
5. ［モデル］特性でズームし、モデル一覧を開きます。
6. ＜見出し文字＞モデルを選択します。



モデル特性の継承

7. ［フォント］特性で ［↓ +］ボタンをクリックします。ボタンの表示が、［↓ X］に変わることを確認してください。
8. ［色］特性で ［↓ +］ボタンをクリックします。ボタンの表示が、［↓ X］に変わることを確認してください。



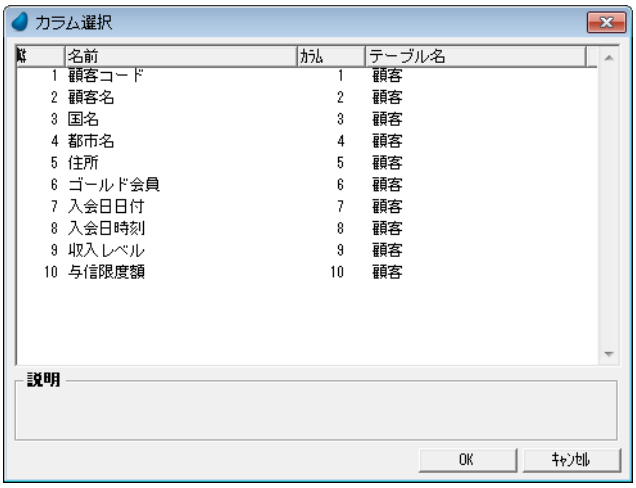
６．フォームモデルを使ったプログラムの自動作成

APG ではフォームモデルを使用することで、あらかじめ定義しておいたフォームを使用するプログラムを作成します。

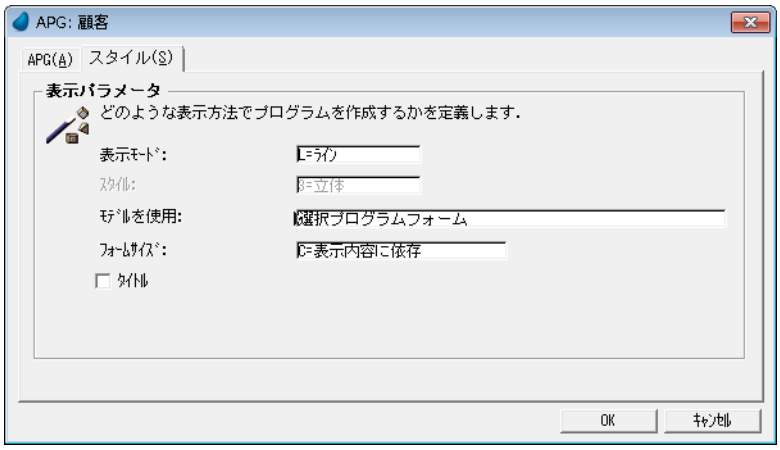
ここでは、プログラムを自動作成するときに＜選択プログラムフォーム＞を使用してみましょう。

1. ［データ］リポジトリ（Shift+F2）を開きます。
2. ＜顧客＞データソースにパークします。
3. ［オプション］メニューから［APG］を選択します。［APG］ダイアログが開きます。
4. APG パラメータの［カラム］欄からズームして［カラム選択］テーブルを開きます。

5. カラム欄 #3 ～ #10 に＜ 0 ＞を設定します。



6. ［選択］ボタンをクリックするか、Enter キーを押下します。
7. ［スタイル］タブをクリックします。
8. ［モデルを使用］欄でズームし、モデル一覧を表示します。
9. ＜選択プログラムフォーム＞モデルを選択します。



10. [OK] をクリックして APG によるプログラム実行を行ないます。実行結果は一時的なオンラインプログラムです。顧客コードと顧客名があらかじめ定義したフォーム上に表示されます。
11. プログラムを終了します。

一時的なプログラムは保存されません。保存したい場合は、APG パラメータの［モード］欄を［G= 作成］にして APG を実行してください。

７．練習問題

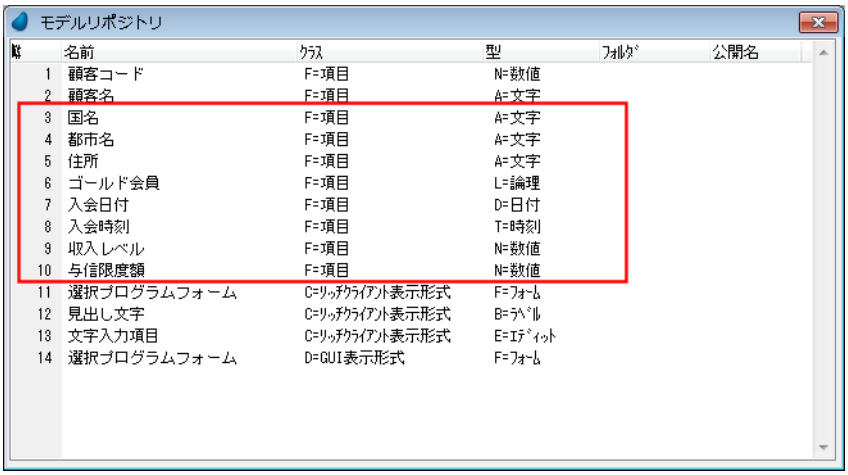
モデルの定義

練習として、＜顧客＞，＜取引先＞，＜商品＞各データソースにモデルを定義しましょう。

1. ［モデル］リポジトリ（Shift+F1）を開きます。
2. ＜顧客名＞（#2）エントリにパークします。
3. F4 キーを八回押下して八行作成し、＜顧客＞データソース用に、下記テーブルに示すモデルを定義してください。

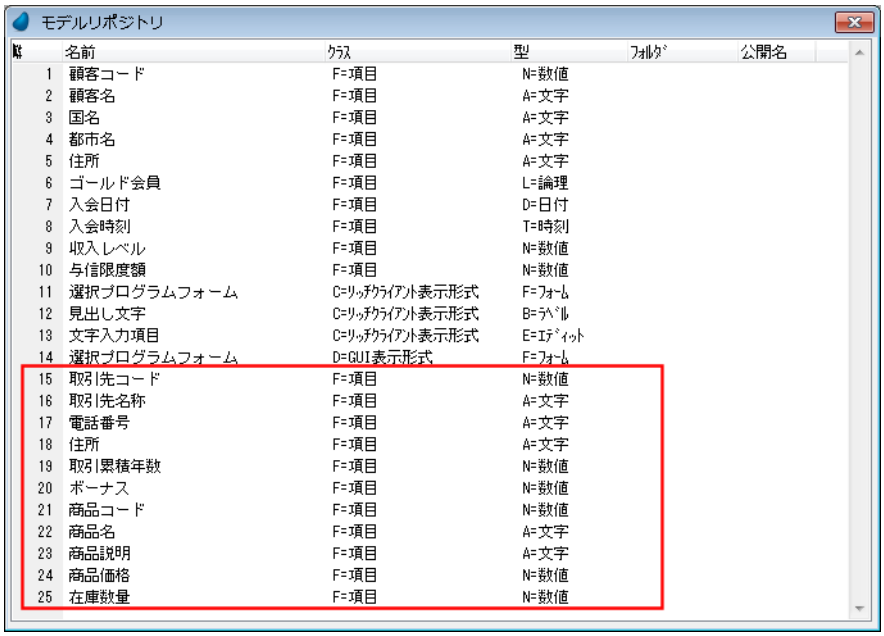
#	名前	クラス	型	フォルダ	公開名	モデル特性
3	国名	F= 項目	A= 文字			［書式］を 20 に設定
4	都市名	F= 項目	A= 文字			［書式］を 20 に設定

#	名前	クラス	型	フォルダ	公開名	モデル特性
5	住所	F= 項目	A= 文字			［書式］を 20 に設定
6	ゴールド会員	F= 項目	L= 論理			デフォルトのまま
7	入会日付	F= 項目	D= 日付			デフォルトのまま
8	入会時刻	F= 項目	T= 時刻			デフォルトのまま
9	収入レベル	F= 項目	N= 数値			［書式］を 12.2C に設定
10	与信限度額	F= 項目	N= 数値			［書式］を 12.2C に設定



4. ＜選択プログラムフォーム＞（#14）エントリにパークします。
5. F4 キーを 11 回押下して 11 行作成し、＜取引先＞および＜商品＞データソース用に、下記に示すモデルを作成してください。

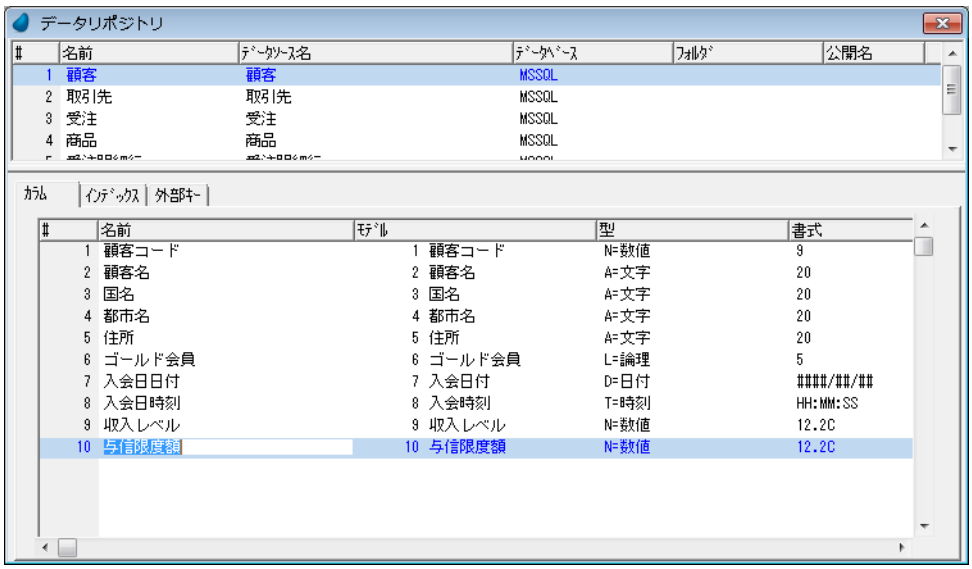
#	名前	クラス	型	フォルダ	公開名	モデル特性
15	取引先コード	F= 項目	N= 数値			［書式］を 9 に設定
16	取引先名称	F= 項目	A= 文字			［書式］を 20 に設定
17	電話番号	F= 項目	A= 文字			［書式］を #12 に設定
18	住所	F= 項目	A= 文字			［書式］を 20 に設定
19	取引累積年数	F= 項目	N= 数値			［書式］を 2 に設定
20	ボーナス	F= 項目	N= 数値			［書式］を 3.2 に設定
21	商品コード	F= 項目	N= 数値			［書式］を 5 に設定
22	商品名	F= 項目	A= 文字			［書式］を 60 に設定
23	商品説明	F= 項目	A= 文字			［書式］を 100 に設定
24	商品価格	F= 項目	N= 数値			［書式］を 8C に設定
25	在庫数量	F= 項目	N= 数値			［書式］を 6C に設定



#	名前	ｸﾗｽ	型	ﾌｫﾙﾀﾞ	公開名
1	顧客コード	F=項目	N=数値		
2	顧客名	F=項目	A=文字		
3	国名	F=項目	A=文字		
4	都市名	F=項目	A=文字		
5	住所	F=項目	A=文字		
6	ゴールド会員	F=項目	L=論理		
7	入会日付	F=項目	D=日付		
8	入会時刻	F=項目	T=時刻		
9	収入レベル	F=項目	N=数値		
10	与信限度額	F=項目	N=数値		
11	選択プログラムフォーム	C=ﾘｯﾁｸﾗｲｱﾝﾄ表示形式	F=ﾌｫｰﾑ		
12	見出し文字	C=ﾘｯﾁｸﾗｲｱﾝﾄ表示形式	B=ﾗﾍﾞﾙ		
13	文字入力項目	C=ﾘｯﾁｸﾗｲｱﾝﾄ表示形式	E=ｴﾃﾞｨｯﾄ		
14	選択プログラムフォーム	D=GUI表示形式	F=ﾌｫｰﾑ		
15	取引先コード	F=項目	N=数値		
16	取引先名称	F=項目	A=文字		
17	電話番号	F=項目	A=文字		
18	住所	F=項目	A=文字		
19	取引累積年数	F=項目	N=数値		
20	ボーナス	F=項目	N=数値		
21	商品コード	F=項目	N=数値		
22	商品名	F=項目	A=文字		
23	商品説明	F=項目	A=文字		
24	商品価格	F=項目	N=数値		
25	在庫数量	F=項目	N=数値		

項目モデルのカラムへの割当て

1. ＜顧客＞データソースの残りのカラムにモデルを割当ててください。
2. 継承設定をするのを忘れないようにしましょう。

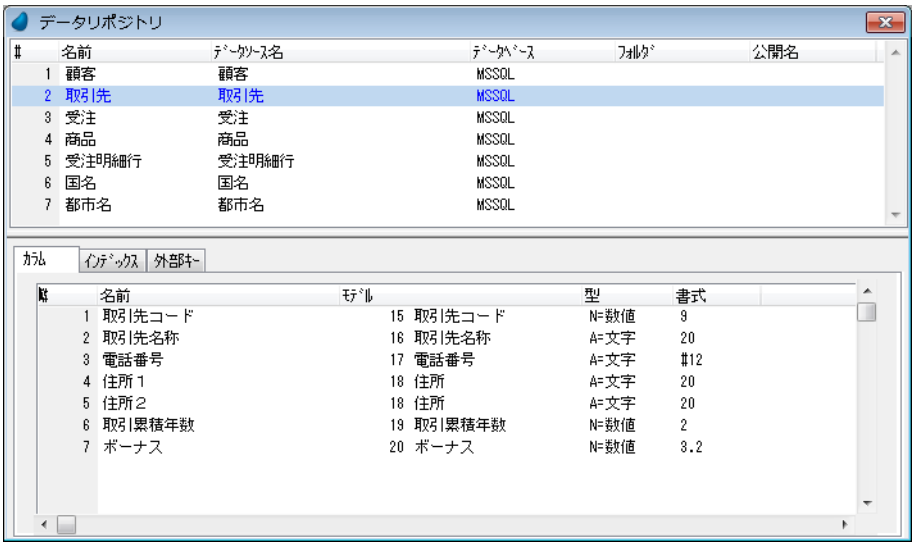


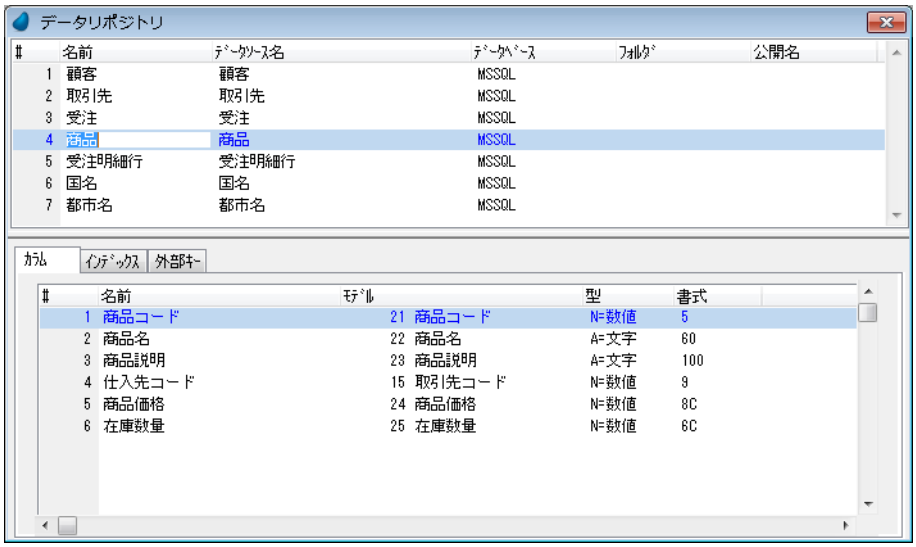
#	名前	ﾃﾞｰﾀｿｰｽ名	ﾃﾞｰﾀﾍﾞｰｽ	ﾌｫﾙﾀﾞ	公開名
1	顧客	顧客	MSSQL		
2	取引先	取引先	MSSQL		
3	受注	受注	MSSQL		
4	商品	商品	MSSQL		

#	名前	ﾓﾃﾞﾙ	型	書式
1	顧客コード	1 顧客コード	N=数値	9
2	顧客名	2 顧客名	A=文字	20
3	国名	3 国名	A=文字	20
4	都市名	4 都市名	A=文字	20
5	住所	5 住所	A=文字	20
6	ゴールド会員	6 ゴールド会員	L=論理	5
7	入会日日付	7 入会日付	D=日付	####/##/##
8	入会日時刻	8 入会時刻	T=時刻	HH:MM:SS
9	収入レベル	9 収入レベル	N=数値	12.2C
10	与信限度額	10 与信限度額	N=数値	12.2C

3. ＜取引先＞および＜商品＞データソースのカラムにモデルを割当ててください。

4. 継承設定をするのを忘れないようにしましょう。

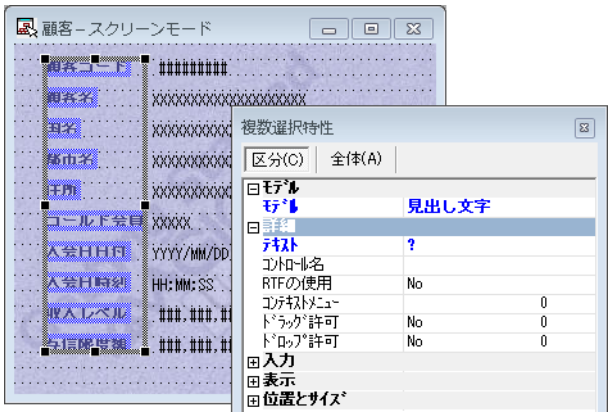




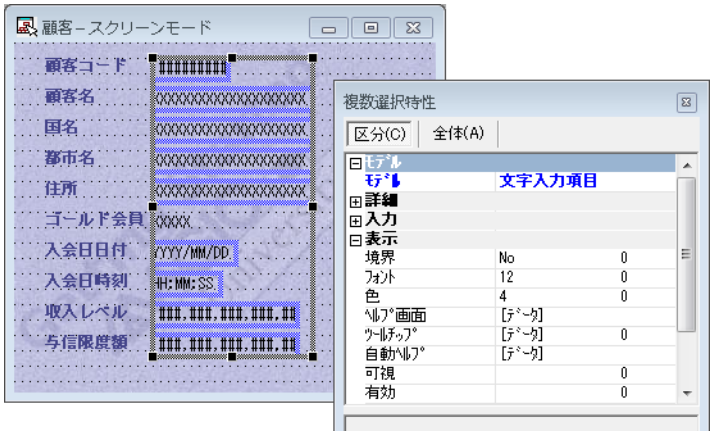
モデルのコントロールへの割当て

次の設定をしましょう。

1. ＜顧客ースクリーンモード＞（#4）プログラムのフォームを開きます。
2. ＜見出し文字＞モデルをすべての［ラベル］コントロールに割当ててください。
3. ＜文字入力項目＞モデルをすべての［エディット］コントロールに割当ててください。
4. 継承設定をするのを忘れないようにしましょう。



（<見出し文字>モデルの割当て）



（<文字入力項目>モデルの割当て）

８．要約

本章では、Magic xpa のモデルについて学習しました。

モデルはオブジェクト特性の集まりです。

モデルの使用は必須ではありませんが、使用することでプロジェクト開発および保守作業の生産性がより高まります。

- モデルを使用することによる効果の例
- 開発時間の短縮
- プロジェクト全体での定義の一貫性保証
- 保守の容易性向上

本章の実習では、モデルの定義と、モデルのカラム，変数，パラメータ，コントロール，フォームなどへの割当てを行ないました。

また APG 機能を利用したプログラム自動作成において、フォームモデルを使用する方法を行ないました。

次章ではプロジェクトで使用するメニューの作成と使用方法について説明します。

第 22 章 メニュー

本章では、Magic xpa のメニューについて学習します。

キーワード

- プルダウンメニュー
- コンテキストメニュー
- ツールバーメニュー
- メニュー生成
- 実行エンジン

学習目標

本章を学ぶことで、次の内容に対する理解を深めましょう。

- メニューの使用目的
- プルダウンメニューの定義方法
- コンテキストメニューの定義方法
- コントロールへのコンテキストメニューの割当て
- フォームへのコンテキストメニューの割当て
- メニューエントリからのイベント発動
- ツールバーへのメニュー追加

参照

Magic ヘルプの［Magic xpa リファレンス／メニュー］

１．はじめに

メニューはユーザが使用することのできるコマンドのリストです。コマンドの詳細知識や使い方を覚えていなくとも利用できるように、よく使われるコマンドのショートカットが登録されているものと考えることができます。

メニューのコマンドを選択する方法は二つあります。

- マウスカーソルでエントリをポイントし、クリックして選択する。
- 一つ以上のキー入力の組み合わせで選択する（たとえば Alt+F など）。

グラフィカルなユーザインタフェースを持つコンピュータは、通常、選択肢を示す文字とシンボルアイコンを組合わせたメニューを持っています。

メニューを共通的に使用することで、さまざまな操作、たとえばファイルへのアクセスや保存、プログラムの終了、データ操作などが便利に行なえるようになります。

プルダウンメニューはメニューバー（ウィンドウまたは画面の最上部）にあるもので、通常は何らかのアクションを起動するコマンドのために利用されます。

本章では、MDI フォーム上で表示するプルダウンメニューを扱います。プログラムを SDI フォームで表示させる場合、フォーム上にプルダウンメニューを表示させることもできます。

コンテキストメニューはマウスの右クリックで表示されます。そのときのコンテキストにしたがって、自動的にメニューから利用できるコマンドの内容が変わります。

伝統的なヒューマンインタフェースとして、メニューの名前にはたとえば、［ファイル］とか［編集］などの単語が用いられていました。

しかし、その後の技術の進展により、最近は大きく変わってきています。単語での表現は時にあいまいで分かりにくいため、複数の単語からなるメニュー名が可能となるよう、プルダウンメニューのように垂直方向にメニューを拡張するアイデアが考案されました。

Magic xpa は、プロジェクトでメニューを簡単に定義することができます。プロジェクトのメニューは、エンドユーザがプロジェクト内のプログラムを起動させるために使用します。

Magic xpa の提供するメニュー機能には次のようなものがあります。

メニュー種別	説明
デフォルトのプルダウンメニュー	Magic xpa はプロジェクトの登録時にデフォルトのプルダウンメニューを作成します。このメニューには実行時に必要となる基本的な機能が登録されています。
ツールバーメニュー	プルダウンメニューの下に表示されるアイコンボタンで、プログラムや諸機能へのショートカットが登録されます。
追加メニュー	必要に応じて作成します。プロジェクトではコンテキストメニューとして使われます。

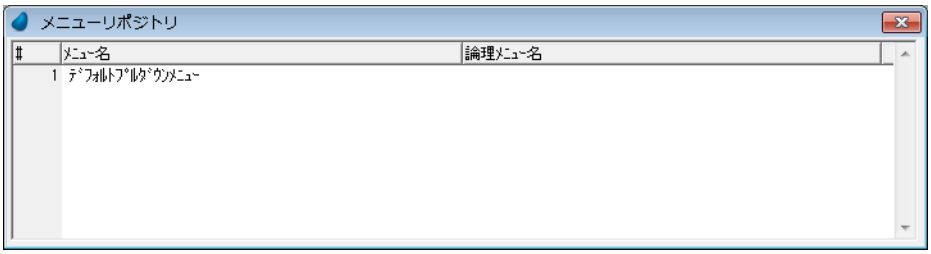
メニューの構成においては論理的な順序を工夫し、エンドユーザが使いやすいものにしましょう。

２．［メニュー］リポジトリ

［メニュー］リポジトリにはあらかじめデフォルトのプルダウンメニューが登録されていますが、さらに追加してゆくことができます。

1. ［プロジェクト］メニューから［メニュー］（Shift+F6）を選択します。［メニュー］リポジトリが開きます。

まだメニューを追加していない状態では、次の図のように、［デフォルトプルダウンメニュー］だけがあります。



［メニュー］リポジトリの各カラムには、次のような機能があります。

カラム	説明
メニュー名	メニュー構造の論理的な名前を定義します。ここで定義された名前は、実行時には表示されません。
論理メニュー名	メニューの表示に関する制御を、Magic xpa のメニューに関連する関数から行う場合に使用します。 関数からの制御を有効に行なうためには、ここにユニークな名前を定義しておく必要があります。 ［¥］記号は、メニューパスの区切り文字として予約されているため、論理名には使用できません。

３．デフォルトのプルダウンメニュー

新しくプロジェクトを登録するときに、デフォルトのプルダウンメニューが作成されます。このデフォルトプルダウンメニューエントリは削除や移動ができません。

デフォルトプルダウンメニューには、実行時に必要な基本的機能があらかじめ登録されています。

たとえば、［編集］メニューにはデータ入力処理に関連する機能、例えば［キャンセル］や［変更取消］、［切り取り］などが含まれています。

このデフォルトプルダウンメニューの内容に対しては、追加，修正，削除が可能です。

1. デフォルトプルダウンメニューのエントリからズームすると、［メニュー定義］テーブルの内容を表示します（下図）。

［メニュー定義］テーブルの各カラムについて

［メニュー定義］テーブルの各カラムの意味を下表にまとめます。

カラム	説明
タイプ	テーブルのメニュー項目タイプには、次のものがあります。 • P ＝プログラム …… パラメータカラムに［プログラム］リポジトリのプログラム番号を指定します。 • O ＝ OS コマンド …… パラメータカラムに OS のコマンドを指定します。 • V ＝イベント …… パラメータカラムに、Magic xpa の内部イベントまたはユーザイベントを指定します。 • E ＝ライン …… メニュー上、この位置に横線（区切り線）が表示されます。 • M ＝メニュー …… メニューを階層化する場合に、この値を指定します。 • W= ウィンドウの一覧 …… メニューは［ウィンドウの一覧］として表示されます。

カラム	説明
ユーザメニュー名	実行時にメニューに表示されるテキスト（コマンド名）を定義します。半角英数字の前に '&'（アンパーサンド）をつけることで、英数字キーがメニューへのショートカットキーとなります。たとえば、' ファイル (&F)' と定義した場合、Alt+F キー入力により、そのメニューが選択されます。
論理メニュー名	メニュー項目（コマンド）の論理名を設定します。この論理名は、エンドユーザには見えません。プログラム内（関数など）から指定する場合はこの論理メニュー名を使用します。
パラメータ	［タイプ］カラムに指定した値（メニューのタイプ）によって内容が異なります。このカラムからズームすると、［タイプ］カラムで指定した値に応じたメニュー定義画面が表示されます。
ショートキー	ショートカットキーを設定する場合、このカラムからズームして［キー設定］ダイアログを表示し、実際のキーの組み合わせを押下します。アクティブなキーの組み合わせが、それぞれユニークである必要があります。

［パラメータ］欄には、メニューのタイプによって、次のようなパラメータを設定します。

タイプ	パラメータ	意味	値
M ＝メニュー	サブメニュー	ズームすると、メニュー定義テーブルが表示されます。このテーブルは、１つ下のレベルのメニュー定義画面です。	このサブメニューレベルのエントリ数
P ＝プログラム	プログラム番号	ズームするとプログラム一覧が表示され、プログラム番号を選択できます。	プログラム番号
V ＝イベント	イベント	イベントを実行し、イベントロジックユニットを駆動させます。	［システム］，［内部］，［ユーザ］のいずれかのタイプに属するイベント
E ＝ OS コマンド	OS コマンド	一時的に OS に抜け、OS のコマンドを実行することができます。スクリプトやバッチファイルなどの実行可能な OS のコマンドを指定できます。 ズームすると Windows の［ファイルを開く］ダイアログでファイル選択することもできます。	文字列 OS コマンドを指定しなかった時には、実行時、このメニュー項目をエンドユーザが選択すると OS シェルへ移行します。
L ＝ライン	N/A	前のメニュー項目と次のメニュー項目との間に横の区切り線を表示します。	

４．デフォルトプルダウンメニューへのエントリ追加

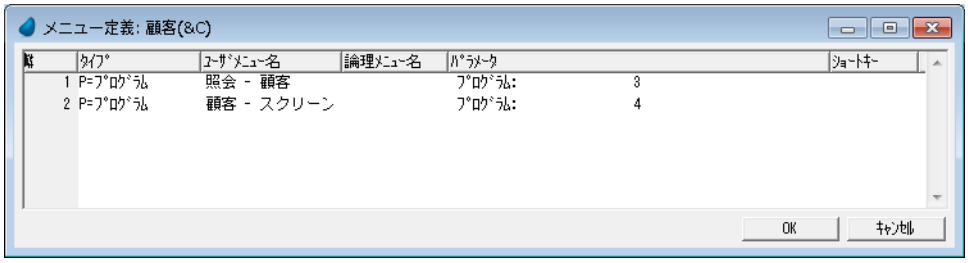
ここで、<顧客>プログラムのためのプルダウンメニューを作成してみましょう。

このメニューは、サブメニューを持つメニュータイプとし、サブメニューの中に顧客関連のプログラムを登録します。

1. ［メニュー定義］テーブルで、#3［オプション］の行にパークします。
2. 一行作成（F4）します。
3. 次のように設定します。

#	タイプ	ユーザメニュー名	論理メニュー名	パラメータ	ショートキー
4	M= メニュー	顧客 (&C)		サブメニュー：0	

4. サブメニューエントリからズームして、新たな［メニュー定義］テーブルを開きます。新しく開いた［メニュー定義］テーブルに＜顧客＞メニューのサブメニューを定義します。
5. F4 キーを押下して一行作成します。
6. ［パラメータ］カラムの［プログラム番号］欄からズームして、プログラム一覧を開きます。
7. ＜照会－顧客＞（#3）プログラムを選択します。
8. ５－７の操作を繰返し、今度は＜顧客－スクリーンモード＞（#4）を追加します。
9. ［OK］をクリックしてメニュー定義を終了し、また、Esc キーを押下して［メニュー］リポジトリも終了します。

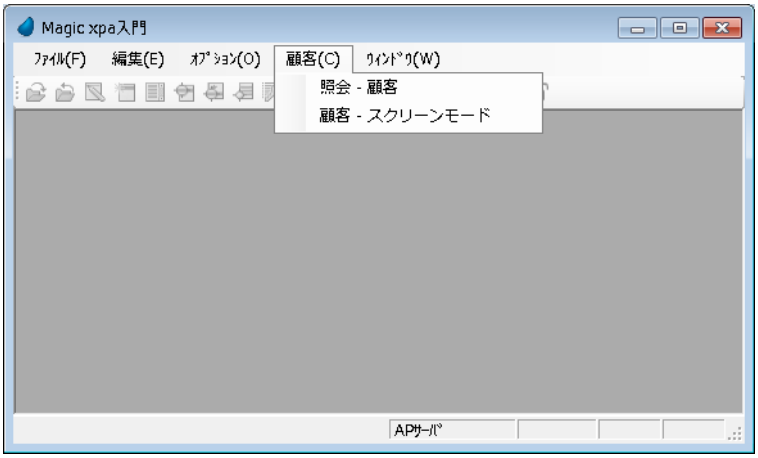


＜顧客＞メニューの動作確認

メニュー表示を確認するためには、プロジェクトを起動する必要があります。

1. ［デバッグ］メニューから［プロジェクトの実行］（Ctrl+F7）を選択します。Magic xpa の RIA 実行エンジンが起動します。プルダウンメニューとして、＜顧客 (C) ＞が新しく追加されていることが分かります。

2. ＜顧客 (C) ＞をクリックするか、Alt+C キーを押下して、サブメニューを開きます。＜照会－顧客＞および、＜顧客－スクリーンモード＞が＜顧客＞メニューの内容として登録されていることが分かります。
3. Esc キーを押下して実行版を終了しましょう。



５．コンテキストメニュー

プルダウンメニューは、一般的によく使用される機能を登録する目的に使用されます。しかし実際にプログラムを動作させているときに、場合によっては追加のメニューオプションが必要になることがあります。コンテキストメニューはそのような追加メニューを用意したいときに使用します。

Magic xpa では追加のメニューを作成し、コンテキストメニューとして使用するための機能を提供しています。

＜商品選択＞コンテキストメニューの作成

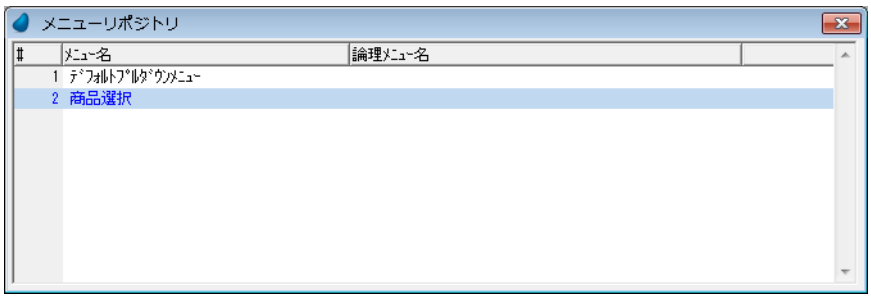
＜受注管理＞プログラムでは、受注明細行の商品は商品一覧から選択して選択できました。

その商品一覧を開くためにエンドユーザがズームするとき、F5 キーを押下しました。

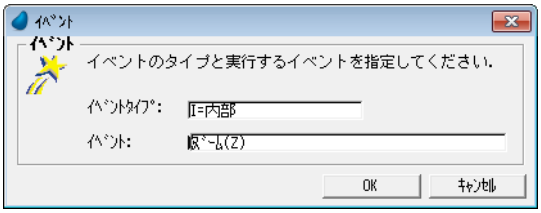
ここでコンテキストメニューを追加し、エンドユーザがそこから商品一覧を開くことができるようにしてみましょう。

1. ［プロジェクト］メニューから［メニュー］（Ctrl+F6）を選択し、［メニュー］リポジトリを開きます。
2. #1：［デフォルトプルダウンメニュー］にパークします。

3. 一行作成します。（F4）
4. ［メニュー名］カラムに **＜商品選択＞** と入力します。

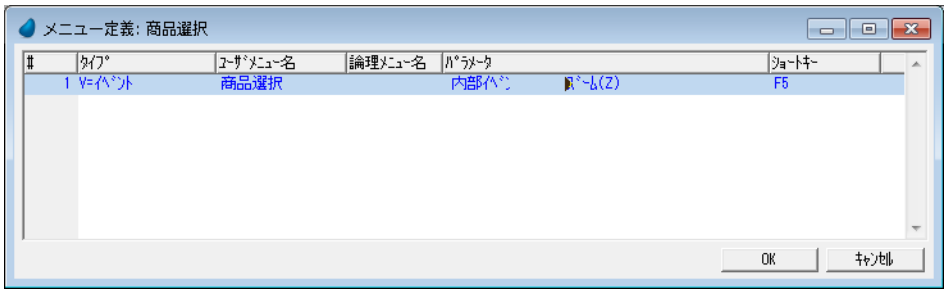


5. ズームして＜商品選択＞［メニュー定義］テーブルを開きます。
6. 一行作成（F4）します。
7. ［タイプ］カラムで、［V= イベント］を選択します。
8. ［ユーザメニュー名］カラムで、**＜商品選択＞** と文字入力します。
9. ［パラメータ］カラムでズームし、［イベント］ダイアログを開きます。
10. ［イベントタイプ］欄で、［I= 内部］を選択します。
11. ［イベント］欄でズームし、イベント一覧を表示します。
12. ［ズーム（Z）］を選択します。
13. ［OK］ボタンをクリックします。



［ショートキー］カラムに自動的に F5 と表示されることに留意してください。

14. 変更内容を保存し、メニュー定義テーブル、［メニュー］リポジトリを閉じます。



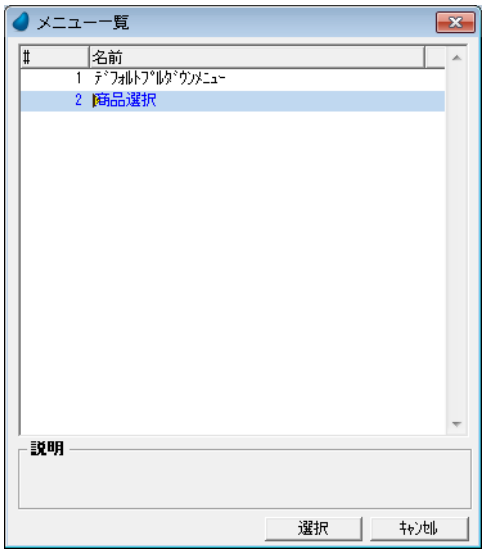
＜商品選択＞コンテキストメニューのコントロールへの割当て

定義したコンテキストメニューを＜受注管理＞プログラムの＜商品コード＞コントロールに割当ててみましょう。

これにより、商品選択コンテキストメニューは、＜商品コード＞コントロールにカーソルがパークしているときだけ有効になります。

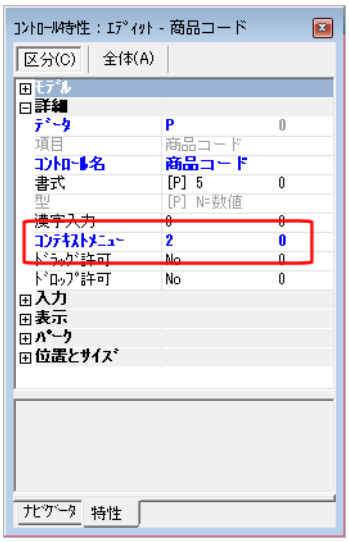
1. ［プログラム］リポジトリを開きます。
2. ＜受注管理＞（#8）プログラムを開きます。
3. ［ナビゲータ］ペインで＜受注明細行管理＞タスクを選択します。

4. タスクフォーム（Ctrl+M）を選択します。
5. ＜商品＞カラムで、＜商品コード＞［エディット］コントロールをクリックしてマークします。
6. ［エディット］コントロールの特性を開き、［コンテキストメニュー］特性からズームし、メニュー一覧を開きます。
7. ＜商品選択＞（#2）メニューを選択します。



8. 変更内容を保存し、プログラムを閉じます。

＜商品コード＞の特性は右図のようになります。



＜受注明細行管理＞タスクフォームは下図のようになります。

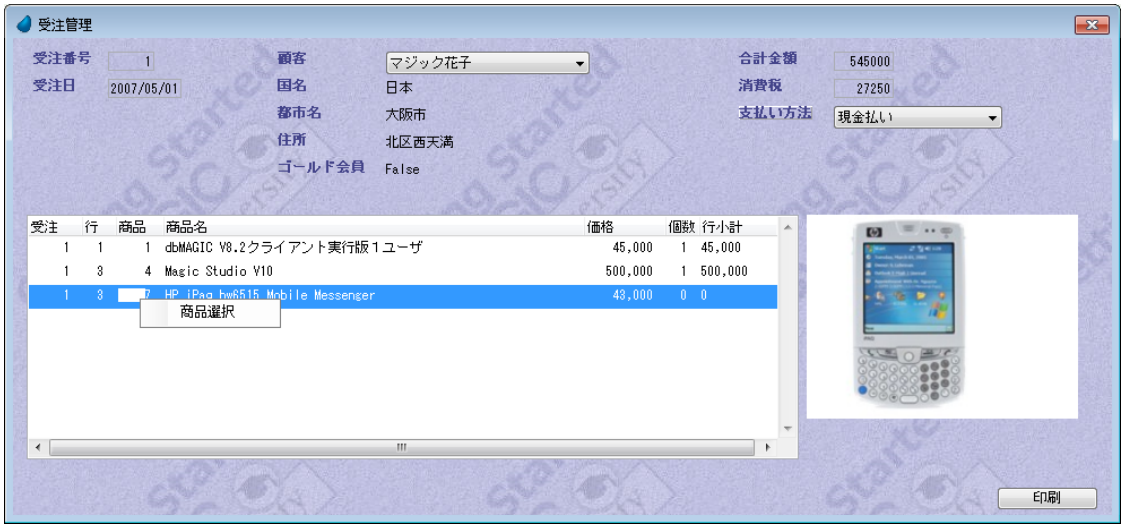
実行結果の確認

1. ＜受注管理＞プログラムを起動します。
2. ＜商品＞カラムで右クリックしてみましょう。

＜商品選択＞コンテキストメニューが表示されます。

＜商品選択＞を選択すると、商品一覧が表示されることが分かります。

すでに説明しているように、商品一覧から商品を選択することは、＜商品＞カラムでのみ有効なアクションです。したがってコンテキストメニューを使用することは、このような用途には最適な方法と言えるでしょう。



＜在庫金額合計＞コンテキストメニューの作成

特定のコントロールに割当てられることなく、プログラムの実行中にどこでも使用できるコンテキストメニューがあります。

これから実習する例では、＜商品照会＞プログラムで＜在庫金額合計＞コンテキストメニューを作成します。このコンテキストメニューは、どのコントロールにパークしていようと、いつでも在庫を確認する処理を起動します。

在庫金額合計イベントの作成

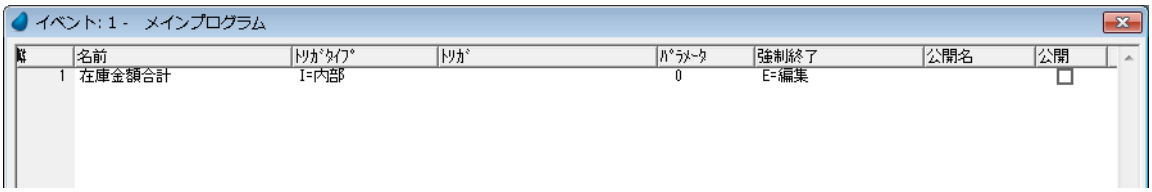
在庫数を計算するために、コンテキストメニューから発動されるイベントを定義する必要があります。イベントは＜商品照会＞プログラムにハンドラを定義します。

Magic xpa では、メインプログラムで定義されたイベントを発動するようにメニュー定義することができます。そこで、まずメインプログラムに在庫金額合計イベントを定義することにします。

1. ［プログラム］リポジトリを開きます。
2. メインプログラムにズームします。
3. ［タスク環境］メニューから［ユーザイベント］を選択します。

4. 一行作成（F4）します。
5. ［名前］カラムに、**＜在庫金額合計＞** と文字入力します。
6. ［強制終了］カラムで、［E= 編集］を選択します。

7. 変更内容を保存し、プログラムを閉じます。

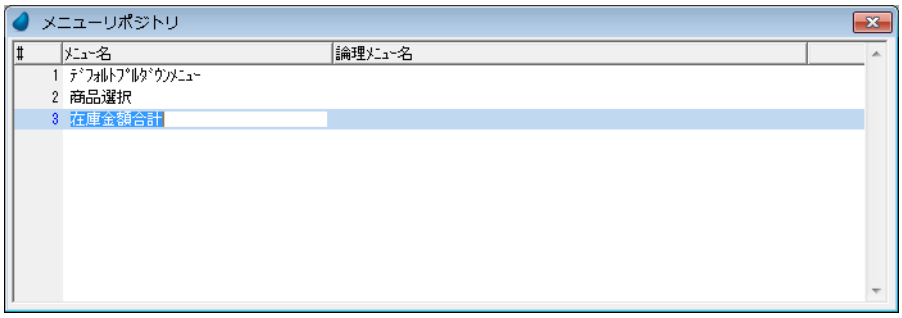


［強制終了］カラムを［E= 編集］に設定しておくのは、＜商品価格＞や＜在庫数量＞の更新後にカーソルがそのコントロールに留まっている場合、更新前の値でイベントハンドラが実行されるためです。［E= 編集］を選択することで、ハンドラは更新された（新しい）値を使用するようになります。

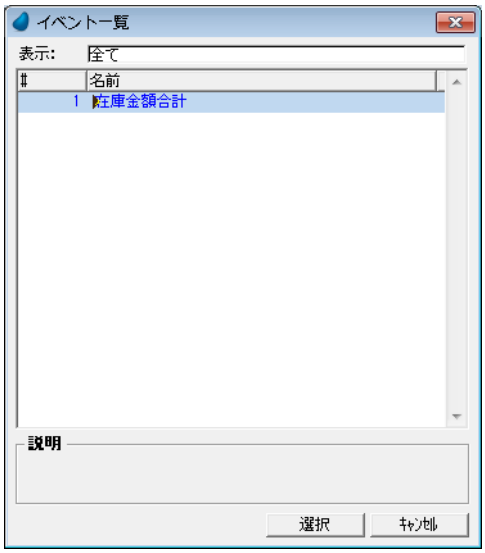
次にメニューを編集します。

1. ［メニュー］リポジトリ（Shift+F6）を開きます。
2. 最終行にパークします。
3. 一行作成（F4）します。

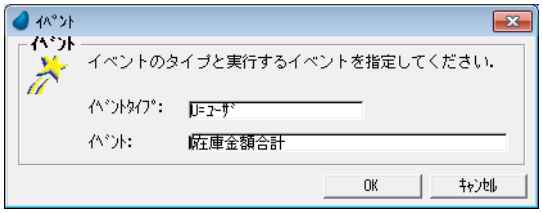
4. ［メニュー名］として、＜在庫金額合計＞と文字入力します。



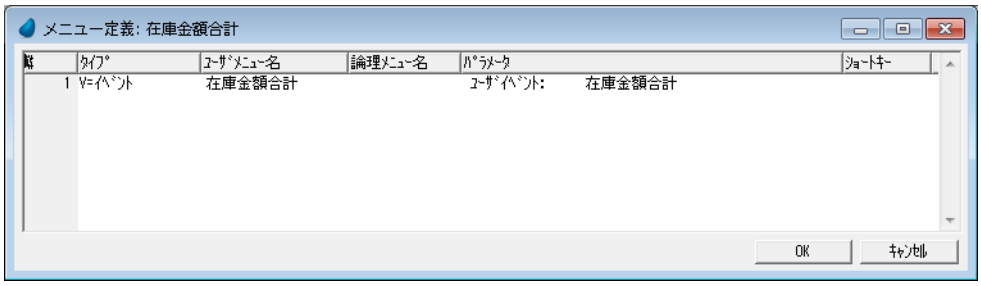
5. ［メニュー定義］テーブルにズームします。
6. 一行作成します。
7. ［タイプ］カラムで、［V= イベント］を選択します。
8. ［ユーザメニュー名］カラムで、＜在庫金額合計＞と文字入力します。



9. ［パラメータ］カラムでズームし、［イベント］ダイアログを開きます。
10. ［イベントタイプ］欄で、［U= ユーザ］を選択します。
11. ［イベント］欄からズームして、イベント一覧を開きます。
12. ＜在庫金額合計＞を選択します。
13. ［OK］ボタンをクリックします。



14.変更内容を保存し、メニュー定義テーブル、［メニュー］リポジトリを閉じます。

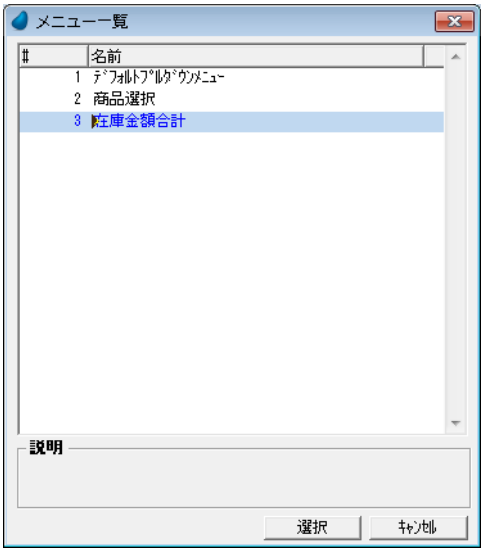


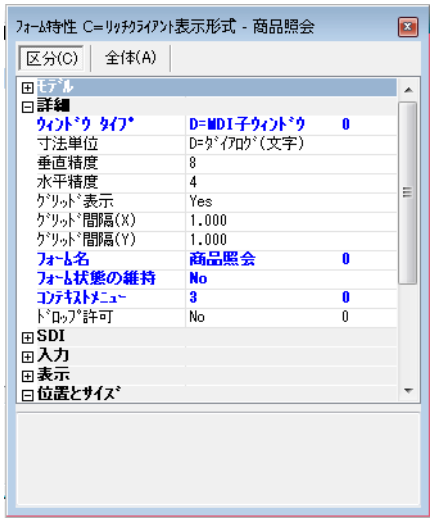
＜在庫金額合計＞コンテキストメニューのフォームへの割当て

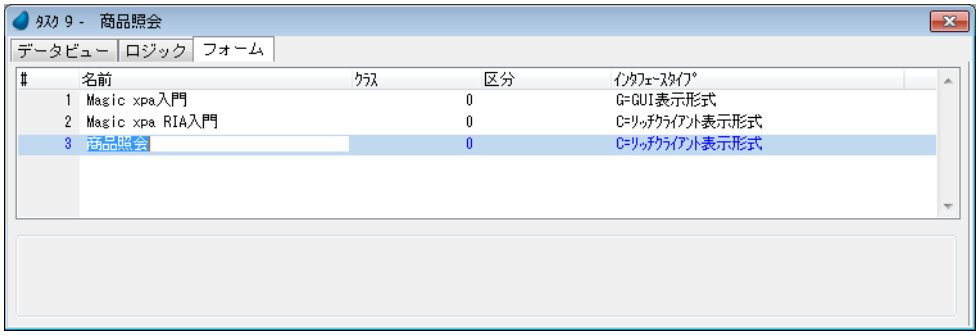
ここでは＜在庫金額合計＞コンテキストメニューを＜商品照会＞フォームに割当てましょう。こうすることで、カーソルがどこにパークしているかに関係なく、コンテキストメニューが使えるようになります。

1. ［プログラム］リポジトリを開きます。
2. ＜商品照会＞（#9）プログラムにズームします。
3. ［フォーム］エディタを選択します。

4. ＜商品照会＞フォームを選択し、特性を表示します。
5. ［ウィンドウタイプ］を「D=MDI 子ウィンドウ」に変更します。
6. ［コンテキストメニュー］からズームし、メニュー一覧を表示します。
7. ＜在庫金額合計＞（#3）を選択します。





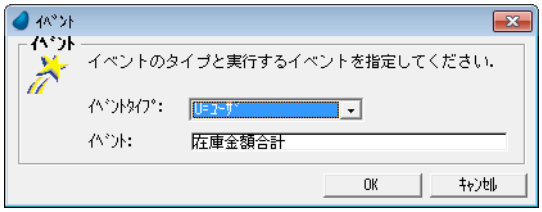


＜在庫金額合計＞イベントのハンドラ

次に、［イベント］ロジックユニットを定義し、＜在庫金額合計＞イベントのハンドラを作成しましょう。

1. ［ロジック］エディタを開きます。
2. ［イベント］ロジックユニット（Ctrl+H）を作成します。
3. イベントエントリからズームして、［イベント］ダイアログを開きます。

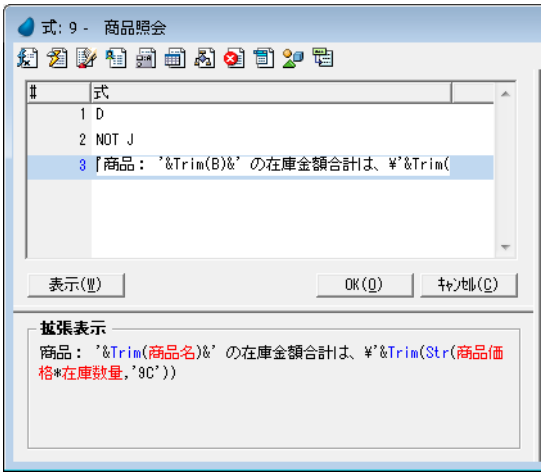
   a. ［イベントタイプ］欄で、［U= ユーザ］を選択します。
   b. ［イベント］欄からズームしてイベント一覧を開きます。
   c. ＜在庫金額合計＞を選択します。
   d. ［OK］ボタンをクリックします。



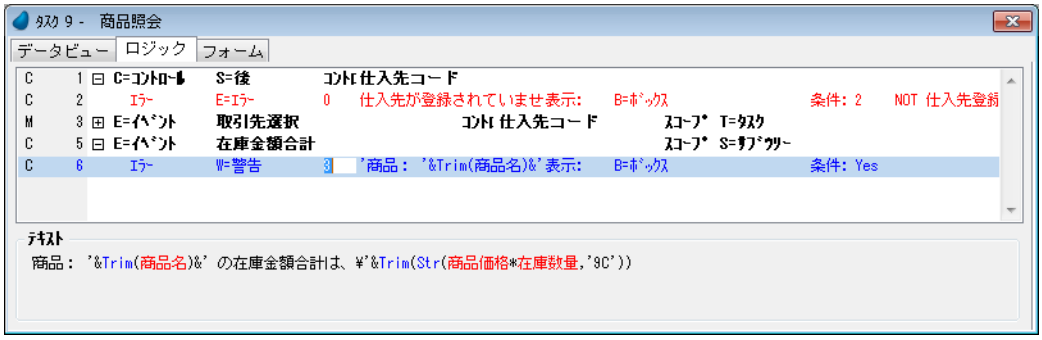
4. ［イベント］ロジックユニットに一行作成します。
5. 処理コマンドとして、［エラー］を選択し、メッセージの［式］欄からズームし、［式］エディタにて一行作成し、次の文字列を入力します。

   ＜’商品：’&Trim(B)&’ の在庫金額合計は、¥’&TRIM(STR(E*F,’9C’))＞

   ここで、B: 商品名、E：商品価格、F: 在庫数量



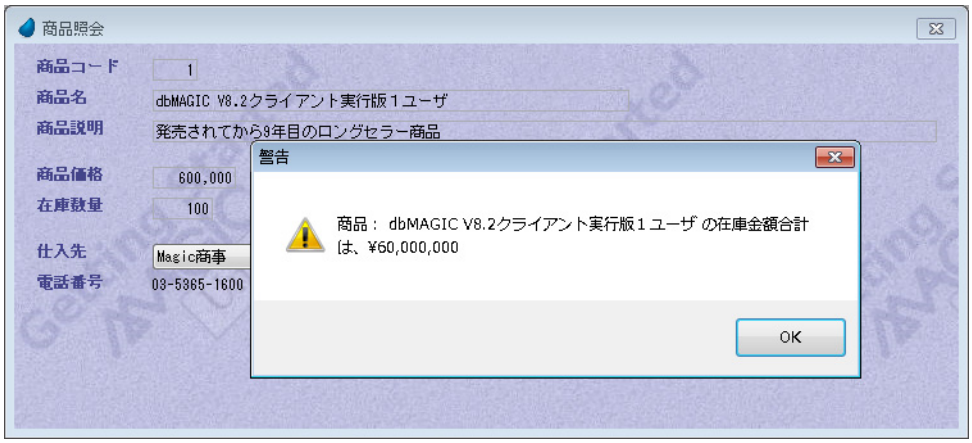
6. 変更内容を保存し、プログラムを閉じます。



実行結果の確認

1. ＜商品照会＞プログラムを実行します。
2. フォーム上の任意の場所で右クリックします。＜在庫金額合計＞コンテキストメニューが表示されます。

3. メニューを選択し、実行結果を確認しましょう。



６．メニュー特性

メニュー特性には 3 つのタブがあります。

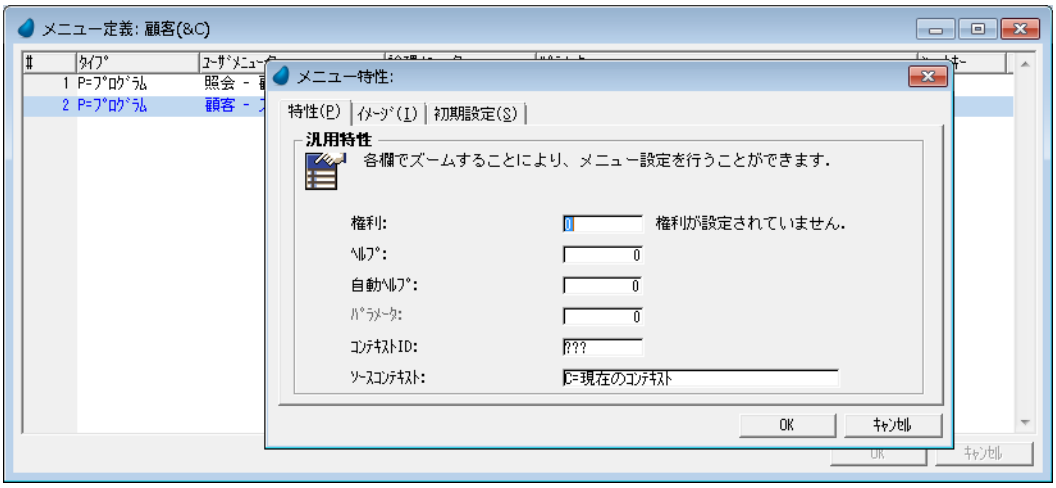
- 特性 …… メニューへの権利設定、ヘルプ設定などを行ないます。
- イメージ …… メニューに対応したイメージやツールチップの設定を行います。
- 初期設定 …… メニューの初期状態を設定します（チェックマーク、可視、有効）。

次の例では、メニューにイメージを設定します。

＜顧客一覧＞メニューへのイメージ設定

1. ［メニュー］リポジトリ（Shift+F6）を開きます。
2. 「デフォルトプルダウンメニュー」にズームし、［メニュー定義］テーブルを開きます。
3. ＜顧客 (&C) ＞（#4）メニューにパークします。

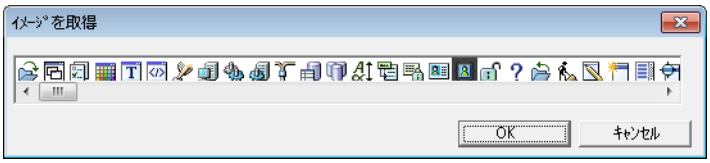
4. サブメニューエントリからズームして、サブメニューの定義テーブルを開きます。
5. ＜顧客ースクリーンモード＞（#2）メニューにパークします。
6. メニュー特性 （Alt+Enter）を開きます。



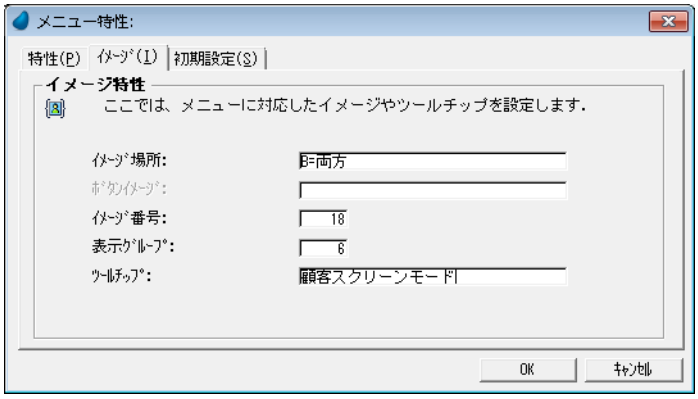
7. ［イメージ］タブをクリックします。

8. ［イメージ場所］特性で、［B= 両方］を選択します。
9. ［イメージ番号］特性で、＜ 18 ＞を指定します。

ここで指定する＜ 18 ＞というのは、Magic xpa 内部のツールイメージの位置を意味する番号です。イメージの番号がわからない場合には、F5 でズームして表示される一覧から選択することもできます。



10.［表示グループ］特性には、＜ 6 ＞と入力します。（この値の意味は次ページで説明します。）
11.［ツールチップ］特性には、＜顧客スクリーンモード＞と入力します。（この欄の説明は次ページで行ないます。）
12.［OK］ボタンをクリックします。
13.変更内容を保存し、メニューを閉じます。



メニューエントリは、この例のように Magic xpa 内部のイメージを使用することができますが、［ボタンイメージ］特性で指定する外部ファイルのイメージを使用することもできます。

メニュー特性のイメージタブについて

メニュー特性の イメージ タブで指定する項目の意味は、以下の通りです。

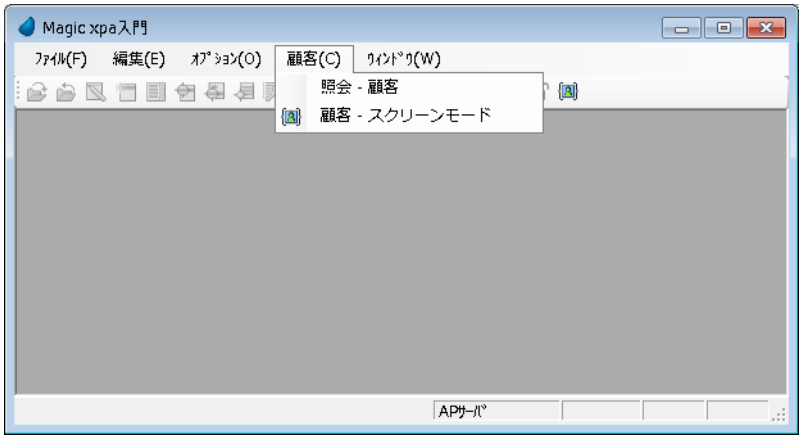
特性	説明
イメージ場所	この特性は、オプションイメージが表示される場所を指定します。次のオプションから選択します。 • N ＝なし …… メニューイメージを表示しません。 • B ＝両方 …… メニューイメージをツールバーとメニューに表示します。 • T ＝ツールバー …… メニューイメージをツールバーだけに表示します。 • M ＝メニュー …… メニューイメージをメニューだけに表示します。
ボタンイメージ	ユーザイメージファイルを指定します。指定したイメージがツールバーボタンに表示されます。
イメージ番号	内部ツールボタンイメージの番号です。ここからズームして［表示内容を取得］ダイアログから選択します。
表示グループ	ツールボタンをグループ分けする場合のグループの番号です（0 ～ 9999）。グループ間にはセパレータが挿入されます。
ツールチップ	50 桁以内の文字列を入力します。マウスカーソルをツールボタン上に置いた時に表示されます。

結果の確認

1. ［デバッグ］メニューから、［プロジェクトの実行］（Ctrl+F7）を選択します。
2. ＜顧客 (C) ＞メニューを開きます。
3. ＜顧客ースクリーンモード＞メニューテキストの左にイメージが表示されていることを確認してください。

4. プルダウンメニューを閉じます。

5. ツールバーの右端に、設定したイメージアイコンが表示されていることを確認してください。
6. イメージアイコンの上にマウスカーソルを移動させると、ツールチップの文字列が表示されることを確認しましょう。



ツールグループについて

ツールイメージをグループに分け、［ツールグループ］特性の番号を使用して、表示順序を決めることができます。

ツールイメージは、［ツールグループ］特性の値にしたがって、左側から順に表示されるようになっています。（最も小さい値のものが一番左になります。）

各グループはセパレータによって分けられます。

本コースの例では 5 つのグループがすでにありました。＜顧客－スクリーンモード＞メニューのメニューから［ツールグループ］特性に＜ 6 ＞を設定したことで、このメニューエントリに、一番右側に表示される新しいグループを作成したことになります。

７．練習問題

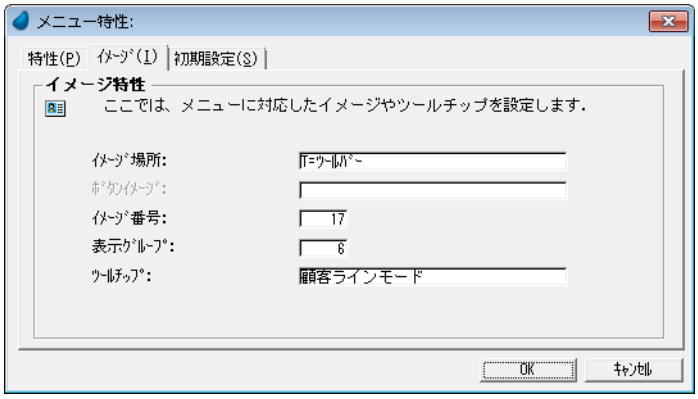
＜顧客 (&C) ＞メニューにエントリを追加しましょう

- 次のメニューエントリを＜顧客 (&C) ＞メニューに追加してください。
  - プログラム #5：＜照会－顧客ラインモード＞
  - プログラム #6：＜顧客－ラインモード＞

ツールバーおよびメニューイメージの設定

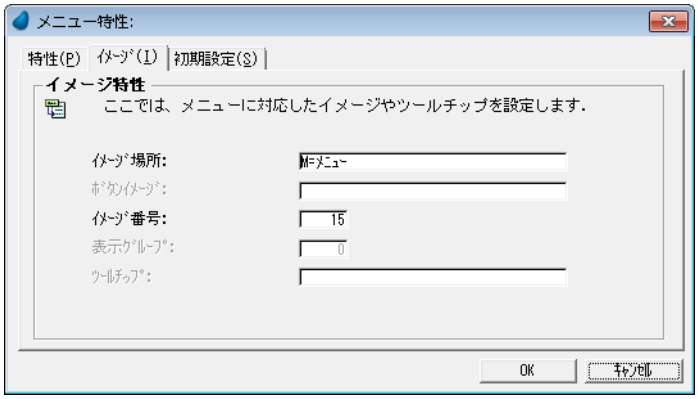
＜照会－顧客ラインモード＞メニューに次の設定をしてください。

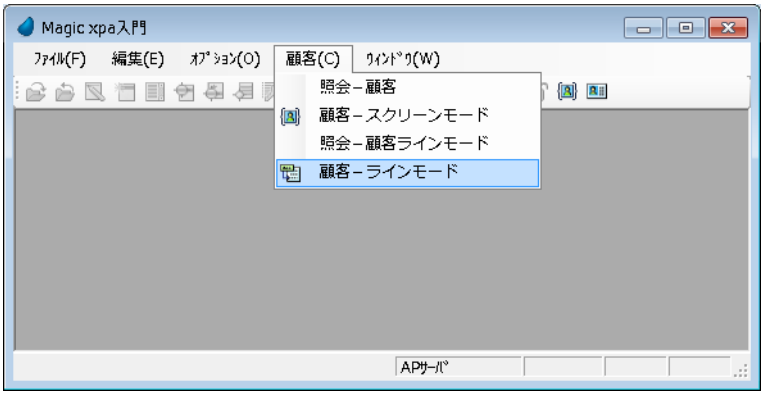
- イメージ場所 …… ［T= ツールバー］
- イメージ番号 …… ＜ 17 ＞
- 表示グループ …… ＜ 6 ＞
- ツールチップ …… ＜顧客ラインモード＞



＜顧客－ラインモード＞メニューに次の設定をしてください。

- イメージ場所 …… ［M= メニュー］
- イメージ番号 …… ＜ 15 ＞





８．要約

メニューの作成は、顧客にアプリケーションを配布する前に行なう、最終的な作業です。

本章では、［メニュー］リポジトリと関連する諸機能について学びました。

メニューには大別してプルダウンメニューとコンテキストメニューがあることが分かりました。

新規プロジェクト登録時に、デフォルトのプルダウンメニューが登録され、そこにエントリの追加、修正、削除が可能であることが分かりました。

コンテキストメニューを作成し、コントロールやフォームに割当てる実習を行ないました。

メニューには特性があり、ツールバーやメニューで使用するイメージをその特性に定義する実習を行ないました。

本章でプロジェクトの作成に関する説明はすべて終了です。基本的にオンラインプログラムを利用した作成方法と同じような手順になっていました。このため次章では、リッチクライアントアプリケーションを作成する上で考慮すべき事項について説明してゆきます。

第 23 章 リッチクライアントに関する考慮事項

本章では RIA を開発するうえで、いくつか考慮する内容について学びます。

キーワード

- 実行環境
- 遅延トランザクション
- サーバ／クライアント処理

学習目標

本章を学ぶことで、次の内容に対する理解を深めましょう。

- RIA の実行環境
- 遅延トランザクションの意味と設定
- 処理の実行場所と最適化

参照

Magic ヘルプの［Magic xpa リファレンス／ Web 開発／リッチクライアントアプリケーション］

１．はじめに

今まで、RIA を開発する上での基本的なプログラム作成の方法について学んできました。しかし、このアプリケーションがインターネット環境で運用されることを前提とした説明はありませんでした。まずは、プログラムを作ってみてもらいました。

このように Magic xpa を使用することで、あまり意識することなくリッチクライアント用のプログラムを作成することができます。

しかし、RIA を実際に運用する場合は、インターネット環境を考慮する必要があります。ここでは、そのような観点で今まで作成したプログラムを確認してみましょう。

２．RIA の運用環境について

RIA は、インターネットアプリケーションです。これは、サーバ上で実行される Magic エンジンにインターネット経由で世界中のコンピュータからアクセスして実行させることができることを意味します。

このため、スタンドアロンやクライアント／サーバ用のアプリケーションと比較して以下のような点を考慮する必要があります。

- 不正アクセスに対するセキュリティ
- ネットワーク上の通信速度やトラフィック
- 遅延トランザクション
- 処理の最適化

不正アクセスに対するセキュリティ

インターネット上では、様々な不正アクセスが発生する場合があります。ネットワーク上の階層ごとに防止策があります。Magic xpa は、暗号化処理や機能の分散化をビルトインで提供することで対応していますが、ネットワーク環境を設定する際も考慮していただくことを推奨します。

Magic xpa には、『リッチクライアント のセキュリティ』というドキュメント が添付されています。ここに、セキュリティ に関する情報を確認することができます。

ネットワーク上の通信速度やトラフィック

インターネット環境は、クライアント／サーバ環境に比べ通信速度が遅かったり、切断したり、場合によっては、クライアント側の利用者が途中で中断するような場合も考慮する必要があります。

Magic xpa の RIA では、クライアント／サーバ間の通信は、圧縮された XML データでやり取りされます。この通信バスはボトルネックとなる場合があり、これによってプログラムのパフォーマンスに影響を及ぼすことになります。このため、効率的なプログラムを作成するには、やり取りする通信量を最小にする必要があります。

また、プログラムを作成する上で表示項目を少なくしたり、表示するイメージのサイズを小さいものにしたり 、更新データを少ないものにする工夫が必要です。

遅延トランザクション

遅延トランザクションは、データビューのレコードに対する変更をコミットする必要が出るまで、データ操作（DML）ステートメントの実行を遅らせる機能です。この機能によって DML ステートメントはキャッシュ内に保存されます。変更をデータベースにコミットする準備が整うと、全ての変更が同時に実行されます。これにより、トランザクションの時間が可能な限り短縮されます。

Magic xpa では、後述する遅延トランザクション機能によって、できるだけデータを更新するタイミングを短くするようにしています。

処理の最適化

Magic xpa では、プログラム内でサーバ側で実行する処理とクライアント側で実行される処理が混在する場合があります。これは、サーバ用のアプリケーションとクライアントアプリケーションを個別に作成する必要がないという利点があるある反面、サーバ側とクライアント側の処理が頻繁に切り替わる場合パフォーマンスの低下につながります。

処理を実行する場所の最適化については後述します。できるだけ、処理する場所の切り替えが発生する回数を減らすように検討してください。Magic xpa には、この作業を支援するための機能が備えられています。これについては、後述します。

３．遅延トランザクションについて

RIA では、DBMS アクセスはサーバ側で発生しますが、ユーザ操作はクライアント側で発生します。このため、一旦データがクライアントに送信された場合、レコードがコミットされる準備ができるまで、ユーザはサーバにアクセスすることなくフィールドからフィールドにタブ移動できるようにする必要があります。このような理由で、リッチクライアントプログラムは、遅延トランザクションを使用しています。

Magic xpa は、遅延トランザクションにデータの変更内容を保存し、トランザクションが終了した時点でコミットします。トランザクションがロールバックされると、保持されたトランザクションは廃棄されます。

遅延トランザクションを使用することで以下の利点があります。

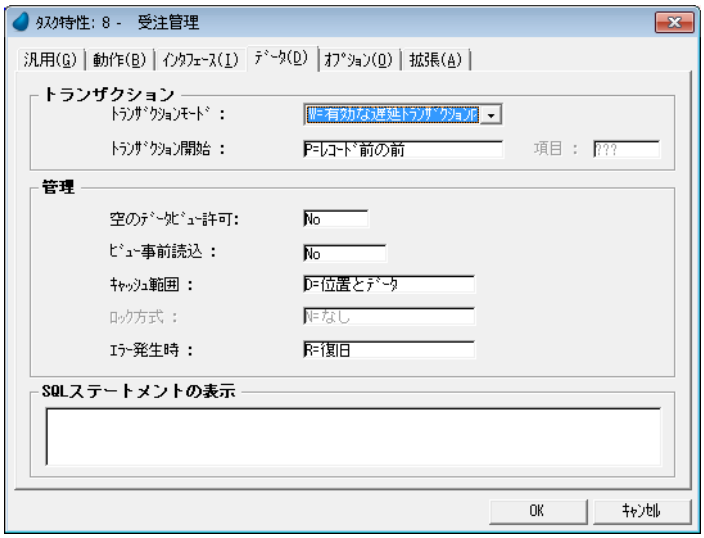
- サーバ側の負荷を減らすことができます。
- アプリケーションとデータベース間の通信処理を減らすことができます。
- パフォーマンスが向上します。
- 並行性が高くなります。

プログラムの確認

親タスクのトランザクション設定

ここでは、以前作成した＜受注管理＞プログラムで、トランザクションの設定内容を確認してみましょう。

1. ［プログラム］リポジトリから＜受注管理＞を開きます（F5）。
2. ［編集］メニューから［特性］（Alt+Enter) を選択して、［タスク特性］を開きます。
3. ［データ］タブをクリックします。



トランザクションモード

リッチクライアントで使用することができるトランザクションモードは以下の通りです。

- なし …… トランザクションはオープンされません。
- 新規の遅延トランザクション …… 新しいトランザクションが開始されます。親のトランザクション内で発生した内容に関係なく、子のトランザクションは子タスクレベルでコミットされます。
- 有効な遅延トランザクション内 …… トランザクションが開始されていない場合、新規の遅延トランザクションを開始します。開始されている場合は、タスクはそのトランザクションの一部になります。

＜受注管理＞プログラムは、「有効な遅延トランザクション内」が定義されています。親タスクのため、実行前はトランザクションは開始されておらず、新規の遅延トランザクションを開始することになります。

トランザクション開始

すべての DML ステートメントは、リッチクライアントモジュールによって維持され、［トランザクション開始］の設定内容に基づいてサーバに送られます。

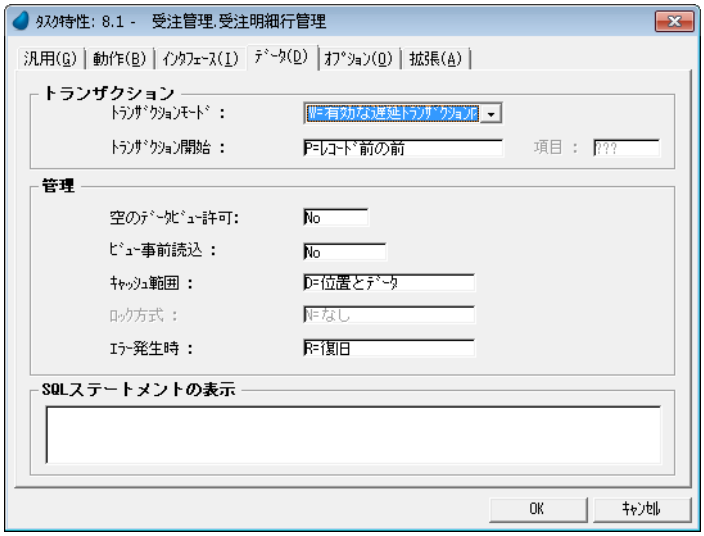
- レコード前の前 …… 変更内容は、レコードレベルのトランザクション内のレコードを終了した後にコミットされます。
- タスク前の前 …… 変更内容は、タスクレベルのトランザクション内でタスクを閉じた後にコミットされます。

＜受注管理＞プログラムは、「レコード前の前」が定義されています。インタラクティブなタスクは、レコードを操作する時間が長くなる可能性があるため、レコードレベルでトランザクション処理を実行させる必要があります。

子タスクのトランザクション設定

次に、子タスクを確認してみましょう。

1. タスクツリーから＜ 受注明細管理＞を開きます。
2. ［編集］メニューから［特性］(Alt+Enter) を選択して、［タスク特性］を開きます。
3. ［データ］タブをクリックします。



このタスクの［タスク特性］にも同じ値が設定されていますが、子タスクの場合は動作が異なります。［トランザクションモード］特性の「有効な遅延トランザクション内」は、親タスクのトランザクションの一部となり、タスクが終了しても親タスクでコミットされるまでサーバ側への書き込みは行われません。これによって、親子タスク内のレコードの整合性が保たれます。

では確認してみましょう。

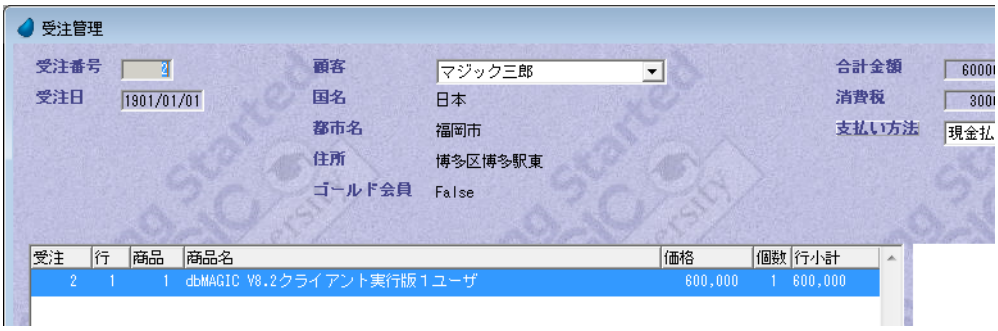
＜受注管理＞プログラムの実行

1. ［プログラム］リポジトリに戻り、＜受注管理＞を実行（F7）します。
2. PageDown キーを［受注番号］が＜ **0** ＞になるまで押下します。
3. 以下のように受注データを入力します（受注番号は、既存の番号と重複しない値を入力してください）。

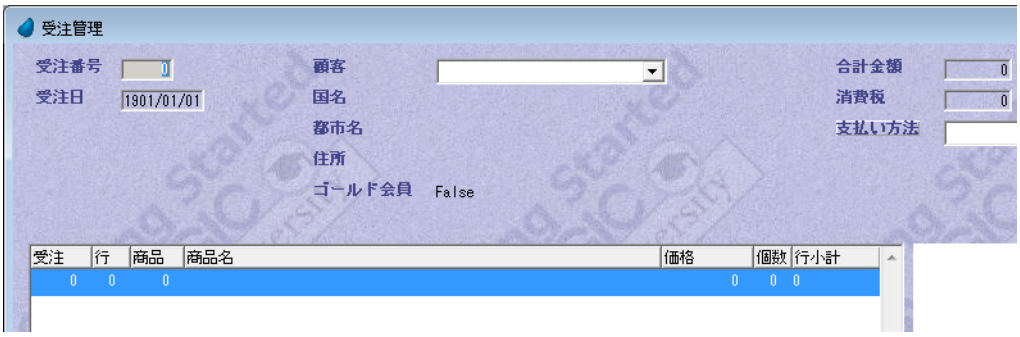
受注番号	顧客	支払い方法
2	マジック三郎	現金払い

4. 受注明細行に移動して、以下のように明細データを入力します。

行	商品	個数
1	1	1



5. この状態でフォーカスを＜受注番号＞に移し、キャンセル（Ctrl+F2）を実行します。入力したデータが全てクリアされた状態になります。



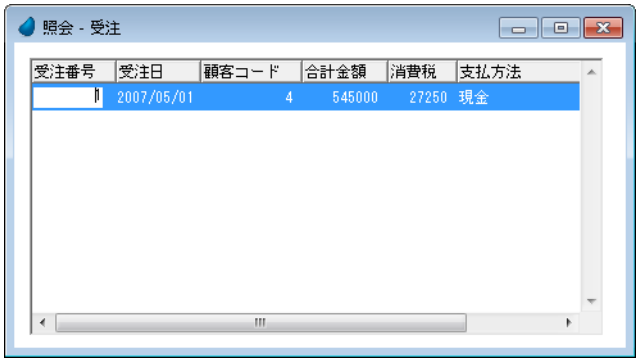
6. Esc キーを押下してプログラムを終了してください。

このとき入力されたデータはどうなっているかを確認してみましょう。

［データ］リポジトリでの確認

1. ［データ］リポジトリ（Shift+F2）を開き、＜受注＞（#3）にカーソルを置き、APG（Ctrl+G）を実行します。
2. APG ダイアログで、［モード］を「E= 実行」、［オプション］を「B= 照会」に設定します。
3. ［OK］ボタンをクリックします。

4. 以下のように表示され、受注番号 =2 のレコードは存在しないことが確認できます。



5. Esc キーを押下して APG を終了してください。
6. ＜受注明細行＞（#5）にカーソルを置き、同じように APG（Ctrl+G）を実行します。
7. 以下のように表示され、やはり受注番号 =2 のレコードは存在しないことが確認できます。

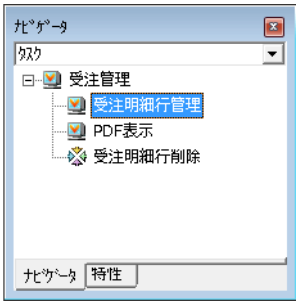
このように、親タスクで修正処理をキャンセル（ロールバック）することで子タスクの入力内容も一緒にキャンセルされます。

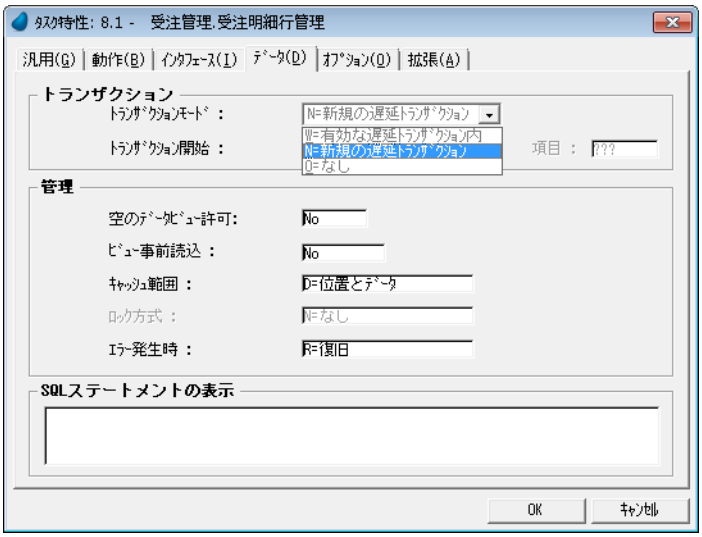
8. Esc キーを押下して APG を終了してください。

では、子タスクの［トランザクションモード］特性を「新規の遅延トランザクション」に設定した場合、どうなるでしょうか？

子タスクのトランザクションモードを変更する

1. ［プログラム］リポジトリ（Shift+F3）に切り替え、＜受注管理＞を開きます（F5）。
2. ［ナビゲータ］から＜受注明細管理＞を選択します。
3. ［タスク特性］（Ctrl+P）を開きます。
4. ［データ］タブをクリックします。
5. ［トランザクションモード］特性を「N= 新規の遅延トランザクション」に変更します。





6. ［OK］ボタンをクリックして設定を保存します。
7. Enter を押下して［プログラム］リポジトリに戻ります。

＜受注管理＞プログラムの実行

1. ［プログラム］リポジトリから＜受注管理＞を実行（F7）します。
2. PageDown キーを［受注番号］が＜ **0** ＞になるまで押下します。
3. 前述と同じように受注データを入力します。

受注番号	顧客	支払い方法
2	マジック三郎	現金払い

4. 受注明細行に移動して、以下のように明細データを入力します。

行	商品	個数
1	1	1

5. フォーカスを［受注番号］に移し、キャンセル（Ctrl+F2）を実行します。入力したデータが全てクリアされた状態になります。
6. Esc キーを押下してプログラムを終了してください。

このとき入力されたデータはどうなっているかを確認してみましょう。

［データ］リポジトリでの確認

1. ［データ］リポジトリ（Shift+F2）を開き、＜受注＞（#3）にカーソルを置き、APG（Ctrl+G）を実行します。
2. APG ダイアログで、［モード］を「E= 実行」、［オプション］を「B= 照会」に設定します。
3. ［OK］ボタンをクリックします。

4. 前回と同じように、受注番号 =2 のレコードは存在しないことが確認できます。

Esc キーを押下して APG を終了してください。

5. ＜受注明細行＞（#5）にカーソルを置き、同じように APG（Ctrl+G）を実行します。
6. 以下のように表示され、受注番号 =2 にレコードは残っていることが確認できます。

このように、子タスクの［トランザクションモード］特性を「N= 新規の遅延トランザクション」にすると、親タスクとは別のトランザクションが開始されるため、親タスクでキャンセルしても子タスクのデータはキャンセルされず親子のデータの関係が連動しなくなってしまうことになります。

トランザクションの設定は、タスクの構成によって様々な組み合わせが考えられます。どの様に動作するかを把握しながら、実際にデータを書き込む時間を短くする工夫やデータの整合性を維持する工夫が必要になります。

４．処理の最適化

RIA では、処理内容によってサーバ側での処理とクライアント側での処理が発生します。

例えば、以下の処理はサーバ側で実行されます。

- バッチタスク
- リッチクライアントタスクの［タスク前］

- ［コール］処理コマンド

また、以下の処理はクライアント側で実行されます。

- ［エラー］処理コマンド

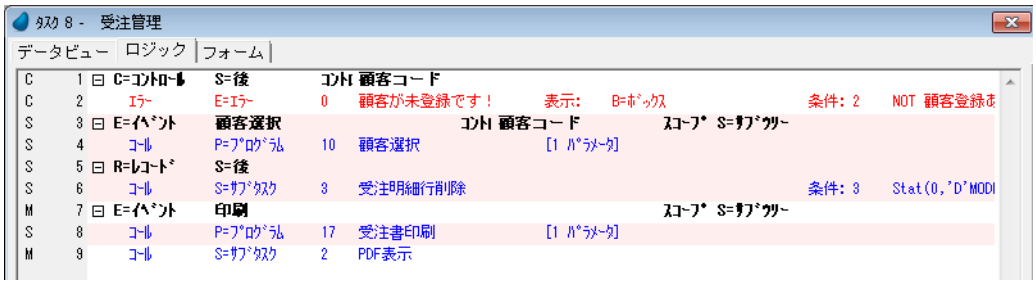
他にも、関数によってはサーバ側で実行されるものとクライアント側で実行されるものがあります。

これらの組み合わせによって、処理コマンドがサーバ側またはクライアント側のどちらで実行されるかが決まる場合もあります。どちらで実行されるかを確認できるようにするため、Magic xpa は、マークを表示したり行を色分けしています。

確認してみましょう。

処理コマンドの実行箇所の確認

1. ［プログラム］リポジトリ（Shift+F3）に切り替え、＜受注管理＞を開きます（F5）。
2. ［ロジック］エディタ（Ctrl+2）に切り替えます。



- クライアントで処理される行（例：#2）は、行番号カラムの先頭に ”C” が表示され、行が白色で表示されます。
- サーバ側で処理される行（例：#4）は、行番号カラムの先頭に ”S” が表示され、行がピンク色で表示されます。
- 行番号カラムの先頭に ”M” が表示されている場合（例 : #9）は、サーバの処理とクライアントの処理が混在していることを示しています。

また、構文チェックユーティリティを使用して確認することもできます。

構文チェックで確認する

Magic xpa には、プログラムが文法的に問題がないかどうかをチェックするための手段として構文チェックユーティリティが備えられています。このユーティリティを使用して、確認してみましょう。

1. ＜受注管理＞を開いた状態で、最終行に移動します。
2. ヘッダ行（Ctrl+H）を追加します。
3. 以下のように入力して、［タスク前］ロジックユニットを定義します。

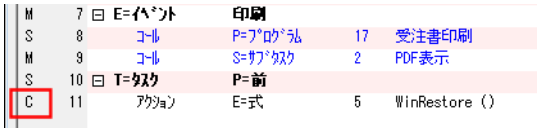
ロジックユニットタイプ	イベント
T= タスク	P= 前

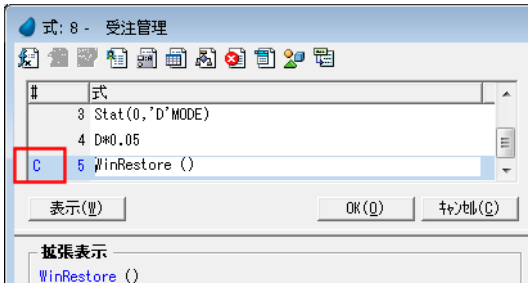
4. 一行作成（F4）して、以下のように［アクション］処理コマンドを入力します。

コマンド		値
T= タスク	E= 式	5 WinRestore()

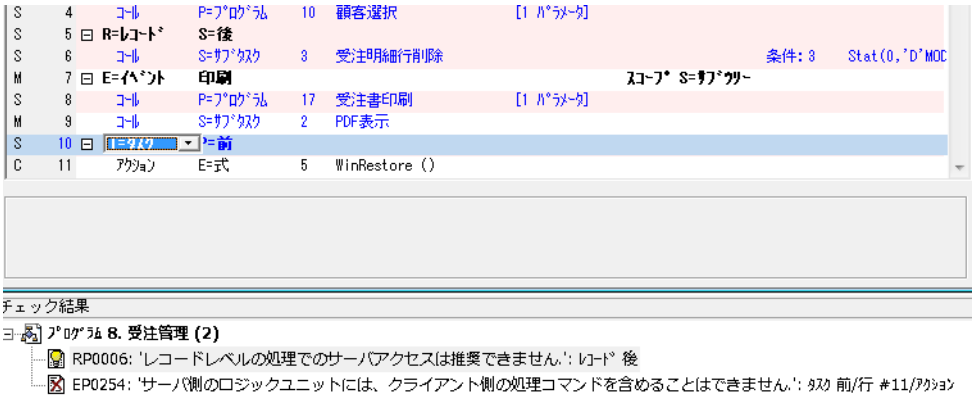
5. ［式］エディタを開いて WinRestore() の定義行を確認してください。

行番号カラムには、やはり ”C” というマークが表示されています。この関数がクライアントで実行されることを示しています。これにより、［アクション］処理コマンドは、クライアントで処理されることになり、［ロジック］エディタ上には、”C” というマークが表示されます。





6. Enter キーを押下してこのプログラムを閉じます。
7. ＜受注管理＞で F8 キーを押下して構文チェックユーティリティを実行します。［チェック結果］ペインが表示され、問題のあるプログラムが表示されます。



この例では、以下のような結果表示になっています。

- RP0006: レコードレベルの処理でのサーバアクセスは推奨されません。…… ［レコード後］でサブタスクを呼び出す処理（#6）を指摘しています。これはエラーではありませんが、レコードレベルでサーバへのアクセスが発生することでパフォーマンスが低下する可能性があることに対する警告表示です。
- EP0254: サーバ側のロジックユニットは、クライアント側の処理コマンドを含めることはできません。…… ［タスク前］は、サーバ側で実行されるため、クライアントで実行される処理を定義できません。このため、エラーとして表示しています。この処理（#11）は削除する必要があります。

サーバ側で実行される処理とクライアントで実行される処理が混在する場合、できるだけ制御の切り替えが少なくなるように工夫する必要がありますが、Magic xpa はこのような状態を視覚的に表示して、改善しやすくなっています。

５．要約

本章ではリッチクライアントのプログラムを作成する上で、気を付ける必要のあることを何点か学びました。

- ネットワーク上のトラフィックを減らすにはどうするか。
- データ更新の時間を短縮するにはどうするか。
- プログラムの最適化とは、その改善方法

次章では、作成したアプリケーションを公開する方法について学習します。

第 24 章 アプリケーションの公開

本章では、作成した RIA を別の PC からアクセスするための方法について説明します。

キーワード

- 環境設定
- リッチクライアントインタフェースビルダ
- 公開用 HTML ファイル

学習目標

本章を学ぶことで、次の内容に対する理解を深めましょう。

- RIA を公開するための環境設定内容
- 公開に必要なファイルの作成方法
- 他の PC からアクセスする方法

参照

Magic ヘルプの［Magic xpa リファレンス／プロジェクトとアプリケーション］
［Magic xpa リファレンス／ Web 開発／リッチクライアントアプリケーション］

１．はじめに

RIA は、（インターネットを含む）リモート上の他の PC からアクセスすることを前提としています。このため、作成したアプリケーションを別の PC からアクセスできるように環境設定を行う必要があります。

Magic xpa で RIA を実行するために必要なモジュールには、以下のものがあります。

- Web サーバ …… リモートクライアントからリクエストを受け取るために、Web サーバが必要です。Web サーバは、Magic xpa サーバにリクエストを送ります。
- インターネットリクエスタ …… クライアントと Magic xpa サーバの間を仲介するモジュールです。このモジュールは、サーバのプールで 1 つのアイドル状態の Magic xpa サーバにリクエストとデータを渡すことができます。これによって負荷を分散します。
- MRB（Magic Request Broker）…… Magic xpa によって提供されるミドルウェアエージェントです。MRB は、利用可能なすべての Magic xpa サーバエンジンを処理して、インターネットリクエスタから利用可能なサーバエンジンへの各リクエストを指示します。様々なフェールオーバーに対応するため、ロードバランシング機能やリカバリ機能を提供します。
- Magic xpa サーバ …… RIA 環境の中心にあります。実際の実行時単位で、各リクエストを処理して、受け取った各タイプのリクエストに対応した全てのアプリケーションロジックを実行します。
- クライアント実行モジュール ……RIA は、クライアント側にも専用の実行モジュールが必要です。Microsoft の Click Once という技術を利用し .Net Framework 上で実行するモジュールをクライアントに配布する必要があります。Magic xpa は、開発したアプリケーションに対応するモジュールを作成し、配布するためのウィザードツールを提供しています。

ここで環境設定とは、以下のようなものを意味します。

- Web サーバ（IIS）の設定
- Magic xpa 側の動作環境
- アプリケーションの公開設定
- クライアント側の実行モジュールのインストール

本章では、アプリケーションを作成後、他の PC からアクセスするまでの環境設定について一通り学んで行きます。他にも、インターネット環境を利用する上での設定がありますが、ここでは省略します。

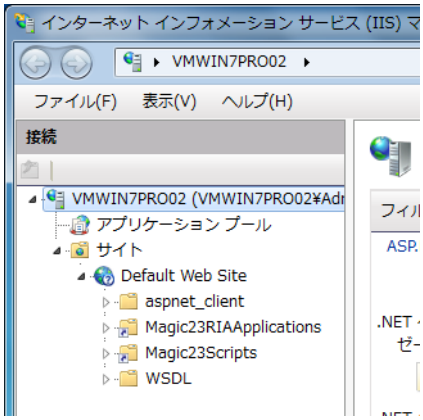
２．Web サーバの設定

Windows で IIS（Internet Information Services）が有効になっている場合、Magic xpa をインストールすると必要な環境設定を行うようになっています。

設定内容を確認してみましょう。

1. Windows のスタートメニューから［コントロールパネル］をクリックします。
2. コントロールパネルから［システムとセキュリティ／管理ツール］を選択します。
3. ［インターネットインフォメーションサービス（IIS）マネージャー］を選択します。IIS マネージャーのウィンドウが開きます。

4. ［接続］ペインのサーバ名をクリックしてツリーを開きます。
5. 「サイト／ Default Web Site」をクリックしていくと以下のエイリアス名が表示されます。
   - Magic23RIAApplications …… アプリケーションファイルの公開用
   - Magic23Scripts …… Magic xpa 用インターネットリクエスタの公開用



確認ができたらウィンドウを閉じてください。

上記のエイリアスはインストール時に自動的に定義されますが、IIS マネージャーで追加することもできます。

エイリアス名のデフォルト値は、Magic xpa のバージョンによって変更される場合があります。また、インストール時に変更することができます。

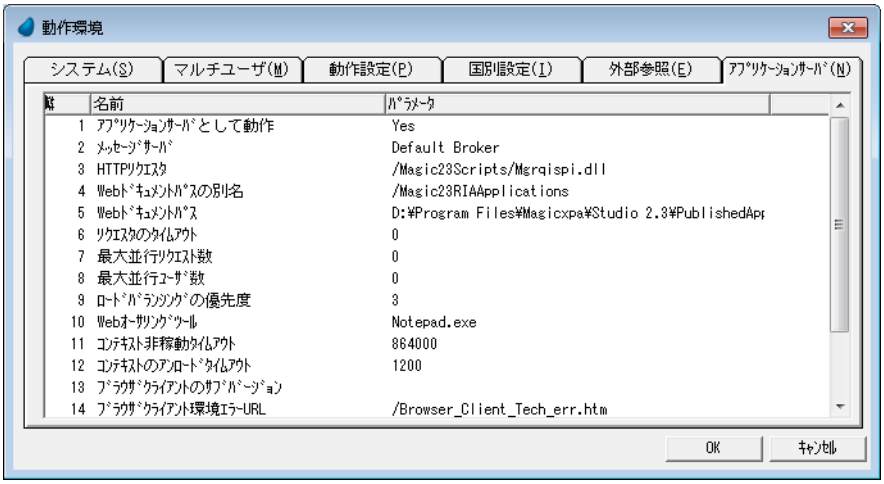
３．Magic xpa の動作環境

Magic xpa で RIA を実行させるための動作環境を確認しましょう。

1. Magic xpa の［オプション］メニューから［動作環境］を選択し、［動作環境］ダイアログを開きます。
2. ［アプリケーションサーバ］タブをクリックします。

RIA では、以下の設定項目が関係します。デフォルト値は、インストーラによって設定されます。

- アプリケーションサーバとして動作 …… Magic xpa がアプリケーションサーバとして動作するかどうかを指定します。
- メッセージサーバ …… MRB（Magic Request Broker）が動作するメッセージサーバを指定します。Magic xpa と MRB が異なる PC で実行する場合に変更する必要があります。
- HTTP リクエスタ …… インターネットリクエスタの ” エイリス名 + ファイル名 ” を設定します。
- Web ドキュメントパスの別名 …… アプリケーションの公開用エイリアス名を設定します。
- Web ドキュメントパス …… アプリケーションの公開用フォルダを設定します。
- 最大並行ユーザ数 …… RIA にアクセスできるユーザ数の最大数を指定します。利用するライセンス数の範囲で指定できます。
- コンテキスト非稼動タイムアウト …… 操作が行われない場合にコンテキストを保持する時間を指定します。
- リッチクライアントキャッシュパス …… RIA のキャッシュデータを保持するフォルダを指定します。



各設定内容の詳細は、リファレンスヘルプを参照してください。

４．アプリケーションの公開準備

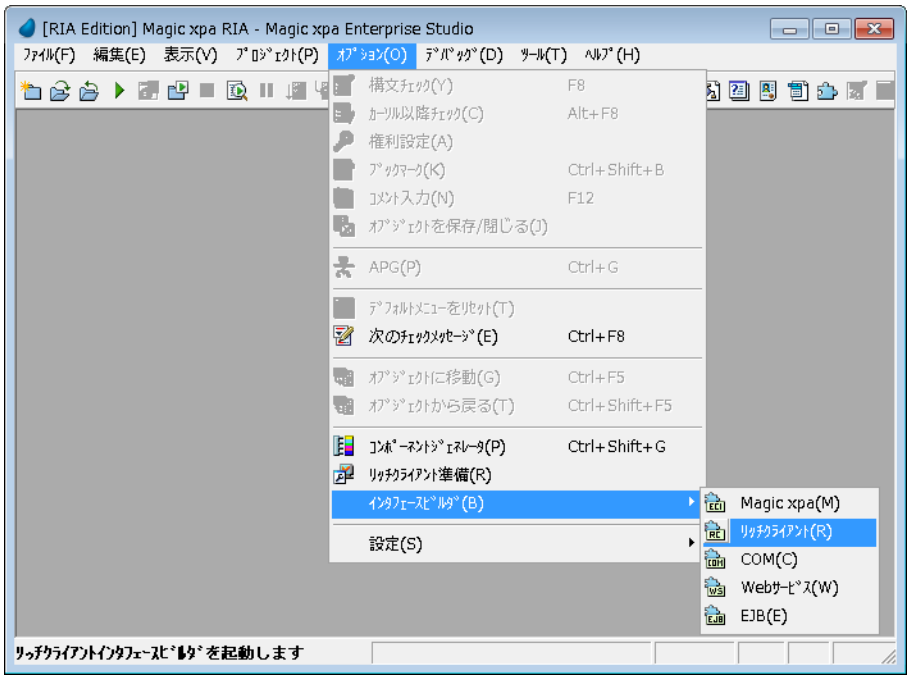
ここでは、RIA を公開するための作業を学習します。RIA を他の PC から実行するためには、以下の２つのファイルを作成する必要があります。

- 公開用 HTML ファイル …… クライアントでの Click Once の有効性をチェックし、必要であればインストールします。ここには、起動プログラムなどのいくつかのアプリケーション実行環境などの XML 定義が含まれています。
- マニフェストファイル …… サーバパスなどのアプリケーション環境が含まれています。

これらのファイルを作成するには、「リッチクライアントインタフェースビルダ」と呼ばれるウィザード形式のユーティリティを使用します。開発環境でリッチクライアントインタフェースビルダを起動して公開用のファイルを作成していきましょう。

リッチクライアントインタフェースビルダの実行

1. ＜ Magic xpa RIA ＞プロジェクトを開いている状態でプルダウンメニューから［オプション / インタフェース ビルダー / リッチクライアント］を選択します。［ようこそ］ダイアログが表示されます。

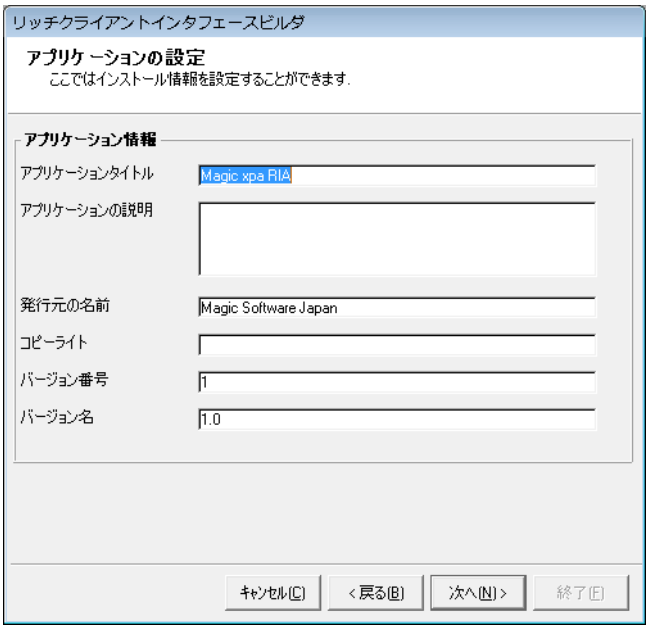


2. ［次へ］をクリックします。［環境設定］ダイアログが表示されます。

3. ［新規］をクリックします。［アプリケーションの設定］ダイアログが表示されます。

   以下は必須項目です。

   - アプリケーションタイトル …… RIA の名前です。デフォルトはプロジェクト名に成りますが変更できます。
   - 発行元の名前 …… 発行元の名前の名前です。
   - バージョン番号 …… バージョン番号を数値で指定します。アップグレード時に比較されます。
   - バージョン名 …… 表示用のバージョン番号を指定します。



4. ［次へ］をクリックします。［サーバ情報］ダイアログが表示されます。

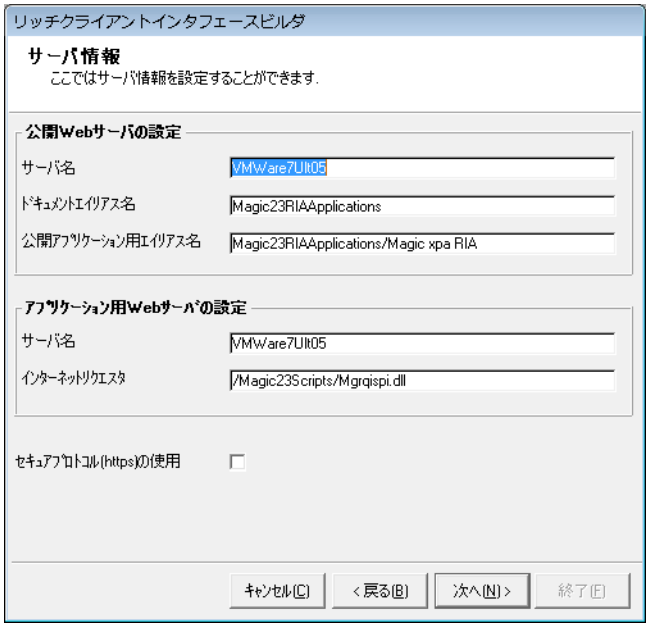
   以下は必須項目です。

   公開用 Web サーバ環境

   - サーバ名 …… HTML ファイルとマニフェストファイルを配置し、エンドユーザに公開するサーバ名または IP アドレス。
   - ドキュメントエイリアス名 …… Web サーバ上のリッチクライアントモジュールフォルダのエイリアス名。
   - 公開アプリケーション用エイリアス名 …… Web サーバ上の RIA アプリケーション用ファイル（HTML ファイルとマニフェストファイル）を格納するフォルダのエイリアス名。

   アプリケーション用 Web サーバ環境

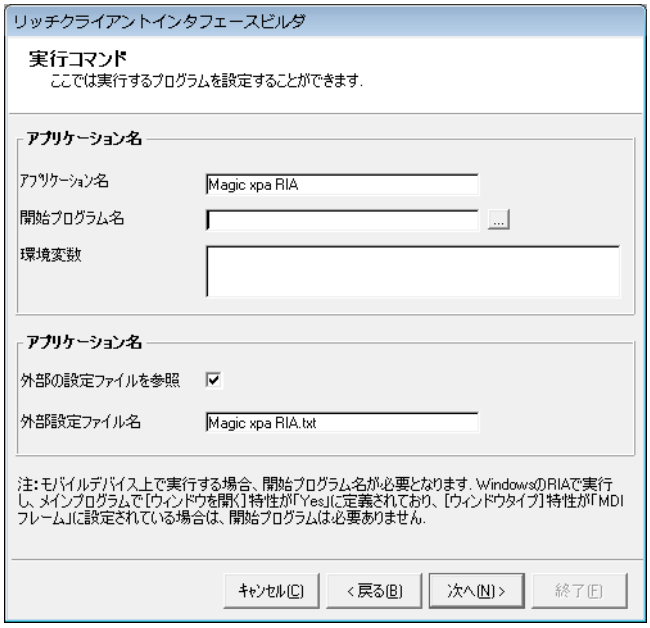
   - サーバ名 …… Magic xpa のアプリケーションファイルが格納されているサーバ名または IP アドレス。
   - インターネットリクエスタ …… 使用するインターネットリクエスタの名前。



5. ［次へ］をクリックします。［実行コマンド］ダイアログが表示されます。

以下は必須項目です。

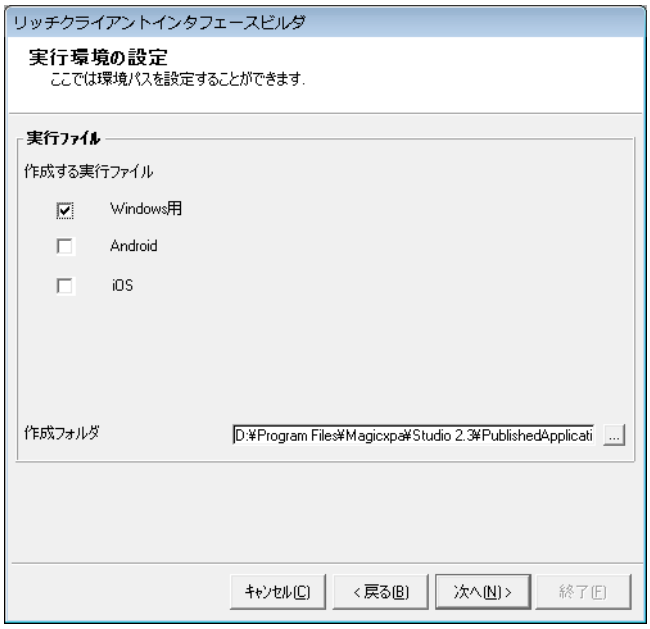
- アプリケーション名 …… 呼び出す Magic アプリケーションの名前。
- 開始プログラム名 …… リッチクライアントアプリケーションで最初に実行するプログラムの公開名。MDI 表示を行う場合は、指定する必要はありません。



6. ［次へ］をクリックします。［実行環境の設定］ダイアログが表示されます。

以下は必須項目です。

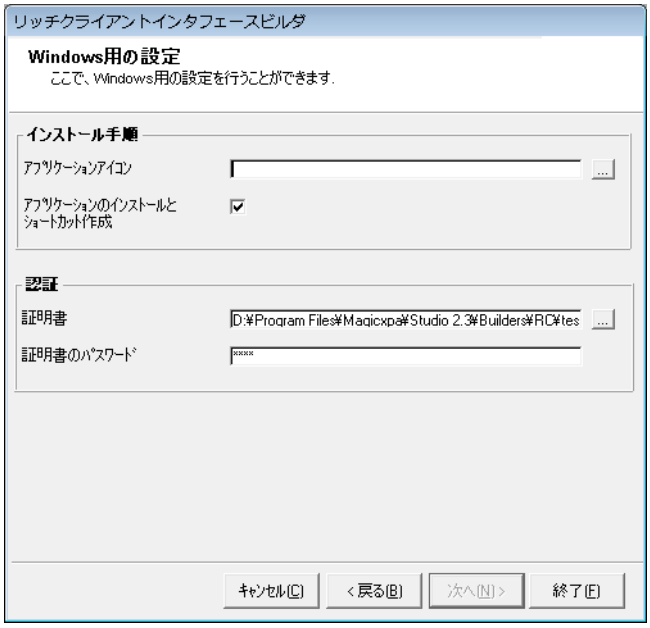
- 実行ファイル …… 公開ファイルをに対応するデバイスを選択します。ここでは、Windows デスクトップのみ選択します。
- 実行ファイルのフォルダ …… 設定ファイルが作成されるフォルダ



7. ［次へ］をクリックします。［Windows 用の設定］ダイアログが表示されます。

   以下は必須項目です。

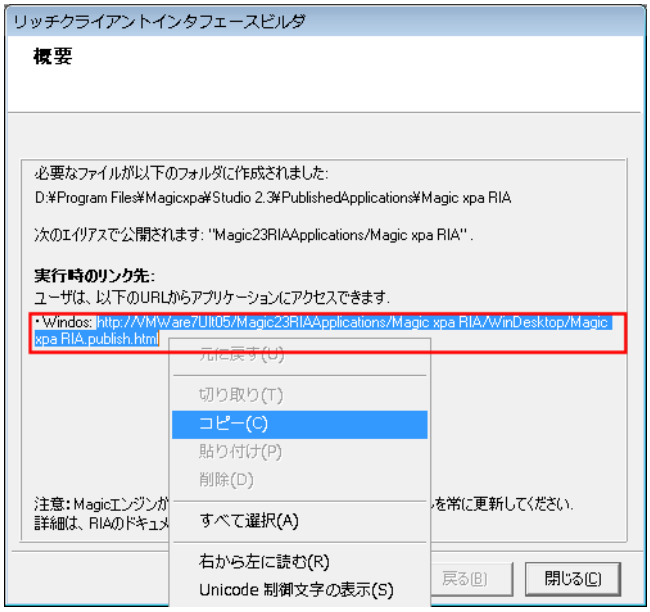
   - 証明書 …… 署名されたマニフェストファイルのファイル名とパスを指定します。pfx 形式の証明書を使用してファイルに署名する必要があります。デフォルトは、Magic xpa によって提供される評価用の証明書ですが、実運用の場合は推奨できません。
   - 証明書のパスワード …… 証明書ファイルを使用するためのパスワード。



8. ［終了］をクリックすると、ファイルの作成処理が実行され、正常に終了すると［概要］ダイアログが表示されます。

   ここには、作成されたファイルの保存先と、公開用に使用するエイリアス名が表示されます。

9. ［実行時のリンク先］に表示される URL を記録しておいてください。マウスで範囲指定し、コンテキストメニューから［コピー］を選択することでクリップボードにコピーすることができます。この URL は後で使用します。
10. ［閉じる］をクリックすると、ウイザードは終了します。



５．クライアント側でのインストール処理

今までの作業で、アプリケーション公開用のファイルが作成されました。次は、このファイルを使用してクライアントからアプリケーションを実行させてみましょう。

Magic xpa が実行している PC でも構いませんし、この PC に TCP/IP で接続できる別の Windows PC があればそこからでも構いません。

まず、プロジェクトを実行し、その後、クライアント側で公開用 HTML ファイルを開きます。

プロジェクトの実行

1. Magic Sutdio を起動し、< Magic xpa RIA >プロジェクトを開きます。
2. ［デバッグ］メニューから、［プロジェクトの実行］（Ctrl+F7）を選択します。プロジェクトは実行状態になります。

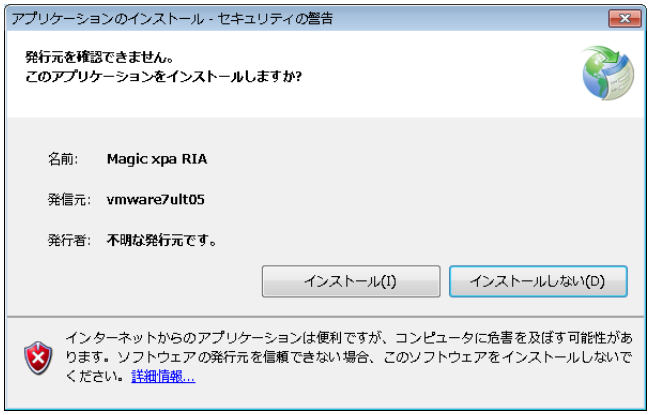
クライアントから実行

1. クライアント PC で IE（Internet Explore）を起動し、リッチクライアントインタフェースビルダの［概要］ダイアログに表示された URL を開きます。前述の例では、以下のようになります。

   http://VMWIN7PRO02/Magic23RIAApplications/Magic%20%xpa%20%RIA/WinDesktop/Magic%20xpa%20RIA.publish.html

2. 以下のようなポータル画面が表示されます。

［起動］をクリックすると、以下のような証明書の確認ダイアログが表示されます。左側は評価用の証明書を使用した場合で、右側は正式な証明書を使用した場合の例です。

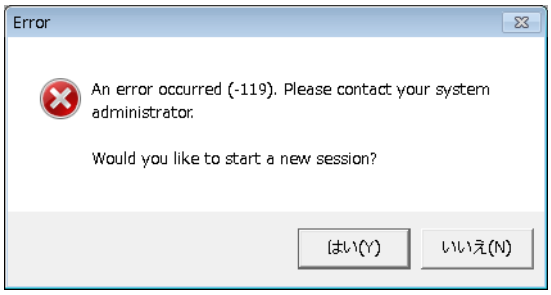


評価用の証明書を使用した場合

正式な証明書を使用した場合の例

［インストール］をクリックするとクライアントモジュールのインストール処理が実行され、RIA が起動します。

しかし、以下のようなエラーダイアログが表示されはずです。

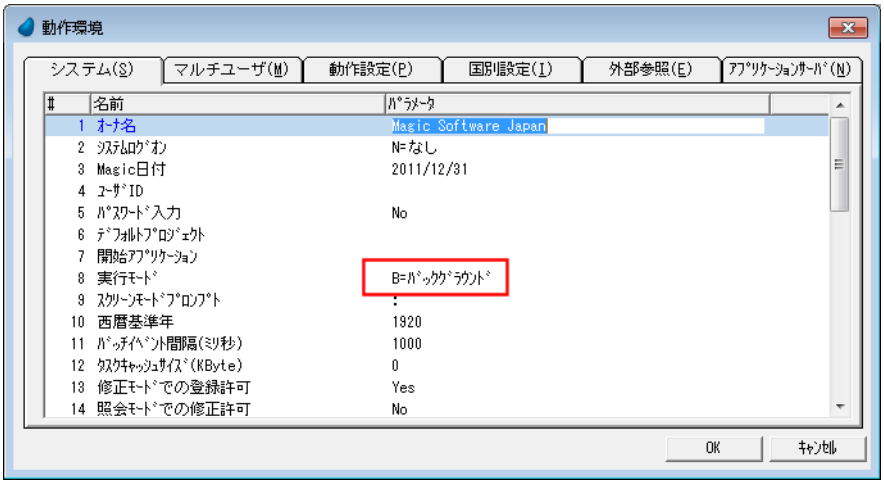


これは、Magic xpa の実行モードが原因です。修正しましょう。

Magic xpa の実行モードを変更する

1. Magic xpa の［デバッグ］メニューから、［停止］を選択します。プロジェクトは停止します。
2. ［オプション］メニューから、［設定／動作環境］を選択します。［動作環境］ダイアログが表示されます。
3. ［実行モード］を「B= バックグラウンド」に変更します。

RIA は、バックグラウンドモードで実行させる必要があります。



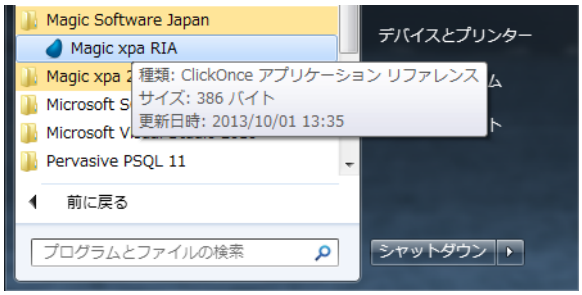
4. ［OK］をクリックしてダイアログを閉じます。
5. 再度、［デバッグ］メニューから、［プロジェクトの実行］（Ctrl+F7）を選択し、プロジェクトを実行状態にします。

クライアントで再度アプリケーションを起動しましょう。今度は、別の方法で起動できます。

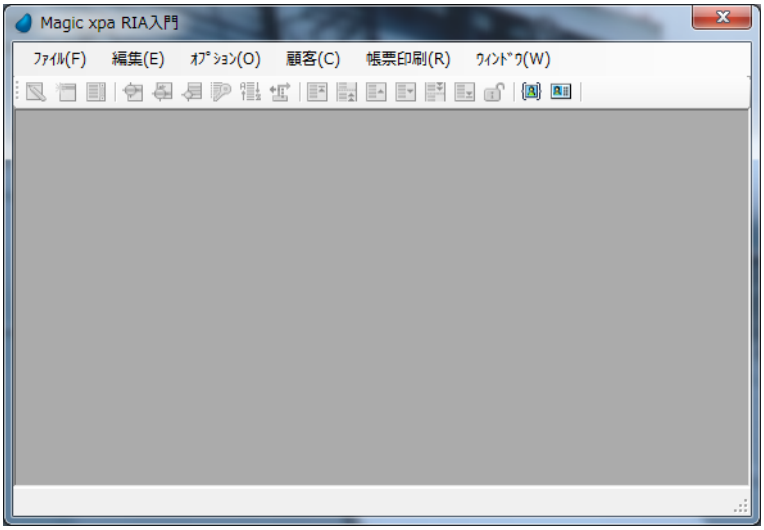
クライアント側のインストール処理を行ったことで、クライアント側の［スタート］メニューに起動用のメニューが登録されました。今後は、これを選択することで実行することができるようになります。

クライアントから実行

1. Windows のスタートメニューを開きます。［すべてのプログラム］を選択します。
2. 「Magic Software Japan」というメニューをクリックすると「Magic xpa RIA」が表示されます。このメニューをクリックするとアプリケーションが起動されます。



「Magic Software Japan」は、リッチクライアント インタフェースビルダで［発行元の名前］に設定した名前です。「Magic xpa RIA」は、［アプリケーション タイトル］に設定した名前です。



今度は、正しく起動されるはずです。起動できない場合は、以下のような原因が考えられます。

- クライアントがサーバに接続できないネットワーク上の問題がある。
- クライアントに .Net Framework がインストールされていない。
- サーバ側でアプリケーションが実行されていない。

リファレンスヘルプの「トラブルシューティング ／リッチクライアント の共通エラー」にエラーメッセージと考えられる原因が記載されています。

６．要約

本章では、サーバ側で実行している RIA をクライアント側でアクセスするための手順について説明しました。

- Web サーバ（IIS）の設定内容
- Magic xpa 側の動作環境の設定内容
- アプリケーションの公開準備
- クライアント側の実行モジュールのインストール

実際の運用環境では、サーバ側に Magic xpa RIA Server をインストールし、開発したアプリケーションを運用環境で 実行させるための環境設定が必要になります。

RIA の開発作業に本格的に取り組むことになった場合は、そのような知識も必要になります。ぜひ、チャレンジしてみてください。

Getting Started for RIA Magic xpa

Copyright 2013 Magic Software Enterprises Ltd.and Magic Software Japan K.K.　All rights reserved.

第一版　2013 年 10 月 11 日
発行　〒 151-0053　　東京都渋谷区代々木三丁目二十五番地三号
あいおいニッセイ同和損保新宿ビル 14 階

Magic Software Japan K.K.