Magic eDeveloper
入門

Integrating Tomorrow's Solutions Today

マジックソフトウェア・ジャパン株式会社

目次

第３章　顧客管理プログラムの作成

第４章　顧客管理システムの機能追加

第５章　バッチ処理プログラムの作成

第１章　　MAGIC eDeveloper の
世界へようこそ

この章では、MAGIC eDeveloper　をはじめてお使いになる方のために、その特長を紹介します。また、MAGIC の起動／終了の方法について説明します。

１．　MAGIC eDeveloper　はこんなソフト
２．　MAGIC eDeveloper　の起動と終了について

１.Magic はこんなソフト

Magicは、クライアント及びサーバーアプリケーションのシステムを開発・実行することのできるツールです。
Magicは、コーディング処理を自動化されたテーブル駆動によるプログラミング技術に置き換え（テーブルに対話形式でコマンドとパラメータを埋め込んでいきます。）、アプリケーションの開発生産性を劇的に向上させる事ができます。これまで3GL(COBOLやC)、あるいは4GLの開発ツールは、ユーザ・インターフェイスの処理などでコーディング排除や部品化に積極的に取り組んできましたが、データアクセスを中心とした核心的な部分については、大量のスクリプト記述が必要となるケースが少なくありません。また、4GLといわれるものの中にもバッチ処理を作成する場合にはかなりの比重で言語記述を必要とするものも多いかと思います。そして何よりも保守工数の増大が問題となっていまが、MagicではMagicの全てのアプリケーションの全ての機能をスクリプト等のステートメントを記述することなく構築することができます。
また　開発者は作成したプログラムをインタプリタのように、コンパイルやリンクなどの中間処理なしに直ちに実行することができます。
これは強力なMagicエンジンにより可能となっています。
この Magic エンジンは、大きく分けて以下の３つの機能から構成されています。

1-A.　３つの機能

①アプリケーション構築機能

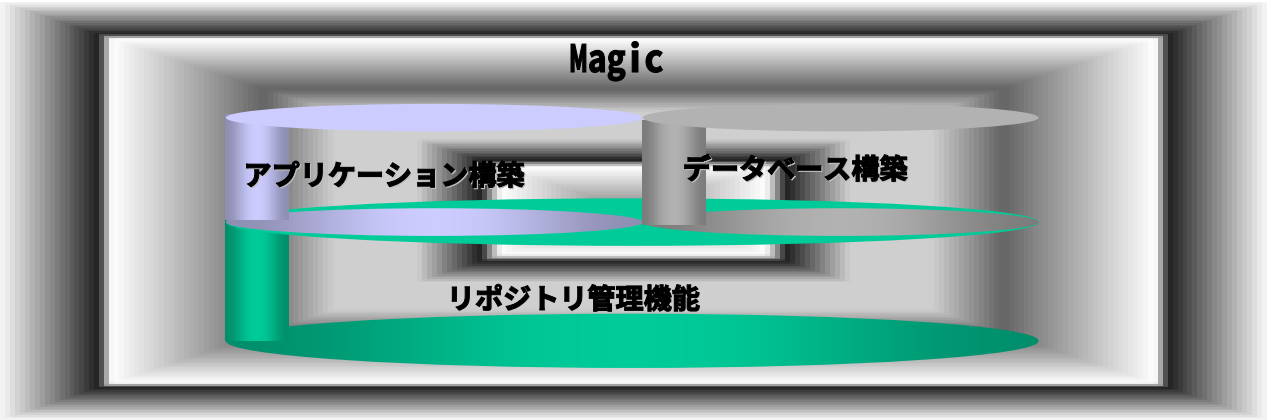
クライアントアプリケーション及びサーバアプリケーションのシステムを構築します

②データベース構築機能

SQLなどを使わずにリレーショナル・データベースを定義・操作・制御でき、必要に応じてMagicが、SQL文を発行します。

③リポジトリ管理機能

CASE ツール的なシステム管理機能を持っています。



1-B.開発システムと実行システム

Magic には、アプリケーション開発用ツール「開発システム」と実行用ツール「実行システム」とが存在します。

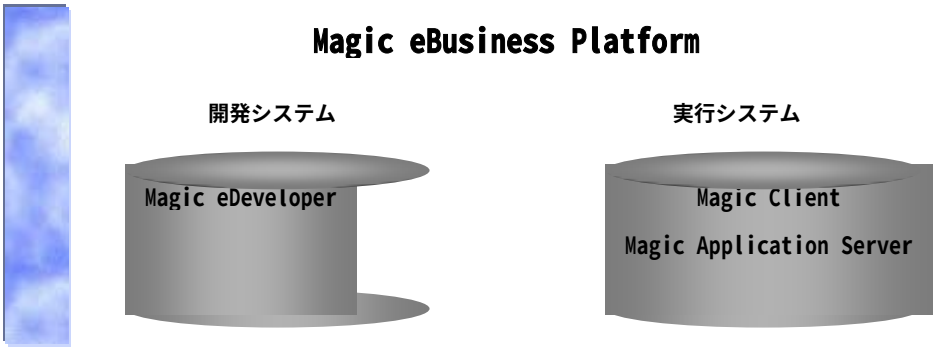
開発システム

開発システムとしてMagic eDeveloper　があります。
開発システムは、従来のコーディングによるプログラミングを排除し、開発資源の集中管理化・効率化を実現した非常に生産性の高い開発用ツールです。開発、運用、保守時に必要となるさまざまなインタフェースやデータアクセス機能を提供します。

実行システム

実行システムとしてMagic Client ・Magic Application Server　の2種類があります。
Magic Client は、開発システム上で作成したアプリケーションをクライアントで実行することができます。 この場合クライアントのCPUで動作します。

Magic Application Server は、アプリケーション・パーティショニングやブラウザを使用したインターネット・ イントラネットシステムが実現できます。 この場合、アプリケーションサーバのCPUで動作します



1-C.　開発者からみた特徴

・従来の開発言語を使ったシステム開発手法に比べ、開発およびメンテナンスの工数を劇的（1/3～1/10）に短縮します。アメリカのDB／Expoでは、毎年名だたる多くの開発ツールがその開発効率を競い合うRealWareコンテストが行われていますが、MagicはRealWare賞をはじめとします各章を受賞している常連でもあります。

・ファイルI/O、並び替え、ロック処理、トランザクション処理、画面制御のみならず、グループ処理を含む処理フローにいたるまでMagicアプリケーションエンジンが自動的に処理するため、実際のプログラミングは、わずか14個のコマンドと各種パラメータを設定するだけで完成します。

・データベースへのアクセスは、Magicエンジンとは別のデータベースゲートウェイモジュールを介して行われます。このモジュールは、各DBMS毎に用意されています。
またモジュールは個々のデータベース専用のゲートウェイになります。
これにより異なるDBMSで個々に管理されているテーブルを、あたかもひとつのDBMSのようにアクセスすることができますので、データ容量の増大や他部署のDBMSとの連携など、必要に応じて簡単にDBMSを切り換えることもできます。

・ユーザーの要求を的確に反映しやすいスパイラル型開発手法*1を実現。従来のウォーターフォール型開発手法*2に比べ顧客満足度が大幅に向上します。

1-D.　エンドユーザからみた特徴

・開発者からみた特長と同じく、ユーザーの要求を的確に反映しやすいスパイラル型開発手法を実現しますので、従来のウォーターフォール型開発手法に比べ顧客満足度が大幅に向上します。

・企業の成長に合わせて、すでに稼動しているシステムでもそのままクライアント、サーバ、DBMSなどを自由に変更することができます。勿論、プログラムやデータはそのまま活用する事ができます。

・インターネットのWebサーバで情報公開するデータベースをMagicで構築することができます。
これにより、基幹業務系と情報系で二重管理することが多いデータを一元管理することができます。

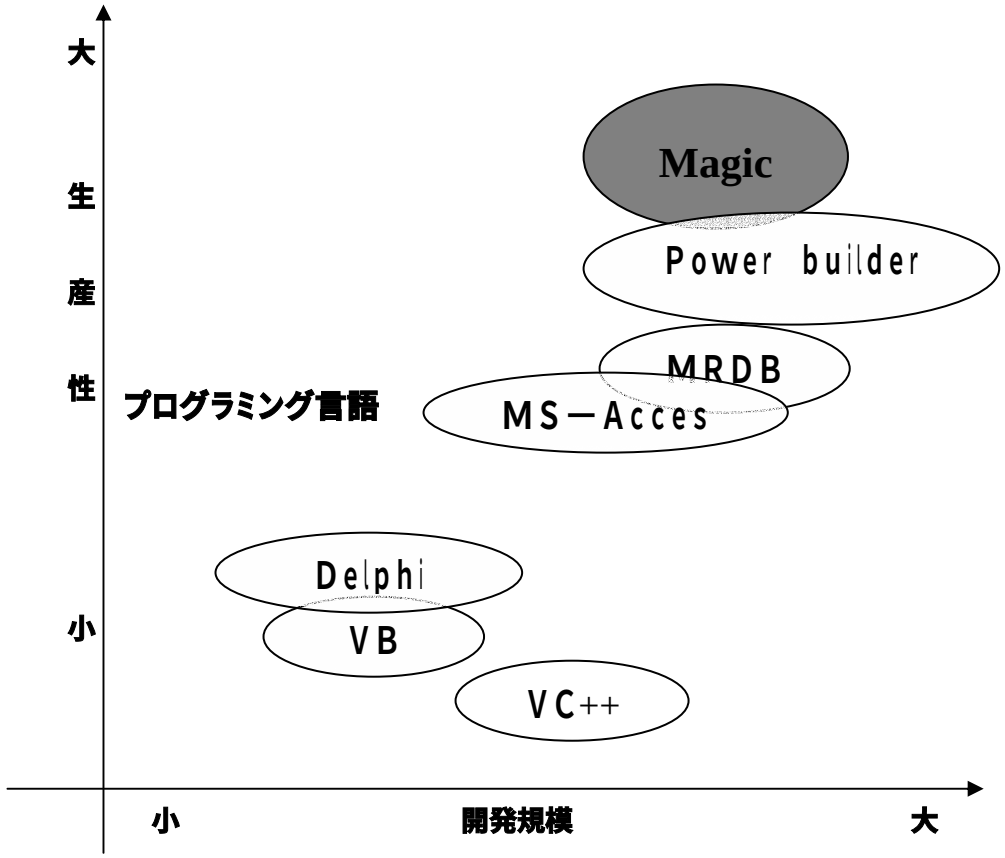
・ドラック＆ドロップによるアプリケーション・ロジックのパーティショニングを可能にします。物理的な環境に依存せず、複数のクライアントと複数の Magic アプリケーション・サーバに分散可能な論理アプリケーションを構築できます

＊1　要求仕様抽出、分析、設計、開発のプロセスを渦巻きのように何度もフィードバックして進める開発手法

＊2　要求仕様抽出から、分析、設計、開発のプロセスを水が流れるように後戻りすることなく進める開発手法

3）Magicの位置付け

位置付けはおおよそ次のようになる。



4）動作環境

ハードウェア	コンピューター	：日本語Microsoft Windows 98、Windows NT 4.0 Windows 2000 Professional、Windows XP Professionalが動作する機種
	CPU	：PentiumIII 500MHｚ以上を推奨
	ディスプレイ	：解像度1024X768以上を推奨
	必要メモリ	：256ＭＢ以上を推奨
	ＨＤＤ空き容量	：71MB（Magic eDeveloper 43MB/Pervasive.SQL 2000i Workgroup 28MB）
	プリンタ	：日本語Microsoft Windows 98、Windows NT 4.0 Windows 2000 Professional、Windows XP Professional に対応す るプリンタ
ソフトウェア	クライアントＯＳ	：日本語Microsoft Windows 98、Windows NT 4.0 Windows 2000 Professional、Windows XP Professional
	サーバＯＳ	：日本語Microsoft Windows NT 4.0 ServerまたはWindows 2000 Server
	ＤＢＭＳ	：Pervasive SQL2000/2000i Workstation(Workgroup)、Oracle8、Oracle8i Oracle9i、MS-SQL7、MS-SQL2000、MSDE

５）システム構成例

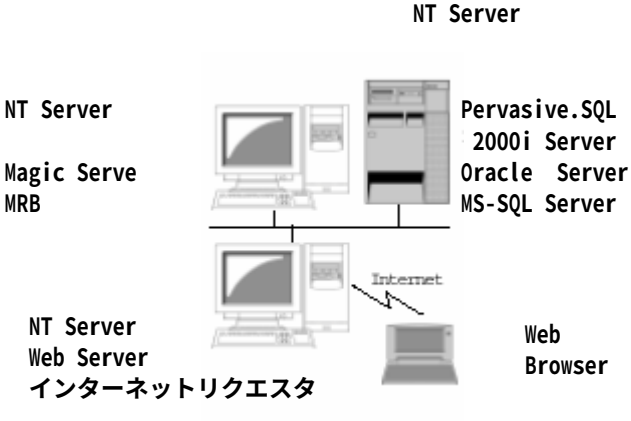
MagicもPervasive.SQL 2000i Workgroup（Magicの標準添付データベースエンジン）もClient PCのCPUパワーを使って動作します。全検索対象データがネットワークを通過するためトラフィックが増大する可能性があります。Client PCにも高パワーなものが求められます。

① Magic Clientは、アプリケーションを実行するとともに同じClient PCにあるPervasive.SQL2000iWorkgroupを介してローカルドライブまたはNt Server上のデータファイルへアクセスします。

Client/Serverとして動作します。(Magic Clientと各DBMS Serverとで負荷を分担)

① Magic Clientは、アプリケーションを実行するとともに同じClient PCにあるPervasive.SQL2000i リクエスタ、SQL*NET等を介してPervasive.SQL2000i Server、Oracle Serverにアクセスします。

② Pervasive.SQL2000i ServerとOracle Server又はMS-SQL Serverはそれぞれのデータファイルにアクセスします。



Ｗｅｂブラウザからアクセスします。

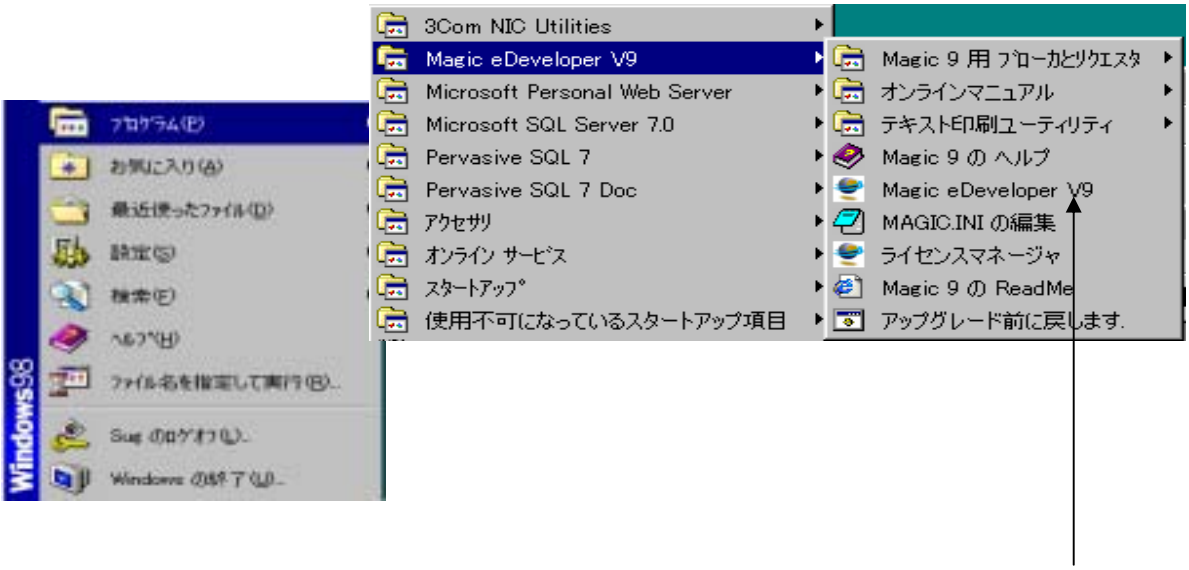
① Ｗｅｂブラウザは、インターネットを介してＷｅｂサーバへアクセスします。
② Ｗｅｂサーバは、MRBとインターネットリクエスタを介してMagic Serverエンジンにアクセスします。
③ Magic Serverエンジンは、データベースエンジンから何らかの結果をインターネットリクエスタに返します。
④ httpプロトコルによりＷｅｂブラウザに表示されます。

＊1　MRBは、クライアントから来るリクエストをキューにため、利用可能なMagic実行エンジンに割り当てます。

＊2 インターネットリクエスタは、NSAPIとISAPIといったウェブ・サーバ固有のAPIをサポートしており、これらのサーバ特有の機能を利用することができます。サーバに依存しないCGIを使用できるウェブ・サーバでアプリケーションを呼び出すことができます

２．Magic eDeveloper の起動と終了について

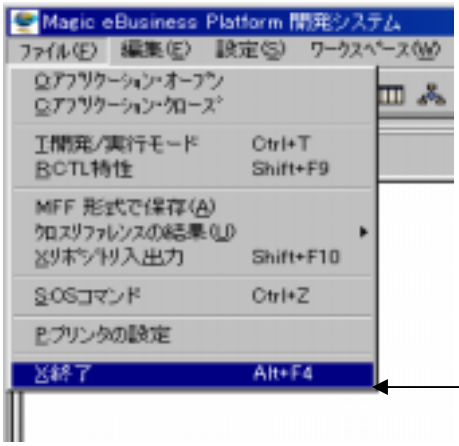
１）起動方法



ここをクリックすると起動します。

２）起動画面

３）終了方法



ここをクリックします。

第２章　　Magicの
アプリケーション構造

この章では、Magicの生産性、メンテナンス性の鍵を握るリポジトリとMagicアプリケーションエンジンについて説明します。

１．リポジトリ
２．Magic のアプリケーションエンジン

１.リポジトリ

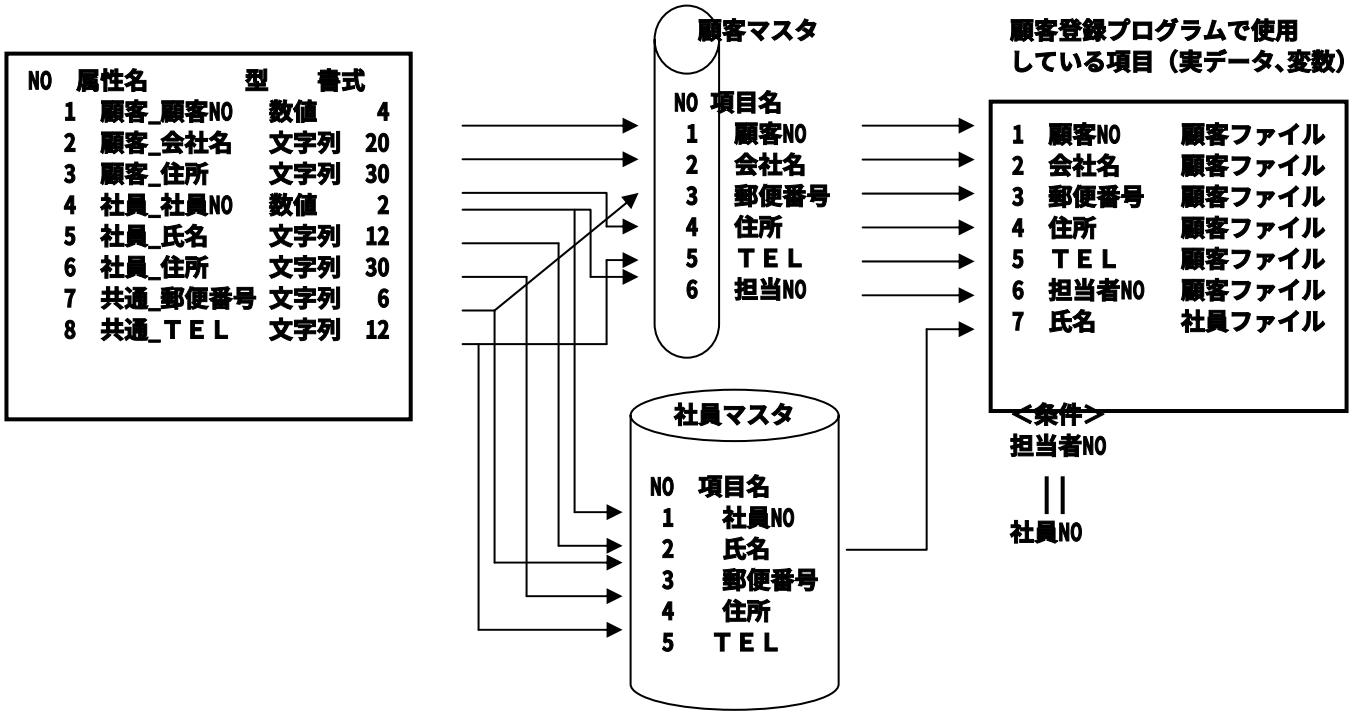
Magic の生産性の第一の鍵を握っているのがリポジトリです。リポジトリとはシステム開発に関するデータベースのデータ項目や、プログラムなどの開発資源を一元管理する保管庫のことです。Magic のリポジトリは、いくつかの構成に分かれていますが、もっとも重要なものが、モデルリポジトリ、テーブルリポジトリ、プログラムリポジトリです*1。また、プログラムで使用する項目（実データ、変数）及び、ファイル定義で使用するデータ項目は、あらかじめモデルリポジトリに登録しておく必要があります*2。

１）　モデルリポジトリ、テーブルリポジトリ、プログラムリポジトリの関係

モデルリポジトリ
テーブルリポジトリ、プログラムリポジトリ等で使用するデータ項目の属性（タイプ）を一元管理する保管庫です。

テーブルリポジトリ
テーブルを一元管理する保管庫です。

プログラムリポジトリ
プログラムを一元管理する保管庫です。



２）メリット

①モデルリポジトリに登録されているデータ項目属性を変更すると（たとえば、桁数を増やすなど）、そのデータ項目を使用しているテーブルとプログラムで使用している項目を瞬時に変更します。テーブルが実データを持っている時は、データコンバート処理も実行できます。

②モデルリポジトリには「型」、「書式」以外に「範囲」や「選択プログラム」などいろいろな属性があります。例えば、モデルリポジトリ4番目の「社員__社員NO」の範囲属性に１－９９を指定すると、プログラムに範囲チェック処理を組み込まなくても、範囲外の値が入力されるとエラーメッセージを表示し再入力を促します。同様に「選択プログラム」属性に社員一覧プログラムを指定すると、プログラムにコールプログラムを組み込まなくても、担当者 NO と社員 NO の項目で社員一覧プログラムを起動させることができるようになります。

③このようにモデルリポジトリを利用することで、ＳＥやプログラマの工数を大幅に削減するだけでなく、アプリケーションの品質も大幅に向上します。

---

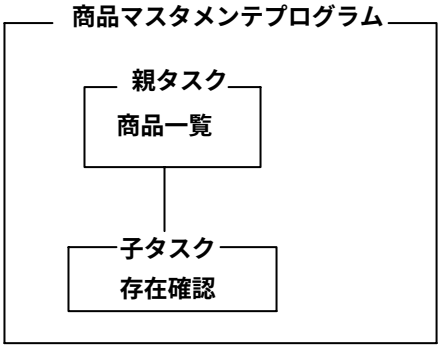
*1　以前はリポジトリのことを辞書と言っていました（例、プログラム辞書）

*2　モデルリポジトリを利用しない方法もあるが、生産性・メンテナンス性を下げるのでお勧めできません。

2.Magic アプリケーションエンジン

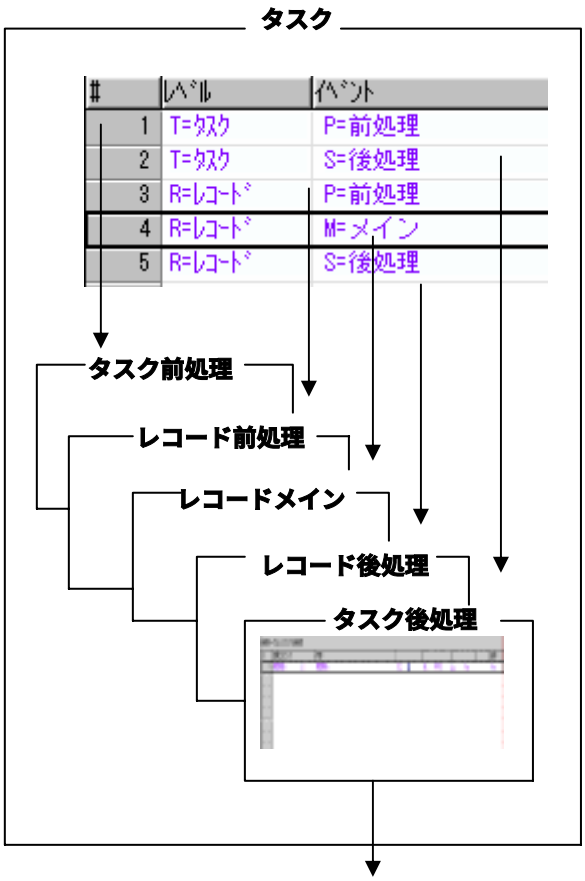
Magicの生産性の第二の鍵はMagicアプリケーションエンジン（以下Magicエンジン）です。
ファイルＩ／Ｏ、並び替え、排他制御、トランザクション処理、画面制御のみならず、グループ処理を含む処理フローにいたるまで、プログラムで頻繁に使用する処理は、Magicエンジンが自動的に処理します。これらを理解するためにはじめに Magic で作成したプログラムの構造について説明いたします。

1）プログラム構造

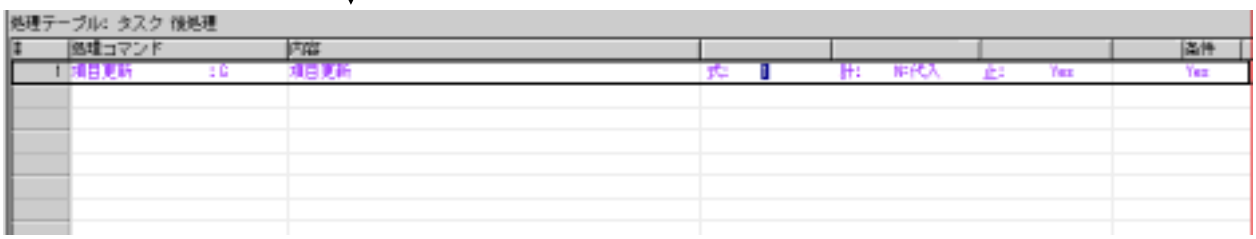


Magicで作ったプログラムはタスクを組み合わせて出来ています。左図は一般的な商品マスタメンテプログラムの例で、商品を一覧表示するタスクと在庫を確認するタスクから出来ています。簡単なプログラムでは1プログラム＝1タスクになります。ただし、ここでのタスクはＯＳ等で使われるタスクとは意味合いが異なり、開発言語の**サブルーチン**や**プロシージャ**のようなものです。

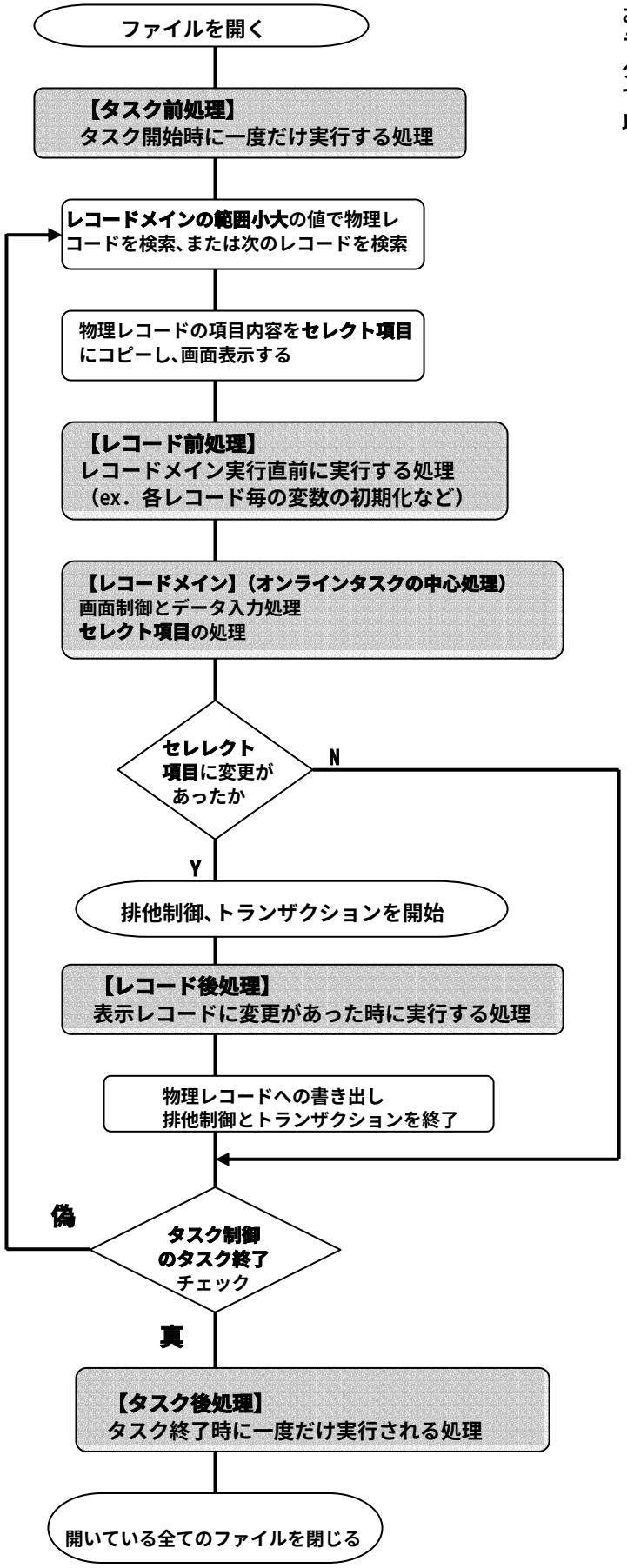
2）タスクの構造



タスクには役割と実行されるタイミング (処理フロー)があらかじめ定められている５種類の処理テーブルがセットになって組み込まれ、それぞれにコマンドを記述することでタスクを作り上げていきます。
実際の記述作業は１４種類のコマンドを所定の場所にはめ込んで行くだけです。　項目名ですらテーブルリポジトリやモデルリポジトリに登録されている項目番号を入力するだけです。
従ってエディタを使った従来のコーディング作業と異なり、存在しないコマンドや項目名は入力すらできません。
これで初歩的な入力ミスは大幅に削除されるでしょう。
またタスクは処理目的に応じてオンラインタスクとバッチタスクの２つに分かれ、それにより処理フローが変化します。
これらについては次ページ以降で説明します。



3）オンライン*1タスクの基本処理フロー



これから学ばれる方にMagic用語を羅列したフロー図をお見せして申し訳ありませんが、内容は一般的なプログラム（マスタメンテなど）と同じです。ただ、実際のプログラムを作成してから見ていただいた方が分かり易いので、最初は読み飛ばしていただいても結構です。
以下に注意点と特徴を述べます。

① 開発言語では開発者がフローを自由自在に変えることができますが、Magic ではあらかじめ許可されている部分*2を除いて変更することは出来ません。理由は、構造の標準化とバグの低減にあります。

②グレーの網掛部分は全て**処理テーブル**（プログラムを記述する領域）です。開発者はここに14種類の処理コマンドを記述してプログラムを作成していきます。中でも**【レコードメイン】**はオンラインタスクの中心となる対話処理を制御する部分なので、大半の工数はここを作成することに費やされます。

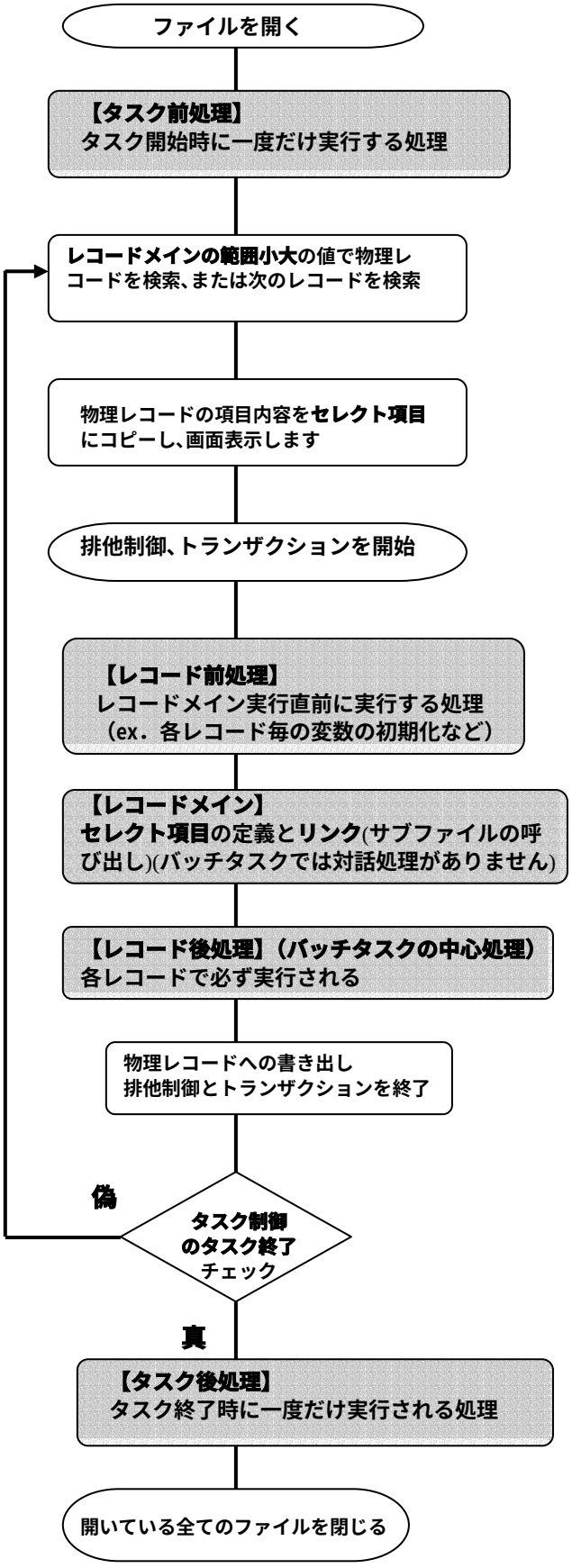
③白い枠の部分はMagicエンジンが自動的に処理してくれる部分です。開発者は必要に応じて簡単なパラメータ（例えば、排他制御やトランザクション処理の有無やそれを開始するタイミングなど）を設定するだけで、複雑な処理 はMagicエンジンが一手に引き受けて処理してくれます。今までに言語で開発された経験がある方なら、これがどれほどすごいことかお分かりになるかと思います。

---

*1　データが発生するたびに逐次処理を行うこと。通常、画面表示とキーボード入力を使って対話的に行われます。
マスタメンテプログラムなどが代表的。

*2　終了条件のチェックのタイミングや排他制御、トランザクションの開始のタイミングなど

4）バッチ*1タスクの基本処理フロー



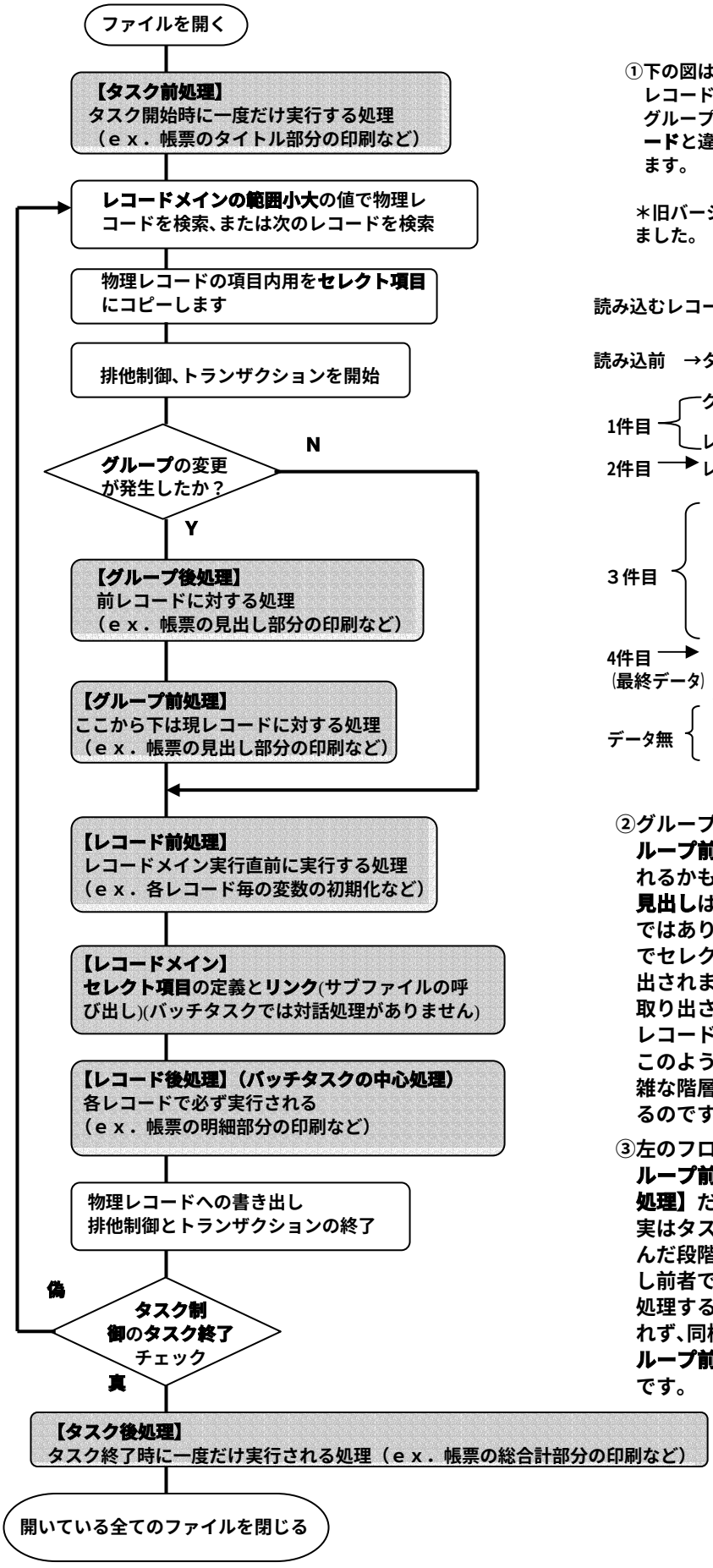
①バッチタスクは対話処理がないタスクです。よってオンラインタスクと異なり**【レコードメイン】**でデータを変更することはできません。変更は**【レコード後処理】**で行われます。従って、**【レコードメイン】**では物理レコードの項目内容を取り出すための**セレクト項目**と**リンク**コマンドのみを記述することになります（それ以外は記述することすらできません）。

②オンラインタスクの**【レコード後処理】**は**【レコードメイン】**でデータの変更があった時実行されますが、バッチタスクでは各レコードで必ず実行されます。この**【レコード後処理】**こそが更新処理の中心となる部分なので、大半の工数はここを作成することに費やされます。

③排他制御とトランザクション処理の開始位置は**【レコード前処理】**の直前で行われます。

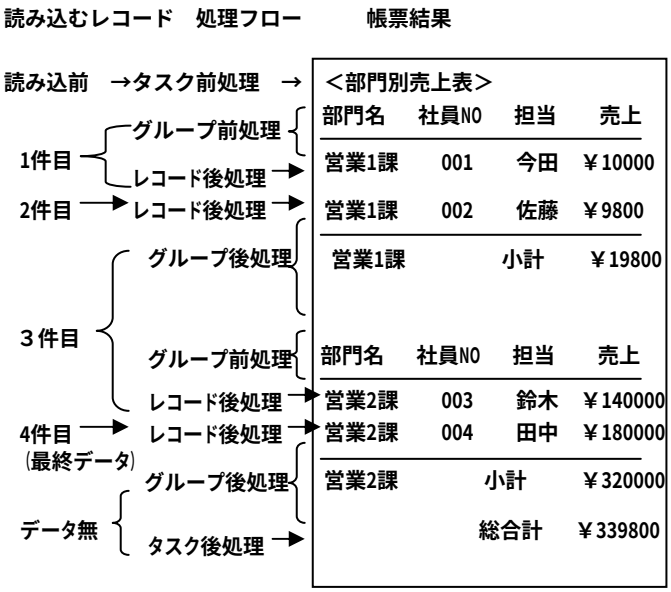
*1 **ある程度データがたまった段階で、一括でデータ処理を行うこと。月次集計や帳票印刷処理などが代表的。**

5）グループ処理*1を伴うバッチタスクの基本フロー



①下の図は、グループを利用した帳票と各**処理テーブル**およびレコードの読み込み状態との関係図です。
グループ項目に**部門名**が指定されており、この項目が**前レコード**と違う値を読み込むと*2、割り込んで**小計**と**見出**を印刷します。

＊旧バージョンではグループ処理をブレイク処理と呼んでいました。



②グループの変更が発生すると**【グループ後処理】→【グループ前処理】**の順で実行されます。順番が逆と感じられるかも知れませんが**小計**は**前レコード**までの事後処理、**見出し**は**現レコード**以降に対する事前処理なので間違いではありません。それだけではなく**【グループ後処理】**でセレクト項目の値を取り出すと前レコードの値が取り出されます。**【グループ前処理】**では現レコードの値が取り出されます*3。これを利用して小計行に部署名（前レコードの値）を印刷しています。
このような仕組みがあらかじめ組み込まれているので、複雑な階層グループですら標準機能で実現することが出来るのです。

③左のフローと異なり帳票例では、タスク起動直後に**【グループ前処理】**だけが、タスク終了直前に**【グループ後処理】**だけが実行されています。何故でしょうか？
実はタスクの起動直後と終了直前でもレコードを読み込んだ段階でグループの変更は発生しているのです。しかし前者では前レコードが存在しないためそのレコードを処理することを目的とした**【グループ後処理】**は実行されず、同様に後者では現レコードが存在しないため**【グループ前処理】～【レコード後処理】**は実行されないのです。

***1　部門別の売り上げ集計などのようにグループ毎の更新・集計作業を効率的に行う方法。**
***2　このことをグループ又はブレイクが発生したと言います。**
***3　Magicの内部には２本のファイルバッファがあり、そのうちの１つは現レコード、他方は前レコード用として使われています。**

第３章　　顧客管理システムの作成

この章では、実際にMagicを使って簡単な顧客管理システムを作成する手順について説明します。

１．プログラム作成の事前確認
２．アプリケーションの登録
３．アプリケーションを開く
４．モデルリポジトリへの登録
５．テーブルリポジトリへの登録
６．プログラムリポジトリへの登録
７．プログラム内容の確認
８．メニューの登録
９．練習問題

１．　プログラム作成の事前確認

１）顧客管理システムの概要

テーブル　　：顧客マスタテーブル（得意先情報テーブル）

プログラム　：顧客マスタメンテナンスプログラム。顧客マスタデータの登録・更新を行う

【完成画面】

２）プログラム作成の手順

①アプリケーションの登録

アプリケーションのタイトル「入門」とその物理ファイル名を登録します。
顧客マスタと社員マスタの物理データファイルを除いた②～⑤の情報は、全てこの一つのコントロールファイルに一元登録されます。

②モデルリポジトリへの登録

次の③で登録しますテーブルの全項目を登録します。ただし、重複する項目は一度だけ登録します。

③テーブルリポジトリへの登録

顧客マスタと社員マスタのテーブル定義を登録します。

④プログラムリポジトリへの登録

顧客メンテナンスプログラムを登録します（実質的なプログラミング作業）。

⑤開発モードで実行

開発モードの状態のままプログラムを実行し動作確認します。

⑥メニューへの登録

顧客メンテナンスプログラムをポップアップメニューへ登録します。

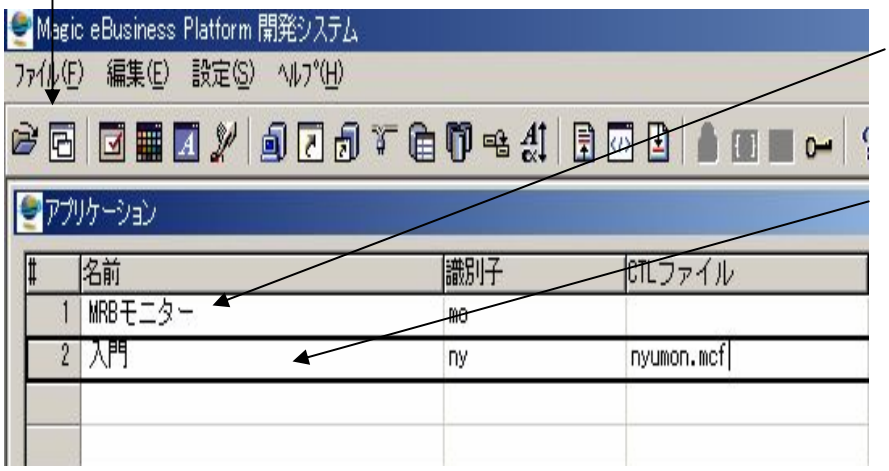
⑦実行モード（エンドユーザの状態）で実行

メニューからプログラムを実行します。

２．　アプリケーションの登録

アプリケーションのタイトルとその物理ファイル（これをコントロールファイル、またはMCFファイルと呼びます）を登録します。物理データファイルを除いた全てのリポジトリ内容は、この一つのコントロールファイルに一元登録されます。この作業は、コントロールファイルを新規に登録するはじめの一回目だけに行います。
一度登録してしまえば、次回からは次の**３．アプリケーションを開く**から実行します。

①アプリケーションをクリックします



②「ＭＲＢモニター」をクリックします。新しい行を挿入する場合、そのひとつ上に移動しておきます。何も登録されていないときは＃の個所にカーソルを持ってきます。「MRBモニター」上でF4を押し一行作成します。

③各項目に次の値を入力します。その他の項目は既定値のままにしておきます。

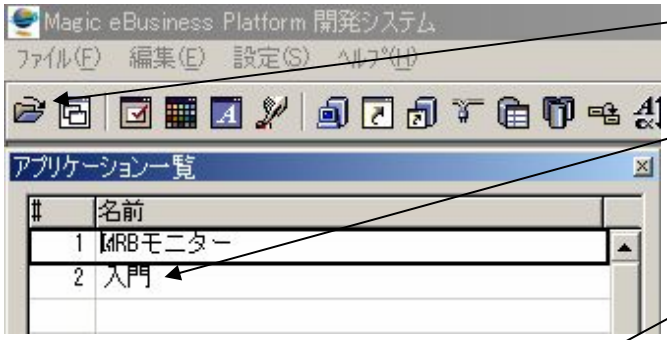
名前　　：入門
識別子　　：ｎｙ
ＣＴＬファイル　：ｎｙｕｍｏｎ．ｍｃｆ

＊識別子は任意の半角英字２文字。
Magic内部の作業ファイル名に使われます。（必須項目）

＊操作を間違ったときは**取消ボタン**を押します。Windowsに慣れてくると反射的に**[CTRL]+[Z]**を押してしまうことがありますが、そうするとDOSプロンプトが起動してしまいます。
これは旧バージョンとの互換性を保つ為の仕様です。
同様に、次項目への移動は**[Enter]**ではなく**[Tab]**になります。
これらキーの操作に関しては付録のキー操作一覧をご覧頂きたい。

④[ＯＫ]ボタンを押します

３．　アプリケーションを開く



①アプリケーション・オープンをクリックします。

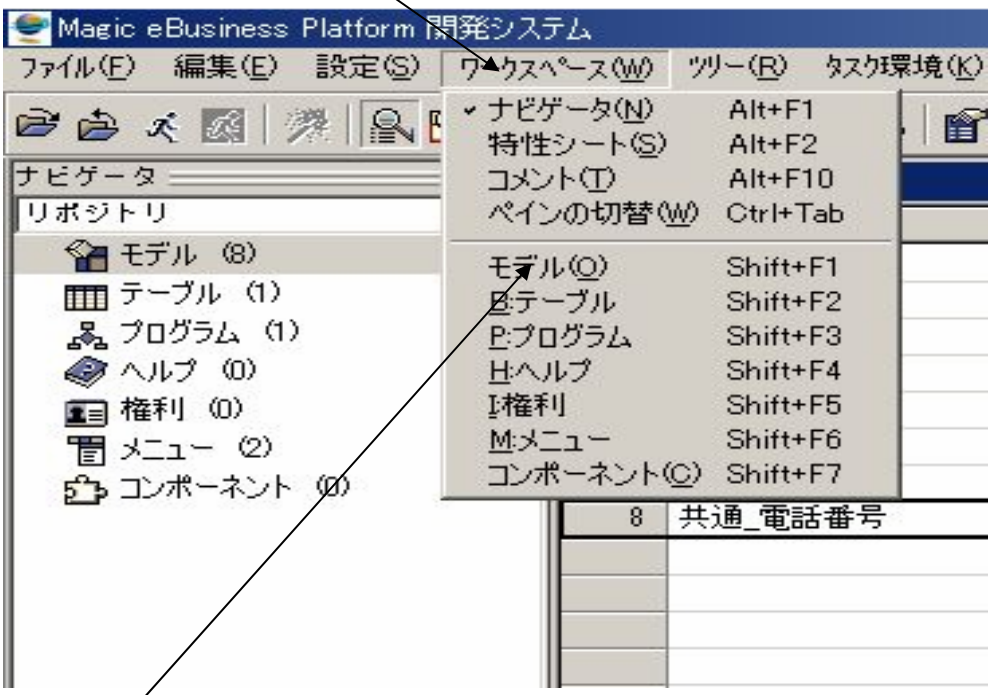
②「入門」をクリックします。

③[選択]ボタンを押します。

④ コントロールファイルがオープンされ、メニューバーとアイコンメニューが開発 モード用のものに切り換わります。

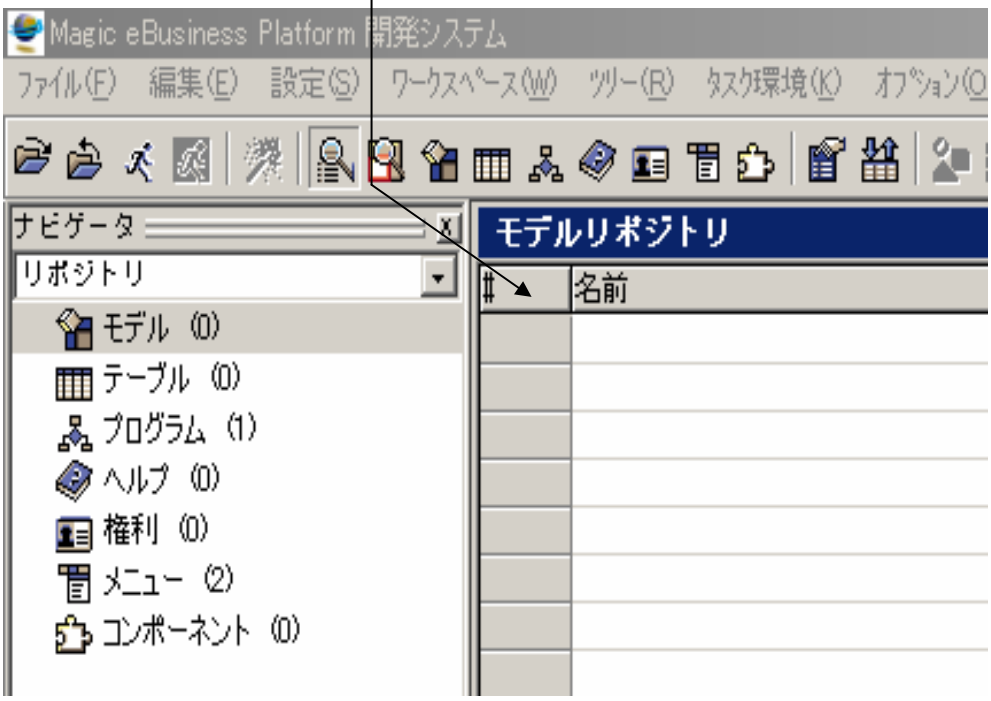
４．　モデルリポジトリへの登録

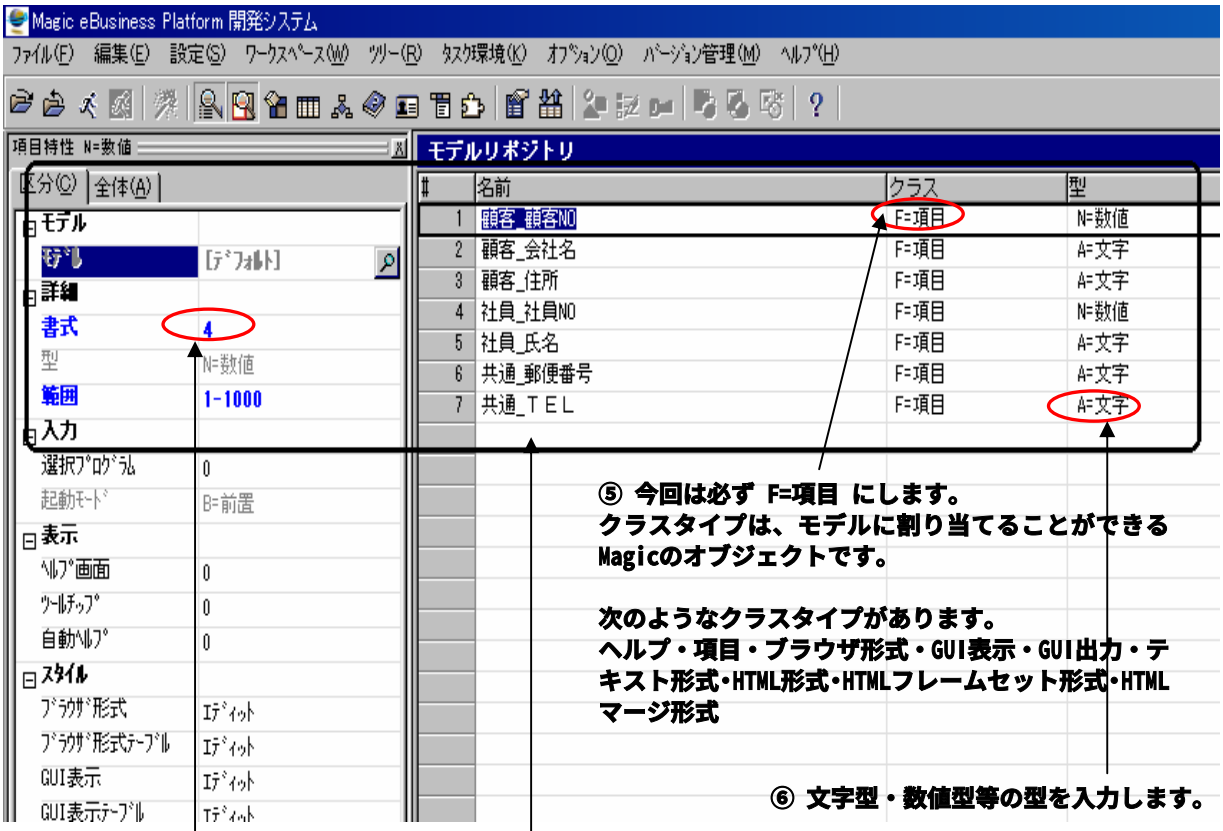
① ワークスペース(W)をクリックします。



② モデルリポジトリ(O)をクリックします。

③ ＃上でＦ４を押し顧客マスタと社員マスタで
使用する7個分の空白行を作成します。
（＃上でＦ４を７回押します。）





1）モデル名称の命名規則について

- Magicではモデルの名称の命名規則を詳細には定められていません。どれを共通で使うモデルにすべきかといった詳細なガイドラインもありません、只これらは開発者が自由に設定することが出来ますので、言わばノウハウが表れるところでもあります。どのモデルがどのテーブル・どのプログラムで使用されているかはクロスリファレンス（CTRL+X）で確認することも出来ますが、複数のテーブルで使われていた場合、それを使うメインテーブルがどれなのかわかりませんので、やはり判別しやすい名前を付けた方がよいでしょう。
  本書では、**主ファイルの略称＋‘_’＋項目名**としました。
  例　顧客_会社名（メインファイル_項目名）

- 共通のモデルは、郵便番号の桁変更（平成10年2月、5桁から7桁に変更）のように、全てのテーブル・プログラムで一括変更されて整合性のとれる項目のみとし、**‘共通_’+項目名**で命名してあります（一つのテーブルでしか使われていなくても、将来共通で使われる可能性があるものは同様に命名してあります）。
  例　共通_郵便番号（共通ファイル_項目名）

- モデルのフォルダを分ける方法もあります。この場合フォルダ欄にフォルダ名を記述します。
  例　共通フォルダ

2）型の種類について

- 項目クラスの型は全部で７種類あります。（A=文字、N=数値、L=論理、D=日付、T=時間、M=メモ、O=BLOB）

3）書式について

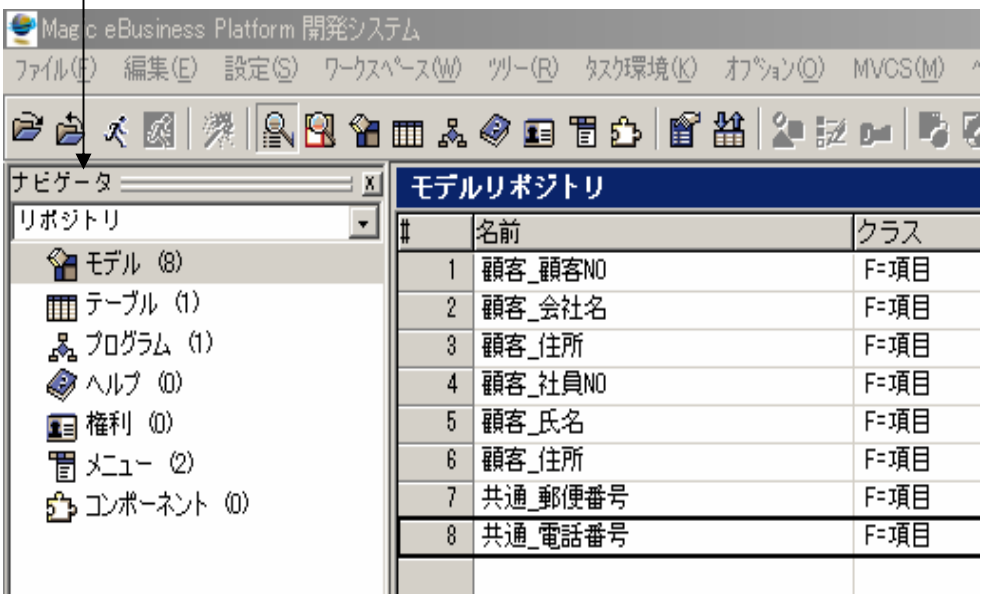
- 桁数以外に、数値項目でもゼロだったら空白表示させる設定なども可能です。

4）範囲について

- あらかじめ実行時に入力される値にモデルリポジトリの段階で範囲を指定しておくことができます。

ナビゲータと特性シートの表示方法

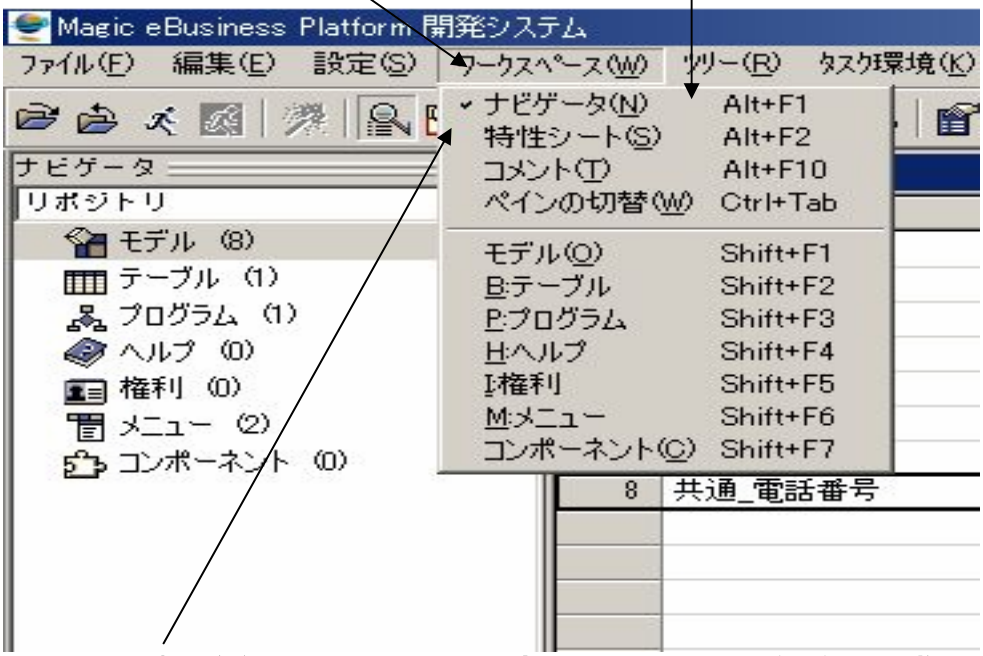
＊ナビゲータ
ナビゲータは、各リポジトリへ素早く移動したり、項目の検索結果を表示したりする時に使用します。
その下のコンボボックスで「リポジトリ・タスク・クロスリファレンス・ブックマーク」の四つが選択できます。



1）ナビゲータの表示

① ワークスペース(W)をクリックします。

②ナビゲータ(N)をクリックします。



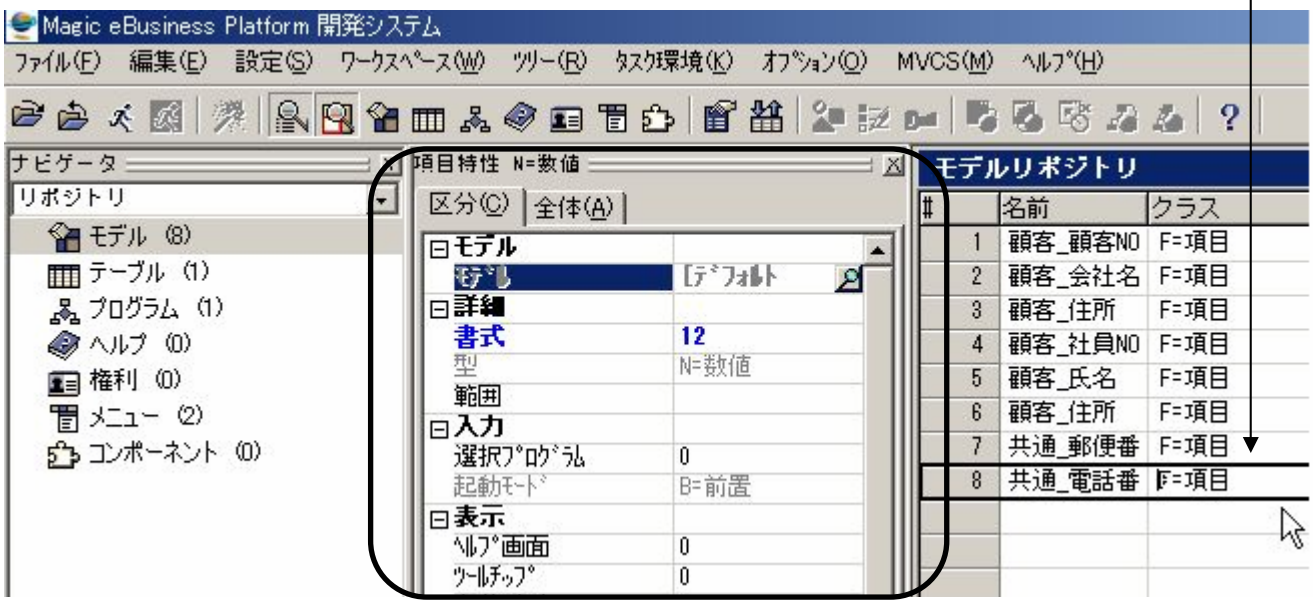
③ナビゲータ(N)をクリックする度にチェック・マークの有り無しを切換える事がでます。
チェックマークの有り無しは、Magic終了時の状態により変動します。

2）特性シートの表示

① ワークスペース(W)をクリックします。

②特性シート(S)をクリックします。

③必要な項目上にカーソルを移動します。

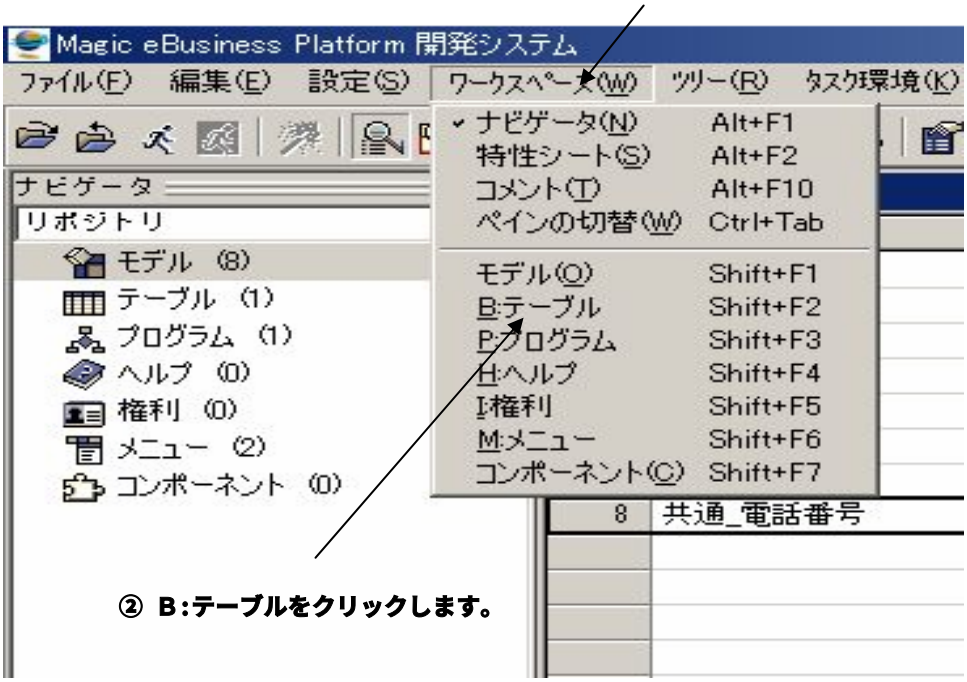


特性シートは、それぞれの項目の型やバイト数等を定義します。

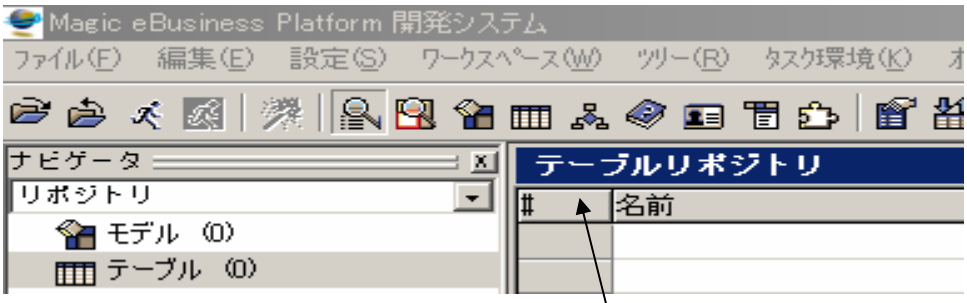
５．　テーブルリポジトリへの登録

１）テーブルの項目定義

① ワークスペース(W)をクリックします。



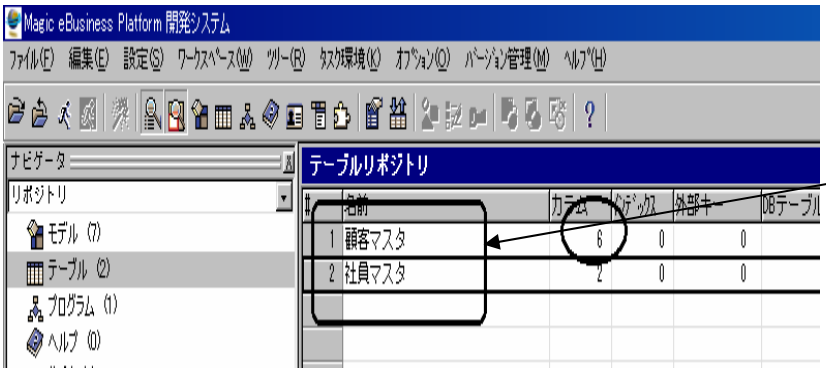
② B:テーブルをクリックします。



③ カーソルを＃の上にもってくます。

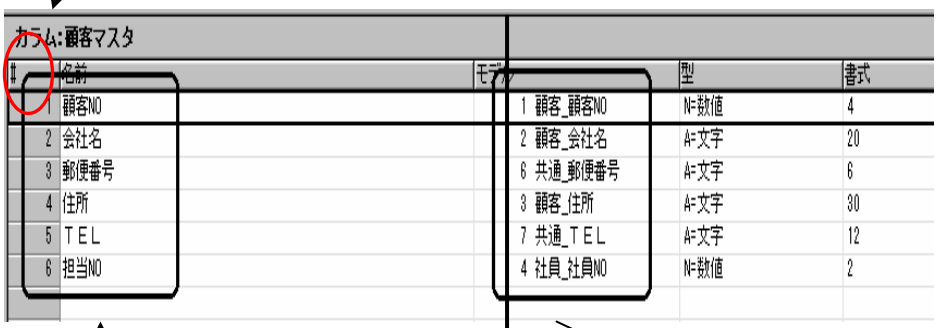
④ 編集(E)をクリックします。

⑤ R:行作成をクリックし、２行作成します。
そして、そこに顧客マスタ、社員マスタと
入力します。

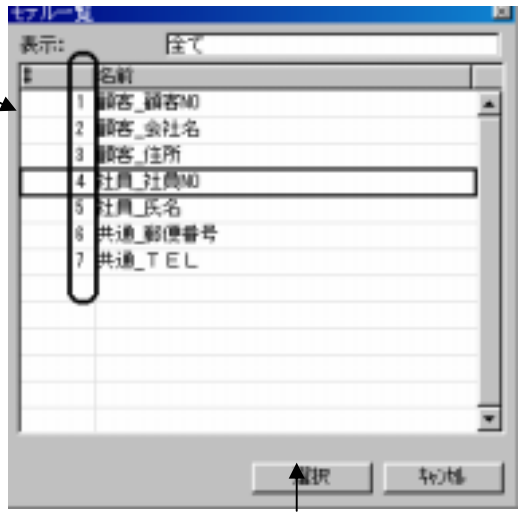


顧客マスタ、社員マスタと入力します。
この名前は物理テーブル名ではありません。

⑦ ＃上で編集(E)のR:行作成をクリックし、６行作成します。
または＃上でＦ４を６回押します。



F5 orダブルクリック



⑧ はじめは空欄ですが全てのモデル欄でダブルクリックまたはF5を押します。

この名前はモデルリポジトリから転記された名前で自由に変更できます。

⑨ 1、2、6、3、7、4、の項目を一つずつクリックして[選択]ボタンを押します。そうしますとモデル名のインデックス番号と名前が前の画面に展開されます。

⑩ 同様にして社員マスタも定義します（次はインデックス定義と物理テーブル名の定義の番になるべきですが、ここは順番を入れ替えても定義できるため、便宜上まとめて説明します。）

カラム:社員マスタ

#	名前	モデル	型	書式
1	社員NO	4 社員_社員NO	N=数値	2
2	氏名	5 社員_氏名	A=文字	12

2）テーブルのインデックス定義

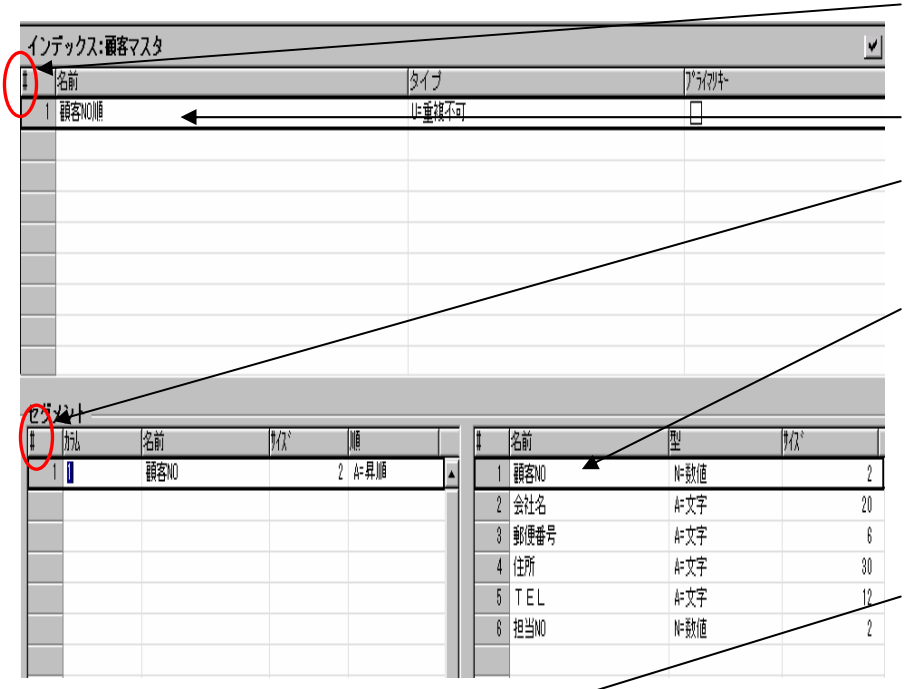
作成するインデックスは２種類
第１インデックス名称　：　顧客NO順　　構成項目　：　顧客NO
第２インデックス名称　：　担当NO・顧客NO順　　構成項目　：　社員NO＋顧客NO

① 顧客マスタのインデックスをダブルクリックまたはF５を押してインデックス定義の画面に入ります。

【インデックス構成項目】【インデックス名称】【テーブル項目】

【操作について】
この画面では３つのテーブルがまとめて表示されています。

A）第１インデックス「顧客NO順」の作成



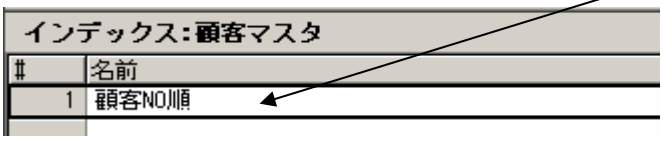
① ＃上で編集(E)のR:行作成をクリックし、１行作成します。
または＃上でＦ４を１回押します。

② 「顧客NO順」と入力します。

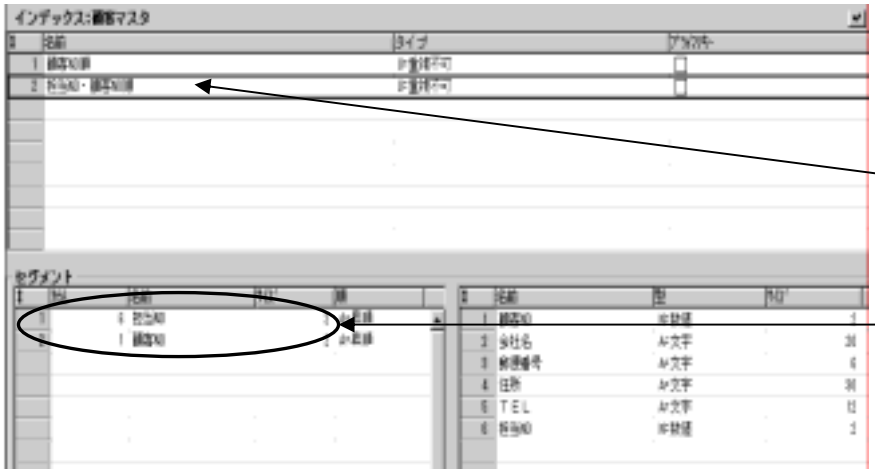
③ ＃上で編集(E)のR:行作成をクリックし、１行作成します。
または＃上でＦ４を１回押します。

④ 顧客NOをダブルクリックします。
そうすると行作成した箇所に値が転記されます。

⑤ [ESC]キーを押すとインデックス名称「顧客NO順」にカーソルが位置付けられます。



B）第２インデックス　「担当 NO・顧客 NO 順」の作成

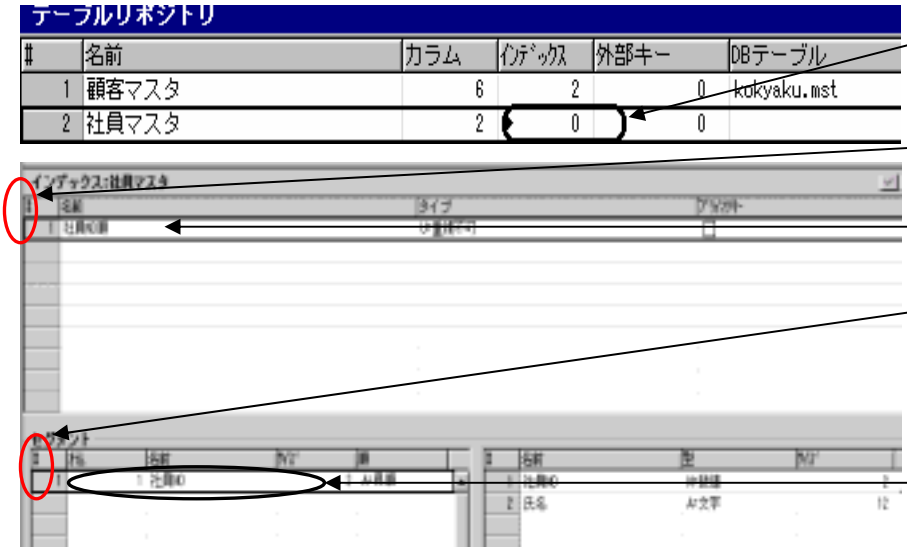


第１インデックスと同様にして「担当NO・顧客NO順」を定義します。
前ページのA)インデックス「顧客NO順」の作成を参考にしてください。

① 「担当NO・顧客NO順」と入力します。

② セグメントとして以下を定義します。
1.担当NO
2.顧客NO

C）社員マスタの第１インデックス　「社員 NO 順」の作成



テーブルリポジトリ

#	名前	カラム	インデックス	外部キー	DBテーブル
1	顧客マスタ	6	2	0	kokyaku.mst
2	社員マスタ	2	0	0	

① テーブルリポジトリの社員マスタインデックスでダブルクリックまたはF５を押します。

② ＃上で編集(E)のR:行作成をクリックし、１行作成します。

③ 「社員NO順」と入力します。

④ ＃上で編集(E)のR:行作成をクリックし、１行作成します。
または＃上でＦ４を１回押します。

⑤セグメントとして以下を定義します。
1.社員NO

3）物理テーブル名の定義

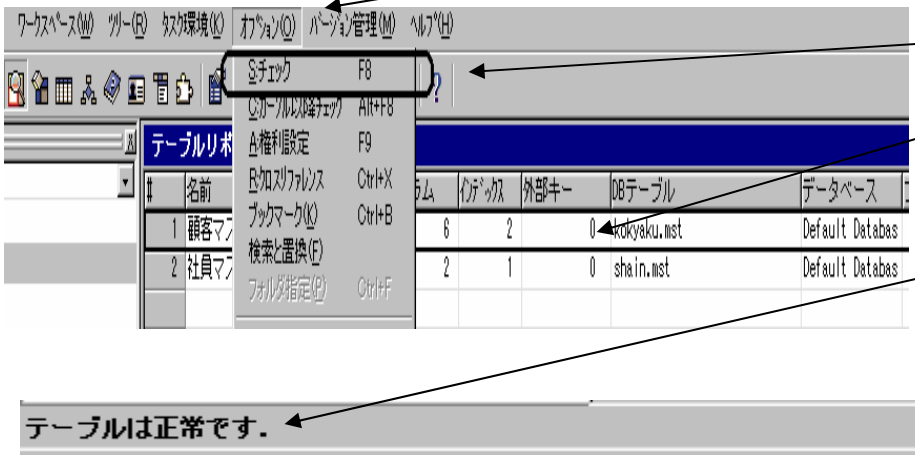
テーブルリポジトリ

#	名前	カラム	インデックス	外部キー	DBテーブル	データベース	フォルダ
1	顧客マスタ	6	2	0	kokyaku.mst	Default Databas	
2	社員マスタ	2	1	0	shain.mst	Default Databas	

① DBテーブルに「kokyaku.mst」と「shain.mst」を入力します（このファイルに実データが保存されます。）

4）テーブル定義のチェック

② オプション(O)をクリックします。



③ S:チェックをクリックします。

① チェックするテーブルをクリックします。

④ テーブル定義内容が正しければ「テーブルは正常です」と表示され、まちがっていればその個所の定義画面が表示されます。

６．　プログラムリポジトリへの登録

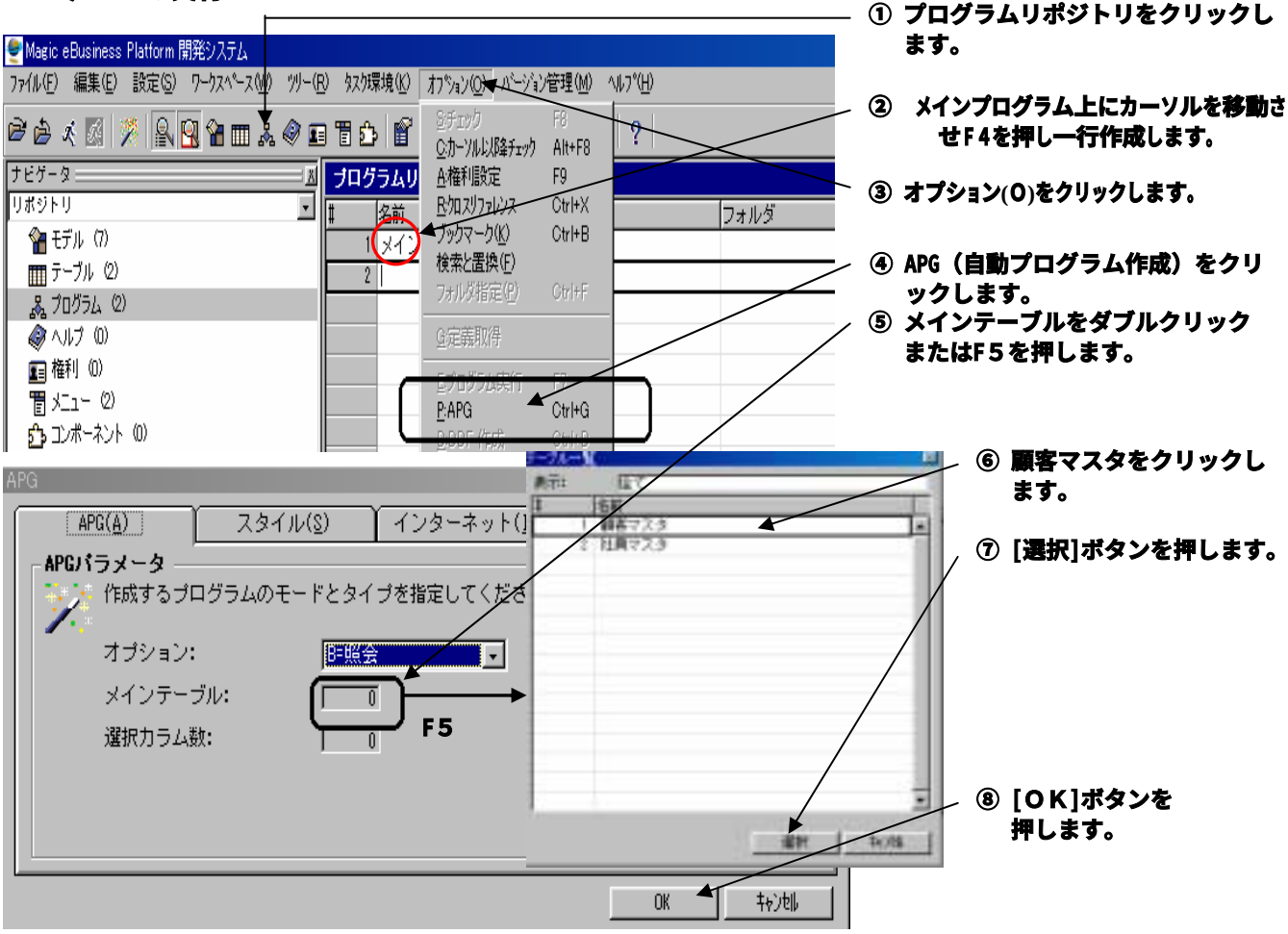
１）Magicによるプログラム作成手法

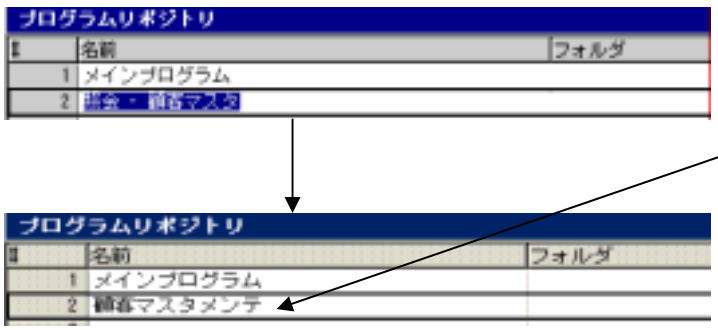
Magicのプログラム作成方法はAPG（自動プログラム作成）によるものと、通常の**手作業**によるものとがあります。勿論APGだけで複雑なプログラムを作成することはできませんので、実際にはAPGでプログラムの大枠を作成し複雑なところを**手作業**で作成します（APGを使わず全て**手作業**で作成することも可能です）。
ただし残念なことにAPGでは、六つの実行モード（B=照会、E=出力、I=入力、P=プリント、N=インターネット、R=ブラウザ）のいずれかのものしか作ることが出来ません。つまり登録や修正のプログラムを作成することはできません。そのようなプログラムを作成する場合は、一度**照会**で作って、手作業で実行モードの設定パラメータを変更する必要があります。

２）顧客マスタプログラムの作成手順

① APGにより大枠のプログラムを作成します。
② 実行モードを修正に変更します。
③ プログラムのチェックをします。
④ 実行（開発モードのまま）します。

３）APG の実行



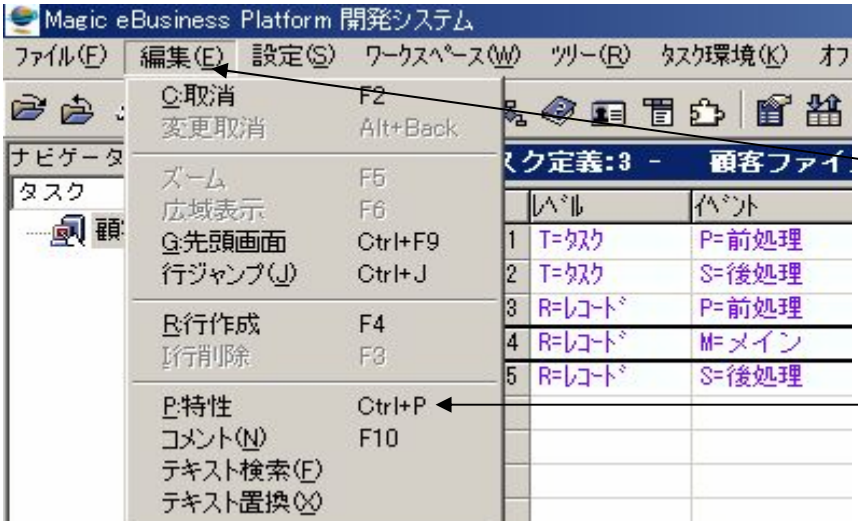


⑨ 「照会－顧客マスタ」を「顧客マスタ」に修正します。

4） 実行モードを修正に変更

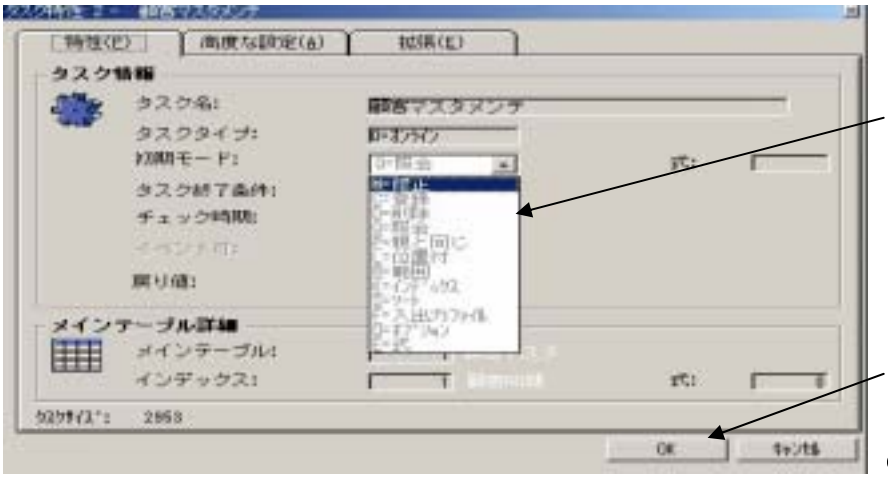
※メインプログラムは必ず出ます。
ここに外部変数やグローバルイベント等を登録します。
また、Magic起動時の開始プログラムもここで設定します。

① プログラム名(顧客マスタ)でダブルクリックまたはF５を押し、タスク一覧を表示します。



② 編集(E)をクリックします。

③ P:特性をクリックします。



④ 照会を修正に変更します。
何故「登録」ではなく「修正」に設定したのでしょうか？
その答えは次のページの実行モード切換を読めば分かるかと思います。

⑤ [ＯＫ]ボタンを押します。

⑥ [ESC]キーを２度押し、プログラムリポジトリ一覧まで戻ります。
途中「変更内容を保存しますか？ 」で「はい」を選択します。

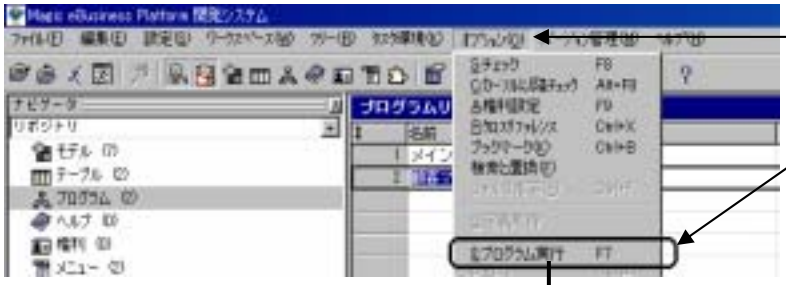
5） プログラムのチェック

① オプション(O)をクリックします。

② S:チェックをクリックします。

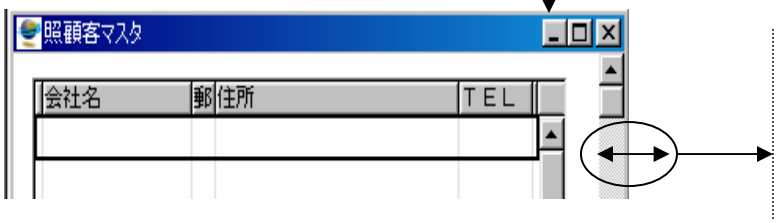
③ プログラム定義内容が正しければ「プログラムは正常です」と表示され、間違っていればその箇所の定義画面が表示されます。

6）開発モードのまま実行



① オプション(O)をクリックします。

② E：プログラム実行をクリックします。



③ APGで作成したデフォルト画面サイズでは全部表示できないのでマウスで広げます。

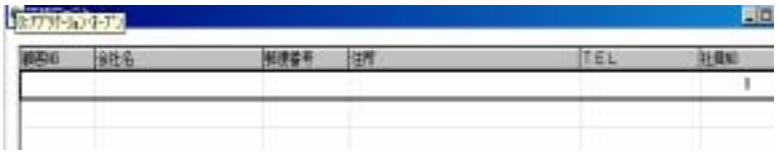
④ 自分の情報等をサンプルデータとして入力します。

＊次項目への移動は[Enter]ではなく[Tab]、終了キーは[ESC]になります。これらキー操作に関しては付録のキー操作一覧をごらんください。

Magicではプログラム実行中でもメニューバーの「オプション(O)」で実行モードを自由に切り換えることができます。

7）　実行モード切換

【C:登録】に切り換えた状態



登録モードは、新規データの追加登録だけを行います。既存データは表示されませんが削除されたわけではありません。

【M：修正】に切り換えた状態

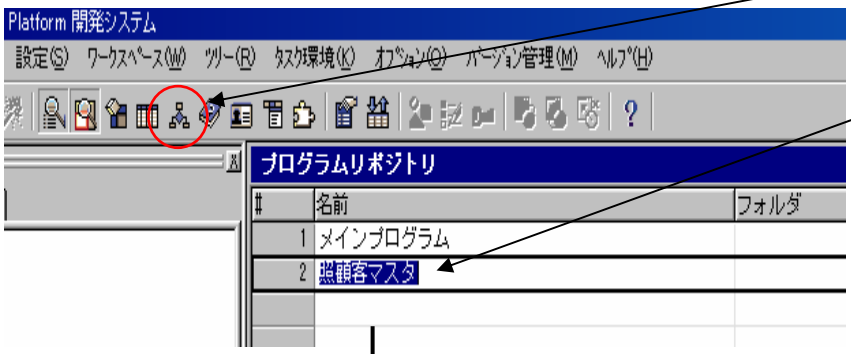
修正モードでは、既存データが表示され、その修正が可能、さらに最終レコードの次の行にカーソルを移動させると（左の例では３行目）何の違和感もなく新規データを追加登録できます。
一覧表形式でメンテナンスする場合はこのモードの方が親和性があります。

【Q：照会】に切り換えた状態

照会モードでは、既存データが表示されますが見るだけで修正することはできません。

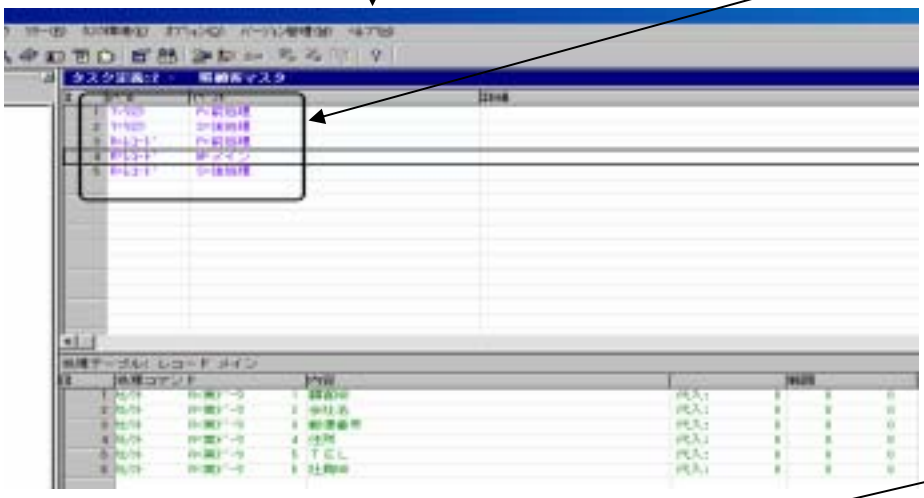
７．　プログラム内容の確認

１）処理テーブルの内容の確認



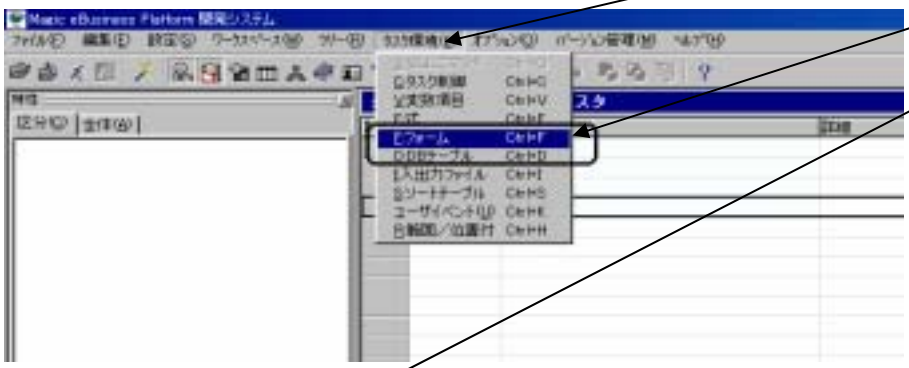
① ツールバーのプログラムリポジトリをクリックします。

② プログラム名でダブルクリックまたはF５を押し、タスク一覧を表示します。



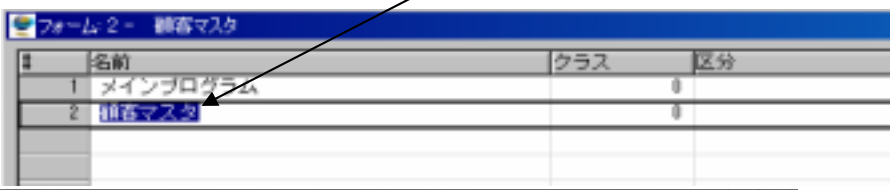
③ ５つの処理レベルをクリックするとそれぞれの処理テーブルが取り出されます。
ここに表示される内容がMagicで作られたプログラムの骨格となる部分です。

２）　画面フォームの確認



① タスク環境(K)をクリックします。

② F:フォームをクリックします。



③ フォーム一覧で「顧客マスタ」をダブルクリックするかF５を押します。

④ APGで自動作成された画面フォームが取り出されます。
ちなみに画面右下にはみ出しているのは、フォームを作成する為の「コントロールパレット」と「コマンドパレット」と呼ばれるツール群です。

⑤ [ESC]キーを４回押すとプログラムリポジトリー覧まで戻ります。
途中「変更内容を保存しますか？」で「はい」を選択します。

「コントロールパレット」

「コマンドパレット」

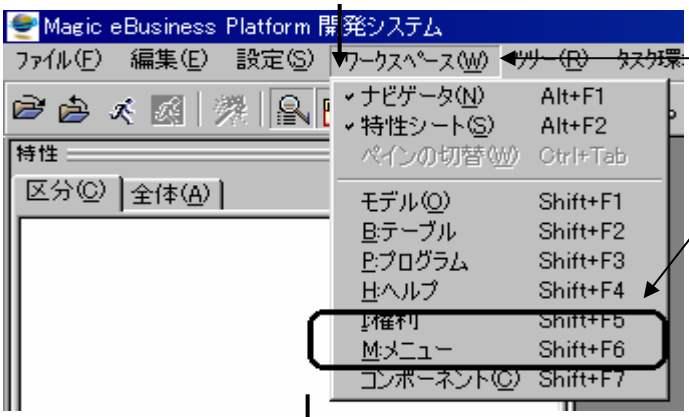
８．　メニューの登録

今度はコンテキストメニュー（ポップアップメニュー）を作成し、そこに今作った「顧客マスタ」を登録し、実際のエンドユーザと同じ様に、そこから起動させてみましょう。

操作の手順の概要

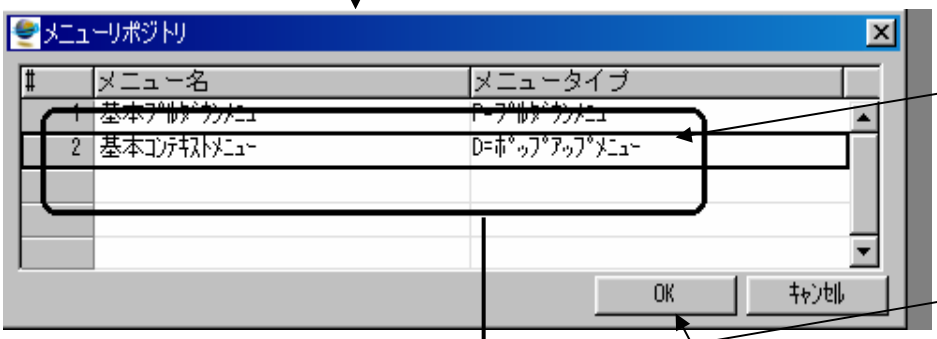
① コンテキストメニューへ顧客マスタメンテを登録します。
② Magicのモードを開発モードから実行モードにします。
③ コンテキストメニューを表示し、プログラムを起動させます。
④ Magic のモードを開発モードに戻します。

１）コンテキストメニューへの登録



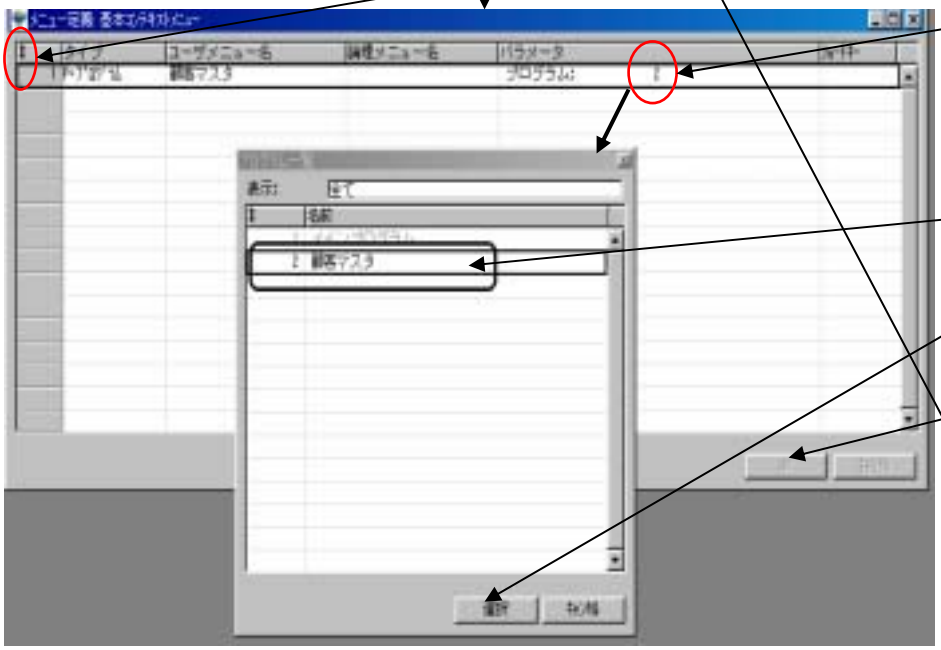
① ワークスペース(W)をクリックします。

② M:メニューをクリックします。



③ コンテキストメニューテーブルの「ポップアップメニュー」をダブルクリックするかＦ５を押します。

④ ＃上で編集(E)のR:行作成をクリックし、１行作成します。



⑤ パラメータのプログラム番号上でダブルクリックするかF５を押します。するとプログラム一覧が表示されます。

⑥ 「顧客マスタ」をクリックします。

⑦ [選択]ボタンをクリックします。

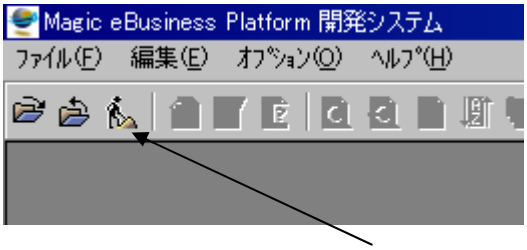
⑧ [ＯＫ]ボタンをクリックします。

⑨ [ＯＫ]ボタンをクリックします。

2）開発モードから実行モードへ

（開発モード）

（実行モード）



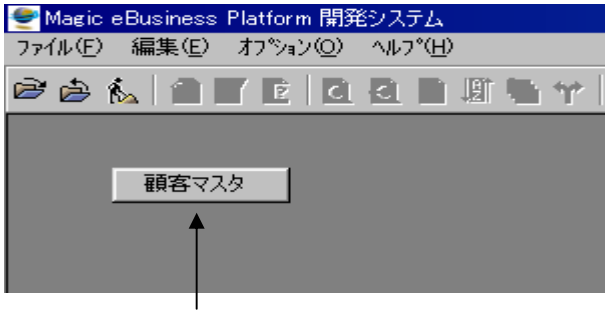
① ツールバーのT:開発／実行モード(左から３番目)をクリックします。

アイコンが変わります

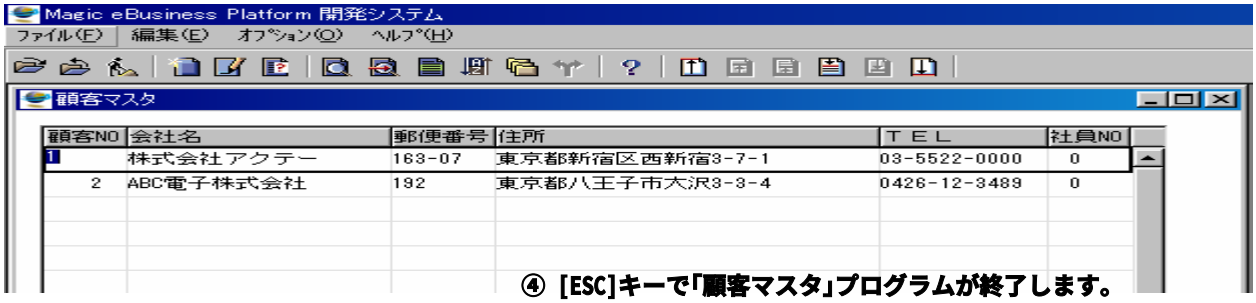
3）コンテキストメニューを取り出し、プログラムを起動させる

② Magicの画面内で
マウスの右ボタンをクリックします。

【起動画面】



③ マウスカーソルの場所にコンテキストメニューが表示されます。
そのままメニューをクリックするとプログラムが起動します。



④ [ESC]キーで「顧客マスタ」プログラムが終了します。

4）開発モードに戻す

アイコンが元に戻ります

⑤ ツールバーのT:開発／実行モード(左から３番目)をクリックします。

ワンポイント

Magicでは、データベースの効率的な構築、プログラミングの生産性およびメンテナンス性を高めるためにプログラム構造の作成手順をルール化し、あらかじめモデルリポジトリ、テーブルリポジトリなどで定義した項目名、書式などをプログラム本体やフォームなどに関連付けて作成する方法をとります。従って、この顧客管理システムを作成する過程では従来のCOBOL等で行っていたステートメントの記述といったコーディング作業はほとんど行われなかったことがお分かりいただけた事と思います。
モデルリポジトリ、テーブルリポジトリはシステム開発時やメンテナンス時の作業効率の向上に非常に威力を発揮します。

COBOLなどのプログラミング言語でも、モデルリポジトリと似たような形で、ファイルのCOPY句などを実際のプログラムとは切り離して定義しておき、ファイルレアウトの変更などが発生した場合には、直接プログラムを触らずに、レイアウト変更を行うことは可能ですが、Magicにおけるモデルリポジトリとはその効率に大きな違いがあります。

以下にその違いをご説明いたします。

（COBOLなどの記述型言語の場合）	（Magicの場合）
1．ファイルレイアウトの変更にあたっては、どのプログラムで変更対象の項目が使用されているのか調査する必要があります。この作業は、関連するワーク用の項目などを含めて検討する必要があり、大変な作業ボリュームとなります。	1．モデルリポジトリやテーブルリポジトリでクロスリファレンス機能が準備されており、変更対象項目がどこで使用されているのか、簡単に確認することができる。
2.プログラム中で定義している作業領域など結局は、プログラムを修正しないといけない。	2.モデルリポジトリやテーブルリポジトを利用して定義していれば、作業用のワークは自動的に修正される。
3.プログラムを作成した後にコンパイル作業が必要。この作業も修正プログラムの本数が多いとかなり大変な作業となります。	3．Magicは開発ｼｽﾃﾑで定義したパラメータの内容を実行ｼｽﾃﾑが直接動作されるので、変更結果を即時に動作させることが可能。
4．ファイルレイアウトを変更すると、当然実際のデータも変更するプログラムを作成し、変換作業を行うことになり作成時間もかかるし、物理データの変換完了後にはキーファイルも生成しなおさなければならない（ＤＢＭＳ にもよる）。	4．物理データ及びキーファイルは自動的に変更することが可能。データ件数が非常に多い場合には、移行ＰＧ作成する場合もあります。
5．項目の特性の変更（例えば、色やフォント）はプログラム中で行う必要があります。	5．モデルリポジトリやプログラムリポジトリの特性設定の変更で対応可能。

など、数多くのメリットがあります。

９．　練習問題

同様に、社員マスタファイルを使って「社員マスタメンテ」プログラムを作って下さい。
モデルリポジトリ、テーブルリポジトリはすでに完成していますので、プログラムリポジトリのみ作成します。

テーブル　：　社員マスタテーブル
プログラム：　社員マスタメンテプログラム。社員マスタデータの登録・更新を行います。

プログラムが完成したら実行し、サンプルで何件かのデータを入力して下さい。入力データは次のプログラムで営業担当者のテーブルとしても使用しますので、社員 NO は１から順番に入力して下さい。

【完成画面】

第４章　　顧客管理システムの機能追加

この章では、第３章で作った簡単な顧客管理システムに機能を追加し、もう少し使いやすいプログラムに修正する手順について説明します。

１．担当者NOを入力し、担当者名を取り出す
２．参照テーブルから担当者名を取り出す

１．　担当 NO を入力し、担当者名を取り出す

１）追加機能の概要

顧客マスタメンテプログラムの担当 NO 欄で社員番号を入力すると担当者名を表示するように変更します。

【完成画面】

２）プログラム変更の手順

① リンクコマンドの追加

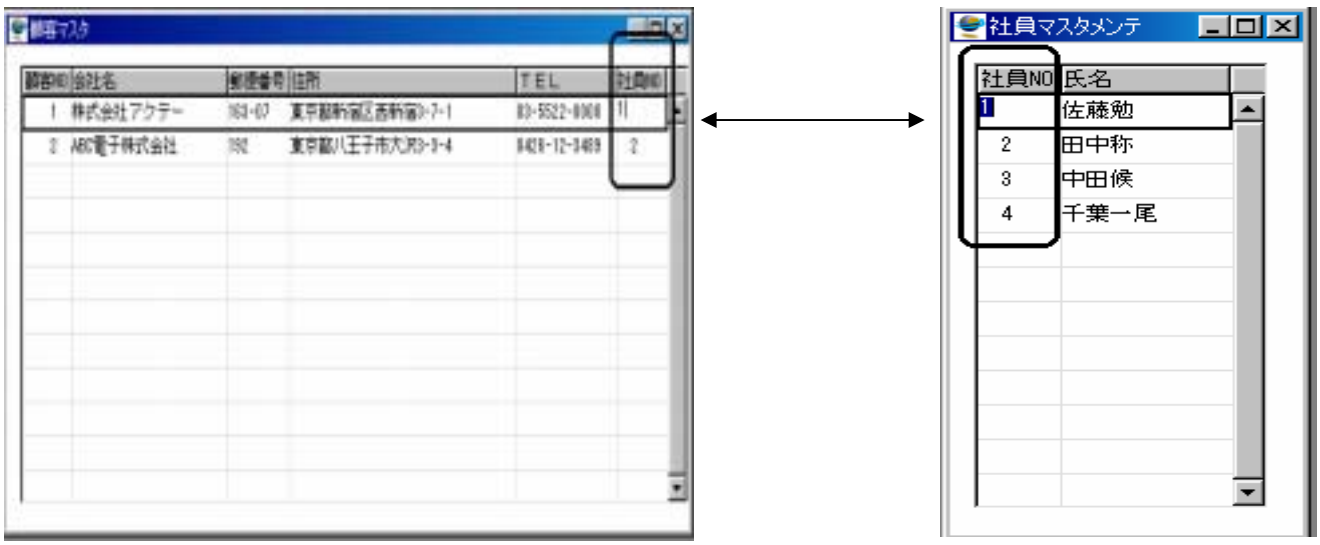
顧客マスタメンテプログラムから社員マスタテーブルのレコードを読み込めるように、**リンクコマンドを追加します。**

② 画面フォームの変更

読み込んだ社員マスタテーブルのレコードを画面表示できるように画面フォームを変更します。変更方法には２つあります。画面フォームを自動作成する方法と既存の画面フォームに[氏名]を追加する方法とがあります。今回は前者を使います。

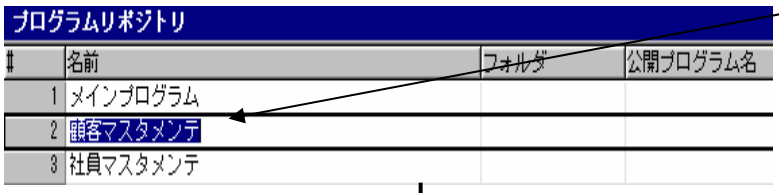
３）リンクコマンドの概要

顧客マスタテーブルと社員マスタテーブルといった別々のテーブルに対し、ある項目の値が等しいレコード同士をメモリ上で一つにつなぎあわせたように見せるときに使われるコマンドです。新しいテーブルが作成されるわけではありません

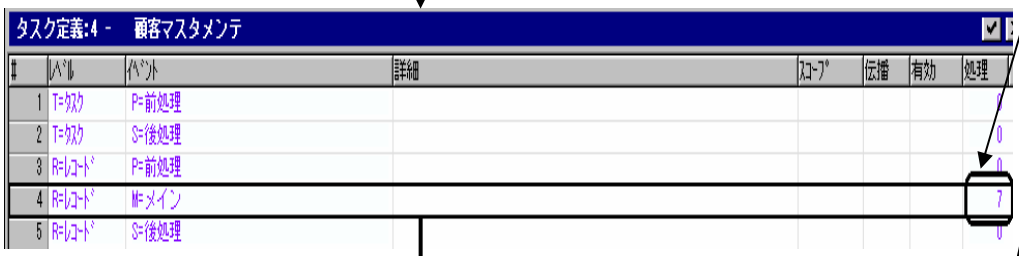


4）リンクコマンドの追加

① ツールバーのプログラムリポジトリをクリックします。



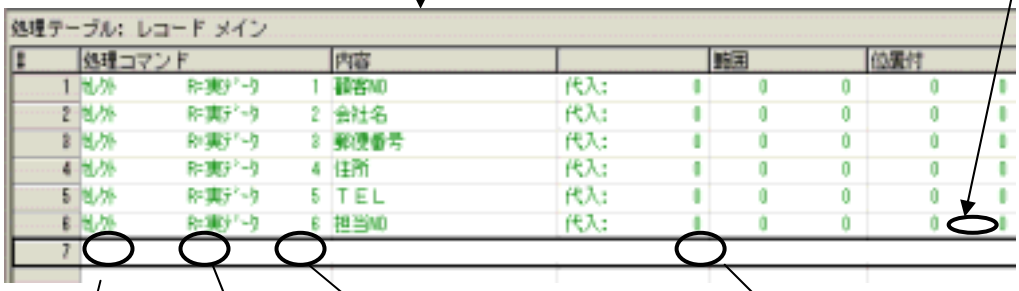
②プログラムリポジトリの「顧客マスタ」のプログラム名を「顧客マスタメンテ」に変更し、ダブルクリックするかＦ５を押します。



③ 画面表示内容が異なる場合は処理レベルテーブルのレコードメインをクリックします。

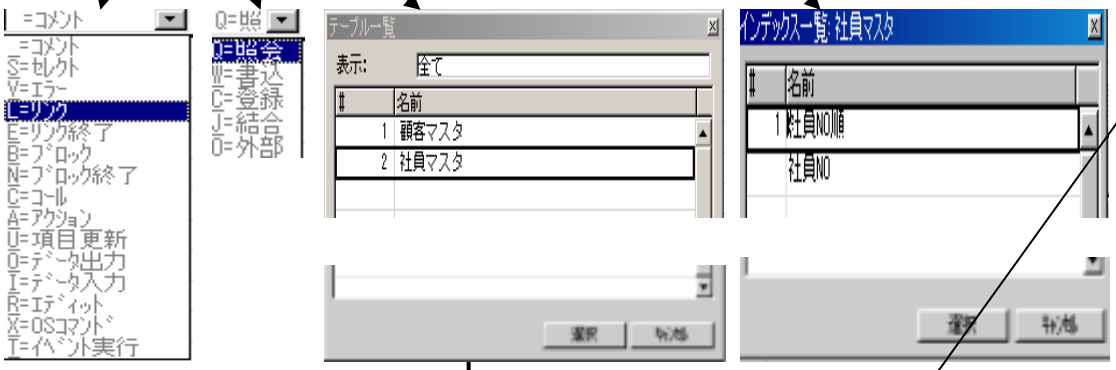
④ 連結条件である「担当NO」の行をクリックします。

⑤ Ｆ４キーを押し、その下にリンクコマンドを記述するための空白行を作成します。



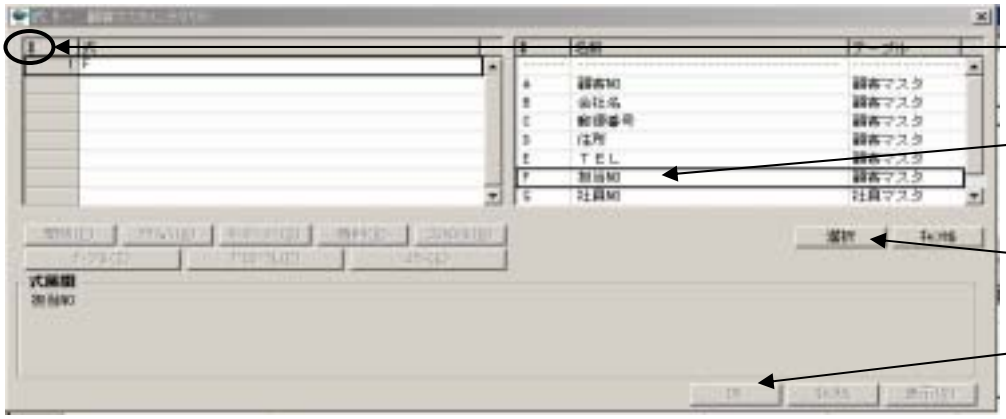
⑥ 空白行の○の個所を左から順にダブルクリックするかＦ５を押し、次の設定をします。

コマンド　　：Ｌ＝リンク
リンクタイプ：Ｑ＝照会
テーブル　　：社員マスタ
インデックス:社員NO順



⑦ リンクとリンク終了の間にある「社員NO」(8行目)位置付けの小(左側)でダブルクリックするかＦ５を押します。

＊顧客マスタテーブルの担当者NO
||
社員マスタテーブルの社員NO
のリンク条件を設定するところが位置付の「小」「大」です。
等号の場合は両方とも同じ項目（または値）を設定します。

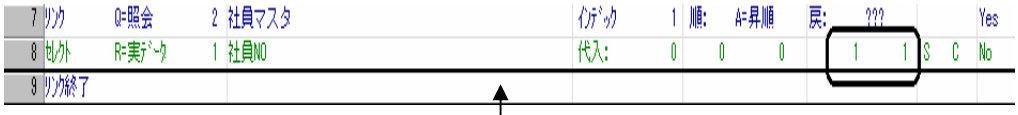


⑧ ＃上でF4を１回押し１行作成します。

⑨ 項目一覧テーブルから「Ｆ：担当NO」をクリックします。

⑩ [選択]ボタンを押します。

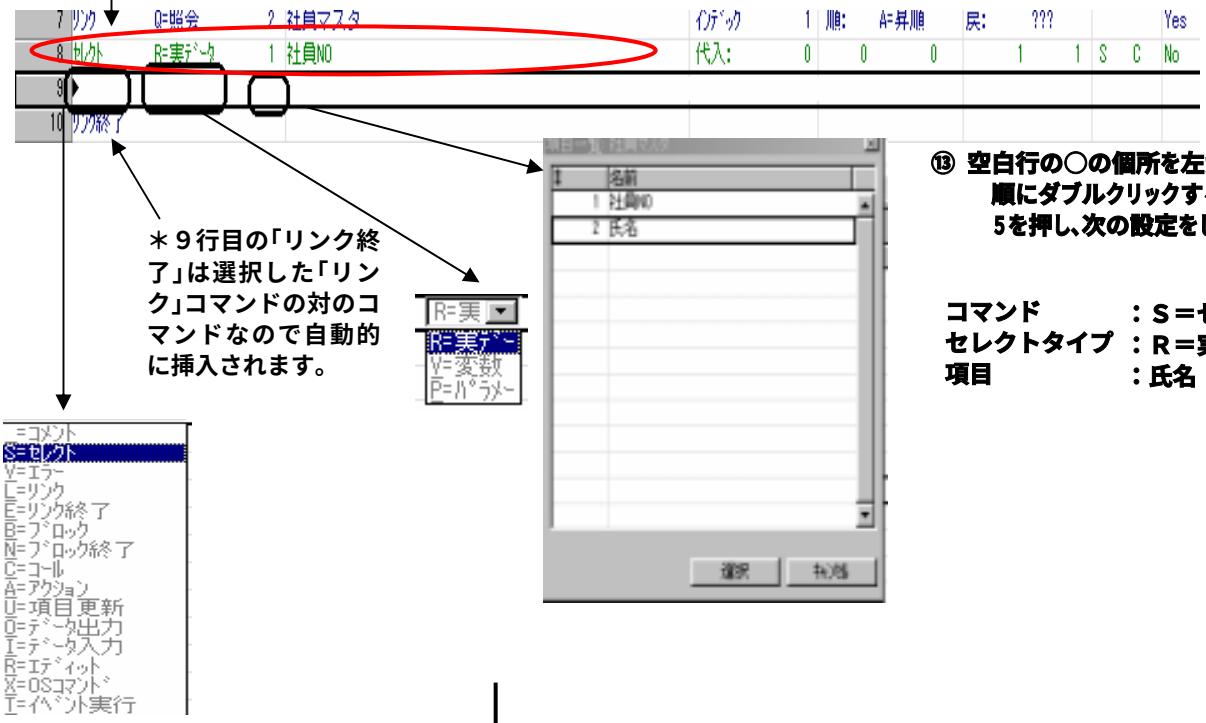
⑪ [ＯＫ]ボタンを押します。



⑫ ２つの異なるテーブルのレコードを等しい条件でリンクする基本設定は完了しました。後は同時に取り出したい残りの項目を「リンク終了」コマンドまでの行に挿入します。「社員NO」の上で、Ｆ４を押し「社員NO」の下に空白行を１行作成します。

＊８行目の「セレクト：社員NO」は選択した「インデックス：社員NO順」の項目なので動的に挿入されます。

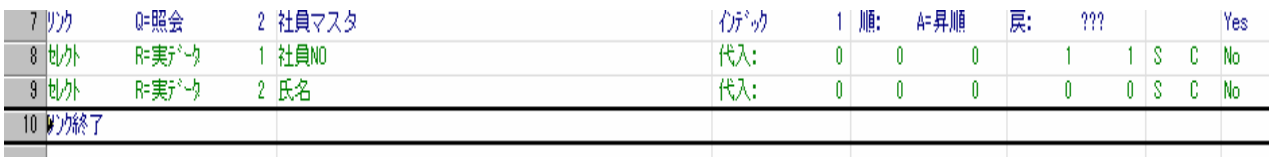
⑪ 位置付けの「小」の値が「１」になっています。これは、先ほどの式のテーブルの行番号を示しています。位置付け「大」も同じ「社員NO」の項目を設定しますので、同じ行番号「１」を設定します。



＊９行目の「リンク終了」は選択した「リンク」コマンドの対のコマンドなので自動的に挿入されます。

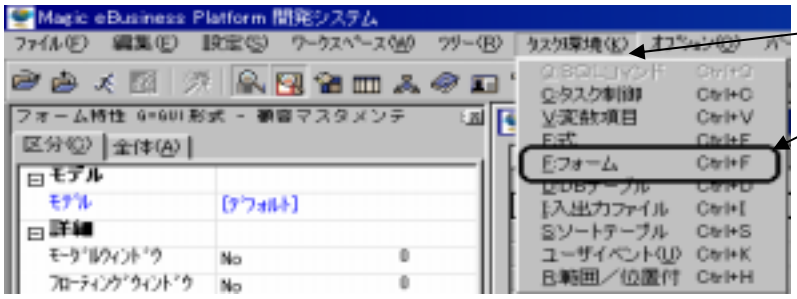
⑬ 空白行の○の個所を左から順にダブルクリックするかF5を押し、次の設定をします。

コマンド　　：Ｓ＝セレクト
セレクトタイプ：Ｒ＝実データ
項目　　：氏名



リンクコマンドの追加に関する処理はこれで完了です。
次は画面フォームを変更します。

5）画面フォームの変更



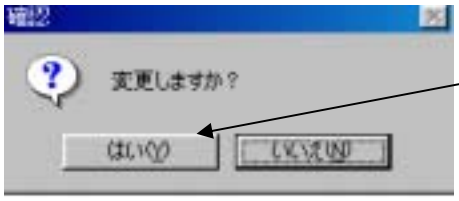
① タスク環境(K)をクリックします。

② Ｆ：フォームをクリックします。

③ フォーム一覧の「顧客マスタメンテ」を修正します。修正方法は２つあります。
ａ）フォームを自動作成する方法
ｂ）フォームに「氏名」を追加する方法
今回はａ）で処理します。

④ オプション(O)をクリックします。

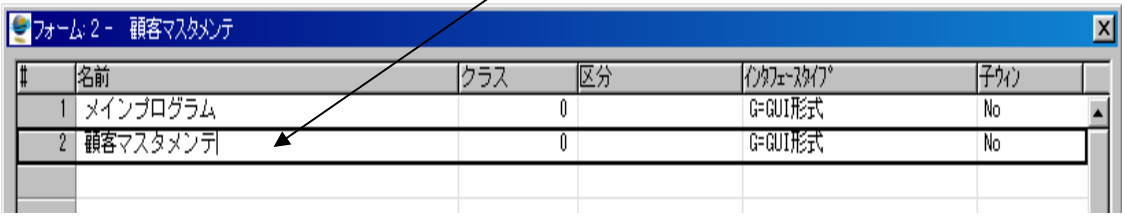
⑤ Ｆ：フォーム自動作成をクリックします。



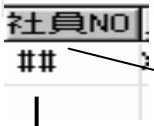
⑥「はい」のボタンを押します。

⑦ [ＯＫ]ボタンを押します。

⑧ 画面フォームの自動作成は完了しましたので、次は微調整です。「顧客マスタメンテ」をダブルクリックするかＦ５を押します。



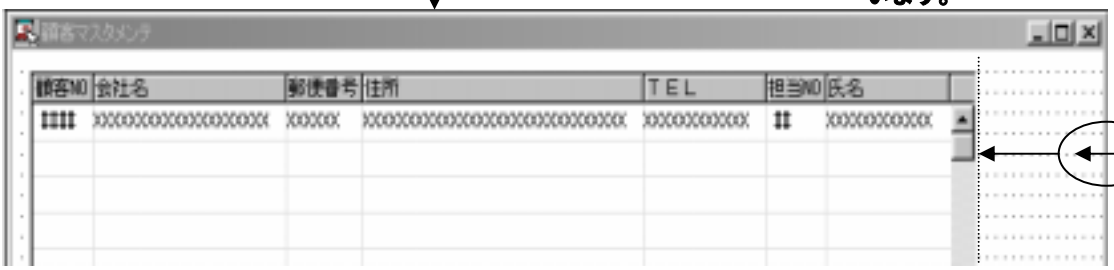
⑨ デフォルトの表示幅は狭いので適切なサイズに広げます。



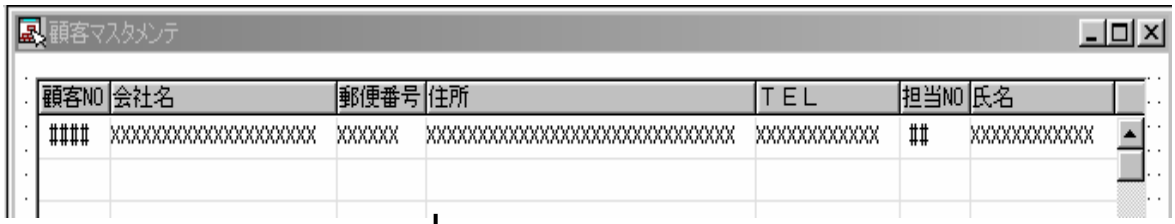
⑩ 追加された項目が組み込まれているのを確認します。ただし「社員NO」は不要です。

⑪ 注意深く「＃＃」の中心部分をマウスでクリックし[ＤＥＬ]を押すと「社員NO」項目全体が削除されます。

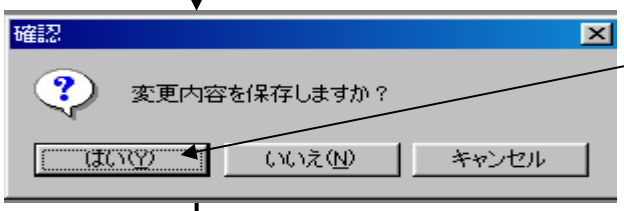
＊ 全体が選択された時の解除はスペ ースバーで行います。



⑫ 表示幅を調整します。



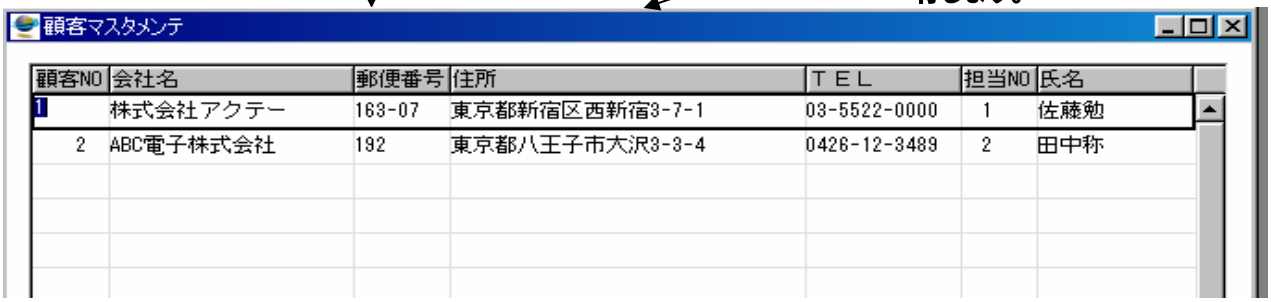
⑬ [ESC]ボタンを押します。



⑭ [はい]ボタンを押します。

⑮ [ESC] キーを3回押しプログラム一覧まで戻ります。途中「変更内容を保存しますか？」と２回聞いてきますので、全て「はい(Y)」を選択してください。

⑯ Ｆ８を押しプログラムチェックをかけ問題がないことを確認しプログラム実行します。

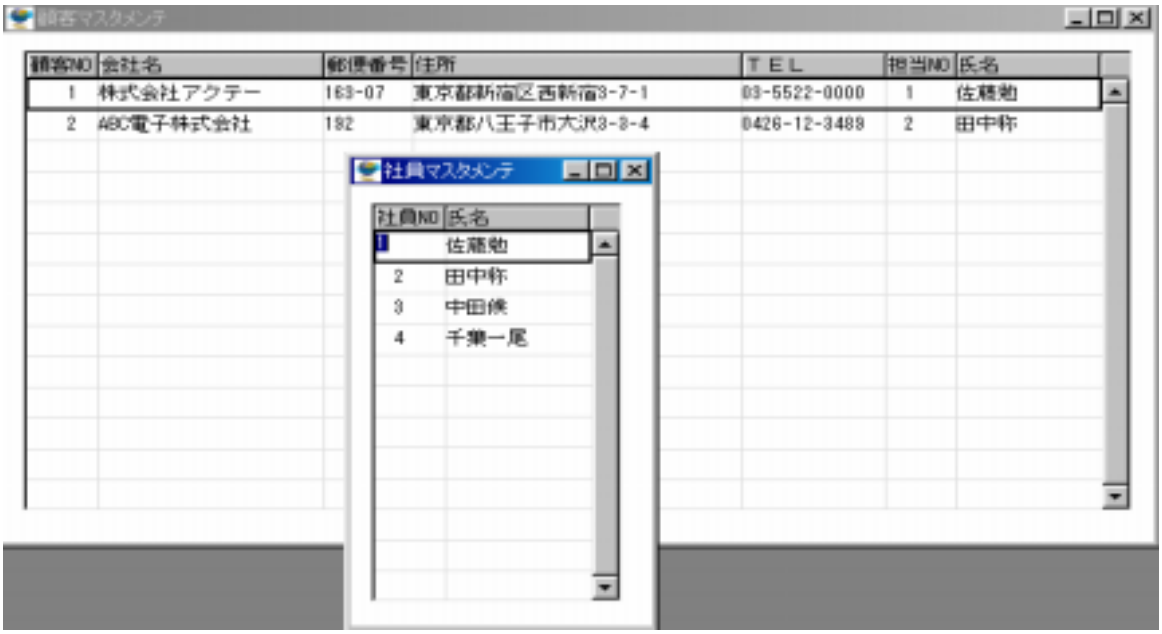


2．　参照テーブルから担当者を取り出す

1）追加機能の概要

顧客マスタメンテプログラムの担当者NO欄でダブルクリックするかＦ５を押すとサブウィンドウに担当者一覧が表示され､一覧から担当者を選択できる機能を追加します。

【完成画面】



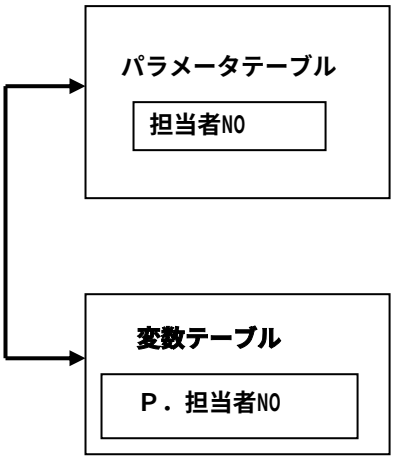
2）プログラム変更の手順

① **コールコマンドの追加**

練習問題で作成した社員マスタメンテプログラムを顧客マスタメンテプログラムから起動する**コールコマンド**を追加します。

② **データの受け渡しパラメータの追加**

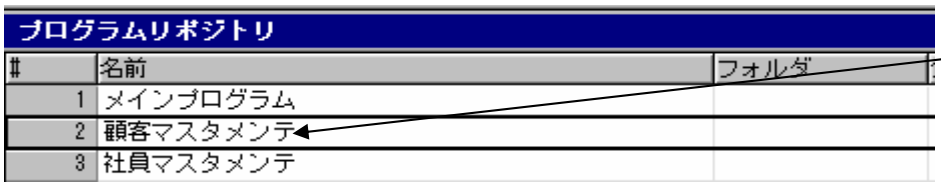
サブウィンドウの社員マスタメンテプログラムで選択したデータを顧客マスタメンテプログラムにパラメータの引渡しが出来るように機能を追加します。



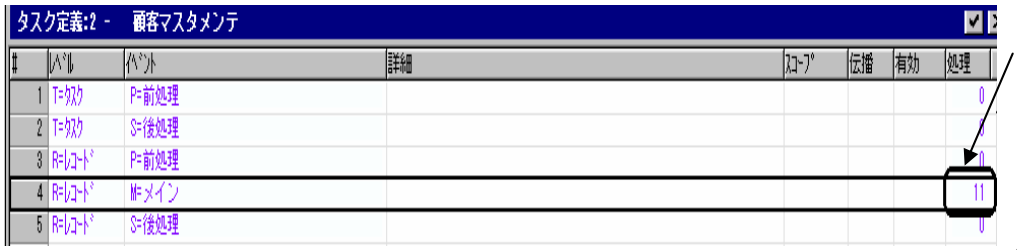
【顧客マスタメンテプログラムの変更】
顧客マスタメンテプログラムで受渡し/受取りに使用する「担当NO」を**コールコマンドのパラメータテーブル**に設定します。「氏名」は既に組み込んである**リンクコマンド**により自動的に表示されるので設定する必要はありません。

【社員マスタメンテプログラムの変更】
親プログラムの内容（パラメータ）を子プログラムに渡す為のエリアを用意します。渡すと書いていますが実際にはこのエリアでパラメータの受渡しと受取りをします。
そして､そのエリアはテーブルの一行目から順番に必要個数分を定義する必要があります。
変数テーブルは一般変数を定義するところでもあるので注意が必要です。

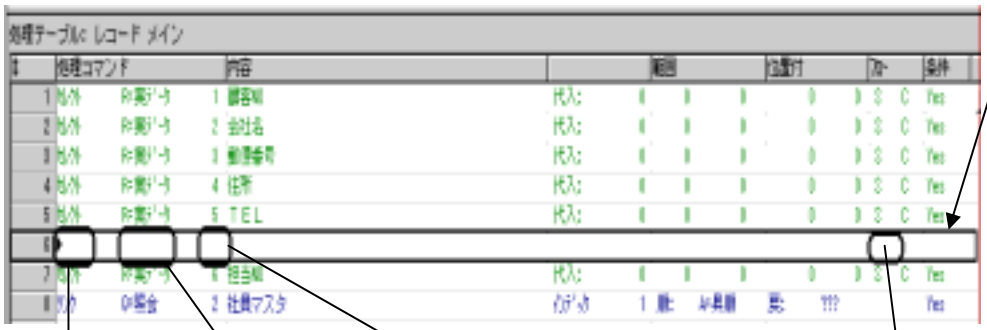
3）コールコマンドの追加



① プログラムリポジトリの「顧客マスタメンテ」をダブルクリックするかＦ５を押します。

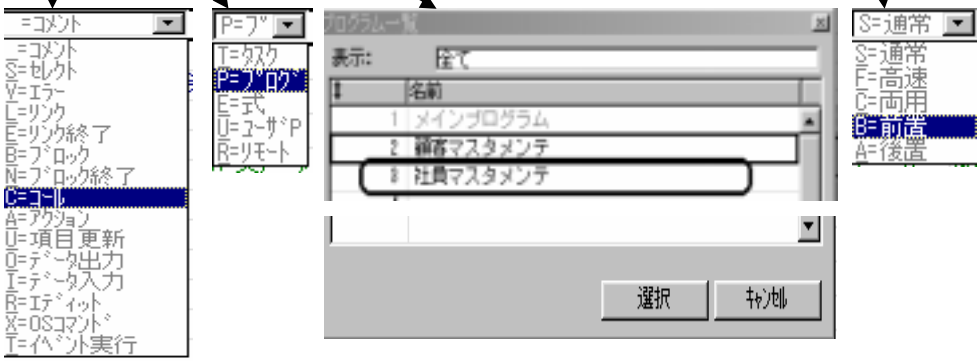


② 画面の表示内容が異なる場合は処理レベルテーブルのレコードメインをクリックします。



③ 「担当NO」の上に空白行を挿入したいので一旦「ＴＥＬ」の行をクリックします。

④ Ｆ４キー（行作成キー）を押し、その下にコールコマンド用の空白行を作成します。

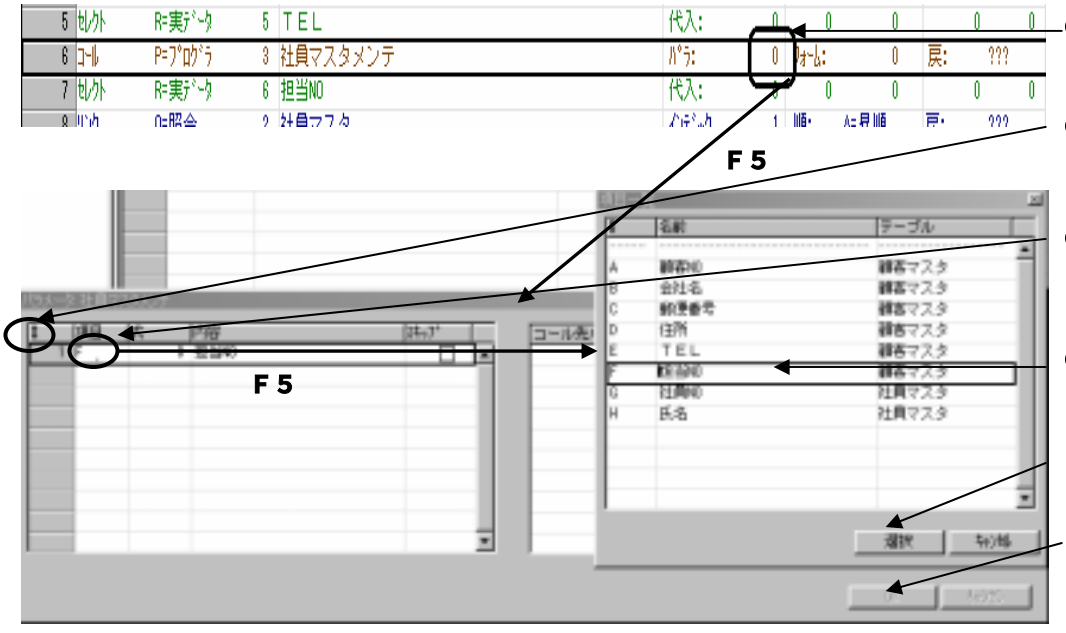


⑤ 空白行の○の個所を左から順にダブルクリックするかＦ５を押し、次の設定をします。

コマンド　　：Ｃ＝コール
コールタイプ：Ｐ＝プログラム
プログラム　　：社員マスタメンテ
フロー　　：Ｂ＝前置

＊ フローの箇所は左側と右側の２箇所あります。左側を選んでください。

4）親プログラムにパラメータ「担当 NO」を追加



① 「パラ」でダブルクリックまたはＦ５を押します。

② ＃上でＦ４キー（行作成キー）を押し、空白行を作成します。

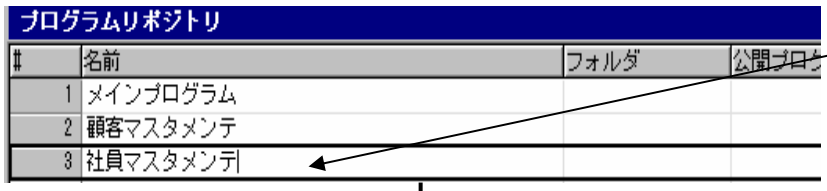
③ １行作成した項目上でF５を押し項目一覧を表示させます。

④ 担当NOをクリックします。

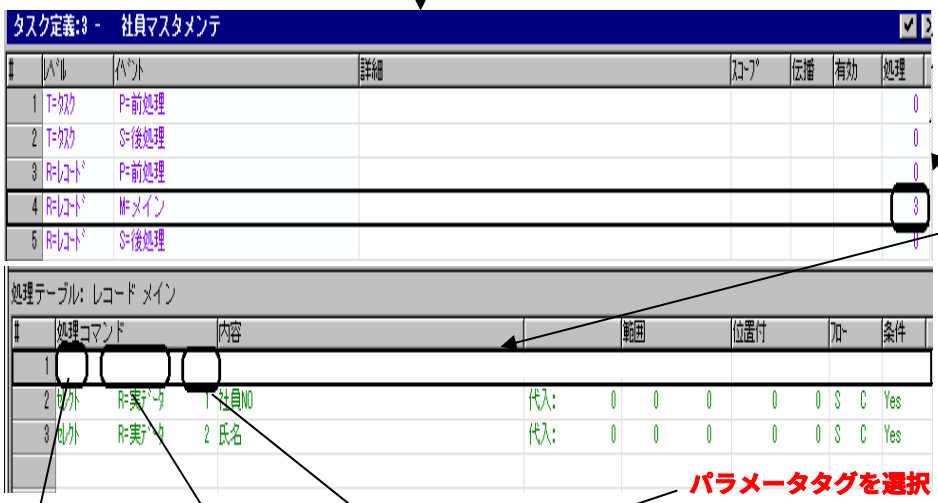
⑤ [選択]ボタンをクリックします。

⑥ [ＯＫ]ボタンを押します。

5）子プログラムにパラメータ用変数を追加



① プログラムリポジトリの「社員マスタンテ」をダブルクリックするかＦ５を押します。

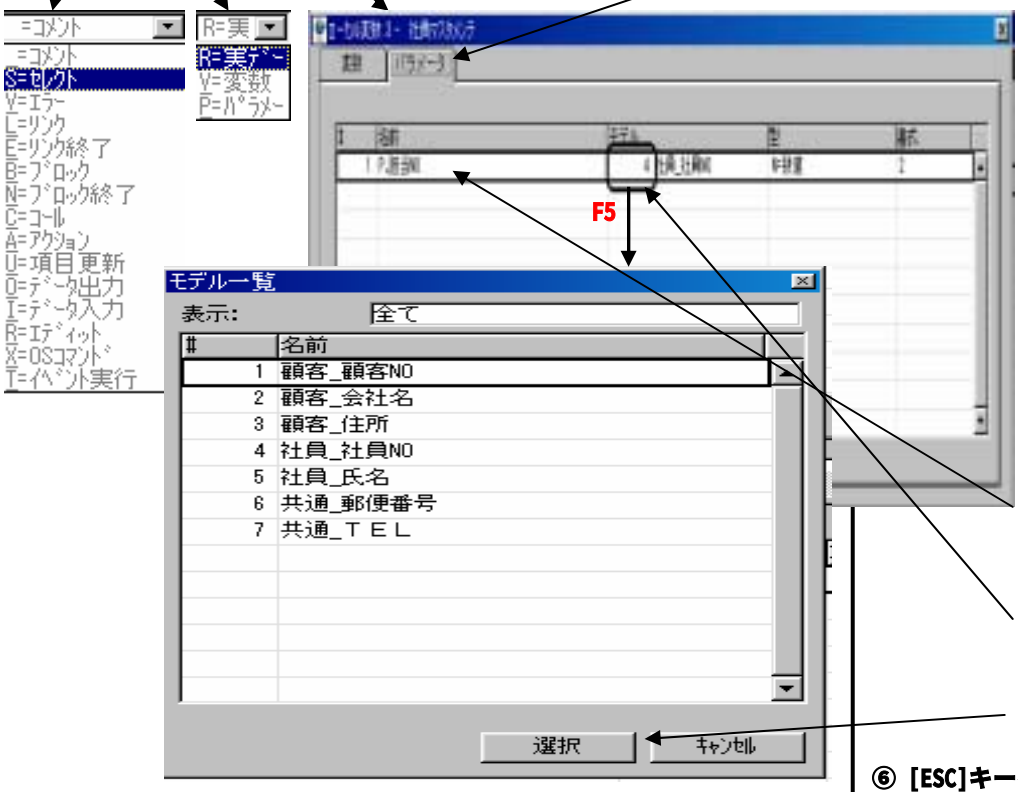


② 画面の表示内容が異なる場合は、処理レベルテーブルのレコードメインをクリックします。

③ 第一行目に空白行を作る場合は見出し行(＃上)をクリックし、カーソルをその前の位置に移動させます。

④ Ｆ４キー（行作成キー）を押しその下にパラメータ変数定義用の空白行を作成します。

パラメータタグを選択



⑤ 空白行の○の個所を左から順にダブルクリックするかＦ５を押し、次の設定をします。

P.担当NOは、モデルを使用します。
名前欄に[P.担当NO]と入力後モデル欄でF５を押しモデル一覧から「社員_社員NO」を選択します。

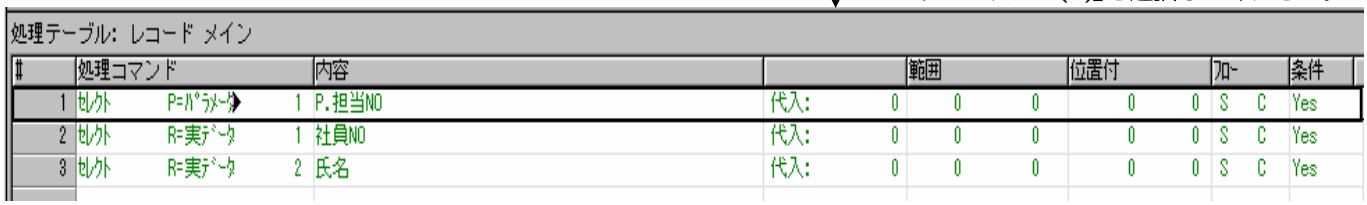
コマンド：Ｓ＝セレクト
セレクト：Ｐ＝パラメータ
項目名　：P.担当NO
モデル　：社員_社員NO

⑤A 名前を[P.担当NO]と入力します。

⑤B モデル欄でＦ５を押します。

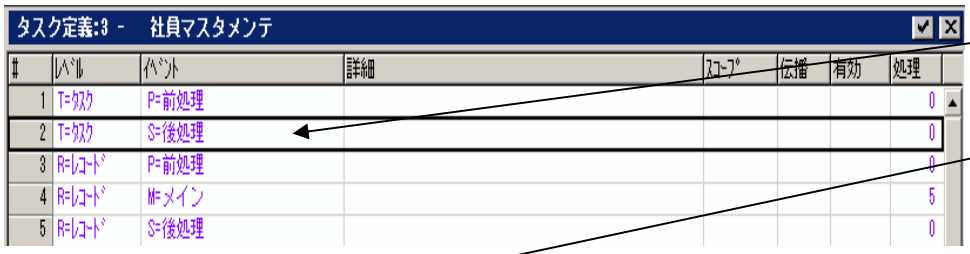
⑤C [選択]ボタンをクリックします。

⑥ [ESC]キーを押します。[ESC]キーを押すと「変更内容を保存 しますか？」と聞いてきますので、「はい(Y)」を選択してください。



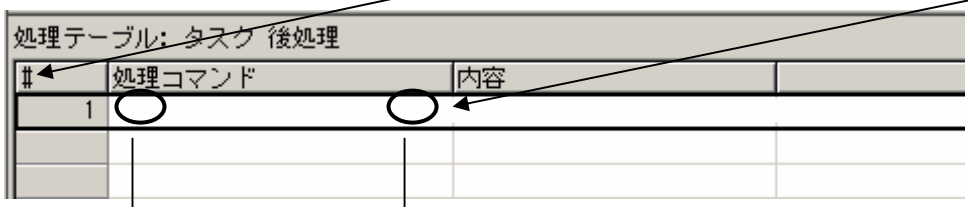
パラメータの設定は完了です。次はプログラム終了時に「社員NO」の値をこの変数に更新します。

6）プログラム終了時にパラメータの値を更新



① タスクの後処理をクリックします。

② 処理テーブルの見出し行(＃の上)をクリックします。



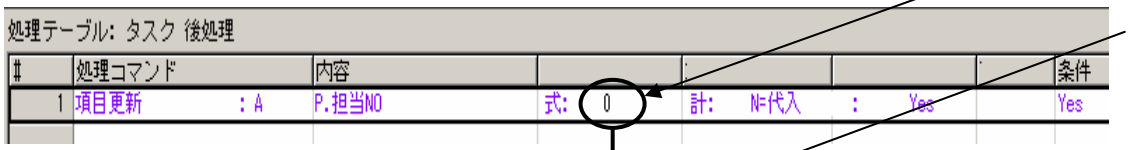
③ Ｆ４キー（行作成キー）を押し、パラメータの値を更新するための空白行を作成します。

④ 空白行の○の個所を左から順にダブルクリックするかＦ５を押し、次の設定をします。
コマンド：Ｕ＝項目更新
項目　　：Ｐ．担当NO

④A 「P.担当NO」をクリックします。

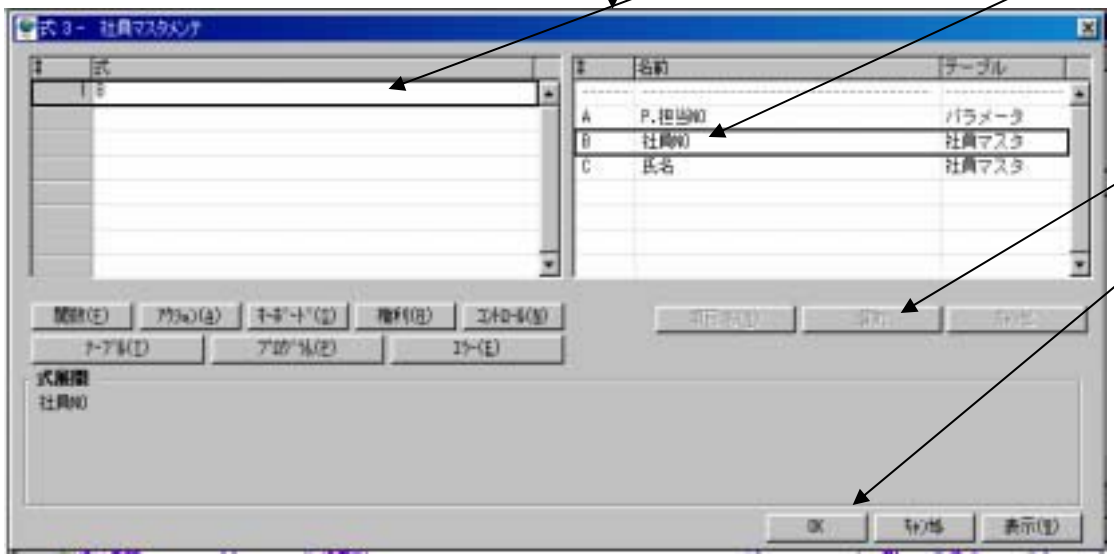
④B [選択]ボタンを押します。

⑤ F5を押します。



⑥ ＃上でＦ４を押して行を作成します。

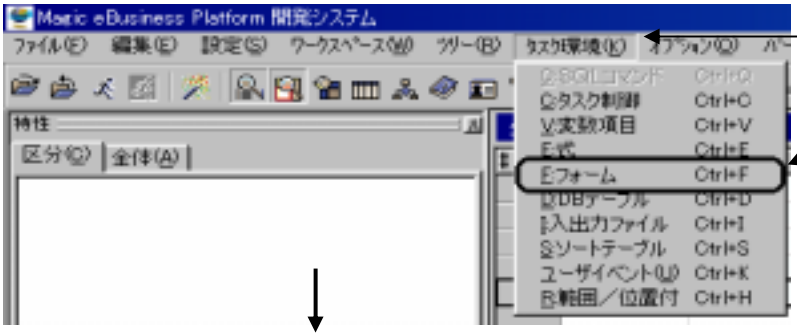
⑦ 項目一覧テーブルから更新元となる「Ｂ：社員NO」をクリックします。



⑧ [選択]ボタンを押します。

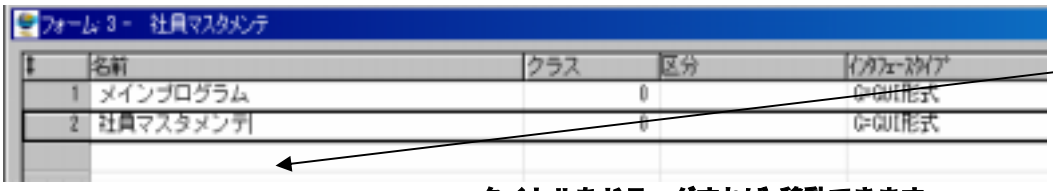
⑨ [ＯＫ]ボタンを押します。

7）画面フォームの変更



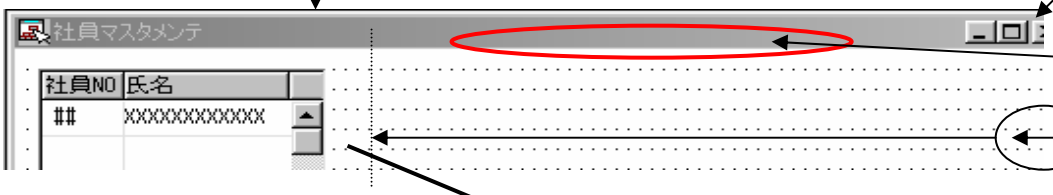
① タスク環境(K)をクリックします。

② Ｆ：フォームをクリックします。



③ 社員マスタメンテ上でダブルクリックをするかＦ５を押します。

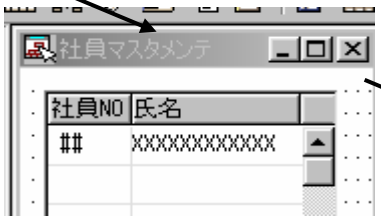
タイトルをドラッグすれば、移動できます。



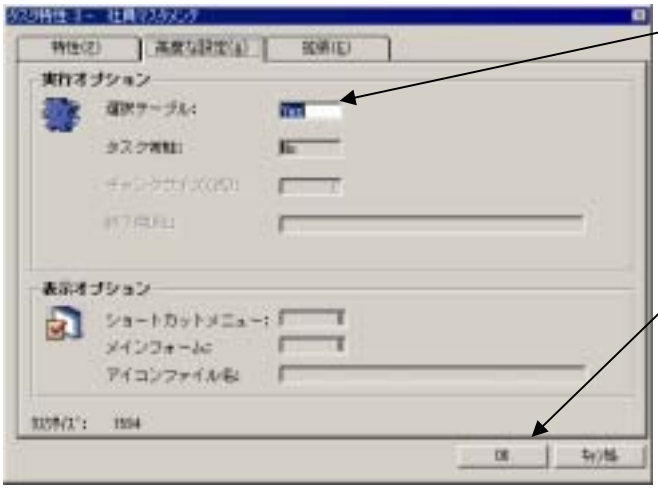
④ 表示幅を調整します。

⑤ 表示位置は見やすいように画面の中央に移動しておきます。

⑥ [ESC]キーを押します。



⑦ 変更内容を保存しますか？で[はい]ボタンを押します。



⑧ Ｃｔｒｌ+Ｐを押してタスク特性画面を開き、高度な設定タブをクリックします。
選択テーブル欄に「Yes」と入力します。

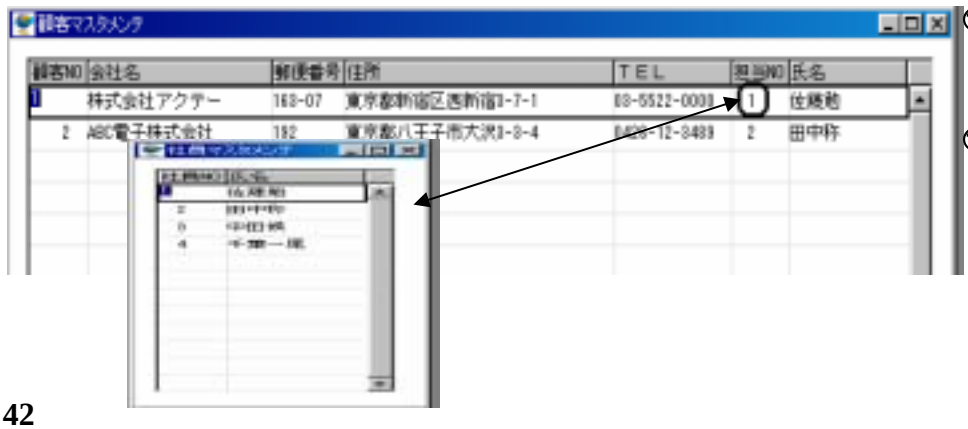
⑨ ＯＫボタンを押します。

⑩ [ESC]キーを押します。

⑪ 変更内容を保存しますか？で[はい]ボタンを押します。

⑫ プログラムリポジトリに戻ってきますので、プログムチェックをかけ「顧客マスタメンテ」上でF7を押し、実行します。

⑬ 次項目への移動は、「Tab」で移動します。



⑭ 顧客マスタメンテの担当NO上でダブルクリックまたはＦ５を押します。

⑮ 社員マスタメンテ上で[ESC]キーを押すと社員マスタメンテが終了します。社員マスタメンテの終了と同時にカーソルがある所の値が顧客マスタメンテに値がコピーされます。

第５章　　バッチ処理プログラムの作成

この章では、顧客マスタファイルを使って簡単なバッチ処理プログラムの作成手順について説明します。

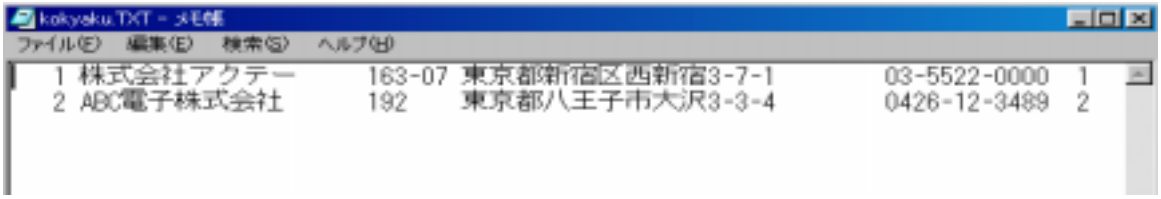
１．テキストファイルの出力と入力処理
２．顧客一覧表の印刷
３．担当者毎に顧客件数を集計して印刷（グループ処理）

１．　テキストファイルの出力と入力処理

１）機能概要

顧客マスタテーブルのデータレコードをテキストファイルに固定長フォーマットで出力します。

【テキストファイルをメモ帳で開いた状態】

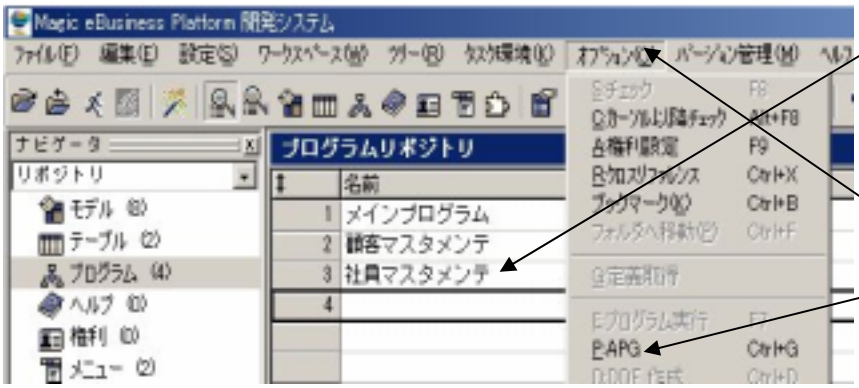


２）プログラム作成の手順

出力処理も入力処理もAPG（プログラム自動作成）で簡単に出来てしまいます。
今回は出力処理を作成し実行します。入力処理の作成方法はAPGの指定方法の説明の中で触れますが、本書では作成も実行もしません。もし既定値の指定で入力処理を作成し実行すると出力処理で出力したテキストファイルを読み出すことになります。そのデータは重複不可インデックス「顧客NO順」と全て重複するため、実際には登録されることはなく、エラーメッセージと警告音をデータ件数分表示して終了します。

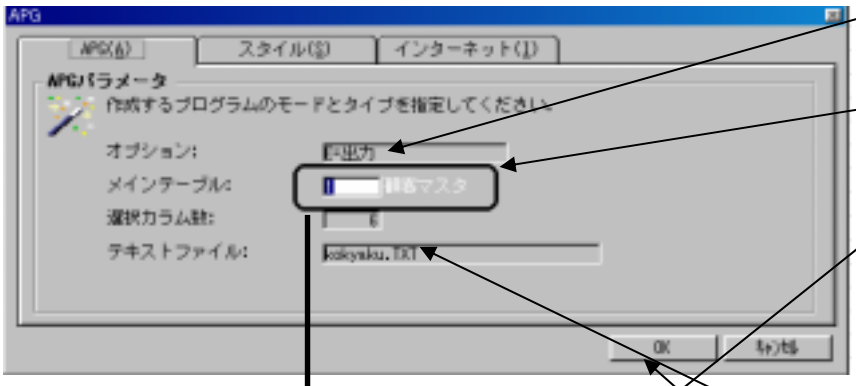
３）テキストファイル出力処理をAPGで作成

① 社員マスタ上でＦ４を押し、顧客マスタで使用する空白行を作成します。



② オプション(O)をクリックします。

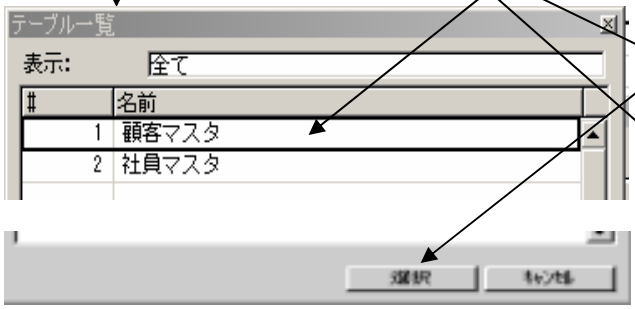
③ APG（自動プログラム作成）をクリックします。



④ APGパラメータのオプションで「E=出力」を選択します。

⑤ APGパラメータのメインテーブルでＦ５を押しテーブル一覧を表示させます。

⑥ テーブル一覧から「顧客マスタ」を選択します。

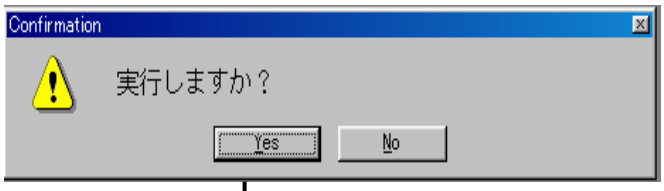


⑧ [選択]ボタンを押します。

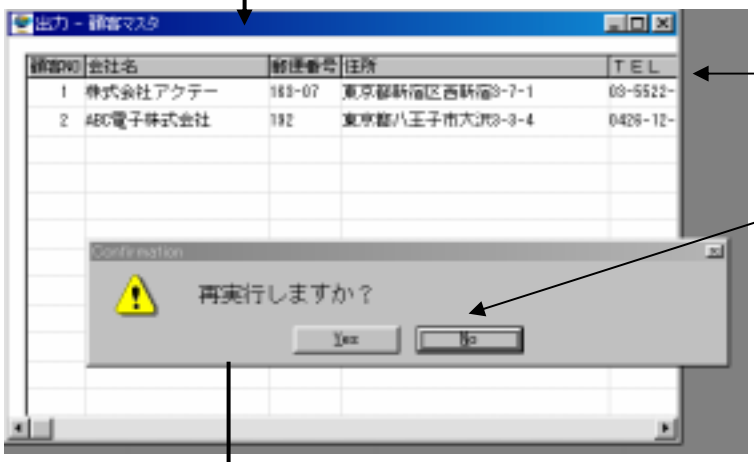
⑨ 出力ファイル名を確認します。今回(「kokyaku.txt」既定値)のままにしておきます。

⑩ [ＯＫ]ボタンを押します。

4）プログラム実行



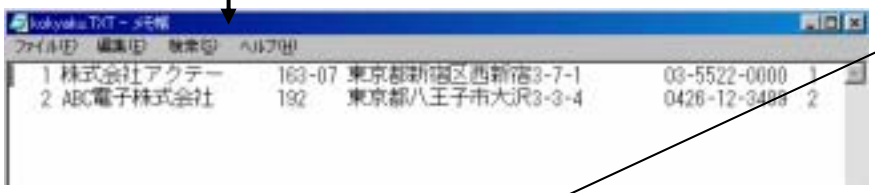
① プログラムリポジトリに戻りプログラム名を「顧客マスタテキスト出力」と変更後プログラムを実行します。
バッチプログラムでは実行確認のダイアログが表示されますので[Yes]ボタンを押します。



② 画面の背景に表示されている画面は処理経過を表示するためのものです。

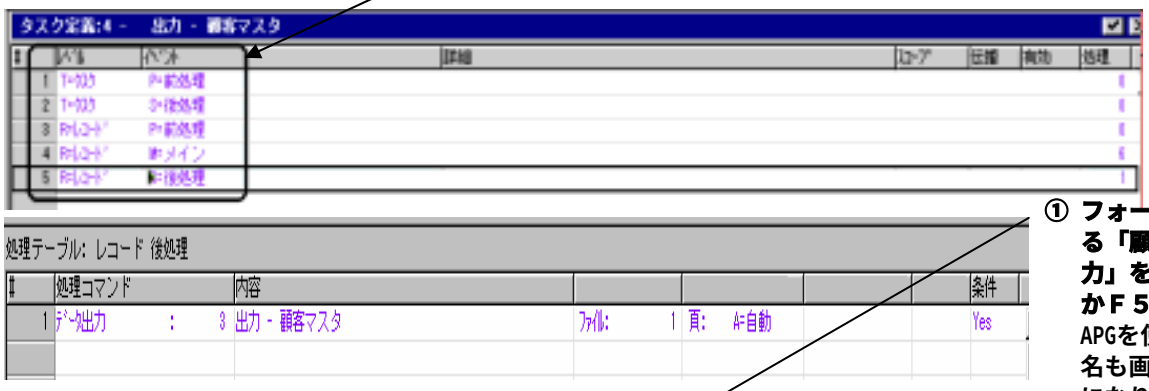
③ バッチプログラムでは再実行を確認するダイアログが表示されますので[No]ボタンを押します。

④「kokyaku.txt」をメモ帳で開いて内容を確認します。Magicをデフォルトでインストールした場合、このファイルが作成されるディレクトリは、C:\Program Files\Magic\Developmentです。



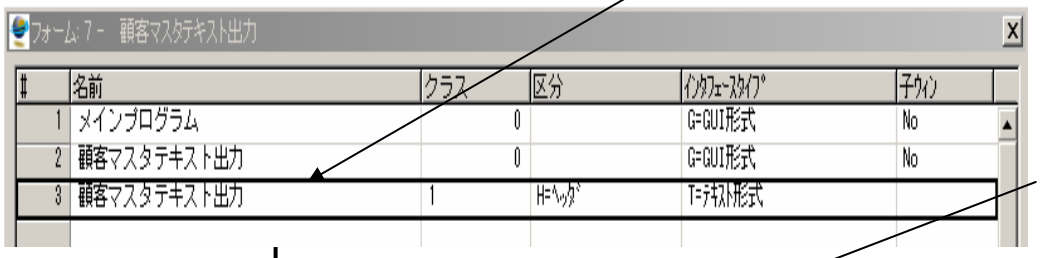
バッチプログラムの中心処理はレコード後処理に記述します。レコードメインではセレクトコマンドで項目が記述されているだけです。

5）プログラム内容の確認



① フォーム一覧で３行目にある「顧客マスタテキスト出力」をダブルクリックするかＦ５を押します。
APGを使うと出力フォーム名も画面フォーム名と同じになります。また明細行なのに「区分」が「H=ヘッダ」になってしまいます。
今回は実用上支障はありませんが注意が必要です。
詳細は有償セミナーを受講してください。

6）プログラム内容の確認



② テキストファイルに固定長フォーマットで出力された出力フォームを確認します。

③ [ESC]キーでフォーム画面に戻ります。

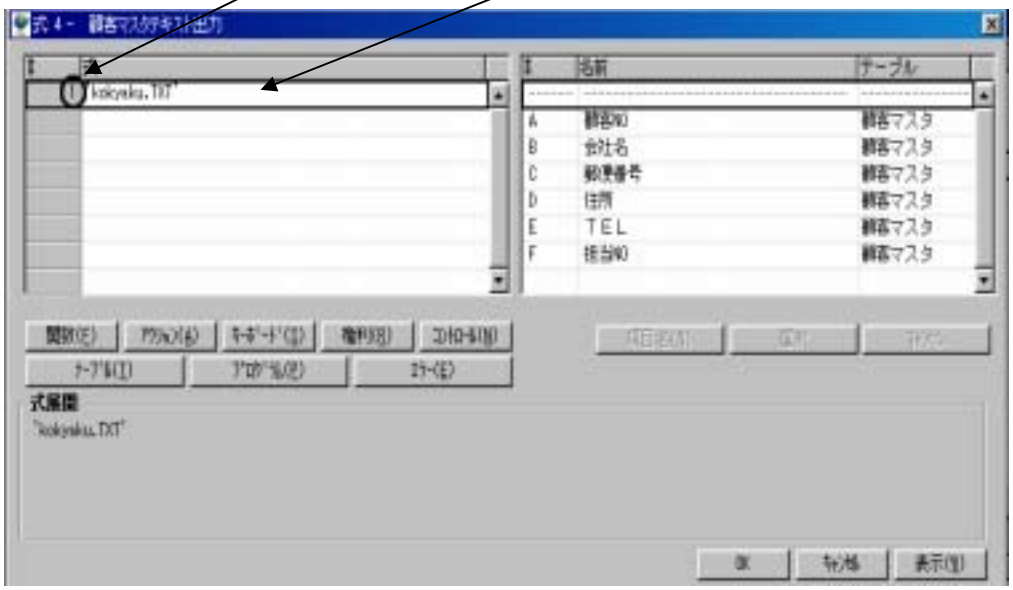
7）出力ファイル名の確認

① タスク環境(K)をクリックします。

② I ：入出力ファイルをクリックします。

③ 式の項目をダブルクリックするかＦ５を押します。
入出力ファイルテーブルはタスク毎にあり主にプリンタとテキストファイルを登録します。
タスクを起動すると始めにこのテーブルのファイルをオープンします。

④ 出力ファイルの物理ファイル名を確認します。
③ の式の値はこの行番号を表しています。



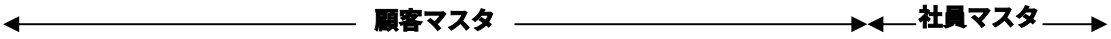
２．　顧客一覧表の印刷

１）機能概要

顧客マスタテーブルと社員マスタテーブルをリンクし､次のフォーマットで印刷します。

【印刷結果】

```
1 株式会社アクテー     163-07   東京都新宿区西新宿3-7-1        03-5522-0000  1  佐藤　勉
2 ABC電子株式会社      192      東京都八王子市大沢3-3-4        0426-12-3489  2  田中　称
```



２）プログラム作成の手順

① プリンタの登録
OSに登録されているプリンタ名（キュー名）を確認し､Magicのプリンタテーブルに登録します。

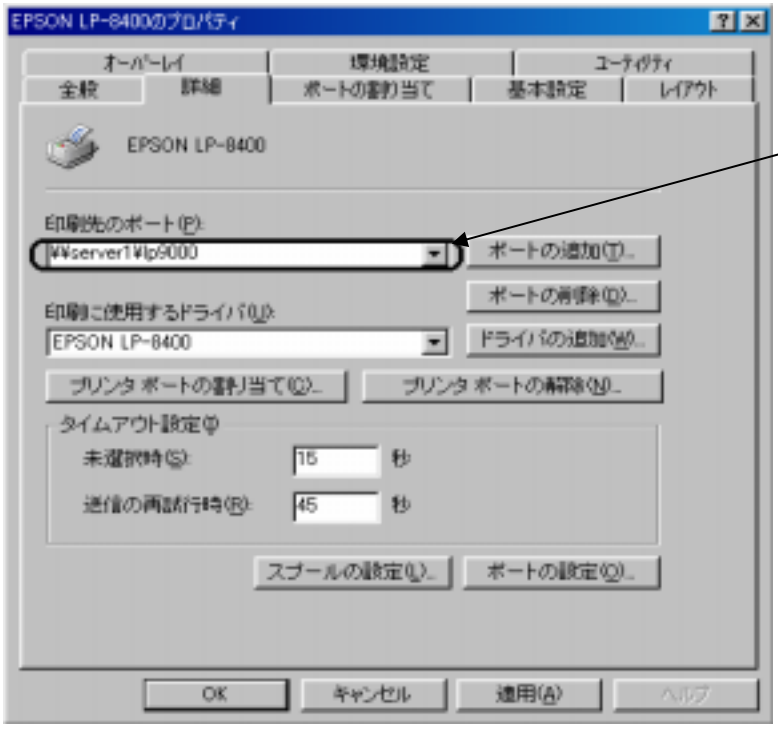
② 顧客マスタテキスト出力プログラムの複写登録
前項のプログラムと似ていますので､それをコピーして流用します。

③ 入出力ファイルテーブルの変更
出力先の設定をテキストファイルからプリンタに変更します。

④ リンクコマンドの追加
担当者名を出力するために社員マスタテーブルをリンクします。

⑤ 出力フォームの修正
④で追加した社員マスタテーブルの「氏名」を出力フォームに追加します。

3）プリンタの登録



① Windowsのコントロールパネルからプリンタを呼び出し、さらに出力に使用するプリンタのプロパティを表示します。

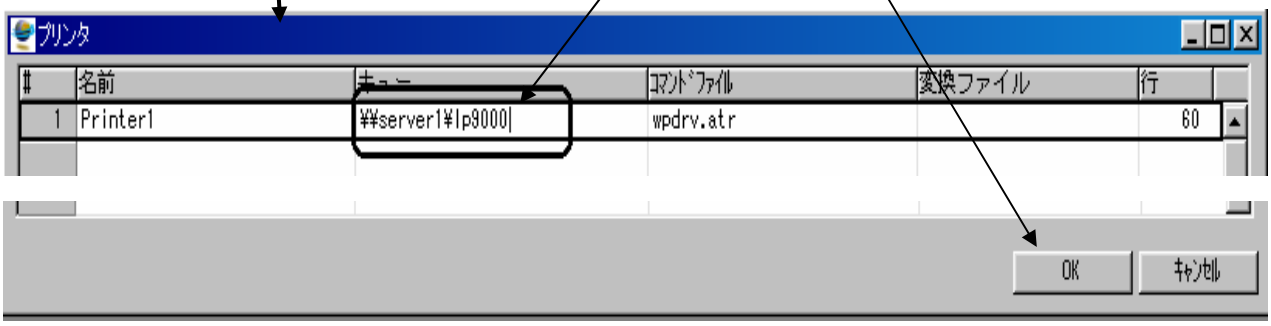
② 「印刷先のポート」に表示されている内容を記録しておきます。

③ Magicに戻り、設定(S)をクリックします。

④ P:プリンタをクリックします。

⑤ プリンタテーブルにはサンプルプリンタが既に登録されていますので、その「キュー」項目を②と同じ内容に変更します。
(Printer1のキュー名を変更します。)

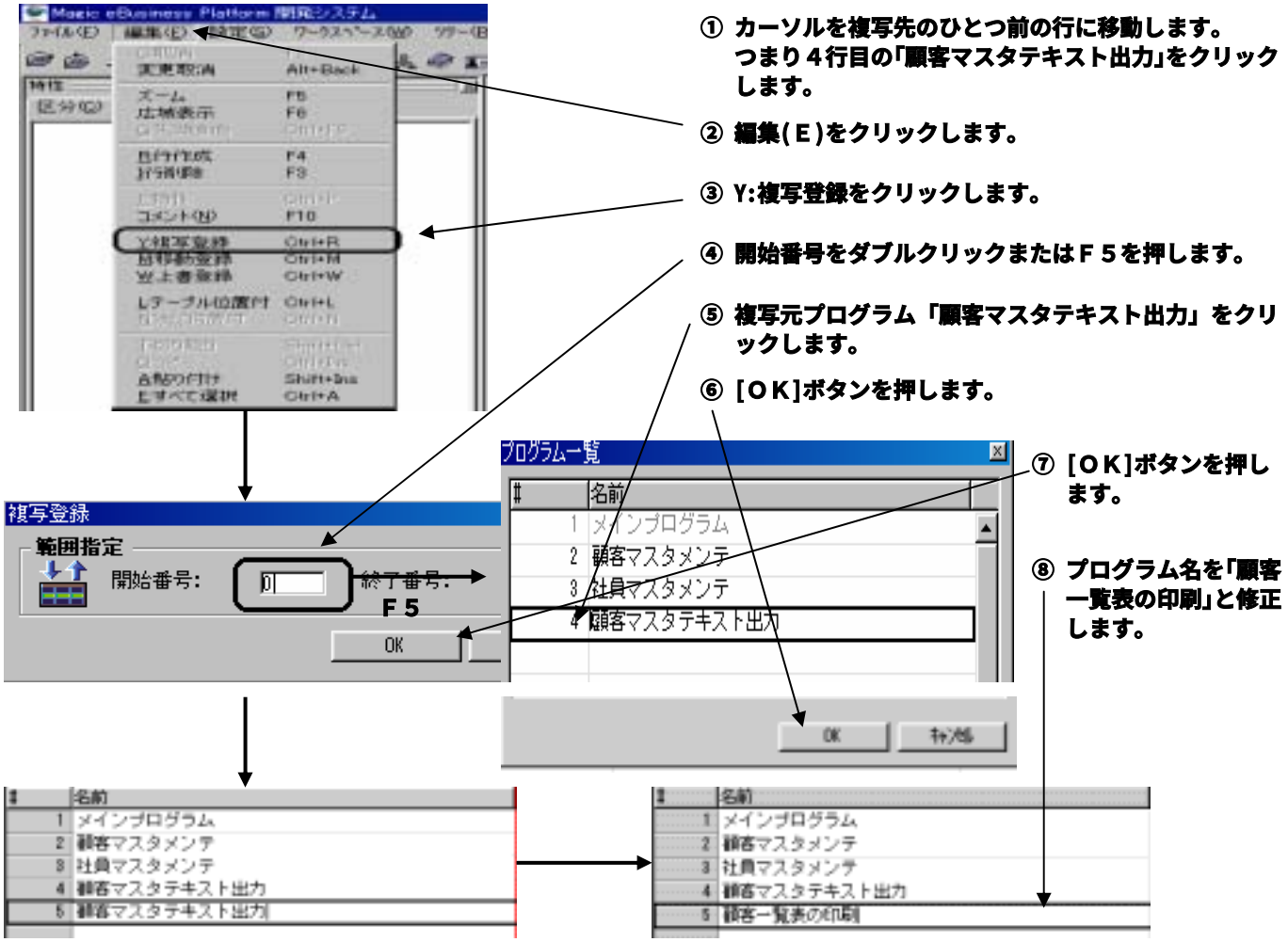
⑥ [OK]ボタンを押します。



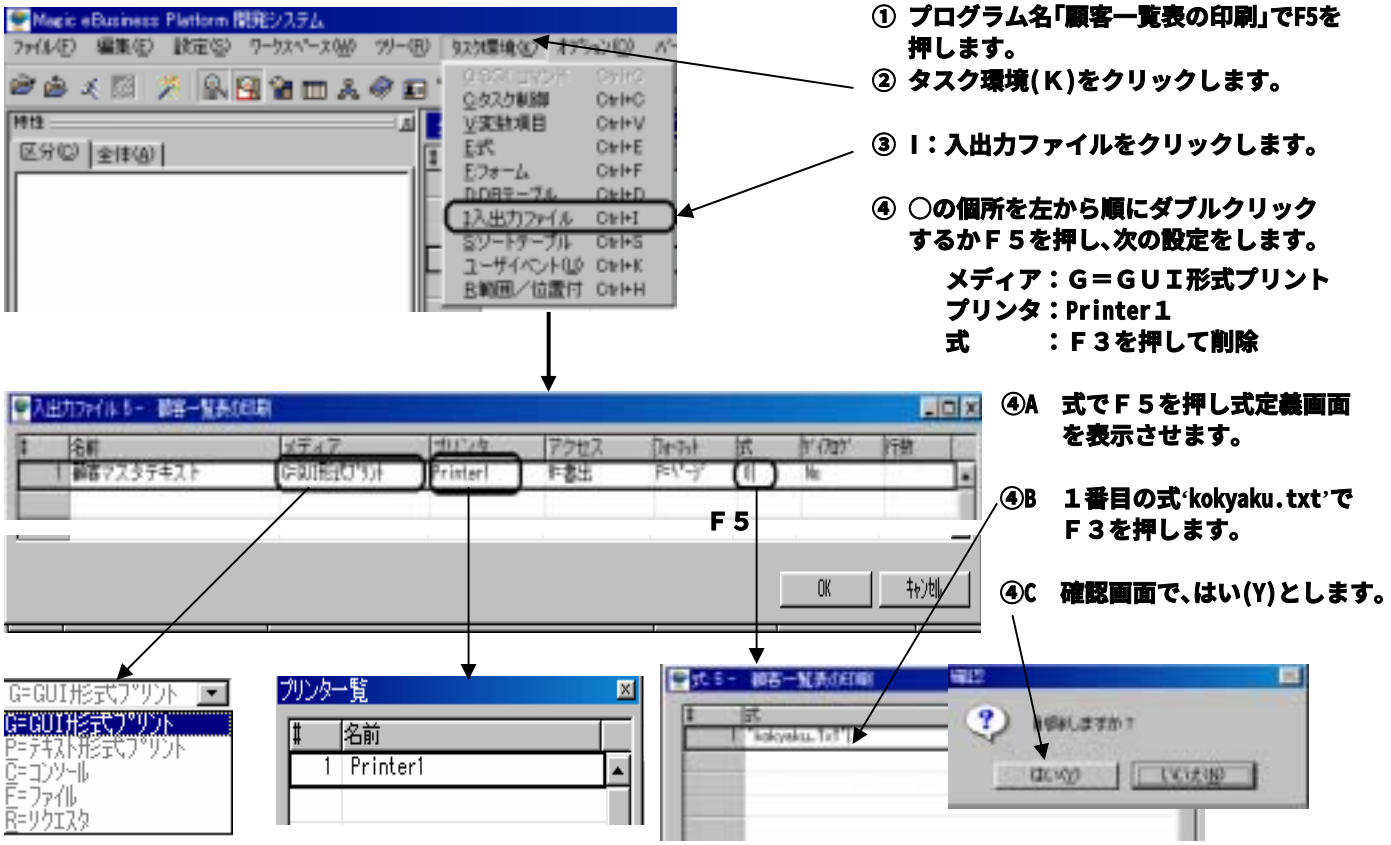
⑦ [ESC]キーを押しプリンタテーブルから抜けます。

⑧ ツールバーのプログラムリポジトリをクリックします。

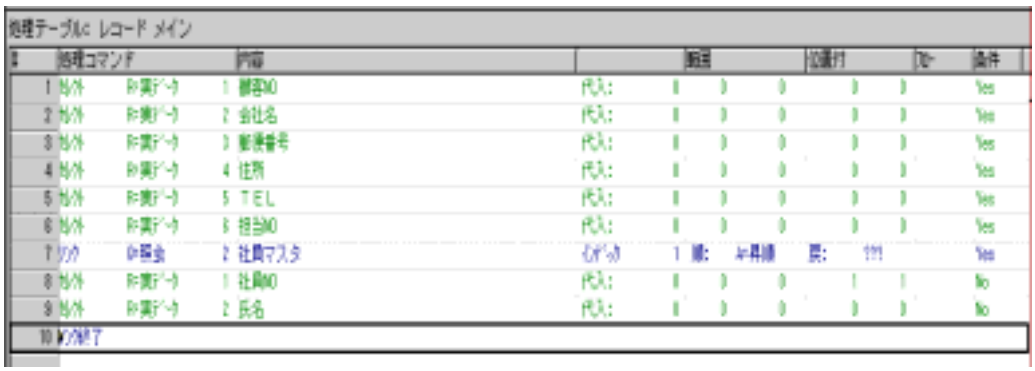
4）顧客マスタテキスト出力プログラムの複写登録



5）入出力ファイルテーブルの変更



6）リンクコマンドの追加

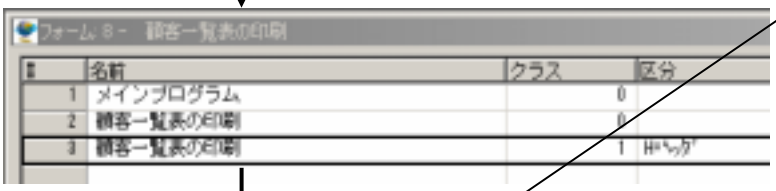


① リンクコマンドを追加します。
ただし、全く同一の設定内容が34、35ページで説明されていますので設定作業はそちらを見て行ってください。

7）出力フォームの修正

① タスク環境(K)をクリックします。

② F：フォームをクリックします。



③ ３行目のフォームの名前を次のように直します。
名前：帳票明細

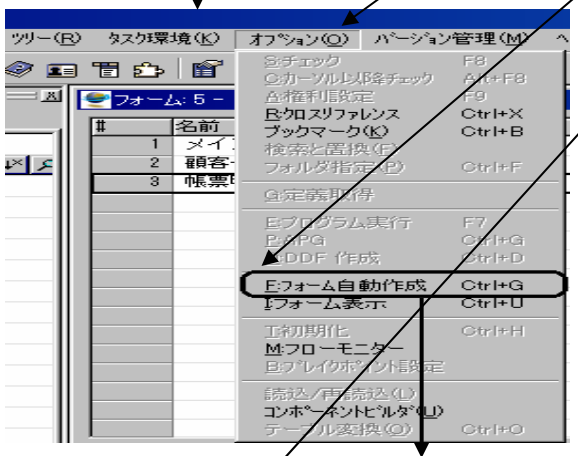
④ オプション(O)をクリックします。

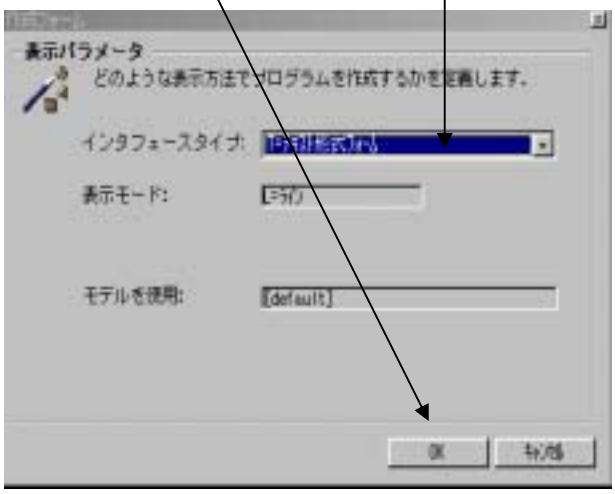
⑤ F：フォームの自動作成をクリックします。

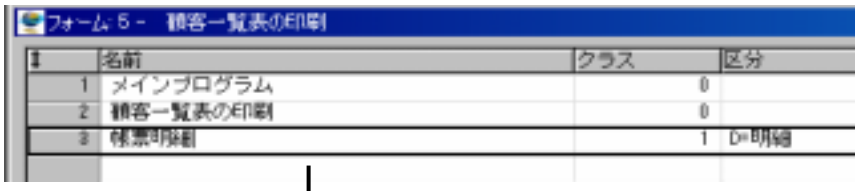
⑥ [はい]ボタンを押します。

⑦ T＝テキスト形式を選択します。

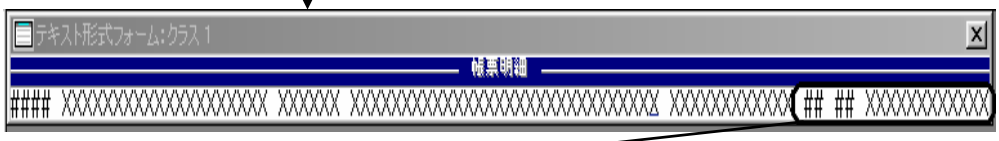
⑧ [ＯＫ]ボタンを押します。



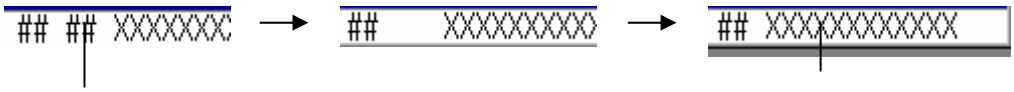




① 名前の項目をダブルクリック
またはＦ５を押します。



フォーム自動作成を使ったので「社員NO」が余分に表示されています。これを削除し､「氏名」を左に寄せます。

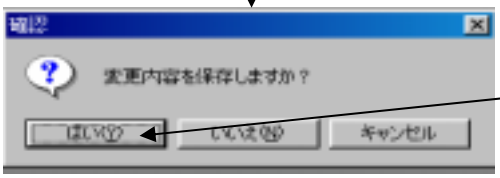


②「社員NO」（右から２番目）の中心を注意深くクリックします。

③ ＤＥＬを押し削除します。

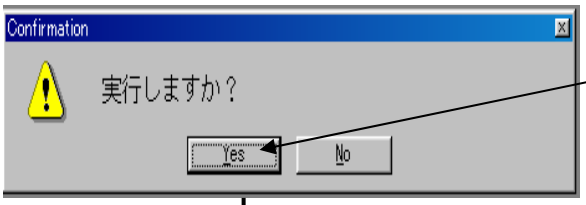
④ 「氏名」（右端）の中心を注意深くクリックした状態のまま、左に移動させます。

⑤ 修正を終えたらＥＳＣを押します

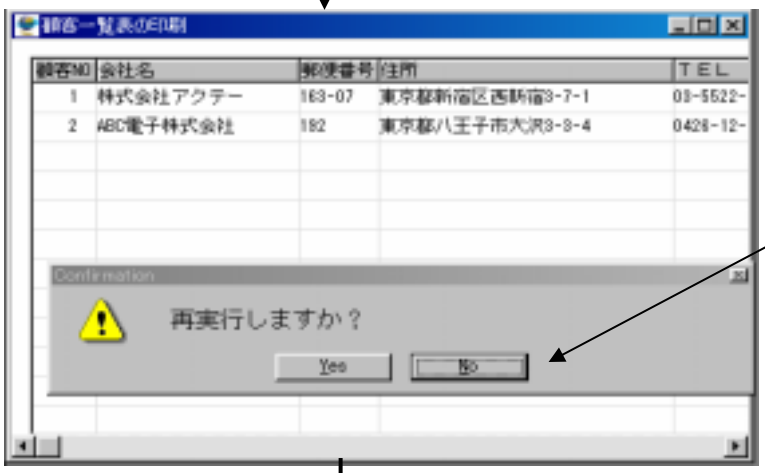


⑥ [はい]ボタンを押します。

8）プログラムの実行



① プログラムリポジトリに戻りプログラムを実行します。バッチプログラムでは実行確認のダイアログが表示されるので[Yes]ボタンを押します。



② 画面の背景に表示されている画面は処理経過を表示するためのものです。

③ バッチプログラムでは再実行を確認するダイアログが表示されるので[No]ボタンを押します。

1	株式会社アクテー	163-07	東京都新宿区西新宿3-7-1	03-5522-0000	1	佐藤　勉
2	ABC電子株式会社	192	東京都八王子市大沢3-3-4	0426-12-3489	2	田中　称

３．　担当者毎に顧客件数を集計して印刷（グループ処理）

１）機能概要

顧客マスタテーブルと社員マスタテーブルをリンクし、次のフォーマットで印刷します。

【印刷結果】

1	株式会社アクテー	163-07	東京都新宿区西新宿3-7-1	03-5522-0000	1　佐藤　勉	グループ
2	株式会社フラン	140	東京都千代田区五反田2-6	03-3578-6666	1　佐藤　勉	
					件数：　2	
2	ABC電子株式会社	192	東京都八王子市大沢3-3-4	0426-12-3489	2　田中　称	グループ
4	株式会社エストロ	684	大阪府北大阪市町7-3	0467-68-4432	2　田中　称	
					件数　2	

＊この例は、データをさらに登録してから実行しています。

インデックス　：担当NO・顧客NO順
グループ項目　：担当NO

２）プログラム作成手順

① 顧客一覧表の印刷プログラムの複製登録
前項のプログラムと似ていますので、それをコピーして流用します。

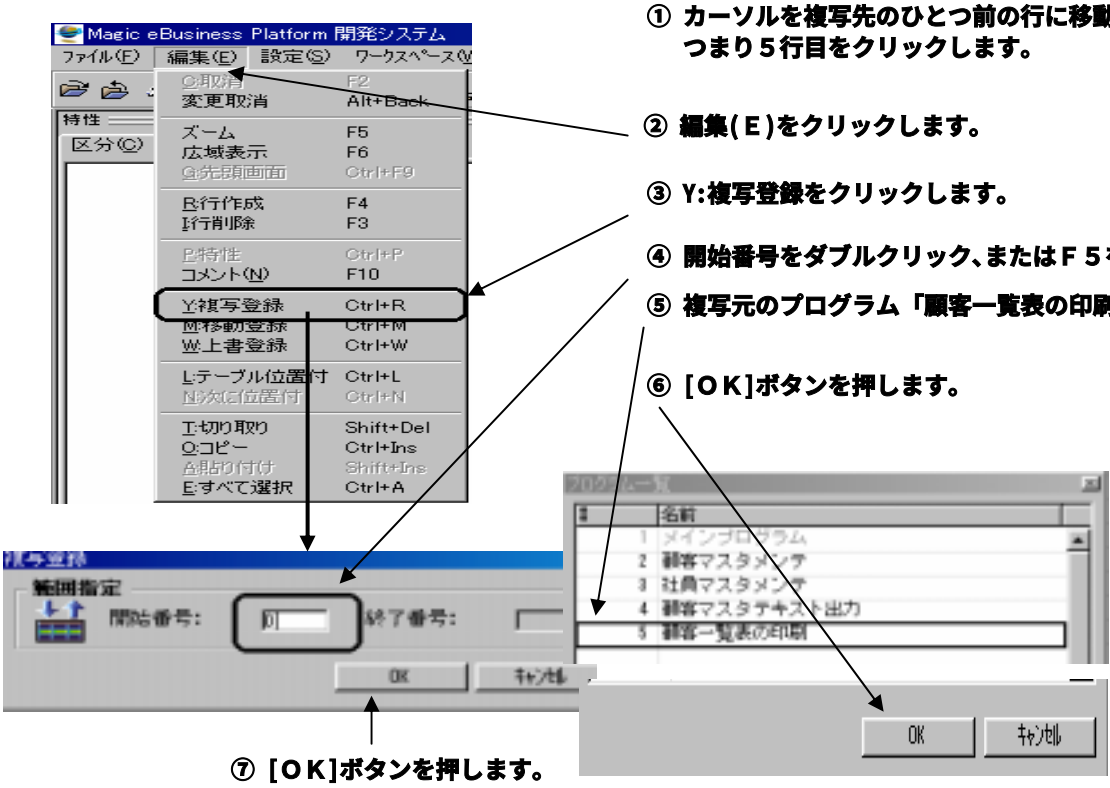
② インデックスの変更
データを取り出すインデックスを「担当NO・顧客NO順」に変更します。

③ 変数の追加
顧客件数を集計する変数「V．件数」を追加します。
毎レコードで**V．件数＝V．件数＋１**を計算します。

④ グループ項目を追加
「担当NO」をグループ項目に設定します。

⑤ グループ処理の追加
グループ処理が発生したら「V．件数」を印刷出力し、その後その値をゼロクリアします。

3）顧客一覧表の印刷プログラムの複写登録



① カーソルを複写先のひとつ前の行に移動します。
つまり５行目をクリックします。

② 編集(Ｅ)をクリックします。

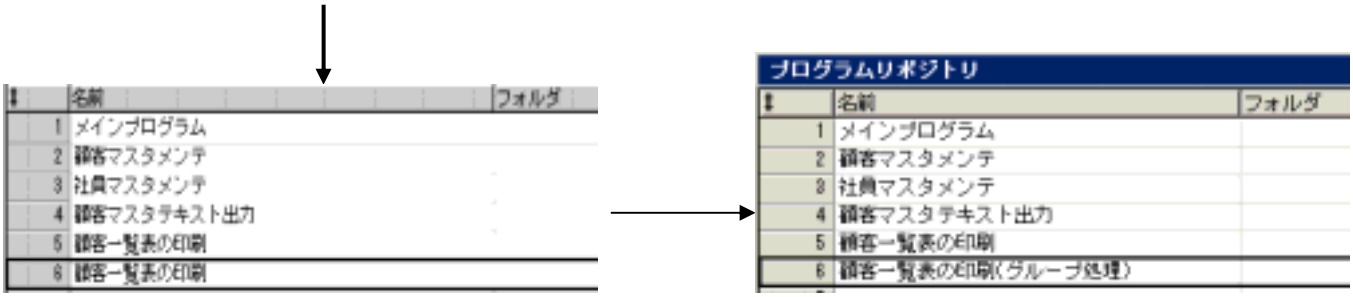
③ Y:複写登録をクリックします。

④ 開始番号をダブルクリック、またはＦ５を押します。

⑤ 複写元のプログラム「顧客一覧表の印刷」をクリックします。

⑥ [ＯＫ]ボタンを押します。

⑦ [ＯＫ]ボタンを押します。



⑧ プログラム名を「顧客一覧表の印刷(グループ処理)」と修正します。

4）インデックスの変更

① プログラム名「顧客一覧表の印刷(グループ処理)」でＦ５を押します。

② Ctrl+Pでタスク特性画面を出します。

③ インデックスをダブルクリックするかＦ５を押します。

④ インデックス一覧から２番目のインデックス「担当NO・顧客NO順」をクリックします。

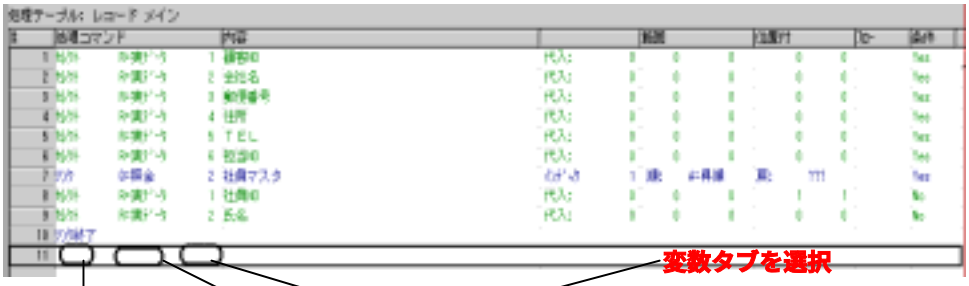
⑤ [選択]ボタンを押します。

⑥ [ＯＫ]ボタンを押します。

5）変数「V．件数」の追加

① 画面の表示内容が異なる場合は処理レベルテーブルのレコードメインをクリックします。

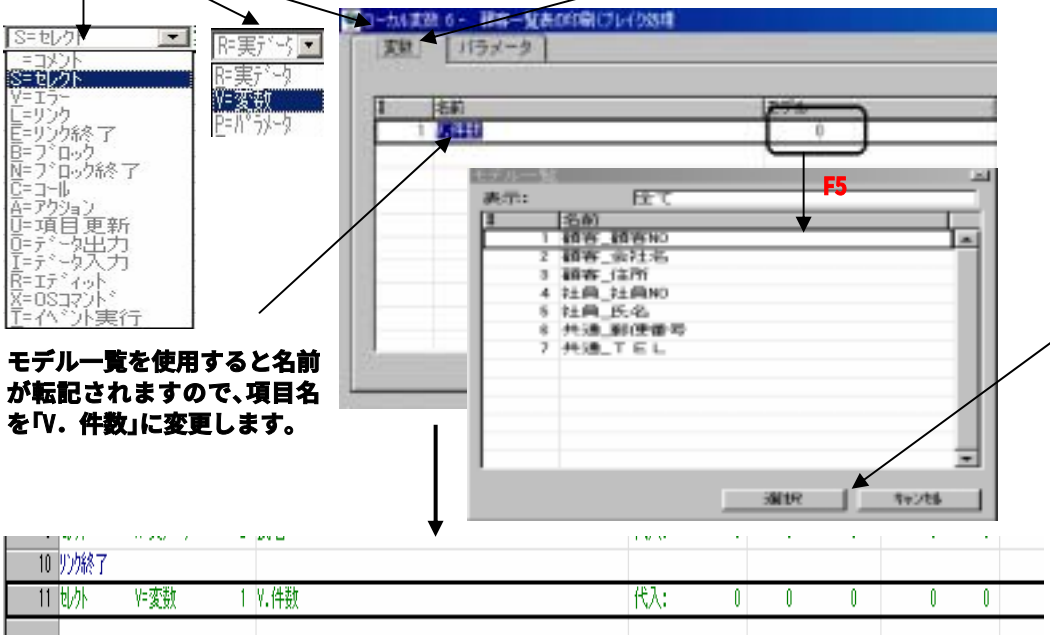
② レコードメインの最終行をクリックします。



③ Ｆ４（行作成キー）を押しその下にセレクトコマンドを登録するための空白行を作成します。

④ 空白行の○の個所を左から順にダブルクリックするかＦ５を押し、次の設定をします。

コマンド：S＝セレクト
ｾﾚｸﾄﾀｲﾌﾟ：V＝変数
名前：V．件数
モデル：顧客_顧客NO



モデル一覧を使用すると名前が転記されますので、項目名を「V．件数」に変更します。

⑤ [選択]ボタンを押します。

⑥ [ESC]キーを押します。

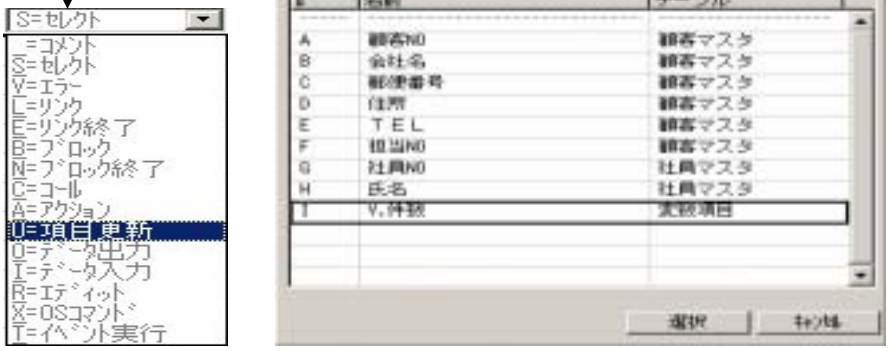
[ESC]キーを押すことにより画面がレコードメイン定義まで戻ってきます。

6）V．件数＝V．件数＋１の計算処理の追加

① 処理レベルテーブルのレコード後処理をクリックします。

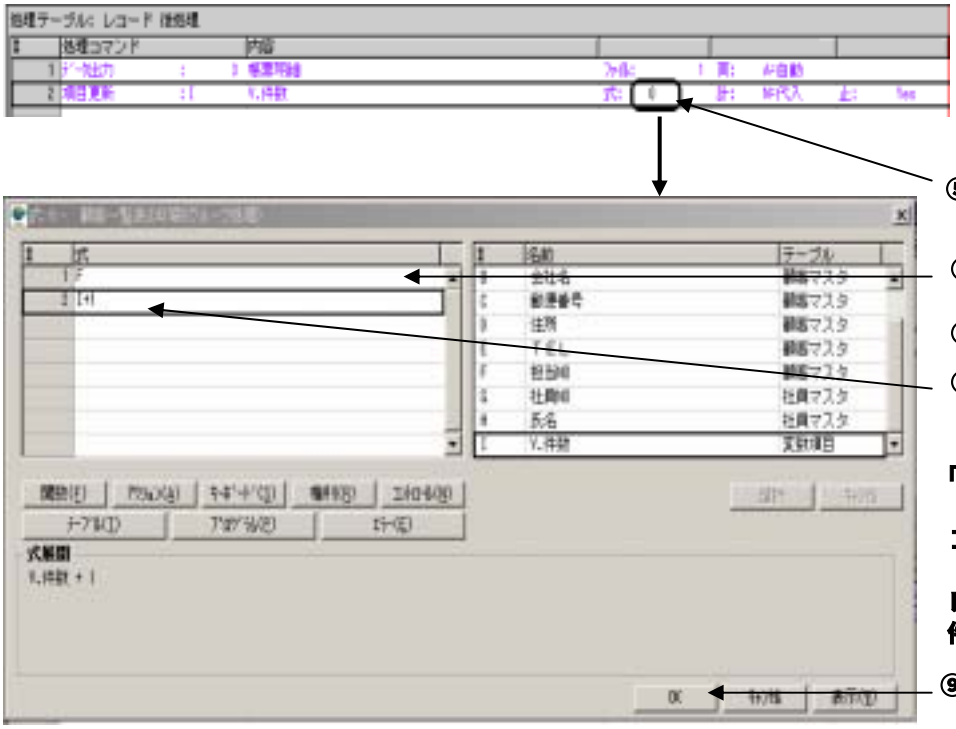
② 最終行をクリックします。

③ Ｆ４（行作成キー）を押し、その下にセレクトコマンドを登録するための空白行を作成します。



④ 空白行の○の個所を左から順にダブルクリックするかＦ５を押し、次の設定をします。

コマンド：U＝項目更新
項目　　：V．件数



⑤ 式でダブルクリックするか
F５を押します。

⑥ 一行目をクリックします。

⑦ Ｆ４を押して行を作成します。

⑧ 「I＋１」を入力します。

「I＋１」は半角入力してください。

Ｉは､V.件数を意味します。

レコード後処理を通過するたびにV.
件数に１加えるという意味です。

⑨ [ＯＫ]ボタンを押します。

7）グループ項目を設定

① タスク後処理の所で
Ｆ４を押すとグループ
レベルが追加されます。

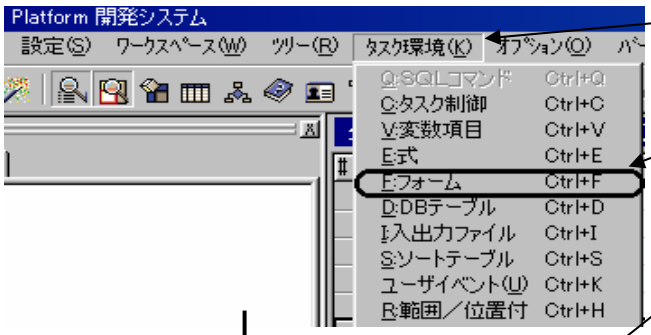
② 項目の所でＦ５を押します。

③ 項目一覧から担当NOをクリックします。

④ [選択]ボタンを押します。

⑤ 同様にグループ後処理も作ります。

8）グループ時の印刷フォームの作成

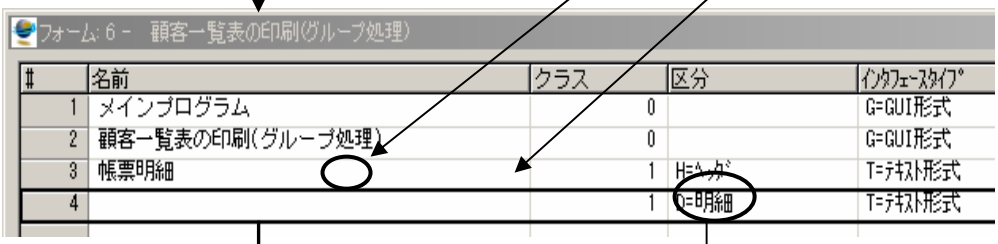


① タスク環境(K)をクリックします。

② F：フォームをクリックします。

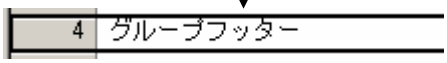
③ 件数を出力する出力フォームを作成するので、まず３行目の帳票明細をクリックします。

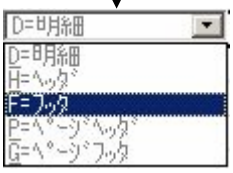
④ F４を押し一行作成します。



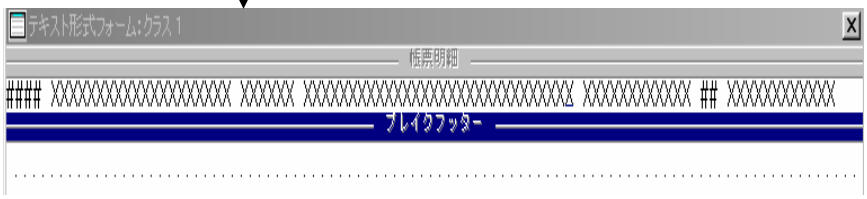
⑤ 空白行の○の個所を左から順にダブルクリックするかＦ５を押し、次の設定をします。

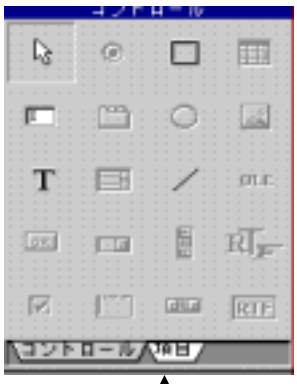
名前 ：「グループフッター」
区分 ： F＝フッタ

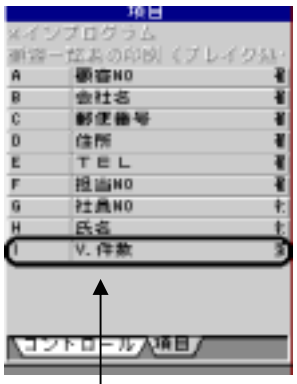


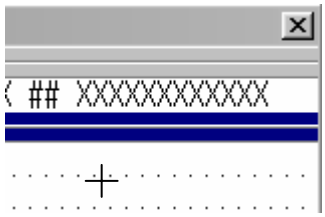


⑥ 「グループフッター」をダブルクリックするかF５を押します。







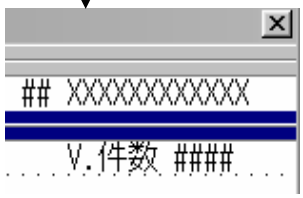


⑦ 項目タブをクリックします。

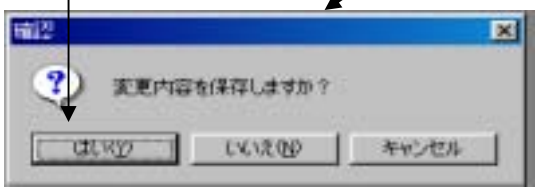
⑧ 「V．件数」をクリックします。

⑨ 図の場所にマウスを移動させクリック（項目は＋マークの場所に現れる）

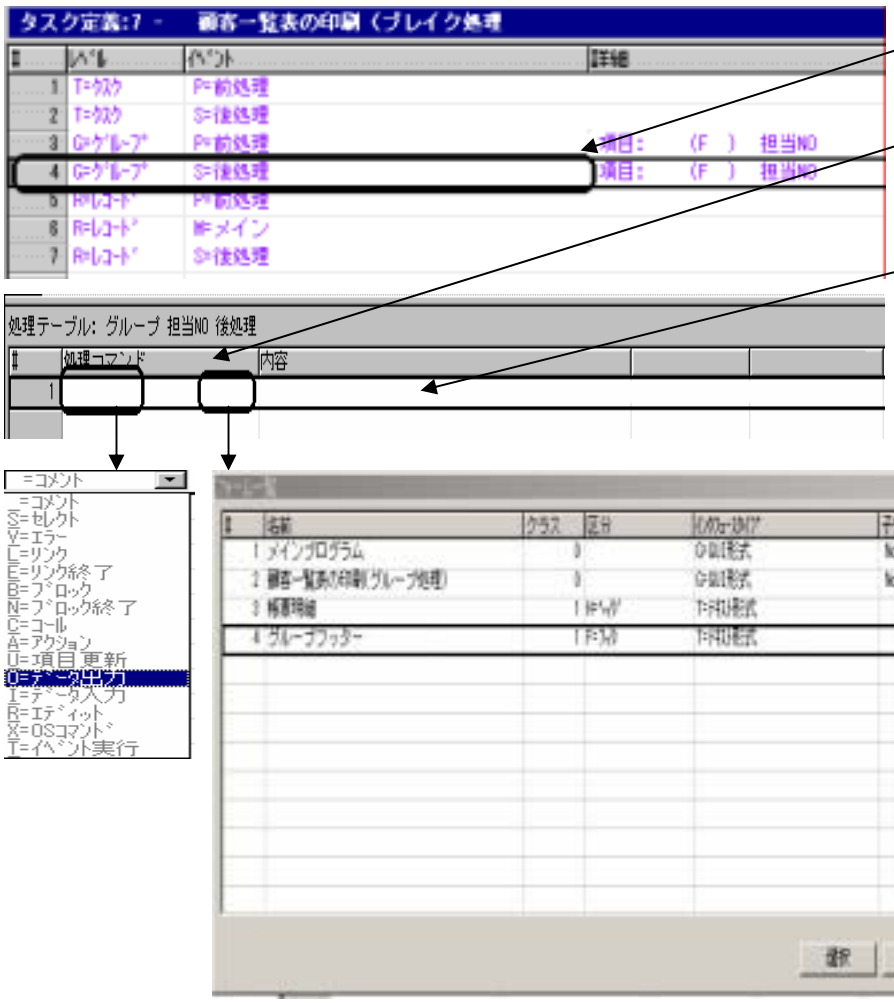
⑩ [ESC]キーを押します



⑪ [はい]ボタンを押します。



９）グループ後処理に件数の印刷処理を追加



① 処理レベルテーブルのグループ後処理をクリックします。

② 第一行目に空白行を作る場合には見出し行をクリックし、カーソルをその前の位置に移動させておきます。

③ Ｆ４キー(行作成キー)を押し、その下にデータ出力コマンドを記述するための空白行を作成します。

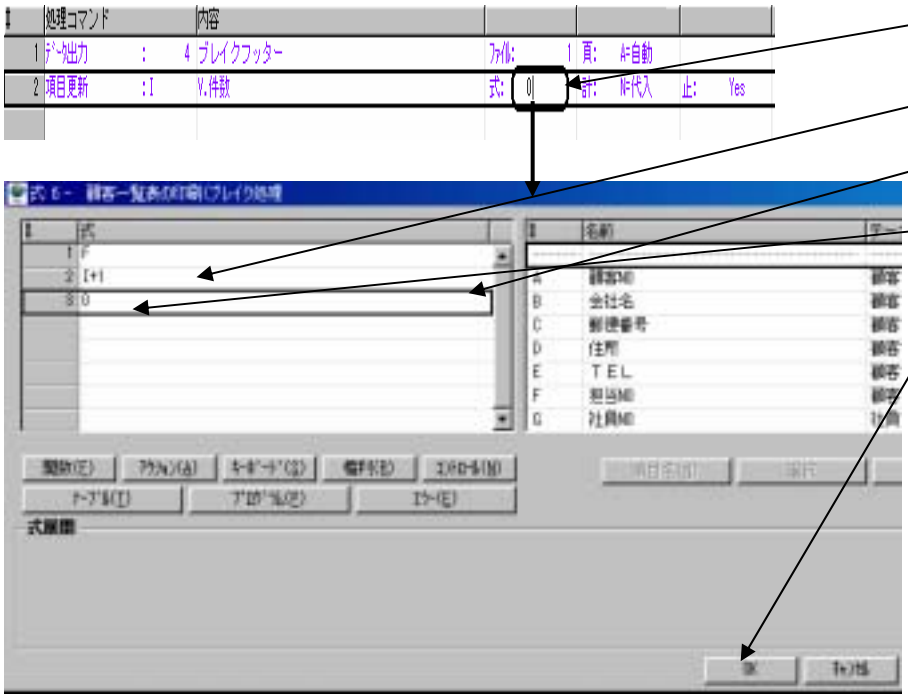
コマンド：O:データ出力
フォーム：グループフッター

１０）Ｖ．件数をゼロクリアします

① Ｆ４キー(行作成キー)を押し、その下に項目更新コマンドを記述するための空白行を作成します。

② 空白行の○の個所を左から順にダブルクリックするかＦ５を押し、次の設定をします。

コマンド：U:項目更新
更新先：Ｖ．件数



③ 式でダブルクリックするか
F５を押します。

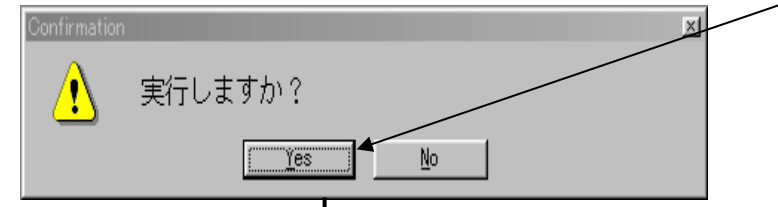
④ ２行目をクリックします。

⑤ F４を押して行を作成します。

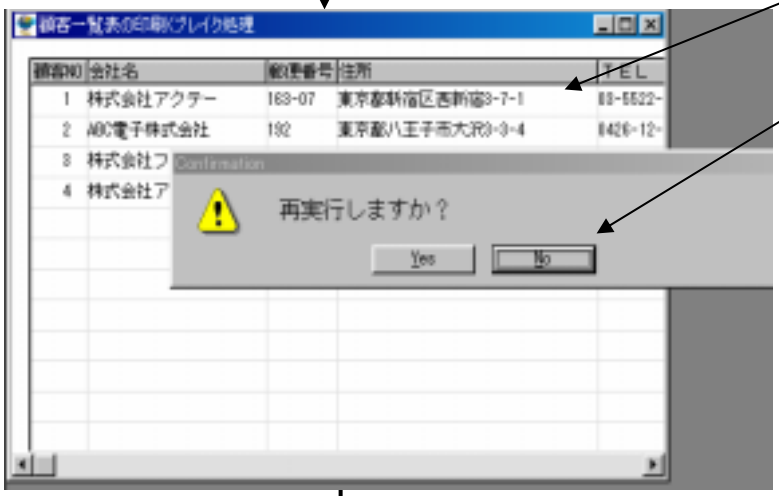
⑥ ０（数値のゼロ）を入力します。

⑦ [ＯＫ]ボタンを押します。

１０）プログラム実行



① プログラムリポジトリに戻りプログラムを実行します。バッチプログラムでは実行確認のダイアログが表示されますので[Yes]ボタンを押します。



② 画面の背景に表示されている画面は処理経過を表示するためのものです。

③ バッチプログラムでは再実行を確認するダイアログが表示されますので[No]ボタン を押します。

1 株式会社アクテー	163-07	東京都新宿区西新宿3-7-1	03-5522-0000	1	佐藤　勉	グループ
3 株式会社フラン	140	東京都千代田区五反田2-6	03-3578-6666	1	佐藤　勉	
2 ABC電子株式会社	192	東京都八王子市大沢3-3-4	0426-12-3489	2	田中　称	グループ
4 株式会社エストロ	684	大阪府北大阪市町7-3	0467-68-4432	2	田中　称	
					件数：　2	

第６章　　モデルリポジトリの
メンテナンス機能

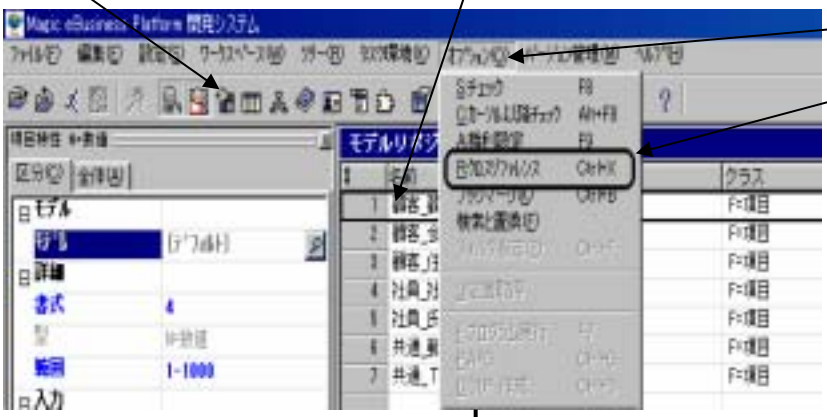
この章では、モデルリポジトリのメンテナンス機能について説明します。

１．クロスリファレンス
２．モデルリポジトリの変更

１．　クロスリファレンス

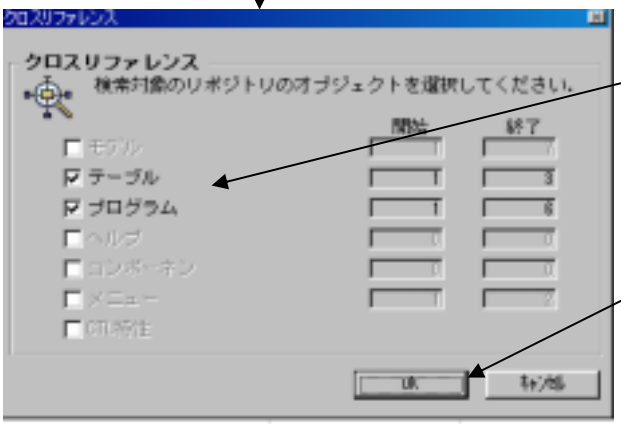
クロスリファレンスとは、モデル・テーブル・プログラムの３つのリポジトリで定義されているそれぞれの情報の参照関係を表したものです。開発言語等でもいろいろなツールを使ってリファレンスを取ることはできるでしょうが、Magicの場合それらを瞬時に取り出せることが大きな特徴です。瞬時にできると言うことは全ての情報が常に一元管理されていることの表れでもあります。
ここでは、モデルリポジトリの「顧客_顧客NO」を使ってクロスリファレンスの取り方を説明しますが、この操作はモデルリポジトリに限られたものではなく、テーブルリポジトリのファイル名、項目、インデックスやプログラムリポジトリのプログラム名等から実行することも出来ます。

① モデルリポジトリをクリックします。② 顧客_顧客NOをクリックします。



③ オプション(O)をクリックします。

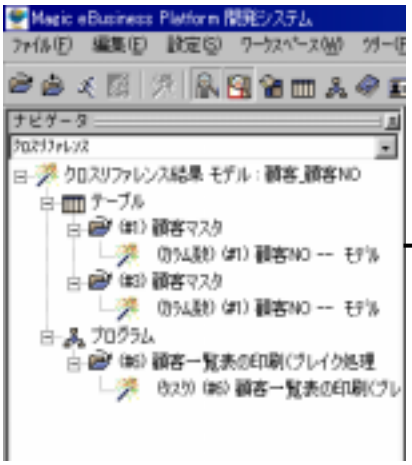
④ R:クロスリファレンスをクリックします。



⑤ 必要な個所にチェックをいれます。

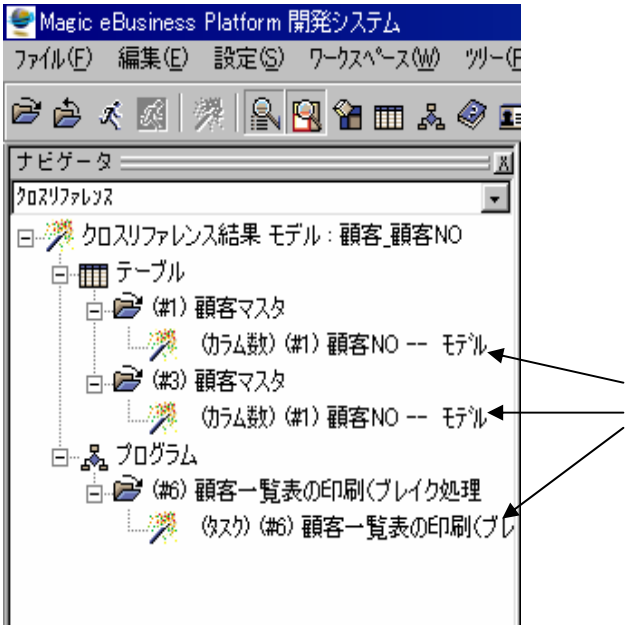
⑥ [ＯＫ]ボタンを押します。

⑦ タイプ名「顧客_顧客NO」を使っている
テーブル名が表示されます。



⑧ 結果ツリーの末端をクリックすると
使用されている場所が表示されます。

クロスリファレンスのメンテナンス方法



ここをクリックすると目的の場所にジャンプします。

必要なければ個別または全部を削除できます。（Ｆ３キーを押します）
※プログラムには影響ありません。

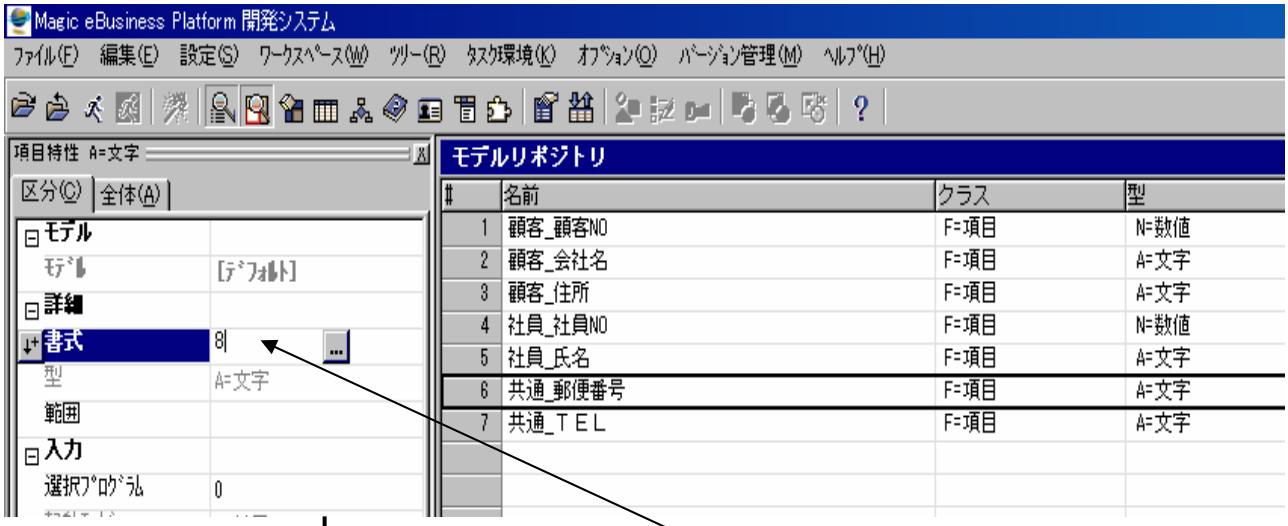
◎クロスリファレンス結果の最大登録数を変更します。（設定－動作環境）

ここで最大数を設定します。

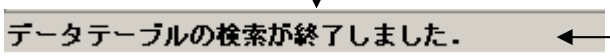
２．　モデルリポジトリの変更

はじめの仕様設計が完璧であれば、それ以降何の仕様変更もなくシステムを開発し本稼動させることが出来ますが、実際には度重なる変更によりはじめの仕様から大きく変わってしまうことさえあります。要因もさまざまですが社会の変革により避けては通れないこともあります。例えば郵便番号です。古いシステムでは６桁の文字列（‘－’を含む）ですが、１９９８年２月から２桁増えました。
従来型の開発言語等の場合、正しいクロスリファレンスがあっても、それを修正するのは人手に頼っていました。やっと作業が完了し、モデル変換も終わったと思ったら一世代前のものだったということもありえます。前項でMagicではクロスリファレンス情報を常に一元管理していると述べました。つまり、Magicならモデルリポジトリを修正するだけで基本的な変更はいとも簡単に終わってしまいます。

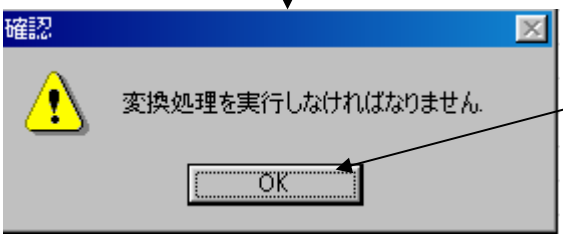
ここでは、モデルリポジトリの「共通_郵便番号」を実際に６桁から８桁に修正し、テーブル変換とプログラム変換を確認します（郵便番号コード体系を変換するわけではありません。また画面や出力フォームもレイアウトの兼ね合いがありますので手作業で直す必要があります）。



① モデルリポジトリの「共通_郵便番号」の書式の値を６桁から



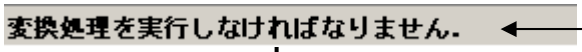
② [ESC]キーを２度押すと、「データテーブルの検索が終了しました。」と表示後「変換処理を実行しなければなりません。」と確認画面が表示されます。



③ [ＯＫ]ボタンを押します。

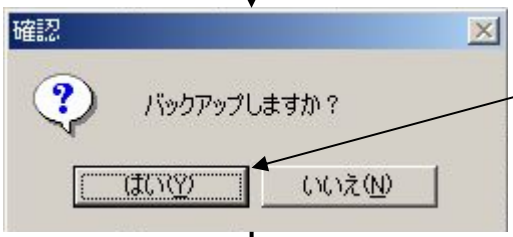
＊データ変換やプログラム変換は以下の操作により変換処理されます。

④ 各テーブル上でAPG([Ctl]+G)を実行



⑤「変換処理を実行しなければなりません。」
と表示されます。

⑥ [はい（Y）]ボタンを押します。



⑦ [はい（Y）　]ボタンを押します。

⑧ [ＯＫ]ボタンを押します。

１．テーブルフォーマット変換
２．データ変換
３．プログラム変換

第７章　　ブラウザ・クライアント
のプログラム作成

この章では、ブラウザ・クライアントの作成方法について説明します。

１．ブラウザクライアントプログラムの作成

１．　ブラウザ・クライアントプログラムの作成

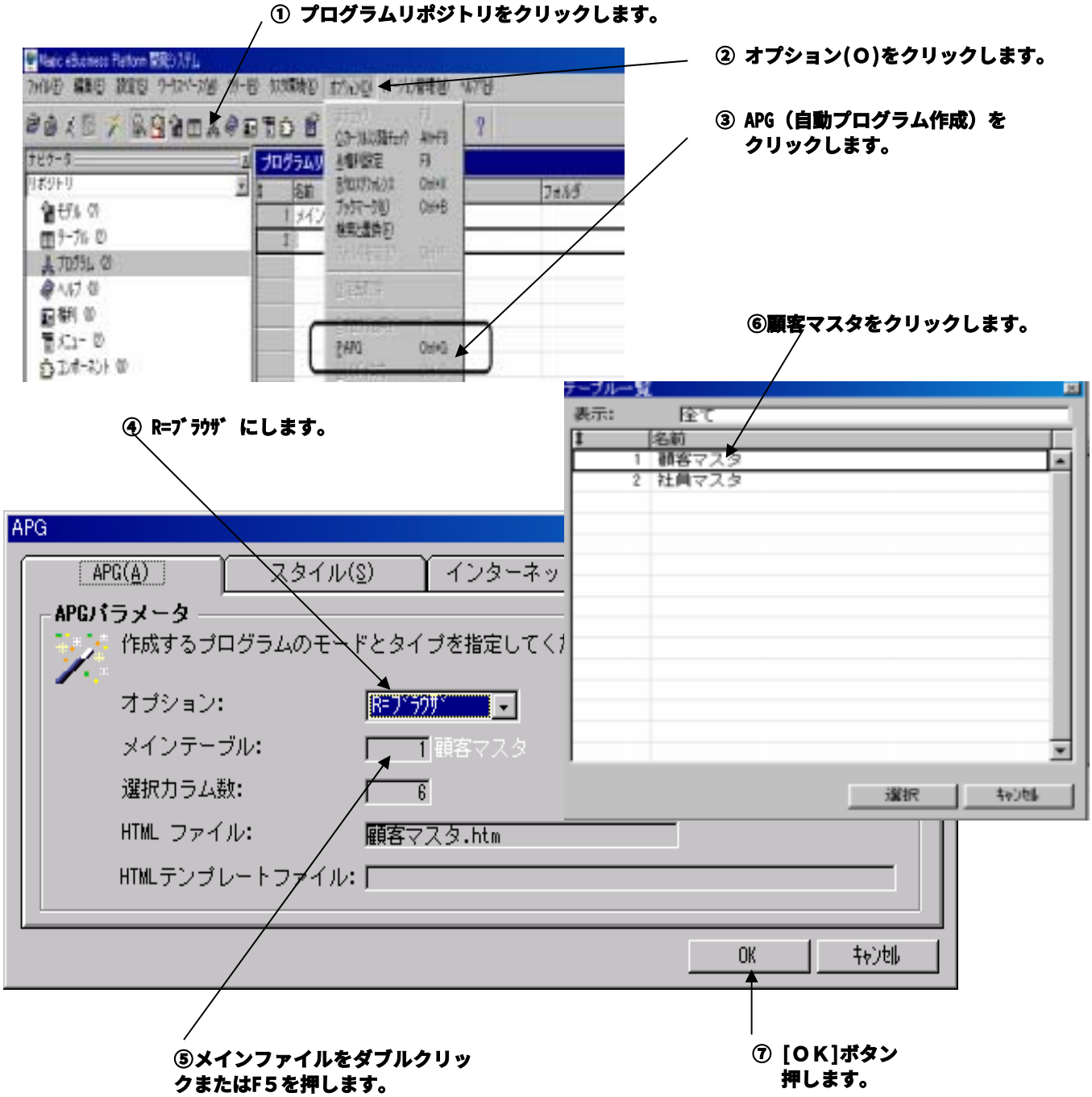
１）ブラウザ・クライアントとは

ブラウザクライアントとは、ブラウザで作られたアプリケーションをあたかもクライアント／サーバのように動作させることが可能な仕組みです。
手作業で一つ一つ作ることも出来ますが、ここではAPG（自動プログラム作成）で作ります。

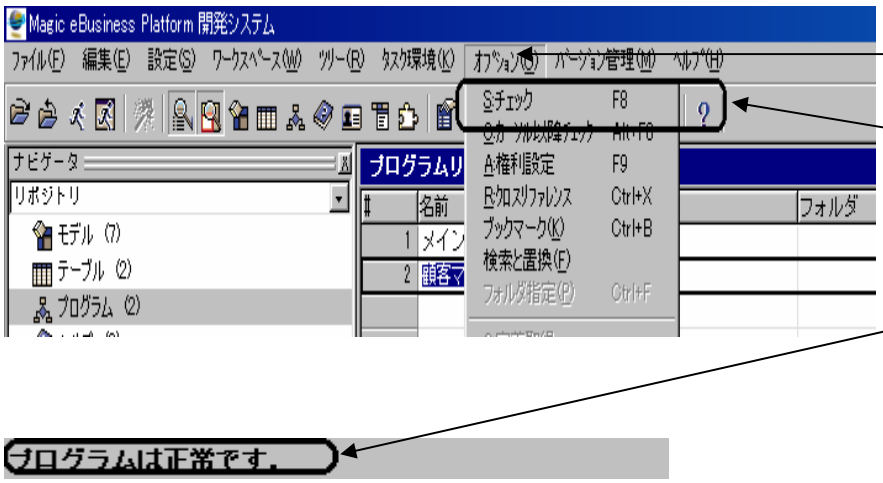
２）ブラウザ・クライアントプログラムの作成手順

① APGによりプログラムを作成します。
② プログラムのチェックをします。
③ 実行（開発モードのまま）します。
④ブラウザ上で修正できるようにします。

３）APGの実行



4）プログラムのチェック

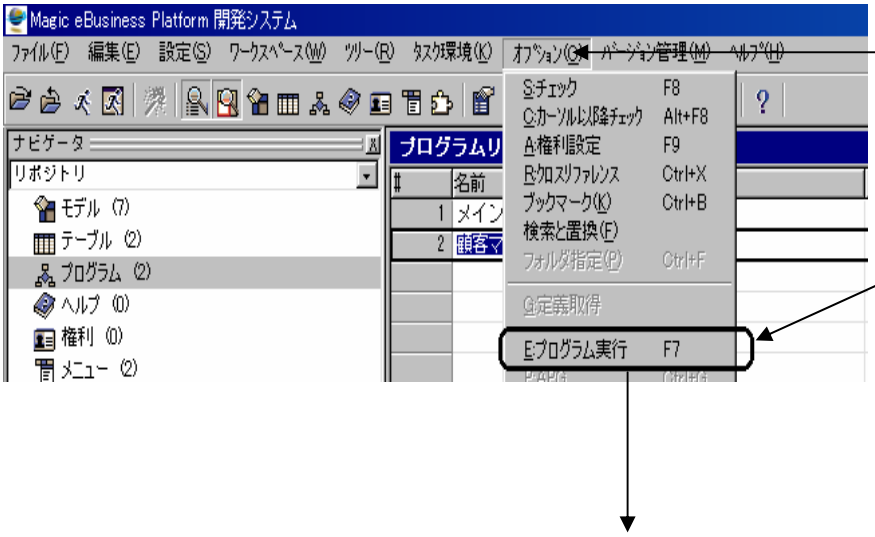


① オプション(O)をクリックします。

② S:チェックをクリックします。

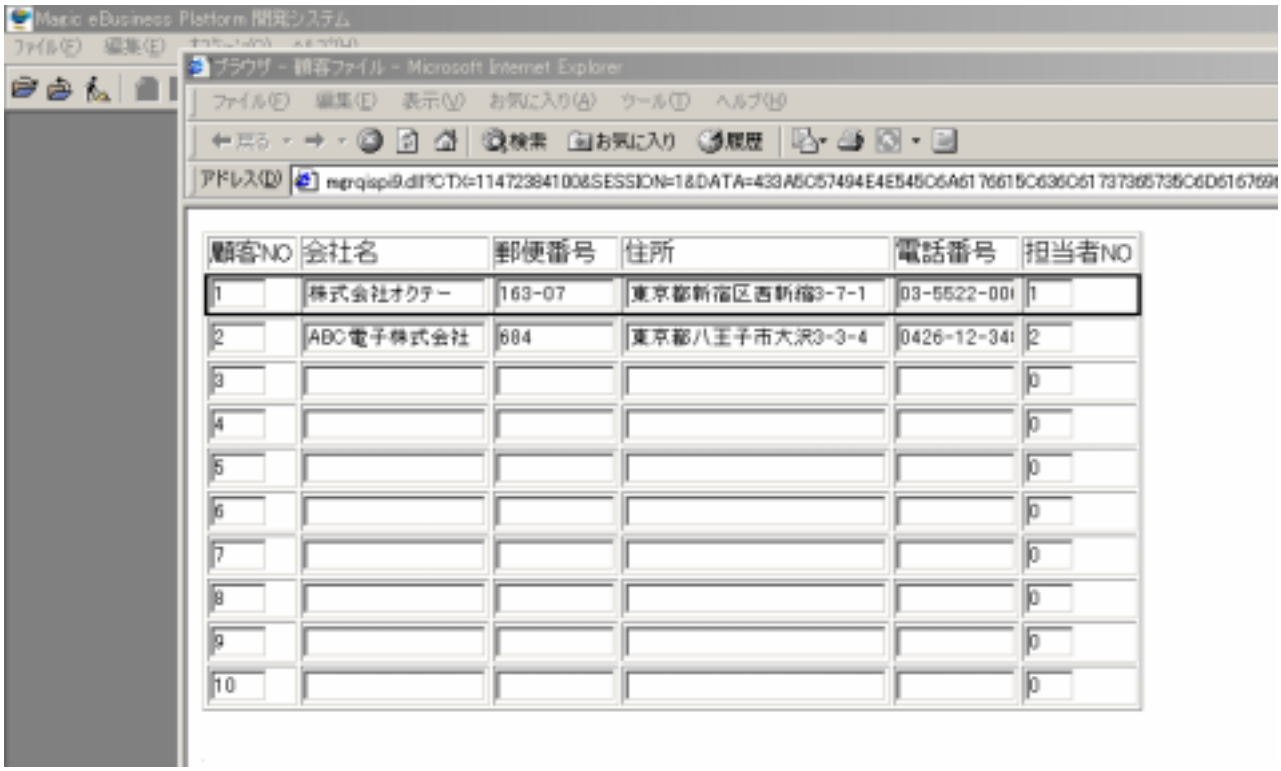
③ プログラム定義内容が正しければ「プログラムは正常です」と表示され、間違っていればその個所の定義画面が表示されます。

5）開発モードのまま実行



① オプション(O)をクリックします。

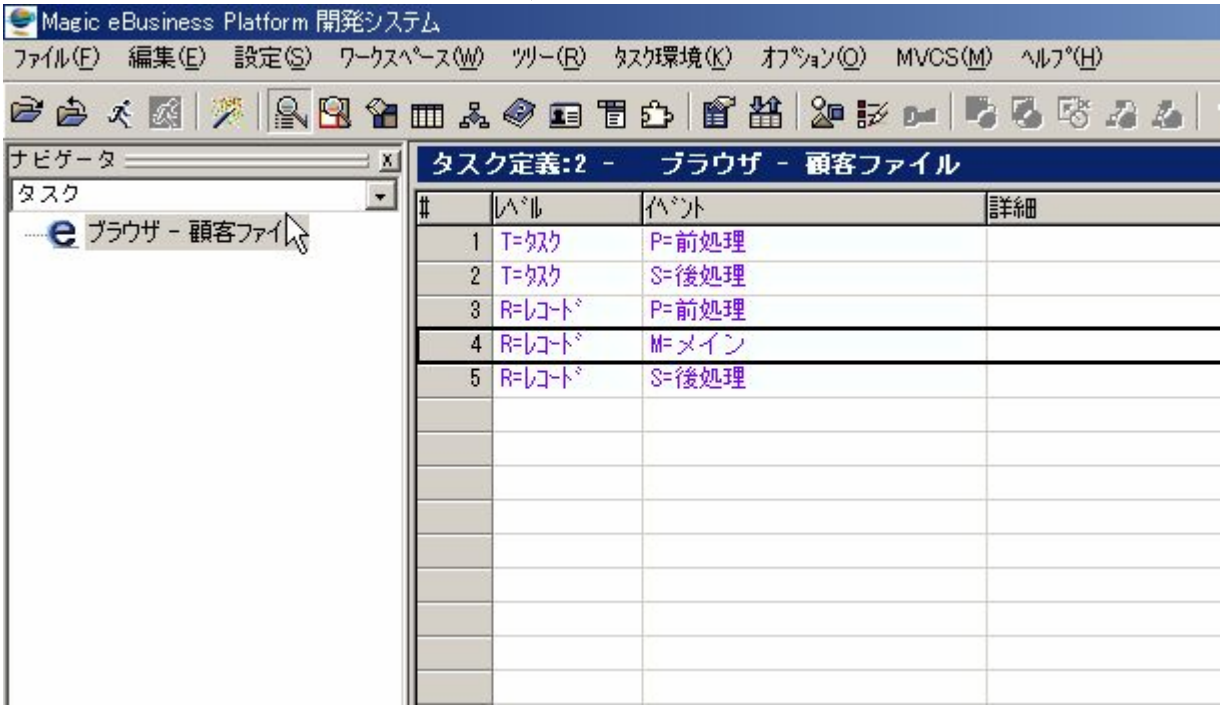
② E：プログラム実行をクリックします。



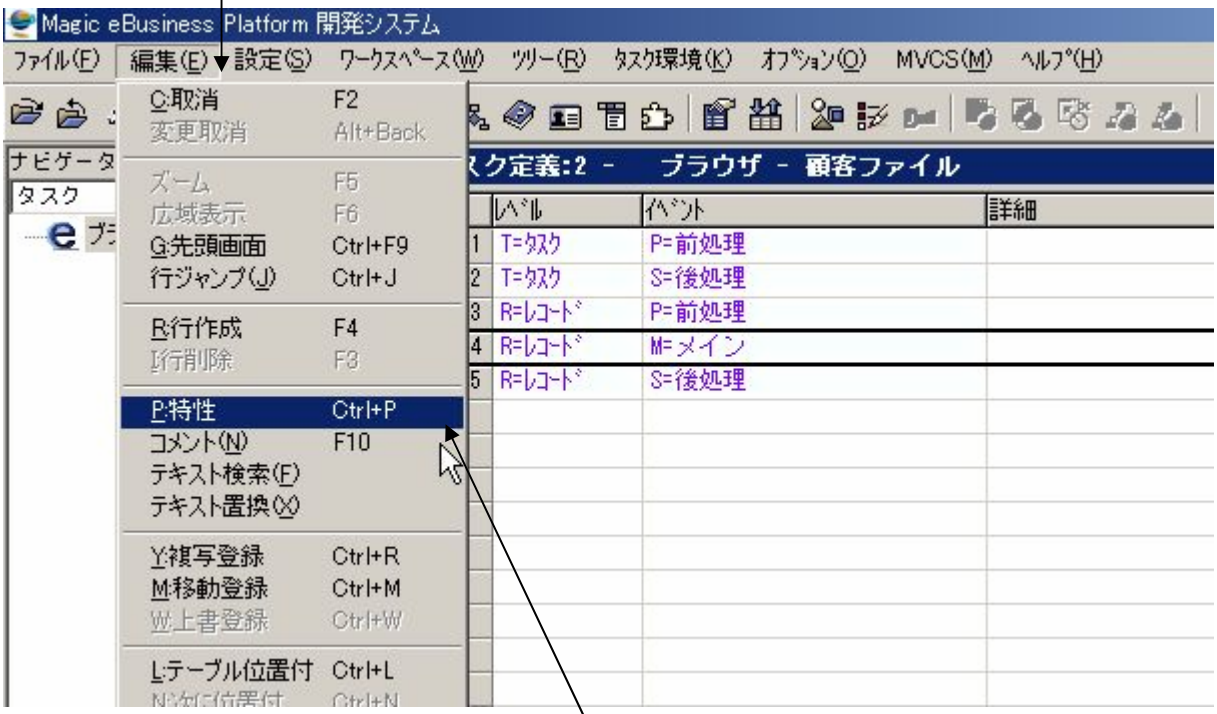
６）ブラウザ上で修正出来るようにする

先ほどのブラウザの起動を終了します。
ブラウザが正常に終了しますと、Magic開発システムの現在実行したPG上にカーソルが停止します。
もししなければ、プログラムリポジトリをオープンし「ブラウザー顧客ファイル」上にカーソルを移動します。

① 編集(Ｅ)をクリックします。　　**② ズームをクリックします。**

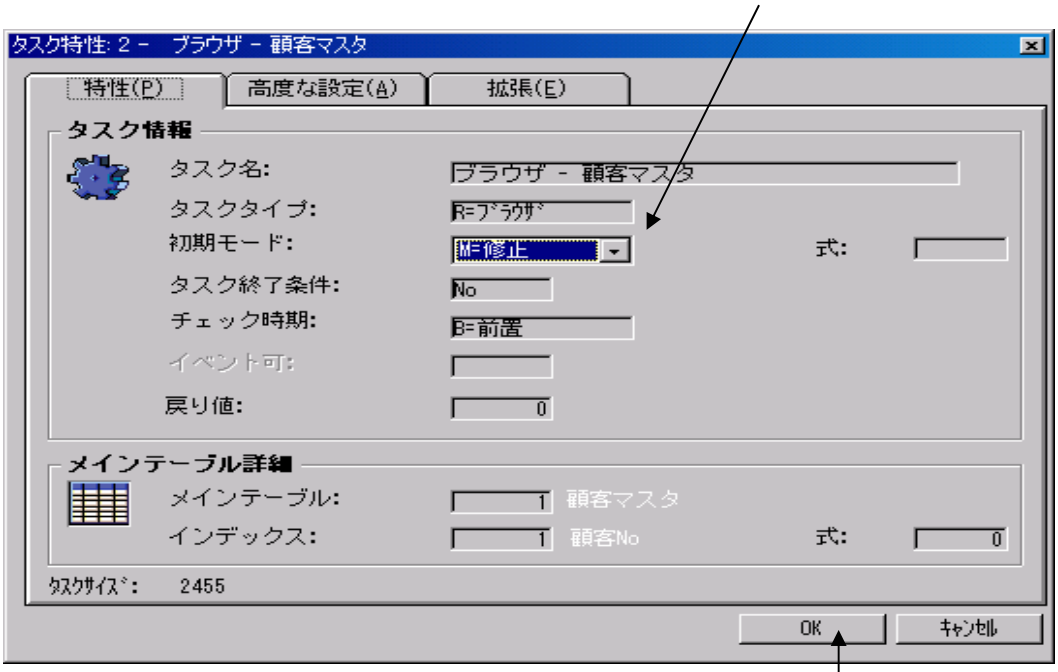


③ 編集(E)をクリックします。



④ P：特性をクリックします。

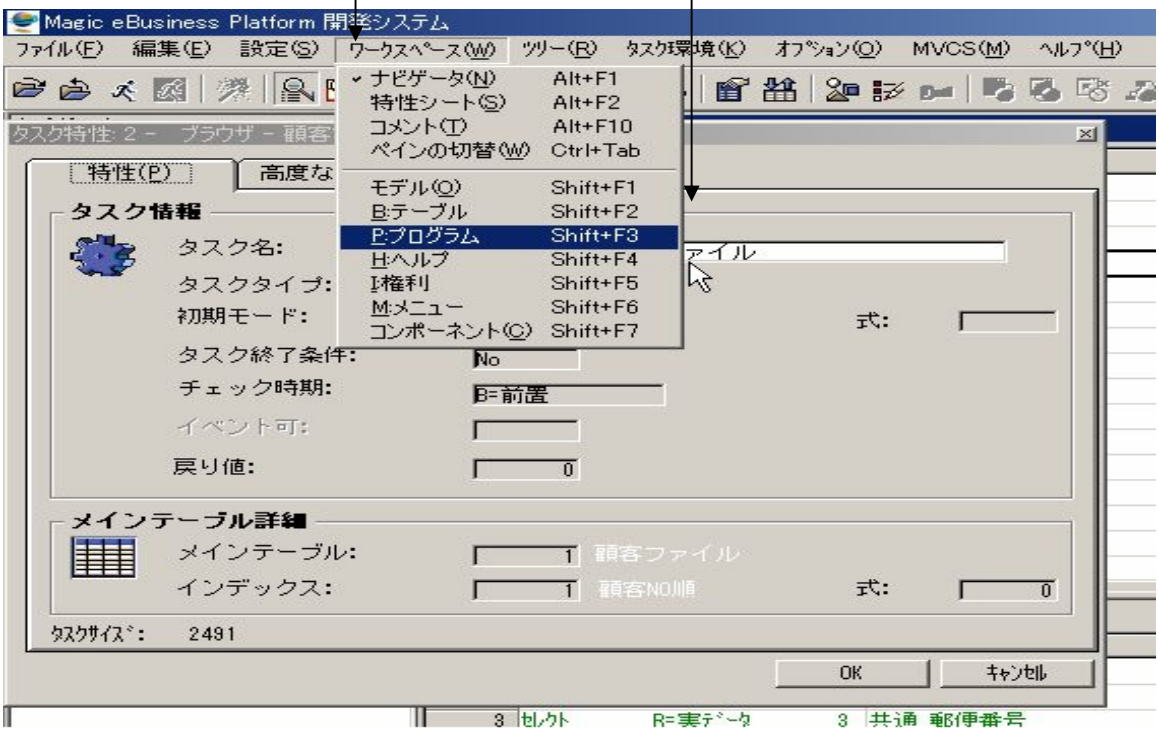
⑤「Q=照会」から「M=修正」に変更します。
この定義だけで、ブラウザ上でデータの変更ができるようになります。



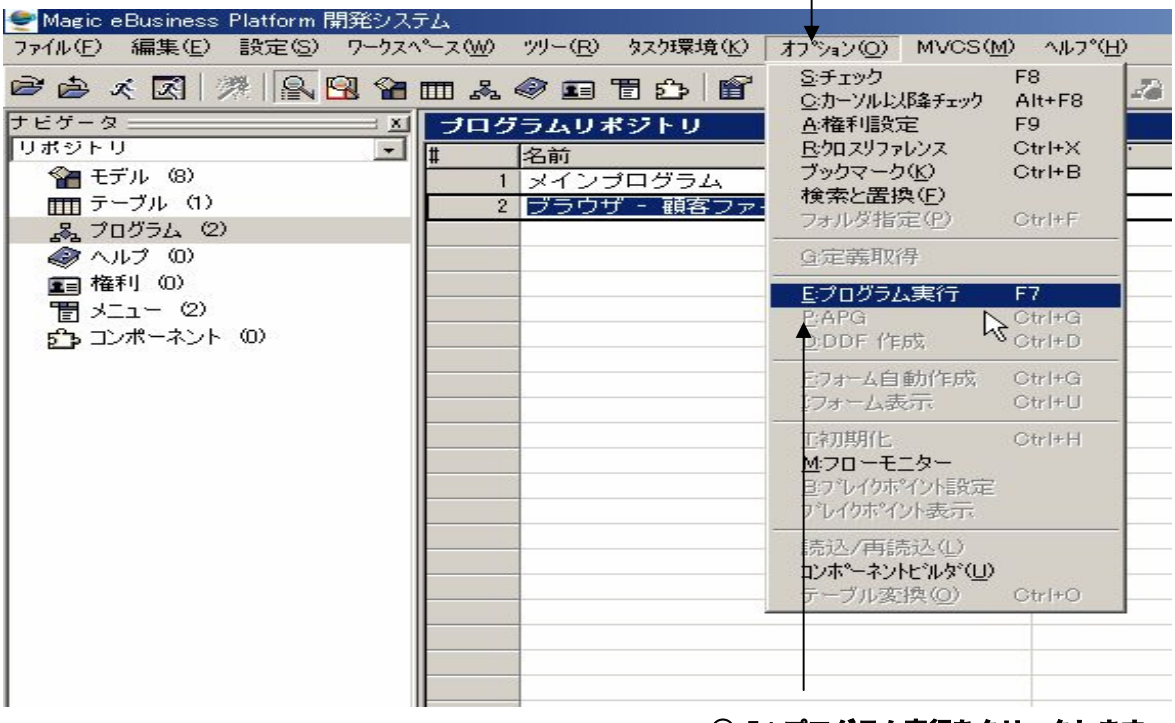
⑥ [OK]ボタンを押します。

⑦ ワークスペース(W)をクリックします。

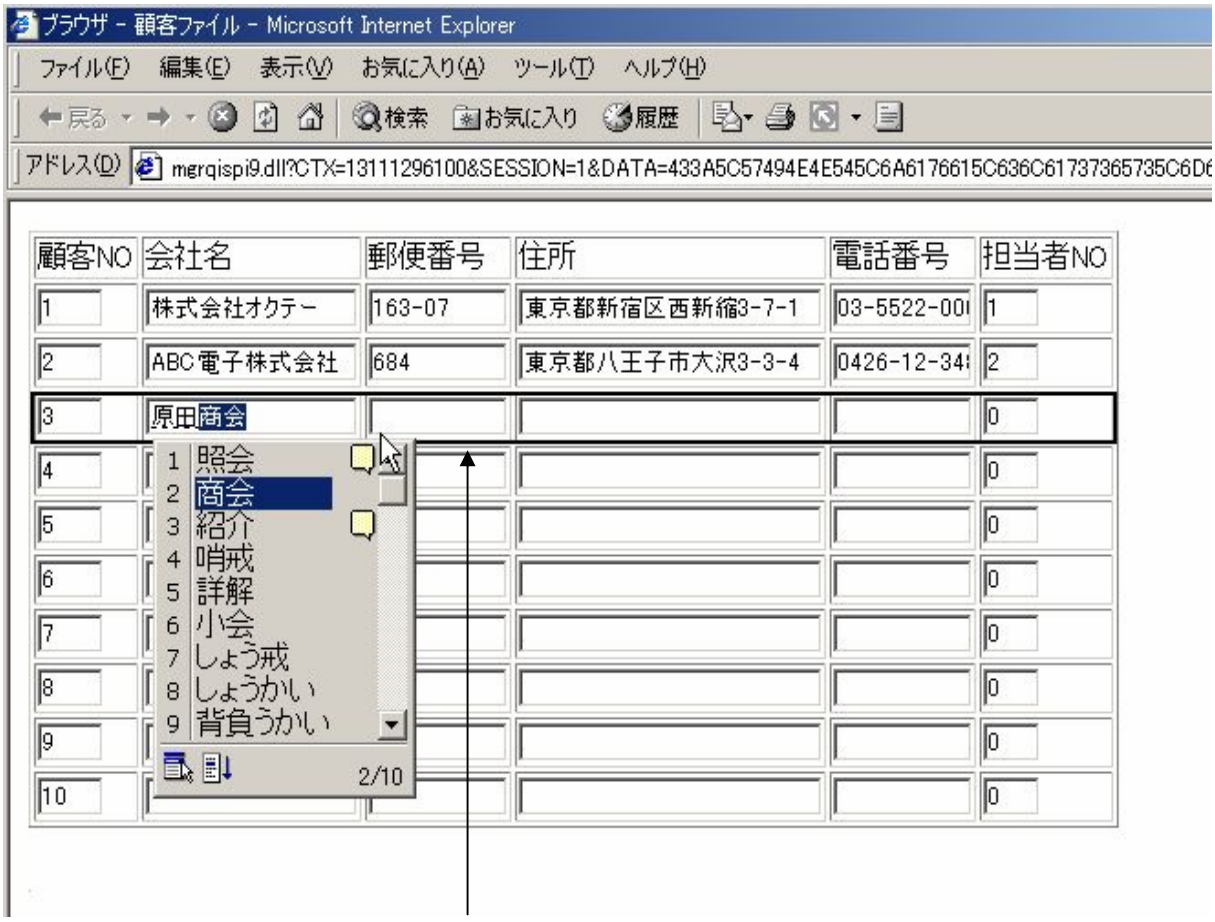
⑧ P：プログラムをクリックします。



⑨ オプション(O)をクリックします。



⑩ E：プログラム実行をクリックします。


ブラウザ - 顧客ファイル - Microsoft Internet Explorer
ファイル(F) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルプ(H)
戻る 検索 お気に入り 履歴
アドレス(D) mgrqispi9.dll?CTX=13111296100&SESSION=1&DATA=433A5C57494E4E545C6A6176615C636C61737365735C6D6
顧客NO 会社名 郵便番号 住所 電話番号 担当者NO
1 株式会社オクテー 163-07 東京都新宿区西新縮3-7-1 03-5522-00 1
2 ABC電子株式会社 684 東京都八王子市大沢3-3-4 0426-12-34 2
3 原田商会 0
4 0
5 0
6 0
7 0
8 0
9 0
10 0
1 照会
2 商会
3 紹介
4 哨戒
5 詳解
6 小会
7 しょう戒
8 しょうかい
9 背負うかい
2/10

⑪ 修正をします。

付録

１．キー操作一覧
２．関数一覧

キー操作一覧

**Magicでの主なキー操作は次のようになります。ただし、これらは
デフォルトのキーボード割り付けであって、カスタマイズしますことも可能です。**

キー	内容
Home	項目の先頭文字に移動します
End	項目の最終文字に移動します
PgUp	前のページに移動します
PgDn	次のページに移動します
Ctrl + PgUp	ページの先頭に移動します
Ctrl + PgDn	ページの最終に移動します
Ctrl + Home	テーブルの先頭に移動します
Ctrl + End	テーブルの最終に移動します
Tab	次項目へ移動します
Shift	前項目へ移動します
→ ←	項目内の文字間を移動します
↑ ↓	項目間を移動します（スクリーンモードの場合） レコード間を移動します（ラインモードの場合）
Alt + →	レコードの最終項目へ移動します
Alt + ←	レコードの先頭項目へ移動します
Insert	挿入モードと上書きモードを切り替えます
Backspace	カーソルの左にある文字を削除します
Delete	カーソル位置の文字を削除します
Alt + Backspace	前回の編集処理を取り消します（アンドゥ）
F2	取り消します
F5	ズームします
F6	広域表示します
F3	レコードを削除します
F4	レコードを挿入します
Ctrl + Insert	カーソルが重なっている項目をコピーします
Shift + Insert	カーソル位置にコピーした内容を貼り付けます
Enter	OKを返す　又は　選択します
Esc	終了　又は　取り消します

フォーム画面におけるキー操作

キー	内容
Home	先頭のコントロールを選択します
End	最終のコントロールを選択します
Tab	次のコントロールを選択します
Shift ＋ Tab	前のコントロールを選択します
矢印キー	選択したコントロール又はファームを移動します
Shift ＋ 矢印キー	選択したコントロール又はファームをサイズ変更します
Esc	操作を取り消します。
Enter	操作を確定します
Insert	フォームにコントロールを挿入します
Delete	選択したコントロールを削除します
スペースバー	コントロールの選択を解除して、フォームを選択します

アクション関数

関数名	説明
BLB2FILE	BLOB オブジェクトをファイルに保存
CALLPROG	Magic プログラムの呼び出し
CASE	値による切り替え
DBCOPY	データベーステーブルのコピー
DBDEL	データベーステーブルの削除
DBDISCNT	現在のデータベース接続の切断
DBRELORD	実行中の常駐テーブルの読み込み
DDEBEGIN	DDE の開始
DDEEND	DDE の終了
DDEGET	DDE サーバから文字列を取得
DDEPOKE	DDE サーバに対する文字列の転送
DDEXEC	DDE サーバに対するコマンド文字列の転送
DELAY	処理の一時停止
EUROCNV	通貨換算（通貨テーブルのユーロ換算レートに基づく）
EURODEL	通貨データの削除
EUROGET	基準通貨の取得
EUROSET	基準通貨の設定
EUROUPD	通貨テーブルの更新
EXPCALC	式の実行
FILE2BLB	ファイルを BLOB オブジェクトに変換
FILE2OLE	ファイルを BLOB オブジェクトに変換し
	リンクまたは埋め込みを行う
FILE2REQ	ファイル内容をリクエスタに送信
FILEDLG	「ファイルを開く」ダイアログのオープン
FLWMTR	フローモニターにメッセージを表示
GROUPADD	ユーザをユーザグループに割り当てる
INIGET	Magic.INI ファイルの値を取得
IOCOPY	ファイルのコピー（ディスク上）
IODEL	ファイルの削除（ディスク上）
IOREN	ファイルの名前の変更（ディスク上）
KBPUT	キー入力シュミレーション
LMCHKIN	ライセンスのチェックイン
LMCHKOUT	ライセンスのチェックアウト
LOCK	リソースをロックする
LOGON	カレントのアプリケーションへのログオン

アクション関数

関数名	説明
MAXMAGIC	Magic のウィンドウを最大表示
MINMAGIC	Magic のウィンドウをアイコン化
MMSTOP	マルチマークの中断
MNUCHECK	メニュー項目のチェックマーク表示／非表示を設定
MNUENABL	メニュー項目の有効／無効を設定
MNUSHOW	メニュー項目の表示／非表示を設定
PROGIDX	プログラム番号の取得
RESMAGIC	Magic のウィンドウを通常サイズに戻す
RIGHTADD	セキュリティファイル上のユーザに権利を割り当てる
SETLANG	言語の設定
UDF	ユーザ定義関数の呼び出し
UNLOCK	リソースのロックを解除
USERADD	セキュリティファイルにユーザを追加
VARSET	項目に値を設定

文字／数値変換関数

関数名	説明
ASC	文字列の文字を ASCII コードへ変換
CHR	ASCII コードを文字に変換
HSTR	10 進数から 16 進数に変換
HVAL	16 進数から 10 進数に変換
MSTR	数値を文字列に変換（長さの指定が可能）
SOUNDX	文字の読み方（発音）の比較
STR	数値を文字列に変換
VAL	文字列を数値に変換

文字列操作関数

関数名	説明
ANSI2OEM	ANSI コードから OEM コードに変換
DEL	文字列中の文字を削除
FILL	文字列を繰り返す
FLIP	文字列を逆転
HAN*	全角の文字を半角に変換
INS	文字列を別の文字列に挿入
INSTR	文字列中で特定の文字列の最初の出現位置を検索

文字列操作関数

関数名	説明
LEFT	文字列の左側の文字を取得
LEN	文字列の長さを取得
LOWER	大文字を小文字に変換
LTRIM	文字列の左側にある空白を削除
MID、MIDV*	文字列から部分文字列を取り出す
OEM2ANSI	OEM コードから ANSI コードに変換
REP、REPV*	文字列内の部分文字列を置き換え
RESTR	文字列内にある特定の文字列を別の文字列で置き換え
RIGHT	文字列の右側から文字を取り出す
RTRIM	文字列の右側にある空白を削除
STRTOKEN	区切り文字で区切られている文字列からトークン
	を取り出す
TRIM	文字列の左右にある空白を削除
UPPER	小文字を大文字に変換
ZEN*、ZENS*	半角文字を全角に変換
ZIMEREAD*	漢字の読みを取得
ZKANA*	ひらがな・カタカナの変換

*日本語版でのみ使用可能

アプリケーションパーティショニング関数

関数	説明
CTXKILL	コンテキストの中断
CTXSTUSE	コンテキストの最終使用時間を取得
CTXNUM	コンテキスト番号の取得
CTXPROG	コンテキストのプログラム名を取得
CTXSIZE	コンテキストのサイズを取得
CTXSTAT	コンテキストの状態を取得
GETPARAM	パラメータの取得
RQEXE	リモートエンジンのロード
RQLOAD	リクエスタのロード
RQQUEDEL	リクエスタのキューの削除
RQQUELST	リクエスタのキュー一覧を取得
RQQUEPRI	リクエスタのキュー優先度を変更
RQREQINF	リクエスタのリクエスト ID 情報を取得
RQREQLST	リクエスタのリクエストエントリ数を取得
RQRTAPP	リクエスタ対応アプリケーション情報を取得
RQRTAPPS	リクエスタのアプリケーション数を取得

アプリケーションパーティショニング関数

関数	説明
RQRTINF	リクエスタの実行エンジンに関する情報を取得
RQRTS	リクエスタの実行エンジンの数を取得
RQRTTRM	実行エンジンを終了
RQSTAT	リクエスタの状態を取得
SETPARAM	パラメータの設定

比較関数	説明
関数	
A　LIKE　A	文字列の比較（パターンマッチング）
MAX	値を比較し、最大値を取り出す
MIN	値を比較し、最小値を取り出す
RANGE	範囲チェック

データベースインタフェース関数

関数	説明
CLRCACHE	現在のタスクのキャッシュをクリア
CURRPOSITION	メインテーブルの現在の位置を取得
DBERR	データベースからのエラーメッセージの取り出しと削除
ERRDATABASENAME	エラーが発生したデータベースの名前を取得
ERRDBMSCODE	エラーが発生したデータベースの DBMS のエラーコードを取得
ERRDBMSMESSAGE	エラーが発生したデータベースの DBMS のエラーメッセージを取得
ERRMagicNAME	エラーが発生した Magic のリテラルを取得
ERRPOSITION	エラーが発生したレコードの位置を取得
ERRTABLENAME	エラーが発生した DB テーブル名を取得
ROLLBACK	トランザクションのロールバック

日付関数

関数	説明
ADDDATE	日付項目に関する計算
BOM	月の最初の日付を取得
BOY	年の最初の日付を取得
CDOW	週の曜日名（英語）を取得
CMONTH	月の名前（英語）を取得
DATE	システムの日付を取得
DAY	日付の日（１～３１）を取得

日付関数

関数	説明
DOW	日付の曜日を示す番号（1～7）を取得
DSTR	日付を文字列に変換
DVAL	文字列を数値(0001／01／01 を起点とする日数)に変換
EOM	月の最後の日付を取得
EOY	年の最後の日付を取得
JCDOW*	週の曜日名（日本語）を取得
JGENGO*	元号名（日本語：平成､昭和､大正､明治）を取得
JMONTH*	月番号（1~12）の月名（日本語）への変換
JNDOW*	曜日番号（1～7）の曜日名（日本語）への変換
JYEAR*	日付の年を元号年に変換
MDATE	Magic の日付を取得
MONTH	日付の月（1～12）を取得
NDOW	曜日番号（1～7）の曜日名（英語）への変換
NMONTH	月番号（1～12）の月名（英語）への変換
YEAR	日付けの年を取得

*日本語版でのみ使用可能

ファイル操作関数

関数	説明
DBCACHE	キャッシュのヒット率を取得
DBDISCNT	現在のデータベース接続を切断
DBDEL	データベーステーブルを削除
DBEXIST	ディスク上のデータベーステーブルの存在確認
DBNAME	データベーステーブル名を取得
DBRECS	データベースの行数を取得
DBSIZE	データテーブルのサイズを取得

DDE 関数

関数	説明
DDEBEGIN	DDE の開始
DDEEND	DDE の終了
DDEGET	DDE サーバから文字列を取得
DDEPOKE	DDE サーバに対する文字列の転送
DDERR	DDE エラーの検出
DDEXEC	DDE サーバに対するコマンド文字列の転送

関数	説明
CHEIGHT	コントロールの高さを取得
CLEFT	コントロールのフォーム内座標位置（X 座標）を取得
CLEFTMDI	コントロールの左辺位置（Magic MID 基準での X 座標）を取得
CLICKCX	コントロール内のクリック位置（X 座標）を取得
CLICKY	コントロール内のリリック位置（Y 座標）を取得
CLICKWX	ウィンドウ内のクリック位置（X 座標）を取得
CLICKWY	ウィンドウ内のクリック位置（Y 座標）を取得
CTOP	コントロールフォーム内座標位置（Y 座標）を取得
CTOPMDI	コントロールの左辺位置（Magic MID 基準での Y 座標）を取得
CTRLHWND	コントロールのウィンドウハンドルを取得
CTRLNAME	コントロール名を取得
CURROW	テーブルコントロール上出の現在の行番号を取得
CWIDTH	コントロールの幅を取得
HITZORDR	コントロール Z オーダー値を取得
INIGET	Magic.INI ファイルの環境設定値を取得
INIGETLN	Magic.INI ファイルの値を取得
INIPUT	Magic.INI ファイルの環境設定値を更新
IOCURR	出力中の入出力ファイル番号を取得
KBGET	直前のキー入力を取得
KBPUT	キー入力を実行
LASTPARK	最後にパークしたコントロールの名前を取得
MENU	メニュー経路を取得
MMCOUNT	マーク行数を取得
MMCURR	現在処理中のマーク行を取得
OWNER	オーナー名を取得
PPD	プロテクションデバイスを読み込みます
PREF	アプリケーションファイルの識別子を取得
PROG	タスクのパスを取得
RIGHTS	ユーザに権利が付与されているかを照会
RUNMODE	エンジンの実行モードに対応する数値コードを取得
SYS	アプリケーション名を取得
TERM	端末番号を取得
TEXT	サーバがバックグラウンドモードであるか確認

識別関数と環境関数

関数	説明
USER	ユーザ情報を取得
WINBOX	ウィンドウの位置とサイズを取得
WINHWND	ウィンドウハンドルを取得

入出力関数

関数	説明
EOF	入出力ファイルの最後のレコードの処理が終了したか確認
EOP	出力ファイルの行数が 1 ページ分を超えたか確認
IOCOPY	ファイルのコピー（ディスク上）
IODEL	ファイルの削除（ディスク上）
IOEXIST	ファイルの存在を確認(ディスク上)
IOREN	ファイルの名前を変更（ディスク上）
IOSIZE	ファイルの照会（ディスク上）
LINE	出力ファイルの行番号を取得
PAGE	出力ファイルのページ番号を取得

言語関数

関数	説明
GETLANG	現在使用されている言語を取得
MLSTRANS	文字列を他の言語に変換
SETLANG	使用します言語を設定

算術関数と三角関数

関数	説明
ABS	絶対値
ACOS	逆余弦（アークコサイン
ASIN	逆正弦（アークサイン
ATAN	逆正接（アークタンジェント）
COS	余弦（コサイン）
EXP	指数
LOG	自然対数
RAND	乱数を生成
SIN	正弦（サイン）三角関数
TAN	正接（タンジェント）三角関数

メニュー関数

関数	説明
MENU	メニュー経路を取得
MNUCHECK	メニュー項目のチェックマークの表示／非表示を設定
MNUENABL	メニュー項目の有効／無効を設定
MNUSHOW	メニュー項目の表示／非表示を設定

数値操作関数

関数	説明
FIX	数値の抽出（範囲外は切り捨て
ROUND	数値の四捨五入

特殊テスト関数

関数	説明
CHKDGT	検査数字を作成
CRC	CRC 方式による計算
FLOW	フローモードのチェック
IDLE	システムアイドル時間をチェック
IF	条件付評価(IF-THEN-ELES)
ISDEFAULT	デフォルト値かどうかのチェック
ISNULL	NULL かどうかのチェック
KBGET	直前のキー入力を取得
LEVEL	タスクの実行レベルを照会
STAT	タスクの処理モードをチェック
TDEPTH	タスクの深さを照会
VARATTR	項目の型を照会
VARPREV	項目のオリジナル値を取得

時刻変数

関数	説明
ADDTIME	時刻に関する計算
HOUR	時刻の時間の値
MINUTE	時刻の分の値
SECOND	時刻の秒の値
TIME	システムの時刻を取得
TSTR	時刻を文字列に変換
TVAL	文字列を時刻に変換

ユーザ定義関数

関数	説明
CALLJS	Java スクリプトの呼び出し
CALLOBJ	オブジェクトの呼び出し
UDF	ユーザ定義関数の呼び出し
CALLDLL、CALLD-LLF、CALLDLLS	外部 DLL の関数呼び出し

項目値操作関数

関数	説明
CNDRANGE	条件を使用して項目に値を設定
NULL	NULL に設定
VARSET	項目に値を設定

ビュー関数

関数	説明
COUNTER	繰り返しカウンタ
THIS	イベントが発生したデータ項目のインデックスまたはタスクの階層をあらわす
VARATTR	項目の型を照会
VARCURR	項目の値を取得
VSRINP	最後の入力項目を照会
VARMOD	項目の値が変更されたか確認
VARNAME	オリジナルのテーブルと項目を取得
VARPREV	[項目一覧]内のシンボル名で示す項目のオリジナル値を検索する
VARSET	項目に値を設定
VIEWMOD	データビューのレコードが修正されたかを確認

マジックソフトウェア・ジャパン株式会社

〒151-0053　東京都渋谷区代々木三丁目 25 番 3 号　あいおい損保新宿ビル 14 階

電話：03-5365-1600（代表）ファックス：03-5365-1630

事業所：札幌　仙台　信越　名古屋　大阪　岡山　広島　福岡

http://www.magicsoftware.co.jp/