IBM Cognos Analytic Server
バージョン 10.1.0

TurboIntegrator ガイド

注記

本書および本書で紹介する製品をご使用になる前に、 95 ページの『特記事項』に記載されている情報をお読みください。

このマニュアルは IBM Cognos Express バージョン 10.1.0 を対象として作成されています。また、その後のリリースも対象となる場合があります。このマニュアルの最新バージョンに関する情報は、IBM Cognos インフォメーション・センター (http://publib.boulder.ibm.com/infocenter/cogic/v1r0m0/index.jsp) で見ることができます。

お客様の環境によっては、資料中の円記号がバックスラッシュと表示されたり、バックスラッシュが円記号と表示されたりする場合があります。

原典： IBM Cognos Analytic Server
Version 10.1.0
TurboIntegrator Guide

発行： 日本アイ・ビー・エム株式会社

担当： トランスレーション・サービス・センター

第1刷 2012.5

目次

はじめに

このマニュアルは、IBM® Cognos® Express® Xcelerator の使用にあたって参照してください。

このマニュアルでは、IBM Cognos Xcelerator TurboIntegrator を使用して、データとメタデータをさまざまな Business Analytics ソースからインポートする方法について説明します。

Xcelerator ソフトウェアが使用するサーバーは、IBM Cognos Analytic Server (ICAS) と呼ばれます。

Business Analytics は、企業全体の財務、運用、顧客、および組織のパフォーマンスを継続して管理し、モニターするためのソフトウェア・ソリューションを提供します。

情報の検索

IBM Cognos 製品のマニュアルを Web で入手するには、 IBM Cognos インフォメーション・センター (http://publib.boulder.ibm.com/infocenter/cogic/v1r0m0/index.jsp) にアクセスしてください。各言語版のマニュアルもすべて用意されています。リリース・ノートはインフォメーション・センターへ直接公開されます。これには、最新の技術情報および APAR へのリンクが含まれています。

サンプルに関する特記事項

Great Outdoors 社、GO 営業、Great Outdoors 名の変形、およびプランニング・サンプルは、IBM および IBM 顧客のサンプル・アプリケーションを開発するために使用されたサンプル・データの架空の企業運営を表しています。これらの架空データには、販売取引、商品流通、財務、および人事のサンプル・データが含まれます。実際の名前、住所、電話番号、または取引額との類似は偶発的なものです。また、サンプル・ファイルの中には、手動またはコンピューターで生成された架空のデータ、学術的ソースまたは公共のソースを基に編集された実際のデータ、版権所有者の許可を得て使われているデータなどが、サンプル・アプリケーションを開発するためのサンプル・データとして使用されている場合もあります。参照される製品名は、それぞれの所有者の商標である可能性があります。無断の複写は禁止されています。

ユーザー補助機能

現在この製品では、ユーザー補助機能はサポートされていません。ユーザー補助機能とは、動作が制限されている方、視力の限られた方など、身体の不自由な方に製品をご使用いただけるように支援する機能のことです。

将来予想に関する記述

このマニュアルでは、本製品の現在の機能について説明しています。一部の内容で、現在利用できない項目について言及している可能性があります。これは、将来

利用できるようになることを意味するものではありません。そのような内容は、資料、コード、または機能の提供に向けた取り組み、確約、あるいは法律上の義務を意味するものではありません。機能の開発、リリース、時期や機能性は IBM の独自の決定によるものとします。

第 1 章 新着情報

このセクションでは、このリリースの新機能、変更された機能、削除された機能のリストを示します。

これは、アップグレードやアプリケーション配布の戦略、およびユーザーに必要なトレーニングを計画するうえで役立ちます。

最新の製品マニュアルを見つけるには、 IBM Cognos Express インフォメーション・センター (http://publib.boulder.ibm.com/infocenter/cx/v10r1m0/index.jsp) にアクセスしてください。

バージョン 10.1.0 の新機能

以下に、前回のリリース後に追加された IBM Cognos Express Xcelerator の新機能をリストします。

Cognos TM1RunTI を使用したコマンド行からの TurboIntegrator プロセスの実行

TM1RunTI は、 IBM Cognos Analytic Server (ICAS) TurboIntegrator (TI) プロセスを開始することのできるコマンド行インターフェース・ツールです。

このツールを使用すると、管理者は Express Xcelerator から外部的に TurboIntegrator プロセスを実行して、それにパラメーターを渡すことができます。また、これを使用して、順次実行される複数の TurboIntegrator プロセスのスケジューリングを準備することもできます。以前は、TurboIntegrator アクティビティーの完了ではなく、時間的なスケジュールに基づいて日課の処理が実行されました。

42 ページの『TM1RunTI の使用』を参照してください。

TurboIntegrator プロセスのシリアライズ

synchronized() を使用すると、TurboIntegrator プロセスをシリアライズして、それらを順次処理できるようになります。

一度に 1 つの TurboIntegrator プロセスだけが更新可能です。明示的に禁止しない限り、複数の TurboIntegrator プロセスが並列的に実行されることがあります。

synchronized() 関数は複数のプロセスが順次完了するように、それらをシリアライズします。こうして効率が改善され、互いのデータに依存するプロセスにより不必要なロック競合が生じるのを防げます。

51 ページの『synchronized() を使用した TurboIntegrator プロセスのシリアライズ』を参照してください。

サーバー始動時における日課の実行

StartupChores は、サーバー始動時に実行される日課のリストを識別する新しい設定パラメーターです。

StartupChores はサーバーが開始すると直ちに、1 つの TurboIntegrator プロセス、またはプロセスから成るセットを実行します。 StartupChores は、すべてのユーザー・ログインの前、およびすべてのスケジュール済み日課の実行前に実行されます。

60 ページの『サーバー始動時における日課の実行』を参照してください。

日課内での TurboIntegrator プロセスの個別コミット

TurboIntegrator 日課では、日課の一部として個別のプロセスをコミットできるようになりました。

通常は、日課の期間にわたってロックが保持されます。この新機能を使用することにより、管理者は TurboIntegrator プロセスを順次化しながらそれらを個別にコミットして、それらのロックを解放することができます。

59 ページの『ChoreCommit の使用』を参照してください。

第 2 章 TurboIntegrator の基礎

このセクションでは、TurboIntegrator を使用して、データを IBM Cognos Xcelerator キューブにインポートする方法に関する基本情報を説明します。 TurboIntegrator では、ソースのデータ構造を認識し、 Xcelerator が必要とする適切な構造への変形するプロセスを設計することができます。一度 TI プロセスが設計された後は、それを再実行したり、動的ソースからデータをインポートするときに使用されるようにスケジュールすることができます。後続のセクションでは、特定のソース・タイプからデータをインポートする手順について説明します。

TurboIntegrator を使い始める前に、この章で説明されている全種類のソースに当てはまる情報を確実に理解しておいてください。

TurboIntegrator で使用可能なデータ・ソース

Xcelerator TurboIntegrator では、次のデータ・ソースのデータをインポートできます。

- ASCII ファイルを含む、コンマ区切りのテキスト・ファイル。
- ODBC データ・ソース経由でアクセス可能なリレーショナル・データベース・テーブル。
- 他のキューブおよびビュー。
- Microsoft Analysis Services。
- SAP 対 RFC。
- IBM Cognos パッケージ

これらのソース・タイプの詳細については、本マニュアルの他のセクションを参照してください。

TurboIntegrator での文字列長の制限

TurboIntegrator は、8000 シングルバイト文字までのサイズの文字列データを一度に扱うことがきます。この制限は、TI プロセスで、変数への値の代入、データの個々のレコードのインポートなどの操作が実行される場合に適用されます。8000 シングルバイト文字より長い値やレコードの末尾は切り捨てられます。

例えば、テキスト・ファイルからデータ行をインポートする場合、テキストの各行の長さが 8000 文字を超えることはできません。コンマ区切りファイルのデータをインポートする場合は、ファイル内の各レコードの長さが 8000 文字を超えることはできません。

インポート処理のオプション

TurboIntegrator を使用してデータをインポートする際には、次のオプションがあります。

- キューブを作成し、ソースからインポートしたデータをキューブに移入する。

- キューブを作成し直す。このオプションでは、現存するキューブが破棄され、再作成されます (そのため、インポート時に、データとメタデータの両方を変更できます)。
- キューブ構造を維持したまま、既存のキューブを更新する。このオプションでは、データを既存のキューブ構造にインポートできます。
- ソースからインポートしたデータでディメンションを作成する。
- インポートしたデータでディメンションを更新する。

TurboIntegrator では、これらの操作を任意に組み合わせて実行できます。

TurboIntegrator 関数

TurboIntegrator には、 Xcelerator オブジェクト (キューブ、ビュー、ディメンション、要素など) をデータのインポート時に操作することができる一連の関数が用意されています。

これらの TurboIntegrator 関数に加えて、TurboIntegrator プロセスには、STET 関数を除く標準的なすべての Xcelerator 規則関数も組み込むことができます。

TurboIntegrator 関数については、『IBM Cognos Xcelerator *参照ガイド*』の「Xcelerator TurboIntegrator 関数」を参照してください。

プロセスと日課

TurboIntegrator でのデータ・インポートは、*プロセス*を定義して行います。プロセスは、次の要素で構成される Xcelerator オブジェクトです。

- データ・ソースの説明
- データ・ソース内の各列に対応する一連の変数
- Xcelerator データベースの変数間の関係およびデータ構造を定義する一連のマップ
- プロローグ・プロシージャー (データ・ソースの処理前に実行する一連の操作で構成されます。)
- メタデータ・プロシージャー (キューブ、ディメンションなどのメタデータ構造を更新または作成する一連の操作で構成されます。)
- データ・プロシージャー (Xcelerator データベース内のデータを更新または変換する一連の操作で構成されます。)
- エピローグ・プロシージャー (データ・ソースの処理後に実行されます。)
- プロセスを一般化し、異なる状況で使用できるようにするための一連のパラメーター。

*日課*は、一連の Xcelerator プロセスのコンテナー・オブジェクトです。日課を使用することで、複数のプロセスを一定の順序で実行したり、プロセスを一定のタイミングで実行するスケジュールを作成できます。詳しくは、57 ページの『第 8 章 日課によるプロセスの自動実行スケジュールの作成』を参照してください。

TurboIntegrator プロセス内での操作の順序

TurboIntegrator プロセスには、複数のプロシージャー (プロローグ、メタデータ、データ、およびエピローグ) が含まれます。これらのプロシージャーは、TurboIntegrator エディターの「詳細」タブのサブタブとして表示することができます。

データ・ソースを定義し、変数を設定して、プロセスのデータ操作を指定すると、TurboIntegrator プロセスを実行するときに実行されるスクリプトが Xcelerator によって生成されます。このスクリプトは、TurboIntegrator エディターのそれぞれのプロシージャー・サブタブに配置されます。また、TurboIntegrator 関数と規則関数を使用して、独自のスクリプトを任意のプロシージャー・サブタブ内に作成することもできます。

TurboIntegrator プロセスを実行すると、次の順序でプロシージャーが実行されます。

1. TurboIntegrator プロセスのデータ・ソースが開かれる*前*に、プロローグ・プロシージャーが実行されます。
2. プロセスのデータ・ソースが「なし」の場合は、プロローグの処理が終了するとすぐにエピローグ・プロシージャーが実行されます。

   注: プロセスのデータ・ソースが「なし」の場合、メタデータ・プロシージャーとデータ・プロシージャーは無視されます。その場合は、プロセスのすべてのスクリプトを、プロローグまたはエピローグのいずれかのプロシージャー内に作成する必要があります。
3. データ・ソースが「なし」以外の場合は、データ・ソースが開かれます。
4. データ・ソースの最初のレコードに対して、メタデータ・プロシージャーのすべての行が順に実行されます。次に、データ・ソースの 2 番目のレコードに対してすべての行が順に実行され、それ以降もすべてのレコードが処理されるまで同じ処理が実行されます。
5. データ・ソースの最初のレコードに対して、データ・プロシージャーのすべての行が順に実行されます。次に、データ・ソースの 2 番目のレコードに対してすべての行が順に実行され、それ以降もすべてのレコードが処理されるまで同じ処理が実行されます。
6. データ・プロシージャーの処理が終了すると、データ・ソースが閉じられます。
7. エピローグ・プロシージャーが実行されます。
8. Xcelerator によって TurboIntegrator プロセスが閉じられます。

TurboIntegrator プロセスに関する注意事項

TurboIntegrator プロセスの作成と編集では、次の点に注意してください。

- TurboIntegrator で新規または変更されたディメンションがコンパイルされるのは、そのディメンションが作成または変更されるプロシージャーが終了した時点でのみです。

  新しいディメンションの場合、これは、ディメンションを作成するプロシージャーがデータ・ソース内のすべてのレコードを処理するまでは、新しいディメンションにアクセスできない (アクセス手段が TurboIntegrator かどうかは問いません)

ことを意味します。変更するディメンションの場合、これは、ディメンションを変更するプロシージャーが処理を完了するまでは、新しい要素にアクセスできないことを意味します。

- TurboIntegrator 関数および (STET を除く) 規則関数は、プロセス内の任意のプロシージャーで使用できます。また、プロシージャーで使用することができる関数に制限はありません (すべての関数が、あらゆる TurboIntegrator プロシージャーで有効です)。
- TI プロセスと規則の論理演算子や算術演算子などのさまざまな演算子の使用方法については、『IBM Cognos Analytic Server *Rules Guide*』の規則の概要の章の「式」を参照してください。
- TurboIntegrator プロセスでは、NULL 値は、数値の場合はゼロに変換され、文字列値の場合は空の文字列に変換されます。
- 集約要素を既存の n レベル要素の下に置こうとする場合、n レベル要素は集約要素に変わり、元の n レベル要素のデータは失われます。

ただし、関数は論理的な順序で使用し、プロセスで目標を必ず達成できるようにすることが必要です。例えば、ディメンションに新しい要素を追加し、その要素のデータ値を更新するプロセスを構築する場合は、新規要素の追加とディメンションのコンパイルが終了してから、その要素のデータ値を更新するようにプロセスが実行されることを確認する必要があります。通常は、メタデータ・プロシージャーで DimensionElementInsert 関数を使用して新規要素を追加してから、データ・プロシージャーで CellPutN 関数を使用して値を更新することになります。

この例で、新規要素を追加する操作、および そのデータ値を更新する操作の両方をデータ・プロシージャーで実行するプロセスを構築した場合、そのプロセスは正しく動作しません。処理が失敗するのは、前述したように、変更されたディメンションは、プロシージャーの終了時のみにコンパイルされるためです。ディメンションがコンパイルされるまでは、新しい要素が存在しません。存在していない要素のデータ値を TurboIntegrator で変更することはできません。このため、プロセスは失敗します。

同じ ICAS サーバーに同時接続。

プロセスが既に実行している*同じ* ICAS サーバーに新しい接続 (ログオン) を作成する TurboIntegrator プロセス内では、いかなる操作も行わないでください。 2 つのログオン間のデッドロックや、スレッドによってサーバーがハングまたはクラッシュする原因になります。

例えば次のような状況は回避してください。

- TI プロセスを使用しないで、*同じ*サーバーで ODBO MDX クエリーを起動 (Xcelerator OLE DB MD Provider を使用)。この場合、プロセスとクエリーの両方が互いに終了するのを待つことになります。
- TI 関数の `ExecuteCommand` を使用しないで、TI プロセスを呼び出して実行し、*同じ* サーバーに再度ログオンする外部プログラムを*待機* (Wait 引数に 1 を設定)。この中には、同じサーバーに再度接続するカスタム・アプリケーションや IBM Cognos アプリケーション (Xcelerator ETLDAP ユーティリティーなど) が含まれます。

Wait 引数に 1 を設定して ExecuteCommand 関数を使用すると、外部プログラムが再度同じサーバーにログオンしない場合でも、サーバーがハングする危険があります。外部プログラムに内部エラーが発生してハングした場合は、基本的に TI プロセスは外部プログラムの待機をハングして、実行を終了します。

TurboIntegrator 関数での別名

別名は、規則上の関連する要素のプリンシパル名の代わりに、また TurboIntegrator において使用することができます。

個人用ワークスペースとサンドボックスを TurboIntegrator プロセスで使用する

このセクションでは、個人用ワークスペースとサンドボックスを TurboIntegrator プロセスおよび関数で使用する方法について説明します。

TurboIntegrator プロセスを個人用ワークスペースかサンドボックスで手動で実行する

そのプロセスの「**アクティブ・サンドボックスを使用**」 プロパティーを選択することで、Server Explorer で現在アクティブなサンドボックスと共にプロセスを手動で実行することができます。アクティブなサンドボックスは、Cube Viewer で現在選択されているサンドボックスにより決定されます。個人用ワークスペースでは、デフォルトのサンドボックスのみを使用することができます。

注: 日課、およびそれらに含まれるプロセスは、個人用ワークスペースまたはサンドボックスで実行できません。プロセスが日課の一部として実行される場合、基本データに対して実行されるだけです。

手順

1. Server Explorer において、Cube Viewer のビューを開きます。
2. 利用可能なサンドボックスのリストからサンドボックスをクリックし、そのプロセスに使用するサンドボックスを選択します。
3. ツリー・ウィンドウで、プロセスを右クリックし、「**アクティブ・サンドボックスを使用**」をクリックしてオプションを有効にします。
4. プロセスを右クリックし、「**実行**」をクリックします。

タスクの結果

プロセスは現在アクティブなサンドボックスを使用して実行されます。

TurboIntegrator 関数をサンドボックスで使用する

次の TurboIntegrator 関数により、TurboIntegrator プロセスが個人用ワークスペースまたはサンドボックスと相互に作用することができるようになります。

- GetUseActiveSandboxProperty
- SetUseActiveSandboxProperty
- ServerActiveSandboxGet

- ServerActiveSandboxSet

これらの関数は、Server Explorer インターフェースで利用できる **「アクティブ・サンドボックスを使用」**プロパティーと同じです。

詳細については、『IBM Cognos Xcelerator *リファレンス・ガイド*』の TurboIntegrator 関数についての節を参照してください。

第 3 章 テキスト・ファイルのインポート

このセクションでは、(ASCII などの) コンマ区切りのテキスト・データを IBM Cognos Xcelerator TurboIntegrator でインポートする方法について説明します。Xcelerator プロセスは個々に異なり、他のタイプのデータ・ソースからインポートする処理ではその内容がわずかに異なりますが、このセクションで説明する手順は、ほとんどのプロセスに共通しています。この手順および例では、ファイル “NewEngland.cma”を使用します。このファイルは、 Xcelerator に付属するサンプル・データの 1 つとしてインストールされています。

テキスト・ファイルからのディメンションの作成

ディメンションは、TurboIntegrator を使用して、データ・ソース内の要素名のリストから作成できます。この方法は、何千何百もの要素を含むディメンションを最も速く作成できます。

TurboIntegrator でディメンションを作成する場合は、プロセスを定義して、 ICAS サーバーにオブジェクトとして保存します。このプロセスは、他のユーザーからもアクセスできます。また、オンデマンドで実行することも、スケジュール設定した間隔で実行することもできます。

TurboIntegrator でディメンションを作成するには :

1. データ・ソースを Xcelerator に定義します。 『データ・ソースの定義』を参照してください。
2. Xcelerator が遭遇する変数を特定します。 11 ページの『データ・ソースの変数の識別』を参照してください。
3. 変数をそのデータ・タイプにマッピングします。 14 ページの『変数のマッピング』を参照してください。
4. プロセスを保存して、実行します。 16 ページの『TurboIntegrator プロセスの保存と実行』を参照してください。

データ・ソースの定義

TurboIntegrator を使用する際はいつでも、データを読み出すデータ・ソースを最初に定義する必要があります。この例では、この TurboIntegrator プロセスのデータ・ソースとして、“NewEngland.cma”という名前の ASCII ファイルを定義します。

手順

1. サーバー・エクスプローラーの左側のウィンドウにある「**プロセス**」を右クリックし、「**プロセス**」 → 「**新規プロセスの作成**」を選択します。
2. 「データ・ソース」タブで「**テキスト**」をクリックします。

   「TurboIntegrator」ウィンドウが開きます。
3. 「**参照**」をクリックします。

   「入力ファイルを選択」ダイアログ・ボックスが開きます。

4. “NewEngland.cma”にナビゲートし、選択して「**開く**」をクリックします。

   “NewEngland.cma”は、PData または SData のいずれかのサンプル・ディレクトリーにあります。 Xcelerator のデフォルト・インストール・ディレクトリーを使用している場合、このファイルの完全なパスは次のとおりです。

   ```
   C:\Program Files\Cognos\TM1\Custom\TM1Data\SData\NewEngland.cma
   ```

   または

   ```
   C:\Program Files\Cognos\TM1\Custom\TM1Data\PData\NewEngland.cma.
   ```

   ファイルの場所を汎用命名規則 (UNC) で指定するように求めるメッセージが表示される場合があります。プロセスを一貫して ASCII ファイルに対して実行する場合は UNC を使用してください。その際、次の条件を満足することも確認してください。

   - Microsoft Windows ICAS サーバーを実行している場合、ASCII ファイルは、サーバーからアクセスできるように、共有 Windows ディレクトリーに存在する必要があります。
   - UNIX オペレーティング・システム上で ICAS サーバーを実行している場合、ファイルは Xcelerator Windows クライアントと ICAS UNIX サーバーの両方から認識できるいずれかの共有ネットワーク・ディレクトリーに存在する必要があります。

   注： UNIX オペレーティング・システム上で ICAS サーバーを実行している場合は、入力ソース・ファイル名に大文字やスペース文字を含めることができません。
5. 警告ボックスで「**OK**」をクリックします。
6. 「TurboIntegrator」ダイアログ・ボックスで次のように指定します。

   “NewEngland.cma”は、区切り文字にコンマ、引用符に二重引用符、タイトル・レコードなし、小数点記号にピリオド、桁区切り記号にコンマを使用した区切りつきソースです。

   このソースを定義するには、次の設定を入力します。
   - 「区切り記号の種類」で「**区切り付き**」を選択します。
   - 「区切り記号」として「**コンマ**」を選択します。
   - 「引用符」に " を入力します。
   - 「タイトル・レコード」のフィールドは空白のままにします。
   - 「小数点記号」に . を入力します。
   - 「桁区切り記号」に , を入力します。
7. 「**プレビュー**」をクリックします。

   ウィンドウの下部に、ソース・データのサンプルが表示されます。

固定長レコードの使用

TurboIntegrator では、フィールド幅が固定されたテキスト・ファイルからデータをインポートすることもできます。データ・ソースが固定フィールド幅を持っている

ことを指定するには、データ・ソースの場所を指定した後で、「区切り記号の種類」で「**固定幅**」を選択してから、「**フィールド幅の設定**」をクリックします。

「データのプレビュー」ダイアログ・ボックスに、ソース・データの最初の 3 レコードが表示されます。データ・ソースのレコードの内容に基づいてフィールド幅を設定するには、次の操作を行います。

手順

1. 列見出し「**1**」をクリックします。

   列見出しに区切り線が表示されます。この線は 3 つのレコードにまで延長されています。
2. 区切り線をクリックして、最初の列と 2 番目の列を仕切る位置までドラッグします。

   新しい列見出し「2」が表示されます。
3. 列見出し「**2**」をクリックして、新しい区切り線を 2 番目と 3 番目の列を仕切る位置までドラッグします。
4. テキスト・ソース内の残りのすべての列について、区切り線を設定します。
5. 「**OK**」をクリックして、「TurboIntegrator」ウィンドウに戻ります。

データ・ソースの変数の識別

データ・ソースの定義が終了すると、TurboIntegrator によってソースの列ごとに変数が割り当てられます。この変数の型と内容を識別する必要があります。

この処理の説明のため、次のテキスト・データについて検討します。

New England, Massachusetts, Boston, Supermart, Feb, 2000000

New England, Massachusetts, Springfield, Supermart, Feb, 1400000

New England, Massachusetts, Worcester, Supermart, Feb, 2200000

New England, Connecticut, Hartford, Supermart, Feb, 1240000

New England, Connecticut, New Haven, Supermart, Feb, 2700000

New England, Connecticut, Greenwich, Supermart, Feb, 1700000

最初の 3 つの列は、ソース・テキスト・ファイルから構築する“Location”ディメンションの階層を構成します。

- “New England”集約は、階層の最上位にあります。
- 州の“Massachusetts”および“Connecticut”は、“New England”の 1 つ下のレベルにあります。
- “Boston”、“Hartford”など都市名がある 3 番目の列は、階層の最下位にある単純要素になります。
- 残りの列は、“Location”ディメンションの作成には使用しません。

このデータ構造の「TurboIntegrator」ウィンドウには「変数」タブがあります。

変数名	変数の型	サンプル値
V1	文字列	New England
Massachusetts	文字列	Massachusetts
Boston	文字列	Boston
SuperMart	文字列	SuperMart
2 月	文字列	2 月
V6	数値	2000000

TurboIntegrator によって、各列に変数名が割り当てられ、各列のサンプル値に基づいて変数の型が割り当てられています。

デフォルトの変数名 (“V1”、“Massachusetts”など) は変更できます。変数には、わかりやすい名前を付けることをお勧めします。わかりやすい名前を付けることで、TurboIntegrator スクリプトが読みやすくなり、トラブルシューティングも容易になります。

変数名を編集するには、「変数名」列の名前をクリックし、新しい名前を入力します。この演習では、最初の 3 つの変数の名前が次のように編集されました。

サンプル値	変数名
New England	Region
Massachusetts	ステータス
Boston	City

変数名は英文字で始まる必要があります。使用できるのは次の文字のみです。

文字	説明
大文字	A から Z まで
小文字	a から z まで
数字	0 から 9 まで
ピリオド	.
アンダースコアー	_
ドル記号	$

「変数の型」フィールドでは、列の内容が識別されます。例えば、このデータの最初の列には、文字列 "New England" が含まれています。この変数型は、TurboIntegrator によって「文字列」と正しく識別されています。

注：一般に、ASCII データの場合は「変数の型」フィールドが正確に設定されますが、ODBC データ・ソースから抽出したデータの場合は正確ではありません。

コンテンツ・フィールドは、次のいずれかの設定で指定できます。

オプション	**説明**
無視	データ・ソースの処理時に、列の内容を無視します。
要素	この列には、作成するディメンションの単純要素が含まれています。
集約	この列には、作成するディメンションの集約要素が含まれています。
データ	この列には、データ値が含まれています。 この例の場合、データ値を含む列は無視してください。データ値を含む列は、ディメンションの作成時にインポートされません。
属性	この列には、作成するディメンションの要素属性が含まれています。
その他	この列には、直前の 4 つのカテゴリーのいずれにも該当しないデータが含まれています。一般にこの設定が使用されるのは、カスタムの変数および数式で処理されるデータが列に含まれる場合です。

この例のテキスト・データには、“Location”ディメンションの要素と集約が含まれています。

- 属性は含まれていません。
- データ値は含まれていますが、それらの値は、他のディメンションの要素と同様に、“Location”ディメンションの作成には関係しません。

“Location”ディメンションの変数を定義するには

手順

1. 「TurboIntegrator」ウィンドウの「**変数**」タブをクリックします。
2. 変数“Region”、“State”、および“City”の「**コンテンツ**」フィールドを、次のように設定します。

変数	**コンテンツ**
Region	集約
ステータス	集約

変数	コンテンツ
City	要素

- これで、変数“Region”は集約として識別されます。
- また、変数“State”も集約として識別されます。
- 変数“City”は、リーフ・レベル (非集約) 要素に識別されています。

変数のマッピング

データ・ソースの変数の識別が終了したら、その変数を要素と集約にマッピングする必要があります。

変数のマッピングは、「TurboIntegrator」ウィンドウで「**マップ**」タブをクリックして始めます。

「マップ」タブには、その他の複数のタブが含まれます。「キューブ」タブは常に使用可能です。他のすべてのタブは、「変数」タブで設定した列のコンテンツに基づいて使用できるようになります。例えば、列に要素が含まれていると指定した場合は、「ディメンション」タブが使用可能になります。列に集約が含まれていると指定した場合は、「集約」タブが使用可能になります (以下、同様)。

キューブのマッピングの無効化

ディメンションを作成する場合は、キューブ操作を実行しません。キューブのマッピングを防ぐには

手順

1. 「**キューブ**」タブをクリックします。
2. 「キューブの操作」ボックスで「**未対処**」を選択します。

ディメンションのマッピング

データ・ソースの列の中に、要素を含んでいると指定した列がある場合は、作成するディメンションにその要素をマッピングする必要があります。

手順

1. 「**ディメンション**」タブをクリックします。
2. 「ディメンション」フィールドに「**Location**」と入力します。

   同じディメンションに複数の要素をマッピングする場合は、ディメンション名を要素ごとに入力します。

   「ディメンション」列に新しいディメンション名を入力すると、「操作」列がデフォルトで「作成」になります。

   既存のディメンション名を入力する場合は、そのディメンションを再作成するか更新するかを選択できます。「再作成」操作を選択すると、既存のディメンション内の要素が削除され、データ・ソースのデータで置き換えられます。「更新」操作を選択した場合は、データ・ソースに新しい要素があればその要素でディメンションが更新されます。

3. 適切な「要素の型」メニューから各要素の型を選択します。要素の型は、要素変数で識別されたデータの型を示します。 Xcelerator では、この設定はほとんどの場合に「数値」です。
4. 「**要素の順序**」オプションを選択します。要素の順序によって、処理中に要素をディメンションに追加する方法が決まります。

   この例のデータには 1 つの数値要素が含まれており、新しい“Location”ディメンションにマッピングされます。操作の完了した「ディメンション」タブは、次のようになります。

データのマッピングの無効化

ディメンションを作成する場合、データのマッピングは実行しません。

例

「キューブ・マッピング」タブ 14 ページの『キューブのマッピングの無効化』 で「未対処」を指定したので、この「データ」タブは使用できなくなります。

集約のマッピング

データ・ソースの列の中に、集約を含んでいると指定した列がある場合は、作成するディメンションの集約パスをマッピングする必要があります。

手順

1. 「**集約**」タブをクリックします。

   このタブには、集約と定義されている変数 (“Region”および“State”) が表示されます。

   ディメンションの集約階層は、集約変数ごとに子の変数を指定して定義できます。
2. “Region”集約変数の直属の子は“State”です。「Region」集約の「子の変数」フィールドの右不等号括弧ボタンをクリックし、「**State**」を選択して、「**OK**」をクリックします。
3. “State”集約変数の直属の子は“City”です。「State」集約の「子の変数」フィールドの右不等号括弧ボタンをクリックし、「**City**」を選択して、「**OK**」をクリックします。
4. 集約ごとに、「**コンポーネントの順序**」ボタンをクリックします。「コンポーネント要素の順序設定」ダイアログ・ボックスが開きます。
5. 「**自動**」、「**名前**」、および「**昇順**」をクリックします。

   注：同じディメンション内に複数の集約を設定する場合は、すべての集約で「コンポーネント要素の順序設定」の設定を同じにする必要があります。同じディメンション内に 2 つの集約がある場合、それぞれの「コンポーネント要素の順序設定」設定が異なっていると、プロセスを保存して実行しようとしたときに、「ソート情報が一致しません」とのエラーが発生します。

TurboIntegrator プロセスの保存と実行

データ・ソースの定義と変数の設定が終了したら、TurboIntegrator プロセスがコンパイルされ、保存されます。ディメンションを作成するには、完了したプロセスを実行します。

手順

1. TurboIntegrator のメニュー・バーから「**ファイル**」 → 「**保存**」をクリックします。

   「プロセスに名前を付けて保存」ダイアログ・ボックスが開きます。
2. プロセスの名前を入力し、「**保存**」をクリックします。

   コンパイル時と保存時に Xcelerator でエラーが検出されると、エラーの内容を示すエラー・メッセージが表示されます。「TurboIntegrator」ウィンドウはアクティブなままなので、エラーがあればすぐに修正できます。

   Xcelerator では、プロセスは、サーバー・エクスプローラーの「プロセス」の下にサーバー・オブジェクトとして保存されます。これで、プロセスを実行したり、修正したりすることができます。

   プロセスを実行してディメンションを作成するには、TurboIntegrator のメニュー・バーから「**ファイル**」 → 「**実行**」をクリックします。サーバー・エクスプローラーからプロセスを直接実行することもできます。それには、プロセスを選択し、「**プロセス**」 → 「**プロセスを実行**」をクリックします。

   プロセスが正常に実行されると、 Xcelerator から確認メッセージが発行されます。

   Xcelerator がプロセスを実行できない場合は、実行時に発生したエラーの詳細を記述したダイアログ・ボックスが表示されます。

   “NewEngland.cma”が処理されると、“Location”ディメンションが新規に作成されます。

テキスト・ファイルからのキューブの作成

また、TurboIntegrator はテキスト・ファイルからキューブ全体を作成できます。この手順では、いくつかのディメンションと要素も構築され、データ操作も実行されます。

キューブを構築する手順は、ディメンションを構築するプロセスと似ています。

1. データ・ソースを Xcelerator に定義します。 17 ページの『キューブのデータ・ソースの定義』を参照してください。
2. Xcelerator が遭遇する変数を特定します。 18 ページの『キューブ変数の定義』を参照してください。
3. 異なる変数を、生成されるキューブ内の異なるデータ・タイプにマッピングします。 18 ページの『ディメンションへのキューブ要素変数のマッピング』、 19 ページの

ページの『キューブのデータ変数のマッピング』、 18 ページの『キューブ変数のマッピング』、および 19 ページの『集約変数のマッピング』を参照してください。

4. プロセスを保存して、実行します。 19 ページの『キューブ・プロセスの保存と実行』を参照してください。

Xcelerator には、“TI_data”というサンプル・データ・ディレクトリーが含まれています。“TI_data”には、“import_cube.csv”というファイルがあります。この例では、“import_cube.csv”からキューブを構築する方法について説明します。

キューブのデータ・ソースの定義

テキスト・ファイルからキューブを作成する最初の手順は、データ・ソースの定義です。

手順

1. サーバー・エクスプローラーの左側のウィンドウで「**プロセス**」アイコンを右クリックし、「**新規プロセスの作成**」を選択します。
2. 「TurboIntegrator」ウィンドウの「**データ・ソース**」タブをクリックします。
3. 「データ・ソース・タイプ」で「**テキスト**」を選択します。
4. 「データ・ソース名」フィールドの横にある「**参照**」ボタンをクリックし、“TI_data”ディレクトリー内の「**import_cube.csv**」ファイルを選択します。デフォルトのインストール・ディレクトリーを使用していた場合、“TI_data”ディレクトリーの完全なパスは次のとおりです。

   ```
   C:\Program Files\Cognos\TM1\Custom\TM1Data\TI_Data.
   ```

5. 「区切り記号の種類」を「**区切り付き**」に設定し、「区切り記号」として「**コンマ**」を選択します。

   この例では、「引用符」および「タイトル・レコード数」フィールドは無視します。

6. 「小数点記号」がピリオド (.) で、「桁区切り記号」がコンマ (,) になっていることを確認します。
7. 「**プレビュー**」をクリックして、データ・ソースの最初の数レコードを表示します。

   “import_cube.csv”の各レコードには 6 つのフィールドがあります。最初の 5 つのフィールドには、要素名として Xcelerator にインポートされる情報が含まれています。 6 番目の列にはキューブのデータが含まれています。

変数名	変数の型	サンプル値	コンテンツ
V1	文字列	Actual	無視
Massachusetts	文字列	Argentina	無視
V3	文字列	S Series 1.8 L Sedan	無視
Units	文字列	Units	無視

変数名	変数の型	サンプル値	コンテンツ
1 月	文字列	1 月	無視
V6	数値	313.00	無視

キューブ変数の定義

TurboIntegrator に対してソース・データを特定した後、ソースのフィールドごとに内容を識別する必要があります。

手順

1. 「**変数**」タブをクリックします。各変数には、TurboIntegrator によってデフォルト値が設定されます。
2. 各変数について、それぞれの「変数の型」メニューから型を選択します。

   この例の場合、「変数の型」フィールドを変更する必要はありません。各変数の型は、 Xcelerator によって正しく識別されます。
3. 各変数について、それぞれの「コンテンツ」メニューからコンテンツ・タイプを選択します。

   この例の場合、V6 を除くすべての変数を「要素」と識別する必要があります。V6 は「データ」と識別する必要があります。

キューブ変数のマッピング

これでデータ、要素、および集約の変数は識別されました。ここからは、変数をマッピングして、新しいキューブを作成する手順を指示する必要があります。

手順

1. 「**マップ**」タブをクリックします。
2. 「**キューブ**」タブをクリックします。
3. 「キューブの操作」で「**作成**」を選択します。
4. 「キューブ名」フィールドに「**import_cube**」と入力します。
5. 「データ操作」で「**値の保存**」を選択します。
6. 「キューブのログ記録を有効化」オプションはオンにしないでください。キューブのログ記録を有効にすると、 Xcelerator は処理中のキューブ・データの変更を記録します。新しいキューブを作成するときに、変更内容をログに記録する必要はありません。

ディメンションへのキューブ要素変数のマッピング

「要素」型を持つと識別したすべての変数を、それぞれのディメンションにマッピングします。

手順

1. 「**ディメンション**」タブをクリックします。
2. 次の表に従って、「ディメンション」タブの値を設定します。

要素の変数	サンプル値	ディメンション	キューブ内の順序
Actual	Actual	actvsbud2	1
Argentina	Argentina	region2	2
V3	S Series 1.8 L Sedan	model2	3
Units	Units	measures	4
1 月	1 月	month2	5

3. すべての要素変数について、「操作」は「**作成**」に設定し、「要素の型」は「**数値**」に設定します。

キューブのデータ変数のマッピング

この例のデータ変数は V6 の 1 つのみです。このデータ変数はマッピングの必要がありません。TurboIntegrator によって自動的にマッピングされます。この例では、データ・タブも有効になりません。

データがキューブに追加されるときは、作成されたディメンションの交差部分に追加されます。「変数」タブで複数の変数がデータとして定義されている場合は、データをキューブに追加する際の位置に関する情報を指定することが必要になります。

データ値をキューブにマッピングする詳細な例については、TurboIntegrator Tutorial を参照してください。

集約変数のマッピング

この例の場合は、「変数」タブで集約として定義された変数がありません。この例では、「集約」タブは有効になりません。

集約をキューブにマッピングする詳細な例については、TurboIntegrator Tutorial を参照してください。

キューブ・プロセスの保存と実行

プロセスを実行するには、その前にプロセスを保存して名前を付ける必要があります。

手順

1. 「**Execute**」ボタンをクリックします。

   プロセスを保存して実行するには、以下を実行します。

   Xcelerator は、プロセスに名前を付けて保存するように指示します。
2. プロセスを“create_newcube”として保存します。

   数秒後に、プロセスが正しく実行されたことを示す確認画面が表示されます。
3. サーバー・エクスプローラーを開きます。キューブ“import_cube”が作成され、データが移入されています。また、必要なディメンションもすべて作成されています。

第 4 章 ODBC ソースからのインポート

TurboIntegrator を使用して、リレーショナル・データベース・テーブルのデータからキューブおよびディメンションを作成できます。そのためには、マシン上に次のソフトウェアが必要です。

- TurboIntegrator を実行するマシンと同じマシンにインストールされているリレーショナル・データベース用のクライアント・ソフトウェア。
- リレーショナル・データベースに対して確立された ODBC データ・ソース。データ・ソースは、Windows の「データ ソース」コントロール・パネルで構築します。

いったん ODBC データ・ソースの定義が完了すると、リレーショナル・データからキューブやディメンションを作成する手順は、テキスト・ファイルからキューブやディメンションを作成する場合とまったく同様です。TurboIntegrator で ODBC ソースを使用し、順を追ってオブジェクトを作成する完全チュートリアルについては、TurboIntegrator Tutorial を参照してください。

注: Xcelerator が Solaris または AIX® の Oracle ODBC ソースにアクセスするには、DataDirect ドライバーが必要です。これらのドライバーは Xcelerator に同梱されていないため、別個に入手する必要があります。

Unicode と DNS

バージョン 11g クライアント/ODBC ドライバーを使用して Oracle データベースから Unicode データをインポートするように DSN を設定する場合は、「Application(アプリケーション)」タブで「Enable Closing Cursors (カーソルのクローズを有効にする)」オプションを必ず指定してください。TI プロセスは、このオプションが指定されていなければ失敗します。

Oracle 11g ODBC ドライバーは、SqlFreeStmt の SQL_CLOSE オプションを完全にサポートしていません。

ODBC データ・ソースの定義

ODBC データ・ソースを定義するには：

手順

1. サーバー・エクスプローラーを開きます。
2. プロセスを作成するサーバーの下の「**プロセス**」アイコン  を右クリックし、「**新規プロセスの作成**」を選択します。

   「TurboIntegrator」ウィンドウが開きます。
3. 「データ・ソース・タイプ」ボックスの最上部の「**ODBC**」を選択します。
   ODBC ソースの定義に必要なフィールドが表示されます。

4. 「**参照**」をクリックし、ODBC データ・ソース名を選択します。 Xcelerator サーバーが実行されているコンピューター上で定義されているデータ・ソースのみがアクセスできます。
5. このソースを使用する必要がある場合は、ターゲット・データベースの有効なユーザー名とパスワードを「**ユーザー名**」と「**パスワード**」フィールドに入力します。
6. 「**クエリー**」ボックスに、ソースからデータを抽出する SQL クエリーを入力します。SQL クエリーの構文と書式は、使用するデータベースのタイプによって異なります。例えば、Microsoft Access Database を使用する場合は、Microsoft Access を実行し、データベースを開き、SQL ビューを使用して、SQL ステートメントをこの「クエリー」ウィンドウにコピーします。

   注: クエリーで参照するテーブル名にスペースが含まれる場合は、名前を二重引用符で囲む必要があります。
7. 「**プレビュー**」をクリックします。

   クエリーが有効であり、接続が正しく定義されている場合は、ターゲット・データベース・テーブルの最初の 10 レコードが「TurboIntegrator」ウィンドウに表示されます。

   ODBC 変数を定義する手順については、データ・ソースの変数の識別を参照してください。

   ODBC のマップ指示を定義する手順については、変数のマッピングを参照してください。

   TurboIntegrator プロセスを保存して実行する方法の詳細については、TurboIntegrator プロセスの保存と実行を参照してください。

MDX 文からの TurboIntegrator プロセスの生成

このセクションでは、MDX 文を使用して ODBO データ・ソースのデータを抽出し、データを Xcelerator にインポートする方法について説明します。

MDX 文は別のユーティリティーを使用して生成し、その MDX 文を、 Xcelerator へのインポートを最終的に行う MDX 文のベースにすることをお勧めします。

データをインポートするときは、列数の限られた MDX 文で始めることが重要です。MDX 文の中には、多数の列を生成するものがあります。このようなクエリーは、インポートの起点として現実的ではありません。

列数を制限するには、例えば、関心がある数値データのみを列に配置します。

MDX TurboIntegrator プロセスの構築

実用的なデータを戻す MDX 文を作成したら、TurboIntegrator プロセスを構築できます。

次の手順を実行して、処理を開始します。

手順

1. サーバー・エクスプローラーで「**プロセス**」を右クリックし、「**新規プロセスの作成**」を選択します。「TurboIntegrator」ウィンドウが開きます。
2. 「データ・ソース・タイプ」ボックスで「**ODBO**」をクリックし、「**MDX クエリー**」を選択します。
3. 必要な接続パラメーターを「TurboIntegrator」ウィンドウの「接続」タブに入力します。接続パラメーターは、ベンダー固有のパラメーターです。
4. 「**接続**」をクリックします。正しく接続すると、接続ボタンがグレー表示になり、「MDX クエリー」タブに進むことができます。
5. 「**MDX クエリー**」タブをクリックします。
6. MDX クエリーをこのタブに入力します。ベースとなる MDX クエリーを別のアプリケーションから切り取って、このタブに貼り付けることもできます。
7. 「**変数**」タブをクリックします。MDX 文で生成される列ごとに、1 つの変数が TurboIntegrator によって生成されます。

   行見出しのある列は、一般にディメンション要素としてマッピングされます。データ要素のある列は、データとしてマッピングされます。
8. 変数のマッピングを参照して、変数を Xcelerator 構造にマッピングします。
   ODBO データ・ソースへの接続と MDX 文の定義が終了すると、TurboIntegrator プロセスを完了する手順は、ODBC データ・インポートの場合と同じです。

第 5 章 Xcelerator のビューまたはサブセットからのインポート

IBM Cognos Xcelerator TurboIntegrator により、キューブ・ビューからデータを抽出し、そのデータを使用して新しいオブジェクトを作成できます。 Xcelerator ビューを使用するためのプロセスを構築する手順は、インポート用に設計したデータのビューを初めに構築することを除き、他のデータ・ソースを定義するのに使用する手順に類似しています。

Xcelerator キューブ・ビューの中には、正しくインポートできないものもありますが、TurboIntegrator から特定のパラメーターを使用してビューを構築することで、インポート処理をいつでも正しく実行できるようになります。

データ・ソースとして Xcelerator キューブ・ビューを使用

データ・ソースとしてキューブ・ビューを定義することができます。

これを行うには、データ・ソースを定義 (『キューブ・プロセスの作成』を参照) してから、テキスト・ファイルのインポートで説明されている手順に従います。

キューブ・プロセスの作成

データ・ソースとしてキューブ・ビューを使用するプロセスを作成することができます。

手順

1. サーバー・エクスプローラーで「**プロセス**」を右クリックし、「**新規プロセスの作成**」を選択します。
2. 「**ICAS**」をクリックして、「データ・ソース・タイプ」ボックスで「**キューブ・ビュー**」を選択します。「データ・ソース名」フィールドが表示されます。
3. 使用可能なビューのリストから選択するために、「**参照**」をクリックします。「サーバー・キューブ・ビューの参照」ダイアログ・ボックスが表示されます。
4. インポートするデータを持つキューブを選択します。
5. データ・ソースとして使用するビューが既に存在する場合は、そのビューを選択します。

   そのようなビューが存在しない場合は、「**ビューを作成**」をクリックし、「ビュー抽出」ウィンドウを開いてビューを作成します。ビューを作成したら、「サーバー・キューブ・ビューの参照」ダイアログ・ボックスでそのビューを選択します。
6. 「**OK**」をクリックします。

   これにより、選択したビューが TurboIntegrator プロセスのデータ・ソースとして表示されます。

   テキスト・ファイルのインポートで説明されている手順に従って、 Xcelerator ビューのインポートを完了します。

データ・ソースとして Xcelerator サブセットを使用

TurboIntegrator により、 Xcelerator ディメンション・サブセットからデータを抽出し、その情報を別の Xcelerator オブジェクトに移すことができます。次の例では、“Region”ディメンションの“Europe”集約が抽出され、“Region_Europe”という新しいディメンションを構成するために使用されています。

ディメンション・サブセットから情報を抽出するときのターゲット・オブジェクトは、一般に別のディメンションです。ディメンション・サブセットから抽出した情報を基にキューブを構築することはできません。

Xcelerator サブセットを使用したデータ抽出の手順は、他の TurboIntegrator プロセスと類似しています。開始するには、『データ・ソースとしてディメンション・サブセットを定義』を参照してください。

データ・ソースとしてディメンション・サブセットを定義

ディメンション・サブセットをデータ・ソースとして使用するプロセスを作成するには、次の手順を実行します。

手順

1. サーバー・エクスプローラーで「**プロセス**」を右クリックし、「**新規プロセスの作成**」を選択します。
2. 「**ICAS**」をクリックして、「データ・ソース・タイプ」ボックスで「**ディメンション・サブセット**」を選択します。キューブ・ビュー・ソースの定義に必要なフィールドが 1 つ表示されます。
3. 使用可能なサブセットのリストから選択するために、「**参照**」をクリックします。

   「サーバー・サブセットの参照」ダイアログ・ボックスが開きます。
4. インポートする要素が含まれるディメンションを選択します。
5. データ・ソースとして使用するサブセットを選択し、「**OK**」をクリックします。
6. 「**プレビュー**」をクリックします。

   選択したディメンション・サブセットの要素が、プレビュー・パネルに表示されます。

ディメンション変数の定義

この例では、サブセット・データ・ソースから抽出する要素が、“All Europe”という最上位集約の子として追加されます。

新しい集約を構築するには、次の手順を実行します。

始める前に

TurboIntegrator で変数を識別し、定義する方法の詳細については、キューブ変数の定義を参照してください。

手順

1. 「**変数を新規作成**」をクリックします。

   「変数」タブに変数“V2”が表示されます。
2. 「**数式**」をクリックします。

   「プロセス変数の数式」ダイアログ・ボックスが開きます。
3. 数式を次のように修正します。

   ```
   V2='All Europe';
   ```
4. 「**OK**」をクリックします。
5. “V2”の「変数の型」を「**文字列**」に変更します。
6. “V2”の「コンテンツ」設定を「**集約**」に変更します。

   次のセクションでは、サブセット・データ・ソースからインポートされる要素を“All Europe”集約に追加します。

ディメンション変数のマッピング

この例では、“Europe”という新しいディメンションを作成するために、「キューブ」、「ディメンション」、「集約」の各タブを設定する必要があります。“Europe”には、“All Europe”という集約が 1 つのみあります。

インポートされるデータを Xcelerator オブジェクトにマッピングする手順の詳細については、変数のマッピングを参照してください。

「キューブ」タブの設定

「キューブ」タブでは、次のオプションを設定します。

操作のタイプ	設定
キューブの操作	未対処
データ操作	値の保存

ディメンション・タブの設定

「ディメンション」タブでは、入ってくるデータを Xcelerator ディメンションにマッピングできます。この例では、“Europe”というディメンションが 1 つのみ作成されます。「ディメンション」タブでは、次のオプションを設定します。

オプション名	設定
要素の変数	Europe
ディメンション	Region
操作	作成
要素の型	数値

集約タブの設定

前の手順で追加した“All Europe”変数が、「集約」タブに表示されます。「サンプル値」が、数式で指定した値に設定されていることに着目してください。このプロセ

スに含まれる変数は 2 つのみなので、 Xcelerator は“region”変数を“V2”変数の子として正しく識別しています。「集約」タブの設定は、変更する必要がありません。

ディメンションの保存と実行

プロセスを保存して実行すると、 Xcelerator は、Europe という新しいディメンションを作成します。このディメンションには、「All Europe」という集約が 1 つ含まれ、その集約にはヨーロッパのあらゆる地域に対するリーフ要素が含まれています。

TurboIntegrator プロセスを保存して実行する方法の詳細については、TurboIntegrator プロセスの保存と実行を参照してください。

第 6 章 MSAS からのインポート

IBM Cognos Xcelerator TurboIntegrator により、Microsoft Analysis Services などの任意の OLE DB for OLAP (ODBO) データ・ソースからデータをインポートできます。このセクションでは、TurboIntegrator を使用して Microsoft Analysis Service からキューブとディメンションをインポートする方法を示します。

OLE DB for OLAP データ・ソース

OLE DB For OLAP データ・ソースは、次のパラメーターで識別されます。

- ODBO プロバイダー名
- ODBO の場所
- ODBO データ・ソース
- ODBO カタログ

ODBO プロバイダー名

この名前は、マルチディメンション・データベース・サーバーを識別する ODBO プロバイダーによって割り当てられます。例えば、 Xcelerator は "TM1 OLE DB MD Provider" を使用し、Microsoft Analysis Services は "Microsoft OLE DB Provider for OLAP Services 8.0" を使用します。

TurboIntegrator には、サーバーにインストール済みの ODBO プロバイダーのみが表示されます。

ODBO の場所

場所フィールドは、管理者によって ODBO プロバイダー・サービスの特定のインスタンスが割り当てられた場所の名前です。

このフィールドの正確な解釈は、ベンダー固有です。

ODBO データ・ソース

これは、管理者が特定の場所で一連のカタログに割り当てる名前です。Microsoft Analysis Services の場合、これは登録されたサーバーの名前です。

ODBC カタログ

これは、管理者がデータベースの特定のコレクション (キューブ、ディメンションなどのオブジェクト) に割り当てる名前です。Microsoft Analysis Services の場合は、データベースの名前になります。

接続文字列: MSAS と Xcelerator

Xcelerator OLE DB for OLAP プロバイダーは、プログラマーが柔軟に接続文字列を作成できるように修正されました。このように修正されたのは、 Xcelerator 接続文字列に MSAS 接続文字列との互換性を持たせるためです。

以前のバージョンの Xcelerator の場合、 Xcelerator OLE DB プロバイダー経由でログオンするには、次のフィールドが必要でした

フィールド	設定例
場所 IBM Cognos Analytic Server Admin Server ホストのマシン名。	MyServer
Datasource Xcelerator サーバーの名前。	Sdata
userID Xcelerator ユーザー名。	Admin
password Xcelerator ユーザーのパスワード。	Apple

上述のパラメーターを使用することもできますが、次の表に示すパラメーターを使用して、 Xcelerator にログオンすることもできます。このパラメーターは、TurboIntegrator から Microsoft Analysis Services に接続する場合にも使用されます。

フィールド	設定例
Datasource IBM Cognos Analytic Server Admin Server ホストのマシン名。	MyServer
カタログ Xcelerator サーバーの名前。	Sdata
userID Xcelerator ユーザー名。	Admin
password Xcelerator ユーザーのパスワード。	Apple

OLE DB for OLAP データ・ソースへの接続 (CAM 認証を使用する場合)

Xcelerator サーバーが Cognos Access Manager (CAM) 認証を使用するように設定されている場合は、ODBO データ・ソースへの接続を確立する際にサーバーで使用される CAM ネームスペース ID を指定する必要があります。

32 ビット・バージョンのサーバーを実行している場合は、TurboIntegrator の「接続」タブにある「その他の接続パラメーター」セクションで CAM ネームスペースを指定できます。 CAM ネームスペース ID は、次の書式で指定する必要があります。

```
Provider String="CAMNamespace=<CAM Namespace ID"
```

<*CAM* ネームスペース *ID*> は、内部で使用する CAM ネーム・スペース ID です (ネームスペースの記述名ではありません)。

64 ビット・バージョンのサーバーを使用している場合は、上述の書式と同じ書式に従って、接続文字列で CAM ネームスペース ID を指定する必要があります。例えば、次の接続文字列では、NTLM_NAMESPACE という CAM ネームスペース ID を指定しています。

```
Provider=TM1OLAP.1;Location=localhost;Data
Source=empty;UserID=tm1user;Password="abc123";
Provider String="CAMNamespace=NTLM_NAMESPACE";InitialCatalog=empty
```

64 ビットのサーバーを実行している場合は、TurboIntegrator のユーザー・インターフェースを使用して CAM ネームスペースを指定することはできません。接続文字列を使用する必要があります。

MAS キューブのインポート

この手順では、単純なキューブを Microsoft Analysis Services から Xcelerator にインポートする方法について説明します。

Microsoft Analysis Services のキューブを Xcelerator にインポートするには

1. **MAS データ・ソースへの接続を確立します。**
   『TurboIntegrator による Analysis Services への接続』を参照してください。
2. **インポートするキューブを指定します。**
   32 ページの『「ODBC キューブのロード」タブによるキューブの指定』を参照してください。
3. **ディメンションを定義します。**
   33 ページの『「キューブ・ディメンション」タブの使用』を参照してください。
4. **プロセスを保存して、実行します。**
   34 ページの『MAS プロセスの保存と実行』を参照してください。

TurboIntegrator による Analysis Services への接続

TurboIntegrator を使用して、Microsoft Analysis Services に接続するプロセスを作成します。

手順

1. Architect を実行し、有効なユーザー名とパスワードを使用してログオンします。
2. 「**プロセス**」を右クリックし、「**新規プロセスの作成**」を選択します。

「TurboIntegrator」ダイアログ・ボックスが開きます。

3. 「**ODBO**」オプションをクリックしてから、「**キューブ**」を選択します。

   ダイアログ・ボックスに、ODBO 接続文字列を作成するためのオプションが表示されます。
4. 接続パラメーターを、ダイアログ・ボックスに次のように入力します。

フィールド	値
ODBO プロバイダー	「**Microsoft OLE DB Provider for OLAP Services**」を選択します。
ODBO の場所	このパラメーターは空白のままにします。
ODBO データ・ソース	Analysis Services をホストするサーバーのマシン名を入力します。
ODBO カタログ	Analysis Services データベースの名前を入力します。例えば、Microsoft サンプル・データベースのデータをインポートする場合は、このフィールドに「**FoodMart 2000**」と入力します。
ODBO ユーザー ID	Analysis Services データベースに有効なユーザー名を入力します。
ODBO パスワード	Analysis Services データベースのこのユーザー名に有効なパスワードを入力します。
その他の接続パラメーター	ODBO サーバーの中には、他のパラメーターも指定しなければ正しく接続できないものがあります。そのパラメーターは、セミコロンで区切ってこのフィールドに入力します。

5. 「**接続**」をクリックします。正しく接続すると、「接続」ボタンがグレー表示になり、「ODBO キューブのロード」タブに進むことができます。

「ODBC キューブのロード」タブによるキューブの指定

「ODBO キューブのロード」タブでは、Analysis Services からインポートするキューブなどを指定できます。次の手順を実行して、このタブに必要事項を入力してください。

手順

1. 「**ODBO キューブのロード**」タブをクリックします。
2. キューブの操作を選択します。次の表で、この選択肢について説明します。

オプション	説明
キューブの作成	データとメタデータを ODBO データ・ソースからコピーし、Xcelerator 内に新しいキューブを作成します。このオプションは、インポートするキューブとディメンションがサーバー上に存在しない場合にのみ使用します。

オプション	説明
キューブの再作成	現存するキューブを破棄し、ODBO データ・ソースのデータとメタデータを使用してキューブを再作成します。このオプションは、キューブおよびディメンションが存在し、その内容を新しい構造とデータで置き換える場合にのみ使用します。
キューブの更新	既存の ODBO キューブのデータをコピーし、既存のキューブに挿入します。このオプションを実行しても、サーバー上のキューブとディメンションの構造は変更されません。
未対処	この画面のデフォルト値。「未対処」を指定したプロセスは、キューブのデータやメタデータに影響しません。このオプションは、プロセスのテストとデバッグに使用します。また、独自のカスタム操作を定義する場合にも使用してください。

この例では、「**キューブの作成**」を選択します。

3. 「**ロード元の ODBO キューブを選択**」をクリックし、 Xcelerator にインポートする Analysis Services キューブを選択します。
4. 「**ロード先の ICAS キューブを選択**」フィールドをクリックします。キューブの一意な名前を入力します。
5. 「データ操作」パネルで、「**値の保存**」を選択します。このオプションを指定すると、ODBO キューブのセル値がキューブに書き込まれます。「値の累計」オプションを選択すると、インポートされる値と合算することができます。

「キューブ・ディメンション」タブの使用

「キューブ・ディメンション」タブでは、 Xcelerator にインポートするディメンションをインポート時に操作できます。

デフォルトでは、ODBO キューブ内のすべてのディメンションがインポートされます。ディメンションは、＜*名前*＞_ として Xcelerator 内に作成されます。例えば、Analysis Services 内の“[customer]”ディメンションをインポートすると、 Xcelerator 内で対応するディメンションは“Customer_”になります。

このダイアログ・ボックスには、次のオプションがあります。

- ODBO ディメンションを、既存のディメンションにマッピングすることができます。それには、「**ICAS ディメンション**」列でディメンションをクリックし、別のディメンションを選択します。
- また、まったく新しいディメンションに ODBO ディメンションの要素をインポートすることもできます。「 ICAS ディメンション」列で対応するセルをクリックしてから、新しいディメンションの名前を入力してください。例えば、“customer_”ディメンションを“MyCustomerDim”というディメンションで置き換えます。
- インポートするディメンションごとに、「 ICAS ディメンションの操作」を選択する必要があります。次のいずれかのオプションを選択してください。

オプション	説明
作成	ODBO キューブのディメンション・データをインポートして、そのディメンション内のすべての要素を持つディメンションを新規に作成します。これはデフォルトの操作です。
フィルターのみ - MDX	ODBO キューブのディメンション・データをインポートし、その一部の要素を持つディメンションを新規に作成します。
未対処	このディメンションは ODBO データ・ソースからインポートしません。

MAS プロセスの保存と実行

「キューブ・ディメンション」タブでの変更作業が終了したら、 をクリックして、プロセスを保存および実行します。

「プロセスに名前を付けて保存」ダイアログ・ボックスが開きます。

新しいプロセスの名前を入力します。プロセスには、インポートするデータに関連する名前を付けてください。この例では、「**ODBO_Sales_Import**」と入力します。

Xcelerator によってデータがインポートされ、新しいキューブが作成されます。ダイアログが表示され、インポートの進捗状況が示されます。

MAS ディメンションのインポート

このセクションでは、Microsoft Analysis Services から Xcelerator にディメンションをインポートする方法について説明します。次の表は、Analysis Services に表示されるディメンションを表しています。

Xcelerator では、同じディメンション内の要素の名前はすべて一意であることが必要です。また、 Xcelerator では、要素の別名もすべて一意の名前であることが必要です。要素名を確実に一意にするために、 Xcelerator では、インポートするディメンション内の集約と要素には、そのすべての親の名前を角括弧で囲み、ピリオドで区切った名前が付けられます。

Xcelerator へのインポート後のサブセットの別名には、Analysis Services の要素名が設定されます。

MAS データをインポートする手順は、他のインポート・プロセスと似ています。

MAS 接続パラメーターの定義

Analysis Services のディメンションを Xcelerator にインポートするには、まず、Analysis Services に接続し、「ODBO ディメンション」オプションを選択します。次の手順を実行してください。

手順

1. Architect を実行し、有効なユーザー名とパスワードを使用してログオンします。
2. 「**プロセス**」を右クリックし、「**新規プロセスの作成**」を選択します。

   「TurboIntegrator」ダイアログ・ボックスが開きます。
3. 「**ODBO**」オプションをクリックしてから、「**ディメンション**」を選択します。
4. 接続パラメーターを、ダイアログ・ボックスに次のように入力します。

フィールド	値
ODBO プロバイダー	「**Microsoft OLE DB Provider for OLAP Services**」を選択します。
ODBO の場所	このパラメーターは空白のままにします。
ODBO データ・ソース	Analysis Services をホストするサーバーのマシン名を入力します。
ODBO カタログ	Analysis Services データベースの名前を入力します。例えば、Microsoft サンプル・データベースのデータをインポートする場合は、「**FoodMart 2000**」と入力します。
ODBO ユーザー ID	Analysis Services データベースに有効なユーザー名を入力します。
ODBO パスワード	Analysis Services データベースのこのユーザーに有効なパスワードを入力します。
その他の接続パラメーター	このフィールドは空白のままにします。

5. 「**接続**」をクリックします。接続ボタンがグレー表示になり、正しく接続されたことを示します。

「ODBO ディメンションのロード」タブの使用

Analysis Services に正しく接続したら、実行するディメンション・ロード・プロセスでのコピー元とコピー先のディメンションに関する情報を指定する必要があります。次の手順を実行してください。

手順

1. 「**ODBO ディメンションのロード**」タブをクリックします。
2. 「 Xcelerator ディメンションの操作」を選択します。次のいずれかのオプションを選択してください。

オプション	説明
ディメンションの作成	ODBO データ・ソースのディメンションをコピーして、新しいディメンションを作成します。
ディメンションの再作成	現存するディメンションを破棄し、ODBO データ・ソースのデータを使用してディメンションを再作成します。
ディメンションの更新	「ディメンションの更新」は、要素を挿入または削除するディメンションが Xcelerator に既に存在していることを前提としています。 • ODBO データ・ソースには要素が存在し、 Xcelerator には存在しない場合、その要素は、ディメンションに追加されます。 • Xcelerator には要素が存在するが、ODBO データ・ソースには存在しない場合、その要素はインポートの影響を受けません。ローカル・ディメンション内の要素は、変更されません。 • 要素が ODBO データ・ソースとローカル・ディメンションに存在する場合は、ODBO データ・ソースの要素がインポートされ、ローカル・ディメンション内に <element_name>_1 として作成されます。これによりディメンションのサイズが増加することに注意してください。
未対処	この画面のデフォルト値。このプロセスは、ディメンションに影響しません。

3. 「**ディメンションを含む ODBO キューブ**」のリストをクリックし、Analysis Services からインポートするディメンションが含まれるキューブを選択します。
4. 「**キューブ・ディメンション**」のリストをクリックし、インポートするディメンションを選択します。
5. ディメンションを更新または再作成する場合は、「**ロードする ICAS ディメンション**」のリストをクリックし、リストからディメンションを選択します。

   ディメンションを新しく作成する場合は、新しいディメンションの名前を「ロードする ICAS ディメンション」フィールドに入力します。

ディメンション MAS プロセスの保存と実行

「ODBO ディメンションのロード」タブの変更作業が終了したら、  をクリックして、プロセスを保存および実行します。

「プロセスに名前を付けて保存」ダイアログ・ボックスが開きます。

新しいプロセスの名前を入力してから、「**保存**」をクリックします。インポートが開始され、 Xcelerator はインポート状況を示すダイアログ・ボックスを表示します。

Xcelerator メッセージ・ログ

プロセスが終了したときに、軽度なエラーが Xcelerator メッセージ・ログに書き込まれる場合があります。その場合は、 Xceleratorでは、メッセージ・ボックスが開いて通知が表示されます。

サーバーのメッセージ・ログをチェックするには、サーバー・エクスプローラーで「 IBM Cognos Analytic サーバー」を右クリックし、「**メッセージ・ログの表示**」を選択します。エラーの詳細を表示するには、メッセージ・ログのエラーをダブルクリックします。

第 7 章 詳細プロシージャーの編集

このセクションでは IBM Cognos Xcelerator TurboIntegrator プロセスの管理について説明します。

一括ロード・モードの使用

一括ロード・モードでは、最適化された特別な単一ユーザー、または単一の日課/プロセス・モードで Xcelerator を実行できます。このモードは、他のアクティビティーがほとんど、あるいはまったく予定されていない場合に、専用タスクのパフォーマンスを最大限に引き上げることができます。

一括ロード・モードの使用例を以下に挙げます。

- メンテナンス操作を手動で実行する必要のある管理者。
- 大容量のデータを読み込む夜間帯。

Xcelerator は通常、複数のユーザー、日課、およびプロセスすべてが同時にデータにアクセスして実行できるマルチユーザー・モードで実行されます。一括ロード・モードでは、 Xcelerator サーバーは他のユーザー、日課、およびプロセスを一時的に停止して同時アクティビティーを阻止するため、マルチユーザー環境で必要な諸経費をなくすことができます。

一括ロード・モードは、実際にユーザーをログアウトするのではなく、 Xcelerator とのやりとりを一時的に停止するだけです。一括ロード・モードが終了すると、以前にログオンしていたユーザー全員が再びアクティブになり、 Xcelerator とユーザーのやりとりが再開されます。

一括ロード・モードは TI プロセス内で直接、または TM1® API を使って有効にすることができます。どちらの場合でも、コマンドを使用して一括ロード・モードを*有効*、*無効*にします。

一括ロード・モード使用時の注意

一括ロード・モードを使用する場合は、次の事項に注意してください。

- 一括ロード・モードでは、エンド・ユーザーへの警告メッセージは表示されません。一括ロード・モードの使用を計画し、それに合わせて調整を行う必要があります。
- 一括ロード・モードでは、単一のユーザーまたはプロセスだけを有効にすることができます。一括ロード・モード中は、サーバーへ新しい接続を確立することはできません。
- TI プロセスでは、再度同じ Xcelerator サーバーにログオンするコマンド行プログラムを起動するために `ExecuteCommand` を使用することはできません。ログオン試行に失敗します。
- 一括ロード・モードが有効な期間中に実行するよう予定されているスケジュール済みの日課は無効になり、実行されません。

一括ロード・モードの開始

サーバーが一括ロード・モードになると、他のスレッドによる処理はすべて一時停止されます。既存のユーザー・スレッドと実行中の日課は一時停止されます。一括ロード・モードを開始したスレッドのみが有効なままになります。一括ロード・モードを開始する日課以外のスケジュール済みの日課はすべて無効になります。すべてのシステム固有のスレッドと Top 接続も一時停止されます。

一括ロード・モードの終了

一括ロード・モードが無効になると、すべてのシステムとユーザー・スレッドが再開され、ユーザー・ログオンが可能になります。

TM1 API を使用して一括ロード・モードを有効にするカスタム・アプリケーションは、必要な TM1 API 関数を呼び出して、一括ロード・モードを終了する必要があります。ただし、クライアント接続が切断されている場合 (ネットワークのエラー、クライアントのログアウト、またはクラッシュや接続切断による)、サーバーは自動的に一括ロード・モードを終了します。

同様に、TI プロセス/日課を一括ロード・モードで実行している最中にプロセスが正常に終了するか、エラーで終了すると、サーバーは自動的に一括ロード・モードを終了します。

サーバーが通常のマルチユーザー・モードに戻ると、無効だった日課がすべて再度アクティブになり、通常のスケジュールに戻ります。実行するようにスケジュールされた日課が一括ロード・モードで止められていた場合は、即座には実行されませんが、スケジュールに従って実行されます。一括ロード・モードが有効な期間中のロックアウトを防ぐには、スケジュールされた日課の実行時間の調整が必要な場合があります。

一括ロード・モードの TurboIntegrator プロセス・コマンド

TI プロセスの「プロローグ」または「エピローグ」セクションのいずれかから「一括ロード・モード」を有効にすることができます。プロセスの「プロローグ」セクションの最初のステートメントまたは最初に非常に近いステートメントで「一括ロード・モード」を有効にすると効率的です。

プロセスで「一括ロード・モード」を有効にすると、「エピローグ」セクションの最後の行でしか無効にできません。プロセスの他の場所で「一括ロード・モード」を無効にしようとすると、プロセスはコンパイルしません。

モードが 1 つの TI プロセスで有効になると、明示的に無効になるまで、または日課が完了するまで有効のままです。つまり、日課内のプロセスでモードを有効にして、無効にする前に連続した TI プロセスを実行することができます。また、日課の特定の重要な部分に対してのみモードを使用して、「一括ロード・モード」の入出を繰り返すこともできます。

TI プロセスで「一括ロード・モード」の有効/無効を切り替えるには、次の TI コマンドを使用します。

```
EnableBulkLoadMode()
```

DisableBulkLoadMode() - この関数は、「一括ロード・モード」を使用する際に、TI プロセスの「エピローグ」セクションの最後の行でのみ使用することができます。

一括読み込みモードの TM1 C API 関数

次の TM1 C API 関数は、「一括読み込みモード」の有効と無効を切り替えるのに使用できます。

- TM1ServerEnableBulkLoadMode
- TM1ServerDisableBulkLoadMode

詳細については、『IBM Cognos Analytic Server *API Guide*』を参照してください。

プロシージャーの編集

データ・ソースを指定し、変数をすべて識別し、マッピング指示をすべて定義したら、「TurboIntegrator」タブで選択したオプションに基づいて、4 つのプロシージャーが生成されます。これらのプロシージャーは、「詳細」タブのサブタブになります。

次のプロシージャーがあります。

タブ	説明
プロローグ	データ・ソースを処理する前に実行する一連の文。
メタデータ	処理中にキューブ、ディメンション、などのメタデータ構造体を更新または作成する一連の文。
データ	データ・ソースの各レコードの値を操作する一連の文。
エピローグ	データ・ソースを処理した後で実行する一連の文。

これらのプロシージャーを編集して TurboIntegrator 関数と Xcelerator 規則関数を記述すると、TurboIntegrator の機能を拡張することができます。例えば、データ・プロシージャーを編集すると、値がゼロのレコードをスキップしたり、インポートしたレコードを外部ファイルに書き込んだりするようにプロセスに対して指示する文を記述できます。

使用可能なすべての TurboIntegrator 関数と Xcelerator 規則関数が記載されたリストは、 『IBM Cognos Xcelerator *参照ガイド*』を参照してください。

プロシージャーを編集するときは、各プロシージャーはプロセス中の特定の時点で特定の操作を実行するように設計されているということを忘れないでください。したがって、作成する操作や文は、そのプロシージャーに適したものにしてください。

注: プロセスのデータ・ソースが「なし」の場合は、プロセスの実行時に、データ・プロシージャーとメタデータ・プロシージャーは無視されます。「データ」および「メタデータ」サブタブ上の関数や文は実行されませんが、 Xcelerator からは、プロセスの一部が実行されなかったことを示すエラーや警告は発行されません。

プロシージャーを編集するには：

手順

1. 「**詳細**」タブをクリックします。
2. 編集するプロシージャーのサブタブをクリックします。
3. 文をテキスト・ボックスに入力します。次の行の*前*か、

   ```
   #****GENERATED STATEMENTS START****
   ```

   次の行の後ろに入力してください。

   ```
   #****GENERATED STATEMENTS FINISH****
   ```

   重要: ユーザーが作成する文は、生成された文の前または後ろに挿入できます。TurboIntegrator によって生成された文の中に挿入することはできません。

オンデマンドでのプロセスの実行

プロセスをオンデマンドで実行するには、サーバー・エクスプローラーでプロセスを選択し、「**プロセス**」 → 「**プロセスを実行**」を選択します。

TurboIntegrator から「**ファイル**」→「**実行**」  を選択することによりプロセスを実行することもできます。

TM1RunTI の使用

TM1RunTI は、オペレーティング・システムのコマンドを実行できる機能を持つアプリケーション内部から IBM Cognos Analytic Server (ICAS) TurboIntegrator (TI) プロセスを開始することのできるコマンド行インターフェース・ツールです。

このユーティリティーは、アプリケーションにおいて、並行実行可能なプロセスが確実に並行実行されるようにするため、TurboIntegrator のプロセスをグループ化することが必要な場合に、特に役立ちます。また、並行実行が不可能なプロセスが、正しい順序でシリアライズされるようにする上でも便利です。注意点として、TM1RunTI は、TurboIntegrator が終了するより前に終了する (制御が戻る) ことはありません。それで、呼び出し側プロセスが TM1RunTI の終了を待機している場合に呼び出しをシリアライズするために使用することができます。

非同期呼び出しと ICAS

Execute コマンドには 2 個のパラメーターがあります。その 2 番目のものは、同期呼び出しか非同期呼び出しかを記述するものです。 ICAS のツールは非同期 (パラメーター 0) でのみ呼び出すようにしてください。これは、TurboIntegrator プロセスによって保持されているロックをシステムが待機し、プロセスがユーティリティーを待機することによるサーバー・デッドロックが発生するのを回避するためです。これと同じアドバイスが、ExecuteCommand によって呼び出される実行可能機能が ICAS にログインする場合についても適用されます。

注: ツールが ICAS にログインする場合は、決して同期呼び出しを使用しないようにしてください。

TM1RunTI の構文

ここでは TM1RunTI の構文について説明します。

```
    tm1runti -?
 or tm1runti -help
 or tm1runt1 [<cmd_parm>...] [<ti_parm>...]

       where <cmd_parm> is one of:
        -i <filespec>
        -process <string>
        -connect <string>
        <connect_parm>...

       where <ti_parm> is:
         <parm_name> '=' <parm_value>

       where <connect_parm> is one of:
        -adminhost <string>
        -server <string>
        -user <string>
        <password_parm>
        -AdminSvrSSLCertAuthority <filespec>
        -AdminSvrSSLCertID <id>
        -AdminSvrSSLCertRevList <filespec>
        -AdminSvrSSLExportKeyId <id>
        -ExportAdminSvrSSLCert <T>
        -CAMNamespace <string>

       where <password_parm> is one of:
         -pwd <string>
         -passwordfile <filespec> -passwordkeyfile <filespec>
```

パラメーター

パラメーターは、設定ファイルで指定することも、コマンド行で渡すこともできます。コマンド行パラメーターは、設定ファイルに含まれるパラメーターよりも優先されます。このため、比較的静的なパラメーター (管理ホスト、サーバーなど) に関する永続的なデフォルト・パラメーターを設定しておき、デフォルトのオーバーライドを必要とするごく少数のパラメーターやデフォルトの指定が難しいごく少数のパラメーター (ユーザー名、TurboIntegrator プロセス名など) の値だけをあとで提供することができます。

パラメーターをコマンド行で渡す場合の形式は異なります。すべてのパラメーターは「-パラメーター名 値」という形式で渡されますが、「パラメーター名=値」として渡されるパラメーターはすべて TurboIntegrator プロセス・パラメーターとして扱われます。

パラメーターには次の 4 種類があります。

- コマンド・パラメーター

  使用する設定ファイル、使用する接続パラメーターのグループ、または実行する TurboIntegrator プロセスを指定するために使われます。
- 接続パラメーター

  サーバー名、ユーザー名、および ICAS サーバーへの接続に必要な他の情報を指定するために使われます。
- パスワード・パラメーター

ユーザー名と標準テキスト形式のパスワード、または暗号化されたパスワードと復号化に使われる関連する鍵ファイルを収めたファイル名のいずれかを指定できます。

- TurboIntegrator パラメーター

  指定された TurboIntegrator に渡されます。

コマンド行で指定するパラメーターは、ダッシュ (-) またはスラッシュ (/) で始まる必要があります。パラメーター値は、1 つのスペースによってパラメーター名と分けられます。値をそのまま指定することも、引用符で囲んで指定することもできます (スペースが含まれる場合)。

以下に例を挙げます。

```
tm1runti –server MyTM1Server –username John –pwd "my secret"
         ti_parm1=yes ti_parm2="my value"
```

TM1RunTI パラメーター

パラメーター	説明 値/必須かどうか/デフォルト
i	設定ファイルのパス 文字列/いいえ/なし
connect	このパラメーターを使用すると、user、pwd、CAMnamespace など、サーバー接続に使われるパラメーターを含む設定ファイル内のセクションを指定できます。 文字列/いいえ/なし
Process	呼び出す TurboIntegrator プロセスの名前 文字列/いいえ/なし
Help	ヘルプ・テキストをコマンド・ウィンドウ (標準出力) に表示します。 該当せず/いいえ/該当せず
?	コマンド行パラメーターの一覧をコマンド・ウィンドウ (標準出力) に表示します。 該当せず/いいえ/該当せず

接続パラメーター

接続パラメーターは ICAS ツール間で共通です。独自のセクション内にこれらを定義することで、再利用を促進し、複数コピーを保守よる負担やリスクを避けることができます。

パラメーター	値/必須かどうか/デフォルト	説明
adminhost	文字列/いいえ/なし	ICAS 管理ホスト
sever	文字列/いいえ/なし	ICAS サーバー名

パラメーター	値/必須かどうか/デフォルト	説明
user	文字列/いいえ/なし	ICAS または CAM 名
AdminSvrSSLCertAuthority	文字列/いいえ/なし	ICAS 管理サーバーの証明書を発行した認証局ファイルの完全なパス。
AdminSvrSSLCertID	文字列/いいえ/なし。API デフォルトは tm1adminserver	ICAS 管理サーバーの証明書が発行される先のプリンシパルの名前。**注:** このパラメーターの値は、Tm1admsrv.ini ファイルの SSLCertificateID パラメーターと同じでなければなりません。
AdminSvrSSLCertRevList	文字列/いいえ/なし	もともと ICAS 管理サーバーの証明書を発行した認証局によって発行された証明書失効ファイルの完全なパス。証明書失効ファイルは、証明書が既に失効した場合にのみ存在します。
ExportAdminSvrSSLCert	ブール値/いいえ/F	もともと ICAS 管理サーバーの証明書を発行した認証局の証明書を、実行時に Microsoft Windows 証明書ストアからエクスポートするかどうかを指定します。このオプションを選択する場合、ここで説明されているように、AdminSvrSSLEx-portKeyID の値を設定する必要もあります。適切な TM1Server 設定については、「*IBM Cognos TM1 インストールおよび設定ガイド*」を参照してください。
AdminSvrSSLExportKeyId	文字列/いいえ/なし	ICAS 管理サーバーの証明書をもともと発行した認証局の証明書を証明書ストアからエクスポートするために使われる識別キー。 このパラメーターが必要となるのは、ExportAdminSvrSSLCert=T を設定することで証明書ストアの使用を選択した場合だけです。適切な TM1Server 設定については、「*IBM Cognos TM1 インストールおよび設定ガイド*」を参照してください。

パラメーター	値/必須かどうか/デフォルト	説明
CAMNamespace	文字列/いいえ/なし	CAM ネームスペース ID。 **注:** これは CAM ネームスペースの名前ではありません。 CAM を使用して ICAS Server が認証を行う場合にのみ、この値が必要です。

TurboIntegrator パラメーター

これらのパラメーターは TurboIntegrator プロセスによって定義されます。これらは適切なタイプ (数値または文字列) である必要があります。

パラメーター	説明 値/必須かどうか/デフォルト
<*ti_parm*>	文字列値または数値 <*value*> を、指定されるパラメーター <*ti_parm*> に提供します (これは、実行される TurboIntegrator で受け入れられる有効なパラメーター名でなければなりません)。 <value>/いいえ/なし

パスワード・パラメーター

パスワードを提供するには、pwd パラメーターを使って平文で指定するか (この方法は推奨されていません)、または passwordfile パラメーターで指定される暗号化ファイルを使用します。

パラメーター	値/必須かどうか/デフォルト	説明
pwd	文字列/いいえ/なし	ICAS または CAM パスワード
passwordfile	文字列/いいえ/なし	指定されたユーザーの暗号化パスワードを含むファイルの絶対パス。パスを指定しない場合、 ICAS サーバー・ディレクトリーが想定されます。このオプションを使用する場合は、-pwd を使用できません。
passwordkeyfile	文字列/いいえ/なし	passwordfile を設定した場合、パスワードを復号化するために、鍵ファイルの絶対パスも必要になります。パスワード・ファイルと鍵ファイルを作成するには TM1Crypt ツールを使用できます。「*IBM Cognos TM1 インストールおよび設定ガイド*」を参照してください。

TM1RunTI 設定ファイル

設定ファイルを使用する場合も使用しない場合も、TM1RunTI は機能することができます。

設定ファイルを指定した場合、そのパラメーターが最初に読み取られます。

その後、コマンド行で指定されるパラメーターが、設定ファイルから得られたパラメーターをオーバーライドして使われます。設定ファイルを読み取る際、TM1RunTI は最初に設定ファイルの [TM1RunTI] セクションからパラメーターを取得します。

接続パラメーターが存在する場合、関連する [Connect <名前>] セクションからパラメーター値が取得され、[TM1RunTI] から読み取った値よりも優先して使われます。

また、-connect パラメーターをコマンド行で指定することもできます。その場合、設定ファイルで検出される接続パラメーターをすべてオーバーライドします。

設定ファイルには以下の内容が含まれます。

1. 1 つの TM1RunTI セクション。
2. 実行される TurboIntegrator プロセスを定義する 1 つ以上のセクション。
3. 接続パラメーターを定義する任意の数のセクション (存在しない場合もある)。

すべてのエントリーは第 1 カラムから始まる必要があります。 # で始まる行はコメントとして扱われます。

セクション名を大括弧 [ ] で囲む必要があります。 1 つのセクション名が繰り返される場合、最初のものだけが使用されます。

セクション内のパラメーターに関しては、

- パラメーター間にブランク行を入れることはできません
- 任意の順序で出現することができます
- keyword=value という形式で指定されます

パラメーター値に空白が含まれる場合、その値を引用符 (") で囲む必要があります。

Connect セクション

開発、テスト、実稼働などのさまざまなサーバー環境を簡単に保守できるようにするために、個別のセクションで環境ごとに接続パラメーターを指定できます。各セクションの名前を指定するには、接頭部「Connect -」の後にユーザー定義の名前を続けます。以下に例を挙げます。

```
[Connect – Production]

[Connect – Test]

[Connect – Development]
```

プロセス・セクション

複数のプロセス・セクションを指定できます。各セクションには、サーバー内のプロセスと一致する名前を指定します。

それぞれの TurboIntegrator プロセス・セクションを使用して、TurboIntegrator プロセスのパラメーターとそのデフォルト値を定義します。

同じ名前の複数のプロセス・セクションがある場合、最初のものだけが使用されます。

設定ファイルの例

この例では、[TM1RunTI] セクションと、1 つの TurboIntegrator プロセス (my_ti_process) のセクションが示されています。これらのパラメーターとそのデフォルト値 (コマンド行で指定されるパラメーターによってオーバーライドされる可能性があります) は、各セクション・ヘッダーの下で定義されます。

```
[TM1RunTI]
process=my_ti_process
connect=Production

[Process - my_ti_process]
num1="value1"
stringX="value2"
stringY="value3"

[Connect – Production]
adminhost=
server=MyTM1server
user="MyTM1AdminServer"
pwdfile="c:\tm1_admin_area\passwords\tm1_password.txt"
AdminSvrSSLCertAuthority=.\ssl\applixca.pem
AdminSvrSSLCertID=tm1adminserver
AdminSvrSSLCertRevList=
CAMNamespace=LOCAL_NTLM
```

処理ロジック

設定パラメーターとコマンド行パラメーターは、以下のような方法で処理されます。

1. -i によって指定される場合、設定ファイルが開かれ、[TM1RunTI] で指定されている接続オプションが最初に処理されます。
2. その後、[TM1RunTI] にある他のすべてのパラメーターが処理されます (接続パラメーターによる指定をオーバーライドすることがあります)。
3. 次に、コマンド行パラメーター `-connect` が存在する場合、それが処理されます。設定ファイルの関連する [Connect - <connection_name>] セクションから値をロードして、これまでの手順でロードされた値をオーバーライドします。
4. 残りのコマンド行パラメーターが処理されます。

例えば、上記の例の設定ファイルを tm1tools.config という名前で保存した後、以下を実行した場合、

```
tm1runti –i ".\tm1tools.config" –passwordkeyfile c:\keystore\prodkey.dat –connect prodsystem
```

-i パラメーターが指定されているため、ツールは以下の操作を行います。

1. 設定ファイルを開いて [tm1runti] セクションをロードします
2. [tm1runti] 内の接続パラメーターを検出すると、[Connect - `testsystem`] からパラメーター値をロードします
3. コマンド行パラメーターを次のように処理します
   a. 接続パラメーターを検出すると、[Connect - `prodsystem`] からパラメーターをロードします

b. passwordkeyfile の値を置き換えます

設定ファイルの名前と場所

コマンド行パラメーター -i を使用して、設定ファイル名を指定できます。これは、環境で複数の IBM Cognos Analytic Servers がサポートされている場合に特に便利です。その理由は、サーバーごとに別の設定ファイルを使用でき、それぞれ異なるパラメーターを使って別々のサーバー内の類似した名前のプロセスを定義できるためです。

TM1RuntTI のリターン・コードとエラー・メッセージ

次のエラー・メッセージは、TM1RunTI で使用されます。

リターン・コードとエラー・メッセージ

リターン・コード

メッセージ: 説明

0 **なし**: プログラムが正常に完了しました。

1 **パスワードが指定されていません**: 引数またはパスワード・ファイルの形式でパスワードが指定されていません。

短いヘルプ・テキスト: 必要なパラメーターが指定されていません (ユーザー、サーバー、プロセス)。短いヘルプが標準出力に送信されます。 -? と同等です。

***<n>* のパラメーターの数が無効です**: 実際にプログラムでサポートされる数より多いパラメーターが検出されました。 *<n>* 番目以降のパラメーターです。

2 **サーバー接続に失敗しました**: プログラムは ICAS サーバーに接続できませんでした。

3 **プロセス *<TI_name>* の呼び出しは完了しましたが、軽度のエラーがあります**: TurboIntegrator プロセスは完了しましたが、軽度のエラーがあります。

4 **プロセス *<TI_name>* の呼び出しがメッセージ付きで完了しました**: TurboIntegrator プロセスは完了しましたが、メッセージが戻されました。

5 **パスワードの取得中にエラーが発生しました**: プログラムは、パスワード・ファイルからパスワードを取得できませんでした。 標準エラー出力でこのメッセージの前にリストされている他のいずれかのエラー・メッセージが、問題の性質をより正確に示している可能性があります。

- **NULL キーが、読み取り中の <ファイル名> キー・パスから戻りました。**
- **NULL パスワードが、読み取り中の <ファイル名> パスワード・ファイルから戻りました。**
- **<ファイル名> のファイル状況を取得中にエラーが発生しました。**
- **<ファイル名> を開いているときにエラーが発生しました。**
- **キーのデータを割り振ることができません。**
- **<ファイル名> キー・ファイルを読み取り中にエラーが発生しました。**

6 **TI プロセス: *<TI 名>* がサーバー <サーバー名> に見つかりません**: TI プロセスが指定のサーバーに見つかりませんでした。

7 **TI プロセス: *<TI 名>* パラメーターを読み取れません**: TurboIntegrator プロセスからパラメーター情報を読み取れません。

8 **TI プロセス: *<TI 名>* の読み取り権限がありません**: 指定のユーザーには TurboIntegrator プロセスの読み取り権限がありません。

9 **呼び出しプロセス: *<TI 名>* が ProcessQuit を呼び出しました**: TurboIntegrator プロセスが ProcessQuit を呼び出しました。

10 **呼び出しプロセス: *<TI 名>* が異常終了しました。**: TurboIntegrator プロセスが異常終了しました。

11 **TI プロセス: *<TI 名>* が数値パラメーター <パラメーター名>=<パラメーター値> の読み取りに失敗しました**: 非数値の値が数値 TurboIntegrator パラメーターに渡されました。

99 **その他の TI エラー**: TurboIntegrator プロセスは完了しましたが、不特定のエラーがあります。

エラーは TM1API からも戻されます。これらのエラーは「(TM1 API エラー) <*xxx*>」と表示されます (<xxx> は TM1API で定義されている値)。

実行モードとエラー処理の制約事項

オペレーティング・システムのバッチ・スクリプト内から、または ICAS TurboIntegrator プロセス内から、スタンドアロン実行可能プログラムとして TM1RunTI を実行できます。

TurboIntegrator 内部から TM1RunTI を実行する最も単純な方法は、ExecuteCommand() 呼び出しを使ってこれを直接実行することです。以下に例を挙げます。

```
ExecuteCommand("tm1runti –i myconfig.config -connect prodserver –process update")
```

設定ファイルで接続 (および他の比較的静的な) パラメーターを定義する機能により、呼び出し側 TurboIntegrator プロセスから TM1RunTI に渡されるパラメーター・リストを単純化することができ、接続情報の一元管理によりメンテナンス上の負担を軽減できます。

ExecuteCommand() を使って TurboIntegrator プロセス内から TM1RunTI を直接実行する方法には、重要な制限があります。 TM1RunTI は障害が起こるとエラー・コードを返しますが、ExecuteCommand() はエラー・コードを返しません。呼び出し後にリターン・コードにアクセスする他のメカニズムが TurboIntegrator にはありません。

考慮すべき別の制限として、プロセスの現行ドライブとディレクトリーは、呼び出し側プロセス (サーバー) と同じ、つまりデータベース・ディレクトリーになります。この点については、4 ページの『TurboIntegrator 関数』で説明されています。

エラーを扱うには、ExecuteCommand によって呼び出されるバッチ・スクリプトから TM1RunTI を実行してください。こうすると、ERRORLEVEL 変数を介して CMD.EXE でエラー・リターン・コードを取得でき、標準エラー出力のリダイレク

トによってエラー・メッセージをログに記録 (またはインターセプト) することができます。これにより、アプリケーション設計者は、エラーを処理するためのさまざまなオプションを使用できます。例えば、

- エラー情報をデータベースに書き込む。
- エラー情報をファイルに書き込み、後続の TurboIntegrator プロセスでその情報を ICAS キューブにロードする。その後、キューブをレポート作成やアラートなどに使用できます。

  注: バージョン 9.5.1 以前では、これにより追加的なロック競合が生じることがあります。
- エラー情報を 1 つ以上のファイルに書き込んだ後、呼び出し側 TurboIntegrator プロセス内で TurboIntegrator プロセス関数 FileExists() を使用して、そのファイルの存在を検査する。バッチ・スクリプトで生成されたファイルが存在するかどうかに基づき、条件付きアクションをプロセスで実行できます。

TM1RunTI に関するその他の考慮事項

これらは、TM1RunTI を使用する際のいくつかの追加考慮事項です。

パスワード・セキュリティー

実働環境への配布の場合、このユーティリティーでは、コマンド行でのパスワードの使用は推奨されません。コマンド行でパスワードを使用する代わりに、暗号化されたパスワードを含むファイルを指定するために passwordfile パラメーターを使ってプログラムにパスワードを渡す必要があります。パスワードを復号化するための鍵ファイルも必要になり、これは passwordkeyfile パラメーターで指定されます。ツールを実行するユーザー名からアクセス可能な場所にこれらのファイルを格納できますが、他のユーザーからアクセスできないように、オペレーティング・システムの保護下に置く必要があります。

標準的な Xcelerator インストールに含まれる TM1Crypt ツールを使用すると、パスワードと鍵の組み合わせを生成できます。詳しくは、「*IBM Cognos TM1 インストールおよび設定ガイド*」を参照してください。

プラットフォーム移植性

このツールは、32 ビットと 64 ビットの Microsoft Window ユーティリティーとして、および AIX ユーティリティーとして使用可能です。プラットフォーム移植性、および tm1top とその他の ICAS サーバー・ツールとの整合性のために、実行可能モジュール名はすべて小文字になっています。

synchronized() を使用した TurboIntegrator プロセスのシリアライズ

synchronized() という名前の IBM Cognos Analytic Server (ICAS) TurboIntegrator (TI) 関数を TurboIntegrator スクリプトで使用すると、TurboIntegrator プロセスから成る指定したセットを強制的に順次実行することができます。

ICAS アプリケーション開発者は、ユーザー・アクションへの応答として実行される、またはバッチ・プロセスとして実行される TurboIntegrator (TI) プロセスを定義できます。明示的に禁止しない限り、複数の TurboIntegrator プロセスが並列的に実

行されることがあります。一部のアプリケーションでは、パフォーマンスや効率を改善するために複数の TurboIntegrator プロセスをシリアライズする必要があります。この新しい関数が導入される以前は、アプリケーション設計者は TurboIntegrator プロセスを確実にシリアライズするためにさまざまな手法を用いました。

1 つの手法は、オブジェクト・ロックを活用してプロセスのシリアライゼーションを強制することです。通常は、排他的アクセス・モードの準備ができたらキューブのロックを起動するように、ステータス値をキューブに書き込みます。しかし、並列相互作用 (PI) の導入が原因でこの方法が失敗する可能性があります。通常、データ書き込み操作は他のデータ書き込み操作と競合します。このため、キューブ内の TurboIntegrator プロセスを実行する場合、ロックを取得して実行を完了できるか、またはロックを取得できるまで待つ必要があるかのいずれかです。 PI モードでは、マルチ・バージョン並行制御により、複数の書き込み操作が即座にそれぞれの書き込みを実行できます。

PI を使用可能にすると従来の手法は有効ではなくなるため、TurboIntegrator プロセス・コードでシリアライゼーションを明示的に起動する synchronized() を使用できます。

この関数の使用法について、詳しくは「*IBM Cognos Express Xcelerator 参照ガイド*」の TurboIntegrator 関数についての章の『プロセス制御 TurboIntegrator 関数』のセクションを参照してください。

synchronized()

synchronized() という名前の IBM Cognos Analytic Server (ICAS) TurboIntegrator (TI) 関数を TurboIntegrator スクリプトで使用すると、TurboIntegrator プロセスから成る指定したセットを強制的に順次実行することができます。 synchronized() 関数は、次のような構文を使用します。

```
synchronized(string)
```

パラメーター

synchronized() は 1 つの必須パラメーター (ロック・オブジェクトを示すユーザー定義の名前) を受け入れます。複数の TurboIntegrator プロセスの実行をグループとしてシリアライズするために、それらの中でこのロック・オブジェクト名を使用できます。

lockName

値 = 文字列

必須か? = 必須

デフォルト - なし

同期化の対象となるロック・オブジェクトを示すユーザー定義の名前。名前は大/小文字を区別せず、組み込みスペースは無視されます。名前の長さは 1023 文字を超えてはなりません。

セマンティクス

1 つの TurboIntegrator プロセスは任意の回数にわたり、任意の数のロック・オブジェクトを指定して synchronized() を呼び出すことができます。シリアライズは synchronized() が呼び出された時点から、それを含むトランザクションが完了する時点まで有効になります。

例えばマスター・プロセス (Pm) またはマスター日課 (Cm) のサブプロセス (Ps) から synchronized() が呼び出された場合、Pm または Cm が完了するとこのロック・オブジェクトは「解放」されます。例外として、SaveDataAll (SDA) は処理の途中でトランザクションの実行を「終了」させます。これはロック・オブジェクトにも適用されます。

synchronized() 呼び出しは TurboIntegrator スクリプト内の任意の場所に配置可能ですが、検出された時点で TurboIntegrator プロセス全体にシリアライゼーションが適用されます。

synchronized() 呼び出しがスクリプトの「中ほど」に指定された TurboIntegrator プロセスがあり、その呼び出しの前に処理 O1 があるとします。この TurboIntegrator プロセスの 2 つのインスタンスが同時に開始する場合があり得ます。 2 番目のインスタンスで synchronized() 呼び出しに到達する前に、1 番目のインスタンスの実行が (synchronized() 呼び出しも含めて) すべて完了する可能性があります。この場合、ユーザーには、この 2 つのプロセスが並行して実行されたように見えます。一方、最初のプロセスが完了する前に 2 番目のプロセスの synchronized() 呼び出しに到達した場合、2 番目のプロセスは完了済みの処理 (O1) をすべて元に戻し、最初のプロセスの完了を待ちます。この場合、ユーザーには 2 つのプロセスがシリアライズされたように見えます。

このような混乱を回避し、最適な方法で synchronized() を使用するために、synchronized() 呼び出しを TurboIntegrator プロセスの最初のステートメントにすることをお勧めします (強制ではありません)。

例

TurboIntegrator プロセス P が 2 つのキューブ Cube_1 および Cube_2 を更新する必要があるとします。

他の TurboIntegrator プロセスもまた、Cube_1 または Cube_2 を更新する必要があるとします。

Cube_1 または Cube_2 を更新するすべての TurboIntegrator プロセスが一度に 1 つずつ実行されるように、P は以下の方法で synchronized() を呼び出すことができます。

```
sCube_1='Cube_1';
sCube_2='Cube_2';
sE1='Elm1';
sE2='Elm2';
sE4='Units';
sE5='Price';

Synchronized( sCube_1 );
Synchronized( sCube_2 );
```

```
CellPutn( 111, sCube_1, sE1, sE2 );
CellPutn( 9.99, sCube_2, sE4, sE5 );

# ...
```

Cube_1 または Cube_2 を更新する他の TurboIntegrator プロセスもまた、同様の方法で `synchronized( sCube_1 )` または `synchronized( sCube_2 )` (あるいは両方) を呼び出す必要があります。

この例では、キューブの名前と同じ 2 つのロック・オブジェクトの名前が選ばれました。しかし、ロック・オブジェクトの名前を他の ICAS オブジェクト (キューブ、ディメンション、サブセットなど) と必ずしも同じにする必要はありません。.

ロック・オブジェクトの保守と命名

ロック・オブジェクトは ICAS によって内部的に管理されます。ユーザーが明示的に作成したり、削除したりする作業は必要ありません。 synchronized() 呼び出しの中で、ロック・オブジェクトを名前で指定するだけです。

ロック・オブジェクト名では大/小文字が区別されず、組み込みブランクが無視されます。例えばロック・オブジェクトの名前が「Abc Def」である場合、「ABCDEF」、「ab cd ef」などの名前を使ってそのロック・オブジェクトを参照できます。つまり、`synchronized( 'Abc Def' )` の呼び出しを使用する TurboIntegrator プロセスの実行は `synchronized( 'ABCDEF' )` の呼び出しを使用するプロセスの実行とシリアライズされます。ロック・オブジェクト名の長さは 1023 文字を超えてはなりません。

実行の順序

同じロック・オブジェクトに対する synchronized() 呼び出しを含む TurboIntegrator プロセスから成るグループは、並行して実行されません。ただし、実際の実行順序はこの影響を受けません。並行して実行されない限り、それらの実行順序は他の多くの要因によって決定されます (アプリケーション設計、オペレーティング・システム・レベルのスケジューリングなど)。例えば 1 つの TurboIntegrator プロセスが他のプロセスによる更新に依存するケースなど、実行順序が重要な場合には、アプリケーション設計者が他の方法を使って適切な実行順序を実現する必要があります。

MaximumTIObjectLocks 設定パラメーター

MaximumTILockObjects パラメーターは、ロックされたオブジェクトのリストのサイズを制限します。「*IBM Cognos TM1 インストールおよび設定ガイド*」を参照してください。

管理者による TurboIntegrator セキュリティーの割り当て

TurboIntegrator プロセスを作成する管理者は、TurboIntegrator プロセスにセキュリティー権限を割り当てます。

TurboIntegrator プロセスを作成できるのは、プロセス作成に必要な管理権限を付与された管理者のみです。管理者は、さまざまな権限をプロセスに割り当てることが

できます。それらの権限は、その TurboIntegrator プロセスを実行しているユーザーにどんな権限が割り当てられているかには関係なく、そのプロセスに付与されます。

非管理ユーザーがインターフェースの中で TurboIntegrator プロセスを表示したり、そのプロセスを実行したりするには、プロセスに対する読み取りアクセスが必要です。しかし、TurboIntegrator プロセス自体は、管理者から割り当てられた権限を保持しています。

例えば、次のようなユーザーと管理者について考慮してみましょう。

- ユーザー U1 には cube_1 に対して読み取りアクセスしかありません。
- 管理者は、cube_1 への CellPutN を実行する TurboIntegrator プロセスを作成します。このプロセスには、キューブへの書き込みアクセス権が必要です。
- 管理者は、U1 に TurboIntegrator プロセスに対する読み取りアクセスを付与します。
- U1 はこの TurboIntegrator プロセスを実行することができます。ユーザーには cube_1 に対して読み取りアクセスしか付与されていなくても、プロセスは CellPutN を実行します。 U1 に cube_1 へのアクセスが付与されていない場合 (None アクセス) でも、同じ結果が得られます。
- TurboIntegrator プロセスへの読み取りアクセスしかないユーザーは、プロセスの表示と実行のみ可能です。そのユーザーは、送信する値を変更したりデータの配置場所を変更したりするためにプロセスを編集することができません。
- 上記のような状態は、ユーザーが TurboIntegrator プロセスを日課内から実行する場合にも当てはまります。

この TurboIntegrator プロセスに U1 からアクセスできないようにするには、IBM Cognos Xcelerator 管理者は、U1 に対して TurboIntegrator プロセスへの読み取りアクセスを付与しないようにしなければなりません。

第 8 章 日課によるプロセスの自動実行スケジュールの作成

プロセスは、オンデマンドで実行できます。また、*日課*を作成して、決まった間隔で実行することもできます。この 2 つの実行方法は、両方が同時に指定されてもかまいません。プロセスは、日課として自動実行するようにスケジュール設定されている場合でも、オンデマンドでいつでも実行できます。

日課は、1 つ以上のプロセスをユーザーが定義した頻度で実行する Xcelerator のオブジェクトです。日課を構成する要素は、次のとおりです。

- 実行するプロセスのリスト
- 日課を最初に実行する開始日時
- それ以降に日課を実行する頻度

定義した日課は、必要に応じて有効または無効にすることができます。

日課機能へのアクセス権は、ユーザー・グループのセキュリティー権限によって制御されます。サーバーに日課を作成するには、ADMIN または DataAdmin グループのメンバーであることが必要です。日課をサーバー・エクスプローラーに表示したり、日課を手動で実行するには、日課の読み取り権限が必要です。

プロセスを日課として自動実行するスケジュールは、TurboIntegrator から設定できます。

手順

1. 「TurboIntegrator」ウィンドウの「**スケジュール**」タブをクリックします。
2. 「**このプロセスのスケジュールに使用する日課名**」オプションを選択します。
3. プロセスの名前を隣のフィールドに入力します。デフォルトでは、プロセスの名前が日課に割り当てられます。
4. カレンダーの日付をクリックして、日課を初めて実行する開始日を指定します。
5. 時刻を入力して、日課を初めて実行するときの開始時刻を指定します。
6. 「日課の実行頻度」ボックスのフィールドにデータを設定して、日課の実行間隔を定義します。
7. 「**ファイル**」 → 「**保存**」を選択し、プロセスをスケジュール情報付きで保存します。

   TurboIntegrator からプロセスのスケジュールを設定すると、その日課が自動的に有効になり、指定した開始日時に実行されます。

   プロセス (またはプロセスの集合) の日課は、サーバー・エクスプローラーから直接作成することもできます。
8. サーバー・エクスプローラーで、日課を作成するサーバーの下にある「**日課**」アイコンを選択します。
9. 「**日課**」→ 「**新規日課の作成**」を選択します。

「日課設定ウィザード」が開きます。

10. 「使用可能」のリストで、日課を作成するプロセスを選択します。
11. 右矢印アイコンをクリックします。
12. 「**次へ**」をクリックします。
13. カレンダーの日付をクリックして、日課を初めて実行する開始日を指定します。
14. 時刻を入力して、日課を初めて実行するときの開始時刻を指定します。
15. 「日課の実行頻度」ボックスのフィールドにデータを設定して、日課の実行間隔を定義します。
16. 「**日課実行はアクティブです。**」ボックスに必要事項を入力します。
17. 「**終了**」をクリックします。

「日課に名前を付けて保存」ダイアログ・ボックスが開きます。

18. 日課の名前を入力し、「**保存**」をクリックします。

日課の開始時刻に関する重要な注意事項

日課の開始日時はグリニッジ標準時 (GMT) 形式で保存され、GMT に基づいて日課が実行されます。 Xcelerator では、夏時間への自動対応は行っていません。サーバー上のシステム・クロックが、地域の夏時間を使用するように設定されている場合は、夏時間が始まるときに日課の開始日時を編集しなければ、日課の実行スケジュールを同じ地域時間に保つことはできません。

夏時間が始まる日に、そのときの日付と希望する開始時刻を使用するように日課を編集してください。

夏時間が終わる日にも、同じように、そのときの日付と希望する開始時刻を使用するように日課を編集してください。

日課の編集

日課を開いて、日課設定ウィザードで編集するには：

手順

1. サーバー・エクスプローラーの左側のウィンドウで日課を選択します。
2. 「**日課**」 → 「**日課の編集**」を選択します。

日課の有効化

現在無効になっている日課を有効にするには：

手順

1. サーバー・エクスプローラーの左側のウィンドウで日課を選択します。
2. 「**日課**」 → 「**有効化**」オプションをオンに切り替えます。

日課の無効化

定期実行するように設定されている日課スケジュールを一時的に停止するには：

手順

1. サーバー・エクスプローラーの左側のウィンドウで日課を選択します。
2. 「**日課**」 → 「**有効化**」オプションをオフに切り替えます。

日課の削除

日課を削除するには：

手順

1. サーバー・エクスプローラーの左側のウィンドウで日課を選択します。
2. 「**日課**」 → 「**削除**」を選択します。

   注: アクティブな日課は削除できません。日課を削除するには、その前に無効にする必要があります。

オンデマンドでの日課の実行

日課をオンデマンドで実行するには：

手順

1. サーバー・エクスプローラーの左側のウィンドウで日課を選択します。
2. 「**日課**」 → 「**実行**」を選択します。

ChoreCommit の使用

ChoreCommit は日課のプロパティーの 1 つであり、これにより、日課に含まれるプロセスが単一トランザクションとしてコミットされるのか、それとも日課に含まれるプロセスが複数トランザクションとしてコミットされるのかを指定できます。

日課は、TurboIntegrator の一連のプロセスを単一のコミット・トランザクションとして実行します。最初のプロセスで獲得されたロックが、最後のプロセスの完了時まで保持されます。したがって、非常に長期間にわたってロックが保持されてしまう場合があります。オプションとして、ChoreCommit により、プロセス完了時に TurboIntegrator の各プロセスが 1 つのトランザクションとしてコミットされるような方法で日課を実行することができます。その場合、ロックは日課の期間ではなく単一プロセスの期間についてのみ保持されます。

Chore プロパティー

日課をセットアップする際、以下のように日課を指定できます。

- 単一コミット・モード

  すべてのプロセスが単一トランザクションとしてコミットされます。これは従来の動作であり、これがデフォルトです。
- 複数コミット・モード

コミットの必要なプロセスは、それぞれが処理された時点でコミットされます。

このプロパティーは、日課が非アクティブの場合にのみ変更可能です。

サーバー始動時における日課の実行

日課を、サーバー始動時に処理される「始動」日課として指定することができます。

サーバー始動時に日課を実行するよう指定するには、設定パラメーター StartupChores を使用して、サーバー始動の前に実行すべき日課のリストを指定します。日課は、順序に従って実行することの可能な一連のタスク (通常は TurboIntegrator プロセス) から成るセットです。このパラメーターについての情報は、「*IBM Cognos TM1 インストールおよび設定ガイド*」を参照してください。

始動時日課は、サーバーを処理前にセットアップするための手段として使用できます。始動時日課は、ユーザー・ログオンより前、かつ他の日課の処理開始より前に実行されます。

始動時日課はログインが可能になるより前に実行されるため、ユーザーが TM1Top で始動時日課をモニターすることはできません。そのため、サーバー・プロセスを強制終了する以外に、始動時日課をキャンセルする方法はありません。

付録 A. TurboIntegrator Tutorial

このチュートリアルは、IBM Cognos Xcelerator TurboIntegrator の高度な機能のガイドです。

このチュートリアルは、組織内で Xcelerator の実装と利用計画の開発を担当するユーザーを対象としています。通常は、上級ユーザーや開発者がキューブやディメンション、およびデータ・インポート・プロセスの作成、保守、および開発を担当します。このチュートリアルを進めるには、 Xcelerator の概念を十分に理解し、Xcelerator の機能について実践的な知識を持っている必要があります。

このチュートリアルでは、TurboIntegrator でディメンションとキューブを作成し、フラット・ファイルと ODBC データ・ソースをインポートする方法を示しています。また、詳細スクリプトの処理機能を使用して、TurboIntegrator の機能を拡張する方法も説明されています。さらに、このチュートリアルには TurboIntegrator の問題に対処するためのヒントも記述されています。

チュートリアルのデータ・ディレクトリーの設定

このチュートリアルでは、 Xcelerator に付属するサンプル・データを使用します。チュートリアルを開始する前に、サンプル・データを参照するようにローカル・サーバーのデータ・ディレクトリーを設定する必要があります。

データ・ディレクトリーを設定するには :

手順

1. サーバー・エクスプローラーの左側のペインで「**ICAS**」をクリックし、「**ファイル**」 → 「**オプション**」の順に選択します。

   「オプション」ダイアログ・ボックスが開きます。
2. 「ローカル・サーバーのデータ・ディレクトリー」の「**参照**」ボタンをクリックし、TurboIntegrator のサンプル・データ・ディレクトリーに移動します。

   サンプル・データ・ディレクトリーの名前は TI_data です。これは <*install_dir*>¥Custom¥TM1Data¥ ディレクトリーにあります。デフォルトのインストール・ディレクトリーにインストールしている場合、このサンプル・データ・ディレクトリーの絶対パスは、C:¥Program Files¥Cognos¥TM1¥Custom¥TM1Data¥TI_Data です。
3. 「オプション」ダイアログ・ボックスの「**OK**」をクリックしてデータ・ディレクトリーを設定し、ローカル・サーバーを再起動します。

TurboIntegrator の概要

Xcelerator TurboIntegrator を使用すると、データのインポート、メタデータの管理などのタスクを自動化するプロセスを作成できます。

プロセスは、次の要素で構成されるオブジェクトです。

- データ・ソースの説明
- データ・ソース内の各列に対応する一連の変数
- Xcelerator データベースの変数間の関係およびデータ構造を定義する一連のマップ
- プロローグ・プロシージャー (データ・ソースの処理前に実行する一連の操作で構成されます。)
- メタデータ・プロシージャー (キューブ、ディメンションなどのメタデータ構造を更新または作成する一連の操作で構成されます。)
- データ・プロシージャー (データ・ソース内のレコードごとに実行する一連の操作で構成されます。)
- エピローグ・プロシージャー (データ・ソースの処理後に実行されます。)
- プロセスを一般化し、異なる状況で使用できるようにするための一連のパラメーター。

TurboIntegrator を使用すると、ODBC ソース、ASCII ファイル、SAP ベース・データ、OLAP マルチディメンション・データ、 Xcelerator キューブ・ビュー、および Xcelerator ディメンション・サブセットからデータをインポートできます。

TurboIntegrator には、プロセス機能を拡張する一連の関数が含まれています。この関を使用して、データを ASCII ファイルや ODBC ソースにエクスポートするスクリプトや、条件式を使用してプロセスを制御するスクリプトを作成できます。このような TurboIntegrator 関数に加えて、プロセス定義には、STET 関数と UNDEFVALS 関数を除くすべての標準 Xcelerator 規則関数も組み込むことができます。

TurboIntegrator へのアクセスは、ユーザー・グループで管理されます。TurboIntegrator のすべての機能を利用し、プロセスをネットワーク上の Xcelerator サーバーに定義するには、ADMIN グループのメンバーであることが必要です。

TurboIntegrator 関数の作成を支援するインターフェースはありません。関数は、「詳細」タブ内の適切なサブタブに、直接手入力する必要があります。TurboIntegrator 関数に対する文字列引数は、一重引用符で囲む必要があります。TurboIntegrator ウィンドウで各関数の末尾を示すには、セミコロン (;) を含める必要があります。

TurboIntegrator プロセスの作成

プロセスは、5 つの手順で作成します。各手順は、「TurboIntegrator」ウィンドウのそれぞれのタブで、オプションを設定するか値を編集することで完了します。

プロセスの作成に必要な手順は、次のとおりです。

手順

1. データ・ソースの定義
2. 変数の設定
3. データのマッピング
4. 詳細スクリプトの編集
5. 作成したプロセスのスケジュール設定

   プロセスを作成するには、「TurboIntegrator」ウィンドウの各タブを順番に処理する必要があります。作業中のタブに必要なすべての情報を指定しない限り、次のタブへは進めません。

TurboIntegrator を使用したディメンションの作成

Xcelerator TurboIntegrator を使用すると、ODBC、ASCII ファイルなどのいずれかのデータ・ソースからディメンションの要素リストを作成できます。リストの要素が膨大にある場合 (カスタマー・ディメンションに何千もの名前があるような場合) は、この方法を使用すると最も速くリストを作成できます。

サンプルの ASCII ファイル

次に、後でディメンションの構築とデータのインポートに使用する区切り付き ASCII ファイル (“example.cma”) を示します。

```
"New England", "Massachusetts", "Boston", "SuperMart",
"Feb" , 2000000"New England", "Massachusetts", "Springfield", "SuperMart",
"Feb" , 1400000"New England", "Massachusetts", "Worcester", "SuperMart",
"Feb" , 2200000
```

このソース・ファイルの各レコードには 6 つのフィールドがあり、その中の 3 つを使用して“Example”ディメンションを作成します。最初の 2 つのフィールドは、集約要素になります。3 番目のフィールドは、数値要素になります。残りのフィールドは無視されます。

Example ディメンションは、ディメンション・エディターで次のような構造を持ちます。

New England

- Massachusetts
  - Boston
  - Springfield
  - Worcester

“Boston”、“Springfield”、および“Worcester”の数値は“Massachusetts”の合計に集約され、その合計は“New England”の合計に集約されます。

ASCII ファイルからのディメンションの作成

サンプル・ファイル“example.cma”を使用してディメンションを作成するには：

手順

1. サーバー・エクスプローラーの左側のウィンドウで、ローカル・サーバーの下にある「**プロセス**」を選択します。

2. 「**プロセス**」 → 「**新規プロセスの作成**」を選択します。

「TurboIntegrator」ウィンドウが開きます。

3. 「データ・ソース・タイプ」で「**テキスト**」を選択します。
4. 「データ・ソース名」の「**参照**」ボタンをクリックし、“TI_data”ディレクトリー内の**“example.cma”**を選択します。
5. 「サーバー上のデータ・ソース名」フィールドは空のままにします。
6. 「区切り記号の種類」を「**区切り付き**」に設定し、「区切り記号」を「**コンマ**」に設定します。
7. 「引用符」および「タイトル・レコード数」フィールドは無視します (入力ファイルに、引用符はなく、タイトル・レコードもありません)。

「小数点記号」はピリオド (.) に、「桁区切り記号」はコンマ (,) に設定されている必要があります。

8. 「**プレビュー**」ボタンをクリックして、“example.cma”ソース・ファイルのレコードを表示します。そのレコードを見て、データ・ソース内のレコード構造を確認できます。

変数の識別:

ソース・データを TurboIntegrator にロードしたら、ソースのフィールドごとに内容を識別する必要があります。ソース内の各フィールドには、 Xcelerator によって変数が割り当てられます。

手順

1. 「**変数**」 タブをクリックし、次に示す情報を表示します。このグリッドに表示される行は、データ・ソース内の各変数に対応しています。

変数名	変数の型	サンプル値	コンテンツ
V1	文字列	New England	無視
Massachusetts	文字列	Massachusetts	無視
Boston	文字列	Boston	無視
Supermart	文字列	Supermart	無視
2 月	文字列	2 月	無視
V6	数値	2000000	無視

グリッドの最初の列は、データ・ソース・フィールドごとに割り当てられる「変数名」です。独自の変数を割り当てるには、そのセルをクリックして新しい変数名を入力します。

2 番目の列は、各変数に割り当てられる「変数の型」です。これにより、ソース・フィールドのデータの型が識別されます。この型は、ドロップダウン・リストから選択して変更できます。

3 番目の列は「サンプル値」で、データ・ソースの最初のレコードの内容が表示されます。上図の場合、“example.cma”の最初のレコードの最初のフィールドの内容は“New England”です。

「コンテンツ」列は、各変数が表すデータ型 (要素、集約、データ、属性、その他、無視) を規定します。この例の最初の 3 つの変数は、地域階層の集約と要素を表しています。

2. 変数“V1”の「コンテンツ」列で、ドロップダウン・リストから「**集約**」を選択します。
3. 同じ操作を“Massachusetts”変数に対して実行します。
4. 変数“Boston”では、「**要素**」を選択します。
5. その他の変数は、ディメンションの作成に使用されないため、すべて「**無視**」を選択します。

変数名	変数の型	サンプル値	コンテンツ
V1	文字列	New England	集約
Massachusetts	文字列	Massachusetts	集約
Boston	文字列	Boston	要素
Supermart	文字列	Supermart	無視
2 月	文字列	2 月	無視
V6	数値	2000000	無視

変数のマッピング:

データ・ソース内の変数を識別したら、その変数を Xcelerator オブジェクトにマッピングする必要があります。

手順

1. 「**マップ**」タブをクリックしてから、「**キューブ**」サブタブをクリックします。
2. キューブは作成しないため、「キューブの操作」ボックスで「**未対処**」を選択します。
3. 「データ操作」は関係ありません (キューブの作成や更新は行いません)。このボックスは無視することができます。
4. 「キューブのログ記録を有効化」オプションは関係ありません (データ値は処理しません)。このオプションは、非選択のままにしておきます。
5. 「**ディメンション**」サブタブをクリックします。

   このグリッドには、内容タイプを「要素」と指定した変数が 1 行ずつ表示されます。要素の型を指定し、その要素をメンバーとするディメンションを指定する必要があります。

6. ディメンションは新規に作成するので、“Boston”変数の「ディメンション」列に「**Example**」と入力します。
7. 「操作」ドロップダウン・リストから「**作成**」を選択します。
8. 「要素の型」ドロップダウン・リストから「**数値**」を選択します。

   これで、“Boston”変数が、“Example”という新しいディメンションの数値要素としてマッピングされました。

   次は、集約として識別した変数をマッピングできます。
9. 「**集約**」サブタブをクリックします。

   2 つの集約変数が新しい“Example”ディメンションのメンバーであることは、Xcelerator で正しく認識されています。必要な処理で残っているのは、それぞれの集約について子の変数を特定することです。
10. “**V1**”集約変数について、「**Massachusetts**」を「子の変数」として選択します。
11. “**Massachusetts**”集約変数について、「**Boston**」を「子の変数」として選択します。
12. 「重み付け」は、どちらの集約変数でも編集しないでください。

    操作が終了すると、「集約」サブタブの表示は次のようになります。

集約変数	ディメンション	子の変数	重み付け	サンプル値	コンポーネントの順序
V1	例	Massachusetts	1.000000	New England	入力順
Massachusetts	例	Boston	1.000000	Massachusetts	入力順

マッピングはすべて完了です。特に必要ではありませんが、「詳細」タブをクリックしてから各種サブタブをクリックすると、TurboIntegrator によって作成されたスクリプト (“Example”を新規に作成し、集約と要素を挿入するスクリプト) を表示できます。TurboIntegrator のスクリプトについては、このチュートリアルの後半で詳しく説明します。

プロセスの保存と実行:

プロセスを保存して実行するには、以下を実行します。

手順

1. 「**実行**」ボタン  をクリックします。

   Xcelerator は、プロセスを保存するよう指示します。
2. “create_Example_dimension”という名前でプロセスを保存します。

   プロセスは、わかりやすい名前で保存することをお勧めします。

   数秒後に、プロセスが正しく実行されたことを示すメッセージ・ボックスが表示されます。

3. 「TurboIntegrator」ウィンドウを閉じます。
4. サーバー・エクスプローラーを開きます。
5. 作成された「Example」ディメンションを右クリックし、「**ディメンション構造の編集**」を選択します。

   ディメンション・エディターで“Example”ディメンションが開きます。

6.  をクリックして、ディメンション・メンバーを階層レベルでソートします。

   “Example”ディメンションが正しく作成されています。“New England”は集約要素で、“Massachusetts”(集約要素) を含みます。その“Massachusetts”は、数値要素の“Boston”、“Springfield”および“Worcester”を含んでいます。

ODBC ソースからのディメンションの作成

チュートリアルのこのセクションでは、手順を追って、ODBC データ・ソースからディメンションを作成します。その手順は、ASCII ファイルからディメンションを作成する方法によく似ています。

データ・ソースの定義:

チュートリアルを進める前に、Microsoft Access データベースを ODBC データ・ソースとして追加し、TurboIntegrator から使用できるようにする必要があります。

手順

1. Windows の「ODBC データ・ソース・アドミニストレーター」ダイアログ・ボックスを開きます。

   このダイアログ・ボックスにアクセスする手順は、実行している Windows のバージョンによって異なります。詳細については、Windows オンライン・ヘルプを参照してください。
2. 「ユーザー DSN」タブで「**追加**」ボタンをクリックします。

   「データ・ソースの新規作成」ダイアログ・ボックスが開きます。
3. 「**Microsoft Access ドライバー**」を選択し、「**完了**」をクリックします。

   「ODBC Access セットアップ」ダイアログ・ボックスが開きます。
4. 「データ・ソース名」フィールドに「**NewDB**」と入力します。
5. 「**選択**」ボタンをクリックします。

   「データベースの選択」ダイアログ・ボックスが開きます。
6. “TI_Data”ディレクトリーに移動して、“**NewDB.mdb**”を選択します。
7. 「**OK**」をクリックして、「データベースの選択」ダイアログ・ボックスを閉じます。
8. 「**OK**」をクリックして、「ODBC アドミニストレーター」ダイアログ・ボックスを閉じます。

   これで、Access データベースの NewDB が ODBC ソースとして使用可能になります。

データ・ソースのクエリー:

データ・ソースをクエリーするには :

手順

1. サーバー・エクスプローラーで「プロセス」アイコンを右クリックし、「**新規プロセスの作成**」を選択します。

   「TurboIntegrator」ウィンドウが開きます。
2. 「データ・ソース・タイプ」として「**ODBC**」を選択します。
3. 「データ・ソース名」フィールドの横にある「**参照**」ボタンをクリックします。
4. 「ODBC データ・ソース」ダイアログ・ボックスが開きます。
5. 「**NewDB**」を選択し、「**OK**」をクリックします。

   “NewDB.mdb”には“ACCOUNT”というテーブルが 1 つあり、このテーブルには 27 個のフィールドが含まれます。その中の 6 つのフィールドの情報を選択する SQL クエリーを作成します。ODBC クエリーでは、その基礎となる DBMS の SQL 方言を使用する必要があります。MS Access クエリーの構文と、Informix® クエリー、SQL Server クエリーなどの構文は異なります。

   クエリーを間違いなく正しい構文で作成するには、まず、基礎となる DBMS のクエリー機能を使用してクエリーを作成し、そのクエリーをコピーして TurboIntegrator の「クエリー」フィールドに貼り付けます。
6. 「クエリー」フィールドに、次の文を正確に入力します。

   ```
   SELECT [ACCOUNT_ID], [PARENT_ID], [NAME], [TYPE], [SALESREP],
   [SALESTEAM] FROM ACCOUNT;
   ```
7. 「**プレビュー**」をクリックし、クエリーが返す最初の 10 レコードを表示します。

SQL 内パラメーターを使用する:

「データソース」フィールドに使用するパラメーターを作成し、クエリーの一部としてそのパラメーターを呼び出します。

次の SQL ステートメントはその例です。

```
SELECT * FROM customer WHERE last_name = 'Smith'
```

パラメーターの Smith の値を 'pLastName' に置き換える場合は、SQL ステートメントは次のようになります。

```
SELECT * FROM customer WHERE last_name = '?pLastName?'
```

パラメーターの作成時は、以下の点に注意してください。

- 最初に ODBC ソースを使用して TI プロセスを作成する必要があります。これは「変数」を作成します。これによって、変数 DATASOURCEQUERY を使用して、「データソース」タブのクエリー・テキスト・ボックスを上書きすることができます。
- 返された設定の列の数は、TI プロセスが開発された時の数と一致している必要があります。
- 列のデータ・タイプも一致している必要があります。

- 文字列パラメーターの場合は、一重引用符でパラメーターを囲むことを忘れないでください。数値パラメーターの場合は、一重引用符を使用しないでください。例えば、数値を使用しているクエリーは次のようになります。

```
SELECT
* FROM customer WHERE last_name = ?pQuantity?
```

パラメーターを作成するには、TurboIntegrator プロセス・ダイアログ・ボックスの「詳細設定」タブを使用してデフォルトの PO パラメーターを使用したいパラメーターに交換します (例えば、 **pLastName**)。

変数の識別:

ソース・データのクエリーが終了したら、クエリー結果のフィールドごとに内容を識別する必要があります。

手順

1. 「**変数**」タブをクリックします。

   「変数名」列に、データベースの列名が正しく設定されていることに着目してください。
2. 「コンテンツ」列の選択内容を、次のように変更します。

変数名	コンテンツ
ACCOUNT_ID	無視
PARENT_ID	無視
名前	要素
TYPE	集約
SALESREP	集約
SALESTEAM	集約

これで、変数をマッピングする準備ができました。

変数のマッピング:

要素をディメンションにマッピングした後、集約変数をマッピングして変数をマッピングします。

手順

1. 要素をディメンションにマッピングします。
   a. 「**マップ**」タブをクリックしてから、「**ディメンション**」サブタブをクリックします。

      要素と識別した 1 つの変数が、グリッドに表示されます。
   b. 「ディメンション」列に「**DB**」と入力します。
   c. 「操作」ドロップダウン・メニューから「**作成**」を選択します。
   d. 「要素の型」ドロップダウン・メニューから「**数値**」を選択します。
2. 集約変数をマッピングします。
   a. 「**集約**」サブタブをクリックします。

各集約変数が DB ディメンションに関係していることは、 Xcelerator で正しく認識されます。

b. 各集約変数の「子の変数」を設定します。

定数の変数	子の変数
TYPE	SALESREP
SALESREP	名前
SALESTEAM	TYPE

プロセスの保存と実行:

プロセスを保存して実行するには、以下を実行します。

手順

1. 「**実行**」ボタン  をクリックします。

   Xcelerator は、プロセスを保存するよう指示します。
2. “create_DB_dimension”という名前でプロセスを保存します。

   数秒後に、プロセスが正しく実行されたことを示す確認画面が表示されます。
3. 「TurboIntegrator」ウィンドウを閉じます。
4. サーバー・エクスプローラーを開きます。
5. 作成された“**DB**”ディメンションをダブルクリックします。

   サブセット・エディターで“DB”ディメンションが開きます。
6. サブセット・エディターのメニュー・バーから「**編集**」、「**ソート**」、「**階層**」を選択して、ディメンションの要素と集約を表示します。

   “DB”ディメンションには、40 を超える要素が含まれています。階層レベルは 4 つあります。

キューブの作成とデータの処理

次の例では、 Xcelerator TurboIntegrator を使用して、キューブ、ディメンション、および要素を作成し、同時にデータを処理する方法を示します。

データ・ソースの定義

次の手順を実行して、データ・ソースを定義します。

手順

1. サーバー・エクスプローラーの左側のウィンドウで「**プロセス**」アイコンを右クリックし、「**新規プロセスの作成**」を選択します。

   「TurboIntegrator」ウィンドウが開きます。
2. 「TurboIntegrator」ウィンドウの「**データ・ソース**」タブをクリックします。

3. 「データ・ソース・タイプ」を「**テキスト**」に、「区切り記号の種類」を「**区切り付き**」に、さらに「区切り記号」を「**コンマ**」に設定します。

   「引用符」および「タイトル・レコード数」フィールドは無視します。
4. 「小数点記号」がピリオド (.) で、「桁区切り記号」がコンマ (,) になっていることを確認します。
5. 「データ・ソース名」フィールドの横にある「**参照**」ボタンをクリックし、“TI_data”ディレクトリー内のファイル“**newcube.csv**”を選択します。
6. 「**プレビュー**」をクリックして、データ・ソースの最初の 10 レコードを表示します。

   “newcube.csv”の各レコードには 20 個のフィールドがあります。表示グリッドをスクロールすると、すべてのフィールドを表示できます。

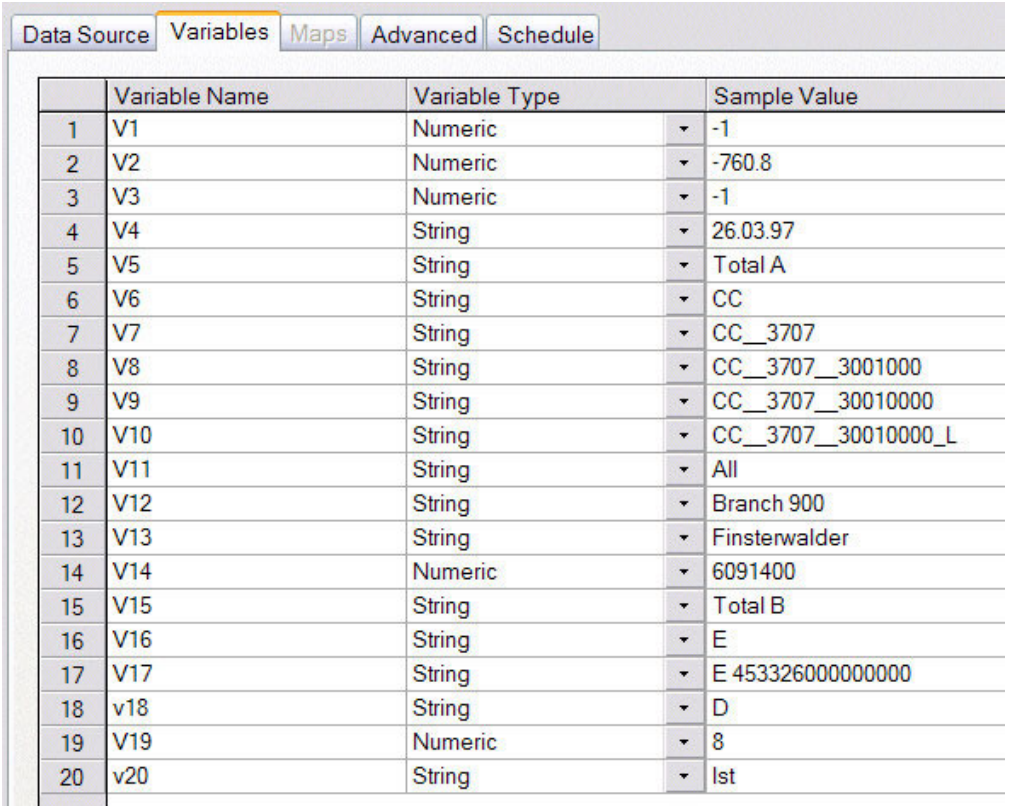
変数の識別

ソース・データを TurboIntegrator にロードしたら、ソースのフィールドごとに内容を識別する必要があります。

手順

1. 「**変数**」タブをクリックします。

   一部の変数には V*n* という形式で名前が付けられ、他の変数にはソース・ファイルの最初のレコードに対応する名前が付けられます。
2. 編集処理を簡単にするために、すべての変数の名前を V*n* という形式に変更します。最初の変数の名前を「V1」、2 番目の変数の名前を「V2」に変更してください (以下同様)。操作が終了すると、「変数」タブの表示は次のようになります。



	Variable Name	Variable Type	Sample Value
1	V1	Numeric	-1
2	V2	Numeric	-760.8
3	V3	Numeric	-1
4	V4	String	26.03.97
5	V5	String	Total A
6	V6	String	CC
7	V7	String	CC__3707
8	V8	String	CC__3707__3001000
9	V9	String	CC__3707__30010000
10	V10	String	CC__3707__30010000_L
11	V11	String	All
12	V12	String	Branch 900
13	V13	String	Finsterwalder
14	V14	Numeric	6091400
15	V15	String	Total B
16	V16	String	E
17	V17	String	E 453326000000000
18	v18	String	D
19	V19	Numeric	8
20	v20	String	Ist

3. 各変数について、それぞれの「変数の型」ドロップダウン・リストから型を選択します。

   変数 V1、V2、V3、V14、および V19 のタイプは「**数値**」です。他の変数のタイプはすべて「**文字列**」です。

4. 各変数について、それぞれの「コンテンツ」ドロップダウン・リストから内容タイプを選択します。次の表を参照して、各変数の内容タイプを指定してください。

変数名	**コンテンツ**	**変数名**	**コンテンツ**
V1	データ	V11	集約
V2	データ	V12	集約
V3	データ	V13	集約
V4	要素	V14	要素
V5	集約	V15	集約
V6	集約	V16	集約
V7	集約	V17	要素
V8	集約	V18	要素

変数名	コンテンツ	変数名	コンテンツ
V9	集約	V19	要素
V10	要素	V20	要素

変数のマッピング

これでデータ、要素、および集約の変数は識別されました。ここからは、変数をマッピングして、新しいキューブを作成する手順を指示する必要があります。

キューブのマッピング:

キューブのマッピング指示を指定するには：

手順

1. 「**マップ**」タブをクリックします。
2. 「**キューブ**」サブタブをクリックします。
3. 「キューブの操作」で「**作成**」を選択します。
4. 「キューブ名」フィールドに「**NewCube**」と入力します。
5. 「データ操作」で「**値の保存**」を選択します。
6. 「キューブのログ記録を有効化」オプションはオンにしないでください。

   キューブのログ記録を有効にすると、 Xcelerator は処理中のキューブ・データの変更を記録します。新しいキューブを作成するときに、変更内容をログに記録する必要はありません。

ディメンションへの要素変数のマッピング:

これで、「要素」型を持つと識別したすべての変数を、それぞれのディメンションにマッピングできます。

手順

1. 「**ディメンション**」サブタブをクリックします。
2. 次の表に従って、要素変数ごとに「ディメンション」、「操作」、および「要素の型」を指定します。

要素の変数	ディメンション	操作	要素の型
V4	date	作成	数値
V10	item	作成	数値
V14	customer	作成	数値
V17	job	作成	数値
V18	地域	作成	数値

要素の変数	ディメンション	操作	要素の型
V19	エージェント	作成	数値
V20	book	作成	数値
Data Variables	measure	作成	数値

各変数の「**キューブ内の順序**」値は、デフォルト値をそのまま使用できます。

データ変数のマッピング:

ここでは、型がデータであると識別した変数をそれぞれの要素にマッピングする必要があります。

手順

1. 「**データ**」サブタブをクリックします。
2. データ変数“V1”について、「**weight**」を変数のマッピング先の要素として入力します。
3. “V2”では、「**conversion**」と入力します。
4. “V3”では、「**pieces**」と入力します。
5. 「要素の型」列は、3 つのすべての要素に対して「**数値**」を選択します。

集約変数のマッピング:

ここでは、コンテンツが集約であると識別したすべての変数について、集約パスをマッピングする必要があります。

手順

1. 「**集約**」サブタブをクリックします。
2. 次の表に従って、集約変数ごとに「ディメンション」および「子の変数」を指定します。

集約変数	ディメンション	子の変数
V5	item	V6
V6	item	V7
V7	item	V8
V8	item	V9
V9	item	V10
V11	customer	V12
V12	customer	V13
V13	customer	V14
V15	job	V16
V16	job	V17

3. 「重み付け」と「コンポーネントの順序」は、すべての集約変数についてデフォルト値をそのまま使用できます。

これで、新しいディメンションを作成し、そのディメンションに要素と集約を挿入し、新しいキューブを作成して、そのキューブにデータを移入するためのマッピングが完了しました。

プロセスの保存と実行:

プロセスを保存して実行するには、以下を実行します。

手順

1. 「**実行**」ボタン  をクリックします。

   Xcelerator は、プロセスを保存するよう指示します。
2. プロセスを“create_newcube”として保存します。

   数秒後に、プロセスが正しく実行されたことを示す確認画面が表示されます。
3. サーバー・エクスプローラーを開いて、キューブ“NewCube”が作成されてデータが移入されていること、および必要なディメンションがすべて作成されていることを確認します。

   新しいキューブ (データはまばらに移入されています) を参照して、新しく作成されたディメンションをチェックしてください。

詳細スクリプトの処理

TurboIntegrator の「**詳細**」タブを使用して、実行時にプロセスに渡すことが可能なパラメーターを作成したり、プロセスのプロシージャーを編集したりすることで、TurboIntegrator の機能を強化できます。プロシージャーは、TurboIntegrator 関数と Xcelerator 規則関数の両方を組み込むスクリプトを作成して編集します。

プロローグ、メタデータ、データ、エピローグの各プロシージャーの編集

TurboIntegrator の機能は、プロセスの動作を定義するプロシージャーを編集して拡張できます。プロシージャーは、 Xcelerator のデータまたはメタデータを操作する文をグループにまとめたものです。

プロセス内には、順番に実行される 4 つのプロシージャーがあります。各プロシージャーには、「TurboIntegrator」ウィンドウのいずれかの画面でユーザーが指定したオプションを基に作成した生成文が含まれます。このプロシージャーは編集できます (ユーザー独自の文を追加して、TurboIntegrator 関数および規則関数を組み込むことができます)。

プロセスには、次のプロシージャーが含まれています。

タブ	説明
プロローグ	データ・ソースを処理する前に実行される一連の操作
メタデータ	処理中にキューブ、ディメンション、などのメタデータ構造体を更新または作成する一連の操作

タブ	説明
データ	データ・ソース内の各レコードに対して実行される一連のデータ操作
エピローグ	データ・ソースを処理した後で実行される一連の操作

プロシージャーを編集するときは、各プロシージャーはプロセス中の特定の時点で特定の操作を実行するように設計されているということを忘れないでください。したがって、作成する操作や文は、そのプロシージャーに適したものにしてください。

例えば、処理するデータを ASCII ファイルにエクスポートするには、データ・プロシージャーに ASCIIOutput 関数を追加することになります。ASCIIOutput はデータを操作する関数なので、処理時に実行する必要があります。したがって、データ・プロシージャーが、この関数を配置するのに適したプロシージャーになります。

プロシージャーの編集

プロシージャーを編集するには：

手順

1. 「TurboIntegrator」ウィンドウの「**詳細**」タブをクリックします。
2. 編集するプロシージャーのサブタブをクリックします。
3. 文をテキスト・ボックスに入力します。次の行の*前*か、

   #****GENERATED STATEMENTS START****

   次の行の後ろに入力してください。

   #****GENERATED STATEMENTS FINISH****

   この 2 つの行の間に生成されている文は編集しないでください。

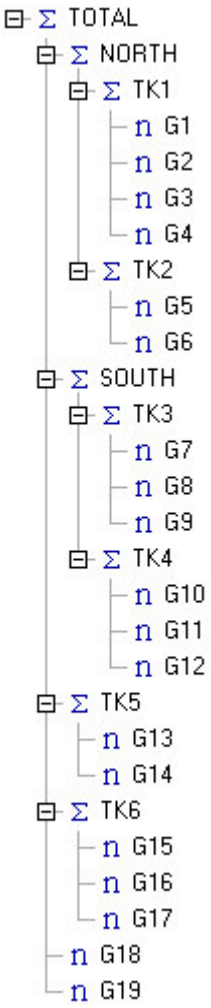
階層のバランスが取れていないディメンションの作成

この演習では、次の入力ファイルを使用して、階層のバランスが取れていないディメンションを作成します。

```
TOTAL,NORTH,TK1,G1
TOTAL,NORTH,TK1,G2
TOTAL,NORTH,TK1,G3
TOTAL,NORTH,TK1,G4
TOTAL,NORTH,TK2,G5
TOTAL,NORTH,TK2,G6
TOTAL,SOUTH,TK3,G7
TOTAL,SOUTH,TK3,G8
TOTAL,SOUTH,TK3,G9
TOTAL,SOUTH,TK4,G10
TOTAL,SOUTH,TK4,G11
TOTAL,SOUTH,TK4,G12
TOTAL,TK5,G13
TOTAL,TK5,G14
TOTAL,TK6,G15
```

```
TOTAL,TK6,G16
TOTAL,TK6,G17
TOTAL,G18
TOTAL,G19
```

最終的結果は、次のようになります。



ディメンションの作成を開始するには：

手順

1. サーバー・エクスプローラーの左側のウィンドウで「**プロセス**」アイコンを右クリックし、「**新規プロセスの作成**」を選択します。

   「TurboIntegrator」ウィンドウが開きます。
2. 「データ・ソース・タイプ」として「**テキスト**」を選択します。
3. 「データ・ソース名」フィールドの横の「**参照**」をクリックし、“TI_data”ディレクトリー内の**“unbalanced.csv”**を選択します。
4. 「データ・ソース」タブにある他のオプションは、すべてデフォルト設定値のままにしておきます。
5. 「**プレビュー**」をクリックして、データ・ソースの最初の 10 レコードを表示します。

変数の識別

ソース・データを TurboIntegrator にロードしたら、ソースのフィールドごとに内容を識別する必要があります。

手順

1. 「**変数**」タブをクリックします。
2. “Total”、“North”、“TK1”の各変数について、「コンテンツ」列で「**集約**」を選択します。
3. 変数“G1”では、「**要素**」を選択します。

変数のマッピング

これで要素と集約の変数が識別されました。次に、変数をディメンションにマッピングし、集約パスを定義する必要があります。

手順

1. 「**マップ**」タブをクリックします。
2. 「**ディメンション**」サブタブをクリックします。
3. 要素変数“G1”について、「ディメンション」として「**unbalanced**」、「操作」として「**作成**」、「要素の型」として「**数値**」を入力します。
4. 「**集約**」サブタブをクリックします。
5. 3 つの変数について、「**ディメンション**」列のドロップダウン・リストから「**unbalanced**」を選択します。
6. 集約変数“Total”について、「子の変数」として「**North**」を選択します。
7. 集約変数“North”について、「子の変数」として「**TK1**」を選択します。
8. 集約変数“TK1”について、「子の変数」として「**G1**」を選択します。

生成された文のコピー

「TurboIntegrator」ウィンドウのオプションを変更すると、 Xcelerator によって動的に文が作成されます。

「詳細」タブの「プロローグ」サブタブと「メタデータ」サブタブに生成された文を編集して、バランスの取れていないディメンション階層に対応します。作業を多少とも楽に進めるため、生成された文をコピーして貼り付けることにします (そうすることにより、「TurboIntegrator」ウィンドウでオプションを変更した後の文を使用できます)。

手順

1. 「**詳細**」タブをクリックしてから、「**プロローグ**」サブタブをクリックします。
2. コメント行の間に挟まれている DimensionDestroy および DimensionCreate 関数をコピーして、

   ```
   #****GENERATED STATEMENTS START****
   #****GENERATED STATEMENTS FINISH****
   ```

   コメント行の下に貼り付けます。

   ```
   #****GENERATED STATEMENTS START****
   DIMENSIONDESTROY('unbalanced');
   ```

```
DIMENSIONCREATE('unbalanced');
DIMENSIONSORTORDER('unbalanced','ByInput','ASCENDING','ByInput','ASCENDING');
****GENERATED STATEMENTS FINISH****
DIMENSIONDESTROY('unbalanced');
DIMENSIONCREATE('unbalanced');
```

3. 「**メタデータ**」サブタブをクリックします。

   次の 2 つの関数があります。

   DimensionElementInsert 関数は、単純な要素 (リーフ要素) をディメンションに追加します。この関数を使用すると、数値と文字列の両方の要素を追加できます。

   DimensionElementComponentAdd 関数は、コンポーネント (子) を集約要素に追加します。

4. 生成されている文をすべてコピーし、最後のコメント行の下に貼り付けます。

```
#****GENERATED STATEMENTS START****
DIMENSIONELEMENTINSERT('unbalanced',",G1,'n');
DIMENSIONELEMENTINSERT('unbalanced',",TOTAL,'c');
DIMENSIONELEMENTINSERT('unbalanced',",NORTH,'c');
DIMENSIONELEMENTINSERT('unbalanced',",TK1,'c');
DIMENSIONELEMENTCOMPONENTADD('unbalanced',TOTAL,NORTH,1.000000);
DIMENSIONELEMENTCOMPONENTADD('unbalanced',NORTH,TK1,1.000000);
DIMENSIONELEMENTCOMPONENTADD('unbalanced',TK1,G1,1.000000);
****GENERATED STATEMENTS FINISH****
DIMENSIONELEMENTINSERT('unbalanced',",G1,'n');
DIMENSIONELEMENTINSERT('unbalanced',",TOTAL,'c');
DIMENSIONELEMENTINSERT('unbalanced',",NORTH,'c');
DIMENSIONELEMENTINSERT('unbalanced',",TK1,'c');
DIMENSIONELEMENTCOMPONENTADD('unbalanced',TOTAL,NORTH,1.000000);
DIMENSIONELEMENTCOMPONENTADD('unbalanced',NORTH,TK1,1.000000);
DIMENSIONELEMENTCOMPONENTADD('unbalanced',TK1,G1,1.000000);
```

生成された文の完全な削除

生成された文を完全に削除するには：

手順

1. 「**変数**」タブをクリックし、「コンテンツ」列の選択内容を「**その他**」に変更します。

   「その他」として識別した変数は、詳細スクリプトで使用することができます。変数が「無視」として識別されている場合、その変数を TurboIntegrator では処理されないため、詳細スクリプトからも参照できません。

2. 文が削除されていることを確認するには、「**詳細**」タブをクリックしてから、「**プロローグ**」および「**メタデータ**」サブタブをクリックします。

   文は次のようになります。

```
Prolog>
#****GENERATED STATEMENTS START****
#****GENERATED STATEMENTS FINISH****
DIMENSIONDESTROY('unbalanced');
DIMENSIONCREATE('unbalanced');
Metadata>
#****GENERATED STATEMENTS START****
#****GENERATED STATEMENTS FINISH****
DIMENSIONELEMENTINSERT('unbalanced','',G1,'n');
DIMENSIONELEMENTINSERT('unbalanced','',TOTAL,'c');
DIMENSIONELEMENTINSERT('unbalanced','',NORTH,'c');
DIMENSIONELEMENTINSERT('unbalanced','',TK1,'c');
DIMENSIONELEMENTCOMPONENTADD('unbalanced',TOTAL,NORTH,1.000000);
DIMENSIONELEMENTCOMPONENTADD('unbalanced',NORTH,TK1,1.000000);
DIMENSIONELEMENTCOMPONENTADD('unbalanced',TK1,G1,1.000000);
```

TurboIntegrator 文の編集

「メタデータ」サブタブに今あるスクリプトの内容を確認します。この文は、次のようになっています。

```
DIMENSIONELEMENTINSERT('unbalanced','',G1,'n');
DIMENSIONELEMENTINSERT('unbalanced','',TOTAL,'c');
DIMENSIONELEMENTINSERT('unbalanced','',NORTH,'c');
DIMENSIONELEMENTINSERT('unbalanced','',TK1,'c');
DIMENSIONELEMENTCOMPONENTADD('unbalanced',TOTAL,NORTH,1.000000);
DIMENSIONELEMENTCOMPONENTADD('unbalanced',NORTH,TK1,1.000000);
DIMENSIONELEMENTCOMPONENTADD('unbalanced',TK1,G1,1.000000);
```

このスクリプトは、“unbalanced.csv”の最初のレコードに基づいて生成されており、フィールドが 4 つあるレコードに対し有効です。このスクリプトは、ソースの各フィールドからディメンション要素を作成し、階層を作成します。ただし、含まれるフィールドが 4 つより少ないレコードには有効でありません。

ソース・ファイルの“unbalanced.csv”には長さが異なるレコードが含まれるため、ソース内の各レコードを評価するようにスクリプトを変更する必要があります。このスクリプトでは、集約のレベルを正しく判断し、集約の各レベルに対して適切な集約パスを指定する必要があります。これは、IF 関数を含むようにスクリプトを編集して実現できます。IF 関数を使用するとで、定義した条件に基づいて別の TurboIntegrator 文を実行できます。

手順

1. 「**詳細**」タブをクリックしてから、「***メタデータ***」サブタブをクリックします。
2. 次の行を挿入します。

   ```
   IF (G1@<>'');
   ```

挿入する位置は、最初の DIMENSIONELEMENTINSERT 文の手前です。この IF 文は、文字列変数“G1”が空でなければ、この IF 文の後に続く文が実行されることを示します。“V4”が空の場合は、この次の条件文に処理がジャンプします。

これで、「メタデータ」サブタブの表示は、次のようになります。

```
#****GENERATED STATEMENTS START****
#****GENERATED STATEMENTS FINISH****
IF (G1@<>'');
DIMENSIONELEMENTINSERT('unbalanced','',G1,'n');
DIMENSIONELEMENTINSERT('unbalanced','',TOTAL,'c');
DIMENSIONELEMENTINSERT('unbalanced','',NORTH,'c');
DIMENSIONELEMENTINSERT('unbalanced','',TK1,'c');
DIMENSIONELEMENTCOMPONENTADD('unbalanced',TOTAL,NORTH,1.000000);
DIMENSIONELEMENTCOMPONENTADD('unbalanced',NORTH,TK1,1.000000);
DIMENSIONELEMENTCOMPONENTADD('unbalanced',TK1,G1,1.000000);
```

IF (G1@<>'') が真の場合は、3 つの集約要素 (“Total”、“North”、“TK1”) と 1 つの数値要素 (“G1”) がバランスが取れていないディメンションに挿入されます。また、“Total”が“North”の親で、“North”が“TK1”の親で、“TK1”が“G1”の親である 4 レベルの階層も作成されます。

3. 次の行を挿入します。

```
ELSEIF (TK1@<>'');
```

挿入する位置は、最後の DIMENSIONELEMENTCOMPONENTADD 文の後ろです。

この条件付き ELSEIF 文は、文字列変数“V3”が空でなければ、この ELSEIF 文の後に続く文が実行されることを示します。V3 が空である 場合は、この次の条件文に処理がジャンプします。

4. ここで、ELSEIF (TK1@<>'') が真の場合に実行する文を挿入する必要があります。

ELSEIF (TK1@<>'') が真の場合は、ソース・レコードに 3 つのフィールドが含まれています。したがって、各フィールドからディメンション要素を作成し、3 レベルの階層を作成する文が必要です。

5. 次の文を、ELSEIF (TK1@<>''); のすぐ後ろに挿入します。

```
DIMENSIONELEMENTINSERT('unbalanced','',TOTAL,'c');
DIMENSIONELEMENTINSERT('unbalanced','',NORTH,'c');
DIMENSIONELEMENTINSERT('unbalanced','',TK1,'n');
DIMENSIONELEMENTCOMPONENTADD('unbalanced',TOTAL,NORTH,1.000000);
DIMENSIONELEMENTCOMPONENTADD('unbalanced',NORTH,TK1,1.000000);
```

IF (TK1@<>'') が真の場合は、2 つの集約要素 (“Total”、“North”) と 1 つの数値要素 (“TK1”) がバランスが取れていないディメンションに挿入されます。また、“TOTAL”が“NORTH”の親で、“NORTH”が“TK1”の親である 3 レベルの階層も作成されます。

6. 次の行を挿入します。

   ```
   ELSE;
   ```

   挿入する位置は、最後の DIMENSIONELEMENTCOMPONENTADD 文の後ろです。

7. ここで、処理が ELSE 文に到達したときに実行する文を挿入する必要があります。(そうなるのは、IF (G1@<>") と ELSEIF (TK1@<>") の両方が偽の場合です。)

   処理が ELSE 文に到達した場合は、ソース・レコードに 2 つのフィールドが含まれています。挿入する文によって、各フィールドからディメンション要素を作成し、2 レベルの階層を作成する必要があります。

8. 次の文を、ELSE; のすぐ後ろに挿入します。

   ```
   DIMENSIONELEMENTINSERT('unbalanced','',TOTAL,'c');
   DIMENSIONELEMENTINSERT('unbalanced','',NORTH,'n');
   DIMENSIONELEMENTCOMPONENTADD('unbalanced',TOTAL,NORTH,1.000000);
   ```

   この文は、集約要素“TOTAL”と数値要素“NORTH”をバランスが取れていないディメンションに挿入し、“TOTAL”が“NORTH”の親である階層を作成するように TurboIntegrator に指示します。

9. 次の行を挿入します。

   ```
   ENDIF;
   ```

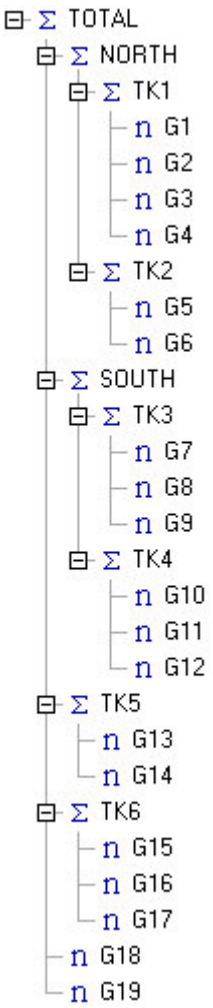
   挿入する位置は、最後の DIMENSIONELEMENTCOMPONENTADD 文の後ろです。ENDIF は、IF 文の終端を示します。

   編集が終了して完成した「メタデータ」サブタブは、次のようになります。

   ```
   #****GENERATED STATEMENTS START****
   #****GENERATED STATEMENTS FINISH****
   IF (G1@<>'');
   DIMENSIONELEMENTINSERT('unbalanced','',G1,'n');
   DIMENSIONELEMENTINSERT('unbalanced','',TOTAL,'c');
   DIMENSIONELEMENTINSERT('unbalanced','',NORTH,'c');
   DIMENSIONELEMENTINSERT('unbalanced','',TK1,'c');
   DIMENSIONELEMENTCOMPONENTADD('unbalanced',TOTAL,NORTH,1.000000);
   DIMENSIONELEMENTCOMPONENTADD('unbalanced',NORTH,TK1,1.000000);
   DIMENSIONELEMENTCOMPONENTADD('unbalanced',TK1,G1,1.000000);
   ELSEIF (TK1@<>'');
   DIMENSIONELEMENTINSERT('unbalanced','',TOTAL,'c');
   DIMENSIONELEMENTINSERT('unbalanced','',NORTH,'c');
   DIMENSIONELEMENTINSERT('unbalanced','',TK1,'n');
   DIMENSIONELEMENTCOMPONENTADD('unbalanced',TOTAL,NORTH,1.000000);
   DIMENSIONELEMENTCOMPONENTADD('unbalanced',NORTH,TK1,1.000000);
   ELSE;
   DIMENSIONELEMENTINSERT('unbalanced','',TOTAL,'c');
   ```

```
DIMENSIONELEMENTINSERT('unbalanced','',NORTH,'n');
DIMENSIONELEMENTCOMPONENTADD('unbalanced',TOTAL,NORTH,1.000000);
ENDIF;
```

10. 「**ファイル**」 → 「**保存**」を選択し、プロセスに“create_unbalanced_dim”という名前を付けます。
11. 「**ファイル**」 → 「**実行**」を選択して、プロセスを実行します。
12. ディメンションが正しく構築されたことを確認するには、ディメンション・エディターでバランスが取れていないディメンションを開きます。ディメンションは、次の図のように表示されます。



サブセットの作成

この演習では、ディメンション・プロセスによって作成されるディメンション“newdim”のサブセットを作成します。

手順

1. 「TurboIntegrator」ウィンドウでプロセス「**subsets**」を開きます。

“TI_data”ディレクトリーの“region.csv”を指すようにデータ・ソースの変更が必要な場合があります。データ・ソースを変更すると、プロセス変数を扱う方法を指定するプロンプトが表示されます。「**すべての変数を維持する**」を選択します。

この例では、 Xcelerator TurboIntegrator 関数の SubsetCreate() と SubsetElementInsert() を使用して、ディメンション・サブセットを作成し、要素を設定します。

ソース・ファイルのプレビューは、次のようになっています。

V0	V1	V2	V3	V4
Sweden	Scandanavia	Europe	International	Europe
Norway	Scandanavia	Europe	International	Europe
Denmark	Scandanavia	Europe	International	Europe
France	Europe	International	世界共通	Europe
Germany	Europe	International	世界共通	Europe
UK	Europe	International	世界共通	Europe
Ireland	Europe	International	世界共通	Europe
Holland	Europe	International	世界共通	Europe
Spain	Europe	International	世界共通	Europe
Italy	Europe	International	世界共通	Europe

次に、サブセットを処理するスクリプトを示します。

```
Prolog>
#****GENERATED STATEMENTS START****
#****GENERATED STATEMENTS FINISH****
SubsetCreate('NewDim','Europe');
SubsetCreate('NewDim','US');
SubsetCreate('NewDim','ROW');
Metadata>
#****GENERATED STATEMENTS START****
#****GENERATED STATEMENTS FINISH****
SubsetElementInsert('NewDim',V4,V0,0);
```

2. プロセスを実行します。
3. サーバー・エクスプローラーで、“newdim”ディメンションを展開して、新しく作成されたアブセットを確認します。

属性の作成

文字列要素属性の値は、AttrPutS 関数で割り当てます。文字列「Europe」を、「NewDim」ディメンションの地域「Sweden」の「Continent」属性に割り当てる場合は、AttrPutS 関数を次のように記述します。

```
AttrPutS('Europe','NewDim','Sweden','Continent');
```

手順

1. TurboIntegrator で「**Attributes**」プロセスを開きます。

   “TI_data”ディレクトリーの“region.csv”を指すようにデータ・ソースの変更が必要な場合があります。データ・ソースを変更すると、プロセス変数を扱う方法を指定するプロンプトが表示されます。「**すべての変数を維持する**」を選択します。
2. 「**変数**」タブをクリックします。

   “V4”と“V5”が属性として識別されていることを確認してください。
3. “V5”の「**数式**」セルをクリックします。

   V5=V0|V4; と表示されます。

   この式は、“V4”変数と“V5”変数の値を連結します。
4. 「**マップ**」タブをクリックし、「**属性**」サブタブをクリックします。

   変数“V4”の属性の型は「テキスト」と定義され、“V5”の型は「別名」と定義されています。
5. 「**詳細**」タブをクリックし、「**データ**」サブタブをクリックして、生成された文と 2 つの追加文を表示します。

   ```
   #****GENERATED STATEMENTS START****
   V5=v0|v4;
   AttrPutS(V4,'newdim',V0,'continent');
   AttrPutS(V5,'newdim',V0,'cont');
    #****GENERATED STATEMENTS FINISH****
   AttrPutS(V4,'newdim',V1,'continent');
   AttrPutS(V4,'newdim',V2,'continent');
   ```

   「変数」タブで“V1”と“V2”がコンテンツと宣言されていないため、上記の 2 つの文は手動で追加されています。ただし、これらの変数には、テキスト属性“Continent”を割り当てる必要があります。
6. “Attributes”プロセスを保存して実行します。

属性の表示

属性値を割り当てたら、次の手順でその割り当て内容を表示できます。

手順

1. サーバー・エクスプローラーで「**newdim**」ディメンションをダブルクリックして、サブセット・エディターを開きます。
2. 「**サブセット All**」  をクリックします。

3. メニューから「**編集**」 → 「**フィルター基準**」 → 「**属性**」を選択して、「属性によるフィルター」ダイアログ・ボックスを表示します。
4. 「属性によるフィルター」ダイアログ・ボックスで、ドロップダウン・リストから属性値を選択して、特定の大陸のすべての地域をサブセット・エディターに表示します。

付録 B. TurboIntegrator の予約語

この付録は、IBM Cognos Xcelerator TurboIntegrator の予約語のリストです。作成する TurboIntegrator スクリプトでエラーが発生しないように、変数を作成する場合は、次の表に示す語と一致する名前を使用しないでください。

TurboIntegrator の予約語は、次の 4 つに分類されます。

- 規則関数の名前
- プロセス関数の名前
- 暗黙的な変数の名前
- TurboIntegrator のキーワード

規則関数の名前

次に、 Xcelerator 規則関数の予約語を示します。

- ABS
- ACOS
- ASIN
- ATAN
- ATTRN
- ATTRS
- AVG
- BANNR
- BDATE
- BDAYN
- CAPIT
- CENTR
- CHAR
- CNT
- CODE
- COL
- Consolidate Children
- COS
- DATE
- DATES
- DATFM
- DAY
- DAYNO
- DBG16

- DBGEN
- DELET
- DFRST
- DIMIX
- DIMNM
- DIMSIZ
- DISPLY
- DNEXT
- DNLEV
- DTYPE
- DYS
- ELCOMP
- ELCOMPN
- ELISANC
- ELISCOMP
- ELISPAR
- ELLEV
- ELPAR
- ELPARN
- ELWEIGHT
- EXP
- FILL
- FV
- HEX
- IF
- INSRT
- INT
- IRR
- ISLEAF
- ISUND
- LIN
- LN
- LOG
- LONG
- LOOK
- LOWER
- MAX
- MEM
- MIN
- MOD

- MONTH
- MOS
- NCELL
- NOW
- NPV
- PAYMT
- PV
- RAND
- RIGHT
- ROUND
- ROUNDP
- SCAN
- SCELL
- SIGN
- SIN
- SLEEP
- SQRT
- STDDV
- STR
- SUBSIZ
- SUBST
- SUM
- TABDIM
- TAN
- TIME
- TIMST
- TIMVL
- TODAY
- TRIM
- UNDEF
- UPPER
- VAR
- WHOAMI
- WIDTH
- YEAR
- YRS

プロセス関数の名前

次に、TurboIntegrator プロセス関数名を示します。

- AddClient
- AddGroup
- AllowExternalRequests
- ASCIIDelete
- ASCIIOutput
- AssignClientPassword
- AssignClientToGroup
- AttrDelete
- AttrInsert
- AttrPutN
- AttrPutS
- AttrToAlias
- BatchUpdateFinish
- BatchUpdateStart
- CellGetN
- CellGetS
- CellIsUpdateable
- CellPutN
- CellPutProportionalSpread
- CellPutS
- ChoreQuit
- CubeCreate
- CubeDestroy
- CubeExists
- CubeGetLogChanges
- CubeLockOverride
- CubeProcessFeeders
- CubeSetConnParams
- CubeSetIsVirtual
- CubeSetLogChanges
- CubeSetSAPVariablesClause
- CubeSetSlicerMembers
- CubeUnload
- DeleteClient
- DeleteGroup
- DimensionCreate
- DimensionDeleteAllElements

- DimensionDestroy
- DimensionEditingAliasSet
- DimensionElementComponentAdd
- DimensionElementComponentDelete
- DimensionElementDelete
- DimensionElementInsert
- DimensionElementInsertByAlias
- DimensionElementPrincipalName
- DimensionExists
- DimensionSortOrder
- ElementSecurityGet
- ElementSecurityPut
- EncodePassword
- ExecuteCommand
- ExecuteProcess
- Expand
- FileExists
- GetProcessErrorFileDirectory
- GetProcessErrorFilename
- IsNull
- ItemReject
- ItemSkip
- LockOff
- LockOn
- NumberToString
- NumberToStringEx
- NumericGlobalVariable
- NumericSessionVariable
- ODBCClose
- ODBCOpen
- ODBCOutput
- ProcessBreak
- ProcessError
- ProcessExitByBreak
- ProcessExitByChoreQuit
- ProcessExitByQuit
- ProcessExitMinorError
- ProcessExitNormal
- ProcessExitOnInit
- ProcessExitSeriousError

- ProcessExitWithMessage
- ProcessQuit
- PublishView
- RemoveClientFromGroup
- ReturnSQLTableHandle
- ReturnViewHandle
- RuleLoadFromFile
- SaveDataAll
- SecurityRefresh
- ServerShutDown
- SetChoreVerboseMessages
- StringGlobalVariable
- StringSessionVariable
- StringToNumber
- StringToNumberEx
- SubsetAliasSet
- SubsetCreate
- SubsetCreateByMDX
- SubsetDeleteAllElements
- SubsetDestroy
- SubsetElementDelete
- SubsetElementInsert
- SubsetExists
- SubsetFormatStyleSet
- SubsetGetElementName
- SubsetGetSize
- SubsetIsAllSet
- SwapAliasWithPrincipalName
- ViewColumnDimensionSet
- ViewColumnSuppressZeroesSet
- ViewConstruct
- ViewCreate
- ViewDestroy
- ViewExists
- ViewExtractSkipRuleValuesSet
- ViewExtractSkipRuleValuesSet
- ViewExtractSkipZeroesSet
- ViewRowDimensionSet
- ViewRowSuppressZeroesSet
- ViewSetSkipCalcs

- ViewSetSkipRuleValues
- ViewSetSkipZeroes
- ViewSubsetAssign
- ViewSuppressZeroesSet
- ViewTitleDimensionSet
- ViewTitleElementSet
- ViewZeroOut
- WildcardFileSearch

暗黙的な変数の名前

次に、TurboIntegrator の暗黙の変数名を示します。

- DatasourceASCIIDecimalSeparator
- DatasourceASCIIDelimiter
- DatasourceASCIIHeaderRecords
- DatasourceASCIIQuoteCharacter
- DatasourceASCIIThousandSeparator
- DatasourceCubeview
- DatasourceDimensionSubset
- DatasourceNameForClient
- DatasourceNameForServer
- DatasourceODBOCatalog
- DatasourceODBOConnectionString
- DatasourceODBOCubeName
- DatasourceODBOHierarchyName
- DatasourceODBOLocation
- DatasourceODBOProvider
- DatasourceODBOSAPClientId
- DatasourceODBOSAPClientLanguage
- DatasourcePassword
- DatasourceQuery
- DatasourceType
- DatasourceUseCallerProcessConnection
- DatasourceUsername
- MinorErrorLogMax
- NValue
- OnMinorErrorDoItemSkip
- SValue
- Value_Is_String

TurboIntegrator のキーワード

次に、予約済みの TurboIntegrator キーワードを示します。

- break
- else
- elseif
- end
- endif
- if
- while

特記事項

本書は IBM が世界各国で提供する製品およびサービスについて作成したものです。

本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。本書には、お客様が購入されたプログラムまたはライセンス資格に含まれない製品、サービス、または機能に関する説明が含まれる場合があります。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒242-8502
神奈川県大和市下鶴間1623番14号
日本アイ・ビー・エム株式会社
法務・知的財産
知的財産権ライセンス渉外

以下の保証は、国または地域の法律に沿わない場合は、適用されません。 IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態で提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。 IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

IBM Software Group
Attention: Licensing
3755 Riverside Dr
Ottawa, ON K1V 1B7
Canada

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性がありますが、その測定値が、一般に利用可能なシステムのものと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確証できません。 IBM 以外の製品の性能に関する質問は、それらの製品の供給者にお願いします。

IBM の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があり、単に目標を示しているものです。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

この情報をソフトコピーでご覧になっている場合は、写真やカラーの図表は表示されない場合があります。

商標

IBM、IBM ロゴ、ibm.com、TM1、Express、および Cognos は、世界の多くの国で登録された International Business Machines Corp. の商標です。他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。現時点での IBM の商標リストについては、http://www.ibm.com/legal/copytrade.shtml の「Copyright and trademark information」をご覧ください。

他の会社名、製品名およびサービス名等はそれぞれ各社の商標です。

- Microsoft、Windows、Windows NT および Windows ロゴは、Microsoft Corporation の米国およびその他の国における商標です。
- Linux は、Linus Torvalds の米国およびその他の国における商標です。
- UNIX は The Open Group の米国およびその他の国における登録商標です。

索引

日本語, 数字, 英字, 特殊文字の順に配列されています。なお, 濁音と半濁音は清音と同等に扱われています。

[ア行]

[カ行]

[サ行]

[タ行]

[ナ行]

[ハ行]

［マ行］

［ヤ行］

［ラ行］

A

C

M

N

O

S

T

U