

IBM Cognos TM1
バージョン 10.2.2

TurboIntegrator ガイド

IBM

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、111 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。

このドキュメントは IBM Cognos TM1 バージョン 10.2.2 に適用され、それ以降のリリースにも適用される場合があります。

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 TM1
Version 10.2.2
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

Licensed Materials - Property of IBM

© Copyright IBM Corporation 2007, 2014.

目次

はじめに	vii
第 1 章 TurboIntegrator の基礎	1
TurboIntegrator で使用可能なデータ・ソース	1
TurboIntegrator での文字列長の制限	1
インポート処理のオプション	2
TurboIntegrator 関数	2
プロセスと日課	2
TurboIntegrator プロセス内での操作の順序	3
TurboIntegrator プロセスに関する注意事項	3
同じ TM1 サーバーに同時接続	4
TurboIntegrator 関数での別名	5
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	5
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	5
TurboIntegrator 関数をサンドボックスで使用する	6
第 2 章 テキスト・ファイルのインポート	7
テキスト・ファイルからのディメンションの作成	7
データ・ソースの定義	7
データ・ソースの変数の識別	9
変数のマッピング	12
TurboIntegrator プロセスの保存と実行	14
テキスト・ファイルからのキューブの作成	14
キューブのデータ・ソースの定義	15
キューブ変数の定義	16
キューブ変数のマッピング	16
ディメンションへのキューブ要素変数のマッピング	16
キューブのデータ変数のマッピング	17
集約変数のマッピング	17
キューブ・プロセスの保存と実行	17
第 3 章 ODBC ソースからのインポート	19
Unicode と DNS	19
ODBC データ・ソースの定義	19
MDX 文からの TurboIntegrator プロセスの生成	20
MDX TurboIntegrator プロセスの構築	20
第 4 章 TM1 のビューまたはサブセットからのインポート	23
データ・ソースとして TM1 キューブ・ビューを使用	23
キューブ・プロセスの作成	23
データ・ソースとして TM1 サブセットを使用	24
データ・ソースとしてディメンション・サブセットを定義	24
ディメンション変数の定義	24
ディメンション変数のマッピング	25
ディメンションの保存と実行	26
第 5 章 MSAS からのインポート	27
OLE DB for OLAP データ・ソース	27
ODBO プロバイダー名	27
ODBO の場所	27

ODBO データ・ソース	27
ODBC カタログ	27
接続文字列: MSAS と TMI	27
OLE DB for OLAP データ・ソースへの接続 (CAM 認証を使用する場合)	28
MAS キューブのインポート	29
TurboIntegrator による Analysis Services への接続	29
「ODBC キューブのロード」タブによるキューブの指定	30
「キューブ・ディメンション」タブの使用	31
MAS プロセスの保存と実行	32
MAS ディメンションのインポート	32
MAS 接続パラメーターの定義	33
「ODBO ディメンションのロード」タブの使用	34
ディメンション MAS プロセスの保存と実行	35
TMI メッセージ・ログ	35

第 6 章 IBM Cognos TM1 Package Connector を使用してデータをインポートする . . . 37

Cognos BI サーバーへの接続を確立する	37
発行されたパッケージに接続する - 「パッケージおよびディメンション」オプション	39
1 つのディメンションのインポート	43
発行されたパッケージに接続する - 「カスタム・クエリー」オプション	44
TurboIntegrator プロセスの保存と実行	46
結果の確認	46
SAP BW データを使用する際のガイドライン	46
Framework Manager でパッケージを使用して SAP BW データを使用する	47
SAP BW クエリーにセグメント化プロンプトを指定する	53
分割プロンプトを使用して並行クエリーを実行する	54
数値データ・クエリー実行時の無効なキー・エラー	54

第 7 章 詳細プロシージャの編集 . . . 55

一括ロード・モードの使用	55
一括ロード・モード使用時の注意	55
一括ロード・モードの TurboIntegrator プロセス・コマンド	56
一括読み込みモードの TMI C API 関数	57
プロシージャの編集	57
オンデマンドでのプロセスの実行	58
TMIRunTI の使用	58
TMIRunTI の構文	59
TMIRunTI 設定ファイル	62
TMIRunTI のリターン・コードとエラー・メッセージ	65
TMIRunTI に関するその他の考慮事項	67
synchronized() を使用した TurboIntegrator プロセスのシリアライズ	67
synchronized()	68
管理者による TurboIntegrator セキュリティーの割り当て	70

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 . . . 73

日課の開始時刻に関する重要な注意事項	74
日課の編集	74
日課の有効化	74
日課の無効化	75
日課の削除	75
オンデマンドでの日課の実行	75
ChoreCommit の使用	75
サーバー始動時における日課の実行	76

付録 A. TurboIntegrator Tutorial . . . 77

チュートリアル用のデータ・ディレクトリーの設定	77
-------------------------	----

TurboIntegrator の概要	78
TurboIntegrator プロセスの作成	78
TurboIntegrator を使用したディメンションの作成	79
キューブの作成とデータの処理	86
詳細スクリプトの処理	91
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の編集	91
サブセットの作成	99
属性の作成	101
付録 B. TurboIntegrator の予約語	103
規則関数の名前	103
プロセス関数の名前	106
暗黙的な変数の名前	109
TurboIntegrator のキーワード	110
特記事項	111
索引	115

はじめに

このマニュアルは、IBM® Cognos® TM1® 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 TM1 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。

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

Web 上でマニュアルを見つけるには、IBM Knowledge Center (<http://www.ibm.com/support/knowledgecenter>) へ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。

サンプルに関する特記事項

Sample Outdoors Company、Great Outdoors Company、GO 販売、Sample Outdoors または Great Outdoors の名前のすべてのバリエーション、および Planning サンプルでは、IBM および IBM のお客様向けのサンプル・アプリケーションを開発するために使用されるサンプル・データにより、架空のビジネス・オペレーションが描出されています。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、著作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。無断の複製は禁止されています。

ユーザー補助機能

ユーザー補助機能は、運動障害または視覚障害など身体に障害を持つユーザーが情報技術製品を快適に使用できるようにサポートします。

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 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 TM1 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、TM1 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

TM1 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できます。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。

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、TM1 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 TM1 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 TM1 参照ガイド』の「TM1 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 TM1 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
- TM1 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 (TM1 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 TM1 プロセスのコンテナ・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、73 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 TM1 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. TM1 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で使用できます。また、プロシージャ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 TM1 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、ストリング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 TM1 サーバーに同時接続。

プロセスが既に実行している同じ TM1 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 (TM1 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 (TM1 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「**アクティブなサンドボックスを使用**」プロパティを選択することで、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、「**アクティブな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 インターフェイスで利用できる「**アクティブなサンドボックスを使用**」プロパティと同じです。

詳細については、『IBM Cognos TM1 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 2 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 TM1 TurboIntegrator でインポートする方法について説明します。TM1 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、TM1 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、TM1 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 TM1 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. TM1 が遭遇する変数を特定します。9 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。12 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。14 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・ディレクトリにあります。TM1 のデフォルト・インストール・ディレクトリ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 TM1 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 TM1 サーバーを実行している場合、ファイルは TM1 Windows クライアントと TM1 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 TM1 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。

表 1. サンプル・テキスト・データ

列 1	列 2	列 3	列 4	列 5	列 6
New England	Massachusetts	Boston	Supermart	2 月	2000000
New England	Massachusetts	Springfield	Supermart	2 月	1400000
New England	Massachusetts	Worcester	Supermart	2 月	2200000
New England	Connecticut	Hartford	Supermart	2 月	1240000
New England	Connecticut	New Haven	Supermart	2 月	2700000
New England	Connecticut	Greenwich	Supermart	2 月	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。

表 2. 「変数」タブ

変数名	変数の型	サンプル値
V1	文字列	New England
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	State
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	_
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
State	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。

- 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。TM1 では、この設定はほとんどの場合に「数値」です。
- 「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 12 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

- 「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せん。

- “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。
- “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。
- 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。
- 「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 TM1 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。

TM1 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、TM1 から確認メッセージが発行されます。

TM1 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 TM1 に定義します。15 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. TM1 が遭遇する変数を特定します。16 ページの『キューブ変数の定義』を参照してください。
3. 生成されるキューブにおける様々なデータ・タイプに対して、異なる変数をマッピングします。16 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』、17 ページの『キューブのデータ変数のマッピング』、16 ページの『キューブ変数のマッピング』、および 17 ページの『集約変数のマッピング』を参照してください。

- プロセスを保存して、実行します。17 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。

TM1 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

- 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」ウィンドウの「データ・ソース」タブをクリックします。
- 「データ・ソース・タイプ」で「テキスト」を選択します。
- 「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

- 「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。

- 「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。
- 「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 TM1 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
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、TM1 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、TM1 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

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
Units	Units	measures	4
1 月	1 月	month2	5

- 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

- 「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。

TM1 は、プロセスに名前を付けて保存するように指示します。

- 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。

- 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 3 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: TM1 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 TM1 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い場合は失敗します。

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。TM1 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 TM1 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を使用して生成し、その MDX 文を、TM1 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 TM1 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 4 章 TM1 のビューまたはサブセットからのインポート

IBM Cognos TM1 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。TM1 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。

TM1 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 TM1 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. 「TM1」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。ビューを作成した後、「サーバー・キューブ・ビューの参照」ダイアログ・ボックスでそれを選択します。

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、TM1 ビューのインポートを完了します。

データ・ソースとして TM1 サブセットを使用

TurboIntegrator により、TM1 デイメンション・サブセットからデータを抽出し、その情報を別の TM1 オブジェクトに移すことができます。次の例では、“Region” デイメンションの“Europe”集約が抽出され、“Region_Europe”という新しいデイメンションを構成するために使用されています。

デイメンション・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、一般に別のデイメンションです。デイメンション・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。

TM1 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・ソースとしてデイメンション・サブセットを定義』を参照してください。

データ・ソースとしてデイメンション・サブセットを定義

デイメンション・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. 「TM1」をクリックして、「データ・ソース・タイプ」ボックスで「デイメンション・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イ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。

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

デイ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 TM1 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 TM1 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、TM1 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、TM1 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 5 章 MSAS からのインポート

IBM Cognos TM1 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、TM1 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 TM1

TM1 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、TM1 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 TM1 の場合、TM1 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 TM1 Admin Server ホストのマシン名。	MyServer
Datasource TM1 サーバーの名前。	Sdata
userID TM1 ユーザー名。	Admin
password TM1 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、TM1 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 TM1 Admin Server ホストのマシン名。	MyServer
カタログ TM1 サーバーの名前。	Sdata
userID TM1 ユーザー名。	Admin
password TM1 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)

TM1 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=TM10LAP.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 TM1 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 TM1 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。

30 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. デイメンションを定義します。

31 ページの『「キューブ・デイメンション」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。

32 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、このタブに必要な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、TM1 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、TM1 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 TM1 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、TM1 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 TM1 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、TM1 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、「**TM1 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。「**TM1 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、「**TM1 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。

TM1 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表示されます。

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 TM1 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。

```

Dimension Members
  . All store2
+ . Canada
- . Mexico
  + . DF
  + . Guerrero
  + . Jalisco
  + . Veracruz
  + . Yucatan
  + . Zacatecas
  . USA
  + . CA
  + . OR
  + . WA

```

図1. サンプル・ディメンション

TM1 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、TM1 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、TM1 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。

TM1 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 TM1 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 デイ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、実行するデイメンション・ロード・プロセスでのコピー元とコピー先のデイ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 デイメンションのロード」タブをクリックします。
2. 「TM1 デイメンションの操作」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
デイメンションの作成	ODBO データ・ソースのデイメンションをコピーして、新しいデイメンションを作成します。
デイメンションの再作成	現存するデイ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イメンションを再作成します。
デイメンションの更新	「デイメンションの更新」は、要素を挿入または削除するデイメンションが TM1 に既に存在していることを前提としています。 <ul style="list-style-type: none">• ODBO データ・ソースには要素が存在し、TM1 には存在しない場合、その要素は、デイメンションに追加されます。• TM1 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・デイ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・デイ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・デイメンション内に <element_name>_1 として作成されます。これによりデイ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、デイメンションに影響しません。

3. 「デイメンションを含む 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イ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・デイメンション」のリストをクリックし、インポートするデイメンションを選択します。
5. デイメンションを更新または再作成する場合は、「ロードする TM1 デイメンション」のリストをクリックし、リストからデイメンションを選択します。

デイメンションを新しく作成する場合は、新しいデイメンションの名前を「ロードする TM1 デイメンション」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、 「実行」アイコンをクリックして、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、TM1 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。

TM1 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 TM1 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、TM1 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「TM1 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 6 章 IBM Cognos TM1 Package Connector を使用してデータをインポートする

現在、IBM Cognos TM1 Package Connector は、SAP Business Warehouse と他のリレーショナルおよび ODBC データ・ソースに対し、IBM Cognos Business Intelligence のパッケージと共に使用するものとしてサポートされています。

サポートされる特定のソフトウェアについて詳しくは、サポートされているハードウェアとソフトウェア (<http://www.ibm.com/support/docview.wss?uid=swg27040698>) のリンクを参照してください。

TM1 Package Connector を使用してパッケージをインポートするとき、以下のタスクを行います。

- SAP BW データ・ソース: SAP BW Infoquery を作成します。Infoqueries は、SAP BW データ・ソースからインポートするときの重要なステップです。
- Framework Manager にパッケージを作成する。
- TM1 Package Connector を使用する TurboIntegrator プロセスを作成する。

このトピックでは、TM1 Package Connector の使用方法について説明し、SAP BW データを使用する際のガイドラインと Framework Manager でパッケージを作成する方法についても説明します。これらの関連項目の詳細は、SAP ドキュメントと IBM Cognos Framework Manager ドキュメントを参照してください。

IBM Cognos TM1 Package Connector は、TM1 Perspectives または TM1 Architect の 64 ビット・バージョンでは機能しません。

IBM Cognos Package Connector は、メイン TM1 インストール・ディスクとは別の CD に収納されているオプションのコンポーネントです。IBM Cognos TM1 Package Connector のインストールと構成についての詳細は、『IBM Cognos TM1 インストール ガイド』の「IBM Cognos TM1 Package Connector のインストール」を参照してください。

Cognos BI サーバーへの接続を確立する

コンポーネントをインストールして設定したら、以下の説明に従って接続を確立します。

手順

1. TM1 を実行します。
2. TM1 サーバー・エクスプローラーの左側のペインで、「プロセス」グループを右クリックし、「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. 「Data Source Type」ダイアログ・ボックスで、「IBM Cognos Package」オプションを選択します。

4. デフォルトで「**Package**」が選択されています。

要確認:

パッケージおよびディメンション

「パッケージおよびディメンション」オプションは、ディメンションを使用してモデル化されたソース (SAP Business Warehouse やディメンションを使用してモデル化されたリレーショナル・ソースなど) からデータを簡単にインポートするためのものであり、カスタマイズは簡単ではありません。

1 つのパッケージの 1 つのディメンションから階層をインポートするだけの場合は、代わりに「**Dimension**」を選択することができます。

カスタム・クエリー

「カスタム・クエリー」オプションは、DMR または非 DMR ソースに対して指定することができ、照会する項目をより柔軟に選択したり、標準の TurboIntegrator プログラミング機能にさらにオープンにアクセスしたりするのを可能にします。

詳しくは、『発行されたパッケージに接続する』を参照してください。

5. 接続情報を入力します。

IBM Cognos BI サーバーが匿名ログインを許可されている場合は、「Log on」をクリックします。そうでない場合は、ここにログオン資格情報を入力します。

• **認証ネームスペース**

プルダウン・リストには使用可能なすべての認証ネームスペースが表示されません。

• **ユーザー ID と パスワード**

選択した認証ネームスペースにユーザーのユーザー ID とパスワードを入力します。

• **サインオン**

TurboIntegrator プロセスに関連するサインオン情報を管理することができます。ボタンは、BI サーバーに正常にログオンすると有効になります。

「BI サーバー・ログオン」というサインオンが、最初の BI サーバーへの認証時に自動的に作成されます。最初のログオンの後に「Connection」タブの資格情報を更新する場合、これらの変更はこのサインオンを更新しません。

「Manage Signons」ダイアログ・ボックスを使用してサインオンの変更を行う必要があります。サインオンは追加、変更、および削除することができます。さらに、「Connection」タブに資格情報を入力してから「Log on」をクリックし、サインオン・ウィンドウでサインオンを修正する場合、「Connection」タブの資格情報は更新されません。

匿名ログオンが無効な場合に、サインオンを削除すると、TurboIntegrator プロセスを実行できなくなります。

ディメンションと数値データを使用する場合に、データ・ソースのサインオンが作成される場合があります。「Signons」ウィンドウからこれらを管理することができます。

- ログオン/ログオフ

「Log on」をクリックして、BI サーバーにログオンします。ログオンすると「Log off」が有効になり、BI サーバーのログアウトが可能になります。

発行されたパッケージに接続する - 「パッケージおよびディメンション」オプション

BI サーバーに接続したら、「パッケージ」タブに移動して、使用するパッケージを選択します。

パッケージを選択した後に数値データをインポートしない場合は、「Dimension」タブに移動します。

手順

1. 「参照」ボタンをクリックして、使用可能なパッケージから選択します。選択にはディメンションがあるパッケージのみが使用できます。

「Browse Metadata」ウィンドウからパッケージを選択すると、「Select Package」フィールドに自動的に入力されます。パッケージを選択した後、「Edit」ボタンが有効になります。

2. 「ロード先の TM1 キューブ」オプションを選択します。キューブを作成する場合は、キューブ名を入力します。既存のキューブを再作成するか、更新する場合は、リストからそれを選択します。
3. 実行するインポート操作を指定します。

他のタイプのデータソースのようにこのインポートの必要に応じてキューブの操作とデータ操作を完了します。これらオプションの詳細については、変数のマッピングを参照してください。

SAP BW ベース・パッケージの場合は、インポートしている選択済みのキー値に関する複数の行が結果セット内に含まれる場合、「Accumulate Values」を選択する必要があります。結果を検証してください。

4. パッケージのディメンションを定義するには、「Dimension」タブをクリックします。
5. 「Select Hierarchies」または「Select Attributes」をクリックし、表示されたダイアログ・ボックスを使用して階層と属性の構造を定義します。

階層と属性の以下の点に注意してください。

- デフォルト階層

階層を事前に選択しないで TM1 ディメンションを指定する場合、TM1 は自動的に最初のをデフォルト階層として選択します。チェック・ボックスを選択または選択解除して選択を変更することができます。

- マッピングの前の選択

ディメンションにマッピングする前に階層とフィルターなどのすべてのサブ項目を選択することもできます。事前選択された階層ディメンション上のディメンションに入力する場合、すべての選択が自動的に適用されます。フィールドのディメンション名をクリアしてマッピングされたディメンションを選択解除すると、対応するディメンションの下にある事前選択された階層すべてが自動的にクリアされます。

- **名前と参照**

指定した階層を明確に識別できるように、階層は階層名と階層参照を表示します。階層名は繰り返されますが、階層参照は一意です。

- **フィルター**

パッケージにフィルターが定義され、階層が選択されると、「Select Filters」ボタンが有効になり、それを使用して適用するフィルターを選択します。

フィルターが選択した階層に適切であることを確認してください。適切でない場合はクエリーが実行できない場合があります。

- **階層内の複数のレベル**

IBM Cognos では、1 つの階層に複数のレベルを定義することができます。階層に複数のレベルを定義する場合、属性名を各レベル全体で繰り返すことができます。

例えば、「City」という名前のディメンションには、「City」、「Geography」、「Geography_link」などが含まれることがあります。各階層は複数のレベルを定義できます。

例えば、「Level Label」、「Level Number」、「Hierarchy Unique Name」などのディメンションを、さまざまな階層の下にあるさまざまなレベルで同一のものとして行うことができます。TM1 は、2 つの要因である属性外部名と属性ロールに基づいて属性の集約を実行します。

優先順位は、属性外部名、属性ロールの順になります。属性に同じ属性外部名値が含まれている場合は、それらの属性は集約されます。

属性に外部名がない場合は、そのロール・シグネチャー (すべてのロールがロール・シグネチャーに合成する場所) が、集約の要因として評価されます。このコンテキストにおいて「合成」とは、複数の同一のロールがまとめて TM1 の単一の属性の中にグループ化されているアイテムのことを指しています。

属性マッピングを指定する場合は、インポート中に外部名やロール・シグネチャーがマッピングされた属性に一致するすべての属性が選択されます。複数の属性をマッピングすることができます。

6. 属性の選択

ディメンション属性を TM1 属性にマッピングする場合は、以下のいずれかを実行します。

- 属性を新しい属性にマッピングするには、その SAP 属性の「TM1 Attribute」フィールドに新しい属性の名前を入力してから、「Attribute Type」を選択します。TurboIntegrator プロセスを実行したときに、新しい属性が作成されます。
- 属性を既存の属性にマッピングするには、対応する「TM1 Attribute」フィールドをクリックし、属性を選択してから、「Attribute Type」を選択します。

属性の型	説明
テキスト	属性をstring値として識別します。
数値	属性を数値として識別します。
別名	属性が関連付けられるディメンションの別名として属性を識別します。この属性を使用すると、TM1 ユーザー・インターフェースにディメンションを別名で表示できます。ディメンションの別名は、他のディメンション別名や実際のディメンション名とは異なった名前であることが必要です。

数値データをディメンションにマッピングする必要があります。

数値データが、「Dimensions」タブの最終行に表示されます。

数値データを選択するには、「Select Measures」をクリックします。

TurboIntegrator は数値データのデフォルトがないので事前を選択しません。

7. 「**Select Measures**」をクリックします。

「Select Measures」ダイアログ・ボックスが開きます。

注: パッケージには複数の数値データ・ディメンションがある場合があります。数値データの名前は、パッケージの数値データ・ディメンション名を含んでいます。

8. TM1 にインポートする各数値データを選択します。
9. 「**Filters**」をクリックして数値データ・クエリーに適用するフィルターを選択します。フィルターが選択した階層に適切であることを確認してください。適切でない場合はクエリーが実行できない場合があります。
10. 「**OK**」をクリックします。
11. 数値データをディメンションにマッピングします。

数値データが既存のディメンションにマッピングする場合は、「**TM1 ディメンション**」列をクリックし、数値データに対応するディメンションを選択します。

数値データから新しいディメンションを作成する場合は、ディメンションの名前を「**TM1 Dimension**」列に入力します。

12. 数値データで「**TM1 Dimension Action**」を選択します。
13. **ディメンション設定**

「Dimension settings」タブを使用して、指定したディメンションに上位集約を定義します。

14. プロンプトの編集

「Prompts」ボタンをクリックして、Cognos プロンプト編集ウィンドウを開いてプロンプトの値を設定して変更します。SAP BW 変数が IBM Cognos パッケージのプロンプトとして表示されます。必須のプロンプトがパッケージにある場合は、TurboIntegrator プロセスがクエリーを実行する前にプロンプト値を入力する必要があります。そうしないと処理が失敗します。SAP BW からデータを抽出する場合は、データのセグメント化にプロンプトを指定してください。データのセグメント化にプロンプトを指定すると、SAP BW に送られるクエリーは抽出用に最適化され、並行クエリーを使用して抽出をさらに高速化できるようになります。46 ページの『SAP BW データを使用する際のガイドライン』を参照してください。

「プロンプト」ボタンを使用して、ユーザー・インターフェースを使用して値や範囲を設定できるようにします。値列の下のセルをクリックしてプロンプト値を設定します。

または、以下の TurboIntegrator プロセス/API を使用することができます。

プロンプト名を知っている場合は、プロンプト名と値を渡して CGAddPromptValues を直接呼び出すことができます。これらの関数は、親 TurboIntegrator プロセスのプロローグに入力する必要があります。

これらの関数は TurboIntegrator プロセスで定義されているプロンプトを取得します。基礎となる TurboIntegrator プロセスにそれらを定義するには、TurboIntegrator プロセスでプロンプト・ウィンドウが起動している必要があります。「Prompts」をクリックし、定義されたクエリーすべてからプロンプトをプルして「OK」をクリックし、これを使用できるようにします。プロンプトに変更を加えるときはいつでも、「OK」をクリックして、プロンプトを使用可能にする必要があります。

TurboIntegrator プロセス API	説明
CGPromptSize()	設定する必要があるプロンプトの合計数を返します
CGPromptGetNextMember(int index)	インデックス (0-(CGPromptSize()-1)) のプロンプトを返します
CGAddPromptValues(promptName, value 1, value 2...)	プロンプト名を指定してプロンプト値を設定します
例	<pre>count=CGPromptSize(); while(i<count); prmptname=CGPromptGetNextMember(i); CGAddPromptValues(prmptname, '1999-01-01','2009-01-01'); i=i+1; end;</pre>

15. マッピングされたディメンション、階層、および数値データがある場合、「Test Queries」をクリックしてクエリーをテストすることができます。

「クエリーをテスト」は、TurboIntegrator プロセスが実行する前にクエリー実行をトリガーします。クエリーが正常に実行するかどうか事前テストしたり、見つからないプロンプト値やサインオン情報を入力する際に使用することができます。

16. 「**Show Namespace**」チェック・ボックスをクリックして、ディメンション・リストにネームスペースを追加します。
17. 必要に応じて「Advanced」タブと「Schedule」タブに記入します。詳細については、詳細プロシージャと日課の編集を参照してください。Package Connector は、各ディメンションで TurboIntegrator プロセスも生成します。これにより、必要に応じて独自の TurboIntegrator プロセスのスクリプト・ステートメントを自由に追加できます。
18. TurboIntegrator プロセスを保存して実行します。TurboIntegrator プロセスの保存と実行を参照してください。

複数階層

数値データ・クエリーの発行時、IBM Cognos TM1 Package Connector には、1 ディメンション当たり 1 つの階層のみが含まれます。

1 つのディメンションの複数階層を選択して、TurboIntegrator プロセスに数値データを含むと、TM1 Package Connector は次々に各階層を置き換えて複数の数値データ・クエリーを発行します。この動作は、各階層のリーフ・レベル・メンバーが互いに異なる場合に有効です。

リーフ・レベル・メンバーがその階層で同じである場合は、必要な数値データに沿って 1 つのディメンションにつき 1 つの階層だけを選択して TurboIntegrator プロセスを作成してください。同じディメンションに複数の階層構造をマージするには、別のプロセスを作成します。

1 つのディメンションのインポート

「データ・ソース」タブのディメンション・プルダウン・オプションを使用すると、簡単に 1 つのディメンションを定義することができます。

IBM Cognos Package オプション・プルダウンから「**Dimension**」を選択し、「Dimension」タブを直接開いて、実行する操作と階層を定義する操作を指定します。

手順

1. パッケージを検索します。

「参照」ボタンをクリックして、使用可能なパッケージから選択します。

2. 「**Dimension to load from:**」を指定します。

プルダウン・リストには選択したパッケージで使用可能なディメンションがリスト表示されます。

3. 「**ロード先の TM1 ディメンション**」を指定します。

既存のディメンションを更新する場合は、プルダウンに使用可能なディメンションがリスト表示されます。

4. 他のデータ・ソースのように「**TM1 ディメンションの操作**」を選択します。詳細については、ディメンションのマッピングを参照してください。
5. 「**Top Consolidation**」を指定します。

ディメンションの上位レベル集約を作成する場合は、最上位集約の名前をここに入力します。

作成されるディメンションには、入力した名前の集約が含まれることとなります。例えば、「**Top Consolidation**」に「*Total*」と入力すると、集約の子としてインポートされたすべての要素と共に「*Total*」という名前の上位レベルの集約がディメンションに含まれることとなります。

6. プロンプトの指定方法について詳しくは、53 ページの『SAP BW クエリーにセグメント化プロンプトを指定する』を参照してください。
7. マッピングされたディメンション、階層、および数値データがある場合、「**Test Queries**」をクリックしてクエリーをテストすることができます。

「クエリーをテスト」は、TurboIntegrator プロセスが実行する前にクエリー実行をトリガーします。クエリーが正常に実行するかどうか事前テストしたり、見からないプロンプト値やサインオン情報を入力する際に使用することができます。

発行されたパッケージに接続する - 「カスタム・クエリー」オプション

IBM Cognos TM1 Package Connector の「カスタム・クエリー」オプションを使用すると、ディメンションを使用してモデル化されたすべての種類のソースに接続してカスタマイズされたクエリーを作成できます。カスタム・クエリーは、非ディメンション・モデル化ソースに接続するための唯一の方法です。

このタスクについて

IBM Cognos Business Intelligence (BI) サーバーに接続した後に、作業対象にするソース・パッケージと、このプロセスで IBM Cognos TM1 にインポートするそのソース・パッケージ内のフィールドを指定します。

手順

1. 「参照」ボタンをクリックして、使用可能なパッケージから選択します。

「メタデータを参照」ウィンドウからパッケージを選択すると、「**パッケージを選択**」フィールドに自動的に入力されます。パッケージを選択した後、「**編集**」および「**クエリーの編集**」ボタンが有効になります。

「**編集**」ボタンはデータ・ソースに関する情報を示し、Content Store のサインオンにパスワードが必要な場合にカスタム・サインオンを作成できるようにします。

2. 「**クエリーの編集**」ボタンをクリックします。ソース・パッケージで使用可能なフィールドが表示されます。
3. 抽出に必要な各フィールドを選択して、「**追加**」を選択します。
4. 「**クエリー・オプション**」を選択します:

プレビュー

縦欄リストを表示します。MDXにより、変換に使用されるコードが表示されます。

- 一致 ソース・パッケージまたはレポートのクエリー・アイテムを、クエリー中の列にリンクします。これが必要なのは、ソースが変更されて、クエリー・アイテムが移動または名前変更された場合のみです。一致オプションを使用すると、アイテムが移動または名前変更された後に、列とクエリー・アイテムを再びマッチングすることができます。

自動集計

デフォルトで選択されています。クエリーは、重複行を集計するためのSQLを生成します。リレーショナル・パッケージに基づくクエリーの場合、「自動集計」機能を有効にすることも、Cognos TM1 Package Connectorがソース・データから取り出す行数を減らして、キューブ作成のパフォーマンス向上に役立ちます。ソース・データが基礎表と同じ細分性で使用されている場合、この項目はチェック・マークを外したままにします。ソース・データが集約されていない場合、この項目にはチェック・マークを付ける必要があります。この設定値が有効になるようにするため、クエリーに適切なIDおよびファクト使用属性が設定されていることを確認します。これらの設定は、ソース(Cognos Framework Manager またはレポート)に設定する必要があります。SQLを検討して、適切なグループ化関数および集計関数が適用されていることを確認します。平均の計算には「自動集計」を使用しませんが、通常は「自動集計」を使用することが推奨されます。

Null 値の抑制

デフォルトで、このオプションは定義されたパッケージのガバナー設定を使用するように設定されています。この設定をオーバーライドするには、「はい」(Null 値を抑制する) または「いいえ」(抑制しない) のいずれかを選択します。

セグメント化の際の並行クエリー・カウント (1-64)

このオプションを適用できるのは、SAP BW からデータを抽出するときに、データのセグメント化に変数が選択されている場合だけです。このオプションが設定され、データがセグメント化されるとき、同時に n 件のクエリーがソース・システムに送られます。これにより、抽出時間が改善されることがあります。

5. 「検証」ボタンを選択して、選択した情報から有効なクエリーを生成できることを検証します。
6. 「OK」をクリックして、「プロンプト」および「プレビュー」ボタンを使用可能にします。

プロンプト

プロンプト値を設定して、データのセグメント化にプロンプトを選択する必要がある場合は、「プロンプト」をクリックします(SAP BWのみ)。

プレビュー

生成されるMDXを確認するために、必要に応じてデータをプレビューするためには、「プレビュー」をクリックします。

7. 「変数」タブをクリックします。これ以降は、他のすべての TurboIntegrator プロセスと同じ手順になります。 9 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。

ヒント: SAP BW データを対象に作業するとき、データのセグメント化プロンプトが使用可能になるのは、ソース・システムからデータとキー値だけを選択した場合に限られます。追加のメタデータをプルする場合は、それらの項目を別のクエリーに定義して、抽出を行うときにデータをセグメント化できるようにします。

TurboIntegrator 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. 「Execute」ボタンをクリックします。

TM1 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. プロセスを保存します。

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。指定したキューブが作成および格納され、必要なすべてのディメンションが作成されていることを確認します。

結果の確認

パッケージを TM1 にインポートして、新しいキューブを作成するプロセスを定義すると、次の処理が行われます。

- キューブに含めるよう選択した各ディメンションで、TM1 は、ディメンションをインポートするプロセスを生成し、対応するディメンションを作成します。
- TM1 は、上述のディメンション作成プロセスを実行し、キューブを構築し、データ値をインポートするためのマスター・プロセスを生成します。
- 階層のレベルのキーとして定義されたクエリー項目は、要素名として使用されません。

SAP BW データを使用する際のガイドライン

Framework Manager で作成された SAP ベース・パッケージを使用する場合の特殊集約があります。

SAP クエリーの作成、Framework Manager での SAP ベース・パッケージの作成、および TM1 での使用方法についての詳細は、Framework Manager でパッケージを使用して SAP BW データを使用するを参照してください。Framework Manager でのパッケージの作成の詳細については、『Framework Manager ユーザー・ガイド』の「パッケージの作成または変更」を参照してください。

Framework Manager でパッケージを使用して SAP BW データを使用する

IBM Cognos BI Framework Manager で作成された SAP ベースのパッケージを使用するときには、特別な考慮事項があります。

Framework Manager で作成して Content Manager に発行した SAP ベース・パッケージを使用して、SAP BW データを IBM Cognos TM1 Package Connector で利用することができます。この方法を使用して SAP BW データを使用することをお勧めします。

TM1 Package Connector を使用して SAP BW クエリー・ソースからディメンションとファクト・データの両方をインポートすることができます。SAP BW キューブを IBM Cognos TM1 キューブとして再構築する方法について、以下で説明します。その場合は、SAP BW クエリー・パッケージを特殊な形式にする必要があります。

IBM Cognos を使用してディメンションとファクトの両方へアクセスする SAP BW クエリーをインポートするには、以下の 3 つの段階があります。

- SAP Business Explorer Query Designer で BW クエリーを作成する
- Framework Manager でパッケージを作成する。
- TM1 Package Connector を使用する TurboIntegrator プロセスを作成する。

この展開プロセスは SAP BW データのみに制限されています。

データ・ソースは SAP BW データ・ソースで定義されている明確に構築されたクエリーでなければなりません。


パッケージの作成の詳細については、『*IBM Cognos Framework Manager ユーザー・ガイド*』の「パッケージの作成または変更」を参照してください。

SAP Business Explorer Query Designer で BW クエリーを作成する

インポートするキューブを含むクエリーを作成する必要があります。クエリーは、データベースの単一の InfoCube に基づくようにします。複数のソースに基づくクエリーは、データ取得中に SAP BW エラーを発生させる場合があります。

使用する Query Designer バージョンによって、使用する手順は異なります。


手順

1. 「Query Designer」で、「New Query」をクリックします。
2. 「New Query」ダイアログ・ボックスで、インポートするキューブを含んでいる情報プロバイダーを選択します。
3. 「Tools」アイコン  をクリックして、「InfoObject」の技術名を表示します。
4. インポートする特性を、左側の列の「InfoObject」カタログからページ右側のいずれかのフィールドにドラッグします。例えば、「Columns」または「Rows」です。

選択する特性は、キューブのメタデータを定義します。特性は以下の制限に従う必要があります。

- データ抽出要求を分割するためには、1 つ以上の任意の変数が必要になります。実質的には、通常の要求より高いスループットになる可能性があります。
 - データ・ソースを代表する特性を検索します。特性は、キューブの数値データか、キューブ・ディメンションのディメンションのいずれかの重要なデータになります。
 - 指定するか変数によって、表示階層にいずれの特性も割り当てないでください。
 - SAP BW クエリーのすべての重要なデータは数値でなければなりません。
 - 「**Currency/Unit**」の特性を選択しないでください。
 - 選択したすべての重要なデータは同じ通貨を使用していることを確認します。
 - TMI Package Connector を使用して抽出する SAP BW クエリーの特性のみを含めます。不必要な特性を含むとデータ量が増加し、パフォーマンスに影響を与えます。
 - 特性は、クエリー定義の「**Columns**」または「**Rows**」フィールドにコピーする必要があります。「**Free Characteristics**」または「**Filter**」フィールドにコピーすると、特性はパッケージからのインポート時にディメンションとして表示されますが、分割されたデータ抽出が発生した場合に使用されるストリーム抽出プロセスは値をフェッチできません。
 - フィルターが定義されている場合は、クエリー定義のいずれかに含まれているディメンションのみを参照する必要があります。
 - フリー特性を含む場合は、重要なデータの抽出の特性に表示される値はありません。フリー特性のフィルターは、返された SAP BW データのフィルターのように動作します。InfoCube のサブセットを定義するフィルターとしてこれを使用することができます。
 - クエリーでは、入力プロンプトでなく、選択リスト・プロンプトを使用します。選択リスト・プロンプトは、データを分割するための値を入力します。
5. キューブを格納するメタデータを定義するには、含めるよう選択した各特性のプロパティを変更する必要があります。特性を右クリックして、「**Properties**」を選択します。
 6. 「**Properties of Characteristic**」ダイアログ・ボックスで、「**Display As**」値を「**Key**」に変更し、「**Suppress Results Rows**」値を「**Always**」に変更します。個々に適用した制限やフィルターは次回まで有効になりません。
 7. 手順 4 で選択した各特性に対して、手順 5 ~ 6 を繰り返します。

注: 必要な特性のみ選択してください。過剰なメモリー使用、システム・パフォーマンスの低下や障害を避けるためには、クエリーに含む特性を慎重に考慮してください。データ量が過剰かどうか確認する場合は、SAP BW 管理者に問い合わせることをお勧めします。

8. 「**Queries Properties**」アイコン  をクリックし、「**Extended**」タブで「**Allow External Access to this Query**」チェック・ボックスにチェック・マークを付けます。これは、クエリーを Framework Manager に公開します。

9. 「Save」をクリックして、「Description」と「Technical Name」に新しいクエリーを入力します。「Technical Name」フィールドには SAP BW の命名規則を使用することをお勧めします。直観的な名前や標準の命名規則の先頭に文字「Z」を付けます。この技術名を書き留めておいてください。Framework Manager でのクエリーの検索に必要になります。

タスクの結果

変数を作成する準備ができました。SAP Query Designerの使用についての詳細は、SAP BW ドキュメントを参照してください。

変数を作成:

クエリーの任意のプロンプト・パラメーターを作成できるようになったので、TM1 Package Connector はより小さなクエリーを SAP に発行して、データ・セット全体を小さなチャンクで取得できます。

始める前に

このテクニックは、データ量が SAP BW サーバーでエラーを発生する場合に使用する必要があります。

TM1 Package Connector で使用するための SAP BW データを抽出する場合の変数使用に対する規則はありません。ただし、多すぎるデータ要求は避けてください。パフォーマンスが低下したり、SAP 環境内にメモリー不足のメッセージが表示されてエラーの原因になります。

変数をデータ抽出の分割に使用すると、TM1 Package Connector はまず、変数が定義されるディメンションに存在するすべてのメンバーをフェッチします。その後、TM1 Package Connector は個々のデータ・フェッチを実行し、変数を満たすためにディメンション内の個々のメンバーのファクト・データを抽出します。

これにより、TM1 Package Connector は SAP BW サーバーが処理できる管理可能なチャンクにデータ抽出を分割できるようになります。これを適用するディメンションについて設定されている標準はありません。最適なパフォーマンスを実現するには、SAP BW データを理解し、ファクト・データを均等に分割するディメンションを決定する必要があります。

変数を定義するディメンションを慎重に選択してください。最適なパフォーマンスを実現するには、いくつか実験する必要がある場合があります。例えば、メンバーとして「United States (US)」、「Canada (CA)」、「Mexico (MX)」の3つの地域が含まれている [REGION] ディメンションがあるとします。ほとんどの取引は US (90%) で、残りの取引 (10%) は「Canada」と「Mexico」で均等に分配されている場合、このディメンションはデータを均等に分割にしません。クエリーの結果は、非常に大きな要求 1 つ (US) と、2 つの小さな要求 (CA と MX) になります。従って、このディメンションは良い候補にはなりません。

多数の非常に小さな要求を発生する 1 つのディメンションに 1 つの変数を適用することは望ましくありません。例えば、SAP BW 環境で良く使用されるディメンション [OMATERIAL] は、多数の小さな要求の実行を発生させるので、良い候補にはなりません。

データの均等な分割に役立つ 10 件の異なるコスト・センターのデータを均等に分割するディメンションを [COSTCENTER] に定義したとします。適切に実行するセクションにデータを分割できるので暦年や暦月は良い候補になります。

データ抽出のクエリーに変数を適用する必要はありません。抽出によっては、変数が適用されていない場合に最適に動作します。

2 つの環境は違うので、式を適用することはできません。ただし、SAP BW 環境を妨害しないよう慎重に扱ってください。

手順

1. 「**Query Designer**」で、前の手順で選択した特性を右クリックし、「**Restrict**」を選択します。

データを確実に均等に配布するには、キューブの代表で、大きな数の値にならない特性を選択します。変数の各値の行数が類似した結果変数にしたい場合は、細かすぎる結果変数 (例えば、1 つの値当たりの行が少ないためにクエリーの数が極端に多くなる) や、粗すぎる変数 (例えば、1 つの値当たりの行が 100 万以上) を避けるべきです。

2. 「**Selection for ...**」ダイアログ・ボックスで、「**Variables**」タブをクリックして、「**Description**」ウィンドウ内の任意の場所を右クリックして、「**New Variable**」を選択します。

注: 既に選択されている特性の 1 つに変数がある場合は、新しい変数の作成を省略して手順 7 に進めます。

3. 「**New Variable Wizard General Information**」ページで、「**Variable Name**」と「**Description**」に入力し、特性としてディメンションを選択します。「**次へ**」をクリックします。
4. 「**Details**」ページで、「**Single Value**」、「**Multiple Single Values**」、または「**Interval**」を「**Variable Represents**」フィールドで選択し、「**Optional**」を「**Variable entry is**」フィールドで選択して「**Ready for Input**」チェック・ボックスを選びます。「**次へ**」をクリックします。
5. 「**Default Values**」ページで「**Default Value**」フィールドが空であることを確認します。
6. 「**Selection for ...**」ダイアログ・ボックスに戻るまで「**次へ**」をクリックします。新しい変数が「**Description**」ウィンドウに表示されます。
7. 変数を選択して右矢印をクリックし、選択した変数を「**Selection**」ウィンドウに移動してから、クエリーを保存します。これで Framework Manager でクエリーをインポートする準備完了です。

Framework Manager でパッケージを作成する

IBM Cognos Framework Manager で必要なパッケージを作成するには

- メタデータ・ウィザードを使用して SAP BW メタデータをインポートします。
メタデータ・ウィザードを使用してインポートする手順

Framework Manager は SAP BW クエリーをモデルにインポートし、Content Manager へエクスポートするパッケージを定義します。

インポートしたら、以下の点に注意してください。

- SAP BW クエリーで選択したディメンションは、「**Import**」ダイアログ・ボックスの「**Dimension Folders**」で使用できます。
 - 各ディメンションには 1 つ以上の階層が含まれます。
 - 階層に一致する名前の第 1 階層を常に選択してください。
 - 他の階層が使用できる場合は、階層内の必要なレベル一式を提供するものを選択します。
 - Framework Manager は、設定パラメーターが有効な場合にのみ SAP BW データからモデルに時間ディメンションをインポートします。時間ディメンションとしての構成の設定は、グローバル・エントリーです。インポートされた各ディメンションは時間文字列として扱われます。
- パッケージ パッケージ作成手順 を作成します。

Content Manager へ発行するためのパッケージを作成する場合は、2 つの階層をインポートしたこれらのディメンションの第 1 階層を非表示にします。第 1 階層は必須で、クエリーを正しく動作させるには、パッケージ中に存在する必要があります。表示したくない場合は、階層を非表示にすることができます。

SAP BW からメタデータをインポートする

SAP BW キューブのメタデータを Framework Manager のモデル定義にインポートする場合に、ディメンションが時間ディメンションかどうかが決まります。

正しく行うには、構成設定を設定する必要があります。デフォルトでは、値が設定されていません。

Framework Manager がインストールされた場所の下にある configuration ディレクトリーには、SAP BW アクセスを制御する設定ファイルがあります。ファイル名は "sapbw_config.xml" になります。インストールでは、このファイルを直接インストールしません。その代わりに、"sapbw_config.xml-sample" という名前のファイルをインストールします。"sapbw_config.xml" が見つからない場合は、"sapbw_config.xml-sample" をコピーするか、"sapbw_config.xml-sample" の名前を変更して作成します。

"sapbw_config.xml" には、次の行が含まれています。

```
<provider name="SAPBWODP">
  <!-- Must be activated for getting staging datatypes -->
  <parameter name="UseStgDTypes" value="true"/>
  <!-- Controls the use of a faster version of GetMembers.
  Default is "true".-->
  <parameter name="UseFastGetMembers" value="true"/>
</provider>
```

必ず "UseStgDTypes" は value = "true" にします。デフォルトでは、値は "false" になっています。

これを実行してから、Framework Manager を起動し、SAP BW からメタデータをインポートしてモデルを作成してください。

この変更によって、メタデータのインポートで、SAP BW キューブのディメンションが "Time" ディメンションであることを認識できるようになります。これは、ディメンションの最低レベルのメンバーに日付であるキー値があるディメンションです。

手順

1. Framework Manager で、「**Create a new project**」をクリックします。
2. 「**New Project**」ダイアログ・ボックスのフィールドを完了します。「**OK**」をクリックします。
3. 「**Metadata Wizard**」に従って操作を完了します。データ・ソースの選択を要求するプロンプトが表示されたら、新しいデータ・ソースを作成する必要がある場合は「**New...**」をクリックします。
4. 「**Select Objects**」ページで、前の操作で SAP BW クエリー・デザイナーで定義したクエリーを検索します。変数を作成した時に入力した技術名にリストをスクロールします。フォルダーの構造は、Hierarchies > Level definitions > Query Item definitions のようになります。
5. レベルに直接関連するメインのクエリー項目を選択します。それは、**(Key)**、**(Name)**などラベル化されたものです。
6. デフォルト値を使用して「**Metadata Wizard**」の残りの操作を完了し、「**次へ**」をクリックします。ディメンションを生成して、メタデータをインポートします。
7. 最後のウィザード画面で結果を確認し、「**Finish**」をクリックします。

パッケージの作成

Content Manager へ発行するためのパッケージを作成する場合は、2 つの階層をインポートしたこれらのディメンションの第 1 階層を非表示にします。第 1 階層は必須で、クエリーを正しく動作させるには、パッケージ中に存在する必要があります。表示したくない場合は、階層を非表示にすることができます。

手順

1. 「**Packages**」フォルダーをクリックして、「**Actions**」メニューから、「**Create**」、「**Package**」の順にクリックします。
2. 「**Provide Name**」ページで、パッケージの名前を入力し、必要に応じて説明と画面のヒントを入力します。「**次へ**」をクリックします。
3. 前の画面でインポートしたクエリーを選択します。
4. 「**Define objects**」ページで、パッケージから子オブジェクトを非表示または除外する場合は、それらを個々に選択する必要があります。親オブジェクトを除外すると、そのすべての子も除外されます。大きなキューブから多数のオブジェクトを除外 (または選択解除) するには、時間がかかります。

注: Framework Manager では Ctrl+Shift と Alt+Shift の機能が使用できます。これらキーのを使用して、キューブに含めるか非表示する複数のオブジェクトを選択します。例えば、大きなブランチに 2 つの項目だけを含む場合は、ブランチ全体を選択し、Ctrl+Shift を使用して含む項目を選択解除し、残りの選択されている項目を非表示にします。

オブジェクトの包含、除外、非表示についての詳細は、『*Framework Manager ユーザー・ガイド*』の「パッケージの作成または変更」を参照してください。

5. パッケージにデフォルトのアクセス権限を使用するかどうかを、以下のように選択します。
 - デフォルトのアクセス権限を使用する場合は、「**Finish**」をクリックします。
 - アクセス権限を設定する場合は、「次へ」をクリックします。
6. 「**Publish Package Wizard**」を開くように要求されたら、「**Yes**」をクリックします。
7. デフォルト値を選択して、「**Publish**」をクリックします。これによりパッケージが Content Store に発行され、TM1 でパッケージへアクセスできるようになります。
8. 最後の画面で結果を確認して、「**Finish**」をクリックします。

タスクの結果

TM1 でモデルを作成する準備が完了しました。パッケージ作成の詳細については、『*Framework Manager ユーザー・ガイド*』の「パッケージの作成または変更」を参照してください。

SAP BW クエリーにセグメント化プロンプトを指定する

セグメント化プロンプトは、ファクト・データを取得するために SAP BW データ・ソースに対してクエリーを実行するときに使用します。

プロンプトを指定すると、データを検索するためのより最適化された抽出のクエリーが IBM Cognos Package Connector によって実行されます。BEx 変数ともいわれるセグメント化プロンプトは、クエリーがファクト・データの代表サンプリングを確実に取得できるようにします。

セグメント化プロンプトには単一値、複数值、または範囲にすることができます。範囲を選択する場合は、包含形式で、範囲の最初と終了の値を含める必要があります。セグメント化プロンプトは任意にする必要があります、デフォルト値はありません。

複数プロンプト、または BEx 変数が許可されています。複数のプロンプトがある場合は、1 つをセグメント化プロンプトとして選択することができます。セグメント化プロンプトはいずれのクエリーでも値を指定されることはできません。セグメント化プロンプトとして指定されていない必須プロンプトでは値を指定する必要があります。セグメント化プロンプトとして指定されていない任意のプロンプトは、必要に応じて値を持ったり、持たなかったりします。

要確認: セグメント化プロンプトは、Cognos Framework Manager でタイプ pickList として示される必要があります。このタイプは大/小文字の区別があるので、小文字の p と大文字の L を使用しなければなりません。

手順

1. 階層と数値データを選択したら、「**Prompts**」をクリックします。
2. データをセグメント化するプロンプトリストで、ファクト・データをセグメント化するプロンプトを選択します。有効なプロンプトのみがリスト表示されます。

3. セグメント化プロンプトにデフォルト値が指定されていないことを確認します。
ヒント: プロンプトの値をクリアするには、現在のプロンプト値リストのプロンプトをクリックして、「値のクリア」をクリックします。
4. 現在のプロンプト値リストに表示されているすべての必須プロンプトに値が指定されていることを確認します。
5. 「OK」をクリックします。

分割プロンプトを使用して並行クエリーを実行する

SAP BW ファクト抽出を分割するプロンプトを定義した場合は、TM1 Package Connector に参照を設定してパフォーマンスを向上させることができます。

手順

1. TM1 Package Connector のインストール・フォルダーを検索して Configuration フォルダーに移動し、テキスト・エディターで **cogtr.xml** ファイルを開きます (XML アウェア・エディターを推奨)。
2. ファイルの "Transformer" セクションに次のエントリーを追加します。

```
<Preference Name="SegmenterParallelQueryCount" Value="16"/>
```
3. TM1 Package Connector が発行する並行クエリーの数値を設定します。デフォルトでは、一度に 1 つ発行します。
4. ファイルを保存します。

TM1 Package Connector は、指定する値に基づいて並行に数値データのクエリーを実行することができます。例えば、月に関する InfoQuery 任意の変数を定義し、特性に 48 カ月が含まれているとします。この変数を分割プロンプトとして数値データ・ディメンションに指定します。TM1 Package Connector は並行して最大 16 クエリーを実行することができるので (各月ごとに 1 つ)、ファクト・データの抽出速度を大幅に改善できます。

SAP 管理者に相談して、データ抽出のクエリーが他のユーザーの SAP BW 環境に悪影響を与えないようにしてください。

数値データ・クエリー実行時の無効なキー・エラー

分割プロンプトを使用して数値データ・クエリーを実行する際に「無効なキー」エラーが発生したら、SAP BW Info Query の各特性の表示形式設定が「キー」に設定されていることを確認してください。設定しない場合は、ディメンションの要素と数値クエリーに返されたキー値が一致しない場合があります。

第 7 章 詳細プロシージャの編集

このセクションでは IBM Cognos TM1 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 TM1 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がある管理者。
- 大容量のデータを読み込む夜間帯。

TM1 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、TM1 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ることができます。

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、TM1 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、TM1 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 TM1 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 TM1 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、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 TM1 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 TM1 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 TM1 規則関数が記載されたリストは、『IBM Cognos TM1 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、TM1 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 TM1 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ル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ルライズするために使用することができます。

TM1RunTI の実行可能ファイル (tm1runti.exe) は TM1 サーバー・インストールの bin ディレクトリーにあります。

デフォルトを受け入れた場合、tm1runti.exe のインストール先は C:\Program Files\IBM\cognos\tm1\bin にあります。

非同期呼び出しと TM1

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。Cognos TM1 のツールは非同期 (パラメーター 0) 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。これと同じアドバースが、ExecuteCommand によって呼び出される実行可能機能が Cognos TM1 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 Cognos TM1 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。

```
tmlrunTI -?  
or tmlrunTI -help  
or tmlrunTI [<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>  
-AdminSvrSSExportKeyId <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 Cognos TM1 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取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 Cognos TM1 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、複数コピーの保守に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	Cognos TM1 管理ホスト
sever	文字列/いいえ/なし	Cognos TM1 サーバー名
user	文字列/いいえ/なし	Cognos TM1 または CAM 名
AdminSvrSSLCertAuthority	文字列/いいえ/なし	Cognos TM1 管理サーバーの証明書を発行した認証局ファイルの絶対パス
AdminSvrSSLCertID	文字列/いいえ/なし。 API デフォルトは tmladminserver	Cognos TM1 管理サーバーの証明書の発行先であるプリンシパルの名前。 注: このパラメーターの値は、Tmladmsrv.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 Cognos TM1 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絶対パス。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	元々 Cognos TM1 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、AdminSvrSSExportKeyID 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。
AdminSvrSSExportKeyId	文字列/いいえ/なし	Cognos TM1 管理サーバーの証明書を元々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 Cognos TM1 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
<ti_parm>	値/必須かどうか/デフォルト 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	Cognos TM1 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、Cognos TM1 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、

- パラメーター間に空白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、

```
tm1run ti -i ".\tm1tools.config" -passwordkeyfile c:\keystore\prodkey.dat -connect prodssystem
```

-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 [tm1run ti] セクションをロードします
2. [tm1run ti] 内の接続パラメーターを検出すると、[Connect - testssystem] 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 - prodssystem] 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、その環境で複数の Cognos TM1 サーバー がサポートされている場合に特に便利です。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違う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。

TM1RunTI 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 Cognos TM1 TM1 Server 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 Cognos TM1 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("tmlrunTI -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、2 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 Cognos TM1 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 Cognos TM1 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 その他の Cognos TM1 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 プロセスのシリアルライズ

synchronized() という名前の IBM Cognos TM1 TurboIntegrator (TI) 関数を TurboIntegrator スクリプトで使用すると、指定した一連の TurboIntegrator プロセスを順次強制実行することができます。

Cognos TM1 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 プロセスをシリアル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ル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ル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・コードでシリアルライゼーションを明示的に起動する `synchronized()` を使用できます。

この関数の用法について、詳しくは「*IBM Cognos TM1 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 TM1 TurboIntegrator (TI) 関数を TurboIntegrator スクリプトで使用すると、指定した一連の TurboIntegrator プロセスを順次強制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ルライズするために、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 = ストリング

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勧めます (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 Cognos TM1 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 Cognos TM1 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、組み込み空白が無視されます。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せん。

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 TM1 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 TM1 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。TM1 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

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 TM1 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 TM1 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、TM1 の概念を十分に理解し、TM1 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、TM1 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「**TM1**」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

TM1 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
- TM1 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ピング
- 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、TM1 キューブ・ビュー、および TM1 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。この関数を使用して、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 TM1 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 TM1 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

TM1 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、TM1 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 TM1 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、TM1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。

変数 V14 および V19 の変数の型を「**文字列**」に変更します。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。

TM1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の保存をお勧めします。

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

チュートリアルは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. 「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、「終了」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連していることは、TM1 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	NAME
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。

TM1 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、TM1 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	String	6091400
15	V15	String	Total B
16	V16	String	E
17	V17	String	E 453326000000000
18	v18	String	D
19	V19	String	8
20	v20	String	lst

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、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、TM1 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	region	作成	数値

要素の変数	ディメンション	操作	要素の型
V19	agent	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。

TM1 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 TM1 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、TM1 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。

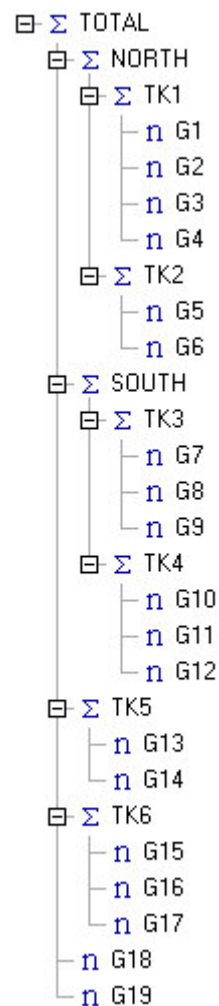
階層のバランスが取れていないディメンションの作成

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、TM1 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はありません。

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、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。

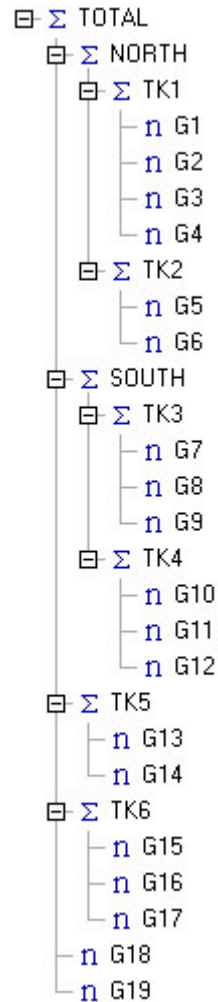
```
*****GENERATED STATEMENTS START*****  
*****GENERATED STATEMENTS FINISH*****  
IF (G1@<>');  
DIMENSIONELEMENTINSERT('unbalanced',' ',G1,'n');  
DIMENSIONELEMENTINSERT('unbalanced',' ',TOTAL,'c');  
DIMENSIONELEMENTINSERT('unbalanced',' ',NORTH,'c');  
DIMENSIONELEMENTINSERT('unbalanced',' ',TK1,'c');  
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、TM1 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	Scandania	Europe	International	Europe
Norway	Scandania	Europe	International	Europe
Denmark	Scandania	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、新しく作成されたサ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 TM1 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、TM1 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。

この記述は、IBM から他の言語で提供されている場合があります。しかし、それを参照するためには、その言語版の製品もしくは製品のコピーを所有していることが必要となる可能性があります。

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。

〒103-8510

東京都中央区日本橋箱崎町19番21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を提供し、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。その測定値が、一般に利用可能なシステムのも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認できません。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ます、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。

この「ソフトウェア・オフファリング」は、Cookie もしくはその他のテクノロジーを使用して個人情報を収集することはありません。

商標

IBM、IBM ロゴ、および [ibm.com](http://www.ibm.com) 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> 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。
- 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]

- 一括ロード・モード 55
- インポート処理のオプション 2
- エピローグ・プロシージャ 3
- エラー・メッセージ 65

[カ行]

- カスタム・クエリー 44
- 関数
 - TurboIntegrator プロセスでの使用 2
- キューブ
 - 作成 14
 - データ・ソース 15
 - 表示 23
 - 変数 16
 - マッピング 12, 16
 - ASCII フラット・ファイル 14
- クエリー
 - SQL 19
- 構文 59, 68
- 固定長レコード 9

[サ行]

- サインオン 37
- サンドボックス
 - TurboIntegrator 関数 6
 - TurboIntegrator プロセスでの実行 5
 - TurboIntegrator プロセスでの使用 5
- サンプルの ASCII ファイル 79
- 始動 76
- 集約
 - 複数 13
 - マッピング 13
- 重要なデータ 39
- 詳細な重要データ 47
- 詳細ファクト・クエリー・サブジェクト 47
- 数値データ
 - IBM Cognos 8 Planning 用 47
- 接続
 - MSAS 文字列 28
 - SAP パラメーター 37
- 設定 63

[タ行]

- チュートリアル
 - 概要 78
 - 作成 79, 86, 99, 101
 - データ・ディレクトリー 77
 - ディメンションの作成 79
 - プロセス内のプロシージャの編集 91
- データ
 - ソース 3, 7, 11, 15, 23, 24
 - プロシージャ 3
 - 変数 17
 - マッピング 13
 - ODBC ソース 19
- データのインポート
 - 概要 1
- データの抽出
 - 変数使用法の SAP BW ガイドライン 49
- データ・ソースのクエリー 84
- データ・ソースのパラメーター 85
- データ・ソース・パラメーター 85
- ディメンション
 - サブセット・データ・ソース 24
 - 集約変数のマッピング 17
 - データ変数のマッピング 17
 - マッピング 12
 - 要素変数のマッピング 16
 - ASCII フラット・ファイル 7
 - ODBO 34
- 同期化 67
- 登録されたサーバー 27
- 特性
 - マッピング 39

[ナ行]

- 日課 76
 - 自動実行 73
 - セットアップ・ウィザード 73
 - 定義 2, 73

[ハ行]

- パスワード 67
- 汎用命名規則
 - データ・ソース 7
- 非ディメンション・モデル化 44
- プロセス
 - 実行 14, 58
 - 定義 2
 - ヒント 3

プロセス (続き)
 プロシージャー 3
 編集 58
 保存 14, 35
 ODBO の実行 35
プロログ・プロシージャー 3
プロンプト
 SAP BW クエリーでセグメント化を指定 53
並行クエリー 54
変数
 キューブ 16
 キューブのマッピング 16
 データ・ソース 11
 ディメンションへの集約のマッピング 17
 ディメンションへのマッピング 16
 デフォルト名 11
 マッピング 12
 SAP BW ファクト・データのガイドライン 49

[マ行]

マッピング
 キューブ 12, 16
 キューブ変数 16
 集約 13
 データ 13
 ディメンション 12
 ディメンションへの集約変数 17
 ディメンション要素変数 16
 変数 12
メタデータ・プロシージャー 3
メッセージ・ログ 35
文字列 3
文字列長の制限 1

[ヤ行]

要素
 データ・ソースからのインポート 7
予約語
 暗黙的な変数の名前 109
 概要 103
 規則関数 103
 プロセス関数 106
 TurboIntegrator のキーワード 110

[ラ行]

レコード、固定長 9

A

ASCII
 サンプル・ファイル 79
 ファイル 1

ASCII (続き)
 フラット・ファイル 14

C

ChoreCommit 75

I

IBM Cognos 8 Planning と SAP BW データ 47

M

MDX 20
Microsoft Analysis Services 27, 32, 35
 キューブのインポート 29
 接続 29
 接続文字列 28
 ディメンションのインポート 32
MSAS
 接続文字列 28

N

Null 値 3

O

ODBC 1
 カタログ 27
 データ・ソース 19
 データ・ソースの定義 19
ODBO
 カタログ 27
 キューブ 30
 キューブの保存 32
 キューブ・ディメンション 30, 32
 データ・ソース 27
 ディメンション 34
 ディメンションの保存 34
 場所 27
 プロバイダー名 27
OLAP 27
OLE DB 27
OLE_LINK1 93

S

SAP
 階層 39
 重要なデータ 39
 接続パラメーター 37
 特性 39
 マッピング 39

SAP (続き)
 BW 37
 BW への接続の確立 37

SAP BW
 ファクト・データ使用のガイドライン 49

SAP BW クエリー
 セグメント化プロンプトを指定 53

SAP BW データ
 Cognos Planning 47

SQL クエリー 19

STET 3

synchronized() 68

T

TI 関数での別名 5

TI プロセス
 推奨事項 4

TM1RunTI 58, 59, 63, 65, 67

TurboIntegrator

 関数 2

 チュートリアル 77

 データのインポート 2

 プロセス 3

 予約語 103

 MDX からのインポート 21

 ODBC 19

TurboIntegrator プロセスのシリアライズ 67

U

UNC 7