

DB2[®] ユニバーサル・データベース



データウェアハウスセンター アプリケーション統合の手引き

バージョン 7

DB2[®] ユニバーサル・データベース



データウェアハウスセンター アプリケーション統合の手引き

バージョン 7

ご注意!

本書、および本書がサポートする製品をご使用になる前に、343ページの『特記事項』にある一般的な情報を必ずお読みください。

本書において、日本では発表されていない IBM 製品 (機械およびプログラム)、プログラミング、またはサービスについて言及または説明する場合があります。しかし、このことは、弊社がこのような IBM 製品、プログラミング、またはサービスを、日本で発表する意図があることを必ずしも示すものではありません。

本マニュアルに関するご意見やご感想は、次の URL からお送りください。今後の参考にさせていただきます。

<http://www.ibm.com/jp/manuals/main/mail.html>

なお、日本 IBM 発行のマニュアルはインターネット経由でもご購入いただけます。詳しくは

<http://www.ibm.com/jp/manuals/> の「ご注文について」をご覧ください。

(URL は、変更になる場合があります)

原典： SC26-9994-00
IBM[®] DB2[®] Universal Database
Data Warehouse Center Application Integration Guide
Version 7

発行： 日本アイ・ビー・エム株式会社

担当： ナショナル・ランゲージ・サポート

第1刷 2000.6

この文書では、平成明朝体™W3、平成明朝体™W9、平成角ゴシック体™W3、平成角ゴシック体™W5、および平成角ゴシック体™W7を使用しています。この(書体*)は、(財)日本規格協会と使用契約を締結し使用しているものです。フォントとして無断複製することは禁止されています。

注* 平成明朝体™W3、平成明朝体™W9、平成角ゴシック体™W3、
平成角ゴシック体™W5、平成角ゴシック体™W7

© Copyright International Business Machines Corporation 1998, 2000. All rights reserved.

Translation: © Copyright IBM Japan 2000

目次

本書について	vii
対象読者	vii

第1部 アプリケーションの統合 1

第1章 アプリケーション統合の計画 3

パートナー・アプリケーションがデータウェア ハウスセンターおよび情報カタログ・マネー ジャーと協働する方法	3
パートナー・アプリケーションの管理	5
メタデータの管理	6
統合のシナリオ	9
ハードウェアおよびソフトウェア要件	11

第2章 メタデータのインポートおよびエクス ポート 13

データウェアハウスセンターにメタデータを インポートする	13
タグ言語ファイルの構築	13
タグ言語ファイルからメタデータをインポ ートする	36
ステップを実行に備えて準備する	37
データウェアハウスセンターからのメタデー タのエクスポート	38
メタデータをエクスポートするオブジェク トの選択	38
タグ言語ファイルへのメタデータのエクス ポート	39

第3章 情報カタログ・マネージャーによるメ タデータのインポートおよびエクスポート . . 41

情報カタログにメタデータをインポートする インポートするメタデータの選択	41
タグ言語ファイルからのメタデータのイン ポート	41
情報カタログ・マネージャーからのメタデー タのエクスポート	47
エクスポートするメタデータの選択	48
タグ言語ファイルのエクスポート	48

第4章 ユーザーが情報カタログ・マネージャ ーからプログラムを開始できることを確認す る	53
情報カタログ・マネージャー (Web 版) ユー ザーのための追加要件	53

第2部 メタデータのリファレンス 57

第5章 メタデータ・テンプレート 59

AgentSite.tag	61
トークン	62
値の例	64
Column.tag	64
トークン	64
値の例	69
HeaderInfo.tag	70
トークン	70
値の例	71
Process.tag	71
トークン	71
値の例	72
StarSchema.tag	73
トークン	73
値の例	74
StarSchemaInputTable.tag	74
トークン	74
値の例	75
Step.tag	76
トークン	76
値の例	79
StepCascade.tag	80
トークン	80
値の例	81
StepInputTable.tag	81
トークン	81
値の例	82
StepOutputTable.tag	83
トークン	83
値の例	83
StepVWOutputTable.tag	84
トークン	84

ACTION.RELATION	253
コンテキスト	253
構文	253
オプション	253
COMMENT	255
構文	255
規則	255
COMMIT	255
コンテキスト	256
構文	256
キーワード	256
規則	257
DISKCNTRL	257
コンテキスト	257
構文	257
キーワード	257
規則	258
INSTANCE	258
コンテキスト	258
構文	258
NL	264
構文	264
規則	264
OBJECT	264
コンテキスト	265
構文	265
PROPERTY	271
構文	271
コンテキスト	271
キーワード	272
規則	274
RELTYPE	275
構文	275
コンテキスト	275
キーワード	275
TAB	277
構文	277
規則	277
第10章 タグ言語ファイルの使い方	279
タグ言語ファイルの DISKCNTRL からの開始	279
追加、変更、および削除の定義	279
行うことの定義	279
情報の定義	280
タグの組み立て	280
データベースへの変更のコミット	281

タグ言語ファイルへのコメントの書き込み	282
-------------------------------	-----

第3部 提供されているプログラムおよびマクロのリファレンス 283

第11章 提供されているデータウェアハウスセンター プログラム	285
VWPEXUNX	285
パラメーター	285
戻りコード	287
ログ・ファイル	288
ISV_Sample	288

第12章 Net.Data® マクロ	291
情報カタログ・マネージャー (Web 版) のファイル	291

第4部 付録および後付け 297

付録A. テンプレート・プランのワークシート	299
----------------------------------	-----

付録B. Visual Warehouse 5.2 によってサポートされるテンプレート	313
---	-----

BusinessView.tag	314
トークン	314
値の例	320
BusinessViewInputTable.tag	321
トークン	321
値の例	323
BusinessViewOutputTable.tag	323
トークン	324
値の例	325
BusinessViewVWPOutputTable.tag	325
トークン	325
値の例	326
ConcurrentCascade.tag	327
トークン	327
値の例	327
PostCascade.tag	328
トークン	328
値の例	329
VWPPProgramInstance.tag	329
トークン	329
値の例	331

付録C. データウェアハウスセンターで使用 するプログラムを独自に作成する	333	特記事項	343
パラメーターを渡す	334	商標	346
状況情報を戻す	335	参考文献	349
情報をデータウェアハウスセンターに転送 する	336	索引	351
フィードバック・ファイルの形式	336	IBM と連絡をとる	361
フィードバックによってステップ状況を判 別する方法	338	製品情報	361

本書について

本書は、データウェアハウジング・ソリューションの開発者が、アプリケーションをデータウェアハウスセンターおよび情報カタログ・マネージャーと統合するために役立つように作成されました。本書を使用して、アプリケーションのメタデータを転送してデータウェアハウスセンターおよび情報カタログ・マネージャーが使用可能な形式に変換するためのプログラムを記述することができます。さらに本書に記載された情報を使用して、情報カタログ・マネージャーの形式を調整することもできます。

対象読者

本書は、他社のデータウェアハウジング・アプリケーションとデータウェアハウスセンターまたは情報カタログ・マネージャーもしくはその両方との間の自動化されたインターフェースを作成する、データウェアハウジング・ソリューションの開発者を対象としています。

情報処理サポートの分野での経験がいくらあっても、ときには企業内の他のサポート担当者の支援が必要なことがあります。本書で解説される統合機能を使用する前に、データウェアハウスセンターおよび情報カタログ・マネージャーについて知っていなければなりません。特に、以下の表にリストされているタスクの実行方法を知っている必要があります。

タスク	参照資料
情報カタログの作成	情報カタログ・マネージャー 管理の手引き
メタデータのインポートおよびエクスポート	情報カタログ・マネージャー 管理の手引き
ウェアハウス・エージェント・サイトの定義	データウェアハウスセンター 管理の手引き およびデータウェアハウスセンターのオンライン・ヘルプ
ステップの作成、プロモート、実行、およびモニター	データウェアハウスセンター 管理の手引き およびデータウェアハウスセンターのオンライン・ヘルプ
データウェアハウスセンター プログラムを作成して、それらをステップで使用する	データウェアハウスセンター 管理の手引き およびデータウェアハウスセンターのオンライン・ヘルプ

タスク	参照資料
データウェアハウスセンター プログラム のパラメーターの変更	データウェアハウスセンター 管理の手引 き およびデータウェアハウスセンターの オンライン・ヘルプ
メタデータのインポートおよびエクスポート	データウェアハウスセンター 管理の手引 き およびデータウェアハウスセンターの オンライン・ヘルプ

データウェアハウスセンターおよび情報カタログ・マネージャーに関する資料のリストは、349ページの『参考文献』をご覧ください。

第1部 アプリケーションの統合

第1章 アプリケーション統合の計画

DB2 ユニバーサル・データベース™・データウェアハウスセンターを使用して、ユーザーがデータウェアハウスを構築および管理するために役立つ種々のアプリケーションを組み合わせることができます。管理したいデータを識別して、そのデータをウェアハウスのユーザーに意味のある情報に変換することができます。

データウェアハウスセンターを使用して、他のデータウェアハウジング・アプリケーションのために種々の情報およびサービスを提供することができます。以下の事柄が含まれます。

- ウェアハウスで使用されるソース・データおよびターゲット・データについてのメタデータの提供。
- SQL の発行または他のウェアハウジング・アプリケーションの実行によるデータの変換。
- 日付またはイベントに基づくデータの抽出および変換のスケジューリング。
- ウェアハウスのユーザーが使用するためのメタデータの公表。

ご使用のアプリケーションをデータウェアハウスセンターに統合すると、ウェアハウス管理者のために単一の制御点が備えられ、また管理者が最良のウェアハウジング・アプリケーションを使用できるようになります。

パートナー・アプリケーションがデータウェアハウスセンターおよび情報カタログ・マネージャーと協働する方法

本書において、パートナー・アプリケーションとはデータウェアハウスセンターから独立して実行するアプリケーションで、データウェアハウジング・ソリューションのために何かのサポートを提供するもののことです。アプリケーションをデータウェアハウスセンターに定義して、そのアプリケーションが複数アプリケーションを含むことのできるウェアハウス構築プロセスに含まれるようにすることができます。

たとえば、IMS™ データベースから操作データをアンロードして、そのデータをクレンジングしてから、クレンジング済みデータを DB2® ウェアハウス・データベースにロードすることができます。その後ユーザーはクレンジング済みデータを照会することができます。次の 3 つのパートナー・アプリケーションがあります。

- パートナー・アプリケーション 1 は、データベースからデータをダウンロードして、表の結合などの簡単な変換を実行してから、変換済みデータをウェアハウス・データベースに書き込みます。
- パートナー・アプリケーション 2 は、データをクレンジングして、そのデータをウェアハウス用に準備します。
- パートナー・アプリケーション 3 は、ウェアハウス内のデータを照会してレポートします。それには、ユーザーが特定の属性について検索できるウェアハウス内の表に関するメタデータが含まれます。ユーザーはメタデータを使用して、必要なデータがどの表に含まれるかを判別します。

以下のプロセスでは、これら 3 つのアプリケーションを共に使用します。

1. パートナー・アプリケーション 1 は、ソース IMS データベース内の複数のセグメントからデータを抽出します。
2. パートナー・アプリケーション 1 は、ソース・セグメントからのデータを結合して、結合済みデータをファイル 1 に書き込みます。
3. パートナー・アプリケーション 1 は、結合済みデータをファイル 1 に書き込みます。
4. パートナー・アプリケーション 2 は、ファイル 1 からデータを読み取ります。
5. パートナー・アプリケーション 2 は、名前の突き合わせおよび他のデータ・クレンジング技法を使用して、データをクレンジングします。
6. パートナー・アプリケーション 2 は、クレンジング済みデータをファイル 2 に書き込みます。
7. パートナー・アプリケーション 1 は、ファイル 2 からデータを読み取ります。
8. パートナー・アプリケーション 1 は、データをウェアハウス・データベースに書き込みます。
9. パートナー・アプリケーション 3 は、ユーザーが照会する表を選択したとき、ウェアハウス内のデータを表示するか、またはウェアハウス内のデータをレポートします。

5ページの図1 は、3 つのパートナー・アプリケーションが協働する方法を示しています。

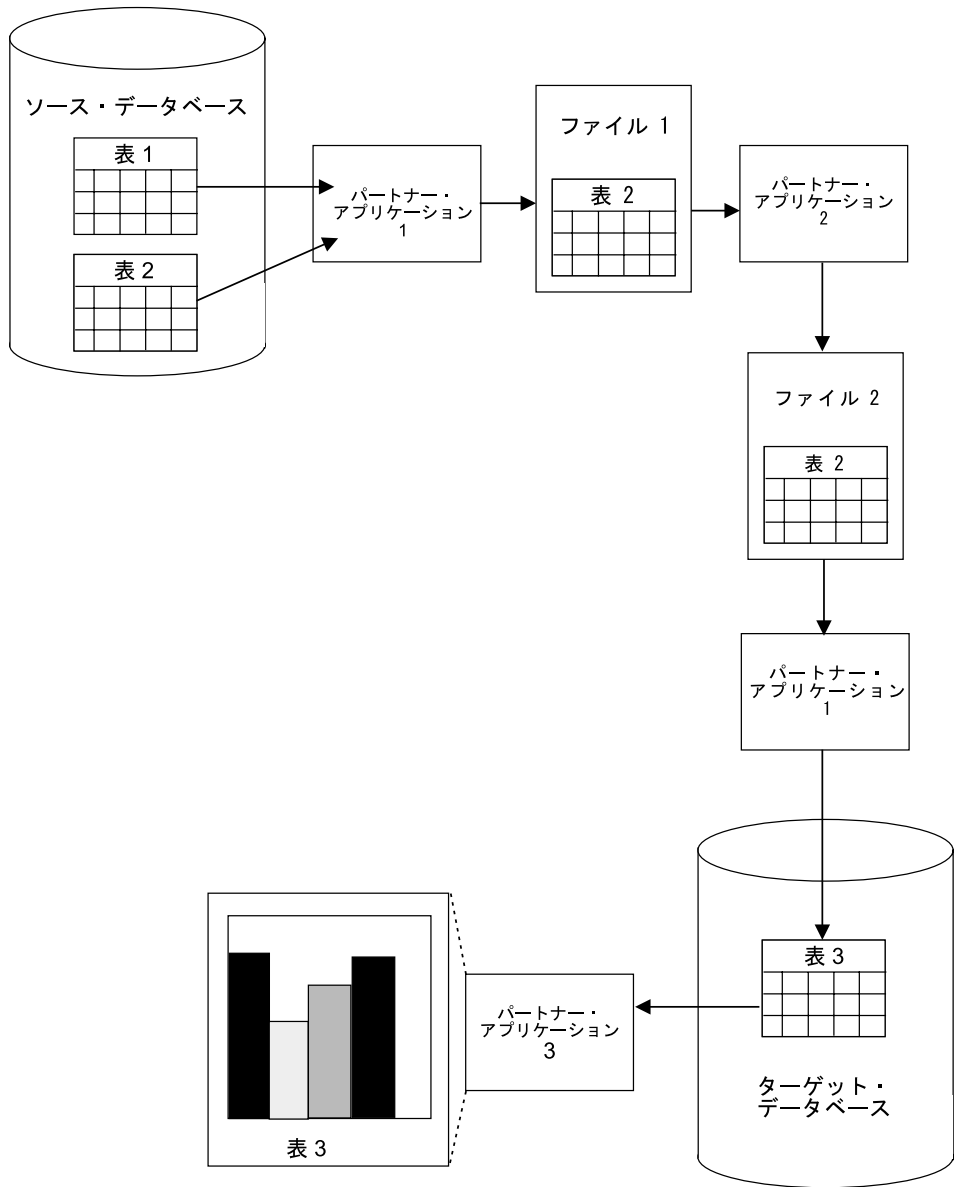


図1. パートナー・アプリケーションを共に使用してウェアハウスを構築する

パートナー・アプリケーションの管理

データウェアハウスセンターのステップを使用して、プロセスを管理することができます。ステップとは、ウェアハウス・プロセスでデータに対する単一の

操作のことです。ほとんどの場合、ステップにはウェアハウス・ソース、変換またはデータの移動、およびウェアハウス・ターゲットが含まれます。ステップはスケジュールに従って実行したり、他のステップからの連鎖で実行することができます。ステップを使用して、データの抽出、変換、および書き込みにおける各ステップを定義およびスケジュールリングします。

基本ステップは以下のタスクを実行します。

- 少なくとも 1 つの表またはファイルからデータを抽出します。
- データウェアハウスセンター SQL 処理を使用してデータを変換するか、またはデータを変換するプログラムを呼び出します。
- 変換済みデータを表に書き込みます。

パートナー・アプリケーションの例では、以下のソースからターゲットへの変換のための、3 つのステップを定義します。

- アンロード・ステップは、タスク 1 から 3 を実行します。
- クレンジング・ステップは、タスク 4 から 6 を実行します。
- ロード・ステップは、タスク 7 から 8 を実行します。

パートナー・アプリケーション 3 はタスク 9 でユーザーの要求に応じてデータを変換するので、タスク 9 のためにはステップを定義しません。

ステップの定義で、ステップを実行する日付および時刻をスケジュールリングできます。その日時に、データウェアハウスセンターは SQL ステートメントを発行したりプログラムを開始することによって、ステップが定義するプロセスを開始します。さらに、2 番目のステップが最初のステップの処理終了後に開始するように指定することもできます。

最初のステップが特定の日時に実行するようにスケジュールリングできます。2 番目のステップは最初のステップの実行後に開始するようにスケジュールリングできます。3 番目のステップは 2 番目のステップの実行後に開始するようにスケジュールリングできます。このようにして、複数のパートナー・アプリケーションの実行プロセスを自動化することができます。

メタデータの管理

このプロセスを定義するためには、パートナー・メタデータをデータウェアハウスセンターにインポートします。本書においてパートナー・メタデータとは、パートナー・アプリケーションが使用してデータウェアハウスセンターの外側に保管するメタデータのことです。

パートナー・アプリケーションの例では、以下のメタデータをデータウェアハウスセンターにインポートします。

- パートナー・アプリケーション 1 から、データベース、ファイル 1、およびアプリケーションに関するメタデータ
- パートナー・アプリケーション 2 から、ファイル 2 およびアプリケーションに関するメタデータ

その後、ファイルに関するメタデータをパートナー・アプリケーションにエクスポートして、両方のパートナー・アプリケーションが同じ情報を使用するようにします。

- ファイル 2 に関するメタデータをパートナー・アプリケーション 1 にエクスポートします。
- ファイル 1 に関するメタデータをパートナー・アプリケーション 2 にエクスポートします。

データウェアハウスセンターから情報カタログ・マネージャーにメタデータをエクスポートして、ウェアハウスのユーザーのためにウェアハウス内のデータに関する情報を示すことができます。ソースとターゲットについてのメタデータ、およびソース形式からターゲット形式へのデータ変換についてのメタデータをインポートすることができます。ウェアハウスのユーザーはインポートされたメタデータによって、ウェアハウス内のデータの系統に関する情報を取得することができます。

パートナー・アプリケーションの例では、ウェアハウス内の表である表 3 に関するメタデータを情報カタログ・マネージャーにエクスポートします。

メタデータを情報カタログ・マネージャーにデータウェアハウスセンターから直接インポートすることができます。さらに、パートナー・アプリケーションが MDIS 形式のメタデータをサポートする場合、メタデータを情報カタログ・マネージャーにインポートすることができます。

メタデータを情報カタログ・マネージャーからパートナー・メタデータ・ストアにエクスポートすることができます。パートナー・アプリケーションの例では、表 3 についてのメタデータを情報カタログ・マネージャーからパートナー・アプリケーション 3 のメタデータ・ストアにエクスポートします。ユーザーは表 3 についてのメタデータを表示して、その内容を検討します。

図2 は、パートナー・アプリケーション、データウェアハウスセンター、および情報カタログ・マネージャーでのメタデータの流れを示しています。

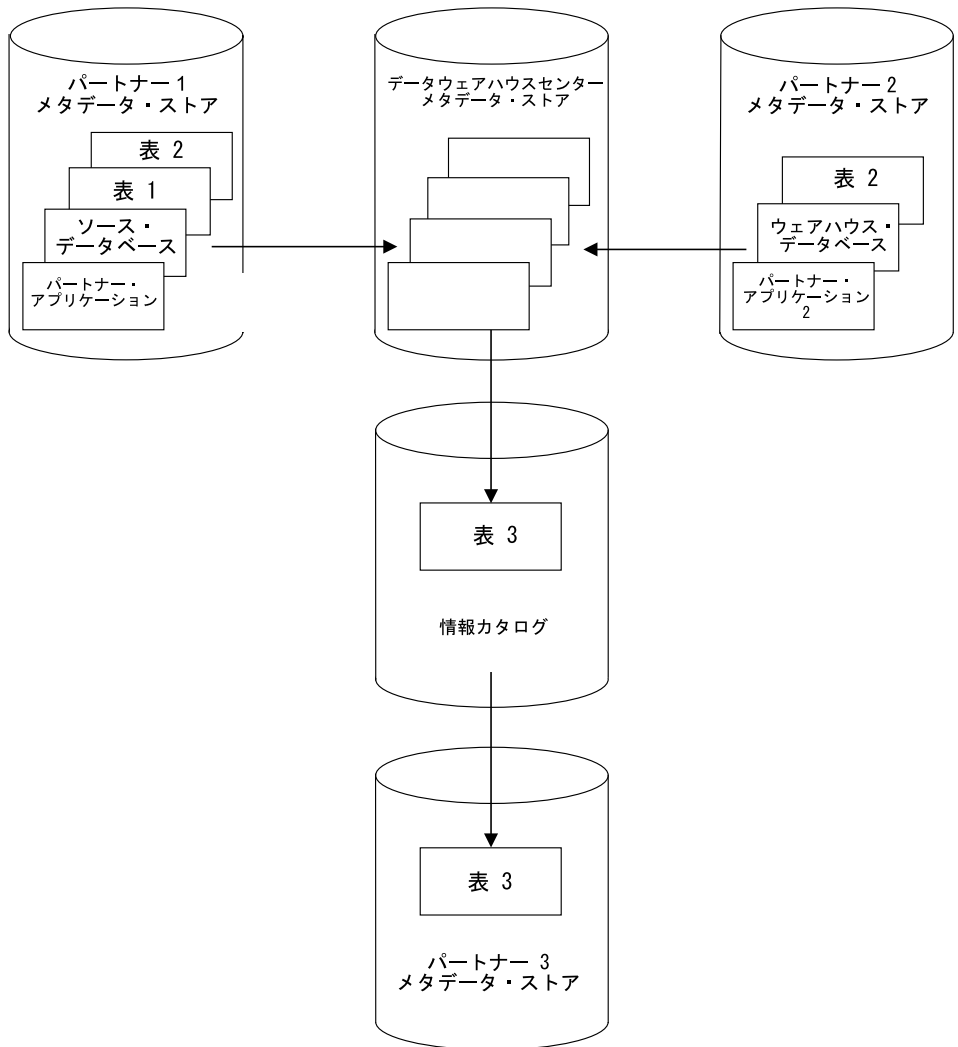


図2. パートナー・アプリケーション、データウェアハウスセンター、および情報カタログ・マネージャーでのメタデータの流れ

本書の残りの部分では、以下のトピックに関してより詳細に解説しています。

- メタデータをデータウェアハウスセンターにインポートすることについての詳細は、13ページの『データウェアハウスセンターにメタデータをインポートする』を参照してください。

- メタデータをデータウェアハウスセンターからエクスポートすることについての詳細は、38ページの『データウェアハウスセンターからのメタデータのエクスポート』を参照してください。
- メタデータを情報カタログ・マネージャーにインポートすることについての詳細は、41ページの『情報カタログにメタデータをインポートする』を参照してください。
- メタデータを情報カタログ・マネージャーからエクスポートすることについての詳細は、47ページの『情報カタログ・マネージャーからのメタデータのエクスポート』を参照してください。

統合のシナリオ

表1 は、いくつかの一般的なタイプのウェアハウス・アプリケーションをリストして、それらをデータウェアハウスセンターに統合する方法を解説しています。

表1. 統合のシナリオ

アプリケーションのタイプ	統合プロセス
データウェアハウジング設計	<p>データウェアハウジング設計アプリケーションからのデータをデータウェアハウスセンターで使用するには、</p> <ol style="list-style-type: none"> 1. メタデータをデータウェアハウスセンターにインポートします。 2. メタデータ同期を使用して、メタデータを情報カタログ・マネージャーに伝達します。
操作データ記述	<p>メタデータをデータウェアハウスセンターに、そしてビジネス・メタデータを情報カタログ・マネージャーにインポートします。</p> <p>メタデータがシステムのためだけに含まれているソース・データのもので、ソース表またはファイルを定義するためのものではない場合、メタデータを情報カタログ・マネージャーに直接インポートしてください。</p>

表 1. 統合のシナリオ (続き)

アプリケーションの タイプ	統合プロセス
データ・クレンジング	<p>操作データをクレンジングするには、</p> <ol style="list-style-type: none"> 1. ソース・データとターゲット・データの動きを管理するのは、データウェアハウスセンターまたはパートナー・アプリケーションのどちらであるかを判別します。 異なるアプリケーションが、ソース・データおよびターゲット・データをそれぞれ管理することが可能です。 2. ソースおよびターゲット定義をインポートするか、またはソースおよびターゲット定義をエクスポートします。 定義を再度入力しなくてもよいように、そのように行ってください。 3. パートナー・アプリケーションをデータウェアハウスセンター プログラムとして定義するか、またはパートナー・アプリケーションを開始するデータウェアハウスセンター プログラムを作成します。 4. パートナー・アプリケーションのパラメーターを設定するユーザー・インターフェースを開発します。 5. メタデータをデータウェアハウスセンターにインポートして、データウェアハウスセンターがデータ・クレンジング・アプリケーションを実行できるようにします。 プログラムのスケジューリングは、順番で行うことも、日付と時刻によって行うことも可能です。 6. ビジネス・メタデータを情報カタログ・マネージャーにインポートして、ユーザーが使用できるようにします。

表 1. 統合のシナリオ (続き)

アプリケーションの タイプ	統合プロセス
代替データ記憶装置 (DB2 OLAP Server™ など)	<p>操作データを代替データ記憶装置にロードするには、</p> <ol style="list-style-type: none"> 1. データウェアハウスセンターから、パートナー記憶装置の構築に必要なデータ定義をエクスポートします。 2. ロード・プログラムをデータウェアハウスセンター プログラムとして定義するか、またはロード・プログラムを開始するデータウェアハウスセンター プログラムを作成します。 3. パートナー・アプリケーションのパラメーターを設定するユーザー・インターフェースを開発します。 4. ロード・プログラムの定義をデータウェアハウスセンターにインポートします。 <p>ロード・プログラムを使用して、操作データ・ストア内の値とパートナー・データ・ストア内の値とを同期させます。</p> <ol style="list-style-type: none"> 5. パートナー・データ・ストアのビジネス・メタデータを情報カタログ・マネージャーにインポートします。
レポート作成 (Brio や Cognos など)	<p>レポート作成アプリケーションをデータウェアハウスセンターに統合するには、</p> <ol style="list-style-type: none"> 1. ビジネス・メタデータを情報カタログ・マネージャーからレポート作成アプリケーションにエクスポートします。 2. レポートについての記述を情報カタログ・マネージャーにインポートします。 3. レポート作成アプリケーションを情報カタログから開始できるようにします。

ハードウェアおよびソフトウェア要件

本書で説明されているモデルおよびテンプレートには、DB2 ユニバーサル・データベースのパッケージに備わっているデータウェアハウスセンターのバージョン 7.1、ウェアハウス・マネージャーのパッケージに備わっている情報カタログ・マネージャー・アドミニストレーターのバージョン 7.1、およびそれらの前提条件となる製品が必要です。

データウェアハウスセンターおよび情報カタログ・マネージャーの前提条件となる製品については、ご使用のプラットフォームの概説およびインストール、および *DB2* ウェアハウス・マネージャー インストールの手引き を参照してください。

第2章 メタデータのインポートおよびエクスポート

この章では、データウェアハウスセンターにメタデータを直接インポートする方法、およびそこからメタデータを直接エクスポートする方法について詳しく解説します。

データウェアハウスセンターにメタデータをインポートする

メタデータをデータウェアハウスセンターにインポートすると、データウェアハウスセンターはウェアハウス用にデータを抽出および変換したり、データを抽出および変換するパートナー・アプリケーションを実行することができます。

メタデータをデータウェアハウスセンターにインポートすることには、以下のタスクが関係します。

1. タグ言語ファイル (インポートするオブジェクトのメタデータを含むファイル) の作成。
2. タグ言語ファイルのインポート。
3. データウェアハウスを実行するステップの準備。

タグ言語ファイルの構築

タグ言語ファイルを構築するには、以下のように行います。

1. メタデータをインポートするオブジェクトを選択します。
2. データウェアハウスセンター メタデータ・テンプレートを使用して、各オブジェクトのメタデータを定義します。データウェアハウスセンター メタデータ・テンプレート は、パートナー・メタデータ値を表すトークンを含むタグ言語ファイルのサブセットです。ご使用のプログラムは、タグ言語ファイルの構文を参照することなく、トークンを検索して値を置換することができます。

メタデータをインポートするオブジェクトの選択

以下のタイプのオブジェクトについて、メタデータをデータウェアハウスセンターにインポートできます。

エージェント・サイト

ウェアハウス・エージェント は、ソース・データベースまたはファイル (ウェアハウス・ソース) と、ターゲット・データベース (ウェアハ

ウス・ターゲット) との間の実際のデータ転送を実行します。さらに、そのデータの変換も実行します。ウェアハウス・エージェントは、ウェアハウス・サーバーからコマンドを受け取ります。その後、エージェントは SQL コマンドを発行し、パートナー・アプリケーションを開始するか、またはパートナー・アプリケーションを開始するデータウェアハウスセンター プログラムを開始します。さらに、ウェアハウス・エージェントは、表定義をインポートします。

エージェント・サイト は、エージェントが実行するマシンです。エージェント・サイトには、ソース・データベースおよびターゲット・データベースを含むマシンにアクセス可能でなければなりません。

ウェアハウス・ソースおよびウェアハウス・ターゲット

ソース・データベース またはソース・ファイル は、データウェアハウスセンターまたはパートナー・アプリケーションがデータを抽出して処理を続けるための、データベースまたはファイルです。一般用語のソース は、データベースまたは 1 つ以上のファイルのグループを意味します。ソースは、1 つ以上の表、ファイル、またはセグメントに関連しています。表、ファイル、またはセグメントは、1 つ以上の列またはフィールドに関連しています。ウェアハウス・ソースは、データウェアハウスセンターに定義されている単一のデータベース、またはファイルのセットからの、表およびビューのサブセットです。

ウェアハウス・ターゲット またはターゲット・ファイル は、データウェアハウスセンターまたはパートナー・アプリケーションが処理した後にデータを書き込むデータベースまたはファイルです。一般用語のターゲット は、データベースまたは 1 つ以上のファイルのグループを意味します。ターゲットは、1 つ以上の表またはファイルと関連しています。表またはファイルは、1 つ以上の列またはフィールドと関連しています。ウェアハウス・ターゲットは、データウェアハウスセンターによって管理される表またはファイルのセットのサブセットです。

ウェアハウス・ターゲットは、ユーザーが照会およびレポート作成を実行するために使用するウェアハウスを含むデータベースです。

データウェアハウスセンター プログラム

データウェアハウスセンター プログラム は、ある種のデータ形式変更を実行するユーザー作成またはパートナー・アプリケーションです。プログラムをデータウェアハウスセンターに定義して、その実行をスケジューリングし、ステップの一部としてその操作をモニターすることができます。データウェアハウスセンター プログラムは、通常 1 つ以上のパラメーターに関連しています。関連したデータウェアハウスセンター

プログラムをデータウェアハウスセンター プログラム・グループに関連付けることにより、それらをグループ化することができます。

サブジェクト・エリア

サブジェクト・エリア を使用して、特定のトピックまたは機能に関連しプロセス（およびプロセス内のステップ、ウェアハウス・ソース、そしてウェアハウス・ターゲット）を論理的にグループ化することができます。たとえば、販売データを移動および変換する一連のプロセスがある場合、Sales サブジェクト・エリアを作成して、それらのプロセスをそのサブジェクト・エリア内に作成することができます。同様に、複数の Marketing プロセスは Marketing サブジェクト・エリアの下にグループ化できます。

プロセス

プロセスは、通常ソース・データに対して操作を行う一連のステップで、データを元の形式から意思決定に役立つ形式に変換します。データウェアハウスセンターのプロセスは、通常 1 つ以上のウェアハウス・ソース、1 つ以上のステップ、および 1 つ以上のウェアハウス・ターゲットから構成されます。

ステップ

ステップとは、データウェアハウスセンターのプロセスでデータに対する単一の操作のことです。プロセスは、通常 1 つ以上のウェアハウス・ソース、1 つ以上のステップ、および 1 つ以上のウェアハウス・ターゲットから構成されます。ほとんどの場合、ステップにはウェアハウス・ソース、データの変換または移動に関する記述、およびウェアハウス・ターゲットが含まれます。ステップを使用して、データの抽出、変換、および書き込みにおける各ステップを定義およびスケジューリングします。ステップのメタデータには、データウェアハウスセンターまたはパートナー・アプリケーションが操作を行うソースおよびターゲット表が含まれます。さらに、発行する SQL ステートメント、または変換の実行を開始するプログラムも含まれます。

ステップ間の連鎖関係

連鎖関係 は、他のステップの処理状況に基づくステップのスケジュールです。ステップが、別のステップの実行の終了後に実行するようにスケジューリングすることができます。

データウェアハウスセンター オブジェクト間の関係

データウェアハウスセンター オブジェクトのメタデータは、他のオブジェクトとの関係を記述しています。たとえば、ステップのメタデータは、ステップが使用するウェアハウス・ソースおよびウェアハウス・ターゲット表との関係を記述しています。

データウェアハウスセンター メタデータのテンプレートを使用してオブジェクトの定義する

データウェアハウスセンターにインポートしたいオブジェクトを定義するためには、1 つ以上のデータウェアハウスセンター メタデータ・テンプレートからタグ言語ファイルを構築します。

各テンプレートは、表などのオブジェクト、または列などのオブジェクトのサブセットに対応します。複数のテンプレートを結合して、オブジェクトに関するすべての詳細を定義します。たとえば、ソース・データベースを定義したい場合、データベース、表、および列のテンプレートを結合します。

パートナー・メタデータ・ストアから値を取得するプログラムを作成して、それらの値によってテンプレート内のトークンを置換します。本書では、この種類のプログラムを交換プログラムと呼びます。

各テンプレートには、交換プログラムが値を指定しなければならないトークンが含まれます。たとえば、トークン `*TableDescription` は表の記述を表します。交換プログラムは `*TableDescription` を検索して、それをリレーショナル・カタログで指定された表の記述を含むストリングに置き換えます。DB2™ ユニバーサル・データベースの表では、システム・カタログの `syscat.tables` 表の `REMARKS` フィールド内にその記述があります。交換プログラムがトークンを値で置き換えるので、ファイル内のメタデータを識別する基礎となるタグ言語の構文を知る必要はありません。

メタデータ・テンプレートのインストール: アプリケーション開発クライアントのインストール時に、テンプレートのインストールを選択できます。

テンプレートをインストールするには、以下のように行います。

1. 「セットアップのタイプ (Setup Type)」ウィンドウで、「**カスタム (Custom)**」を選択します。
2. 「データウェアハウス ISV ツールキット (**Data Warehouse ISV Toolkit**)」をクリックします。
3. テンプレートのためのディレクトリーを選択します。

ISV ツールキットのデフォルトのディレクトリーは、`x:\¥sqllib¥templates` です。データウェアハウスセンターは、`VWS_TEMPLATES` 環境変数を ISV ツールキットの位置に設定します。ご使用のプログラムは、`VWS_TEMPLATES` の値を照会して、テンプレートの位置を見つけることができます。

データウェアハウスセンターは、`VWS_TEMPLATES` が設定したディレクトリーのサブディレクトリーにファイルをインストールします。17ページの表2 は、

インストールされるファイルのタイプ、およびそれらのファイルがインストールされるサブディレクトリーをリストしています。

表2. テンプレートのファイル・タイプおよびサブディレクトリー

ファイルのタイプ	サブディレクトリー
テンプレート	ISV
サンプル	Samples
ヘッダー・ファイル	Include

交換プログラムの作成: 交換プログラムを作成するとき、以下を行う必要があります。

- ヘッダー・ファイルを組み込む。
- 適切なテンプレートをコピーして変更する。
- テンプレートの各コピーにチェックポイントを設定する。
- テンプレートの変更済みコピーをタグ言語ファイルに追加する。

さらに、データウェアハウスセンターが処理メッセージのログに使用するディレクトリーと同じディレクトリーに処理メッセージのログを記録することができます。

ISV_defines.h ヘッダー・ファイルを組み込む: ISV_Defines.h ヘッダー・ファイルを使用すると、テンプレートのトークンが変更してもプログラム・ロジックを同じに保つことができます。その場合に必要なのは、プログラムを再コンパイルすることだけです。

テンプレートをコピーして変更する: テンプレートを使用して作業するために、作成するプログラムは以下の手順を使用しなければなりません。

1. `VWS_TEMPLATES` 環境変数を使用して、テンプレートが保管されているディレクトリーを取得します。その値に `¥ISV¥` を追加して、テンプレートの完全パスを取得します。
2. テンプレートのコピーをプログラム内にローカルに読み取ります。
3. テンプレートを検索してテンプレート内のトークンを見つけ、それらのトークンをパートナー・アプリケーションからのメタデータで置き換えます。

タグ言語ファイルの形式にプログラミングする代わりに、検索と置換の手法を使用します。トークンを使用することにより、テンプレート・ファイル内で使用されているタグ言語の変更からプログラムが独立していることとなります。

テンプレート内では、各トークンは括弧で囲まれています。右小括弧は値の終了を示します。作成するプログラムがトークンだけを値に置換して、括弧を除去しないようにしてください。

トークンの値を置換するストリングは、以下の規則に従っていなければなりません。

- ストリングに組み込みタブ文字を含めることはできません。
- ストリング内に括弧がある場合、それを単一引用符で囲みます。

たとえば、*DatabaseNotes トークンを以下の値で置き換えると仮定します。

```
This is my database (managed by the Finance group).
```

この値を、以下のように変更しなければなりません。

```
This is my database '('managed by the Finance group)'
```

交換プログラムにトークンの値がない場合、そのトークンを定数 ISV_DEFAULTVALUE (ISV_defines.h 内で定義されている) に置き換えるようにします。ただし、必須のトークンには ISV_DEFAULTVALUE 以外の値を指定しなければなりません。

セキュリティー・グループのテンプレートは存在しないので、作成するプログラムでは値 ISV_DEFAULTSECURITYGROUP を *SecurityGroup トークンのインスタンスに指定しなければなりません。

テンプレートは、データウェアハウスセンターに特定のメタデータについてはデフォルト値を使用します。たとえば、ウェアハウス・ソースおよびウェアハウス・ターゲットの再試行回数および再試行間隔は、データウェアハウスセンターのデフォルト値に設定されます。

チェックポイントを設定する: 各テンプレートには、タグ言語ファイルをインポートするときの処理の追跡に使用できる *CurrentCheckPointID++ トークンを含んでいます。作成するプログラムがトークンの値を設定するとき、*CurrentCheckPointID++ の最初の出現を 0 に設定するようにしてください。プログラムは、*CurrentCheckPointID++ が出現するたびにその値を 1 ずつ増加させるようにします。データウェアハウスセンターはタグ言語ファイルのインポート時にこれらのチェックポイントをログ・ファイルに書き込みます。

テンプレートをタグ言語ファイルに追加する: 表3、表4、および表5は、プログラムがテンプレートをタグ言語ファイルに追加しなければならない順序をリストしています。それらの表には、テンプレートが必須または任意選択となるときの条件が示されています。

ヘッダーを除き、それぞれのテンプレートについて必要な数のコピーを定義することができます。各タグ言語ファイルで定義するヘッダーのコピーは 1 つだけです。

表3. テンプレートの相互関係および条件

順序	テンプレート	必須または任意選択
1	HeaderInfo.tag	常に必須
2	AgentSite.tag	デフォルトのエージェント・サイトを使用しない場合は必須
3	VWPGroup.tag	データウェアハウスセンター プログラムを定義している場合は必須
4	VWPProgramTemplate.tag	データウェアハウスセンター プログラムを定義している場合は必須
5	VWPProgramTemplateParameter.tag	データウェアハウスセンター プログラムを定義している場合は必須
6	SourceDataBase.tag WarehouseDataBase.tag	ウェアハウス・ソースまたはウェアハウス・ターゲットを定義している場合は必須
7	Table.tag	ウェアハウス・ソースまたはウェアハウス・ターゲットを定義している場合は必須
8	Column.tag	ウェアハウス・ソースまたはウェアハウス・ターゲットを定義している場合は必須

Column.tag テンプレートをタグ言語ファイルに追加した後、一連のテンプレート、およびテンプレートをタグ言語ファイルに追加する順序は、ステップまたはスタースキーマのどちらかを定義したいかに依存します。

ステップを定義している場合、表4 に示されている順序で以下のテンプレートをタグ言語ファイルに追加します。

表4. ステップを定義するときのテンプレートの相互関係および条件

順序	テンプレート	必須または任意選択
9	SubjectArea.tag	ステップを定義している場合は必須
10	Process.tag	ステップを定義している場合は必須

表 4. ステップを定義するときのテンプレートの相互関係および条件 (続き)

11	Step.tag	ソースおよびターゲット・データとの間での SQL 変換を生成している場合、またはデータウェアハウスセンターが実行するプログラムを定義している場合は必須。
12	StepInputTable.tag	以下のタイプのステップを定義している場合は必須。 ISV_StepType_Editioned_Append ISV_StepType_Full_Replace ISV_StepType_Uneditioned_Append 以下のタイプのステップを定義している場合は任意選択。 ISV_StepType_VWP_Population
13	StepOutputTable.tag	以下のタイプのステップを定義している場合は必須。 ISV_StepType_Editioned_Append ISV_StepType_Full_Replace ISV_StepType_Uneditioned_Append 以下のタイプのステップには、StepOutputTable を使用できません。 ISV_StepType_VWP_Population
14	StepVWPOutputTable.tag	以下のタイプのステップを定義している場合は任意選択。 ISV_StepType_VWP_Population
15	StepCascade.tag	ステップを連鎖関係にリンクするために必須
16	StepVWPProgramInstance.tag	ステップがデータウェアハウスセンター プログラムを使用する場合に必須

表4. ステップを定義するときのテンプレートの相互関係および条件 (続き)

17	VWPProgramInstanceParameter.tag	ステップが、パラメーターが渡されることを期待し、パラメーターを持っているデータウェアハウスセンター プログラムを使用する場合に必須。
----	---------------------------------	--

スタースキーマを定義している場合、表5 に示されている順序で以下のテンプレートをタグ言語ファイルに追加します。

表5. スタースキーマを定義するときのテンプレートの相互関係および条件

順序	テンプレート	必須または任意選択
9	StarSchema.tag	以下のタイプのスタースキーマを定義している場合は必須。
10	StarSchemaInputTable.tag	以下のタイプのスタースキーマを定義している場合は必須。

これらのテンプレートに関する詳細は、59ページの『第5章 メタデータ・テンプレート』を参照してください。

処理メッセージのログ記録: 交換プログラムは、ログ処理メッセージまたはトレース・ファイルを `VWS_LOGGING` 環境変数が指定するディレクトリーに書き込みます。データウェアハウスセンターはこのディレクトリーを、ログ・ファイルおよびトレース・ファイルのために使用します。

タグ言語ファイルのためのヘッダーを定義する: タグ言語ファイルに含めることのできるオブジェクトを定義するため、ヘッダーを定義しなければなりません。

ヘッダーを定義するには、以下のように行います。

1. 適当なテンプレートをコピーします。
2. トークンを実際の値に置換します。

テンプレートをコピーする: プログラムは、HeaderInfo.tag テンプレート・ファイルをコピーして変更しなければなりません。

値を置換する: プログラムは、以下の値を提供しなければなりません。

- デフォルトのセキュリティー・グループ、`ISV_DEFAULTSECURITYGROUP`
- ヘッダーのメタデータのための、`CurrentCheckPointID++` トークンの値

テンプレート内のトークンに関する情報については、70ページの『HeaderInfo.tag』を参照してください。

プログラム・ロジック: 図3 は、タグ言語ファイルのヘッダー部分を構築するためにプログラムで使用できるロジックの疑似コードです。

```
Initialize native metadata environment (need to include ISV_defines.h)
Read a copy of the HeaderInfo.tag template (from the templates directory)
Search for and replace tokens with the metadata from your native metadata
  store (or defaults)
Write the output to a target file
```

図3. ヘッダーをタグ言語ファイルに追加するための疑似コード

ISV_Sample プログラムは、タグ言語ファイルのヘッダー部分の例を示しています。プログラムのソース・コードは、VWS_TEMPLATES 環境変数によって設定されたディレクトリーの Samples サブディレクトリーにあります。

エージェント・サイトの定義

以下のエージェント・サイト・タイプの 1 つを使用できます。

- ウェアハウス・コントロール・データベース内ですでに定義されているエージェント・サイト。
既存のエージェント・サイトを使用するためには、*AgentSite トークンのすべての出現をエージェント・サイト名に置き換えます。
- デフォルトのエージェント・サイト。
デフォルトのエージェント・サイトを使用するためには、*AgentSite トークンのすべての出現を ISV_DEFAULTAGENTSITE に置き換えます。
- AgentSite テンプレートを使用して定義した新規のエージェント・サイト。
新規のエージェント・サイトを定義するには、AgentSite テンプレート内のトークンの値を定義します。*AgentSite トークンのすべての出現を、新規のエージェント・サイト名に置き換えます。

新規のエージェント・サイトを定義するには、以下のように行います。

1. 適当なテンプレートをコピーします。
2. トークンを実際の値に置換します。

テンプレートをコピーする: プログラムは、AgentSite.tag テンプレート・ファイルをコピーして変更しなければなりません。AgentSite.tag テンプレートには、前提条件として HeaderInfo.tag テンプレートが必要です。

値を置換する: 新規のエージェント・サイトを定義するために、プログラムはウェアハウス・エージェントがインストールされているワークステーションについてのメタデータを取得しなければなりません。プログラムは取得した値でテンプレート内の適切なトークンを置換しなければなりません。

プログラム・ロジック: 図4 は、タグ言語ファイルに新規のエージェント・サイトを追加するためにプログラムで使用できるロジックの疑似コードを示しています。

```
If the ISV wants to create an AgentSite specific to the ISV:
    Search for and replace tokens with the metadata from your native
    metadata store (or defaults)
    Append the output to a target file
Else
    Set AgentSite token to default agentsite value
```

図4. *AgentSite.tag* テンプレートを変更する疑似コードの例

ISV_Sample プログラムは、パートナー・ツールに特定のエージェント・サイトをタグ言語ファイルに追加する例を示しています。プログラムのソース・コードは、*VWS_TEMPLATES* 環境変数によって設定されたディレクトリーの **Samples** サブディレクトリーにあります。

ソースおよびターゲットの定義

データウェアハウスセンターまたはパートナー・アプリケーションがソースから読み取るようにしたい場合、それらのソースを定義します。同様に、データウェアハウスセンターまたはパートナー・アプリケーションがデータをターゲットに書き込むようにしたい場合、それらのターゲットを定義します。以下の条件に適合する場合を除き、使用するすべてのソースおよびターゲットを定義しなければなりません。

- そのソースまたはターゲットがすでにウェアハウス・コントロール・データベース内にある。
- データウェアハウスセンター プログラムを使用するステップだけを使用している。

ソースおよびターゲットを定義するには、以下のように行います。

1. 適当なテンプレートをコピーします。
2. トークンを実際の値に置換します。

テンプレートをコピーする: 以下のタイプのソース・オブジェクトを定義できます。

- リレーショナル・データベース
- IMS データベース
- ファイル・システム
- ファイル

リレーショナル・データベースをターゲット・オブジェクトとして定義することができます。

表6 および 表7 は、ソースおよびターゲット・オブジェクトの各タイプを定義するために、プログラムがコピーして変更しなければならないテンプレートをリストしています。

リレーショナル表: 表6 は、リレーショナル・データベースを定義するために、プログラムがコピーしなければならないテンプレートを示しています。

表6. リレーショナル・ソースおよびターゲット定義のためのテンプレート

ソースまたは ターゲットの定義	テンプレートの コピー数	コピーするテンプレート	前提条件となるテンプレート
データベース	使用したいデータベースごとに 1 コピー	SourceDataBase.tag (87ページを参照) WarehouseDataBase.tag (109ページを参照)	HeaderInfo.tag (70ページを参照) デフォルト・エージェントを使用していない場合、AgentSite.tag (61ページを参照)
表	データベースに定義したい表ごとに 1 コピー	Table.tag (92ページを参照)	SourceDataBase.tag (87ページを参照) WarehouseDataBase.tag (109ページを参照)
列	各表に定義したい列ごとに 1 コピー	Column.tag (64ページを参照)	Table.tag (92ページを参照)

テンプレート内に共通の値を指定することによって、表のテンプレートとデータベースのテンプレートとを関連付けます。同様に、テンプレート内に共通の値を指定することによって、列のテンプレートと表のテンプレートとを関連付けます。

25ページの図5 は、データベース、表、および列のテンプレートの関係を示しています。1 から m の表記は、1 対多の関係を示します。ここで、多には 0 も含まれます。



図5. DataBase.tag、Table.tag、および Column.tag のテンプレートの関係

IMS データベース: 表7 は、IMS データベースを定義するために、プログラムがコピーしなければならないテンプレートを示しています。これらの IMS オブジェクトにアクセスするためには、データウェアハウスセンター ODBC ドライバーを使用しなければなりません。

表7. IMS ソース定義のテンプレート

ソースまたはターゲットの定義	テンプレートのコピー数	コピーするテンプレート	前提条件となるテンプレート
データベース	使用したいデータベースごとに 1 コピー —	SourceDataBase.tag (87ページを参照)	HeaderInfo.tag (70ページを参照) デフォルト・エージェントを使用していない場合、AgentSite.tag (61ページを参照)
セグメント	データベースで使用したいセグメントごとに 1 コピー	Table.tag (92ページを参照)	SourceDataBase.tag (87ページを参照)
フィールド	各セグメントで使用したいフィールドごとに 1 コピー	Column.tag (64ページを参照)	Table.tag (92ページを参照)

表の関係を定義したときと同様の方法で、データベース、セグメント、およびフィールドの各テンプレート間の関係を定義します。(図5 を参照)

ファイル: 表7 は、ファイル・システムおよび関連ファイル、または単一のファイルのどちらかを定義するために、プログラムがコピーしなければならないテンプレートを示しています。

表 8. ファイル・システムまたは単一のファイルのためのテンプレート

ソースまたはターゲットの定義	テンプレートのコピー数	コピーするテンプレート	前提条件となるテンプレート
ファイル・システム	ファイル・システムごとに 1 コピー	SourceDataBase.tag (87ページを参照)	HeaderInfo.tag (70ページを参照) デフォルト・エージェントを使用していない場合、AgentSite.tag (61ページを参照)
ファイル	ファイル・システムで使いたいファイルごとに 1 コピー	Table.tag (92ページを参照)	SourceDataBase.tag (87ページを参照)
フィールド	各ファイルで使いたいフィールドごとに 1 コピー	Column.tag (64ページを参照)	Table.tag (92ページを参照)

表の関係を定義したときと同様の方法で、ファイル・システム、ファイル、およびフィールドの各テンプレート間の関係を定義します。(25ページの図5を参照)

値を置換する: プログラムは、パートナー・メタデータ・ストアからデータベースまたはファイルを説明する値を取得しなければなりません。プログラムは取得した値でテンプレート内の適切なトークンを置換しなければなりません。

データベース: プログラムは、ソース・データベースまたはターゲット・データベースについての以下のメタデータを提供しなければなりません。

- 定義するソース・データベース、または定義するターゲット・データベース
- データベースが存在するマシン
- 定義する各データベース内の表
- 定義する各表内の列

ファイル: プログラムは、ソース・ファイルについての以下のメタデータを提供しなければなりません。

- ファイルを含むファイル・システム
- 定義するソース・ファイル、または定義するターゲット・ファイル
- ファイルが存在するマシン
- 定義する各ファイル内のフィールド

プログラム・ロジック: 図6 は、ソースまたはターゲットを定義するデータ・リソースを作成または更新するために、プログラムで使用できるロジックの疑似コードを示しています。

```
For each source or target to be defined:
  Read a copy of the SourceDatabase.tag or WarehouseDatabase.tag template
  Search for and replace tokens with the metadata from your native metadata source
  (or defaults)
  Append the output to a target file
  For each table, file, or segment that is to be defined:
    Read a copy of the Table.tag template
    Search for and replace tokens with the metadata from your native metadata source
    (or defaults)
    Append the output to a target file
    For each column or field that the table contains:
      Read a copy of the Column.tag template
      Search for and replace tokens with the metadata from your native metadata source
      (or defaults)
      Append the output to a target file
    End (for each column)
  End (for each table)
End (for each source or target data source)
```

図6. ソースまたはターゲットを定義するデータ・リソースを作成または更新するための疑似コード。このロジックを作成または更新したいソースまたはターゲット定義ごとに使用します。

ISV_Sample プログラムは、ソースまたはターゲット定義のためのデータ・ソースを作成または更新する例を示しています。プログラムのソース・コードは、`VWS_TEMPLATES` 環境変数によって設定されたディレクトリーの `Samples` サブディレクトリーにあります。

データウェアハウスセンター プログラムを定義する

データウェアハウスセンターがパートナー・アプリケーションをスケジューリングして実行するようにしたい場合、最初にそのアプリケーションをデータウェアハウスセンター プログラムとして定義しなければなりません。その後、それを 1 つ以上のステップで使用して、プログラムをスケジューリングおよび実行することができます。

タグ言語ファイルがデータウェアハウスセンター プログラムを含める場合、以下のオブジェクトを順序どおりに定義しなければなりません。

1. データウェアハウスセンター プログラムを含める 1 つ以上のプログラム・グループ。
2. データウェアハウスセンターにプログラムの基本定義を示す 1 つ以上のデータウェアハウスセンター プログラム・テンプレート。

3. データウェアハウスセンターがプログラムに渡すデフォルトのパラメーターを提供する、1つ以上のデータウェアハウスセンター プログラム・テンプレートのパラメーター。

ステップのためにプログラム・パラメーターのインスタンスを定義することによって、特定のステップで使用されるパラメーターを変更できます。データウェアハウスセンター プログラムをステップで使用するについての詳細は、30ページの『ステップを定義する』を参照してください。

データウェアハウスセンターと共に使用するプログラムを作成することについての詳細は、333ページの『付録C. データウェアハウスセンターで使用するプログラムを独自に作成する』を参照してください。

データウェアハウスセンター プログラムを定義するには、以下のように行います。

1. 適当なテンプレートをコピーします。
2. トークンを実際の値に置換します。

テンプレートをコピーする: 表9 は、データウェアハウスセンター プログラムを定義するために、プログラムがコピーして変更しなければならないテンプレートを示しています。

表9. データウェアハウスセンター プログラムのテンプレート

定義	テンプレート		前提条件となるテンプレート
	のコピー数	コピーするテンプレート	
データウェアハウスセンター プログラム・グループ	定義するプログラム・グループごとに1コピー	VWPGGroup.tag (99ページを参照)	HeaderInfo.tag (70ページを参照)
データウェアハウスセンター プログラム・テンプレート	プログラム・グループ内のデータウェアハウスセンター プログラムごとに1コピー	VWPPProgramTemplate.tag (103ページを参照)	VWPGGroup.tag (99ページを参照)

表9. データウェアハウスセンター プログラムのテンプレート (続き)

定義	テンプレートのコピー数	コピーするテンプレート	前提条件となるテンプレート
データウェアハウスセンタープログラム・テンプレートのパラメーター	データウェアハウスセンタープログラムに渡されるパラメーターごとに1コピー	VWPPProgramTemplateParameter.tag (106ページを参照)	VWPPProgramTemplate.tag (103ページを参照)

テンプレート内に共通の値を指定することによって、データウェアハウスセンター プログラム・グループのテンプレートとデータウェアハウスセンター プログラムのテンプレートとを関連付けます。同様に、テンプレート内に共通の値を指定することによって、パラメーターのテンプレートとデータウェアハウスセンター プログラムのテンプレートとを関連付けます。

図7 は、データウェアハウスセンター プログラム・グループ、データウェアハウスセンター プログラム、およびデータウェアハウスセンター プログラム・パラメーターの間の関係を示しています。

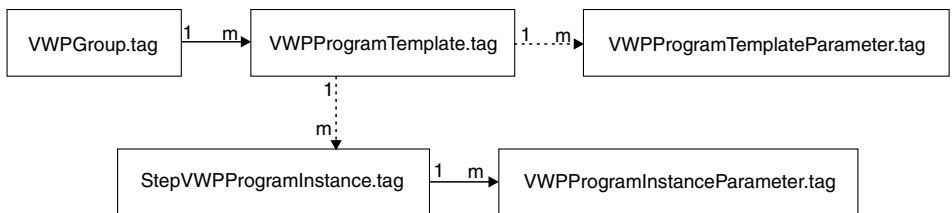


図7. VWPGroup.tag、VWPPProgramTemplate.tag、および VWPPProgramTemplateParameter.tag の各テンプレート間の関係

データウェアハウスセンター プログラムをステップに関連付けることについての詳細は、30ページの『ステップを定義する』を参照してください。

値を置換する: プログラムは、パートナー・メタデータ・ストアからデータウェアハウスセンター プログラムを説明する値を取得しなければなりません。

- 定義するデータウェアハウスセンター プログラム・グループ
- 定義するデータウェアハウスセンター プログラム
- 定義する各データウェアハウスセンター プログラム内のパラメーター

プログラムは取得した値でテンプレート内の適切なトークンを置換しなければなりません。

プログラム・ロジック: 図8 は、データウェアハウスセンターによって管理および実行されるアプリケーションを定義するため、プログラムで使用できるロジックの疑似コードを示しています。

```
Read a copy of the VWPGroup.tag template
Search for and replace tokens with the metadata from your native metadata store
(or defaults)
Append the output to a target file
For each application that is to be managed by the Data Warehouse Center:
  Read a copy of the VWPPProgramTemplate.tag template
  Search for and replace tokens with the metadata from your native metadata store
  (or defaults)
  Append the output to a target file

  For each parameter the application needs passed:
    Read a copy of the VWPPProgramTemplateParameter.tag template
    Search for and replace tokens with the metadata from your native metadata store
    (or defaults)
    Append the output to a target file
  End (for each parameter)
End (for each application)
```

図8. データウェアハウスセンター プログラムを定義するための疑似コード

ISV_Sample プログラムは、データウェアハウスセンター プログラムをタグ言語ファイルに追加する例を示しています。プログラムのソース・コードは、**VWS_TEMPLATES** 環境変数によって設定されたディレクトリーの **Samples** サブディレクトリーにあります。

ステップを定義する

ステップとは、ウェアハウス・プロセスでデータに対する単一の操作のことです。ほとんどの場合、ステップにはウェアハウス・ソース、データの変換または移動、およびウェアハウス・ターゲットが含まれます。ステップはスケジュールに従って実行したり、他のステップからの連鎖で実行することができます。ステップを使用して、データの抽出、変換、および書き込みにおける各ステップを定義およびスケジューリングします。データウェアハウスセンターが管理するようにしたい変換プロセスの部分ごとに、ステップを定義しなければなりません。データウェアハウスセンター オンライン・ヘルプの情報を使用するよりも、この節の情報を使用してステップを定義する方法を決めてください。ユーザー・インターフェースを使用してステップを定義すると、テンプレートには複数の異なる関係が必要となります。

ステップのためのサブジェクト・エリアを定義しなければなりません。特定のパートナー・アプリケーションを使用するステップのために、サブジェクト・エリアを定義することができます。

タグ言語ファイルにステップが含まれる場合、以下のオブジェクトを順序どおりに定義しなければなりません。

1. プロセスを含めるための 1 つ以上のサブジェクト・エリア。
2. ステップを含めるための 1 つ以上のプロセス。
3. 1 つ以上のステップ。
4. ステップが SQL を使用してソース・ターゲット・マッピングを行う場合、ステップごとに 1 つ以上のソース表および 1 つのターゲット表との関係。ステップがデータウェアハウスセンター プログラムを使用する場合、ソース表およびターゲット表は任意指定です。
5. ステップがデータウェアハウスセンター プログラムを使用する場合、
 - a. データウェアハウスセンター プログラムのインスタンス。
 - b. データウェアハウスセンター プログラムに関連したパラメーター。
 - c. 任意指定で、データウェアハウスセンター プログラムの出力表。

ステップを定義するには、以下のように行います。

1. 適当なテンプレートをコピーします。
2. トークンを実際の値に置換します。

テンプレートをコピーする: 表10 は、ステップを定義するために、プログラムがコピーして変更しなければならないテンプレートを示しています。

表 10. ステップのテンプレート

定義	テンプレートのコピー数		前提条件となるテンプレート
	サブジェクト・エリアごと	コピーするテンプレート	
サブジェクト・エリア	サブジェクト・エリアごとに 1 コピー	SubjectArea.tag (90ページを参照)	HeaderInfo.tag (70ページを参照) デフォルト・エージェントを使用していない場合、AgentSite.tag (61ページを参照)
プロセス	プロセスごとに 1 コピー	Process.tag (71ページを参照)	SubjectArea.tag (90ページを参照)
ステップ	ステップごとに 1 コピー	Step.tag (76ページを参照)	SubjectArea.tag (90ページを参照) Process.tag (71ページを参照)

表 10. ステップのテンプレート (続き)

定義	テンプレートのコピー数	コピーするテンプレート	前提条件となるテンプレート
ステップのソース表	ステップのソース表ごとに 1 コピー	StepInputTable.tag (81ページを参照)	Table.tag (92ページを参照) Step.tag (76ページを参照) Process.tag (71ページを参照)
ステップのターゲット表	ステップにターゲット表がある場合、1 コピー	StepOutputTable.tag (83ページを参照)	Table.tag (92ページを参照) Step.tag (76ページを参照) Process.tag (71ページを参照)
データウェアハウスセンタープログラムを使用するステップのターゲット表	プログラムによって更新される各ターゲット表を記述するために 1 コピー	StepOutputTable.tag (83ページを参照)	Table.tag (92ページを参照) Step.tag (76ページを参照)
データウェアハウスセンタープログラム・インスタンス	ステップがデータウェアハウスセンタープログラムを使用する場合、1 コピー	StepVWPPProgramInstance.tag (85ページを参照)	VWPPProgramTemplate.tag (103ページを参照) Step.tag (76ページを参照)
データウェアハウスセンタープログラム・インスタンスのパラメーター	ステップで使用されるパラメーターごとに 1 コピー	VWPPProgramInstanceParameter.tag (100ページを参照)	StepVWPPProgramInstance.tag (85ページを参照)

テンプレート内に共通の値を指定することによって、サブジェクト・エリアのテンプレートとプロセスのテンプレートとを関連付けます。同様に、テンプレート内に共通の値を指定することによって、ステップのテンプレートと入力表および出力表のテンプレートとを関連付けます。さらに、テンプレート内に共通の値を指定することによって、ステップのテンプレートとプログラム・インスタンスのテンプレートとを関連付けることができます。

図9 は、サブジェクト・エリア、ステップ、ステップ入力表、ステップ出力表、ステップ VWP プログラム・インスタンス、および VWP プログラム・インスタンス・パラメーター・タグの間の関係を示しています。

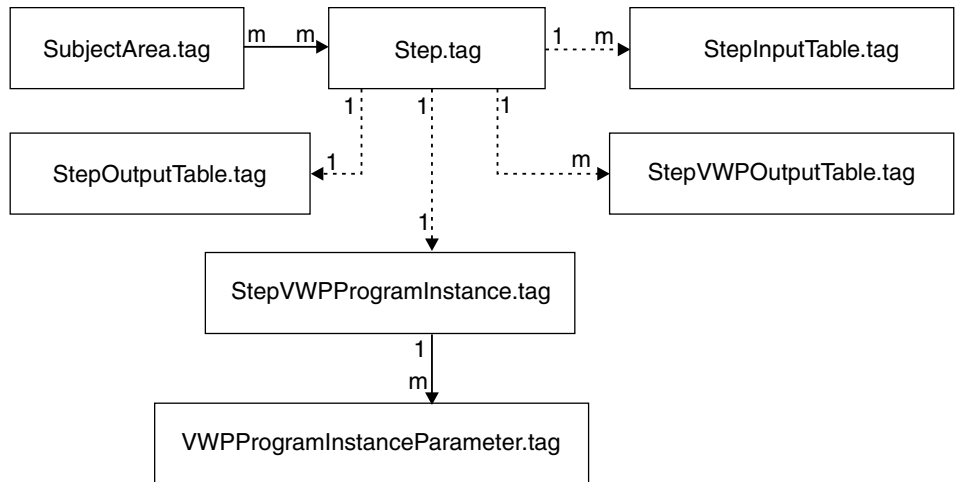


図9. SubjectArea.tag、Process.tag、Step.tag、StepInputTable.tag、StepOutputTable.tag、StepVWPOutputTable.tag、StepVWPProgramInstance.tag、および VWPPProgramInstanceParameter.tag の各テンプレートの間の関係。データウェアハウスセンター プログラム・テンプレートのテンプレートと他のデータウェアハウスセンター プログラムのテンプレートとがどのように関連するかは、29ページの図7 をご覧ください。

値を置換する: プログラムは、パートナー・メタデータ・ストアからサブジェクト・エリアまたはステップを説明する値を取得しなければなりません。

- ステップを含むプロセスを含むサブジェクト・エリア
- 定義するステップ
- ステップごとのソース表
- 該当する場合、ステップごとのターゲット表
- 該当する場合、データウェアハウスセンター プログラムおよびステップのパラメーター

プログラムは取得した値でテンプレート内の適切なトークンを置換しなければなりません。

プログラム・ロジック: 34ページの図10 は、タグ言語ファイル内にステップを定義するためにプログラムで使用できる、ロジックの疑似コードを示しています。

```

Read a copy of the SubjectArea.tag template
Search for and replace tokens with the metadata from your native metadata store (or defaults)
Append the output to a target file
Read a copy of the process

For each step to be defined:
  Read a copy of the Step.tag template
  Search for and replace tokens with the metadata from your native metadata store
  (or defaults)
  Append the output to a target file
  If the step is to execute your application:
    Read a copy of the StepVWPPProgramInstance.tag template
    Search for and replace tokens with the metadata from your native metadata store
    (or defaults)
    Append the output to a target file
    For each parameter that your application needs:
      Read a copy of the VWPPProgramInstanceParameter.tag template
      Search for and replace tokens with the metadata from your native metadata store
      (or defaults)
      Append the output to a target file
    End (for each parameter)

    If the step is to be related to its VWP output target data:
      Read a copy of the StepVWPOutputTable.tag template
      Search for and replace tokens with the metadata from your native metadata store
      (or defaults)
      Append the output to a target file
    End (step relation to its output)
  End (if step to execute your application)

  If the step is to be related to its input source data:
    Read a copy of the StepInputTable.tag template
    Search for and replace tokens with the metadata from your native metadata store
    (or defaults)
    Append the output to a target file
  End (step relation to its source)

  If the step is to be related to its output target data:
    Read a copy of the StepOutputTable.tag template
    Search for and replace tokens with the metadata from your native metadata store
    (or defaults)
    Append the output to a target file
  End (step relation to its target)
End (for each step)

```

図 10. タグ言語ファイルにステップを定義するための疑似コード

ISV_Sample プログラムは、ステップをタグ言語ファイルに追加する例を示しています。プログラムのソース・コードは、`VWS_TEMPLATES` 環境変数によって設定されたディレクトリーの `Samples` サブディレクトリーにあります。

連鎖ステップを定義する

タグ言語ファイル内で、ステップが他のステップを開始するように指定できます。

- 後処理連鎖関係を定義することにより、1 つのステップが別のステップの処理が正常に終了した後に開始するように指定できます。

連鎖ステップを定義するには、以下のように行います。

1. 適当なテンプレートをコピーします。
2. トークンを実際の値に置換します。

テンプレートをコピーする: 表11 は、連鎖関係を定義するために、プログラムがコピーして変更しなければならないテンプレートを示しています。

表 11. 連鎖関係のテンプレート

定義	テンプレートのコピー数	コピーするテンプレート	前提条件となるテンプレート
ステップの連鎖関係	関係ごとに 1 コピー	80ページの『StepCascade.tag』	80ページの『StepCascade.tag』

値を置換する: 以下を行うために、プログラムはステップの名前および別のステップの名前を指定しなければなりません。

- 最初のステップの後に開始する。

プログラムは取得した値でテンプレート内の適切なトークンを置換しなければなりません。

プログラム・ロジック: 図11 は、アプリケーションが 2 つのステップを関連付けて、1 つのステップが別のステップの完了時に開始するようにするためにプログラムで使用できる、ロジックの疑似コードを示しています。

```
Read a copy of the StepCascade.tag template
Search for and replace tokens with the metadata from your native metadata store
(or defaults)
Append the output to a target file
End (relate steps for cascaded processing)
```

図 11. 連鎖処理のためにステップを関連付ける疑似コード

ISV_Sample プログラムは、タグ言語ファイル内でステップを関連付けて連鎖処理を行う方法の例を示しています。プログラムのソース・コードは、*VWS_TEMPLATES* 環境変数によって設定されたディレクトリーの *Samples* サブディレクトリーにあります。

タグ言語ファイルからメタデータをインポートする

コマンド・ウィンドウまたはユーザー・インターフェースを使用して、メタデータをタグ言語ファイルからインポートすることができます。この節では、コマンド・ウィンドウの使用方法を解説します。ユーザー・インターフェースの使用方法については、データウェアハウスセンターのオンライン・ヘルプを参照してください。

タグ言語ファイルをインポートするためには、DOS コマンド・プロンプトから以下のコマンドを入力します。

```
iwh2imp2 tag-filename log-pathname target-control-db userid password  
[PREFIX = schema]
```

tag-filename

タグ言語ファイルの全パスおよびファイル名。

log-pathname

ログ・ファイルの完全修飾パス名。

target-control-db

インポートのターゲット・データベースであるウェアハウス・コントロール・データベースの名前。

userid ウェアハウス・コントロール・データベースにアクセスするためのユーザー ID。

password

ウェアハウス・コントロール・データベースにアクセスするためのパスワード。

[*PREFIX = schema*]

メタデータ表の表修飾子。

接頭部が指定されない場合、デフォルト値は *IWH* です。

インポート・コマンドのパラメーターに関するヘルプを表示するには、コマンドだけを入力します。

インポート・ユーティリティーはメタデータをタグ言語ファイルからインポートするときに、以下のファイル名とファイル拡張子を持つログ・ファイルを作成します。

- ファイル名はタグ言語ファイルと等しい。
- ファイル拡張子は LOG。

インポート・プロセスは、戻りコードおよび最後に完了したチェックポイントをログ・ファイルの末尾に記録します。

system() 呼び出しまたは rexec() 呼び出しを使用して、戻りコードを交換プログラムにコーディングすることもできます。使用する呼び出しは、プログラムを実行しているオペレーティング・システムに依存します。

メタデータをデータウェアハウスセンターにインポートすることについての詳細は、[データウェアハウスセンター 管理の手引き](#) を参照してください。

ステップを実行に備えて準備する

メタデータをデータウェアハウスセンターにインポートした後、以下の手順を完了して、ウェアハウスのための自動処理をセットアップしなければなりません。

1. 以下のオブジェクトのためのパスワードを指定します。
 - インポートしたエージェント・サイト
 - インポートしたウェアハウス・ソースまたはウェアハウス・ターゲット (ソースおよびターゲット)
2. SQL ステップでは、ソース表またはファイルをターゲット表に直接マップする場合、ソース列をターゲット列にマップしてください。
3. データウェアハウスセンターでオブジェクトが作成された後、データウェアハウスセンターを使用して特定の日および時刻のスケジュールをステップに定義します。さらに、タグ言語ファイル内で定義していない場合、連鎖関係を定義することもできます。
4. ステップをテスト・モードに昇格します。
5. ステップをテストするには、「新規ステップの実行 (Run New Step)」ウィンドウでそれらを選択して実行します。

変更を加える必要がある場合、以下のように行います。

 - a. 必要であれば、ステップを開発モードに降格します。
 - b. 変更を加えます。
 - c. 再び、ステップをテスト・モードに昇格させます。

プログラムを更新して、これらの変更が反映されるようにします。
6. ステップを実働モードに昇格して、それらのスケジュールを活動化します。これで、ステップは自動スケジュールによって実行するようになりました。

データウェアハウスセンターからのメタデータのエクスポート

パートナー・アプリケーションがデータウェアハウスセンターに定義されたデータ・ソースまたはターゲットを操作するようにしたい場合、メタデータをデータウェアハウスセンターからエクスポートします。

メタデータをデータウェアハウスセンターからエクスポートすることには、以下の手順が関係します。

1. メタデータをエクスポートするオブジェクトを選択します。
2. メタデータをタグ言語ファイルにエクスポートします。

メタデータをエクスポートするオブジェクトの選択

ほとんどのデータウェアハウスセンター オブジェクトは、データウェアハウスセンターに特定のものであります。しかし、データベース、表、および列についてのメタデータを使用して、パートナー・アプリケーションのためのソースおよびターゲット・データベースを定義することができます。この機能を使用して、同じウェアハウスのデータを変換するパートナー・アプリケーションと、ソースおよびターゲット情報を共有することができます。

たとえば、あるパートナー・ツールはデータベースからターゲット・ファイルにデータをダウンロードすると仮定します。別のパートナー・ツールは、そのファイルを使用して以下を実行するとします。

- そのファイルからデータを読み取ります。
- データを変換します。
- データを別のデータ・ファイルに書き込みます。

3 番目のパートナー・ツールは、ファイルからデータを読み取って、ターゲット・データベースにロードするとします。データベースおよびファイルについてのメタデータをデータウェアハウスセンターからエクスポートする場合、すべてのパートナー・ツールが同じデータ定義を使用していることを保証することができます。

ソース・データベースを定義するには、1 つ以上のウェアハウス・ソースをエクスポートします (すべての表および列は自動的に組み込まれます)。ターゲット・データベースを定義するには、ウェアハウス・ターゲットをエクスポートします (すべての表および列は自動的に組み込まれます)。

オブジェクトをエクスポートするとき、データウェアハウスセンターはタグ言語形式を使用してオブジェクトをファイルに書き込みます。ソース・データベースおよびターゲット・データベースのメタデータを識別するタグについての

情報は、115ページの『第6章 データウェアハウスセンターのメタデータ』を参照してください。タグ言語ファイルの構文および構造については、239ページの『第9章 タグ言語』 および 279ページの『第10章 タグ言語ファイルの使い方』を参照してください。

表12 は、論理データウェアハウスセンター オブジェクトと論理オブジェクトを表すタグ言語オブジェクトとの間のマッピングを示しています。

表 12. ソースおよびターゲット・データベースの論理オブジェクト

データウェアハウスセンター論理オブジェクト	タグ言語ファイル内のオブジェクト	説明	参照先:
ウェアハウス・ソース	DATABASE	ソース・データベースまたはファイル	115ページの『DATABASE オブジェクト』
ウェアハウス・ターゲット	DATABASE	ターゲット・データベースまたはファイル	115ページの『DATABASE オブジェクト』
表	TABLES	ソースまたはターゲット・データベース内の表、ファイル、またはセグメント	121ページの『TABLES オブジェクト』
列	COLUMN	列または表内のフィールドまたはファイル内のフィールド	127ページの『COLUMN オブジェクト』

タグ言語ファイルへのメタデータのエクスポート

データウェアハウスセンターのユーザー・インターフェースまたはコマンド・ウィンドウを使用して、メタデータをデータウェアハウスセンターからエクスポートすることができます。この節では、コマンド・ウィンドウの使用方法を解説します。ユーザー・インターフェースの使用方法については、データウェアハウスセンターのオンライン・ヘルプ、およびデータウェアハウスセンター管理の手引きを参照してください。

最初に、エクスポートしたいウェアハウス・ソースおよびウェアハウス・ターゲットのリストを含む .INP ファイルを作成します。以下に例を示します。

<IR>
LOG_STAT_IR
LOG_STAT_REP

LOG_STAT_IR はウェアハウス・ソースであり、LOG_STAT_REP はウェアハウス・ターゲットです。データウェアハウスセンターは LOG_STAT_IR および LOG_STAT_REP に関連した表および列を自動的にエクスポートします。

その後、タグ言語ファイルをエクスポートするためには、DOS コマンド・プロンプトから以下のコマンドを入力します。

```
iwh2exp2 INPfilename controlDBname userid password [PREFIX = schema]
```

INPfilename

.INP ファイルの全パスおよびファイル名。

データウェアハウスセンターはタグ言語ファイルをこのディレクトリーに書き込むため、このファイルを読み取り / 書き込みディレクトリーに作成してください。データウェアハウスセンターはタグ言語ファイルに *INPfilename.TAG* という名前を付けます。

controlDBname

制御データベースの名前。

userID 制御データベースにアクセスするために必要なユーザー ID。

password

制御データベースにアクセスするために必要なパスワード。

[*PREFIX = schema*]

メタデータ表の表修飾子。

接頭部が指定されない場合、デフォルト値は *IWH* です。

メタデータをデータウェアハウスセンターからエクスポートすることについての詳細は、データウェアハウスセンター 管理の手引き を参照してください。

インポート形式およびエクスポート形式は、リリースによって異なります。以前のリリースからのエクスポート・ファイルを使用して、データウェアハウスセンターのあるリリースから別のリリースへの移行を行うことはできません。移行についての詳細は、ご使用のプラットフォームの概説およびインストールを参照してください。

第3章 情報カタログ・マネージャーによるメタデータのインポートおよびエクスポート

この章では、情報カタログ・マネージャーにメタデータを直接インポートする方法、およびそこからメタデータを直接エクスポートする方法について詳しく解説します。

情報カタログにメタデータをインポートする

データウェアハウスセンターからメタデータをインポートして、ウェアハウスのユーザーのためにウェアハウス内のデータに関する情報を示すことができます。同様にいくつかのカタログ機能を備えたパートナー・アプリケーションから、メタデータをインポートすることができます。

メタデータを情報カタログ・マネージャーにインポートすることには、以下のタスクが関係します。

1. インポートするメタデータのタイプの選択。
2. 情報カタログ・マネージャーへのメタデータのインポート。

インポートするメタデータの選択

メタデータを情報カタログにインポートするとき、タグ言語を以下の 2 つの形式でインポートすることができます。

- 情報カタログ・マネージャーおよびデータウェアハウスセンターの両方によって使用される形式。
- MDIS に準拠する形式。

タグ言語ファイルからのメタデータのインポート

メタデータを、MDIS 形式または情報カタログ・マネージャーおよびデータウェアハウスセンターで使用される形式のタグ言語ファイルからインポートすることができます。情報カタログ・マネージャー・オブジェクト・タイプを MDIS 名にマップする方法については、163ページの『第8章 情報カタログ・マネージャー・オブジェクト・タイプ』を参照してください。MDIS タグ言語ファイルについては、<http://www.MDCinfo.com> にある Meta Data Coalition の Web サイトをご覧ください。

他の製品および Visual Warehouse 3.1 と共に MDIS を使用している場合、48 ページの『タグ言語ファイルのエクスポート』の注記をご覧ください。

MDIS タグ言語ファイルを情報カタログ・マネージャー・タグ言語ファイルに変換したい場合、情報カタログ・マネージャー 管理の手引き を参照してください。

MDIS 対応のタグ言語ファイルをインポートする

MDIS タグ言語ファイルを情報カタログに直接インポートするには、情報カタログ・マネージャー・コマンドを MS-DOS コマンド・プロンプトから入力します。コマンド構文に関する以下の規則を守ってください。

- 指定のないすべての部分については、大文字小文字を区別しません。
- 各キーワードの前に、スラッシュ (/) またはハイフン (-) のどちらかが必要です。
- DGUIDE コマンドに続くすべてのキーワードは必須です。 /MDIS_IMPORT キーワードに続くすべてのキーワードは必須です。
- 下線のある選択項目はデフォルトです。

```
DGUIDE /USERID userid /PASSWORD password /DGNAME dname /MDIS_IMPORT filename /LOGFILE filename name/ADMIN
```

任意指定のキーワードは以下のとおりです。

```
/TRACE 0|1|2|3|4
```

たとえば、MDIS メタデータを情報カタログにインポートする場合、以下のコマンドをタイプします (改行は入力しません)。

```
DGUIDE /USERID longods /PASSWORD secret /DGNAME ICMSAMP /ADMIN /MDIS_IMPORT c:%mdis.tag /LOGFILE c:%mdis.log
```

/ADMIN

管理者としてログオンすることを指定します。メタデータをインポートするためには、アドミニストレーターとしてログオンする必要があります。

/DGNAME

情報カタログ名。

情報カタログがローカルである場合、データベース名を指定します。情報カタログがリモートである場合、それがカタログ化される別名を指定します。

例:

```
/DGNAME ICMSAMP
```

/LOGFILE

このパラメーターは必須です。

情報カタログ・マネージャーが MDIS のインポートまたは MDIS のエクスポートの際に生成するメッセージを送るファイルの宛先を指定します。ドライブ、パス、およびファイル名を含めた全体を指定するのでなければ、情報カタログ・マネージャーはそのファイルを DGWPATH 環境変数で指定されたパスに置きます。固定ドライブを指定しなければなりません。

例:

```
/LOGFILE d:¥tagfile.log
```

/MDIS_IMPORT

指定した MDIS 対応のタグ言語ファイルをインポートします。ドライブ、パス、およびファイル名を含めた全体を指定するのでなければ、情報カタログ・マネージャーはそのファイルが DGWPATH 環境変数で指定されたパスに存在すると想定します。

例:

```
/MDIS_IMPORT d:¥tagfile.tag
```

MDIS メタデータをインポートする情報カタログには、有効な MDIS オブジェクト・タイプ定義が含まれていなければなりません (それ以外のものが含まれていることも可能)。

/PASSWORD

このユーザー ID に対するパスワード。

例:

```
/PASSWORD secret
```

以下のオペレーティング・システム上にあるデータベースにアクセスするためのパスワードは大文字小文字の区別をするため、指定されたとおりにタイプしなければなりません。

- AIX
- Windows NT および Windows 2000
- Solaris 実行環境版

/TRACE

情報カタログ・マネージャー・トレース・ファイルに送るトレース情報のレベル。高位のレベルには下位のレベルの機能が含まれます (たとえば、3にはレベル 0、1、2、および 3 の機能が含まれます)。情報カタログ・マ

ネージャーの問題を診断するために IBM ソフトウェア・サポートを呼び出すには、高いレベルを指定しなければならない場合があります。

- 0 デフォルト。すべてのメッセージ、警告、エラー、および重大エラー状態が含まれます。
- 1 最高レベルの情報カタログ・マネージャー機能の入り口レコードおよび出口レコードが含まれます。
- 2 情報カタログ・マネージャー機能の入り口レコードおよび出口レコードで、一般的なものが含まれます。
- 3 入力および出力パラメーター（入力または出力構造を除く）が含まれます。
- 4 情報カタログ・マネージャーに渡されて使用されたすべての入力または出力構造が含まれます。

/USERID

情報カタログのユーザー ID。情報カタログがあるデータベースに必要なユーザー ID を入力します。たとえば、ユーザー ID はローカル、LAN、AS/400、AIX、または OS/390 TSO ユーザー ID であることがあります。

例:

```
/USERID longods
```

タグ言語ファイルをコマンド行からインポートする

MS-DOS コマンド・プロンプトから DGUIDE コマンドを使用して、情報カタログをオープンし、タグ言語ファイルをインポートすることができます。

DGUIDE コマンドを使用するとき、コマンド構文に関する以下の規則に注意してください。

- 指定のないすべての部分については、大文字小文字を区別しません。
- 各キーワードの前にはスラッシュ (/) またはハイフン (-) 文字のいずれかを付けます。
- インポートを選択する場合、45ページの図12 に示されているように /IMPORT に続くすべてのキーワードは必須です。
- 下線のある選択項目はデフォルトです。

DGUIDE /**USERID** *userid* /**PASSWORD** *password* /**DGNAME** *dgname*

任意指定のキーワードは以下のとおりです。

/ADMIN
/TRACE 0|1|2|3|4
/IMPORT *filename* /**LOGFILE** *filename* /**RESTART** B|C

任意指定のインポート・キーワードは以下のとおりです。

/ICOPATH *iconpath*

図 12. *i* をオープンしてメタデータをインポートするための **DGUIDE** コマンド・パラメーター

以下の例は、サンプルの情報カタログを管理者としてオープンするための必須パラメーターを示しています。

```
DGUIDE /USERID longods /PASSWORD secret /DGNAME ICMSAMP /ADMIN
```

以下のリストは、**DGUIDE** コマンドに追加するパラメーターを示しています。タグ言語ファイルをインポートするための任意指定または必須キーワードについては、注記されています。

/ADMIN

管理者としてログオンすることを指定します。この任意指定のキーワードを **DGUIDE** コマンドに指定しない場合、ユーザーとしてログオンします。その場合、管理者タスクを実行することはできません。

/DGNAME

情報カタログ名。

情報カタログがローカルである場合、データベース名を指定します。情報カタログがリモートである場合、それがカタログ化される別名を指定します。

例:

```
/DGNAME ICMSAMP
```

/ICOPATH

/IMPORT; がある場合にのみ有効、任意指定。

アイコンをインポートすることを示し、インポート・ファンクションが使用するアイコン・パスを指定します。ドライブおよびパスの全体を指定しない場合、情報カタログ・マネージャーはパスが情報カタログ・マネージャーをインストールしたパスと同じであると想定します。固定ドライブを指定しなければなりません。

例:

/ICOPATH d:%icons¥

/IMPORT

指定したタグ言語ファイルをインポートします。ドライブ、パス、およびファイル名を含めた全体を指定するのでなければ、情報カタログ・マネージャーはそのファイルが DGWPATH 環境変数で指定されたパスに存在すると想定します。

例:

```
/IMPORT d:%tagfile.tag
```

このキーワードは情報カタログ・マネージャー・ユーザー・インターフェースをう回して、インポート・ファンクションをバッチ処理として実行します。

/LOGFILE

/IMPORT; がある場合にのみ有効、/IMPORT がある場合には必須。

情報カタログ・マネージャーがインポートの際に生成するメッセージを送るファイルの宛先を指定します。ドライブ、パス、およびファイル名を含めた全体を指定するのでなければ、情報カタログ・マネージャーはそのファイルを DGWPATH 環境変数で指定されたパスに置きます。固定ドライブを指定しなければなりません。

例:

```
/LOGFILE d:%tagfile.log
```

/PASSWORD

このユーザー ID に対するパスワード。

例:

```
/PASSWORD secret
```

以下のオペレーティング・システム上にあるデータベースにアクセスするためのパスワードは大文字小文字の区別をするため、指定されたとおりにタイプしなければなりません。

- AIX
- Windows NT および Windows 2000
- Solaris 実行環境版

/RESTART

/IMPORT; がある場合にのみ有効、/IMPORT がある場合には必須。

インポート・ファンクションが使用するオプションを示します。有効なオプションは以下のとおりです。

- B** タグ言語ファイルを最初からインポートします。
- C** デフォルト。タグ言語ファイルを、最後に情報カタログ・マネージャーが情報カタログへの変更を正常にコミットした箇所からインポートします。

/TRACE

トレース・ファイルに送るトレース情報のレベル。高位のレベルには下位のレベルの機能が含まれます (3 にはレベル 0、1、2、および 3 の機能が含まれます)。情報カタログ・マネージャーの問題を診断するために IBM ソフトウェア・サポートを呼び出すには、高いレベルを指定しなければならない場合があります。

- 0** デフォルト。すべてのメッセージ、警告、エラー、および重大エラー状態が含まれます。
- 1** 最高レベルの情報カタログ・マネージャー機能の入り口レコードおよび出口レコードが含まれます。
- 2** 情報カタログ・マネージャー機能の入り口レコードおよび出口レコードで、一般的なものが含まれます。
- 3** 入力および出力パラメーター (入力または出力構造を除く) が含まれます。
- 4** 情報カタログ・マネージャーに渡されて使用されたすべての入力または出力構造が含まれます。

/USERID

情報カタログのユーザー ID。オープンしている情報カタログのデータベース位置に応じて、データベースのために必要なユーザー ID をタイプします。たとえば、ユーザー ID はローカル、LAN、AS/400、AIX、または OS/390 TSO ユーザー ID であることがあります。

例:

```
/USERID longods
```

情報カタログ・マネージャーからのメタデータのエクスポート

メタデータを情報カタログ・マネージャーからエクスポートして、パートナー・アプリケーションが使用できるようにすることができます。たとえば、情報カタログ・マネージャー・メタデータをエクスポートして、アプリケーション開発者がデータウェアハウス用のアプリケーション開発に使用する CASE ツールで使用することができます。

メタデータを情報カタログ・マネージャーからエクスポートすることには、以下のタスクが関係します。

1. エクスポートするメタデータのタイプの選択。
2. 情報カタログ・マネージャーからのメタデータのエクスポート。

メタデータを情報カタログ・マネージャーからエクスポートするとき、タグ言語を 2 つの形式で生成することができます。以下に例を示します。

- 情報カタログ・マネージャー製品のウィンドウまたは FLGExport API を使用してエクスポートする場合、生成されるタグ言語は情報カタログ・マネージャー・タグ言語形式です。メタデータは、Windows® 95、Windows NT®、または Windows 2000 のコマンド行からエクスポートすることができます。詳細については、[情報カタログ・マネージャー 管理の手引き](#) を参照してください。
- FLGMDisExport API を使用してエクスポートする場合、生成されるタグ言語は MDIS 形式です。情報カタログ・マネージャー API についての詳細は、[情報カタログ・マネージャー プログラミングの手引きおよび解説書](#) を参照してください。

エクスポートするメタデータの選択

情報カタログ・マネージャーからエクスポート可能なメタデータは、オブジェクト・タイプの形式です。オブジェクト・タイプとは、表、レポート、または画像など、ビジネス情報のタイプを反映するために使用されるオブジェクトの種別です。

情報カタログには、事前定義のオブジェクト・タイプ、および情報カタログの管理者が定義したオブジェクト・タイプを含めることができます。事前定義のオブジェクト・タイプとは、定義が情報カタログ・マネージャーの出荷時に備わっているオブジェクト・タイプです。それらのオブジェクトに関する説明は、163ページの『第8章 情報カタログ・マネージャー・オブジェクト・タイプ』を参照してください。

情報カタログ・マネージャー製品のウィンドウまたはタグ言語を使用してオブジェクト・タイプを作成するための情報は、[情報カタログ・マネージャー 管理の手引き](#) を参照してください。

タグ言語ファイルのエクスポート

MDIS を他の製品および **Visual Warehouse 3.1** と共に使用しているユーザーへの注記: MDIS 構成およびプロファイル・ファイルがすでに存在する場合、Visual Warehouse インストール・プログラムはそれらを上書きさせ

ん。しかし、情報カタログ・マネージャーの MDIS 機能を最初に使用する前に、情報カタログ・マネージャー MDIS プロファイルおよび構成ファイル内の情報と既存のファイルとをマージしなければなりません。以下のステップを実行してください。

1. MDIS 環境変数の設定値を調べて、既存の MDIS プロファイル・ファイル (MDISTOOL.PRO) および構成ファイル (MDISTOOL.CFG) を見つけます。
2. テキスト・エディターを使用して、
X:¥VWSLIB¥METADATA¥PROFILES¥MDISTOOL.PRO の内容を既存のプロファイル・ファイルに追加します。(X は情報カタログ・マネージャーのインストール先ドライブです。)
3. テキスト・エディターを使用して、
X:¥VWSLIB¥METADATA¥PROFILES¥MDISTOOL.CFG の内容を既存の構成ファイルに追加します。(X は情報カタログ・マネージャーのインストール先ドライブです。)

MDIS 対応のタグ言語ファイルをエクスポートする

MDIS タグ言語ファイルを情報カタログから直接エクスポートするには、DGUIDE コマンドを MS-DOS コマンド・プロンプトから入力します。コマンド構文に関する以下の規則を守ってください。

- 指定のないすべての部分については、大文字小文字を区別しません。
- 各キーワードの前に、スラッシュ (/) またはハイフン (-) のどちらかが必要です。
- DGUIDE コマンドに続くすべてのキーワードは必須です。 /MDIS_EXPORT キーワードに続くすべてのキーワードは必須です。

```
DGUIDE /USERID userid /PASSWORD password /DGNAME dname /MDIS_EXPORT filename  
/LOGFILE filename /OBJTYPE object_type /OBJECTS name
```

任意指定のキーワードは以下のとおりです。

```
/ADMIN  
/TRACE 0|1|2|3|4
```

たとえば、MDIS メタデータを情報カタログからファイルにエクスポートする場合、以下のコマンドをタイプします (改行は入力しません)。

```
DGUIDE /USERID longods /PASSWORD secret /DGNAME ICMSAMP /ADMIN  
/MDIS_EXPORT c:¥mdis.tag /LOGFILE c:¥mdis.log  
/OBJTYPE database /OBJECTS server01.payroll.valdezma
```

/ADMIN

管理者としてログオンすることを指定します。この任意指定のキーワードを DGUIDE コマンドに指定しない場合、ユーザーとしてログオンしま

す。メタデータをユーザーとしてエクスポートすることは可能ですが、その場合、管理者タスクのすべてを実行することはできません。

/DGNAME

情報カタログ名。

情報カタログがローカルである場合、データベース名を指定します。情報カタログがリモートである場合、それがカタログ化される別名を指定します。

例:

```
/DGNAME ICMSAMP
```

/LOGFILE

情報カタログ・マネージャーが MDIS のインポートまたは MDIS のエクスポートの際に生成するメッセージを送るファイルの宛先を指定します。

ドライブ、パス、およびファイル名を含めた全体を指定するのであれば、情報カタログ・マネージャーはそのファイルを DGWPATH 環境変数で指定されたパスに置きます。固定ドライブを指定しなければなりません。

例:

```
/LOGFILE d:%tagfile.log
```

/MDIS_EXPORT

MDIS 対応のメタデータを指定した名前 MDIS 対応のタグ言語ファイルとしてエクスポートします。ドライブ、パス、およびファイル名を含めた全体を指定するのであれば、情報カタログ・マネージャーはそのファイルを DGWPATH 環境変数で指定されたパスに置きます

例:

```
/MDIS_EXPORT d:%tagfile.tag
```

MDIS メタデータをエクスポートする情報カタログには、MDIS メタデータ以外のメタデータを含めることができます。しかし、/MDIS_EXPORT によってエクスポートされるのは、MDIS に準拠するメタデータだけです。

/OBJECTS

このパラメーターは必須です。

エクスポートしたいオブジェクトを指定します。/OBJTYPE キーワード上に指定したオブジェクト・タイプに応じて、*name* 値はピリオドによって区切られた 3 から 5 のプロパティ値で構成されます。

/OBJTYPE /OBJECTS

Database	<i>ServerName.DatabaseName.OwnerName</i>
Dimension	<i>ServerName.DatabaseName.OwnerName.DimensionName</i>
Subschema	<i>ServerName.DatabaseName.OwnerName.SubschemaName</i>
Record	<i>ServerName.DatabaseName.OwnerName.RecordName</i>
Element	<i>ServerName.DatabaseName. OwnerName. RecordName. ElementName</i>

このリストでは、名前の一部が MDIS 名で表されています。対応する情報カタログ名を見つけるには、データウェアハウスセンター Web サイト (<http://www.software.ibm.com/data/vw/>) で入手可能なデータウェアハウスセンター アプリケーション統合の手引き を参照してください。

/OBJTYPE

これは必須パラメーターです。

エクスポートしたい以下の MDIS オブジェクト・タイプの 1 つを指定します。

Database
Dimension
Subschema
Record
Element

オブジェクト・タイプ名は大文字小文字の区別をしません。

例:

```
/MDIS_EXPORT d:%tagfile.tag /OBJTYPE record
```

/PASSWORD

このユーザー ID に対するパスワード。

例:

```
/PASSWORD secret
```

以下のオペレーティング・システム上にあるデータベースにアクセスするためのパスワードは大文字小文字の区別をするため、指定されたとおりにタイプしなければなりません。

- AIX
- Windows NT および Windows 2000
- Solaris 実行環境版

/TRACE

情報カタログ・マネージャー・トレース・ファイルに送るトレース情報のレベル。高位のレベルには下位のレベルの機能が含まれます (たとえば、3にはレベル 0、1、2、および 3 の機能が含まれます)。情報カタログ・マネージャーの問題を診断するために IBM ソフトウェア・サポートを呼び出すには、高いレベルを指定しなければならない場合があります。

- 0** デフォルト。このレベルには、すべてのメッセージ、警告、エラー、および重大エラー状態が含まれます。
- 1** 最高レベルの情報カタログ・マネージャー機能の入力レコードおよび出力レコードが含まれます。
- 2** 情報カタログ・マネージャー機能の入力レコードおよび出力レコードで、一般的なものが含まれます。
- 3** 入力および出力パラメーター (入力または出力構造を除く) が含まれます。
- 4** 情報カタログ・マネージャーに渡されて使用されたすべての入力または出力構造が含まれます。

/USERID

情報カタログのユーザー ID。情報カタログがあるデータベースに必要なユーザー ID を入力します。たとえば、ユーザー ID はローカル、LAN、AS/400、AIX、または OS/390 TSO ユーザー ID であることがあります。

例:

```
/USERID longods
```

第4章 ユーザーが情報カタログ・マネージャーからプログラムを開始できることを確認する

情報カタログ内にオブジェクトを設定して、ユーザーがアプリケーション・プログラムを実行してオブジェクトに記述された実際の情報を処理できるようにします。ユーザーは使い慣れたアプリケーション・プログラムを実行することができます。それには、情報を作成するときに最初に使用したプログラムも含まれます。

以下の要件に適合していることを確認してください。

- ユーザーには、ワークステーションまたは LAN 上にインストールされた適切なアプリケーション・ソフトウェアが必要です。
- ユーザーは、プログラムのインストール場所に関係なくコマンド行からパスを指定しないコマンド `start program_name` によって開始できる、任意のプログラムを立ち上げることができます。

プログラムの多くは、インストール時にパスをプログラム・レジストリーに書き込みます。 `start` コマンドによって、そのパスが検索されます。プログラムがパスをプログラム・レジストリーに記述しない場合、ユーザーのワークステーションの `path` 環境変数にそのプログラムへのディレクトリー・パスを追加する必要がある場合があります。

- ユーザーは、使用したい情報が保管されているデータベースまたはファイル・システムに対する必要な権限を持っていないければなりません。
- 情報カタログ内のプログラム・オブジェクトには、ユーザーがプログラムを実行するオペレーティング・システムのための適切な呼び出し構文が含まれていないければなりません。

情報カタログ・マネージャー (Web 版) ユーザーのための追加要件

情報カタログ・マネージャー (Web 版) ユーザーのための Web 環境を設定するとき、以下の要件に適合していることを確認してください。

- ユーザーがアプリケーション・プログラムで使用したいデータは、Web サーバーからアクセス可能でなければなりません。たとえば、情報カタログ・マネージャーのサンプル・データ・ファイルは Web サーバー上のディレクトリーにあります。

ユーザーが情報カタログ・マネージャーからプログラムを開始できることを確認する

- ユーザーが開始したいプログラムは、Web クライアント上にインストールされていなければなりません。

たとえば、ユーザーが Lotus® 1-2-3 ファイル® にアクセスする場合、Lotus 1-2-3 は Web クライアント上にインストールされていなければなりません。

アプリケーション・プログラムが Java™ アプレットである場合、そのアプリケーションはインストールされている必要はありません。そのアプリケーションに Web ブラウザーから直接アクセスできるからです。

クライアントには、必要なブラウザー・プラグイン・プログラムもなければなりません。情報カタログ・マネージャー (Web 版) サーバーは、プラグイン・プログラムによって使用される関連ファイルを見つけることが可能でなければなりません。たとえば、ユーザーが Adobe Acrobat ファイルを表示したい場合、Acrobat Reader 用のブラウザー・プラグイン・ファイルを情報カタログ・マネージャー (Web 版) クライアントにインストールしておく必要があります。情報カタログ・マネージャー (Web 版) サーバーは、ユーザーが表示したいファイルを見つけてクライアントにダウンロードすることが可能でなければなりません。

- ユーザーが開始するアプリケーション・プログラムに必要な MIME タイプは、Web サーバーの構成ファイルで識別されていなければなりません。ユーザーが開始したいプログラムのファイル拡張子を指定した AddType 指示を、構成ファイルに含めなければなりません。たとえば、ユーザーが Lotus 1-2-3 スプレッドシートをファイル・タイプ WK4 で使用したい場合、Lotus Domino™ Go Webserver の AddType 指示を以下のように定義します。

```
AddType .WK4 application/x-lotus1-2-3 binary
```

ユーザーが Lotus Domino Go Webserver 以外の Web サーバーを使用している場合、MIME タイプの定義はこれとは異なります。詳細については、ご使用の Web サーバーのマニュアルを参照してください。

- Websphere IBM HTTP WebServer を使用している場合、MIME タイプは `¥conf¥mime.types` ファイル内で以下の例のように定義されます。

```
application/vnd.lotus-1-2-3 wks 123 wk1 wk2 wk3 wk4
```

- Netscape Navigator の一部のバージョンでは、支援プログラムがファイル・タイプを識別して対応するアプリケーション・プログラムを開始します。Microsoft Internet Explorer は、支援プログラムを使用しません。その代わりに、Internet Explorer は Windows Explorer によって使用されるファイル・タイプとプログラムとの関連を使用します。ファイル・タイプを識別するように Internet Explorer を設定する必要はありません。

ユーザーが情報カタログ・マネージャーからプログラムを開始できることを確認する

- ユーザーがプログラムを開始したいオブジェクトについて、「**データにアクセスするための URL (URL to access data)**」プロパティを定義しなければなりません。プロパティの値は、プログラムを直接立ち上げるためのリンクです。

情報カタログ・マネージャー (Web 版) オブジェクトからプログラムを開始するには、以下のように行います。

1. リスト・ペインで、プログラムを開始したいオブジェクトをクリックします。

記述ペイン内にオブジェクト記述ページがオープンします。

2. 「**データにアクセスするための URL (URL to access data)**」プロパティを見つけます。
3. プロパティ値をクリックします。

Web browser は、プロパティ値によって指定された Web アドレスを使用して立ち上げられます。

ユーザーが情報カタログ・マネージャーからプログラムを開始できることを確認する

第2部 メタデータのリファレンス

第5章 メタデータ・テンプレート

この章では、データウェアハウスセンターおよび情報カタログ・マネージャーに備わっている各テンプレートについての詳しい情報を示します。各テンプレートのセクションには、そのテンプレートのトークンがリストされています。そこには、各トークンに許可される値、および値の長さが示されています。

ご使用の交換プログラムにトークンの値がない場合、トークンを `ISV_DEFAULTVALUE` に設定してください。ただし、必須のトークンには `ISV_DEFAULTVALUE` 以外の値を指定しなければなりません。

セキュリティー・グループのテンプレートは存在しないので、ご使用のプログラムでは値 `ISV_DEFAULTSECURITYGROUP` を `*SecurityGroup` トークンのインスタンスに指定しなければなりません。

テンプレートがデータウェアハウスセンター パラメーターを設定しない場合、データウェアハウスセンター定義はパラメーターのデフォルトの値を持つこととなります。たとえば、データウェアハウスセンターはソース・データベースの「再試行回数 (Retry Count)」および「再試行間隔 (Retry Interval)」パラメーターをデフォルト値に設定します。

表13 には、データウェアハウスセンターに備わっているメタデータ・テンプレート、および各テンプレートについて解説するセクションがリストされています。

表 13. データウェアハウスセンターに備わっているメタデータ・テンプレート

テンプレート	説明	参照先:
AgentSite.tag	エージェントがデータ・ソースまたはターゲット・ウェアハウスにアクセスする、またはデータウェアハウスセンター プログラムが実行される、エージェント・サイトを定義します。	61ページの『AgentSite.tag』
Column.tag	表、セグメント、またはファイル内の列またはフィールドを定義します。	64ページの『Column.tag』

表 13. データウェアハウスセンターに備わっているメタデータ・テンプレート (続き)

テンプレート	説明	参照先:
HeaderInfo.tag	データウェアハウスセンターがタグ言語ファイルを宣言するために必要なすべてのオブジェクト・タイプ定義を宣言します。	70ページの『HeaderInfo.tag』
Process.tag	プロセスを定義します。	71ページの『Process.tag』
StarSchema.tag	スタースキーマを定義します。	73ページの『StarSchema.tag』
StarSchemaInputTable.tag	表とスタースキーマとの間の関係を定義します。	74ページの『StarSchemaInputTable.tag』
Step.tag	ステップを定義します。	76ページの『Step.tag』
StepCascade.tag	ステップ間の連鎖関係を定義します。	80ページの『StepCascade.tag』
StepInputTable.tag	ステップとそのソース表との間の関係を定義します。	81ページの『StepInputTable.tag』
StepOutputTable.tag	ステップとそのターゲットとの間の関係を定義します。	83ページの『StepOutputTable.tag』
StepVWPOutputTable.tag	ステップとウェアハウス・ターゲットとの間の関係を定義します。	84ページの『StepVWPOutputTable.tag』
StepVWPProgramInstance.tag	ステップによって使用される特定のインスタンスを定義します。	85ページの『StepVWPProgramInstance.tag』
SourceDataBase.tag	ウェアハウス・ソースを定義します。	87ページの『SourceDataBase.tag』
SubjectArea.tag	作成されているプロセスおよびステップを含むサブジェクト・エリアを定義します。	90ページの『SubjectArea.tag』
Table.tag	データウェアハウスセンターのアクセス先となる表またはファイルを定義します。	92ページの『Table.tag』

表 13. データウェアハウスセンターに備わっているメタデータ・テンプレート (続き)

テンプレート	説明	参照先:
VWPGroup.tag	定義されているデータウェアハウスセンター プログラムを含むことになるグループを定義します。	99ページの『VWPGroup.tag』
VWPProgramInstanceParameter.tag	データウェアハウスセンターが特定のステップによって使用されるデータウェアハウスセンター プログラムのインスタンスに渡すパラメーターを、追加または変更します。	100ページの『VWPProgramInstanceParameter.tag』
VWPProgramTemplate.tag	データウェアハウスセンター プログラムを定義します。	103ページの『VWPProgramTemplate.tag』
VWPProgramTemplateParameter.tag	データウェアハウスセンターがデータウェアハウスセンター プログラムに渡すプログラムを定義します。	106ページの『VWPProgramTemplateParameter.tag』
WarehouseDataBase.tag	ウェアハウス・ターゲットを定義します。	109ページの『WarehouseDataBase.tag』

AgentSite.tag

このテンプレートは、以下のエージェント・サイトの定義に使用します。

- エージェントがデータ・ソースまたはターゲット・ウェアハウスにアクセスするエージェント・サイト。
- データウェアハウスセンター プログラムが実行されるエージェント・サイト。

以下のエージェント・サイト・タイプの 1 つを使用できます。

- ウェアハウス・コントロール・データベース内ですでに定義されているエージェント・サイト。

既存のエージェント・サイトを使用するためには、`*AgentSite` トークンのすべての出現をエージェント・サイト名に置き換えます。

- デフォルトのエージェント・サイト。

AgentSite.tag

デフォルトのエージェント・サイトを使用するためには、`*AgentSite` トークンのすべての出現を `ISV_DEFAULTAGENTSITE` に置き換えます。

- `AgentSite.tag` テンプレートを使用して定義した新規のエージェント・サイト。

新規のエージェント・サイトを定義するには、`AgentSite.tag` テンプレート内のトークンの値を定義します。`*AgentSite` トークンのすべての出現を、新規のエージェント・サイト名に置き換えます。

トークン

表14 は、このテンプレート内の各トークンに関する情報を示しています。

表 14. `AgentSite.tag` のトークン

トークン	説明	許可される値
エンティティ・パラメーター		
<code>*AgentSite</code>	<p>新規のエージェント・サイトの名前、またはエージェントが新規でない場合はデフォルトのエージェント・サイトの名前。</p> <p>新規の名前を指定する場合、ウェアハウス・コントロール・データベース内で固有の名前にしなければなりません。</p> <p>このトークンは必須ですが、デフォルトのエージェント・サイト <code>ISV_DEFAULTAGENTSITE</code> を指定することもできます。</p>	<p>最長 80 バイトのテキスト・ストリング。</p> <p>新規のエージェント・サイトを作成したくない場合、デフォルトのエージェント・サイトとして <code>ISV_DEFAULTAGENTSITE</code> を使用してください。</p>
<code>*AgentSiteContact</code>	このエージェントに責任を持つ人または組織の名前。	テキスト・ストリング。
<code>*AgentSiteDescription</code>	<p>エージェント・サイトについての短い説明。</p> <p>このトークンは任意指定です。</p>	最長 254 バイトのテキスト・ストリング。

表 14. AgentSite.tag のトークン (続き)

トークン	説明	許可される値
*AgentSiteNotes	エージェント・サイトについての長い説明。 このトークンは任意指定です。	最長 32700 バイトのテキスト・ストリング。
*AgentSiteOSType	エージェント・サイト上で実行するオペレーティング・システムのタイプ。 このトークンは必須です。	以下の値の 1 つ: ISV_windowsNT Windows NT [®] ISV_AIX AIX [®] ISV_os2 OS/2 [®] ISV_as400 AS/400 [®] ISV_Solaris SUN ISV_MVS MVS
*AgentSiteTCP/IPHostname	エージェント・サイトの TCP/IP ホスト名。 このトークンは必須です。	最長 200 バイトのテキスト・ストリング。
*AgentSiteUserid	その下でエージェントが実行されるユーザー ID。 このトークンは必須です。	最長 36 バイトのテキスト・ストリング。
関係パラメーター		
*CurrentCheckPointID++	0 から始まり、トークンに置換されるたびに増加する索引。 このトークンは必須です。	数値。

AgentSite.tag

値の例

表15 は、各トークンに指定するメタデータの種類を例示するために、各トークンの値の例を示しています。

表 15. AgentSite.tag トークンの値の例

トークン	値の例
*AgentSite	My agent site
*AgentSiteContact	DEPT W24A
*AgentSiteDescription	This is the description of my agent site
*AgentSiteNotes	These are the notes for my agent site.
*AgentSiteOSType	ISV_Solaris
*AgentSiteTCP/IPHostname	CHI11W71.stl.ibm.com
*AgentSiteUserid	VWADMIN
*CurrentCheckPointID++	1

Column.tag

このテンプレートを使用して、表内の列、またはセグメントかファイル内のフィールドを定義します。このテンプレートを使用して、ソースおよびターゲットの両方の列またはフィールドを定義できます。

このテンプレートは、列またはフィールドと、その列またはフィールドを含む表、セグメント、またはファイルとの間の関係を定義します。 Table.tag テンプレートを使用してソースまたはターゲットを定義した場合、このテンプレートを含めなければなりません (92ページを参照)。

トークン

65ページの表16 は、このテンプレート内の各トークンに関する情報を示しています。

表 16. Column.tag トークン

トークン	説明	許可される値
エンティティ・パラメーター		
*ColumnName	列またはフィールドの名前。 その名前は、表またはフィールド内で固有でなければなりません。 このトークンは必須です。	最長 80 バイトのテキスト・ストリング。
*ColumnDescription	列またはフィールドについての短い説明。 このトークンは任意指定です。	最長 254 バイトのテキスト・ストリング。
*ColumnNotes	列またはフィールドについての長い説明。 このトークンは任意指定です。	最長 32700 バイトのテキスト・ストリング。
*ColumnOffsetFromZero	ファイルの先頭からこのフィールドのデータが開始する位置までのバイト数で表したオフセット。	数値または 0。
*ColumnOrdinalNumber	列の位置順序。通常は *ColumnPositionNumber と同じです。	数値または 0。
*ColumnUserActions	この列またはフィールド上でユーザーが実行できるアクション。 このトークンは任意指定です。	最長 254 バイトのテキスト・ストリング。
*ColumnLength	作成されている列またはフィールドの長さ。 このトークンは必須です。	数値。

Column.tag

表 16. Column.tag トークン (続き)

トークン	説明	許可される値
*ColumnPrecision	10 進データ・タイプによる列またはフィールドの、列またはフィールド精度。 このトークンは必須です。	数値または 0。
*ColumnKeyPosition	この列がキーの一部である場合、キー内の列の位置。 このトークンは必須です。	数値。精度の値がない場合、0 を指定します。
*ColumnPositionNumber	行内での列の順位を示す、0 から始まる数値。 このトークンは必須です。	数値。
*ColumnAllowsNulls	列またはフィールドにヌル・データを入れられるかどうかを示すフラグ。 このトークンは必須です。	以下の値の 1 つ: ISV_NULLSYES カラムにはヌル・データを入れることができます。 ISV_NULLSNO カラムにはヌル・データを入れることができません。
*ColumnDataIsText	列またはフィールドが文字タイプの場合、入れられるのがテキスト・データだけかどうかを示すフラグ。 このトークンは必須です。	以下の値の 1 つ: ISV_IStEXTYES カラムにはテキスト・データだけが含まれます。 ISV_IStEXTNO カラムにはテキスト以外のデータも含まれます。
*ColumnEditionType	列にデータウェアハウスセンターのエディション情報が含まれるかどうかを示します。	以下の値の 1 つ: ISV_ColumnIsEditionColumn カラムはエディション・カラムです。 ISV_ColumnIsNormal カラムは通常のカラムです。

表 16. Column.tag トークン (続き)

トークン	説明	許可される値
*ColumnNativeDataType	データベース・マネージャ またはファイル・システムに定義された、列または フィールドのデータ・タイ プ。 このトークンは必須です。	以下の値の 1 つ: ISV_NATIVE_CHAR ISV_NATIVE_VARCHAR ISV_NATIVE_LONGVARCHAR ISV_NATIVE_VARCHAR2 ISV_NATIVE_GRAPHIC ISV_NATIVE_VARGRAPHIC ISV_NATIVE_LONGVARGRAPHIC ISV_NATIVE_CLOB ISV_NATIVE_INT ISV_NATIVE_TINYINT ISV_NATIVE_BLOB ISV_NATIVE_SMALLINT ISV_NATIVE_INTEGER ISV_NATIVE_FLOAT ISV_NATIVE_SMALLFLOAT ISV_NATIVE_DOUBLE ISV_NATIVE_REAL ISV_NATIVE_DECIMAL ISV_NATIVE_SMALLMONEY ISV_NATIVE_MONEY ISV_NATIVE_NUMBER

Column.tag

表 16. *Column.tag* トークン (続き)

トークン	説明	許可される値
* <i>ColumnNativeDataType</i> (続き)	データベース・マネージャーまたはファイル・システムに定義された、列またはフィールドのデータ・タイプ。 このトークンは必須です。	以下の値の 1 つ: ISV_NATIVE_NUMERIC ISV_NATIVE_DATE ISV_NATIVE_TIME ISV_NATIVE_TIMESTAMP ISV_NATIVE_LONG ISV_NATIVE_RAW ISV_NATIVE_LONGRAW ISV_NATIVE_DATETIME ISV_NATIVE_SMALLDATETIME ISV_NATIVE_SYSNAME ISV_NATIVE_TEXT ISV_NATIVE_BINARY ISV_NATIVE_VARBINARY ISV_NATIVE_LONGVARBINARY ISV_NATIVE_BIT ISV_NATIVE_IMAGE ISV_NATIVE_SERIAL ISV_NATIVE_DBCLOB ISV_NATIVE_BIGINT ISV_NATIVE_DATETIMEYEARTOFRACTION
関係パラメーター		
* <i>CurrentCheckPointID++</i>	0 から始まり、トークンに置換されるたびに増加する索引。 このトークンは必須です。	数値。
* <i>DatabaseName</i>	ウェアハウス・ソースまたはウェアハウス・ターゲットのビジネス上の名前。 このトークンは必須です。	最長 40 バイトのテキスト・ストリング。

表 16. Column.tag トークン (続き)

トークン	説明	許可される値
*TablePhysicalName	データベース・マネージャ またはファイル・システムに定義された、列を含む 表またはファイルの物理 名。 このトークンは必須です。	最長 80 バイトのテキスト・ストリング。
*TableOwner	列を含む表の所有者、高レ ベル修飾子、集合、または スキーマ。 このトークンは必須です。	最長 15 バイトのテキスト・ストリング。
*CurrentCheckPointID++	0 から始まり、トークンに 置換されるたびに増加する 索引。 このトークンは必須です。	数値。

値の例

表17 は、各トークンに指定するメタデータの種類を例示するために、各トークンの値の例を示しています。

表 17. Column.tag トークンの値の例

トークン	値の例
*ColumnName	Country_code
*ColumnDescription	This column contains the country code
*ColumnNotes	The valid values for this column can be found in the Geography reference manual
*ColumnOffsetFromZero	0
*ColumnOrdinalNumber	0
*ColumnUserActions	User cannot directly view a single column
*ColumnLength	10
*ColumnPrecision	0
*ColumnKeyPosition	0
*ColumnAllowsNulls	ISV_NULLSNO
*ColumnDataIsText	ISV_ISTEXTYES

Column.tag

表 17. *Column.tag* トークンの値の例 (続き)

トークン	値の例
<i>*ColumnNativeDataType</i>	ISV_NATIVE_CHAR
<i>*DatabasePhysicalName</i>	FINANCE
<i>*TableOwner</i>	DB2ADMIN
<i>*TablePhysicalName</i>	GEOGRAPHY
<i>*CurrentCheckPointID++</i>	8

HeaderInfo.tag

このテンプレートを使用して、データウェアハウスセンターがタグ言語ファイルを処理するために必要なすべてのオブジェクト・タイプ定義を宣言します。このテンプレートには、インポート中のオブジェクトを含むことになるセキュリティ・グループなど、データウェアハウスセンターが他の定義と関連付ける定義も含まれます。このテンプレートは常に必須で、タグ言語ファイルの先頭に存在しなければなりません。

トークン

表18 は、このテンプレート内の各トークンに関する情報を示しています。

表 18. *HeaderInfo.tag* のトークン. このテンプレートには、関係パラメーターだけが含まれます。

トークン	説明	許可される値
<i>*SecurityGroup</i>	インポート中のすべてのオブジェクトを含むことになるセキュリティ・グループ。 このトークンは必須であり、デフォルトのセキュリティ・グループを指定しなければなりません。	デフォルトのセキュリティ・グループとして ISV_DEFAULTSECURITYGROUP。
<i>*CurrentCheckPointID++</i>	0 から始まり、トークンに置換されるたびに増加する索引。 このトークンは必須です。	数値。

値の例

表19 は、各トークンに指定するメタデータの種類を例示するために、各トークンの値の例を示しています。

表19. Header.tag トークンの値の例

トークン	値の例
*SecurityGroup	ISV_DEFAULTSECURITYGROUP
*CurrentCheckpointID++	0

Process.tag

このテンプレートは、ステップをグループ化するために使用します。各ステップは、1 つのプロセスだけに含まれていなければなりません。プロセスはサブジェクト・エリアに関連付けられて、各パートナー・アプリケーションにはプロセスが常駐する少なくとも 1 つのサブジェクト・エリアがなければなりません。このテンプレートは、サブジェクト・エリアとパートナー・アプリケーションのセキュリティー・グループとの間、およびプロセスとサブジェクト・エリアとの間の関係を定義します。

パートナー・アプリケーションがデータウェアハウスセンターへのステップを定義しているとき、このテンプレートは必須です。

新規のプロセス・オブジェクトを作成する場合、*ProcessName トークンに指定する値は、ウェアハウス・コントロール・データベース内に定義されたすべてのプロセスに対して固有でなければなりません。

トークン

表20 は、このテンプレート内の各トークンに関する情報を示しています。

表20. Process.tag のトークン. このテンプレートには、関係パラメーターだけが含まれます。

トークン	説明	許可される値
エンティティ・パラメーター		
*ProcessName	プロセスの固有名。	最長 80 バイトのテキスト・ストリング。
*ProcessDescription	プロセスに関連した説明。	最長 254 バイトのテキスト・ストリング。
*ProcessNotes	プロセスに関連した長い説明。	最長 32,700 バイトのテキスト・ストリング。

Process.tag

表 20. *Process.tag* のトークン (続き). このテンプレートには、関係パラメーターだけが含まれます。

トークン	説明	許可される値
<i>*ProcessContact</i>	このステップに関する質問 または考慮事項についての 連絡先となる個人またはグ ループの名前。	テキスト・ストリング。
<i>*ProcessType</i>	ソース・データが存在しな い場合の処理オプション。	以下の値の 1 つ: ISV_ProcessType_Normal プロセスは通常のユーザー・プロ セスです。
関係パラメーター		
<i>*SubjectArea</i>	作成中またはこのプロセス に追加中のプロセスおよび ステップを含むことになる サブジェクト・エリアの名 前。	最長 80 バイトのテキスト・ストリング。
<i>*SecurityGroup</i>	インポート中のすべてのオ ブジェクトを含むことにな るセキュリティ・グルー プ。 このトークンは必須であ り、デフォルトのセキュリ ティ・グループを指定し なければなりません。	デフォルトのセキュリティ・グループと して ISV_DEFAULTSECURITYGROUP。
<i>*CurrentCheckPointID</i>	0 から始まり、トークンに置 換されるたびに増加する索 引。 このトークンは必須です。	数値。

値の例

表21 は、各トークンに指定するメタデータの種類を例示するために、各トークンの値の例を示しています。

表 21. *Process.tag* トークンの値の例

トークン	値の例
<i>*ProcessName</i>	Marketing process

表 21. Process.tag トークンの値の例 (続き)

トークン	値の例
*ProcessDescription	A collection of steps that is used by the marketing organization
*ProcessNotes	Steps that create the star schema that is used by the marketing organization
*ProcessContact	Marketing
*ProcessType	ISV_ProcessType_2
*SubjectArea	Group of processes generated for this partner application
*SecurityGroup	ISV_DEFAULTSECURITYGROUP
*CurrentCheckPointID	9

StarSchema.tag

このテンプレートを使用して、スタースキーマに関連したテーブルをグループ化するメカニズムとして定義します。スタースキーマは、(DB2 OLAP Integration Server によってさらに使用するために) 同一の物理データベース内の表を関連付けるため、または複数のデータベースにある表を関連付けて論理グループ化を行うために使用できます。

トークン

表 22 は、このテンプレート内の各トークンに関する情報を示しています。

表 22. StarSchema.tag のトークン

トークン	説明	許可される値
エンティティ・パラメーター		
*StarSchemaName	作成中または関連付け中のスタースキーマの固有名。	最長 80 バイトのテキスト・ストリング。
*StarSchemaDescription	スタースキーマに関連した説明。	最長 254 バイトのテキスト・ストリング。
*StarSchemaNotes	ステップに関連した長い説明。	最長 32,700 バイトのテキスト・ストリング。
*StarSchemaContact	このステップに関する質問または考慮事項についての連絡先となる個人またはグループの名前。	テキスト・ストリング。

StarSchema.tag

表 22. *StarSchema.tag* のトークン (続き)

トークン	説明	許可される値
<i>*StarSchemaDBName</i>	作成中のデータベースのビジネス名。	テキスト・ストリング。
関係パラメーター		
<i>*CurrentCheckPointID++</i>	0 から始まり、トークンに置換されるたびに増加する索引。 このトークンは必須です。	数値。

値の例

表23 は、各トークンに指定するメタデータの種類を例示するために、各トークンの値の例を示しています。

表 23. *StarSchema.tag* トークンの値の例

トークン	値の例
<i>*StarSchemaName</i>	Marketing schema
<i>*StarSchemaDescription</i>	This star schema represents the marketing division's internal databases
<i>*StarSchemaNotes</i>	Tables used for the marketing division
<i>*StarSchemaContact</i>	Marketing group
<i>*StarSchemaDBName</i>	Marketing
<i>*CurrentCheckPointID++</i>	3

StarSchemaInputTable.tag

このテンプレートを使用して、スタースキーマとその入力ソースとの間の関係を定義します。この関係はすべてのスタースキーマに必須です。

トークン

75ページの表24 は、このテンプレート内の各トークンに関する情報を示しています。

表 24. SourceDataBase.tag のトークン

トークン	説明	許可される値
エンティティ・パラメーター		
*StarSchemaName	作成中または関連付け中のデータベーススキーマの名前。	テキスト・ストリング。
関係パラメーター		
*DatabaseName	作成中のデータベースのビジネス名。	テキスト・ストリング。
*TableOwner	説明されている表の所有者、高レベル修飾子、集合、またはスキーマ。 この値は ODBC の規則に定義されている有効な修飾子でなければなりません。	テキスト・ストリング。
*TablePhysicalName	ODBC に知られている物理表名 (システム DSN 名)。	テキスト・ストリング。
*CurrentCheckPointID++	0 から始まり、トークンに置換されるたびに増加する索引。 このトークンは必須です。	数値。

値の例

表25 は、各トークンに指定するメタデータの種類を例示するために、各トークンの値の例を示しています。

表 25. SourceDataBase.tag トークンの値の例

トークン	値の例
*StarSchemaName	Finance schema
*DatabaseName	Finance Warehouse
*TableOwner	DB2ADMIN
*TablePhysicalName	DB2ADMIN.GEOGRAPHY
*CurrentCheckPointID++	7

Step.tag

このテンプレートを使用して、データウェアハウスセンターが管理するステップを定義します。このテンプレートには、セキュリティ・グループ、プロセス、およびエージェントへの関係についての情報が含まれています。

このテンプレートは、ソースおよびターゲット・データとの間の関係を生成している、またはデータウェアハウスセンターが実行するプログラムを定義している、すべてのパートナー・アプリケーションで必須です。

新規のステップ・オブジェクトを作成する場合、*StepName トークンに指定する値は、ウェアハウス・コントロール・データベース内に定義されたすべてのステップに対して固有でなければなりません。

トークン

表26 は、このテンプレート内の各トークンに関する情報を示しています。

表 26. Step.tag のトークン

トークン	説明	許可される値
エンティティ・パラメーター		
*StepName	作成中または関連付け中のステップの固有名。	最長 80 バイトのテキスト・ストリング。
*StepDescription	ステップに関連した説明。	最長 254 バイトのテキスト・ストリング。
*StepNotes	ステップに関連した長い説明。	最長 32,700 バイトのテキスト・ストリング。
*StepDataNotPresent	ソース・データが存在しない場合の処理オプション。	以下の値の 1 つ: ISV_StepDataNotPresent_OK データが存在しない場合、処理を続けます。 ISV_StepDataNotPresent_Warning データが存在しない場合、警告を発行して処理を続けます。 ISV_StepDataNotPresent_Error データが存在しない場合、エラー・メッセージを発行して処理を停止します。

表 26. Step.tag のトークン (続き)

トークン	説明	許可される値
<i>*StepSelectStatementGenerated</i>	データウェアハウスセンターが SQL を生成するかどうか、または SQL ステートメントがトークン <i>*StepSelectStatement</i> によって含まれるかどうかを指定するフラグ。	以下の値の 1 つ: ISV_StepSelectStatementNo SELECT ステートメントは生成されませんが、 <i>*StepSelectStatement</i> に含まれています。 ISV_StepSelectStatementYes SELECT ステートメントが生成されて、 <i>*StepSelectStatement</i> は無視されます。
<i>*StepSelectStatement</i>	SQL ステートメントは ISV_StepSelectStatementNo であれば発行されます。	SQL ストリング。
<i>*StepContact</i>	このステップに関する質問または考慮事項についての連絡先となる個人またはグループの名前。	テキスト・ストリング。
<i>*StepExternalPopulation</i>	ステップがデータウェアハウスセンター環境の外側で実行される予定かどうかを示すフラグ。	以下の値の 1 つ: ISV_StepExternalNo 表には他の方法で外部的なデータが取り込まれることはありません。 ISV_StepExternalYes 表には他の方法で外部的なデータが取り込まれます。

Step.tag

表 26. Step.tag のトークン (続き)

トークン	説明	許可される値
*StepType	作成中のステップのタイプ。	以下の値の 1 つ: ISV_StepType_Editioned_Append ステップが実行されると、表内のデータは追加されます。 ISV_StepType_Full_Replace ステップが実行されると、表内のデータは置換されます。 ISV_StepType_Uneditioned_Append ステップが実行されると、表内のデータは追加されます。 ISV_StepType_VWP_Population 表内のデータはデータウェアハウスセンター プログラムによって取り込まれます。
*StepSQLWarning	SQL 警告が生じた場合の処理オプション。	以下の値の 1 つ: ISV_StepSQLWarning_OK SQL 警告が生じた場合、処理を続けます。 ISV_StepSQLWarning_Warning SQL 警告が生じた場合、警告を発行して処理を続けます。 ISV_StepSQLWarning_Error SQL 警告が生じた場合、エラーを発行して処理を停止します。
*StepCommit	*StepCommitAfterNumberRows がステップのターゲット表に挿入された後、データウェアハウスセンターが断続的にコミットするかどうかを指定するフラグ。	以下の値の 1 つ: ISV_Step_Incremental_Commit_On データはターゲットにおいて断続的にコミットされます。 ISV_Step_Incremental_Commit_Off データはターゲットにおいて断続的にコミットされません。

表 26. Step.tag のトークン (続き)

トークン	説明	許可される値
*StepCommitAfterNumberRows	コミットの前に挿入される行数。	数値。
関係パラメーター		
*SecurityGroup	インポート中のすべてのオブジェクトを含むことになるセキュリティ・グループ。	デフォルトのセキュリティ・グループとして ISV_DEFAULTSECURITYGROUP。
	このトークンは必須であり、デフォルトのセキュリティ・グループを指定しなければなりません。	
*ProcessName	プロセスの名前。	最長 80 バイトのテキスト・ストリング。
	このトークンは必須です。	
*AgentSite	新規のエージェント・サイトの名前、またはエージェントが新規でない場合はデフォルトのエージェント・サイトの名前。	最長 80 バイトのテキスト・ストリング。
	新規の名前を指定する場合、データウェアハウスセンターコントロール・データベース内で固有の名前にしなければなりません。	新規のエージェント・サイトを作成したくない場合、デフォルトのエージェント・サイトとして ISV_DEFAULTAGENTSITE を使用してください。
	このトークンは必須ですが、デフォルトのエージェント・サイト ISV_DEFAULTAGENTSITE を指定することもできます。	
*CurrentCheckPointID++	0 から始まり、トークンに置換されるたびに増加する索引。	数値。
	このトークンは必須です。	

値の例

80ページの表27 は、各トークンに指定するメタデータの種類を例示するために、各トークンの値の例を示しています。

Step.tag

表 27. Step.tag トークンの値の例

トークン	値の例
*StepName	Revenue by location
*StepDescription	This step will pull data to create the revenue for each location in a DB2 table
*StepNotes	Revenue for Geography 7 comes from 4 source Oracle tables
*StepDataNotPresent	ISV_StepDataNotPresent_Error
*StepSelectStatementGenerated	ISV_StepSelectStatementNo
*StepSelectStatement	SELECT * FROM IWH.REVENUE_BY_LOCATION
*StepContact	Jason Smythe
*StepExternalPopulation	ISV_StepExternalNo
*StepType	ISV_StepType_Full_Replace
*StepSQLWarning	ISV_StepSQLWarning_Warning
*StepCommit	ISV_Step_Incremental_Commit_On
*StepCommitAfterNumberRows	10000
*SecurityGroup	ISV_DEFAULTSECURITYGROUP
*ProcessName	Marketing process
*AgentSite	My agent site
*CurrentCheckpointID++	5

StepCascade.tag

このテンプレートを使用して 2 つのステップ間の関係を定義し、指定されたステップが完了すると別のステップが開始するようにします。

このテンプレートが必須なのは、パートナー・アプリケーションが複数のステップを連鎖関係に定義している場合だけです。

トークン

表28 は、このテンプレート内の各トークンに関する情報を示しています。

表 28. StepCascade.tag のトークン

トークン	説明	許可される値
	エンティティ・パラメーター	
*StepName	関連付け中のステップの名前。	テキスト・ストリング。

表 28. StepCascade.tag のトークン (続き)

トークン	説明	許可される値
*PostStepName	他のステップが完了すると開始されるステップの名前。	テキスト・ストリング。
関係パラメーター		
*CurrentCheckPointID++	0 から始まり、トークンに置換されるたびに増加する索引。 このトークンは必須です。	数値。

値の例

表29 は、各トークンに指定するメタデータの種類を例示するために、トークンの値の例を示しています。

表 29. StepCascade.tag トークンの値の例

トークン	値の例
*StepName	Revenue by location
*PostStepName	Revenue for all Geographies
*CurrentCheckPointID++	12

StepInputTable.tag

このテンプレートは、ステップとその入力ソースとの間の関係を定義します。

この関係は、ISV_StepType_Editioned_Append、 ISV_StepType_Full_Replace、および ISV_StepType_Uneditioned_Append のタイプのステップでは必須です。

この関係は、ISV_StepType_VWP_Population のタイプのステップでは任意指定です。

ISV は、テンプレートを入力ソースの固有のインスタンスごとに再使用することによって、複数の入力ソースをステップに関連付けることができます。

トークン

82ページの表30 は、このテンプレート内の各トークンに関する情報を示しています。

StepInputTable.tag

表 30. StepInputTable.tag のトークン

トークン	説明	許可される値
エンティティ・パラメーター		
*StepName	関連付け中のステップの名前。	テキスト・ストリング。
関係パラメーター		
*DatabaseName	作成中のデータベースのビジネス名。	テキスト・ストリング。
*TableOwner	説明されている表の所有者、高レベル修飾子、集合、またはスキーマ。 この値は ODBC の規則に定義されている有効な修飾子でなければなりません。	テキスト・ストリング。
*TablePhysicalName	ODBC に知られている物理表名 (システム DSN 名)。	テキスト・ストリング。
*ProcessName	関連付け中のプロセスの名前。	テキスト・ストリング。
*CurrentCheckPointID++	0 から始まり、トークンに置換されるたびに増加する索引。 このトークンは必須です。	数値。

値の例

表31 は、各トークンに指定するメタデータの種類を例示するために、各トークンの値の例を示しています。

表 31. StepInputTable.tag トークンの値の例

トークン	値の例
*StepName	Revenue by product
*DatabaseName	Finance Warehouse
*TableOwner	FINADMIN
*TablePhysicalName	INVENTORY
*ProcessName	Inventory process
*CurrentCheckPointID++	6

StepOutputTable.tag

このテンプレートを使用して、ステップとその出力ターゲットとの間の関係を定義します。

この関係は、ISV_StepType_Editioned_Append、ISV_StepType_Full_Replace、および ISV_StepType_Uneditioned_Append のタイプのステップでは必須です。

トークン

表32 は、このテンプレート内の各トークンに関する情報を示しています。

表 32. SourceDataBase.tag のトークン

トークン	説明	許可される値
エンティティ・パラメーター		
*StepName	作成中または関連付け中のステップの名前。	テキスト・ストリング。
関係パラメーター		
*DatabaseName	関連付け中のデータベースのビジネス名。	テキスト・ストリング。
*TableOwner	説明されている表の所有者、高レベル修飾子、集合、またはスキーマ。	テキスト・ストリング。
	この値は ODBC の規則に定義されている有効な修飾子でなければなりません。	
*TablePhysicalName	ODBC に知られている物理表名 (システム DSN 名)。	テキスト・ストリング。
*ProcessName	関連付け中のプロセスの名前。	テキスト・ストリング。
*CurrentCheckPointID++	0 から始まり、トークンに置換されるたびに増加する索引。	数値。
	このトークンは必須です。	

値の例

84ページの表33 は、各トークンに指定するメタデータの種類を例示するために、各トークンの値の例を示しています。

StepOutputTable.tag

表 33. StepOutputTable.tag トークンの値の例

トークン	値の例
*StepName	Revenue by product
*DatabaseName	Finance Warehouse
*TableOwner	FINADMIN
*TablePhysicalName	INVENTORY
*ProcessName	Marketing process
*CurrentCheckpointID++	4

StepVWPOutputTable.tag

このテンプレートを使用して、ISV_StepType_VWP_Population タイプのステップとその出力ターゲットとの間の関係を任意選択で定義します。

トークン

表34 は、このテンプレート内の各トークンに関する情報を示しています。

表 34. StepVWPOutputTable.tag のトークン

トークン	説明	許可される値
エンティティ・パラメーター		
*StepName	関連付け中のステップの名前。	テキスト・ストリング。
関係パラメーター		
*DatabaseName	作成中のデータベースのビジネス名。	テキスト・ストリング。
*TableOwner	説明されている表の所有者、高レベル修飾子、集合、またはスキーマ。	テキスト・ストリング。
	この値は ODBC の規則に定義されている有効な修飾子でなければなりません。	
*TablePhysicalName	ODBC に知られている物理表名 (システム DSN 名)。	テキスト・ストリング。
*ProcessName	作成中または関連付け中のプロセスの名前。	テキスト・ストリング。

表 34. StepVWPOutputTable.tag のトークン (続き)

トークン	説明	許可される値
*CurrentCheckPointID++	0 から始まり、トークンに置換されるたびに増加する索引。 このトークンは必須です。	数値。

値の例

表35 は、各トークンに指定するメタデータの種類を例示するために、各トークンの値の例を示しています。

表 35. StepVWPOutputTable.tag トークンの値の例

トークン	値の例
*StepName	Revenue by product
*DatabaseName	Finance Warehouse
*TableOwner	FINADMIN
*TablePhysicalName	INVENTORY
*ProcessName	Marketing process
*CurrentCheckPointID++	1

StepVWPProgramInstance.tag

このテンプレートを使用して、ウェアハウス・エージェントによって実行されるデータウェアハウスセンター プログラムのインスタンスを定義します。このテンプレートはさらに、VWPTemplate と呼ばれるデータウェアハウスセンター プログラム定義への関係、およびデータウェアハウスセンター プログラムを使用するステップを定義します。このテンプレートは、データウェアハウスセンター プログラムを使用するステップごとに必要です。

トークン

86ページの表36 は、このテンプレート内の各トークンに関する情報を示しています。

StepVWPPProgramInstance.tag

表 36. StepVWPPProgramInstance.tag のトークン

トークン	説明	許可される値
エンティティ・パラメーター		
*VWPPProgramInstanceKey	このプログラムのインスタンスを固有に識別するキー。このキーは、タグ言語ファイル内の他のすべてのキーに対して固有でなければなりません。	数値。
	ヒント: VWPPProgramInstance.tag テンプレートの処理を終了してから、キーの値を増加させてください。	
	このトークンは必須です。	
関係パラメーター		
*StepName	関連付け中のステップの名前。	テキスト・ストリング。
*VWPPProgramTemplateName	作成中のデータウェアハウスセンターのビジネス名。	テキスト・ストリング。
*CurrentCheckPointID++	0 から始まり、トークンに置換されるたびに増加する索引。	数値。
	このトークンは必須です。	

値の例

表37 は、各トークンに指定するメタデータの種類を例示するために、各トークンの値の例を示しています。

表 37. StepVWPPProgramInstance.tag トークンの値の例

トークン	値の例
*VWPPProgramInstanceKey	070001
*StepName	Revenue by location
*VWPPProgramTemplateName	My ISV Program
*CurrentCheckPointID++	2

SourceDataBase.tag

このテンプレートを使用して、データウェアハウスセンターにインポートするソース・データベース、ファイル・システム、またはファイルを定義します。このテンプレートを使用して、リレーショナルの非 DB2 ソース・データベース、および DB2 ソース・データベースを定義できます。

このテンプレートはさらに、以下のオブジェクト間の関係も定義します。

- ソース・データベース
- ソース・データベースに使用するエージェント・サイト
- ソース・データベースを定義するセキュリティー・グループ

トークン

表38 は、このテンプレート内の各トークンに関する情報を示しています。

表 38. SourceDataBase.tag のトークン

トークン	説明	許可される値
エンティティ・パラメーター		
*DatabaseName	データベースの名前。 その名前は、ウェアハウス・コントロール・データベース内で固有でなければなりません。 このトークンは必須です。	最長 80 バイトのテキスト・ストリング。
*DatabaseDescription	データベースについての短い説明。 このトークンは任意指定です。	最長 254 バイトのテキスト・ストリング。
*DatabaseNotes	データベースについての長い説明。 このトークンは任意指定です。	最長 32700 バイトのテキスト・ストリング。
*DatabaseContact	このデータベースに関する情報の問い合わせ先。 このトークンは任意指定です。	最長 64 バイトのテキスト・ストリング。

SourceDataBase.tag

表 38. SourceDataBase.tag のトークン (続き)

トークン	説明	許可される値
*DatabaseServerName	データベースが存在するサーバーの名前。 このトークンはフラット・ファイル LAN ファイルに必要です。その他の場合は、任意指定です。	最長 64 バイトのテキスト・ストリング。
*DatabaseVersion	データベースのバージョン。	テキスト・ストリング。
*DatabasePhysicalName	ODBC に知られている、データベース・マネージャーに定義されたデータベースの物理データベース名。 このトークンは必須です。	最長 40 バイトのテキスト・ストリング。
*DatabaseType	データベース・ファミリーのタイプ。 このトークンは必須です。	以下の値の 1 つ: ISV_IR_DB2Family DB2 ファミリー ISV_IR_Oracle Oracle ISV_IR_Sybase Sybase ISV_IR_MSSQLServer Microsoft® SQLServer ISV_IR_Informix Informix ISV_IR_GenericODBC 汎用 ODBC ISV_IR_FFJan フラット・ファイル LAN ISV_IR_VSAM VSAM ISV_IR_IMS IMS

表 38. SourceDataBase.tag のトークン (続き)

トークン	説明	許可される値
*DatabaseTypeExtended	AS/400 システムまたはファイルのタイプ。 このトークンは必須です。	以下の値の 1 つ: ISV_IR_DB2400CISC CISC 用の DB2 UDB (AS/400 版) ISV_IR_DB2400RISC RISC 用の DB2 UDB (AS/400 版) ISV_IR_FFLanLocalCmd ローカル・フラット・ファイル ISV_IR_FFLanFTPCopy リモート・システムから FTP を使用して送信されたローカル・フラット・ファイル
*DatabaseUserid	データベースにアクセスするためのユーザー ID。 このトークンは任意指定です。	最長 36 バイトのテキスト・ストリング。
関係パラメーター		
*SecurityGroup	ソースまたはターゲット・データベースを作成するためのセキュリティ・グループ。 このトークンは必須であり、デフォルトのセキュリティ・グループを指定しなければなりません。	デフォルトのセキュリティ・グループとして ISV_DEFAULTSECURITYGROUP。
*AgentSite	ソースまたはターゲット・データベースに使用するエージェント・サイト。 このトークンは必須ですが、デフォルトのエージェント・サイトを指定することもできます。	最長 80 バイトのテキスト・ストリング。 デフォルトのエージェント・サイトは ISV_DEFAULTAGENTSITE です。

SourceDataBase.tag

表 38. SourceDataBase.tag のトークン (続き)

トークン	説明	許可される値
*CurrentCheckPointID++	0 から始まり、トークンに置換されるたびに増加する索引。 このトークンは必須です。	数値。

値の例

表39 は、各トークンに指定するメタデータの種類を例示するために、各トークンの値の例を示しています。

表 39. SourceDataBase.tag トークンの値の例

トークン	値の例
*DatabaseName	Finance Warehouse
*DatabaseDescription	This database contains financial information.
*DatabaseNotes	This is the warehouse where all geographies keep financial information.
*DatabaseContact	Valerie Zieman
*DatabaseServerName	CHI11W71
*DatabaseVersion	V6.1.0
*DatabasePhysicalName	FINANCE
*DatabaseType	ISV_IR_DB2Family
*DatabaseTypeExtended	ISV_DEFAULTVALUE
*DatabaseUserid	DB2ADMIN
*SecurityGroup	ISV_DEFAULTSECURITYGROUP
*AgentSite	My agent site
*CurrentCheckPointID++	5

SubjectArea.tag

このテンプレートを使用して、作成するプロセスおよびステップを入れるためのサブジェクト・エリアを定義します。タグ言語ファイルごとに、作成するプロセスおよびステップを入れるためのサブジェクト・エリアが少なくとも 1 つあります。このテンプレートは、プロセスおよびステップを定義している場合は必須です。

このテンプレートはさらに、サブジェクト・エリアとヘッダー・ファイルが指定するセキュリティー・グループとの関係も定義します (70ページの『HeaderInfo.tag』を参照してください)。

トークン

表40 は、このテンプレート内の各トークンに関する情報を示しています。

表 40. SubjectArea.tag のトークン

トークン	説明	許可される値
エンティティ・パラメーター		
*SubjectArea	特定のサブジェクト・エリアに作成または追加されたすべてのプロセスおよびステップを入れるグループの名前。 その名前は、ウェアハウス・コントロール・データベース内で固有でなければなりません。このトークンは必須です。	最長 80 バイトのテキスト・ストリング。
*SubjectAreaContact	このサブジェクト・エリアに責任を持つ人または組織の名前。	テキスト・ストリング。
*SubjectAreaDescription	プロセスおよびステップのグループについての短い説明。 このトークンは任意指定です。	最長 254 バイトのテキスト・ストリング。
*SubjectAreaNotes	プロセスおよびステップのグループについての長い説明。 このトークンは任意指定です。	最長 32700 バイトのテキスト・ストリング。
関係パラメーター		

SubjectArea.tag

表 40. SubjectArea.tag のトークン (続き)

トークン	説明	許可される値
*SecurityGroup	サブジェクト・エリアを作成するためのセキュリティ・グループ。 このトークンは必須であり、デフォルトのセキュリティ・グループを指定しなければなりません。	デフォルトのセキュリティ・グループとして ISV_DEFAULTSECURITYGROUP。
*CurrentCheckPointID++	0 から始まり、トークンに置換されるたびに増加する索引。 このトークンは必須です。	数値。

値の例

表41 は、各トークンに指定するメタデータの種類を例示するために、各トークンの値の例を示しています。

表 41. SubjectArea.tag トークンの値の例

トークン	値の例
*SubjectArea	Group of processes and steps generated for the partner tool
*SubjectAreaContact	DEPT W24A
*SubjectAreaDescription	This subject area contains all the processes and steps generated for Data Warehouse Center by the partner tool.
*SubjectAreaNotes	The processes and steps in this subject area will be used to evaluate the product.
*SecurityGroup	ISV_DEFAULTSECURITYGROUP
*CurrentCheckPointID++	9

Table.tag

このテンプレートを使用して、ソースとターゲット表の両方、およびデータウェアハウスセンターがアクセスするソース・ファイルとセグメントを定義します。このテンプレートを使用して、ソースとターゲット表、ファイル、およびセグメントを定義します。

このテンプレートは、ODBC データ・ソース内の表および DB2 ターゲット表をデータウェアハウスセンターが定義するために必要なすべてのメタデータを定義します。テンプレートはさらに、表とこの表を含むデータベースとの間の関係も定義します。

トークン

表42 は、このテンプレート内の各トークンに関する情報を示しています。

表 42. Table.tag のトークン

トークン	説明	許可される値
エンティティ・パラメーター		
<i>*TableFullName</i>	<p>リレーショナル表またはファイルの完全修飾名。</p> <p>表の場合、この名前は <i>*TableOwner</i> および <i>*TablePhysicalName</i> トークンの値を連結して、ピリオドで区切ったものです。</p> <p>ファイルの場合、<i>*TableOwner</i> はブランクのまま、<i>*TableFullName</i> および <i>*TablePhysicalName</i> の値は同じにします。</p> <p>その名前は、ウェアハウス・コントロール・データベース内で固有でなければなりません。</p> <p>このトークンは必須です。</p>	最長 80 バイトのテキスト・ストリング。
<i>*TableDescription</i>	<p>表についての短い説明。</p> <p>このトークンは任意指定です。</p>	最長 254 バイトのテキスト・ストリング。
<i>*TableNotes</i>	<p>表についての長い説明。</p> <p>このトークンは任意指定です。</p>	最長 32700 バイトのテキスト・ストリング。

Table.tag

表 42. Table.tag のトークン (続き)

トークン	説明	許可される値
<i>*TableOwner</i>	表の所有者、高レベル修飾子、集合、またはスキーマ。 このトークンは必須です。	最長 15 バイトのテキスト・ストリング。
<i>*TablePhysicalName</i>	データベース・マネージャまたはファイル・システムに定義された、物理表名。 このトークンは必須です。	最長 80 バイトのテキスト・ストリング。
<i>*TableBinaryIfFile</i>	表がファイルを表す場合にファイルがバイナリー・データだけを含むかどうかを示すフラグ。 このトークンは任意指定です。	以下の値の 1 つ: ISV_DR_FILE_IS_BINARY ファイルはバイナリーです。 ISV_DR_FILE_IS_NOT_BINARY ファイルは ASCII、または混合形式です。
<i>*TableFirstRowNamesIfFile</i>	表がファイルを表す場合に、ファイルの最初の行に列名が含まれるかどうかを示すフラグ。 このトークンは任意指定です。	以下の値の 1 つ: ISV_DR_ROW_CONTAINS_NAMES ファイルの最初の行には、列名が含まれます。 ISV_DR_ROW_DOES_NOT_CONTAIN_NAMES ファイルの最初の行には、データが含まれます。

表 42. Table.tag のトークン (続き)

トークン	説明	許可される値
*TableTypeIfFile	表がファイルを表す場合に、ファイルのタイプ。 このトークンは任意指定です。	以下の値の 1 つ: ISV_DR_REL_TABLE 表はリレーショナル表です。 ISV_DR_COMMA_DELIMITED ファイル内の列はコンマで区切られます。 ISV_DR_FIXED_FORMAT ファイル内の列は混合形式です。 ISV_DR_TAB_DELIMITED ファイル内の列はタブで区切られます。 ISV_DR_CHAR_DELIMITED ファイル内の列は、 *TableDelimiterIfFile の値で区切られます。
*TableDelimiterIfFile	ファイル・タイプが ISV_DR_CHAR_DELIMITED の場合に、複数のフィールドを分ける区切り文字の値。 このトークンは任意指定です。	長さ 1 バイトのテキスト・ストリング。
*TableIsAView	表がビューかどうかを示すトークン。	以下の値の 1 つ: ISV_TableIsAView 表はビューです。 ISV_TableIsNotAView 表はビューではありません。
*TableIsADimensionTable	表がスタースキーマの一部で、次元データを含むかどうかを示すトークン。	以下の値の 1 つ: ISV_TableIsADimensionalTable 表は次元表です。 ISV_TableIsNotADimensionalTable 表は次元表ではありません。

Table.tag

表 42. Table.tag のトークン (続き)

トークン	説明	許可される値
<i>*TableIsAnAlias</i>	表が実際には他の表の別名であるかどうかを示すトークン。	以下の値の 1 つ: ISV_TableIsAnAlias この表は他の表の別名です。 ISV_TableIsNotAnAlias この表は他の表の別名ではありません。
<i>*TableCreatedByDWC</i>	データウェアハウスセンターがこの表を作成して管理するかどうかを示すトークン。	以下の値の 1 つ: ISV_TableIsToBeCreatedByDWC 表はデータウェアハウスセンターによって作成されます。 ISV_TableIsNotToBeCreatedByDWC 表はデータウェアハウスセンターによって作成されません。
<i>*TableGrantedToPublic</i>	表の作成時にデータウェアハウスセンターがこの表にパブリックのアクセスを許可するかどうかを示すトークン。これが有効なのは、データウェアハウスセンターが表を作成するときだけです。	以下の値の 1 つ: ISV_GrantTableAccessToPublic データウェアハウスセンターはこの表に PUBLIC アクセスを許可します。 ISV_DoNotGrantTableAccessToPublic データウェアハウスセンターはこの表に PUBLIC アクセスを許可しません。
<i>*TableIsPersistent</i>	表内のデータが、この表を使用する複数のステップの実行と実行の間に保持されるかどうかを示すトークン。表が持続しない場合、表内のデータは各使用の後に削除されます。	以下の値の 1 つ: ISV_TableIsPersistent 表は持続的なものとみなされます。 ISV_TableIsTransient 表は一時的なものとみなされます。
<i>*TableMaximumEditions</i>	表がエディションをサポートする場合、表が持つことのできるエディションの最大数。	数値。

表 42. Table.tag のトークン (続き)

トークン	説明	許可される値
*TableGenerateCreateStatement	データウェアハウスセンターが表作成ステートメントを生成するかどうかを示すトークン。	以下の値の 1 つ: ISV_GenerateCreateTableStmt データウェアハウスセンターは、CREATE TABLE ステートメントを生成することになります。 ISV_DoNotGenerateCreateTableStmt データウェアハウスセンターは、CREATE TABLE ステートメントを生成しないことになります。
*TableIsAFactTable	表がスタースキーマの一部で、表にファクト情報が含まれるかどうかを示すトークン。	以下の値の 1 つ: ISV_TableIsAFactTable 表はファクト表です。 ISV_TableIsNotAFactTable 表はファクト表ではありません。
*TableCreateStatement	表を作成するための DDL。 ISV_DoNotGenerateCreateTableStmt が指定されている場合、このトークンだけを使用してください。	テキスト・ストリング。
関係パラメーター		
*DatabaseName	表を含むデータベースの名前。 その名前は、ウェアハウス・コントロール・データベース内で固有でなければなりません。 このトークンは必須です。	最長 80 バイトのテキスト・ストリング。
*DatabasePhysicalName	表を含むデータベースの物理データベース名。 このトークンは必須です。	最長 40 バイトのテキスト・ストリング。

Table.tag

表 42. Table.tag のトークン (続き)

トークン	説明	許可される値
*CurrentCheckpointID++	0 から始まり、トークンに置換されるたびに増加する索引。 このトークンは必須です。	数値。

値の例

表43 は、各トークンに指定するメタデータの種類を例示するために、各トークンの値の例を示しています。

表 43. Table.tag トークンの値の例

トークン	値の例
*TableFullName	DB2ADMIN.GEOGRAPHY
*TableDescription	Contains geography information
*TableNotes	This table contains all the information about geographies serviced by our company
*TableOwner	DB2ADMIN
*TablePhysicalName	GEOGRAPHY
*TableBinaryIfFile	ISV_DEFAULTVALUE
*TableFirstRowNamesIfFile	ISV_DEFAULTVALUE
*TableTypeIfFile	ISV_DEFAULTVALUE
*TableDelimiterIfFile	ISV_DEFAULTVALUE
*TableIsAView	ISV_TableIsAView
*TableIsADimensionTable	ISV_TableIsNotADimensionTable
*TableIsAnAlias	ISV_TableIsAnAlias
*TableCreatedByDWC	ISV_TableIsToBeCreatedByDWC
*TableGrantedToPublic	ISV_GrantTableAccessToPublic
*TableIsPersistent	ISV_TableIsTransient
*TableMaximumEditions	12
*TableGenerateCreateStatement	ISV_GenerateCreateTableStmt
*TableIsAFactTable	ISV_TableIsAFactTable
*TableCreateStatement	Create table xyz
*DatabaseName	Finance warehouse
*DatabasePhysicalName	FINANCE

表 43. Table.tag トークンの値の例 (続き)

トークン	値の例
*CurrentCheckpointID++	7

VWPGroup.tag

このテンプレートを使用して、定義中のデータウェアハウスセンター プログラムを入れるグループを定義します。このテンプレートは、データウェアハウスセンター プログラムを定義している場合には必須です。

トークン

表44 は、このテンプレート内の各トークンに関する情報を示しています。

表 44. VWPGroup.tag のトークン

トークン	説明	許可される値
エンティティ・パラメーター		
*VWPGroup	作成中のすべてのデータウェアハウスセンター プログラムを含めるプログラム・グループの固有名。 その名前は、ウェアハウス・コントロール・データベース内で固有でなければなりません。 このトークンは必須です。	最長 80 バイトのテキスト・ストリング。
*VWPGroupDescription	データウェアハウスセンター プログラムのグループについての短い説明。 このトークンは任意指定です。	最長 254 バイトのテキスト・ストリング。
*VWPGroupNotes	データウェアハウスセンター プログラムのグループについての長い説明。 このトークンは任意指定です。	最長 32700 バイトのテキスト・ストリング。

VWPGroup.tag

表 44. VWPGroup.tag のトークン (続き)

トークン	説明	許可される値
関係パラメーター		
*CurrentCheckPointID++	0 から始まり、トークンに置換されるたびに増加する索引。 このトークンは必須です。	数値。

値の例

表45 は、各トークンに指定するメタデータの種類を例示するために、各トークンの値の例を示しています。

表 45. VWPGroup.tag トークンの値の例

トークン	値の例
*VWPGroup	Group of programs for the partner tool
*VWPGroupDescription	This group contains all the programs used by Data Warehouse Center for the partner tool
*VWPGroupNotes	These programs can be used to determine the relationship between sales and location.
*CurrentCheckPointID++	2

VWPProgramInstanceParameter.tag

このテンプレートを使用して、データウェアハウスセンターが特定のステップのためのデータウェアハウスセンター プログラムのインスタンスに渡すパラメーターを、追加または変更します。たとえば、ホスト名パラメーターのデフォルト値は VWPProgramTemplateParameter.tag ファイルで設定します (106ページを参照してください)。このテンプレートを使用して、この特定のステップが実行されるときにデータウェアハウスセンター プログラムに渡される値を変更します。

データウェアハウスセンター プログラムがパラメーターを渡す先としてデータウェアハウスセンターを必要とする場合、このテンプレートは必須です。このテンプレートを各パラメーターに含めることによって、データウェアハウスセンターが複数のパラメーターをプログラムに渡すように指定することができます。

このテンプレートはさらに、パラメーターとそのプログラム・インスタンスとの間の関係も定義します。

トークン

表46 は、このテンプレート内の各トークンに関する情報を示しています。

表 46. VWPPProgramInstanceParameter.tag のトークン

トークン	説明	許可される値
エンティティ・パラメーター		
*VWPPProgramInstanceParameterName	データウェアハウスセンタープログラムに渡すパラメーターの固有名または説明。 このトークンは必須です。	最長 80 バイトのテキスト・ストリング。
*VWPPProgramInstanceParameterOrder	パラメーター・リスト内でのパラメーターの順位を示す、0 から始まる数値。 このトークンは必須です。	数値。
*VWPPProgramInstanceParameterData	パラメーターの値としてデータウェアハウスセンタープログラムに渡されるデータ。 このトークンは必須です。	長さが 240 バイトまでのテキスト・ストリングまたは数値。
*VWPPProgramInstanceParameterKey	このプログラム・パラメーターのインスタンスを固有に識別するキー。このキーは、交換ファイル内の他のすべてのパラメーター・キーに対して固有でなければなりません。 ヒント: VWPPProgramInstanceParameter.tag テンプレートの処理を終了してから、キーの値を増加させてください。 このトークンは必須です。	最長 10 バイトのテキスト値。

VWPPProgramInstanceParameter.tag

表 46. VWPPProgramInstanceParameter.tag のトークン (続き)

トークン	説明	許可される値
*VWPPProgramInstanceParameterType	このパラメーターに含まれる値のタイプ。たとえば、文字、数値、またはパスワード・データ。	以下の値の 1 つ: ISV_ParameterTypeNone パラメーター・タイプが不明であるか、または該当しません。 ISV_ParameterTypeCharacter パラメーター・タイプは文字です。 ISV_ParameterTypeNumeric パラメーター・タイプは数値です。 ISV_ParameterTypePassword パラメーター・タイプはパスワードです。
関係パラメーター		
*VWPPProgramInstanceKey	このプログラムのインスタンスを固有に識別するキー。このキーは、交換ファイル内の他のすべてのキーに対して固有でなければなりません。 ヒント: VWPPProgram Instance.tag テンプレートの処理を終了してから、キーの値を増加させてください。 このトークンは必須です。	最長 10 バイトのテキスト値。
*CurrentCheckPointID++	0 から始まり、トークンに置換されるたびに増加する索引。 このトークンは必須です。	数値。

値の例

以下の表は、各トークンに指定するメタデータの種類を例示するために、各トークンの値の例を示しています。

表 47. VWPPProgramInstanceParameter.tag トークンの値の例

トークン	値の例
*VWPPProgramInstanceParameterName	DB2 UDB user ID
*VWPPProgramInstanceKey	070000
*VWPPProgramInstanceParameterOrder++	1
*VWPPProgramInstanceParameterData	my_userid
*VWPPProgramInstanceParameterKey	012994
*VWPPProgramInstanceParameterType	ISV_ParameterTypeNumeric
*VWPPProgramInstanceKey	070001
*CurrentCheckPointID++	12

VWPPProgramTemplate.tag

このテンプレートを使用して、データウェアハウスセンター プログラムを定義します。ウェアハウス・プログラムがデータウェアハウスセンター コントロール・データベースにすでに存在するのでなければ、タグ言語ファイルがデータウェアハウスセンター プログラムを参照する場合にこのテンプレートは必須です。

このテンプレートはさらに、ウェアハウス・プログラム定義とプログラムが属するデータウェアハウスセンター プログラム・グループとの間の関係を定義します。

トークン

表48 は、このテンプレート内の各トークンに関する情報を示しています。

表 48. VWPPProgramTemplate.tag のトークン

トークン	説明	許可される値
	エンティティ・パラメーター	

VWPPProgramTemplate.tag

表 48. VWPPProgramTemplate.tag のトークン (続き)

トークン	説明	許可される値
*VWPPProgramTemplateName	データウェアハウスセンター プログラム・テンプレートの名前。 その名前は、ウェアハウス・コントロール・データベース内で固有でなければなりません。 このトークンは必須です。	最長 80 バイトのテキスト・ストリング。
*VWPPProgramTemplateDescription	データウェアハウスセンター プログラムおよびその機能についての短い説明。 このトークンは任意指定です。	最長 254 バイトのテキスト・ストリング。
*VWPPProgramTemplateNotes	データウェアハウスセンター プログラムおよびその機能についての長い説明。 このトークンは任意指定です。	最長 32700 バイトのテキスト・ストリング。
*VWPPProgramTemplateExecutableName	ステップが実行されるときに実行されるデータウェアハウスセンター プログラムの完全修飾プログラム名。 データウェアハウスセンター プログラムがシステム・パスにインストールされている場合、ウェアハウス・プログラム名が完全修飾である必要はありません。 このトークンは必須です。	最長 240 バイトのテキスト・ストリング。

表 48. VWPPProgramTemplate.tag のトークン (続き)

トークン	説明	許可される値
*VWPPProgramTemplateType	プログラムのタイプ。 このトークンは必須です。	以下の値の 1 つ: ISV_PROGRAMTYPEDLL データウェアハウスセンター プログラムはダイナミック・リンク・ライブラリー (DLL) からロードされるか、またはロード・モジュールです。 ISV_PROGRAMTYPECOMMAND データウェアハウスセンター プログラムはコマンド・ファイルです。
ISV_PROGRAMTYPEEXECUTABLE データウェアハウスセンター プログラムは実行可能ファイルです。		
*VWPPProgramTemplateFunctionName	*VWPPProgramTemplateType の値が ISV_PROGRAMTYPEDLL である場合にデータウェアハウスセンターが呼び出す DLL 内の入り口点の名前。 *VWPPProgramTemplateType の値が ISV_PROGRAMTYPEDLL である場合に、このトークンは必須です。	最長 80 バイトのテキスト・ストリング。
関係パラメーター		
*VWPPGroup	データウェアハウスセンター プログラムを含めるグループの名前。 このトークンは必須です。	最長 80 バイトのテキスト・ストリング。

VWPPProgramTemplate.tag

表 48. VWPPProgramTemplate.tag のトークン (続き)

トークン	説明	許可される値
*CurrentCheckPointID++	0 から始まり、トークンに置換されるたびに増加する索引。 このトークンは必須です。	数値。
*AgentSite	ソースまたはターゲットに使用するエージェント・サイト。 このトークンは必須です。	最長 80 バイトのテキスト・ストリング。 デフォルトのエージェント・サイトは、ISV_DEFAULTAGENTSITE を指定します。

値の例

表49 は、各トークンに指定するメタデータの種類を例示するために、各トークンの値の例を示しています。

表 49. VWPPProgramTemplate.tag トークンの値の例

トークン	値の例
*VWPPProgramTemplateName	My ISV program
*VWPPProgramTemplateDescription	This program exports data from an ODBC database.
*VWPPProgramTemplateNotes	This program will export data from an ODBC database, process it, and place it into another database.
*VWPPProgramTemplateExecutableName	c:\ISV\BIN\MYPROG.EXE
*VWPPProgramTemplateType	ISV_PROGRAMTYPEPEEXECUTABLE
*VWPPProgramTemplateFunctionName	My_Prog_Func_Name
*VWPPGroup	Group of programs for partner tool
*CurrentCheckPointID++	3

VWPPProgramTemplateParameter.tag

このテンプレートを使用して、データウェアハウスセンターがデータウェアハウスセンター プログラムに渡すパラメーターを定義します。

データウェアハウスセンター プログラムがパラメーターを渡す先としてデータウェアハウスセンターを必要とする場合、このテンプレートは必須です。この

テンプレートを各パラメーターに含めることによって、複数のパラメーターをデータウェアハウスセンター プログラムに渡すように指定することができます。

このテンプレートは VWPPProgramTemplate.tag ファイル (103ページの『VWPPProgramTemplate.tag』) と共に使用します。このテンプレートは、パラメーターとそのデータウェアハウスセンター プログラム定義 (VWPPProgramTemplate.tag) との関係を定義します。

トークン

表50 は、このテンプレート内の各トークンに関する情報を示しています。

表 50. VWProgramTemplateParameter.tag のトークン

トークン	説明	許可される値
エンティティ・パラメーター		
*VWPPProgramTemplateParameterName	データウェアハウスセンター プログラムに渡すパラメーターの名前または説明。 その名前は、データウェアハウスセンター プログラム内で固有でなければなりません。 このトークンは必須です。	最長 80 バイトのテキスト・ストリング。
*VWPPProgramTemplateParameterOrder	パラメーター・リスト内でのパラメーターの順位を示す、0 から始まる数値。 このトークンは必須です。	数値。
*VWPPProgramTemplateParameterData	パラメーターの値としてデータウェアハウスセンター プログラムに渡されるデータ。 このトークンは必須です。	長さが 240 バイトまでのテキスト・ストリングまたは数値。

VWPPProgramTemplateParameter.tag

表 50. VWPPProgramTemplateParameter.tag のトークン (続き)

トークン	説明	許可される値
*VWPPProgramTemplateParameterKey	このプログラム・パラメーターのテンプレートを固有に識別するキー。このキーは、交換ファイル内の他のすべてのキーに対して固有でなければなりません。 ヒント: VWPPProgramTemplateParameter.tag テンプレートの処理を終了してから、キーの値を増加させてください。 このトークンは必須です。	数値。
*VWPPProgramInstanceParameterType	このパラメーターに含まれる値のタイプ。たとえば、文字、数値、またはパスワード・データ。	以下の値の 1 つ: ISV_ParameterTypeNone パラメーター・タイプが不明であるか、または該当しません。 ISV_ParameterTypeCharacter パラメーター・タイプは文字です。 ISV_ParameterTypeNumeric パラメーター・タイプは数値です。 ISV_ParameterTypePassword パラメーター・タイプはパスワードです。
関係パラメーター		
*VWPPProgramTemplateName	このパラメーターを使用するデータウェアハウスセンター プログラムの名前。 このトークンは必須です。	最長 80 バイトのテキスト・ストリング。

表 50. VWProgramTemplateParameter.tag のトークン (続き)

トークン	説明	許可される値
*CurrentCheckPointID++	0 から始まり、トークンに置換されるたびに増加する索引。 このトークンは必須です。	数値。

値の例

表51 は、各トークンに指定するメタデータの種類を例示するために、各トークンの値の例を示しています。

表 51. VWPPProgramTemplateParameter.tag トークンの値の例

トークン	値の例
*VWPPProgramTemplateParameterName	DB2 UDB user ID
*VWPPProgramTemplateParameterOrder	1
*VWPPProgramInstanceKey	070000
*VWPPProgramTemplateParameterData	my_userid
*VWPPProgramTemplateParameterKey	012994
*VWPPProgramInstanceParameterType	ISV_ParameterTypePassword
*VWPPProgramTemplateName	My ISV program
*CurrentCheckPointID++	4

WarehouseDataBase.tag

このテンプレートを使用して、データウェアハウスセンターにインポートするターゲット・ウェアハウス・データベースを定義します。

このテンプレートはさらに、以下のオブジェクト間の関係も定義します。

- ターゲット・ウェアハウス・データベース
- ターゲット・ウェアハウス・データベースに使用するエージェント・サイト
- ターゲット・ウェアハウス・データベースを定義するセキュリティー・グループ

トークン

110ページの表52 は、このテンプレート内の各トークンに関する情報を示しています。

WarehouseDataBase.tag

表 52. WarehouseDataBase.tag のトークン

トークン	説明	許可される値
エンティティ・パラメーター		
<i>*DatabaseName</i>	データベースの固有名。 その名前は、ウェアハウ ス・コントロール・デー タベース内で固有でなけ ればなりません。 このトークンは必須で す。	最長 80 バイトのテキスト・ストリング。
<i>*DatabaseDescription</i>	データベースについての 短い説明。 このトークンは任意指定 です。	最長 254 バイトのテキスト・ストリング。
<i>*DatabaseNotes</i>	データベースについての 長い説明。 このトークンは任意指定 です。	最長 32700 バイトのテキスト・ストリング。
<i>*DatabaseContact</i>	このデータベースに関す る情報の問い合わせ先。 このトークンは任意指定 です。	最長 64 バイトのテキスト・ストリング。
<i>*DatabaseServerName</i>	データベースが存在する サーバーの名前。 このトークンは任意指定 です。	最長 64 バイトのテキスト・ストリング。
<i>*DatabaseVersion</i>	データベースのバージョ ン。	テキスト・ストリング。
<i>*DatabasePhysicalName</i>	データベース・マネージ ャーに定義されたデー タベースの物理データベ ース名。 このトークンは必須で す。	最長 40 バイトのテキスト・ストリング。

表 52. WarehouseDataBase.tag のトークン (続き)

トークン	説明	許可される値
*DatabaseType	データベース・ファミリーのタイプ。 このトークンは必須です。	以下の値の 1 つ: ISV_IR_DB2Family DB2 ファミリー ISV_IR_Oracle Oracle ISV_IR_Sybase Sybase ISV_IR_MSSQLServer Microsoft SQLServer ISV_IR_Informix Informix ISV_IR_GenericODBC 汎用 ODBC ISV_IR_FFLan フラット・ファイル LAN ISV_IR_VSAM VSAM ISV_IR_IMS IMS
*DatabaseTypeExtended	AS/400 システムまたはファイルのタイプ。 このトークンは必須です。	以下の値の 1 つ: ISV_IR_DB2400CISC CISC 用の DB2 UDB (AS/400 版) ISV_IR_DB2400RISC RISC 用の DB2 UDB (AS/400 版) ISV_IR_FFLanLocalCmd ローカル・フラット・ファイル ISV_IR_FFLanFTPCopy リモート・システムから FTP を使用して送信されたローカル・フラット・ファイル

WarehouseDataBase.tag

表 52. WarehouseDataBase.tag のトークン (続き)

トークン	説明	許可される値
*DatabaseUserid	データベースにアクセスするためのユーザー ID。 このトークンは任意指定です。	最長 36 バイトのテキスト・ストリング。
関係パラメーター		
*SecurityGroup	ソースまたはターゲット・データベースを作成するためのセキュリティ・グループ。 このトークンは必須ですが、デフォルトのセキュリティ・グループを指定することもできます。	最長 80 バイトのテキスト・ストリング。 デフォルトのセキュリティ・グループは、ISV_DEFAULTSECURITYGROUP を指定します。
*AgentSite	ソースまたはターゲットに使用するエージェント・サイト。 このトークンは必須です。	最長 80 バイトのテキスト・ストリング。 デフォルトのエージェント・サイトは、ISV_DEFAULTAGENTSITE を指定します。
*CurrentCheckpointID++	0 から始まり、トークンに置換されるたびに増加する索引。 このトークンは必須です。	数値。

値の例

表53 は、各トークンに指定するメタデータの種類を例示するために、各トークンの値の例を示しています。

表 53. WarehouseDataBase.tag トークンの値の例

トークン	値の例
*DatabaseName	Finance Warehouse
*DatabaseDescription	This database contains financial information.

表 53. WarehouseDataBase.tag トークンの値の例 (続き)

トークン	値の例
*DatabaseNotes	This is the warehouse where all geographies keep financial information.
*DatabaseContact	Valerie Zieman
*DatabaseServerName	CHI11W71
*DatabaseVersion	V6.1.0
*DatabasePhysicalName	FINANCE
*DatabaseType	DB2 Family
*DatabaseTypeExtended	ISV_DEFAULTVALUE
*DatabaseUserid	DB2ADMIN
*SecurityGroup	ISV_DEFAULTSECURITYGROUP
*AgentSite	My agent site
*CurrentCheckPointID++	6

第6章 データウェアハウスセンターのメタデータ

この章では、ソース・データベースとターゲット・データベースを記述する、データウェアハウスセンターのメタデータについて説明します。異なるアプリケーション同士で、メタデータをエクスポートして、データベースに関する情報を共用することができます。

表54 では、タグ言語ファイル内の各オブジェクトと、データウェアハウスセンター内の対応する論理オブジェクトとの間のマッピングを示します。

表 54. ソースおよびターゲット・データベース用の論理オブジェクト

タグ言語ファイル内の オブジェクト	データウェアハウスセンター の論理オブジェクト	参照先:
DATABASE	ウェアハウス・ソースまたは ウェアハウス・ターゲット	『DATABASE オブジェク ト』
TABLE	表、ファイル、または IMS セグメント	121ページの『TABLES オ ブジェクト』
COLUMN	列またはフィールド	127ページの『COLUMN オブジェクト』

データウェアハウスセンターでは、データベース、表、および列の関係も定義されます。各オブジェクトに関する節では、オブジェクトが参加する関係がリストされています。これは、パートナー・アプリケーションにとって有用です。

DATABASE オブジェクト

DATABASE オブジェクトには、ソース・データベース / ターゲット・データベース、ファイル・システム、またはファイルに関するメタデータが含まれます。

プロパティ

116ページの表55 では、DATABASE オブジェクトのプロパティに関する情報を提供します。

DATABASE オブジェクト

表 55. DATABASE オブジェクトのプロパティ

タグ言語の プロパティ名	説明	許容値
NAME	ソースのビジネス名。	テキスト・ストリング (80 バイトまで)。
DBNAME	データベース管理プログラムに定義された物理データベース名。 総称 ODBC データベース、Sybase データベース、IMS データベース、およびファイル・システムの場合、この値はヌルです。	テキスト・ストリング (40 バイトまで)。
SHRTDESC	ソースの短い記述。	テキスト・ストリング (200 バイトまで)。
LONGDESC	ソースの長い記述。	テキスト・ストリング (32700 バイトまで)。
DBTYPE	データベースまたはファイルのファミリー。	次のいずれかの値: 1 DB2 ファミリー 20 Oracle 30 Sybase 40 Microsoft SQLServer 50 Informix 60 総称 ODBC 70 フラット・ファイル LAN 80 VSAM 90 IMS

表 55. DATABASE オブジェクトのプロパティ (続き)

タグ言語の プロパティ名	説明	許容値
DBETYPE	あるファミリー内でのデータベースまたはファイルのタイプ。	次のいずれかの値: 1 DB2/2 3 DB2 MVS 4 AS/400 CISC 5 AS/400 RISC 6 DB2/6000 8 DB2 HP 9 DB2 SUN 11 DB2 NT 12 DB2 VM 13 DB2 SINIX 14 DB2 SCO 15 DB2 VSE 16 DB2 EEE 18 DB2 ファミリー 19 DataJoiner 20 Oracle 30 Sybase 40 Microsoft SQLServer 50 Informix 60 ユーザー定義の ODBC

DATABASE オブジェクト

表 55. DATABASE オブジェクトのプロパティ (続き)

タグ言語の		
プロパティ名	説明	許容値
DBETYPE (続き)	あるファミリー内でのデータベースまたはファイルのタイプ。	次のいずれかの値: 70 フラット・ファイル LAN ローカル・コマンド 71 フラット・ファイル LAN FTP コピー 80 VSAM 90 IMS
ISWH	このソースがウェアハウス・ターゲットであるか、ウェアハウス・ソースであるかを示すフラグ。	次のいずれかの値: Y このソースはウェアハウス・ターゲット。 N このソースはウェアハウス・ソース。
USERID	データウェアハウスセンターがソースへの接続に使用するユーザー ID。	テキスト・ストリング (36 バイトまで)。
CONTACT	ソースについて責任がある担当者名前。	テキスト・ストリング (64 バイトまで)。
USEODBC	ユーザー提供の接続ストリングを使用するか、ストリングを生成するかを指定するフラグ。ファイルの場合、N を使用します。	次のいずれかの値: Y ユーザー定義の接続ストリングを使用する。 N 接続ストリングを生成する。
ODBCSTR	USEODBC が Y に設定されている場合、ユーザー定義の ODBC 接続ストリング。それ以外の場合、このプロパティはヌルです。	テキスト・ストリング (254 バイトまで)。
PREACCMD	ソースがローカル・フラット・ファイル LAN ソースの場合、リモート・ファイルにアクセスするために実行するコマンド。	テキスト・ストリング (64 バイトまで)。
POSTACMD	ソースがローカル・フラット・ファイル LAN ソースの場合、リモート・ファイルにアクセスした後で実行するコマンド。	テキスト・ストリング (64 バイトまで)。

表 55. DATABASE オブジェクトのプロパティ (続き)

タグ言語の プロパティ名	説明	許容値
RETRYCNT	エラーの際に、このソースからのデータの抽出を試行する回数。	数値。
RETRYINT	データの抽出を試行する時間の間隔。	数値。
VERSION	使用中の DB2 のバージョン。	テキスト・ストリング (128 バイトまで)。
DBMSSERV	ODBC 接続用のデータベース・インスタンス / サブシステム / サーバー名。	テキスト・ストリング (128 バイトまで)。
DFLTDEL	システム 390 データベースのデフォルトの文字ストリング区切り文字。	テキスト・ストリング (1 バイトまで)。

図13 では、ターゲット・ウェアハウス・データベースを定義する DATABASE オブジェクトの例を示します。

```
:COMMENT. Begin DATABASE Instance
:COMMENT.
:ACTION.OBJINST(MERGE)
:OBJECT.TYPE(DATABASE)
:INSTANCE.
  NAME(iwhtar)
  DBNAME(IWHTAR)
  DBTYPE(1)
  DBETYPE(11)
  ISWH(Y)
  USERID(marlow)
  USEODBC(N)
  CODEPAGE(437)
  RETRYCNT(3)
  RETRYINT(30)
```

図 13. ターゲット・データベースの DATABASE オブジェクト

120ページの図14 では、ソース・ファイルを定義する DATABASE オブジェクトの例を示します。

DATABASE オブジェクト

```
:ACTION.OBJINST(MERGE)
:OBJECT.TYPE(DATABASE)
:INSTANCE.
  NAME(TBC Operations)
  SHRTDESC(The Beverage Company operational data sources)
  DBTYPE(70)
  DBETYPE(70)
  ISWH(N)
  LOCATION(Thirsty City)
  USERID(XXXXXXXX)
  USEODBC(N)
  CODEPAGE(437)
  RETRYCNT(0)
  RETRYINT(0)
```

図 14. ソース・ファイルの DATABASE オブジェクト

関係

表56 では、DATABASE オブジェクトが参加する関係を示します。これは、パートナー・アプリケーションにとって有用です。「ソース」欄と「ターゲット」欄では、関係のソース・オブジェクトまたはターゲット・オブジェクトが関係に参加できる回数が見られています。たとえば、表56 の「1」と「M」の値は、1 つのデータベースが複数の表に関係できるが、1 つの表は1 つのデータベースにのみ関係できることを意味します。

表 56. DATABASE オブジェクトが参加する関係

ソース	ソースのタグ		ターゲット	ターゲットのタグ	
	言語オブジェクト・タイプ	関係タイプ		言語オブジェクト・タイプ	説明
1	DATABASE	CONTAIN	M	TABLES	データベースまたはファイル・システムに含まれる表またはファイル。

121ページの図15 では、DATABASE オブジェクトと TABLES オブジェクトの間の関係の例を示します。

```

:COMMENT. Relation: DATABASE to TABLES
:COMMENT.
:ACTION.RELATION(ADD)
:RELTYPE.TYPE(CONTAIN) SOURCETYPE(DATABASE) TARGETTYPE(TABLES)
:INSTANCE.
    SOURCEKEY(NAME(TBC Operations) DBNAME() )
    TARGETKEY(DBNAME(TBC Operations) OWNER() TABLES(d:¥iwhdemo¥outcusti.txt) )

```

図 15. DATABASE オブジェクトの TABLES オブジェクトへのリンク

TABLES オブジェクト

このオブジェクトには、ウェアハウス・ソース表、セグメント、またはファイル、あるいはターゲット表に関するメタデータが含まれます。このオブジェクトは、DATABASE オブジェクトに関連付けられます (115ページの『DATABASE オブジェクト』を参照)。

プロパティ

表 57 では、TABLES オブジェクトのプロパティに関する情報を提供します。

表 57. TABLES オブジェクトのプロパティ

タグ言語のプロパティ名	説明	許容値
NAME	表、ファイル、または IMS セグメントの名前。 表名には、IWH.TABLE1 のように、高位修飾子、スキーマ、またはコレクションが含まれます。 データベース名と表名の組み合わせは固有です。 このプロパティは、ファイルの完全修飾パスとファイル名です。	テキスト・ストリング (80 バイトまで)。
SHRTDESC	ファイルまたはセグメントの短い記述。	テキスト・ストリング (200 バイトまで)。
LONGDESC	表の長い記述。	テキスト・ストリング (32700 バイトまで)。
DBNAME	この表またはファイルを含むソースのビジネス名。	テキスト・ストリング (80 バイトまで)。

TABLES オブジェクト

表 57. TABLES オブジェクトのプロパティ (続き)

タグ言語の プロパティ名	説明	許容値
OWNER	表の所有者、高位修飾子、またはコレクション。 ファイルまたは IMS セグメントの場合、このプロパティはヌルです。	テキスト・ストリング (15 バイトまで)。
TABLES	データベース管理プログラムまたはファイル・システムに定義された物理表、ファイル、またはセグメント名。 ファイルおよび IMS セグメントの場合、この値は NAME の値と同じです。	テキスト・ストリング (80 バイトまで)。
TBLISBIN	フラット・ファイル LAN ファイルのファイル転送モードを指定するフラグ。	次のいずれかの値: Y ファイル転送モードはバイナリー。 N ファイル転送モードは ASCII。
TBLNAMESP	DB2 表スペースの名前。	テキスト・ストリング (90 バイトまで)。
TBLFTYPE	ファイルの場合、ファイルのタイプ。	次のいずれかの値: 1 固定 2 コンマ 3 タブ 4 文字
TBLLINAM	ファイルの最初の行に列名が含まれるかどうかを指定するフラグ。	次のいずれかの値: Y ファイルの最初の行には列名が含まれる。 N ファイルの最初の行にはデータが含まれる。
CHARDELM	ファイルの場合、文字区切り文字 (ファイル・タイプが文字の場合)。	テキスト・ストリング (1 バイト)。

表 57. TABLES オブジェクトのプロパティ (続き)

タグ言語の プロパティ名	説明	許容値
CREATYPE	データウェアハウスセンターで表を定義するの に使用された方式。	次のいずれかの値:
		1 表は手作業で定義された。
		2 表定義はデータベース管理プログラムからインポートされた。
		3 表定義は情報カタログ・マネージャーからインポートされた。
4 表は、あるステップがテスト・モードにプロモートされるときに、そのステップ用にデータウェアハウスセンターによって作成された。		
TABALIAS	表に別名があるかどうかを指定するフラグ。	次のいずれかの値:
		Y 表には別名がある。
		N 表には別名がない。
IWHCRTAR	ターゲット表がデータウェアハウスセンターによって作成されるかどうかを指定するフラグ。	次のいずれかの値:
		Y ターゲット表はデータウェアハウスセンターによって作成される。
		N ターゲット表はデータウェアハウスセンターによって作成されない。
IWHGRANT	表について GRANT TO PUBLIC が使用可能にされるかどうかを指定するフラグ。	次のいずれかの値:
		Y 表について GRANT TO PUBLIC が使用可能にされる。
		N 表について GRANT TO PUBLIC が使用可能にされない。

TABLES オブジェクト

表 57. TABLES オブジェクトのプロパティ (続き)

タグ言語の プロパティ名	説明	許容値
IWHDRATN	ウェアハウス・ターゲットの期間 (一時または永続)。	次のいずれかの値: Y 表は永続。 N 表は一時。
IWHMAXED	表の版の最大数。	数値。
IWHCREGN	作成ステートメントが自動的に生成されるかどうかを指定するフラグ。	次のいずれかの値: Y 作成ステートメントは自動的に生成される。 N 作成ステートメントは自動的に生成されない。
IWHCRERU	表の作成ステートメント。	テキスト・ストリング (32,700 バイトまで)。
IDSFACT	表がファクト表として使用されるかどうかを指定するフラグ。	次のいずれかの値: Y 表はファクト表として使用される。 N 表はファクト表として使用されない。
CDSSHEMA	複製用の表スキーマ。	テキスト・ストリング (128 バイトまで)。
CDTABNAM	複製用の表名。	テキスト・ストリング (128 バイトまで)。
BEFORIMG	複製の変更前イメージの接頭部。	テキスト・ストリング (4 バイトまで)。
IDSREPL	表が複製に使用されるかどうかを指定するフラグ。	次のいずれかの値: Y 表は複製に使用される。 N 表は複製に使用されない。
NAMINDEX	DB2 表名の索引。	テキスト・ストリング (90 バイトまで)。

表 57. TABLES オブジェクトのプロパティ (続き)

タグ言語の プロパティ名	説明	許容値
PARTTBSP	表が区分表スペースにあるかどうかを指定するフラグ。	次のいずれかの値: Y 表は区分表スペースにある。 N 表は区分表スペースにない。
DBNAM390	システム 390 データベース名。	テキスト・ストリング (8 バイトまで)。

図16 では、リレーショナル表用の TABLES オブジェクトの例を示します。

```
:COMMENT. Begin TABLES Instance
:COMMENT.
:ACTION.OBJINST(MERGE)
:OBJECT.TYPE(TABLES)
:INSTANCE.
    NAME(IWH.ATOMICED)
    DBNAME(iwhtar)
    OWNER(IWH)
    TABLES(ATOMICED)
    TBLISBIN(N)
    TBLFTYPE(0)
    TBLLINAM(N)
    CREATYPE(4)
:COMMENT.
:COMMENT. End TABLES Instance
```

図 16. リレーショナル表用の TABLES オブジェクト

126ページの図17 では、ファイル用の TABLES オブジェクトの例を示します。

TABLES オブジェクト

```
:COMMENT. Begin TABLES Instance
:COMMENT.
:ACTION.OBJINST(MERGE)
:OBJECT.TYPE(TABLES)
:INSTANCE.
  NAME(d:¥iwhdemo¥outcusti.txt)
  SHRTDESC(File containing operational data for Institutions Customers)
  DBNAME(TBC Operations)
  OWNER()
  TABLES(d:¥iwhdemo¥outcusti.txt)
  TBLISBIN(Y)
  TBLFTYPE(3)
  TBLLINAM(N)
  CREATYPE(1)
:COMMENT.
:COMMENT. End TABLES Instance
```

図 17. ファイル用の TABLES オブジェクト

関係

表58 では、TABLES オブジェクトが参加する関係をリストします。これは、パートナー・アプリケーションにとって有用です。「ソース」欄と「ターゲット」欄では、関係のソース・オブジェクトまたはターゲット・オブジェクトが関係に参加できる回数が示されています。

表 58. TABLES オブジェクトが参加する関係

ソース	ソースのタグ 言語オブジェク ト・タイプ		ターゲット	ターゲットのタグ 言語オブジェク ト・タイプ	
	関係タイプ	説明		説明	説明
1	DATABASE	CONTAIN	M	TABLES	この表またはファイルが関連付けられるデータベースまたはファイル・システム。
1	TABLE	CONTAIN	M	COLUMN	この表に関連付けられる列。

127ページの図18 では、TABLES オブジェクトと DATABASE オブジェクトの間の関係の例を示します。


```
:COMMENT. Relation: DATABASE to TABLES
:COMMENT.
:ACTION.RELATION(ADD)
:RELTYPE.TYPE(CONTAIN) SOURCETYPE(DATABASE) TARGETTYPE(TABLES)
:INSTANCE.
    SOURCEKEY(NAME(TBC Operations) DBNAME() )
    TARGETKEY(DBNAME(TBC Operations) OWNER() TABLES(d:¥iwhdemo¥outcusti.txt) )
```

図 18. TABLES オブジェクトの DATABASE オブジェクトへのリンク

図19 では、TABLES オブジェクトと COLUMN オブジェクトの間の関係の例を示します。

```
:COMMENT. Relation: TABLES to COLUMN
:COMMENT.
:ACTION.RELATION(ADD)
:RELTYPE.TYPE(CONTAIN) SOURCETYPE(TABLES) TARGETTYPE(COLUMN)
:INSTANCE.
    SOURCEKEY(DBNAME(TBC Operations) OWNER() TABLES(d:¥iwhdemo¥outcusti.txt) )
    TARGETKEY(DBNAME(TBC Operations) OWNER() TABLES(d:¥iwhdemo¥outcusti.txt)
    COLUMNS(Zipcode) )
```

図 19. TABLES オブジェクトの COLUMN オブジェクトへのリンク

COLUMN オブジェクト

COLUMN オブジェクトには、ソース表、ターゲット表、またはファイル内の列またはフィールドに関するメタデータが含まれます。このオブジェクトは、TABLES オブジェクトに関連付けられます (121ページの『TABLES オブジェクト』を参照)。

プロパティ

表59 では、COLUMN オブジェクトのプロパティに関する情報を提供します。

表 59. COLUMN オブジェクトのプロパティ

タグ言語の		
プロパティ名	説明	許容値
NAME	列またはフィールドの名前。 データベース名、表名、および列名の組み合わせは固有です。	テキスト・ストリング (80 バイトまで)。
SHRTDESC	列またはフィールドの短い記述。	テキスト・ストリング (200 バイトまで)。

COLUMN オブジェクト

表 59. COLUMN オブジェクトのプロパティ (続き)

タグ言語の プロパティ名	説明	許容値
LONGDESC	列またはフィールドの長い記述。	テキスト・ストリング (32700 バイトまで)。
DATATYPE	データベース管理プログラム・データ・タイプからマップされる ODBC データ・タイプ。 データウェアハウスセンターは、ネイティブ・データ・タイプからデータ・タイプを導出します。 VSAM データベース内の表に GRAPHIC データ・タイプを追加することはできません。	次のいずれかの値: CHAR NUMERIC DECIMAL INTEGER SMALLINT FLOAT DOUBLE DATE TIME TIMESTAMP VARCHAR LONG_VARCHAR GRAPHIC VARGRAPHIC LONG_VARGRAPHIC BLOB CLOB DBCLOB TINYINT BIT REAL BIGINT
LENGTH	列またはフィールドの長さ。	数値。
SCALE	DECIMAL データ・タイプの列またはフィールドの場合、列またはフィールドの精度。	数値。
POSNO	表またはファイルの行における列またはフィールドの索引 (0 から始まる)。	数値。
NULLS	列またはフィールドでヌル・データが許可されるかどうかを指定するフラグ。	次のいずれかの値: Y 列ではヌル・データが許可される。 N 列ではヌル・データが許可されない。

表 59. COLUMN オブジェクトのプロパティ (続き)

タグ言語の プロパティ名	説明	許容値
ISTEXT	列またはフィールド・データがバイナリー・データか、テキスト・データかを指定するフラグ。	次のいずれかの値: Y 列データはバイナリー・データ。 N 列データはテキスト・データ。
DBNAME	この表またはファイルを含むソースまたはターゲットのビジネス名。	テキスト・ストリング (80 バイトまで)。
OWNER	表の所有者、高位修飾子、またはコレクション。 ファイルまたは IMS セグメントの場合、このプロパティはヌルです。	テキスト・ストリング (15 バイトまで)。
TABLES	データベース管理プログラムまたはファイル・システムに定義された物理表、ファイル、またはセグメント名。 ファイルおよび IMS セグメントの場合、この値は NAME の値と同じです。	テキスト・ストリング (80 バイトまで)。
NATIVEDT	列またはフィールドのネイティブ・データ・タイプ。	データベース管理プログラムに定義された列のデータ・タイプ。 データ・タイプは、40 バイトまでのテキスト・ストリングです。 多くの場合、このプロパティの値は DATATYPE の値と一致します。 データベース管理プログラム・データ・タイプの ODBC データ・タイプへのマッピングについては、データウェアハウスセンターのオンライン・ヘルプを参照してください。
ORDINAL	列またはフィールドの序数。	数値。
OFFSET	固定長ファイルにおけるフィールドのオフセット。	数値。

COLUMN オブジェクト

表 59. COLUMN オブジェクトのプロパティ (続き)

タグ言語の プロパティ名	説明	許容値
COLTYPE	DPROPR 用の列タイプ。	次のいずれかの値: A 変更後イメージ列 B 変更前イメージ列

図20 では、COLUMN オブジェクトの例を示します。

```
:ACTION.OBJINST(MERGE)
:OBJECT.TYPE(COLUMN)
:INSTANCE.
  NAME(CORR_COEF)
  SHRTDESC(Correlation Coefficient)
  DATATYPE(DOUBLE)
  LENGTH(0)
  SCALE(0)
  POSNO(4)
  NULLS(Y)
  ITEXT(N)
  DBNAME(TRANSFORMER_TARGET)
  OWNER(IWH)
  TABLES(TR_CORRELATION_06)
  COLUMNS(CORR_COEFF)
  NATIVEDT(DOUBLE)
  TRANSNAM(Correlation Coefficient(r))
```

図 20. COLUMN オブジェクト

関係

131ページの表60 では、COLUMN オブジェクトが参加する関係を示します。この関係は、パートナー・アプリケーションにとって有用です。「ソース」欄と「ターゲット」欄では、関係のソース・オブジェクトまたはターゲット・オブジェクトが関係に参加できる回数が見られています。

表 60. COLUMN オブジェクトが参加する関係

ソース	ソースのタグ 言語オブジェクト		ターゲット	ターゲットのタグ 言語オブジェクト・	
	ト・タイプ	関係タイプ		タイプ	説明
1	TABLES	CONTAIN	M	COLUMN	この列が関連付けられる表。

図21 では、COLUMN オブジェクトと TABLES オブジェクトの間の関係の例を示します。

```
:COMMENT. Relation: TABLES to COLUMN
:COMMENT.
:ACTION.RELATION(ADD)
:RELTYPE.TYPE(CONTAIN) SOURCETYPE(TABLES) TARGETTYPE(COLUMN)
:INSTANCE.
SOURCEKEY(DBNAME(TBC Operations) OWNER() TABLES(d:¥iwhdemo¥outcusti.txt) )
TARGETKEY(DBNAME(TBC Operations) OWNER() TABLES(d:¥iwhdemo¥outcusti.txt)
COLUMNS(Zipcode) )
```

図 21. COLUMN オブジェクトの TABLES オブジェクトへのリンク

第7章 情報カタログ・マネージャー・システム表およびメタデータ・モデル

以下の表は、情報カタログ・マネージャー・システムによる使用のために定義されています。

- 付加関係表: FLG.ATCHREL
- チェックポイント作業表: FLG.CHECKPT
- コメント表: FLG.COMMENTS
- 交換表: FLG.EXCHANGE
- ヒストリー表: FLG.HISTORY
- オブジェクト名インスタンス表: FLG.NAMEINST
- オブジェクト・タイプ登録表: FLG.OBJTYREG
- 長い記述のオーバーフロー表: FLG.OVERDESC
- システム・パラメーター表: FLG.PARMS
- プログラム表: FLG.PROGRAMS
- オブジェクト・タイプのプロパティ表: FLG.PROPERTY
- 関係インスタンス表: FLG.RELINST
- ユーザー表: FLG.USERS
- Windows アイコン表: FLG.WINICON

FLG.ATCHREL 表

FLG.ATCHREL 表は、オブジェクトとコメントの間関係を定義するために使用されます。

RELTYPE、SOURCE、および TARGET 列は、表の基本キーを形成します。

RELTYPE 列は、表の索引です。

134ページの表61 では、FLG.ATCHREL 表の各列に関する情報を提供します。

表 61. FLG.ATCHREL 表の列のプロパティ

列名	データ・タイプ	説明	ヌル可能	NLS
RELTYPE	CHAR(1)	関係タイプ: A 付加関係 L リンク関係 M コメント関係	いいえ	SBCS
SOURCE	CHAR(16)	ソース・オブジェクトを表す FLGID。	いいえ	SBCS
TARGET	CHAR(16)	ターゲット・オブジェクトを表す FLGID。	いいえ	SBCS
注:				
NLS: 各国語サポート				
SBCS: 1 バイト文字セット				
DBCS: 2 バイト文字セット				

FLG.CHECKPT 表

FLG.CHECKPT 表は、インポート API で、あるチェックポイントでインポート・プロセスを再開するために使用されます。

この表は、インポート API によってデータが取り込まれます。この表には、常に、ゼロ個以上の行が含まれる可能性があります。

TAGFNAME 列は、表の基本キーです。

COMMITID、LASTUPDT、および USERID 列は、すべて表の索引です。

表62 では、FLG.CHECKPT 表の各列に関する情報を提供します。

表 62. FLG.CHECKPT 表の列のプロパティ

列名	データ・タイプ	説明	ヌル可能	NLS
TAGFNAME	VARCHAR(240)	タグ言語ファイルの名前 (パス情報を除く)。	いいえ	SBCS と DBCS の 両方

表 62. FLG.CHECKPT 表の列のプロパティ (続き)

列名	データ・タイプ	説明	ヌル可能	NLS
COMMITID	CHAR(26)	最後の COMMIT チェックポイントの ID。この ID は、タグ言語ファイル内の適切な場所に置かれた COMMIT タグで、ユーザーによって提供されます。ID は、システム・タイム・スタンプまたは任意の文字の列になります。	いいえ	SBCS と DBCS の両方
LASTUPDT	TIMESTAMP	このエントリーが作成または更新された時点のシステム・タイム・スタンプ。最終更新フィールドは、常に 26 バイト全体を占有するため、埋め込みを必要としません。	いいえ	なし
USERID	CHAR(8)	情報カタログ管理者のユーザー ID。	いいえ	SBCS と DBCS の両方
ENTSAVED	INTEGER	保管域に保管されているエントリーの合計数。	いいえ	なし
SAVEAREA	LONG VARCHAR	オブジェクト・タイプ名のリスト用のストレージ域。各オブジェクト・タイプ名は 8 バイトです。	いいえ	SBCS
注:				
NLS: 各国語サポート				
SBCS: 1 バイト文字セット				
DBCS: 2 バイト文字セット				

FLG.COMMENTS 表

FLG.COMMENTS 表には、情報カタログ・マネージャーの情報カタログ内のオブジェクトに関するすべてのコメントが含まれます。

この表には、常に、ゼロ個以上の行が含まれる可能性があります。

INSTIDNT 列は、表の基本キーです。

NAME、CREATOR、および CREATSTP 列は、表の固有索引を形成します。

NAME、CREATOR、CREATSTP、および UPDATIME 列は、表の索引です。

表63 では、FLG.COMMENTS 表の各列に関する情報を提供します。

表 63. FLG.COMMENTS 表の列のプロパティ

列名	データ・タイプ	説明	ヌル可能	NLS
OBJTYPID	CHAR(6)	情報カタログ・マネージャーによって生成された 6 桁のオブジェクト・タイプ ID。この ID は、情報カタログ内の特定のオブジェクト・タイプを表します。	いいえ	SBCS
INSTIDNT	CHAR(10)	情報カタログ・マネージャーによって生成された固有なインスタンス ID。この ID は、FLGID の 2 番目の部分であり、オブジェクト・タイプの中でこのインスタンスを固有に識別する 10 桁のシリアル番号です。	いいえ	SBCS
NAME	VARCHAR(80)	それぞれのユーザー定義オブジェクトを識別するために、情報カタログ・ユーザーによって入力された名前。	いいえ	SBCS と DBCS の両方
UPDATIME	CHAR(26)	メタデータの作成または最終更新の日時。この値は情報カタログ・マネージャーによって生成されます。	はい	なし
UPDATEBY	CHAR(8)	最後にインスタンスを更新した情報カタログ管理者のユーザー ID。	はい	SBCS と DBCS の両方
CREATOR	CHAR(8)	コメント・オブジェクトの作成者。システムは、作成者の値を現行ユーザー ID に設定します。	いいえ	SBCS と DBCS の両方
CREATSTP	CHAR(26)	コメント・オブジェクトが作成された日時を示すタイム・スタンプ。このタイム・スタンプは、インスタンスの作成時にシステムによって提供されます。	いいえ	なし
STATUS	CHAR(80)	コメントの状況。ユーザーは、この値についての独自の規則を作成することができません。	はい	SBCS と DBCS の両方
ACTIONS	VARCHAR(250)	ユーザーが行わなければならない処置を指定します。	はい	SBCS と DBCS の両方
EXTRA	VARCHAR(80)	エクストラ情報のために使用されます。	はい	SBCS と DBCS の両方

表 63. FLG.COMMENTS 表の列のプロパティ (続き)

列名	データ・タイプ	説明	ヌル可能	NLS
注:				
NLS:		各国語サポート		
SBCS:		1 バイト文字セット		
DBCS:		2 バイト文字セット		

FLG.EXCHANGE 表

FLG.EXCHANGE 表は、情報カタログ・マネージャー、データウェアハウスセンター、および DB2 OLAP Server™ の間で同期化されたオブジェクトを記録するために使用されます。

この表は、インストール時に、メタデータ交換プログラムによってデータが取り込まれます。

OBJNAME および OBJTYPE 列は、表の基本キーを形成します。

表64 では、FLG.EXCHANGE 表の各列に関する情報を提供します。

表 64. FLG.EXCHANGE 表の列のプロパティ

列名	データ・タイプ	説明	ヌル可能	NLS
PRODUCT	VARCHAR(40)	製品、バージョン、およびリリース番号の組み合わせ。	いいえ	SBCS
OBJNAME	VARCHAR(200)	オブジェクト名 (たとえば、ステップ)。	いいえ	SBCS と DBCS の両方
IMPDATE	TIMESTAMP	インポートのタイム・スタンプ。	いいえ	なし
OBJTYPE	CHAR(5)	OBJTYPE は、次のいずれかの値になります。 <ul style="list-style-type: none"> • IR (交換されたソース・メタデータを表す) • DR (ターゲット・メタデータを表す) • BV (ステップ・メタデータを表す) • OLAP (OLAP メタデータを表す) 	いいえ	SBCS

表 64. FLG.EXCHANGE 表の列のプロパティ (続き)

列名	データ・タイプ	説明	ヌル可能	NLS
注:				
NLS: 各国語サポート				
SBCS: 1 バイト文字セット				
DBCS: 2 バイト文字セット				

FLG.HISTORY 表

FLG.HISTORY 表は、情報カタログ・マネージャーおよびデータウェアハウスセンターから削除されたオブジェクトを記録するために使用されます。

この表は、ユーザーがオブジェクトを削除し、削除履歴の記録フラグが ON になっているときに、データが取り込まれます。この表には、常に、ゼロ個以上の行が含まれる可能性があります。

HISSEQ 列は、表の基本キーです。

表65 では、FLG.HISTORY 表の各列に関する情報を提供します。

表 65. FLG.HISTORY 表の列のプロパティ

列名	データ・タイプ	説明	ヌル可能	NLS
HISSEQ	TIMESTAMP	削除履歴のシーケンス番号。	いいえ	なし
HISTYPE	INTEGER	削除履歴のタイプ。 <ul style="list-style-type: none"> この列の値が 1 の場合は、情報カタログからの削除を示します。 この列の値が 2 の場合は、データウェアハウスセンターからの削除を示します。 	いいえ	なし
HISTAG	LONG VARCHAR	この列には、削除されるオブジェクトの ID が保管されます。	はい	SBCS と DBCS の 両方
注:				
NLS: 各国語サポート				
SBCS: 1 バイト文字セット				
DBCS: 2 バイト文字セット				

FLG.NAMEINST 表

FLG.NAMEINST 表には、情報カタログ内のそれぞれのオブジェクトの名前が含まれます。

FLGID 列は、表の基本キーです。

INSTNAME および TYPENAME 列は、表の索引です。

表66 では、FLG.NAMEINST 表の各列に関する情報を提供します。

表 66. FLG.NAMEINST 表の列のプロパティ

列名	データ・タイプ	説明	ヌル可能	NLS
FLGID	CHAR(16)	16 文字のオブジェクト ID。	いいえ	SBCS
TYPENAME	VARCHAR(80)	オブジェクト・タイプの外部名。	いいえ	SBCS と DBCS の 両方
INSTNAME	VARCHAR(80)	オブジェクトの外部名。	いいえ	SBCS と DBCS の 両方
注:				
NLS: 各国語サポート				
SBCS: 1 バイト文字セット				
DBCS: 2 バイト文字セット				

FLG.OBJTYREG 表

FLG.OBJTYREG 表は、すべてのオブジェクトとそれらのオブジェクト・タイプ、および情報カタログ・マネージャーによって作成された表を記録するために使用されます。

OBJTYPID 列は、FLG.OBJTYREG の基本キーで、情報カタログ内のオブジェクト・タイプを固有に識別し、すべてのインスタンス ID の接頭部として使用されます。

PTNAME、NAME、および DPNAME 列は、FLG.OBJTYREG 表の固有索引キーです。

CATEGORY、CREATOR、および UPDATEBY 列は、表の索引キーです。

表67 では、FLG.OBJTYREG 表の各列に関する情報を提供します。

表 67. FLG.OBJTYREG 表の列のプロパティ

列名	データ・タイプ	説明	ヌル可能	NLS
OBJTYPID	CHAR(6)	情報カタログ・マネージャーによって生成された 6 桁のオブジェクト・タイプ ID。この ID は、情報カタログ内の特定のオブジェクト・タイプを表します。	いいえ	SBCS
PTNAME	CHAR(30)	オブジェクト・タイプの名前。この名前は、ユーザーの表の名前としても使用されます。PTNAME の実際のサイズは、FLG.PARMS 表の ENVSIZE の値 (インストール時に定義される) によって判別されます。	いいえ	SBCS
DPNAME	CHAR(8)	情報カタログ内の固有なオブジェクト・タイプ名。	いいえ	SBCS
NAME	VARCHAR(80)	このオブジェクト・タイプの外部名。	いいえ	SBCS と DBCS の両方
CATEGORY	CHAR(1)	情報カタログ・マネージャー区分: Elemental (基本) - E、Grouping (グループ化) - G、Program (プログラム) - P、Contact (接点) - C、Dictionary (辞書) - D、Support (サポート) - S、および Attachment (付加) - A。	いいえ	SBCS
CREATOR	CHAR(8)	オブジェクト・タイプを作成した情報カタログ管理者のユーザー ID。オブジェクト・タイプが登録されるときには、これはブランクになります。オブジェクト・タイプが削除されたが、登録が除去されていないときにも、これはブランクになります。	はい	SBCS と DBCS の両方
UPDATIME	CHAR(26)	オブジェクト・タイプが作成されたか、またはそのプロパティが拡張された日時。	はい	SBCS
UPDATEBY	CHAR(8)	最後にオブジェクト・タイプを拡張した (プロパティを追加した) 情報カタログ管理者のユーザー ID。	はい	SBCS と DBCS の両方

表 67. *FLG.OBJTYREG* 表の列のプロパティ (続き)

列名	データ・タイプ	説明	ヌル可能	NLS
LASTINID	INTEGER	このオブジェクト・タイプについての最後のシステム生成インスタンス ID。 これは内部プロパティであり、情報カタログ・ユーザーにとって不可視です。このプロパティは、インスタンスの作成 IPI によってのみアクセスされ、更新されません。	いいえ	なし
OBJJCON	LONG VARCHAR FOR BIT DATA	オブジェクト・タイプに対応するアイコンのビットマップ。	いいえ	なし
注:				
NLS: 各国語サポート				
SBCS: 1 バイト文字セット				
DBCS: 2 バイト文字セット				

FLG.OVERDESC 表

FLG.OVERDESC 表には、すべての長い記述のプロパティが含まれます。それぞれの長い記述は、3 KB のチャンクに分割されます。

OBJTYPID、INSTIDNT、PHYPRPNM、および SEQNO 列は、FLG.OVERDESC 表の基本キーを形成します。

表 68 では、FLG.OVERDESC 表の各列に関する情報を提供します。

表 68. *FLG.OVERDESC* 表の列のプロパティ

列名	データ・タイプ	説明	ヌル可能	NLS
OBJTYPID	CHAR(6)	情報カタログ・マネージャーによって生成された 6 桁のオブジェクト・タイプ ID。 この ID は、情報カタログ内の特定のオブジェクト・タイプを表します。	いいえ	SBCS

表 68. FLG.OVERDESC 表の列のプロパティ (続き)

列名	データ・タイプ	説明	ヌル可能	NLS
INSTIDNT	CHAR(10)	情報カタログ・マネージャーによって生成された固有なインスタンス ID。この ID は、FLGID の 2 番目の部分であり、オブジェクト・タイプの中でこのインスタンスを固有に識別するシリアル番号の 10 桁部分です。	いいえ	SBCS
PHYPRPNM	CHAR(8)	ユーザーによって定義された元のプロパティまたは列名。	いいえ	SBCS
SEQNO	SMALLINT	同じ入力ソースを反映する行の数を記録するためのシーケンス番号。	いいえ	なし
ODESC	VARCHAR(3000)	このエントリーでは、長い記述 (最大 32700 バイト) のセグメントがより小さく、より扱いやすいバッファーに保管されます。	いいえ	SBCS と DBCS の両方
注:				
NLS: 各国語サポート				
SBCS: 1 バイト文字セット				
DBCS: 2 バイト文字セット				

FLG.PARMS 表

FLG.PARMS 表には、メタデータは含まれません。情報カタログ・マネージャーの内部のグローバル・パラメーターが含まれます。この表は、情報カタログ・マネージャーの永続的なパラメーター (たとえば、バージョン、ログオン・メッセージ、およびコード・ページ) 用のグローバル・ストレージ域です。

FLG.PARMS には、システム・パラメーターが保管されます。この表の値は、情報カタログ・マネージャーのカタログ作成ユーティリティ (情報カタログ・マネージャー 管理の手引き を参照) を使用する際に設定されます。また、情報カタログ・マネージャー API (情報カタログ・マネージャー プログラミングの手引きおよび解説書 を参照) を使用して、値を変更することもできます。

143ページの表69 では、FLG.PARMS 表の各列に関する情報を提供します。

表 69. FLG.PARMS 表の列のプロパティ

列名	データ・タイプ	説明	ヌル可能	NLS
VERSION	CHAR(20)	情報カタログのバージョン (たとえば、V1R0M0 または V1R1M0)。このデータは、インストールまたは移行時に取り込まれます。	はい	SBCS
LOGONMSG	VARCHAR(254)	情報カタログ・マネージャーのログオン・メッセージ (たとえば、"情報カタログ・マネージャーへようこそ!")。	はい	SBCS と DBCS の 両方
CODEPAGE	CHAR(4)	情報カタログのコード・ページ番号。	はい	SBCS
LANGUAGE	CHAR(4)	言語コード (たとえば、米国英語を表す ENU)。このデータは、ストリング・ファイルからロードされます。	はい	SBCS
DTOKEN	CHAR(1)	未指定データ・フィールドを表現するために使用される、情報カタログ・マネージャー環境のデフォルト・トークン。この適用不能記号は、インポートおよびエクスポート機能によって使用されます。 この値は、インストール時に設定されません。	はい	SBCS
ENVSIZE	SMALL INTEGER	データベース・サーバーの環境サイズ。 この値は、インストール時に設定され、情報カタログ・マネージャー表、列、および索引の適切な名前の長さを指定するために使用されます。 この値は、DB2 UDB (AS/400 版) の場合は 10、他の大部分の IBM リレーショナル・データベースの場合は 18、非 IBM データベースの場合は最大 30 バイトになります。	はい	なし
LASTYPID	INTEGER	あるオブジェクト・タイプについての最後のシステム生成 ID。この ID は、登録の作成 IPI によってアクセスされ、更新されません。	はい	なし
LISTMAX	INTEGER	リストまたは検索結果からの検索可能オブジェクトの最大数。	はい	なし
ISTGROUP	CHAR(8)	DB2 for OS/390 [®] (DB2 (OS/390 版)) データベース用の索引ストレージ・グループの名前。	はい	SBCS

表 69. FLG.PARMS 表の列のプロパティ (続き)

列名	データ・タイプ	説明	ヌル可能	NLS
TSTGROUP	CHAR(8)	DB2 (OS/390 版) データベース用の表ストレージ・グループの名前。	はい	SBCS
MDBNAME	CHAR(8)	DB2 (OS/390 版) データベースの名前。	はい	SBCS
TBSPAC32	CHAR(8)	DB2 (OS/390 版) データベース用の 32 KB 表スペースの名前。	はい	SBCS
TBSPAC04	CHAR(8)	DB2 (OS/390 版) データベース用の 4 KB 表スペースの名前。	はい	SBCS
PARMFLAG	INTEGER	フラグ標識。 FLG_PARMS_RECORD_DELETE_HISTORY 削除ヒストリーを記録する。 FLG_PARMS_MVS_FOLD_UP DB2 (OS/390 版) 情報カタログで、オブジェクト値を大文字で保管する。情報カタログ・マネージャーでは、これらの値を大文字または小文字で検索することができます。	はい	なし
CMTSTAT	VARCHAR(800)	この列には、コメントの状況のリストが保管されます。それぞれの状況は 80 バイトです。	はい	SBCS と DBCS の 両方
注:				
NLS: 各国語サポート				
SBCS: 1 バイト文字セット				
DBCS: 2 バイト文字セット				

FLG.PROGRAMS 表

FLG.PROGRAMS 表は、情報カタログ内のすべてのプログラム・オブジェクトを記録するために使用されます。

INSTIDNT 列は、FLG.PROGRAMS 表の基本キーです。

UUICLASS、UUIQUAL1、UUIQUAL2、UUIQUAL3、および UUIIDENT 列は、FLG.PROGRAMS 表の固有索引を形成します。

NAME、UPDATEBY、UPDATIME、UICLASS、UIQUAL1、UIQUAL2、UIQUAL3、UIDENT、および HANDLES 列は、表の索引です。

表70 では、FLG.PROGRAMS 表の各列に関する情報を提供します。

表 70. FLG.PROGRAMS 表の列のプロパティ

列名	データ・タイプ	説明	原点	NLS
OBJTYPID	CHAR(6)	情報カタログ・マネージャーによって生成された 6 桁のオブジェクト・タイプ ID。この ID は、特定のオブジェクト・タイプを表します。	いいえ	SBCS
INSTIDNT	CHAR(10)	情報カタログ・マネージャーによって生成された固有なインスタンス ID。この ID は、FLGID の 2 番目の部分であり、オブジェクト・タイプの中でこのインスタンスを固有に識別する 10 桁のシリアル番号です。	いいえ	SBCS
NAME	VARCHAR(80)	この名前は、それぞれのユーザー定義オブジェクトを識別するために、情報カタログ・ユーザーによって入力されます。	いいえ	SBCS と DBCS の両方
UPDATIME	CHAR(26)	メタデータの作成または最終更新の日時。この値は情報カタログ・マネージャーによって生成されます。	はい	SBCS
UPDATEBY	CHAR(8)	最後にインスタンスを更新した情報カタログ管理者のユーザー ID。	はい	SBCS と DBCS の両方
UICLASS	CHAR(25)	汎用固有 ID (UII) の部分 1 の名前。	いいえ	SBCS と DBCS の両方
UIQUAL1	VARCHAR(48)	UII の部分 2 の名前。	いいえ	SBCS と DBCS の両方
UIQUAL2	VARCHAR(48)	UII の部分 3 の名前。	いいえ	SBCS と DBCS の両方
UIQUAL3	VARCHAR(48)	UII の部分 4 の名前。	いいえ	SBCS と DBCS の両方

表 70. FLG.PROGRAMS 表の列のプロパティ (続き)

列名	データ・タイプ	説明	原点	NLS
UIDENT	VARCHAR(70)	UUI の部分 5 の名前。	いいえ	SBCS と DBCS の両方
HANDLES	CHAR(8)	このプログラムによって処理されるオブジェクト・タイプ。	はい	SBCS
STARTCMD	VARCHAR(250)	呼び出されるプログラム名。プログラムは、.exe、.cmd、.com、または .bat の拡張子を持つ可能性があります。	いいえ	SBCS と DBCS の両方
PARMLIST	VARCHAR(1800)	オブジェクトを処理するためにパラメーター・リストが必要な場合は、PARMLIST プロパティによってパラメーターの値が指定されます。	はい	SBCS と DBCS の両方
SHRTDESC	VARCHAR(250)	プログラムの短い記述。	はい	SBCS と DBCS の両方
注:				
NLS: 各国語サポート				
SBCS: 1 バイト文字セット				
DBCS: 2 バイト文字セット				

FLG.PROPERTY 表

FLG.PROPERTY 表は、あるオブジェクト・タイプのプロパティを定義するために使用されます。この表で定義されるそれぞれのオブジェクト・タイプのプロパティごとに、1 つの行があります。情報カタログ・マネージャー・オブジェクト・タイプとオブジェクト・タイプのプロパティについては、163 ページの『第8章 情報カタログ・マネージャー・オブジェクト・タイプ』を参照してください。

OBJTYPID 列は、表の索引です。

147ページの表71 では、FLG.PROPERTY 表の各列に関する情報を提供しません。

表 71. FLG.PROPERTY 表の列のプロパティー

列名	データ・タイプ	説明	ヌル可能	NLS
OBJTYPID	CHAR(6)	それぞれのオブジェクト・タイプを表す 6 桁の固有なシステム生成 ID。	いいえ	SBCS
PHYPRPNM	CHAR(8)	オブジェクト・タイプ内のプロパティーの物理名。この名前は、ユーザーのオブジェクト表で列名を生成するために使用されま す。	いいえ	SBCS
PROPNAME	VARCHAR(80)	このオブジェクト・タイプ・プロパティーの外部名。	いいえ	SBCS と DBCS の 両方
DATATYPE	CHAR(30)	プロパティーのデータ・タイプ (CHAR、VARCHAR、LONG VARCHAR、 および TIMESTAMP)。	いいえ	SBCS
LENGTH	INTEGER	プロパティーの長さ。	いいえ	なし
OPTIONS	CHAR(1)	このフィールドでヌル値が許可されるかど うかを示す値フラグ。 R 値は必須 (ヌル不可) O 値はオプション (ヌル可能) S システム生成の値	いいえ	SBCS
UISEQNO	CHAR(1)	オブジェクト・タイプ内のプロパティーの UII シーケンス番号。	はい	SBCS
PROPSEQ	INTEGER	プロパティーのシーケンス番号。	いいえ	なし
注:				
NLS: 各国語サポート				
SBCS: 1 バイト文字セット				
DBCS: 2 バイト文字セット				

FLG.RELINST 表

FLG.RELINST 表では、2 つのオブジェクトの間関係が定義されます。この表には、ソース・オブジェクトとターゲット・オブジェクトの関係ごとに、1 つの行が入ります。

RELTYPE、SOURCE、および TARGET 列は、表の基本キーを形成します。

RELTYPE、SRCCAT、SOURCE、SRCTNAME、SRCINAME、TRGCAT、TARGET、TRGTNAME、および TRGINAME 列は、表の索引です。

表72 では、FLG.RELINST 表の各列に関する情報を提供します。

表 72. FLG.RELINST 表の列のプロパティ

列名	データ・タイプ	説明	ヌル可能	NLS
RELTYPE	CHAR(1)	関係タイプ: C 包含 T 接点	いいえ	SBCS
SRCCAT	CHAR(1)	ソース・オブジェクトの区分。	いいえ	SBCS
SOURCE	CHAR(16)	ソース・オブジェクトを表す FLGID。	いいえ	SBCS
SRCTNAME	VARCHAR(80)	ソース・オブジェクト・タイプの外部名。	いいえ	SBCS と DBCS の 両方
SRCINAME	VARCHAR(80)	ソース・オブジェクトの外部名。	いいえ	SBCS と DBCS の 両方
TRGCAT	CHAR(1)	ターゲット・オブジェクトの区分。	いいえ	SBCS
TARGET	CHAR(16)	ターゲット・オブジェクトを表す FLGID。	いいえ	SBCS
TRGTNAME	VARCHAR(80)	ターゲット・オブジェクト・タイプの外部名。	いいえ	SBCS と DBCS の 両方
TRGINAME	VARCHAR(80)	ターゲット・オブジェクトの外部名。	いいえ	SBCS と DBCS の 両方
注:				
NLS: 各国語サポート				
SBCS: 1 バイト文字セット				
DBCS: 2 バイト文字セット				

FLG.USERS 表

FLG.USERS 表には、すべての情報カタログ管理者と、特殊な管理特権を持つユーザーのリストが含まれます。情報カタログ・マネージャーの他の保管表とは異なり、FLG.USERS 表にはメタデータが含まれません。この表には、さまざまなタイプの情報カタログ・ユーザーの定義と、それらのユーザーの状況が含まれます。

USERTYPE および DGUSER 列は、表の基本キーを形成します。

DGUSER 列は、表の索引です。

表73 では、FLG.USERS 表の各列に関する情報を提供します。

表 73. FLG.USERS 表の列のプロパティー

列名	データ・ タイプ	説明	ヌル可能	NLS
DGUSER	CHAR(8)	情報カタログ管理者のユーザー ID。この ID は、インストール時に入力されます。	いいえ	SBCS と DBCS の両方
USERTYPE	CHAR(1)	DGUSER のタイプ。このタイプは、情報カタログ管理者、特殊な更新特権を持つユーザー、またはユーザーになります。 この値は、インストール時に設定されます。	いいえ	SBCS
ACTIVEKA	CHAR(1)	情報カタログ・マネージャーに現在ログオンしている情報カタログ管理者を示すフラグ。一度に 1 人の情報カタログ管理者のみがログオンすることができます。	はい	SBCS
注: NLS: 各国語サポート SBCS: 1 バイト文字セット DBCS: 2 バイト文字セット				

FLG.WINICON 表

FLG.WINICON 表には、それぞれのオブジェクト・タイプごとに、関連する Windows アイコンが含まれます。

OBJTYPID 列は、表の基本キーです。

表74 では、FLG.WINICON 表の各列に関する情報を提供します。

表 74. FLG.WINICON 表の列のプロパティ

列名	データ・タイプ	説明	ヌル可能	NLS
OBJTYPID	CHAR(6)	6 桁のオブジェクト・タイプ ID。	いいえ	SBCS
OBJICON	LONG VARCHAR FOR BIT DATA (30000)	Windows アイコンのビットマップ。	はい	なし
注:				
NLS: 各国語サポート				
SBCS: 1 バイト文字セット				
DBCS: 2 バイト文字セット				

情報カタログ・マネージャーのメタデータ・モデル

以下の節では、情報カタログ・マネージャーのメタデータ・モデルについて説明します。『情報カタログ・マネージャー・システム表のモデル』では、情報カタログ・マネージャー・システム表の間の関係を説明します。155ページの『論理メタデータ・モデル』では、情報カタログ・マネージャーのオブジェクト・タイプ区分におけるオブジェクト間の関係を説明します。

情報カタログ・マネージャー・システム表のモデル

以下の図では、さまざまな情報カタログ・マネージャー・システム表や、オブジェクト・タイプ表の間の関係が示されています。関係の例としては、2 つの列の間の結合があります。以下の情報カタログ・マネージャー・システム表は、他のシステム表と関係しません。

- FLG.PARMS
- FLG.HISTORY
- FLG.USERS

- FLG.EXCHANGE
- FLG.CHECKPT

番号が付けられているそれぞれの関係については、この図の下の注を参照してください。

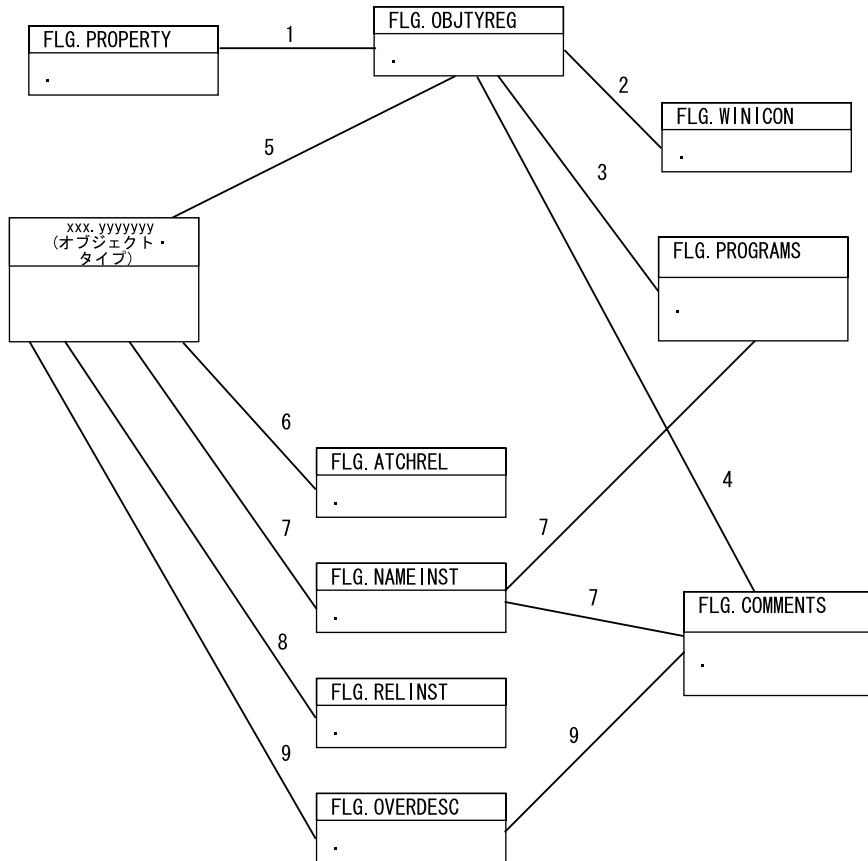


図22. 情報カタログ・マネージャー・システム表

図22 の注

1. 2 つの表の間関係は、表の OBJTYPID 列の値が等しい場合に存在します。関係は、OBJTYPID 列に基づく、2 つの表の間の結合です。
2. 2 つの表の間関係は、表の OBJTYPID 列の値が等しい場合に存在します。関係は、OBJTYPID 列に基づく、2 つの表の間の結合です。

- 2つの表の間関係は、表の DPNAME および HANDLES 列の値が等しい場合に存在します。関係は、DPNAME および HANDLES 列に基づく、2つの表の間の結合です。
- 表の間関係は、FLG.OBJTYREG 表の PTNAME および CREATOR 列と、FLG.COMMENTS 表の物理名から導出されます。
たとえば、図23 では、PTNAME 列の最初のエントリーが COMMENTS で、CREATOR 列の最初のエントリーが FLG です。これらの値を連結すると、完全修飾の表名 FLG.COMMENTS になります。

FLG.OBJTYREG

OBJTYPID	PTNAME	DPNAME	NAME	CATEGORY	CREATOR	...
000001	COMMENTS	COMMENTS	コメント	G	FLG	...
000002	PRESENT	PRESENT	プレゼンテーション	E	DGADMIN	...
000003	COLUMNS	COLUMN	リレーショナルDB中の列またはフィールド	G	DGADMIN	...

FLG.COMMENTS

OBJTYPID	INSTIDNT	Name	UPDATIME	UPDATEBY	SHRTDESC
000001	0000016465	「個人プレゼンテーション」オブジェクトのコメント
000001	0000003435	XYZ プレゼンテーションのコメント
000001	0000064459	コメント 3

DGADMIN.PRESENT

OBJTYPID	INSTIDNT	Name	UPDATIME	UPDATEBY	SHRTDESC
000002	0000001111	個人プレゼンテーション	プレゼンテーション・オブジェクト
0000021	0000002222	XYZ プレゼンテーション	情報カタログにある別のプレゼンテーション・オブジェクト

図 23. FLG.OBJTYREG 表とオブジェクト・タイプ表の間関係

- FLG.OBJTYREG 表とオブジェクト・タイプ表の間関係は、FLG.OBJTYREG 表の PTNAME 列と CREATOR 列を連結することによって導出されます。結果の名前は、オブジェクト・タイプ表の名前です。

たとえば、152ページの図23 では、PTNAME 列の 2 番目のエントリーが PRESENT で、 CREATOR 列の 2 番目のエントリーが DGADMIN です。これらの値を連結すると、完全修飾名 DGADMIN.PRESENT になります。

6. 関係がタイプ A (付加) の場合、FLG.ATCHREL 表に保管される関係は、ソース表のオブジェクト・タイプ ID およびインスタンス ID をターゲット表のオブジェクト・タイプ ID およびインスタンス ID に連結することによって導出されます。

たとえば、154ページの図24 では、DGADMIN.PRESENT のオブジェクト・タイプとインスタンス ID が FLG.ATCHREL 表のソース列で連結されます。DGADMIN.PRESENT のプレゼンテーション・オブジェクトに付加される、関連するコメントのオブジェクト・タイプとインスタンス ID を連結したものがターゲット列に保管されます。

FLG.COMMENTS

OBJTYPID	INSTIDNT	Name	UPDATIME	UPDATEBY	SHRTDESC
000001	0000016465	「個人プレゼンテーション」オブジェクトのコメント
000001	0000064459	コメント 3
000001	0000003435	XYZ プレゼンテーションのコメント

FLG.ATCHREL

RELTYPE	SOURCE	TARGET
A	0000020000001111	0000010000016465
A	0000020000002222	0000010000003435
A	0000030000123456	0000010000004459

DGADMIN.PRESENT

OBJTYPID	INSTIDNT	Name	UPDATIME	UPDATEBY	SHRTDESC
000002	0000001111	個人プレゼンテーション	プレゼンテーション・オブジェクト
000002	0000002222	XYZ プレゼンテーション	情報カタログにある別のプレゼンテーション・オブジェクト

図 24. FLG.ATCHREL 表、ソース、およびターゲットの間の関係

- それぞれの表の対の間の関係は、表の FLGID から導出されます。FLGID は、表の OBJTYPID 列と INSTIDNT 列を連結したものです。
- FLG.RELINST に保管される関係は、包含、リンク、および接点です。(オブジェクト区分の関係については、155ページの『論理メタデータ・モデル』を参照してください。) 関係は、ソース表とターゲット表の FLGID から導出されます。情報カタログ・マネージャー・オブジェクト・タイプの詳細については、168ページの『事前定義の情報カタログ・マネージャー・オブジェクト・タイプ』を参照してください。

9. それぞれの表の対の間の関係は、2 つの表の FLGID から導出されます。FLG.OVERDESC 表には、複数行のデータが含まれる可能性があります。その場合、行は、FLG.OVERDESC 表の SEQNO 列によって順序付けられます。

論理メタデータ・モデル

それぞれのオブジェクト・タイプは、情報カタログ・マネージャー区分に属さなければなりません。オブジェクト・タイプの区分は、情報カタログ・マネージャーがそれを処理する方法に影響を与えます。以下のリストでは、情報カタログ・マネージャー区分のそれぞれで作成することができるオブジェクト・タイプについて説明します。

グループ化

他のオブジェクト・タイプを含むことができるオブジェクト・タイプ。

基礎的 他の情報カタログ・マネージャー・オブジェクト・タイプの組み立てブロックとなる非グループ化オブジェクト・タイプ。

コンタクト

オブジェクトに関する詳細情報への参照を識別するオブジェクト・タイプ。詳細情報には、オブジェクトが表現する情報を作成した人の名前や、情報の保守に責任がある部門などが含まれます。

プログラム

情報カタログ・マネージャー・オブジェクト・タイプによって表現される実際の情報を処理することができるアプリケーションを識別し、記述するプログラム・オブジェクト・タイプ。プログラム区分に属する唯一のオブジェクト・タイプはプログラムであり、これは情報カタログの作成時に定義されます。

辞書 ビジネスに特有の用語を定義するオブジェクト・タイプ。

サポート

情報カタログまたは企業に関する追加情報を提供するオブジェクト・タイプ。

接続 別の情報カタログ・マネージャー・オブジェクトに付加される追加情報を識別するコメント・オブジェクト・タイプ。接続区分に属する唯一のオブジェクト・タイプはコメントであり、これは情報カタログの作成時に定義されます。

156ページの表75 では、情報カタログ・マネージャーのオブジェクト・タイプ区分の間の関係を要約します。157ページの図25 では、これらの関係を図で表現します。

表 75. 情報カタログ・マネージャー区分の関係

区分	～を含むことができる / ～に含まれる	～とリンクする	接点に関連付けられる	コメントが付加される	～からプログラムが立ち上げられる
グループ化	他のグループ化または基礎的オブジェクトを含む	他のグループ化または基礎的オブジェクト	はい	はい	はい
基礎的	グループ化オブジェクトに含まれる	他のグループ化または基礎的オブジェクト	はい	はい	はい
コンタクト	なし	なし	いいえ	はい	はい
プログラム	なし	なし	いいえ	はい	いいえ
辞書	なし	なし	いいえ	はい	はい
サポート	なし	なし	いいえ	はい	はい
接続	なし	なし	いいえ	いいえ	はい

情報カタログ用のオブジェクト・タイプは、次のいずれかの方法で確立することができます。

- サンプル情報カタログで情報カタログ・マネージャーと一緒に提供されるオブジェクト・タイプを使用する (サンプル情報カタログの作成に関する情報と、それに含まれるオブジェクト・タイプの説明については、168ページの『事前定義の情報カタログ・マネージャー・オブジェクト・タイプ』を参照してください)。
- 情報カタログ・マネージャーと一緒に提供されるオブジェクト・タイプを組織の要件に合わせて変更する (オブジェクト・タイプの変更については、情報カタログ・マネージャー 管理の手引き を参照してください)。
- 独自のオブジェクト・タイプを作成する。

157ページの図25 では、オブジェクト・タイプ区分内のオブジェクト間関係を示します。図で、オブジェクト・タイプ区分名を囲んでいる括弧は、そのオブジェクト・タイプ区分が拡張不能であることを示します。オブジェクト・タイプ名を囲んでいる括弧は、そのオブジェクト・タイプが拡張不能であることを示します。拡張可能なオブジェクト・タイプの詳細については、163ページの『第8章 情報カタログ・マネージャー・オブジェクト・タイプ』を参照してください。

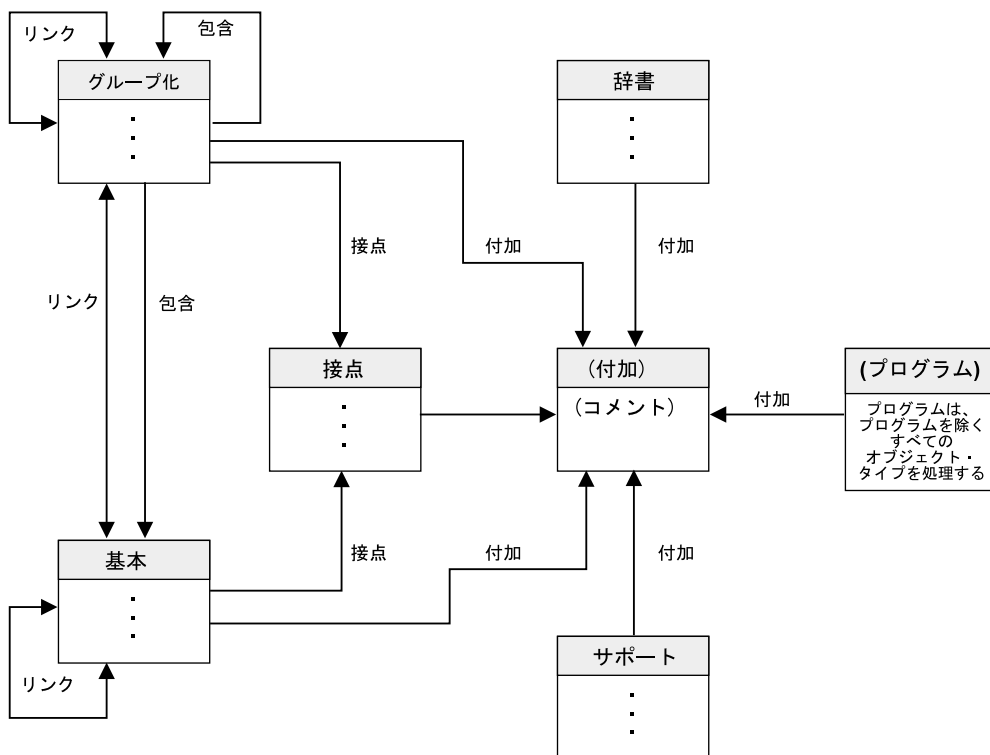


図 25. オブジェクト・タイプ区分間の関係

上の図25 では、以下の関係が示されています。

包含 1つのオブジェクトは複数のオブジェクトを含むことができ、また、複数のオブジェクトに含まれることができます。

たとえば、1つのグループ化オブジェクトは複数の基礎的オブジェクトを含むことができ、1つの基礎的オブジェクトは複数のグループ化オブジェクトに含まれることができます。

リンク 1つのオブジェクトは複数のオブジェクトにリンクすることができます。リンク関係にあるオブジェクトは対等であり、一方のオブジェクトがもう一方の下になることはありません。

たとえば、1つのグループ化オブジェクトは複数の基礎的オブジェクトにリンクすることができます、1つの基礎的オブジェクトは複数のグループ化オブジェクトにリンクすることができます。

接点 1つのオブジェクトには複数のコンタクト・オブジェクトを関連付けることができ、1つのコンタクト・オブジェクトは複数のオブジェクトに関連付けることができます。

たとえば、1つのグループ化オブジェクトは複数のコンタクト・オブジェクトに関連付けることができ、1つのコンタクト・オブジェクトは複数のグループ化および基礎的オブジェクトに関連付けることができます。

付加 1つのオブジェクトには複数の接続オブジェクトを関連付けることができますが、1つの接続オブジェクトは1つのオブジェクトにのみ関連付けることができます。

たとえば、1つのグループ化オブジェクトには複数の接続オブジェクトを関連付けることができますが、1つの接続オブジェクトは1つのグループ化オブジェクトにのみ関連することができます。

プログラム

この関係では、1つのオブジェクト・タイプに複数のプログラム・オブジェクトを関連付けることができます。ただし、1つのプログラム・オブジェクトは1つのオブジェクト・タイプにのみ関連付けることができます。

たとえば、基礎的オブジェクト・タイプには複数のプログラム・オブジェクトを関連付けることができますが、1つのプログラム・オブジェクトは1つのオブジェクト・タイプにのみ関連付けることができます。

SQL によるメタデータへのアクセス

SQL を使用して、情報カタログを構成するデータベース表から直接にメタデータを抽出することができます。この節では、例を示します。

1. 情報カタログに存在するオブジェクト・タイプ定義を判別するには、次の SQL ステートメントを入力します。

```
SELECT OBJTYPID, DPNAME, NAME, CREATOR, PTNAME FROM FLG.OBJTYREG
```

このステートメントでは、以下の情報が戻されます。

OBJTYPID

オブジェクト・タイプの内部 ID

DPNAME

オブジェクト・タイプ名

NAME オブジェクト・タイプの外部名

CREATOR, PTNAME

そのタイプのオブジェクトが保管されている表 (オブジェクト表)

- (158ページの1 のステップから) オブジェクト・タイプを判別した後、特定のオブジェクト・タイプについてのプロパティ名を判別するには、次の SQL ステートメントを入力します。

```
SELECT PHYPRPNM, PROPNAME, DATATYPE, LENGTH, OPTIONS, UUISEQNO,  
       PROPSEQ FROM FLG.PROPERTY WHERE OBJTYPID = 'object_type_ID'  
ORDER BY PROPSEQ
```

このステートメントでは、以下の情報が戻されます (プロパティが作成された順に)。

PHYPRPNM

オブジェクト・タイプのプロパティにマップする、オブジェクト表内の物理列名

PROPNAME

プロパティのビジネス名

DATATYPE

プロパティのデータ・タイプ

LENGTH

プロパティの長さ

OPTIONS

オブジェクト内のこのプロパティについて値が必須であるかどうか

UUISEQNO

UUI 標識 (0 でない場合、シーケンス番号)

PROPSEQ

プロパティがプロパティ表に追加された順序

- (158ページの1 のステップから) オブジェクトが保管されている物理表を判別し、(2 のステップから) 必要なプロパティを判別した後、特定のオブジェクト・タイプのインスタンスを検索するには、次の SQL ステートメントを入力します。

```
SELECT OBJTYPID, INSTIDNT, NAME, phyprpnm1, phyprpnm2...  
       FROM creator.ptname  
       WHERE phyprpnm LIKE '%search_criteria%'
```

このステートメントでは、以下の情報が戻されます。

OBJTYPID

オブジェクト・タイプの内部 ID

INSTIDNT

このオブジェクト・タイプのインスタンスの内部 ID

phyprpnm1

SELECT ステートメントで指定されたプロパティの値

phyprpnm2

SELECT ステートメントで指定されたプロパティの値

さらに、LONG VARCHAR データ・タイプのプロパティ値を検索するには、次の SELECT ステートメントを入力しなければなりません。

```
SELECT PHYPRPNM, ODESC FROM FLG.OVERDESC
WHERE OBJTYPID = object_type_ID
AND INSTIDNT = object_instance_ID
ORDER BY SEQNO
```

ここで、*object_type_ID* および *object_instance_ID* は、159ページの3のステップで SELECT ステートメントを生成した後に入手した値です。このステートメントでは、以下の情報が戻されます。

PHYPRPNM

LONG VARCHAR のプロパティの物理名

ODESC

LONG VARCHAR の値 (それぞれのプロパティ値には、複数の ODESC が存在する可能性があり、順序はシーケンス番号に従います)

4. 情報カタログ内のすべてのオブジェクトのリストを検索するには、次の SQL ステートメントを入力します。

```
SELECT FLGID, INSTNAME, TYPENAME FROM FLG.NAMEINST
```

このステートメントでは、以下の情報が戻されます。

FLGID オブジェクトを表す、連結されたオブジェクト・タイプとインスタンス ID

INSTNAME

オブジェクトの外部名

TYPENAME

オブジェクトのタイプ (オブジェクト・タイプの外部名)

5. オブジェクト間の階層または接点関係を判別するには、次のステートメントを入力します。

```
SELECT SOURCE, TARGET, RELTYPE FROM FLG.RELINST
```

このステートメントでは、以下の情報が戻されます。

SOURCE

関係のソースであるオブジェクトを表す、連結されたオブジェクト・タイプとインスタンス ID

TARGET

関係のターゲットであるオブジェクトを表す、連結されたオブジェクト・タイプとインスタンス ID

RELTYPE

関係タイプ (コンテナーを表す C または接点を表す T)

オブジェクト間のリンクまたは付加関係を判別するには、次のステートメントを入力します。

```
SELECT SOURCE, TARGET, RELTYPE FROM FLG.ATCHREL
```

このステートメントでは、以下の情報が戻されます。

SOURCE

関係のソースであるオブジェクトを表す、連結されたオブジェクト・タイプとインスタンス ID

TARGET

関係のターゲットであるオブジェクトを表す、連結されたオブジェクト・タイプとインスタンス ID

RELTYPE

関係タイプ (付加を表す A またはリンクを表す L)

SOURCE および **TARGET** 値を使用すると、オブジェクト表内のインスタンス情報を検索することができます。また、162ページの4 のステップで示されているように、SQL ステートメントに条件を付けて、特定のオブジェクト値を選択することもできます。

例: あるアプリケーションについて、Employee という名前のリレーショナル表に関するメタデータを表示し、さらに、その表のすべての列を表示したいとします。Employee のオブジェクト・タイプは TABLES であり、列のオブジェクト・タイプは COLUMN です。アプリケーションには、以下の SQL ステートメントが組み込まれます。

1. TABLES オブジェクトが保管されている表の名前を検索します。

```
SELECT OBJTYPID, DPNAME, NAME, CREATOR, PTNAME FROM FLG.OBJTYREG  
WHERE DPNAME = 'TABLES'
```

このステートメントでは、以下の情報が戻されます。

```
'000001', 'TABLES', 'Relational Tables', 'USERXYZ', 'TABLES'
```

2. COLUMN オブジェクトの OBJTYPID を検索します。

```
SELECT OBJTYPID, DPNAME, CREATOR, PTNAME from FLG.OBJTYREG  
WHERE DPNAME = 'COLUMN'
```

このステートメントでは、以下の情報が戻されます。

```
'000007', 'COLUMN', 'Columns or fields', 'USERXYZ', 'COLUMN'
```

3. メタデータを表示したい特定の TABLES オブジェクトに関する情報を検索します。

```
SELECT OBJTYPID, INSTIDNT, NAME, DBNAME, OWNER, TABLES  
FROM USERXYZ.TABLES  
WHERE NAME = 'Employee'
```

このステートメントでは、以下の情報が戻されます。

```
'000001', '0040608795', 'Employee', 'MYDBASE', 'USERABC', 'EMPL_TAB'
```

4. TABLES インスタンス (SOURCE) と COLUMN インスタンス (TARGET) の間の関係を検索します。

```
SELECT TARGET FROM FLG.RELINST  
WHERE SOURCE = '0000010040608795'  
AND TARGET LIKE '000007%'  
AND RELTYPE = 'C'
```

このステートメントでは、次の 2 つのオブジェクトが戻されます。

```
('0000079238400354')  
( '0000079843095410')
```

5. 戻された 2 つの COLUMN オブジェクトに関する情報を検索します。

```
SELECT NAME, SHRTDESC, DATATYPE, LENGTH FROM USERXYZ.COLUMNS  
WHERE INSTIDNT IN ('9238400354', 9843095410')
```

このステートメントでは、以下の情報が戻されます。

```
('Name', 'Employee name information', 'CHAR', '80')  
( 'Address', 'Employee address information', 'CHAR', '220')
```

第8章 情報カタログ・マネージャー・オブジェクト・タイプ

この章では、情報カタログ・マネージャー・オブジェクト・タイプに関する詳しい情報を提供します。

すべてのオブジェクト・タイプのデフォルト・プロパティ

情報カタログ・マネージャーでは、汎用オブジェクト・タイプのデフォルト・プロパティのセットが用意されています。これらのデフォルト・プロパティは、ユーザー定義の表の基礎となります。プロパティには、情報カタログ・マネージャーによって生成されるもの、必須のもの、およびオプションのものがあります。

FLGID インスタンスを固有に識別する、情報カタログ・マネージャーによって生成される ID。

FLGID ID は 16 桁で、最初の 6 桁はオブジェクト・タイプ ID (OBJTYPID) に使用され、次の 10 桁はインスタンス ID (INSTIDNT) に使用されます。FLGID の形式は、次のとおりです。

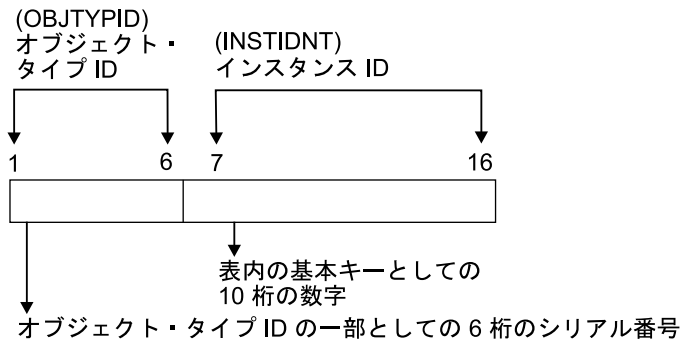


図 26. FLGID の形式

Name ステップの名前。名前は、用語集、ニュース照会、およびその他のオブジェクトで使用することができます。これは必須プロパティであり、ヌル不可です。この名前は、情報カタログ・マネージャー・ウィンドウで表示されます。

すべてのオブジェクト・タイプのデフォルト・プロパティ

UPDATIME

インスタンスの作成または最終更新の日時を示すシステム・タイム・スタンプ。

UPDATEBY

インスタンスを最後に更新した情報カタログ管理者またはユーザー (特殊な特権を持つ)。 Attachment (付加) オブジェクトの場合、このフィールドは情報カタログ・ユーザーのユーザー ID になる可能性があります。

デフォルト・プロパティの要約

情報カタログ管理者は、事前定義テンプレートを使用して、オブジェクト・タイプを作成することができます。また、テンプレートを組織に合わせてカスタマイズするために、テンプレートに属性を追加することができます。事前定義テンプレートには、いくつかのオプション・フィールドがあります。表76では、デフォルト・プロパティを示します。

表 76. 事前定義テンプレートのデフォルト・プロパティ

列名	データ・タイプ	説明	ヌル可能	NLS
OBJTYPID	CHAR(6)	情報カタログ・マネージャーによって生成された 6 桁のオブジェクト・タイプ ID。この ID は、特定のオブジェクト・タイプを表します。	いいえ	SBCS
INSTIDNT	CHAR(10)	情報カタログ・マネージャーによって生成された固有なインスタンス ID。この ID は、FLGID の 2 番目の部分であり、オブジェクト・タイプの中でこのインスタンスを固有に識別する 10 桁のシリアル番号です。	いいえ	SBCS
NAME	VARCHAR(80)	この名前は、製品の中でそれぞれのユーザー定義オブジェクトを識別するために、情報カタログ・ユーザーによって入力されます。	いいえ	SBCS と DBCS の両方
UPDATIME	CHAR(26)	メタデータの作成または最終更新の日時。この値は情報カタログ・マネージャーによって生成されます。	いいえ	SBCS
UPDATEBY	CHAR(8)	インスタンスを最後に更新した情報カタログ管理者またはユーザー (特殊な更新特権を持つ)。 Attachment (付加) オブジェクトの場合、このフィールドは情報カタログ・ユーザーのユーザー ID になる可能性があります。この値は情報カタログ・マネージャーによって生成されます。	いいえ	SBCS と DBCS の両方

表 76. 事前定義テンプレートのデフォルト・プロパティ (続き)

列名	データ・ タイプ	説明	ヌル可能	NLS
注:				
NLS: 各国語サポート				
SBCS: 1 バイト文字セット				
DBCS: 2 バイト文字セット				

拡張可能なオブジェクト・タイプに関するガイドライン

1. オブジェクト・タイプが拡張可能なのは、それを変更することができる場合です。オブジェクト・タイプ区分が変更可能なのは、それに他のオブジェクトを追加することができる場合です。PROGRAMS、QUERY、IMAGE、REPORT、ビジネス・グループ (BUSNSGP)、TABLES、COLUMNS、GLOSSARY、CONTACTS、および NEWS を含め、大部分の情報カタログ・マネージャー・オブジェクトは拡張可能です。COMMENTS オブジェクト・タイプは拡張不能です。Programs (プログラム) および Attachment (付加) 区分は拡張不能です。
2. すべての情報カタログ・マネージャー・オブジェクトは、以下の区分に編成されます。

Elemental (基本) - E

他のオブジェクトを含むことができないオブジェクト・タイプ (たとえば、REPORT、QUERY、および IMAGE オブジェクト)。

Grouping (グループ化) - G

他の Grouping (グループ化) または Elemental (基本) オブジェクトを含むことができるオブジェクト・タイプ (たとえば、INFOGRPS および TABLES オブジェクト・タイプ)。

Program (プログラム) - P

実行可能なオブジェクト・タイプ (たとえば、PROGRAMS オブジェクト・タイプ)。

Contact (接点) - C

他のオブジェクトに関する質問が生じた場合の連絡先となる人や組織を識別するために使用される特殊なオブジェクト・タイプ (たとえば、CONTACTS オブジェクト・タイプ)。

すべてのオブジェクト・タイプのデフォルト・プロパティー

Dictionary (辞書) - D

ユーザーがビジネス環境で使用されている用語の定義または同義語を見付けるのに役立つオブジェクト・タイプ (たとえば、GLOSSARY オブジェクト・タイプ)。

Support (サポート) - S

情報カタログまたはビジネス環境に関する追加情報を提供するオブジェクト・タイプ (たとえば、NEWS オブジェクト・タイプ)。

Attachment (付加) - A

他のオブジェクトに追加情報を付加するために使用されるオブジェクト・タイプ (たとえば、COMMENTS オブジェクト・タイプ)。

オブジェクト・タイプの作成、削除、および更新に使用されるプロセスは、PROGRAMS および COMMENTS オブジェクト・タイプを除き、すべてのオブジェクト・タイプで同じです。

PROGRAMS オブジェクト・タイプは、情報カタログ・マネージャーで事前定義されており、Program (プログラム) 区分で使用される唯一のオブジェクト・タイプです。Program (プログラム) 区分のもとで別のオブジェクト・タイプを作成することはできず、また、PROGRAMS オブジェクト・タイプを削除することはできません。

COMMENTS オブジェクト・タイプは、情報カタログ・マネージャーで事前定義されており、Attachment (付加) 区分で使用される唯一のオブジェクト・タイプです。Attachment (付加) 区分のもとで別のオブジェクト・タイプを作成することはできず、また、COMMENTS オブジェクト・タイプを削除することはできません。

3. VIDEO や AUDIO などの新しいオブジェクト・タイプを使用して、独自のオブジェクト・タイプを作成することができます (そのオブジェクト・タイプの DPname が情報カタログ・マネージャーの中で固有であれば)。
4. すべてのオブジェクトは、オブジェクト・タイプ定義の一部として汎用固有 ID (UUI) を含まなければなりません。UUI は、インポート・プロセス中に、ターゲット情報カタログ内の類似した ID との比較に使用されます。
5. プロパティーのデータ・タイプが LONG VARCHAR などの場合、情報カタログ・マネージャーは、自動的にプロパティーとそのメタデータを独立したオーバーフロー表に入れ、ユーザーがプロパティーを検索できるように、それをより小さいセグメントに分割します。プロパティーのサイズのために、検索の処理速度は遅くなります。
6. 情報カタログ・マネージャーでは、5 つのデータ・タイプがサポートされません。

すべてのオブジェクト・タイプのデフォルト・プロパティ

CHAR 固定長文字ストリング (最大 254 文字)。

VARCHAR

可変長文字ストリング (最大 4000 文字)。表の行の最大長も 4000 です。

LONG VARCHAR

可変長文字ストリング (最大 32700 文字)。

情報カタログ・マネージャーは、このタイプのメタデータを独立した表に保管し、ユーザーがストリングを検索できるように、メタデータをより小さいセグメントに分割します。メタデータが取り出されるときには、情報カタログ・マネージャーはセグメントを結合します。

TIMESTAMP

年、月、日、時、分、秒、およびマイクロ秒から構成される 7 部分値 (26 バイトの文字ストリング)。形式は yyyy-mm-dd-hh.mm.ss.nnnnnn です。

LONG VARCHAR FOR BIT DATA

ビットマップなどのバイナリー・データ。

関係タイプ

1. 情報カタログ・マネージャーでは、同じ **FLGRelation** API によって作成および削除される以下のタイプの関係がサポートされます。それぞれのタイプの関係にアクセスするためには、**FLGNavigate**、**FLGWhereUsed**、および **FLGListContacts** など、異なる API が使用されます。これらの API は、ユーザーの要求を完了させるために、対応する **IPI** を呼び出します。
 - a. 包含 (C)

たとえば、階層的なビジネス構造や、リレーショナル列に対するリレーショナル表。

この関係は、**FLGNavigate** および **FLGWhereUsed** などの API によって取り出されます。
 - b. 接点 (T)

たとえば、特定のオブジェクトに関してサービスを提供する人の名前。

この関係にアクセスするには、**FLGListContacts** API が使用されます。
 - c. 付加関係 (A)

たとえば、特定のオブジェクトに関するコメント。

この関係を取り出すには、**FLGListAssociates** および **FLGFoundIn** API が使用されます。

すべてのオブジェクト・タイプのデフォルト・プロパティ

d. リンク関係 (L)

Grouping (グループ化) または Elemental (基本) 区分のオブジェクト・タイプ・インスタンスは、他の Grouping (グループ化) または Elemental (基本) 区分のオブジェクト・タイプ・インスタンスにリンクすることができます。

この関係を取り出すには、FLGListAssociates API が使用されます。

2. 情報カタログ・マネージャー定義の区分に基づく関係の規則は、155ページの『論理メタデータ・モデル』で説明されています。

オブジェクトは、必ずしも関係を持つ必要はありません。情報カタログ・マネージャー・ウィンドウ (情報カタログ・マネージャー 管理の手引き を参照) または FLGSearch API を使用するか、あるいは FLG.NAMEINST 表を表示すると、すべてのオブジェクトを検出することができます。情報カタログ・マネージャー API の詳細については、情報カタログ・マネージャー プログラミングの手引きおよび解説書 を参照してください。FLG.NAMEINST 表については、139ページの『FLG.NAMEINST 表』を参照してください。

関係インスタンス

2 つのオブジェクトの間に関係がある場合、このオブジェクト間の関係が関係インスタンス表に追加されます。

表の形式は、次のとおりです。

ソースの FLGID (16 桁)	ターゲットの FLGID (16 桁)	RELTYPE C/T/L/A
-------------------------	---------------------------	--------------------

表のプロパティの詳細については、147ページの『FLG.RELINST 表』を参照してください。

事前定義の情報カタログ・マネージャー・オブジェクト・タイプ

情報カタログ・マネージャーには、他のデータウェアハウスセンター構成要素および他の MDIS 準拠製品 (IBM および他社の) からのメタデータと交換することができる事前定義オブジェクト・タイプが組み込まれています。この節では、オブジェクト・タイプのプロパティの MDIS オブジェクト・タイプへのマップを含め、事前定義のすべての情報カタログ・マネージャー・オブジェクト・タイプについて説明します。完全な MDIS オブジェクト・タイプ定義を含め、メタデータ交換仕様については、Meta Data Coalition の Web サイト (<http://www.MDCinfo.com>) を参照してください。

事前定義の情報カタログ・マネージャー・オブジェクト・タイプ

情報カタログ・マネージャーでは、サンプル情報カタログで、事前定義オブジェクト・タイプと、各タイプのサンプル・オブジェクトの両方が提供されます。サンプル情報カタログには、7つの情報カタログ・マネージャー区分ごとに、少なくとも1つのオブジェクト・タイプが組み込まれています。この節では、サンプル情報カタログを作成する方法を説明します。情報カタログ・マネージャー・オブジェクト・タイプの能力の詳細については、[情報カタログ・マネージャー 管理の手引き](#)を参照してください。

表77では、サンプル情報カタログ内のすべてのオブジェクト・タイプがリストされています。オブジェクト・タイプは、データを表現するか、または2つのオブジェクト・タイプの間を表現します。

データを表現するオブジェクト・タイプ

事前定義オブジェクト・タイプの大部分は、図表または文書オブジェクト・タイプなど、データのタイプを表現します。

関係を表現するオブジェクト・タイプ

トランスフォーメーション・オブジェクト・タイプは、他の2つのオブジェクト・タイプの間を表現する特殊なオブジェクト・タイプです。具体的には、データのソース・フォーマットからターゲット・フォーマットへのトランスフォーメーションを表現します。トランスフォーメーション・オブジェクト・タイプを使用すると、ターゲット・リレーショナル・データベース内でデータの系統に関する情報を提供することができます。

表 77. 事前定義データ・オブジェクト・タイプの要約

オブジェクト・タイプ名	説明	プロパティが定義されているページ
アプリケーション・データ	内部使用のみ	178
オーディオ・クリップ	音声情報を含むファイルを表現する	215
ビジネス・サブジェクト・エリア	オブジェクトの論理的なグループ分けを表現する	180
図表	印刷された、または電子的な図表を表現する	216
列またはフィールド	リレーショナル表内の列、ファイル内のフィールド、またはIMSセグメント内のフィールドを表現する	181
コメント	情報カタログ内の他のオブジェクトに関するコメントを含む	234

事前定義の情報カタログ・マネージャー・オブジェクト・タイプ

表 77. 事前定義データ・オブジェクト・タイプの要約 (続き)

オブジェクト・ タイプ名	説明	プロパティーが定義 されているページ
データベース	リレーショナル・データベースを表現する	184
情報カタログ・マネ ージャー・ニュース	情報カタログへの変更に関する情報を伝達 する	229
多次元データベース 内の次元	多次元データベース内の次元を表現する	186
文書	ブック、資料、および技術文書を表現する	217
エレメント	『列またはフィールド』 オブジェクト・ タイプに直接にマップしない MDIS Element (エレメント) オブジェクトを表現 する	188
ファイル	ファイル・システム内のファイルを表現す る	191
用語集エントリー	情報カタログ内で使用される用語の定義を 表現する	227
イメージまたはグラ フィックス	グラフィック・イメージ (ビットマップな ど) を表現する	218
IMS データベース定 義 (DBD)	IMS データベース定義を表現する	193
IMS プログラム制御 ブロック (PCB)	IMS プログラム制御ブロックを表現する	195
IMS プログラム仕様 ブロック (PSB)	IMS プログラム仕様ブロックを表現する	197
IMS セグメント	IMS セグメントを表現する	198
インターネット文書	Web サイトや、インターネット上のその 他の関連文書を表現する	219
Lotus® Approach® 照 会	組織のデータに対して使用可能な Lotus Approach 照会を表現する	220
多次元データベース 内のメンバー	多次元データベース内のメンバーを表現す る	201
多次元データベース	多次元データベースを表現する	203
オンライン・ニュー ス・サービス	オンラインでアクセスできるニュースおよ び情報サービスを表現する	230
オンライン資料	オンライン・サービスによってアクセスで きる資料およびその他の文書を表現する	231

事前定義の情報カタログ・マネージャー・オブジェクト・タイプ

表 77. 事前定義データ・オブジェクト・タイプの要約 (続き)

オブジェクト・タイプ名	説明	プロパティが定義されているページ
連絡先	情報カタログ内の 1 つまたは複数のオブジェクトについて責任がある人またはグループを識別する	227
プレゼンテーション	印刷された、または電子的なプレゼンテーションを表現する	222
情報カタログ・マネージャー・オブジェクトから呼び出すことができるプログラム	特定のオブジェクト・タイプを処理することができるアプリケーションを定義する	233
レコード	『ファイル』 または 『リレーショナル表および視点』 オブジェクト・タイプに直接にマップしない MDIS Record (レコード) オブジェクトを表現する	205
リレーショナル表および視点	リレーショナル・データベースの表または視点を表現する	207
サブスキーマ	データベース内のレコードの論理的なグループ分けを表現する	211
トランスフォーマーション	ターゲット・リレーショナル・データベース内の列にデータを取り込むために使用される式または論理を表現する	212
スプレッドシート	デスクトップのスプレッドシート (たとえば、Lotus 1-2-3 [®] または Microsoft Excel スプレッドシート) を表現する	223
テキスト・ベース・レポート	印刷された、または電子的なレポートを表現する	224
ビデオ・クリップ	ビデオ情報を含むファイルを表現する	225

事前定義オブジェクト・タイプのモデル

事前定義の情報カタログ・マネージャー・オブジェクト・タイプは、図27～図32で示されている6つのデータ・モデルに参加します。

図27では、リレーショナル・モデルに参加するオブジェクト・タイプを示します。

事前定義オブジェクト・タイプのモデル

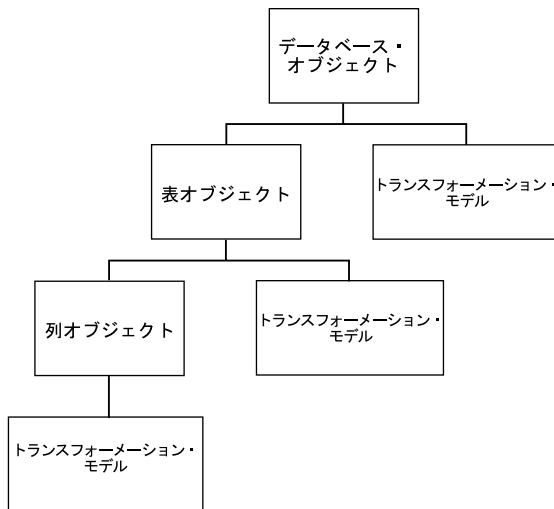


図27. リレーショナル・モデルと事前定義オブジェクト・タイプ

図28 では、階層モデルに参加するオブジェクト・タイプを示します。

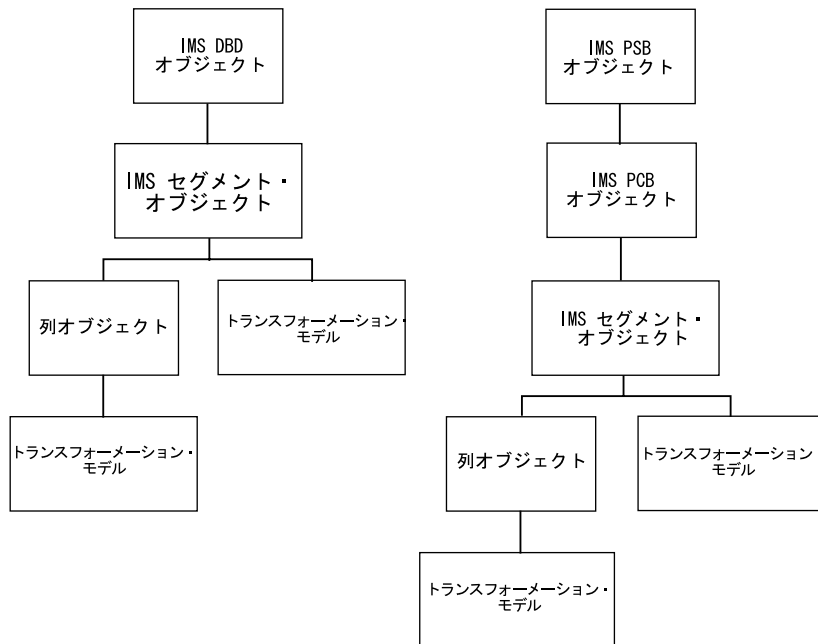


図28. 階層モデルと事前定義オブジェクト・タイプ

図29 では、ファイル・モデルに参加するオブジェクト・タイプを示します。

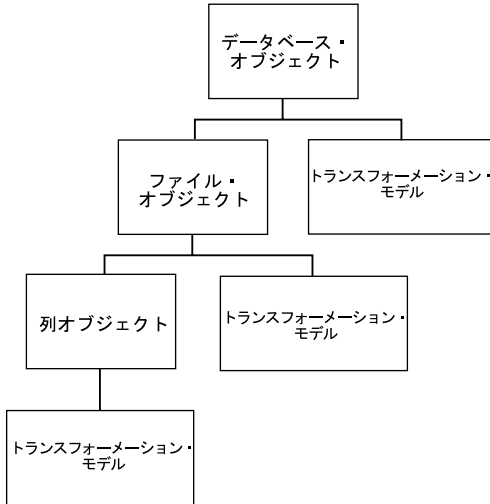


図 29. ファイル・モデルと事前定義オブジェクト・タイプ

図30 では、多次元 (OLAP) モデルに参加するオブジェクト・タイプを示します。

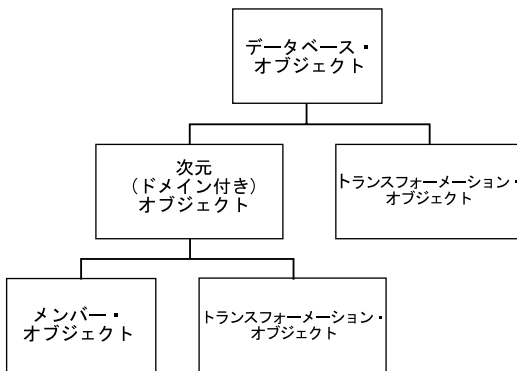


図 30. 多次元モデルと事前定義オブジェクト・タイプ

図31 では、トランスフォーメーション・モデルに参加するオブジェクト・タイプを示します。

事前定義オブジェクト・タイプのモデル

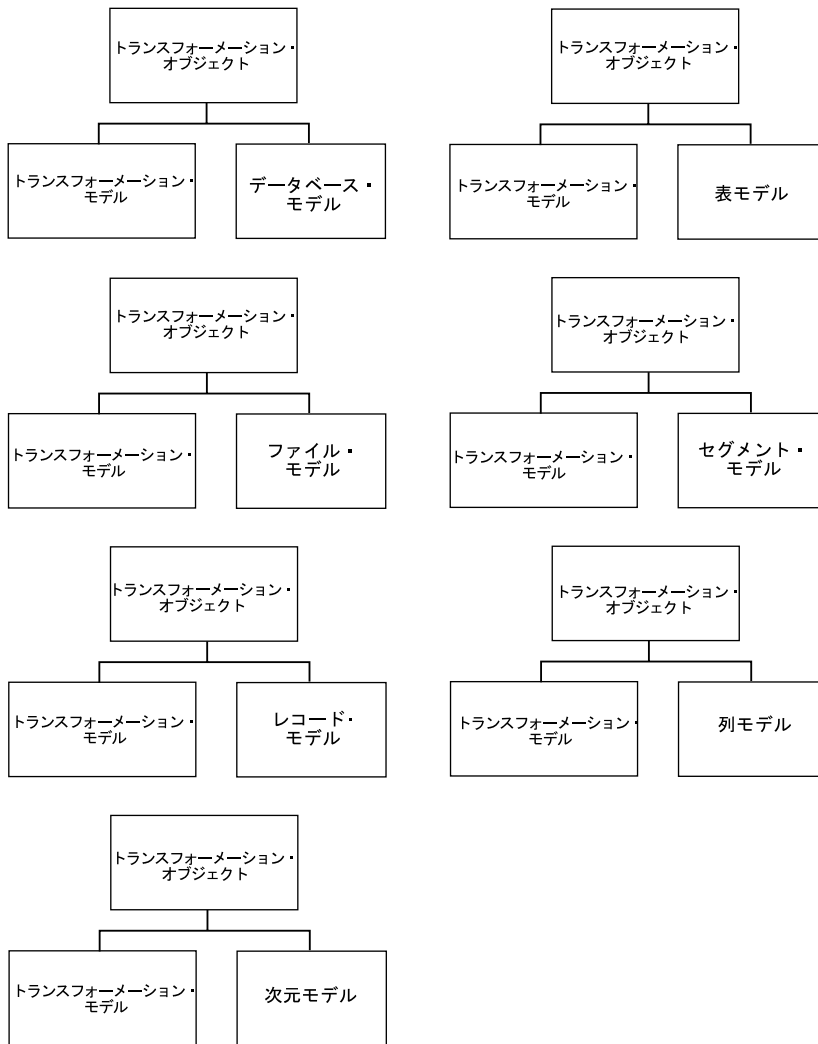


図31. トランスフォーメーション・モデルと事前定義オブジェクト・タイプ

図32 では、サブジェクト・エリア・モデルに参加するオブジェクト・タイプを示します。

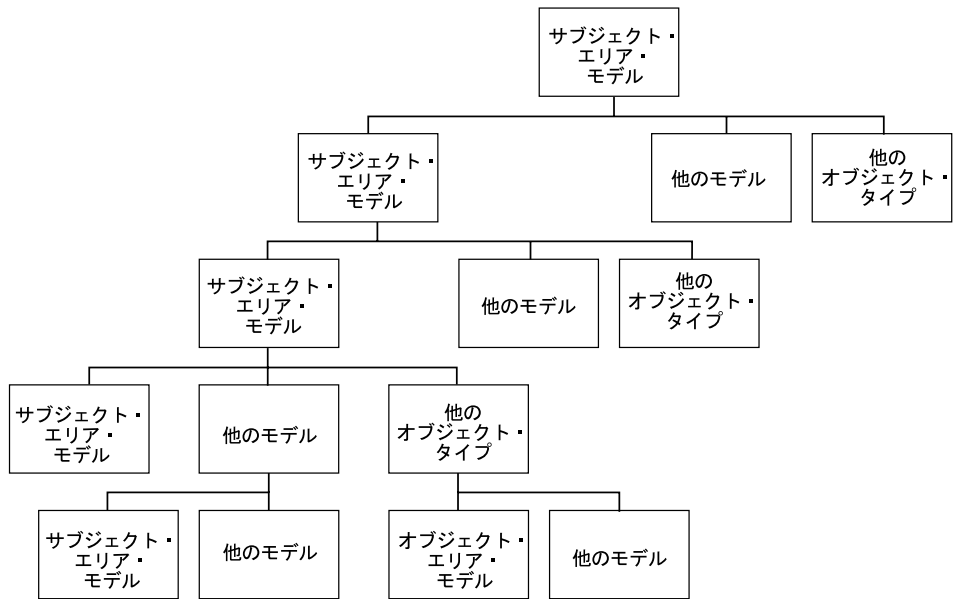


図 32. サブジェクト・エリア・モデルと事前定義オブジェクト・タイプ

事前定義オブジェクト・タイプの説明

サンプル情報カタログ・マネージャーのオブジェクト・タイプは、区分別に編成されており、177ページから始まる表で定義されています。

それぞれの表では、そのオブジェクト・タイプのプロパティがリストされています。それぞれのプロパティは、その仕様の面から説明されています。プロパティの仕様により、そのオブジェクト・タイプのオブジェクトを作成または更新する際にそのプロパティに与えることができる値が左右されます。

プロパティの仕様は、以下のとおりです。

EXTNAME	プロパティの名前 (たとえば、ビジネス名)。
DT	プロパティの値のデータ・タイプ (たとえば、CHAR または VARCHAR)。
DL	プロパティの値の長さ (VARCHAR または LONG VARCHAR データ・タイプの場合、最大長)。
SHRTNAME	情報カタログ・マネージャー・データ・ストア内でプロパティを識別するために使用される名前。

サンプル情報カタログ内の事前定義オブジェクト・タイプ

NULLS	R プロパティの値は必須 (タグ言語の NULLS の値は N)。 O プロパティの値はオプション (タグ言語の NULLS の値は Y)。 S 情報カタログ・マネージャーによって生成される値。これは、オブジェクトの作成時に情報カタログ・マネージャーによって値が提供されることを意味します。この値を指定することはできません。
UISEQ	プロパティが UII の一部である場合、この番号は、UII の中でのプロパティの位置を示します。

MDIS マッピング

情報カタログ・マネージャー・オブジェクト・タイプのプロパティを説明する表は、177ページから始まります。メタデータ交換仕様 (MDIS) に準拠しているそれぞれのオブジェクト・タイプについては、**MDIS 名へのマップ**という欄で、プロパティごとに同等の MDIS 名が示されています。

1. エクスポートしようとしているオブジェクト・タイプの表を見付けます。
2. **MDIS 名へのマップ**欄で MDIS 名を見付けます。
3. **プロパティ名**および**プロパティ短縮名**欄で、同等の情報カタログ・マネージャー名を見付けます。

以下のオブジェクト・タイプのプロパティ表で説明されているそれぞれのプロパティは、情報カタログ・マネージャー DB2 ストレージ表 `XXX.object_type_name` の中で同じプロパティ短縮名を持つ列に対応します。ここで、`object_type_name` は、表で説明されているオブジェクト・タイプの名前です。プロパティのデータ・タイプが `LONG VARCHAR` である場合、プロパティは、情報カタログ・マネージャー DB2 ストレージ表 `FLG.OVERDESC` の行に対応します。

グループ化区分

グループ化区分には、次のオブジェクト・タイプが含まれます。

- 177ページの『アプリケーション・データ (Application data)』
- 179ページの『ビジネス・サブジェクト・エリア (Business subject areas)』
- 180ページの『列またはフィールド (Columns or fields)』
- 183ページの『データベース (Databases)』

サンプル情報カタログ内の事前定義オブジェクト・タイプ

- 185ページの『多次元データベース内の次元 (Dimensions within a multidimensional database)』
- 187ページの『DWC プロセス (DWC Process)』
- 188ページの『エレメント (Elements)』
- 190ページの『ファイル (Files)』
- 193ページの『IMS データベース定義 (DBD) (IMS database definitions)』
- 195ページの『IMS プログラム制御ブロック (PCB) (IMS program control blocks)』
- 196ページの『IMS プログラム仕様ブロック (PSB) (IMS program specification blocks)』
- 198ページの『IMS セグメント (IMS segments)』
- 200ページの『多次元データベース内のメンバー (Members within a multidimensional database)』
- 202ページの『多次元データベース (Multidimensional databases)』
- 204ページの『レコード (Records)』
- 206ページの『リレーショナル表および視点 (Relational tables and views)』
- 209ページの『スタースキーマ (Star Schemas)』
- 210ページの『サブスキーマ (Subschemas)』
- 212ページの『トランスフォーメーション (Transformations)』

アプリケーション・データ (Application data): 一部の MDIS メタデータの交換のために情報カタログ・マネージャーによって使用されます。このオブジェクト・タイプのオブジェクトが情報カタログ内で見られることはありますが、このオブジェクト・タイプはオブジェクトの作成には使用されません。

このオブジェクト・タイプを定義するためのタグ言語は、
¥VWSWIN¥DGWIN¥TYPES ディレクトリーのファイル FLGNYAPL.TYP に入っています。

このオブジェクト・タイプ用の情報カタログ・マネージャー DB2 ストレージ表の名前は XXX.APPLDATA です。表名の詳細については、176ページの『MDIS マッピング』を参照してください。

178ページの表78 では、アプリケーション・データ・オブジェクト・タイプのプロパティに関する情報を提供します。

アプリケーション・データ

表 78. アプリケーション・データ・オブジェクト・タイプのプロパティ

プロパティ名	データ・タイプ	サイズ	プロパティ短縮名	値フラグ	UUI 順序
オブジェクト・タイプ ID	CHAR	6	OBJTYPID	S	
インスタンス ID	CHAR	10	INSTIDNT	S	
名前	VARCHAR	80	NAME	O	
最終変更日時	TIMESTAMP	26	UPDATIME	S	
最終変更者	CHAR	8	UPDATEBY	S	
短い記述	VARCHAR	250	SHRTDESC	O	
長い記述	LONG VARCHAR	32700	LONGDESC	O	
ソース・オブジェクト ID	CHAR	16	FLGID	R	1
アプリケーション・データ・フィールド 0	LONG VARCHAR	32700	APPLDAT0	O	
アプリケーション・データ・フィールド 1	LONG VARCHAR	32700	APPLDAT1	O	
アプリケーション・データ・フィールド 2	LONG VARCHAR	32700	APPLDAT2	O	
アプリケーション・データ・フィールド 3	LONG VARCHAR	32700	APPLDAT3	O	
アプリケーション・データ・フィールド 4	LONG VARCHAR	32700	APPLDAT4	O	

表 78. アプリケーション・データ・オブジェクト・タイプのプロパティ (続き)

プロパティ名	データ・タイプ	サイズ	プロパティ短縮名	値フラグ	UUI 順序
アプリケーション・データ・フィールド 5	LONG VARCHAR	32700	APPLDAT5	O	
アプリケーション・データ・フィールド 6	LONG VARCHAR	32700	APPLDAT6	O	
アプリケーション・データ・フィールド 7	LONG VARCHAR	32700	APPLDAT7	O	
アプリケーション・データ・フィールド 8	LONG VARCHAR	32700	APPLDAT8	O	
アプリケーション・データ・フィールド 9	LONG VARCHAR	32700	APPLDAT9	O	
ソース定義の作成のタイム・スタンプ	CHAR	26	CRTTIME	O	
ソース定義の最終変更のタイム・スタンプ	CHAR	26	SRCDATCF	O	

注: S = 情報カタログ・マネージャー生成、R = 必須、O = オプション

ビジネス・サブジェクト・エリア (Business subject areas): オブジェクトの論理的なグループ分けを表現します。

このオブジェクト・タイプを定義するためのタグ言語は、
 ¥VWSWIN¥DGWIN¥TYPES ディレクトリーのファイル FLGNYINF.TYP に入
 っています。

ビジネス・サブジェクト・エリア

このオブジェクト・タイプ用の情報カタログ・マネージャー DB2 ストレージ表の名前は XXX.INFOGRPS です。表名の詳細については、176ページの『MDIS マッピング』を参照してください。

表79 では、ビジネス・サブジェクト・エリア・オブジェクト・タイプのプロパティーに関する情報を提供します。

表79. ビジネス・サブジェクト・エリア・オブジェクト・タイプのプロパティー

プロパティー名	データ・タイプ	サイズ	プロパティー短縮名	値フラグ	UII 順序
オブジェクト・タイプ ID	CHAR	6	OBJTYPID	S	
インスタンス ID	CHAR	10	INSTIDNT	S	
名前	VARCHAR	80	NAME	R	1
最終変更日時	TIMESTAMP	26	UPDATIME	S	
最終変更者	CHAR	8	UPDATEBY	S	
短い記述	VARCHAR	250	SHRTDESC	O	
長い記述	LONG VARCHAR	32700	LONGDESC	O	
処置	VARCHAR	254	ACTIONS	O	
ファイル名	VARCHAR	254	FILENAME	O	
データにアクセスするための URL	VARCHAR	254	URL	O	
詳細の問い合わせ先	VARCHAR	80	CONTACT	O	

注: S = 情報カタログ・マネージャー生成、R = 必須、O = オプション

列またはフィールド (Columns or fields): リレーショナル表内の列、ファイル内のフィールド、または IMS セグメント内のフィールドを表現します。

このオブジェクト・タイプを定義するためのタグ言語は、`¥VWSWIN¥DGWIN¥TYPES` ディレクトリーのファイル `FLGNYCOL.TYP` に入っています。

このオブジェクト・タイプ用の情報カタログ・マネージャー DB2 ストレージ表の名前は XXX.COLUMNS です。表名の詳細については、176ページの『MDIS マッピング』を参照してください。

表80 では、列またはフィールド・オブジェクト・タイプのプロパティに関する情報を提供します。

表 80. 列またはフィールド・オブジェクト・タイプのプロパティ. このオブジェクト・タイプの MDIS 名は Element (エレメント) です。

プロパティ名	データ・タイプ	サイズ	プロパティ短縮名	値フラグ	MDIS 名へのマップ
オブジェクト・タイプ ID	CHAR	6	OBJTYPID	S	
インスタンス ID	CHAR	10	INSTIDNT	S	
名前	VARCHAR	80	NAME	R	ElementLongName
最終変更日時	TIMESTAMP	26	UPDATIME	S	
最終変更者	CHAR	8	UPDATEBY	S	
短い記述	VARCHAR	250	SHRTDESC	O	BriefDescription
長い記述	LONG VARCHAR	32700	LONGDESC	O	LongDescription
処置	VARCHAR	254	ACTIONS	O	ApplicationData
カタログ注釈	VARCHAR	254	REMARKS	O	ApplicationData
詳細の問い合わせ先	VARCHAR	80	RESPNSBL	O	ContactName
列またはフィールドの最後のリフレッシュ	CHAR	26	FRESHDAT	O	ElementLastRefreshDate
列またはフィールドのデータ・タイプ	CHAR	30	DATATYPE	O	ElementDataType
列またはフィールドの長さ	CHAR	20	LENGTH	O	ElementLength
列またはフィールドの位取り	CHAR	5	SCALE	O	ApplicationData
列またはフィールドの精度	CHAR	5	PRECDIG	O	ElementPrecision

列またはフィールド

表 80. 列またはフィールド・オブジェクト・タイプのプロパティ (続き). このオブジェクト・タイプの MDIS 名は Element (エレメント) です。

プロパティ名	データ・タイプ	サイズ	プロパティ短縮名	値フラグ	UUI 順序	MDIS 名へのマップ
列またはフィールドがヌル可能	CHAR	1	NULLS	O		ElementNulls
列またはフィールドの序数	CHAR	5	ORDINAL	O		ElementOrdinality
列またはフィールドの位置	CHAR	5	POSNO	O		ElementPosition
列またはフィールドの始まりからのバイト・オフセット	CHAR	10	STARTPOS	O		ApplicationData
列またはフィールドがキーの一部	CHAR	1	ISKEY	O		ApplicationData
列またはフィールドが固有キー	CHAR	1	UNIQKEY	O		ApplicationData
キーの中での列またはフィールドの位置	CHAR	5	KEYPOSNO	O		ElementKeyPosition
データベースのホスト・サーバー名	VARCHAR	80	SERVER	O		ServerName
データベースまたはサブシステム名	VARCHAR	80	DBNAME	R	1	DatabaseName
表の所有者	VARCHAR	80	OWNER	R	2	OwnerName
表名	VARCHAR	80	TABLES	R	3	RecordName
列またはフィールド名	VARCHAR	254	COLUMNS	R	4	ElementName
ファイル名	VARCHAR	254	FILENAME	R	5	ApplicationData
データにアクセスするための URL	VARCHAR	254	URL	O		ApplicationData

表 80. 列またはフィールド・オブジェクト・タイプのプロパティ (続き). このオブジェクト・タイプの MDIS 名は Element (エレメント) です。

プロパティ名	データ・タイプ	サイズ	プロパティ短縮名	値フラグ	UI 順序	MDIS 名へのマップ
含む次元	VARCHAR	80	DIMENSION	O		DimensionName
データは変更前イメージ、変更後イメージ、または計算	CHAR	50	COLIMAGE	O		ApplicationData
列にデータを取り込むために使用されるソース列、フィールド名、または式	VARCHAR	254	COLEXPR	O		ApplicationData
ヌル値を表現するために使用されるストリング	VARCHAR	30	IDSNREP	O		ApplicationData
日付のレゾリューション	CHAR	1	IDSRES	O		ApplicationData
データがテキスト	CHAR	1	ISTEXT	O		ApplicationData
ソース定義の作成のタイム・スタンプ	CHAR	26	CRTTIME	O		DateCreated、TimeCreated
ソース定義の最終変更のタイム・スタンプ	CHAR	26	SRCDATCF	O		DateUpdated、TimeUpdated

注: S = 情報カタログ・マネージャー生成、R = 必須、O = オプション

データベース (Databases): リレーショナル・データベースを表現します。

このオブジェクト・タイプを定義するためのタグ言語は、
 ¥VWSWIN¥DGWIN¥TYPES ディレクトリーのファイル FLGNYDAT.TYP に入
 っています。

データベース

このオブジェクト・タイプ用の情報カタログ・マネージャー DB2 ストレージ表の名前は XXX.DATABAS です。表名の詳細については、176ページの『MDIS マッピング』を参照してください。

表81 では、データベース・オブジェクト・タイプのプロパティに関する情報を提供します。

表 81. データベース・オブジェクト・タイプのプロパティ. このオブジェクト・タイプの MDIS 名は Database (データベース) です。

プロパティ名	データ・タイプ	サイズ	プロパティ短縮名	値フラグ	UI 順序	MDIS 名へのマップ
オブジェクト・タイプ ID	CHAR	6	OBJTYPID	S		
インスタンス ID	CHAR	10	INSTIDNT	S		
名前	VARCHAR	80	NAME	R		DatabaseLongName
最終変更日時	TIMESTAMP	26	UPDATIME	S		
最終変更者	CHAR	8	UPDATEBY	S		
短い記述	VARCHAR	250	SHRTDESC	O		BriefDescription
長い記述	LONG VARCHAR	32700	LONGDESC	O		LongDescription
処置	VARCHAR	254	ACTIONS	O		ApplicationData
詳細の問い合わせ先	VARCHAR	80	RESPNSBL	O		ContactName
データベースの所有者	VARCHAR	80	OWNER	O		OwnerName
データベースのホスト・サーバー名	VARCHAR	80	SERVER	R	1	ServerName
データベースのサーバー・タイプ	VARCHAR	80	SRVRTYPE	O		ServerType
データベースまたはサブシステム名	VARCHAR	80	DBNAME	R	2	DatabaseName
データベースのタイプ	VARCHAR	80	DBTYPE	R	3	DatabaseType
データベースの拡張タイプ	VARCHAR	40	DBETYPE	O		DatabaseExtendedType

表 8I. データベース・オブジェクト・タイプのプロパティ (続き). このオブジェクト・タイプの MDIS 名は Database (データベース) です。

プロパティ名	データ・タイプ	サイズ	プロパティ短縮名	値フラグ	UUI 順序	MDIS 名へのマップ
データベースの状況	VARCHAR	80	DBSTAT	O		DatabaseStatus
データベースの位置	VARCHAR	80	LOCATION	O		ApplicationData
データにアクセスするための URL	VARCHAR	254	URL	O		ApplicationData
システム・コード・ページ	VARCHAR	10	CODEPAGE	O		ApplicationData
エージェント・タイプ	VARCHAR	80	AGENTTYPE	O		ApplicationData
ソース定義の作成のタイム・スタンプ	CHAR	26	CRTTIME	O		DateCreated、TimeCreated
ソース定義の最終変更のタイム・スタンプ	CHAR	26	SRCDATCF	O		DateUpdated、TimeUpdated

注: S = 情報カタログ・マネージャー生成、R = 必須、O = オプション

多次元データベース内の次元 (Dimensions within a multidimensional database): 多次元データベース内の次元を表現します。次元はメンバーから構成されます。

このオブジェクト・タイプを定義するためのタグ言語は、
 ¥VWSWIN¥DGWIN¥TYPES ディレクトリーのファイル FLGNYDIM.TYP に
 入っています。

このオブジェクト・タイプ用の情報カタログ・マネージャー DB2 ストレージ表の名前は XXX.DIMENSION です。表名の詳細については、176ページの『MDIS マッピング』を参照してください。

186ページの表82 では、多次元データベース内の次元オブジェクト・タイプのプロパティに関する情報を提供します。

多次元データベース内の次元

表 82. 多次元データベース内の次元オブジェクト・タイプのプロパティ。このオブジェクト・タイプの MDIS 名は Dimension (次元) です。

プロパティ 名	データ・ タイプ	サイズ	プロパティ 短縮名	値フラグ	UUI 順序	MDIS 名へのマップ
オブジェク ト・タイプ ID	CHAR	6	OBJTYPID	S		
インスタンス ID	CHAR	10	INSTIDNT	S		
名前	VARCHAR	80	NAME	R		DimensionLongName
最終変更日時	TIMESTAMP	26	UPDATIME	S		
最終変更者	CHAR	8	UPDATEBY	S		
短い記述	VARCHAR	250	SHRTDESC	O		BriefDescription
長い記述	LONG VARCHAR	32700	LONGDESC	O		LongDescription
処置	VARCHAR	254	ACTIONS	O		ApplicationData
詳細の問い合 わせ先	VARCHAR	80	RESPNSBL	O		ContactName
データベース の最後のリフ レッシュ	CHAR	26	FRESHDAT	O		ApplicationData
データベー スのホスト・サ ーバー名	VARCHAR	80	SERVER	R	1	ServerName
データベー スまたはサブシ ステム名	VARCHAR	80	DBNAME	R	2	DatabaseName
使用中のア プリケーション 名	VARCHAR	80	APPLNAME	R	3	ApplicationData
次元の所有者	VARCHAR	80	OWNER	O		OwnerName
次元名	VARCHAR	80	DIMENSON	R	4	DimensionName
次元のクラス またはタイプ	VARCHAR	80	TYPE	O		DimensionType
合計メンバ ー数	CHAR	10	TOTALCNT	O		DimensionCount
レベル数	CHAR	10	LEVELCNT	O		DimensionLevelCount

表 82. 多次元データベース内の次元オブジェクト・タイプのプロパティ (続き). このオブジェクト・タイプの MDIS 名は Dimension (次元) です。

プロパティ名	データ・タイプ	サイズ	プロパティ短縮名	値フラグ	MDIS 名へのマップ
アプリケーション固有の情報	VARCHAR	512	APPLDATA	O	ApplicationData
データにアクセスするための URL	VARCHAR	254	URL	O	ApplicationData
ソース定義の作成のタイム・スタンプ	CHAR	26	CRTTIME	O	DateCreated、TimeCreated
ソース定義の最終変更のタイム・スタンプ	CHAR	26	SRCDATCF	O	DateUpdated、TimeUpdated

注: S = 情報カタログ・マネージャー生成、R = 必須、O = オプション

DWC プロセス (DWC Process): データウェアハウスセンター内のプロセスを表現します。

このオブジェクト・タイプを定義するためのタグ言語は、
`¥VWSWIN¥DGWIN¥TYPES` ディレクトリーのファイル `FLGNYINF.TYP` に入っています。

このオブジェクト・タイプ用の情報カタログ・マネージャー DB2 ストレージ表の名前は `XXX.DWCPROC` です。表名の詳細については、176ページの『MDIS マッピング』を参照してください。

表 83 では、DWC プロセス・オブジェクト・タイプのプロパティに関する情報を提供します。

表 83. DWC プロセス・オブジェクト・タイプのプロパティ

プロパティ名	データ・タイプ	サイズ	プロパティ短縮名	値フラグ	MDIS 名へのマップ
名前	VARCHAR	80	NAME	R	1
短い記述	VARCHAR	250	SHRTDESC	O	
長い記述	LONG VARCHAR	32700	LONGDESC	O	

表 83. DWC プロセス・オブジェクト・タイプのプロパティ (続き)

プロパティ名	データ・タイプ	サイズ	プロパティ短縮名	値フラグ	UUI 順序
処置	VARCHAR	254	ACTIONS	O	
詳細の問い合わせ先	VARCHAR	80	RESPNSBL	O	
データにアクセスするための URL	VARCHAR	254	URL	O	
ソース定義の最終変更のタイム・スタンプ	CHAR	26	SRCDATCF	O	

注: S = 情報カタログ・マネージャー生成、R = 必須、O = オプション

エレメント (Elements): "列またはフィールド" オブジェクト・タイプに直接にマップしない MDIS Element (エレメント) オブジェクトを表現します。

このオブジェクト・タイプを定義するためのタグ言語は、`¥VWSWIN¥DGWIN¥TYPES` ディレクトリーのファイル `FLGNYELE.TYP` に入っています。

このオブジェクト・タイプ用の情報カタログ・マネージャー DB2 ストレージ表の名前は `XXX.ELEMENT` です。表名の詳細については、176ページの『MDIS マッピング』を参照してください。

表84 では、エレメント・オブジェクト・タイプのプロパティに関する情報を提供します。

表 84. エレメント・オブジェクト・タイプのプロパティ. このオブジェクト・タイプの MDIS 名は Element (エレメント) です。

プロパティ名	データ・タイプ	サイズ	プロパティ短縮名	値フラグ	UUI 順序
オブジェクト・タイプ ID	CHAR	6	OBJTYPID	S	
インスタンス ID	CHAR	10	INSTIDNT	S	
名前	VARCHAR	80	NAME	R	

表 84. エレメント・オブジェクト・タイプのプロパティ (続き). このオブジェクト・タイプの MDIS 名は Element (エレメント) です。

プロパティ名	データ・タイプ	サイズ	プロパティ短縮名	値フラグ	UI 順序
最終変更日時	TIMESTAMP	26	UPDATIME	S	
最終変更者	CHAR	8	UPDATEBY	S	
短い記述	VARCHAR	250	SHRTDESC	O	
長い記述	LONG VARCHAR	32700	LONGDESC	O	
処置	VARCHAR	254	ACTIONS	O	
詳細の問い合わせ先	VARCHAR	80	RESPNSBL	O	
エレメントの最後のリフレッシュ	CHAR	26	FRESHDAT	O	
データベースのホスト・サーバー名	VARCHAR	80	SERVER	R	1
データベースまたはサブシステム名	VARCHAR	80	DBNAME	R	2
エレメントの所有者	VARCHAR	80	OWNER	R	3
次元またはレコード名	VARCHAR	80	DIMRECNM	R	4
エレメント名	VARCHAR	80	ELEMNAME	R	5
データにアクセスするための URL	VARCHAR	254	URL	O	
エレメントのデータ・タイプ	CHAR	30	DATATYPE	O	
エレメントの長さ	CHAR	20	LENGTH	O	
エレメントの位取り	CHAR	5	SCALE	O	
エレメントの精度	CHAR	5	PRECDIG	O	

エレメント

表 84. エレメント・オブジェクト・タイプのプロパティ (続き). このオブジェクト・タイプの MDIS 名は Element (エレメント) です。

プロパティ名	データ・タイプ	サイズ	プロパティ短縮名	値フラグ	UI 順序
エレメントがヌル可能	CHAR	1	NULLS	O	
基本キーの中でのエレメントの位置	CHAR	5	KEYPOSNO	O	
エレメントの位置	CHAR	5	POSNO	O	
エレメントの序数	CHAR	5	ORDINAL	O	
ソース定義の作成のタイム・スタンプ	CHAR	26	CRTTIME	O	
ソース定義の最終変更のタイム・スタンプ	CHAR	26	SRCDATCF	O	

注: S = 情報カタログ・マネージャー生成、R = 必須、O = オプション

ファイル (Files): ファイル・システム内のファイルを表現します。

このオブジェクト・タイプを定義するためのタグ言語は、
 ¥VWSWIN¥DGWIN¥TYPES ディレクトリーのファイル FLGNYFIL.TYP に入っています。

このオブジェクト・タイプ用の情報カタログ・マネージャー DB2 ストレージ表の名前は XXX.FILE です。表名の詳細については、176ページの『MDIS マッピング』を参照してください。

191ページの表85 では、ファイル・オブジェクト・タイプのプロパティに関する情報を提供します。

表 85. ファイル・オブジェクト・タイプのプロパティ。このオブジェクト・タイプの MDIS 名は Record (レコード) です。

プロパティ名	データ・タイプ	サイズ	プロパティ短縮名	値フラグ	UUI 順序	MDIS 名へのマップ
オブジェクト・タイプ ID	CHAR	6	OBJTYPID	S		
インスタンス ID	CHAR	10	INSTIDNT	S		
名前	VARCHAR	80	NAME	R		RecordLongName
最終変更日時	TIMESTAMP	26	UPDATIME	S		
最終変更者	CHAR	8	UPDATEBY	S		
短い記述	VARCHAR	250	SHRTDESC	O		BriefDescription
長い記述	LONG VARCHAR	32700	LONGDESC	O		LongDescription
処置	VARCHAR	254	ACTIONS	O		ApplicationData
詳細の問い合わせ先	VARCHAR	80	RESPNSBL	O		ContactName
データベースのホスト・サーバー名	VARCHAR	80	SERVER	R	1	ServerName
データベースまたはサブシステム名	VARCHAR	80	DBNAME	R	2	DatabaseName
ファイルの所有者	VARCHAR	80	OWNER	R	3	OwnerName
ファイルのパスまたはディレクトリー	VARCHAR	254	FILEPATH	R	4	ApplicationData
ファイルのファイル名	VARCHAR	254	FILENAME	R	5	RecordName
ファイル・データの最後のリフレッシュ	CHAR	26	FRESHDAT	O		RecordLastRefreshDate

ファイル

表 85. ファイル・オブジェクト・タイプのプロパティ (続き). このオブジェクト・タイプの MDIS 名は Record (レコード) です。

プロパティ名	データ・タイプ	サイズ	プロパティ短縮名	値フラグ	UUI 順序	MDIS 名へのマップ
トランスフォーメーション・プログラムの最後の実行	CHAR	26	LASTRUN	0		ApplicationData
トランスフォーメーション・プログラムの実行頻度	VARCHAR	80	RUNFREQ	0		RecordUpdateFrequency
トランスフォーメーション・プログラムのタイプ	VARCHAR	32	SOURCE	0		ApplicationData
ファイルの部分または完全コピー/更新	CHAR	1	COPYCOMP	0		ApplicationData
コピー/更新されたデータが整合状態にある	CHAR	1	CONSIST	0		ApplicationData
トランスフォーメーション・プログラムの最後の変更	CHAR	26	PGMGEND	0		ApplicationData
トランスフォーメーション・プログラムの最後のコンパイル	CHAR	26	PGMCOMP	0		ApplicationData
ファイルのクラスまたはタイプ	VARCHAR	80	TYPE	0		RecordType
データにアクセスするための URL	VARCHAR	254	URL	0		ApplicationData

表 85. ファイル・オブジェクト・タイプのプロパティ (続き). このオブジェクト・タイプの MDIS 名は Record (レコード) です。

プロパティ名	データ・タイプ	サイズ	プロパティ短縮名	値フラグ	UI 順序	MDIS 名へのマップ
ソース定義の作成のタイム・スタンプ	CHAR	26	CRTTIME	O		DateCreated、TimeCreated
ソース定義の最終変更のタイム・スタンプ	CHAR	26	SRCDATCF	O		DateUpdated、TimeUpdated

注: S = 情報カタログ・マネージャー生成、R = 必須、O = オプション

IMS データベース定義 (DBD) (IMS database definitions): IMS データベース定義を表現します。

このオブジェクト・タイプを定義するためのタグ言語は、
 ¥VWSWIN¥DGWIN¥TYPES ディレクトリーのファイル FLGNYDBD.TYP に入っています。

このオブジェクト・タイプ用の情報カタログ・マネージャー DB2 ストレージ表の名前は XXX.IMSDBD です。表名の詳細については、176ページの『MDIS マッピング』を参照してください。

表86 では、IMS データベース定義 (DBD) オブジェクト・タイプのプロパティに関する情報を提供します。

表 86. IMS データベース定義 (DBD) オブジェクト・タイプのプロパティ. このオブジェクト・タイプの MDIS 名は Database (データベース) です。

プロパティ名	データ・タイプ	サイズ	プロパティ短縮名	値フラグ	UI 順序	MDIS 名へのマップ
オブジェクト・タイプ ID	CHAR	6	OBJTYPID	S		
インスタンス ID	CHAR	10	INSTIDNT	S		
名前	VARCHAR	80	NAME	R		DatabaseLongName
最終変更日時	TIMESTAMP	26	UPDATIME	S		
最終変更者	CHAR	8	UPDATEBY	S		

IMS データベース定義 (DBD)

表 86. IMS データベース定義 (DBD) オブジェクト・タイプのプロパティ (続き). このオブジェクト・タイプの MDIS 名は Database (データベース) です。

プロパティ名	データ・タイプ	サイズ	プロパティ短縮名	値フラグ	UUI 順序	MDIS 名へのマップ
短い記述	VARCHAR	250	SHRTDESC	O		BriefDescription
長い記述	LONG VARCHAR	32700	LONGDESC	O		LongDescription
処置	VARCHAR	254	ACTIONS	O		ApplicationData
データベースの最後のリフレッシュ	CHAR	26	FRESHDAT	O		ApplicationData
詳細の問い合わせ先	VARCHAR	80	RESPNSBL	O		ContactName
データベースの所有者	VARCHAR	80	OWNER	O		OwnerName
データベースのホスト・サーバー名	VARCHAR	80	SERVER	R	1	ServerName
データベースのサーバー・タイプ	VARCHAR	80	SRVRTYPE	O		ServerType
データベースまたはサブシステム名	VARCHAR	80	DBNAME	R	2	DatabaseName
データベースのタイプ	VARCHAR	80	DBTYPE	R	3	DatabaseType
データベースの拡張タイプ	VARCHAR	40	DBETYPE	O		ApplicationData
データベースの状況	VARCHAR	80	DBSTAT	O		DatabaseStatus
IMS のアクセス方式	VARCHAR	80	IMSACC	O		ApplicationData
オペレーティング・システムのアクセス方式	VARCHAR	80	OSACC	O		ApplicationData
共用索引名	VARCHAR	320	SHRINDEX	O		ApplicationData

表 86. IMS データベース定義 (DBD) オブジェクト・タイプのプロパティ (続き). このオブジェクト・タイプの MDIS 名は Database (データベース) です。

プロパティ名	データ・タイプ	サイズ	プロパティ短縮名	値フラグ	UI 順序	MDIS 名へのマップ
データにアクセスするための URL	VARCHAR	254	URL	O		ApplicationData
ソース定義の作成のタイム・スタンプ	CHAR	26	CRTTIME	O		DateCreated、TimeCreated
ソース定義の最終変更のタイム・スタンプ	CHAR	26	SRCDATCF	O		DateUpdated、TimeUpdated

注: S = 情報カタログ・マネージャー生成、R = 必須、O = オプション

IMS プログラム制御ブロック (PCB) (IMS program control blocks): IMS プログラム制御ブロックを表現します。

このオブジェクト・タイプを定義するためのタグ言語は、
 ¥VWSWIN¥DGWIN¥TYPES ディレクトリーのファイル FLGNYPCB.TYP に入
 っています。

このオブジェクト・タイプ用の情報カタログ・マネージャー DB2 ストレージ
 表の名前は XXX.IMSPCB です。表名の詳細については、176ページの『MDIS
 マッピング』を参照してください。

表87 では、IMS プログラム制御ブロック (PCB) オブジェクト・タイプのプロ
 パティに関する情報を提供します。

表 87. IMS プログラム制御ブロック (PCB) オブジェクト・タイプのプロパティ. このオブジェクト・タイプの MDIS 名は Subschema (サブスキーマ) です。

プロパティ名	データ・タイプ	サイズ	プロパティ短縮名	値フラグ	UI 順序	MDIS 名へのマップ
オブジェクト・タイプ ID	CHAR	6	OBJTYPID	S		
インスタンス ID	CHAR	10	INSTIDNT	S		
名前	VARCHAR	80	NAME	R		SubschemaLongName

IMS プログラム制御ブロック (PCB)

表 87. IMS プログラム制御ブロック (PCB) オブジェクト・タイプのプロパティ (続き). このオブジェクト・タイプの MDIS 名は Subschema (サブスキーマ) です。

プロパティ名	データ・タイプ	サイズ	プロパティ短縮名	値フラグ	UI 順序	MDIS 名へのマップ
最終変更日時	TIMESTAMP	26	UPDATIME	S		
最終変更者	CHAR	8	UPDATEBY	S		
短い記述	VARCHAR	250	SHRTDESC	O		BriefDescription
長い記述	LONG VARCHAR	32700	LONGDESC	O		LongDescription
処置	VARCHAR	254	ACTIONS	O		ApplicationData
詳細の問い合わせ先	VARCHAR	80	RESPNSBL	O		ContactName
データベースのホスト・サーバー名	VARCHAR	80	SERVER	R	1	ServerName
データベースまたはサブシステム名	VARCHAR	80	DBNAME	R	2	DatabaseName
PCB 名	VARCHAR	80	PCBNAME	R	3	SubschemaName
PCB の所有者	VARCHAR	80	OWNER	O		OwnerName
データにアクセスするための URL	VARCHAR	254	URL	O		ApplicationData
ソース定義の作成のタイム・スタンプ	CHAR	26	CRTTIME	O		DateCreated、TimeCreated
ソース定義の最終変更のタイム・スタンプ	CHAR	26	SRCDATCF	O		DateUpdated、TimeUpdated

注: S = 情報カタログ・マネージャー生成、R = 必須、O = オプション

IMS プログラム仕様ブロック (PSB) (IMS program specification blocks):
IMS プログラム仕様ブロックを表現します。

このオブジェクト・タイプを定義するためのタグ言語は、
¥VWSWIN¥DGWIN¥TYPES ディレクトリーのファイル FLGNYPBS.TYP に入っています。

このオブジェクト・タイプ用の情報カタログ・マネージャー DB2 ストレージ表の名前は XXX.PSB です。表名の詳細については、176ページの『MDIS マッピング』を参照してください。

195ページの表87 では、IMS プログラム仕様ブロック (PSB) オブジェクト・タイプのプロパティに関する情報を提供します。

表 88. IMS プログラム仕様ブロック (PSB) オブジェクト・タイプのプロパティ. このオブジェクト・タイプの MDIS 名は Subschema (サブスキーマ) です。

プロパティ名	データ・タイプ	サイズ	プロパティ短縮名	値フラグ	UI 順序	MDIS 名へのマップ
オブジェクト・タイプ ID	CHAR	6	OBJTYPID	S		
インスタンス ID	CHAR	10	INSTIDNT	S		
名前	VARCHAR	80	NAME	R		DatabaseLongName
最終変更日時	TIMESTAMP	26	UPDATIME	S		
最終変更者	CHAR	8	UPDATEBY	S		
短い記述	VARCHAR	250	SHRTDESC	O		BriefDescription
長い記述	LONG VARCHAR	32700	LONGDESC	O		LongDescription
処置	VARCHAR	254	ACTIONS	O		ApplicationData
詳細の問い合わせ先	VARCHAR	80	RESPNSBL	O		ContactName
データベースのホスト・サーバー名	VARCHAR	80	SERVER	R	1	ServerName
データベースのサーバー・タイプ	VARCHAR	80	SRVRTYPE	O		ServerType
データベースのタイプ	VARCHAR	80	DBTYPE	R	3	DatabaseType
データベースの拡張タイプ	VARCHAR	40	DBETYPE	O		ApplicationData
データベースの状況	VARCHAR	80	DBSTAT	O		DatabaseStatus
PSB 名	VARCHAR	80	PSBNAME	R	2	DatabaseName
PSB の所有者	VARCHAR	80	OWNER	O		OwnerName

IMS プログラム仕様ブロック (PSB)

表 88. IMS プログラム仕様ブロック (PSB) オブジェクト・タイプのプロパティ (続き). このオブジェクト・タイプの MDIS 名は Subschema (サブスキーマ) です。

プロパティ名	データ・タイプ	サイズ	プロパティ短縮名	値フラグ	UI 順序	MDIS 名へのマップ
データにアクセスするための URL	VARCHAR	254	URL	O		ApplicationData
ソース定義の作成のタイム・スタンプ	CHAR	26	CRTTIME	O		DateCreated、TimeCreated
ソース定義の最終変更のタイム・スタンプ	CHAR	26	SRCDATCF	O		DateUpdated、TimeUpdated

注: S = 情報カタログ・マネージャー生成、R = 必須、O = オプション

IMS セグメント (IMS segments): IMS セグメントを表現します。

このオブジェクト・タイプを定義するためのタグ言語は、
 ¥VWSWIN¥DGWIN¥TYPES ディレクトリーのファイル FLGNYSEG.TYP に入っています。

このオブジェクト・タイプ用の情報カタログ・マネージャー DB2 ストレージ表の名前は XXX.IMSSEG です。表名の詳細については、176ページの『MDIS マッピング』を参照してください。

表89 では、IMS セグメント・オブジェクト・タイプのプロパティに関する情報を提供します。

表 89. IMS セグメント・オブジェクト・タイプのプロパティ. このオブジェクト・タイプの MDIS 名は Record (レコード) です。

プロパティ名	データ・タイプ	サイズ	プロパティ短縮名	値フラグ	UI 順序	MDIS 名へのマップ
オブジェクト・タイプ ID	CHAR	6	OBJTYPID	S		
インスタンス ID	CHAR	10	INSTIDNT	S		
名前	VARCHAR	80	NAME	R		RecordLongName
最終変更日時	TIMESTAMP	26	UPDATIME	S		
最終変更者	CHAR	8	UPDATEBY	S		

表 89. IMS セグメント・オブジェクト・タイプのプロパティ (続き). このオブジェクト・タイプの MDIS 名は Record (レコード) です。

プロパティ名	データ・タイプ	サイズ	プロパティ短縮名	値フラグ	UUI 順序	MDIS 名へのマップ
短い記述	VARCHAR	250	SHRTDESC	O		BriefDescription
長い記述	LONG VARCHAR	32700	LONGDESC	O		LongDescription
処置	VARCHAR	254	ACTIONS	O		ApplicationData
セグメントの最後のリフレッシュ	CHAR	26	FRESHDAT	O		RecordLastRefreshDate
詳細の問い合わせ先	VARCHAR	80	RESPNSBL	O		ContactName
データベースのホスト・サーバー名	VARCHAR	80	SERVER	O		ServerName
データベースまたはサブシステム名	VARCHAR	80	DBNAME	R	1	DatabaseName
セグメント名	VARCHAR	80	SEGNAME	R	2	RecordName
セグメントの所有者	VARCHAR	80	OWNER	O		OwnerName
セグメント・タイプ	VARCHAR	80	TYPE	O		RecordType
セグメントの最大長	CHAR	5	MAXLEN	O		ApplicationData
セグメントの最小長	CHAR	5	MINLEN	O		ApplicationData
実際の論理子セグメントのソース	CHAR	20	PSEGSRC	O		ApplicationData
論理親の連結キーのソース	CHAR	20	LPCKSRC	O		ApplicationData
トランスフォーメーション・プログラムの最後の実行	CHAR	26	LASTRUN	O		ApplicationData

IMS セグメント

表 89. IMS セグメント・オブジェクト・タイプのプロパティ (続き). このオブジェクト・タイプの MDIS 名は Record (レコード) です。

プロパティ名	データ・タイプ	サイズ	プロパティ短縮名	値フラグ	UI 順序	MDIS 名へのマップ
トランスフォーメーション・プログラムの実行頻度	VARCHAR	80	RUNFREQ	O		RecordUpdateFrequency
データにアクセスするための URL	VARCHAR	254	URL	O		ApplicationData
ソース定義の作成のタイム・スタンプ	CHAR	26	CRTTIME	O		DateCreated、TimeCreated
ソース定義の最終変更のタイム・スタンプ	CHAR	26	SRCDATCF	O		DateUpdated、TimeUpdated

注: S = 情報カタログ・マネージャー生成、R = 必須、O = オプション

多次元データベース内のメンバー (Members within a multidimensional database):

多次元データベース内のメンバーを表現します。メンバーは次元の一部であり、次元は多次元データベースの一部です。

このオブジェクト・タイプを定義するためのタグ言語は、

¥VWSWIN¥DGWIN¥TYPES ディレクトリーのファイル FLGNYMEM.TYP に入っています。

このオブジェクト・タイプ用の情報カタログ・マネージャー DB2 ストレージ表の名前は XXX.MEMBER です。表名の詳細については、176ページの『MDIS マッピング』を参照してください。

201ページの表90 では、多次元データベース内のメンバー・オブジェクト・タイプのプロパティに関する情報を提供します。

表 90. 多次元データベース内のメンバー・オブジェクト・タイプのプロパティ。このオブジェクト・タイプの MDIS 名は Element (エレメント) です。

プロパティ名	データ・タイプ	サイズ	プロパティ短縮名	値フラグ	UUI 順序	MDIS 名へのマップ
オブジェクト・タイプ ID	CHAR	6	OBJTYPID	S		
インスタンス ID	CHAR	10	INSTIDNT	S		
名前	CHAR	80	NAME	R		ElementLongName
最終変更日時	TIMESTAMP	26	UPDATIME	S		
最終変更者	CHAR	8	UPDATEBY	S		
短い記述	VARCHAR	250	SHRTDESC	O		BriefDescription
長い記述	LONG VARCHAR	32700	LONGDESC	O		LongDescription
処置	VARCHAR	254	ACTIONS	O		ApplicationData
詳細の問い合わせ先	VARCHAR	80	RESPNSBL	O		ContactName
メンバーの最後のリフレッシュ	CHAR	26	FRESHDAT	O		ElementLastRefreshDate
メンバーの所有者	VARCHAR	80	OWNER	O		OwnerName
データベースのホスト・サーバー名	VARCHAR	80	SERVER	R	1	ServerName
データベースまたはサブシステム名	VARCHAR	80	DBNAME	R	2	DatabaseName
使用中のアプリケーション名	VARCHAR	80	APPLNAME	R	3	ApplicationData
次元名	VARCHAR	80	DIMENSON	R	4	DimensionName
メンバー名	VARCHAR	80	MEMBER	R	5	ElementName
メンバーのデータ・タイプ	CHAR	30	DATATYPE	O		ElementDataType
メンバーの長さ	CHAR	20	LENGTH	O		ElementLength

多次元データベース内のメンバー

表 90. 多次元データベース内のメンバー・オブジェクト・タイプのプロパティ (続き). このオブジェクト・タイプの MDIS 名は Element (エレメント) です。

プロパティ名	データ・タイプ	サイズ	プロパティ短縮名	値フラグ	UI 順序	MDIS 名へのマップ
メンバーの位取り	CHAR	5	SCALE	O		ApplicationData
メンバーの精度	CHAR	5	PRECDIG	O		ElementPrecision
メンバーがヌル可能	CHAR	1	NULLS	O		ElementNulls
基本キーの中でのメンバーの位置	CHAR	5	KEYPOSNO	O		ElementKeyPosition
メンバーの位置	CHAR	5	POSNO	O		ElementPosition
メンバーの序数	CHAR	5	ORDINAL	O		ElementOrdinality
派生元	VARCHAR	512	DERIVED	O		ApplicationData
アプリケーション固有の情報	VARCHAR	512	APPLDATA	O		ApplicationData
データにアクセスするための URL	VARCHAR	254	URL	O		ApplicationData
ソース定義の作成のタイム・スタンプ	CHAR	26	CRTTIME	O		DateCreated、TimeCreated
ソース定義の最終変更のタイム・スタンプ	CHAR	26	SRCDATCF	O		DateUpdated、TimeUpdated

注: S = 情報カタログ・マネージャー生成、R = 必須、O = オプション

多次元データベース (Multidimensional databases): 多次元データベースを表現します。

このオブジェクト・タイプを定義するためのタグ言語は、
 ¥VWSWIN¥DGWIN¥TYPES ディレクトリーのファイル FLGNYOLA.TYP に入っています。

このオブジェクト・タイプ用の情報カタログ・マネージャー DB2 ストレージ表の名前は XXX.OLAPMODL です。表名の詳細については、176ページの『MDIS マッピング』を参照してください。

表91 では、多次元データベース・オブジェクト・タイプのプロパティに関する情報を提供します。

表 91. 多次元データベース・オブジェクト・タイプのプロパティ. このオブジェクト・タイプの MDIS 名は Database (データベース) です。

プロパティ 名	データ・ タイプ	サイズ	プロパティ 短縮名	値フラグ	UUI 順序	MDIS 名へのマップ
オブジェ クト・タイプ ID	CHAR	6	OBJTYPID	S		
インスタンス ID	CHAR	10	INSTIDNT	S		
名前	VARCHAR	80	NAME	R		DatabaseLongName
最終変更日時	TIMESTAMP	26	UPDATIME	S		
最終変更者	CHAR	8	UPDATEBY	S		
短い記述	VARCHAR	250	SHRTDESC	O		BriefDescription
長い記述	LONG VARCHAR	32700	LONGDESC	O		LongDescription
処置	VARCHAR	254	ACTIONS	O		ApplicationData
詳細の問い合 わせ先	VARCHAR	80	RESPNSBL	O		ContactName
データベー スの最後のリフ レッシュ	CHAR	26	FRESHDAT	O		ApplicationData
データベー スの所有者	VARCHAR	80	OWNER	O		OwnerName
データベー スのホスト・サ ーバー名	VARCHAR	80	SERVER	R	1	ServerName
データベー スのサーバー・ タイプ	VARCHAR	80	SRVRTYPE	O		ServerType
データベー スまたはサブシ ステム名	VARCHAR	80	DBNAME	R	2	DatabaseName

多次元データベース

表 91. 多次元データベース・オブジェクト・タイプのプロパティ (続き). このオブジェクト・タイプの MDIS 名は Database (データベース) です。

プロパティ名	データ・タイプ	サイズ	プロパティ短縮名	値フラグ	UI 順序	MDIS 名へのマップ
データベースのタイプ	VARCHAR	80	DBTYPE	O		DatabaseType
データベースの拡張タイプ	VARCHAR	20	DBETYPE	O		ApplicationData
データベースの状況	VARCHAR	80	DBSTAT	O		DatabaseStatus
使用中のアプリケーション名	VARCHAR	80	APPLNAME	R	3	ApplicationData
アプリケーション固有の情報	VARCHAR	512	APPLDATA	O		ApplicationData
データにアクセスするための URL	VARCHAR	254	URL	O		ApplicationData
ソース定義の作成のタイム・スタンプ	CHAR	26	CRTTIME	O		DateCreated、TimeCreated
ソース定義の最終変更のタイム・スタンプ	CHAR	26	SRCDATCF	O		DateUpdated、TimeUpdated

注: S = 情報カタログ・マネージャー生成、R = 必須、O = オプション

レコード (Records): 『ファイル』 または 『リレーショナル表および視点』 オブジェクト・タイプに直接にマップしない MDIS Record (レコード) オブジェクトを表現します。レコードはエレメントから構成されます。

このオブジェクト・タイプを定義するためのタグ言語は、`¥VWSWIN¥DGWIN¥TYPES` ディレクトリーのファイル `FLGNYREC.TYP` に入っています。

このオブジェクト・タイプ用の情報カタログ・マネージャー DB2 ストレージ表の名前は `XXX.RECORD` です。表名の詳細については、176ページの『MDIS マッピング』を参照してください。

表92 では、レコード・オブジェクト・タイプのプロパティに関する情報を提供します。

表 92. レコード・オブジェクト・タイプのプロパティ. このオブジェクト・タイプの MDIS 名は Record (レコード) です。

プロパティ名	データ・タイプ	サイズ	プロパティ短縮名	値フラグ	UUI 順序	MDIS 名へのマップ
オブジェクト・タイプ ID	CHAR	6	OBJTYPID	S		
インスタンス ID	CHAR	10	INSTIDNT	S		
名前	VARCHAR	80	NAME	R		RecordLongName
最終変更日時	TIMESTAMP	26	UPDATIME	S		
最終変更者	CHAR	8	UPDATEBY	S		
短い記述	VARCHAR	250	SHRTDESC	O		BriefDescription
長い記述	LONG VARCHAR	32700	LONGDESC	O		LongDescription
処置	VARCHAR	254	ACTIONS	O		ApplicationData
詳細の問い合わせ先	VARCHAR	80	RESPNSBL	O		ContactName
データベースのホスト・サーバー名	VARCHAR	80	SERVER	R	1	ServerName
データベースまたはサブシステム名	VARCHAR	80	DBNAME	R	2	DatabaseName
レコードの所有者	VARCHAR	80	OWNER	R	3	OwnerName
レコード名	VARCHAR	80	RECNAME	R	4	RecordName
レコード・データの最後のリフレッシュ	CHAR	26	FRESHDAT	O		RecordLastRefreshDate
トランスフォーメーション・プログラムの最後の実行	CHAR	26	LASTRUN	O		ApplicationData

レコード

表 92. レコード・オブジェクト・タイプのプロパティ (続き). このオブジェクト・タイプの MDIS 名は Record (レコード) です。

プロパティ名	データ・タイプ	サイズ	プロパティ短縮名	値フラグ	MDIS 名へのマップ
トランスフォーメーション・プログラムの実行頻度	VARCHAR	80	RUNFREQ	O	RecordUpdateFrequency
レコード・タイプ	VARCHAR	80	TYPE	O	RecordType
データにアクセスするための URL	VARCHAR	254	URL	O	ApplicationData
ソース定義の作成のタイム・スタンプ	CHAR	26	CRTTIME	O	DateCreated、TimeCreated
ソース定義の最終変更のタイム・スタンプ	CHAR	26	SRCDATCF	O	DateUpdated、TimeUpdated

注: S = 情報カタログ・マネージャー生成、R = 必須、O = オプション

リレーショナル表および視点 (Relational tables and views): リレーショナル・データベースの表または視点を表現します。

このオブジェクト・タイプを定義するためのタグ言語は、`¥VWSWIN¥DGWIN¥TYPES` ディレクトリーのファイル `FLGNYTAB.TYP` に入っています。

このオブジェクト・タイプ用の情報カタログ・マネージャー DB2 ストレージ表の名前は `XXX.TABLES` です。表名の詳細については、176ページの『MDIS マッピング』を参照してください。

207ページの表93 では、リレーショナル表および視点オブジェクト・タイプのプロパティに関する情報を提供します。

表 93. リレーショナル表および視点オブジェクト・タイプのプロパティ。このオブジェクト・タイプの MDIS 名は Record (レコード) です。

プロパティ名	データ・タイプ	サイズ	プロパティ短縮名	値フラグ	UUI 順序	MDIS 名へのマップ
オブジェクト・タイプ ID	CHAR	6	OBJTYPID	S		
インスタンス ID	CHAR	10	INSTIDNT	S		
名前	VARCHAR	80	NAME	R		RecordLongName
最終変更日時	TIMESTAMP	26	UPDATIME	S		
最終変更者	CHAR	8	UPDATEBY	S		
短い記述	VARCHAR	250	SHRTDESC	O		BriefDescription
長い記述	LONG VARCHAR	32700	LONGDESC	O		LongDescription
処置	VARCHAR	254	ACTIONS	O		ApplicationData
カタログ注釈	VARCHAR	254	REMARKS	O		ApplicationData
詳細の問い合わせ先	VARCHAR	80	RESPNSBL	O		ContactName
データベースのホスト・サーバー名	VARCHAR	80	SERVER	O		ServerName
ローカル・データベース別名	CHAR	8	DBALIAS	O		ApplicationData
データベースまたはサブシステム名	VARCHAR	80	DBNAME	R	1	DatabaseName
表の所有者	VARCHAR	80	OWNER	R	2	OwnerName
表名	VARCHAR	80	TABLES	R	3	RecordName
基礎表の所有者名	CHAR	30	SRCOWNER	O		ApplicationData
基礎表の名前	CHAR	128	SRCTBNAM	O		ApplicationData
表データの最後のリフレッシュ	CHAR	26	FRESHDAT	O		RecordLastRefreshDate

リレーショナル表および視点

表 93. リレーショナル表および視点オブジェクト・タイプのプロパティ (続き). このオブジェクト・タイプの MDIS 名は Record (レコード) です。

プロパティ名	データ・タイプ	サイズ	プロパティ短縮名	値フラグ	UI 順序	MDIS 名へのマップ
トランスフォーメーション・プログラムの実行モード	CHAR	30	RUNMODE	O		ApplicationData
トランスフォーメーション・プログラムの最後の実行	CHAR	26	LASTRUN	O		ApplicationData
トランスフォーメーション・プログラムの実行頻度	VARCHAR	80	RUNFREQ	O		RecordUpdateFrequency
トランスフォーメーション・プログラムのタイプ	VARCHAR	32	SOURCE	O		ApplicationData
表の部分または完全コピー/更新	CHAR	1	COPYCOMP	O		ApplicationData
コピー/更新されたデータが整合状態にある	CHAR	1	CONSIST	O		ApplicationData
カタログのリフレッシュ/更新頻度	VARCHAR	80	REFRESH	O		ApplicationData
トランスフォーメーション・プログラムの最後の変更	CHAR	26	PGMGEND	O		ApplicationData

表 93. リレーショナル表および視点オブジェクト・タイプのプロパティ (続き). このオブジェクト・タイプの MDIS 名は Record (レコード) です。

プロパティ名	データ・タイプ	サイズ	プロパティ短縮名	値フラグ	UUI 順序	MDIS 名へのマップ
トランスフォーメーション・プログラムの最後のコンパイル	CHAR	26	PGMCOMP	O		ApplicationData
表タイプ	VARCHAR	80	TYPE	O		RecordType
定義が視点を表現する	CHAR	1	TABLVIEW	O		ApplicationData
表の内部名	CHAR	18	IDSINAME	O		ApplicationData
表が次元表として使用される	CHAR	1	IDSDIM	O		ApplicationData
データにアクセスするための URL	VARCHAR	254	URL	O		ApplicationData
ソース定義の作成のタイム・スタンプ	CHAR	26	CRTTIME	O		DateCreated、TimeCreated
ソース定義の最終変更のタイム・スタンプ	CHAR	26	SRCDATCF	O		DateUpdated、TimeUpdated

注: S = 情報カタログ・マネージャー生成、R = 必須、O = オプション

スタースキーマ (Star Schemas): リレーショナル・データを表現します。

このオブジェクト・タイプを定義するためのタグ言語は、
 ¥VWSWIN¥DGWIN¥TYPES ディレクトリーのファイル FLGNYSUB.TYP に入
 っています。

このオブジェクト・タイプ用の情報カタログ・マネージャー DB2 ストレージ
 表の名前は XXX.STARSCHM です。表名の詳細については、176ページの
 『MDIS マッピング』を参照してください。

210ページの表94 では、スタースキーマ・オブジェクト・タイプのプロパティ
 ーに関する情報を提供します。

スタースキーマ

表 94. スタースキーマ・オブジェクト・タイプのプロパティ

プロパティ名	データ・タイプ	サイズ	プロパティ短縮名	値フラグ	UUI 順序
名前	VARCHAR	80	NAME	R	1
短い記述	VARCHAR	250	SHRTDESC	O	
長い記述	LONG VARCHAR	32700	LONGDESC	O	
処置	VARCHAR	254	ACTIONS	O	
詳細の問い合わせ先	VARCHAR	80	RESPNSBL	O	
データにアクセスするための URL	VARCHAR	254	URL	O	
ソース定義の最終変更のタイム・スタンプ	CHAR	26	SRCDATCF	O	

注: S = 情報カタログ・マネージャー生成、R = 必須、O = オプション

サブスキーマ (Subschemas): データベース内のレコードの論理的なグループ分けを表現します。

このオブジェクト・タイプを定義するためのタグ言語は、`¥VWSWIN¥DGWIN¥TYPES` ディレクトリーのファイル `FLGNYSUB.TYP` に入っています。

このオブジェクト・タイプ用の情報カタログ・マネージャー DB2 ストレージ表の名前は `XXX.SUBSCHEM` です。表名の詳細については、176ページの『MDIS マッピング』を参照してください。

211ページの表95 では、サブスキーマ・オブジェクト・タイプのプロパティに関する情報を提供します。

表 95. サブスキーマ・オブジェクト・タイプのプロパティ。このオブジェクト・タイプの MDIS 名は Subschema (サブスキーマ) です。

プロパティ名	データ・タイプ	サイズ	プロパティ短縮名	値フラグ	UI 順序	MDIS 名へのマップ
オブジェクト・タイプ ID	CHAR	6	OBJTYPID	S		
インスタンス ID	CHAR	10	INSTIDNT	S		
名前	VARCHAR	80	NAME	R		SubschemaLongName
最終変更日時	TIMESTAMP	26	UPDATIME	S		
最終変更者	CHAR	8	UPDATEBY	S		
短い記述	VARCHAR	250	SHRTDESC	O		BriefDescription
長い記述	LONG VARCHAR	32700	LONGDESC	O		LongDescription
処置	VARCHAR	254	ACTIONS	O		ApplicationData
詳細の問い合わせ先	VARCHAR	80	RESPNSBL	O		ContactName
データベースのホスト・サーバー名	VARCHAR	80	SERVER	R	1	ServerName
データベースまたはサブシステム名	VARCHAR	80	DBNAME	R	2	DatabaseName
サブスキーマの所有者	VARCHAR	80	OWNER	O		OwnerName
サブスキーマ名	VARCHAR	80	SSNAME	R	3	SubschemaName
データにアクセスするための URL	VARCHAR	254	URL	O		ApplicationData
ソース定義の作成のタイム・スタンプ	CHAR	26	CRTTIME	O		DateCreated、TimeCreated

サブスキーマ

表 95. サブスキーマ・オブジェクト・タイプのプロパティ (続き). このオブジェクト・タイプの MDIS 名は Subschema (サブスキーマ) です。

プロパティ名	データ・タイプ	サイズ	プロパティ短縮名	値フラグ	UUI 順序	MDIS 名へのマップ
ソース定義の最終変更のタイム・スタンプ	CHAR	26	SRCDATCF	O		DateUpdated、TimeUpdated

注: S = 情報カタログ・マネージャー生成, R = 必須, O = オプション

トランスフォーメーション (Transformations): ターゲット・リレーショナル・データベース内の列にデータを取り込むために使用される式または論理を表現します。トランスフォーメーション・オブジェクトは、ソース・オペレーション・データをターゲット列に変換するために使用される式、またはソース・フィールドとターゲット列の 1 対 1 のマッピングを示します。

このオブジェクト・タイプを定義するためのタグ言語は、`¥VWSWIN¥DGWIN¥TYPES` ディレクトリーのファイル `FLGNYFLT.TYP` に入っています。

このオブジェクト・タイプ用の情報カタログ・マネージャー DB2 ストレージ表の名前は `XXX.FILTER` です。表名の詳細については、176ページの『MDIS マッピング』を参照してください。

表96 では、トランスフォーメーション・オブジェクト・タイプのプロパティに関する情報を提供します。

表 96. トランスフォーメーション・オブジェクト・タイプのプロパティ. このオブジェクト・タイプの MDIS 名は Relationship (リレーションシップ) です。

プロパティ名	データ・タイプ	サイズ	プロパティ短縮名	値フラグ	UUI 順序	MDIS 名へのマップ
オブジェクト・タイプ ID	CHAR	6	OBJTYPID	S		
インスタンス ID	CHAR	10	INSTIDNT	S		
名前	VARCHAR	80	NAME	R		RelationshipLongName
最終変更日時	TIMESTAMP	26	UPDATIME	S		
最終変更者	CHAR	8	UPDATEBY	S		

表 96. トランスフォーメーション・オブジェクト・タイプのプロパティ (続き). このオブジェクト・タイプの MDIS 名は Relationship (リレーションシップ) です。

プロパティ名	データ・タイプ	サイズ	プロパティ短縮名	値フラグ	UUI 順序	MDIS 名へのマップ
短い記述	VARCHAR	250	SHRTDESC	O		BriefDescription
長い記述	LONG VARCHAR	32700	LONGDESC	O		LongDescription
処置	VARCHAR	254	ACTIONS	O		ApplicationData
トランスフォーメーション・プログラム名	VARCHAR	80	FPNAME	R	1	ApplicationData
トランスフォーメーション ID	VARCHAR	254	FIDENT	R	2	RelationshipName
トランスフォーメーションのクラスまたはタイプ	VARCHAR	80	TYPE	R	3	RelationshipType
ソース列/フィールド名、またはパラメーター	LONG VARCHAR	32700	FEXPRESS	O		RelationshipExpression
データベースのホスト・サーバー名	VARCHAR	80	SERVER	O		ServerName
トランスフォーメーションの所有者	VARCHAR	80	OWNER	O		OwnerName
ソースのシーケンス	CHAR	5	SRCSEQ	O		SourceSequenceOrder
トランスフォーメーションの序数	CHAR	5	ORDINAL	O		RelationshipOrdinality
トランスフォーメーションの両方向性	CHAR	1	DIRECT	O		RelationshipBidirectional

トランスフォーメーション

表 96. トランスフォーメーション・オブジェクト・タイプのプロパティ (続き). このオブジェクト・タイプの MDIS 名は Relationship (リレーションシップ) です。

プロパティ名	データ・タイプ	サイズ	プロパティ短縮名	値フラグ	UUI 順序	MDIS 名へのマップ
データにアクセスするための URL	VARCHAR	254	URL	O		ApplicationData
ソース定義の作成のタイム・スタンプ	CHAR	26	CRTTIME	O		DateCreated、TimeCreated
ソース定義の最終変更のタイム・スタンプ	CHAR	26	SRCDATCF	O		DateUpdated、TimeUpdated
詳細の問い合わせ先	VARCHAR	80	RESPNSBL			ContactName

注: S = 情報カタログ・マネージャー生成、R = 必須、O = オプション

基礎的区分

基礎的区分には、次のオブジェクト・タイプが含まれます。

- 『オーディオ・クリップ (Audio clips)』
- 215ページの『図表 (Charts)』
- 217ページの『文書 (Documents)』
- 218ページの『イメージまたはグラフィックス (Images or graphics)』
- 219ページの『インターネット文書 (Internet documents)』
- 220ページの『Lotus Approach 照会 (Lotus Approach queries)』
- 221ページの『プレゼンテーション (Presentations)』
- 222ページの『スプレッドシート (Spreadsheets)』
- 224ページの『テキスト・ベース・レポート (Text-based reports)』
- 225ページの『ビデオ・クリップ (Video clips)』

オーディオ・クリップ (Audio clips): 音声情報を含むファイルを表現します。これらのオブジェクトでは、電子的な (AUD ファイル)、または媒体に記録された (たとえば、CD、テープ) 音声情報を表現することができます。

このオブジェクト・タイプを定義するためのタグ言語は、
 ¥VWSWIN¥DGWIN¥TYPES ディレクトリーのファイル FLGNYAUD.TYP に入
 っています。

このオブジェクト・タイプ用の情報カタログ・マネージャー DB2 ストレージ
 表の名前は XXX.AUDIO です。表名の詳細については、176ページの『MDIS
 マッピング』を参照してください。

表97 では、オーディオ・クリップ・オブジェクト・タイプのプロパティーに関
 する情報を提供します。

表97. オーディオ・クリップ・オブジェクト・タイプのプロパティー

プロパティー 名	データ・ タイプ	サイズ	プロパティー 短縮名	値フラグ	UUI 順序
オブジェク ト・タイプ ID	CHAR	6	OBJTYPID	S	
インスタンス ID	CHAR	10	INSTIDNT	S	
名前	VARCHAR	80	NAME	R	
最終変更日時	TIMESTAMP	26	UPDATIME	S	
最終変更者	CHAR	8	UPDATEBY	S	
短い記述	VARCHAR	250	SHRTDESC	O	
長い記述	LONG VARCHAR	32700	LONGDESC	O	
処置	VARCHAR	254	ACTIONS	O	
オーディオ・ クリップのフ ァイル名	VARCHAR	254	FILENAME	R	1
オーディオ・ クリップのク ラスまたはタ イプ	VARCHAR	80	TYPE	R	2
データにアク セスするため の URL	VARCHAR	254	URL	O	

注: S = 情報カタログ・マネージャー生成、R = 必須、O = オプション

図表 (Charts): 印刷された、または電子的な図表を表現します。

図表

このオブジェクト・タイプを定義するためのタグ言語は、
 ¥VWSWIN¥DGWIN¥TYPES ディレクトリーのファイル FLGNYCHA.TYP に入
 っています。

このオブジェクト・タイプ用の情報カタログ・マネージャー DB2 ストレージ
 表の名前は XXX.CHARTS です。表名の詳細については、176ページの『MDIS
 マッピング』を参照してください。

表98 では、図表オブジェクト・タイプのプロパティーに関する情報を提供しま
 す。

表 98. 図表オブジェクト・タイプのプロパティー

プロパティー 名	データ・ タイプ	サイズ	プロパティー 短縮名	値フラグ	UUI 順序
オブジェク ト・タイプ ID	CHAR	6	OBJTYPID	S	
インスタンス ID	CHAR	10	INSTIDNT	S	
名前	VARCHAR	80	NAME	R	
最終変更日時	TIMESTAMP	26	UPDATIME	S	
最終変更者	CHAR	8	UPDATEBY	S	
短い記述	VARCHAR	250	SHRTDESC	O	
長い記述	LONG VARCHAR	32700	LONGDESC	O	
処置	VARCHAR	254	ACTIONS	O	
図表の表題	VARCHAR	254	TITLE	O	
図表の発行日 付	CHAR	26	RPRTDATE	O	
図表の表示形 式	VARCHAR	80	RPRTFRMT	O	
図表の表示要 件	VARCHAR	254	DPPRESNT	O	
図表の所有者	VARCHAR	80	OWNER	O	
図表のファイ ル名	VARCHAR	254	FILENAME	R	1
図表のクラス またはタイプ	VARCHAR	80	TYPE	R	2

表 98. 図表オブジェクト・タイプのプロパティ (続き)

プロパティ名	データ・タイプ	サイズ	プロパティ短縮名	値フラグ	UI 順序
データにアクセスするための URL	VARCHAR	254	URL	O	

注: S = 情報カタログ・マネージャー生成、R = 必須、O = オプション

文書 (Documents): ブックまたは技術文書を表現します。これらの資料は、印刷または電子形式で、ローカルに、あるいはライブラリー内で見つけることができます。

このオブジェクト・タイプを定義するためのタグ言語は、`¥VWSWIN¥DGWIN¥TYPES` ディレクトリーのファイル `FLGNYDOC.TYP` に入っています。

このオブジェクト・タイプ用の情報カタログ・マネージャー DB2 ストレージ表の名前は `XXX.DOCS` です。表名の詳細については、176ページの『MDIS マッピング』を参照してください。

表99 では、文書オブジェクト・タイプのプロパティに関する情報を提供します。

表 99. 文書オブジェクト・タイプのプロパティ

プロパティ名	データ・タイプ	サイズ	プロパティ短縮名	値フラグ	UI 順序
オブジェクト・タイプ ID	CHAR	6	OBJTYPID	S	
インスタンス ID	CHAR	10	INSTIDNT	S	
名前	VARCHAR	80	NAME	R	
最終変更日時	TIMESTAMP	26	UPDATIME	S	
最終変更者	CHAR	8	UPDATEBY	S	
短い記述	VARCHAR	250	SHRTDESC	O	
長い記述	LONG VARCHAR	32700	LONGDESC	O	
処置	VARCHAR	254	ACTIONS	O	
文書の作成者	VARCHAR	80	AUTHOR	R	1

表 99. 文書オブジェクト・タイプのプロパティ (続き)

プロパティ名	データ・タイプ	サイズ	プロパティ短縮名	値フラグ	UI 順序
文書の位置	VARCHAR	254	LOCATION	R	2
文書のファイル名	VARCHAR	254	FILENAME	R	3
データにアクセスするための URL	VARCHAR	254	URL	O	

注: S = 情報カタログ・マネージャー生成、R = 必須、O = オプション

イメージまたはグラフィックス (Images or graphics): グラフィック・イメージ (ビットマップなど) を表現します。

このオブジェクト・タイプを定義するためのタグ言語は、`¥VWSWIN¥DGWIN¥TYPES` ディレクトリーのファイル `FLGNYIMA.TYP` に入っています。

このオブジェクト・タイプ用の情報カタログ・マネージャー DB2 ストレージ表の名前は `XXX.IMAGES` です。表名の詳細については、176ページの『MDIS マッピング』を参照してください。

表100 では、イメージまたはグラフィックス・オブジェクト・タイプのプロパティに関する情報を提供します。

表 100. イメージまたはグラフィックス・オブジェクト・タイプのプロパティ

プロパティ名	データ・タイプ	サイズ	プロパティ短縮名	値フラグ	UI 順序
オブジェクト・タイプ ID	CHAR	6	OBJTYPID	S	
インスタンス ID	CHAR	10	INSTIDNT	S	
名前	VARCHAR	80	NAME	R	
最終変更日時	TIMESTAMP	26	UPDATIME	S	
最終変更者	CHAR	8	UPDATEBY	S	
短い記述	VARCHAR	250	SHRTDESC	O	
長い記述	LONG VARCHAR	32700	LONGDESC	O	

表 100. イメージまたはグラフィックス・オブジェクト・タイプのプロパティ (続き)

プロパティ名	データ・タイプ	サイズ	プロパティ短縮名	値フラグ	UI 順序
処置	VARCHAR	254	ACTIONS	O	
イメージのファイル名	VARCHAR	254	FILENAME	R	1
イメージのクラスまたはタイプ	VARCHAR	80	TYPE	R	2
データにアクセスするための URL	VARCHAR	254	URL	O	

注: S = 情報カタログ・マネージャー生成、R = 必須、O = オプション

インターネット文書 (Internet documents): Web サイトや、インターネット上のその他の関連文書を表現します。

このオブジェクト・タイプを定義するためのタグ言語は、
`¥VWSWIN¥DGWIN¥TYPES` ディレクトリーのファイル `FLGNYINT.TYP` に入っています。

このオブジェクト・タイプ用の情報カタログ・マネージャー DB2 ストレージ表の名前は `XXX.INTERNET` です。表名の詳細については、176ページの『MDIS マッピング』を参照してください。

表101 では、インターネット文書オブジェクト・タイプのプロパティに関する情報を提供します。

表 101. インターネット文書オブジェクト・タイプのプロパティ

プロパティ名	データ・タイプ	サイズ	プロパティ短縮名	値フラグ	UI 順序
オブジェクト・タイプ ID	CHAR	6	OBJTYPID	S	
インスタンス ID	CHAR	10	INSTIDNT	S	
名前	VARCHAR	80	NAME	R	
最終変更日時	TIMESTAMP	26	UPDATIME	S	
最終変更者	CHAR	8	UPDATEBY	S	

インターネット文書

表 101. インターネット文書オブジェクト・タイプのプロパティ (続き)

プロパティ名	データ・タイプ	サイズ	プロパティ短縮名	値フラグ	UUI 順序
短い記述	VARCHAR	250	SHRTDESC	O	
長い記述	LONG VARCHAR	32700	LONGDESC	O	
処置	VARCHAR	254	ACTIONS	O	
データにアクセスするための URL	VARCHAR	254	URL	R	1
ローカル・ファイル名	VARCHAR	254	FILENAME	R	2
インターネット文書のクラスまたはタイプ	VARCHAR	80	TYPE	O	

注: S = 情報カタログ・マネージャー生成、R = 必須、O = オプション

Lotus Approach 照会 (Lotus Approach queries): 組織のデータに対して使用可能な Lotus Approach 照会を表現します。

このオブジェクト・タイプを定義するためのタグ言語は、`¥VWSWIN¥DGWIN¥TYPES` ディレクトリーのファイル `FLGNYAPR.TYP` に入っています。

このオブジェクト・タイプ用の情報カタログ・マネージャー DB2 ストレージ表の名前は `XXX.APPROACH` です。表名の詳細については、176ページの『MDIS マッピング』を参照してください。

表102 では、Lotus Approach 照会オブジェクト・タイプのプロパティに関する情報を提供します。

表 102. Lotus Approach 照会オブジェクト・タイプのプロパティ

プロパティ名	データ・タイプ	サイズ	プロパティ短縮名	値フラグ	UUI 順序
オブジェクト・タイプ ID	CHAR	6	OBJTYPID	S	

表 102. Lotus Approach 照会オブジェクト・タイプのプロパティ (続き)

プロパティ名	データ・タイプ	サイズ	プロパティ短縮名	値フラグ	UI 順序
インスタンス ID	CHAR	10	INSTIDNT	S	
名前	VARCHAR	80	NAME	R	
最終変更日時	TIMESTAMP	26	UPDATIME	S	
最終変更者	CHAR	8	UPDATEBY	S	
短い記述	VARCHAR	250	SHRTDESC	O	
長い記述	LONG VARCHAR	32700	LONGDESC	O	
処置	VARCHAR	254	ACTIONS	O	
Approach オブジェクトのファイル名	VARCHAR	254	FILENAME	R	1
データにアクセスするための URL	VARCHAR	254	URL	O	

注: S = 情報カタログ・マネージャー生成、R = 必須、O = オプション

プレゼンテーション (Presentations): 印刷された、または電子的なプレゼンテーションを表現します。これらのプレゼンテーションには、製品、顧客、品質、および状況のプレゼンテーションが含まれます。

このオブジェクト・タイプを定義するためのタグ言語は、
¥VWSWIN¥DGWIN¥TYPES ディレクトリーのファイル FLGNYPRE.TYP に入っています。

このオブジェクト・タイプ用の情報カタログ・マネージャー DB2 ストレージ表の名前は XXX.PRESENT です。表名の詳細については、176ページの『MDIS マッピング』を参照してください。

222ページの表103 では、プレゼンテーション・オブジェクト・タイプのプロパティに関する情報を提供します。

プレゼンテーション

表 103. プレゼンテーション・オブジェクト・タイプのプロパティ

プロパティ名	データ・タイプ	サイズ	プロパティ短縮名	値フラグ	UUI 順序
オブジェクト・タイプ ID	CHAR	6	OBJTYPID	S	
インスタンス ID	CHAR	10	INSTIDNT	S	
名前	VARCHAR	80	NAME	R	
最終変更日時	TIMESTAMP	26	UPDATIME	S	
最終変更者	CHAR	8	UPDATEBY	S	
短い記述	VARCHAR	250	SHRTDESC	O	
長い記述	LONG VARCHAR	32700	LONGDESC	O	
処置	VARCHAR	254	ACTIONS	O	
プレゼンテーションのファイル名	VARCHAR	254	FILENAME	R	1
プレゼンテーションのクラスまたはタイプ	VARCHAR	80	TYPE	O	
プレゼンテーション・スクリプト	VARCHAR	254	SCRIPTFN	O	
データにアクセスするための URL	VARCHAR	254	URL	O	

注: S = 情報カタログ・マネージャー生成、R = 必須、O = オプション

スプレッドシート (Spreadsheets): デスクトップのスプレッドシート (たとえば、Lotus 1-2-3 または Microsoft Excel スプレッドシート) を表現します。

このオブジェクト・タイプを定義するためのタグ言語は、`¥VWSWIN¥DGWIN¥TYPES` ディレクトリーのファイル `FLGNYSSTH.TYP` に入っています。

このオブジェクト・タイプ用の情報カタログ・マネージャー DB2 ストレージ表の名前は XXX.SSHEETS です。表名の詳細については、176ページの『MDIS マッピング』を参照してください。

表104 では、スプレッドシート・オブジェクト・タイプのプロパティーに関する情報を提供します。

表 104. スプレッドシート・オブジェクト・タイプのプロパティー

プロパティー名	データ・タイプ	サイズ	プロパティー短縮名	値フラグ	UUI 順序
オブジェクト・タイプ ID	CHAR	6	OBJTYPID	S	
インスタンス ID	CHAR	10	INSTIDNT	S	
名前	VARCHAR	80	NAME	R	
最終変更日時	TIMESTAMP	26	UPDATIME	S	
最終変更者	CHAR	8	UPDATEBY	S	
短い記述	VARCHAR	250	SHRTDESC	O	
長い記述	LONG VARCHAR	32700	LONGDESC	O	
処置	VARCHAR	254	ACTIONS	O	
スプレッドシートのクラスまたはタイプ	VARCHAR	80	TYPE	O	
スプレッドシートのファイル名	VARCHAR	254	FILENAME	R	1
スプレッドシートのビットマップ <キャプチャー> ファイル名	VARCHAR	254	BITMAP	O	
データにアクセスするための URL	VARCHAR	254	URL	O	

注: S = 情報カタログ・マネージャー生成、R = 必須、O = オプション

テキスト・ベース・レポート

テキスト・ベース・レポート (Text-based reports): 印刷された、または電子的なレポートを表現します。

このオブジェクト・タイプを定義するためのタグ言語は、
¥VWSWIN¥DGWIN¥TYPES ディレクトリーのファイル FLGNYREP.TYP に入
っています。

このオブジェクト・タイプ用の情報カタログ・マネージャー DB2 ストレージ
表の名前は XXX.REPORTS です。表名の詳細については、176ページの
『MDIS マッピング』を参照してください。

表105 では、テキスト・ベース・レポート・オブジェクト・タイプのプロパテ
ィーに関する情報を提供します。

表 105. テキスト・ベース・レポート・オブジェクト・タイプのプロパティ

プロパティ 名	データ・ タイプ	サイズ	プロパティ 短縮名	値フラグ	UI 順序
オブジェ クト・タイプ ID	CHAR	6	OBJTYPID	S	
インスタ ンスID	CHAR	10	INSTIDNT	S	
名前	VARCHAR	80	NAME	R	
最終変更 日時	TIMESTAMP	26	UPDATIME	S	
最終変更 者	CHAR	8	UPDATEBY	S	
短い記 述	VARCHAR	250	SHRTDESC	O	
長い記 述	LONG VARCHAR	32700	LONGDESC	O	
処置	VARCHAR	254	ACTIONS	O	
レポート の表題	VARCHAR	254	TITLE	R	
レポート の発行 日付	CHAR	26	RPRTDATE	O	
レポート の表示 形式	VARCHAR	80	RPRTFRMT	O	
レポート の表示 要件	VARCHAR	254	DPPRESNT	O	
レポート の所有 者	VARCHAR	80	OWNER	O	

表 105. テキスト・ベース・レポート・オブジェクト・タイプのプロパティ (続き)

プロパティ名	データ・タイプ	サイズ	プロパティ短縮名	値フラグ	UII 順序
レポートのファイル名	VARCHAR	254	FILENAME	R	1
レポートのクラスまたはタイプ	VARCHAR	80	TYPE	R	2
データにアクセスするための URL	VARCHAR	254	URL	O	

注: S = 情報カタログ・マネージャー生成、R = 必須、O = オプション

ビデオ・クリップ (Video clips): ビデオ情報を含むファイルを表現します。これらのオブジェクトでは、電子的な (AVI ファイル)、または媒体に記録された (たとえば、ビデオ・テープまたはレーザー・ディスク) ビデオ情報を表現することができます。

このオブジェクト・タイプを定義するためのタグ言語は、
`¥VWSWIN¥DGWIN¥TYPES` ディレクトリーのファイル `FLGNYVID.TYP` に入っています。

このオブジェクト・タイプ用の情報カタログ・マネージャー DB2 ストレージ表の名前は `XXX.VIDEO` です。表名の詳細については、176ページの『MDIS マッピング』を参照してください。

表106 では、ビデオ・クリップ・オブジェクト・タイプのプロパティに関する情報を提供します。

表 106. ビデオ・クリップ・オブジェクト・タイプのプロパティ

プロパティ名	データ・タイプ	サイズ	プロパティ短縮名	値フラグ	UII 順序
オブジェクト・タイプ ID	CHAR	6	OBJTYPID	S	
インスタンス ID	CHAR	10	INSTIDNT	S	
名前	VARCHAR	80	NAME	R	
最終変更日時	TIMESTAMP	26	UPDATIME	S	

ビデオ・クリップ

表 106. ビデオ・クリップ・オブジェクト・タイプのプロパティ (続き)

プロパティ名	データ・タイプ	サイズ	プロパティ短縮名	値フラグ	UUI 順序
最終変更者	CHAR	8	UPDATEBY	S	
短い記述	VARCHAR	250	SHRTDESC	O	
長い記述	LONG VARCHAR	32700	LONGDESC	O	
処置	VARCHAR	254	ACTIONS	O	
ビデオ・クリップのファイル名	VARCHAR	254	FILENAME	R	1
ビデオ・クリップのクラスまたはタイプ	VARCHAR	80	TYPE	R	2
データにアクセスするための URL	VARCHAR	254	URL	O	

注: S = 情報カタログ・マネージャー生成、R = 必須、O = オプション

コンタクト区分

コンタクト区分には、『連絡先 (People to contact)』オブジェクト・タイプが含まれます。

連絡先 (People to contact): 連絡先オブジェクト・タイプは、情報カタログ内のオブジェクトについて責任がある人またはグループを識別します。

このオブジェクト・タイプを定義するためのタグ言語は、
¥VWSWIN¥DGWIN¥TYPES ディレクトリーのファイル FLGNYCON.TYP に入っています。

このオブジェクト・タイプ用の情報カタログ・マネージャー DB2 ストレージ表の名前は XXX.CONTACT です。表名の詳細については、176ページの『MDIS マッピング』を参照してください。

227ページの表107 では、連絡先オブジェクト・タイプのプロパティに関する情報を提供します。

表 107. 連絡先オブジェクト・タイプのプロパティ

プロパティ名	データ・タイプ	サイズ	プロパティ短縮名	値フラグ	UUI 順序
オブジェクト・タイプ ID	CHAR	6	OBJTYPID	S	
インスタンス ID	CHAR	10	INSTIDNT	S	
名前	VARCHAR	80	NAME	R	1
最終変更日時	TIMESTAMP	26	UPDATIME	S	
最終変更者	CHAR	8	UPDATEBY	S	
短い記述	VARCHAR	250	SHRTDESC	O	
長い記述	LONG VARCHAR	32700	LONGDESC	O	
処置	VARCHAR	254	ACTIONS	O	
連絡先の責任	VARCHAR	254	RESPONSE	R	2
連絡先の電話番号	CHAR	15	PHONE	R	
連絡先の電子メール・アドレス	VARCHAR	254	EMAIL	R	
連絡先のピクチャー・ファイル名	VARCHAR	254	FILENAME	O	
データにアクセスするための URL	VARCHAR	254	URL	O	

注: S = 情報カタログ・マネージャー生成、R = 必須、O = オプション

辞書区分

辞書区分には、『用語集エントリー (Glossary entries)』オブジェクト・タイプが含まれます。

用語集エントリー (Glossary entries): 用語集エントリー・オブジェクト・タイプは、情報カタログ内で使用される用語の定義を表現します。このオブジェクト・タイプのプロパティは、228ページの表108 で示されています。

用語集エントリー

このオブジェクト・タイプを定義するためのタグ言語は、
 ¥VWSWIN¥DGWIN¥TYPES ディレクトリーのファイル FLGNYGLO.TYP に入
 っています。

このオブジェクト・タイプ用の情報カタログ・マネージャー DB2 ストレージ
 表の名前は XXX.GLOSSARY です。表名の詳細については、176ページの
 『MDIS マッピング』を参照してください。

表108 では、用語集エントリー・オブジェクト・タイプのプロパティーに関する
 情報を提供します。

表 108. 用語集エントリー・オブジェクト・タイプのプロパティー

プロパティー 名	データ・ タイプ	サイズ	プロパティー 短縮名	値フラグ	UII 順序
オブジェク ト・タイプ ID	CHAR	6	OBJTYPID	S	
インスタンス ID	CHAR	10	INSTIDNT	S	
名前	VARCHAR	80	NAME	R	1
最終変更日時	TIMESTAMP	26	UPDATIME	S	
最終変更者	CHAR	8	UPDATEBY	S	
短い記述	VARCHAR	250	SHRTDESC	O	
長い記述	LONG VARCHAR	32700	LONGDESC	O	
処置	VARCHAR	254	ACTIONS	O	
キーワード	VARCHAR	254	KEYWORD	O	
用語集定義の コンテキスト	CHAR	32	CONTEXT	O	
用語集定義を 含むファイル 名	VARCHAR	254	FILENAME	O	
用語集のクラ スまたはタイ プ	VARCHAR	80	TYPE	O	
データにアク セスするため の URL	VARCHAR	254	URL	O	

注: S = 情報カタログ・マネージャー生成、R = 必須、O = オプション

サポート区分

サポート区分には、次のオブジェクト・タイプが含まれます。

- 『情報カタログ・マネージャー・ニュース (Information Catalog Manager news)』
- 230ページの『オンライン・ニュース・サービス (Online news services)』
- 231ページの『オンライン資料 (Online publications)』

情報カタログ・マネージャー・ニュース (Information Catalog Manager news): 情報カタログ・マネージャー・ニュース・オブジェクト・タイプには、情報カタログへの変更に関する情報が含まれます。

このオブジェクト・タイプを定義するためのタグ言語は、
¥VWSWIN¥DGWIN¥TYPES ディレクトリーのファイル FLGNYDGN.TYP に入っています。

このオブジェクト・タイプ用の情報カタログ・マネージャー DB2 ストレージ表の名前は XXX.DGNEWS です。表名の詳細については、176ページの『MDIS マッピング』を参照してください。

表109 では、情報カタログ・マネージャー・ニュース・オブジェクト・タイプのプロパティーに関する情報を提供します。

表 109. 情報カタログ・マネージャー・ニュース・オブジェクト・タイプのプロパティー

プロパティー 名	データ・ タイプ	サイズ	プロパティー 短縮名	値フラグ	UI 順序
オブジェク ト・タイプ ID	CHAR	6	OBJTYPID	S	
インスタンス ID	CHAR	10	INSTIDNT	S	
名前	VARCHAR	80	NAME	R	1
最終変更日時	TIMESTAMP	26	UPDATIME	S	
最終変更者	CHAR	8	UPDATEBY	S	
短い記述	VARCHAR	250	SHRTDESC	O	
長い記述	LONG VARCHAR	32700	LONGDESC	O	
処置	VARCHAR	254	ACTIONS	O	

表 109. 情報カタログ・マネージャー・ニュース・オブジェクト・タイプのプロパティ
(続き)

プロパティ 名	データ・ タイプ	サイズ	プロパティ 短縮名	値フラグ	UI 順序
ニュース項目 の日付	CHAR	26	NEWSDATE	R	
ニュース・ク リップ	VARCHAR	254	ABSTRACT	R	
全ニュース項 目	LONG VARCHAR	32700	NEWSITEM	O	
データにアク セスするため の URL	VARCHAR	254	URL	O	

注: S = 情報カタログ・マネージャー生成、R = 必須、O = オプション

オンライン・ニュース・サービス (Online news services): オンライン・ニュース・サービス・オブジェクト・タイプは、オンラインでアクセスできるニュースおよび情報サービスを表現します。

このオブジェクト・タイプを定義するためのタグ言語は、
¥VWSWIN¥DGWIN¥TYPES ディレクトリーのファイル FLGNYOLN.TYP に入
っています。

このオブジェクト・タイプ用の情報カタログ・マネージャー DB2 ストレージ
表の名前は XXX.OLNEWS です。表名の詳細については、176ページの『MDIS
マッピング』を参照してください。

表110 では、オンライン・ニュース・サービス・オブジェクト・タイプのプロ
パティに関する情報を提供します。

表 110. オンライン・ニュース・サービス・オブジェクト・タイプのプロパティ

プロパティ 名	データ・ タイプ	サイズ	プロパティ 短縮名	値フラグ	UI 順序
オブジェク ト・タイプ ID	CHAR	6	OBJTYPID	S	
インスタンス ID	CHAR	10	INSTIDNT	S	
名前	VARCHAR	80	NAME	R	1

表 110. オンライン・ニュース・サービス・オブジェクト・タイプのプロパティ (続き)

プロパティ名	データ・タイプ	サイズ	プロパティ短縮名	値フラグ	UUI 順序
最終変更日時	TIMESTAMP	26	UPDATIME	S	
最終変更者	CHAR	8	UPDATEBY	S	
短い記述	VARCHAR	250	SHRTDESC	O	
長い記述	LONG VARCHAR	32700	LONGDESC	O	
処置	VARCHAR	254	ACTIONS	O	
サービス名	VARCHAR	254	SERVNAME	R	
データにアクセスするための URL	VARCHAR	254	URL	O	

注: S = 情報カタログ・マネージャー生成、R = 必須、O = オプション

オンライン資料 (Online publications): オンライン資料オブジェクト・タイプは、オンライン・サービスによってアクセスできる資料およびその他の文書を表現します。

このオブジェクト・タイプを定義するためのタグ言語は、
`¥VWSWIN¥DGWIN¥TYPES` ディレクトリーのファイル `FLGNYOLP.TYP` に入っています。

このオブジェクト・タイプ用の情報カタログ・マネージャー DB2 ストレージ表の名前は `XXX.OLPUBS` です。表名の詳細については、176ページの『MDIS マッピング』を参照してください。

表111 では、オンライン資料オブジェクト・タイプのプロパティに関する情報を提供します。

表 111. オンライン資料オブジェクト・タイプのプロパティ

プロパティ名	データ・タイプ	サイズ	プロパティ短縮名	値フラグ	UUI 順序
オブジェクト・タイプ ID	CHAR	6	OBJTYPID	S	
インスタンス ID	CHAR	10	INSTIDNT	S	
名前	VARCHAR	80	NAME	R	1

表 111. オンライン資料オブジェクト・タイプのプロパティ (続き)

プロパティ名	データ・タイプ	サイズ	プロパティ短縮名	値フラグ	UUI 順序
最終変更日時	TIMESTAMP	26	UPDATIME	S	
最終変更者	CHAR	8	UPDATEBY	S	
短い記述	VARCHAR	250	SHRTDESC	O	
長い記述	LONG VARCHAR	32700	LONGDESC	O	
処置	VARCHAR	254	ACTIONS	O	
サービス名	VARCHAR	254	SERVNAME	R	
データにアクセスするための URL	VARCHAR	254	URL	O	

注: S = 情報カタログ・マネージャー生成、R = 必須、O = オプション

プログラム区分

プログラム区分には、プログラム・オブジェクト・タイプのみが含まれます。

プログラム・オブジェクト・タイプは、情報カタログの作成時に作成されます。サンプル情報カタログ DGV5SAMP では、プログラム・オブジェクト・タイプに、情報カタログ・マネージャー・オブジェクトから呼び出すことができるプログラム (*Programs that can be invoked from Information Catalog Manager objects*) という名前が付けられています。

情報カタログ・マネージャー・オブジェクトから呼び出すことができるプログラム (*Programs that can be invoked from Information Catalog Manager objects*): 特定のオブジェクト・タイプを処理することができるアプリケーションを定義するために使用されます。

このオブジェクト・タイプ用の情報カタログ・マネージャー DB2 ストレージ表の名前は XXX.PROGRAMS です。表名の詳細については、176ページの『MDIS マッピング』を参照してください。

233ページの表112 では、情報カタログ・マネージャー・オブジェクトから呼び出すことができるプログラム・オブジェクト・タイプのプロパティに関する情報を提供します。

情報カタログ・マネージャー・オブジェクトから呼び出すことができるプログラム

表 112. "情報カタログ・マネージャー・オブジェクトから呼び出すことができるプログラム" オブジェクト・タイプのプロパティ

プロパティ名 ¹	データ・タイプ	サイズ	プロパティ短縮名	値フラグ ²	UI 順序
オブジェクト・タイプ ID	CHAR	6	OBJTYPID	S	
インスタンス ID	CHAR	10	INSTIDNT	S	
名前	VARCHAR	80	NAME	R	
最終変更日時	TIMESTAMP	26	UPDATIME	S	
最終変更者	CHAR	8	UPDATEBY	S	
クラス	CHAR	25	UIICLASS	R	1
修飾子 1	VARCHAR	48	UIQUAL1	R	2
修飾子 2	VARCHAR	48	UIQUAL2	R	3
修飾子 3	VARCHAR	48	UIQUAL3	R	4
ID	VARCHAR	70	UIIDENT	R	5
このプログラムによって処理されるオブジェクト・タイプ	CHAR	8	HANDLES	O	
開始時に呼び出すコマンド	VARCHAR	250	STARTCMD	R	
パラメーター・リスト	VARCHAR	1800	PARMLIST	O	
短い記述	VARCHAR	250	SHRTDESC	O	

注:

1. 必須プロパティの説明と例は、*情報カタログ・マネージャー 管理の手引き* に記載されています。
2. S = 情報カタログ・マネージャー生成、R = 必須、O = オプション

接続区分

接続区分には、234ページの『コメント (Comments)』オブジェクト・タイプのみが含まれます。

コメント・オブジェクト・タイプは、情報カタログの作成時に作成されます。

コメント

コメント (Comments): 情報カタログ内の他のオブジェクトにコメントを付けるために使用されます。

表113 では、コメント・オブジェクト・タイプのプロパティに関する情報を提供します。

表 113. コメント・オブジェクト・タイプのプロパティ

プロパティ名	データ・タイプ	サイズ	プロパティ短縮名	値フラグ	UI 順序
オブジェクト・タイプ ID	CHAR	6	OBJTYPID	S	
インスタンス ID	CHAR	10	INSTIDNT	S	
名前	VARCHAR	80	NAME	R	1
最終変更日時	TIMESTAMP	26	UPDATIME	S	
最終変更者	CHAR	8	UPDATEBY	S	
作成者	CHAR	8	CREATOR	R	2
作成のタイムスタンプ	TIMESTAMP	26	CREATSTP	R	3
状況	CHAR	80	STATUS	O	
処置	VARCHAR	250	ACTIONS	O	
エクストラ情報	VARCHAR	80	EXTRA	O	
長い記述	LONG VARCHAR	32700	LONGDESC	O	

注: S = 情報カタログ・マネージャー生成、R = 必須、O = オプション

事前定義プログラム・オブジェクト

表114 で示されているプログラム・オブジェクト・タイプは、サンプル情報カタログで提供されるものです。この表では、プログラムの立ち上げ時に情報カタログ・マネージャー・プログラム・オブジェクトへの関連付けに使用するプロパティ名も示されています。

表 114. サンプル情報カタログ内の事前定義の汎用プログラム・オブジェクト

情報のタイプ	プログラム名	オブジェクト・タイプ	プロパティ名
マルチメディア・ファイル	Microsoft Media Player	オーディオ・クリップ	オーディオ・クリップのファイル名

表 114. サンプル情報カタログ内の事前定義の汎用プログラム・オブジェクト (続き)

情報のタイプ	プログラム名	オブジェクト・タイプ	プロパティ名
	Microsoft Media Player	ビジネス・サブジェクト・エリア	ファイル名
	Microsoft Media Player	プレゼンテーション	プレゼンテーションのファイル名
	Microsoft Media Player	ビデオ・クリップ	ビデオ・クリップのファイル名
ビットマップ・ファイル	Microsoft Paint	イメージまたはグラフィックス	グラフィックのファイル名
	Microsoft Paint	連絡先	連絡先のピクチャー・ファイル名
スプレッドシート・ファイル	Microsoft Excel	スプレッドシート	スプレッドシートのファイル名
	Microsoft Paint	スプレッドシート	スプレッドシートのファイル名
	Lotus 1-2-3	スプレッドシート	スプレッドシートのファイル名
Web ページ	Netscape Navigator	オンライン・ニュース	データにアクセスするための URL
	Netscape Navigator	オンライン資料	データにアクセスするための URL
	Microsoft Internet Explorer	インターネット文書	データにアクセスするための URL
	Microsoft Internet Explorer	オンライン・ニュース	データにアクセスするための URL
	Microsoft Internet Explorer	オンライン資料	データにアクセスするための URL

表115 では、情報カタログ・マネージャーに統合されているアプリケーションの提供元である特定の IBM ビジネス・パートナーがリストされています。この表の情報は、234ページの表114 のものと類似しています。

表 115. サンプル情報カタログ内の事前定義プログラム・オブジェクト -- IBM ビジネス・パートナー

情報のタイプ	プログラム名	オブジェクト・タイプ	プロパティ名
Lotus	Approach	Lotus Approach	Approach オブジェクトのファイル名

事前定義プログラム・オブジェクト

表 115. サンプル情報カタログ内の事前定義プログラム・オブジェクト -- IBM ビジネス・パートナー (続き)

情報のタイプ	プログラム名	オブジェクト・タイプ	プロパティ名
	Freelance Graphics	プレゼンテーション	プレゼンテーション・オブジェクトのファイル名
Hyperion	Lotus 1-2-3 (Essbase Spreadsheet アドイン付き)	スプレッドシート	スプレッドシートのファイル名
	Microsoft Excel (Essbase Spreadsheet アドイン付き)	スプレッドシート	スプレッドシートのファイル名
Brio	Brio Query	テキスト・ベース・レポート	レポートのファイル名
	Netscape Navigator (Brio.Insights プラグインと一緒に使用される)	テキスト・ベース・レポート	データにアクセスするための URL
	Microsoft Internet Explorer (Brio.Insights プラグインと一緒に使用される)	テキスト・ベース・レポート	データにアクセスするための URL
BusinessObjects	BusinessObjects	データベース	なし
	BusinessObjects	テキスト・ベース・レポート	レポートのファイル名
	Microsoft Excel (BusinessQuery アドインと一緒に使用される)	スプレッドシート	スプレッドシートのファイル名
	Microsoft Internet Explorer (WebIntelligence Java [®] アプレットにアクセスするために使用される)	インターネット文書	データにアクセスするための URL
	Netscape Navigator (WebIntelligence Java アプレットにアクセスするために使用される)	インターネット文書	データにアクセスするための URL
Cognos	PowerPlay	テキスト・ベース・レポート	レポートのファイル名
	Impromptu	テキスト・ベース・レポート	レポートのファイル名

表 115. サンプル情報カタログ内の事前定義プログラム・オブジェクト -- IBM ビジネス・パートナー (続き)

情報のタイプ	プログラム名	オブジェクト・タイプ	プロパティ名
	Microsoft Internet Explorer (Impromptu Web Query と一緒に使用される)	インターネット文書	データにアクセスするための URL
	Netscape Navigator (Impromptu Web Query と一緒に使用される)	インターネット文書	データにアクセスするための URL
	Netscape Navigator (PowerPlay の Web 版 HTML ページにアクセスするために使用される)	インターネット文書	データにアクセスするための URL
Wired for OLAP	Wired for OLAP ビュー	テキスト・ベース・レポート	デフォルト・ユーザー・ログインおよび始動オプションの構成
	Netscape 内の Wired for OLAP ホーム・ページ	テキスト・ベース・レポート	デフォルト・ユーザー・ログインおよび始動オプションの構成
	Microsoft Internet Explorer 内の Wired for OLAP ホーム・ページ	テキスト・ベース・レポート	デフォルト・ユーザー・ログインおよび始動オプションの構成
Seagate	Crystal Reports	テキスト・ベース・レポート	レポートのファイル名
Microsoft Access	Microsoft Access	データベース	
Microsoft PowerPoint	Microsoft PowerPoint ビューアー	テキスト・ベース・レポート	レポートのファイル名
	Netscape 内の Microsoft PowerPoint ビューアー	テキスト・ベース・レポート	データにアクセスするための URL
	Microsoft Internet Explorer 内の Microsoft PowerPoint ビューアー	テキスト・ベース・レポート	データにアクセスするための URL

第9章 タグ言語

情報カタログ・マネージャーのタグ言語を使用すると、記述的データを情報カタログにインポートできるように、フォーマットを設定することができます。タグ言語は、情報カタログ・マネージャーに、それがインポートする記述的データに対して行うべきことを指示します。また、情報カタログ・マネージャーは、ユーザーが情報カタログをバックアップしたり、ある情報カタログから別の情報カタログにデータを転送したりすることができるように、記述的データをタグ言語ファイルにエクスポートします。

タグ言語を使用して記述的データのフォーマットを設定することによって、ある情報カタログから別の情報カタログに記述的データを移動したり、情報カタログ・マネージャー・オブジェクト・タイプおよびオブジェクトを定義したりすることができます。また、独自の抽出プログラムを作成して、他のソース（リレーショナル・データベース・カタログなど）から記述的データを抽出し、情報カタログにインポートすることもできます。表116 では、タグ言語のタグと、これらのタグによって実行されるアクションを示します。

表 116. 情報カタログ・マネージャー・タグ

作業	タグ名	参照先
ディスクットのシーケンスを記録する	DISKCNL	257 ページ
入力データに対して行われるアクションを識別する	ACTION.OBJINST	243 ページ
	ACTION.OBJTYPE	249 ページ
	ACTION.RELATION	253 ページ
情報カタログにデータを記述する	OBJECT	264 ページ
	PROPERTY	271 ページ
	INSTANCE	258 ページ
	RELTYPE	275 ページ
変更がコミットされた時点と、チェックポイントが発生した場所を識別する	COMMIT	255 ページ
ユーザーのコメントを識別する	COMMENT	255 ページ
データのフォーマットを設定する	NL	264 ページ
	TAB	277 ページ

タグ言語ファイルの作成に関する規則

この節で説明されている規則は、すべてのタグ言語ファイルに適用されます。

- それぞれのタグ名は、コロンで始まり、ピリオドで終わらなければなりません。コロンとタグ名の間、あるいはタグ名とピリオドの間にスペースを入れないでください。たとえば、次のようにします。

```
:ACTION.OBJINST.
```

タグ名は、239ページの表116 でリストされているタグ名のいずれかでなければなりません。

- COMMENT、NL、または TAB を除くすべてのタグでは、少なくとも 1 つのキーワードを組み込んでください。
- キーワードとその値は、次のように書いてください。

```
keyword(value)
```

- キーワードは、どのような順序で指定しても構いません。唯一の例外は、INSTANCE タグの SOURCEKEY キーワードが最初のキーワードでなければならないことです。
- キーワードを分離するには、ブランクを使用します。
- キーワードの値は、括弧で囲んでください。値に括弧が含まれる場合は、括弧をアポストロフィの対で囲みます。たとえば、次のようにします。

```
keyword(value>('1'))
```

- PROPERTY または INSTANCE タグで、OBJTYPID、INSTIDNT、UPDATIME、または UPDATEBY をプロパティ短縮名 (*short_name*) として使用しないでください。
- 以下のプロパティ名は、情報カタログ・マネージャーによって予約されています。

```
OBJTYPID  
INSTIDNT  
NAME  
UPDATIME  
UPDATEBY
```

ACTION.OBJTYPE(ADD) または ACTION.OBJTYPE(MERGE) の使用時に、NAME をオブジェクト・タイプの UII プロパティとして識別する場合は、次のように、PROPERTY タグで NAME を *short_name* として指定することができます。

```
:PROPERTY.SHRTNAME(NAME) UIISEQ(1)
```

情報カタログ・マネージャーがタグ言語ファイルを読み取る方法

タグ言語ファイルをコーディングする際には、情報カタログ・マネージャーがタグ言語ファイルを読み取る方法を考慮に入れてください。

- 情報カタログ・マネージャーは、タグ言語ファイル全体を連続したデータ・ストリームとして読み取ります。
- 情報カタログ・マネージャーは、X'20' より下の 16 進値を持つ文字（プロパティ値の中で指定されたタブおよび改行文字タグを除く）を制御文字として扱い、その文字を無視します。
- 情報カタログ・マネージャーが、あるタグが完結したとみなすのは、タグ言語ファイルの中で次のタグを検出したときです。
- タグおよびキーワードは、各国語に翻訳されません。
- 2 バイト文字セット (DBCS) サポートで使用可能なのは、表117 のキーワードの値のみです。

表 117. DBCS が使用可能なキーワード値

タグ名	キーワード	可変値
OBJECT	EXTNAME	<i>ext_name</i>
	ICWFILE	<i>Windows_ICON_file_name</i>
PROPERTY	EXTNAME	<i>ext_name</i>
COMMIT	CHKPID	<i>checkpoint_id</i>
INSTANCE	<i>UUI_short_name</i>	<i>UUI_property_value</i>
	または	または
	<i>short_name</i>	<i>property_value</i>

すべてのユーザー定義プロパティ値では、DBCS 文字を使用することができます。

- 情報カタログ・マネージャーは、表118 で示されているキーワード値の中でのみ DBCS ブランクを受け入れます。タグ言語ファイル内のほかの場所に DBCS ブランクがあると、エラーが発生する可能性があります。

表 118. DBCS ブランク文字が使用可能なキーワード値

タグ名	キーワード
ACTION	OBJTYPE
	OBJINST
	RELATION
OBJECT	すべてのキーワード
PROPERTY	すべてのキーワード
RELTYPE	すべてのキーワード

表 118. DBCS ブランク文字が使用可能なキーワード値 (続き)

タグ名	キーワード
COMMIT	CHKPID
INSTANCE	<i>UII_short_name</i> または <i>short_name</i>

情報カタログ・マネージャーの記述的データについて有効なデータ・タイプ

表119 では、情報カタログ・マネージャーの記述的データについて有効なデータ・タイプを示します。

表 119. 情報カタログ・マネージャーの記述的データについて有効なデータ・タイプ

データ・タイプ	説明
CHAR	固定長文字ストリング (1 ~ 254 バイト)。 値がプロパティについて定義されているデータ長よりも短い場合は、値の右側を末尾ブランクで埋めてください。
TIMESTAMP	次の形式の 26 文字のタイム・スタンプ: <i>yyyy-mm-dd-hh.mm.ss.nnnnnn</i>
LONG VARCHAR	長い可変長文字ストリング (1 ~ 32 700 バイト)。 データ・タイプが LONG VARCHAR のプロパティを UII プロパティとして指定することはできません。
VARCHAR	可変長文字ストリング (1 ~ 4 000 バイト)。

情報カタログ・マネージャーは、要求を妥当性検査し、受け入れる前に、自動的に可変値から末尾ブランクを除去し、値の長さを適宜に調整します。

必須の値が指定されていないか、または全体がブランクの場合、情報カタログ・マネージャーは、表120 で示されている値を挿入します。

表 120. 情報カタログ・マネージャーによって提供される値

データ・タイプ	提供される値
CHAR	先頭文字としての該当しないシンボルと、定義された長さを埋めるための末尾ブランク。
TIMESTAMP	9999-12-31-24.00.00.000000
LONG VARCHAR	該当しないシンボル
VARCHAR	該当しないシンボル

タグ言語の構文図の読み方

タグおよびキーワードは、テキストで示されているとおりに正確にコーディングしてください。タグおよびキーワードは、次のように示されています。

:tagname.keyword() keyword()

変数を置換できる有効な値は、キーワード・リストの中で説明されています。値は、次のように示されています: *variable*

タグの説明で、キーワードまたは値のそれぞれの対の中にある縦線は、対のいずれかをタグに組み込まなければならないことを意味します。たとえば、PROPERTY タグの構文には、NULLS キーワードの値として NULLS(Y|N) が組み込まれています。NULLS(Y) または NULLS(N) のいずれかをコーディングしなければなりません。

ACTION.OBJINST

ACTION タグに続くタグで記述されているオブジェクトに対して実行されるアクションを識別します。

コンテキスト

ACTION.OBJINST は、情報カタログ・マネージャー・オブジェクトの作成、削除、または保守に使用されます。

ACTION.OBJINST の後には、アクションの対象となるオブジェクトを定義する 1 つまたは複数の OBJECT および INSTANCE タグが続きます。

構文

:ACTION.OBJINST(*option*)

オプション

ACTION.OBJINST では、以下のオプションが有効です。

ADD
DELETE
DELETE_TREE_ALL
DELETE_TREE_REL
MERGE
UPDATE

ACTION.OBJINST

ACTION.OBJINST(ADD)

オブジェクトを追加します。

コンテキスト:

```
:ACTION.OBJINST(ADD)
:OBJECT.TYPE()
:INSTANCE.short_name()
:INSTANCE.short_name()

:OBJECT.TYPE()
:INSTANCE.short_name()
:INSTANCE.short_name()
```

図 33. オブジェクトの追加時の ACTION.OBJINST タグの使用

規則:

- オブジェクトはすでに存在していなければなりません。
- ACTION.OBJINST(ADD) タグの後には、OBJECT と INSTANCE の両方のタグを続けなければなりません。
 - OBJECT タグは、新規のオブジェクトのオブジェクト・タイプを識別します。
 - INSTANCE タグは、新規のオブジェクトのプロパティ値を指定します。
- 1 つの OBJECT の後には、1 つまたは複数の INSTANCE タグを続けることができます (オブジェクトが同じオブジェクト・タイプのものであれば)。
- ACTION.OBJINST(ADD) の後に OBJECT タグ (INSTANCE タグを伴う) の複数のセットを続けると、追加する別々のオブジェクト・タイプのオブジェクトを記述することができます。

ACTION.OBJINST(DELETE)

オブジェクトを削除します。

コンテキスト:

```
:ACTION.OBJINST(DELETE)
:OBJECT.TYPE()
:INSTANCE.SOURCEKEY(UUI_short_name()...)
```

```
:OBJECT.TYPE()
:INSTANCE.SOURCEKEY(UUI_short_name()...)
```

図 34. オブジェクトの削除時の ACTION.OBJINST タグの使用

規則:

- 指定するオブジェクトはすでに存在していなければなりません。
- ACTION.OBJINST(DELETE) タグの後には、OBJECT と INSTANCE の両方のタグを続けなければなりません。
 - OBJECT タグは、削除されるオブジェクトのオブジェクト・タイプを識別します。
 - INSTANCE タグは、削除されるオブジェクトの UUI プロパティ値を指定します。
- 1 つの OBJECT の後には、1 つまたは複数の INSTANCE タグを続けることができます (オブジェクトが同じオブジェクト・タイプのものであれば)。
- ACTION.OBJINST(DELETE) の後に OBJECT タグ (INSTANCE タグを伴う) の複数のセットを続けると、削除する別々のオブジェクト・タイプのオブジェクトを記述することができます。
- 削除するオブジェクトがグループ化オブジェクトである場合は、それに別のオブジェクトが含まれていてはなりません。別のオブジェクトが含まれていると、削除は失敗します。この場合は、代わりに ACTION.OBJINST(DELETE_TREE_ALL) を使用してください。

ACTION.OBJINST(DELETE_TREE_ALL)

グループ化区分のオブジェクト、それに付加されているすべてのコメント・オブジェクト、およびそれが参加しているすべての ATTACHMENT、CONTACT、および LINK 関係を削除します。さらに、グループ化区分のオブジェクトに含まれるすべてのオブジェクト、それらに付加されているすべてのコメント・オブジェクト、およびそれらが参加しているすべての ATTACHMENT、CONTACT、および LINK 関係を削除します。

ACTION.OBJINST

コンテキスト:

```
:ACTION.OBJINST(DELETE_TREE_ALL)
:OBJECT.TYPE()
:INSTANCE.SOURCEKEY(UII_short_name()...)

:OBJECT.TYPE()
:INSTANCE.SOURCEKEY(UII_short_name()...)
```

図 35. グループ化区分のオブジェクトと、それに含まれるオブジェクトの削除時の ACTION.OBJINST タグの使用

規則:

- 指定するオブジェクトはすでに存在していなければならず、グループ化区分のオブジェクトでなければなりません。
- ACTION.OBJINST(DELETE_TREE_ALL) タグの後には、OBJECT と INSTANCE の両方のタグを続けなければなりません。
 - OBJECT タグは、削除されるオブジェクトのオブジェクト・タイプを識別します。
 - INSTANCE タグは、削除されるオブジェクトの UII プロパティ値を指定します。
- 1 つの OBJECT の後には、1 つまたは複数の INSTANCE タグを続けることができます (オブジェクトが同じオブジェクト・タイプのものであれば)。
- ACTION.OBJINST(DELETE_TREE_ALL) の後に OBJECT タグ (INSTANCE タグを伴う) の複数のセットを続けると、削除する別々のオブジェクト・タイプのオブジェクトを記述することができます。

ACTION.OBJINST(DELETE_TREE_REL)

グループ化区分のオブジェクト、それに付加されているすべてのコメント・オブジェクト、およびそれが参加しているすべての ATTACHMENT、CONTACT、CONTAIN、および LINK 関係を削除します。

コンテキスト:

```
:ACTION.OBJINST(DELETE_TREE_REL)
:OBJECT.TYPE()
:INSTANCE.SOURCEKEY(UII_short_name()...)

:OBJECT.TYPE()
:INSTANCE.SOURCEKEY(UII_short_name()...)
```

図 36. グループ化区分のオブジェクトと関係の削除時の ACTION.OBJINST タグの使用

規則:

- 指定するオブジェクトはすでに存在していなければならず、グループ化区分のオブジェクトでなければなりません。
- ACTION.OBJINST(DELETE_TREE_REL) タグの後には、OBJECT と INSTANCE の両方のタグを続けなければなりません。
 - OBJECT タグは、削除されるオブジェクトのオブジェクト・タイプを識別します。
 - INSTANCE タグは、削除されるオブジェクトの UII プロパティ値を指定します。
- 1 つの OBJECT の後には、1 つまたは複数の INSTANCE タグを続けることができます (オブジェクトが同じオブジェクト・タイプのものであれば)。
- ACTION.OBJINST(DELETE_TREE_REL) の後に OBJECT タグ (INSTANCE タグを伴う) の複数のセットを続けると、削除する別々のオブジェクト・タイプのオブジェクトを記述することができます。

ACTION.OBJINST(MERGE)

入力オブジェクトが存在するかどうかを調べるために、情報カタログ内で入力オブジェクトの UII を検索します。

オブジェクトが存在する場合、情報カタログ・マネージャーは情報カタログ内のオブジェクトのプロパティ値を更新します。オブジェクトが存在しない場合、情報カタログ・マネージャーは新規のオブジェクトを作成します。

コンテキスト:

```
:ACTION.OBJTYPE(MERGE)
:OBJECT.TYPE() CATEGORY() EXTNAME() PHYNAME() ICOFILE() ICWFILE()
:PROPERTY.EXTNAME() DT() DL() SHRTNAME() NULLS() UIISEQ()
```

```
:ACTION.OBJINST(MERGE)
:OBJECT.TYPE()
:INSTANCE.short_name()
```

図 37. オブジェクトの組み合わせ時の ACTION.OBJINST タグの使用

規則:

- オブジェクトが存在する場合、情報カタログ・マネージャーは情報カタログ内のオブジェクトのプロパティ値を更新します。オブジェクトが存在しない場合、情報カタログ・マネージャーは新規のオブジェクトを作成します。
- ACTION.OBJINST(MERGE) タグの後には、OBJECT と INSTANCE の両方のタグを続けなければなりません。

ACTION.OBJINST

- OBJECT タグは、組み合わせられるオブジェクトのオブジェクト・タイプを識別します。
- INSTANCE タグは、組み合わせられるオブジェクトのプロパティ値を指定します。
- タグ言語ファイルの中で、ある特定のオブジェクト・タイプに対する ACTION.OBJTYPE(MERGE) タグは、同じオブジェクト・タイプに対する ACTION.OBJINST(MERGE) タグよりも先に指定しなければなりません。これにより、インポート先の情報カタログにそのオブジェクト・タイプが存在するように保証されるため、情報カタログ・マネージャーがオブジェクトを追加または更新 (組み合わせ) できるようになります。

新規のプログラムまたは接続オブジェクト・タイプを作成することはできないため、プログラムまたは接続区分に属するオブジェクト・タイプに対して ACTION.OBJTYPE(MERGE) を使用することはできません。ただし、ACTION.OBJTYPE(MERGE) を最初に指定せずに、プログラム・オブジェクトに対して ACTION.OBJINST(MERGE) を使用することはできます。

ACTION.OBJINST(UPDATE)

オブジェクトの値を更新します。

コンテキスト:

```
:ACTION.OBJINST(UPDATE)
:OBJECT.TYPE()
:INSTANCE.SOURCEKEY(UII_short_name()...) short_name()
```

図 38. オブジェクトの更新時の ACTION.OBJINST タグの使用

規則:

- 指定するオブジェクトはすでに存在していなければなりません。
- ACTION.OBJINST(UPDATE) タグの後には、OBJECT と INSTANCE の両方のタグを続けなければなりません。
 - OBJECT タグは、更新されるオブジェクトのオブジェクト・タイプを識別します。
 - INSTANCE タグは、更新されるオブジェクトを識別する UII プロパティ値と、更新されるプロパティ値を指定します。

INSTANCE タグで指定したプロパティ値のみが更新されます。

ACTION.OBJTYPE

ACTION.OBJTYPE タグに続くタグで記述されているオブジェクトに対して実行されるアクションを識別します。

コンテキスト

ACTION.OBJTYPE は、情報カタログ・マネージャー・オブジェクト・タイプの作成、削除、または保守に使用されます。

ACTION.OBJTYPE の後には、アクションの対象となるオブジェクト・タイプを定義する 1 つまたは複数の OBJECT および PROPERTY タグが続きます。

構文

```
:ACTION.OBJTYPE(option)
```

オプション

ACTION.OBJTYPE では、以下のオプションが有効です。

```
ADD
APPEND
DELETE
DELETE_EXT
MERGE
UPDATE
```

ACTION.OBJTYPE(ADD)

オブジェクト・タイプを作成します。

コンテキスト:

```
:ACTION.OBJTYPE(ADD)
:OBJECT.TYPE() CATEGORY() EXTNAME() PHYNAME() ICOFILE() ICWFILE()
:PROPERTY.EXTNAME() DT() DL() SHRTNAME() NULLS() UUISEQ()
:PROPERTY.EXTNAME() DT() DL() SHRTNAME() NULLS() UUISEQ()
```

図 39. オブジェクト・タイプの追加時の ACTION.OBJTYPE タグの使用

規則:

- オブジェクト・タイプが存在してはなりません。

ACTION.OBJTYPE

- ACTION.OBJTYPE(ADD) タグの直後には、OBJECT タグとそれに関連する PROPERTY タグを続けなければなりません。
 - OBJECT タグは、新規のオブジェクト・タイプの属性を定義します。
 - PROPERTY タグは、新規のオブジェクト・タイプに属するプロパティを定義します。情報カタログ・マネージャーは、それぞれのオブジェクト・タイプについて、次の必須プロパティを自動的に定義します。

OBJTYPID
INSTIDNT
NAME
UPDATIME
UPDATEBY

- プログラムまたは接続区分に属するオブジェクト・タイプを追加することはできません。

ACTION.OBJTYPE(APPEND)

既存のオブジェクト・タイプにプロパティを追加します。

コンテキスト:

```
:ACTION.OBJTYPE(APPEND)  
:OBJECT.TYPE()  
:PROPERTY.EXTNAME() DT() DL() SHRTNAME() NULLS() UIUSEQ()
```

図 40. オブジェクト・タイプへのプロパティの追加時の ACTION.OBJTYPE タグの使用

規則:

- オブジェクト・タイプは存在していなければなりません。
- 追加するプロパティが存在してはなりません。
- プロパティに 0 (デフォルト) 以外の UIUSEQ 値を割り当てないください。追加するプロパティは、NULLS(Y) のオプションでなければならず、UIUI の一部にすることはできません。
- ACTION.OBJTYPE(APPEND) タグの直後には、OBJECT タグとそれに関連する PROPERTY タグを続けなければなりません。
 - OBJECT タグは、追加先のオブジェクト・タイプを識別します。
 - それぞれの PROPERTY タグは、追加されるプロパティを定義します。
- 接続区分に属するオブジェクト・タイプへの追加を行うことはできません。

ACTION.OBJTYPE(DELETE)

オブジェクト・タイプを削除します。

コンテキスト:

```
:ACTION.OBJTYPE(DELETE)
:OBJECT.TYPE()
```

図 41. オブジェクト・タイプの削除時の ACTION.OBJTYPE タグの使用

規則:

- オブジェクト・タイプは存在していなければなりません。そのオブジェクト・タイプのオブジェクトが存在してはなりません。
- ACTION.OBJTYPE(DELETE) タグの後には、1 つまたは複数の OBJECT タグを続けなければなりません。それぞれの OBJECT タグは、削除されるオブジェクト・タイプを識別します。
- プログラムまたは接続区分に属するオブジェクト・タイプを削除することはできません。

ACTION.OBJTYPE(DELETE_EXT)

オブジェクト・タイプと、そのオブジェクト・タイプのオブジェクトを削除します。

コンテキスト:

```
:ACTION.OBJTYPE(DELETE_EXT)
:OBJECT.TYPE()
```

図 42. オブジェクト・タイプと、そのタイプのすべてのオブジェクトの削除時の ACTION.OBJTYPE タグの使用

規則:

- オブジェクト・タイプは存在していなければなりません。
- オブジェクトに、異なるオブジェクト・タイプのオブジェクトを含めることはできません。
- ACTION.OBJTYPE(DELETE_EXT) タグの後には、1 つまたは複数の OBJECT タグを続けなければなりません。それぞれの OBJECT タグは、削除されるオブジェクト・タイプを識別します。
- プログラムまたは接続区分に属するオブジェクト・タイプを削除することはできません。

ACTION.OBJTYPE(MERGE)

入力オブジェクト・タイプ名が存在するかどうかを調べるために、情報カタログをチェックします。

ACTION.OBJTYPE

オブジェクト・タイプが存在する場合、情報カタログ・マネージャーは、入力オブジェクト・タイプのプロパティを保管済みのオブジェクト・タイプのプロパティと比較します。プロパティが一致した場合、オブジェクト・タイプは同一として扱われ、一致しない場合、入力オブジェクト・タイプは無効です。

オブジェクト・タイプが存在しない場合、情報カタログ・マネージャーは新規のオブジェクト・タイプを作成します。

コンテキスト:

```
:ACTION.OBJTYPE(MERGE)
:OBJECT.TYPE() CATEGORY() EXTNAME() PHYNAME() ICOFILE() ICWFILE()
:PROPERTY.EXTNAME() DT() DL() SHRTNAME() NULLS() UUISEQ()
```

```
:ACTION.OBJINST(MERGE)
:OBJECT.TYPE()
:INSTANCE.short_name()
```

図 43. オブジェクト・タイプの組み合わせ時の ACTION.OBJTYPE タグの使用

規則:

- ACTION.OBJTYPE(MERGE) タグの直後には、OBJECT タグとそれに関連する PROPERTY タグを続けなければなりません。
 - OBJECT タグは、組み合わせられるオブジェクト・タイプを識別します。
 - それぞれの PROPERTY タグは、そのオブジェクト・タイプに属するプロパティを定義します。
- オブジェクトを組み合わせるには、事前にオブジェクト・タイプを組み合わせることによって、ターゲット情報カタログに有効なオブジェクト・タイプが存在するように保証しなければなりません。したがって、タグ言語ファイルの中で、ACTION.OBJTYPE(MERGE) タグは ACTION.OBJINST(MERGE) タグの前に指定しなければなりません。
- プログラムまたは接続区分に属するオブジェクト・タイプを組み合わせることはできません。

ACTION.OBJTYPE(UPDATE)

オブジェクト・タイプの外部名および ICON ファイル情報を変更します。

コンテキスト:

```
:ACTION.OBJTYPE(UPDATE)
:OBJECT.TYPE() EXTNAME() ICOFILE() ICWFILE()
```

図44. オブジェクト・タイプの更新時の ACTION.OBJTYPE タグの使用

規則:

- オブジェクト・タイプはすでに存在していなければなりません。
- ACTION タグの後には、1 つまたは複数の OBJECT タグを続けなければなりません。

ACTION.RELATION

ACTION.RELATION タグに続くタグで記述されている関係に対して実行されるアクションを識別します。

コンテキスト

ACTION.RELATION は、情報カタログの関係を作成または削除するために使用されます。

ACTION.RELATION の後には、アクションの対象となる関係を定義する 1 つまたは複数の RELTYPE および INSTANCE タグが続きます。

構文

```
:ACTION.RELATION(option)
```

オプション

ACTION.RELATION では、以下のオプションが有効です。

```
ADD
DELETE
```

ACTION.RELATION(ADD)

ATTACHMENT、CONTACT、CONTAIN、または LINK 関係を定義します。

ACTION.RELATION

コンテキスト:

```
:ACTION.RELATION(ADD)  
:RELTYPE.TYPE() SOURCETYPE() TARGETTYPE()  
:INSTANCE.SOURCEKEY(UII_short_name()...) TARGETKEY(UII_short_name()...)
```

図 45. 関係の追加時の ACTION.RELATION タグの使用

規則:

- 指定した関係が存在しない場合は、関係が追加されます。指定した関係が存在する場合は、情報カタログ・マネージャーによって通知メッセージが書き込まれ、処理が継続されます。
- ACTION.RELATION(ADD) タグの直後には、RELTYPE タグと 1 つまたは複数の INSTANCE タグを続けなければなりません。
 - RELTYPE タグは、追加される関係のタイプを定義し、関連付けられるオブジェクトのオブジェクト・タイプを指定します。
 - それぞれの INSTANCE タグは、関連付けられる 2 つのオブジェクトを識別する UII プロパティ値を指定します。

ACTION.RELATION(DELETE)

関係を削除します。

コンテキスト:

```
:ACTION.RELATION(DELETE)  
:RELTYPE.TYPE() SOURCETYPE() TARGETTYPE()  
:INSTANCE.SOURCEKEY(UII_short_name()...) TARGETKEY(UII_short_name()...)
```

図 46. 関係の削除時の ACTION.RELATION タグの使用

規則:

- 関係が存在する場合は、それが削除されます。それ以外の場合は、情報カタログ・マネージャーによって通知メッセージが書き込まれ、処理が継続されます。
- ACTION.RELATION(DELETE) タグの直後には、RELTYPE タグと 1 つまたは複数の INSTANCE タグを続けなければなりません。
 - RELTYPE タグは、削除される関係のタイプを定義し、関連付けられたオブジェクトのオブジェクト・タイプを指定します。
 - それぞれの INSTANCE タグは、2 つの関連付けられたオブジェクトを識別する UII プロパティ値を指定します。

COMMENT

タグ言語ファイル内のコメントを識別します。このタグは、ファイル内の完結したタグ指定の間に置いてください。

情報カタログ・マネージャーは、タグ言語ファイルのインポート時にコメントを無視します。

構文

```
:COMMENT.your comments
```

```
:COMMENT.This is the text of a comment.
```

図 47. COMMENT タグの例

規則

- COMMENT タグは、別のタグとそのキーワードの間、あるいはキーワードの間に置くことはできません。
- それぞれのタグは、ファイルの終わり、または次の有効なタグの始まりで終わるため、コメント・テキストに情報カタログ・マネージャー・タグ (たとえば、:ACTION.) を含めることはできません。

COMMIT

コミット・ポイントを識別します。情報カタログ・マネージャーに、データベースに対する現行の変更をコミットするよう要求します。

情報カタログ・マネージャーは、タグ言語ファイルのインポート中にエラーを検出すると、変更が最後にコミットされた時点から情報カタログに対して行われたすべての変更をロールバックします。

情報カタログ・マネージャーのタグ言語ファイルをより効率的にインポートするために、COMMIT を一定の間隔で組み込んでください。

オブジェクト・タイプ、オブジェクトのセット、および関係のセットを定義または削除する前後に COMMIT チェックポイントを組み込むと、記述的データの保全性を維持するのに役立ちます。

COMMIT

定期的な COMMIT チェックポイントにより、情報カタログ・マネージャーが情報カタログをロールバックする際に取り消される変更の数が制限されます。

頻繁に COMMIT チェックポイントを組み込むと、タグ言語ファイルにエラーがある場合に、エコー・ファイルがより読みやすくなります。COMMIT タグが正常に処理されると、情報カタログ・マネージャーは、COMMIT タグの前に処理されたタグのエコー・ファイルをクリアします。そのため、エコー・ファイルには、コミットされていない変更を記述するタグのみが含まれます。

コンテキスト

このタグは、1 つまたは複数の完結したアクション指定 (ACTION、OBJECT、RELTYPE、および INSTANCE タグのセット) の後に置いてください。

構文

```
:COMMIT.CHKPID(checkpt_id)
```

```
:COMMIT.CHKPID(Added_relationships)
```

図 48. COMMIT タグの例

キーワード

CHKPID

必須キーワード。

checkpt_id

情報カタログ・マネージャーが COMMIT タグの処理時に保管する ID。

COMMIT タグが正常に処理された後でタグ言語ファイルのインポートが失敗した場合は、最後のチェックポイントからタグ言語ファイルの残りの部分をインポートしなければなりません。このオプションは、インポート・ファンクションで使用することができます。情報カタログ・マネージャーは、保管された *checkpt_id* を使用して、適切な COMMIT タグを見付けます。

checkpt_id の値は、それぞれのタグ言語ファイルの中で固有でなければなりません。そうでなければ、再開処理の結果は予測不能です。

checkpt_id の最大長は 26 文字です。

ckpt_id は、大文字小文字が区別されません。

規則

COMMIT タグは、データに整合性があるときに指定してください。

ターゲット情報カタログのトランザクション・ログが満杯になるのを防ぐために、タグ言語ファイル内で COMMIT タグを一定の間隔で指定してください。

同じタグ言語ファイル内の追加のデータを処理する必要がある場合は、COMMIT タグの後に ACTION タグを続けなければなりません。

DISKCNTL

タグ言語ファイルが 1 つまたは複数のディスクセットに保管されている場合に、ディスクセットのシーケンス番号を識別します。

コンテキスト

タグ言語ファイルが 1 つまたは複数のディスクセットに保管されている場合、DISKCNTL は各ディスクセットの最初のタグになります。

構文

```
:DISKCNTL.SEQUENCE(nn, + | -)
```

```
:DISKCNTL.SEQUENCE(01,+)
```

図 49. 一連のディスクセットの始まりを表す DISKCNTL タグの例

キーワード

SEQUENCE

必須キーワード。

nn 一連のディスクセットの番号を示す 1 桁または 2 桁の番号。

一連のディスクセットの最初の番号は、1 または 01 でなければなりません。この値は、後続のディスクセットについて 1 ずつ増加します。3 つのディスクセットのセットを表す番号は、1、2、3、あるいは 01、02、03 です。

DISKCNTRL

- + このディスクレットの後に、タグ言語ファイルを含む追加のディスクレットが続きます。
- タグ言語ファイルを含む最後または唯一のディスクレット。

規則

このタグを指定する場合は、各タグ言語ファイルの最初のタグにしなければなりません。このタグが欠落しており、タグ言語ファイルがディスクレットに入っている場合、インポート・プログラムは、タグ言語ファイルが 1 つのディスクレットに含まれているものとみなします。

タグ言語ファイルがハード・ディスクに保管されている場合、このタグは適用不能です。このタグが存在する場合は、無視されます。

INSTANCE

アクションの対象となるオブジェクトまたは関係を定義または識別します。

コンテキスト

このタグは、次のタグの後に続けることが必要です。

:ACTION.OBJINST INSTANCE タグは、OBJECT タグの後に続きます。

:ACTION.RELATION INSTANCE タグは、RELTYPE タグの後に続きます。

構文

INSTANCE タグには、ACTION タグの形式に応じて、4 つの形式があります。

ACTION.OBJINST(ADD) または ACTION.OBJINST(MERGE)

オブジェクトの追加または組み合わせ

:INSTANCE.short_name (property_value) . . .

コンテキスト:

```
:ACTION.OBJINST(ADD)
:OBJECT.TYPE()
:INSTANCE.short_name()
```

図 50. オブジェクトの追加時の INSTANCE タグの使用

```
:ACTION.OBJINST(MERGE)
:OBJECT.TYPE()
:INSTANCE.short_name()
:short_name()
:short_name()
```

図 51. オブジェクトの組み合わせ時の INSTANCE タグの使用

キーワード:*short_name*

それぞれのプロパティをその 8 文字の短縮名によって識別します。この値は、大文字小文字が区別されないため、大文字または小文字を使用して指定することができます。INSTANCE タグに複数の短縮名を関連付ける場合は、図 51 で示されているように、1 つの INSTANCE タグのみを使用し、それに短縮名を続けてください。

property_value

特定のオブジェクトについてのプロパティの値を指定します。この値は、大文字小文字が区別されます。

規則:

- オブジェクトの追加時:
 - すべての UI 値、NAME プロパティの値、および必須として定義されているその他のすべてのプロパティの値を指定しなければなりません。
 - 追加される値がないプロパティは INSTANCE タグから省略することができます。ただし、省略したプロパティが CHAR、VARCHAR、または LONG VARCHAR データ・タイプの必須プロパティである場合には、該当しないシンボルが生成され、情報カタログに保管されます。省略した必須プロパティのデータ・タイプが TIMESTAMP である場合は、情報カタログ・マネージャーによって値 9999-12-31-24.00.00.000000 が生成され、保管されます。
- オブジェクトの組み合わせ時:

INSTANCE

- すべての UUI 値を指定して、突き合わせられるオブジェクトが識別できるように保証しなければなりません。
- 追加または更新される値がないプロパティは省略することができます。ただし、定義したオブジェクトが存在せず、省略したプロパティが必須である場合には、該当しないシンボルが生成され、情報カタログに保管されます。

ACTION.OBJINST(DELETE)、ACTION.OBJINST(DELETE_TREE_ALL)、または ACTION.OBJINST(DELETE_TREE_REL)

オブジェクトの削除

```
:INSTANCE.SOURCEKEY(UUI_short_name (UUI_property_value) . . . )
```

コンテキスト:

```
:ACTION.OBJINST(DELETE)
:OBJECT.TYPE()
:INSTANCE.SOURCEKEY(UUI_short_name()...)
```

図 52. オブジェクトの削除時の *INSTANCE* タグの使用

```
:ACTION.OBJINST(DELETE_TREE_ALL)
:OBJECT.TYPE()
:INSTANCE.SOURCEKEY(UUI_short_name()...)
```

図 53. グループ化区分のオブジェクトと、それに含まれるオブジェクトの削除時の *INSTANCE* タグの使用

```
:ACTION.OBJINST(DELETE_TREE_REL)
:OBJECT.TYPE()
:INSTANCE.SOURCEKEY(UUI_short_name()...)
```

図 54. グループ化区分のオブジェクトと関係の削除時の *INSTANCE* タグの使用

キーワード:

SOURCEKEY

ある特定のオブジェクトを識別する UUI プロパティ値。

SOURCEKEY は、*INSTANCE* タグの最初のキーワードでなければなりません。

UUI_short_name

UUI 名をその 8 文字の短縮名によって識別します。

UUI_short_name(UUI_property_value) のすべての組み合わせを指定してください。 *UUI_short_name* は、大文字小文字が区別されないため、大文字または小文字を使用して指定することができます。

UUI_property_value

特定のオブジェクトについての UUI プロパティの値を指定します。この値は、大文字小文字が区別されます。

規則: オブジェクト・タイプの UUI プロパティとして定義されているプロパティごとに、1 つの *UUI_short_name(value)* の組み合わせを指定しなければなりません。それぞれのオブジェクト・タイプには、UUI プロパティとして定義された 1 つまたは複数のプロパティがあります。これらのプロパティは、情報カタログの中でオブジェクトを固有に識別します。

ACTION.OBJINST(UPDATE)

オブジェクトについてのプロパティ値の更新

```
:INSTANCE.SOURCEKEY(UUI_short_name (UUI_property_value) . . .)
                    short_name (property_value) . . .
```

コンテキスト:

```
:ACTION.OBJINST(UPDATE)
:OBJECT.TYPE()
:INSTANCE.SOURCEKEY(UUI_short_name()...) short_name()
```

図 55. オブジェクトの更新時の *INSTANCE* タグの使用

キーワード:**SOURCEKEY**

ある特定のオブジェクトを識別する UUI プロパティ値。

SOURCEKEY は、INSTANCE タグの最初のキーワードでなければなりません。

UUI_short_name

UUI プロパティをその 8 文字の短縮名によって識別します。

UUI_short_name は、大文字小文字が区別されないため、大文字または小文字を使用して指定することができます。

INSTANCE

UUI_property_value

この値は、大文字小文字が区別されます。 *UUI_short_name* とともに、特定のオブジェクトについての UUI プロパティの値を指定します。

short_name

更新されるプロパティをその 8 文字の短縮名によって識別します。

short_name は、大文字小文字が区別されないため、大文字または小文字を使用して指定することができます。

次のプロパティは更新不能であるため、これらの短縮名を指定することはできません: OBJTYPID、INSTIDNT、UPDATIME、UPDATEBY。

property_value

short_name とともに、特定のオブジェクトについてのプロパティの新しい値を指定します。この値は、大文字小文字が区別されます。

規則: オブジェクト・タイプの UUI プロパティとして定義されているプロパティごとに、1 つの *UUI_short_name(value)* の組み合わせを指定しなければなりません。それぞれのオブジェクト・タイプには、UUI プロパティとして定義された 1 つまたは複数のプロパティがあります。これらのプロパティは、情報カタログの中でオブジェクトを固有に識別します。

プロパティ値を指定すると、情報カタログ内でその値が更新されます。プロパティ値を指定しないと、その値は更新されません。

ACTION.RELATION(ADD) または ACTION.RELATION(DELETE)

関係の追加または削除

```
:INSTANCE.SOURCEKEY(UUI_short_name (UUI_property_value)...)  
      TARGETKEY(UUI_short_name (UUI_property_value)...)
```

コンテキスト:

```
:ACTION.RELATION(ADD)  
:RELTYPE.TYPE() SOURCETYPE() TARGETTYPE()  
:INSTANCE.SOURCEKEY(UUI_short_name()...) TARGETKEY(UUI_short_name()...)
```

図 56. 関係の追加時の INSTANCE タグの使用


```
:ACTION.RELATION(DELETE)
:RELTYPETYPE() SOURCETYPE() TARGETTYPE()
:INSTANCE.SOURCEKEY(UII_short_name())... TARGETKEY(UII_short_name())...
```

図 57. 関係の削除時の INSTANCE タグの使用

キーワード:

SOURCEKEY

関係における最初のオブジェクトを識別する UII プロパティ値。

関係:	SOURCEKEY が識別するオブジェクト:
包含	グループ化区分のオブジェクト
接点	接点の対象であるオブジェクト
付加	コメントの対象であるオブジェクト
リンク	リンクするオブジェクトのいずれか一方

SOURCEKEY は、INSTANCE タグの最初のキーワードでなければなりません。

TARGETKEY

関係における 2 番目のオブジェクトを識別する UII プロパティ値。

関係:	TARGETKEY が識別するオブジェクト:
包含	基礎的区分のオブジェクト
接点	コンタクト区分のオブジェクト
付加	接続区分のオブジェクト
リンク	リンクするオブジェクトのいずれか一方

TARGETKEY は、INSTANCE タグの 2 番目のキーワードでなければなりません。

UII_short_name

UII 名をその 8 文字の短縮名によって識別します。この値は、大文字小文字が区別されないため、大文字または小文字を使用して指定することができます。

UII_property_value

特定のオブジェクトについての UII プロパティの値を指定します。この値は、大文字小文字が区別されます。

INSTANCE

規則: それぞれのオブジェクトについて、オブジェクト・タイプの UUI プロパティーとして定義されているプロパティーごとに、1 つの *UUI_short_name(value)* の組み合わせを指定しなければなりません。それぞれのオブジェクト・タイプには、UUI プロパティーとして定義された 1 つまたは複数のプロパティーがあります。これらのプロパティーは、情報カタログの中でオブジェクトを固有に識別します。

図58 で示されているように、*UUI_short_name(value)* と *short_name(value)* のそれぞれの対は、空白で分離しなければなりません。

```
:INSTANCE.SOURCEKEY(UUIname1(value1) UUIname2(value2)) sname3(value3) sname4(value4)
```

図 58. 複数の短縮名を伴う *INSTANCE* タグの例

ある値を囲む括弧の間に組み込まれた先行空白はその値の一部となり、末尾空白は除去されます。情報カタログ・マネージャーは、値の長さが有効かどうかを判別する際に、これらの空白をデータ長の一部として数えます。値全体が許容される最大長よりも長くなるような、余分な先行空白または末尾空白を組み込むと、エラーが発生します。

NL

プロパティー値の中で改行を指定します。

情報カタログ・マネージャーは、非 UUI プロパティー値の中で指定された NL タグのみを読み取り、それ以外はすべて無視します。

構文

```
:NL.
```

規則

NL タグは、*INSTANCE* タグの *property_values* の指定の中でのみ使用してください。

OBJECT

オブジェクト・タイプの属性を定義するか、またはオブジェクト・タイプを識別します。

コンテキスト

このタグは、次のタグの直後に続けることが必要です。

```
ACTION.OBJTYPE
ACTION.OBJINST
```

構文

```
:OBJECT.TYPE(type) CATEGORY(category) EXTNAME(ext_name)
  PHYNAME(table_name) ICOFILE( )
  ICWFILE(Windows_ICON_file_name)
```

OBJECT タグの前に指定する ACTION タグのタイプによって、OBJECT タグの必須または有効キーワードは異なります。

ACTION.OBJTYPE(ADD) または ACTION.OBJTYPE(MERGE)

オブジェクト・タイプの追加または組み合わせ

コンテキスト:

```
:ACTION.OBJTYPE(ADD)
:OBJECT.TYPE() CATEGORY() EXTNAME() PHYNAME() ICOFILE() ICWFILE()
:PROPERTY.EXTNAME() DT() DL() SHRTNAME() NULLS() UUISEQ()
```

図 59. オブジェクト・タイプの追加時の OBJECT タグの使用

```
:ACTION.OBJTYPE(MERGE)
:OBJECT.TYPE() CATEGORY() EXTNAME() PHYNAME() ICOFILE() ICWFILE()
:PROPERTY.EXTNAME() DT() DL() SHRTNAME() NULLS() UUISEQ()
```

図 60. オブジェクト・タイプの組み合わせ時の OBJECT タグの使用

キーワード:

TYPE

オブジェクト・タイプの名前を指定します。

必須キーワード。

type

特定のオブジェクト・タイプを表す短縮名を定義または識別します。

type の値は、あるオブジェクト・タイプを含むすべての関連する情報カタログにわたって、そのオブジェクト・タイプに固有でなければなりません。これにより、このオブジェクト・タイプのオブジェクトが、関連する

OBJECT

情報カタログの間で共用できるようになります。 *type* の値がすでに存在する場合、それは検索引き数として使用されます。

値の最大長は 8 文字です。値は大文字で保管されます。この値は、文字 A ~ Z、@、#、または \$ で始まり、これらの文字と、0 ~ 9 および _ を含むことができます。先行空白または組み込み空白は許可されません。

オブジェクト・タイプを作成した後で、*type* の値を変更することはできません。

CATEGORY

このオブジェクト・タイプが属する区分を指定します。

必須キーワード。

category

情報カタログ・マネージャー・オブジェクトの区分を指定します。この値は、次のいずれかにすることができます。

GROUPING
ELEMENTAL
SUPPORT
CONTACT
DICTIONARY

PROGRAM または ATTACHMENT を新規のオブジェクト・タイプの区分として指定することはできません。

オブジェクト・タイプが定義された後で、このキーワードの情報を変更することはできません。

EXTNAME

オブジェクト・タイプの長い記述名を指定します。必須キーワード。

ext_name

オブジェクト・タイプの長い記述名を指定します。 *ext_name* の最大長は 80 文字です。

この名前は、関連する情報カタログの中で固有でなければなりません。

ext_name の値は大文字小文字混合で保管されます。

オブジェクト・タイプが定義された後で、このキーワードの情報を変更することができます。

PHYNAME

このオブジェクト・タイプに関する情報を含むデータベース表の作成時に使用する名前を指定します。

オプション・キーワード。

table_name

オブジェクト・タイプの情報を含むデータベース表の作成時に使用する名前を指定します。

名前の最大長は、情報カタログ・マネージャーのインストール時に定義されます。 *table_name* の値は、情報カタログの中で固有でなければならず、SQL 予約語を含むことはできません。

table_name は、デフォルトでは、**TYPE** キーワードに指定された *type* です。この値は、大文字小文字が区別されないため、大文字または小文字を使用して指定することができます。

この値は、文字 A ~ Z、@、#、または \$ で始まり、これらの文字と、0 ~ 9 および _ を含むことができます。先行ブランクまたは組み込みブランクは許可されません。この値は、情報カタログに関して使用されるデータベースを表す SQL 予約語にすることはできません。

表が作成された後で、その名前を変更することはできません。

ICWFILE

オブジェクト・タイプに関連付けられる Windows アイコンを含むファイルを指定します。

オプション・キーワード。

Windows_ICON_file_name

オブジェクト・タイプに関連付ける Windows アイコン・ファイルの名前を指定します。 *Windows_ICON_file_name* の最大長は 254 文字です。ただし、この名前とアイコン・パス (ICOPATH) の組み合わせの最大長が 259 であるため、実際の最大長はアイコン・パスの長さによって異なります。このファイルには、任意の拡張子を付けることができます。この値は、大文字小文字が区別されないため、大文字または小文字を使用して指定することができます。

このキーワードを使用して、アイコン・ファイルが存在する場所を識別するドライブおよびパス情報を指定することはできません。この情報は、FLGImport API 呼び出し (情報カタログ・マネージャー プログラミングの手引きおよび解説書 を参照)、ユーザー・インターフェースのインポート・ファンクション (44ページの『タグ言語ファイルをコマンド行からインポートする』を参照)、または DGUIDE コマンドの IMPORT オプション (44ページの『タグ言語ファイルをコマンド行からインポートする』を参照) への入力パラメーターとして指定しなければなりません。

OBJECT

この値は、ACTION.OBJTYPE(UPDATE) によってオブジェクト・タイプが作成された後で変更することができます。あるオブジェクト・タイプに関連付けるアイコン・ファイルを指定した後で、関連付けられたアイコンを変更することはできますが、そのオブジェクト・タイプは常にアイコンに関連付けられていなければなりません。

ACTION.OBJTYPE(APPEND)

コンテキスト:

```
:ACTION.OBJTYPE(APPEND)
:OBJECT.TYPE()
:PROPERTY.EXTNAME() DT() DL() SHRTNAME() NULLS() UUISEQ()
```

図 61. オブジェクト・タイプへのプロパティの追加時の OBJECT タグの使用

キーワード:

TYPE

オブジェクト・タイプの名前 (*type*) を指定します。

必須キーワード。

type

特定のオブジェクト・タイプをその 8 文字の短縮名によって識別します。

ACTION.OBJTYPE(DELETE) または ACTION.OBJTYPE(DELETE_EXT)

既存のオブジェクト・タイプの削除

コンテキスト:

```
:ACTION.OBJTYPE(DELETE)
:OBJECT.TYPE()
```

図 62. オブジェクト・タイプの削除時の OBJECT タグの使用

```
:ACTION.OBJTYPE(DELETE_EXT)
:OBJECT.TYPE()
```

図 63. オブジェクト・タイプと、そのタイプのすべてのオブジェクトの削除時の OBJECT タグの使用

キーワード:

TYPE

オブジェクト・タイプの名前 (*type*) を指定します。
必須キーワード。

type

特定のオブジェクト・タイプをその 8 文字の短縮名によって識別します。

ACTION.OBJTYPE(UPDATE)

オブジェクト・タイプの情報の更新

コンテキスト:

```
:ACTION.OBJTYPE(UPDATE)
:OBJECT.TYPE() EXTNAME() ICOFILE() ICWFILE()
```

図 64. オブジェクト・タイプの更新時の *OBJECT* タグの使用

キーワード:**TYPE**

オブジェクト・タイプの名前 (*type*) を指定します。
必須キーワード。

type

特定のオブジェクト・タイプをその 8 文字の短縮名によって識別します。
この値を更新することはできません。

EXTNAME

オブジェクト・タイプの記述名を指定します。オプション・キーワード。

ext_name

オブジェクト・タイプの長い記述名を指定します。 *ext_name* の最大長は 80 文字です。

この値は更新することができます。

この名前は、関連する情報カタログの中で固有でなければなりません。

ext_name の値は大文字小文字混合で保管されます。

ICWFILE

オブジェクト・タイプに関連付けられる Windows アイコンを含むファイル
を指定します。

オプション・キーワード。

OBJECT

Windows_ICON_file_name

オブジェクト・タイプに関連付ける Windows アイコン・ファイルの名前を指定します。

この値は更新することができます。

Windows_ICON_file_name の最大長は 254 文字です。このキーワードを使用して、アイコン・ファイルが存在する場所を識別するドライブおよびパス情報を指定することはできません。この情報は、FLGImport API 呼び出し、ユーザー・インターフェースのインポート・ファンクション、または情報カタログ・マネージャー・コマンドの IMPORT オプションへの入力パラメーターとして指定しなければなりません。

ACTION.OBJINST

オブジェクトの追加、更新、削除、または組み合わせ

コンテキスト:

```
:ACTION.OBJINST(ADD)
:OBJECT.TYPE()
:INSTANCE.short_name()
```

図 65. オブジェクトの追加時の *OBJECT* タグの使用

```
:ACTION.OBJINST(MERGE)
:OBJECT.TYPE()
:INSTANCE.short_name()
```

図 66. オブジェクトの組み合わせ時の *OBJECT* タグの使用

```
:ACTION.OBJINST(UPDATE)
:OBJECT.TYPE()
:INSTANCE.SOURCEKEY(UII_short_name()...) short_name()
```

図 67. オブジェクトの更新時の *OBJECT* タグの使用

```
:ACTION.OBJINST(DELETE)
:OBJECT.TYPE()
:INSTANCE.SOURCEKEY(UII_short_name()...)
```

図 68. オブジェクトの削除時の *OBJECT* タグの使用

キーワード:

TYPE

オブジェクト・タイプの名前 (*type*) を指定します。

必須キーワード。

type

特定のオブジェクト・タイプをその 8 文字の短縮名によって識別します。

PROPERTY

オブジェクト・タイプに属するプロパティを定義します。

このタグは、次の ACTION タグの後に続けることが必要です。

```
:ACTION.OBJTYPE(ADD)
```

```
:ACTION.OBJTYPE(MERGE)
```

```
:ACTION.OBJTYPE(APPEND)
```

構文

```
:PROPERTY.EXTNAME(ext_name) DT(data_type) DL(data_length)  
SHRTNAME(short_name) NULLS(Y | N) UUISEQ(UUI_number)
```

コンテキスト

```
:ACTION.OBJTYPE(ADD)  
:OBJECT.TYPE() CATEGORY() EXTNAME() PHYNAME() ICOFILE() ICWFILE()  
:PROPERTY.EXTNAME() DT() DL() SHRTNAME() NULLS() UUISEQ()
```

図 69. オブジェクト・タイプの追加時の PROPERTY タグの使用

```
:ACTION.OBJTYPE(MERGE)  
:OBJECT.TYPE() CATEGORY() EXTNAME() PHYNAME() ICOFILE() ICWFILE()  
:PROPERTY.EXTNAME() DT() DL() SHRTNAME() NULLS() UUISEQ()
```

図 70. オブジェクト・タイプの組み合わせ時の PROPERTY タグの使用

PROPERTY

```
:ACTION.OBJTYPE(APPEND)
:OBJECT.TYPE()
:PROPERTY.EXTNAME() DT() DL() SHRTNAME() NULLS() UUISEQ()
```

図 71. オブジェクト・タイプへのプロパティの追加時の *PROPERTY* タグの使用

キーワード

EXTNAME

プロパティの記述名を指定します。

必須キーワード。

ext_name

長い記述名を指定します。

ext_name の最大長は 80 文字です。 *ext_name* は、オブジェクト・タイプの中で固有でなければなりません。 *ext_name* は大文字小文字混合で保管されます。

DT

プロパティのデータ・タイプを指定します。

必須キーワード。

data_type

プロパティのデータ・タイプ。この値は、大文字または小文字で指定することができます。有効な値は、以下のとおりです。

C	CHAR
V	VARCHAR
L	LONG VARCHAR
T	TIMESTAMP

DL

プロパティのデータ長または最大データ長を指定します。

必須キーワード。

data_length

プロパティのデータ長または最大データ長。 *data_length* の有効な値は、このプロパティについて定義される *data_type* によって異なります。

data_type	data_length の最大値
C (CHAR)	最大長は 254

V (VARCHAR)	最大長は 4000
L (LONG VARCHAR)	最大長は 32700
T (TIMESTAMP)	常に 26 文字

SHRTNAME

プロパティ短縮名を指定します。

必須キーワード。

short_name

プロパティの短縮名。 *short_name* の値は、8 文字までの長さにすることができます。この値には、SBCS 文字だけを含めることができます。

この値は、大文字として保管され、小文字は大文字に変換されます。

この値は、文字 A ~ Z、@、#、または \$ で始まり、これらの文字と、0 ~ 9 および _ を含むことができます。先行空白または組み込み空白は許可されません。

この値は、情報カタログに関して使用されるデータベースを表す SQL 予約語にすることはできません。それぞれの情報カタログ・マネージャー・オブジェクト・タイプについて、次の必須プロパティの短縮名は指定しないでください: OBJTYPID、INSTIDNT、UPDATIME、または UPDATEBY。

NULLS

それぞれのオブジェクトについてプロパティの値が必須であるかどうかを指定します。この値は、大文字または小文字で指定することができます。

必須キーワード。

Y は、この値がヌル可能であることを示します。

ACTION.OBJTYPE(APPEND) タグを使用して新規のプロパティを追加する際には、NULLS(Y) を指定しなければなりません。これは、追加するプロパティがオプションでなければならないためです。

N は、このプロパティの値が必須であることを示します。情報カタログへのオブジェクトの追加時に、必須プロパティにデータが存在しない場合、データ・タイプが CHAR、VARCHAR、および LONG VARCHAR の必須値については、該当しないシンボルが入力されます。必須値のデータ・タイプが TIMESTAMP の場合は、次の値が入力されます:

9999-12-31-24.00.00.000000

UISEQ

UII で使用されるプロパティを識別します。

PROPERTY

オプション・キーワード。デフォルト値は 0 です。 UISEQ キーワードは、UII の一部ではないプロパティについてはオプションです。 UII は、それぞれのオブジェクトを固有に識別するキーとして、管理者によって定義されるプロパティのセットです。

UII_number

UII シーケンスにおけるプロパティの位置を指定します。有効な値は 0、1、2、3、4、および 5 です。値 0 は、プロパティが UII の一部ではないことを意味します。 *UII_number* の非ゼロ値は、プロパティが UII の一部であることを示します。

タグ言語ファイルで定義されるすべてのオブジェクト・タイプは、UII の一部であるプロパティを少なくとも 1 つ持たなければなりません。UII は、最大 5 つのプロパティから構成することができます。

少なくとも 1 つのプロパティを UII の一部として定義しなければなりません。

複数のプロパティに *UII_number* 値を割り当てる際には、UII プロパティの番号は、1 から UII プロパティの数までの範囲でなければなりません。たとえば、3 つのプロパティが UII の一部として定義される場合、*UII_number* 値は 1、2、および 3 でなければなりません。シーケンス内の番号をスキップすることはできません。*UII_number* 値は、プロパティを指定するのと同じ順序にする必要はありません。

規則

- 新規のオブジェクト・タイプを追加するか、またはオブジェクト・タイプを組み合わせる際には、予約済みのプロパティ NAME を UII の一部として定義することができます。図 72 では、NAME を UII プロパティとして識別するための一般的な構文を示します。

```
:ACTION.OBJTYPE(ADD)
:OBJECT.TYPE() CATEGORY() EXTNAME() PHYNAME() ICOFILE() ICWFILE()
:PROPERTY.SHRTNAME(NAME) UISEQ()
```

図 72. NAME プロパティを UII の一部として指定する例

この図にある空の括弧は、タグ言語ファイルの中でユーザーが指定しなければならない値を示しています。

- UII フィールドの最大長は 254 バイトです。

RELTYPE

追加または削除される関係のタイプと、関係に参加するオブジェクトのオブジェクト・タイプを識別します。

このタグは、次のタグの直後に続けることが必要です。

```
:ACTION.RELATION(ADD)
:ACTION.RELATION(DELETE)
```

構文

```
:RELTYPE.TYPE(CONTAIN | CONTACT | ATTACHMENT | LINK)
      SOURCETYPE(source_type) TARGETTYPE(target_type)
```

コンテキスト

```
:ACTION.RELATION(ADD)
:RELTYPE.TYPE() SOURCETYPE() TARGETTYPE()
:INSTANCE.SOURCEKEY(UI_short_name(...)) TARGETKEY(UI_short_name(...))
```

図 73. 関係の追加時の RELTYPE タグの使用

```
:ACTION.RELATION(DELETE)
:RELTYPE.TYPE() SOURCETYPE() TARGETTYPE()
:INSTANCE.SOURCEKEY(UI_short_name(...)) TARGETKEY(UI_short_name(...))
```

図 74. 関係の削除時の RELTYPE タグの使用

キーワード

TYPE

関係のタイプを指定します。

必須キーワード。

有効な値は、以下のとおりです。

ATTACHMENT

付加関係: ターゲット・オブジェクトがソース・オブジェクトに付加されます。

CONTACT

コンタクト関係: ソース・オブジェクトがターゲットのコンタクト・オブジェクトに関連付けられます。

CONTAIN

包含関係: ソース・オブジェクトがターゲット・オブジェクトを含みます。

LINK リンク関係: ソース・オブジェクトがターゲット・オブジェクトにリンクされます。

SOURCETYPE

ソース・オブジェクト・タイプを識別します。

必須キーワード。

source_type

ソース・オブジェクト・タイプ名 *source_type* は、OBJECT タグの TYPE キーワードの *type* 値と一致します。 *source_type* の最大長は 8 文字です。この値は、大文字小文字が区別されないため、大文字または小文字を使用して指定することができます。

付加関係の場合、*source_type* は接続以外のオブジェクト・タイプ名です。

包含関係の場合、*source_type* はコンテナ・オブジェクト・タイプ名です。

接点またはリンク関係の場合、*source_type* はグループ化または基礎的オブジェクト・タイプ名です。

TARGETTYPE

ターゲット・オブジェクト・タイプを識別します。

必須キーワード。

target_type

ターゲット・オブジェクト・タイプ名。 *target_type* は、OBJECT タグの TYPE キーワードの *type* 値と一致します。 *target_type* の最大長は 8 文字です。この値は、大文字小文字が区別されないため、大文字または小文字を使用して指定することができます。

付加関係の場合、*target_type* は接続オブジェクト・タイプ名です。

包含関係の場合、*target_type* は包含されるオブジェクト・タイプ名です。

接点関係の場合、*target_type* はコンタクト・オブジェクト・タイプ名です。

リンク関係の場合、*target_type* はグループ化または基礎的オブジェクト・タイプ名です。

TAB

プロパティ値の中でタブを指定します。

情報カタログ・マネージャーは、非 UII プロパティ値の中で指定された TAB タグのみを読み取り、それ以外はすべて無視します。

構文

:TAB.

規則

TAB タグは、INSTANCE タグの *property_values* の指定の中でのみ使用してください。

TAB

第10章 タグ言語ファイルの使い方

タグを使用すると、オブジェクト・タイプとオブジェクトを追加、削除、および更新することができます。情報カタログ・マネージャー・タグはコンテキストに依存するため、行いたいことに応じてさまざまな組み合わせでタグを指定します。

タグ言語ファイルの DISKCNTL からの開始

ファイルが取り外し可能ディスク (ディスクセットなど) にある場合は、タグ言語ファイルを DISKCNTL タグで開始してください。たとえば、次のようになります。

```
:DISKCNTL.SEQUENCE(01,+)
```

タグ言語ファイルが複数のディスクセットに入っている場合は、各ディスクセット上のタグ言語ファイルのそれぞれのセクションで、DISKCNTL を最初のタグにしなければなりません。タグ言語ファイルがハード・ディスクにある場合、DISKCNTL は無視されます。

追加、変更、および削除の定義

タグ言語では、アクションと、それらのアクションのオブジェクトを定義します。

行うことの定義

ACTION タグは、行うべきことを情報カタログ・マネージャーに指示するために使用します。キーワードでは、保守する情報の種類を指定し、オプションでは、実行するタスクを指定します。

```
:ACTION.OBJINST(option)
```

オブジェクトの保守。

```
:ACTION.OBJTYPE(option)
```

オブジェクト・タイプの保守。

```
:ACTION.RELTYPE(option)
```

オブジェクトの関係の保守。

情報の定義

行いたいことを指定した後、追加、変更、または削除する情報を正確に定義することが必要です。

定義する情報:

既存のオブジェクト・タイプ
組み合わせられるオブジェクト・タイプ
新規のオブジェクト・タイプ
オブジェクト・タイプについての新規のプロパティ
新規または既存のオブジェクト
オブジェクトの新規または既存の関係

使用するタグ:

OBJECT
OBJECT および PROPERTY
OBJECT および PROPERTY
OBJECT および PROPERTY
OBJECT および INSTANCE
RELTYPE および INSTANCE

タグの組み立て

OBJECT、INSTANCE、および PROPERTY タグに必要なキーワードと値は、それらのタグで追加、変更、または削除するために識別するものによって異なります。それぞれの ACTION タグの中でのタグの順序は、以下のとおりです。

:ACTION.OBJINST(*option*)

```
:ACTION.OBJINST(ADD)
:OBJECT.TYPE()
:INSTANCE.short_name() ...

:ACTION.OBJINST(DELETE)
:OBJECT.TYPE()
:INSTANCE.SOURCEKEY(UII_short_name()...)

:ACTION.OBJINST(DELETE_TREE_ALL)
:OBJECT.TYPE()
:INSTANCE.SOURCEKEY(UII_short_name()...)

:ACTION.OBJINST(DELETE_TREE_REL)
:OBJECT.TYPE()
:INSTANCE.SOURCEKEY(UII_short_name()...)

:ACTION.OBJINST(MERGE)
:OBJECT.TYPE()
:INSTANCE.short_name() ...

:ACTION.OBJINST(UPDATE)
:OBJECT.TYPE()
:INSTANCE.SOURCEKEY(UII_short_name()...) short_name()
```

:ACTION.OBJTYPE(*option*)

```
:ACTION.OBJTYPE(ADD)
:OBJECT.TYPE() CATEGORY() EXTNAME() PHYNAME() ICOFILE()
:PROPERTY.EXTNAME() DT() DL() SHRTNAME() NULLS() UIISEQ()
```

```

:ACTION.OBJTYPE(APPEND)
:OBJECT.TYPE()
:PROPERTY.EXTNAME() DT() DL() SHRTNAME() NULLS() UUISEQ()

:ACTION.OBJTYPE(DELETE)
:OBJECT.TYPE()

:ACTION.OBJTYPE(DELETE_EXT)
:OBJECT.TYPE()

:ACTION.OBJTYPE(MERGE)
:OBJECT.TYPE() CATEGORY() EXTNAME() PHYNAME() ICOFILE() ICWFILE()
:PROPERTY.EXTNAME() DT() DL() SHRTNAME() NULLS() UUISEQ()

:ACTION.OBJTYPE(UPDATE)
:OBJECT.TYPE() EXTNAME() ICOFILE() ICWFILE()

```

:ACTION.RELATION(*option*)

```

:ACTION.RELATION(ADD)
:RELTYPE.TYPE(CONTAIN | CONTACT | ATTACHMENT | LINK) SOURCETYPE(type)
TARGETTYPE(type)
:INSTANCE.SOURCEKEY(UII_short_name()...) TARGETKEY(UII_short_name()...)

:ACTION.RELATION(DELETE)
:RELTYPE.TYPE(CONTAIN | CONTACT | ATTACHMENT | LINK) SOURCETYPE(type)
TARGETTYPE(type)
:INSTANCE.SOURCEKEY(UII_short_name()...) TARGETKEY(UII_short_name()...)

```

INSTANCE、OBJECT、および PROPERTY タグの形式に関する具体的な情報については、258ページの『INSTANCE』、264ページの『OBJECT』、または271ページの『PROPERTY』を参照してください。

データベースへの変更のコミット

COMMIT タグは、情報カタログ・データベースへの変更をコミットします。COMMIT タグが処理されると、次のタグのセットの処理が開始される前に、エコー・ファイルが空にされます。これにより、エコー・ファイルには、コミットされていない変更を記述するタグのみが含まれるようになります。

情報カタログ・マネージャーは、エラーを検出すると、最後にコミットされたチェックポイントまでデータベースをロールバックします。データの整合性を維持し、データベースのロールバック時に取り消される変更の数を制限するために、ファイルに COMMIT タグを挿入してください。

COMMIT タグは、あるアクションを定義するタグの完結したセットの後に挿入することができます。ある ACTION タグと、その ACTION タグに関連するデータを定義する最後のタグの間に、COMMIT タグを挿入しないでください。

```
:COMMIT.CHKPT(20)
```

タグ言語ファイルへのコメントの書き込み

COMMENT タグを使用すると、情報カタログにインポートしたくない情報（注釈やラベルなど）をタグ言語ファイルに書き込むことができます。

```
:COMMENT.Updating the LASTDATE property
```

第3部 提供されているプログラムおよびマクロのリファレンス

第11章 提供されているデータウェアハウスセンター プログラム

データウェアハウスセンターは、データウェアハウスセンターとの統合をサポートするために以下のプログラムを提供しています。

- VWPEXUNX
- ISV_Sample

VWPEXUNX

VWPEXUNX プログラムは、リモート側でコマンドを発行またはプログラムを実行します。VWPEXUNX は、Windows NT、Windows 2000、および UNIX® 上で実行します。

VWPEXUNX プログラムを Windows NT または Windows 2000 上で実行している場合、REXECD プログラムもワークステーション上で実行していなければなりません。

パラメーター

表121 は、VWPEXUNX プログラムのパラメーター・リストを示しています。このリストには、パラメーターのための事前定義トークン (存在する場合) も含まれています。

表 121. VWPEXUNX のパラメーター

順序	説明
1	リモート・ホスト名。
2	リモート・ユーザー ID。
3	実行するリモート・プログラム。
4	リモート・エラー・ファイル。
5	リモート警告ファイル。警告ファイルが存在しない場合、- (該当しないことを示す記号) を指定します。
6	リモート・ログ (要約) ファイル。ログ・ファイルが存在しない場合、- (該当しないことを示す記号) を指定します。
7	リモート・オペレーティング・システムのタイプ。UNIX、WINNT、または WIN2000 のいずれかを指定します。

表 121. VWPEXUNX のパラメーター (続き)

順序	説明
8	パスワードのタイプ。 PasswordNotRequired、 EnterPassword、または GetPassword のいずれかを指定します。
9	パスワード・タイプが EnterPassword である場合のパスワード値。 パスワード・タイプが PasswordNot Required である場合は、- (該当しないことを示す記号)。 パスワード・タイプが GetPassword である場合は、パスワード・プログラム。パスワード・プログラムはステップのために選択されたエージェント・サイトに存在しなければなりません。そのプログラムは、使用するパスワードをファイルの最初の行に含むファイルを記述しなければなりません。正常に実行した場合には 0 を戻さなければなりません。
10	パスワード・タイプが GetPassword である場合は、パスワード・プログラム・パラメーター。

以下の例は、VWPEXUNX プログラムをコマンド行から開始する方法を示しています。コマンドはすべて同一行にタイプしなければなりません。この例における改行には意味はありません。

```
wvpexunx tomari labriejj db2cmd ¥usr¥labriejj¥db2cmd.err - -
UNIX EnterPassword mypass
```

tomari リモート・ホストの名前

labriejj リモート・ホストへのアクセスに使用するユーザー ID

db2cmd 実行するリモート・プログラム

¥usr¥labriejj¥db2cmd.err
 リモート・エラー・ファイルのパスおよび名前

- リモート警告ファイルは存在しない

- リモート・ログ (要約) ファイルは存在しない

UNIX リモート・オペレーティング・システム

EnterPassword
 パスワードのタイプ

mypass パスワード

戻りコード

VWPEXUNX プログラムはリモート・エラー・ファイルを使用して、リモート・コマンドまたはプログラムが成功したかエラーとなったかを、以下のように判別します。

- エラー・ファイルが空であるか存在しない場合、VWPEXUNX プログラムは成功を示すエラー・コードを戻します。
- エラー・ファイルが空ではない場合、VWPEXUNX プログラムは:
 - エラー・ファイルの内容を一時ファイルに保管します。
 - 失敗を示すエラー・コードを戻します。

VWPEXUNX プログラムは、リモート・エラー・ファイルの内容は調べません。

表122 は、VWPEXUNX プログラムの戻りコードをリストしています。

表 122. VWPEXUNX プログラムの戻りコード

戻りコード	説明
0	プログラムは正常に実行しました。
4	プログラムの実行時に警告が出されました。 プログラムはパスワード・プログラムの実行後にパスワード・ファイルを削除できませんでした。
8	パラメーター・エラー。 プログラムに供給されたパラメーターの数が多過ぎるか少な過ぎます。または無効値がパラメーターとして指定されました。
16	内部エラー。 プログラムは、一時ファイルのオープン、作成、書き込み不可などの内部エラーを検出しました。
48	環境変数エラー。 VWS_LOGGING 環境変数が設定されていません。
52	パスワード・プログラム・エラーの取得。 プログラムは、プログラムの欠落、名前の無効、パラメーター数の不適切などのパスワード・プログラム・エラーを検出しました。

表 122. VWPEXUNX プログラムの戻りコード (続き)

戻りコード	説明
56	<p>リモート実行エラー。</p> <p>プログラムは、以下のエラーのようなりモート実行エラーを検出しました。</p> <ul style="list-style-type: none"> 指定されたユーザー ID またはパスワードが不適切です。 リモート・ファイルが見つかりません。 リモート・ホストが応答しません。 指定されたユーザー ID は、リモート・ファイルの作成または読み取りを許可されていません。

ログ・ファイル

VWPEXUNX プログラムは、トレース・ファイルを `VWS_LOGGING` 環境変数が指定するディレクトリーに書き込みます。

ISV_Sample

ISV_Sample プログラムは、ODBC データ・ソースからメタデータを読み取り、そのメタデータからデータウェアハウスセンター オブジェクトを生成します。ISV_Sample プログラムは、Windows NT および Windows 2000 上で実行します。

表123 は、ISV_Sample プログラムのパラメーター・リストを示しています

このパラメーターのための事前定義されたトークンはありません。

表 123. ISV_Sample のパラメーター

順序	説明
1	メタデータを取り出す ODBC DSN
2	ODBC ユーザー ID
3	ODBC パスワード

以下の例は、ISV_Sample プログラムを開始する方法を示しています。

```
ISV_Sample SAMPLE labriejj mypass
```

SAMPLE メタデータを読み取る ODBC DSN

labriejj ODBC DSN へのアクセスに使用するユーザー ID

mypass ODBC DSN へのアクセスに使用するパスワード

ISV_Sample プログラムは、ISV_VWP プログラムを使用します。ステップは ISV_VWP プログラムを呼び出して、入力パラメーターを出力ファイルに書き込みます。

第12章 Net.Data[®] マクロ

情報カタログ・マネージャー (Web 版) は Net.Data[®] マクロを使用して Web 上のデータを表示し、データベース内のデータを検索します。Net.Data およびそのマクロについてよく知っていれば、企業の必要に適合するようにマクロをカスタマイズすることができます。

たとえば、デフォルトでは情報カタログ・マネージャー (Web 版) はユーザー ID およびパスワードの入力を必要とします。マクロをカスタマイズすると、代わりに独自の機密保護プログラムを呼び出すようにすることができます。

この章では、情報カタログ・マネージャー (Web 版) に含まれるファイルをリストします。Net.Data およびそのマクロについての詳細は、*Net.Data 管理およびプログラミングの手引き* および *Net.Data 解説書* を参照してください。

情報カタログ・マネージャー (Web 版) のファイル

情報カタログ・マネージャー (Web 版) のファイルに対して作業を行うためには、まずアドミニストレーション・クライアントのカスタム・インストールを実行して、情報カタログ・マネージャー (Web 版) を選択しなければなりません。ファイルは `x:%sqllib%icuweb` ディレクトリーにインストールされます。

ファイル名は、AIX[®] 命名規則に従うために、小文字となります。

表124 は、Net.Data マクロを含む情報カタログ・マネージャー (Web 版) のファイルをリストしています。これらのファイルは、`x:%sqllib%icuweb%macro` ディレクトリーにあります。

表 124. 情報カタログ・マネージャー Web Net.Data マクロ

ファイル名	説明
<code>dg_list.mac</code>	検索、ツリー、またはサブジェクト呼び出しの結果を表示する
<code>dg_desc.mac</code>	説明ビューの結果を表示する
<code>dg_frame.mac</code>	3 フレームのページを作成する
<code>dg_advsearch.mac</code>	拡張検索を実行する
<code>dg_comment.mac</code>	コメントを作成または更新する

情報カタログ・マネージャー (Web 版) のファイル

表 124. 情報カタログ・マネージャー Web Net.Data マクロ (続き)

ファイル名	説明
dg_home.mac	情報カタログ・マネージャー・ホーム・ページを表示する
dg_tableviewer.mac	サンプル・データを表示する

表125 は、Net.Data 組み込みファイルを含む情報カタログ・マネージャー (Web 版) のファイルをリストしています。これらのファイルは、`x:\$sqllib¥icuweb¥macro` ディレクトリーにあります。

表 125. Net.Data 組み込みファイル

ファイル名	説明
dg_desc.hti	説明ビューのための共通関数を含む組み込みファイル
dg_home.hti	情報カタログ・マネージャー・ホーム・ページ上に表示する情報カタログのリストを含む組み込みファイル
dg_strings.hti	変換可能ストリングを含む組み込みファイル
dg_config.hti	インストールで設定可能な変数を含む組み込みファイル
dg_graphics.hti	図形のルック・アンド・フィール定義を含む組み込みファイル

表126 は、HTML を含む情報カタログ・マネージャー (Web 版) のファイルを示しています。これらのファイルは、`x:\$sqllib¥icuweb¥html` ディレクトリーにあります

表 126. 情報カタログ・マネージャー (Web 版) HTML ファイル

ファイル名	説明
*.htm	ヘルプ・ファイル

293ページの表127 は、情報カタログ・マネージャー (Web 版) の図形ファイルをリストしています。これらのファイルは、`x:\$sqllib¥icuweb¥icons` ディレクトリーにあります。

以下にリストする図形ファイルに加えて、情報カタログ・マネージャー内に作成した新規のオブジェクト・タイプを示す固有のアイコンを作成することもで

きます。オブジェクト・タイプのアイコンを作成する方法についての詳細は、
 情報カタログ・マネージャー 管理の手引き を参照してください。

表 127. 情報カタログ・マネージャー (Web 版) 図形ファイル

ファイル名	説明
dg_ibmlogo.gif	IBM ロゴ
dg_lgudblogo.gif	ホーム用の大きな DB2 ロゴ
dg_smudblogo.gif	ヘッダー用の小さな DB2 ロゴ
dg_curve.gif	小さな曲線結合ヘッダーおよびメニュー
dg_lgappldata.gif	大きなアプリケーション・データ
dg_smappldata.gif	小さなアプリケーション・データ
dg_lgapproach.gif	大きな Lotus Approach
dg_smapproach.gif	小さな Lotus Approach
dg_lgaudio.gif	大きなオーディオ・クリップ
dg_smaudio.gif	小さなオーディオ・クリップ
dg_lgcharts.gif	大きな図表
dg_smcharts.gif	小さな図表
dg_lgcolumns.gif	大きなカラム
dg_smccolumns.gif	小さなカラム
dg_lgcomments.gif	大きなコメント
dg_smcomments.gif	小さなコメント
dg_lgcontact.gif	連絡先の大きな人
dg_smcontact.gif	連絡先の小さな人
dg_lgdatas.gif	大きなデータベース
dg_smdatas.gif	小さなデータベース
dg_lgimsdbd.gif	大きな IMS データベース定義 (DBD)
dg_smimsdbd.gif	小さな IMS データベース定義 (DBD)
dg_lgdgnews.gif	大きなニュース
dg_smdgnews.gif	小さなニュース
dg_lgdimension.gif	多次元データベース内の大きな次元
dg_smdimension.gif	多次元データベース内の小さな次元
dg_lgdocs.gif	大きな文書
dg_smdocs.gif	小さな文書
dg_lgelement.gif	大きな要素
dg_smelement.gif	小さな要素

表 127. 情報カタログ・マネージャー (Web 版) 図形ファイル (続き)

ファイル名	説明
dg_lgfile.gif	大きなファイル
dg_smfile.gif	小さなファイル
dg_lgfilter.gif	大きな変換
dg_smfilter.gif	小さな変換
dg_lgglossary.gif	大きな用語集項目
dg_smglossary.gif	小さな用語集項目
dg_lgimages.gif	大きなイメージまたは図形
dg_smimages.gif	小さなイメージまたは図形
dg_lginfogrps.gif	大きなビジネス対象エリア
dg_sminfogrps.gif	小さなビジネス対象エリア
dg_lginternet.gif	大きなインターネット文書
dg_sminternet.gif	小さなインターネット文書
dg_lgmember.gif	多次元データベース内の大きなメンバー
dg_smmember.gif	多次元データベース内の小さなメンバー
dg_lgolapmodl.gif	大きな多次元データベース
dg_smolapmodl.gif	小さな多次元データベース
dg_lgolnews.gif	大きなオンライン・ニュース・サービス
dg_smolnews.gif	小さなオンライン・ニュース・サービス
dg_lgolpubs.gif	大きなオンライン・ニュース・サービス
dg_smolpubs.gif	小さなオンライン・ニュース・サービス
dg_lgiimspcb.gif	大きな IMS プログラム制御ブロック (PCB)
dg_smimspcb.gif	小さな IMS プログラム制御ブロック (PCB)
dg_lgpresent.gif	大きな表示
dg_smpresent.gif	小さな表示
dg_lgimspsb.gif	大きな IMS プログラム仕様 (PSB)
dg_smimspsb.gif	小さな IMS プログラム仕様 (PSB)
dg_lgrecord.gif	大きなレコード
dg_smrecord.gif	小さなレコード
dg_lgreports.gif	大きなテキスト・ベースの報告書
dg_smreports.gif	小さなテキスト・ベースの報告書
dg_lgmsseg.gif	大きな IMS セグメント
dg_smmsseg.gif	小さな IMS セグメント

表 127. 情報カタログ・マネージャー (Web 版) 図形ファイル (続き)

ファイル名	説明
dg_lgssheets.gif	大きなスプレッドシート
dg_smssheets.gif	小さなスプレッドシート
dg_lgsubschem.gif	大きなサブスキーマ
dg_smssubschem.gif	小さなサブスキーマ
dg_lgtables.gif	大きなりレーショナル表およびビュー
dg_smtables.gif	小さなりレーショナル表およびビュー
dg_lgvideo.gif	大きなビデオ・クリップ
dg_smvideo.gif	小さなビデオ・クリップ
dg_lggrouping.gif	大きなグループ化 - デフォルトのカテゴリ ー・アイコン
dg_smgrouping.gif	小さなグループ化 - デフォルトのカテゴリ ー・アイコン
dg_lgelemental.gif	大きな基礎的 - デフォルトのカテゴリ ー・アイコン
dg_smelemental.gif	小さな基礎的 - デフォルトのカテゴリ ー・アイコン
dg_lgcontact.gif	大きなコンタクト - デフォルトのカテゴリ ー・アイコン
dg_smcontact.gif	小さなコンタクト - デフォルトのカテゴリ ー・アイコン
dg_lgdictionary.gif	大きな辞書 - デフォルトのカテゴリ ー・アイコン
dg_smdictionary.gif	小さな辞書 - デフォルトのカテゴリ ー・アイコン
dg_lgsupport.gif	大きなサポート - デフォルトのカテゴリ ー・アイコン
dg_smsupport.gif	小さなサポート - デフォルトのカテゴリ ー・アイコン
dg_lgattachment.gif	大きな接続 - デフォルトのカテゴリ ー・アイコン
dg_smattachment.gif	小さな接続 - デフォルトのカテゴリ ー・アイコン
dg_collapse.gif	ツリー - 縮小アイコン
dg_expand.gif	ツリー - 展開アイコン

情報カタログ・マネージャー (Web 版) のファイル

表 127. 情報カタログ・マネージャー (Web 版) 図形ファイル (続き)

ファイル名	説明
dg_lmore.gif	記述 - 長いプロパティ (より多くの矢印)
dg_clear.gif	スペーシングのための明確な図形

第4部 付録および後付け

付録A. テンプレート・プランのワークシート

パートナー・アプリケーションから必要な値を収集するために、このワークシートを使用してください。

トークンの値を表に書き込みます。許可される値のリストが示されているトークンについては、許可される値の 1 つを丸で囲みます。

表 128. テンプレートに必要なメタデータのトークン

トークン	値
<i>*AgentSite</i>	
<i>*AgentSiteContact</i>	
<i>*AgentSiteDescription</i>	
<i>*AgentSiteNotes</i>	
<i>*AgentSiteOSType</i>	以下の値の 1 つ: ISV_windowsNT Windows NT ISV_AIX AIX ISV_os2 OS/2 ISV_as400 AS/400 ISV_Solaris SUN ISV_MVS MVS
<i>*AgentSiteTCPIPHostName</i>	
<i>*AgentSiteUserId</i>	

表 128. テンプレートに必要なメタデータのトークン (続き)

トークン	値
<i>*ColumnAllowsNulls</i>	<p>以下の値の 1 つ:</p> <p>ISV_NULLSYES カラムにはヌル・データを入れることができます。</p> <p>ISV_NULLSNO カラムにはヌル・データを入れることができません。</p>
<i>*ColumnDataIsText</i>	<p>以下の値の 1 つ:</p> <p>ISV_ISTEXTYES カラムにはテキスト・データだけが含まれます。</p> <p>ISV_ISTEXTNO カラムにはテキスト以外のデータも含まれます。</p>
<i>*ColumnDescription</i>	
<i>*ColumnEditionType</i>	<p>以下の値の 1 つ:</p> <p>ISV_ColumnIsEditionColumn カラムはエディション・カラムです。</p> <p>ISV_ColumnIsNormal カラムは通常のカラムです。</p>
<i>*ColumnKeyPosition</i>	
<i>*ColumnLength</i>	
<i>*ColumnName</i>	

表 128. テンプレートに必要なメタデータのトークン (続き)

トークン	値
<i>*ColumnNativeDataType</i>	以下の値の 1 つ: ISV_NATIVE_CHAR ISV_NATIVE_VARCHAR ISV_NATIVE_LONGVARCHAR ISV_NATIVE_VARCHAR2 ISV_NATIVE_GRAPHIC ISV_NATIVE_VARGRAPHIC ISV_NATIVE_LONGVARGRAPHIC ISV_NATIVE_CLOB ISV_NATIVE_INT ISV_NATIVE_TINYINT ISV_NATIVE_BLOB ISV_NATIVE_SMALLINT ISV_NATIVE_INTEGER ISV_NATIVE_FLOAT ISV_NATIVE_SMALLFLOAT ISV_NATIVE_DOUBLE ISV_NATIVE_REAL ISV_NATIVE_DECIMAL ISV_NATIVE_SMALLMONEY ISV_NATIVE_MONEY ISV_NATIVE_NUMBER ISV_NATIVE_NUMERIC ISV_NATIVE_DATE ISV_NATIVE_TIME ISV_NATIVE_TIMESTAMP ISV_NATIVE_LONG ISV_NATIVE_RAW ISV_NATIVE_LONGRAW ISV_NATIVE_DATETIME ISV_NATIVE_SMALLDATETIME ISV_NATIVE_SYSNAME

表 128. テンプレートに必要なメタデータのトークン (続き)

トークン	値
* <i>ColumnNativeDataType</i> (続き)	以下の値の 1 つ: ISV_NATIVE_TEXT ISV_NATIVE_BINARY ISV_NATIVE_VARBINARY ISV_NATIVE_LONGVARBINARY ISV_NATIVE_BIT ISV_NATIVE_IMAGE ISV_NATIVE_SERIAL ISV_NATIVE_DATETIMEYEARTOFRACTION ISV_NATIVE_DBCLOB ISV_NATIVE_BIGINT
* <i>ColumnNotes</i>	
* <i>ColumnOffsetFromZero</i>	
* <i>ColumnOrdinalNumber</i>	
* <i>ColumnPositionNumber</i>	
* <i>ColumnPrecision</i>	
* <i>ColumnUserActions</i>	
* <i>CurrentCheckpointID++</i>	
* <i>DatabaseContact</i>	
* <i>DatabaseDescription</i>	
* <i>DatabaseName</i>	
* <i>DatabaseNotes</i>	
* <i>DatabasePhysicalName</i>	

表 128. テンプレートに必要なメタデータのトークン (続き)

トークン	値
<i>*DatabaseType</i>	以下の値の 1 つ: ISV_IR_DB2Family DB2 ファミリー ISV_IR_Oracle Oracle ISV_IR_Sybase Sybase ISV_IR_MSSQLServer Microsoft SQLServer ISV_IR_Informix Informix ISV_IR_GenericODBC 汎用 ODBC ISV_IR_FFLan フラット・ファイル LAN ISV_IR_VSAM VSAM ISV_IR_IMS IMS
<i>*DatabaseTypeExtended</i>	以下の値の 1 つ: ISV_IR_DB2400CISC CISC 用の DB2 UDB (AS/400® 版) ISV_IR_DB2400RISC RISC 用の DB2 UDB (AS/400 版) ISV_IR_FFLanLocalCmd ローカル・フラット・ファイル ISV_IR_FFLanFTPCopy リモート・システムから FTP を使用して送信されたローカル・フラット・ファイル
<i>*DatabaseServerName</i>	
<i>*DatabaseUserid</i>	

表 128. テンプレートに必要なメタデータのトークン (続き)

トークン	値
<i>*DatabaseVersion</i>	
<i>*PostStepName</i>	
<i>*ProcessContact</i>	
<i>*ProcessDescription</i>	
<i>*ProcessName</i>	
<i>*ProcessNotes</i>	
<i>*ProcessType</i>	以下の値の 1 つ: ISV_ProcessType_Normal プロセスは通常のユーザー・プロセスです。 ISV_ProcessType_Meta_pub プロセスはメタデータ資料プロセスです。 ISV_ProcessType_Notify プロセスは通知プロセスです。
<i>*SecurityGroup</i>	ISV_DEFAULTSECURITYGROUP
<i>*StarSchemaContact</i>	
<i>*StarSchemaDBName</i>	
<i>*StarSchemaDescription</i>	
<i>*StarSchemaName</i>	
<i>*StarSchemaNotes</i>	
<i>*StepCommit</i>	以下の値の 1 つ: ISV_Step_Incremental_Commit_On データはターゲットにおいて増分的にコミットされます。 ISV_Step_Incremental_Commit_Off データはターゲットにおいて増分的にコミットされません。
<i>*StepCommitAfterNumberRows</i>	
<i>*StepContact</i>	

表 128. テンプレートに必要なメタデータのトークン (続き)

トークン	値
<i>*StepDataNotPresent</i>	<p>以下の値の 1 つ:</p> <p>ISV_StepDataNotPresent_OK データが存在しない場合、処理を続けます。</p> <p>ISV_StepDataNotPresent_Warning データが存在しない場合、警告を発行して処理を続けます。</p> <p>ISV_StepDataNotPresent_Error データが存在しない場合、エラー・メッセージを発行して処理を停止します。</p>
<i>*StepDescription</i>	
<i>*StepExternalPopulation</i>	<p>以下の値の 1 つ:</p> <p>ISV_StepExternalNo 表には他の方法で外部的なデータが取り込まれることはありません。</p> <p>ISV_StepExternalYes 表には他の方法で外部的なデータが取り込まれます。</p>
<i>*StepName</i>	
<i>*StepNotes</i>	
<i>*StepSelectStatement</i>	
<i>*StepSelectStatementGenerated</i>	<p>以下の値の 1 つ:</p> <p>ISV_StepSelectStatementNo SELECT ステートメントは生成されませんが、<i>*StepSelectStatement</i> に含まれています。</p> <p>ISV_StepSelectStatementYes SELECT ステートメントが生成されて、<i>*StepSelectStatement</i> は無視されます。</p>

表 128. テンプレートに必要なメタデータのトークン (続き)

トークン	値
*StepSQLWarning	<p>以下の値の 1 つ:</p> <p>ISV_StepSQLWarning_OK SQL 警告が生じた場合、処理を続けます。</p> <p>ISV_StepSQLWarning_Warning SQL 警告が生じた場合、警告を発行して処理を続けます。</p> <p>ISV_StepSQLWarning_Error SQL 警告が生じた場合、エラーを発行して処理を停止します。</p>
*StepType	<p>以下の値の 1 つ:</p> <p>ISV_StepType_Editioned_Append ステップが実行されると、表内のデータは追加されます。</p> <p>ISV_StepType_Full_Replace ステップが実行されると、表内のデータは置換されます。</p> <p>ISV_StepType_Uneditioned_Append ステップが実行されると、表内のデータは追加されます。</p> <p>ISV_StepType_VWP_Population 表内のデータはデータウェアハウスセンター プログラムによって取り込まれます。</p>
*SubjectArea	
*SubjectAreaContact	
*SubjectAreaDescription	
*SubjectAreaNotes	

表 128. テンプレートに必要なメタデータのトークン (続き)

トークン	値
<i>*TableBinaryIfFile</i>	<p>以下の値の 1 つ:</p> <p>ISV_DR_FILE_IS_BINARY ファイルはバイナリーです。</p> <p>ISV_DR_FILE_IS_NOT_BINARY ファイルは ASCII、または混合形式です。</p>
<i>*TableCreatedByDWC</i>	<p>以下の値の 1 つ:</p> <p>ISV_TableIsToBeCreatedByDWC 表はデータウェアハウスセンターによって作成されます。</p> <p>ISV_TableIsNotToBeCreatedByDWC 表はデータウェアハウスセンターによって作成されません。</p>
<i>*TableCreateStatement</i>	
<i>*TableDelimiterIfFile</i>	
<i>*TableDescription</i>	
<i>*TableFirstRowNamesIfFile</i>	<p>以下の値の 1 つ:</p> <p>ISV_DR_ROW_CONTAINS_NAMES ファイルの最初の行には、列名が含まれます。</p> <p>ISV_DR_ROW_DOES_NOT_CONTAIN_NAMES ファイルの最初の行には、データが含まれません。</p>
<i>*TableFullName</i>	

表 128. テンプレートに必要なメタデータのトークン (続き)

トークン	値
<i>*TableGenerateCreateStatement</i>	<p>以下の値の 1 つ:</p> <p>ISV_GenerateCreateTableStmt データウェアハウスセンターは、 CREATE TABLE ステートメントを生成 することになります。</p> <p>ISV_DoNotGenerateCreateTableStmt データウェアハウスセンターは、 CREATE TABLE ステートメントを生成 しないことになります。</p>
<i>*TableGrantedToPublic</i>	<p>以下の値の 1 つ:</p> <p>ISV_GrantTableAccessToPublic この表に PUBLIC アクセスを権限付与し ます。</p> <p>ISV_DoNotGrantTableAccessToPublic この表に PUBLIC アクセスを権限付与し ません。</p>
<i>*TableIsAnAlias</i>	<p>以下の値の 1 つ:</p> <p>ISV_TableIsAnAlias この表は他の表の別名です。</p> <p>ISV_TableIsNotAnAlias この表は他の表の別名ではありません。</p>
<i>*TableIsADimensionTable</i>	<p>以下の値の 1 つ:</p> <p>ISV_TableIsADimensionalTable 表は次元表です。</p> <p>ISV_TableIsNotADimensionalTable 表は次元表ではありません。</p>

表 128. テンプレートに必要なメタデータのトークン (続き)

トークン	値
<i>*TableIsAFactTable</i>	以下の値の 1 つ: ISV_TableIsAFactTable 表はファクト表です。 ISV_TableIsNotAFactTable 表はファクト表ではありません。
<i>*TableIsAView</i>	以下の値の 1 つ: ISV_TableIsAView 表はビューです。 ISV_TableIsNotAView 表はビューではありません。
<i>*TableIsPersistent</i>	以下の値の 1 つ: ISV_TableIsPersistent 表は持続的なものとみなされます。 ISV_TableIsTransient 表は一時的なものとみなされます。
<i>*TableMaximumEditions</i>	
<i>*TableNotes</i>	
<i>*TableOwner</i>	
<i>*TablePhysicalName</i>	

表 128. テンプレートに必要なメタデータのトークン (続き)

トークン	値
<i>*TableTypeIfFile</i>	以下の値の 1 つ: ISV_DR_REL_TABLE 表はリレーショナル表です。 ISV_DR_COMMA_DELIMITED ファイル内の列はコンマで区切られます。 ISV_DR_FIXED_FORMAT ファイル内の列は混合形式です。 ISV_DR_TAB_DELIMITED ファイル内の列はタブで区切られます。 ISV_DR_CHAR_DELIMITED ファイル内の列は、 <i>*TableDelimiterIfFile</i> の値で区切られます。
<i>*VWPGGroup</i>	
<i>*VWPGGroupDescription</i>	
<i>*VWPGGroupNotes</i>	
<i>*VWPPProgramInstanceKey</i>	
<i>*VWPPProgramInstanceParameterData</i>	
<i>*VWPPProgramInstanceParameterKey</i>	
<i>*VWPPProgramInstanceParameterName</i>	
<i>*VWPPProgramInstanceParameterOrder</i>	
<i>*VWPPProgramInstanceParameterType</i>	以下の値の 1 つ: ISV_ParameterTypeNone パラメーター・タイプは不明です。 ISV_ParameterTypeCharacter パラメーター・タイプは文字です。 ISV_ParameterTypeNumeric パラメーター・タイプは数値です。 ISV_ParameterTypePassword パラメーター・タイプはパスワードです。

表 128. テンプレートに必要なメタデータのトークン (続き)

トークン	値
<i>*VWPPProgramTemplateDescription</i>	
<i>*VWPPProgramTemplateExecutableName</i>	
<i>*VWPPProgramTemplateFunctionName</i>	
<i>*VWPPProgramTemplateName</i>	
<i>*VWPPProgramTemplateNotes</i>	
<i>*VWPPProgramTemplateType</i>	以下の値の 1 つ: ISV_PROGRAMTYPEDLL データウェアハウスセンター プログラムはダイナミック・リンク・ライブラリー (DLL) からロードされるか、またはロード・モジュールです。 ISV_PROGRAMTYPECOMMAND データウェアハウスセンター プログラムはコマンド・ファイルです。 ISV_PROGRAMTYPEEXECUTABLE データウェアハウスセンター プログラムは実行可能ファイルです。
<i>*VWPPProgramTemplateParameterData</i>	
<i>*VWPPProgramTemplateParameterKey</i>	
<i>*VWPPProgramTemplateParameterName</i>	
<i>*VWPPProgramTemplateParameterOrder</i>	
<i>*VWPPProgramTemplateParameterType</i>	以下の値の 1 つ: ISV_ParameterTypeNone パラメーター・タイプは不明です。 ISV_ParameterTypeCharacter パラメーター・タイプは文字です。 ISV_ParameterTypeNumeric パラメーター・タイプは数値です。 ISV_ParameterTypePassword パラメーター・タイプはパスワードです。

付録B. Visual Warehouse 5.2 によってサポートされるテンプレート

バージョン 5.2 の Visual Warehouse および DataGuide® によって提供され、サポートされるテンプレートに関する詳細な情報については、この章を参照してください。各テンプレートのセクションには、そのテンプレートのトークンがリストされています。そこには、各トークンに許可される値、および値の長さが示されています。

注: この章で解説されるテンプレートは、使用禁止となっていて、拡張されることはありません。

ご使用の交換プログラムにトークンの値がない場合、トークンを ISV_DEFAULTVALUE に設定してください。ただし、必須のトークンには ISV_DEFAULTVALUE 以外の値を指定しなければなりません。

セキュリティー・グループのテンプレートは存在しないので、ご使用のプログラムでは値 ISV_DEFAULTSECURITYGROUP を *SecurityGroup トークンのインスタンスに指定しなければなりません。

テンプレートが Visual Warehouse パラメーターを設定しない場合、Visual Warehouse 定義はパラメーターのデフォルトの値を持つこととなります。たとえば、Visual Warehouse はソース・データベースの「再試行回数 (Retry Count)」および「再試行間隔 (Retry Interval)」パラメーターをデフォルト値に設定します。

表129 には、Visual Warehouse に備わっているメタデータ・テンプレート、および各テンプレートについて解説するセクションがリストされています。

表 129. Visual Warehouse 5.2 によってサポートされるメタデータ・テンプレート

BusinessView.tag	Visual Warehouse によって管理されるビジネス・ビューを定義します。	314ページの『BusinessView.tag』
BusinessViewInputTable.tag	ビジネス・ビューが指定されたソース表を使用することを指定します。	321ページの『BusinessViewInputTable.tag』

表 129. Visual Warehouse 5.2 によってサポートされるメタデータ・テンプレート (続き)

BusinessViewOutputTable.tag	ビジネス・ビューが指定されたターゲット表を使用することを指定します。	323ページの『BusinessViewOutputTable.tag』
BusinessViewVWOutputTable.tag	Visual Warehouse プログラムを使用するビジネス・ビューと Visual Warehouse プログラムの出力表との間の関係を指定します。	325ページの『BusinessViewVWOutputTable.tag』
ConcurrentCascade.tag	2 つのビジネス・ビューが同時に開始されることを示します。	327ページの『ConcurrentCascade.tag』
VWPPProgramInstance	Visual Warehouse プログラムの定義を、特定のビジネス・ビューが使用するために変更します。	329ページの『VWPPProgramInstance.tag』

BusinessView.tag

このテンプレートを使用して、ビジネス・ビューを定義します。パートナー・アプリケーションがデータ・ソースとターゲットとの間の関係を生成する場合、または Visual Warehouse が実行するプログラムを含む場合に、このテンプレートを使用しなければなりません。

このテンプレートにはさらに、セキュリティー・グループ、サブジェクト・エリア、および 1 つ以上のエージェント・サイトへの関係が含まれます。

トークン

315ページの表130 は、このテンプレート内の各トークンに関する情報を示しています。

表 130. BusinessView.tag のトークン

トークン	説明	許可される値	ウィンドウまたはノートブック: フィールド
エンティティ・パラメーター			
*BVName	<p>ビジネス・ビューの名前。</p> <p>その名前は、Visual Warehouse コントロール・データベース内で固有でなければなりません。</p> <p>このトークンは必須です。</p>	最長 80 バイトのテキスト・ストリング。	ビジネス・ビュー: ビジネス名
*BVDescription	<p>ビジネス・ビューについての短い説明。</p> <p>このトークンは任意指定です。</p>	最長 200 バイトのテキスト・ストリング。	ビジネス・ビュー: 説明
*BVNotes	<p>ビジネス・ビューについての長い説明。</p> <p>このトークンは任意指定です。</p>	最長 32700 バイトのテキスト・ストリング。	ビジネス・ビュー: 注記

BusinessView.tag

表 130. BusinessView.tag のトークン (続き)

トークン	説明	許可される値	ウィンドウまたはノートブック: フィールド
*BVDataNotPresent	<p>ビジネス・ビューのために抽出するデータがないことをエージェントが知ったとき、警告を扱う方法についての設定。</p> <p>このトークンは必須です。</p>	<p>以下の値の 1 つ:</p> <p>ISV_BVDataNotPresent_OK 抽出するデータがないことをエージェントが知ったとき、ビジネス・ビューは正常に処理を行います。</p> <p>ISV_BVDataNotPresent_Warning 抽出するデータがないことをエージェントが知ったとき、ビジネス・ビューは失敗します。</p> <p>ISV_BVDataNotPresent_Error 抽出するデータがないことをエージェントが知ったとき、ビジネス・ビューは警告を出して処理を続けます。</p>	ビジネス・ビュー: 行が戻されない処理オプション
*BVSelectStatementGenerated	<p>Visual Warehouse が SQL を生成するのか、または *BVSelectStatement トークンの値として SQL が備えられるのかを示すフラグ。</p> <p>このトークンは必須です。</p>	<p>以下の値の 1 つ:</p> <p>ISV_BVSELECTSTATEMENTYES Visual Warehouse は SQL を生成します。</p> <p>ISV_BVSELECTSTATEMENTNO SQL は *BVSelectStatement トークンの値として備えられます。</p>	なし
*BVSelectStatement	<p>実行される SQL ステートメント。</p> <p>*BVSelectStatementGenerated が N に設定されている場合、このトークンは必須です。</p>	<p>最長 32700 バイトの SQL ステートメント。</p>	<p>ビジネス・ビュー SQL の変更: SQL ステートメント</p>

表 130. BusinessView.tag のトークン (続き)

トークン	説明	許可される値	ウィンドウまたはノートブック: フィールド
*BVContact	このビジネス・ビューに関する質問についての連絡先となる個人またはグループの名前。 このトークンは任意指定です。	最長 64 バイトのテキスト・ストリング。	ビジネス・ビュー: 管理上の連絡
*BVExternalPopulation	外部アプリケーションが表にデータを入れるかどうかを示すフラグ。 このトークンは必須です。	以下の値の 1 つ: ISV_BVEXTERNALYES 外部アプリケーションは表にデータを入れられます。 ISV_BVEXTERNALNO Visual Warehouse だけが表にデータを入れられます。	ビジネス・ビュー: 外部的に取り込む
*BVCreateTargetTable	ビジネス・ビューが状況をテストするように要求されたとき、Visual Warehouse がターゲット表を作成するかどうかを示すフラグ。 このトークンは必須です。	以下の値の 1 つ: ISV_BVCREATETABLEYES Visual Warehouse はターゲット表を作成します。 ISV_BVCREATETABLENO Visual Warehouse はターゲット表を作成しません。	ビジネス・ビュー: Visual Warehouse の作成した表

BusinessView.tag

表 130. BusinessView.tag のトークン (続き)

トークン	説明	許可される値	ウィンドウまたはノートブック: フィールド
*BVType	ビジネス・ビューのタイプ。 このトークンは必須です。	以下の値の 1 つ: ISV_BVType_EditionedAppend ビジネス・ビューを実行するたびに、データの新しい版をターゲット表に追加します。 ISV_BVType_Full_Replace ビジネス・ビューを実行するたびに、ターゲット表のすべてのデータを置換します。 ISV_BVType_Uneditioned_Append ビジネス・ビューを実行するたびに、既存のデータに新規のデータを追加します。 ISV_BVType_VWP_Population Visual Warehouse プログラムを使用してデータを管理します。	なし

表 130. BusinessView.tag のトークン (続き)

トークン	説明	許可される値	ウィンドウまたはノートブック: フィールド
*BVSQLError	SQL 警告コードが発行された場合にビジネス・ビューの処理を続けるかどうかを示す設定。 このトークンは必須です。	以下の値の 1 つ: ISV_BVSQLError_OK SQL 警告コードが発行された場合に、ビジネス・ビューは正常に処理を続けます。 ISV_BVSQLError_Warning SQL 警告コードが発行された場合に、ビジネス・ビューは警告を出して処理を続けます。 ISV_BVSQLError_Error SQL 警告コードが発行された場合に、ビジネス・ビューは失敗します。	ビジネス・ビュー: SQL 警告処理オプション
関係パラメーター			
*SecurityGroup	インポート中のすべてのオブジェクトを作成するセキュリティ・グループ。 このトークンは必須であり、デフォルトのセキュリティ・グループを指定しなければなりません。	デフォルトのセキュリティ・グループとして ISV_DEFAULTSECURITYGROUP。	ビジネス・ビュー: セキュリティ・グループの更新
*SubjectArea	ビジネス・ビューのグループの名前。 このトークンは必須です。	最長 80 バイトのテキスト・ストリング。	サブジェクト: 名前

BusinessView.tag

表 130. BusinessView.tag のトークン (続き)

トークン	説明	許可される値	ウィンドウまたはノートブック: フィールド
*AgentSite	<p>ビジネス・ビューに使用するエージェント・サイト: 新規のエージェント・サイト、またはデフォルトのエージェント・サイトのいずれか。</p> <p>このトークンは必須ですが、デフォルトのエージェント・サイトを指定することもできます。</p>	<p>最長 80 バイトのテキスト・ストリング。</p> <p>デフォルトのエージェント・サイトは、ISV_DEFAULTAGENTSITE を指定します。</p>	ビジネス・ビュー: エージェント・サイト
*CurrentCheckPointID++	<p>0 から始まり、トークンに置換されるたびに増加する索引。</p> <p>このトークンは必須です。</p>	数値。	なし

値の例

表131 は、各トークンに指定するメタデータの種別を例示するために、各トークンの値の例を示しています。

表 131. BusinessView.tag トークンの値の例

トークン	値の例
*BVName	Revenue_by_Geography_7
*BVDescription	This business view will extract Geography 7 data and write it to an UDB table
*BVNotes	The Revenue for Geography 7 data comes from four source Oracle tables.
*BVDataNotPresent	ISV_BVDataNotPresent_Warning1
*BVSelectStatementGenerated	ISV_BVSELECTSTATEMENTNO

表 131. BusinessView.tag トークンの値の例 (続き)

トークン	値の例
*BVSelectStatement	"SELECT * FROM IWH.REVENUE_BY_GEOGRAPHY7"
*BVContact	Greg Holland
*BVExternalPopulation	ISV_BVEXTERNALNO
*BVCreateTargetTable	ISV_CREATETABLEYES
*BVType	ISV_BVType_VWP_Population
*BVSQLWarning	ISV_BVSQLWarning_Error
*SecurityGroup	ISV_DEFAULTSECURITYGROUP
*Subject Area	Group of business views generated for the partner tool
*AgentSite	My agent site
*CurrentCheckPointID++	10

BusinessViewInputTable.tag

このテンプレートを 사용하여、ビジネス・ビューとそのソース表との関係を定義します。テンプレートをソース表の固有のインスタンスごとに再使用することによって、複数のソース表をビジネス・ビューに関連付けることができます。

このテンプレートは、以下のタイプのビジネス・ビューに含めなければなりません。

- エディションの追加 (*BVType は ISV_BVType_EditionedAppend)
- 既存データの置換 (*BVType は ISV_BVType_Full_Replace)
- エディションのないデータの追加 (*BVType は ISV_BVType_Uneditioned_Append)

このテンプレートは、Visual Warehouse プログラムを使用するビジネス・ビュー (*BVType は ISV_BVType_VWP_Population) では任意選択です。

トークン

322ページの表132 は、このテンプレート内の各トークンに関する情報を示しています。

BusinessViewInputTable.tag

表 132. *BusinessViewInputTable.tag* のトークン. このテンプレートには、関係パラメーターだけが含まれます。

トークン	説明	許可される値	ウィンドウまたはノートブック: フィールド
<i>*BVName</i>	ビジネス・ビューの名前。 その名前は、Visual Warehouse コントロール・データベース内で固有でなければなりません。 このトークンは必須です。	最長 80 バイトのテキスト・ストリング。	ビジネス・ビュー: ビジネス名
<i>*DatabaseName</i>	表を含むデータベースの名前。 このトークンは必須です。	最長 80 バイトのテキスト・ストリング。	情報リソース: データベース
<i>*TableOwner</i>	表の所有者、高レベル修飾子、集合、またはスキーマ。 所有者は ODBC の規則に従う有効な修飾子でなければなりません。 このトークンは必須です。	最長 15 バイトのテキスト・ストリング。	表: 名前 ビジネス・ビュー: 表名修飾子
<i>*TablePhysicalName</i>	データベース・マネージャーまたはファイル・システムに定義された、物理表名。 このトークンは必須です。	最長 80 バイトのテキスト・ストリング。	表: 名前 ビジネス・ビュー: データベース表名

表 132. *BusinessViewInputTable.tag* のトークン (続き). このテンプレートには、関係パラメーターだけが含まれます。

トークン	説明	許可される値	ウィンドウまたはノートブック: フィールド
*CurrentCheckPointID++	0 から始まり、トークンに置換されるたびに増加する索引。 このトークンは必須です。	数値。	なし

値の例

表 133 は、各トークンに指定するメタデータの種類を例示するために、各トークンの値の例を示しています。

表 133. *BusinessViewInputTable.tag* トークンの値の例

トークン	値の例
*BVName	Revenue_by_Geography_1
*DatabaseName	Operational_system_files
*TableOwner	ISV_DEFAULTVALUE
*TablePhysicalName	z:¥geography¥regions¥geo1.file
*CurrentCheckPointID++	13

BusinessViewOutputTable.tag

このテンプレートを使用して、ビジネス・ビューとその出力ターゲットとの間の関係を定義します。

このテンプレートは、以下のタイプのビジネス・ビューに含めなければなりません。

- エディションの追加 (*BVType は ISV_BVType_EditionedAppend)
- 既存データの置換 (*BVType は ISV_BVType_Full_Replace)
- エディションのないデータの追加 (*BVType は ISV_BVType_Uneditioned_Append)

このテンプレートは、Visual Warehouse プログラムを使用するビジネス・ビュー (*BVType は ISV_BVType_VWP_Population) では任意選択です

BusinessViewOutputTable.tag

トークン

表134 は、このテンプレート内の各トークンに関する情報を示しています。

表 134. *BusinessViewOutputTable.tag* のトークン. このテンプレートには、関係パラメーターだけが含まれません。

トークン	説明	許可される値	ウィンドウまたはノートブック: フィールド
*BVName	ビジネス・ビューの名前。 このトークンは必須です。	最長 80 バイトのテキスト・ストリング。	ビジネス・ビュー: ビジネス名
*DatabaseName	表を含むデータベースの名前。 このトークンは必須です。	最長 80 バイトのテキスト・ストリング。	情報リソース: データベース
*TableOwner	表の所有者、高レベル修飾子、集合、またはスキーマ。 このトークンは必須です。	最長 15 バイトのテキスト・ストリング。	表: 名前 ビジネス・ビュー: 表名修飾子
*TablePhysicalName	データベース・マネージャーまたはファイル・システムに定義された、物理表名。 このトークンは必須です。	最長 80 バイトのテキスト・ストリング。	表: 名前 ビジネス・ビュー: データベース表名
*CurrentCheckpointID++	0 から始まり、トークンに置換されるたびに増加する索引。 このトークンは必須です。	数値。	なし

値の例

表135 は、各トークンに指定するメタデータの種類を例示するために、各トークンの値の例を示しています。

表 135. BusinessViewOutputTable.tag トークンの値の例

トークン	値の例
*BVName	Revenue_by_Geography_7
*DatabaseName	Finance Warehouse
*TableOwner	DB2ADMIN
*TablePhysicalName	GEOGRAPHY
*CurrentCheckPointID++	14

BusinessViewVWPOutputTable.tag

このテンプレートを使用して、Visual Warehouse プログラムを使用するビジネス・ビューと Visual Warehouse プログラムの出力ターゲットとの間の関係を定義します。

トークン

表136 は、このテンプレート内の各トークンに関する情報を示しています。

表 136. BusinessViewVWPOutputTable.tag のトークン. このテンプレートには、関係パラメーターだけが含まれます。

トークン	説明	許可される値	ウィンドウまたはノートブック: フィールド
*BVName	ビジネス・ビューの名前。 このトークンは必須です。	最長 80 バイトのテキスト・ストリング。	ビジネス・ビュー: ビジネス名
*DatabaseName	表を含むデータベースの名前。 このトークンは必須です。	最長 80 バイトのテキスト・ストリング。	情報リソース: データベース

BusinessViewVWPOutputTable.tag

表 136. *BusinessViewVWPOutputTable.tag* のトークン (続き). このテンプレートには、関係パラメーターだけが含まれます。

トークン	説明	許可される値	ウィンドウまたはノートブック: フィールド
* <i>TableOwner</i>	表の所有者、高レベル修飾子、集合、またはスキーマ。 このトークンは必須です。	最長 15 バイトのテキスト・ストリング。	表: 名前 ビジネス・ビュー: 表名修飾子
* <i>TablePhysicalName</i>	データベース・マネージャーまたはファイル・システムに定義された、物理表名。 このトークンは必須です。	最長 80 バイトのテキスト・ストリング。	表: 名前 ビジネス・ビュー: データベース表名
* <i>CurrentCheckPointID++</i>	0 から始まり、トークンに置換されるたびに増加する索引。 このトークンは必須です。	数値。	なし

値の例

表137 は、各トークンに指定するメタデータの種類を例示するために、各トークンの値の例を示しています。

表 137. *VWPOutputTable.tag* トークンの値の例

トークン	値の例
* <i>BVName</i>	Revenue_by_Geography_7
* <i>DatabaseName</i>	Finance Warehouse
* <i>TableOwner</i>	DB2ADMIN
* <i>TablePhysicalName</i>	GEOGRAPHY
* <i>CurrentCheckPointID++</i>	15

ConcurrentCascade.tag

このテンプレートを使用して、Visual Warehouse が同時に 2 つのビジネス・ビューを開始することを指定します。このテンプレートが必要なのは、複数のビジネス・ビューを同時に開始したい場合だけです。

トークン

表138 は、このテンプレート内の各トークンに関する情報を示しています。

表 138. *ConcurrentCascade.tag* のトークン. このテンプレートには、関係パラメーターだけが含まれます。

トークン	説明	許可される値	ウィンドウまたはノートブック: フィールド
*BVName	ビジネス・ビューの名前。 このトークンは必須です。	最長 80 バイトのテキスト・ストリング。	ビジネス・ビュー: ビジネス名
*ConcurrentBVName	他のビジネス・ビューと同時に開始したいビジネス・ビューの名前。 このトークンは必須です。	最長 80 バイトのテキスト・ストリング。	ビジネス・ビュー: 同時に開始: ビジネス・ビュー名
*CurrentCheckPointID++	0 から始まり、トークンに置換されるたびに増加する索引。 このトークンは必須です。	数値。	なし

値の例

表139 は、各トークンに指定するメタデータの種類を例示するために、各トークンの値の例を示しています。

表 139. *ConcurrentCascade.tag* トークンの値の例

トークン	値の例
*BVName	Revenue_by_Geography_7
*ConcurrentBVName	Revenue_by_Geography_6

ConcurrentCascade.tag

表 139. *ConcurrentCascade.tag* トークンの値の例 (続き)

トークン	値の例
* <i>CurrentCheckPointID</i> ++	16

PostCascade.tag

このテンプレートを使用して、Visual Warehouse が指定したビジネス・ビューの処理の終了後に別のビジネス・ビューを開始することを示します。このテンプレートが必要なのは、複数のビジネス・ビューを連鎖関係にリンクしたい場合だけです。

トークン

表140 は、このテンプレート内の各トークンに関する情報を示しています。

表 140. *PostCascade.tag* のトークン. このテンプレートには、関係パラメーターだけが含まれます。

トークン	説明	許可される値	ウィンドウまたはノートブック: フィールド
* <i>BVName</i>	次のビジネス・ビューを開始する前に処理を終了するビジネス・ビューの名前。 このトークンは必須です。	最長 80 バイトのテキスト・ストリング。	ビジネス・ビュー: ビジネス名
* <i>PostBVName</i>	別のビジネス・ビューが処理を終了したときに処理を開始するビジネス・ビューの名前。 このトークンは必須です。	最長 80 バイトのテキスト・ストリング。	ビジネス・ビュー: 開始: ビジネス・ビュー名
* <i>CurrentCheckPointID</i> ++	0 から始まり、トークンに置換されるたびに増加する索引。 このトークンは必須です。	数値。	なし

値の例

表141 は、各トークンに指定するメタデータの種類を例示するために、各トークンの値の例を示しています。

表 141. PostCascade.tag トークンの値の例

トークン	値の例
*BVName	Revenue by geography 7
*PostBVName	Revenue for all geographies
*CurrentCheckPointID++	17

VWPPProgramInstance.tag

このテンプレートを使用して、Visual Warehouse プログラムの定義を特定のビジネス・ビューが使用するために変更します。このテンプレートは、Visual Warehouse プログラムを使用するビジネス・ビューごとに必要です。

このテンプレートを使用する前に、Visual Warehouse プログラムの基本定義を VWPPProgramTemplate.tag に定義しなければなりません (103ページを参照)。このテンプレートは、基本 Visual Warehouse プログラム定義 (VWPPProgramTemplate.tag) への関係、および Visual Warehouse プログラムを使用するビジネス・ビューへの関係を定義します。

トークン

表142 は、このテンプレート内の各トークンに関する情報を示しています。

表 142. VWPPProgramInstance.tag のトークン

トークン	説明	許可される値	ウィンドウまたはノートブック: フィールド
エンティティ・パラメーター			
*VWPInstanceNotes	Visual Warehouse プログラムおよびその機能についての長い説明 このトークンは任意指定です。	最長 32700 バイトのテキスト・ストリング。	なし

VWPPProgramInstance.tag

表 142. VWPPProgramInstance.tag のトークン (続き)

トークン	説明	許可される値	ウィンドウまたはノートブック: フィールド
*VWPPProgramInstanceKey	<p>このプログラムのインスタンスを固有に識別するキー。このキーは、タグ言語ファイル内の他のすべてのキーに対して固有でなければなりません。</p> <p>ヒント</p> <p>: VWPPProgramInstance.tag テンプレートの処理を終了してから、キーの値を増加させてください。</p> <p>このトークンは必須です。</p>	数値。	なし
関係パラメーター			
*BVName	<p>ビジネス・ビューの名前。</p> <p>このトークンは必須です。</p>		ビジネス・ビュー: ビジネス名
*VWPPProgramTemplateName	<p>この Visual Warehouse プログラム・インスタンスに対する親 Visual Warehouse プログラム・テンプレートの名前。</p> <p>このトークンは必須です。</p>	最長 80 バイトのテキスト・ストリング。	プログラム: ビジネス名
*CurrentCheckPointID++	<p>0 から始まり、トークンに置換されるたびに増加する索引。</p> <p>このトークンは必須です。</p>	数値。	なし

値の例

表143 は、各トークンに指定するメタデータの種類を例示するために、各トークンの値の例を示しています。

表 143. VWPPProgramInstance.tag トークンの値の例

トークン	値の例
<i>*VWPInstanceNotes</i>	This program exports data from the Geography database.
<i>*VWPPProgramInstanceKey</i>	070000
<i>*BVName</i>	Revenue by geography
<i>*VWPPProgramTemplateName</i>	My partner program
<i>*CurrentCheckPointID++</i>	11

VWPProgramInstance.tag

付録C. データウェアハウスセンターで使用するプログラムを独自に作成する

実行可能、バッチ・プログラム、またはダイナミック・リンク・ライブラリーのプログラム・タイプのいずれかをサポートする任意の言語で、データウェアハウスセンター プログラムを作成することができます。

プログラムのプログラム・タイプが実行可能、コマンド・ファイル、またはダイナミック・リンク・ライブラリーである場合、それはエージェント・サイトに存在しなければなりません。データウェアハウスセンター エージェントはスケジュールされた時刻にプログラムを開始します。Windows NT および Windows 2000 では、デフォルトではエージェントがシステム・プロセスとして実行します。プログラムはユーザー ID を必要とするリソースまたはプログラムにアクセスできません。さらに、プログラムがアクセスする必要がある環境変数はシステム変数でなければなりません。

データウェアハウスセンター サーバー、ロガー、およびエージェント・デーモン・プロセスをユーザー・プロセスとして実行するように変更するためには、以下のように行います。

1. 「コントロール パネル」フォルダーの「サービス (Services)」アイコンをダブルクリックします。
2. 「エージェント・サービス (Agent service)」を停止します。
3. 「エージェント・サービス (Agent service)」を選択して、「起動 (startup)」をクリックします。
4. 「このアカウント (This Account)」をクリックします。
5. 「このアカウント (This Account)」フィールドの後の押しボタンをクリックして、ユーザー ID を選択します。

ユーザーID には Windows NT または Windows 2000 での管理者権限、および必要なネットワーク・ドライブへの権限が必要です。

6. ユーザー ID に対するパスワードを 2 回タイプします。
7. 「OK」をクリックします。
8. ワークステーションを再始動します。

Object REXX を使用するプログラムを作成する場合、以下の手順を完了してこれらのプログラムが Windows NT または Windows 2000 上で実行するようにします。

1. データウェアハウスセンター エージェントまたはサーバー・サービスを、Windows NT または Windows 2000 デスクトップと対話できるシステム・プロセスとして定義します。
 - a. 「サービス (Service)」リストから、エージェントまたはサーバー・サービスを選択します。
 - b. 「起動 (startup)」をクリックします。
 - c. 「システム・アカウント (System Account)」をクリックします。
 - d. 「サービスがデスクトップと対話するようにする (Allow Service to Interact with Desktop)」チェック・ボックスを選択します。
2. エージェントまたはサーバーがプログラムを開始する前に、Object REXX 環境を初期化します。コマンド行から任意の Object REXX プログラムを実行することにより、環境を初期化できます。
3. Object REXX プログラムが DB2 CONNECT ステートメントを発行する場合、以下の例に示すように、ステートメントにユーザー ID およびパスワードが含まれることを確認してください。

```
DB2 CONNECT TO testdb USER vwadmin USING vwpass
```

パラメーターを渡す

実行時に、データウェアハウスセンターはプログラムへの入力として渡すためのコマンド行パラメーター・リストを生成します。いつでも可能なときに、プログラムをステップで使用する前にそれをコマンド行からテストしてください。

例: データウェアハウスセンター プログラム VW 5.2 DB2 ロード置換 (VWPLOADR) は、ファイルからデータを選択してデータベースにロードします。以下のパラメーターを使用します。

- ソース・ファイル名
- ターゲット・データベース名
- ターゲット・データベースのユーザー ID
- ターゲット・データベースのパスワード
- ターゲット表名
- 列の区切り文字

プログラムは、図75 に示されているようにパラメーターを受け取ります。

```
char * sourceFile;
sourceFile = argv[1];
char * dbName;
dbName = argv[2];
char * dbUser;
dbUser = argv[3];
char * dbPassword;
dbPassword = argv[4];
char * dbTable;
dbTable = argv[5];
char * fileMod;
if(argc>6) fileMod = argv[6];
else fileMod = NULL;
```

図 75. コマンド行からパラメーターを読み取る

図76 に示されているように、プログラムはターゲット・パラメーターを使用してターゲット・データベースに接続します。

```
rc = SQLConnect (hdbc, (SQLCHAR *)dbName, SQL_NTS,
                (SQLCHAR *)dbUser, SQL_NTS, /* UID */
                (SQLCHAR *)dbPassword, SQL_NTS); /* Password */
```

図 76. ターゲット・データベースに接続する

その後、プログラムは DB2 ロード・ユーティリティーを使用してデータをデータベースにロードします。

状況情報を戻す

データウェアハウスセンター プログラムの実行後に、戻りコードがプログラムを使用したステップに戻されなければなりません。戻りコードは正の整数でなければなりません。ご使用のプログラムが戻りコードを戻さない場合、プログラムを使用しているステップは失敗します。 **Error RC1** の値が 8410 である場合、データウェアハウスセンターは「ログ詳細 (Log Details)」ウィンドウの「**エラー RC2 (Error RC2)**」フィールドに戻りコードを表示します。

データウェアハウスセンター プログラムは、以下に示す追加の状況情報をデータウェアハウスセンターに戻すことができます。

- 別の戻りコード。これはデータウェアハウスセンター プログラムから戻されるコードと同じでも異なっても可能です。

- データウェアハウスセンターが戻りコードを警告として処理するように指示するフラグ。プログラムがこのフラグを立てると、このプログラムを使用するステップは「進行中の操作作業 (Operations Work in Progress)」ウィンドウに「警告 (Warning)」状況を示します。
- 「ログ・ビューアーの詳細 (Log Viewer Details)」ウィンドウの「システム・メッセージ (System Message)」フィールドに表示されるメッセージ。
- プログラムが処理したデータの行数。
データウェアハウスセンターは、ステップについての数を「ログ・ビューアーの詳細 (Log Viewer Details)」ウィンドウに表示します。
- プログラムが処理したデータのバイト数。
データウェアハウスセンターは、ステップについての数を「ログ・ビューアーの詳細 (Log Viewer Details)」ウィンドウに表示します。
- データウェアハウスセンターが「ログ・ビューアーの詳細 (Log Viewer Details)」ウィンドウの「SQL 状況 (SQL state)」フィールドに表示する SQLSTATE 戻りコード。

データウェアハウスセンター エージェントは、追加の状況情報をウェアハウス・サーバーに転送します。

情報をデータウェアハウスセンターに転送する

追加の状況情報をウェアハウス・エージェントに転送するために、プログラムは追加の状況情報を含むフィードバック・ファイルと呼ばれるファイルを作成しなければなりません。フィードバック・ファイルのパスおよびファイル名は、VWP_LOG 環境変数の値でなければなりません。(ファイル名は *processid.log* で、*processid* はエージェント・プロセスの ID です。) エージェントは、プログラムを呼び出す前に VWP_LOG を設定します。プログラムの実行が終了した後、エージェントはフィードバック・ファイルが存在するかどうかを調べます。存在する場合、エージェントはそのファイルを処理します。存在しない場合、エージェントは何も行いません。プログラムがファイルを作成できない場合、それは実行を続ける必要があります。

フィードバック・ファイルの形式

プログラムは追加の状況情報をフィードバック・ファイルに任意の順序で書き込むことができますが、情報を識別するために以下の形式を使用しなければなりません。以下のリストに示すように、戻される各項目を開始タグ <tag> および終了タグ </tag> で囲みます。それぞれの開始タグの次には対応する終了タグがなければなりません。2 つの開始タグを連続して使用することはできません。たとえば、以下のタグ形式は有効です。

<RC>...</RC>...<MSG>...</MSG>

以下の組み込みタグ形式は無効です。

<RC>...<MSG>...</RC>...</MSG>

フィードバック・ファイル内に以下の情報を指定できます。

戻りコード

<RC>*return code*</RC>。 *return code* は正の整数です。

戻りコード警告標識

<WARNING>1</WARNING> は、戻りコード警告標識をオンに設定します。

データウェアハウスセンター システム・メッセージ

<MSG>*message text*¥n</MSG>

message text

1 つ以上のメッセージのテキスト

¥n 改行文字。複数のメッセージがある場合、この文字を各メッセージの終わりに含めます。

コメント

<COMMENT>*comment text*</COMMENT>。 *comment text* は、コメントのテキストです。

処理されたデータの行数

<ROWS>*number of rows*</ROWS>。 *number of rows* は正の整数です。

処理されたデータのバイト数

<BYTES>*number of bytes*</BYTES>。 *number of bytes* は正の整数です。

SQLSTATE

<SQLSTATE>*sqlstate string*</SQLSTATE>。 *sqlstate string* は長さが 0 より大きく、5 桁以下の任意のストリングです。

338ページの図77 は、フィードバック・ファイルの例を示しています。

```
<RC> 20</RC>
<ROWS>2345</ROWS>
<MSG>The parameter type is not correct</MSG>
<COMMENT> Please supply the correct parameter type (PASSWORD
NOTREQUIRED, GETPASSWORD, ENTERPASSWORD)</COMMENT>
<BYTES> 123456</BYTES>
<WARNING> 1</WARNING>
<SQLSTATE>12345</SQLSTATE>
```

図 77. フィードバック・ファイルの例

フィードバックによってステップ状況を判別する方法

Log Viewer に表示されるプログラムの戻りコードおよびステップ状況には、さまざまなものがあります。それらはプログラムによって設定される以下の値に依存します。

- プログラムが戻す戻りコードの値
- フィードバック・ファイルが存在するかどうか
- フィードバック・ファイル内の戻りコードの値
- 警告標識がオンに設定されているかどうか

339ページの表144 は、これらの値の可能な組み合わせと、それぞれの組み合わせから生じる結果とを示しています。

表 144. フィードバック・ファイルの状態および結果

状態			結果		
			ステップ状況 ¹	エラー RC1 および RC2 の値	
データウェア ハウスセンタ ー プログラム の戻りコード が 0	フィードバック・ファイルが存在しない ²		成功	RC1 = 0; RC2 = 0	
	フィードバック・ファイル が存在する ²	フィードバック・ファイル 内の <RC> の 値が 0 ³	<WARNING> がフィードバ ック・ファイ ル内で設定さ れていない	成功	RC1 = 0; RC2 = 0
			フィードバック・ファイル 内の <WARNING> の値が 1	警告	RC1 = 0; RC2 = 0
		フィードバック・ファイル 内の <RC> の 値が 0 ではな い ³	<WARNING> がフィードバ ック・ファイ ル内で設定さ れていない	失敗	RC1 = 8410 (プログラムは 失敗した)、 RC2 = フィー ドバック・フ ァイル内の <RC> の値
			フィードバック・ファイル 内の <WARNING> の値が 1	警告	RC1 = 0、 RC2 = フィー ドバック・フ ァイル内の <RC> の値

表 144. フィードバック・ファイルの状態および結果 (続き)

状態				結果	
				ステップ状況 ¹	エラー RC1 および RC2 の値
データウェアハウスセンター プログラムの戻りコードがゼロではない	フィードバック・ファイルが存在しない ²			失敗	RC1 = 8410 (データウェアハウスセンター プログラムは失敗した)、RC2 = データウェアハウスセンター プログラムによって戻されるコード
	フィードバック・ファイルが存在する ²	フィードバック・ファイル内の <RC> の値が 0 ³	<WARNING> がフィードバック・ファイル内で設定されていない	成功	RC1 = 0; RC2 = 0
			フィードバック・ファイル内の <WARNING> の値が 1	警告	RC1 = 0; RC2 = 0
	フィードバック・ファイル内の <RC> の値がゼロではない	<WARNING> がフィードバック・ファイル内で設定されていない		失敗	RC1 = 8410 (データウェアハウスセンター プログラムは失敗した)、RC2 = データウェアハウスセンター プログラムによって戻されるコード
			フィードバック・ファイル内の <WARNING> の値が 1	警告	RC1 = 0、RC2 = フィードバック・ファイル内の <RC> の値

表 144. フィードバック・ファイルの状態および結果 (続き)

状態	結果	
	ステップ状況 ¹	エラー RC1 および RC2 の値
<p>注:</p> <ol style="list-style-type: none"> 「進行中の作業 (Work in Progress)」ウィンドウに表示される、ステップの処理状況。 プログラムの戻りコードが 0 であってもなくても、データウェアハウスセンターはフィードバック・ファイルが存在するかどうかを検査します。 データウェアハウスセンターは常に、フィードバック・ファイル内の <RC> の値を「ログ詳細 (Log Details)」ウィンドウ内の 「RC2」 フィールドの値として表示します。 		

特記事項

本書において、日本では発表されていない IBM 製品 (機械およびプログラム)、プログラミングまたはサービスについて言及または説明する場合があります。しかし、このことは、弊社がこのような IBM 製品、プログラミングまたはサービスを、日本で発表する意図があることを必ずしも示すものではありません。本書で、IBM ライセンス・プログラムまたは他の IBM 製品に言及している部分があっても、このことは当該プログラムまたは製品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。これらのプログラムまたは製品に代えて、IBM の知的所有権を侵害することのない機能的に同等な他社のプログラム、製品またはサービスを使用することができます。ただし、IBM によって明示的に指定されたものを除き、これらのプログラムまたは製品に関連する稼働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。

IBM および他社は、本書で説明する主題に関する特許権 (特許出願を含む)、商標権、または著作権を所有している場合があります。本書は、これらの特許権、商標権、および著作権について、本書で明示されている場合を除き、実施権、使用権等を許諾することを意味するものではありません。実施権、使用権等の許諾については、下記の宛先に、書面にてご照会ください。

〒106-0032 東京都港区六本木 3 丁目 2-31
AP 事業所
IBM World Trade Asia Corporation
Intellectual Property Law & Licensing

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。

本書に含まれる情報には、技術的に不正確なもの、または誤植が含まれる場合があります。これらに対する変更は、定期的に行われます。これらの変更は、資料の改訂版に含まれます。IBM は、本書で説明している製品、プログラムに対して、予告なく改良、変更を加える場合があります。

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、お客様になんら義務も負わせない適切な方法で、使用もしくは配布することがあります。

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 Canada Limited
Office of the Lab Director
1150 Eglinton Ave. East
North York, Ontario
M3C 1H7
CANADA

本プログラムに関する上記の情報は、適切な条件の下で使用することができませんが、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。

本書に含まれるパフォーマンス・データは、制御された環境下で決定されています。したがって、その他の稼働環境で得られる結果とは、かなり異なる可能性もあります。一部の測定値は、開発中のシステムを使用している場合があり、これらの測定値が一般的に提供可能なシステムで同様の数値になることを保証するものではありません。さらに、一部の測定値が推定されたものもあります。実測値と異なる場合があります。本書のユーザーは、使用される特定の環境での該当データを確認してください。

IBM 以外の製品については、当該製品の提供者から直接、出版されている資料または一般公開されている情報から入手しました。IBM は、これらの製品についてはテストを行っておらず、これらの IBM 以外の製品に関する性能、互換性またはその他の主張について確認することはできません。IBM 以外の製品の機能に対する質問は、それぞれの製品提供者にお問い合わせください。

IBM の将来の方向性または意図については、予告なしに変更または中止する場合があります。IBM の目的および目標のみ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が、これは説明に具体性を与えるために記載されたものであり、それらの例には、個人、企業、ブランドの、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。それらの名前はすべて架空のものであり、また名称や住所が類似する企業が実在しても、それは偶然に過ぎません。

著作権：

本書に含まれる情報には、サンプル・アプリケーション・プログラムがソース言語の形式で含まれており、様々な、オペレーティング・プラットフォームでのプログラミング技法を示しています。お客様は、これらのサンプル・プログラムが書かれているオペレーティング・プラットフォームでアプリケーション・プログラミング・インターフェースが実行可能となるためのアプリケーション・プログラムを開発、使用、販売または配布もしくは転送する目的のためだけに、サンプル・プログラムを、IBM に対する別途料金を支払うことなく、複製、変更、配布または転送することができます。これらのサンプルは、すべての条件下で十分にテストを行っていません。したがって、IBM は、これらのプログラムの信頼性、実用性または機能について、いかなる保証も負いません。

サンプル・プログラムまたはその改変版の複製物には、全部複製か部分複製かを問わず、次の著作権表示を必ず行うものとします。

© (お客様の会社名) (西暦年). このコードの一部は IBM Corp. のサンプル・プログラムから取られています。 © Copyright IBM Corp. _年_. All rights reserved.

商標

以下は、IBM Corporation の商標です。

ACF/VTAM	IBM
AISPO	IMS
AIX	IMS/ESA
AIX/6000	LAN DistanceMVS
AIXwindows	MVS/ESA
AnyNet	MVS/XA
APPN	Net.Data
AS/400	OS/2
BookManager	OS/390
CICS	OS/400
C Set++	PowerPC
C/370	QBIC
DATABASE 2	QMF
DataHub	RACF
DataJoiner	RISC System/6000
DataPropagator	RS/6000
DataRefresher	S/370
DB2	SP
DB2 Connect	SQL/DS
DB2 Extenders	SQL/400
DB2 OLAP Server	System/370
DB2 Universal Database	System/390
Distributed Relational Database Architecture	SystemView VisualAge
DRDA	VM/ESA
eNetwork	VSE/ESA
Extended Services	VTAM
FFST	WebExplorer
First Failure Support Technology	WIN-OS/2

次のものは、他社の商標または登録商標です。

Tivoli および NetView は、米国およびその他の国における Tivoli Systems Inc. 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。

Java およびすべての Java 関連の商標およびロゴは Sun Microsystems, Inc. の米国およびその他の国における商標または登録商標です。

UNIX は、The Open Group がライセンスしている米国およびその他の国における登録商標です。

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または登録商標です。

参考文献

データウェアハウスセンターの使用方法については、オンライン・ヘルプを参照してください。データウェアハウスセンターには特定のウィンドウに関するヘルプ、およびウェアハウス・ソースの作成やステップなどの汎用タスクに関するヘルプが備わっています。

データウェアハウスセンターに関連した IBM 製品についての情報は、<http://www.software.ibm.com/data/> の IBM Data Management Web サイトをご覧ください。

データウェアハウスセンター ライブラリーには、以下の資料が含まれます。

IBM DB2: DB2 ウェアハウス・マネージャー インストールの手引き (SC88-8572)

IBM DB2: Messages and Reason Codes (データウェアハウスセンター フォルダに含まれる HTML ブック)

IBM DB2: 情報カタログ・マネージャー 管理の手引き (SC88-8547)

IBM DB2: 情報カタログ・マネージャー プログラミングの手引きおよび解説書 (SC88-8549)

IBM DB2 OLAP Server: DB2 OLAP Server 使用の手引き (SD88-7081)

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]

アプリケーション・データ、サンプル・オブジェクト・タイプ 177
イメージまたはグラフィックス、サンプル・オブジェクト・タイプ 218
インターネット文書、サンプル・オブジェクト・タイプ 219
ウェアハウス・ソース・メタデータ
関係 120
プロパティ 115
ウェアハウス・ターゲット・メタデータ
関係 120
プロパティ 115
ウェアハウス・データベース 14
エージェント 13
エージェント・サイト
疑似コード 23
定義 14
提供する値 23
テンプレート 22
エコー (ECH) ファイル 281
エレメント、サンプル・オブジェクト・タイプ 188
オーディオ・クリップ、サンプル・オブジェクト・タイプ 214
オブジェクト
COLUMN
関係 130
プロパティ 127
DATABASE
関係 120
プロパティ 115

オブジェクト (続き)

TABLES

関係 126
プロパティ 121

オブジェクト・タイプ

関係 155

基礎的区分

イメージまたはグラフィックス、サンプル情報カタログで提供される 218

インターネット文書、サンプル情報カタログで提供される 219

オーディオ・クリップ、サンプル情報カタログで提供される 214

サンプル情報カタログで提供される 214

図表、サンプル情報カタログで提供される 215

スプレッドシート、サンプル情報カタログで提供される 222

テキスト・ベース・レポート、サンプル情報カタログで提供される 224

ビデオ・クリップ、サンプル情報カタログで提供される 225

プレゼンテーション、サンプル情報カタログで提供される 221

文書、サンプル情報カタログで提供される 217

Lotus Approach 照会、サンプル情報カタログで提供される 220

グループ化区分

アプリケーション・データ、サンプル情報カタログで提供される 177

オブジェクト・タイプ (続き)

グループ化区分 (続き)

エレメント、サンプル情報カタログで提供される 188

サブスキーマ、サンプル情報カタログで提供される 210
サンプル情報カタログで提供される 176

スタースキーマ、サンプル情報カタログで提供される 209

多次元データベース、サンプル情報カタログで提供される 202

多次元データベース内の次元、サンプル情報カタログで提供される 185

多次元データベース内のメンバー、サンプル情報カタログで提供される 200

データベース、サンプル情報カタログで提供される 183

トランスフォーメーション、サンプル情報カタログで提供される 212

ビジネス・サブジェクト・エリア、サンプル情報カタログで提供される 180

ファイル、サンプル情報カタログで提供される 190

リレーショナル表および視点、サンプル情報カタログで提供される 206

列またはフィールド、サンプル情報カタログで提供される 180

レコード、サンプル情報カタログで提供される 204

DWC プロセス、サンプル情報カタログで提供される 187

オブジェクト・タイプ (続き)

IMS セグメント、サンプル情報カタログで提供される 198

IMS データベース定義 (DBD)、サンプル情報カタログで提供される 193

IMS プログラム仕様ブロック (PSB)、サンプル情報カタログで提供される 196

IMS プログラム制御ブロック (PCB)、サンプル情報カタログで提供される 195

コンタクト区分
サンプル情報カタログ 226
連絡先、サンプル情報カタログで提供される 226

サポート区分
オンライン資料、サンプル情報カタログで提供される 231
オンライン・ニュース・サービス、サンプル情報カタログで提供される 230
サンプル情報カタログで提供される 229
情報カタログ・マネージャー・ニュース、サンプル情報カタログで提供される 229

辞書区分
サンプル情報カタログ 227
用語集エントリ、サンプル情報カタログで提供される 227

情報カタログでの確立 156

接続区分
定義済みのコメント・オブジェクト・タイプ 233
プログラム区分、定義済みのプログラム・オブジェクト・タイプ 232

オブジェクト・タイプ間の関係 155

オプション
ACTION.RELATION 262
ADD
ACTION.OBJINST 244

オプション (続き)

ADD (続き)
ACTION.OBJTYPE 249, 265
ACTION.RELATION 253, 262

APPEND 250

DELETE 262
ACTION.OBJINST 244
ACTION.OBJTYPE 250
ACTION.RELATION 254
OBJINSTの 260
DELETE_EXT 251
DELETE_TREE_ALL
ACTION.OBJINST 245
OBJINST の 260
DELETE_TREE_REL
ACTION.OBJINST 246
OBJINST の 260

MERGE
ACTION.OBJINST 247
ACTION.OBJTYPE 251, 265

UPDATE
ACTION.OBJINST 248, 261
ACTION.OBJTYPE 252

オンライン資料、サンプル・オブジェクト・タイプ 231

オンライン・ニュース・サービス、サンプル・オブジェクト・タイプ 230

[力行]

該当しないシンボル
MDIS エクスポートの際に指定する 50

各国語サポート (NLS) 241

可変値 242

キーワード
各国語についてサポートされない 241
コンテキストに依存した 280

ATTACHMENT 275

CATEGORY 265

CHKPID 256

CONTACT 275

CONTAIN 275

DL、タグ言語リファレンス 272

DT、タグ言語リファレンス 272

キーワード (続き)

EXTNAME
OBJECT タグの 265, 269
PROPERTY タグの 272

ICOFIELD、OBJECT のオプション・キーワード 265, 269

ICWFIELD、OBJECT のオプション・キーワード 265, 269

LINK 275

OBJTYPE 249

PHYNAME 265

RELATION 253

RELTYPE 275, 277

SEQUENCE 257

SOURCEKEY 260
ACTION.OBJINST(DELETE) 261
ACTION.RELATION 263

SOURCETYPE 275

TARGETKEY 263

TARGETTYPE 275

TYPE
OBJTYPE(ADD) 265
OBJTYPE(APPEND) 268
OBJTYPE(DELETE) 268, 270
OBJTYPE(MERGE) 265
OBJTYPE(UPDATE) 269, 270
RELTYPE 275

UIUSEQ 272

基礎的区分

オブジェクト・タイプ
イメージまたはグラフィックス、サンプル情報カタログで提供される 218

インターネット文書、サンプル情報カタログで提供される 219

オーディオ・クリップ、サンプル情報カタログで提供される 214

サンプル情報カタログで提供される 214

図表、サンプル情報カタログで提供される 215

スプレッドシート、サンプル情報カタログで提供される 222

基礎的区分 (続き)

- テキスト・ベース・レポート、サンプル情報カタログで提供される 224
- ビデオ・クリップ、サンプル情報カタログで提供される 225
- プレゼンテーション、サンプル情報カタログで提供される 221
- 文書、サンプル情報カタログで提供される 217
- Lotus Approach 照会、サンプル情報カタログで提供される 220
- 他の区分との関係 155
- 定義 155

疑似コード

- エージェント・サイト 23
- ステップ 33, 35
- ソースおよびターゲット・データベース 27
- データウェアハウスセンター プログラム 30

区分

基礎的

- サンプル情報カタログ内の Lotus Approach 照会オブジェクト・タイプ 220
- サンプル情報カタログ内のイメージまたはグラフィックス・オブジェクト・タイプ 218
- サンプル情報カタログ内のインターネット文書オブジェクト・タイプ 219
- サンプル情報カタログ内のオーディオ・クリップ・オブジェクト・タイプ 214
- サンプル情報カタログ内のオブジェクト・タイプ 214
- サンプル情報カタログ内の図表オブジェクト・タイプ 215

区分 (続き)

基礎的 (続き)

- サンプル情報カタログ内のスプレッドシート・オブジェクト・タイプ 222
- サンプル情報カタログ内のテキスト・ベース・レポート・オブジェクト・タイプ 224
- サンプル情報カタログ内のビデオ・クリップ・オブジェクト・タイプ 225
- サンプル情報カタログ内のプレゼンテーション・オブジェクト・タイプ 221
- サンプル情報カタログ内の文書オブジェクト・タイプ 217
- 他の区分との関係 155
- 定義 155

グループ化

- サンプル情報カタログ内の DWC プロセス・オブジェクト・タイプ 187
- サンプル情報カタログ内の IMS セグメント・オブジェクト・タイプ 198
- サンプル情報カタログ内の IMS データベース定義 (DBD) オブジェクト・タイプ 193
- サンプル情報カタログ内の IMS プログラム仕様ブロック (PSB) オブジェクト・タイプ 196
- サンプル情報カタログ内の IMS プログラム制御ブロック (PCB) オブジェクト・タイプ 195
- サンプル情報カタログ内のアプリケーション・データ・オブジェクト・タイプ 177
- サンプル情報カタログ内のエレメント・オブジェクト・タイプ 188
- サンプル情報カタログ内のオブジェクト・タイプ 176

区分 (続き)

グループ化 (続き)

- サンプル情報カタログ内のサスキーマ・オブジェクト・タイプ 210
- サンプル情報カタログ内のスタースキーマ・オブジェクト・タイプ 209
- サンプル情報カタログ内の多次元データベース内の次元オブジェクト・タイプ 185
- サンプル情報カタログ内の多次元データベース内のメンバー・オブジェクト・タイプ 200
- サンプル情報カタログ内の多次元データベース・オブジェクト・タイプ 202
- サンプル情報カタログ内のデータベース・オブジェクト・タイプ 183
- サンプル情報カタログ内のトランスフォーメーション・オブジェクト・タイプ 212
- サンプル情報カタログ内のビジネス・サブジェクト・エリア・オブジェクト・タイプ 180
- サンプル情報カタログ内のファイル・オブジェクト・タイプ 190
- サンプル情報カタログ内のリレーショナル表および視点オブジェクト・タイプ 206
- サンプル情報カタログ内の列またはフィールド・オブジェクト・タイプ 180
- サンプル情報カタログ内のレコード・オブジェクト・タイプ 204
- 他の区分との関係 155
- 定義 155
- コンタクト
- サンプル情報カタログ内のオブジェクト・タイプ 226

区分 (続き)

サンプル情報カタログ内の連絡先オブジェクト・タイプ 226

他の区分との関係 155

定義 155

サポート

サンプル情報カタログ内のオブジェクト・タイプ 229

サンプル情報カタログ内のオンライン資料オブジェクト・タイプ 231

サンプル情報カタログ内のオンライン・ニュース・サービス・オブジェクト・タイプ 230

サンプル情報カタログ内の情報カタログ・マネージャー・ニュース・オブジェクト・タイプ 229

他の区分との関係 155

定義 155

辞書

サンプル情報カタログ内のオブジェクト・タイプ 227

サンプル情報カタログ内の用語集エントリ・オブジェクト・タイプ 227

他の区分との関係 155

定義 155

接続

他の区分との関係 155

定義 155

定義済みのコメント・オブジェクト・タイプ 233

プログラム

他の区分との関係 155

プログラム、定義済みのプログラム・オブジェクト・タイプ 232

クリア、エコー・ファイルの 281

グループ化区分

オブジェクト・タイプ

アプリケーション・データ、サンプル情報カタログで提供される 177

グループ化区分 (続き)

オブジェクト・タイプ (続き)

エレメント、サンプル情報カタログで提供される 188

サブスキーマ、サンプル情報カタログで提供される 210

サンプル情報カタログで提供される 176

スタースキーマ、サンプル情報カタログで提供される 209

多次元データベース、サンプル情報カタログで提供される 202

多次元データベース内の次元、サンプル情報カタログで提供される 185

多次元データベース内のメンバー、サンプル情報カタログで提供される 200

データベース、サンプル情報カタログで提供される 183

トランスフォーメーション、サンプル情報カタログで提供される 212

ビジネス・サブジェクト・エリア、サンプル情報カタログで提供される 180

ファイル、サンプル情報カタログで提供される 190

リレーショナル表および視点、サンプル情報カタログで提供される 206

列またはフィールド、サンプル情報カタログで提供される 180

レコード、サンプル情報カタログで提供される 204

DWC プロセス、サンプル情報カタログで提供される 187

IMS セグメント、サンプル情報カタログで提供される 198

IMS データベース定義 (DBD)、サンプル情報カタログで提供される 193

グループ化区分 (続き)

オブジェクト・タイプ (続き)

IMS プログラム仕様ブロック (PSB)、サンプル情報カタログで提供される 196

IMS プログラム制御ブロック (PCB)、サンプル情報カタログで提供される 195

他の区分との関係 155

定義 155

交換プログラム

作成 17

定義 16

構文図 243

コマンド

DGUIDE、情報カタログのオープン 44

コミット・チェックポイント 255

コメント・オブジェクト・タイプ 234

コンタクト区分

オブジェクト・タイプ

サンプル情報カタログ 226

連絡先、サンプル情報カタログで提供される 226

他の区分との関係 155

定義 155

[サ行]

作成、タグ言語ファイルの 240

サブスキーマ、サンプル・オブジェクト・タイプ 210

サポート区分

オブジェクト・タイプ

オンライン資料、サンプル情報カタログで提供される 231

オンライン・ニュース・サービス、サンプル情報カタログで提供される 230

サンプル情報カタログで提供される 229

情報カタログ・マネージャー・ニュース、サンプル情報カタログで提供される 229

- サポート区分 (続き)
 - 他の区分との関係 155
 - 定義 155
 - サポートされないタグおよびキーワード 241
 - サンプル情報カタログ
 - 事前定義プログラム・オブジェクト 234
 - 定義済みのオブジェクト・タイプ 175
 - 辞書区分
 - オブジェクト・タイプ 227
 - 用語集エントリ、サンプル情報カタログで提供される 227
 - 他の区分との関係 155
 - 定義 155
 - システム・プロセス 333
 - 情報カタログ
 - オブジェクト・タイプの確立 156
 - コマンド行からのオープン 44
 - 情報カタログ・マネージャーと一緒に提供されるサンプル
 - 事前定義プログラム・オブジェクト 234
 - 定義済みのオブジェクト・タイプ 175
 - MDIS メタデータのインポート 42, 48, 49
 - MDIS メタデータのエクスポート 42, 48, 49
 - 情報カタログ・マネージャー
 - 情報カタログをコマンド行からオープンする 44
 - メタデータ・モデル 150
 - 情報カタログ・マネージャー (Web版)
 - 図形ファイル 293
 - HTML ファイル 292
 - Net.Data ファイル 291
 - 情報カタログ・マネージャーへのログオン
 - コマンド行から 44
 - 情報カタログ・マネージャー・オブジェクトから呼び出すことができるプログラム 232
 - 情報カタログ・マネージャー・ニュース、サンプル・オブジェクト・タイプ 229
 - スターズキーマ、サンプル・オブジェクト・タイプ 209
 - ステップ
 - 疑似コード 33, 35
 - 状況 338
 - データウェアハウスセンター プログラム・フィードバック 338
 - 定義 5
 - 提供する値 33
 - テンプレート 31
 - 図表、サンプル・オブジェクト・タイプ 215
 - スプレッドシート、サンプル・オブジェクト・タイプ 222
 - 接続区分
 - 他の区分との関係 155
 - 定義 155
 - 接続区分、定義済みのコメント・オブジェクト・タイプ 233
 - ソース
 - 提供する値 26
 - テンプレート 23
 - ソース・データベース
 - 疑似コード 27
 - 定義 14
 - ソース・ファイル 14
- ## [タ行]
- ターゲット
 - 提供する値 26
 - テンプレート 23
 - ファイル 14
 - ターゲット・データベース
 - 疑似コード 27
 - 定義 14
 - タグ
 - 各国語についてサポートされない 241
 - コンテキストに依存した使用 279
 - タグ (続き)
 - 情報を定義するための 280
 - ACTION
 - 順序 280
 - タグ言語リファレンス 243, 255
 - ヒント 279
 - OBJINST キーワード 260
 - COMMENT
 - タグ言語リファレンス 255
 - ヒント 282
 - COMMIT
 - タグ言語リファレンス 255, 257
 - ヒント 281
 - DISKCNTRL
 - タグ言語リファレンス 257
 - ヒント 279
 - INSTANCE 258, 264
 - NL 264
 - NULLS 272
 - PROPERTY 271, 275
 - TAB 277
 - タグ言語
 - オブジェクト
 - COLUMN 127
 - TABLE 121
 - 概要 239
 - 構文規則 240
 - 情報の定義 280
 - データウェアハウスセンター
 - インポート 36
 - エクスポート 39
 - 構築 13
 - 定義 13
 - ファイル
 - 情報カタログ・マネージャーが読み取る方法 241
 - MDIS 対応の、インポートおよびエクスポート 42, 48, 49
 - リファレンス 239, 277
 - タグ言語の構文規則 240
 - タグ言語ファイル・ヘッダー 21
 - 多次元データベース、サンプル・オブジェクト・タイプ 202

多次元データベース内の次元、サンプル・オブジェクト・タイプ 185
多次元データベース内のメンバー、サンプル・オブジェクト・タイプ 200
データウェアハウスセンター
エージェント 13
タグ言語
インポート 36
エクスポート 39
構築 13
メタデータ
インポート 13
エクスポート 38
表 121
列 127
DATABASE オブジェクト 115
データウェアハウスセンター プログラム
位置 333
エージェントをユーザー・プロセスに変更する 333
およびステップ状況 338
疑似コード 30
作成 333
定義 14
提供する値 29
テンプレート 28
パラメーター 334
フィードバック 335
戻りコード 335
DB2 UDB ロード・置換 334
Object REXX for Windows 333
データウェアハウスセンター メタデータのエクスポート 38
データベース
ウェアハウス・ソース 14
ウェアハウス・ターゲット 14
ロールバック 281
データベース、サンプル・オブジェクト・タイプ 183
データ・タイプ 242, 272
テキスト・ベース・レポート、サンプル・オブジェクト・タイプ 224

テンプレート
定義 13
ヘッダー・ファイル 17
AgentSite.tag
値の例 64
トークン 62
BusinessViewInputTable.tag
値の例 323
トークン 321
BusinessViewOutputTable.tag
値の例 325
トークン 324
BusinessViewVWOutputTable.tag
値の例 326
トークン 325
BusinessView.tag
値の例 320
トークン 314
Column.tag
値の例 69
トークン 64
ConcurrentCascade.tag
値の例 327
トークン 327
HeaderInfo.tag 70
PostCascade.tag 328
SourceDataBase.tag
値の例 90
トークン 87
SubjectArea.tag
値の例 92
トークン 91
Table.tag
値の例 98
トークン 93
VWPGGroup.tag 99
VWPPProgramInstanceParameter.tag
値の例 102
トークン 101
VWPPProgramInstance.tag
値の例 331
トークン 329
VWPPProgramParameter.tag
値の例 109
トークン 107

テンプレート (続き)
VWPPProgramTemplate.tag
値の例 106
トークン 103
WarehouseDataBase.tag
値の例 112
トークン 109
トランスフォーメーション、サンプル・オブジェクト・タイプ 212

[八行]

パートナー・アプリケーション 3
パートナー・メタデータ 6
汎用固有 ID
プロパティ値 261
ビジネス・サブジェクト・エリア、サンプル・オブジェクト・タイプ 180
ビデオ・クリップ、サンプル・オブジェクト・タイプ 225
ファイル、サンプル・オブジェクト・タイプ 190
フィードバック・ファイル 336
ブランク、可変値から除去される 242
プレゼンテーション、サンプル・オブジェクト・タイプ 221
プログラム、情報カタログ・マネージャから開始 53
プログラム区分
他の区分との関係 155
プログラム区分、定義済みのプログラム・オブジェクト・タイプ 232
プログラム・オブジェクト・タイプ 232
プロパティ
値 175
基礎的区分のサンプル・オブジェクト・タイプの仕様
イメージまたはグラフィックス 218
インターネット文書 219
オーディオ・クリップ 214
図表 215
スプレッドシート 222

プロパティ (続き)

- テキスト・ベース・レポート 224
- ビデオ・クリップ 225
- プレゼンテーション 221
- 文書 217
- Lotus Approach 照会 220
- グループ化区分のサンプル・オブジェクト・タイプの仕様
アプリケーション・データ 177
- エレメント 188
- サブスキーマ 210
- スタースキーマ 209
- 多次元データベース 202
- 多次元データベース内の次元 185
- 多次元データベース内のメンバ 200
- データベース 183
- トランスフォーメーション 212
- ビジネス・サブジェクト・エリア 180
- ファイル 190
- リレーショナル表および視点 206
- 列またはフィールド 180
- レコード 204
- DWC プロセス 187
- IMS セグメント 198
- IMS データベース定義 (DBD) 193
- IMS プログラム仕様ブロック (PSB) 196
- IMS プログラム制御ブロック (PCB) 195
- コンタクト区分のサンプル・オブジェクト・タイプの仕様、連絡先 226
- サポート区分のサンプル・オブジェクト・タイプの仕様
オンライン資料 231
- オンライン・ニュース・サービ 230

プロパティ (続き)

- サポート区分のサンプル・オブジェクト・タイプの仕様 (続き)
情報カタログ・マネージャ
ー・ニュース 229
- 辞書区分のサンプル・オブジェクト・タイプの仕様、用語集エントリ 227
- 仕様 175
- 接続区分のオブジェクト・タイプの仕様、コメント 234
- プログラム区分のオブジェクト・タイプの仕様 232
- 文書、サンプル・オブジェクト・タイプ 217

[マ行]

マクロ

- Net.Data 291
- メタデータ
ウェアハウス・ソース 115
- データウェアハウスセンターからのエクスポート 38
- データウェアハウスセンターへのインポート 13
- 表
関係 126
- プロパティ 121
- モデル、情報カタログ・マネージャ 150
- 列
関係 130
- プロパティ 127
- メタデータ交換仕様 (MDIS) 7

[ヤ行]

- ユーザー・プロセス 333
- 用語集エントリ、サンプル・オブジェクト・タイプ 227
- 読み方、構文図の 243
- 予約語 240

[ラ行]

- リレーショナル表および視点、サンプル・オブジェクト・タイプ 206
- 列またはフィールド、サンプル・オブジェクト・タイプ 180

例

関係

- COLUMN オブジェクトの TABLES オブジェクトへの 131
- DATABASE オブジェクト 120
- TABLE オブジェクトの COLUMN オブジェクトへの 127
- TABLES オブジェクトの DATABASE オブジェクトへの 126
- タグ言語ファイルの定義 22
- COLUMN オブジェクト 130
- DATABASE オブジェクト
ソース 119
- ターゲット 119
- TABLE オブジェクト
ファイル 125
- リレーショナル表 125
- レコード、サンプル・オブジェクト・タイプ 204
- 連鎖関係 15
- 連鎖ステップ 35
- 連絡先、サンプル・オブジェクト・タイプ 226
- ロールバック、データの 281

[数字]

- 2 バイト文字セット (DBCS) 241

A

- ACTION タグ
順序 280
- タグ語リファレンス 243, 255
- ヒント 279
- OBJINST キーワード 243, 260
- OBJTYPE キーワード 249

ACTION タグ (続き)
RELATION キーワード 253
ADD オプション
ACTION.OBJINST 244
ACTION.OBJTYPE 249, 265
ACTION.RELATION 253
AgentSite.tag テンプレート
値の例 64
トークン 62
APPEND オプション 250
ATTACHMENT キーワード 275

B

BusinessViewInputTable.tag テンプレート
値の例 323
トークン 321
BusinessViewOutputTable.tag テンプレート
値の例 325
トークン 324
BusinessViewVWPOutputTable.tag テンプレート
値の例 326
トークン 325
BusinessView.tag テンプレート
値の例 320
トークン 314

C

CATEGORY キーワード 265
CHAR データ・タイプ、PROPERTY
タグ 272
ckpt_id 識別子 256
CHKPID キーワード 256
COLUMN オブジェクト
関係 130
プロパティ 127
Column.tag テンプレート
値の例 69
トークン 64
COMMENT タグ
タグ言語リファレンス 255
ヒント 282

COMMIT タグ
タグ言語リファレンス 255, 257
ヒント 281
ConcurrentCascade.tag テンプレート
値の例 327
トークン 327
CONTACT キーワード 275
CONTAIN キーワード 275

D

DATABASE オブジェクト
関係 120
プロパティ 115
DB2 UDB ロード置換データウェア
ハウスセンター プログラム 334
DBCS 241
DELETE オプション
ACTION.OBJINST 244
ACTION.OBJTYPE 250
ACTION.RELATION 254
OBJINST キーワード 260
DELETE_EXT オプション、
ACTION.OBJTYPE の 251
DELETE_TREE_ALL オプション
ACTION.OBJINST 245
OBJINST キーワード 260
DELETE_TREE_REL オプション
ACTION.OBJINST 246
OBJINST キーワード 260
DGUIDE コマンド 48
ADMIN キーワード 45
DGNAME キーワード、情報カ
タログを指定するための 45
IMPORT キーワード
ICOPATH 45
LOGFILE 46
RESTART 46
PASSWORD キーワード 46
TRACE キーワード 47
USERID キーワード 47
DGUIDE コマンド、情報カタログ・
マネージャーを起動するための
42, 49
ADMIN キーワード 42, 49
DGNAME キーワード、情報カ
タログを指定するための 42, 50

DGUIDE コマンド、情報カタログ・
マネージャーを起動するための
42, 49 (続き)
LOGFILE、MDIS_IMPORT キー
ワード 43, 50
MDIS_EXPORT キーワード
LOGFILE 43, 50
OBJECT 50
OBJTYPE 51
PASSWORD キーワード 43, 51
TRACE キーワード 43, 52
USERID キーワード 44, 52
DGV3SAMP サンプル情報カタログ
175
DISKCNTRL タグ
タグ言語リファレンス 257
ヒント 279
DL キーワード 272
DT キーワード 272
DWC プロセス、サンプル・オブジェ
クト・タイプ 187

E

EXTNAME キーワード
OBJECT タグの 265, 269
PROPERTY タグの 272

H

HeaderInfo.tag テンプレート 70

I

ICOFIELD キーワード 265
タグ言語リファレンス 269
ICWFILE キーワード 265
タグ言語リファレンス 269
IMS セグメント、サンプル・オブジ
ェクト・タイプ 198
IMS データベース定義 (DBD)、サン
プル・オブジェクト・タイプ 193
IMS プログラム仕様ブロック
(PSB)、サンプル・オブジェクト・
タイプ 196

IMS プログラム制御ブロック
(PCB)、サンプル・オブジェクト・
タイプ 195

INSTANCE タグ
タグ言語リファレンス 258, 264

ACTION.OBJINST
(DELETE_TREE_ALL) 260

ACTION.OBJINST
(DELETE_TREE_REL) 260

ACTION.OBJINST(ADD) 258

ACTION.OBJINST(DELETE) 260

ACTION.OBJINST(MERGE) 258

ACTION.OBJINST(UPDATE) 261

ACTION.RELATION(ADD) 262

ACTION.RELATION(DELETE) 262

ISV_defines.h ファイル 17

L

LINK キーワード 275

LONG VARCHAR データ・タイプ、
PROPERTY タグ 272

Lotus Approach 照会、サンプル・オ
ブジェクト・タイプ 220

M

MDIS 168
タグ言語ファイル、インポート
42
タグ言語ファイル、エクスポート
49
マップする事前定義オブジェク
ト・タイプ 168
エレメント 188
サブスキーマ 210
多次元データベース 202
多次元データベース内の次元
185
多次元データベース内のメン
バー 200
データベース 183
トランスフォーメーション
212
ファイル 190
リレーショナル表および視点
206

MDIS 168 (続き)
マップする事前定義オブジェク
ト・タイプ 168 (続き)
列またはフィールド 180
レコード 204
IMS セグメント 198
IMS データベース定義
(DBD) 193
IMS プログラム仕様ブロック
(PSB) 196
IMS プログラム制御ブロック
(PCB) 195

MERGE オプション
ACTION.OBJINST 247
ACTION.OBJTYPE 251, 265

N

Net.Data
組み込みファイル
情報カタログ・マネージャー
(Web 版) 292

マクロ
情報カタログ・マネージャー
(Web 版) 291

NL タグ 264

NULLS キーワード 272

O

Object REXX for Windows 333

OBJECT タグ
タグ言語リファレンス 264, 271

ACTION.OBJTYPE(ADD) 265

ACTION.OBJTYPE(APPEND) 268

ACTION.OBJTYPE(DELETE) 268

ACTION.OBJTYPE(DELETE_EXT) 268

ACTION.OBJTYPE(MERGE) 265

ACTION.OBJTYPE(UPDATE) 269

P

PHYNAME キーワード 265

PostCascade.tag テンプレート 328

PROPERTY タグ 271, 275

R

RELTYPE タグ 275, 277

S

SEQUENCE キーワード 257

SHRTNAME キーワード 272

SourceDataBase.tag テンプレート
値の例 90
トークン 87

SOURCEKEY キーワード
タグ言語リファレンス 260
ACTION.OBJINST(DELETE) 261
ACTION.RELATION 263

SOURCETYPE キーワード 275

SubjectArea.tag テンプレート
値の例 92
トークン 91

T

TAB タグ 277

TABLE オブジェクト
関係 126
プロパティ 121

Table.tag テンプレート
値の例 98
トークン 93

TARGETKEY キーワード 263

TARGETTYPE キーワード 275

TIMESTAMP データ・タイプ、
PROPERTY タグ 272

TYPE キーワード
OBJTYPE(ADD) 265
OBJTYPE(APPEND) 268
OBJTYPE(DELETE) 268, 270
OBJTYPE(MERGE) 265
OBJTYPE(UPDATE) 269, 270
RELTYPE 275

U

UPDATE オプション
ACTION.OBJINST 248, 261
ACTION.OBJTYPE 252

UUI

プロパティ値 261

UUISEQ キーワード 272

UUI_property_value 260

UUI_short_name の値 260

V

VARCHAR データ・タイプ、

PROPERTY タグ 272

VWPGroup.tag テンプレート 99

VWPProgramInstanceParameter.tag テンプレート

値の例 102

トークン 101

VWPProgramInstance.tag テンプレート

値の例 331

トークン 329

VWPProgramTemplateParameter.tag テンプレート

値の例 109

トークン 107

VWPProgramTemplate.tag テンプレート

値の例 106

トークン 103

W

WarehouseDataBase.tag テンプレート

値の例 112

トークン 109

IBM と連絡をとる

技術上の問題がある場合は、時間をとって「問題判別の手引き」に定義されている処置を検討し、それらの提案を実行した後で、DB2 顧客サービスに連絡をとってください。この資料には、DB2 顧客サービスがお客さまを支援するために必要とする情報が説明されています。

製品情報

以下の情報は英語で提供されます。内容は英語版製品に関する情報です。

<http://www.ibm.com/software/data/>

DB2 World Wide Web ページには、ニュース、製品説明、研修スケジュールなどの DB2 に関する最新情報が提供されています。ただし、提供されている情報は英語です。

<http://www.ibm.com/software/data/db2/library/>

「DB2 Product and Service Technical Library」では、よくされる質問 (FAQ)、修正内容、資料、および最新の DB2 技術情報などの情報へのアクセスが提供されています。

注: この情報のご提供は英語のみとなりますのでご注意ください。

<http://www.elink.ibm.com/pbl/pbl/>

「International Publications」注文用 Web サイトでは、マニュアルの注文方法についての情報を提供しています。ただし、提供されている情報は英語です。

<http://www.ibm.com/education/certify/>

IBM の「Professional Certification Program」Web サイトでは、DB2 を含むさまざまな IBM 製品の認証テストの情報を提供しています。ただし、提供されている情報は英語です。

<ftp.software.ibm.com>

匿名でログオンしてください。ディレクトリー /ps/products/db2 には、DB2 および多数の他製品に関連したデモ、修正プログラム、情報、およびツールがあります。ただし、提供されている情報は英語です。

comp.databases.ibm-db2, bit.listserv.db2-l

これらのインターネット・ニュースグループは、ユーザーが DB2 製品に関する自分の経験について話し合うために利用できます。ただし、提供されている情報は英語です。

CompuServe: GO IBMDB2

このコマンドを入力すると、IBM DB2 Family forum にアクセスできます。すべての DB2 製品が、このフォーラムでサポートされています。ただし、提供されている情報は英語です。

米国以外の国で IBM に連絡する方法については、*IBM Software Support Handbook* の Appendix A を参照してください。この資料にアクセスするには、Web ページ: <http://www.ibm.com/support/> にアクセスし、ページの最下部にある「IBM Software Support Handbook」リンク・ボタンを選択します。

注: 国によっては、IBM が承認している販売業者が、IBM サポート・センターの代わりにそれら販売業者のサポート・センターに連絡する場合があります。



部品番号: CT60KJA

Printed in Japan

SC88-8546-00



日本アイ・ビー・エム株式会社
〒106-8711 東京都港区六本木3-2-12

CT60KJA



Spine information:



DB2® ユニバーサル・デー
タベース データウェアハウスセンター アプリケー
ション統合の手引き バージョン7