

ORACLE®

PEOPLESOFT

PeopleSoft 9.2: アプリケーション インテ グレーション フレームワーク

January 2014

ORACLE®

ライセンス制約の保証と結果的に生じる損害の免責

このソフトウェアおよび関連ドキュメントの使用と開示は、ライセンス契約の制約条件に従うものとし、知的財産に関する法律により保護されています。ライセンス契約で明示的に許諾されている場合もしくは法律によって認められている場合を除き、形式、手段に関係なく、いかなる部分も使用、複写、複製、翻訳、放送、修正、ライセンス供与、送信、配布、発表、実行、公開または表示することはできません。このソフトウェアのリバース・エンジニアリング、逆アセンブル、逆コンパイルは互換性のために法律によって規定されている場合を除き、禁止されています。

保証免責

ここに記載された情報は予告なしに変更される場合があります。また、誤りが無いことの保証はいたしかねます。誤りを見つけた場合は、オラクルまでご連絡ください。

制限付権利

このソフトウェアまたは関連ドキュメントを、米国政府機関もしくは米国政府機関に代わってこのソフトウェアまたは関連ドキュメントをライセンスされた者に提供する場合は、次の通知が適用されます。

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

危険な用途への使用について

このソフトウェアまたはハードウェアは様々な情報管理アプリケーションでの一般的な使用のために開発されたものです。このソフトウェアまたはハードウェアは、危険が伴うアプリケーション(人的傷害を発生させる可能性があるアプリケーションを含む)への用途を目的として開発されていません。このソフトウェアまたはハードウェアを危険が伴うアプリケーションで使用する場合、安全に使用するために、適切な安全装置、バックアップ、冗長性(redundancy)、その他の対策を講じることは使用者の責任となります。このソフトウェアまたはハードウェアを危険が伴うアプリケーションで使用したこと起因して損害が発生しても、Oracle Corporationおよびその関連会社は一切の責任を負いかねます。

商標と登録商標について

OracleおよびJavaはオラクルおよびその関連会社の登録商標です。その他の名称は、それぞれの所有者の商標または登録商標です。

Intel、Intel Xeonは、Intel Corporationの商標または登録商標です。すべてのSPARCの商標はライセンスをもとに使用し、SPARC International, Inc.の商標または登録商標です。AMD、Opteron、AMDロゴ、AMD

Opteronロゴは、Advanced Micro Devices, Inc.の商標または登録商標です。UNIXは、The Open Groupの登録商標です。

第三者のコンテンツ、製品、サービスに対する免責

このソフトウェアまたはハードウェア、そしてドキュメントは、第三者のコンテンツ、製品、サービスへのアクセス、あるいはそれらに関する情報を提供することがあります。適用されるお客様とOracle Corporationとの間の契約に別段の定めがある場合を除いて、Oracle Corporationおよびその関連会社は、第三者のコンテンツ、製品、サービスに関して一切の責任を負わず、いかなる保証もいたしません。適用されるお客様とOracle Corporationとの間の契約に定めがある場合を除いて、Oracle Corporationおよびその関連会社は、第三者のコンテンツ、製品、サービスへのアクセスまたは使用によって損失、費用、あるいは損害が発生しても一切の責任を負いかねます。

ドキュメントのアクセシビリティについて

オラクルのアクセシビリティについての詳細情報は、Oracle Accessibility ProgramのWeb サイト(<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>)を参照してください。

Oracle Supportへのアクセス

サポートをご契約のお客様には、My Oracle Supportを通して電子支援サービスを提供しています。詳細情報は(<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info>)か、聴覚に障害のあるお客様は(<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs>)を参照してください。

目次

はじめに.....	ix
PeopleSoft オンライン ヘルプおよび PeopleBook について.....	ix
PeopleSoft ホスト ドキュメント.....	ix
ローカルにインストールされたヘルプ.....	ix
ダウンロード可能な PeopleBook のPDF ファイル.....	x
共通のヘルプ ドキュメント.....	x
フィールドおよびコントロールの定義.....	x
表記規則.....	x
ISO の国コードおよび通貨コード.....	xi
地域および業種の識別子.....	xi
PeopleSoft オンライン ヘルプの使用および管理.....	xii
PeopleSoft エンタープライズ コンポーネントの関連リンク.....	xii
連絡先.....	xii
フォロー.....	xiii
第1章: アプリケーション インテグレーション フレームワークの概要.....	15
アプリケーション インテグレーション フレームワークについて.....	15
アプリケーション インテグレーション フレームワークの実装.....	15
値マップの定義.....	15
アプリケーション エンジン変換プログラムの作成.....	15
サービス オペレーションのルーティングの更新.....	16
第2章: アプリケーション インテグレーション フレームワークについて.....	17
アプリケーション インテグレーション フレームワーク.....	17
アプリケーション インテグレーション フレームワークのユース ケース.....	19
マップ.....	19
ドメイン値マップ.....	20
クロスリファレンス マップ.....	20
クロスリファレンスと DVM をロードおよび管理するための関数.....	21
ユース ケース: AIA ミドルウェアなしのインテグレーション ブロカー変換.....	22
ユース ケース: インテグレーション ブロカーによるポイントツーポイント変換.....	26
ユース ケース: サードパーティが AIA ミドルウェアを使用するインテグレーション ブロカー変換.....	30
第3章: ドメイン値マップの定義とロード.....	35
値マップについて.....	35
マップ オプションの定義.....	39
値マップのオプションの定義に使用するページ.....	39
値マップ オプション ページ.....	39
値マップの定義.....	42
値マップの定義に使用するページ.....	42
値マップの定義 検索ページ.....	42
値マップの定義 - エレメント ページ.....	43
値マップのオプション定義ページ.....	45

値マップの定義 - ドメイン ページ.....	46
ドメイン値マップのロード.....	48
ドメイン値マップのロードに使用するページ.....	48
ドメイン値マップ ページ.....	48
値マップのインポート.....	49
値マップのインポートに使用するページ.....	49
値マップで使用するインポート ファイル タイプについて.....	49
値マップのインポート ページ.....	51
値マップのエクスポート.....	53
エクスポート スキーマについて.....	53
値マップのエクスポート.....	54
第4章: 変換プログラムの作成とサービス オペレーションの更新.....	55
変換プログラムについて.....	55
変換プログラムの作成.....	56
サービス オペレーションのルーティングの更新.....	57
第5章: XSLT 拡張を使用したマップへのアクセス.....	59
XSLT 拡張関数について.....	59
クロスリファレンス関数.....	59
xref:BulkPopulateDomainData.....	59
xref:BulkPopulateElementData.....	60
xref:populateXRefRow.....	61
xref:populateXrefRowNVP.....	64
xref:markForDelete.....	66
xref:markForDeleteNVP.....	67
xref:lookupXRef.....	68
xref:lookupXRefNVP.....	69
ドメイン値マップ関数.....	71
dvm:lookupValue.....	71
dvm:lookupValueNVP.....	72
dvm:lookup-dvm.....	73
Generate-Guid 関数.....	75
generate-guid.....	75
SetID 関数.....	75
SetID:lookupSetCtrlValues.....	75
第6章: PeopleCode を使用したマップへのアクセス.....	79
アプリケーション インテグレーション フレームワーク クラスについて.....	79
アプリケーション インテグレーション フレームワーク Type クラスのインポート方法.....	79
DVM ユーティリティ クラス メソッド.....	80
ExtractData.....	80
LookupValue.....	81
LookupValue1M.....	82
LookupValueNVP.....	83
DVM ユーティリティ クラス プロパティ.....	84
exceptionCaught.....	84
exceptionDetails.....	85
SetID ユーティリティ クラス メソッド.....	85
lookupSetCtrlValues.....	85

lookupSetID.....	86
Xref ユーティリティ クラス メソッド.....	88
BulkPopulateDomainData.....	88
BulkPopulateElementData.....	89
ExtractData.....	90
LookupValue.....	91
LookupValue1M.....	92
LookupValueNVP.....	93
MarkForDelete.....	94
MarkForDeleteNVP.....	95
PopulateData.....	96
PopulateValue.....	98
PopulateValueNVP.....	99
Xref クラス プロパティ.....	101
exceptionCaught.....	101
exceptionDetails.....	101
第7章: Web サービスを使用したマップへのアクセス.....	103
アプリケーション インテグレーション フレームワーク Web サービスについて.....	103
EOTF_DVM サービス.....	104
EOTF_DVM_LOOKUP.....	104
EOTF_DVM_IMPORT.v2.....	105
EOTF_DVM_IMPORT.v1.....	106
EOTF_XREF サービス.....	107
EOTF_XREF_ADD.....	108
EOTF_XREF_LINK.....	109
EOTF_XREF_UPDATE.....	110
EOTF_XREF_DELETE.....	111
EOTF_XREF_LOOKUP.....	112
EOTF_XREF_IMPORT.v2.....	113
EOTF_XREF_IMPORT.v1.....	115
クロスリファレンス検索 Web サービスの例.....	116
付録A: アプリケーション インテグレーション フレームワークの例.....	119
例の概要.....	119
動的値マップの定義.....	120
静的値マップの定義とロード.....	121
変換プログラムでの XSLT 拡張関数の使用.....	124
キー値の変換.....	124
ドメイン値の変換.....	126
サービス オペレーションのルーティングの更新.....	126

はじめに

PeopleSoft オンライン ヘルプおよび PeopleBook について

PeopleSoft オンライン ヘルプは、PeopleSoft アプリケーションおよび PeopleTools の全てのヘルプ コンテンツを表示できる Web サイトです。このヘルプでは、標準ナビゲーションと全文検索、さらに状況依存オンライン ヘルプが PeopleSoft ユーザーに提供されます。

PeopleSoft ホスト ドキュメント

オラクル社の PeopleSoft ホストドキュメント Web サイトの PeopleSoft オンライン ヘルプにアクセスすると、Oracle ホスト サーバーの全てのヘルプ Web サイトおよび状況依存ヘルプに直接アクセスすることができます。ホストドキュメントは定期的に更新されるため、ユーザーは最新のドキュメントにアクセスすることができます。ホストの Web サイト コンテンツにはアプリケーションの保守に関するドキュメントがすぐに反映されるため、これらの個別のドキュメント記録を My Oracle Support で確認する必要がなくなります。ホストドキュメントの Web サイトは英語版のみ利用可能です。

注: 最新リリースのホストドキュメントのみが定期的に更新されます。新しいリリースが登録されても、前のリリースは引き続き利用できますが、更新はされません。

ローカルにインストールされたヘルプ

組織においてファイアウォールの制限により、ホストドキュメントの Web サイトを利用できない場合は、PeopleSoft オンライン ヘルプをローカルでインストールすることができます。ローカルでヘルプをインストールした場合は、ユーザーがアクセス可能なドキュメントをより詳細に管理したり、自分の組織のカスタムドキュメントへのリンクをヘルプ ページに組み込むことができます。

また、PeopleSoft オンライン ヘルプをローカルにインストールすると、全文検索にあらゆる検索エンジンを使用することができます。Oracle Secure Enterprise Search を設定して全文検索を行う手順については、ご使用のインストールドキュメントに記載されています。

ご使用のデータベース プラットフォーム向けの『*PeopleTools Installation*』の「Installing PeopleSoft Online Help」を参照してください。Secure Enterprise Search を使用しない場合は、選択した検索エンジンのドキュメントを参照してください。

注: ローカルでインストールしたヘルプ Web サイトで検索エンジンにアクセスするには、検索のポートレットとリンクを有効にする必要があります。手順については、PeopleSoft オンライン ヘルプのいずれかのページで [ヘルプ] リンクをクリックしてください。

ダウンロード可能な PeopleBook の PDF ファイル

従来の PeopleBook 形式であり、ダウンロード可能な PDF 版のヘルプ コンテンツにアクセスすることもできます。PeopleBook PDF のコンテンツは PeopleSoft オンライン ヘルプのコンテンツと同じですが、構造は異なり、オンライン ヘルプで使用できるインタラクティブなナビゲーション機能はありません。

共通のヘルプドキュメント

共通のヘルプドキュメントには、複数のアプリケーションに適用される情報が含まれています。共通のヘルプの 2 つの主要なタイプは、次のとおりです。

- アプリケーションの基礎
- PeopleSoft アプリケーションの使い方

ほとんどの製品ファミリーでは、システムの設定や設計に関する基本的な情報を説明する、一連のアプリケーションの基礎ヘルプトピックが提供されます。この情報は、PeopleSoft 製品ファミリーの多くの、または全てのアプリケーションに適用されます。単一のアプリケーション、製品ファミリー内のアプリケーションの組合せ、または製品ファミリー全体のいずれを実装化に関係なく、該当するアプリケーションの基礎ヘルプのコンテンツについて理解している必要があります。これらは、基本的な実装タスクの開始点となります。

また、『*PeopleTools: Applications User's Guide*』では、PeopleSoft ピュア インターネット アーキテクチャのさまざまな要素が紹介されています。また、そこではシステムをナビゲートする際に、ナビゲーションの階層、コンポーネントおよびページを使用して基本的な機能を実行する方法についても説明しています。アプリケーションや導入内容は異なる場合がありますが、このユーザーズ ガイドのトピックでは PeopleSoft アプリケーションの使用方法に関する一般的な情報が説明されています。

フィールドおよびコントロールの定義

PeopleSoft のドキュメントには、アプリケーション ページに表示されるほとんどのフィールドおよびコントロールの定義が含まれています。これらの定義で、フィールドまたはコントロールの使用法や、移入される値の発生元、さらに特定の値を選択したときの結果などが説明されています。フィールドまたはコントロールが定義されていない場合は、追加の説明が必要ないか、ドキュメントの共通要素のセクションに既に記載されています。たとえば、日付フィールドは追加の説明がほとんど必要ないため、ドキュメント内で定義が複数のページにわたって記載されないことがあります。

表記規則

次の表に、オンライン ヘルプで使用される表記規則を示します。

表記規則	説明
キー + キー	キー操作の組合せを示します。たとえば、キーの間のプラス記号 (+) は、最初のキーを押しながら 2 番目のキーを押す必要があることを意味します。Alt+W の場合、Alt キーを押しながら W キーを押します。

表記規則	説明
... (省略記号)	PeopleCode 構文で、先行する項目または一連の項目を任意の回数繰り返すことができることを示します。
{ } (波かっこ)	PeopleCode 構文で、2つのオプションからの選択肢を示します。オプションはパイプ () で区切られます。
[] (角かっこ)	PeopleCode 構文で、オプションの項目を示します。
& (アンパサンド)	PeopleCode 構文でパラメータの前にアンパサンドが記述されている場合は、そのパラメータがインスタンス作成済みのオブジェクトであることを示します。 また、アンパサンドは全ての PeopleCode 変数の前に付きます。
⇒	この継続文字は、コードの行の最後、ページのマージンで折り返された場所に挿入されます。コードは、継続文字がない1つの連続した行のコードとして表示または入力される必要があります。

ISO の国コードおよび通貨コード

PeopleSoft オンライン ヘルプのトピックでは、国別の情報や通貨金額を識別するために国際標準化機構 (ISO) の国コードおよび通貨コードを使用しています。

PeopleSoft ドキュメントでは、ISO 国コードは国の識別子として表示される場合があります、ISO 通貨コードは通貨の識別子として表示される場合があります。ドキュメントでの ISO 国コードの参照は、全ての ISO 国コードがアプリケーションに含まれていることを意味しません。"(FRA) Hiring an Employee" は、国固有の見出しの例です。

PeopleSoft の通貨コード テーブル (CURRENCY_CD_TBL) には、サンプルの通貨コード データが含まれています。通貨コード テーブルは、ISO 標準 4217 "Codes for the representation of currencies" に基づいており、さらに国テーブル (COUNTRY_TBL) の ISO 国コードにも依存します。通貨コードおよび国情報を管理するページへのナビゲーションは、使用している PeopleSoft アプリケーションによって異なります。通貨コード テーブルおよび国テーブルを管理するページにアクセスするには、アプリケーションのオンライン ヘルプで詳細を確認してください。

地域および業種の識別子

特定の地域や業種に対してのみ適用される情報は、その前に標準の識別子がかっこ書きで付きます。通常、この識別子はセクション見出しの先頭に記述されますが、注意書きや他のテキストの先頭に記述されることもあります。

地域固有の見出しの例として、"(Latin America) Setting Up Depreciation" があります。

地域識別子

地域は地域名によって識別します。PeopleSoft オンライン ヘルプでは、次の地域識別子が使用される場合があります。

- Asia Pacific
- Europe
- Latin America
- North America

業種の識別子

業種は、業種名または業種の略称によって識別します。PeopleSoft オンライン ヘルプには、次の業種識別子が現れることがあります。

- USF (U.S. Federal)
- E&G (Education and Government)

PeopleSoft オンライン ヘルプの使用および管理

次のトピックに関する情報を参照するには、PeopleSoft オンライン ヘルプの任意のページのユニバーサル ナビゲーション ヘッダーで [ヘルプ] リンクをクリックします。

- PeopleSoft オンライン ヘルプの新機能。
- PeopleSoft オンライン ヘルプのアクセシビリティ。
- PeopleSoft オンライン ヘルプのアクセス、ナビゲーションおよび検索。
- ローカルにインストールされた PeopleSoft オンライン ヘルプ Web サイトの管理。

PeopleSoft エンタープライズ コンポーネントの関連リンク

[PeopleSoft 情報ポータル](#)

[My Oracle Support](#)

[Oracle University の PeopleSoft 研修](#)

連絡先

[提案を送信する](#) ご使用の PeopleTools およびアプリケーションのリリース番号を含めてください。

フォロー



Facebook で PeopleSoft の最新の更新を入手します。



Twitter@PeopleSoft_Info で PeopleSoft をフォローします。

第1章

アプリケーション インテグレーション フレームワーク の概要

アプリケーション インテグレーション フレームワークについて

この製品ドキュメントでは、アプリケーション インテグレーション フレームワークを使用して、共通フレームワークを基に、異種システム間のインテグレーションを作成する方法を説明します。

アプリケーション インテグレーション フレームワークの実装

アプリケーション インテグレーション フレームワークは、PeopleSoft インテグレーション ブローカー フレームワークを使用します。次の実装ステップでは、PeopleSoft インテグレーション ブローカーが設定され、サービス オペレーションがインテグレーション用に作成されていることを前提としています。

アプリケーション インテグレーション フレームワークを実装するには、以下の作業を行います。

- 値マップを定義します。
- ドメイン値マップをロードします。
- アプリケーション エンジン変換プログラムを作成します。
- サービス オペレーションのルーティングを更新します。

『PeopleTools: PeopleSoft Integration Broker』を参照してください。

値マップの定義

値マップを定義するには、以下のステップを実行します。

ステップ	参照
値マップの定義	「 値マップの定義 」を参照してください。

アプリケーション エンジン変換プログラムの作成

アプリケーション エンジン変換プログラムを作成するには、以下のステップを実行します。

ステップ	参照
アプリケーション エンジン プログラムの作成	『PeopleTools: PeopleSoft Integration Broker』の「Applying Filtering, Transformation and Translation」、「Defining Transform Programs」を参照してください。
XSLT ステップのコード化	『PeopleTools: PeopleSoft Integration Broker』の「Creating Transform Programs and Updating Service Operations」、「Adding XSLT Steps to Transformation Programs」を参照してください。

サービス オペレーションのルーティングの更新

サービス オペレーションのルーティングを更新するには、以下のステップを実行します。

ステップ	参照
ルーティングの更新	『PeopleTools: PeopleSoft Integration Broker』の「Creating Transform Programs and Updating Service Operations」、「Updating Service Operation Routing」を参照してください。

第2章

アプリケーション インテグレーション フレームワーク について

アプリケーション インテグレーション フレームワーク

アプリケーション インテグレーション フレームワークは、PeopleSoft インテグレーション ブロカー (IB) 機能を拡張し、データの表現、分類、保管、クエリー、パブリッシュ、取得、呼出しの標準的な手段を提供することで、PeopleSoft アプリケーション ビジネス メッセージ (ABM) とその他のアプリケーション間でエレメント名、ストラクチャおよび値をマッピングします。PeopleSoft インテグレーション ブロカーは、他の PeopleSoft システムやサードパーティ システムとのメッセージの送受信を行うためのフレームワークを提供します。システム間でメッセージ構造が異なる場合、変換プログラムを使用して、受信または送信メッセージを PeopleSoft システムが理解できるメッセージ フォーマットに変換します。

情報は同じでも、それを表すデータ値や ID が、インテグレーション対象のアプリケーションごとに異なる可能性があります。たとえば、PeopleSoft アプリケーションの新しい顧客は、*[PS1001]* などの一意の ID を使用して、新しい行として、顧客データベースに挿入されます。Oracle E-Business Suite アプリケーションと Siebel アプリケーションに同じ情報を移動すると、*[EBS1001]* と *[SBL1001]* のように、それぞれ別の ID で新しい行が挿入されます。アプリケーション インテグレーション フレームワークを使用すると、このデータを変換できます。

アプリケーション インテグレーション アーキテクチャ

アプリケーション インテグレーション アーキテクチャ (AIA) は、Oracle の サービス指向アーキテクチャ (SOA) に従い、共通のアーキテクチャとエンタープライズ ビジネス オブジェクト (EBO) と呼ばれるビジネス オブジェクトの共通の定義に基づき、サードパーティのアプリケーションも含めてアプリケーション間でビジネス プロセスのインテグレーションを行う統合アプローチとして、構築されています。これらのアプリケーションは異なるテクノロジーを使用して設計されているため、同じビジネス オブジェクトを表現していても使用する名前やストラクチャが異なります。AIA は、異種システム間で送信されるメッセージを変換し、関係するシステムごとにポイントツーポイント インテグレーションを確立することなく、複数のアプリケーションをインテグレーションできる基盤です。

AIA ミドルウェアを使用して、ビジネス オブジェクトを共通オブジェクトに変換できます。Oracle Fusion Middleware には、AIA と Oracle 製品向けの事前構築済のインテグレーションが含まれています。

Oracle Application Integration Architecture Foundation Pack

AIA プロセス インテグレーション パック (PIP) は、Siebel CRM、Oracle E-Business Suite、Agile PLM、Oracle Communications Billing、Revenue Management など、Oracle アプリケーション間の事前構築済インテグレーションのセットです。これらのインテグレーションは EBO と エンタープライズ ビジネス

サービス (EBS) で構成されています。エンタープライズ ビジネス メッセージ (EBM) は操作固有になるように設計されています。

AIA は、PeopleSoft システムとその他のサードパーティ システム間の中間層を提供します。サードパーティの Siebel の顧客が AIA を使用している場合、その顧客のメッセージが PeopleSoft システムに届くと、PeopleSoft アプリケーションから、Siebel の名前と値ではなく、共通の AIA 名と値で認識されます。一方、Siebel 顧客が AIA PIP を購入していない場合、PeopleSoft アプリケーションから認識されるのは Siebel の名前と値になります。

注: PIP は Oracle E-Delivery Web サイトから入手できます。

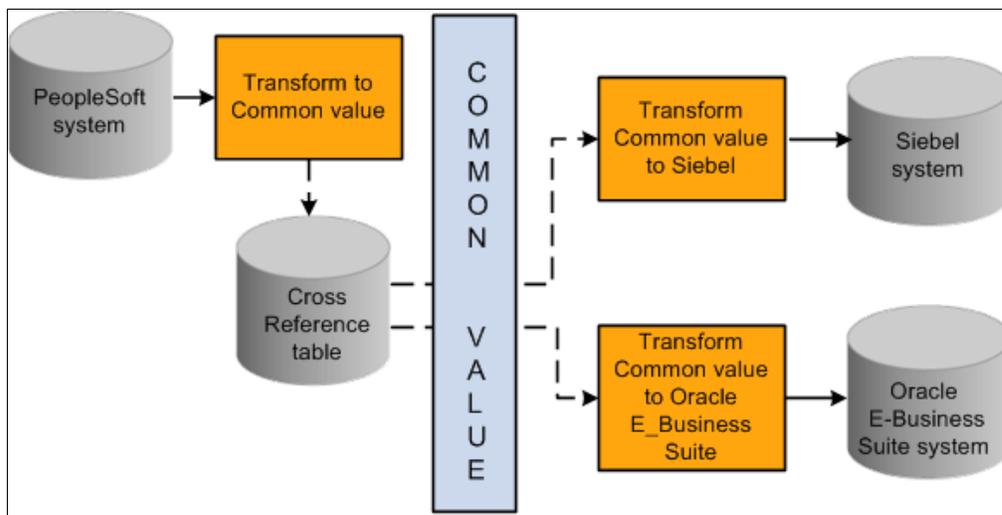
アプリケーション インテグレーション フレームワークによるインテグレーション

アプリケーション インテグレーション フレームワークは、AIA 対応のインテグレーションを作成する必要があるタスクの支援を目的としています。インテグレーションは、標準の (AIA) EBM と同じセマンティック コンテンツと、ほぼ同じ形式で ABM を作成するたため、変換要件が最小限に抑えられています。PeopleSoft アプリケーションが提供する各 AIA 対応インテグレーションには、インテグレーションをできるだけ厳密に EBO にマッピングできるように、サンプル変換が含まれています。この戦略によって、PeopleSoft をお使いの場合は、IB だけでパートナー インテグレーションを確立でき、ミドルウェアによって変換を実行する必要はありません。もちろん、Oracle AIA ミドルウェアを購入することにしたか、既に所有している顧客は、提供されている IB サンプルを使用する代わりに、またはそれらと併用して、EBS の機能を使用して変換を行うことができます。

注: PeopleSoft が提供するインテグレーションは、メッセージを EBM フォーマットに変換し、他の Oracle 製品との直接インテグレーションを実現します。

画像: 送信メッセージの共通値のマッピング

次の図は、PeopleSoft メッセージの共通値への変換を示しています。



アプリケーション インテグレーション フレームワークのユース ケース

ユース ケースは以下の 2 つのカテゴリに分類されます。

- 動的クロスリファレンス フレームワークを使用するキー マッピング変換
- ドメイン値マップ フレームワークを使用する静的値変換

この 2 つのフレームワークは別物だと見なされますが、実際には共通の要素があります。キーの値は自由形式で、通常、拡張されるため、変換前ではなく、変換プロセス中にプログラムを使用して作成されます。静的値変換では、通常、変換プロセス前に全ての値がマップに入力され、値が拡張される頻度は少なくなります。

この 2 つのカテゴリの他に、インテグレーションのユース ケースは、サードパーティが AIA ミドルウェアを使用しているかどうかにも左右されます。

次の表に、AIA ミドルウェアの使用時と未使用時の、変換が実行される状況を示します。

ユースケース	PeopleSoft インテグレーションブローカー	AIA ミドルウェア	サードパーティ
AIA ミドルウェアなしのインテグレーションブローカー変換	送信用に PeopleSoft ABM を EBM に変換 受信用に EBM を PeopleSoft ABM に変換	未使用	受信用に EBM をサードパーティの ABM に変換 送信用にサードパーティの ABM を EBM に変換
インテグレーションブローカーによるポイントツーポイント変換	送信用に PeopleSoft ABM をサードパーティの ABM に変換 送信用に サードパーティの ABM を PeopleSoft ABM に変換	未使用	変換不要 オプションで、サードパーティは PeopleSoft ABM への送信用変換を実行可能
サードパーティが AIA ミドルウェアを使用するインテグレーションブローカー変換	送信用に PeopleSoft ABM を EBM に変換 受信用に EBM を PeopleSoft ABM に変換	PeopleSoft アプリケーションから受信用に EBM をサードパーティの ABM に変換 PeopleSoft アプリケーションへの受信用にサードパーティの ABM を EBM に変換	変換の実行なし

マップ

ドメイン値マップ (DVM) とクロスリファレンス マップ (XREF) の 2 種類のマップを使用できます。これらのマップは、どちらもドメイン名を基に要素をグループ化し、ドメイン間で特定の値をマッピングする要素で構成されているという点は同じです。技術的な観点での違いは、XREF 値は常に作成され続けるのに対し、DVM 値は比較的静的であることです。

XREF	DVM
通常、キー	通常、属性
データはプログラムを使用して管理	データはユーザー インターフェイスから手動で管理

マップのカーディナリティによって、マッピング機能を個別に分類できます。マップは、単一の値 (1:1) または複数值 (N:N) ベースで作成されます。

ドメイン値マップ

ドメイン値マップ (DVM) は、国コードや都道府県/州など、比較的静的で、合計数が比較的限られた値に使用されます。DVM では、通常、変換プロセス前に全ての値がマップに入力され、クロスリファレンスマップと比べて、値が拡張される頻度は少なくなります。

DVM は、ドメイン、マップ、およびエレメントで構成されます。

ドメイン

インテグレーション対象のシステム (Siebel や Oracle E-Business Suite など)

注: 1 つのマップに対して複数のドメインを管理できます。

マップ

マッピングの管理が必要な静的オブジェクト (通貨コードや国コードなど)。

エレメント

最小要素で実行されるローカルまたはリモート メッセージのデータの単位。最小要素は、これ以上細かい変換に分解できないレベルの要素のことです。通常、エレメントに含まれるデータ値は 1 つだけですが、そうでない場合もあります。たとえば、住所は 1 つの文字列として表現されることも、市、都道府県、番地、および家タイプを表す個別の文字列で構成されることもあります。DVM エレメントの例は、国コードと共通値です。

クロスリファレンス マップ

クロスリファレンス マップは、インスタンスの識別に使用されるキー エレメントなど、動的な性質の値に使用されます。これらは、キー マッピングやクロスリファレンスと呼ばれます。より大きなデータ エンティティの名前が、値になります。

クロスリファレンスは、ドメイン、マップ、およびエレメントで構成されます。

ドメイン

インテグレーション対象のシステム (Siebel や E-Business Suite など)

注: 1 つのマップに対して複数のドメインを管理できます。

マップ	キーのクロスリファレンスの管理対象になるトランザクション オブジェクト (伝票や取引先など)。
エレメント	キー情報の一部として取得された個々のエレメント (SETID、VENDOR_ID など)。

クロスリファレンスと DVM をロードおよび管理するための関数

変換プロセス中に DVM およびクロスリファレンス データのクエリーおよび管理に使用できる関数が 3 種類あります。

アプリケーション クラス メソッド アプリケーション クラス メソッドは、PeopleCode で使用します。これらのクラスは、通常、アプリケーション エンジン プログラムにおいて PeopleCode ステップとして変換を実装する場合に使用します。このような PeopleCode クラスは、XPath 拡張関数と Web サービスによって内部的に使用されます。

XPath 拡張関数 XPath 拡張関数は、アプリケーション エンジン プログラムの XSLT ステップで使用され、TransformEx PeopleCode API を使用して XSLT 変換を呼び出します。

Web サービス 外部システムがクロスリファレンス検索を実行するために使用します。

これらの関数を使用することで、返還中にクロスリファレンス (XREF) データと DVM データのクエリー、操作、および削除を実行できます。関数のカーディナリティは、API 命名規則に組み込まれています。各関数セットは、さらに、検索、ロード、および削除の 3 種類のアクティビティに分けることができます。

提供されている関数をクラス、アクティビティ、およびフォームにより分類すると、次のようになります。

クラス	アクティビティ	フォーム	説明
XREF	ロード	populateValue	単一値エレメントの変換アイテムをロードします。
XREF	ロード	populateValue1M	1 対多マッピングの変換エレメントをロードします。
XREF	ロード	populateValueNVP	複数値エレメントの変換アイテムをロードします。
XREF	削除	markForDelete	単一値エレメントの変換エレメントを削除指定します。削除指定したアイテムは、後で再アクティブ化できます。

クラス	アクティビティ	フォーム	説明
XREF	削除	markForDeleteNVP	複数値エレメント (名前と値のペア) の変換エレメントを削除指定します。
XREF	検索	lookupValue	クロスリファレンス値を検索します。
XREF	検索	lookupValue1M	参照エレメントの特定の1つの値に対応する複数値 (1対多) のクロスリファレンスエレメントを検索します。
XREF	検索	lookupValueNVP	複数値エレメントのクロスリファレンス値を検索します。
DVM	検索	lookupValue	ドメイン値を検索します。
DVM	検索	lookupValue1M	参照エレメントの特定の1つの値に対応する複数のドメイン値 (1対多) を検索します。
DVM	検索	lookupValueNVP	複数値エレメントのドメイン値を検索します。
DVM	検索	lookup-dvm	ドメイン値を検索します。

ユース ケース: AIA ミドルウェアなしのインテグレーション ブローカー変換

このセクションでは、アプリケーション インテグレーション フレームワークによって、PeopleSoft インテグレーション ブローカー内で PeopleSoft およびサードパーティの両方のマップ変換が実行されるユース ケースについて説明します。このユース ケースでは、AIA ミドルウェアを購入しなくても、PeopleSoft アプリケーションから標準のインテグレーション モデルを利用できます。

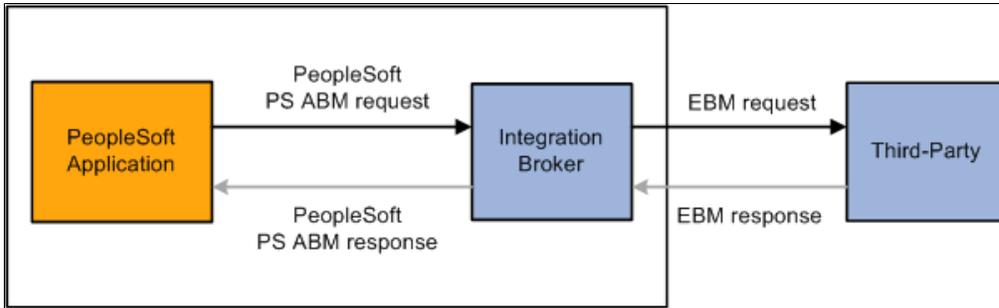
次のユース ケースがあります。

- サードパーティへのアウトバウンド リクエストまたはアウトバウンド送信
- サードパーティからのインバウンド リクエストまたはインバウンド送信

サードパーティへのアウトバウンド リクエストまたはアウトバウンド送信

画像: AIA ミドルウェアなしでのサードパーティへのアウトバウンド リクエスト

次の図は、サードパーティへのアウトバウンド リクエストまたはアウトバウンド送信を示しています。



このシナリオでは、次のステップが実行されます。

1. PeopleSoft システムのユーザーが、サードパーティ システムへの AIA 対応インテグレーションのパブリッシングをトリガします。
2. PeopleSoft インテグレーション ブローカー内で、アウトバウンド ルーティングに対する変換によって、EBM が生成されます。データ値の変換に使用されるメソッドは、トランザクションによって決まります。次の表に、トランザクション タイプと必要な変換アクションを示します。

トランザクション	アクション
動的なキーの追加リクエスト	トランザクションが追加リクエストの場合、変換によって新しい共通キー (GUID) が作成され、適切な形式の populate XRef XPath 拡張関数を使用して、新しい共通キーとそれに対応する PeopleSoft キーが PeopleSoft クロスリファレンス フレームワークに入力されます。
動的なキーの検索	トランザクションが追加リクエストではない場合、変換では、PeopleSoft キーを入力とし、適切な形式の lookup XRef XPath 拡張関数を使用して、共通キーを検索します。共通キーが存在しない場合、新しい共通キーを作成する、エラーを生成する、または、インテグレーションがサポートする場合は返信メッセージでエラー メッセージを返すよう、開発者が指定します。
動的なキーの削除リクエスト	キーとなるエンティティを削除する非同期通知 (リクエスト専用) インテグレーションの場合、開発者は履歴目的でクロスリファレンス値をそのまま残すことも、必要に応じてクロスリファレンス値を削除することもできます。PeopleSoft キーを削除指定する場合、変換では、PeopleSoft キーを入力として、適切な形式の deleteXRef XPath 拡張関数を使用します。

トランザクション	アクション
静的な値の検索	トランザクションがドメイン値マップを使用する場合、変換では、PeopleSoft 値を入力として、適切な形式の lookupDVM XPath 拡張関数を使用して、PeopleSoft 値を検索します。値が見つからない場合、変換によってデフォルトで PeopleSoft 値を提供する、空白のままにする、エラーをスローする、または、インテグレーションがサポートする場合は、返信メッセージでエラー メッセージを返すように開発者が指定します。

3. EBM は処理のためにサードパーティシステムに転送されます。

レスポンス メッセージをサポートするインテグレーションには、追加のステップがあります。

1. サードパーティシステムは、リクエストを処理し、返信メッセージのフォーマットを整え、共通キーまたは値と共に返信メッセージを返します。
2. PeopleSoft インテグレーション ブローカー内で、非同期のリクエスト/返信処理において、共通キーを入力として、適切な形式の **lookupXref XPath** 拡張関数を使用して、PeopleSoft キーを検索する変換が実行されます。その後、PeopleSoft キーが PeopleSoft ABM に格納されます。同期処理の場合は、既に初期リクエストによって PeopleSoft キーが PeopleSoft アプリケーションに認識されているので、この検索は必要ありません。

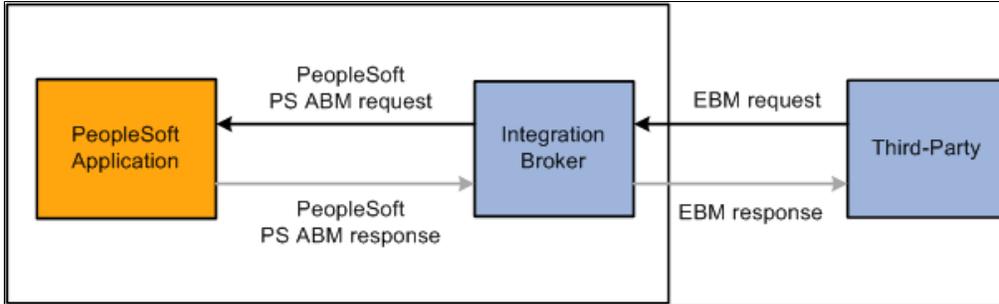
トランザクション	アクション
動的なキー値を使用する非同期のリクエスト/返信処理	共通キーを入力とし、適切な形式の lookupXref XPath 拡張関数を使用して、PeopleSoft キーを検索する変換プログラムが必要です。その後、PeopleSoft キーが PeopleSoft ABM に格納されます。
動的なキー値を使用する同期処理	既に初期リクエストによって PeopleSoft キーが PeopleSoft アプリケーションに認識されているので、検索は必要ありません。
DVM 値を使用する非同期のリクエスト/返信処理	共通値を入力として、適切な形式の lookupDVM XPath 拡張関数を使用して、PeopleSoft 値を検索する変換プログラムが必要です。その後、PeopleSoft キーが PeopleSoft ABM に格納されます。

3. その後、PeopleSoft ABM 返信が、リクエストを送った PeopleSoft アプリケーションに返されます。

サードパーティからのインバウンド リクエストまたはインバウンド送信

画像: AIA ミドルウェアなしでのサードパーティからのインバウンド リクエスト

次の図は、サードパーティからのインバウンド リクエストまたはインバウンド送信を示しています。



1. サードパーティ システムのユーザーが、PeopleSoft システムへのインテグレーションのパブリッシングをトリガします。
2. EBM は処理のために PeopleSoft システムに転送されます。
3. PeopleSoft インテグレーション ブローカー内で、インバウンド ルーティングに対する変換によって、トランザクション タイプに応じて次のアクションが実行されます。

トランザクション	アクション
動的なキーの追加リクエスト	トランザクションが追加リクエストの場合、変換では、ABM 内の PeopleSoft キーを空白のままにし、ヘッダーで共通キーを使用するために渡します。
動的なキーの検索	トランザクションが追加リクエストではない場合、変換では、EBM の共通キーを入力とし、適切な形式の lookup XRef XPath 拡張関数を使用して、PeopleSoft キーを検索します。オプションで、この処理は、共通キーを入力として、適切な PeopleSoft lookup XRef API を利用して、PeopleSoft インバウンド ハンドラ内でも実行できます。
DVM 検索	インバウンド ルーティングに対する変換によって、EBM の共通値を使って適切な形式の lookupDVM XPath 拡張関数を使用して、PeopleSoft 値が検索されます。値が見つからない場合、変換によって PeopleSoft 値を提供する、省略する、エラーをスローする、または、インテグレーションがサポートする場合は、返信メッセージでエラー メッセージを返すように開発者が指定します。

4. PeopleSoft インバウンド ハンドラによって、メッセージが処理されます。

トランザクション	アクション
動的なキーの追加リクエスト	トランザクションが追加リクエストの場合、PeopleSoft キーが作成され、適切な形式の populate XRef API を使用して、PeopleSoft キーとそれに対応する共通キーが PeopleSoft クロスリファレンス フレームワークに追加されます。キーを作成できない場合やフレームワークに追加できない場合は、エラーが発生するか、インテグレーションがサポートする場合は返信メッセージでエラー メッセージが返されます。
動的なキーの検索	トランザクションが追加リクエストではない場合、メッセージは ABM の PeopleSoft キーを使用して処理されません。PeopleSoft キー値が見つからない場合、エラーが発生するか、インテグレーションがサポートする場合は返信メッセージでエラー メッセージが返されます。

レスポンス メッセージをサポートするインテグレーションには、追加のステップがあります。

1. PeopleSoft アプリケーションがリクエストを処理し、ABM 返信を PeopleSoft インバウンド ハンドラに返します。
2. PeopleSoft インバウンド ハンドラは EBM 返信のフォーマットを整え、共通キーを使用してサードパーティ システムに返します。
3. オプションで、サードパーティ システムがキー マッピングの維持に PeopleSoft クロスリファレンス フレームワークを使用している場合、非同期リクエスト/返信処理では、EBM 返信の共通キーを入力とし、適切な形式の **lookup XRef** Web サービスを使用して、サードパーティのキーを検索する必要があります。

ユース ケース: インテグレーション ブローカーによるポイントツーポイント変換

このセクションでは、PeopleSoft インテグレーション ブローカーによって、サードパーティとの変換の全てを実行するユース ケースについて説明します。前のユース ケースでは、メッセージは EBM フォーマットに変換されましたが、このユース ケースでは、メッセージはサードパーティの ABM に変換されます。サードパーティが別の PeopleSoft システムの場合は、変換は不要です。

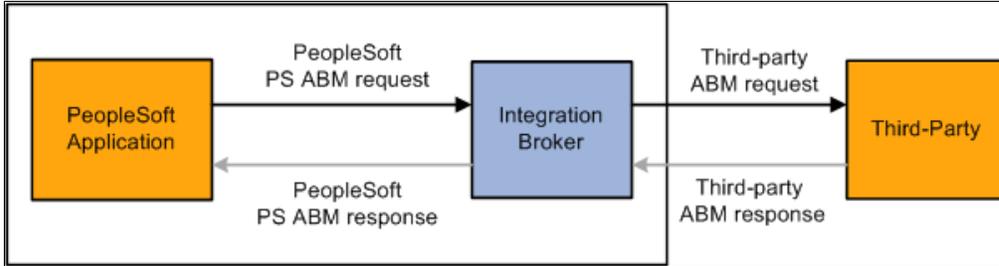
次のユース ケースがあります。

- サードパーティへのアウトバウンド リクエストまたはアウトバウンド送信
- サードパーティからのインバウンド リクエストまたはインバウンド送信

サードパーティへのアウトバウンド リクエストまたはアウトバウンド送信

画像: サードパーティへのアウトバウンドのポイントツーポイント リクエスト

次の図は、サードパーティへのアウトバウンドのポイントツーポイントのリクエストまたは送信を示しています。



このシナリオでは、次のステップが実行されます。

1. PeopleSoft システムのユーザーが、サードパーティ システムへのポイントツーポイント インテグレーションのパブリッシングをトリガします。
2. PeopleSoft インテグレーション ブローカー内で、アウトバウンド ルーティングに対する変換によって、サードパーティの ABM が生成されます。データ値の変換に使用されるメソッドは、トランザクションによって決まります。次の表に、トランザクション タイプと必要な変換アクションを示します。

トランザクション	アクション
動的なキーの追加リクエスト	トランザクションが追加リクエストの場合、変換によって新しい共通キー (GUID) が作成され、適切な形式の populate XRef XPath 拡張関数を使用して、新しい共通キー、それに対応する PeopleSoft キー、およびサードパーティのキーが PeopleSoft クロスリファレンス フレームワークに入力されます。
動的なキーの検索	トランザクションが追加リクエストではない場合、変換では、PeopleSoft キーを入力とし、適切な形式の lookup XRef XPath 拡張関数を使用して、キーを検索します。共通キーが存在しない場合、新しい共通キーを作成する、エラーを生成する、または、インテグレーションがサポートする場合は返信メッセージでエラー メッセージを返すよう、開発者が指定します。
動的なキーの削除リクエスト	キーとなるエンティティを削除する非同期通知 (リクエスト専用) インテグレーションの場合、開発者は履歴目的でクロスリファレンス値をそのまま残すことも、必要に応じてクロスリファレンス値を削除することもできます。PeopleSoft キーを削除指定する場合、変換では、PeopleSoft キーを入力として、適切な形式の deleteXRef XPath 拡張関数を使用します。

トランザクション	アクション
静的な値の検索	トランザクションがドメイン値マップを使用する場合、アウトバウンド ルーティングでの変換では、PeopleSoft 値を入力として、適切な形式の lookupDVM XPath 拡張関数を使用して、共通値を検索します。値が見つからない場合、変換によってデフォルトで PeopleSoft 値を提供する、省略する、エラーをスローする、または、インテグレーションがサポートする場合は、返信メッセージでエラー メッセージを返すように開発者が指定します。

3. サードパーティの ABM は処理のためにサードパーティ システムに転送されます。

レスポンス メッセージをサポートするインテグレーションには、追加のステップがあります。

1. サードパーティ システムは、リクエストを処理し、返信メッセージのフォーマットを整え、共通キーまたは値と共に返信メッセージを返します。
2. PeopleSoft インテグレーション ブローカー内で、非同期のリクエスト/返信処理において、サードパーティのキーを入力として、適切な形式の検索用 **XRef** XPath 拡張関数を使用して、PeopleSoft キーを検索する変換が実行されます。その後、PeopleSoft キーが PeopleSoft ABM に格納されます。同期処理の場合は、既に初期リクエストによって PeopleSoft キーが PeopleSoft アプリケーションに認識されているので、この検索は必要ありません。

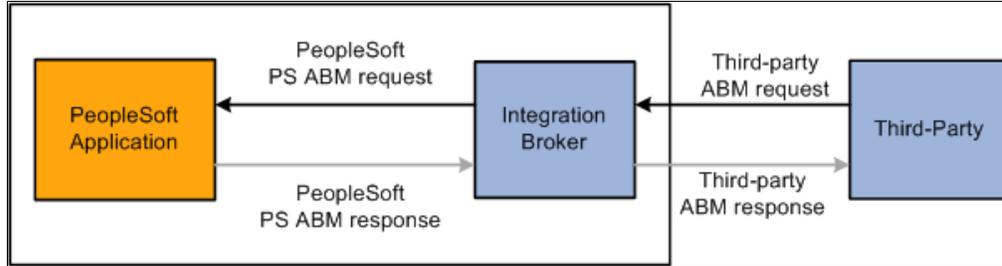
トランザクション	アクション
動的なキー値を使用する非同期のリクエスト/返信処理	サードパーティのキーを入力とし、適切な形式の lookupXRef XPath 拡張関数を使用して、PeopleSoft キーを検索する変換プログラムが必要です。その後、PeopleSoft キーが PeopleSoft ABM に格納されます。
動的なキー値を使用する同期処理	既に初期リクエストによって PeopleSoft キーが PeopleSoft アプリケーションに認識されているので、検索は必要ありません。
DVM 値を使用する非同期のリクエスト/返信処理	サードパーティの値を入力とし、適切な形式の lookupDVM XPath 拡張関数を使用して、PeopleSoft 値を検索する変換プログラムが必要です。その後、PeopleSoft キーが PeopleSoft ABM に格納されます。

3. PeopleSoft ABM 返信が、リクエストを送った PeopleSoft アプリケーションに返されます。

サードパーティからのインバウンド リクエストまたはインバウンド送信

画像: サードパーティへのインバウンドのポイントツーポイント リクエスト

次の図は、サードパーティからのインバウンドのポイントツーポイントのリクエストまたは送信を示しています。



1. サードパーティ システムのユーザーが、PeopleSoft システムへのインテグレーションのパブリッシングをトリガします。
2. この例では、リクエストはサードパーティの ABM フォーマットで送信されます。

注: 多くの場合、サードパーティは PeopleSoft ABM フォーマットを認識し、メッセージを送信する前に変換を実行します。この場合では、変換は不要です。

3. PeopleSoft インテグレーション ブローカー内で、インバウンド メッセージに対する変換によって、トランザクション タイプに応じて次のアクションが実行されます。

トランザクション	アクション
動的なキーの追加リクエスト	トランザクションが追加リクエストの場合、変換時に ABM 内の PeopleSoft キーは空白のままになり、サードパーティのキーをヘッダーに含めて渡し、これが使用されるようになります。
動的なキーの検索	トランザクションが追加リクエストではない場合、変換では、サードパーティの ABM 内のリファレンス キーを入力とし、適切な形式の lookup XRef XPath 拡張関数を使用して、PeopleSoft キーを検索します。
DVM 検索	インバウンド ルーティングに対する変換では、サードパーティの ABM のリファレンス キーを使い、適切な形式の lookupDVM XPath 拡張関数を使用して、PeopleSoft 値を検索します。値が見つからない場合、変換によって PeopleSoft 値を提供する、省略する、エラーをスローする、または、インテグレーションがサポートする場合は、返信メッセージでエラー メッセージを返すように開発者が指定します。

4. PeopleSoft インテグレーション ブローカー内で、インバウンド メッセージに対する変換によって、トランザクション タイプに応じて次のアクションが実行されます。

5. PeopleSoft インバウンド ハンドラによって、メッセージが処理されます。

トランザクション	アクション
動的なキーの追加リクエスト	トランザクションが追加リクエストの場合、PeopleSoft キーが作成され、適切な形式の populate XRef API を使用して、PeopleSoft キーとそれに対応する UniqueGUID が PeopleSoft クロスリファレンス フレームワークに追加されます。キーを作成できない場合やフレームワークに追加できない場合は、エラーが発生するか、インテグレーションがサポートする場合は返信メッセージでメッセージが返されます。
動的なキーの検索	トランザクションが追加リクエストではない場合、メッセージは ABM の PeopleSoft キー値を使用して処理されます。PeopleSoft キー値が見つからない場合、エラーが発生するか、インテグレーションがサポートする場合は返信メッセージでエラー メッセージが返されます。

レスポンス メッセージをサポートするインテグレーションには、追加のステップがあります。

1. PeopleSoft アプリケーションはリクエストを処理し、PeopleSoft インバウンド ハンドラが ABM 返信のフォーマットを整えて、共通キーまたは値と共に返信を返します。
2. この変換においてキーの変換は必要ないので、EBM 返信が ABM 返信の共通キーと共にサードパーティシステムに返されます。

ユース ケース: サードパーティが AIA ミドルウェアを使用するインテグレーション ブローカー変換

PeopleSoft からすると、このユース ケースは AIA ミドルウェアなしの変換と同じフローになります。しかし、サードパーティからすると、フローは似ていますが、実装では異なります。このケースでは、サードパーティシステムは、EBM を使用して PeopleSoft ソフトウェアとのインテグレーションを確立するのではなく、サードパーティの ABM を使用して AIA とのインテグレーションを確立します。全てのサードパーティの値マップと、EBM およびサードパーティの ABM 間の変換は、サードパーティシステムではなく、AIA レイヤーにおいて EBS グラフィカル マッパーと XPath 拡張関数によって処理されます。

注: これは、AIA PIP で使用されるモデルです。

次のユース ケースがあります。

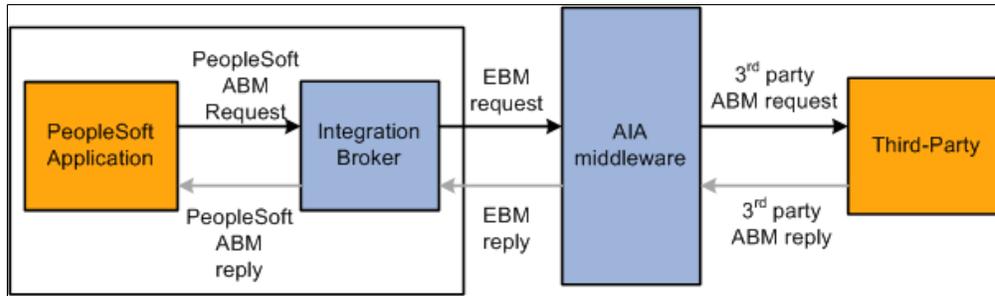
- サードパーティへのアウトバウンド リクエストまたはアウトバウンド送信
- サードパーティからのインバウンド リクエストまたはインバウンド送信

サードパーティへのアウトバウンド リクエストまたはアウトバウンド送信

このユース ケースでは、PeopleSoft アプリケーションからサードパーティにリクエストが送信され、サードパーティは自身の AIA ミドルウェアを使用してメッセージ変換を実行します。

画像: AIA ミドルウェアを使用するサードパーティへのアウトバウンド リクエスト

次の図は、AIA ミドルウェアを使用するサードパーティへのアウトバウンド リクエストまたはアウトバウンド送信を示しています。



1. PeopleSoft システムのユーザーが、AIA を介して、サードパーティシステムへの AIA 対応インテグレーションのパブリッシングをトリガします。
2. PeopleSoft インテグレーション ブローカー内で、アウトバウンド ルーティングに対する変換が実行され、EBM が生成されます。データ値の変換に使用されるメソッドは、トランザクションによって決まります。次の表に、トランザクション タイプと必要な変換アクションを示します。

トランザクション	アクション
動的なキーの追加リクエスト	トランザクションが追加リクエストの場合、変換によって新しい共通キー (GUID) が作成され、適切な形式の populateXRef XPath 拡張関数を使用して、新しい共通キーとそれに対応する PeopleSoft キーが PeopleSoft クロスリファレンス フレームワークに入力されます。
動的なキーの検索	トランザクションが追加リクエストではない場合、変換では、PeopleSoft キーを入力とし、適切な形式の lookupXRef XPath 拡張関数を使用して、共通キーを検索します。共通キーが存在しない場合、新しい共通キーを作成する、エラーを生成する、または、インテグレーションがサポートする場合は返信メッセージでエラー メッセージを返すよう、開発者が指定します。
動的なキーの削除リクエスト	キーとなるエンティティを削除する非同期通知 (リクエスト専用) インテグレーションの場合、開発者は履歴目的でクロスリファレンス値をそのまま残すことも、必要に応じてクロスリファレンス値を削除することもできます。PeopleSoft キーを削除指定する場合、変換では、PeopleSoft キーを入力として、適切な形式の deleteXRef XPath 拡張関数を使用します。

トランザクション	アクション
DVM 検索	トランザクションがドメイン値マップを使用する場合、変換では、EBM の共通値を使い、適切な形式の lookupDVM XPath 拡張関数を使用して、PeopleSoft 値を検索します。値が見つからない場合、変換によってデフォルトで PeopleSoft 値を提供する、省略する、エラーをスローする、または、インテグレーションがサポートする場合は、返信メッセージでエラーを返すように開発者が指定します。

- EBM は AIA に転送されます。
- EBM リクエストを受信すると、AIA は共通キーをサードパーティのキーに変換し、サードパーティの ABM を使用して変換したリクエストをサードパーティに送信します。

レスポンス メッセージをサポートするインテグレーションには、追加のステップを実行します。

- サードパーティはリクエストを処理し、返信メッセージのフォーマットを整え、返信メッセージを AIA ミドルウェアに返します。
- AIA は、返信内のサードパーティのキーを共通キーに変換して、EBM 返信を PeopleSoft ソフトウェアに返します。
- PeopleSoft インテグレーション ブローカー内で、非同期のリクエスト/返信処理において、共通キーまたは共通値を変換して、PeopleSoft ABM に格納する変換プログラムが実行されます。同期処理の場合は、既に初期リクエストによって PeopleSoft のキーまたは値が PeopleSoft アプリケーションに認識されているので、この検索は必要ありません。

トランザクション	アクション
動的なキーの検索	変換によって、共通キーを入力とし、適切な形式の lookup XRef XPath 拡張関数を使用して、PeopleSoft キーが検索されます。
動的なキーの削除	(オプション) キーとなるエンティティを削除するインテグレーションの場合、開発者は履歴目的でクロスリファレンス値をそのまま残すことも、必要に応じてクロスリファレンス値を削除することもできます。PeopleSoft キーを削除指定する場合、変換では、PeopleSoft キーを入力として、適切な形式の deleteXRef XPath 拡張関数を使用します。
DVM 検索	変換によって、PeopleSoft 値を入力とし、適切な形式の検索用 DVM XPath 拡張関数を使用して、共通値が検索されます。

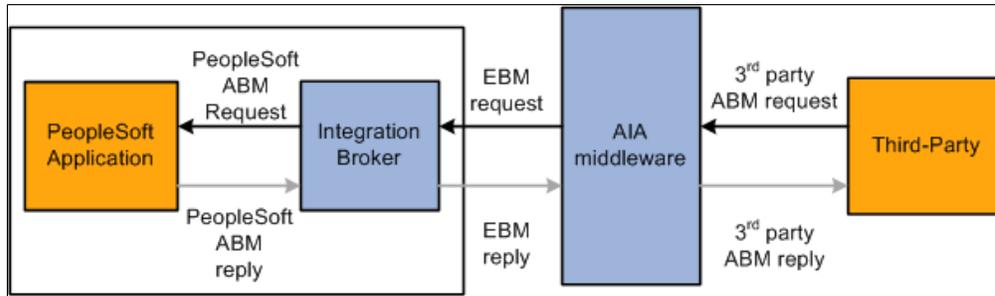
- PeopleSoft ABM 返信が、リクエストを送った PeopleSoft アプリケーションに返されます。

サードパーティからのインバウンド リクエストまたはインバウンド送信

このユース ケースでは、サードパーティのアプリケーションが、サードパーティの AIA ミドルウェアを使用して変換されるリクエストを作成し、PeopleSoft アプリケーションに送信します。

画像: AIA ミドルウェアを使用するサードパーティからのインバウンド リクエスト

次の図は、AIA ミドルウェアを使用するサードパーティからのインバウンド リクエストまたはインバウンド送信を示しています。



1. サードパーティ システムのユーザーが、AIA を介して、PeopleSoft システムへのインテグレーションのパブリッシングをトリガします。
2. AIA は、サードパーティの ABM キーおよび値を共通キーおよび値に変換して、EBM リクエストを PeopleSoft システムに処理のために送信します。
3. PeopleSoft インテグレーション ブローカー内で、インバウンド ルーティングに対する変換によって、トランザクションに応じて次の方法の 1 つ以上を使用して、EBM が PeopleSoft ABM に変換されます。

トランザクション	アクション
動的なキーの追加	トランザクションが追加リクエストの場合、変換では、ABM 内の PeopleSoft キーを空白のままにし、ヘッダーで共通キーを使用するために渡します。
動的なキーの検索	トランザクションが追加リクエストではない場合、変換では、EBM の共通キーを入力とし、適切な形式の lookupXRef XPath 拡張関数を使用して、PeopleSoft キーを検索します。オプションで、この処理は、共通キーを入力として、適切な PeopleSoft 検索用 Xref API を利用して、PeopleSoft インバウンド ハンドラ内でも実行できます。
DVM 検索	インバウンド ルーティングに対する変換によって、EBM の共通値を使って適切な形式の検索用 DVM XPath 拡張関数を使用して、PeopleSoft 値が検索されます。値が見つからない場合、変換によってデフォルトで PeopleSoft 値を提供する、空白のままにする、エラーをスローする、または、インテグレーションがサポートする場合は、返信メッセージでエラーメッセージを返すように開発者が指定します。

同期および非同期のリクエスト/返信メッセージの場合、次の追加のステップが実行されます。

1. PeopleSoft アプリケーションはリクエストを処理し、PeopleSoft インバウンド ハンドラが ABM 返信のフォーマットを整えて、共通キーまたは値と共に返信を返します。
2. この IB 変換においてキーの変換は必要ないので、EBM 返信は AIA に返され、そこからサードパーティシステムに返送されます。
3. AIA は、共通キーおよび値をサードパーティの ABM キーおよび値に変換して、サードパーティの ABM 返信をサードパーティシステムに返送します。

ドメイン値マップの定義とロード

値マップについて

値マップの定義コンポーネントを使用すると、値マップとエレメントを定義および分類できます。各値マップは動的 (クロスリファレンス) または静的 (DVM) のいずれかに定義する必要があります。

値マップは、次のマップのタイプをサポートします。

- 1 対 1
- 1 対多
- ドメインごとの複数のエレメント

1 対 1 のクロスリファレンス

クロスリファレンス マップは、2つのシステム間の 1 対 1 関係をサポートします。たとえば、PeopleSoft システムでは 顧客 A に対して特定の ID を使用し、Enterprise Business Object (EBO) または共通値では GUID を使用し、外部システムでは同じ顧客に対して別の ID を使用します。

このクロスリファレンス マップは、次のように定義されます。

PeopleSoft	UniqueGUID	外部システム
PS	UniqueGUID	EXT
PS100	<guid1>	EXT-100
PS102	<guid2>	EXT-102

このシナリオでは、PeopleSoft アプリケーションが新規顧客を作成し、顧客の作成メッセージを外部システムに送信する際、ルーティング中に、変換プログラムがクロスリファレンス マップを使用したデータ変換を実行します。

1 対 多のクロスリファレンス

クロスリファレンス マップは、2つのシステム間の 1 対 多 関係をサポートします。あるシステム内の複数の値が、別のシステムの 1 つの値に対応している場合があります。たとえば、PeopleSoft システムに存在する 3 種類の職務コードが、外部システムの 1 つの職務コードに対応している可能性があります。

このクロスリファレンス マップは、次のように定義されます。

PeopleSoft	UniqueGUID	外部システム
PS	UniqueGUID	EXT
AS01	<guid1>	ASST
AS02	<guid2>	
AS03	<guid3>	
MG01	<guid4>	MNGR

このシナリオでは、PeopleSoft アプリケーションが職務コードを割り当てて変更し、メッセージを外部システムに送信する際、ルーティング中に、変換プログラムがクロスリファレンス マップを使用したデータ変換を実行します。インテグレーションにおいて、外部システム値を PeopleSoft 値に変換する必要がある場合、検索から返される複数の PeopleSoft の値をどのように処理するかを開発者が定義する必要があります。その方法としては、各値のソース データの複製、デフォルト値を選択するメソッドの実装、およびトランザクションに対するエラーの生成が含まれます。

複数のドメインがあるクロスリファレンス

システムは複合キーを使用して機能を実装することがよくあります。クロスリファレンス フレームワークでは、複数のエレメントによって値セットを特定することで、この処理をサポートします。たとえば、PeopleSoft アプリケーションから外部システムへのアイテムのインテグレーションでは、クロスリファレンス マップを使用して、PeopleSoft システムのセットID/アイテム ID を外部システムの "製品" に変換します。

このクロスリファレンス マップは、次のように定義されます。

PeopleSoft	PeopleSoft	UniqueGUID	外部システム
セットID	アイテム ID	UniqueGUID	製品
SHARE	1001	<guid1>	RP001
SHARE	1002	<guid2>	RP002
SHARE	1003	<guid3>	RP003
SHR03	1000	<guid4>	RP006

このシナリオでは、PeopleSoft アプリケーションがアイテムを作成し、メッセージを外部システムに送信する際、ルーティング中に、変換プログラムがセットID およびアイテム ID を共通エレメント (UniqueGUID) にマッピングします。外部システムは変換されたデータを受け取ります。

1 対 1 の DVM

ドメイン値マップでは、1 対 1 関係には実際のデータ値が含まれます。たとえば、PeopleSoft アプリケーションは州コードに略称を使用し、EBO は正式名を使用して共通値として州コードを定義し、外部アプリケーションは州コードの略称を使用します。

この DVM の定義の例は、次のとおりです。

PeopleSoft	共通	外部システム
略称	正式名	略称
MA	マサチューセッツ	Mass
CA	カリフォルニア	Calif

このシナリオでは、PeopleSoft アプリケーションが外部システムへの州コードを含むアウトバウンド メッセージを作成する際、ルーティング中に、変換プログラムが略称を正式名にマッピングします。外部システムでは、正式名から略称への変換が必要になります。外部システム向けの変換は、AIA ミドルウェアまたは外部システム固有の変換によって実行できます。

1 対 多の DVM

ドメイン値マップの場合、1 対多関係は、別のドメインの 1 つの値にマッピングする、あるドメインの複数の値を入力することで作成されます。たとえば、PeopleSoft アプリケーションの複数の個人タイプを、外部アプリケーションの 1 つの個人タイプにマッピングできます。

この DVM の定義の例は、次のとおりです。

PeopleSoft	UniqueGUID	外部システム 2
PS	UniqueGUID	EXT2
PS001 PS002	<guid1>	SB001
PS003	<guid2>	SBL002

このシナリオでは、PeopleSoft アプリケーションが外部システムへの個人タイプを含むアウトバウンド メッセージを作成する際、ルーティング中に、変換プログラムがデータ値を変換します。

ドメインごとに複数のエレメントを持つ DVM

DVM では、複数のエレメントをマッピングして、1 つの値セットを定義できます。たとえば、PeopleSoft アプリケーションにあるビジネス ユニット/チャートフィールドの組合せが、外部アプリケーションにある元帳/セグメントの組合せにマッピングされることがあります。

このシナリオでは、PeopleSoft アプリケーションが外部システムへの個人タイプを含むアウトバウンドメッセージを作成する際、ルーティング中に、変換プログラムがデータ変換を実行します。

この DVM の定義の例は、次のとおりです。

PeopleSoft	PeopleSoft	UniqueGUID	外部システム 1	
ビジネス ユニット	チャートフィールド	UniqueGUID	セグメント	元帳
US001	ACCOUNT	<guid1>	SEGMENT1	元帳 1
US001	DEPARTMENT	<guid2>	SEGMENT2	元帳 1
US001	PRODUCT	<guid3>	SEGMENT3	元帳 1
US002	ACCOUNT	<guid4>	SEGMENT1	元帳 2

このシナリオでは、ビジネス ユニット/チャートフィールドを含む外部システムへのアウトバウンドメッセージをPeopleSoft アプリケーションで作成する際、複数エレメントのデータ値を1つの UniqueGUID に変換する変換プログラムがルーティングに含められます。その後、外部システムは、UniqueGUID を同等の "セグメント/元帳" に変換する必要があります。

複数の識別子を持つ DVM

識別子は、マッピング値を一意に識別するための追加のコンテキストを提供します。追加のコンテキスト情報によって修飾されないと、マッピングは有効にならない場合があります。たとえば、市コードと市町村名のマッピングを含むドメイン値マップでは、市町村名は同じでも国が異なるため、複数のマッピングが定義されている場合があります。たとえば、ケンジントンという都市は、カナダにも U.S. にもあります。

この DVM の定義の例は、次のとおりです。

EXT (識別子 1)	EXT (識別子 2)	共通	EXT	PSFT
PSFT (識別子 1)	PSFT (識別子 2)			
国	都道府県/州	共通	市コード	市区町村名
USA	ミネソタ	BELG_MN	BELG	ベルグレード
USA	ノースカロライナ	BELG_NC	BELG	ベルグレード
USA	カンサス	KN_KS	KN	ケンジントン
カナダ	プリンスエドワードアイランド	KN_PEI	KN	ケンジントン

このシナリオでは、ビジネス ユニット/チャートフィールドを含む外部システムへのアウトバウンド メッセージをPeopleSoft アプリケーションで作成する際、市区町村名、国、および州を共通値にマッピングする変換プログラムがルーティングに含まれます。

マップ オプションの定義

このセクションでは、値マップのオプションの定義方法について説明します。

値マップのオプションの定義に使用するページ

ページ名	定義名	用途
<u>値マップ オプション ページ</u>	EOTF_MAP_OPTIONS	このページは、値マップの定義で使用可能なオプションのデフォルト値の選択に使用します。

値マップ オプション ページ

値マップ オプション ページ (EOTF_MAP_OPTIONS) を使用して、値マップの定義で使用可能なオプションのデフォルト値を選択します。

ナビゲーション

[エンタープライズ コンポーネント], [インテグレーション定義], [エンタープライズ コンポーネント], [マップ オプションの定義]

画像: 値マップ オプション ページ

次の例では、値マップ オプション ページのフィールドおよびコントロールを説明します。このページのフィールドおよびコントロールの定義は後で説明します。

キャッシュ (分)

マップ定義および DVM データのキャッシュでは、この値を使用して無効の日時を決定します。この時刻をすぎると、キャッシュは使用する前にリフレッシュされます。

連結区切り

データのインポートまたはエクスポート時の、連結の区切りを指定します。V1 (Fusion Middleware FMW) スキーマでは、エクスポート中に複数エレメントのドメインの値が 1 つの列に連結される場合や、データのインポート中に連結解除される場合に、この区切り値が使用されます。

インポート オプション

以下のオプションは、インポート コンポーネントおよび作成されるマップにデフォルトで設定されます。インポート プロセスでは、マップ固有のオプションが使用されます。

インポート モード

定義をインポートし、値マップの定義コンポーネントを使用してその定義を変更して、その後、データのみをインポートできるようにします。これにより、データをインポートする前に、エレメントの名前を変更したり、場合によっては 1 ドメインに複数のエレメントを指定したりできます。

有効値は以下のとおりです。

- 定義のみ

データなしで定義を環境間で移動できます。

- 定義とデータ

定義と関連データの両方をインポートします。

- 値のみ

ファイル内の定義ではなく、データベースにあるマップ定義を使用して、データをインポートできます。

マップが存在する場合のエラー

ユーザーが既存の名前を使用してマップをインポートしようとした場合に、マップが誤って消去されないようにする場合は、このチェック ボックスをオンにします。デフォルトではオンになっています。

インポート前にマップ/データを削除

インポート前に、既存の定義およびデータを削除する場合は、このチェック ボックスをオンにします (デフォルト)。このチェック ボックスをオフにすると、追加やマージが使用されます。追加やマージでは、既存のデータを削除することなく、既存のマップにエレメント、ドメインまたはデータを追加できます。

注: このオプションを使用する場合は、その影響を理解している必要があるため、上級ユーザー以外は使用しないでください。

エクスポート オプション

以下のオプションは、作成されるマップにデフォルトで設定されます。エクスポート プロセスでは、マップ固有のオプションが使用されます。

エクスポート モード

有効なオプションは以下のとおりです。

- 定義のみ

データなしで定義を環境間で移動できます。

これは、XREF のデフォルト値です。

- 定義とデータ

定義と関連データの両方をエクスポートします。

これは、DVM のデフォルト値です。

DVM.V1 に UniqueGUID をエクスポート

マップを Fusion Middleware (FMW) に移動するときに、DVM から PeopleSoft に必要な UniqueGUID ドメインを削除する場合は、オンにします。

値マップの定義

このセクションでは、値マップの定義方法について説明します。

値マップの定義に使用するページ

ページ名	定義名	用途
値マップの定義 - エlement ページ	EOTF_DEFINE_MAPS	マップにElementを割り当てます。
値マップのオプション定義ページ	EOTF_MAP_OPT_SEC	マップレベル オプションを設定します。新しいマップの場合、このオプションの値は定義されているシステムレベル オプションの値にデフォルトで設定されます。
値マップの定義 - ドメイン ページ	EOTF_DEFINE_MAPS2	ドメインにElementを割り当てます。

値マップの定義 検索ページ

値マップの定義 検索ページを使用して、新しい値を追加します。

ナビゲーション

[エンタープライズ コンポーネント], [インテグレーション定義], [変換フレームワーク], [値マップの定義]

画像: 値マップの定義 検索ページ: [値の追加] タブ

次の例では、値マップの定義 検索ページの [値の追加] タブのフィールドおよびコントロールを説明します。このページのフィールドおよびコントロールの定義は後で説明します。

値マップを追加するには、次の手順に従います。

1. [値の追加] タブをクリックします。
2. [マップ名] フィールドにマップの名前を入力します。
3. [マッピング タイプ] フィールドでマップ タイプを選択します。
4. [追加] ボタンをクリックします。

エレメント ページが表示され、そこでマップのエレメントを定義できます。

注: マップを保存したら、マップ タイプは変更できません。

マップ タイプ

マップは静的または動的になります。

[ドメイン値マップ (静的)]

ドメイン値のロード コンポーネントによって値が提供される静的なマップです。

クロスリファレンス (動的)

キー情報を基に値が提供される動的なマップです。

値マップの定義 - エレメント ページ

値マップの定義 - エレメント ページ (EOTF_DEFINE_MAPS) を使用して、マップにエレメントを割り当てます。

ナビゲーション

[エンタープライズ コンポーネント], [インテグレーション定義], [変換フレームワーク], [値マップの定義], [エレメント]

画像: 値マップの定義 - エレメント ページ

次の例では、"値マップの定義 - エレメント" ページのフィールドおよびコントロールを説明します。このページのフィールドおよびコントロールの定義は後で説明します。

The screenshot shows the 'Elements' tab of a 'Value Map Definition' page. The 'Map Name' is 'StateCodeDVM' and the 'Type' is 'Static'. The 'Description' is 'State & Country Table DVM'. Below the description is a 'Comments' text area. A table titled 'Assign Elements to Maps' contains the following data:

Order	Element Name	Data Type	Length	Required		
1	UniqueGUID	String	36	<input checked="" type="checkbox"/>	+	-
2	COUNTRY	String	3	<input checked="" type="checkbox"/>	+	-
3	STATE	String	6	<input checked="" type="checkbox"/>	+	-
4	COMMON	String	32	<input checked="" type="checkbox"/>	+	-

Buttons for 'Export' and 'Delete' are located at the bottom right of the table area.

説明

値マップの説明を入力します。

オプション

このリンクを使用して、このマップのオプションを変更します。このリンクをクリックすると、値マップのオプション定義ページが開き、オプションを設定できる他、マップの既存のキャッシュを削除できます。[OK] をクリックすると、オプションがマップに適用されます。

新しいマップの場合、このオプションの値は定義されているシステムレベル オプションの値にデフォルトで設定されます。

「[値マップ オプション ページ](#)」を参照してください。

コメント

値マップのコメントを入力します。

順序

共通エレメントには、常に順序 1 が割り当てられます。その他のエレメントには、2 以上の順序が割り当てられます。

エレメント名

エレメント名を入力します。共通エレメントとして 1 エレメントのみを選択してください。共通エレメントには、常に、動的マップと静的マップの両方に入力される各データ行に一意の値を含める必要があります。

動的マップの場合、共通エレメントは一意の GUID である必要があります。

	静的マップの場合、共通エレメントは必要に応じて割り当てることができます。
データタイプ	値は以下のとおりです。 <ul style="list-style-type: none">• 文字列• 数値 データタイプは、データ値を入力するときの確認に使用されます。
長さ	エレメントの長さを入力します。 この長さは、データ値を入力するときの確認に使用されます。
必須	必須エレメントに指定する場合に選択します。
エクスポート	ユーザーはこのボタンを使用して、値マップに定義されているオプションを使用することで、マップ定義およびデータを XML 形式でパブリッシュできます。 「 値マップのエクスポート 」を参照してください。
削除	このボタンを使用して、値マップを削除します。

値マップのオプション定義ページ

値マップのオプション定義ページ (EOTF_MAP_OPT_SEC) を使用して、マップレベル オプションを設定します。新しいマップの場合、このオプションの値は定義されているシステムレベル オプションの値にデフォルトで設定されます。

ナビゲーション

"値マップの定義 - エlement" ページの [オプション] リンクをクリックします。

画像: 特定のマップの値マップのオプション定義ページ

次の例では、特定のマップの値マップのオプション定義ページのフィールドおよびコントロールを説明します。このページのフィールドおよびコントロールの定義は後で説明します。

Define options for a value map

Map Name: StateCodeDVM

Short Name:

Cache Minutes:

Concatenation Separator:

Import Options

*Import Mode:

Error When Map Exists

Delete Map/Data Before Import

Export Options

*Export Mode:

Export UniqueGUID to DVM.V1

OK Cancel Delete Cache

このページは、デフォルト オプションの定義に使用する値マップのオプション定義ページに似ていますが、次の違いがあります。

- オプションは、選択されている特定のマップに適用されます。
- マップの既存のキャッシュを削除できる [キャッシュの削除] ボタンがあります。

[OK] ボタンをクリックすると、オプションがマップに適用され、値マップの定義ページに戻ります。

「[値マップ オプション ページ](#)」を参照してください。

値マップの定義 - ドメイン ページ

値マップの定義 - ドメイン ページ (EOTF_DEFINE_MAPS2) を使用して、ドメインにエレメントを割り当てます。

ナビゲーション

[エンタープライズ コンポーネント], [インテグレーション定義], [変換フレームワーク], [値マップの定義], [ドメイン]

画像: 値マップの定義 - ドメイン ページ

次の例では、"値マップの定義 - ドメイン" ページのフィールドおよびコントロールを説明します。このページのフィールドおよびコントロールの定義は後で説明します。

The screenshot shows the 'Domains' tab in a software interface. The main configuration is for 'Map Name: StateCodeDVM' and 'Type: Domain Value Map (static)'. Below this, there are three sections for assigning domains to value maps.

Domain 1: Domain Name: AIA, Is Unique: . Assigned Elements: 1 COMMON.

Domain 2: Domain Name: PSFT, Is Unique: . Assigned Elements: 1 COUNTRY, 2 STATE.

Domain 3: Domain Name: UniqueGUID, Is Unique: . Assigned Elements: 1 UniqueGUID.

値マップに含まれる各エレメントは、少なくとも 1 つのドメインに割り当てられている必要があります。

ドメイン ネーム

関係するシステムのドメイン ネームを入力します。

固有

ドメインに含まれるエレメントが一意のインスタンスになることを指定する場合に選択します。

エレメント名

割り当てるエレメントを選択します。

注: マップに含まれるデータ行ごとに一意の値を管理する必要があるため、"UniqueGUID" というドメインネームが自動的に生成されます。エレメントのデータ値がドメインではなくマップに対して一意であることをユーザーが認識している限り、複数のドメイン間でエレメントを共有できます。一般に、DVM ドメインには通常 1 エレメントのみが含まれ、PeopleSoft XREF ドメインには通常複数のエレメントが含まれますが、これは決まりではありません。

ドメイン値マップのロード

このセクションでは、ドメイン値マップのロード方法について説明します。

ドメイン値マップのロードに使用するページ

ページ名	定義名	用途
ドメイン値マップ ページ	EOTF_POPULATE_DVMS	ドメイン値マップをロードします。

ドメイン値マップ ページ

ドメイン値マップ ページ (EOTF_POPULATE_DVMS) を使用して、ドメイン値マップをロードします。

ナビゲーション

[エンタープライズ コンポーネント], [インテグレーション定義], [変換フレームワーク], [ドメイン値マップのロード]

画像: ドメイン値マップ ページ

次の例では、ドメイン値マップ ページのフィールドおよびコントロールを説明します。このページのフィールドおよびコントロールの定義は後で説明します。

Domain Value Map			
Map Name: StateCodeDVM		Description: State & Country Table DVM	
Assign Data to Value Maps			
COUNTRY	STATE	COMMON	
1 USA	CA	California	+ -
2 USA	NY	New York	+ -
3 USA	MD	Maryland	+ -
4 USA	MI	Michigan	+ -
5 CAN	NF	Newfoundland	+ -
6 USA	AK	Alaska	+ -

ドメイン値マップを使用して、静的な値マップにデータを入力して管理します。値マップに定義するエレメントが、ページに表示される列になります。エレメントは、マップ定義に指定する際に、順序が付けられ、確認されます。

どの DVM でも、各データ行に一意の値が入力される必要があります。ページを保存すると、コンポーネントによってキャッシュ済行セットが検索され、破棄されます。キャッシュは、次回呼び出されるときに再ロードされます。

値マップのインポート

このセクションでは、値マップで使用するインポート ファイル タイプの概要および値マップのインポート方法について説明します。

値マップのインポートに使用するページ

ページ名	定義名	用途
値マップのインポート ページ	EOTF_MAP_IMPORT	指定されたファイルから DVM または XREF をインポートします。

値マップで使用するインポート ファイル タイプについて

変換フレームワークでは、2 種類のインポート ファイルがサポートされています。

- XML ファイル
- CSV ファイル

XML ファイルのインポート

変換フレームワークでは、2 種類のスキーマがサポートされています。

- V1 Fusion Middleware (FMW)

これは、11g フォーマットを使用する FMW に使用されるスキーマです。関連付けられているサービス オペレーションは EOTF_DVM_IMPORT.v1 です。これは、サービス オペレーションのデフォルトバージョンではないため、この形式で値マップがインポートされる場合は、バージョン変換プログラムが実行されます。

「[EOTF_DVM_IMPORT.v1](#)」、「[EOTF_XREF_IMPORT.v1](#)」を参照してください。

- PeopleSoft フォーマット

これは、PeopleSoft に使用されるスキーマです。関連付けられているサービス オペレーションは EOTF_DVM_IMPORT.v2 です。これは、デフォルトのサービス オペレーションです。

「[EOTF_DVM_IMPORT.v2](#)」、「[EOTF_XREF_IMPORT.v2](#)」を参照してください。

指定されたファイルに XML が含まれる場合、XML は インポート Web サービスがサポートするスキーマの 1 つに準拠する必要があります。指定されたファイルに XML が含まれる場合、ユーザーはマップ定義、マップ データ、またはその両方をインポートできます。FMW (V1) スキーマを使用するインポートでは、インポート プロセス中に複合値を元素値に分解する必要がある場合があります。そのためには、まず定義をインポートします。次に、値マップの定義コンポーネントを使用して、その定義を変更し、1 つ以上のドメインに対して複数の元素を指定します。そのうえで前に戻り、データ値のみをインポートします。このシ

ナリオでは、指定された区切りを使用して、値の連結が解除されます。必須エレメントに値が入力されていない場合、エラーがスローされます。

CSV ファイルのインポート

ファイルに CSV データが含まれる場合、次のルールが適用されます。

- ファイルの 1 行目では、1 列目に値マップのタイプ (*DVM* または *XREF*) と、2 列目にインポートする値マップの名前が指定されている必要があります。
- ファイルの 2 行目の各列には、インポート先のドメイン名またはエレメント名 (の必ずどちらか一方)、または参照データが指定されている必要があります。既存のマップにインポートする場合、ドメイン名とエレメント名は、そのマップの定義を基に検証されます。ファイルにドメイン名やエレメント名が指定されていないか、ドメインの必須エレメントが全て含まれていない場合は、エラーがスローされます。インポートプロセスによって無視される参照データ列を定義するには、列名にアスタリスクをプレフィックスとして付加するか、名前を空欄にします。
- ファイル内の後続の行では、インポート対象のデータ値を識別し、2 行目と同じ数の列が含まれる必要があります。
- 列に含まれるデータ値が、ドメインの各エレメントを示す複合 (連結) 値で、ファイルの 2 行目がドメイン名を示している場合、インポートプロセスは、複数のエレメントが定義されている各ドメインの区切りを使用して、値を連結解除します。必須エレメントに値が入力されていない場合、エラーがスローされます。
- マップに既に他のドメインのデータが含まれていて、当該ドメインの値を既存のデータ行に追加する場合は、1 列目のデータ列で参照されているドメインを使用して、既存のデータ行が特定されます。データ行が見つからない場合は、新しいデータ行が作成されます。複数の行が見つかり、ファイルに追加するドメインが重複を許可していない場合は、エラーが発生します。重複が許可されている場合は、既存のデータ行が更新され、ファイル内の他のドメインのデータ値が追加されます。
- UniqueGUID が空欄か、ファイルに指定されていない場合は、新しいデータ行ごとに値が作成されます。
- 定義をインポートするとき、各列は、1 つのエレメントが定義されている同名のドメインとして処理されます。まず定義のみをインポートして、値マップの定義コンポーネントを使用して定義を変更し、その後、データのみをインポートすることで、複合 (連結) 値や複数のエレメントが定義されているドメインをインポートできます。

Microsoft Excel に表示されたクロスリファレンス値マップの csv ファイルの例を示します。

画像: クロスリファレンス値マップのインポートに使用するサンプルの csv ファイル

次の例では、クロスリファレンス値マップのインポートに使用するサンプルの csv ファイルを説明します。このページのフィールドおよびコントロールの定義は後で説明します。

	A	B	C	D
1	XREF	XrefTest		
2	UniqueGUID	PSFT	*FusionValue	FusionGUID
3	asdf1	US001::VEND000001	V0101	1
4	asdf2	US002::VEND000002	V0202	2
5	asdf3	US003::VEND000003	V0303	3
6	asdf4	US004::VEND000004	V0404	4
7	asdf5	US005::VEND000005	V0405	5
8	asdf6	US006::VEND000006	V0406	6
9	asdf7	US007::VEND000007	V0407	7
10				

Microsoft Excel に表示された DVM 値マップの csv ファイルの例を示します。

画像: DVM 値マップのインポートに使用するサンプルの csv ファイル

次の例では、DVM 値マップのインポートに使用するサンプルの csv ファイルを説明します。このページのフィールドおよびコントロールの定義は後で説明します。

	A	B	C
1	DVM	StateCodes	
2	PeopleSoft	*Description	ExternalSystem
3	MA	Massachusetts	Mass
4	CA	California	Calif
5	NV	Nevada	Nev
6	OR	Oregon	Oreg
7			

値マップのインポート ページ

値マップのインポート ページ (EOTF_MAP_IMPORT) を使用して、指定されたファイルから DVM または XREF をインポートします。

ナビゲーション

[エンタープライズ コンポーネント], [インテグレーション定義], [変換フレームワーク], [値マップのインポート]

画像: 値マップのインポート ページ

次の例では、値マップのインポート ページのフィールドおよびコントロールを説明します。このページのフィールドおよびコントロールの定義は後で説明します。

値マップのインポートコンポーネントは、指定されたファイルから DVM または XREF をインポートする機能を提供します。インポートするファイルは、XML またはカンマ区切り値 (CSV) フォーマットである必要があります。インポート時にデータ損失が発生した場合、確認のダイアログが表示されます。オプションのデフォルト値は、値マップ オプション コンポーネントを使用してユーザーが設定したシステム デフォルトから取得されます。

ファイル名

パスも含めたファイル名。ファイルは XML または CSV フォーマットで、マップ名が指定されている必要があります。

インポート モード

定義をインポートし、値マップの定義コンポーネントを使用してその定義を変更して、その後、データのみをインポートできるようにします。これにより、データをインポートする前に、定義をインポートしたうえで、エレメントの名前を変更したり、場合によっては 1 ドメインに複数のエレメントを指定したりできます。有効値は以下のとおりです。

- 定義のみ
- 定義とデータ
- 値のみ

インポート前にマップ/データを削除

インポート前に、既存の定義およびデータを削除する場合は、オンにします (デフォルト)。このチェック ボックスをオフにすると、追加やマージが使用されます。追加やマージでは、既存のデー

データを削除することなく、既存のマップにエレメント、ドメインまたはデータを追加できます。

注: このオプションを使用する場合は、その影響を理解している必要があるため、上級ユーザー以外は使用しないでください。

列区切り

CSV ファイルでは、カンマ以外の文字を使用して、列値を区切ることができます。ここに入力された値は、指定されたファイルの読取り時に使用される区切り文字の値になります。タブ文字を指定するには、区切り文字の値として `[t]` を入力します。

連結区切り

V1 (FMW) スキーマでは、エクスポート中に複数エレメントのドメインからの値が1つの列に連結される場合や、データのみインポート中に連結解除される場合に、この区切り値が使用されます。

インポート

オプションを検証し、インポートプロセスを開始します。

値マップのエクスポート

このセクションでは、エクスポート スキーマの概要および値マップのエクスポート方法について説明します。

エクスポート スキーマについて

値マップは、値マップの定義ページから [エクスポート] ボタンを使用して、エクスポートできます。マップ定義およびデータは、値マップに定義されているオプションを使用して、XML 形式でパブリッシュされます。XML は、EOTF_DVM_EXPORT サービス オペレーションのアクティブなバージョンで指定されているスキーマに準拠します。

- FMW フォーマット (V1)

FMW スキーマを使用してエクスポートするには、EOTF_DVM_EXPORT.v1 サービス オペレーションに対して有効なルーティングをアクティブ化し、提供する必要があります。

V1 スキーマへのエクスポートでは、ドメイン名が列名として使用されます。複数エレメントのドメインが V1 スキーマにエクスポートされる場合は、エレメント値は連結されて、ドメインごとに複合値が作成されます。

- PeopleSoft

PeopleSoft スキーマを使用してエクスポートするには、EOTF_DVM_EXPORT.v2 サービス オペレーションに対して有効なルーティングをアクティブ化し、提供する必要があります。

- 両方

EOTF_DVM_EXPORT.v1 および EOTF_DVM_EXPORT.v2 の両方に対するルーティングをアクティブ化して提供します。

『*PeopleTools: PeopleSoft Integration Broker*』の「Managing Service Operations」、「Configuring Service Operation Definitions」を参照してください。

値マップのエクスポート

値マップをエクスポートするには、次の手順に従います。

1. [エンタープライズ コンポーネント], [インテグレーション定義], [変換フレームワーク], [値マップの定義] を選択します。
2. エクスポートするマップを選択します。
3. [エクスポート] ボタンをクリックします。

変換プログラムの作成とサービス オペレーションの更新

変換プログラムについて

変換プログラムは、PeopleSoft アプリケーション エンジン プログラムのタイプの 1 つです。新しい変換アプリケーション エンジン プログラムを作成したら、プログラムにステップとアクションを追加し、さらにステップとアクションにデータ変換と翻訳を実行するコードを追加します。

変換プログラムを開発するには、変換するメッセージの初期の構造および場合によってはその内容と、変換後に得たい構造 (とコンテンツ) を把握しておく必要があります。

適用する変換プログラムは、サービス オペレーションのルーティング定義内で指定します。

変換のプログラミング言語

変換ロジックを作成するプログラミング言語には、PeopleCode または Extensible Stylesheet Language Transformation (XSLT) を使用できます。XSLT は XML 構造の操作に適した標準言語として広く認められているため、変換には XSLT の使用が強く推奨されます。

PeopleSoft アプリケーションは、値マップの検索、削除、およびロードを実行する XSLT 拡張関数および PeopleCode API を提供します。

注: XSLT を使用してプログラムを作成する場合、手動で XSLT をコーディングするか、Oracle XSL マッパーを使用して、グラフィックを用いてレコードおよびフィールドを関連付けることができます。Oracle XSL マッパーの場合、関連付けを行うと、自動的に XSLT コードが作成されます。

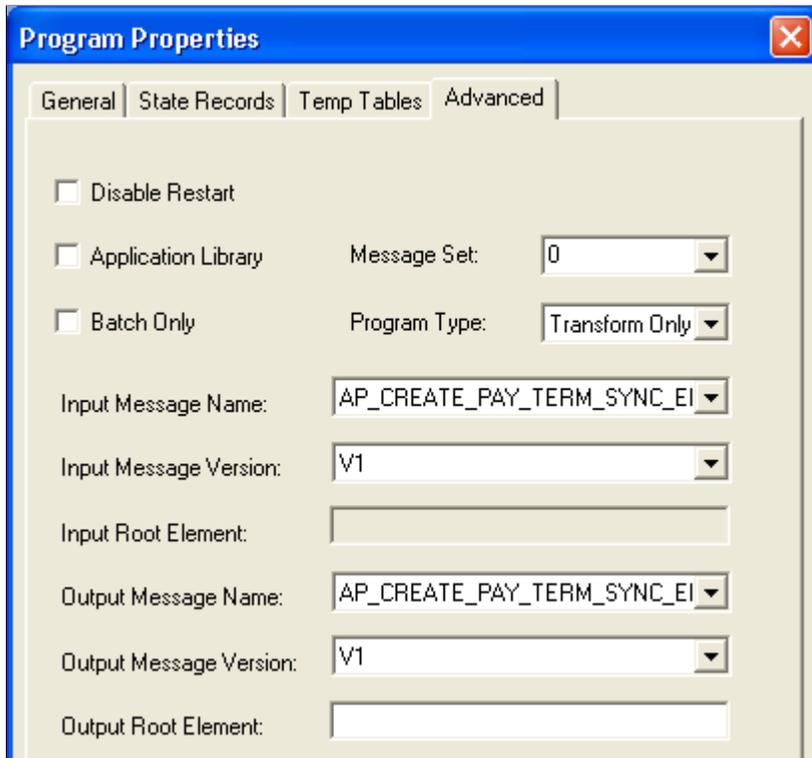
『*PeopleTools: PeopleSoft Integration Broker*』の「Applying Filtering, Transformation and Translation」、
「Developing Transform Programs」を参照してください。

変換プログラムの作成

アプリケーション エンジン プログラムは、アプリケーション デザイナを使用して作成します。ツールバーの [プロパティ] ボタンをクリックし、[詳細] タブを選択します。

画像: 変換プログラムのアプリケーション エンジン プログラム プロパティ

次の例では、変換プログラムのアプリケーション エンジン プログラム プロパティのフィールドおよびコントロールを説明します。このページのフィールドおよびコントロールの定義は後で説明します。



アプリケーション エンジン プログラムは、プログラム タイプを変換のみに定義する必要があります。オプションで、入出力メッセージの名前とバージョンを指定することもできます。入出力メッセージの名前とバージョンは、Oracle グラフィカル マッパーを使用して変換プログラムの XSLT を作成する場合は、指定する必要があります。

注: Oracle グラフィカル マッパーを使用した変換プログラムの作成の詳細は、『*PeopleTools: PeopleSoft Integration Broker*』の「Applying Filtering, Transformation and Translation」、「Developing Transform Programs」を参照してください。

クロスリファレンス値マップおよびドメイン値マップに使用する構文については、トピック「[XSLT 拡張関数について](#)」を参照してください。

注: 付録「アプリケーション インテグレーション フレームワークの例」には、コード スニペットが記載されています。

サービス オペレーションのルーティングの更新

ルーティング定義には、サービス オペレーションに関連付けられている各インバウンドリクエスト、インバウンドレスポンス、アウトバウンドリクエスト、およびアウトバウンドレスポンスのルーティング パラメータが指定されています。リクエストまたはレスポンスごとに、ルーティング別名、変換前と後のメッセージ名、および変換プログラム名を定義します。

サービス オペレーションのルーティング定義で変換プログラム名を指定すると、その変換プログラムは PeopleSoft インテグレーション ブローカーによって呼び出されます。

ルーティング パラメータの追加

ルーティング定義のページにアクセスして、ルーティング パラメータを追加するには、次の手順に従います。

1. [PeopleTools], [インテグレーション ブローカー], [インテグレーション設定], [サービス オペレーション]を選択します。
2. ルーティング ページを選択します。
3. 既存のルーティングのリンクをクリックするか、新しいルーティングを追加します。
4. 新しいルーティングの場合は、ルーティング情報を追加します。

5. パラメータ ページを選択します。

画像: 非同期の一方向のサービス オペレーションの ルーティング パラメータ ページ

次の例では、非同期の一方向のサービス オペレーションの ルーティング パラメータ ページのフィールドおよびコントロールを説明します。このページのフィールドおよびコントロールの定義は後で説明します。

Routing Definitions		Parameters	Routing Properties
Routing Name:	SCM_CONTACT1_TO_CONTACT_RB1		
Service Operation:	CONTACT_SYNC_EFF		
Service Operation Version:	VERSION_RB_1		
Sender Node:	PSFT_EP		
Receiver Node:	CR910TS1		
Parameters			
Type:	Inbound Request		
External Alias:	<input type="text" value="CONTACT_SYNC_EFF.VERSION_1"/>		
Alias References			
Message.Ver into Transform 1:	<input type="text" value="CONTACT_SYNC_EFF.VERSION_1"/>		
Transform Program 1:	<input type="text" value="RB_CONT_IN"/>		
Transform Program 2:	<input type="text"/>		
Message.Ver out of Transforms:	<input type="text" value="CONTACT_SYNC_EFF.VERSION_RB_1"/>		
<input type="button" value="Save"/>		<input type="button" value="Return"/>	

アプリケーション エンジン変換プログラムに変換の入力および出力のメッセージが含まれる場合、それらのメッセージは自動的にこれらの値と合わせてロードされます。アプリケーション エンジン プログラムにメッセージ名が指定されていない場合、適切なメッセージ名を入力して、ページを保存します。

注: 同期サービス オペレーションの場合は、アウトバウンド メッセージにもインバウンド メッセージにも変換を定義できます。

『PeopleTools: PeopleSoft Integration Broker』の「Managing Routing Definitions」、 「Creating Routing Definitions」を参照してください。

XSLT 拡張を使用したマップへのアクセス

XSLT 拡張関数について

値マップを作成している場合は、メッセージ内のエレメントをマッピングし、値マップを使用してデータ変換を実行する変換プログラムを作成する必要があります。

PeopleSoft アプリケーションは、値マップの検索、削除、およびロードを実行する XSLT 拡張関数を提供します。クロスリファレンス用と DVM 用の 2 種類の、ほぼ同じ内容の関数セットが用意されています。

クロスリファレンス関数

このセクションでは、アルファベット順で、クロスリファレンスの XSLT 拡張関数について説明します。

xref:BulkPopulateDomainData

構文

```
xref:BulkPopulateDomainData(mapName, domainList, dataRowsPath, dataValuesPath, mode)
```

説明

この XREF ドメイン データのバルク ロード関数は、XML からクロスリファレンス データをロードします。XML ノードセットに設定されているデータ行を順番に処理し、指定されているドメイン リストとデータ値のパスを使用して、各行に設定されている各ドメインの値にアクセスします。その後、各データ行は、できるだけ効率よく処理するため、一括挿入を使用して、指定されたマップの XREF フレームワークに挿入されます。この関数は、大量のデータの処理に使用します。これにより、XREF フレームワークの物理的なデータ永続性層の抽象化層が提供されます。挿入はバルク モードで実行されるため、挿入シーケンスが完了するまで重複は検索されません。重複が見つかった場合、データ行はまったく挿入されません。ドメインに複数のエレメントが含まれる場合、ドメインの各値には、ドメインに含まれる各エレメントの値を、値マップ オプションで指定されている区切り文字で区切った複合値を指定する必要があります。このシナリオでは、変換フレームワークにデータを挿入する前に、この関数によって、ドメイン エレメント値の連結が解除されません。

パラメータ

パラメータ	説明
mapName	動的 (クロスリファレンス) 値マップの名前を文字列として指定します。
domainList	ドメイン名をカンマで区切ったリスト。インポートするデータ値の順序を指定します。
dataRowsPath	一連の XML ノードを生成する XPath 式。各ノードは 1 つのデータ行を表します。
dataValuesPath	データ行内のデータ値を含む各ノードを指定するために使用する (各データ行に関連する) XPath 式。このパスが評価されるときに返されるノードセットには、指定された domainList 内のドメインと同じ数のノードが含まれている必要があります。
mode	ADD モードのみがサポートされます。

戻り値

プロセスが成功したか失敗したかを示すブール値。

xref:BulkPopulateElementData

構文

xref:BulkPopulateElementData (*mapName*, *elementList*, *dataRowsPath*, *dataValuesPath*, *mode*)

説明

この XREF エlement データのバルクロード関数は、XML からクロスリファレンス データをロードします。XML ノードセットに設定されているデータ行を順番に処理し、指定されている Element と XPath 式のペアを使用して、各行に設定されている Element の値にアクセスします。その後、各データ行は、できるだけ効率よく処理するため、一括挿入を使用して、指定されたマップの XREF フレームワークに挿入されます。この関数は、大量のデータの処理に使用します。これにより、XREF フレームワークの物理的なデータ永続性層の抽象化層が提供されます。挿入はバルク モードで実行されるため、挿入シーケンスが完了するまで重複は検索されません。重複が見つかった場合、データ行はまったく挿入されません。

パラメータ

パラメータ	説明
mapName	動的 (クロスリファレンス) 値マップの名前を文字列として指定します。

パラメータ	説明
elementList	エレメント名をカンマで区切ったリスト。インポートするデータ値の順序を指定します。
dataRowsPath	一連の XML ノードを生成する XPath 式。各ノードは 1 つのデータ行を表します。
dataValuesPath	データ行内のデータ値を含む各ノードを指定するために使用する (各データ行に関連する) XPath 式。このパスが評価されるときに返されるノードセットには、指定された domainList 内のドメインと同じ数のノードが含まれている必要があります。
mode	ADD モードのみがサポートされます。

戻り値

プロセスが成功したか失敗したかを示すブール値。

xref:populateXRefRow

構文

```
xref:populateXRefRow(mapName, referenceElementName, referenceValue, elementName, elementValue, mode)
```

説明

populateXRefRow 関数を使用して、クロスリファレンス エレメントに値をロードします。

パラメータ

パラメータ	説明
mapName	クロスリファレンス マップの名前を文字列として指定します。
referenceElementName	参照エレメントの名前を文字列として指定します。
referenceValue	参照エレメント名に対応する値を文字列として指定します。
elementName	ロードするエレメントの名前を文字列として指定します。
elementValue	エレメントのロードに使用する値を文字列として指定します。

パラメータ	説明
mode	この関数がエレメントをロードするモードを示します。ADD、LINK、または UPDATE のいずれかの値を指定できます。モードパラメータの値は、大文字と小文字が区別され、大文字だけで指定される必要があります。

戻り値

この関数は、ロードされるクロスリファレンス値を文字列として返します。

次の表に、Xref.populateXRefRow 関数の結果を示します。

モード	参照値	追加される値	結果
ADD	なし	なし	成功
	あり	なし	例外
	あり	あり	例外
LINK	なし	なし	例外
	あり	なし	成功
	あり	あり	例外
UPDATE	なし	なし	例外
	あり	なし	例外
	あり	あり	成功

例

次の表に、モードとその説明および例外の理由の例を示します。

モード	説明	例外の理由
ADD	<p>参照値と値を追加します。以下に例を示します。</p> <pre>xref:populateXRefRow("customers", "PS", "PS101", "Common", "CM001", "ADD")</pre> <p>参照値 <i>PS101</i> を <i>PS</i> エlementに追加し、値 <i>CM001</i> を <i>customers</i> クロスリファレンス マップの <i>Common</i> Elementに追加します。</p>	<p>次の理由で、例外が発生する可能性があります。</p> <ul style="list-style-type: none"> • 指定されたクロスリファレンス マップが見つかりません。 • 指定されたElementが見つかりません。 • 提供された値が空です。 • 追加する値は、対象のマップの対象のElementにおいて一意ではありません。 • 対象の行のElementには既に値が存在します。 • 参照値は存在しています。
LINK	<p>既存の参照値に対応するクロスリファレンス値を追加します。以下に例を示します。</p> <pre>xref:populateXRefRow("customers", "Common", "CM001", "SBL", "SB-101", "LINK")</pre> <p>値 [<i>SB-101</i>] を <i>customers</i> クロスリファレンス マップの <i>SBL</i> Elementに追加し、それを <i>Common</i> Elementの値 [<i>CM001</i>] にリンクします。</p>	<p>次の理由で、例外が発生する可能性があります。</p> <ul style="list-style-type: none"> • 指定されたクロスリファレンス マップが見つかりません。 • 指定されたElementが見つかりません。 • 提供された値が空です。 • 参照値が見つかりません。 • リンクする値が、対象のマップの対象のElementに存在しています。
UPDATE	<p>既存の参照Elementと値のペアに対応するクロスリファレンス値を更新します。以下に例を示します。</p> <pre>xref:populateXRefRow("customers", "PS", "PS100", "PS", "PS1001", "UPDATE")</pre> <p><i>customers</i> クロスリファレンス マップの <i>PS</i> Elementの値 [<i>PS100</i>] を値 [<i>PS1001</i>] に更新します。</p>	<p>次の理由で、例外が発生する可能性があります。</p> <ul style="list-style-type: none"> • 指定されたクロスリファレンス マップが見つかりません。 • 指定されたElementが見つかりません。 • 提供された値が空です。 • 更新する値は、対象のマップの対象のElementにおいて一意ではありません。 • 更新するElementに対して複数の値が見つかりました。 • 参照値が見つかりません。 • 対象の行のElementに値がありません。

xref:populateXrefRowNVP

構文

```
xref:populateXrefRowNVP(mapName, referenceDomain, referenceNVP, targetDomain, targetNVP, mode)
```

説明

xref:populateXrefRowNVP 関数を使用して、クロスリファレンス マップ内の複数のエレメントに値をロードします。

パラメータ

パラメータ	説明
mapName	クロスリファレンス マップの名前を文字列として指定します。
referenceDomain	参照ドメインの名前を文字列として指定します。
rreferencesNVP	参照エレメントおよび値の NVP リストを文字列として指定します。
targetDomain	ロードするドメインの名前を文字列として指定します。
targetNVP	エレメントおよび当該エレメントにロードする値の NVP リストを文字列として指定します。
mode	この関数がエレメントをロードするモードを示します。ADD、LINK、または UPDATE のいずれかの値を指定できます。モード パラメータの値は、大文字と小文字が区別され、大文字だけで指定される必要があります。

戻り値

次の表に、populateXrefRowNVP 関数の結果を示します。

モード	参照値	追加される値	結果
ADD	なし	なし	成功
	あり	なし	例外
	あり	あり	例外

モード	参照値	追加される値	結果
LINK	なし	なし	例外
	あり	なし	成功
	あり	あり	例外
UPDATE	なし	なし	例外
	あり	なし	例外
	あり	あり	成功

例

次の表に、モードとその説明および例外の理由を示します。

モード	説明	例外の理由
ADD	<p>参照値と追加する値を追加します。以下に例を示します。</p> <pre>xref:populateXRefRowNVP("Items", "PeopleSoft", "<Setid>SHARE</Setid> <ItemID>1005</ItemID>", "Common", "<Common>" generate-guid() "/Common>", "ADD")</pre> <p>PeopleSoftドメインの参照値 <i>SHARE/1005</i> および <i>Common</i>ドメインの値 <i><guid1></i> を追加します。</p>	<p>次の理由で、例外が発生する可能性があります。</p> <ul style="list-style-type: none"> 指定されたクロスリファレンス マップが見つかりません。 指定されたドメインが見つかりません。 指定されたエレメントが見つかりません。 提供された値が空です。 追加しようとしている値は、マップのドメイン内で一意ではありません。 対象の行のエレメントには既に値が存在します。 参照値は存在しています。
LINK	<p>既存の参照値に対応するクロスリファレンス値を追加します。以下に例を示します。</p> <pre>xref:populateXRefRowNVP("Items", "PeopleSoft", "<Setid>SHARE</Setid> <ItemID>1005</ItemID>", "Retail", "<Product>RP0005</Product>", "LINK")</pre> <p>値 <i>RP005</i> を <i>Retail</i>ドメインに追加し、それを <i>PeopleSoft</i>ドメインの参照値 <i>SHARE/1005</i> にリンクします。</p>	<p>次の理由で、例外が発生する可能性があります。</p> <ul style="list-style-type: none"> 指定されたクロスリファレンス マップが見つかりません。 指定されたドメインが見つかりません。 指定されたエレメントが見つかりません。 提供された値が空です。 参照値が見つかりません。 リンクする値が、対象のマップの対象のドメインに存在しています。

モード	説明	例外の理由
UPDATE	<p>既存の参照エレメントと値のペアに対応するクロスリファレンス値を更新します。以下に例を示します。</p> <pre>xref:populateXRefRowNVP("Items", "PeopleSoft", "<Setid>SHARE</Setid> <ItemID>1000</ItemID>", "PeopleSoft", "<Setid>SHARE</Setid> <ItemID>10000</ItemID>", "UPDATE")</pre> <p>PeopleSoft ドメインの <i>ItemID</i> エレメントの値 1000 を値 10000 に更新します。</p>	<p>次の理由で、例外が発生する可能性があります。</p> <ul style="list-style-type: none"> 指定されたクロスリファレンス マップが見つかりません。 指定されたドメインが見つかりません。 指定されたエレメントが見つかりません。 提供された値が空です。 更新する値は、対象のマップの対象のドメインにおいて一意ではありません。 更新するドメインに対して複数の値が見つかりました。 参照値が見つかりません。 対象の行のエレメントに値がありません。

xref:markForDelete

構文

```
xref:markForDelete(mapName, elementName, elementValue)
```

説明

指定されているエレメントがシングルドメインの唯一のエレメントである場合、xref:markForDelete 関数を使用して、クロスリファレンス マップ内の値を削除します。エレメントの値は削除済としてマークされます。複数のドメインがエレメントを参照しているか、エレメントを参照しているドメインに複数のプライマリ エレメントがある場合は、xref:markForDeleteNVP 関数を使用します。

クロスリファレンス マップの各行には、少なくとも 2 つのマッピングが設定されている必要があります。従って、行に 2 つのマッピングしかなく、一方の値を削除指定すると、別のエレメントの値も削除されます。

削除指定されたエレメント値は、存在していないものとして扱われます。従って、xref:populateXRefRow 関数を ADD モードで使用して、同じエレメントをロードできます。ただし、削除指定されているエレメント値は、xref:populateXRefRow 関数の LINK モードで、参照値として使用できません。

パラメータ

パラメータ	説明
mapName	クロスリファレンス マップ名を文字列として指定します。

パラメータ	説明
elementName	値を削除するエレメントの名前を文字列として指定します。
elementValue	削除する値を文字列として指定します。

戻り値

この関数は、削除が成功した場合 "真" を返し、失敗した場合は "偽" を返します。

次の理由で、例外が発生する可能性があります。

- 指定された名前のクロスリファレンス マップが見つかりません。
- 指定されたエレメント名が見つかりません。
- 指定されたエレメント名は、ドメインに対して一意ではありません。
- 指定された値が空です。
- 指定された値がエレメント内にありません。
- 複数の値が見つかりました。

例

次のコードでは、*customers* クロスリファレンス マップの PS エレメントの値 *PS001* が削除されます。

```
xref:markForDelete("customers","PS","PS001")
```

xref:markForDeleteNVP

構文

```
xref:markForDeleteNVP(mapName, referenceDomain, referenceNVP)
```

説明

xref:markForDeleteNVP 関数を使用して、指定されたドメインのクロスリファレンス マップ内の値セットを削除します。エレメントの値は削除済としてマークされます。

クロスリファレンス マップの各行には、少なくとも 2 つのマッピングが設定されている必要があります。従って、行に 2 つのマッピングしかなく、一方の値を削除指定すると、もう一方のドメインの値も削除されます。

削除指定された値は、存在していないものとして扱われます。従って、xref:populateXRefRowNVP 関数を ADD モードで使用して、同じエレメントをロードできます。ただし、削除指定されているエレメント値は、xref:populateXRefRowNVP 関数の LINK モードで、削除指定されているエレメント値は、参照値として使用できません。

パラメータ

パラメータ	説明
mapName	クロスリファレンス マップ名を文字列として指定します。
referenceDomain	参照ドメインの名前を文字列として指定します。
referenceNVP	参照エレメントおよび削除する値の NVP リストを文字列として指定します。

戻り値

この関数は、削除が成功した場合 "真" を返し、失敗した場合は "偽" を返します。

次の理由で、例外が発生する可能性があります。

- 指定された名前のクロスリファレンス マップが見つかりません。
- 指定されたエレメント名が見つかりません。
- このドメインの全てのエレメントが指定されていません。
- 指定された値が空です。
- 指定された値がエレメント内にありません。
- 複数の値が見つかりました。

例

次のコードでは、*PeopleSoft* ドメインの *Setid* エレメントと *ItemID* エレメントに指定された値が、*Items* クロスリファレンス マップから削除されます。

```
xref:markForDeleteNVP ("Items","PeopleSoft", "<Setid>SHARE</Setid><ItemID>1000</ItemID>")
```

xref:lookupXRef

構文

```
xref:lookupXRef (mapName, referenceElementName, xrefReferenceValue, elementName, needAnException)
```

説明

lookupXRef 関数を使用して、参照エレメントの特定の 1 つの値に対応する 1 つの値のクロスリファレンス エレメントを検索します。

パラメータ

パラメータ	説明
mapName	クロスリファレンス マップの名前を文字列として指定します。
referenceElementName	参照エレメントの名前を文字列として指定します。
referenceValue	参照エレメント名に対応する値を文字列として指定します。
elementName	値を検索するエレメントの名前を文字列として指定します。
needAnException	"真" または "偽" を指定します。 needAnException パラメータが "真" に設定されていると、検索対象の値がマップ内に見つからない場合、例外が発生します。needAnException パラメータが "偽" に設定されていると、検索対象の値がマップ内に見つからない場合、空の値が返されます。

戻り値

リクエストされたエレメントの値。

次の理由で、例外が発生する可能性があります。

- 指定された名前のクロスリファレンス マップが見つかりません。
- 指定されたエレメント名が見つかりません。
- 指定された参照値が空です。
- 複数のターゲット値が見つかりました。

例

次のコードでは、*customers* クロスリファレンス マップの *Common* エレメントから、*PS* エレメントの値 *PS001* に対応する値が検索されます。

```
xref:lookupXRef("customers","PS","PS001","Common",true())
```

xref:lookupXRefNVP

構文

```
xref:lookupXRefNVP (mapName, referenceDomain, referenceNVP, targetDomain,  
needAnException)
```

説明

lookupXRefNVP 関数を使用して、指定された参照ドメインの値セットに対応する、クロスリファレンス値を検索します。参照ドメインの全てのプライマリ エレメントは、NVP 参照リストに含める必要がありますが、識別子エレメントは任意です。

パラメータ

パラメータ	説明
mapName	クロスリファレンス マップの名前を文字列として指定します。
referenceDomain	参照ドメインの名前を文字列として指定します。
referenceNVP	参照エレメントおよび値の NVP リストを文字列として指定します。
targetDomain	値を検索するドメインの名前を文字列として指定します。
needAnException	"真" または "偽" を指定します。 needAnException パラメータが "真" に設定されていると、検索対象の値がマップ内に見つからない場合、例外が発生します。needAnException パラメータが "偽" に設定されていると、検索対象の値がマップ内に見つからない場合、空の値が返されます。

戻り値

戻り文字列には、ターゲットドメインの全てのプライマリ エレメントと識別子エレメントの値が NVP リスト形式で含まれます。

次の理由で、例外が発生する可能性があります。

- 指定された名前のクロスリファレンス マップが見つかりません。
- 指定されたドメイン名が見つかりません。
- 指定されたエレメント名が見つかりません。
- 指定された参照値が空です。
- 複数のターゲット値が見つかりました。

例

次のコードでは、*PeopleSoft* ドメインの値 *SHARE/1000* に対応する、*Items* クロスリファレンス マップの *Common* ドメインに含まれる全ての要素の値が検索されます。

```
xref:lookupXRefNVP("Items","PeopleSoft","<Setid>SHARE</Setid><ItemID>1000</ItemID>"⇒
,"Common",true())
```

ドメイン値マップ関数

このセクションでは、ドメイン値マップ関数について説明します。

dvm:lookupValue

構文

```
dvm:lookupValue(mapName, referenceElementName, referenceValue, elementName,
defaultValue, needAnException)
```

説明

dvm:lookupValue 関数は、ドメイン値マップの参照要素値を検索し、指定されている要素名の等値を文字列として返します。この形式の DVM 検索は、単一の参照要素を検索して、単一の要素値を返す場合に使用されます。参照ドメインまたは返されるドメインに複数の要素がある検索の場合は、dvm:lookupValueNVP 関数を使用して実行する必要があります。

パラメータ

パラメータ	説明
mapName	ドメイン値マップ名を文字列として指定します。
referenceElementName	ソース エlement名を文字列として指定します。
referenceValue	ソース値 (XSLT 変換のソースドキュメントにバインドされている XPath 式) を文字列として指定します。
elementName	ターゲット エlement名を文字列として指定します。
defaultValue	値が見つからない場合、デフォルト値が文字列として返されます。

パラメータ	説明
needAnException	"真" または "偽" を指定します。 needAnException パラメータが "真" に設定されていると、検索対象の値がマップ内に見つからない場合、例外が発生します。needAnException パラメータが "偽" に設定されていると、検索対象の値がマップ内に見つからない場合、空の値が返されます。

戻り値

dvm:lookupValue は、エレメントの値を含む文字列を返します。

次の理由で、例外が発生する可能性があります。

- 指定された名前の DVM が見つかりません。
- 指定されたエレメントが見つかりません。
- 指定されたソース値が空です。

例

次のコードでは、Long エレメントの値 *California* に対応する、StateCodes DVM クロスリファレンス マップの Short エレメントの値が検索されます。

```
dvm:lookupValue("StateCodes", "Long", "California", "Short", "CouldNotBeFound", True)
```

dvm:lookupValueNVP

構文

```
dvm:lookupValueNVP(mapName, referenceDomain, referenceNVP, targetDomain, defaultNVP, needAnException)
```

説明

dvm:lookupValueNVP 関数は、DVM の参照ドメイン エレメント値を検索し、指定されたドメインの全てのエレメントの等値を NVP リストとして返します。この形式の DVM 検索は、参照ドメインまたは返されるドメインに複数のエレメントが存在する場合に使用します。参照ドメインの全ての必須エレメントは、NVP 参照リストに含める必要がありますが、オプションのエレメント (おそらく識別子) は含める必要はありません。戻り文字列には、必須エレメントであるかどうかを問わず、ターゲットドメインの全てのエレメントの値が NVP リスト形式で含まれます。

パラメータ

パラメータ	説明
mapName	ドメイン値マップ名を文字列として指定します。
referenceDomain	ソースドメイン名を文字列として指定します。
referenceNVP	ソース エlementおよび値の NVP リストを文字列として指定します。
targetDomain	ターゲットドメイン名を文字列として指定します。
defaultNVP	値が見つからない場合、指定されているデフォルト値が文字列として返されます。
needAnException	"真" または "偽" を指定します。

戻り値

戻り文字列には、必須Elementであるかどうかを問わず、ターゲットドメインの全てのElementの値がNVPリスト形式で含まれます。

次のいずれかの理由で、例外が発生する可能性があります。

- 指定された名前の DVM が見つかりません。
- 指定されたドメインが見つかりません。
- 指定されたElementが見つかりません。
- 指定されたソース値が空です。

例

次のコードでは、*ChartElements* DVM の *PeopleSoft* ドメインの *BusinessUnit* Elementと *Chartfield* Elementに指定されている値を検索し、*UniqueGUID* Elementの値を返します。

```
dvm:lookupValueNVP ("ChartElements", "PeopleSoft", "<BusinessUnit>US100</BusinessUnit>=>
<Chartfield>ACCOUNT</Chartfield>", "UniqueGUID", "<UniqueGUID>CouldNotBeFound</Unique=>
GUID>", True)
```

dvm:lookup-dvm

構文

```
dvm:lookup-dvm (mapName, referenceElementName, referenceValue, elementName,
defaultValue, needAnException)
```

説明

`dvm:lookup-dvm` 関数は、ドメイン値マップの参照エレメント値を検索し、指定されているエレメント名の等値を文字列として返します。この形式の DVM 検索は、単一の参照エレメントを検索して、単一のエレメント値を返す場合に使用されます。参照ドメインまたは返されるドメインに複数のエレメントがある検索の場合は、`dvm:lookupValueNVP` 関数を使用して実行する必要があります。この関数の目的と機能は、`dvm:lookupValue` 関数と同じです。DVM の ESB 実装に定義されている関数名を模倣するために存在しています。

パラメータ

パラメータ	説明
<code>mapName</code>	ドメイン値マップ名を文字列として指定します。
<code>referenceElementName</code>	DVM 内のソース エレメントの名前を文字列として指定します。
<code>referenceValue</code>	ソース値 (XSLT 変換のソース ドキュメントにバインドされている XPath 式) を文字列として指定します。
<code>elementName</code>	DVM 内のターゲット エレメントの名前を文字列として指定します。
<code>defaultValue</code>	値が見つからない場合に、ターゲット エレメントに割り当てるデフォルト値を、文字列として指定します。
<code>needAnException</code>	"真" または "偽" を指定します。 <code>needAnException</code> パラメータが "真" に設定されていると、検索対象の値がマップ内に見つからない場合、例外が発生します。 <code>needAnException</code> パラメータが "偽" に設定されていると、検索対象の値がマップ内に見つからない場合、空の値が返されます。

戻り値

この関数は、DVM から、ソース エレメントの値がソース値と同じである、ターゲット エレメントの値を検索します。ソース値は、XSLT 変換のソース ドキュメントにバインドされている XPath 式です。例外は変換中に評価され、結果値は検索のソース値として渡されます。

次の理由で、例外が発生する可能性があります。

- 指定された名前の DVM が見つかりません。
- 指定されたエレメントが見つかりません。

- 指定されたソース値が空です。

例

次のコードでは、*Abbrev* エLEMENTの値 *Calif*に対応する、*StateCodes* DVM マップの *Short* エLEMENTの値が検索されます。

```
dvm:lookup-dvm("StateCodes","Abbrev","Calif","Short","CouldNotBeFound", True)
```

Generate-Guid 関数

このセクションでは、`generate-guid` 関数について説明します。

generate-guid

構文

```
generate-guid()
```

説明

この関数を使用して、GUID を作成します。

パラメータ

なし

戻り値

この関数は、ランダムに生成されたグローバル一意識別子 (GUID) を含む文字列を返します。

例

次のコードは、新しい共通キー値として使用できる、ランダム GUID を作成します。

```
generate-guid()
```

SetID 関数

このセクションでは、`SetID` XSLT 拡張関数について説明します。

SetID:lookupSetCtrlValues

構文

```
SetID:lookupSetCtrlValues (SetId, LookupType, dvmTranslate, dvmMapName,  
SourceElementName, TargetElementName, needAnException)
```

説明

lookupSetCtrlValues 関数を使用して、レコード グループまたはレコードのコンテキストのセットID に関連付けられているセット コントロール値のリストを検索します。マップ名、ソース エlement名、およびターゲット Element名が指定されている場合、オプションで、各セット コントロール値を DVM マップを利用して変換できます。

パラメータ

パラメータ	説明
SetId	検索対象のセットID 値。
LookupType	検索タイプは、検索のタイプの指定に使用されます。レコードグループ名には [1] を、レコード名には [2] を指定します。
LookupValue	値は、LookupType によって指定されているレコード名またはレコードグループ ID のいずれかです。
dvmTranslate	指定された DVM 名を使用した共通 ID への変換を行う場合は、"真" を指定します。変換が不要な場合は、"偽" を指定します。
dvmMapName	リクエストされた場合に変換で使用する DVM マップ。
SourceElementName	リクエストされた場合に DVM 変換で使用するソース Element名。
TargetElementName	リクエストされた場合に DVM 変換で使用するターゲット Element名。
needAnException	"真" または "偽" を指定して、セット コントロール値が見つからない場合、例外が発生するかどうかを示します。

戻り値

この関数は、XSLT で解析できる連結文字列として、セット コントロール値のリストまたは変換後のセット コントロール値のリストを返します。

例

この例では、VENDOR レコード グループのセットID SHARE に関連付けられているセット コントロール値 (この例では PeopleSoft ビジネス ユニットを表します) を検索し、BusinessUnit DVM マッピングによってビジネス ユニットの共通 ID に変換します。セット コントロール値が見つからない場合、例外がリクエストされます。

```
xref:lookupSetCtrlValues("SHARE","1", "VENDOR",true(),"BusinessUnit","PSFT_BU","COM⇒MON",true())
```

次の理由で、例外が発生する可能性があります。

- 指定された DVM マップ名が無効です。
- 指定されたソース エlement名は、DVM マップに関連付けられていません。
- 指定されたターゲット Element名は、DVM マップに関連付けられていません。
- DVM マップ内で、セット コントロール値に対する変換後の値が見つかりません。
- `needAnException` が "真" に設定されていると、指定されたセットID/レコードまたはレコード グループ ID に対してセット コントロール値が見つからない場合、例外が発生します。

この例では、*VENDOR_LOC*レコードのセットID *SHARE* に関連付けられているセット コントロール値 (この例では PeopleSoft ビジネス ユニットを表します) を検索し、*BusinessUnit* DVM マッピングによってビジネス ユニットの共通 ID に変換します。セット コントロール値が見つからない場合、例外はリクエストされません。

```
xref:lookupSetCtrlValues("SHARE","2", "VENDOR_LOC",true(),"BusinessUnit","PSFT_BU",⇒  
"COMMON",false())
```

この例では、*VENDOR*レコード グループのセットID *SHARE* に関連付けられているセット コントロール値 (この例では PeopleSoft ビジネス ユニットを表します) を検索します。セット コントロール値が見つからない場合、例外がリクエストされます。

```
xref:lookupSetCtrlValues("SHARE","1", "VENDOR",false(),"","",true())
```

指定されたセットID/レコードまたはレコード グループ ID に対してセット コントロール値が見つからない場合、例外が発生します。

PeopleCode を使用したマップへのアクセス

アプリケーション インテグレーション フレームワーク クラスについて

アプリケーション インテグレーション フレームワーク クラスは、値マップの検索、削除、およびロードを実行する関数を提供します。次の関数は、XSLT 変換用に提供されている XPath 拡張関数と同じ機能を提供します。

DVM ユーティリティ クラス

PeopleCode 開発において、変換中にドメイン値マップ (DVM) のデータにアクセスするための関数です。この関数は、XSLT 変換用に提供されている XPath 拡張関数と同じ機能を提供します。

SetID ユーティリティ クラス

PeopleCode 開発において、変換中に DVM に格納されている SetID データにアクセスするための関数です。

Xref ユーティリティ クラス

PeopleCode 開発において、変換中に動的 (クロスリファレンス) 値マップのデータにアクセスするための関数です。

アプリケーション インテグレーション フレームワーク Type クラスのインポート方法

アプリケーション インテグレーション フレームワーク Type クラスは、Rowset、Field、Record などとは異なり、ビルトイン クラスではなく、これらはアプリケーション クラスです。作成する PeopleCode プログラム内でこれらのクラスを使用するには、その前に、これらをプログラムにインポートする必要があります。

インポート文では、パッケージ内の全てのクラスを宣言する、または特定のアプリケーション クラスを宣言します。アプリケーション インテグレーション フレームワークのクラスをインポートする場合、特定のニーズに限定したアプリケーション パッケージ内の関数クラスをインポートすることをお勧めします。

この関数クラスは、次のアプリケーション パッケージに格納されています。

- EOTF_CORE:DVM
- EOTF_CORE:SetId
- EOTF_CORE:Xref

次のいずれかのインポート文を使用します。

```
import EOTF_CORE:DVM:Functions;
```

```
import EOTF_CORE:Setid:Functions;
import EOTF_CORE:Xref;
```

DVM ユーティリティ クラス メソッド

このセクションでは、DVM ユーティリティ クラス メソッドについて説明します。メソッドは、アルファベット順で説明します。

ExtractData

構文

```
ExtractData (mapName, domainList, tempRecName, instance)
```

説明

DVM データ抽出関数を使用して、セット ベースの SQL を作成および実行し、指定されたテーブルに DVM データを抽出できます。データを抽出する前に、指定されたプロセス インスタンスの一時テーブルから全てのデータが削除されます。エレメント値は、指定されているドメイン リストの順序を使用し、次に、値マップ定義で指定されているエレメントの順序を使用して、一時レコードのフィールドにマッピングされます。ドメインが指定されていない場合は、エレメント値は、値マップ定義で指定されているエレメントの順序のみを使用して、一時レコードのフィールドにマッピングされます。この関数は、大量のデータの処理に使用します。これにより、DVM フレームワークの物理的なデータ永続性層の抽象化層が提供されます。

一時テーブルには、次の設計上の制約があります。

1. 1 列目は PROCESS_INSTANCE になります。
2. 2 列目は UniqueGUID (EOTF_COMMONELEMENT) に使用されます。
3. 残りの列は、指定されたドメイン リストのエレメントに対応します。
4. 一時テーブルの列名は、DVM のエレメント名に一致している必要はありません。
5. 一時テーブルでは、ProcessInstance、UniqueGUID がキーになります。
6. 指定された各ドメインのエレメントのインデックスを作成すると、パフォーマンスが向上する可能性があります。

パラメータ

パラメータ	説明
mapName	静的値マップの定義 (DVM) の名前を文字列として指定します。

パラメータ	説明
domainList	データを抽出する値マップドメインの名前。マップに含まれる全てのドメインのデータを抽出するには、このパラメータに null または空の配列を指定します。
tempRecName	抽出したデータを格納する一時テーブルの名前を文字列として指定します。
instance	プロセス インスタンスを適用する場合に使用する数値を整数で指定します。

戻り値

プロセスが成功したか失敗したかを示すブール値。

例

```
Local EOTF_CORE:DVM:Functions &dvm = create EOTF_CORE:DVM:Functions();
Local string &mapName = "States";
Local array of string &domainList = CreateArray("PS");
Local string &tempRecName = Record.HR_STATES_TAO;
Local boolean &success =
&dvm.ExtractData(&mapName, &domainList, &tempRecName, &instance);
```

LookupValue

構文

LookupValue (*mapName*, *referenceElementName*, *referenceValue*, *elementName*, *defaultValue*, *needAnException*)

説明

ドメイン値マップの参照エレメント値を検索し、指定されているエレメント名の等値を返します。この形式の DVM 検索は、単一の参照エレメントを検索して、単一のエレメント値を返す場合に使用されます。

パラメータ

パラメータ	説明
mapName	静的 (DVM) 値マップの定義の名前を文字列として指定します。
referenceElementName	値を検索する DVM 内のエレメントの名前を文字列として指定します。
referenceValue	検索対象の DVM 内のエレメントの値を文字列として指定します。

パラメータ	説明
elementName	等値を返す DVM 内のエレメントの名前を文字列として指定します。
defaultValue	needAnException が "偽" で、エラーが発生した場合に、文字列として返されるデフォルト値。
needAnException	エラー メッセージを返す場合は "真"、デフォルト値を返す場合は "偽"。

戻り値

DVM の elementName の等値、またはデフォルト値を返します。

例

この例は、&TestName DVM の UniqueGUID エレメントの値 &guid1 を検索し、&EBS1 エレメントの等値を返す場合に使用されます。

```
Local string &returnValue = &dvm.LookupValue(&TestName, &UniqueGUID, &guid1, &EBS1, ⇒
  "Value not found.", True);
```

この例は、&TestName DVM の &EBS1 エレメントに存在しない値を検索し、NeedAnException パラメータが "偽" であった場合に、デフォルト値が返されることを確認します。

```
&returnValue = &dvm.LookupValue(&TestName, &EBS1, "NotFound", &UniqueGUID, &ValueNo ⇒
  tFound, False);
```

LookupValue1M

構文

```
LookupValue1M (mapName, referenceElementName, referenceValue, elementName,
  needAnException)
```

説明

ドメイン値マップの参照エレメント値を検索し、指定されているエレメント名の等値を NVP リストとして返します。この形式の DVM 検索は、単一の参照エレメントを検索して、指定されたエレメント用の 1 対多の等値を返す場合に使用されます。

パラメータ

パラメータ	説明
mapName	静的 (DVM) 値マップの定義の名前を文字列として指定します。

パラメータ	説明
referenceElementName	値を検索する DVM 内のエレメントの名前を文字列として指定します。
referenceValue	検索対象の DVM 内のエレメントの値を文字列として指定します。
elementName	等値を返す DVM 内のエレメントの名前を文字列として指定します。
needAnException	エラーメッセージを返す場合は "真"、デフォルト値を返す場合は "偽"。

戻り値

DVM 内の `elementName` の等値を含む文字列の配列を返します。

例

この例では、`&TestName` DVM の `UniqueGUID` エレメントの値 `&guid1` を検索し、`&EBS1` エレメントの等値を返します。

```
&returnValue = &dvm.LookupValue1M(&TestName, &UniqueGUID, &guid1, &EBS1, True);
```

この例では、`&TestName` DVM の `UniqueGUID` エレメントに存在しない値を検索し、`NeedAnException` が "偽" の場合に返される値がないことを確認します。

```
&returnValue = &dvm.LookupValue1M(&TestName, &UniqueGUID, "NotFound", &EBS1, False⇒);
```

LookupValueNVP

構文

```
LookupValueNVP (mapName, referenceDomain, referenceNVP, targetDomain, defaultNVP,  
needAnException)
```

説明

DVM の参照ドメイン エレメント値を検索し、指定されたドメインの全てのエレメントの等値を NVP リストとして返します。この形式の DVM 検索は、参照ドメインまたは返されるドメインに複数のエレメントが存在する場合に使用します。参照ドメインの全ての必須エレメントは、NVP 参照リストに含める必要がありますが、オプションのエレメント (おそらく識別子) は含める必要はありません。戻り文字列には、必須エレメントであるかどうかを問わず、ターゲットドメインの全てのエレメントの値が NVP リスト形式で含まれます。

パラメータ

パラメータ	説明
mapName	静的 (DVM) 値マップの定義の名前を文字列として指定します。
referenceDomain	値を検索する DVM 内のドメインの名前を文字列として指定します。
referenceNVP	検索対象の、DVM ドメイン内の要素と値のペアを、DataElement の配列として指定します。
targetDomain	等値を返すドメインの名前を文字列として指定します。
defaultNVP	needAnException が "偽" で、エラーが発生した場合に、DataElement の配列として返されるデフォルト値 (NVP)。
needAnException	エラー メッセージを返す場合は "真"、デフォルト値が設定された NVP を返す場合は "偽"。

戻り値

DataElement の配列。ターゲットドメインの要素の等値を含む名前と値のペア、またはデフォルト値。

例

この例は、&TestName DVM の UniqueGUID エLEMENTの値 &guid1 を検索し、&RTK ドメインの等値を返す場合に使用されます。

```
Local array of EOTF_CORE:Common:DataElement &UniqueGUIDrequestValues = CreateArray(⇒
create EOTF_CORE:Common:DataElement (&UniqueGUID));

&UniqueGUIDrequestValues [1].value = &guid1;
Local array of EOTF_CORE:Common:DataElement &returnValue = &dvm.LookupValueNVP(&TestName, &UniqueGUID, &UniqueGUIDrequestValues, &RTK, &DefaultRTKValues, True);
```

DVM ユーティリティ クラス プロパティ

このセクションでは、DVM ユーティリティ クラス プロパティについて説明します。

exceptionCaught

説明

needAnException パラメータが "偽" になっていたために、検索メソッドによって例外が抑制される場合は、値が "真" になります。

exceptionDetails

説明

発生した例外の詳細を含む例外オブジェクトです。

SetID ユーティリティ クラス メソッド

このセクションでは、SetId ユーティリティ クラス メソッドについて説明します。メソッドは、アルファベット順で説明します。

lookupSetCtrlValues

構文

lookupSetCtrlValues (*SetId, LookupType, LookupName, dvmTranslate, dvmMapName, PsftElementName, CommonElementName, needAnException*)

説明

lookupSetCtrlValues 関数を使用して、レコードまたはレコード グループのコンテキストのセットID に関連付けられているセット コントロール値を検索できます。さらに、指定の DVM マップを利用して、PeopleSoft ドメインの要素から共通ドメインの要素にセット コントロール値を変換するようにリクエストすることができます。通常の PeopleSoft 実装では、セット コントロール値は PeopleSoft ビジネス ユニットの表します。その場合、変換用のビジネス ユニットの マッピングに、DVM を定義します。

パラメータ

パラメータ	説明
SetId	検索対象のセットID 値を文字列として指定します。
LookupType	実行する検索のタイプ。レコード グループ ID の場合は 1、レコード名の場合は 2 が有効な値です。
LookupName	レコード グループ ID またはレコード名を文字列として指定します。指定された LookupType に対応する値を指定します。
dvmTranslate	指定された静的値マップ (DVM) を使用してセット コントロール値の変換を行う場合は、"真"を指定します。
dvmMapName	変換で使用する静的な値マップ (DVM) の名前を文字列として指定します。
PsftElementName	検索で使用する、DVM 内の PeopleSoft ドメインに属する要素の名前を、文字列として指定します。

パラメータ	説明
CommonElementName	等値を返す、DVM 内の共通ドメインに属するエレメントの名前を、文字列として指定します。
needAnException	セット コントロール値が見つからない場合に、例外を発生させる場合は、"真"。そのような場合に空の文字列をリクエストする場合は、"偽"。

戻り値

文字列の配列として、等値が返されます。

例

この例では、*VENDOR* レコード グループのセットID *SHARE* に関連付けられているセット コントロール値 (この例では PeopleSoft ビジネス ユニットを表します) を検索し、*BusinessUnit* DVM マッピングによってビジネス ユニットの共通 ID に変換します。セット コントロール値が見つからない場合、例外がリクエストされます。

```
&oSetIdUtil = create EOTF_CORE:SetId:Functions();

    Local array of string &arrReturnValue = CreateArrayRept("", 0);

    try

        &arrReturnValue = &oSetIdUtil.lookupSetCtrlValues("SHARE", "1", "VENDOR", true,
e, "BusinessUnit", "PSFT_BU", "COMMON", true);

    catch Exception &exReturn
    . . .
        {Your Exception Handling Logic Here}
    . . .
    end-try;
```

lookupSetID

構文

```
lookupSetID (LookupValue, dvmTranslate, dvmMapName, PsftElementName,
CommonElementName, LookupType, LookupName, needAnException)
```

説明

lookupSetID 関数を使用して、レコードまたはレコード グループのコンテキストのセット コントロール値に対応するセットIDを検索できます。さらに、指定の DVM マップを利用して、共通ドメインのエレメントから PeopleSoft ドメインのエレメントに、検索に使用するセット コントロール値を変換するようにリクエストすることができます。通常の PeopleSoft 実装では、セット コントロール値は PeopleSoft ビジネス ユニットを表します。その場合、ビジネス ユニット マッピング変換に、DVM を定義します。

パラメータ

パラメータ	説明
LookupValue	セットID の検索で使用する値を文字列として指定します。dvmTranslate が "偽" の場合はセットコントロール値、dvmTranslate が "真" の場合は共通エレメントのいずれかを値に指定します。
dvmTranslate	指定された静的値マップ (DVM) を使用してセットコントロール値の変換を行う場合は、"真" を指定します。
dvmMapName	変換で使用する静的な値マップ (DVM) の名前を文字列として指定します。
PsftElementName	相当するセットコントロール値を取得する、DVM 内の PeopleSoft ドメインに属するエレメントの名前を、文字列として指定します。
CommonElementName	検索で使用する、DVM 内の共通ドメインに属するエレメントの名前を、文字列として指定します。
LookupType	実行する検索のタイプ。レコードグループ ID の場合は 1、レコード名の場合は 2 が有効な値です。
LookupName	レコードグループ ID またはレコード名を文字列として指定します。指定された LookupType に対応する値を指定します。
needAnException	セットID が見つからない場合に、例外を発生させる場合は、"真"。そのような場合に空の文字列をリクエストする場合は、"偽"。

戻り値

等値を文字列として指定します。

例

次のコードは、*VENDOR* レコードグループの *BusinessUnit* DVM マッピングに含まれるビジネス 有効な共通 ID *BUID001* に関連付けられているセットID を検索します。setID が見つからない場合、例外はリクエストされません (空の文字列が返されます)。

```
&oSetIdUtil = create EOTF_CORE:SetId:Functions ();
Local string &ReturnValue;

&ReturnValue = &oSetIdUtil.lookupSetId("BUID001", "BusinessUnit", "PSFT_BU", "COMMO⇒
N", "1", "VENDOR", false);
```

Xref ユーティリティ クラス メソッド

このセクションでは、Xref クラス メソッドについて説明します。メソッドは、アルファベット順で説明します。

BulkPopulateDomainData

構文

BulkPopulateDomainData (*mapName*, *domainList*, *dataRows*, *dataValuesPath*, *mode*)

説明

この XREF ドメイン データのバルク ロード関数は、XML からクロスリファレンス データをロードします。XML ノードセットに設定されているデータ行を順番に処理し、指定されているドメイン リストとデータ値のパスを使用して、各行に設定されている各ドメインの値にアクセスします。その後、各データ行は、できるだけ効率よく処理するため、一括挿入を使用して、指定されたマップの XREF フレームワークに挿入されます。この関数は、大量のデータの処理に使用します。これにより、XREF フレームワークの物理的なデータ永続性層の抽象化層が提供されます。挿入はバルク モードで実行されるため、挿入シーケンスが完了するまで重複は検索されません。重複が見つかった場合、データ行はまったく挿入されません。ドメインに複数のエレメントが含まれる場合、ドメインの各値には、ドメインに含まれる各エレメントの値を、値マップ オプションで指定されている区切り文字で区切った複合値を指定する必要があります。このシナリオでは、変換フレームワークにデータを挿入する前に、この関数によって、ドメイン エレメント値の連結が解除されます。

パラメータ

パラメータ	説明
mapName	動的 (クロスリファレンス) 値マップの名前を文字列として指定します。
domainList	ドメイン名の配列。インポートするデータ値の順序を指定します。
dataRows	XML ノードの配列で、配列内の各ノードは 1 つのデータ行に相当します。
dataValuesPath	データ行内のデータ値を含む各ノードを指定するために使用する (各データ行に関連する) XPath 式。このパスが評価されるときに返されるノードセットには、指定された domainList 内のドメインと同じ数のノードが含まれている必要があります。
mode	ADD モードのみがサポートされます。

戻り値

プロセスが成功したか失敗したかを示すブール値。

例

```
Local EOTF_CORE:Xref:Functions &xref = create EOTF_CORE:Xref:Functions();
Local string &mapName = "VendorIDs";
Local array of string &domainList = ...;
Local array of XmlNode &dataRows = ...;
Local string &dataValuesPath = "...";
Local string &mode = "ADD";
Local boolean &success =
&xref.BulkPopulateDomainData(&mapName, &domainList, &dataRows, &dataValuesPath, &mode);
```

BulkPopulateElementData

構文

```
BulkPopulateElementData (mapName, elementList, dataRows, dataValuesPath, mode)
```

説明

この XREF エlement データのバルクロード関数は、XML からクロスリファレンス データをロードします。XML ノードセットに設定されているデータ行を順番に処理し、指定されている Element と XPath 式のペアを使用して、各行に設定されている Element の値にアクセスします。その後、各データ行は、できるだけ効率よく処理するため、一括挿入を使用して、指定されたマップの XREF フレームワークに挿入されます。この関数は、大量のデータの処理に使用します。これにより、XREF フレームワークの物理的なデータ永続性層の抽象化層が提供されます。挿入はバルク モードで実行されるため、挿入シーケンスが完了するまで重複は検索されません。重複が見つかった場合、データ行はまったく挿入されません。

パラメータ

パラメータ	説明
mapName	動的 (クロスリファレンス) 値マップの名前を文字列として指定します。
elementList	Element 名の配列。インポートするデータ値の順序を指定します。
dataRows	XML ノードの配列で、配列内の各ノードは 1 つのデータ行に相当します。
dataValuesPath	データ行内のデータ値を含む各ノードを指定するために使用する (各データ行に関連する) XPath 式。このパスが評価されるときに返されるノードセットには、指定された elementList 内の Element と同じ数のノードが含まれている必要があります。

パラメータ	説明
mode	ADD モードのみがサポートされます。

戻り値

プロセスが成功したか失敗したかを示すブール値。

例

```
Local EOTF_CORE:Xref:Functions &xref = create EOTF_CORE:Xref:Functions();
Local string &mapName = "VendorIDs";
Local array of string &elementList = ...;
Local array of XmlNode &dataRows = ...;
Local string &dataValuesPath = "...";
Local string &mode = "ADD";
Local boolean &success = &xref.BulkPopulateElementData(&mapName, &elementList, &dataRows, &dataValuesPath, &mode);
```

ExtractData

構文

```
ExtractData (mapName, domainList, tempRecName, instance)
```

説明

XREF データ抽出関数を使用して、セット ベースの SQL を作成および実行し、指定されたテーブルにクロスリファレンス データを抽出できます。データを抽出する前に、指定されたプロセス インスタンスの一時テーブルから全てのデータが削除されます。エレメント値は、指定されているドメイン リストの順序を使用し、次に、値マップ定義で指定されているエレメントの順序を使用して、一時レコードのフィールドにマッピングされます。ドメインが指定されていない場合は、エレメント値は、値マップ定義で指定されているエレメントの順序のみを使用して、一時レコードのフィールドにマッピングされます。この関数により、XREF フレームワークの物理的なデータ永続性層の抽象化層が提供されます。この関数は、大量のデータの処理に使用します。

一時テーブルには、次の設計上の制約があります。

- 1 列目は PROCESS_INSTANCE になります。
- 2 列目は UniqueGUID (EOTF_COMMONELEMENT) に使用されます。
- 3 残りの列は、指定されたドメインのエレメントに対応します。

注: 3 列目の名前が EOTF_IMPORT_FLG の場合、エクスポート中に *[N]* という値が設定されます。

- 4 一時テーブルの列名は、マップのエレメント名に一致している必要はありません。
- 5 一時テーブルでは、ProcessInstance、UniqueGUID が一意のキーになります。

6. 指定された各ドメインの要素のインデックスを作成すると、パフォーマンスが向上する可能性があります。

パラメータ

パラメータ	説明
mapName	動的なクロスリファレンス マップの名前を文字列として指定します。
domainList	データを抽出する値マップドメインの名前。マップに含まれる全てのドメインのデータを抽出するには、このパラメータに null または空の配列を指定します。
tempRecName	抽出したデータを格納する一時テーブルの名前を文字列として指定します。
instance	プロセス インスタンスを適用する場合に使用する数値を整数で指定します。

戻り値

プロセスが成功したか失敗したかを示すブール値。

例

```
Local EOTF_CORE:Xref:Functions &xref = create EOTF_CORE:Xref:Functions();
Local string &mapName = "VendorIDs";
Local array of string &domainList = CreateArray("PS");
Local string &tempRecName = Record.HR_VENDOR_TAO;
Local boolean &success =
&xref.ExtractData(&mapName, &domainList, &tempRecName, &instance);
```

LookupValue

構文

```
LookupValue (mapName, referenceElementName, referenceValue, elementName,
needAnException)
```

説明

クロスリファレンス値マップの参照要素値を検索し、指定されている要素名の等値を返します。この形式の検索は、単一の参照要素を検索して、単一の要素値を返す場合に使用されます。

パラメータ

パラメータ	説明
mapName	動的 (クロスリファレンス) 値マップの定義の名前を文字列として指定します。
referenceElementName	値を検索する Xref 内のエレメントの名前を文字列として指定します。
referenceValue	検索対象の Xref 内のエレメントの値を文字列として指定します。
elementName	等値を返す Xref 内のエレメントの名前を文字列として指定します。
needAnException	エラー メッセージを返す場合は "真"、空白を返す場合は "偽"。

戻り値

Xref 内の `elementName` の等値を含む文字列の配列。

例

この例では、`&TestName` クロスリファレンス マップの `UniqueGUID` エレメントの値 `&guid1` を検索し、`&EBS1` エレメントの等値を返します。

```
Local string &returnValue = &xref.LookupValue(&TestName, &UniqueGUID, &guid1, &EBS1 =>
, True);
```

この例は、`&TestName` クロスリファレンス マップの `&EBS1` エレメントに存在しない値を検索し、`NeedAnException` パラメータが "偽" であった場合に、デフォルト値 (空白) が返されることを確認します。

```
&returnValue = &xref.LookupValue(&TestName, &EBS1, "NotFound", &UniqueGUID, False);
```

LookupValue1M

構文

```
LookupValue1M ( mapName, referenceElementName, referenceValue, elementName,
needAnException )
```

説明

クロスリファレンス値マップの参照エレメント値を検索し、名前付きの値ペア (NVP) リストとして、指定されているエレメント名の等値を返します。この形式の検索は、単一の参照エレメントを検索して、指定されたエレメント用の 1 対多の等値を返す場合に使用されます。

パラメータ

パラメータ	説明
mapName	動的 (クロスリファレンス) 値マップの定義の名前を文字列として指定します。
referenceElementName	値を検索する Xref 内のエレメントの名前を文字列として指定します。
referenceValue	検索対象の Xref 内のエレメントの値を文字列として指定します。
elementName	等値を返す Xref 内のエレメントの名前を文字列として指定します。
needAnException	エラー メッセージを返す場合は "真"、空白を返す場合は "偽"。

戻り値

クロスリファレンス内の `elementName` の等値を含む文字列の配列を返します。

例

この例は、`&TestName` クロスリファレンス マップの `UniqueGUID` エレメントの値 `&guid1` を検索し、`&EBS1` エレメントの等値を返します。

```
Local array of string &returnValue;
&returnValue = &xref.LookupValue1M(&TestName, &UniqueGUID, &guid1, &EBS1, True);
```

この例では、`&TestName` クロスリファレンス マップの `UniqueGUID` エレメントに存在しない値を検索し、`NeedAnException` が "偽" の場合に返される値がないことを確認します。

```
&returnValue = &xref.LookupValue1M(&TestName, &UniqueGUID, "NotFound", &EBS1, False);
```

LookupValueNVP

構文

```
LookupValueNVP (mapName, referenceDomain, referenceNVP, targetDomain, needAnException)
```

説明

クロスリファレンス マップの参照ドメイン エレメント値を検索し、指定されたドメインの全てのエレメントの等値を NVP リストとして返します。この形式の検索は、参照ドメインまたは返されるドメインに複数のエレメントが存在する場合に使用します。参照ドメインの全ての必須エレメントは、NVP 参照リストに含める必要がありますが、オプションのエレメント (おそらく識別子) は含める必要はありません。戻り文字列には、

必須要素であるかどうかを問わず、ターゲット ドメインの全ての要素の値が NVP リスト形式で含まれます。

パラメータ

パラメータ	説明
mapName	動的 (クロスリファレンス) 値マップの定義の名前を文字列として指定します。
referenceDomain	値を検索する XREF 内のドメインの名前を文字列として指定します。
referenceNVP	検索対象の、参照ドメイン内の要素と値のペアを、DataElement の配列として指定します。
targetDomain	等値を返すドメインの名前を文字列として指定します。
needAnException	エラー メッセージを返す場合は "真"、デフォルト値が設定された NVP を返す場合は "偽"。

戻り値

ターゲット ドメインの要素の等値を含む、名前と値のペアの DataElement 配列、またはデフォルト値。

例

この例では、&TestName クロスリファレンス マップの UniqueGUID ドメインの値 &guid1 を検索し、&RTK ドメインの等値を返します。

```
Local array of EOTF_CORE:Common:DataElement &returnValue = &xref.LookupValueNVP(&T⇒estName, &UniqueGUID, &UniqueGUIDrequestValues, &RTK, True);
```

MarkForDelete

構文

```
MarkForDelete (mapName, elementName, elementValue)
```

説明

指定されている要素がシングル ドメインの唯一の要素である場合、クロスリファレンス マップ内の値を削除します。複数のドメインが要素を参照しているか、複数のプライマリ要素があるドメインで要素が使用されている場合は、xref:markForDeleteNVP 関数を使用します。要素の値は削除済としてマークされます。1 行に 2 つのマッピングしかない場合に、一方の値が削除指定されると、もう一方のドメインの値も削除されます。削除指定された要素値は、存在していないものとして扱われます。従って、xref:populateXRefRow 関数を ADD モードで使用して、同じ要素をロードできます。

ただし、削除指定されているエレメント値を、`xref:populateXRefRow` 関数の LINK モードで参照値として使用すると、エラーが発生します。

パラメータ

パラメータ	説明
mapName	動的 (クロスリファレンス) 値マップの定義の名前を文字列として指定します。
elementName	値を削除する Xref 内のエレメントの名前を文字列として指定します。
elementValue	削除対象の Xref 内のエレメントの値を文字列として指定します。

戻り値

削除が成功した場合は "真"。

例

この例では、*Customers* クロスリファレンス マップの *PS* エレメントの値 *PS001* が削除されます。

```
&return=&xref.MarkForDelete("Customers", "PS", "PS001")
```

MarkForDeleteNVP

構文

```
MarkForDeleteNVP (mapName, referenceDomain, referenceNVP)
```

説明

指定されたドメインのクロスリファレンス マップ内の値セットを削除します。エレメントの値は削除済としてマークされます。1 行に 2 つのマッピングしかない場合に、一方の値が削除指定されると、もう一方のドメインの値も削除されます。削除指定された値は、存在していないものとして扱われます。従って、`xref:populateXRefRowNVP` 関数を ADD モードで使用して、同じエレメントをロードできます。ただし、削除指定されているエレメント値を、`xref:populateXRefRowNVP` 関数の LINK モードで参照値として使用すると、エラーが発生します。

パラメータ

パラメータ	説明
mapName	動的 (クロスリファレンス) 値マップの定義の名前を文字列として指定します。

パラメータ	説明
referenceDomain	値を削除する Xref 内のドメインの名前を文字列として指定します。
referenceNVP	削除対象の、参照ドメイン内のエレメントと値のペアを、DataElement の配列として指定します。

戻り値

削除が成功した場合は "真"。

例

この例では、Items クロスリファレンス マップから、EBS ドメインの指定された値を削除します。

```
Local array of EOTF_CORE:Common:DataElement &ebsNVP1 = CreateArrayRept(create EOTF_
CORE:Common:DataElement(&EBS1), 1);
&ebsNVP1 [1].value = &value1;
Local boolean &return = &xref.MarkForDeleteNVP("Items", &EBS1, &ebsNVP1);
```

PopulateData

構文

PopulateData (*mapName*, *domainName*, *tempRecName*, *recName*, *instance*)

説明

XREF データ ロード関数を使用して、セット ベースの SQL を作成および実行し、指定されたソースレコード (テーブルまたはビュー) からクロスリファレンス データをロードできます。この関数により、XREF フレームワークの物理的なデータ永続性層の抽象化層が提供されます。この関数は、大量のデータの処理に使用します。

この関数では、次のステップが作成および実行されます。

1. 指定されたプロセス インスタンスの一時テーブルから全てのデータが削除されます。
2. 指定されたドメインの既存の XREF データが、インポート フラグが *[N]* に設定されている一時テーブルに抽出されます。

注: エレメント値は、(名前ではなく) マップ エレメントの順序に従って、レコードのフィールドにマップされます。

3. 更新フラグが *[Y]* に設定されている場合、重複のないステージ データの行が、一時テーブルに既に存在していない場合は挿入されます。

注: 一致するフィールド名を基に、一時テーブルからステージングへと、フィールドがマッピングされま

4. 一時テーブルで GUID が空白の部分には、GUID が作成されます。
5. インポート フラグが [Y] に設定されている場合、XREF データが一時テーブルから挿入されます。
6. ステージ データが更新され、一時テーブルの GUID が設定されます。

一時テーブルには、次の設計上の制約があります。

1. 1 列目は PROCESS_INSTANCE になります。
2. 2 列目は UniqueGUID (EOTF_COMMONELEMENT) に使用されます。
3. 3 列目の名前は EOTF_IMPORT_FLG である必要があります (この関数による内部使用に必要)。
4. 残りの列は、指定されたドメインの要素に対応します。
5. 一時テーブルの列名は、マップの要素名に一致している必要はありません。
6. 一時テーブルの列名は、ステージ テーブルの対応しているフィールドと一致する必要があります。
7. 一時テーブルでは、ProcessInstance、UniqueGUID がキーになり、重複 (空白) が許されます。
8. 指定された各ドメインの要素のインデックスを作成すると、パフォーマンスが向上する可能性があります。

注: この関数はどのデータも検証しないため、ロードするデータが、値マップ定義で定義されている制約に違反していないことをユーザーが確認する必要があります。データ ルールに違反している場合、将来、マップの使用に影響が出る可能性があります。既存のデータ値よりも一時テーブルのフィールドを短く定義したことで、既存のデータ値が切り捨てられる可能性については、この関数は対処しません。このような切り捨てが起きた場合、エラーはスローされませんが、既存の値はステージ値と適切に対応付けされず、意図せず重複データが作成される可能性があります。

パラメータ

パラメータ	説明
mapName	動的 (クロスリファレンス) 値マップの名前を文字列として指定します。
domainName	データのロード先の値マップドメインの名前を文字列として指定します。
tempRecName	抽出した既存の XREF データを格納する一時テーブルの名前を、文字列として指定します。
recName	データをロードするステージングレコードの名前を文字列として指定します。

パラメータ	説明
instance	プロセス インスタンスを適用する場合に使用する数値を整数で指定します。

戻り値

プロセスが成功したか失敗したかを示すブール値。

例

```
Local EOTF_CORE:Xref:Functions &xref = create EOTF_CORE:Xref:Functions();
Local string &mapName = "VendorIDs";
Local string &domainName = "PS";
Local string &tempRecName = Record.HR_VENDOR_TAO;
Local string &recName = Record.HR_VENDOR_STG;
Local boolean &success =
&xref.PopulateData(&mapName, &domainName, &tempRecName, &recName, &instance);
```

PopulateValue

構文

PopulateValue (*mapName, referenceElementName, referenceValue, elementName, elementValue, mode*)

説明

クロスリファレンス マップの参照エレメント値を検索し、同じデータ行の別のエレメントに等値をロードします。この形式のロードは、単一の参照エレメントを検索して、単一のエレメント値をロードする場合に使用されます。

パラメータ

パラメータ	説明
mapName	動的 (クロスリファレンス) 値マップの定義の名前を文字列として指定します。
referenceElementName	値を検索する Xref 内のエレメントの名前を文字列として指定します。
referenceValue	検索対象の Xref 内のエレメントの値を文字列として指定します。
elementName	等値の提供先の Xref 内のエレメントの名前を文字列として指定します。
elementValue	エレメントに提供する等値を文字列として指定します。

パラメータ	説明
mode	この関数がエレメントをロードするモードを示します。ADD、LINK、または UPDATE のいずれかの値を指定できます。モードパラメータの値は、大文字と小文字が区別され、大文字だけで指定される必要があります。

戻り値

データがロードされた Xref 行の UniqueGUID 値。

例

この例は、*&TestName* クロスリファレンス マップの *UniqueGUID* エレメントの値 *&guid1* を検索し、*&EBS1* エレメントの等値を *&value1* に更新します。

```
&value1 = &value1 | "0";
&returnValue = &xref.PopulateValue(&TestName, &UniqueGUID, &guid1, &EBS1, &value1 =>
, &UPDATE);
```

この例は、*UniqueGUID* エレメントの値 *&guid3* と *EBS1* エレメントの値 *&value3* を *&TestName* クロスリファレンス マップ データに追加します。

```
Local string &guid3 = UuidGen();
Local string &value3 = "00003";
&returnValue = &xref.PopulateValue(&TestName, &EBS1, &value3, &UniqueGUID, &guid3, =>
&ADD);
```

この例は、*EBS1* エレメントの値 *&value4* を *&TestName* クロスリファレンス マップ データに追加し、コードによって自動的にランダムな *UniqueGUID* を作成します。

```
Local string &value4 = "00004";
&returnValue = &xref.PopulateValue(&TestName, &EBS1, &value4, &UniqueGUID, "", &AD =>
D);
```

PopulateValueNVP

構文

```
PopulateValueNVP (mapName, referenceDomain, DataElement referenceNVP, targetDomain,
targetNVP, mode)
```

説明

クロスリファレンス マップの参照ドメイン エレメント値を検索し、同じデータ行の別のドメインのエレメントに等値をロードします。この形式のロードは、参照ドメインまたはターゲットドメインに複数のエレメントが存在する場合に使用します。参照ドメインの全ての必須エレメントは、NVP 参照リストに含める必要がありますが、オプションのエレメント (識別子など) は含める必要はありません。

パラメータ

パラメータ	説明
mapName	動的 (クロスリファレンス) 値マップの定義の名前を文字列として指定します。
referenceDomain	値を検索する Xref 内のドメインの名前を文字列として指定します。
referenceNVP	検索対象の、参照ドメイン内の要素と値のペアを、DataElement の配列として指定します。
targetDomain	等値のロード先のドメインの名前を文字列として指定します。
targetNVP	Xref マップ内の、ロード対象の要素名とその等値 (NVP) を、DataElement の配列として指定します。
mode	この関数が要素をロードするモードを示します。ADD、LINK、または UPDATE のいずれかの値を指定できます。モードパラメータの値は、大文字と小文字が区別され、大文字だけで指定される必要があります。

戻り値

データがロードされた Xref 行の UniqueGUID 値。

例

この例は、&TestName クロスリファレンス マップの UniqueGUID エLEMENTの値 &guid1 を検索し、&EBS1 エLEMENTの等値を &value1 に更新します。

```
&value1 = &value1 | "0";
&ebsNVP1 [1].value = &value1;
&returnValue = &xref.PopulateValueNVP(&TestName, &UniqueGUID, &guidNVP1, &EBS, &ebsNVP1, &UPDATE);
```

この例は、UniqueGUID エLEMENTの値 &guid3 と EBS1 エLEMENTの値 &value3 を &TestName クロスリファレンス マップ データに追加します。

```
Local string &guid3 = UuidGen();
Local array of EOTF_CORE:Common:DataElement &guidNVP3 = CreateArrayRept(create EOTF_
CORE:Common:DataElement(&UniqueGUID), 1);
&guidNVP3 [1].value = &guid3;
Local string &value3 = "00003";
Local array of EOTF_CORE:Common:DataElement &ebsNVP3 = CreateArrayRept(create EOTF_
CORE:Common:DataElement(&EBS1), 1);
&ebsNVP3 [1].value = &value3;
&returnValue = &xref.PopulateValueNVP(&TestName, &EBS, &ebsNVP3, &UniqueGUID, &guidNVP3, &ADD);
```

この例は、*EBSI* エLEMENTの値 *&value4* を *&TestName* クロスリファレンス マップ データに追加し、コードによって自動的にランダムな UniqueGUID を作成します。

```
Local array of EOTF_CORE:Common:DataElement &guidNVP4 = CreateArrayRept(create EOTF_
CORE:Common:DataElement (&UniqueGUID), 1);
Local string &value4 = "00004";
Local array of EOTF_CORE:Common:DataElement &ebsNVP4 = CreateArrayRept(create EOTF_
CORE:Common:DataElement (&EBS1), 1);
&ebsNVP4 [1].value = &value4;
&returnValue = &xref.PopulateValueNVP (&TestName, &EBS, &ebsNVP4, &UniqueGUID, &guidNVP4, &ADD);
&requestValues = CreateArrayRept(create EOTF_CORE:Common:DataElement (&EBS1), 1);
&requestValues [1].value = &value4;
```

Xref クラス プロパティ

このセクションでは、Xref クラス プロパティについて説明します。

exceptionCaught

説明

`needAnException` パラメータが "偽" になっていたために、検索メソッドによって例外が抑制される場合は、値が "真" になります。

exceptionDetails

説明

発生した例外の詳細を含む例外オブジェクトです。

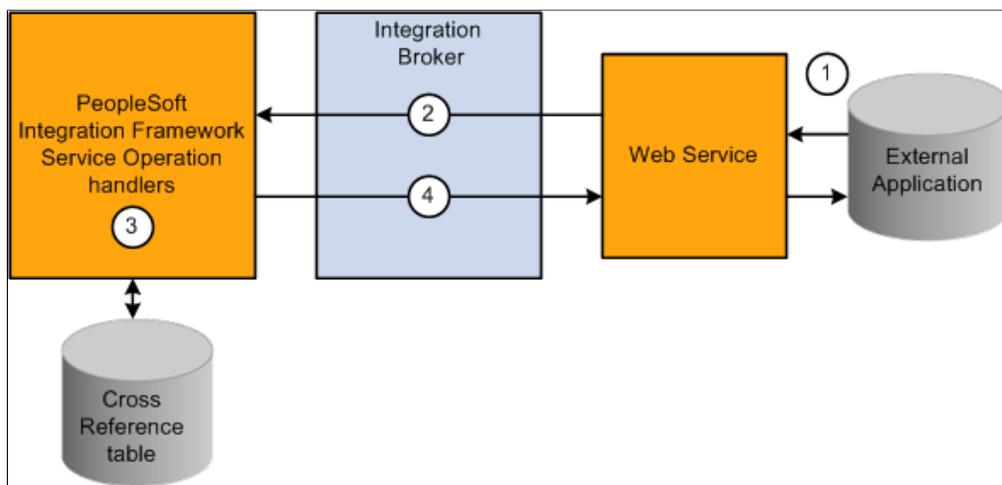
Web サービスを使用したマップへのアクセス

アプリケーション インテグレーション フレームワーク Web サービスについて

アプリケーション インテグレーション フレームワーク Web サービスは、Web サービスベースでアプリケーション インテグレーション フレームワークのマップ情報にアクセスする手段を外部アプリケーションに提供します。Web サービスは、PeopleTools インテグレーション ブローカー (IB) フレームワークによって実装されます。インテグレーション ゲートウェイ Web アプリケーションが、全ての Web サービス リクエストを受信し、インテグレーション エンジン (アプリケーション サーバー) により処理されるよう転送します。

画像: 外部アプリケーションとのアプリケーション インテグレーション フレームワーク Web サービスの使用

次の図は、外部アプリケーションがクロスリファレンス検索 Web サービスを使用して、アプリケーション インテグレーション フレームワークのクロスリファレンス マップ内の値を検索するようすを示しています。



1. 外部アプリケーションが、Web サービス オペレーションの 1 つを呼び出します。
2. PeopleSoft インテグレーション ブローカーが、サービス オペレーションを受取、WS セキュリティ資格証明を検証します。
3. リクエストが、処理のためにアプリケーション サーバーに渡されます。アプリケーション サーバーがサービス オペレーションを認証し、対応するハンドラに転送します。ハンドラが PeopleCode を実行し、応答をインテグレーション ゲートウェイに送信します。
4. インテグレーション ブローカーが、応答を外部アプリケーションに送信します。

EOTF_DVM サービス

このセクションでは、DVM の Web サービスとして使用できるサービス オペレーションについて説明します。

- EOTF_DVM_LOOKUP
- EOTF_DVM_IMPORT.v2
- EOTF_DVM_IMPORT.v1

EOTF_DVM_LOOKUP

このサービス オペレーションは、DVM の値の検索に使用します。

別名: dvmLookupValue

タイプ: 同期

リクエスト メッセージ: EOTF_DVM_LOOKUP_REQ.V1

リクエストの SOAP ボディの例を示します。

```
<soapenv:Body>
  <dvm:dvmLookupValue>
    <dvm:mapname>?</dvm:mapname>
    <!--2 or more repetitions:-->
    <dvm:Values>
      <dvm:domain>?</dvm:domain>
      <!--1 or more repetitions:-->
      <dvm:element name="?" />
    </dvm:Values>
    <dvm:Values>
      <dvm:domain>?</dvm:domain>
      <!--1 or more repetitions:-->
      <dvm:element name="?" />
    </dvm:Values>
    <dvm:targetDomain>?</dvm:targetDomain>
  </dvm:dvmLookupValue>
</soapenv:Body>
```

エレメント名	説明
mapname	値を検索するドメイン値マップの名前。
Values	このエレメントには、DVM 内で見つかったエレメントと値が保持されません。
domain	検索リクエストで指定されているターゲットドメインの名前。
element	DVM エレメントとその値を表します。
targetDomain	検索リクエストによって値が返されるドメインの名前。

レスポンス メッセージ: EOTF_DVM_RESPONSE.V1

エレメント名	説明
status	オペレーションのステータス。[成功] または [例外] を返します。
reason	例外が発生した場合、理由が返されます。
requestValues	このエレメントには、検索オペレーションに使用される参照値が保持されます。一部適格の検索の場合、これらの値はリクエストに指定されている値と異なる可能性があります。
domain	検索リクエストで指定されている参照ドメインの名前。
element	DVM エレメントとその値。
responseValues	検索オペレーションに返される値。
domain	検索リクエストで指定されているターゲットドメインの名前。
element	ターゲットドメインの DVM エレメントとその値。

EOTF_DVM_IMPORT.v2

このサービス オペレーションを使用して、PeopleSoft スキーマ ベースで、ドメイン値マップの定義と値をエンタープライズ変換フレームワークへインポートします。

タイプ: 非同期 - 一方向

別名: import_dvm

デフォルト: はい

リクエスト メッセージ: EOTF_DVM.V2

PeopleSoft スキーマを使用したインポート リクエストの SOAP ボディの例を示します。

```
<soapenv:Body>
  <dvm:dvm>
    <dvm:name>?</dvm:name>
    <!--Optional:-->
    <dvm:description>?</dvm:description>
    <!--Optional:-->
    <dvm:comments>?</dvm:comments>
    <dvm:elements>
      <!--2 or more repetitions:-->
      <dvm:element name="?" order="?" dataType="string" dataLength="?" isComm⇒
on="false"/>
      <dvm:element name="?" order="?" dataType="string" dataLength="?" isComm⇒
on="false"/>
    </dvm:elements>
    <dvm:domains>
      <!--1 or more repetitions:-->
```

```

    <dvm:domain name="?">
      <!--1 or more repetitions:-->
      <dvm:element name "?" qualifier="false"/>
    </dvm:domain>
  </dvm:domains>
  <!--Optional:-->
  <dvm:rows>
    <!--1 or more repetitions:-->
    <dvm:row>
      <!--2 or more repetitions:-->
      <dvm:cell?></dvm:cell>
      <dvm:cell?></dvm:cell>
    </dvm:row>
  </dvm:rows>
</dvm:dvm>
</soapenv:Body>

```

エレメント名	説明
description	DVM の説明。 オプション
comments	DVM についてのコメント。 オプション
elements	このエレメントには、DVM のエレメントリストが保持されます。
element	DVM エレメントを表します。
domains	このエレメントには、DVM のドメインリストが保持されます。
domain	このエレメントは、DVM 内のドメインを表します。
element	DVM ドメイン内のエレメントを表します。
rows	DVM 内の全てのデータ行を表します。
row	DVM の各値行。
cell	エレメントに定義されている順序と同じ順序で、各エレメントのこの行の値を示します。

EOTF_DVM_IMPORT.v1

このサービス オペレーションを使用して、FMW スキーマ ベースで、ドメイン値マップの定義と値をエンタープライズ変換フレームワークへインポートします。

タイプ: 非同期 - 一方向

別名: import_dvm

デフォルト: いいえ

リクエスト メッセージ: EOTF_DVM.V1

FMW スキーマを使用したインポートリクエストの SOAP ボディの例を示します。

```
<soapenv:Body>
  <dvm:dvm name="?">
    <!--Optional:-->
    <dvm:description?></dvm:description>
    <dvm:columns>
      <!--2 or more repetitions:-->
      <dvm:column name="" qualifier="false" order=""/>
      <dvm:column name="" qualifier="false" order=""/>
    </dvm:columns>
    <!--Optional:-->
    <dvm:rows>
      <!--1 or more repetitions:-->
      <dvm:row>
        <!--2 or more repetitions:-->
        <dvm:cell?></dvm:cell>
        <dvm:cell?></dvm:cell>
      </dvm:row>
    </dvm:rows>
  </dvm:dvm>
</soapenv:Body>
```

エレメント名	説明
description	DVM の説明。 オプション
columns	このエレメントには、DVM の列リストが保持されます。
column	DVM 列を表します。
rows	DVM 内の全てのデータ行を表します。
row	DVM の各値行。
cell	列に定義されている順序と同じ順序で、各列のこの行の値を示します。

EOTF_XREF サービス

このセクションでは、クロスリファレンスで使用できるサービス オペレーションについて説明します。

- EOTF_XREF_ADD
- EOTF_XREF_LINK
- EOTF_XREF_UPDATE

- EOTF_XREF_DELETE
- EOTF_XREF_LOOKUP
- EOTF_XREF_IMPORT.v2
- EOTF_XREF_IMPORT.v1

EOTF_XREF_ADD

このサービス オペレーションを使用して、クロスリファレンス マップに値を追加します。

別名: add_xref_value

タイプ: 同期

リクエスト メッセージ: EOTF_XREF_ADD_REQ.V1

リクエストの SOAP ボディの例を示します。

```
<soapenv:Body>
  <xref:xrefAddValue>
    <xref:mapname?></xref:mapname>
    <!--2 or more repetitions:-->
    <xref:Values>
      <xref:domain?></xref:domain>
      <!--1 or more repetitions:-->
      <xref:element name="?" />
    </xref:Values>
    <xref:Values>
      <xref:domain?></xref:domain>
      <!--1 or more repetitions:-->
      <xref:element name="?" />
    </xref:Values>
  </xref:xrefAddValue>
</soapenv:Body>
```

エレメント名	説明
mapname	値を追加するクロスリファレンス マップの名前。
Values	このエレメントには、ロード対象のクロスリファレンス エレメントと値が保持されます。
domain	エレメントが含まれているドメインの名前。
element	エレメントとその値を表します。

レスポンス メッセージ: EOTF_XREF_RESPONSE.V1

エレメント名	説明
status	オペレーションのステータス。[成功] または [例外] を返します。

エレメント名	説明
reason	例外が発生した場合、理由が返されます。
<i>responseValues</i>	検索オペレーションに返される値。
domain	検索リクエストで指定されているターゲットドメインの名前。
element	ターゲットドメインのエレメントとその値を表します。

EOTF_XREF_LINK

このサービス オペレーションを使用して、クロスリファレンス マップに値をリンクします。

別名: link_xref_value

タイプ: 同期

リクエスト メッセージ: EOTF_XREF_LINK_REQ.V1

リクエストの SOAP ボディの例を示します。

```
<soapenv:Body>
  <xref:xrefLinkValue>
    <xref:mapname>?</xref:mapname>
    <xref:ReferenceValues>
      <xref:domain>?</xref:domain>
      <!--1 or more repetitions:-->
      <xref:element name="?" />
    </xref:ReferenceValues>
    <!--1 or more repetitions:-->
    <xref:TargetValues>
      <xref:domain>?</xref:domain>
      <!--1 or more repetitions:-->
      <xref:element name="?" />
    </xref:TargetValues>
  </xref:xrefLinkValue>
</soapenv:Body>
```

エレメント名	説明
mapname	値を追加するクロスリファレンス マップの名前。
<i>ReferenceValues</i>	このエレメントには、ターゲット値にリンクする、クロスリファレンス内の既存のエレメントと値が保持されます。
domain	参照エレメントが含まれているドメインの名前。
element	クロスリファレンス エレメントとその値を表します。
<i>targetValues</i>	このエレメントには、追加対象のクロスリファレンス エレメントと値が保持されます。

エレメント名	説明
domain	ターゲット エレメントが含まれているドメインの名前。
element	クロスリファレンス エレメントとその値を表します。

レスポンス メッセージ: EOTF_XREF_RESPONSE.V1

エレメント名	説明
status	オペレーションのステータス。 <i>[成功]</i> または <i>[例外]</i> を返します。
reason	例外が発生した場合、理由が返されます。
responseValues	検索オペレーションに返される値。
domain	検索リクエストで指定されているターゲット ドメインの名前。
element	ターゲット ドメインのエレメントとその値を表します。

EOTF_XREF_UPDATE

このサービス オペレーションを使用して、クロスリファレンス マップの値を更新します。

別名: update_xref_value

タイプ: 同期

リクエスト メッセージ: EOTF_XREF_UPDATE_REQ.V1

リクエストの SOAP ボディの例を示します。

```
<soapenv:Body>
  <xref:xrefUpdateValue>
    <xref:mapname>?</xref:mapname>
    <xref:ReferenceValues>
      <xref:domain>?</xref:domain>
      <!--1 or more repetitions:-->
      <xref:element name="?" />
    </xref:ReferenceValues>
    <!--1 or more repetitions:-->
    <xref:TargetValues>
      <xref:domain>?</xref:domain>
      <!--1 or more repetitions:-->
      <xref:element name="?" />
    </xref:TargetValues>
  </xref:xrefUpdateValue>
</soapenv:Body>
```

エレメント名	説明
mapname	値を更新するクロスリファレンス マップの名前。

エレメント名	説明
<i>ReferenceValues</i>	このエレメントには、更新する行の検索に使用する、既存のクロスリファレンス エレメントと値が保持されます。
domain	参照エレメントが含まれているドメインの名前。
element	クロスリファレンス エレメントとその値を表します。
<i>targetValues</i>	このエレメントには、更新対象のクロスリファレンス エレメントと新しい値が保持されます。
domain	ターゲット エレメントが含まれているドメインの名前。
element	クロスリファレンス エレメントとその値を表します。

レスポンス メッセージ: EOTF_XREF_RESPONSE.V1

エレメント名	説明
status	オペレーションのステータス。 <i>[成功]</i> または <i>[例外]</i> を返します。
reason	例外が発生した場合、理由が返されます。
<i>responseValues</i>	検索オペレーションに返される値。
domain	検索リクエストで指定されているターゲットドメインの名前。
element	ターゲットドメインのエレメントとその値を表します。

EOTF_XREF_DELETE

このサービス オペレーションを使用して、クロスリファレンス マップから値を削除します。

別名: delete_xref_value

タイプ: 同期

リクエスト メッセージ: EOTF_XREF_DELETE_REQ.V1

リクエストの SOAP ボディの例を示します。

```
<soapenv:Body>
  <xref:xrefDeleteValue>
    <xref:mapname>?</xref:mapname>
    <!--2 or more repetitions:-->
    <xref:Values>
      <xref:domain>?</xref:domain>
      <!--1 or more repetitions:-->
      <xref:element name="?" />
    
```

```

    </xref:Values>
    <xref:Values>
      <xref:domain>?</xref:domain>
      <!--1 or more repetitions:-->
      <xref:element name="?" />
    </xref:Values>
  </xref:xrefDeleteValue>
</soapenv:Body>

```

エレメント名	説明
mapname	値を削除するクロスリファレンス マップの名前。
Values	このエレメントには、削除対象のクロスリファレンス エレメントと値が保持されます。
domain	エレメントが含まれているドメインの名前。
element	エレメントとその値を表します。

レスポンス メッセージ: EOTF_XREF_RESPONSE.V1

エレメント名	説明
status	オペレーションのステータス。[成功] または [例外] を返します。
reason	例外が発生した場合、理由が返されます。
responseValues	応答値は返されません。

EOTF_XREF_LOOKUP

このサービス オペレーションを使用して、クロスリファレンス マップから値を検索します。

別名: lookup_xref_value

タイプ: 同期

EOTF_XREF_LOOKUP_REQ.V1

リクエストの SOAP ボディの例を示します。

```

<soapenv:Body>
  <xref:xrefLookupValue needFault="true">
    <xref:mapname>?</xref:mapname>
    <xref:Values>
      <xref:domain>?</xref:domain>
      <!--1 or more repetitions:-->
      <xref:element name="?">?</xref:element>
    </xref:Values>
    <xref:targetDomain>?</xref:targetDomain>
  </xref:xrefLookupValue>
</soapenv:Body>

```

エレメント名	説明
mapname	値を検索するクロスリファレンス マップの名前。
<i>Values</i>	このエレメントには、検索対象のクロスリファレンス エレメントと値が保持されます。
domain	参照エレメントが含まれているドメインの名前。
element	クロスリファレンス エレメントとその値を表します。
targetDomain	検索リクエストによってエレメントおよび値が返されるドメインの名前。

レスポンス メッセージ: EOTF_XREF_RESPONSE.V1

エレメント名	説明
status	オペレーションのステータス。[成功] または [例外] を返します。
reason	例外が発生した場合、理由が返されます。
<i>responseValues</i>	検索オペレーションに返される値。
domain	検索リクエストで指定されているターゲット ドメインの名前。
element	ターゲット ドメインのエレメントとその値を表します。

EOTF_XREF_IMPORT.v2

このサービス オペレーションを使用して、クロスリファレンス マップの定義と値を PeopleSoft スキーマを使ってインポートします。

別名: xref

タイプ: 非同期 - 一方向

デフォルト: はい

リクエスト メッセージ: EOTF_XREF.V2

リクエストの SOAP ボディの例を示します。

```
<soapenv:Body>
  <xref:xref>
    <xref:name>?</xref:name>
    <!--Optional:-->
    <xref:description>?</xref:description>
    <!--Optional:-->
  </xref:xref>
</soapenv:Body>
```

```

    <xref:comments>?</xref:comments>
    <xref:elements>
      <!--2 or more repetitions:-->
      <xref:element name="?" order="?" dataType="string" dataLength="?" isCom⇒
mon="false"/>
      <xref:element name="?" order="?" dataType="string" dataLength="?" isCom⇒
mon="false"/>
    </xref:elements>
    <xref:domains>
      <!--1 or more repetitions:-->
      <xref:domain name="?">
        <!--1 or more repetitions:-->
        <xref:element name="?"/>
      </xref:domain>
    </xref:domains>
    <!--Optional:-->
    <xref:rows>
      <!--1 or more repetitions:-->
      <xref:row>
        <!--2 or more repetitions:-->
        <xref:cell>?</xref:cell>
        <xref:cell>?</xref:cell>
      </xref:row>
    </xref:rows>
  </xref:xref>
</soapenv:Body>

```

エレメント名	説明
description	クロスリファレンス マップの説明。 オプション
comments	クロスリファレンスについての説明。 オプション
elements	このエレメントには、クロスリファレンスのエレメントリストが保持されま す。
element	クロスリファレンス エレメントを表します。
domains	このエレメントには、クロスリファレンスのドメインリストが保持されます。
domain	このエレメントは、クロスリファレンス マップ内のドメインを表し ます。
element	クロスリファレンスドメイン内のエレメントを表します。
rows	クロスリファレンスマップ内の全てのデータ行を表します。
row	クロスリファレンスの各値行。
cell	エレメントに定義されている順序と同じ順序で、各エレメント のこの行の値を示します。

EOTF_XREF_IMPORT.v1

このサービス オペレーションを使用して、クロスリファレンス マップの定義と値を PeopleSoft スキーマを使ってインポートします。

別名: xref

タイプ: 非同期 - 一方向

デフォルト: いいえ

変換: v2 へ

リクエスト メッセージ: EOTF_XREF.V1

リクエストの SOAP ボディの例を示します。

```
<soapenv:Body>
  <xref:xref>
    <xref:table name="?">
      <!--Optional:-->
      <xref:description?></xref:description>
      <!--Optional:-->
      <xref:columns>
        <!--1 or more repetitions:-->
        <xref:column name="?">
        </xref:column>
      </xref:columns>
      <!--Optional:-->
      <xref:rows>
        <!--1 or more repetitions:-->
        <xref:row>
          <!--1 or more repetitions:-->
          <xref:cell colName="?">
          </xref:cell>
        </xref:row>
      </xref:rows>
    </xref:table>
  </xref:xref>
</soapenv:Body>
```

エレメント名	説明
table	このエレメントには、テーブル エレメントが保持されます。
description	DVM の説明。 オプション
columns	このエレメントには、DVM の列リストが保持されます。
column	DVM 列を表します。
rows	DVM 内の全てのデータ行を表します。
row	DVM の各値行。

エレメント名	説明
cell	列に定義されている順序と同じ順序で、各列のこの行の値を示します。

クロスリファレンス検索 Web サービスの例

サービス オペレーション EOTF_XREF_LOOKUP.v1 は、外部システムがクロスリファレンス値の検索に使用します。外部システムが Web サービスにアクセスできるようにするには、PeopleSoft アプリケーションで Web サービスがパブリッシュされる必要があります。

『PeopleTools: PeopleSoft Integration Broker』の「Providing Services」を参照してください。

例: クロスリファレンス マップの定義

この例では、サードパーティシステムが、支払条件のクロスリファレンス検索をリクエストします。このページでは、PaymentTermsGUID 値マップに含まれるエレメントを説明します。

画像: 例で使用されているドメイン値マップ

次の例では、例で使用されているドメイン値マップのフィールドおよびコントロールを説明します。このページのフィールドおよびコントロールの定義は後で説明します。

Map Name: PaymentTermGUID Type: Cross-reference (dynamic)

*Description: Payment Terms Cross-Reference

Comments:

Delete

Order	*Element Name	*Data Type	Length	Required
1	UniqueGUID	String	36	<input checked="" type="checkbox"/>
2	SETID	String	5	<input checked="" type="checkbox"/>
3	PYMNT_TERMS_CD	String	5	<input checked="" type="checkbox"/>

UniqueGUID、SETID、および PYMNT_TERMS_CD の 3 つのエレメントが定義されています。

このページでは、ドメイン値マップに定義されているドメインを説明します。

画像: この例で使用されている "値マップ - ドメイン" ページ

次の例では、この例で使用されている "値マップ - ドメイン" ページのフィールドおよびコントロールを説明します。このページのフィールドおよびコントロールの定義は後で説明します。

The screenshot shows a web interface for configuring domains for a value map. The map name is 'PaymentTermGUID' and its type is 'Cross-reference (dynamic)'. There are two domain entries:

- Domain Name: UniqueGUID** (checked 'Is Unique'). Below it, a table 'Assign Elements to the Domain' shows one element: 'UniqueGUID'.
- *Domain Name: PSFT** (checked 'Is Unique'). Below it, a table 'Assign Elements to the Domain' shows two elements: 'SETID' and 'PYMNT_TERMS_CD', each with a dropdown menu and '+'/'-' buttons.

このマップには、次の2つのドメインが定義されています。

1. *UniqueGUID*. サードパーティによって使用されるドメインで、UniqueGUID エlementが設定されています。
2. *PSFT*. PeopleSoft アプリケーションを表し、SETID および PYMNT_TERMS_CD Elementが設定されています。

例: Web サービスのリクエストおよび応答

これは、共通 GUID 値に対応する PeopleSoft 値を取得するサンプル リクエストです。

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xref="http://xmlns.oracle.com/Enterprise/tools/schema/xrefLookupValue.v1">
  <soapenv:Header/>
  <soapenv:Body>
    <xref:xrefLookupValue needFault="true">
      <xref:mapname>PaymentTermGUID</xref:mapname>
      <xref:Values>
        <xref:domain>UniqueGUID</xref:domain>
        <!--1 or more repetitions:-->
        <xref:element name="UniqueGUID">9d266732-90e2-11dd-a062-96c8921a7858</xref:element>
      </xref:Values>
      <xref:targetDomain>PSFT</xref:targetDomain>
    </xref:xrefLookupValue>
  </soapenv:Body>
</soapenv:Envelope>
```

これはその応答です。

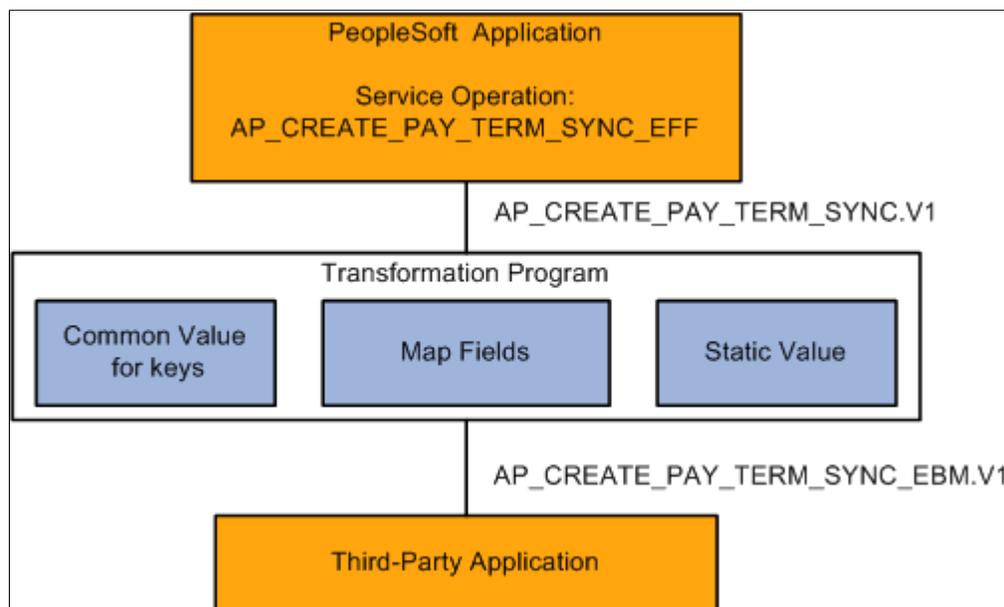
```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:s=>
oapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:xsd="http://www.w3.org/200=>
1/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <xrefResponse xmlns="http://xmlns.oracle.com/Enterprise/tools/schema/xrefResp=>
onse.v1">
      <status>Success</status>
      <responseValues>
        <domain>PSFT</domain>
        <element name="SETID">SHARE</element>
        <element name="PYMNT_TERMS_CD">CD01</element>
      </responseValues>
    </xrefResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

アプリケーション インテグレーション フレームワーク の例

例の概要

画像: 支払条件のインテグレーション フロー

次の図では、このアプリケーションのシナリオを説明します。



この例では、新しい支払条件が追加されるたびに、PeopleSoft アプリケーションが、サードパーティ アプリケーションを更新します。この変換では、PeopleSoft ABM エlementと EBM Elementのマッピングと、キーと静的値のマッピングが必要です。この例では、アプリケーション インテグレーション フレームワーク内の次のデータ変換を説明します。

- SETID フィールドと PYMNT_TERMS_CD フィールドの組合せが、PeopleSoft アプリケーションのキーとして使用されます。この値に、EBM の共通 GUID を割り当てる必要があります。
- PeopleSoft アプリケーションは、3 文字の言語コードを使用します。サードパーティ システムは、数値コードを使用します。

このインテグレーションには、次の処理を実行する変換が必要です。

- メッセージ構造のマッピング
- キー フィールドの共通値の作成

- タイプの異なる静的フィールドのデータ値の変換

動的値マップの定義

新しい動的値マップを作成するには、次の手順に従います。

1. [エンタープライズ コンポーネント], [インテグレーション定義], [変換フレームワーク], [値マップの定義] を選択します。
2. 新規値の追加ページを選択します。
3. マップ名として [PaymentTermsGUID] を入力します。
4. マップタイプとして [相互参照 (動的)] を選択します。
5. [追加] をクリックします。

マップにはエレメントの UniqueGUID が設定されます。

6. さらに 2 つのエレメント、SETID および PYMNT_TERMS_CD を追加します。

画像: 例で使用されているドメイン値マップ

次の例では、例で使用されているドメイン値マップのフィールドおよびコントロールを説明します。このページのフィールドおよびコントロールの定義は後で説明します。

The screenshot shows a web interface for defining a domain value map. The 'Map Name' is 'PaymentTermGUID' and the 'Type' is 'Cross-reference (dynamic)'. The description is 'Payment Terms Cross-Reference'. Below this is a table titled 'Assign Elements to Maps' with the following data:

Order	*Element Name	*Data Type	Length	Required		
1	UniqueGUID	String	36	<input checked="" type="checkbox"/>	+	-
2	SETID	String	5	<input checked="" type="checkbox"/>	+	-
3	PYMNT_TERMS_CD	String	5	<input checked="" type="checkbox"/>	+	-

UniqueGUID、SETID、および PYMNT_TERMS_CD の 3 つのエレメントが定義されています。

ドメインを追加するには、次の手順に従います。

1. ドメイン ページを選択します。
UniqueGUID ドメインが表示されます。
2. [新規行の追加] アイコンをクリックして、ドメインを作成します。

3. ドメイン名として *[PSFT]* を入力します。
4. 1つ目のエレメントとして *[SETID]* を選択します。
5. [新規行の追加] アイコンをクリックして、行を追加します。
6. 2つ目のエレメントとして *[PYMNT_TERMS_CD]* を選択します。
7. ページを保存します。

画像: この例で使用されている "値マップ - ドメイン" ページ

次の例では、この例で使用されている "値マップ - ドメイン" ページのフィールドおよびコントロールを説明します。このページのフィールドおよびコントロールの定義は後で説明します。

The screenshot shows the 'Assign Domains to Value Maps' page. At the top, it displays 'Map Name: PaymentTermGUID' and 'Type: Cross-reference (dynamic)'. Below this, there are two sections for domain configuration. The first section is for 'Domain Name: UniqueGUID' with a checked 'Is Unique' box. Below it, a table 'Assign Elements to the Domain' shows one element: '1 UniqueGUID'. The second section is for '*Domain Name: PSFT' with a checked 'Is Unique' box. Below it, a table 'Assign Elements to the Domain' shows two elements: '1 SETID' and '2 PYMNT_TERMS_CD'. Each element has a dropdown arrow and '+'/'-' buttons.

このマップには、次の2つのドメインが定義されています。

- *UniqueGUID*。サードパーティによって使用されるドメインで、UniqueGUID エレメントが設定されています。
- *PSFT*。PeopleSoft アプリケーションを表し、SETID および PYMNT_TERMS_CD エレメントが設定されています。

静的値マップの定義とロード

新しいドメイン値マップを作成するには、次の手順に従います。

1. [エンタープライズ コンポーネント], [インテグレーション定義], [変換フレームワーク], [値マップの定義] を選択します。
2. 新規値の追加ページを選択します。

3. マップ名として *[LanguageCodeDVM]* を入力します。
4. マップ タイプとして *[ドメイン値マップ (静的)]* を選択します。
5. **[追加]** をクリックします。

マップにはエレメントの UniqueGUID が設定されます。

6. さらに 2 つのエレメント、LANGUAGE_CD および COMMON を追加します。

画像: 言語コードのドメイン値マップ

次の例では、言語コードのドメイン値マップのフィールドおよびコントロールを説明します。このページのフィールドおよびコントロールの定義は後で説明します。

Order	*Element Name	*Data Type	Length	Required		
1	UniqueGUID	String	36	<input checked="" type="checkbox"/>	+	-
2	LANGUAGE_CD	String	3	<input checked="" type="checkbox"/>	+	-
3	COMMON	String	32	<input checked="" type="checkbox"/>	+	-

ドメインを追加するには、次の手順に従います。

1. ドメイン ページを選択します。
UniqueGUID ドメインが表示されます。
2. **[新規行の追加]** アイコンをクリックして、ドメインを作成します。
3. ドメイン名として *[PSFT]* を入力します。
4. 1 つ目のエレメントとして *[LANGUAGE_CD]* を選択します。
5. ヘッダーの **[新規行の追加]** アイコンをクリックして、ドメインを追加します。
6. ドメイン名として *[AIA]* を入力します。
7. 1 つ目のエレメントとして *[COMMON]* を選択します。

8. ページを保存します。

画像: 言語コードのドメイン値マップドメイン

次の例では、言語コードのドメイン値マップドメインのフィールドおよびコントロールを説明します。このページのフィールドおよびコントロールの定義は後で説明します。

The screenshot displays the configuration for a Domain Value Map (DVM) named 'LanguageCodeDVM'. The interface is divided into three sections, each representing a different domain. Each section includes a 'Domain Name' field, a checked 'Is Unique' checkbox, and a table for assigning elements to that domain.

Assign Domains to Value Maps	
Domain Name: UniqueGUID	<input checked="" type="checkbox"/> Is Unique
Assign Elements to the Domain	
1 UniqueGUID	
*Domain Name: PSFT <input checked="" type="checkbox"/> Is Unique	
Assign Elements to the Domain	
1 LANGUAGE_CD	
*Domain Name: AIA <input checked="" type="checkbox"/> Is Unique	
Assign Elements to the Domain	
1 COMMON	

AIA、PSFT、および UniqueGUID の 3 つのドメインが定義されています。

DVM をロードするには、次の手順に従います。

1. [エンタープライズ コンポーネント], [インテグレーション定義], [変換フレームワーク], [ドメイン値マップのロード] を選択します。
2. [LanguageCodeDVM] を選択します。
3. LANGUAGE_CD 列に、PeopleSoft (PSFT) ドメインで定義されているデータ値を入力します。
4. 対応する COMMON 列に、AIA ドメインに使用する値を入力します。
5. 全ての静的値のマッピングに必要な数の行を追加します。

6. ページを保存します。

画像: ドメイン値マップ ページ

次の例では、ドメイン値マップ ページのフィールドおよびコントロールを説明します。このページのフィールドおよびコントロールの定義は後で説明します。

Domain Value Map	
Map Name: LanguageCodeDVM Description: Language Code DVM	
Assign Data to Value Maps Customize Find View All First 1 of 1 Last	
LANGUAGE_CD	COMMON
1 ENG	500

変換プログラムでの XSLT 拡張関数の使用

PeopleSoft ABM メッセージのフィールドを EBM メッセージ内の対応するフィールドにマッピングする変換プログラムを作成します。その後、XSLT 関数を使用して、データ値をマッピングします。

このセクションでは、次の機能に関する変換アプリケーション エンジン プログラムのセクションのコーディング例を紹介します。

- キー値の変換
- ドメイン値の変換

キー値の変換

次の例では、変換する必要がある ABM の XML のエレメント、変換のコード、および変換後の EBM 内のエレメントを説明します。

PeopleSoft ABM 内の変換対象のキー エレメント

XML メッセージに含まれる要素は、次のとおりです。

```
<MsgData>
  <Transaction>
    <PYMT_TRMS_HDR class="R">
      <SETID IsChanged="Y">SHARE</SETID>
      <PYMNT_TERMS_CD IsChanged="Y">DIT91</PYMNT_TERMS_CD>
```

XSLT コード

以下の XSLT コードのスニペットは、アプリケーション エンジン プログラムの変換 XSLT を示しています。

1. SETID と PYMT_TERMS_CD を連結して、新しい変数を作成します。

```
<corecom:ApplicationObjectKey>
  <corecom:ID>
    <xsl:attribute name="schemeID">
      <xsl:text disable-output-escaping="no">PSFT</xsl:text>
    </xsl:attribute>
```

```

<xsl:attribute name="schemeAgencyID">
  <xsl:text disable-output-escaping="no">PSFT_COMMON</xsl:text>
</xsl:attribute>
  <xsl:variable name="NamedValuePair" select='concat("&lt;SETID&gt;", PYMT_T⇒
RMS_HDR/SETID, "&lt;/SETID&gt;&lt;PYMNT_TERMS_CD&gt;", PYMT_TRMS_HDR/PYMNT_TERMS⇒
_CD, "&lt;/PYMNT_TERMS_CD&gt;")' />

```

2. generate-guid ユーティリティを使用して、一意の GUID を作成します。

```

<xsl:variable name="CommonGuid" select='concat("&lt;UniqueGUID&gt;", util⇒
ity:generate-guid(), "&lt;/UniqueGUID&gt;")' />

```

注: このステップは必須ではありません。GUID が存在しない場合、適切な xref:populate 関数によつて自動的に一意の GUID が作成されます。

3. xref:populateValueNVP 関数を使用して、新しい GUID をクロスリファレンスに追加します。

```

<xsl:variable name="GUIDAdd" select='xref:populateValueNVP("PaymentTermGUID", "⇒
PSFT", $NamedValuePair, "UniqueGUID", $CommonGuid, "ADD")' />
  <xsl:call-template name="Process-GUID">
    <xsl:with-param name="returnValue">
      <xsl:value-of select="$GUIDAdd"/>
    </xsl:with-param>
    <xsl:with-param name="statusDelimiter"></xsl:with-param>
  </xsl:call-template>
</corecom:ID>

```

4. EBM メッセージに値を挿入します。

```

<corecom:ContextID>
  <xsl:attribute name="schemeID">
    <xsl:text disable-output-escaping="no">SETID</xsl:text>
  </xsl:attribute>
  <xsl:attribute name="schemeAgencyID">
    <xsl:text disable-output-escaping="no">PSFT</xsl:text>
  </xsl:attribute>
  <xsl:value-of select="PYMT_TRMS_HDR/SETID"/>
</corecom:ContextID>
<corecom:ContextID>
  <xsl:attribute name="schemeID">
    <xsl:text disable-output-escaping="no">Payment Terms Code</xsl:text>
  </xsl:attribute>
  <xsl:attribute name="schemeAgencyID">
    <xsl:text disable-output-escaping="no">PSFT</xsl:text>
  </xsl:attribute>
  <xsl:value-of select="PYMT_TRMS_HDR/PYMNT_TERMS_CD"/>
</corecom:ContextID>

```

EBM メッセージに含まれる変換後のエレメント

変換後の EBM メッセージには、共通の GUID 値が含まれます。

```

<corecom:ApplicationObjectKey>
  <corecom:ID schemeAgencyID="PSFT_COMMON" schemeID="PSFT">b15f3c34-72bc-11d⇒
d-b7dd-aaf7c4308a71</corecom:ID>
  <corecom:ContextID schemeAgencyID="PSFT" schemeID="SETID">SHARE</corecom:⇒
ContextID>
  <corecom:ContextID schemeAgencyID="PSFT" schemeID="Payment Terms Code">DI⇒
T91</corecom:ContextID>

```

ドメイン値の変換

次の例では、変換する必要がある ABM の XML のドメイン値エレメント、変換のコード、および変換後の EBM 内のエレメントを説明します。

PeopleSoft ABM 内の変換対象のドメイン値エレメント

これは、PeopleSoft ABM メッセージに含まれる言語を指定するエレメントです。

```
<<PSCAMA class="R">
  <LANGUAGE_CD>ENG</LANGUAGE_CD>
```

XSLT コード

以下の XSLT コードのスニペットは、変換アプリケーション エンジン プログラムの DVM 検索を示しています。

```
<xsl:variable name="MsgLang" select='dvm:lookup-dvm ("LanguageCodeDVM", "LANGUAGE_CD⇒
", //MsgData/Transaction/PSCAMA/LANGUAGE_CD, "COMMON", //MsgData/Transaction/PSCAMA/LA⇒
NGUAGE_CD, false())' />
<xsl:variable name="BaseLang" select='dvm:lookup-dvm ("LanguageCodeDVM", "LANGUAGE_CD⇒
", //MsgData/Transaction/PSCAMA/BASE_LANGUAGE_CD, "COMMON", //MsgData/Transaction/PSCA⇒
MA/BASE_LANGUAGE_CD, false())' />
<xsl:attribute name="languageCode">
  <xsl:value-of select="substring-after ($MsgLang, ';')"/>
</xsl:attribute>
```

EBM メッセージに含まれる変換後のエレメント

これは結果の EBM メッセージに含まれる変換後のエレメントです。

```
<?xml version="1.0"?>
<paytermcreate:CreatePaymentTermEBM languageCode="500" xmlns:corepaymenttermcust="h⇒
ttp://xmlns.oracle.com/EnterpriseObjects/Core/Custom/EBO/PaymentTerm/V1" xmlns:payt⇒
ermcreate="http://xmlns.oracle.com/EnterpriseObjects/Core/EBO/CreatePaymentTermEBM/⇒
V1">
  <corecom:EBMHeader languageCode="500" xmlns:corecom="http://xmlns.oracle.com/Ente⇒
priseObjects/Core/Common/V2">
    <corecom:Sender>
      <corecom:ID>E900B20</corecom:ID>
    </corecom:Sender>
  </corecom:EBMHeader>
```

サービス オペレーションのルーティングの更新

サービス オペレーションのルーティングを更新するには、次の手順に従います。

1. [PeopleTools], [インテグレーション ブローカー], [インテグレーション設定], [サービス オペレーション]を選択します。
2. 更新が必要なサービス オペレーションを選択します。
3. ルーティング ページを選択します。

4. 新しいルーティングを入力するか、既存のルーティングのリンクをクリックします。
5. 新しいルーティングの場合は、ルーティング情報を入力します。
6. パラメータ ページにアクセスします。

変換前メッセージ バージョン

これは、PeopleSoft ABM メッセージです。

変換プログラム 1

これは、このインテグレーション用に作成された変換プログラムです。

変換後メッセージ バージョン

これは、サードパーティに送信される EBM メッセージです。

7. ルーティングを保存します。

画像: サービス オペレーション ルーティング - パラメータ ページ

次の例では、「サービス オペレーション - ルーティング パラメータ」ページのフィールドおよびコントロールを説明します。このページのフィールドおよびコントロールの定義は後で説明します。

Routing Definitions	Parameters	Routing Properties
Routing Name:	SCM_CONTACT1_TO_CONTACT_RB1	
Service Operation:	CONTACT_SYNC_EFF	
Service Operation Version:	VERSION_RB_1	
Sender Node:	PSFT_EP	
Receiver Node:	CR910TS1	
Parameters		
Type:	Inbound Request	
External Alias:	<input type="text" value="CONTACT_SYNC_EFF.VERSION_1"/>	
	Alias References	
Message.Ver into Transform 1:	<input type="text" value="CONTACT_SYNC_EFF.VERSION_1"/> 	
Transform Program 1:	<input type="text" value="RB_CONT_IN"/> 	
Transform Program 2:	<input type="text"/> 	
Message.Ver out of Transforms:	<input type="text" value="CONTACT_SYNC_EFF.VERSION_RB_1"/> 	
<input type="button" value="Save"/> <input type="button" value="Return"/>		

