

平成 18 年度 経済産業省委託事業

流通システム標準化事業

ナショナル・レジストリ システム設計書

目次

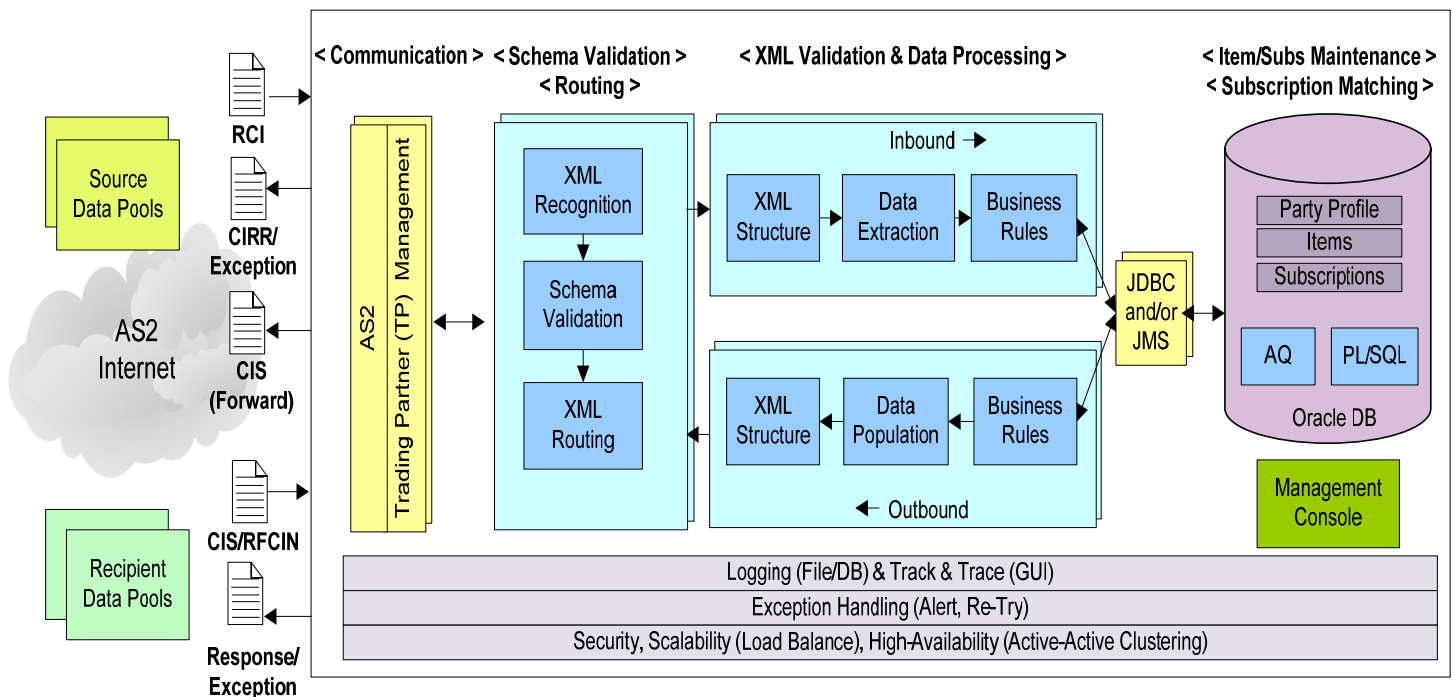
1. アーキテクチャ、デザインに関する考察	3
1.1 信頼できる AS2 企業間通信	4
1.2 ユーザー企業 (TRADING PARTNER) 管理	4
1.3 XML メッセージのバリデーション、転送、処理	4
1.4 オラクルDBとの連携	5
1.5 アイテム登録/サブスクリプションの管理、マッチングエンジン	5
1.6 ログの管理及びメッセージトラッキング	6
1.7 国際化サポート	6
1.8 高可用性(HIGH-AVAILABILITY)とスケーラビリティ	6
1.9 信頼性 (RELIABILITY)	8
1.10 セキュリティ (SECURITY)	8
2 機能要件および主なタスク	9
2.1 EANUCC XML の認識、ルーティング	9
2.2 EANUCC XML バリデーション	10
2.3 共通変換、ユーティリティサービス	14
2.4 共通エクセプション処理	14
2.5 RCI の処理	14
2.6 CIS の処理	15
2.7 RFCIN の処理	15
2.8 レスポンス、エクセプション処理	16
2.9 CIS / RFCIN の送信処理	17
2.10 RCI、CIS、RFCIN のマッチングエンジン	19
2.11 パーティプロフィールの管理	19
3 プラットフォーム、運用要件、デザイン	20
3.1 WEBMETHODS 6.5 の検証および設定	20
3.2 プログラミングテンプレート	20
3.3 ファイル処理、ネーミングコンベンション	21
3.4 AS2 設定、改善	21
3.5 ORACLE (10g) 検証—DB アーキテクト	21
4 環境セットアップ	22
5 別紙 A: 物理アーキテクチャ	23
7 別紙 B: Xerces スキーマバリデーション	24
8 別紙 C: JNR 設定ファイル	25
9 別紙 D: RCI の受信と処理	27
10 別紙 E: CIS の受信と処理	29
11 別紙 F: RFCIN の受信と処理	31
12 別紙 G: データベースプロセス図およびテーブルレイアウト	33
12.1 オラクルデータベース	33
12.2 データベースプロセス図	34
12.3 データベーステーブルおよびレイアウト	35
12.4 アドバンスキュー	49
12.5 HEARTBEAT	50
12.6 クリーンアップ	50
13 別紙 H: Node Iterator を使った XML 処理	51
14 別紙 I: JNR webMethods パッケージ一覧	52
14 別紙 J: 技術参照ドキュメントの入手先	53

1. アーキテクチャ、デザインに関する考察

JNR のコア機能要件は、GS1 グローバルレジストリで使用する XML メッセージ(RCI, CIS, RFCIN, GDSN レスポンス)のバリデーションおよびハンドリングであり、アイテム、サブスクリプションの保持、サブスクリプションのマッチング、マッチしたサブスクリプションの SDP への転送を含む。JNR アーキテクチャの基本構想は、webMethods Integration Suite を使って通信、XML メッセージの処理を行い、オラクルデータベースを利用してアイテム、サブスクリプションの保存や大量のサブスクリプションのマッチングを可能にするものである。

アーキテクチャの基本構想は、以下の原則に従うものとする。

1. 信頼できる、検証済みのロバストな技術的ツール、フレームワークを使用すること
2. 開発時間を短縮するため、既製品の機能を最大限活用すること
3. 各コンポーネントの互いの依存を最小限にし、平行して機能し、それらの統合が容易であること



< Japanese National Registry >

上記の図は JNR の主な機能的、技術的コンポーネントである通信、スキーマバリデーション、データ処理、転送、XML バリデーション、データ処理、サブスクリプションマッチングを表している。これらのコンポーネントは互いの依存度が低く、平行して機能し、互いに交信する。メッセージを受信、保存した後、技術的問題が起きても、各レベルでメッセージの再処理が可能なアーキテクチャであるため、エンドユーザーへの影響を最小限に抑えることができる。また、各コンポーネントの拡張、取替えが容易に行える。例えば、スキーマバリデーションエンジンは、他のサードパーティのものに簡単に入れ替えることができる。もし、スキーマバリデーションが負担のかかる作業であれば、より高速の CPU を搭載したサーバーに移すこともできる。サブスクリプションのマッチングコンポーネントは、通信、バリデーション、データマッピング処理用のコンポーネントに影響を与えることなく交換できる。下記のセクションはそれぞれのコンポーネントについての詳細説明である。

1.1 信頼できる AS2 企業間通信

AS2 はインターネット経由で重要なメッセージを送受信する際に用いられる、安全かつ信頼性の高い業界の標準プロトコルである。JNR は webMethods の AS2 アダプタを使ってデータプールと EANUCC XML メッセージの送受信を行う。webMethods の AS2 アダプタは Drummond Group の認証を受けており、その信頼性はエージェントリクスのさまざまなソリューションの本番環境で使われたことで検証済みである。同アダプタは、さらに自動再送機能を有しており、エンドユーザー間の確実なデータ送信を保障できる。

1.2 ユーザー企業 (TRADING PARTNER) 管理

企業名、GLN、連絡先、AS2 接続情報、電子証明書などのデータプールのプロフィール（企業情報）は、webMethods Trading Networks (TN) バージョン 6.5 を使って管理される。webMethods TN は使いやすい GUI が用意されており、ユーザー企業の管理を効率的に行うことが可能である。

JNR はデータプールとしかデータ通信を行わないため、小売や卸などのプロフィールを webMethods TN に格納される必要はない。また、これらの情報は webMethods への依存を減らし、webMethods API をバイパスすることで検索の速度を早くする意味でも、独立した DB テーブル（パーティ情報）を使って、GLN や所属 DP などの小売、卸、メーカー情報を管理した方が望ましい。

1.3 XML メッセージのバリデーション、転送、処理

EANUCC XML メッセージの定義、識別、転送は、webMethods TN の標準機能で対応する。まず、各 XML メッセージのドキュメントタイプを、ルートタグ、ネームスペース、メッセージタイプタグで定義し、いくつかのカスタム属性については、TN の中で定義し、XPATH クエリーを利用して抽出することができる。たとえば、メッセージ ID は、下記の XPATH クエリーを利用して抽出することができる：

```
/sh:StandardBusinessDocument/sh:StandardBusinessDocumentHeader/sh:DocumentIdentification/sh:InstanceIdentifier
```

ドキュメントタイプを定義した後は、ドキュメントごとのプロセッシングルールを作ることによって、XML メッセージを処理する。プロセッシングルールは、送信者、受信者、ドキュメントタイプ、ステータス、エラー状況などによって実行される。JNR は登録されたデータプールからの、バリデーション検証済みの XML メッセージのみ処理する。定義されていないドキュメントタイプ、未登録のデータプールからのメッセージ、メッセージ ID の重複などはここで検知される。また、これらのルールは、スキーマバリデーションを行うためのサービスを呼び出すためにも使われ、webMethods の標準バリデーションエンジン、またはサードパーティの Java ベースのバリデーションエンジンを使うことも可能である。EANUCC XML の著者は GS1 であるが、GS1 は EANUCC 2.1 スキーマバリデーションのために Xerces と XML Schema Validator(XSV)を使っている。JNR システムにおいても、これらのサードパーティのバリデーションエンジンのどれかの性能を検証し、スキーマバリデーションに使う予定である。

TN の他に、webMethods Business Process Management (BPM)を使用することで、複雑なビジネスプロセスを制御でき、よりわかりやすいランタイムステータスを得ることができる。このステータスを利用すれば、メッセージトラッキング以外も、今後のための分析や最適化などに活用できる。Agentrics データプールのトラッキングモジュールの成功事例を受け、JNR のアーキテクチャは webMethods BPM(6.5 バージョン)を使用し、アイテム登録やサブスクリプションのマッチングプロセスを行うことにした。具体的にいうと、RCI の処理、CIS の処理、RFCIN の処理の 3 つのプロセスが必要である。BPM プロセスのランタイムインスタンスは、メッセージがシステムに取り込まれた時間、バリデーション処理にかかった時間、データベースがアップデートされた時間、サブスクリプションがマッチされるまでにかかった時間、マッチしたサブスクリプション及び各メッセージのレスポンスが送信状況を表示することができる。

注：RFCIN は、JNR 開発の対象範囲外であるため、設計は行いが実際の開発には反映しないものとする。

XML データ変換、マッピング、XML メッセージの生成に関しては、webMethods Integration Server (6.5 バージョン)の XML ライブラリー、レコードストラクチャ、マッピングツール GUI が使用される。GUI はドラッグ&ドロップによるマッピングや、ツールに組み込まれた変換サービスを提供する。大容量のドキュメントをマッピングする際は、メモリーの過度使用を防ぐために、webMethods node iterator を使うことを推奨する。

JNR の GDSN バリデーションおよび他のビジネスルールは、webMethods フローサービス言語と Java でプログラムを組むことができる。スムーズなシステム管理のため、バリデーションルール管理のフレームワークを開発すれば、簡単にバリデーションルールの追加、修正、削除、適用、非適用設定を行うことが可能となる。これは GUI による設定、もしくはコンフィギュレーションファイルの形となる。

1.4 オラクルDBとの連携

webMethods の JDBC アダプタを使い、必要な SQL トランザクションを実行し、オラクル DB へのアイテム、サブスクリプションの保存を行う。このアダプタは、認証、SQL 実行、接続の再利用、データベースへの自動再接続を処理する。

オラクル DB からデータを抽出する際は、JDBC polling を使用する。JDBC polling は、データベース読み取りの競合を避けるため、一つのアダプタインスタンスのみをアクティブにする。処理能力の確保のため、アダプタは XML メッセージの作成と、外部送信に別々のスレッドを立てる。詳細は、セクション 4.9 を参照。

1.5 アイテム登録/サブスクリプションの管理、マッチングエンジン

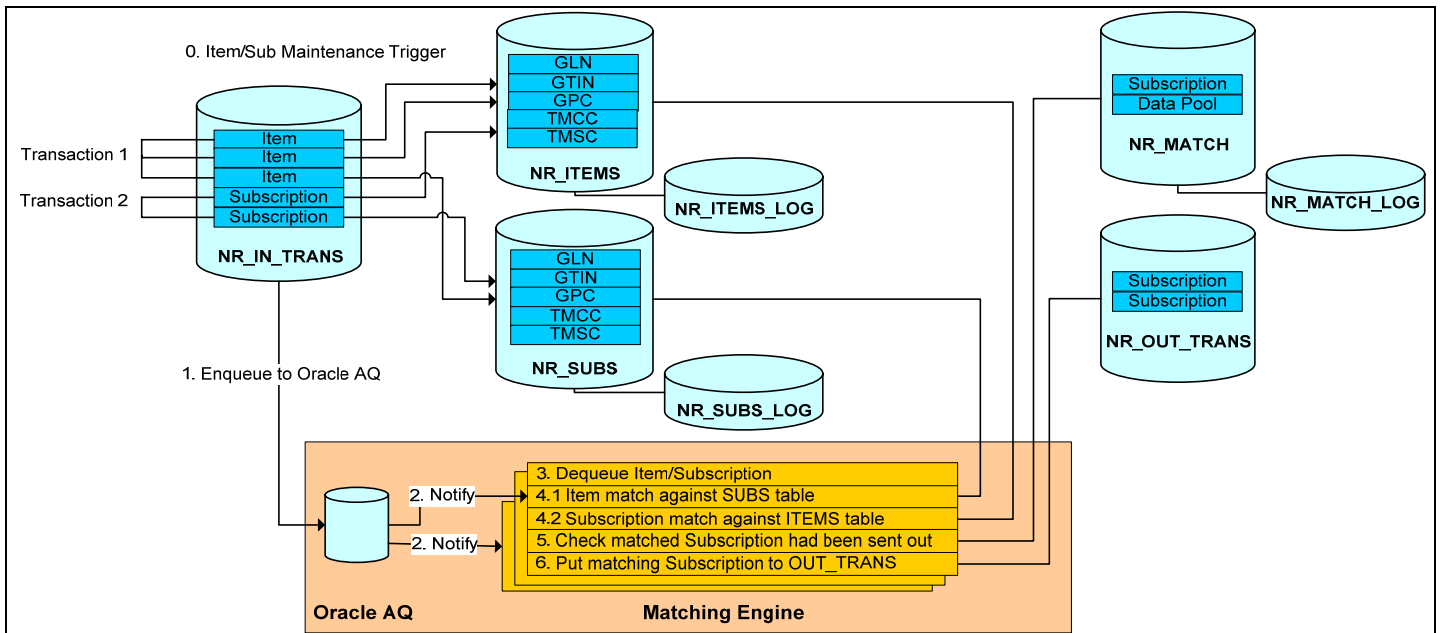
アイテム情報、サブスクリプションの保存は、Oracle 10g データベースを使用する。サブスクリプションのマッチングエンジンは、PL/SQL stored procedure でプログラミングする。マッチングエンジンの管理は、オラクル Advance Queuing (AQ)と Job Queue(JQ)を使用する。

アイテム、サブスクリプションの管理

次ページの図は、データベーステーブルレイアウトとロジカルフローを表している。アイテム、サブスクリプションは、NR_IN_TRANS テーブルに挿入され、トランザクションごとに commit もしくは rollback される。たとえば、3つのアイテムが一つのトランザクションに入った RCI メッセージの場合、すべてのアイテムが挿入され、commit または rollback される。アイテム、サブスクリプションの恒久的なテーブル (NR_ITEMS、NR_SUBS) と AQ のアップデートには、データベーストリガーを使用する。すべての恒久的テーブル関連処理はログが記録され、オラクル AQ は稼働中のマッチングエンジンに、サブスクリプションマッチングを行うよう通知する。

サブスクリプションのマッチングエンジン

マッチングエンジンは、最初は Oracle Job Queue(JQ)によって起動されるが、エンジンはいつでも処理できるよう常に稼働状態にある。AQ からマッチング処理を行うよう通知されると、エンジンはウェイクアップする。AQ により、追加の管理コマンド(suspend、restart)をエンジンに送信することも可能である。サブスクリプションのマッチングが完了した後、エンジンはマッチしたサブスクリプションがすでに送信されているかをチェックする。NR_Match テーブルは、データプールに送信されたすべてのサブスクリプションを保持し、もし、サブスクリプションがまだ一度も転送されていなければ、サブスクリプションは NR_OUT_TRANS テーブルに送られ、送信処理が行われる。オラクルデータベースアーキテクチャに関する詳細は、別紙 A を参照すること。セクション 4.10 は、アイテムとサブスクリプションのマッチングロジックに関して記述している。



1.6 ログ管理とメッセージトラッキング

企業向けシステムの管理は、機能的、運用的ステップおよびタスクは、さまざまなレベルでログが必要であり、すべてのエラー、エクセプションを探索し、サポートエンジニアに通知しなければならない。それぞれのビジネスプロセスやシナリオの設計の際、開発者とプロダクトマネージャーは、どのステップにログが必要かを判断し、トラッキング情報や問題解決のための詳細手順を、サポートエンジニアへの通知できるようにしなければならない。

通常のソフトウェア開発同様、JNR はシステム全体で共通のログやエラー処理サービスを使い、システム全体の一貫性を保つとともに、トラッキングを可能にする必要がある。唯一の相違点はオラクル AQ の中でログとエラーで、別途のデータベーステーブルを用意することで、データベース運用上のエラーログを管理する。これらのログを閲覧、集約するための、カスタム GUI の開発も必要である。

モニタリングの際、サポートエンジニアは webMethods TN と webMethods BPM を使い、トラッキングを行い、トラブルに対応する。webMethods Transaction Analysis を使えば、すべての送受信メッセージと関連する処理に簡単にアクセスできる。webMethods BPM を使うことで、サポートエンジニアは複雑なビジネスプロセスを、わかりやすいグラフィックで表示することができる。また、TN と BPM は、両方ともメッセージの再送や再処理のための機能を持つ。

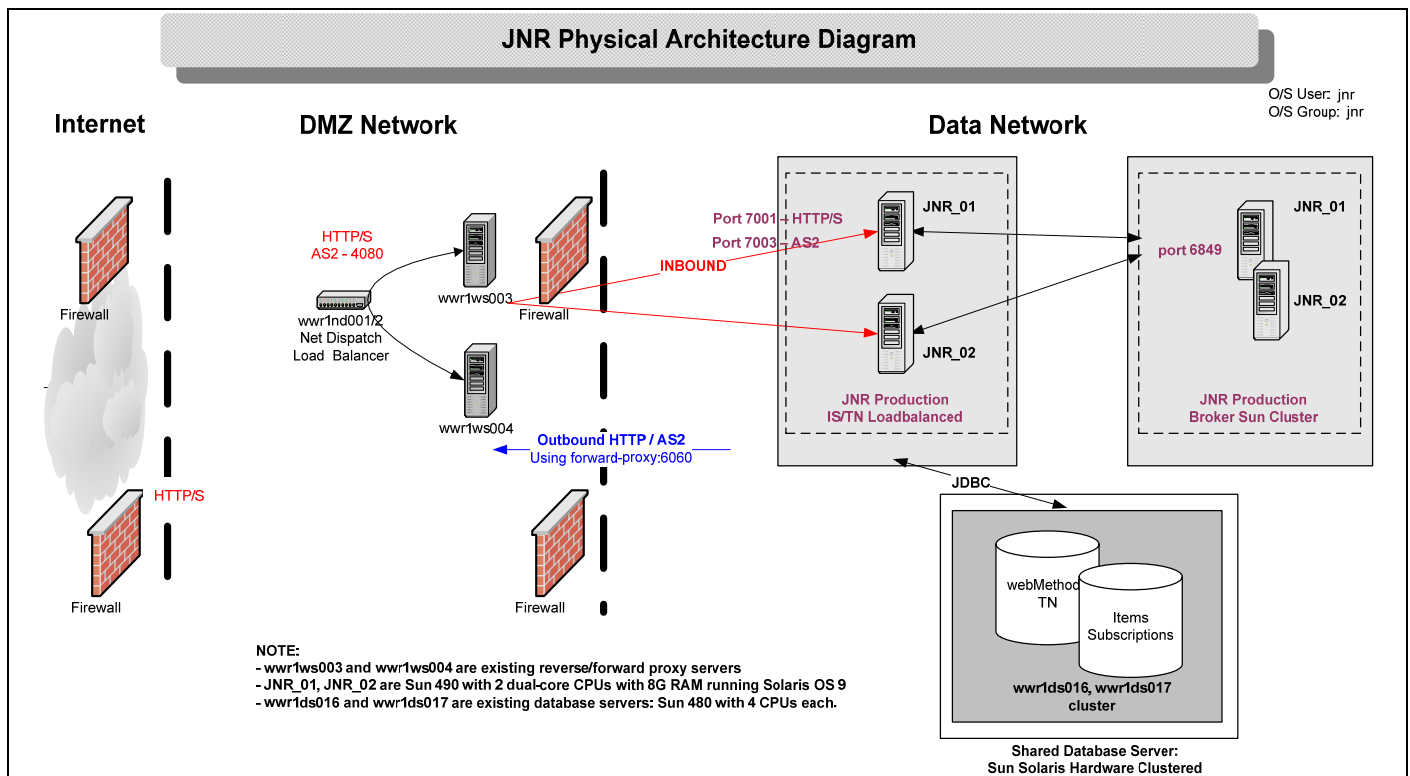
1.7 国際化サポート

JNR は、日本語を含み、マルチバイトキャラクターをサポートしなければならない。すべての XML メッセージは、webMethods Integration Suite の中で国際標準の UTF-8 エンコーディングを使用する。オラクルデータベースは、マルチバイト文字処理のための設定を行わなければならない。JNR の場合、現段階では、企業名や住所などいくつかのパーティ項目のみ日本語をサポートする必要がある。

1.8 高可用性(HIGH-AVAILABILITY)とスケーラビリティ

JNR は、365 日、24 時間稼動することが前提であり、実務利用を前提にした高可用性が求められる。Agentrics の現状のロードバランサーは、webMethods クラスタリングと複数の webMethods Integration Server のノードを同時運用

することができる。下記のアーキテクチャは、2つの UNIX サーバー（Sun Solaris 490、デュアルコア CPU x 2、メモリ8Gb）に webMethods Integration Server (IS)がインストールされていることを表す。webMethods Broker は BPM のもう一つの重要なコンポーネントであり、webMethods Broker とオラクルデータベースは、Sun clustering を使い、アクティブ・スタンバイ高可用性設定を行うことで、システムがダウンすることを最小限にする。



マルチスレッド JVM スケーラビリティ

webMethods IS は、XML バリデーションや処理を行うコアエンジンである。稼動時は、Java Virtual Machine (JVM) を使い、トランザクションをマルチスレッド方式で処理する。JVM のメモリ制限(3.1G)を克服するため、クラスタを利用し、複数の webMethods IS の JVM インスタンスを動かす設定を行う。

ハードウェアスケーラビリティ

webMethods IS はサーバー数を増やすことで、horizontal scale up が可能であり、CPU を追加することで vertical scale up が可能である。これは webMethods のクラスタリング技術とロードバランサを両方使うことで実現可能であり、オラクルデータベースも同じ方式でスケールアップできるが、vertical scale up がより一般的である。

流動制御

XML メッセージの処理は、webMethods に組み込まれた流動制御（フローコントロール）機能を使う。この機能はタイムアウト制御や同時に処理されるトランザクションのスレッド数の設定を可能にする。

データベースのアイテム、サブスクリプションのマッチングは、オラクル AQ がジョブの流れを制御する。

1.9 信頼性 (RELIABILITY)

webMethods IS とオラクル DB は、両方とも commit、rollback モードで運用されるロバストなトランザクションサーバーであり、タスクやジョブの自動再スタート機能は、サーバーに組み込まれている。ただし、メモリ上で作業が行われるカスタム処理については、手動再処理が必要になる場合もある。

JNR の開発において、webMethods TN と BPM はトランザクションステータスのトラッキング以外に、信頼できるトランザクションの処理完了にも係わっている。処理した XML メッセージは保存され、それぞれのビジネスプロセスやステップにおいて、ユーザーに影響を与えることなく再処理が可能である。いくつかのステップは自動で処理されるが、手動処理が必要な場合もある。例えば RCI メッセージは、スキーマバリデーションの後、原本のメッセージがなくても再処理が可能で、CIS はサブスクリプションマッチングを繰り返すことなく、転送することができる。

1.10 セキュリティ (SECURITY)

webMethods IS は、ユーザーID、パスワード、電子証明書による認証、ユーザーやグループレベルの認証、電子証明書による暗号化・複合化、IP アドレスドメインによるアクセス制御など、さまざまなセキュリティ機能を有する。webMethods セキュリティ機能に関する詳細は、webMethods Fabric 6.5 Security Best Practice を参照すること。以下のセクションは、セキュリティに関して JNR が webMethods IS と Oracle DB をどのように活用するかについて説明する。

ホスティング

JNR は Agentrics のセキュリティホスティングルールに準拠してホスティングされる。webMethods IS は、DMZ ネットワークによって分離されたアプリケーション・データベースレイヤーで稼動する。Sun One プロキシサーバーにより、DMZ ネットワークの中に AS2 メッセージを通した後、Sun Solaris Unix サーバーで、webMethods IS はスーパーユーザーの権限を求めない「jnr」として稼動する。また、既存のデータベースクラスタの中に、新しいオラクル DB のインスタンスも追加する。

アクセス

JNR へのアクセスは 1) インターネット経由の inbound/outbound の AS2 メッセージ 2) アドミン用のシステム内部アクセスの 2 つの方法が提供される。すべての AS2 メッセージは、特定のポート(4080)を経由し、ユーザー企業の電子証明書により、暗号化され、署名されている。Outbound メッセージは、forward プロキシを経由し、JNR の電子証明書を使って暗号化、署名される。すべての内部管理アクセスは、ユーザーID/パスワードの他、適切なアクセスコントロールを用いる。

2 機能要件および主なタスク

2.1 EANUCC XML の認識、ルーティング

すべての送受信 XML メッセージは、webMethods TN で認識、送信される。

- TN ドキュメント認識 (document recognition) – RCI、CIS、RFCIN、レスポンス (レスポンス、CIRR、またはエクセプション)
 - 下記のドキュメント名を使い、ドキュメントタイプを作る
 - ネームスペースを抽出する
 - 下記の XPATH を使い、すべてのメッセージの SenderID、ReceiverID、ドキュメント ID を抽出する。
 - レスポンスは、CIRR、レスポンス、エクセプションを含むかを識別する。メッセージは CIRR、eANUCCResponse、gDSNException を複数持つことができる。

ドキュメント名	変数名	内容	XPATH
EANUCC 21 – RCI	SenderID	Sender GLN (SDP GLN)	/sh:StandardBusinessDocument[0]/sh:StandardBusinessDocumentHeader[0]/sh:Sender[0]/sh:Identifier[0]
	ReceiverID	Receiver GLN (JNR GLN)	/sh:StandardBusinessDocument[0]/sh:StandardBusinessDocumentHeader[0]/sh:Receiver[0]/sh:Identifier[0]
	DocumentID	Message ID	/sh:StandardBusinessDocument[0]/sh:StandardBusinessDocumentHeader[0]/sh:DocumentIdentification[0]/sh:InstanceIdentifier[0]
	ConversationID	For Modeler	//sh:Receiver/sh:Identifier \$union\$ //sh:Sender/sh:Identifier \$union\$ //sh:DocumentIdentification/sh:InstanceIdentifier \$union\$ //sh:Scope/sh:CorrelationInformation/sh:RequestingDocumentInstanceIdentifier
EANUCC 21 – CIS	SenderID	Sender GLN (SDP GLN)	/sh:StandardBusinessDocument[0]/sh:StandardBusinessDocumentHeader[0]/sh:Sender[0]/sh:Identifier[0]
	ReceiverID	Receiver GLN (JNR GLN)	/sh:StandardBusinessDocument[0]/sh:StandardBusinessDocumentHeader[0]/sh:Receiver[0]/sh:Identifier[0]
	DocumentID	Message ID	/sh:StandardBusinessDocument[0]/sh:StandardBusinessDocumentHeader[0]/sh:DocumentIdentification[0]/sh:InstanceIdentifier[0]
	ConversationID	For Modeler	//sh:Receiver/sh:Identifier \$union\$ //sh:Sender/sh:Identifier \$union\$ //sh:DocumentIdentification/sh:InstanceIdentifier \$union\$ //sh:Scope/sh:CorrelationInformation/sh:RequestingDocumentInstanceIdentifier
EANUCC 21 – Response	SenderID	Sender GLN (JNR GLN)	/sh:StandardBusinessDocument[0]/sh:StandardBusinessDocumentHeader[0]/sh:Sender[0]/sh:Identifier[0]
	ReceiverID	Receiver GLN (DP GLN)	/sh:StandardBusinessDocument[0]/sh:StandardBusinessDocumentHeader[0]/sh:Receiver[0]/sh:Identifier[0]
	DocumentID	Message ID	
	ConversationID	Correlation ID = Message ID of original RCI or CIS For Modeler	
	ResponseType	Response, CIRR, or Exception	

- レコードストラクチャ – RCI、CIS、RFCIN、レスポンス
 - スキーマから作成する
 - サンプルファイルを使い、スキーマに対してテストする
- プロセスルール
 - 未登録の送信者：サービスを呼び出し、アラートを出す
 - 未設定のドキュメント：サービスを呼び出し、アラートを出す
 - JNR が受信者ではない：サービスを呼び出し、アラートを出す
 - 登録済みのサーバー (DP) → JNR : プロセスルールを実行する
 - JNR→登録済みのサーバー(DP) : プロセスルールを実行する

2.2 EANUCC XML バリデーション

JNR は、EANUCC XML のバリデーションを2つの段階で実施する。

1. スキーマバリデーション - スキーマによる XML メッセージの妥当性チェック
2. ビジネスバリデーション - GDSN Operations Guide に明記されたルール

スキーマバリデーションは、webMethods に組み込まれたバリデーションエンジンを使うか、サードパーティ製の XML スキーマバリデーションエンジンを流用する。

ビジネスバリデーションは、カスタム webMethods/Java services を開発して実施する。必要なデータは XML データベースとオラクルデータベースに保存される。

別紙Cに実際に導入した XML バリデーションエンジンに関する記述がある。

関連する開発タスク:

- スキーマバリデーション
 - Java API、もしくは UNIX プロセスとして使用できるフリーウェア、シェアウェアのスキーマバリデーションエンジンをリサーチする。Sun Solaris 9 で動くことが条件。
 - ここを探す <http://www.w3.org/XML/Schema>
 - GS1 が使用したバリデーションエンジンを検討する。
 - XML Schema Validator (XSV)
 - Xerces (Apache)
 - Well-formed でない場合、正しいネームスペースを参照していない場合 (ERROR 136)
 - 処理結果は、レスポンスメッセージにマッピングできるよう、実用的かつ明確な説明を持たなければならない。

■ GDSN 運用バリデーション – 構造バリデーション

StandardBusinessDocumentHeader
Sender
Authority
<gln>
Receiver
DocumentIdentification
Standard
タイプ Version
InstanceIdentifier
タイプ
Multiple タイプ
CreationDateAndTime
eanucc.message
entityIdentification – message id
eanucc.transaction
entityIdentification – transaction id
command
eanucc.documentCommand
documentCommandHeader
entityIdentification – command id
documentCommandOperand
gdsn.<message タイプ>
<message タイプ>Identification – document id

EANUCC XML 構造概要

○ メッセージ、トランザクション、コマンド、ドキュメントレベルのユニークチェック

レベル	ユニーク値	XPATH
Message	Transaction ID	/sh:StandardBusinessDocument/eanucc:message/eanucc:transaction/entityIdentification/uniqueCreatorIdentification
Transaction	Command Header ID	/sh:StandardBusinessDocument/eanucc:message/eanucc:transaction/command/eanucc:documentCommand/documentCommandHeader/entityIdentification/uniqueCreatorIdentification
Command	Document ID – RCI	/sh:StandardBusinessDocument/eanucc:message/eanucc:transaction/command/eanucc:documentCommand/documentCommandOperand/gdsn:registryCatalogueItem/registryCatalogueItemIdentification/uniqueCreatorIdentification
Command	Document ID – CIS	
Command	Document ID – RFCIN	
Command	Document ID – Response	

- 各レベルで共通の値のチェック:

名前	レベル/値	XPATH
GLN	Message Level contentOwner gln	/sh:StandardBusinessDocument/eanucc:message/entityIdentification/contentOwner/gln
	SBDH sender gln	/sh:StandardBusinessDocument/sh:StandardBusinessDocumentHeader/sh:Sender/sh:Identifier
contentOwner GLN [GDSN_VAL 300]	Transaction	/sh:StandardBusinessDocument/eanucc:message/eanucc:transaction/entityIdentification/contentOwner/gln
	Command	/sh:StandardBusinessDocument/eanucc:message/eanucc:transaction/command/eanucc:documentCommand/documentCommandHeader/entityIdentification/contentOwner/gln
	Document – RCI	/sh:StandardBusinessDocument/eanucc:message/eanucc:transaction/command/eanucc:documentCommand/documentCommandOperand/gdsn:registryCatalogueItem/registryCatalogueItemIdentification/contentOwner/gln
	Document – CIS	
	Document – RFCIN	
	Document – Response	

■ GDSN 運用バリデーション – 不変値

名前	値	XPATH
SBDH HeaderVersion	1.0	/sh:StandardBusinessDocument/sh:StandardBusinessDocumentHeader/sh:HeaderVersion
SBDH Sender.Identifier.Authority	EAN.UCC	/sh:StandardBusinessDocument/sh:StandardBusinessDocumentHeader/sh:Sender/sh:Identifier/@Authority
SBDH Receiver.Identifier.Authority	EAN.UCC	/sh:StandardBusinessDocument/sh:StandardBusinessDocumentHeader/sh:Receiver/sh:Identifier/@Authority
SBDH DocumentIdentification.Standard	EAN.UCC	/sh:StandardBusinessDocument/sh:StandardBusinessDocumentHeader/sh:DocumentIdentification/sh:Standard
SBDH DocumentIdentification.TypeVersion	2.1	/sh:StandardBusinessDocument/sh:StandardBusinessDocumentHeader/sh:DocumentIdentification/sh:タイプ Version
SBDH DocumentIdentification.Type	registryCatalogueItem, catalogueItemSubscription, requestForCatalogueItemNotification, or GDSNResponse	/sh:StandardBusinessDocument/sh:StandardBusinessDocumentHeader/sh:DocumentIdentification/sh:タイプ
gdsn.registryCatalogueItem.contentVersion.versionIdentification	2.1	/sh:StandardBusinessDocument/eanucc:message/eanucc:transaction/command/eanucc:documentCommand/documentCommandOperand/gdsn:registryCatalogueItem/contentVersion/versionIdentification
gdsn.registryCatalogueItem.documentStructureVersion.versionIdentification	2.1	/sh:StandardBusinessDocument/eanucc:message/eanucc:transaction/command/eanucc:documentCommand/documentCommandOperand/gdsn:registryCatalogueItem/documentStructureVersion/versionIdentification
SBDH BusinessScope.Type	GDSN	/sh:StandardBusinessDocument/sh:StandardBusinessDocumentHeader/sh:BusinessScope/sh:Scope/sh:タイプ
SBDH DocumentIdentification.Multiple Type	Should not exist	

■ 共通 GDSN 運用バリデーション – 変数チェック

- GLN – 13 桁&チェックデジット
- GTIN – 14 桁&チェックデジット
- TMCC – 3 桁チェック
- キャンセル日、停止日を XML に追加可能
- SDP、RDP、メーカー、卸、小売の登録 GLN チェック [GDSN_VAL 343, 344]
- ユニークアイテムチェック – IP GLN, GTIN、TMCC、TMSC、GPC
- TMCC – コードリスト = (国コード (2 文字) と数字 (3 桁))
 - 例: US 840
- TMSC – コードリスト = <2 文字国コード>-<2 文字地域コード>
- TMSC を TMCC に対しバリデーションをかける 例 TMCC = 840, TMSC US-CA
 - TMSC の最初の 2 桁を探し、TMCC 一覧と比べる
- GPC – コードリスト [GDSN_VAL 39]

** TMCC、TMSC、GPC を DB のキャッシュテーブルと webMethods キャッシュに保存して、検索を速くすることを推奨する

2.3 共通変換、ユーティリティサービス

- 変換サービス:
 - 日付変換 - xs:dateTime¹ ↔ Julian date (オラクルで使用)
 YYYYMMDDThh:mm:ss<.ss>±hh:mm
 例: 1997-07-16T19:20.33-05:00 (7:20.33 EST)
- ユーティリティサービス:
 - ユニークな InstanceIdentifier を生成する

2.4 共通エクセプション処理

エクセプションの処理と webMethods のサービスエラーには、以下の2つの方法を用いる。

1. 直接呼び出すことのできるメインサービスの通常の try-catch ブロックを使う
2. webMethods の標準機能である Event Manager を使う

Developer に付随する Event Manager は、エクセプションイベントの取得に使用され、発生したエクセプションの閲覧、編集、削除ができる。Event Manager はサーバーをモニターしてエクセプションイベントを探し、Event Handler を呼び出す。エクセプションが発生した際、Event Handler は Developer に設定されたユーザー指定可能な汎用のエラー処理サービス呼び出す。イベントフィルターを使うことで、特定のサービス、パッケージ、ユーザー、ポートで発生したイベントのみを取得させることも可能である。

ただし、DB 接続の遮断、データロックの解除など特定の状況の場合は、個別に処理しなければならない。

2.5 RCI の処理

- RCI バリデーション
 - 共通 GDSN 運用バリデーション
 - GTIN が存在しない[GDSN_VAL 16]
 - 登録後、GTIN の CBR または CORRECT はできない [GDSN_VAL 41]
 - アイテム登録後、データソースは変更できない [GDSN_VAL 42]
 - 登録された RCI のキャンセル日は、その日付が過ぎた後は変更できない。Change_by_refresh コマンドでは、この項目をアップデートできない。CORRECT コマンドのみが使用可能である。[GDSN_VAL 44]
 - 登録された RCI の停止日は、その日付が過ぎた後は変更できない。Change_by_refresh コマンドでは、この項目をアップデートできない。CORRECT コマンドのみが使用可能である。[GDSN_VAL 45]
 - RCI メッセージによるアイテムの更新、修正を行うためには、GTIN/GLN/TM が JNR に既に登録されていなければならない。 [GDSN_VAL 47]
- DB への挿入
- レスポンス生成
 - BusinessScope の追加(Correlation)
 - CIRR
 - エクセプションの生成

RCI の処理方法については、別紙 D を参照。

¹ GS1 XML Release 2.0 Technical Users Guide (Pg 18-20)

2.6 CIS の処理

- CIS バリデーション
 - 共通 GDSN 運用バリデーション
 - データ受領者 GLN は必ず存在しなければならない [GDSN_VAL 82]
 - 次の項目のうち最低でも一つ、もしくは複数存在しなければならない: targetMarket、gtin または GPC または GLN [GDSN_VAL 86]
 - GPC と GTIN が同時に存在してはならない [GDSN_VAL 87]
- DB への挿入
- レスポンス生成
 - BusinessScope の追加(Correlation)
 - レスポンス/エクセプションの生成

CIS の詳細については、別紙 E を参照。

2.7 RFCIN の処理

- RFCIN バリデーション
 - 共通 GDSN 運用バリデーション
 - データ受領者 GLN は必ず存在しなければならない [GDSN_VAL 82]
 - 次の項目のうち最低でも一つ、もしくは複数存在しなければならない: targetMarket、gtin または GPC または GLN [GDSN_VAL 86]
 - GPC と GTIN が同時に存在してはならない [GDSN_VAL 87]
- DB への挿入
- レスポンスの作成
 - BusinessScope の追加(Correlation)
 - レスポンス/エクセプションの生成

RFCIN の処理方法については、別紙 F を参照。

2.8 レスポンス、エクセプション処理

- 受信したメッセージに対して、1 対 1 で送信しなければならない(RCI、CIS、RFCIN)。ただし、エクセプションは例外とする [GDSN_Ops 5.5]
- CIRR と eANUCCResponse の場合、レスポンスのステータスは必ず ACCEPTED でなければならない。[GDSN_Ops]

メッセージ種類	ステータス値	XPATH
Response - CIRR	ACCEPTED	/sh:StandardBusinessDocument/eanucc:message/gdsn:catalogueItemRegistrationResponse/@responseStatus
Response - eANUCCResponse	ACCEPTED	/sh:StandardBusinessDocument/eanucc:message/gdsn:eANUCCResponse/@responseStatus

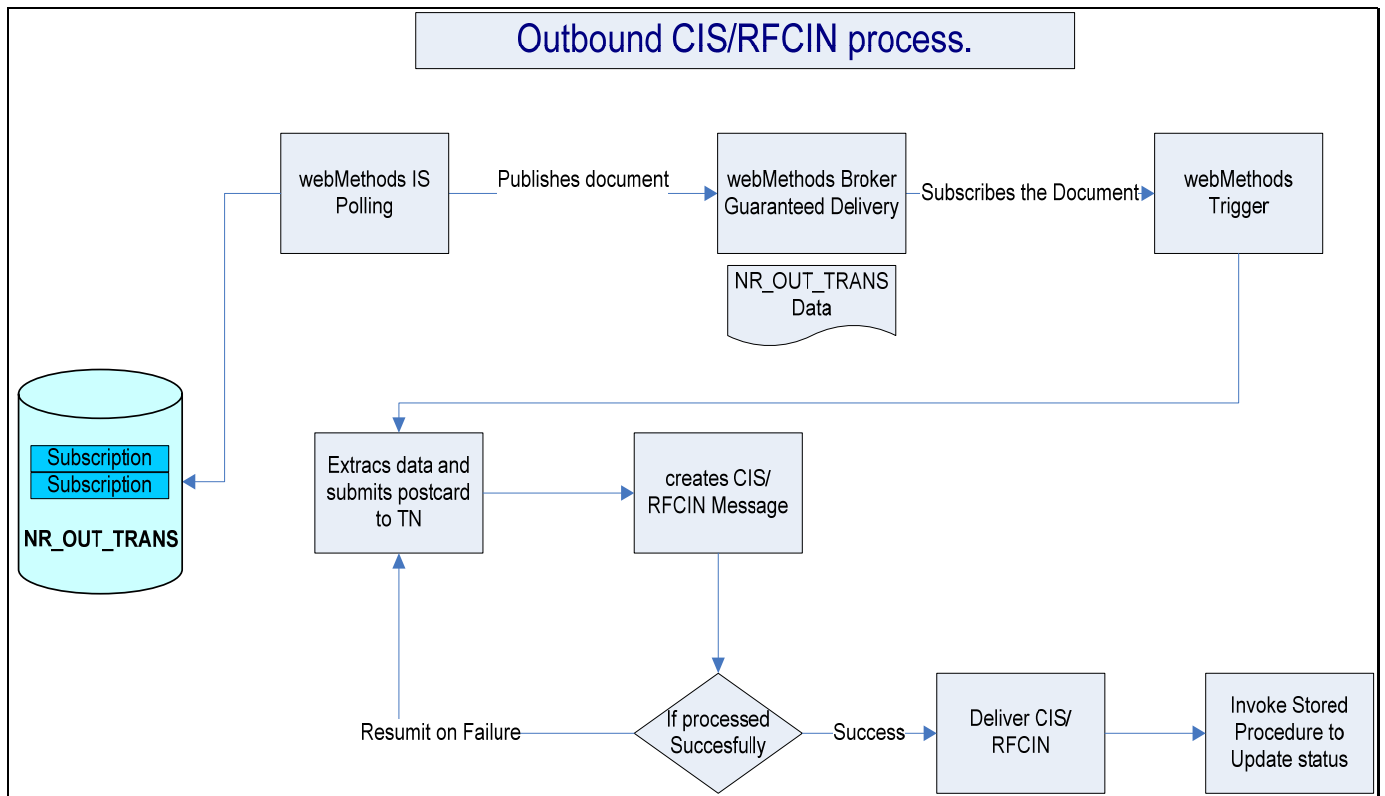
- eANUCCResponse と gDSNException の場合、以下の送信者と受信者の値は、元のメッセージの SBDH の送信者と受信者と一致しなければならない [GDSN_Ops 5.5]:

値	レスポンスタイプ	XPATH
送信者 GLN	eANUCCResponse	/sh:StandardBusinessDocument/eanucc:message/gdsn:gDSNException/sender
	RCI、CIS、RFCIN の SBDH	/sh:StandardBusinessDocument/sh:StandardBusinessDocumentHeader/sh:Sender/sh:Identifier
受信者 GLN	eANUCCResponse	/sh:StandardBusinessDocument/eanucc:message/gdsn:gDSNException/receiver
	RCI、CIS、RFCIN の SBDH	/sh:StandardBusinessDocument/sh:StandardBusinessDocumentHeader/sh:Receiver/sh:Identifier
送信者 GLN	gDSNException	/sh:StandardBusinessDocument/eanucc:message/gdsn:gDSNException/sender
	RCI、CIS、RFCIN の SBDH	/sh:StandardBusinessDocument/sh:StandardBusinessDocumentHeader/sh:Sender/sh:Identifier
受信者 GLN	gDSNException	/sh:StandardBusinessDocument/eanucc:message/gdsn:gDSNException/receiver
	RCI、CIS、RFCIN の SBDH	/sh:StandardBusinessDocument/sh:StandardBusinessDocumentHeader/sh:Receiver/sh:Identifier

- CIRR の場合、レスポンスはドキュメントレベルで返す [GDSN_Ops 5.5]
- eANUCCResponse の場合、レスポンスはトランザクションレベルで返す [GDSN_Ops 5.5]
- gDSNException の場合、エクセプションは対応するメッセージのエラーの種類、エラーの発生場所によって以下のレベルで返す必要がある [GDSN_Ops 5.5]:
 - メッセージレベル – もし SBDH やメッセージレベルで問題があった場合 例：メッセージタグ
 - トランザクションレベル
 - コマンドレベル
 - ドキュメントレベル
 - アトリビュート（項目）レベル – 最下位レベル

○

2.9 CIS / RFCIN の送信処理



オラクル DB でサブスクリプションのマッチングが行われた後、結果は NR_OUT_TRANS テーブルに記録され、webMethods の JDBC ポーリング通知により、送信向けの CIS/RFCIN の処理がトリガーされる。決められた時間間隔で NR_OUT_TRANS テーブルからデータがポーリングされ、webMethods Broker に対して公開される。webMethods のトリガーは公開されたドキュメントを取得し、ポストカードを作成するためのサービス呼び出して TN に送信する。ポストカードは、マッチング情報をすべて含んでおり、もしシステムの問題があった場合、再送処理のために使われる。その後、TN のプロセッシングルールが別のサービス呼び出し、送信用の CIS/RFCIN を作成し、該当データプールに送信する。

サンプルポストカード(NR XML)

```
<? xml version="1.0"?>
<NR_XML>
  <Header>
    <sender>OracleApps</sender>
    <receiver>0821620001112</receiver>
    <doc 名前>NR_XML</doc 名前>
    <application>NR</application>
    <transaction タイプ>CIS - ADD</transaction タイプ>
    <dateTime>2006-10-10 14:35:49.220</dateTime>
  </Header>
  <Payload>
    <additionalInfo>
      <matchParams>
        <ID>1303</ID>
        <matchID>1213</matchID>
        <action>A</action>
      </matchParams>
    </additionalInfo>
  </Payload>
</NR_XML>
```

webMethods トリガーおよびポーリング通知

webMethods ポーリング通知は、特定の時間間隔で実行されるようにスケジュールされている。ポーリング通知が成功した後、テーブルの記録は削除される。webMethods Broker のトリガーの設定により、重複ドキュメントの探知や処理が行われる。下記のトリガー設定は、重複ドキュメントをチェックするため、Developer の中で設定される。

重複の探知 - true

重複の探知は、ドキュメントに対して 1 回のみ実行される。

ヒストリーの使用 - true

webMethods によって管理されるヒストリーテーブルにより、重複を検知するプロパティ。

History time to live - 時間 (秒 / 分 / 時 / 日 / 週 / 月 / 年).

トリガーによって受信したドキュメントのヒストリーを、データベースが管理する期間を定める

2.10 RCI、CIS、RFCIN のマッチングエンジン

マッチングエンジンは、1) アイテムをサブスクリプションに対してマッチさせる (RCI) 2) サブスクリプションをアイテムに対してマッチさせる (CIS、RFCIN) の 2 つのタイプのマッチを行う。マッチングは、すべての項目が提供された場合も、そうではない場合も同じように実施される。クエリーはロジカルヌルスイッチを使い、提供された項目の値がヌルであると、いかなる値もマッチを認める。マッチングは必ずアクティブなアイテムとサブスクリプションに対して行われ、RCI によってトリガーされた場合は、RFCIN が既存してもマッチングは行わない。(RCI のマッチングは必ず保存された CIS に対して行われる)

マッチングロジックのスク립ト例：

```
and (gln_supplier = el_gln_supplier or gln_supplier is null)
and (gtin         = el_gtin         or gtin         is null)
and (tmcc        = el_tmcc        or tmcc        is null)
and (tmsc        = el_tmsc        or tmsc        is null)
and (gpc         = el_gpc         or gpc         is null);
```

2.11 パーティプロフィールの管理

- webMethods TN を活用する
 - DP プロフィールの管理で使用を検討
- DP に所属するユーザー企業 (trading partners) の GLN 情報保存用の DB を構築を検討
 - 管理用の UI が必要になる

3 プラットフォーム、運用要件、デザイン

3.1 WEBMETHODS 6.5 の検証および設定

- サービスパック、Feature Pack、修正パッチ、既知バグのチェック
- 6.1 の問題点の修正が 6.5 に反映されているかチェック
- Developer の検証および使用
- TN の大容量ファイル処理設定
- ブローカーのタイムアウト設定
- スレッド数の調整、接続数の調整、タイムアウト値の設定 (HTTP/S、BPM steps)

3.2 プログラミングテンプレート

- パッケージ、サービスのネーミングコンベンション
 - NR_<package Name>.NR_<package Name>
 - config folder – 特定パッケージのための設定が入っている
 - getConfig, setConfig, startConfig, shutdownConfig
 - inbound folder – 受信メッセージの処理
 - outbound folder – 送信メッセージの処理
 - utils folder – パッケージごとのユーティリティサービス
 - rec folder – レコードストラクチャ
 - パッケージの一覧:
 - NR_EANUCC – RCI、CIS、RFCIN、CIRR、レスポンスのレコードストラクチャ
 - NR_Validation – スキーマバリデーション
 - NR_RCI – RCI 処理
 - NR_CIS – CIS 処理
 - NR_RFCIN – RFCIN 処理
 - NR_Uutilities – 共通ユーティリティサービス、java エラー処理
 - NR_Common – インターフェース全般の共通サービス
 - NR_Admin – アドミン、運用サービス
 - サービス名:
 - 小文字で始まる
 - 例 processRCI, processMessage
- IS フローテンプレートと debug、exception ブロック
 - エラーハンドリングの呼び出しを含む
- 下記の項目対するエラーハンドリング、及びログ
 - レベル – クリティカル、エラー、ログ
 - クリティカル → ページャー送信、メール送信、ログ
 - エラー → メール送信、ログ
 - ログ → ログ
 - モジュール
 - 通信
 - バリデーション
 - プロセス - RCI
 - CIS の処理
 - プロセス – RFCIN
 - 題名
 - 詳細
 - ログファイル名の決定及び保存

3.3 ファイル処理、ネーミングコンベンション

- ファイル保存のタイミングと保存先を決める
 - ディレクトリーの構造を考える – (GLN、年、月、日)または(年、月、日、GLN)
- ユニークなファイル名を使う
 - オラクルのシーケンサーを使うか、オラクルが使えない場合デフォルトを使う

3.4 AS2 設定、改善

- 自動再送
- 送信失敗の場合アラートを出す – 送信失敗の際の通知機能
- Sun と Apache のリバース、フォワードプロキシを検証する
 - 現状
 - リバースプロキシ – SunOne 6.1 SP5 と SunOne pass-through または BEA plugin
 - フォワードプロキシ – Netscape Proxy server 3.6
 - Apache を検証

3.5 ORACLE (10G) 検証—DB アーキテクト

- 10g の既知の問題点の確認
- XML データベース機能を活用する/10g で使用する

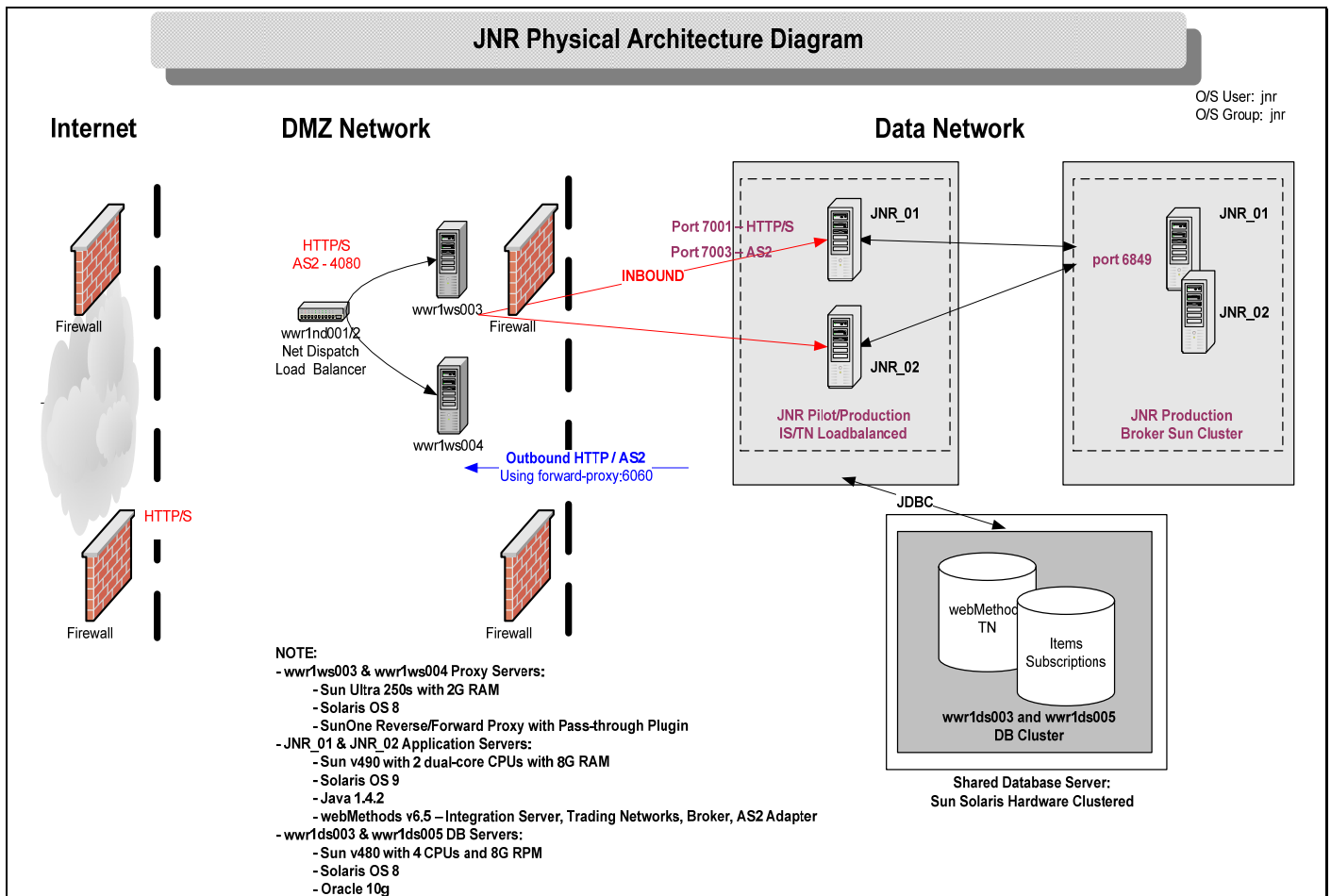
-

4 環境セットアップ

webMethods 6.5 IS/TN と Broker を以下の環境にインストールする:

- Development (開発環境)
- QA、Staging 環境
- パイロット環境– クラスタを使用

5 別紙 A: 物理アーキテクチャ



システムのスペック、概要は以下の通りである。

	wwr1ws003 / wwr1ws004 プロキシ サーバー	JNR_01 / JNR_02 アプ リケーションサーバー	wwr1ds003 / wwr1ds005 DB サー バー
ハードウェア	Sun Ultra 250s with 2G RAM	Sun v490 with 2 dual- core CPUs with 8G RAM	Sun v480 with 4 CPUs and 8G RAM
OS	Solaris OS 8	Solaris OS 9	Solaris OS 8
アプリケーシ ョンサーバー	SunOne Reverse/Forward Proxy with Pass- through Plug-in	webMethods v 6.5 – Integration Server, Trading Networks, Broker, AS2 Adapter	Oracle 10g
JVM		Java 1.4.2	

7 別紙 B: Xerces スキーマバリデーション

スキーマバリデーションは、Xerces 2.6.2 を使用した。Xerces 2.6.2 は SAX2.0.2 と SAX2 Extensions 1.1 を含む。

8 別紙 C: JNR 設定ファイル

以下の設定ファイルは、JNR アプリケーションをサポートするためのランタイムプロパティを含む。

1. **attribute_xpath.properties** – GDSN レスポンスで通知されるバリデーションエラー発生時の項目の xpath を含む。

例：：

```
rci_uniqueCreatorIdentification=sh: StandardBusinessDocument/eanucc: message/eanucc: transaction/command/eanucc: documentCommand/documentCommandOperand/gdsn: registryCatalogueItem/registryCatalogueItemIdentification/uniqueCreatorIdentification
```

2. **eanucc.properties** – EANUCC SBDH のバリデーションを行うための値を含む。また、送信用の XML メッセージを作成するためのメッセージタイプ、JNR の GLN などの情報も含む。

例：：

- headerVersion=**1.0**
- standard=**EAN.UCC**
- typeVersion=**2.1**
- businessScope=**GDSN**
- responseStatus=**ACCEPTED**
- messageType.response=**GDSNResponse**
- messageType.rfcin=**requestForCatalogueItemNotification**
- messageType.cis=**catalogueItemSubscription**
- nrGLN=**0821620001112**

3. **environment.properties** - JNR アプリケーションをサポートするための設定値を含む

- environmentID=**JNRDEV** – サーバーインスタンス ID
- serverRootDir=**/nr/jnr/wm65_predev** – サーバーが動いているフォルダ
- inboundDataDir=**/nr/jnr/data_predev/inbound** – 受信ファイルが保存されるフォルダ
- outboundDataDir=**/nr/jnr/data_predev/outbound** – 送信ファイルが保存されるフォルダ
- logFileLocation=**/nr/jnr/data_predev/logs** – ログファイルが保存されるフォルダ
- pagerEmailID=**nr_dev_support@agentrics.com** – 処理エラー通知をサポートチームに送信するためのメールアドレス（モバイル機器用のメールアドレス）
- systemEmailID=**b2b@agentrics.com** – システムエラーをシステム管理チームに送信するためのメールアドレス
- systemEmailccID=**nr_dev_support@agentrics.com** – システムエラーをシステム管理チームに送信するためのメールアドレス
- nrSupportEmailID=**nr_dev_support@agentrics.com** -処理エラー通知をサポートチームに送信するためのメールアドレス
- webm.mail.host 名前=**localhost** – メールサーバーの ID

4. **nrErrorCodes.properties** – GDSN レスポンスで報告されたエラーメッセージの内容を含む。GDSN バリデーションエラーのすべてのエラーコードには、エラーメッセージがついている。

例：： 101=cancelDate は今日の日付と同じか、今日以降でなければならない。

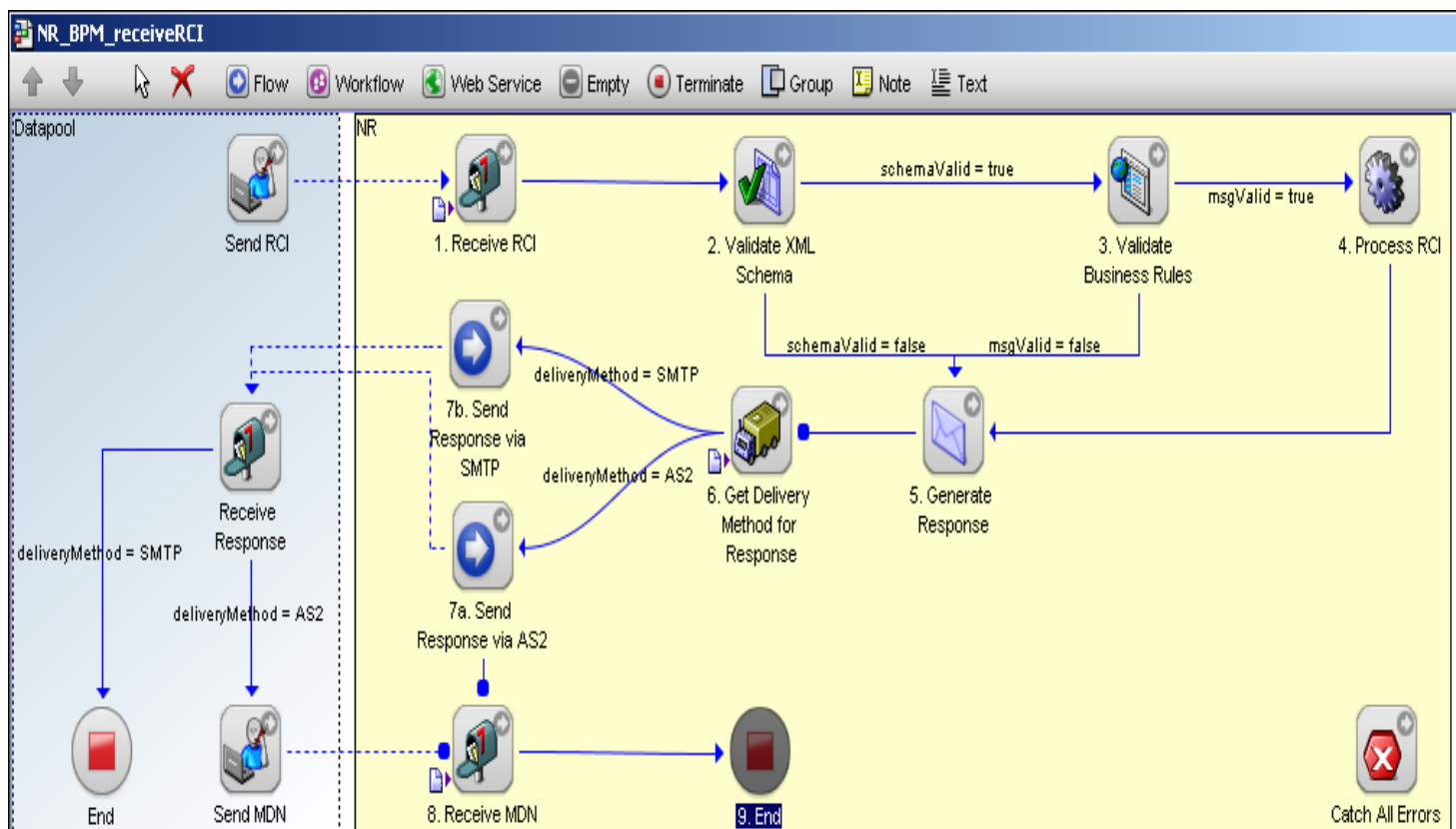
5. **schemaValidation.properties** – 取り込んだ XML メッセージのスキーマバリデーションを行うためのスキーマのロケーション情報を含む。それぞれの EANUCC XML メッセージのバリデーションを行うためのスキーマの一覧は、以下の通りである：
- response=/nr/jnr/data_predev/schemas/ResponseProxy.xsd
 - rfcIn=/nr/jnr/data_predev/schemas/RequestForCatalogueItemNotificationProxy.xsd
 - cis=/nr/jnr/data_predev/schemas/CatalogueItemSubscriptionProxy.xsd
 - sbdh=/nr/jnr/data_predev/schemas/sbdh/StandardBusinessDocumentHeader.xsd
 - webm.nr.jdbc.lt.connection=NR_JDBC_CON.connections\NR
 - rci=/nr/jnr/data_predev/schemas/RegistryCatalogueItemProxy.xsd
6. **validationServices.properties** – ランタイム中に実施されるバリデーションの一覧を含む。JNR ビジネスバリデーションは以下の 4 つのレベルで行われる：
- メッセージレベル
 - トランザクションレベル
 - コマンドレベル
 - アイテムレベル （メッセージとコマンドによって違う 例：RCI ADD）

各バリデーションルールは適用、非適用の設定が可能である。これによって、GDSN ビジネスバリデーションルールのフレキシブルな管理ができる。

例：：MSG.validateSBDHvalues=true

9 別紙 D: RCI の受信と処理

RCI 受信プロセスは、RCI メッセージの受信、パーシング、バリデーション、アイテム情報の NR_IN_TRANS テーブルへの挿入、レスポンスの作成、AS2 または SMTP を使ったデータプールへのレスポンス送信で構成される

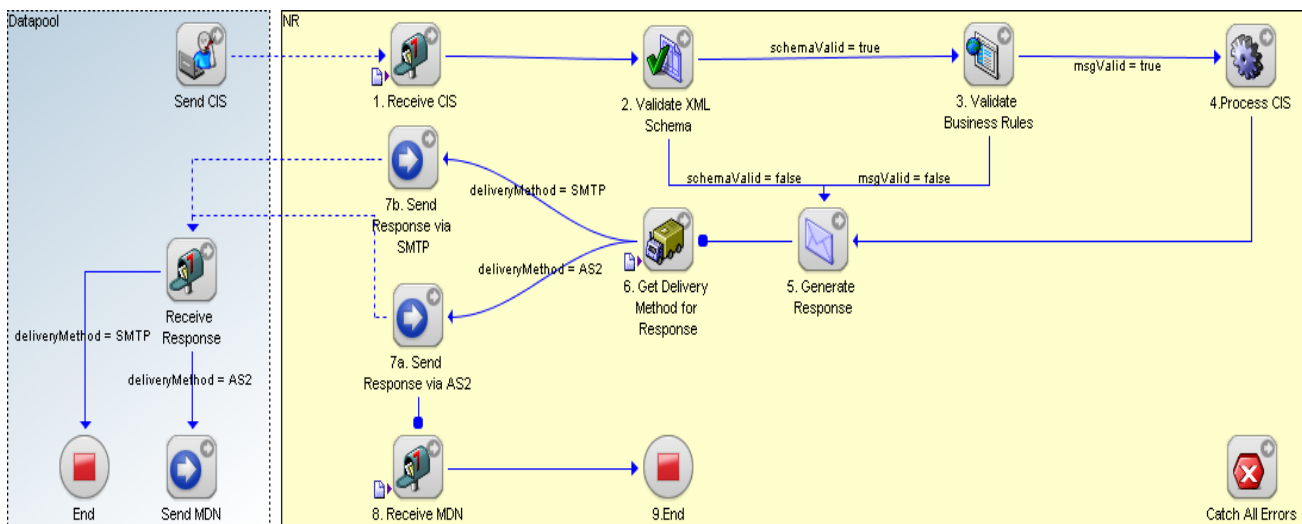


BPM ステップ名	サービス名	再送	説明	依存サービス
1. RCI 受信	NR_BPM_receiveRCI.Design_Server: 1_receiveRCI	Yes	1. TN からドキュメントを取得する 2. メッセージレベルの情報を取得し、NR_TRNS_HIST に挿入する 3. ディスクに保存	1. NR_Common.services: getInactiveValsForPartner 2. NR_Common.utils: persistToDisk
2. スキーマバリデーション	NR_BPM_receiveRCI.Design_Server: 2_validateXML	Yes	1. Xerces バリデーションエンジンを使い、メッセージを GDSN スキーマに対してチェックする	1. NR_XMLValidation: xercesValidate
3. ビジネスバリデーションの実行	NR_BPM_receiveRCI.Design_Server: 3_validateMsg	Yes	1. メッセージバリデーション	1. NR_Validation.gdsn.msg: validateGDSNMessage

4. RCI 処理	NR_BPM_receiveRCI.Design_Server:4_processTransaction	Yes	<p>トランザクションごとにプロセスを繰り返す。 トランザクションレベルのバリデーションを行う コマンドレベルのバリデーションを行う アイテムレベルのバリデーションを行う</p> <p>NR_IN_TRANS テーブルに挿入する GDSN レスポンス・GDSN エクセプションを作成する 注：データベースセッションは、commit、rollback を可能にするため、トランザクションごとに維持される – 該当トランザクションは必ず commit または roll back される。</p>	
5. レスポンス作成	NR_BPM_receiveRCI.Design_Server:5_sendResponse	Yes	レスポンス XML の作成 TN にポストカードを送信	1. NR_Common.outbound: processGDSNReponse 2. NR_Common.utils:createSBDHHeader
6. レスポンスの送信方法を取得	NR_BPM_receiveRCI.Design_Server:6_getDeliveryMethod	Yes	1. TN からレスポンスドキュメントを要求する (EANUCC 2.1 GDSN レスポンス) . 2. TN パートナープロフィールから送信方法を取得する	1. NR_Utility.tn: getPreferredDestination
7a. AS2 でレスポンス送信	NR_BPM_receiveRCI.Design_Server:7a_sendAS2	Yes	1. AS2 でレスポンスを送信する	1. NR_Utility.tn: getAS2Params 2. wm.EDIINT:send
7b. SMTP でレスポンス送信	NR_BPM_receiveRCI.Design_Server:7b_sendSMTP	Yes	1. SMTP でレスポンスを送信する	1. NR_Utility.email: sendEmailWithAttachments
8. MDN 受信	NR_BPM_receiveRCI.Design_Server:8_MDN_RECEIVE	NA	1. TN からの MDN ドキュメントを要求する (EDIINT MDN)	1. wm.EDIINT.rules: processMDN
9. 終了	NR_BPM_receiveRCI.Design_Server:9_End	No	プロセスの終了	
エラーのキャッチ	NR_BPM_receiveRCI.Design_Server:Catch_All_Errors	Yes		1. NR_Utility.exception: errorHandler

10 別紙 E: CIS の受信と処理

受信した CIS メッセージの処理プロセスは、CIS の受信、パーシング、バリデーション、NR_IN_TRANS テーブルへのサブスクリプション情報の挿入、レスポンスの生成、AS2 または SMTP を使ったデータプールへの CIS 転送が含まれる。

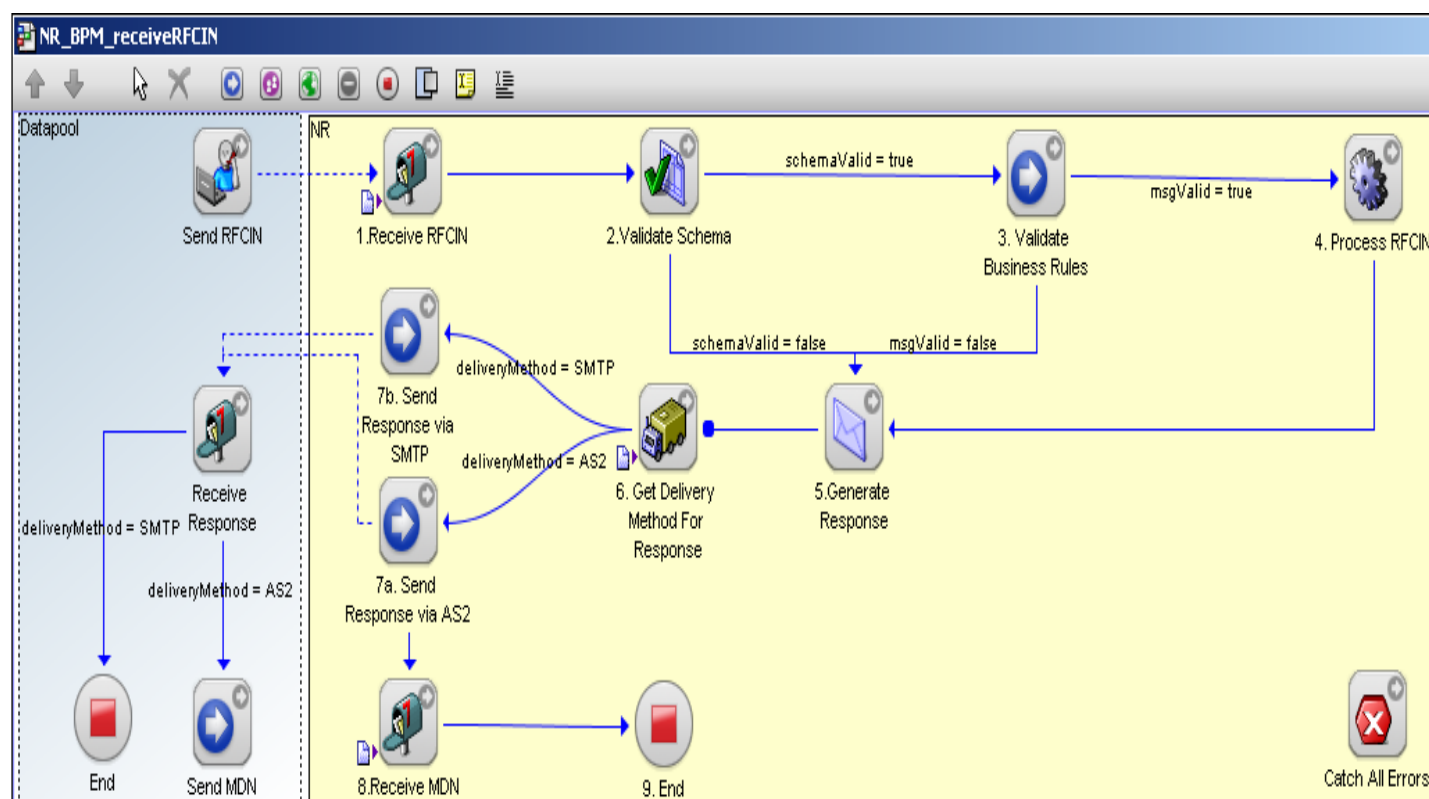


BPM ステップ名	サービス名	再送	説明	依存サービス
1. CIS の受信	NR_BPM_CIS_RECEIVE.Design_Server: 1_CIS の受信	Yes	1. TN からドキュメントを取得する 2. メッセージレベルの情報を取得し、NR_TRNS_HIST に挿入する 3. ディスクに保存	1. NR_Common.services: getInactiveValsForPartner 2. NR_Common.utils: persistToDisk
2. スキーマバリデーション	NR_BPM_CIS_RECEIVE.Design_Server: 2_validateXML	Yes	1. Xerces バリデーションエンジンを使い、メッセージを GDSN スキーマに対してチェックする	1.NR_XMLValidation: xercesValidate
3. ビジネスバリデーションの実行	NR_BPM_CIS_RECEIVE.Design_Server: 3_validateMsg	Yes	1. メッセージバリデーション	1.NR_Validation.gdsn.msg: validateGDSNMessage
4. CIS の処理	NR_BPM_CIS_RECEIVE.Design_Server: 4_processTransaction	Yes	1. トランザクションごとにプロセスを繰り返す。 2. トランザクションレベルのバリデーションを行う コマンドレベルのバリデーションを行う アイテムレベルのバリデーションを行う NR_IN_TRANS テーブルに挿入する 3. GDSN レスポンス・GDSN エクセプションを作成する	

BPM ステップ名	サービス名	再送	説明	依存サービス
			注：データベースセッションは、commit、rollback を可能にするため、トランザクションごとに維持される - 該当トランザクションは必ず commit または roll back される。	
5. レスポンス作成	NR_BPM_CIS_RECEIVE.Design_Server:5_sendResponse	Yes	1. レスポンス XML の作成 2. TN にポストカードを送信	1. NR_Common.outbound:processGDSNReponse 2. NR_Common.utils:createSBDHHeader
6. レスポンスの送信方法を取得	NR_BPM_CIS_RECEIVE.Design_Server:6_getDeliveryMethod	Yes	1. TN からレスポンスドキュメントを要求する (EANUCC 2.1 GDSN レスポンス) . 2. TN パートナープロフィールから送信方法を取得する	1. NR_Uilities.tn:getPreferredDestination
7a. AS2 でレスポンス送信	NR_BPM_CIS_RECEIVE.Design_Server:7a_sendAS2	Yes	1. AS2 でレスポンスを送信する	1. NR_Uilities.tn:getAS2Params 2. wm.EDIINT:send
7b. SMTP でレスポンス送信	NR_BPM_CIS_RECEIVE.Design_Server:7b_sendSMTP	Yes	1. SMTP でレスポンスを送信する	1. NR_Uilities.email:sendEmailWithAttachments
8. MDN 受信	NR_BPM_CIS_RECEIVE.Design_Server:8_MDN_RECEIVE	NA	1. TN からの MDN ドキュメントを要求する(EDIINT MDN)	1. wm.EDIINT.rules:processMDN
9. 終了	NR_BPM_CIS_RECEIVE.Design_Server:9_End	No	プロセスの終了	
エラーのキャッチ	NR_BPM_CIS_RECEIVE.Design_Server: Catch_All_Errors	Yes		1. NR_Uilities.exception:errorHandler

11 別紙 F: RFCIN の受信と処理

RFCIN の受信処理は、RFCIN メッセージの受信、パーシング、バリデーション、サブスクリプション情報の NR_IN_TRANS テーブルへの挿入、レスポンスの作成、AS2 または SMTP を使ってデータプールに送信するプロセスを含む。



BPM ステップ名	サービス名	再送	説明	依存サービス
1. RFCIN 受信	NR_BPM_receiveRFCIN.Design_Server: 1_receiveRFCIN	Yes	1. TN からドキュメントを取得する 2. メッセージレベルの情報を取得し、NR_TRNS_HIST に挿入する 3. ディスクに保存	1. NR_Common.services: getInactiveValsForPartner 2. NR_Common.utils: persistToDisk
2. スキーマバリデーション	NR_BPM_receiveRFCIN.Design_Server: 2_validateXML	Yes	1. Xerces バリデーションエンジンを使い、メッセージを GDSN スキーマに対してチェックする	1.NR_XMLValidation:xercesValidate
3. ビジネスバリデーションの実行	NR_BPM_receiveRFCIN.Design_Server: 3_validateMsg	Yes	1. メッセージバリデーション	1.NR_Validation.gdsn.msg:validateGDSNMessage

BPM ステップ名	サービス名	再送	説明	依存サービス
4. RFCIN 処理	NR_BPM_receiveRFCIN.Design_Server:4_processTransaction	Yes	<ol style="list-style-type: none"> 1. トランザクションごとにプロセスを繰り返す 2. トランザクションレベルのバリデーションを行う コマンドレベルのバリデーションを行う アイテムレベルのバリデーションを行う NR_IN_TRANS テーブルに挿入する 3. GDSN レスポンス・GDSN エクセプションを作成する <p>注：データベースセッションは、commit、rollback を可能にするため、トランザクションごとに維持される – 該当トランザクションは必ず commit または rollback される。</p>	
5. レスポンス作成	NR_BPM_receiveRFCIN.Design_Server:5_sendResponse	Yes	レスポンス XML の作成 TN にポストカードを送信	<ol style="list-style-type: none"> 1. NR_Common.outbound:processGDSNReponse 2. NR_Common.utils:createSBDHHeader
6. レスポンスの送信方法を取得	NR_BPM_receiveRFCIN.Design_Server:6_getDeliveryMethod	Yes	<ol style="list-style-type: none"> 1. TN からレスポンスドキュメントを要求する (EANUCC 2.1 GDSN レスポンス) . 2. TN パートナープロフィールから送信方法を取得する 	1. NR_Uilities.tn: getPreferredDestination
7a. AS2 でレスポンス送信	NR_BPM_receiveRFCIN.Design_Server:7a_sendAS2	Yes	1. AS2 でレスポンスを送信する	<ol style="list-style-type: none"> 1. NR_Uilities.tn: getAS2Params 2. wm.EDIINT:send
7b. SMTP でレスポンス送信	NR_BPM_receiveRFCIN.Design_Server:7b_sendSMTP	Yes	1. SMTP でレスポンスを送信する	1. NR_Uilities.email: sendEmailWithAttachments
8. MDN 受信	NR_BPM_receiveRFCIN.Design_Server:8_MDN_RECEIVE	NA	1. TN からの MDN ドキュメントを要求する (EDIINT MDN)	1. wm.EDIINT.rules: processMDN
9. 終了	NR_BPM_receiveRFCIN.Design_Server:9_End	No	プロセスの終了	
エラーのキャッチ	NR_BPM_receiveRFCIN.Design_Server: Catch_All_Errors	Yes		1. NR_Uilities.exception: errorHandler

12 別紙 G: データベースプロセス図およびテーブルレイアウト

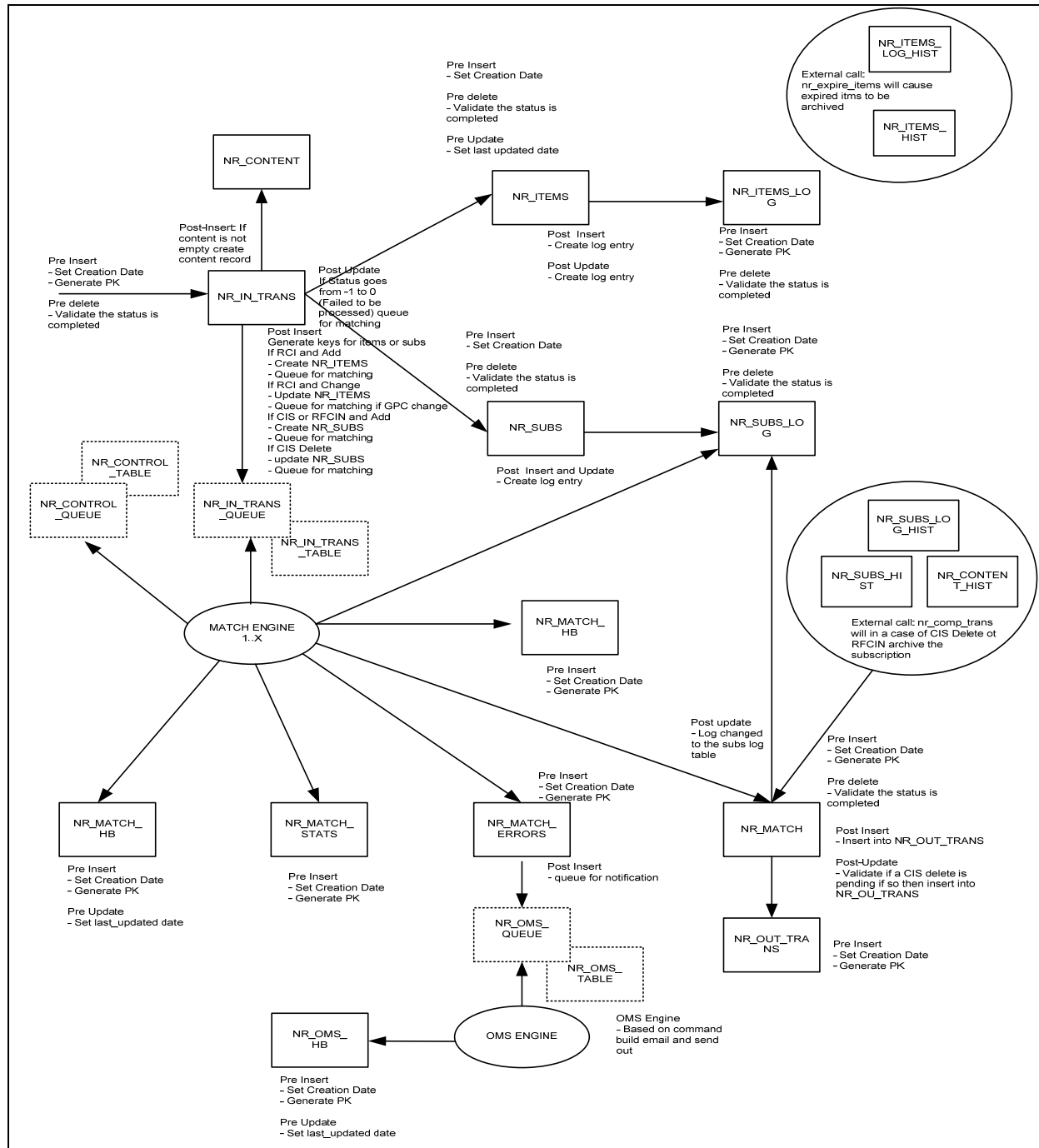
12.1 オラクルデータベース

通常のオラクル DB インスタンスは、サービスプロセス、SGA(Shared Global Area/Memory)、バックグラウンドプロセス、disk storage に構成される。RCI、CIS、RFCIN トランザクションは、専用または共有サーバープロセス経由で、TNS リスナーによってデータベースに挿入される。マッチングプログラムは、Job キューのサーバープロセスによって実行される。SGA またはメモリーは、実際のクエリーを最適化する。1 ギガ以上のメモリーを割り当てること、ほとんどのマッチングクエリーは、disk storage を使うことなく shared memory の中で行われる。



12.2 データベースプロセス図

下記の図は、送受信アイテム、サブスクリプションの処理プロセスと、マッチングエンジンの働きを表している。



12.3 データベーステーブルおよびレイアウト

NR_SUBS

このテーブルは、active もしくは inactive の受信済みのサブスクリプションを保存する。また、RFCIN の情報をテンポラリ情報として保持する。T

項目	タイプ	参照	必須	コメント
ID	NUMBER	PK	Y	NR_SUBS_SEQ シーケンスからのシステム生成番号
CREATION_DATE	DATE		Y	データベースレコードの作成日。
DELETED_DATE	DATE			ステータスが inactive になった日付
ACTIVE	NUMBER(1)		Y	0=TRUE, 1=FALSE
STATUS	NUMBER(1)		Y	0 = permanent - CIS 1 = temporary - RFCIN
DOCUMENT_ID	VARCHAR(80)		N	XML ファイルのドキュメント ID
TRANSACTION_ID	VARCHAR(80)		N	XML ファイルのトランザクション ID
CIS_ID	VARCHAR(80)		N	受信ファイルの Creator Identification
COMMAND_ID	VARCHAR(80)		N	受信ファイルのコマンド ID
UNIQ_ID	NUMBER		N	RFCIN のユニーク ID (pk と同じ) これによって複数の RFCIN を同時処理できる
GLN_RETAILER	VARCHAR(13)		Y	RDP の GLN
GLN_SUPPLIER	VARCHAR(13)		N	SDP の GLN (マッチしたもの)
GTIN	VARCHAR	FK	N	GTIN が IR_ITEM_ID を参照しているが、アイテムが存在しないかも知れないため、まだ FK ではない
TMCC	VARCHAR(5)		N	対象市場国コード
TMSC	VARCHAR(6)		N	対象市場国地域コード
GPC	VARCHAR(14)		N	GPC/JICFS

インデックス

NR_SUBS_PK (ID)

NR_SUBS_IDX1 (GTIN, TMCC, TMSC, GPC, ACTIVE, STATUS, UNIQ_ID)

アクション

Post Insert	NR_SUBS_LOG に初期エントリーを作成する
Post update of active status to FALSE	Deleted_date をシステム日付にセットし、NR_SUBS_LOG にエントリーを作成する s
Pre delete	アクティブステータスが FALSE であることを確認する

NR_SUBS_LOG

このテーブルは、NR_SUBS に対するアクションのヒストリーログを保存する。

項目	タイプ	参照	必須	コメント
ID	NUMBER	PK	Y	NR_SUBS_LOG_SEQ シーケンスからのシステム作成番号
CREATION_DATE	DATE		Y	データベースレコードの作成日。
TRANSACTION_ID	VARCHAR(80)		N	XML ファイルのトランザクション ID
DOCUMENT_ID	VARCHAR(80)		Y	XML ファイルのドキュメント ID
NR_SUBS_ID	NUMBER	FK	Y	NR_SUBS_PK への参照
ACTION	VARCHAR(10)		Y	CREATION, INACTIVE, INFO
TEXT	VARCHAR(128)		N	アクションに関するコメント用フリーテキスト

インデックス

NR_SUBS_LOG_PK (ID)

Constraints

FK Cascade delete based on NR_SUBS

NR_SUBS_HIST

取り込まれた NR_SUBS テーブルの情報を保存する。

項目	タイプ	参照	必須	コメント
ID	NUMBER	PK	Y	NR_SUBS_SEQ シーケンスからのシステム生成番号
CREATION_DATE	DATE		Y	データベースレコードの作成日。
DELETED_DATE	DATE			ステータスが inactive になった日付
ACTIVE	NUMBER(1)		Y	0=TRUE, 1=FALSE
STATUS	NUMBER(1)		Y	0 = permanent 1 = temporary
DOCUMENT_ID	VARCHAR(80)		N	XML ファイルのドキュメント ID
TRANSACTION_ID	VARCHAR(80)		N	XML ファイルのトランザクション ID
CIS_ID	VARCHAR(80)		N	受信ファイルの Creator Identification
COMMAND_ID	VARCHAR(80)		N	受信ファイルのコマンド ID
GLN_RETAILER	VARCHAR(13)		Y	RDP の GLN
GLN_SUPPLIER	VARCHAR(13)		N	SDP の GLN (マッチしたもの)
GTIN	VARCHAR	FK	N	GTIN が IR_ITEM_ID を参照しているが、アイテムが存在しないかも知れないため、まだ FK ではない
TMCC	VARCHAR(5)		N	対象市場国コード
TMSC	VARCHAR(6)		N	対象市場国地域コード
GPC	VARCHAR(14)		N	GPC/JICFS

インデックス

NR_SUBS_HIST_PK (ID)

NR_SUBS_HIST_IDX1 (GTIN, TMCC, TMSC, GPC)

NR_SUBS_LOG_HIST

このテーブルは、NR_SUBS_HIST テーブルに対するアクションのヒストリーログを保持する。アーカイブされたサブスクリプションに対して適用される。

項目	タイプ	参照	必須	コメント
ID	NUMBER	PK	Y	NR_SUBS_LOG_SEQ シーケンスからのシステム作成番号
CREATION_DATE	DATE		Y	データベースレコードの作成日。
TRANSACTION_ID	VARCHAR(80)		N	XML ファイルのトランザクション ID
DOCUMENT_ID	VARCHAR(80)		Y	XML ファイルのドキュメント ID
NR_SUBS_ID	NUMBER	FK	Y	NR_SUBS_PK への参照
ACTION	VARCHAR(10)		Y	CREATION, INACTIVE, INFO
TEXT	VARCHAR(128)		N	アクションに関するコメント用フリーテキスト

インデックス

NR_SUBS_LOG_HIST_PK (ID)

Constraints

FK Cascade delete based on NR_SUBS

NR_ITEMS

このテーブルは、アイテム情報（永久版）を保持する。

項目	タイプ	参照	必須	コメント
ID	NUMBER	PK	Y	シーケンスから生成されるシステム生成番号 NR_ITEMS_SEQ
CREATION_DATE	DATE		Y	データベースレコードの作成日。
UPDATED_DATE	DATE		N	文字列項目 (description) が変更された日付
DELETED_DATE	DATE		N	ステータスが inactive になった日付
CANCEL_DATE	DATE		N	このアイテムがキャンセルされる日付
DISCONTINUE_DATE	DATE		N	このアイテムが停止される日付
ACTIVE	NUMBER(1)		Y	TRUE =サブスクリプションが inactive FALSE =期限切れ. 0=TRUE 1=FALSE
STATUS	NUMBER(1)		Y	0 = permanent 1 = temporary
TRANSACTION_ID	VARCHAR(80)		N	XML ファイルのトランザクション ID
DOCUMENT_ID	VARCHAR(80)		Y	XML ファイルのドキュメント ID
GLN_SUPPLIER	VARCHAR(14)		Y	GLN
GTIN	VARCHAR	FK	Y	GTIN
TMCC	VARCHAR(5)		Y	対象市場国コード
TMSC	VARCHAR(6)		Y	対象市場国地域コード
GPC	VARCHAR(14)		Y	GPC/JICFS

インデックス

NR_ITEMS_PK (ID)

NR_ITEMS_IND1 (GTIN, GLN , TMCC, TMSC, GPC)

アクション

Post Insert	NR_ITEMS_LOG に初期エントリーを作成する
Post update of active status to FALSE	Deleted_date をシステム日付に設定し、NR_SUBS_LOG にエントリーを作成する
Pre delete	アクティブステータスが FALSE であることを確認する
pre update	いくつかの項目に対してアップデートログを保持する (GLN、GTIN、TMCC、TMSC、CANCEL_DATE、DELETED_DATE)
Pre-update	更新日をシステム日付にセットする

NR_ITEMS_LOG

このテーブルは、NR_ITEMS テーブルに対するアクションのヒストリーログを保存する。

項目	タイプ	参照	必須	コメント
ID	NUMBER	PK	Y	シーケンスから生成されるシステム生成番号 NR_ITEMS_LOG_SEQ
DATED	DATE		Y	データベースレコードの作成日。
NR_ITEMS_ID	NUMBER	FK	Y	NR_ITEMS_PK への参照
ACTION	VARCHAR(10)		Y	CREATION, INACTIVE, INFO
TRANSACTION_ID	VARCHAR(80)		N	XML ファイルのトランザクション ID
DOCUMENT_ID	VARCHAR(80)		Y	XML ファイルのドキュメント ID
TEXT	VARCHAR(128)		N	アクションに関するコメント用フリーテキスト

インデックス

NR_ITEMS_LOG_PK (ID)

Constraints

FK Cascade delete based on NR_ITEMS

NR_ITEMS_HIST

このテーブルは、アーカイブされた NR_ITEMS を保存する。

項目	タイプ	参照	必須	コメント
ID	NUMBER	PK	Y	シーケンスから生成されるシステム生成番号 NR_ITEMS_SEQ
CREATION_DATE	DATE		Y	データベースレコードの作成日。
UPDATED_DATE	DATE		N	文字列項目 (description) が変更された日付
DELETED_DATE	DATE		N	ステータスが inactive になった日付
CANCEL_DATE	DATE		N	アイテムがキャンセルされる日付
DISCONTINUE_DATE	DATE		N	アイテムが停止される日付
ACTIVE	NUMBER(1)		Y	TRUE =サブスクリプションが inactive FALSE =サブスクリプションが期限切れ。 0=TRUE 1=FALSE
STATUS	NUMBER(1)		Y	0 = permanent 1 = temporary
TRANSACTION_ID	VARCHAR(80)		N	XML ファイルのトランザクション ID
DOCUMENT_ID	VARCHAR(80)		Y	XML ファイルのドキュメント ID
GLN_SUPPLIER	VARCHAR(14)		Y	GLN
GTIN	VARCHAR	FK	Y	GTIN
TMCC	VARCHAR(5)		Y	対象市場国コード
TMSC	VARCHAR(6)		Y	対象市場国地域コード
GPC	VARCHAR(14)		Y	GPC/JICFS

インデックス

NR_ITEMS_HIST_PK (ID)

NR_ITEMS_HIST_INDX1 (GTIN, GLN , TMCC, TMSC, GPC)

NR_ITEMS_LOG_HIST

NR_ITEMS_HIST テーブルのヒストリーログを保存する。このテーブルの項目はアーカイブされる。

項目	タイプ	参照	必須	コメント
ID	NUMBER	PK	Y	シーケンスから生成されるシステム生成番号 NR_ITEMS_LOG_SEQ
DATED	DATE		Y	データベースレコードの作成日。
NR_ITEMS_ID	NUMBER	FK	Y	NR_ITEMS_PK への参照
ACTION	VARCHAR(10)		Y	CREATION, INACTIVE, INFO
TRANSACTION_ID	VARCHAR(80)		N	XML ファイルのトランザクション ID
DOCUMENT_ID	VARCHAR(80)		Y	XML ファイルのドキュメント ID
TEXT	VARCHAR(128)		N	アクションに関するコメント用フリーテキスト

インデックス

NR_ITEMS_LOG_HIST_PK (ID)

Constraints

FK Cascade delete based on NR_ITEMS

NR_IN_TRANS

このテーブルは、RCI、CIS、RFCIN のトランザクションを保存する。

項目	タイプ	参照	必須	コメント
ID	NUMBER	PK	Y	シーケンスから生成されるシステム生成番号 NR_IN_TRANS_SEQ
CREATION_DATE	DATE		Y	データベースレコードの作成日。
COMPLETION_DATE	DATE		N	トランザクションが完了した日付(例: マッチの完了)
CANCEL_DATE	DATE		N	アイテムがキャンセルになった日
DISCONTINUE_DATE	DATE		N	アイテムが停止になった日
TRANSACTION_タイプ	NUMBER		Y	0 = RCI 1 = CIS 2 = RFCIN
ACTION	VARCHAR(1)		Y	A = ADD C=UPDATE D=DELELET
STATUS	NUMBER(1)		Y	0 = 作成されたが、マッチはされていない 1 = 完了 -1 = 失敗
GLN_RETAILER	VARCHAR(13)		Y	Content owner データプールの GLN

GLN_SUPPLIER	VARCHAR(13)		N	CIS の受信者 GLN (マッチングのみ使用)
GTIN	VARCHAR		N	GTIN
TMCC	VARCHAR(5)		N	対象市場国コード
TMSC	VARCHAR(6)		N	対象市場国地域コード
GPC	VARCHAR(14)		N	GPC/JICFS
DOCUMENT_ID	VARCHAR(80)		Y	XML ファイルのドキュメント ID
TRANSACTION_ID	VARCHAR(80)		N	XML ファイルのトランザクション ID
CIS_ID	VARCHAR(80)		N	受信ファイルの Creator Identification
COMMAND_ID	VARCHAR(80)		N	受信ファイルのコマンド ID
REF_ID	NUMBER		N	Nt_items、または nr_subs の PK への参照 (トランザクションタイプによる)
CONTENT	BLOB		N	XML メッセージ

インデックス

NR_IN_TRANS_PK (ID)

NR_IN_TRANS_INDX1(GLN,GTIN)

アクション

Post Insert	<p>タイプとアクションによって以下の処理が行われる。</p> <p>RCI and I = NR_ITEMS を作成 (マッチのためのキュー)</p> <p>RCI and U = NR_ITEMS をアップデート (マッチのためのキュー)</p> <p>RCI and D = NR_ITEMS を inactive にする</p> <p>CIS and I = NR_SUBS を作成 (マッチのためのキュー)</p> <p>CIS and D = NR_SUBS を inactive にする</p> <p>RFCIN and I = NR_SUBS を作成 (マッチのためのキュー)</p> <p>RFCIN and D = NR_SUBS を inactive にする</p> <p>キューは NR_IN_TRANS_AQ に対して行われる</p>
Pre delete	ステータスが完了であることを検証する

NR_CATEGORY

GPC/JICFS の参照テーブル

項目	タイプ	参照	必須	コメント
CATEGORY	VARCHAR(14)		Y	GPC または JICFS コード
DESCRIPTION	VARCHAR(80)		N	カテゴリの説明
CREATION_DATE	DATE		Y	レコード作成の日付・時間（トリガーによる）

インデックス

NR_JICFS_PK(JICFS)

アクション

PRE_INSERT 作成日をシステム日付に設定

NR_CONFIG

設定のためのさまざまなパラメータを保存するテーブル。

項目	タイプ	参照	必須	コメント
CONFIG_OWNER	VARCHAR2(32)		Y	設定が適用されるサブシステム
CONFIG_NAME	VARCHAR2(32)		Y	設定の ID
CONFIG_VALUE	VARCHAR2(80)			設定の値

NR_PROFILES

企業情報を保存するためのテーブル。

項目	タイプ	参照	必須	コメント
ID	NUMBER	PK	Y	シーケンスから生成されるシステム生成番号 NR_PROFILES_SEQ
GLN	VARCHAR(13)		Y	パーティ GLN
ROLE	NUMBER		Y	パーティロール 1=小売/ 2= メーカー/ 3=卸 4=データプール
ASSOCIATED_DP	VARCHAR(13)			所属データプール GLN
CORP_Name	VARCHAR(128)		Y	データソース (content owner) の企業名
GLN_PARENT	VARCHAR(13)			
STATUS	NUMBER(1)		Y	0 = Inactive 1 = Active
ADDRESS	VARCHAR(254)			パーティの住所
CITY	VARCHAR(50)			市町村

STATE	VARCHAR(50)			都道府県
ZIP	VARCHAR(15)			郵便番号
COUNTRY	VARCHAR(50)			国
PO_BOX_NO	VARCHAR(80)			PO ボックス番号
COUNTRY_CODE	VARCHAR(5)			国コード
LANG_CODE	VARCHAR(7)			言語コード
PHONE	VARCHAR(20)			電話番号
FAX	VARCHAR(20)			ファックス番号
EMAIL	VARCHAR(50)			メールアドレス
CREATION_DATE	DATE		Y	データベースレコードの作成日.
UPDATED_DATE	DATE		Y	データベースレコードの作成日.
DELETED_DATE	DATE			ステータスが inactive になった日付

インデックス

NR_PROFILES_PK(ID)
NR_PROFILES_INDX1(GLN)

アクション

PRE_INSERT 作成日を sysdate に設定し、ID が存在しない場合 ID を作成する
PRE_UPDATE 更新日を sysdate に設定する

NR_COUNTRY_CODES

国コードを保存するためのテーブル.

項目	タイプ	参照	必須	コメント
TMCC_CODE	VARCHAR(5)		Y	対象市場国コード
TMCC_DESC	VARCHAR(50)			対象市場国名
TMSC_CODE	VARCHAR(6)		Y	対象市場国地域コード
TMCC_DESC	VARCHAR(50)			対象市場国地域名
CREATION_DATE	DATE		Y	レコード作成日

インデックス

NR_COUNTRY_CODES_PK(TMCC, TMSC)

アクション

PRE_UPDATE 更新日を sysdate に設定する

NR_MATCH

送信されたすべてのマッチング結果を保存するテーブル。

Description	タイプ	参照	必須	コメント
ID	NUMBER		Y	内部キー
NR_SUBS_ID	NUMBER	FK	Y	サブスクリプションへの参照
ASSOCIATED_DP	VARCHAR(13)		Y	Sub が送信されたデータプールの GLN
STATUS	NUMBER		Y	0 CIS/RFCIN (ADD) の送信が必要 1 CIS/RFCIN (ADD) 送信済み 2 CIS (Delete) 送信必要 3 CIS(Delete) 送信必要
STATUS_ADD	NUMBER		Y	0 CIS/RFCIN (ADD) 送信必要 1 CIS/RFCIN (ADD)
STATUS_DELETE	NUMBER		Y	-1 削除なし 0 CIS (削除) の送信が必要 1 CIS(削除) の送信が必要
ACTIVE	NUMBER		Y	0=active 1=inactive
TRANSACTION_ID	VARCHAR(80)		N	XML ファイルのトランザクション ID
DOCUMENT_ID	VARCHAR(80)		Y	XML ファイルのドキュメント ID
CREATION_DATE	DATE		Y	レコード作成日

インデックス

NR_MATCH_PK(ID)

NR_MATCH_INDX1(NR_SUBS_ID, ASSOCIATED_DP) unique

アクション

POST_INSERT 送信用トランザクションテーブルにエントリーを挿入する。(NR_OUT_TRANS Insert)

PRE_INSERT 作成日をシステム日付に設定

NR_MATCH_TRACE

送信済みのすべてのマッチ結果を保存するテーブル

項目	タイプ	参照	必須	コメント
ID	NUMBER		Y	内部キー
CREATION_DATE	DATE		Y	レコード作成日
NAME	VARCHAR(15)			マッチングエンジン名 (複数存在する場合)
TEXT	VARCHAR(80)			トレースのためのフリーテキスト

インデックス

NR_MATCH_TRACE_PK(ID)

アクション

PRE_INSERT 作成日をシステム日付に設定し、PK キーを作成する

NR_MATCH_HB

マッチエンジンの HEARTBEAT を管理するテーブル。稼働中のすべてのエンジンはこのテーブルの中にエントリーを持ち、60 秒ごとにアップデートされる（デフォルトは nt_config MATCH、HEARTBEAT のエントリーのよって変更可能）。AQ をブロックするため、時間はキューの待ち時間により変わる可能性がある。（デフォルトは 10 秒だが、nr_config MATCH、QUEUE_WAIT によって設定可能である）。

項目	タイプ	参照	必須	コメント
CREATION_DATE	DATE		Y	レコード作成日
UPDATED_DATE	DATE			レコード更新日
LAST_STARTED	DATE			現在使用中のエンジンが起動した日付
START_COUNT	NUMBER			最初にレコードを作成してから、エンジンが起動した回数
NAME	VARCHAR(15)		Y	複数エンジンが存在する場合、該当するエンジンの名前
STATUS	VARCHAR(15)		N	稼働中のエンジンのステータス RUNNING: 稼働中 STOP: エンジンが正常に停止している DOWN: エンジンが HEARTBEAT に対してリトライ時間（デフォルト 3 * 60）の間にレポートしていない上、 STOP 状態ではない（match_control が稼働中の場合のみ起こりうる） TRACE: エンジンがトレースモードである SUSPEND: エンジンがサスペンド状態である TRACE/SUSPEND: エンジンがトレースモード状態でサスペンドされている

インデックス

NR_MATCH_HB_PK(名前)

アクション

PRE_INSERT 作成日をシステム日付に設定
 PRE__UPADATE 更新日をシステム日付に設定

NR_MATCH_STATS

マッチエンジンのすべてのマッチに対しての統計情報を管理するテーブル

項目	タイプ	Ref	必須	コメント
CREATION_DATE	DATE		Y	レコード作成日
BEGIN_DATE	DATE		Y	マッチトランザクションが開始された日付・時間
END_DATE	DATE		Y	マッチトランザクションが終了した日付・時間
NR_IN_TRANS_ID	NUMBER		Y	該当トランザクションを処理した in_trans レコードへの参照
TRANSACTION_TYPE	NUMBER		Y	統計情報が記録される該当トランザクショ

				ンタイプ
TOTAL_COUNT	NUMBER		Y	マッチの総数
DUPLICATE_COUNT	NUMBER		Y	重複マッチの総数

アクション

PRE_INSERT 作成日をシステム日付に設定

NR_MATCH_ERRORS

マッチエンジンのエラーで、処理されていないものを管理するテーブル

項目	タイプ	参照	必須	コメント
ID	NUMBER		Y	シーケンスから生成されるシステム生成番号 NR_MATCH_ERRORS_SEQ
CREATION_DATE	DATE		Y	レコード作成日
ERR_NUM	VARCHAR(10)		Y	オラクル SQLCODE
ERR_MSG	VARCHAR(200)		Y	オラクルエラーメッセージ
NR_IN_TRANS_ID	NUMBER		N	失敗したトランザクション の参照
NAME	VARCHAR(15)		N	エラーが発生したサブシステム

アクション

PRE_INSERT 作成日をシステム日付に設定

NR_OMS_HB

OMS エンジン Heartbeat を管理するテーブル。すべての稼働中のエンジンは、このテーブルにエントリー情報を持ち、60 秒ごとにアップデートされる。（nr_config OMS、HEARTBEAT のエントリーによってデフォルト設定を変更できる）

項目	タイプ	参照	必須	コメント
CREATION_DATE	DATE		Y	レコード作成日
UPDATED_DATE	DATE			レコード更新日
LAST_STARTED	DATE			エンジンが起動した日付
START_COUNT	NUMBER			レコードが最初に作成されてからエンジンが起動した回数
NAME	VARCHAR(15)		Y	マッチングエンジン名（複数存在する場合）

インデックス

NR_OMS_HB_PK(Name)

アクション

PRE_INSERT 作成日をシステム日付に設定
PRE__UPADATE 更新日をシステム日付に設定

NR_OUT_TRANS

パラメータ設定値を保存するテーブル。

項目	タイプ	参照	必須	コメント
ID	NUMBER		Y	
NR_MATCH_ID	NUMBER	FK	Y	
ACTION	VARCHAR(1)		Y	A = CIS 新規追加 D = CIS 削除
CREATION_DATE	DATE		Y	レコード作成日

インデックス

NR_OUT_TRANS_PK(ID)

アクション

PRE_INSERT 作成日をシステム日付に設定

NR_TRANS_HIST

webMethods のメッセージ処理のログテーブル

Description	タイプ	参照	必須	コメント
DOCUMENT_ID	VARCHAR(80)		Y	XML ファイルのドキュメント ID
TRANSACTION_ID	VARCHAR(80)		N	XML ファイルのトランザクション ID
TOTAL_COMMANDS	NUMBER			メッセージ、トランザクション内のコマンド総数
TOTAL_GTIN	NUMBER			メッセージ、トランザクション内の GTIN 総数
STATUS	VARCHAR(24)		Y	エラーまたは成功コード
ASSOCIATED_DP	VARCHAR(13)			ファイル名作成に使用する
FILE_SUFFIX	VARCHAR(24)			ファイル作成に使うタイムスタンプ
COMMAND_TYPE	VARCHAR(24)			トランザクション内のコマンドタイプ
CREATION_DATE	DATE		Y	レコード作成日

インデックス

NR_TRANS_HIST_INDX01(DOCUMENT_ID, ASSOCIATED_DP)

アクション

PRE_INSERT 作成日をシステム日付にセットしてキーを生成する。

NR_CONTENT

In_trans トランザクションに関連する XML メッセージ

Description	タイプ	参照	必須	コメント
ID	NUMBER		Y	内部キー
TRANSACTION_タイプ	VARCHAR(80)		Y	トランザクションのタイプ
CREATION_DATE	DATE		Y	レコード作成日
REF_ID	NUMBER		Y	サブスクリプション、またはアイテムキーの参照
ACTION	VARCHAR(1)		Y	A/C/D トランザクション (nr_in_trans と同じ)
CONTENT	BLOB		N	XML メッセージ
タイプ	NUMBER(2)		N	メッセージタイプ (処理されたメッセージを特定するため)

インデックス

NR_CONTENT_PK (ID)

NR_CONTENT_INDX1(REF_ID, TRANSACTION_TYPE, ACTION)

アクション

PRE_INSERT 作成日をシステム日付にセットしてキーを生成する。

NR_CONTENT_HIST

In_trans のトランザクションヒストリー情報に関する XML メッセージ

項目	タイプ	参照	必須	コメント
ID	NUMBER		Y	内部キー
TRANSACTION_TYPE	VARCHAR(80)		Y	トランザクションタイプ
CREATION_DATE	DATE		Y	レコード作成日
REF_ID	NUMBER		Y	サブスクリプション、またはアイテムキーの参照
ACTION	VARCHAR(1)		Y	A/C/D トランザクション (nr_in_trans と同じ)
CONTENT	BLOB		N	XML メッセージ
タイプ	NUMBER(2)		N	メッセージタイプ (処理されたメッセージを特定するため)

インデックス

NR_CONTENT_PK (ID)

NR_CONTENT_INDX1(REF_ID, TRANSACTION_タイプ, ACTION)

アクション

PRE_INSERT 作成日をシステム日付にセットし、キーを作成する

12.4 アドバンスキュー

マッチングエンジンは、2種類のキューを使い、それぞれ違う目的で使用される。最初のキューは、nr_in_trans_queue (テーブルは nr_in_trans_table)、これはトランザクションのマッチングに使われる。このキューのタイプである nr_in_trans_msg_type は Rowid と nr_in_trans_id で構成され、両方とも nr_in_trans_table を指している。

2番目のキューは、nr_control_queue (テーブルは nr_control_table) で、これはマッチングエンジンのコントロールキューとして使われ、タイプは nr_control_type (name, command) である。

キュー	テーブル	タイプ
NR_IN_TRANS_QUEUE	NR_IN_TRANS_TABLE	NR_IN_TRANS_MSG_TYPE
NR_CONTROL_QUEUE	NR_CONTROL_TABLE	NR_CONTROL_TYPE

コントロールキューは、マッチングエンジンにメッセージを送るために使われ、将来機能強化されれば、特定のキューにメッセージを送ることも可能であるが、現状は稼働中のマッチングエンジンの数の分メッセージを送ることになっている。この処理は、過去 2 分間の HEARTBEAT の受信状況で決まる。

タイプ	説明	実行
STOP	エンジンの停止	Nr_match_engine.stop_all
TRACE	エンジンをトレースする	Nr_match_engine.trace_all_on
STOP_TRACE	エンジンのトレースを停止する	Nr_match_engine.trace_all_off
SUSPEND	マッチ機能を停止する (シャットダウンはしない)	Nr_match_engine.suspend_all
RESUME	マッチ機能を再開させる (機能停止後)	Nr_match_engine.resume_all

タイプ	名前	値	デフォルト	説明
MATCH	QUEUE_WAIT	10	10	マッチングエンジンがメッセージをキューから取り出すまでの待ち時間。
MATCH	AQ_TRACE	N	N	エンジンが自らをトレースする必要がある場合はデフォルト設定される。この設定は、ランタイム上メッセージを送信することによって変更できる。(上記メッセージを参照)
MATCH	HISTORICAL_DUPS	Y	Y	Nr_subs_logs テーブルにおける重複マッチ
<engine anme>	TRACE	N	N	In_trans_match で nr_match_engine を使用する場合のトレース。これはサポート作業のみ使われるべきもので、キューを無視して実行される。
MATCH	HEARTBEAT	60	60	HEARTBEAT をレポートするためにかかった時間 (秒単位)

OMS システムもコマンドを OMS システムに送信する時にキューを使用する。

キュー	テーブル	タイプ
NR_OMS_QUEUE	NR_OMS_TABLE	NR_OMS_MSG_タイプ

送信可能なコマンド:

タイプ	説明	実行
STOP	エンジンを停止させる	Nr_oms_engine.stop

TEST	デフォルト email アカウントにテストメッセージを送信する	Nr_oms_engine.test
NR_MATCH_ERRORS	該当するメールアカウントに、エラーIDに関連するメッセージを送信する	Nr_oms_engine.test_error(id)

12.5 HEARTBEAT

マッチングエンジンは DB の中で動いているため、稼働状況を確認することが難しく、稼働状況を確認するための装置を導入する必要がある。もっとも基本的なものは heartbeat 構造であり、これによって稼働中のすべてのエンジンが稼働し続けているか確認することができる。デフォルト設定では、エンジンは稼働状況を 60 秒+キュー解除待ち時間（デフォルト 10 秒）=70 秒ごとに nr_match_hb テーブルにレポートする。このテーブルは、プロセス開始時間（last_started_field）やレコードが最初に作成されてからのプロセス開始回数も管理する。これをモニターすることで、時間内に何回起動したかを確認できる。マッチングエンジンは、dbms jobs scheduler で起動するので、なんらかの理由でこれが停止したりプロセスが失敗すると 1 分後に再起動する。

Heartbeat は以下の設定が可能である

OWNER	名前	値	デフォルト	内容
MATCH	HEARTBEAT	60	60	The frequency the matching engine will report that they are still up in seconds

12.6 クリーンアップ

夜 1 時に、システムのいくつかの部分でキリンナップを実行するジョブが起動する。クリーンアップ作業は、nr_misc_cleanup を呼び出すことで初期化することができる。現状のクリーンアップは、以下の内容で処理される。

NR_MATCH_TRACE	日付 x より古いすべてのエントリーを NR_MATCH_TRACE テーブルから削除する
NR_MATCH_ERRORS	日付 x より古いすべてのエントリーを NR_MATCH_ERRORS テーブルから削除する
NR_IN_TRANS	日付 x より古いすべてのエントリー、ステータスが completed になっているエントリーを NR_IN_TRANS テーブルから削除する

これは NR_CONFIG テーブルを使って設定できる。

OWNER	名前	値	デフォルト	内容
CLEANUP	MATCH_TRACE	2	2	日付 x より古い情報を削除する
CLEANUP	MATCH_ERRORS	7	7	日付 x より古い情報を削除する
CLEANUP	IN_TRANS	30	30	日付 x より古い情報を削除する
CLEANUP	COMMIT	1000	1000	レコード X を削除した後コミットする

13 別紙 H: Node Iterator を使った XML 処理

webMethods IS は、大容量の XML ファイルを小さく分けて読み込むためのパブリックサービス (`pub.web.*`)を提供している。これは XML ファイルのサイズが大きいほど、処理はメモリーへの負担が大きくなるからである。

XML ファイルはストリームとして読み込まれた後、`WmPublic/pub.xmll:StringToXMLNode` サービスに移されてノードが作られる。このノードはストリームへのポインターの役割をし、`pub:web:getNodeIterator` によって `NodeIterator` が作成、提供される。

`getNodeIterator` は、オンデマンドのドキュメントのローディング、パーシングに適している。大容量、または低速のドキュメントの処理は、希望するデータを抽出できるところまでロードすることが有効的だからである

`NodeIterator` は、moving window mode により、メモリー内にあるノードが `getNextNode()`によって返された最後のノードと認識する。`getNextNode()`が呼び出されると、これまで `getNextNode()`によって返されたノードはすべて無効になり、最新のもののみ有効とみなされる。`getNextNode()`を再度実行する前に、現状のノードはすべて処理されなければならない。Moving window mode では、ドキュメント処理の際、最新のノードを保つためのメモリーしか使用しない。

14 別紙 I: JNR webMethods パッケージ一覧

XML の処理、GDSN バリデーション、DB への挿入などは、webMethods の中で記述され、パッケージとして整理されている。パッケージの中身は、ソースコードとデータベース接続、ブローカーのトリガー、ドキュメントタイプなどのコンポーネントで構成される。各パッケージは、「NR_〇〇」の形式で名前がつけられ、webMethods が提供する基本パッケージと区別される。ソースコードは webMethods のフロー言語と java で書かれている。

JNR アプリケーションをサポートするパッケージ一覧と、それぞれの内容は以下の通りである。

パッケージ名	内容
NR_JDBC_CON	JDBC Connections を含む
NR_BrokerDocTypes	ブローカーに対して公開するドキュメントタイプを含む
NR_Common	複数のサービスで呼び出される共通サービスを含む
NR_Validations	バリデーション実施に関するサービスを含む
NR_Uilities	複数のパッケージで使用される共通ユーティリティサービスを含む
NR_XMLValidation	XML バリデーションに使用される xerces サービスを含む
NR_BPM_receiveRCI	受信 RCI 処理するプロセスを含む
NR_BPM_CIS_RECEIVE	受信 CIS 処理するプロセスを含む
NR_BPM_receiveRFCIN	受信 RFCIN を処理するプロセスを含む
NR_CIS	送信向けの CIS と RFCIN を処理するサービスを含む
NR_AS2	AS2 通信のプロセスに関するサービスを含む

14 別紙 J: 技術参照ドキュメントの入手先

ドキュメントの入手先一覧

技術エリア	ウェブサイト / ドキュメント保管場所
Oracle	www.oracle.com
Solaris	www.sun.com
webMethods	www.advantage.webMethods.com
JNR	www.agentrics.com