



CHAPTER 2

Service Link

- 「概要」 (P.2-1)
- 「Service Link の設計と開発」 (P.2-4)
- 「Service Link トランザクションのモニタリング」 (P.2-32)
- 「Service Link アダプタ」 (P.2-41)
- 「Integration ウィザード」 (P.2-58)
- 「Service Link のトラブルシューティングと管理」 (P.2-64)
- 「事前作成済みの関数」 (P.2-66)

概要

はじめに

Service Link は、外部システムとの統合を可能にする、Cisco Service Portal (Service Portal) のモジュールです。フレームワークが提供され、Service Portal ワークフロー内で定義された配信タスク、承認、または確認を、他のシステムおよびユーザ インターフェイスで実行できるようにインターフェイスを設定し、インターフェイスの動作をモニタリングできます。

Service Link の使用のための最も一般的なシナリオは、サービスが十分なレベルで提供されるように、配信計画タスクに関連付けられたデータを Service Portal の外部に渡す必要がある場合です。たとえば、調達処理のためにメッセージをハードウェアベンダーに渡す場合や、データレコードの更新のためにインベントリまたは資産管理システムにメッセージを渡す場合があります。その後、場合によっては、外部アプリケーションが 1 つまたは複数のメッセージを Service Portal に返します。次に各メッセージは、外部システム内の現在のタスクのステータスで Service Portal を更新し、最終的にはタスクが完了し、Service Portal ワークフロー (配信計画) で後続のタスクを継続できることを示します。

Service Link は多数の組み込みアダプタがあり、ファイルの交換、データベースの更新、HTTP POST 要求または Web サービスを介した Web 通信、およびキューベースのメッセージングなどのさまざまな転送メカニズムを使用して、外部アプリケーションとの通信に役立ちます。これらのデフォルトのアダプタに加えて、開発者は Service Link Adapter Development Kit (ADK) を使用してカスタムアダプタを開発し、導入できます。追加アダプタキットは、HP Open View Service Desk、BMC Remedy、IBM Identity Management などのサードパーティ製品とのインターフェイスとして設計されたもので、Cisco Advanced Services から利用できます。

Service Link の前提条件

Service Link 統合の開発には、幅広い技術スキルが必要です。次の作業を行います。

- Active Form Components (AFC) でのディクショナリ使用の設定方法や配信計画でのタスクの設定方法など、サービス設計に関する理解。
- アプリケーションをホストしているサーバを含め、ターゲットのサードパーティ システムに関する詳細な知識。
- VMware アダプタ以外のすべてのアダプタでは、XML タグ構造の基本的な理解。Service Link は Service Portal と外部システムの間で XML メッセージを送信することによって動作するためです。
- VMware アダプタ以外のすべてのアダプタでは、Service Link ウィザードの使用によって適用できる XML 変換を補完するため、XML Stylesheet Language (XSL) 変換の設定の中程度の理解。
- データベース アダプタを使用する場合は、SQL に関する知識も必要です。
- http/Web サービス アダプタを Service Portal と Web サービスの間でメッセージを渡すために使用する場合、SOAP、WSDL、および Web サービス セキュリティなどの Web サービス コンポーネントの知識が役立ちます。

Service Link の設計方法とコンポーネント

Service Portal では、設計の統合のための 2 つのアプローチがあります。

- Service Designer で使用できる Integration ウィザードは、Web サービスの統合を作成するためのウィザード形式のアプローチです。
- Service Link モジュールには、外部システムとの通信に使用されるメッセージング プロトコルに係なく、すべての統合の作成および維持するための機能があります。

統合が作成されると、Service Link から使用できる高度な設定機能によって表示および維持が可能になります。上級ユーザはこの機能を使用し、必要に応じてウィザードをバイパスして Web サービス統合を作成することもできます。

また、管理者は Service Link を使用して、実稼働環境での統合の管理およびトラブルシューティングを実行します。

統合は次のコンポーネントで構成されます。

- アダプタ

アダプタは転送コンポーネントを論理的に表現したもので、これを使って Service Portal は XML 文書またはその他のメッセージをサードパーティ システムに送信します。事前にパッケージ化されたアダプタは、ファイル、http/Web サービス、JMS、IBM MQ、データベースを含め、さまざまなメッセージ転送プロトコルをサポートします。

アダプタは次の 2 つのコンポーネントで構成されます。

- 着信アダプタ

着信アダプタは、外部システムから着信するメッセージを管理します。外部システム メッセージは、データを Request Center で解釈できるように、変換によって「標準」の nsXML (旧称 newScale XML) 形式に変更できます。

着信アダプタには、ポーラーとリスナーの 2 種類があります。ポーラーは定期的に起動して着信メッセージを検出するスレッドで、リスナーは着信外部メッセージによって起動されるまで待機します。ポーラーの例としては、メッセージの定期的なチェックを必要とする着信ファイルアダプタがあります。リスナー アダプタの例としては、HTTP 応答を受信するまで待機する Web サービス リスナー アダプタがあります。

– 発信アダプタ

発信アダプタは **Service Portal** から発信される XML メッセージを管理し、設定された外部システムに送信します。「標準」の nsXML 発信メッセージが **Service Link** に送信されると、変換によってメッセージが変更され、外部システムに転送できるメッセージの形式になります。その後、発信アダプタによって適切なプロトコルおよび論理を適用し、外部システムにメッセージに送信します。

• エージェント

エージェントは転送メカニズムを論理的に表現したもので、これを使って **Service Portal** はサードパーティシステムと通信します。エージェントは、適切なサードパーティの宛先にタスクを指示するサービス設計者が使用します。このアクションを外部システムに指示するエージェントを指定することによって、タスクだけではなく、承認と確認を外部で実行できます。

エージェントは、着信アダプタおよび発信アダプタ、オプションのメッセージ変換 (XSLT) コンポーネント、オプションパラメータ、エラー状態を解決するためのその他の設定で構成されます。

• 変換

XML スタイルシート (XSL) 変換は、発信メッセージをサードパーティシステムで解釈される形式に変換し、着信メッセージを **Service Portal** で解釈できる形式に変換します。

発信アダプタが含まれるエージェントは、現在の要求およびタスクに関連する情報が含まれる nsXML メッセージを自動的に作成します。エージェントに関連付けられた変換では、メッセージを外部メッセージに変換でき、その外部メッセージはエージェントに対して設定された発信アダプタを介して外部システムに配信されます。同様に、着信エージェントは関連付けられたアダプタを介して外部メッセージを受信します。変換では、**Service Portal** のワークフロー マネージャ **Business Engine** によって認識および処理される着信メッセージタイプにメッセージを変換する必要があります。

• ディクショナリとアクティブ フォーム コンポーネント

ディクショナリは、特定のサービス要求を実行するために必要なデータのフィールドを保持するサービス設計コンポーネントです。ディクショナリ フィールドにマッピングされたエージェントパラメータ (またはサービス要求で使用できるその他のデータ) は、外部システムで容易に理解される標準の発信メッセージ形式です。外部システムから受信した着信メッセージのエージェントパラメータは、**Service Link** に、これらのパラメータにマッピングされたディクショナリ フィールドの値を更新するように指示します。変更されたフォーム データは、すぐにサービス フォームで使用できるようになります。また、ディクショナリが含まれるアクティブ フォーム コンポーネントは、**Service Link** 統合を実装するサービスに含まれている必要があります。

Integration ウィザードは、エージェントおよび変換と、統合ディクショナリ、アクティブ フォーム コンポーネントを自動的に作成して、エージェント設定を完成します。これらのコンポーネントが作成されると、**Service Link** および **Service Designer** によって維持されるようになります。

Business Engine と nsXML

Service Link を理解するうえで重要なことは、**Business Engine** との相互動作を理解することです。**Business Engine** は、すべてのワークフローを処理するコンポーネントです。ワークフローには、次のアクションがあります。

- 正しい順序で配信計画のタスクを開始する。
- 完了のすべての要件を満たしている場合に、タスクを完了とマークする。
- イベントのトリガーが発生したときに、設定されたとおりに電子メールを送信する。

Service Link を使用しない（すべてのタスクが Service Portal 内部で行われる）タスク計画では、Business Engine の動作がほとんど意識されません。Business Engine の使用は Service Link 内部で認識されます。これは、外部タスクのステータスを変更するために、Business Engine が理解できるメッセージを Service Link が処理または生成する必要があるためです。

Business Engine は外部タスク（Service Link で処理されるタスク）を開始するときに、発信 nsXML メッセージを生成します。発信タスクを処理する Service Link エージェントは、ターゲットシステムが理解できる形式に nsXML メッセージを変換し、エージェントで指定された発信転送メカニズム（アダプタ）を使ってターゲットシステムに配信します。

同様に、Service Link が外部システムからの着信メッセージを受信するように設定されている場合、そのメッセージを Business Engine が理解できる着信 nsXML メッセージに変換する必要があります。着信メッセージを使用して、現在の要求のためにサービス フォーム データを更新したり、現在のタスクを実行したり、現在の要求にユーザ コメントを追加したりできます。

有効な nsXML メッセージについては、この後で詳細に説明します。

Service Link の設計と開発

概要

Request Center では、Service Link とサードパーティ システムとの統合を設計、開発、導入するための 2 つのアプローチがあります。

- **Integration ウィザード**：統合を Web サービスを使って行う場合、Service Designer から Integration ウィザードを使用して、すべての Request Center 統合コンポーネントを作成できます。このアプローチを使用するには、Web Service 定義言語ファイル (wsdl) が使用可能になっている必要があります。
- **Service Link コンフィギュレーション**：統合で他の転送メカニズムを使用する場合や、Integration ウィザードを介して作成されたコンポーネントの確認や変更を行う場合、Service Link および Service Designer によって表示される画面を使用して、統合コンポーネントを設定します。

Service Link および Service Designer コンフィギュレーションでは、次の方法を使用して、サードパーティ システムとの統合を設計、開発、および導入します。

- サードパーティのターゲットシステムとの間で使用される通信プロトコルを設計します。これには、着信アダプタと発信アダプタの選択、メッセージの形式と内容が含まれます。
- 必要に応じて、カスタム Service Link アダプタを作成および導入します。
- Service Designer を使用して、Service Link 統合を実装するサービスを設計します。通常、設計コンポーネントには 1 つまたは複数のディクショナリが含まれており、ディクショナリにはエージェント パラメータを介して外部システムに渡されるデータ、これらのディクショナリが含まれるアクティブ フォーム コンポーネントが含まれ、これらのディクショナリのフィールドの表示プロパティを設定します。
- 適切な発信アダプタと着信アダプタを選択し、それぞれのプロパティと、いずれかの方向で渡されるパラメータを定義することによって、エージェントを作成します。Service Link には、エージェント パラメータの定義を部分的に自動化するためのウィザードとドロップダウン リストが含まれています。
- 必要に応じて、Service Portal がサードパーティ システムからのメッセージを理解し、サードパーティ システムが Service Portal からのメッセージを理解できるようにするための変換を作成します。

- Service Designer を使用して、サービスの配信計画内のタスクにエージェントを関連付けます。必要に応じて、エージェント パラメータが、サービスで使用されるディクショナリ フィールドに正しくマッピングされていることを確認します。
- タスクが含まれるサービスを要求し、該当する Service Link ページからメッセージおよび外部タスクをモニタリングすることによって、コンフィギュレーションをテストします。

このプロセスについては、この後で詳細に説明します。

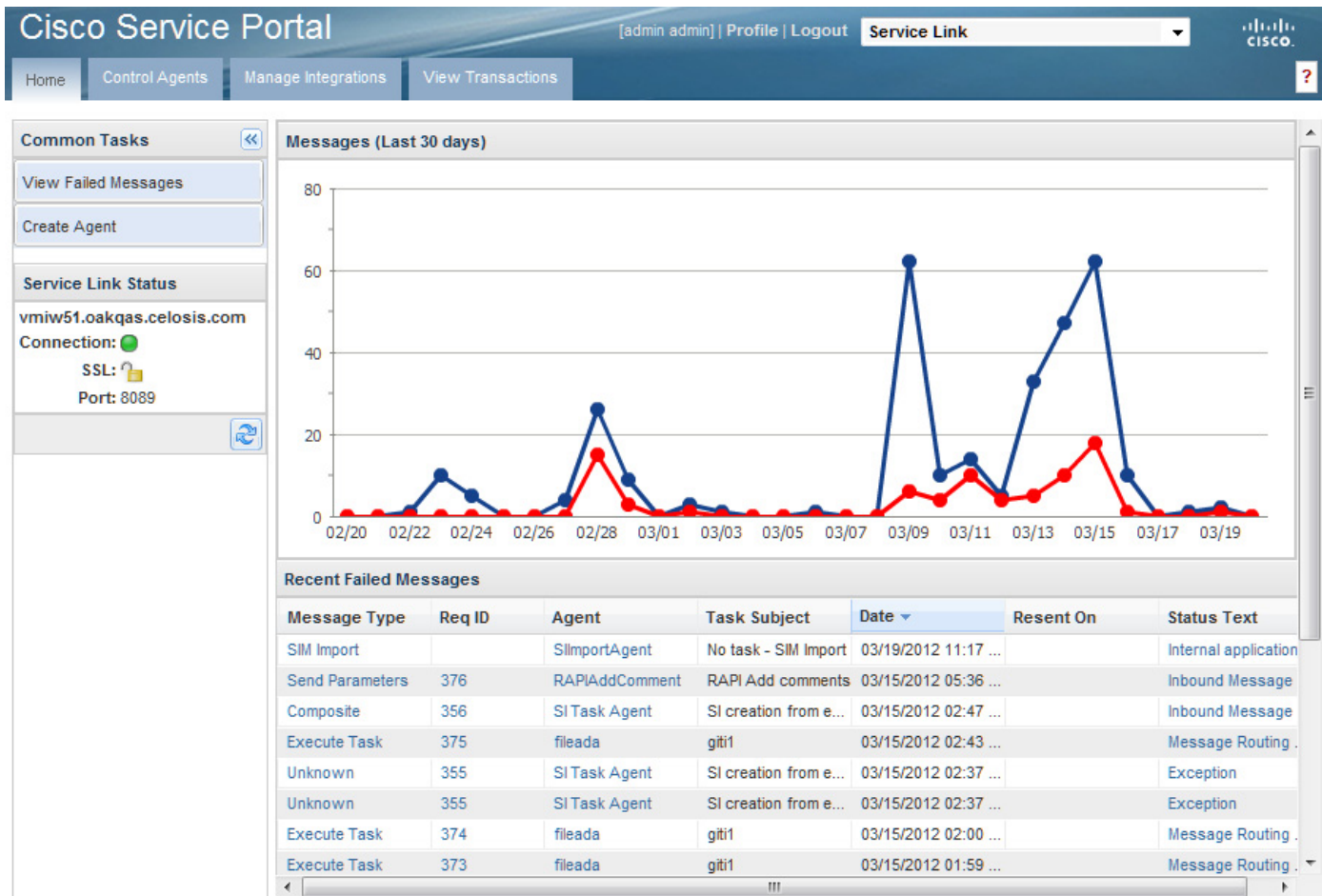
Integration ウィザードについては、「[Integration ウィザード](#)」(P.2-58) で説明します。

Service Link へのアクセス

Service Link にアクセスするには、モジュール メニューの [Service Link] を選択します。

Service Portal System My Workspace
My Services My Services Executive
Relationship Manager Service Level Manager Service Manager
Organization Designer Portal Designer Portfolio Designer Service Designer Service Item Manager
Administration Catalog Deployer Service Link
Reporting Advanced Reporting

次に示す、Service Link のホーム ページが表示されます。

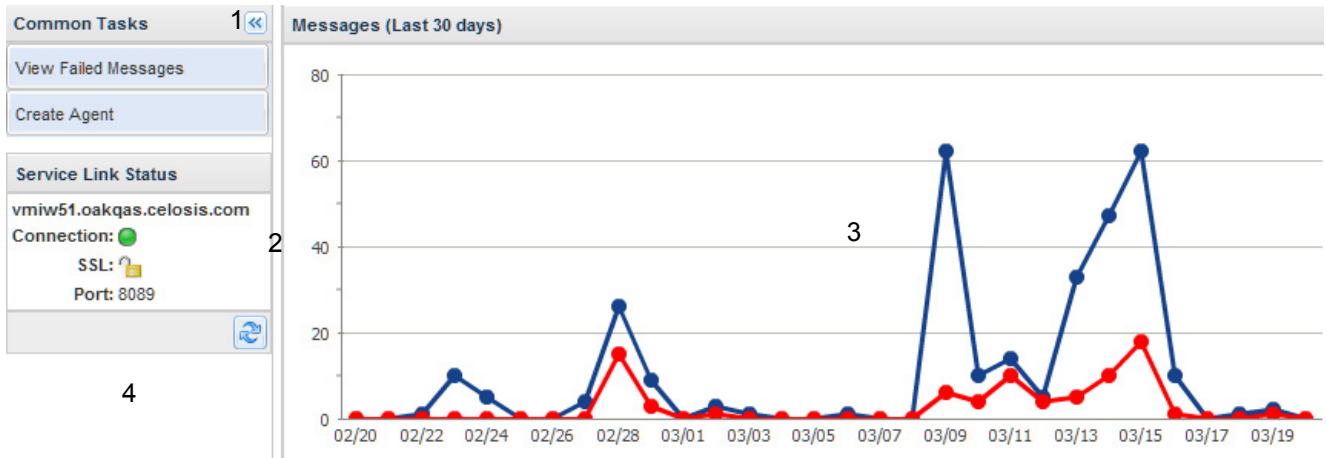


Service Link のホーム ページには、次の要素が含まれています。

要素	説明
メニュー バー	ページの上部にあり、Service Link の環境を管理し、Service Link の統合の開発および維持に使用されるタブが含まれています。
Common Tasks	ページの左側にあり、失敗したメッセージの全リストを表示し、エージェントを作成するためのクイック リンクが含まれています。
Service Link Status	ページの左側にあり、Service Link の現在のステータスが表示されます。
Messages (Last 30 days)	ページの右側にあり、このグラフには最近 30 日間のメッセージの大まかな量が表示されます。
Recent Failed Messages	ページの右下にあり、正常に配信されなかった最新の Service Link メッセージが一覧表示されます。メッセージ、要求、およびエージェントの詳細へのハイパーリンクがあります。

Service Link 画面の管理

ホーム ページと同様に、ほとんどの Service Link ページは左側のリスト パネルと、右側のコンテンツ パネルで構成されています。リスト パネルは、パネルの縮小および展開アイコンをクリックして縮小および展開できます。リスト パネルが表示されている状態でも、デバイダをドラッグしてコンテンツ パネルの幅を変更できます。



1	パネルの縮小アイコン	3	コンテンツ パネル
2	デバイダ	4	リスト パネル

[Recent Failed Messages] グリッド（および Service Link によって表示されるその他のグリッド）の外観を制御することもできます。

Recent Failed Messages						
Message Type	Req ID	Agent	Task Subject	Date	Resent On	Status Text
SIM Import			No task - SIM Import	03/19/2012 11:17 AM		Internal application e
Send Parameters	376		RAPI Add comments	03/15/2012 05:36 PM		Inbound Message Fi
Composite	356		SI creation from exter...	03/15/2012 02:47 AM		Inbound Message Fi
Execute Task	375			03/15/2012 02:43 AM		Message Routing Ex
Unknown	355	SI Task Agent		03/15/2012 02:37 AM		Exception
Unknown	355	SI Task Agent		03/15/2012 02:37 AM		Exception
Execute Task	374	fileada		03/15/2012 02:00 AM		Message Routing Ex
Execute Task	373	fileada		03/15/2012 01:59 AM		Message Routing Ex
Execute Task	372	fileada		03/15/2012 01:57 AM		Message Routing Ex
Execute Task	370	vmware_ceberus2		03/15/2012 01:32 AM		Inbound Message Fi
Composite	359	RAPIAddComment		03/14/2012 11:08 PM		Internal application e
Composite	361	SI Task Agent	SI inbound actions	03/14/2012 10:59 PM		Inbound Message Fi
Composite	361	SI Task Agent	SI inbound actions	03/14/2012 10:54 PM		Inbound Message Fi
Send Parameters	359	RAPIAddComment	RAPI Add comments	03/14/2012 10:38 PM		Inbound Message Fi
Send Parameters	358	RAPIAddComment	RAPI Add comments	03/14/2012 10:37 PM		Inbound Message Fi
Composite	358	RAPIAddComment	RAPI Add comments	03/14/2012 10:30 PM		Internal application e
Send Parameters	359	RAPIAddComment	RAPI Add comments	03/14/2012 10:29 PM		Inbound Message Fi
Send Parameters	358	RAPIAddComment	RAPI Add comments	03/14/2012 10:26 PM		Inbound Message Fi

1	ドロップダウン アイコン	3	カラム選択
2	ソート オプション	4	カラム幅カーソル

- カラムの幅を変更するには、カラム間の線の上にカーソルを置きます。カーソルが、反対方向を指す 2 つの矢印が付いた二重線に変わります。新しいカーソルが表示されたら、マウスの左ボタンを押しながらドラッグして、カラムの幅を調整できます。
- 表示をソートするには、ソートとカラムのヘッダーの上にマウスを移動します。ヘッダーの右にドロップダウンアイコンが表示されます。アイコンをクリックしてソート オプションを表示し、目的のオプションをクリックします。
- グリッドに表示されるカラムを変更するには、ドロップダウン リストで [Columns] オプションを強調表示し、表示するカラムをオンにしたり、グリッドから削除するカラムをオフにしたりします。
- グリッド内のカラムの順序を変更するには、移動するカラムのラベル内をマウスでクリックし、マウスをドラッグしてカラムの目的の位置で放します。

通信プロトコルの設計

Service Link で実行するコンフィギュレーションおよびテストに加えて、サードパーティ システムが担当するテクニカル リソースで同等の作業を実行する必要があります。インターフェイスが確実に動作するためには、設計を十分に検討することが不可欠です。

通常は、通信相手となるシステムのインターフェイス機能によって、統合の基本設計、つまり使用されるアダプタが決まります。

ファイル アダプタおよびデータベース アダプタは、設定が最も簡単です。JMS、MQ または http/Web サービス アダプタを導入する場合は、接続の問題が原因でシステム間でデータが移動できなくなることを防ぐために、ネットワーク管理チームの専門知識が必要な場合があります。ターゲット システムが企業のネットワーク外に存在している場合は、データ セキュリティの懸念も要因となります。たとえば、外部ベンダーとの通信に http または https で送信される SOAP メッセージを使用する場合があります。カスタム アダプタが必要な場合は、さらに多くの技術リソースが必要となり、統合の完了までに多くの時間を見積もっておく必要があります。

通常、発信 Service Link 通信に必要なデータを最初に評価します。すぐに使用可能なフィールド (nsXML 発信メッセージを使用) と、提供する必要のある追加データ (フォーム データおよびエージェント パラメータを使用) を検討する必要があります。発信通信はタスクごとに (タスクの開始時に) 一度だけ発生しますが、複数の独立した着信通信がサポートされます。作業実行の指示を受信すると、サードパーティ システムでは完了時に単一の (着信) 通信が発生する場合があります。または、作業が完了する前に、複数の更新を送信できます。たとえば、参照 ID を通信する場合、テキストのステータス更新を返送する必要があり、最終的に完了確認が通信されます。

アダプタ

Service Link アダプタは、使用できるように事前設定されています。追加のアダプタは Adapter Development Kit を使用して開発できます。Service Link の [Manage Integrations] タブの [Adapters] ページを使用して、使用可能なアダプタを確認できます。

ステップ 1 Service Link メニューから [Manage Integrations] を選択します。[Adapters] サブタブをクリックします。

[Adapters] ページが表示されます。

Name	Last Updated	Modified By
DB Adapter	02/09/2012 04:59 PM	
Dummy Adapter	02/09/2012 04:59 PM	
File Adapter	02/09/2012 04:59 PM	
HTTPWS Adapter	02/09/2012 04:59 PM	
JMS Adapter	02/09/2012 04:59 PM	
MQ Adapter	02/09/2012 04:59 PM	
Service Item Listener Adapter	02/28/2012 12:04 AM	
VMware Adapter	02/09/2012 04:59 PM	
Web Service Listener Adapter	02/09/2012 04:59 PM	

ステップ 2 [Name] カラムで目的のアダプタをクリックします。

選択したアダプタの [Manage Adapter] ページが表示されます。次に示すように、データベース アダプタの詳細が表示されます。

Manage Adapter	
Name	DB Adapter
Description	The purpose of the DB Adapter is to allow creation of Agents that communicate with other systems via tables in a database. It supports polling a table in a database and reading messages found there, as well as writing messages to a table in a database.
Created On	02/10/2012
Created By	admin admin
Last Updated On	02/10/2012
Last Updated By	null
Message Support	Bidirectional
Inbound Model	Poller
Inbound Class	com.newscale.is.adapter.db.DBInboundAdapter
Outbound Class	com.newscale.is.adapter.db.DBOutboundAdapter
Exception Class	
Polling Interval (in milliseconds)	<input type="text" value="10000"/>
Retry Interval (in milliseconds)	<input type="text" value="0"/>
Maximum Attempts	<input type="text" value="0"/>

これらの一般的なプロパティの大半は、通常、Service Link 開発者が変更すべきではありません。[Polling Interval]、[Retry interval]、および [Maximum Attempts] は要件に応じて変更が必要となる場合があります。すべての変更は、このアダプタ タイプを使用するすべてのエージェントに継承されます。

追加の発信および着信のプロパティは、アダプタがエージェントで使用される際に指定されます。これらのプロパティについては、「[Service Link アダプタ](#)」(P.2-41) で説明します。

エージェント

Create Agent ウィザードでは、エージェントの設定を行います。ウィザードは 8 つのページで構成されます。一部のページは、前のページで選択されたオプションに応じてスキップされます。

Create Agent ウィザードのページについて、次の表 2-1 にまとめます。

表 2-1 Create Agent ウィザードのページ

ページ	説明 / 用途
General Information	エージェントの名前、アクション、説明と、その設定および動作に関するその他の全般情報
End Points	エージェントで使用されるアダプタおよび変換
Outbound Adapter	発信アダプタの詳細な設定オプション
Inbound Adapter	着信アダプタの詳細な設定オプション
Outbound Request Parameters	すべてのアダプタ タイプの発信パラメータ (VMware アダプタを除く)
Outbound Response Parameters	http/ws アダプタに送信される発信メッセージに対する同期応答で受信されるパラメータ
Inbound Parameters	着信メッセージの一部として受信されるパラメータ
Summary	[Summary] ページには、前もって入力されたすべての情報が表示されます

エージェントを作成するには、次の手順を実行します。

ステップ 1 Service Link のホーム ページの [Common Tasks] 領域で [Create Agent] を選択するか、Service Link モジュールの [Manage Integrations] タブにある [Agents] サブタブをクリックし、[Create Agent] をクリックします。

Create Agent ウィザードの [General Information] ページが表示されます。これは、ウィザードを構成している 8 つのページの最初のページです。一部のページは特定のエージェント設定には関係がなく、スキップできます。

The screenshot shows the 'General Information' page of the Create Agent wizard. The page title is 'Step 1 of 8'. At the top right, there are buttons for 'Back', 'Next', 'Save', and 'Cancel'. The main content area contains the following fields:

- Name: [Text input field]
- Action: [Text input field]
- Outgoing Content: [Dropdown menu with 'Data and Parameters, No Service Details (default, small)' selected]
- Failed Email: [Dropdown menu with 'None' selected]
- Context Type: [Dropdown menu with 'Service Task' selected]
- Description: [Text area]

次の表 2-2 で説明するフィールドに値を入力し、[Next] をクリックします。

表 2-2 エージェントの作成：一般属性

設定	説明
Name	エージェントの名前。名前は固有である必要があります。
Action	エージェントによって実行されるアクションの説明。たとえば、「Service Portal To Remedy - Create Ticket」。アクションに一意性は求められませんが、タスクへのエージェントの割り当てを求められたときに選択リストに表示されるため、一意になるようにしてください。
Outgoing Content	このエージェントへの発信 nsXML メッセージを生成する必要があるノードを指定するオプション。詳細については、以下で説明します。
Failed Email	発信メッセージを宛先に配信できない場合に送信される電子メールのテンプレート。電子メールのテンプレートは、Administration モジュールの [Notifications] オプションで定義する必要があります。
Context Type	エージェントのタイプ。Service Link エージェントは「Service Tasks」であり、エージェントを配信計画の外部タスクとして使用できます。「Service Item」エージェントは、Service Item Manager によって設定された定義/データまたはサービス項目をインポートするために使用します。
Description	エージェントの詳細な説明。ここに表示される詳細な説明は、統合のサポートに役立ちます。

Failed Email

Failed Email 通知は、発信メッセージを配信できない場合に生成されます。Service Link の Failed Emails には、タスクに関する電子メールに使用可能な名前空間の標準セットに加えて、障害発生時に生成されるメッセージについての詳細が含まれています。電子メールのテンプレートにこのような名前空間を含めると、問題の診断に役立ちます。使用可能な名前空間の詳細については、『Cisco Service Portal Designer Guide』を参照してください。

Failed Email は着信メッセージに適用されません。Service Link が着信メッセージの処理に失敗した場合、通知は生成されません。

Outbound Message Content

発信アダプタは、Service Portal データベースに保存される nsXML メッセージを生成します。このメッセージは変換の対象となり、その結果として生成される外部メッセージは、指定されたアダプタを介して目的の宛先に配信されます。

メッセージの形式については、第 3 章「Service Link Adapter Development Kit」に説明があり、ISEE.war/WEB-INF/classes/nsxml.xsd にある、アプリケーション サーバで使用可能な対応する XML スキーマで使用できます。メッセージ全体には、サービス要求についてのすべての情報が含まれています。このようなメッセージはサイズが非常に大きくなる可能性があり（サービスで使用されるディクショナリやフィールドの数に応じて、500 K を超えることがよくあります）、結果的にデータベース内で大量のストレージを消費し、生成するために CPU の使用率も非常に高くなります。

リソースの消費を減らすため、Service Portal には次のオプションが用意されています。

- 「Compress messages」の管理設定によって、データベースに保存される Service Link メッセージを圧縮できます。これによって、ストレージの要件が大幅に減少しますが、メッセージは DBA やサービス担当者などの人間が判読不可になるため、トラブルシューティングが複雑になります。

- **Service Link Message Purge** スクリプトが使用可能です。このようなスクリプトは、完了したタスクに関するメッセージをトランザクション データベースから消去できます。これによって、データベース内のメッセージに対するストレージ要件が減少します。**Message Purge** スクリプトの使用に関する詳細については、「[Service Link のトラブルシューティングと管理](#)」(P.2-64) を参照してください。
- 「**Outgoing message content**」プロパティを操作すると、nsXML メッセージの選択したノードだけを含めるように発信内容を設定できます。これによって、発信メッセージを処理するためのメモリおよび CPU の要件が減少します。

デフォルトのメッセージ内容は「**Data and parameters; no Service Details (small)**」で、要求されたサービスを説明するコンテンツ ノードが含まれない nsXML メッセージが生成されます。エージェント パラメータおよび変換は、発信内容のタイプを考慮して設計する必要があります。要求されるすべての内容が nsXML メッセージに含まれていることを確認してください。特に、発信メッセージからディクショナリ データを除外する場合、エージェント パラメータを適切なフォーム フィールド（または定数）にマッピングする必要があります。外部タスクで必要とされない多数のフォーム フィールドがサービスにある場合、XML サイズや、関連する CPU 使用率が大幅に減少します。

発信内容のオプションの概要については、次の表 2-3 を参照してください。

表 2-3 発信メッセージの内容 : オプション

オプション	説明
All Message Details (large)	Service Link メッセージ全体が生成されます。
Data, Form and Parameters (medium-large)	サービスとそのタスクに関する情報は省略されます。メッセージはデータ (サービス フォームのフィールド値)、フォーム (サービス フォームのディクショナリおよびフィールドについての完全なメタデータ)、およびパラメータ (エージェント パラメータで指定される値) に制限されます。
Data and Parameters (medium)	メッセージは、要求についての情報、サービス フォームに入力されたすべてのデータ値、パラメータ値に制限されます。
Data and Parameters; No Service Details (small)	メッセージは、要求についての情報、サービス フォームに入力されたすべてのデータ値、パラメータ値 (デフォルト) に制限されます。 VMware アダプタに対しては、「 small 」オプションを指定する必要があります。
Only Parameters (minimal)	メッセージは、要求およびエージェント パラメータに関する情報に制限されます。

アダプタの選択

エージェントは、発信と着信の両方の通信、発信の通信だけ、または着信データだけを管理するように設定できます。各方向に異なるアダプタ タイプを使用できます。たとえば、データベース アダプタを使用して発信データを書き込みますが、システムはディレクトリにファイルを書き込むことで応答し、そのファイルが着信ファイル アダプタによって読み込まれるようにできます。

アダプタ タイプを選択すると、ウィザードの後続のページは、アダプタ タイプおよび使用方法 (発信または着信) に関連するプロパティを表示するように調整されます。プロパティ値は、エージェント定義の一部として指定する必要があります。

Create Agent ウィザードの [End Points] ページ (2/8 ページ) では、エージェントで使用するアダプタ、および発信または着信メッセージに適用される変換を指定します。変換は、[Manage Integrations] オプションの [Transformations] サブタブを使用して事前に定義しておく必要があります。

Step 2 of 8 Back Next Save Cancel

End Points

Outbound Adapter:

Inbound Adapter:

Outbound Transformation:

Inbound Transformation:

表 2-4 エージェントの作成 : End Point のプロパティ

設定	説明
Outbound Adapter	Service Portal から外部システムへのメッセージの送信に使用される、デフォルトまたはカスタム アダプタ。
Outbound Transformation	外部システムが理解可能な外部メッセージを生成するために、発信アダプタによって生成される nsXML メッセージに適用する変換（または、なし）。
Inbound Adapter	外部システムからのメッセージの受信に使用されるアダプタ、または着信メッセージがないことが予想される場合は「Auto Complete」。
Inbound Transformation	Business Engine で理解できる nsXML メッセージを生成するために、着信メッセージに適用する変換（または、なし）。Service Tasks だけに適用されます。

アダプタの各タイプに適用可能なプロパティについては、このマニュアルで後述します。実際には、ウィザードの発信アダプタと着信アダプタの設定のための次の 2 つのページは、選択したアダプタに応じて異なります。ただし、ダミー アダプタと自動完了オプションの 2 つの特別な場合について説明する必要があります。

ダミー アダプタは、発信アダプタまたは着信アダプタのいずれかとしてエージェント内で設定できます。ダミー アダプタを選択すると、エージェントが単一方向だけで動作します。たとえば、ダミー着信アダプタが設定されたエージェントは、発信の通信だけを行うこととなります。一方、タスクの更新と終了専用の、別の（着信だけの）エージェントを設定することもできます。

自動完了オプションは、エージェントの着信アダプタ部分に対する選択肢としてのみ使用できます。その効果は「ダミー アダプタ」を選択することと同様で、エージェントは発信の通信だけを管理します。主な違いは、タスクに関連付けられた出力の通信が送信された後、タスクが自動的に完了し、残りの配信計画の実行が継続されることです。

エージェント パラメータ

エージェント パラメータは、発信アダプタと着信アダプタの両方で使用できます。

エージェント定義の一部として指定されたパラメータ マッピングでは、使用されるデフォルト値が指定されます。これらのマッピングは、Service Designer でサービスのタスク定義を編集することによって、サービスごとに上書きできます。

発信要求パラメータ

発信メッセージで使用されるエージェント パラメータは、標準の nsXML 発信メッセージのコンテンツ ノードを追加データで補完し、処理しやすい形式でコンテンツ ノードを整理する手段です。これらのパラメータには XSL 変換を介して簡単にアクセスでき、外部メッセージに含めることができます。

Step 5 of 8 Back Next Save Cancel

Outbound Request Parameters

nsXML elements +
Dictionaries +
Prebuilt Functions +

Outbound Parameter Mappings

Add Mapping Remove Selected

Service Data Mapping	Parameter
	New Parameter

Edit Parameter Values

Parameter:

Service Data Mapping:

Apply

パラメータ マッピングは、[Service Data Mapping] 領域にソース要素を入力するか、[Parameter Mappings] パネルの左側にあるドロップダウン リストに表示される要素を使用して式を作成することによって割り当てます。マッピングは次の組み合わせで構成されます。

- 定数値
- サービス フォームのディクショナリ フィールド
- nsXML 要素
- 上記のいずれかの要素に適用される事前作成済みの関数

エージェント パラメータのマッピングには、次の利点があります。

- パラメータ値を抽出する変換では、値が入力されるディクショナリ フィールド名を参照する必要はありませんが、単にエージェント パラメータ名を参照することはできます。これによって、異なるサービスおよびディクショナリ間でエージェントを再利用しやすくなります。
- パラメータはすべてのコンテンツ タイプに含まれるため、メッセージで必要とされる他のすべてのコンテンツがサポートされる場合、サイズの小さい発信メッセージ コンテンツ タイプを使用できます。
- 変換を使用せずにはアクセスできない nsXML 要素および XPATH 操作を、外部メッセージに含めることができます。

発信パラメータを作成するには、次の手順に従います。

ステップ 1 [Outbound Request Parameters] ページで、[Add Mapping] をクリックします。

ページの下部に [Edit Parameter Values] ダイアログが表示されます。

ステップ 2 パラメータの名前を入力します。

パラメータ名にはスペースを使用できますが、XML メッセージで意味を持つ特殊文字（「>」や「&」など）は使用できません。

ステップ 3 次に示すガイドラインを使用して、パラメータの値/マッピングを指定します。

定数値

要求またはサービスの詳細に依存しない定数値を、外部システムに渡すことが必要になる場合があります。たとえば、システムで外部システムのソース名が必要とされる場合、「Service Portal」や「Request Center」は単に Service Data Mapping として入力できます（引用符なし）。

発信 nsXML マッピング

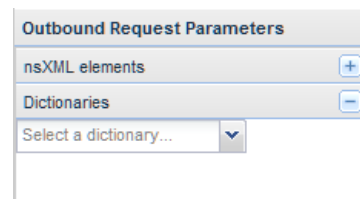
標準の nsXML 発信メッセージの選択された要素を、エージェント パラメータへのマッピングに使用できます。これについて、次の表にまとめます。

nsXML 要素	内容 / 説明
Customer	要求に対するカスタマーのログイン名
Initiator	要求の要求者（発信者）のログイン名
requisition-entry-id	要求内でサービス要求を一意に識別する番号。通常は、買い物カゴ内の複数のサービスを区別するために使用されます。
expected-cost	サービスの取引価格
expected-duration	サービスの標準期間
requisition-id	要求を一意に識別する番号。My Services と Service Manager で参照されます
channel-id	外部タスクを識別するグローバルな固有識別子（GUID）

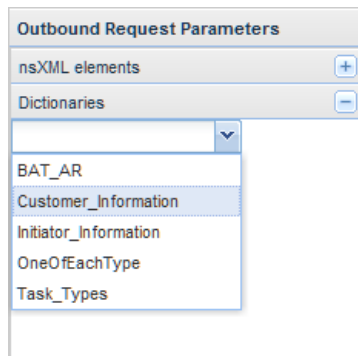
ディクショナリのマッピング

エージェント パラメータをディクショナリ フィールドにマッピングするには、次の手順に従います。

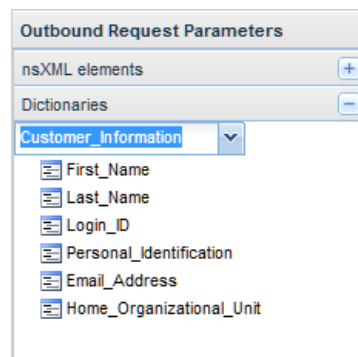
ステップ 1 [Dictionaries] ノードを展開し、[Select a dictionary...] オプションを表示します。



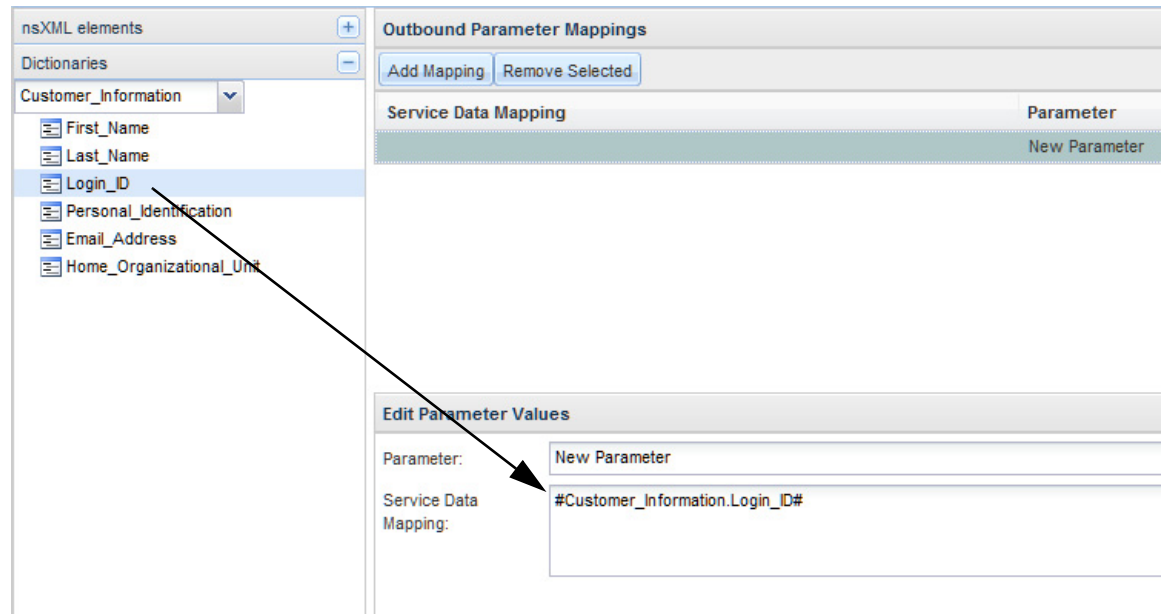
ステップ 2 [Select a dictionary...] ドロップダウン矢印をクリックし、すべての Request Center ディクショナリのリストを表示します。



ステップ 3 エージェントパラメータにマッピングするフィールドが含まれるディクショナリを選択します。ディクショナリのすべてのフィールドのリストが表示されます。



ステップ 4 エージェントパラメータにマッピングするフィールドを選択し、[Service Data Mapping] テキスト領域にドラッグします。ドラッグアイコンが緑色のチェックマークに変わったら、マウスを放します。選択されたフィールドの Lightweight 名前空間が表示されます。



エージェントはサービスとは関係なく定義されるため、グリッドディクショナリを除く任意のディクショナリフィールドを選択できます。参照されるディクショナリが、実際にこのエージェントが使用されるサービスに含まれるようにすることは、サービス設計者が行います。

1 つまたは複数のグリッドディクショナリの内容を渡す統合の場合、複数のディクショナリインスタンスとして保存される各グリッドディクショナリの行を処理する変換が必要です。この変換は、発信 nsXML の <data-values> セクションで、「DictionaryName-n」の命名規則に従います。n はグリッド行番号です。たとえば、VMOperation という名前のグリッドディクショナリで、サービス要求に 2 行のデータがある場合、その値は次のように表現されます。

```
<data-values>
  <data-value multi-valued="true">
    <name>VMOperation-1.Name</name>
    <value>vmgw01</value>
  </data-value>
  <data-value multi-valued="false">
    <name>VMOperation-1.GuestOS_Name</name>
    <value>winNetStandardGuest</value>
  </data-value>
  <data-value multi-valued="false">
    <name>VMOperation-1.CPUCount</name>
    <value>1</value>
  </data-value>
  <data-value multi-valued="false">
    <name>VMOperation-1.Memory</name>
    <value>2048 MB</value>
  </data-value>
  <data-value multi-valued="true">
    <name>VMOperation-2.Name</name>
    <value>vmgw01</value>
  </data-value>
  <data-value multi-valued="false">
    <name>VMOperation-2.GuestOS_Name</name>
    <value>winNetStandardGuest</value>
  </data-value>
  <data-value multi-valued="false">
    <name>VMOperation-2.CPUCount</name>
    <value>1</value>
  </data-value>
  <data-value multi-valued="false">
    <name>VMOperation-2.Memory</name>
    <value>2048 MB</value>
  </data-value>
</data-values>
```

着信エージェントパラメータマッピングおよびグリッドディクショナリフィールドの更新はサポートされません。

事前作成済みの関数

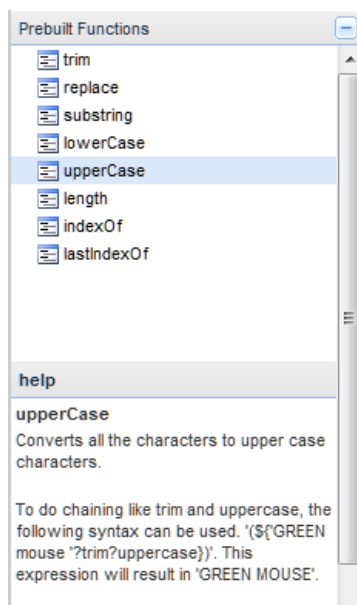
パラメータ値がターゲットシステムで予測されるセマンティックスまたはフォーマットの要件を満たすよう、マッピングされた要素に事前作成済み関数を適用できます。たとえば、ターゲットシステムのフィールドのデータ定義に Service Portal で維持されるよりも少ない文字が含まれている場合、substring 関数を適用することによって、フィールドを短くできます。次の表に、事前作成済み関数をまとめます。詳細については、「事前作成済みの関数」(P.2-66) を参照してください。

機能	用途 / 説明
trim	値から先頭と末尾のスペースを削除します。特に、値が CDATA タグで囲まれる、データベースアダプタのフォームデータや着信メッセージに役立ちます。
replace	1 文字またはパターンを別の文字またはパターンと置換します。
substring	開始点とオプションの長さで指定して、文字列の一部を選択します。
lowerCase	値をすべて小文字に変換します。
upperCase	値をすべて大文字に変換します。
length	文字列の文字数を返します。
indexOf	指定したサブストリングがその文字列で最初に出現する場所を示すインデックスを返します。
lastIndexOf	指定したサブストリングがその文字列で最後に出現する場所を示すインデックスを返します。

エージェント パラメータに事前作成済み関数を適用するには、次の手順に従います。

ステップ 1 [Prebuilt Functions] ノードを展開し、関数名を表示します。

ステップ 2 使用する関数を強調表示します。パネルの下部に、関数の使用方法を説明する [Help] が表示されます。



ステップ 3 関数をパラメータの [Service Data Mapping] ボックスにドラッグします。ドラッグアイコンが緑色のチェックマークに変わったら、マウスを放します。

Edit Parameter Values

Parameter:

Service Data Mapping:

ステップ 4 関数が定義され、`$Parameter$` が実際の値のプレースホルダになります。

Edit Parameter Values

Parameter:

Service Data Mapping:

ステップ 5 プレースホルダを、使用する nsXML メッセージのディクショナリ フィールドまたは要素と置換します。フィールドまたは nsXML 要素を [Service Data Mapping] テキスト ボックスにドラッグし、パラメータ定義を手作業で編集する必要があります。

Edit Parameter Values

Parameter:

Service Data Mapping:

エージェントパラメータが指定された発信 nsXML

エージェントパラメータは、発信 nsXML メッセージの末尾に追加されます。たとえば、次に示すエージェントパラメータは nsXML スニペットを直後に生成します。

Service Data Mapping	Parameter
\$initiator\$	Initiator
\$expectedCost\$	Transactional Price
\$customer\$	Customer

```
<agent-parameter multi-valued="false">
  <name>Initiator</name><value>ltierstein</value></agent-parameter>
<agent-parameter multi-valued="false">
  <name>Transactional Price</name><value>0.0</value></agent-parameter>
<agent-parameter multi-valued="false">
  <name>Customer</name><value>Customer</value></agent-parameter>
```

発信応答パラメータ

発信応答パラメータは、発信 http/Web サービス アダプタと組み合わせて使用できます。アダプタの Process Response 設定が true の場合、元の要求への応答が処理されます。この応答には「Send Parameters」メッセージタイプが含まれる場合があります。パラメータは着信エージェントパラメータとして定義されます。

着信エージェントパラメータ

エージェントパラメータを着信「Send Parameters」メッセージタイプと組み合わせて使用すると、パラメータがマッピングされるディクショナリ フィールドを外部タスクで更新できます。

表 2-5 着信エージェントパラメータ設定

設定	説明
Parameter	パラメータ名。
Dictionary Field	すべてのディクショナリ フィールドのリストが表示されたドロップダウンリストから、ディクショナリ フィールドを選択します。フィールド名は次の形式です。 DictionaryName.FieldName ハッシュ マーク (#) で囲みません。
Mapping	指定されたディクショナリ フィールドの更新に使用すべき値の取得に適用される事前作成済み関数。
Mandatory	着信メッセージにこのフィールドが存在する必要がある場合は、[Mandatory] をオンにします。サービスに変更が加えられてパラメータが不要になった場合、通常はこのチェックボックスをオフにします。不要になったパラメータを削除しないでください。

変換

変換を作成するには、次の手順を実行します。

- ステップ 1 [Manage Integrations] タブの [Transformations] サブタブを選択します。
- ステップ 2 [Create Transformation] ボタンを選択します。
[Create Transformation] ページが表示されます。

変換は [Direction] を指定することによって、発信メッセージと着信メッセージのいずれかに適用できます。[Process Response] 設定がオンになっている場合、Web サービス発信メッセージに 2 つの変換の使用が必要になる場合があります。1 つは発信要求に適用され、もう 1 つはその要求への応答に適用されます。

[Validate] ボタンをクリックし、XSLT を解析して、変換の形式が適切であることを確認します。適切ではない場合 (XML タグのスペル間違いや不足など)、診断メッセージが表示されます。エラーを修正してから変換を保存してください。

ただし、Service Link が変換を検証することはありません。たとえば、元のメッセージに存在していない要素を変換が参照している場合、エラーは検出されません。正しい形式の XML メッセージが生成されますが、ターゲットシステムに対しては有効ではありません。したがって、Service Link がターゲットアプリケーションで認識できない外部メッセージ、または Business Engine で認識できない着信メッセージを生成した場合は、ランタイム エラーが生成されます。

XML 開発環境にアクセスでき、使用方法がわかっている場合、その環境を使用して変換をテストできます。出力変換では、エージェントによって生成された nsXML を XML 開発環境にコピーし (変換の適用前)、これをソース XML として使用するだけです。変換を検証した後、XSLT コードを Service Link 変換にコピーして貼り付け、適切なエージェントに関連付けます。

Integration ウィザードを使用して開発された発信 Web サービスの統合には、手動で変換を開発してデバッグするプロセスが不要です。Web サービス用の変換が自動的に作成され、これにより、発信 nsXML が、指定された WSDL と互換性のある形式に変換されます。ただし、wsdl または統合の要件が変更された場合、上記で説明した手順に従って、変換を更新する必要があります。

ステップ 3 次の表 2-6 で説明するフィールドに値を入力します。

表 2-6 変換の設定

設定	説明
Name	変換の名前。変換名は、インターフェイスの性質を表すわかりやすい名前にします。「Service Portal To Remedy」など、ソース システムとターゲット システムの名前を含めてもかまいません。
Direction	[Inbound] または [Outbound]。
Description	変換の説明（必要な場合）。
Request subtab	発信アダプタの nsXML メッセージおよび着信アダプタの外部メッセージに適用される XSLT コード。
Response subtab	エージェントの [Process Response] 設定が [true] に設定されている場合、http/Web サービス要求から受信する応答に適用される XSLT コード。

ステップ 4 [Validate] をクリックして、変換に正しい形式の XSL が含まれていることを確認します。

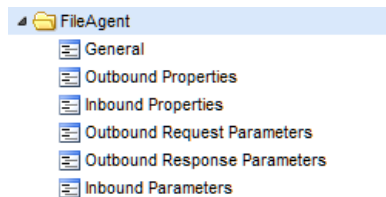
ステップ 5 [Save] をクリックします。

エージェント定義とプロパティ シートの確認

エージェントが Service Link または Integration ウィザードのどちらで作成された場合でも、すべてのエージェントの定義を確認または変更できます。[Manage Integrations] タブの [Agents] サブタブを表示し、次のいずれかを実行します。

- ページの左側の [List] パネルを使用し、エージェント名をクリックします。
- ページの右側に一覧表示されたエージェント情報をスクロールし、エージェント名をクリックします。

[List] パネルの [Agent] エントリが展開されます。編集または確認を行うエージェント定義の該当部分のプロパティシートをクリックします。プロパティシートが表示されたら、変更を入力し、[Save] をクリックして変更内容を保存します。



エージェント名をクリックすると、エージェント定義の概要が表示されます。

Agent Information	
Name:	FileAgent
Action:	FileAction
Status:	Not sending or receiving
Description:	This is a file agent
Service Tasks using this agent	
Service	Name
Configurable Tasks File Adapter	Second Task
Configurable Tasks File Adapter	Delivery Task
Configurable Tasks File Adapter	Service Group Review
Configurable Tasks File Adapter	Service Group Authorization
<input type="button" value="Reset Agent Parameters"/> Page 1 of 1 <input type="button" value="Display"/>	

このページは、[Reset Agent Parameters] のボタンを除いて読み取り専用です。

Service Link エージェントを使用するタスクの設定

エージェントを定義すると、エージェントを起動するワークフローの外部タスクを作成することによって、サービスで使用できるようになります。ワークフローが設定されると、含まれるエージェントに対して定義されたエージェント パラメータを確認したり、上書きしたりできます。

外部タスクの作成

Service Link エージェントを使用して、外部（サードパーティ）アプリケーションに対してタスクを指示するには、次の手順に従います。

- ステップ 1** Service Designer を起動します。外部タスクを含めるサービスを選択します。[Plan] タブをクリックします。
- ステップ 2** 外部タスクの指定には、[Tasks] サブタブまたは [Graphical Designer] サブタブを使用できます。
 - サービスの [Tasks] サブタブを使用して、タスクを定義し、配信計画に正しいシーケンスで配置します。
 - [Graphical Designer] サブタブを使用して、図上にタスクを作成し、Associate ツールを使用して、図に正しいシーケンスで配置します。タスクをダブルクリックすると、そのプロパティシートが表示されます。
- ステップ 3** [General] タブの [Workflow Type] ドロップダウン リストを使用して、ドロップダウン リストから目的のアクションを選択します。これは Service Link モジュール定義済みのアクションであり、このドロップダウン リストに一覧表示されるエージェント名ではないことに注意してください。
- ステップ 4** ワークフローとタスク計画を保存します。Workflow Designer を使用していた場合は、[Tasks] サブタブに戻ります。
- ステップ 5** 外部タスクを保存すると、[Workflow Type] の横に省略符号ボタンが表示されるようになります。省略符号をクリックすると、エージェント（存在している場合）に対して現在有効なパラメータ マッピングを確認したり、この特定のサービスのマッピングを変更したりできます。省略符号ボタンをクリックします。

General Participants Email Task Instructions Checklist

Save

Workflow Type: Update Inventory ... Create Agent

Task name: Update Inventory

ステップ 6 [Agent Parameter Override] ポップアップ ウィンドウが表示されます。エージェント パラメータを確認したり、1 つまたは複数のパラメータを変更します。各パラメータを変更するたびに [Apply] を必ずクリックし、ウィンドウを閉じる前に [Save] を必ずクリックしてください。

Agent Parameter Override

Outbound Request Outbound Response Inbound

nsXML elements +

Dictionaries +

Prebuilt Functions +

Outbound Parameter Mappings

Save

Service Data Mapping	Parameter
#RAPI_GetServiceDef.RQ_Password#	RQ_Password
#RAPI_GetServiceDef.RQ_Username#	RQ_Username
#RAPI_GetServiceDef.RQ_commentText#	RQ_commentText
#RAPI_GetServiceDef.RQ_loginUserName#	RQ_loginUserName
#RAPI_GetServiceDef.RQ_requisitionId#	RQ_requisitionId

Edit Parameter Values

Parameter: RQ_Password

Service Data Mapping: #RAPI_GetServiceDef.RQ_Password#

Apply

ステップ 7 [Participants] タブでは、実行者（個人、キュー、または役職）を定義できます。タスクの期限の計算には、エンティティの実行者のカレンダーが使用されます。外部タスクの [Participants] タブに値を設定していない場合は、タスクの期限の計算に、Default Service Queue のカレンダーが使用されます。このようにして、計画の期限が設定され、外部タスクの配信 Operating Level Agreement (OLA) コンプライアンスおよびこのようなタスクを含むサービスの Service Level Agreement (SLA) とのコンプライアンスを Service Portal で計算できます。

エージェント マッピングとサービス定義の同期

エージェントを使用するタスクを作成して保存する場合、そのエージェントに指定されたエージェント パラメータ マッピングは、それぞれのタスクによって自動的に継承されます。上記のように、サービス設計者はすべてのエージェント マッピングをタスク レベルで上書きできます。

ただし、その後、別のセットのエージェント パラメータ マッピングを含めるためにエージェントを変更しても、そのエージェントを使用するように以前に定義したタスクによってその変更が自動的に継承されることはありません。次のような変更が行われる可能性があります。

- エージェント パラメータの追加
- エージェント パラメータの削除
- 既存のエージェント パラメータのマッピングの変更

エージェントを使用するサービスへの変更の伝達は、次の手順を実行することによって、自動化できます。

ステップ 1 [Service Link Manage Integrations] タブの [Agents] サブタブをクリックします。

ステップ 2 パラメータ マッピングが変更されたエージェントの名前をクリックします。
[Agent Information] ページが表示されます。

Agent Information	
Name:	FileAgent
Action:	FileAction
Status:	Not sending or receiving
Description:	This is a file agent

Service Tasks using this agent	
Service	Name
Configurable Tasks File Adapter	Service Group Authorization
Configurable Tasks File Adapter	Service Group Review
Configurable Tasks File Adapter	Delivery Task
Configurable Tasks File Adapter	Second Task

ステップ 3 更新されたエージェント定義とエージェント パラメータ マッピングを再同期する必要があるサービスを選択します。

ステップ 4 [Reset Selected Tasks] ボタンをクリックします。このボタンをクリックすると、すべてのパラメータ マッピングがそのエージェントのデフォルトに自動的にリセットされるため、タスクに固有のマッピングを再適用する必要があります。

nsXML メッセージ

変換には正しい形式の XML を含める必要があるだけでなく、正しい形式の有効な nsXML メッセージを生成する必要があります。すべての nsXML メッセージが nsXML スキーマ (XML 文書の構造を記述する XML 文書) に従っている必要があります。スキーマはアプリケーション サーバの ISEE.war/WEB-INF/classes/nsxml.xsd にあります。

発信 nsXML メッセージ

外部タスクのステータスが **Ongoing** に変わると、発信 nsXML メッセージが生成されます。

各メッセージに生成された nsXML は、[Message Details] ページで nsXML メッセージをクリックすると、Service Link モジュールに表示できます。これには、タスク関連のデータと親要求関連のデータが含まれます。

nsXML 内で最も重要な要素は **channel-id** であり、外部タスクを一意に識別する ID です。この ID はサードパーティ システムに渡され、そのビジネス エンジンによって、対応するデータ更新が正常に適用された場合に、応答に示される必要があります。

channel-id は **Globally Unique Identifier (GUID)** としてフォーマット設定されます。GUID はほとんどの場合、次のように 16 進数のシーケンスでテキストに書き込まれます。

```
3F2504E0-4F89-11D3-9A0C-0305E82C3301
```

このテキスト表記は 5 セットのデータで構成され、各セットはハイフンで区切られます。GUID は 32 文字と 4 つのハイフンで構成され、合計 36 文字の長さになります。

発信メッセージは 2 種類あります。

task-started

task-started メッセージ タイプは、外部タスクの開始時に生成されます。**task-started** メッセージの要素の詳細については、第 3 章「[Service Link Adapter Development Kit](#)」を参照してください。

```
<message channel-id="3F2504E0-4F89-11D3-9A0C-0305E82C3301">
  <task-started task-type="task">
    <task>
      .
      .
      .
    </task>
  </task-started>
</message>
```

task-canceled

task-canceled メッセージ タイプは、外部タスクを含む要求のキャンセル時に生成されます。(サービスの配信計画の対応する設定から) 外部タスクが実行された後、ユーザが要求のキャンセルを許可されていない場合、このメッセージは生成されません。ただし、要求のキャンセルが許可されている場合は、外部タスクを実行するエージェントで使用される変換が「十分にスマート」で、**task-started** メッセージと **task-canceled** メッセージの両方を処理できる必要があります。変換で **task-canceled** メッセージ タイプのテストを行い、適切なメッセージを外部システムに送信する必要があります。

```
<xsl:if test="/message/task-started">
  <!-- Original XSLT goes here -->
</xsl:if>

<xsl:if test="/message/task-canceled">
  <!-- XSLT for the cancel message goes here -->
</xsl:if>
```

着信 nsXML メッセージ

サードパーティ システムから信号を受信すると、データが次のメッセージ タイプのいずれかになるように処理される必要があります。

take-action

take-action メッセージは承認または配信タスクに適用でき、タスクのステータスを変更します。**take-action** タグのアクション属性は、実行するアクションを識別します。次に、有効なアクションの概要を示します。

アクション	タスク タイプ	説明
done	配信タスク	配信タスクに完了済みのマークを付けます。
cancel	配信タスク	配信タスクをキャンセルします。

ok	確認タスク	確認が完了したマークを付けます。
reject	承認タスク	承認を拒否します。
approve	承認タスク	承認を許可します。

タスク計画の最後の配信タスクに **done** のマークが付くと、要求がクローズ（完了）します。**take-action** タグの「アクション」属性を対応する値に設定すると、許可タスクを **Approved** または **Rejected** とマークできます。

```
<?xml version="1.0" encoding="UTF-8"?>
<message channel-id="3F2504E0-4F89-11D3-9A0C-0305E82C3301">
  <take-action action="done"/>
</message>
```

send-parameters

パラメータは、エージェント定義内のディクショナリ フィールドにバインドされるデータ要素です。**send-parameters** メッセージ タイプでは、指定された 1 つまたは複数のパラメータを更新し、続いて、サービスの対応するディクショナリ フィールドを更新できます。このタイプの着信メッセージの使用は、サービス要求で使用されるディクショナリ フィールドを外部システムで更新する推奨方法です。

```
<?xml version="1.0" encoding="UTF-8"?>
<message channel-id="3F2504E0-4F89-11D3-9A0C-0305E82C3301">
  <send-parameters>
    <agent-parameter>
      <name>Status</name>
      <value>Resolved</value>
    </agent-parameter>
    <agent-parameter>
      <name>ResolvedBy</name>
      <value>Help Desk</value>
    </agent-parameter>
  </send-parameters>
</message>
```

add-comments

add-comments メッセージは、要求の [System Comments] セクションにコメントを追加するために使用されます。

```
<?xml version="1.0" encoding="UTF-8"?>
<message channel-id="3F2504E0-4F89-11D3-9A0C-0305E82C3301">
  <add-comments>
    <comment>Test Comment</comment>
  </add-comments>
</message>
```

サービス項目の作成/更新/削除メッセージ

サービス項目は、**Lifecycle Center** で定義されたエンティティです。そのライフサイクルは、サービス項目インスタンスがプロビジョニングされた時点から、廃棄される時点までで、サービス要求に関連付けられます。サービス項目のライフサイクル イベントが外部システムによって処理される場合、**Service Link** サービス項目の作成、更新、および削除のメッセージを介して、サービス項目データを **Lifecycle Center** と同期することができます。これらのメッセージ タイプは、**Web** サービスベースのサービス項目リスナー アダプタだけでサポートされます（「サービス項目リスナー アダプタ」(P.2-54)を参照）。

これらのメッセージには、1 つまたは複数のサービス項目タイプおよびサービス項目インスタンスを含めることができます。1 つのメッセージに複数のサービス項目操作を組み合わせることはできません。言い換えると、操作の作成、更新、または削除は個別の着信メッセージで送信する必要があります。エラー状態が発生すると、同じメッセージのすべてのサービス項目の操作がロールバックされます。

日時タイプのサービス項目属性は、YYYY-MM-DD HH:MI:SS または YYYY-MM-DD の形式で指定する必要があります。すべての時刻は UTC 時間で保存されます。

サービス項目のサブスクリプションは、サービス項目インスタンスの作成時にオプションで含めることができます。作成操作にサブスクリプション情報が含まれていない場合、項目は要求のカスタマーおよびそのホーム組織単位に割り当てられます。メッセージのサブスクリプションセクションにカスタマーのログイン ID または組織単位名が指定されている場合、これらの値はデフォルトのサービス項目の割り当てを上書きするために使用されます。サブスクリプションの処理ルールの詳細については、『Cisco Service Portal Designer Guide』の「Service Designer」の章を参照してください。

更新メッセージでは、サービス項目属性を省略すると、属性値は変更されません。メッセージで属性が明示的に指定されているものの、値が含まれない場合、そのサービス項目の属性値はテキストフィールドでは空白に設定され、数値フィールドでは 0 に設定されます。削除メッセージについては、サービス項目属性およびサブスクリプション情報が無視されます。

```
<?xml version="1.0" encoding="UTF-8"?>
<message channel-id="3F2504E0-4F89-11D3-9A0C-0305E82C3301">
  <create>
    <serviceitem>
      <name>LaptopComputer</name>
      <serviceItemData>
        <serviceItemAttribute name="Name">LT-LENVT60-17032</serviceItemAttribute>
        <serviceItemAttribute name="Model">Thinkpad T60</serviceItemAttribute>
        <serviceItemAttribute name="Brand">LENOVO</serviceItemAttribute>
        <serviceItemAttribute name="Price">899.99</serviceItemAttribute>
        <serviceItemAttribute name="Memory">3</serviceItemAttribute>
        <serviceItemAttribute name="ManufactureDate">2009-04-15
12:00:00</serviceItemAttribute>
        <subscription>
          <loginID>jsmith</loginID>
          <ouname>Finance</ouname>
        </subscription>
      </serviceItemData>
    </serviceitem>
    <serviceitem>
      <name>DesktopComputer</name>
      <serviceItemData>
        <serviceItemAttribute name="Name">DT-DELLV200-02274</serviceItemAttribute>
        <serviceItemAttribute name="Model">Vostro 200</serviceItemAttribute>
        <serviceItemAttribute name="Brand">DELL</serviceItemAttribute>
        <serviceItemAttribute name="Price">755.99</serviceItemAttribute>
        <serviceItemAttribute name="Memory">4</serviceItemAttribute>
        <serviceItemAttribute name="ManufactureDate">2010-03-01
12:00:00</serviceItemAttribute>
      </serviceItemData>
    </serviceitem>
  </create>
</message>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<message channel-id="3F2504E0-4F89-11D3-9A0C-0305E82C3301">
  <update>
    <serviceitem>
      <name>LaptopComputer</name>
      <serviceItemData>
```

```

    <serviceItemAttribute name="Name">LT-LENVT60-6122</serviceItemAttribute>
    <serviceItemAttribute name="Memory">4</serviceItemAttribute>
    <subscription>
      <loginID>dcohen</loginID>
    </subscription>
  </serviceItemData>
</serviceitem>
<serviceitem>
  <name>DesktopComputer</name>
  <serviceItemData>
    <serviceItemAttribute name="Name">DT-DELLV200-00394</serviceItemAttribute>
    <subscription>
      <loginID></loginID>
      <ouname></ouname>
    </subscription>
  </serviceItemData>
</serviceitem>
</update>
</message>

<?xml version="1.0" encoding="UTF-8"?>
<message channel-id="3F2504E0-4F89-11D3-9A0C-0305E82C3301">
  <delete>
    <serviceitem>
      <name>LaptopComputer</name>
      <serviceItemData>
        <serviceItemAttribute name="Name">LT-TOSH900-0021</serviceItemAttribute>
      </serviceItemData>
    </serviceitem>
    <serviceitem>
      <name>DesktopComputer</name>
      <serviceItemData>
        <serviceItemAttribute name="Name">DT-DELLV100-0394</serviceItemAttribute>
      </serviceItemData>
    </serviceitem>
  </delete>
</message>

```

複合メッセージ

上記のメッセージタイプを組み合わせて、単一の着信メッセージにすることができます。このような組み合わせは「複合」メッセージと呼ばれます。実行順序が重要です。**take-action** タグを含める前に、パラメータの送信やコメントの追加を実行し、最後にサービス項目操作タグを配置する必要があります。

```

<?xml version="1.0" encoding="UTF-8"?>
<message channel-id="3F2504E0-4F89-11D3-9A0C-0305E82C3301">
  <add-comments>
    <comment>Task closed per override ...</comment>
  </add-comments>
  <send-parameters>
    <agent-parameter>
      <name>Status</name>
      <value>Resolved</value>
    </agent-parameter>
  </send-parameters>
  <take-action action="done"/>
<update>
  <serviceitem>
    <name>LaptopComputer</name>
    <serviceItemData>

```



```

<serviceItemAttribute name="Name">LT-LENVT60-6122</serviceItemAttribute>
<serviceItemAttribute name="Memory">4</serviceItemAttribute>
<subscription>
  <loginID>dcohen</loginID>
</subscription>
</serviceItemData>
</serviceitem>
</update>
</message>

```

SIM Import メッセージ

Service Item Manager (SIM) Import メッセージタイプでは、外部システムから Service Portal へのサービス項目と標準の定義、およびデータのインポートがサポートされます。サービス項目の作成/更新/削除の操作とは異なり、SIM のインポートは、特定のディレクトリに配置された着信ファイルポーリングする File Adapter プロトコルに基づいています。サービス項目インスタンスの操作に加えて、SIM Import ではサービス項目グループおよびサービス項目タイプのメンテナンスもサポートされます。Service Item Manager のインポートの詳細については、『Cisco Service Portal Designer Guide』を参照してください。

変換と nsXML

発信 nsXML メッセージは通常、非常にサイズが大きく複雑で、500KB を超えることもよくあります。変換を使用してメッセージ形式を変更することは必須ではありませんが、外部システムが nsXML を読み取るように設定されていることはまずありません。このため、通常は、変換を使用して発信メッセージの形式を変更ことが不可欠です。

ただし、着信メッセージの形式についてはサードパーティ システムの担当者と相談することが多いため、nsXML メッセージ形式に近い仕様に合意することも可能です。この場合は、着信変換が、対応する発信変換よりも大幅に単純になる可能性があります。

シスコでは XSL 変換 (XSLT) と呼んでいますが、使用されるテクノロジーは実際には eXtensible Stylesheet Language と呼ばれ、XPath も含まれます。XPath は、情報を検索したり、XML 文書内の要素や属性のナビゲーションを行うための言語です。XPath には、文字列値、数値、日付および時刻の比較、シーケンスの操作、ブール値、およびその他のメソッドのための組み込みの関数が含まれます。

Service Link エージェントの作成と導入

次の手順では、ファイルアダプタを使用する Service Link の統合の導入に必要なタスクの一般的なシーケンスを示します。これは Service Link のインストールの検証にも使用できます。

-
- ステップ 1** 発信ファイルアダプタを使用するエージェントを作成するには、Service Link のホーム ページの [Common Tasks] 領域から [Create Agent] ウィザードを選択し、場所のフィールドに入力し、その他の発信アダプタ プロパティを指定します。(このようなプロパティの詳細については、「[ファイルアダプタ](#)」(P.2-46) で説明します)。
- ステップ 2** エージェントを起動するには、[Control Agents] タブに移動してエージェントを見つけ、エージェント名 (エージェント定義へのリンク) 以外の任意の行をマウスでクリックして選択し、ページの右上にある [Start Selected] ボタンをクリックします。起動したかどうかを確認します。起動しない場合は、Service Link コンフィギュレーション設定のいずれかが間違っているか、Integration Server (ISEE) が正しく起動していません。

- ステップ 3** 入力したファイル ディレクトリがアプリケーション サーバに存在していることを確認します。なければ作成します。Service Portal と外部アプリケーションの両方に、ディレクトリへの適切なアクセス権（書き込み/読み取り）があることを確認します。これらの条件を満たしていない場合、実行時にファイルの送信が失敗します。
- ステップ 4** [Service Designer] に移動し、このエージェントを使用するサービスを作成します。
- ステップ 5** [My Services] に移動し、サービスをオーダーします。
- ステップ 6** 要求が正常に作成された場合は、ISEE 発信キューが動作しています。[our apologies] ページが表示された場合は、JMS キューが動作していません。
- ステップ 7** [View Transactions] タブからアクセス可能な [Messages] ページに移動します。作成した要求のメッセージが表示されれば成功です。メッセージのステータスが「Message sent」になっているはずですが。
- ステップ 8** 発信ファイル ディレクトリ（C:\cisco\SL\OutboundFiles など）に移動します。このディレクトリに XML ファイルが存在している場合（XML ファイルのタイム スタンプを確認して、新しい要求に対応したものであることを確認します）、ファイル エージェントの発信トリップは完了です。これですべての作業は完了しました。発信 XML ファイルは有効な nsXML メッセージになります。
- ステップ 9** [Message Type] カラムで、要求の [Execute Task] リンクをクリックします。[Message Details] ページが表示されます。
- ステップ 10** 要求 ID が正しいことを確認します。[Message Details] 画面から「Channel ID」をコピーします。
- ステップ 11** 次のように、SampleInbound.xml という名前の XML ファイルを作成します。「insert your Channel ID here」と表示されたら、前の手順でコピーした Channel ID の値を貼り付けます。（二重引用符もそのまま貼り付けます）。
- ```
<?xml version="1.0" encoding="UTF-8"?>
<message channel-id="insert your Channel ID here">
 <take-action action="done"/>
</message>
```
- たとえば、Channel ID 値を貼り付けた後、SampleInbound.xml ファイルは次のようになります。
- ```
<?xml version="1.0" encoding="UTF-8"?>
<message channel-id="3F2504E0-4F89-11D3-9A0C-0305E82C3301">
  <take-action action="done"/>
</message>
```
- ステップ 12** 着信ファイル ディレクトリ（C:\cisco\SL\InboundFiles など）に SampleInbound.xml ファイルを配置します。
- ステップ 13** File Agent が入力をポーリングすると、着信ファイルが自動的に選択されます。（File Adapter の Polling Interval Time のデフォルト設定は 10 秒間隔です）。SampleInbound.xml ファイルが正常に処理された場合は、ディレクトリ内から削除されます。
- ステップ 14** [View Transactions] タブの [Messages] ページに移動し、要求を検索します。要求に対して、Type=Take Action および Status=Inbound Message Completed となった別のメッセージがあれば、ラウンドトリップが完了しています。
- ステップ 15** [Requisition ID] リンクをクリックし、[Requisition Status] ページを開きます。要求のステータスが「Closed (1 of 1 completed)」になっていることを確認します。

Service Link トランザクションのモニタリング

Service Link の使用状況をモニタリングするには、次のように複数の方法があります。

- Service Link のホーム ページに、過去 30 日間のメッセージ量のグラフが表示され、他のモニタリング オプションにアクセスするための [Common Tasks] タブや [View Transactions] タブが表示されます。
- また、Service Link のホーム ページには [Recent Failed Messages] を表示するオプションもあり、配信できなかったすべてのメッセージが表示されます。
- [View Transactions] タブからアクセス可能なメッセージを表示するオプションでは、Service Link で送受信するすべてのメッセージが表示され、管理者は目的のメッセージを表示するためにフィルタリングや検索を実行できます。
- [View Transactions] タブからアクセスできる外部タスクを表示するためのオプションでは、Service Link メッセージを配信できなかったために進行中のままになっているすべてのタスクが表示され、管理者は目的のタスクを表示するためにフィルタリングや検索を実行できます。

Service Link のモニタリングや管理のためのすべてのページが、設定可能な「データ テーブル」を使用して表示されます。これらのテーブルの外観（表示されるカラム、各カラムの幅、データの表示順）はカスタマイズできます。さらに、管理者はフィルタおよび検索機能を使用して、目的の行だけを表示できます。

Service Link ホーム ページからのメッセージの表示

Service Link のホーム ページの [Recent Failed Messages] ペインには、過去 30 日以内に宛先に配信できなかった Service Link メッセージが表示されます。デフォルトでは、送信された日付と時刻に基づいて、メッセージが最新のものから順に表示されます。

Recent Failed Messages						
Message Type	Req ID	Agent	Task Subject	Date ▼	Resent On	Status Text
Take Action	166	Douglas_DB_Agent	configurable task d...	12/05/2011 03:55 PM		Inbound Message f
Take Action	166	Douglas_DB_Agent	configurable task d...	12/05/2011 03:54 PM		Inbound Message f
Composite	181	Douglas_DB_Agent	configurable task d...	12/05/2011 03:51 PM		Inbound Message f
Composite	180	Douglas_DB_Agent	DT1	12/05/2011 03:51 PM		Inbound Message f
Composite	179	Douglas_DB_Agent	configurable task d...	12/05/2011 03:50 PM		Inbound Message f
Composite	178	Douglas_DB_Agent	configurable task d...	12/05/2011 03:49 PM		Inbound Message f
Composite	177	Douglas_DB_Agent	configurable task d...	12/05/2011 03:47 PM		Inbound Message f
Unknown		Douglas_DB_Agent		12/05/2011 03:21 PM		Unknown Channel
Take Action	172	Douglas_DB_Agent	configurable task d...	12/02/2011 02:58 PM		Inbound Message f
Take Action	171	Douglas_DB_Agent	configurable task d...	12/02/2011 02:56 PM		Inbound Message f
Take Action	170	Douglas_DB_Agent	DT1	12/02/2011 02:06 PM		Inbound Message f
Take Action	170	Douglas_DB_Agent	DT1	12/02/2011 02:05 PM		Inbound Message f
Take Action	169	Douglas_DB_Agent	configurable task d...	12/02/2011 02:05 PM		Inbound Message f

[Failed Messages] グリッドのカラム リンクのいずれかをクリックすると、次の関連情報が表示されます。

表 2-7 Service Link の [Failed Messages] のクリック可能なカラム

カラム	リンク
Message Type	Service Link の [Message Details] ポップアップ ページに表示されるメッセージの詳細
Req ID	要求の詳細
Agent	Service Link の [Agents] ページに表示されるエージェントの詳細

[View Transactions] タブにある [Messages] ページでは、ステータスに関係なく着信と発信の両方のメッセージを表示し、ページに表示するメッセージを明示的にフィルタリングし、指定された検索条件と一致するメッセージを検索できます。

メッセージの表示

[Messages] ページには、設定したフィルタに応じて、すべてまたは選択した Service Link メッセージが表示されます。[Messages] ページを表示するには、Service Link のホーム ページの [View Transactions] タブを選択します。次に、[Messages] サブタブを選択します。Service Link ホーム ページの [Common Tasks] 領域にある [View Failed Messages] リンクでは、ステータスが「Failed」のメッセージだけを表示するようにフィルタが設定された [Messages] ページが表示されます。

[Messages] ページが表示されます。

Cisco Service Portal [admin admin] | Profile | Logout Service Link

Home Control Agents Manage Integrations View Transactions

Messages External Tasks

Filter and Search

Direction	Message Type	Status	Status Text	Date	Req ID	Agent Name	Task Subject	Resent on
Inbound	SIM Import	Completed	Inbound Messag...	03/19/2012 11:19...		SIimportAgent	No task - SIM Imp...	
Inbound	SIM Import	Failed	Internal applicatio...	03/19/2012 11:17...		SIimportAgent	No task - SIM Imp...	
Inbound	SIM Import	Completed	Inbound Messag...	03/17/2012 06:06...		SIimportAgent	No task - SIM Imp...	
Outbound	Execute Task	Completed	Message sent	03/16/2012 12:02...	378	DummyAgentMin...	Dummy adapter -...	
Outbound	Execute Task	Waiting	Agent Stopped.M...	03/16/2012 12:02...	378	DummyAgentMed...	Dummy adapter -...	
Outbound	Execute Task	Waiting	Agent Stopped.M...	03/16/2012 12:02...	378	DummyAgentMed...	Dummy adapter -...	
Outbound	Execute Task	Completed	Message sent	03/16/2012 12:02...	378	DummyAgentLar...	Dummy adapter -...	
Outbound	Execute Task	Completed	Message sent	03/15/2012 06:36...	377	DummyAgentMed...	Dummy adapter -...	
Outbound	Execute Task	Completed	Message sent	03/15/2012 06:36...	377	DummyAgentMin...	Dummy adapter -...	
Outbound	Execute Task	Completed	Message sent	03/15/2012 06:36...	377	DummyAgentMed...	Dummy adapter -...	
Outbound	Execute Task	Completed	Message sent	03/15/2012 06:36...	377	DummyAgentLar...	Dummy adapter -...	
Outbound	Execute Task	Completed	Message sent	03/15/2012 05:36...	376	RAPIAddComment	RAPI Add come...	
Outbound	Send Parameters	Failed	Inbound Messag...	03/15/2012 05:36...	376	RAPIAddComment	RAPI Add come...	
Outbound	Execute Task	Completed	Message sent	03/15/2012 03:51...	131	SI Task Agent	SI creation from ...	
Outbound	Execute Task	Completed	Message sent	03/15/2012 03:51...	131	SI Task Agent	SI creation from ...	
Outbound	Execute Task	Completed	Message sent	03/15/2012 03:50...	131	SI Task Agent	SI creation from ...	
Inbound	Composite	Failed	Inbound Messag...	03/15/2012 02:47...	356	SI Task Agent	SI creation from ...	
Inbound	Composite	Processing	Service Item Mes...	03/15/2012 02:47...	356	SI Task Agent	SI creation from ...	
Inbound	Add Comment	Completed	Inbound Messag...	03/15/2012 02:47...	356	SI Task Agent	SI creation from ...	
Inbound	Add Comment	Completed	Inbound Messag...	03/15/2012 02:45...	361	SI Task Agent	SI inbound actions	

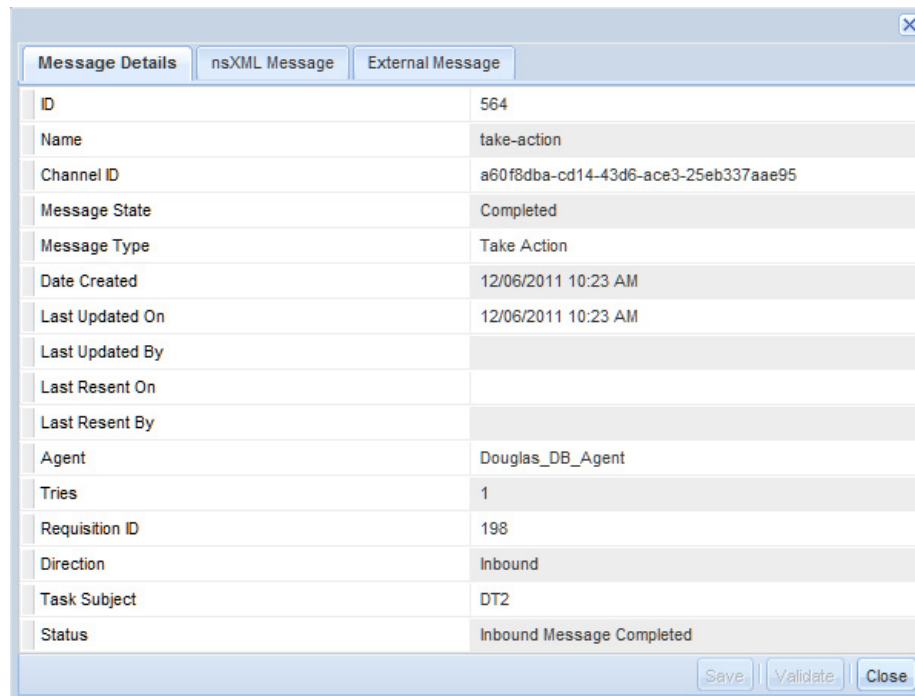
Resend Message Page 1 of 16 Displaying 1 - 20 of 306

表 2-8 Service Link の [Messages] のクリック可能なカラム

カラム	リンク
Message Type	Service Link の [Message Details] ポップアップ ページに表示されるメッセージの詳細。
Status Text	失敗したメッセージを参照するために、アダプタに固有のログ ファイルおよびサーバ ログに書き込まれたエラー メッセージへのリンクを使用できます。詳細については、「 Service Link のトラブルシューティングと管理 」(P.2-64) を参照してください。
Req ID	要求の詳細。
Agent	Service Link の [Agents] ページに表示されるエージェントの詳細。
Task Subject	Service Manager のタスクの詳細。

Message Details

[Message Details] ポップアップ ページでは、Service Portal と外部メッセージのすべてのメッセージが表示できます。このページには、この要求のタスクを一意に識別するチャンネル Id も表示されます。サードパーティ システムに関する問題を調査する場合は、この Id を使用できます。



The screenshot shows a dialog box titled "Message Details" with three tabs: "Message Details", "nsXML Message", and "External Message". The "Message Details" tab is active, displaying a table of message information. At the bottom right of the dialog are three buttons: "Save", "Validate", and "Close".

Field	Value
ID	564
Name	take-action
Channel ID	a60f8dba-cd14-43d6-ace3-25eb337aae95
Message State	Completed
Message Type	Take Action
Date Created	12/06/2011 10:23 AM
Last Updated On	12/06/2011 10:23 AM
Last Updated By	
Last Resent On	
Last Resent By	
Agent	Douglas_DB_Agent
Tries	1
Requisition ID	198
Direction	Inbound
Task Subject	DT2
Status	Inbound Message Completed

[Message Details] ポップアップ ページでいずれかのタブをクリックすると、関連情報が表示されます。

表 2-9 Service Link の [Message Details] のサブタブ

カラム	リンク
Message Details	メッセージの詳細。
nsXML Message	発信メッセージの場合は、Business Engine によって生成されたメッセージが、Service Link エージェントによって処理（変換）されます。着信メッセージの場合は、外部システムから受信したメッセージが、エージェント変換（存在する場合）によって変換され、Business Engine によって処理されます。
External System Message	発信メッセージの場合、エージェントに関連付けられた変換が適用された後のメッセージです。着信メッセージの場合、外部システムから受信したメッセージです。

フィルタおよび検索

検索機能を使用して、メッセージのサブセット（ステータスが **Failed** のすべてのメッセージなど）を表示できます。検索では、検索対象として [Messages] ウィンドウのいずれかのカラムを指定し、照合する値を選択または入力します。

検索条件は、セッションや Service Link のホーム ページの表示間で維持されます。検索条件を削除するには、[Filter and Search] ボタン ([Messages] ページの上部) をクリックします。次に、[Clear] をクリックします。

[Filter and Search] ダイアログでは、次のことも実行できます。

- カラムの意味に適した関係演算子を使用して、特定のカラムをフィルタリングする。たとえば、日付の範囲を選択したり、指定されたステータスと等しくないステータスを選択したりできます。
- カラムに指定されたすべての条件の論理「AND」によってフィルタリングする。

[Filter and Search] ダイアログ ボックスはノンモーダルです。目的の条件を入力して [Apply] をクリックすると、現在の設定の結果が表示されます。必要に応じて、設定を調整して再適用します。任意のカラムの昇順または降順でメッセージを表示したり、「Service Link 画面の管理」(P.2-7) で説明した方法を使用して、表示されるカラムを変更することもできます。

失敗したメッセージの再送信

Service Link の開発中は、エージェントまたは変換設定のエラーのために配信に失敗した、多数のメッセージが生成されることがあります。これらのメッセージは再送信しないでください。同様に、Service Item Import タスクによって生成されたメッセージも再送信しないでください。インポートファイル形式を変更して、インポート タスクを再実行してください。

ただし、実稼働環境では、外部システムの停止または解決することが可能なその他の外部要因のために、メッセージの配信が失敗する場合があります。配信失敗の原因の修正後は、失敗したメッセージを再送信できます。

失敗したメッセージを再送信するには、次の手順を実行します。

-
- ステップ 1** [View Transactions] タブの [Messages] ページに一覧表示された、失敗したメッセージが含まれる行を選択します。
- ステップ 2** [Messages] ページの左下にある [Resend Message] ボタンをクリックします。
-

Service Link は指定された宛先に、メッセージを再送信しようとします。再送信に成功すると、メッセージのステータスおよび日付が更新され、再送信日が記録されて、[Resent On] カラムに表示されます。

メッセージの再送信時、変換は再適用されません。エージェントは、すでに変換済みのメッセージを宛先に送信しようとします。

サービス項目の操作に関する失敗した着信メッセージの再送信はサポートされません。このプロセスでは、再実行タスク アクションが試行されます。したがって、このようなメッセージの宛先は、Service Item インポート プロセッサではなく、Business Engine になります。

外部タスクの表示

外部タスクを表示するには、次の手順を実行します。

-
- ステップ 1** Service Link のホーム ページの [View Transactions] タブを選択します。次に、[External Tasks] サブタブを選択します。
- 次に示す、[External Tasks] ページが表示されます。

Messages		External Tasks				
Task Subject	Started On	Req ID	Status	Completed On	Agent Name	Service
DT2	12/06/2011 10:22 AM	198	Completed	12/06/2011 10:23 AM	Douglas_DB_Agent	configurable task db
DT1	12/06/2011 10:22 AM	198	Completed	12/06/2011 10:22 AM	Douglas_DB_Agent	configurable task db
configurable task db needs...	12/06/2011 10:22 AM	198	Skipped	12/06/2011 10:22 AM	Douglas_DB_Agent	configurable task db
configurable task db needs...	12/06/2011 10:22 AM	198	Skipped	12/06/2011 10:22 AM	Douglas_DB_Agent	configurable task db
DT2	12/06/2011 10:21 AM	197	Completed	12/06/2011 10:22 AM	Douglas_DB_Agent	configurable task db
DT1	12/06/2011 10:21 AM	197	Completed	12/06/2011 10:21 AM	Douglas_DB_Agent	configurable task db
configurable task db needs...	12/06/2011 10:21 AM	197	Skipped	12/06/2011 10:21 AM	Douglas_DB_Agent	configurable task db
configurable task db needs...	12/06/2011 10:21 AM	197	Skipped	12/06/2011 10:21 AM	Douglas_DB_Agent	configurable task db
DT2	12/06/2011 10:20 AM	196	Completed	12/06/2011 10:21 AM	Douglas_DB_Agent	configurable task db
DT1	12/06/2011 10:19 AM	196	Completed	12/06/2011 10:20 AM	Douglas_DB_Agent	configurable task db
configurable task db needs...	12/06/2011 10:19 AM	196	Skipped	12/06/2011 10:19 AM	Douglas_DB_Agent	configurable task db
configurable task db needs...	12/06/2011 10:19 AM	196	Skipped	12/06/2011 10:19 AM	Douglas_DB_Agent	configurable task db
DT2	12/06/2011 10:19 AM	195	Completed	12/06/2011 10:19 AM	Douglas_DB_Agent	configurable task db
DT1	12/06/2011 10:18 AM	195	Completed	12/06/2011 10:19 AM	Douglas_DB_Agent	configurable task db
configurable task db needs...	12/06/2011 10:18 AM	195	Skipped	12/06/2011 10:18 AM	Douglas_DB_Agent	configurable task db
configurable task db needs...	12/06/2011 10:18 AM	195	Skipped	12/06/2011 10:18 AM	Douglas_DB_Agent	configurable task db
DT1	12/05/2011 05:08 PM	194	Ongoing		Douglas_DB_Agent	configurable task db
configurable task db needs...	12/05/2011 05:08 PM	194	Skipped	12/05/2011 05:08 PM	Douglas_DB_Agent	configurable task db
configurable task db needs...	12/05/2011 05:08 PM	194	Skipped	12/05/2011 05:08 PM	Douglas_DB_Agent	configurable task db
DT1	12/05/2011 05:07 PM	193	Ongoing		Douglas_DB_Agent	configurable task db

Send Manual Message Page 1 of 30 Displaying 1 - 20 of 593

ステップ 2 次のカラム リnkのいずれかをクリックすると、関連情報が表示されます。

表 2-10 Service Link の [External Tasks] のクリック可能なカラム

カラム	リンク
Task Subject	Service Manager のタスクの詳細
Req	[My Services] の要求の概要
Agent Name	Service Link の [Agents] ページに表示されるエージェントの詳細

フィルタおよび検索

[Messages] の表示と同様に、[External Tasks] ページではデータ テーブルに表示されるデータの列および順序をカスタマイズしたり、データのフィルタリングや検索を実行できます。

手動メッセージの送信

開始済みで、着信メッセージの受信を待機しているタスクは、「Ongoing」状態です。通常、着信メッセージではタスクが更新されるか、そのステータスが変更されます。メッセージが受信され、タスクが完了するまでは、要求の配信計画にある後続のタスクを実行できません。予測されるメッセージが送信されたものの、「消失した」と思われる（または、外部システムの管理者に問い合わせることによって確認できる）場合、手動メッセージを送信することによって、メッセージの受信をエミュレートできます。

手動メッセージを使用して、失敗したサービス項目の操作をエミュレートすることはできません。

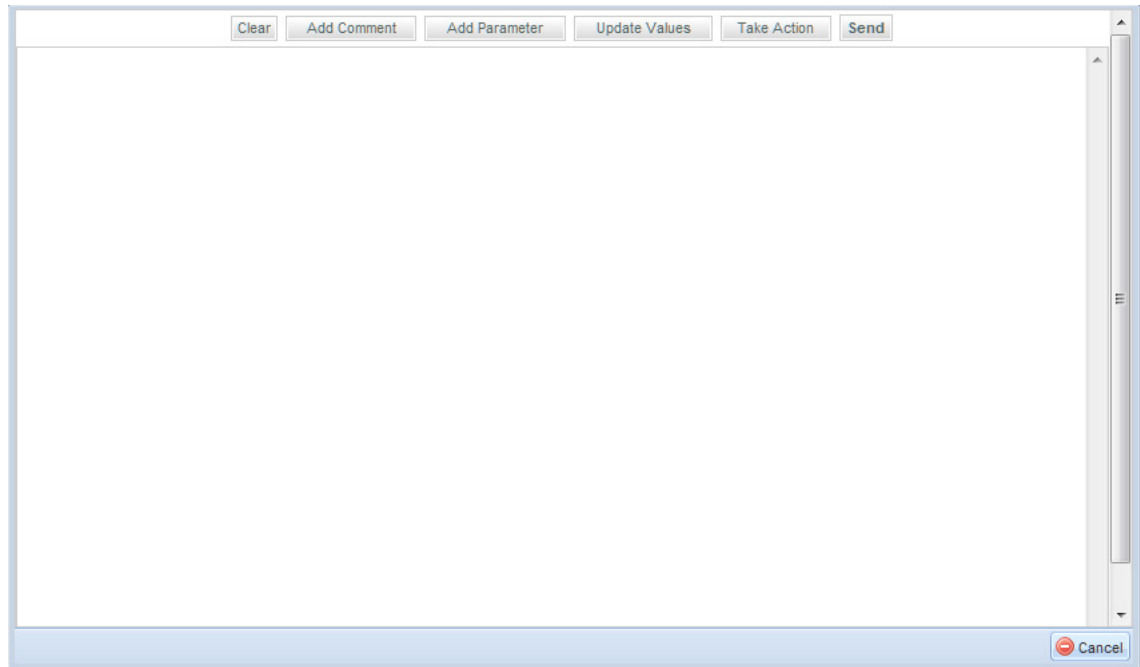


(注)

この機能を利用する場合は注意が必要です。この機能はシステム内のすべての通信プロトコルを上書きします。この機能を使用すると、サードパーティ システムの作成物がそのままになり、Service Link が応答できなくなる可能性があります。また、この機能を使用して要求をキャンセルした場合、たとえば、Service Link が関係先に通知を行わないため、自分でフォローアップする必要があります。

手動メッセージを Business Engine に送信するには、次の手順に従います。

- ステップ 1** Service Link のホーム ページの [View Transactions] をクリックします。次に、[External Tasks] をクリックします。
[External Tasks] ページが表示されます。
- ステップ 2** 手動メッセージを送信するタスクが含まれた行を選択します。
- ステップ 3** [External Tasks] ページの左下にある [Send Manual Message] ボタンをクリックします。
[Send Manual Message] ダイアログが表示されます。



- ステップ 4** 送信するメッセージのタイプに応じて [Add Comment]、[Add Parameter]、[Update Values]、または [Take Action] をクリックします。

アクション	説明
Add Comment	add-comments メッセージを送信し、システム コメントを要求に追加します。
Add Parameter	send-parameters メッセージを送信します。1 つまたは複数の着信エージェント パラメータ値と、パラメータがバインドされる対応するディクショナリ フィールドの値を変更します。
Update Values	指定されたディクショナリ フィールドの内容を変更するメッセージを送信します。(このメッセージタイプは、主に以前のバージョンとの下位互換性のために用意されています。通常、フィールドの内容は着信エージェント パラメータで更新する必要があります)。
Take Action	take-action メッセージを送信します。Mark タスクが実行された (完了した) かキャンセルされたかをマークし、承認を許可または拒否し、あるいは確認が OK とマークします。

- ステップ 5** 選択したメッセージタイプに関連するポップアップ ダイアログ ボックスに応答します (ポップアップ ブロックはオフにします)。これによって、メッセージ ウィンドウに適切なタイプの正しい形式の XML メッセージが入力されます。また、このメッセージは、通常のチャンネルを介して受信されず、手動で生成されたことを示す <add-comments> メッセージに含まれます。
- ステップ 6** 必要な場合は、生成されたメッセージを編集できます。メッセージの全文を作成したら、[Send] ボタンをクリックします。着信メッセージが Business Engine に送信されます。

Service Link メッセージの再発行

まれに、Service Link アプリケーションが長時間停止したり、コンフィギュレーションが正しくないために、外部タスクに Service Link で作成された対応する発信メッセージを含めることができないことがあります。

Service Link で根本原因が解決し、アプリケーションが再び起動すると、問題の外部タスクを Service Link に再発行し、発信メッセージを作成して配信プロセスを再開することができます。

発信メッセージを再発行するには、次の手順に従います。

-
- ステップ 1** Service Link のホーム ページの [View Transactions] タブを選択します。次に、[Message Republish] サブタブを選択します。
 - ステップ 2** 左側のパネルに、1 つまたは複数の不明な発信 Service Link メッセージがある要求の Requisition ID を入力します。要求に関連するすべての承認タスクおよび配信タスクが評価され、発信メッセージの作成のために再発行が必要なタスクだけが処理されます。一度に最大 20 の要求を入力できます。
 - ステップ 3** [Republish] をクリックします。
 - ステップ 4** 再発行プロセスが完了したら、右側のパネルで処理のステータスを確認します。
-

Service Link アダプタ

すべての Service Link アダプタで、データ交換形式として nsXML がサポートされます。nsXML 形式の詳細については、第 3 章「[Service Link Adapter Development Kit](#)」を参照してください。

すべてのポーラーベースのアダプタで、一度の呼び出しにつき 1 つのメッセージの処理だけがサポートされます。

すべてのアプリケーション インスタンスに次の Service Link アダプタがインストールされます。

- [ダミー アダプタ](#)
- [データベース アダプタ](#)
- [ファイル アダプタ](#)
- [HTTP/WS アダプタ](#)
- [JMS アダプタ](#)
- [MQ アダプタ](#)
- [サービス項目リスナー アダプタ](#)
- [VMware アダプタ](#)
- [Web サービス リスナー アダプタ](#)

これらのアダプタに加えて、Service Link では [Auto-Complete アダプタ](#) がサポートされます。

追加アダプタは、Service Link Adapter Development Kit (ADK) を使用してインストールおよび設定できます。このようなカスタム アダプタも [Adapters] ページに表示され、プロパティを確認できます。カスタム アダプタの作成およびインストールの詳細については、第 3 章「[Service Link Adapter Development Kit](#)」を参照してください。

次の項では、これらのアダプタについて説明します。

Auto-Complete アダプタ

Auto-Complete アダプタでは、エージェントが発信メッセージを送信し、外部システムからの確認応答の受信を待機することなく、タスクが完了済みであるとマークできます。発信メッセージが正常に送信された（たとえば、発信ファイルアダプタによってファイルが特定のディレクトリに書き込まれた）場合、auto-complete アダプタが同じタスクの着信メッセージを生成します。着信メッセージのメッセージタイプは「take-action」です。通常、このメッセージは Business Engine によって処理され、アクションが完了済みであるとマークされ、外部タスクが完了します。

ダミー アダプタ

ダミー アダプタはプレースホルダです。これは、次のいくつかの処理シナリオで使用できます。

- ダミーアダプタを着信アダプタとして使用すると、Service Link で外部タスクを開始し、「Ongoing」ステータスのままにすることができます。
- ダミーアダプタを発信アダプタとして使用し、auto-complete アダプタを着信アダプタとして使用すると、サービス設計者は外部タスクで Auto-Complete エージェントを実装できます。次に、このタスクをワークフローの一部として使用できます。たとえば、参加者に電子メールを送信したり、他のタスクがない要求をクローズにしたりできます。また、この組み合わせを使用して、Request Center と Service Link の間の通信が正常に動作していることを確認できます。

データベース アダプタ

データベース (DB) アダプタでは、データベースの 1 つまたは複数のテーブルを使用して、Service Portal と外部アプリケーションの間でデータを渡します。

データベース接続

着信および発信データベースアダプタは、ANSI 規格の SQL をサポートする、JDBC に準拠したリレーショナルデータベースと通信できます。JDBC URL、データベースドライバ、および有効な接続基準を指定する必要があります。外部データベースが SQLServer または Oracle の場合、シスコ提供のドライバを使用できます。シスコが提供するドライバには、次のものがあります。

```
com.newscale.jdbc.sqlserver.SqlServerDriver
com.newscale.jdbc.oracle.OracleDriver
com.newscale.jdbc.UnifiedDriver
```

この統合ドライバでは、Service Portal でサポートされるすべてのデータベース (SQLServer および Oracle) がサポートされ、データベース固有のドライバよりも優先して使用する必要があります。データベース固有のドライバは、以前のバージョンの Service Portal との下位互換性のためにサポートされます。

JDBC URL は次の形式です。

```
jdbc:newscale:dbtype://machine:port;DatabaseName=dbname
jdbc:newscale:dbtype://machine:port;SID=sid
```

ここで、

- dbtype は sqlserver または oracle です
- machine は、データベース サーバの名前です

- port はデータベースに接続するポートです。通常は SQLServer の場合は 1433、Oracle の場合は 1521 です
- SQLServer にはデータベース名を指定し、Oracle には SID (システム ID) を指定する必要があります

例：

```
jdbc:newscale:sqlserver://localhost:1433;DatabaseName=RCDev
```

```
jdbc:newscale:oracle://PRODSRVR2:1521;SID=RCProd
```

サポート jar ファイルが Request Center ディレクトリ構造のディレクトリ ISEE.war/WEB-INF/lib にインストールされている場合、ユーザ指定のドライバを使用できます

-
- ステップ 1** 適切なサードパーティ JDBC ドライバを入手します。たとえば、Sybase JDBC Driver は Sybase の Web サイトからダウンロードできます。Oracle ドライバはオンラインで入手するか、Oracle 製品に付属しています。
- ステップ 2** 必要なすべての jar を ISEE.war/WEB-INF/lib フォルダにコピーします。
- ステップ 3** カスタム ドライバおよび正しい JDBC URL 形式を使用するようにエージェントの設定を変更します。たとえば、Sybase ドライバの JDBC URL は次の形式です。
- ```
jdbc:sybase:Tds:host:port/database
```
- Oracle シン ドライバは次の形式です。
- ```
jdbc:oracle:thin:@host:port
```
- ステップ 4** Service Link および Request Center サービスを再起動します。
-

JDBCUrl の形式は、Service Link が導入されたアプリケーション サーバによっても異なります。たとえば、WebSphere アプリケーション サーバから SQLServer データベースへの接続を確立する JDBC URL は次のようになります。

```
jdbc:newscale:sqlserver://hostname;DatabaseName=databasename;selectMethod=direct;alwaysReportTriggerResults=true;insensitiveResultSetBufferSize=16384;useServerSideUpdatableCursors=false;maxPooledStatements=0
```

着信プロパティ

データベース アダプタを着信アダプタとして使用する場合、エージェントのプロパティに、指定したデータベース接続に対して実行される SQL ステートメントを含めます。通常、SQL は行のセットを返す select コマンドです。これらの行は外部 XML メッセージの形式です。メッセージは、(エージェントで指定される) 着信変換によって有効な nsXML 着信メッセージに変換する必要があります。次に、メッセージが Business Engine に渡されます。Business Engine が、着信メッセージで指定された Channel ID で識別される open タスクを見つけると、着信メッセージが処理され、指定されたアクションが実行されます。

データベース着信アダプタのプロパティシートでは、次に示すプロパティ名に「DBInboundAdapter」というプレフィックスが付けられます。

表 2-11 DB アダプタの着信プロパティ

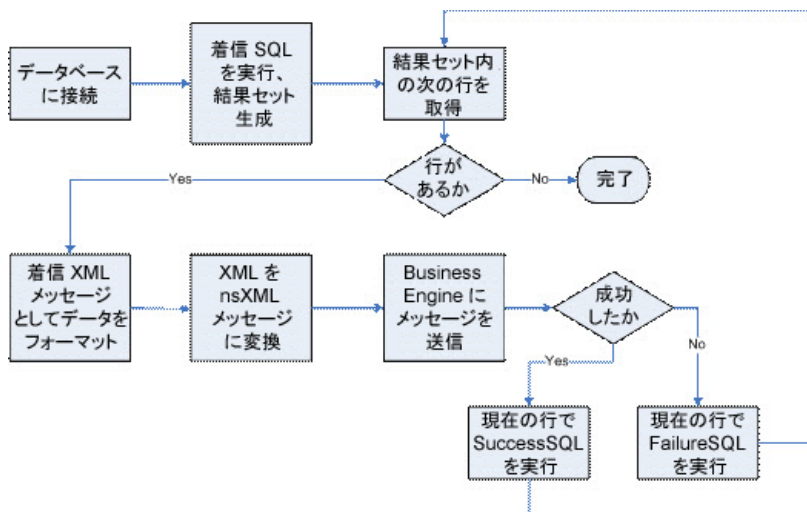
プロパティ	説明
DBPassword	指定されたユーザのパスワード。

表 2-11 DB アダプタの着信プロパティ (続き)

DBUserName	データベース ユーザの名前。
InboundSql	着信トランザクションに対して実行される SQL ステートメント。これは、行のセットを返す SELECT ステートメントにする必要があります。トランザクション SQL (プロシージャ) はサポートされません。
InboundSuccessSql	着信トランザクションの成功時に実行する SQL です。通常は現在の行を正常に処理されたとマークする SQL の update または delete ステートメントです。
InboundFailureSql	着信トランザクションの失敗時に実行する SQL です。通常は現在の行を正常に処理されたとマークする SQL の update または delete ステートメントです。
JDBCUrl	データベースに接続する JDBC URL
JDBCDriverClass	データベースへの接続に使用されるドライバのクラス名。

着信メッセージとワークフロー

次に、着信データベース アダプタの処理フローを示します。



結果セットの各行に対して、アダプタは次の構造の XML メッセージを生成します。

- メッセージのルート要素は <inbound-results> です。
- 必要な子要素は <row> です。メッセージごとに <row> 要素は 1 つだけです。
- 各 <row> 要素には複数の <column> 要素があり、各カラムの要素は、アダプタに指定された InboundSQL ステートメントに含まれます。
- <row> 要素にはカラム名 (<name>) と JDBC データタイプ (<type>)。文字の場合は 12、数値の場合は 1) の属性があります。
- 各 <column> 要素の値は、SQL ステートメントの対応するカラムに返される値です。

たとえば、次の SQL ステートメントがあるとします。

```
SELECT channelId, task, status, processType FROM rcInterface
WHERE status = 'UPDATED'
```


この場合、次のような XML ストリームが生成されます。

```
<?xml version='1.0' encoding='UTF-8'?>
<inbound-results>
  <row>
    <column name="channelid" type="12" >
      "3F2504E0-4F89-11D3-9A0C-0305E82C3301"
    </column>
    <column name="task" type="12" >Task</column>
    <column name="status" type="12" >UPDATED</column>
    <column name="processtype" type="1" >null</column>
  </row>
</inbound-results>
```

次に、変換がこの XML ストリームに適用され、有効な nsXML 着信メッセージが生成される必要があります。たとえば、進行中タスクを完了する変換には、次のコードが含まれます。

```
<xsl:template match="/inbound-results/row">
  <xsl:variable name="status" select="column[@name='status']" />
  <xsl:choose>http://www.w3schools.com/xsl
    <xsl:when test="$status='Complete'">http://www.w3schools.com/xsl
      <message>http://training2.cisco.com/RequestCenter
        <xsl:attribute name="channel-id">
          <xsl:value-of select="column[@name='channelId']" />
        </xsl:attribute>http://training2.cisco.com/ServiceLink
        <take-action action="done" />
      </message>
    <xsl:otherwise>
```

得られた nsXML メッセージを Business Engine が処理します。メッセージが正常に適用された場合、エージェントで指定された SuccessSQL が実行されます。通常、SuccessSQL は、処理用の行が選択されたソース テーブル内のカラムを更新し、次のポーリング間隔でその行が再び検出されることがありません。Service Link が現在の行を更新するように指定するには、行の固有識別子を構成するカラムを識別します。これらのカラムは着信 SQL ステートメントに含まれてはならずです。次に例を示します。

```
UPDATE rcInterface
  SET status = 'DONE'
  WHERE channelId = #ChannelId#
```

同様に、Business Engine が nsXML メッセージの適用に失敗した場合、たとえば、メッセージの処理中にエラーが発生した場合、FailureSQL が実行されます。通常、FailureSQL は現在の行が正しく処理されなかったことを示すように、その行のステータスを更新します。次に例を示します。

```
UPDATE rcInterface
  SET status = 'FAILED'
  WHERE channelId = #ChannelId#
```

発信プロパティ

データベース アダプタを発信アダプタとして使用する場合は、Service Portal と外部システムの間に「ステージング テーブル」スタイルのインターフェイスを提供します。Business Engine によってエージェントに提供される nsXML 発信メッセージは、1 つまたは複数の SQL ステートメントが含まれる外部メッセージに変換する必要があります。次に、指定された接続を使用して、指定されたデータベースでこれらの SQL ステートメントが実行されます。

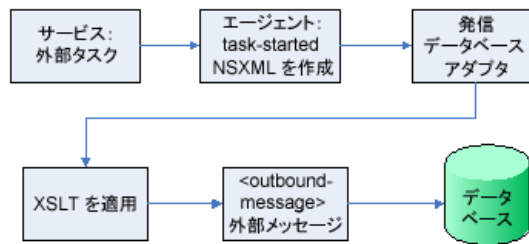
DB 発信アダプタのプロパティ シートでは、次に示すプロパティ名に「DBOutboundAdapter」というプレフィックスが付けられます。

表 2-12 DB アダプタの発信プロパティ

プロパティ	説明
DBPassword	データベース ユーザのパスワード
DBUserName	データベースのユーザ名
JDBCUrl	データベースに接続する JDBC URL
JDBCDriverClass	データベースへの接続に使用される特定のドライバのクラス名

発信メッセージとワークフロー

次に、発信データベース アダプタの処理フローを示します。



XSLT 変換によって生成された発信メッセージは、次の形式になっている必要があります。

```

<?xml version="1.0" encoding="UTF-8"?>
<outbound-message>
  <execute-sql-list>
    <execute-sql> SQLStatement
  </execute-sql>
</execute-sql-list>
</outbound-message>
  
```

メッセージには複数の SQL ステートメントを含めることができ、それぞれは `<execute-sql>` タグで囲みます。通常、これらのステートメントでは SQL テーブルに行が挿入されるか、更新されます。アダプタに対して指定された JDBC ドライバでサポートされる、SQL ステートメントを使用できます。ストアドプロシージャ (SQLServer Transact-SQL または Oracle PL/SQL 内) はサポートされませんが、SQL ステートメントにユーザ定義の関数を含めることはできます。各外部タスクは Channel ID によって一意に識別されますが、着信メッセージによってタスクを更新できるように、発信 SQL ステートメントのターゲット テーブルには Channel ID のカラムを含める必要があります。

ファイル アダプタ

ファイル アダプタでは、指定されたディレクトリからのファイルの読み取り、または指定されたディレクトリへのファイルの書き込みがサポートされます。

- 複数のディレクトリ、または指定されたディレクトリのサブディレクトリにあるファイルを処理するように、アダプタを設定することはできません。
- 着信ファイル アダプタの起動時に、ディレクトリにある複数のファイルの中で最も古いファイルが処理されます。
- 指定されたディレクトリに対して、ファイルを処理するように設定するエージェントは 1 つだけにする必要があります。

- 指定するディレクトリ（場所）は、Request Center がインストールされているか、またはアプリケーション サーバからアクセスできるアプリケーション サーバのファイル システム上にする必要があります。Service Link の処理が進行すると、1 つのディレクトリから別のディレクトリにファイルが移動されるため、すべてのディレクトリが同じ物理デバイス上に存在している必要があります。

ファイル アダプタの着信プロパティ

次に、ファイル アダプタのプロパティとデフォルト値を示します。

ファイル着信アダプタのプロパティシートでは、次に示すプロパティ名に「FileInboundAdapter」というプレフィックスが付けられます。

表 2-13 ファイル アダプタの着信プロパティ

プロパティ	説明
BackupLocation	Final Resolution または OnError プロパティが「Preserve」の場合、ファイルが処理後にバックアップされる場所。
BackupSuffix	バックアップ ファイルのファイル拡張子。デフォルトは .bak。
FileLocation	読み込まれる着信ファイルがポーリングされる場所（ディレクトリ）。各着信ファイル アダプタに対して一意の場所を使用する必要があります。
FileNameDateFormat	ファイルの日付形式。デフォルトは .yyyyMMddHHmmssSSS。
FinalResolution	トランザクションの完了後にファイルに対して実行するアクション。オプションは次のとおりです。 <ul style="list-style-type: none"> [Preserve]：ファイルをバックアップ場所に移動します [Delete]：ファイルを削除します デフォルトは [Preserve] です。
OnError	エラーの発生時にファイルに対して実行するアクション。オプションは次のとおりです。 <ul style="list-style-type: none"> [Preserve]：ファイルをバックアップ場所に移動します [Delete]：ファイルを削除します デフォルトは [Preserve] です。
TempLocation	着信ファイルの処理に使用される一時フォルダ。

ファイル アダプタの発信プロパティ

発信ファイル アダプタは、指定されたファイルの場所に XML ファイルを生成します。ファイル名に channel-id が含まれます。これは、エージェントが含まれ、メッセージを作成する外部タスクの固有識別子です。ファイル名の末尾には、発信プロパティとして指定された日付形式が付加されます。

ファイル発信アダプタのプロパティシートでは、次に示すプロパティ名に「FileOutboundAdapter」というプレフィックスが付けられます。

表 2-14 ファイルアダプタの発信プロパティ

プロパティ	説明
BackupLocation	発信ファイルのバックアップ場所。アプリケーション サーバからアクセス可能な、任意の有効なファイル システム ディレクトリなどです。
BackupSuffix	バックアップ ファイルのファイル拡張子。デフォルトは .bak。
ConflictResolution	発信ファイル名に競合がある場合に実行するアクション。オプションは次のとおりです。 <ul style="list-style-type: none"> [Preserve] : ファイルをバックアップ場所に移動します [Delete] : ファイルを削除します デフォルトは [Rename] です。
FileLocation	ファイルが書き込まれる場所。アプリケーション サーバからアクセス可能な任意のディレクトリなどです。
FileNameDateFormat	ファイル名の日付形式。デフォルトは .yyyyMMddHHmmssSSS です。
OnError	エラーの発生時にファイルに対して実行するアクション。オプションは次のとおりです。 <ul style="list-style-type: none"> [Preserve] : ファイルをバックアップ場所に移動します [Delete] : ファイルを削除します デフォルトは [Preserve] です。
TempLocation	発信ファイルの処理に使用される一時フォルダ。

HTTP/WS アダプタ

HTTP/WS アダプタは、HTTP 要求または Web サービス要求および応答を送受信するために使用されます。HTTPS もサポートされます。

Web サービスの呼び出しに使用された場合は、同期呼び出しだけが可能です。発信変換は、メッセージが SOAP ラッパーで囲まれるように設定する必要があります。

発信プロパティ

HTTP/WS アダプタ発信プロパティは、発信アダプタの動作を指定します。

http/ws 発信アダプタのプロパティ シートでは、次に示すプロパティ名に「HttpOutboundAdapter」というプレフィックスが付けられます。

表 2-15 HTTP/WS アダプタの発信プロパティ

プロパティ	説明
WsdURL	実行される操作が含まれた wsdl の URL。Integration ウィザードだけで使用されます。
WsdOperation	Web サービスによって実行される操作。Integration ウィザードを使用する場合を除いて、ドキュメントだけです。指定された WSDL に含まれるすべての操作のドロップダウン リストを使用できます。

表 2-15 HTTP/WS アダプタの発信プロパティ (続き)

RoutingURL	ポストされるすべての発信メッセージをルーティングする URL。Web サービスのエンドポイント。
AcceptUntrustedURL	外部システムからの信頼できない証明書を許容可能にするオプション。デフォルトは true です。
ContentType	コンテンツ タイプ。デフォルトは「text/xml; charset=ISO-8859-1」。
TimeOut	http url 接続を取得するためのタイムアウト。デフォルトは 180,000 マイクロ秒です。
ProcessResponse	着信メッセージとしての SOAP メッセージのポストまたは応答の結果を処理するオプション。デフォルトは false です。
RequestHeaders	HTTP 要求ヘッダーに含める必要があるカスタム ヘッダー パラメータ。通常は名前と値のペアの形式で、パラメータをアンパサンド (&) で区切ります。たとえば、SOAPAction は次の形式で入力します。 SOAPAction=OpCreate SOAPAction および「参照元」と呼ばれるカスタム ヘッダーは次の形式で入力します。 SOAPAction=OpCreate&referrer=www.test.com
AuthenticationSchema	Web サービスを要求するか、URL にポストするために使用される認証のタイプ。オプションは basic、anonymous、digest、または NTLM です。これらのオプションについては、後述の説明を参照してください。
AuthenticationScopeHost	認証クレデンシャルを適用するホスト。クレデンシャルを任意のホストに適用できる場合、空白にします。
AuthenticationScopePort	認証クレデンシャルを適用するポート。クレデンシャルを任意のポートに適用できる場合、空白にします。
AuthenticationScopeRealm	認証クレデンシャルを適用するレルム。クレデンシャルを任意のレルムに適用できる場合、空白にできます。
Username	ターゲットシステムへの認証のユーザ名。
Password	ターゲットシステムへの認証のパスワード。
Host	一部の認証スキーム (NTLM など) で必要になることがあるホスト クレデンシャル。
Domain	一部の認証スキームで必要になることがあるドメイン クレデンシャル。 NTLM ではレルムの概念が使用されません。認証ドメインは、「realm」属性の値として指定する必要があります。クレデンシャルを任意のドメインに適用できる場合、空白にします。
SaveRefField	ProcessResponse が true の場合に使用されるブール値。外部システムがこのタスクの固有識別子 (または TopicID) として使用するフィールドが、応答に含まれていることを示します。詳細については、下記の説明を参照してください。
RefFieldXPath	参照フィールド (TopicID) を識別する、応答の XPath 式。

表 2-15 HTTP/WS アダプタの発信プロパティ (続き)

RefFieldPattern	参照フィールドに適用される正規表現。
CancelIdentifierXPath	存在すると、進行中タスクをキャンセルする必要があることを指定する XPath 式。

認証スキーム

発信 http/ws アダプタの一部のプロパティは、特定の認証スキームだけに必要です。ほとんどの場合、認証がカスタマイズされた Web サーバだけが対象です。次の表に、発信 http/ws アダプタによってサポートされる認証スキームについてまとめます。

認証タイプ	説明
Anonymous	ユーザ クレデンシャルの提供を要求する必要はありません。通常は、Web サーバへのアクセスはサービス アカウントで行います。
Basic	ユーザ名とパスワードが必要です。パスワードはクリア テキスト形式で送信されます。
Digest	ユーザ名とパスワードが必要ですが、パスワードは MD5 ハッシュで送信されます。
NTLM	統合 Windows 認証。

http/ws 要求への応答

要求が Web サイトにポストされるか、またはメッセージが Web サービスに送信される場合、通常、ターゲット サイトはメッセージ発信者に応答を送信します。応答に Service Link にとって有用な情報が含まれている可能性が低い場合、*Process Response* プロパティを `false` に設定し、Service Link にこのようなメッセージを無視するように指定できます。ただし、このような応答に、外部システムのチケット番号や、Request Center で生成されたタスクに割り当てられたケース番号などの、追加情報が含まれていることがあります。この場合、*Process Response* と *Save Ref Field* の両方のプロパティを `true` に設定し、Service Link の Reference フィールドの `xpath` を指定して、Web サービスの応答から参照をキャプチャできます。さらに、変換を応答に適用し、サービス フォームを外部システムからの情報で更新するアクションを起動できます。

参照フィールドと TopicID

通常、外部システムには独自の手段があり、インシデント、要求、またはその他のオブジェクトが、サードパーティ システムで開かれているか、製品のユーザ インターフェイスで維持されているかを識別します。指定した参照フィールド (TopicID) によって、Request Center で外部システムの固有識別子と Request Center の Channel-Id の間の相互参照が維持されます。Web サービス要求に対する最初の応答で TopicID が識別され、保存されると、Web サービス リスナー アダプタを介して受信する外部システムからのその後のメッセージで TopicID を使用して Request Center 外部タスクを識別できます。

着信プロパティ

- HTTP/WS 着信アダプタはリスナー アダプタで、呼び出しに基づくポーリングはサポートされません。
- 指定の URL に対して HTTP/WS 着信エージェントを 1 つだけ設定する必要があります。http プロトコルまたは https プロトコルが使用されます。

着信 http アダプタにはプロパティが指定されません。すべての http post を Integration Server の URL に転送する必要があります。

```
<ServerName>:<Port>/IntegrationServer/ishttplistener?channel-id=<channel-id>
```

ここで、

- <ServerName> は Service Portal アプリケーション サーバです。
- <Port> は Service Portal が待ち受けるポートです。
- <channel-id> は、着信メッセージの影響を受けるタスクを一意に識別するチャンネル ID です。channelID が実行中のタスクに適用されない場合、サードパーティ システムにエラー 503 (アプリケーション エラー) が返されます。

Web サービスの呼び出し

Web サービスはそれほど単純ではない「HTTP による XML」です。発信アダプタの場合、XML メッセージが http (または https) で Web サービスに送信されます。メッセージ (変換のアプリケーションによって作成された発信メッセージ) は、SOAP エンベロープで囲む必要があります。Web サービスへのサンプル XML メッセージは次のようになります。

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">
<soap:Header
  soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <AuthenticationInfo>
    <userName>ns28sbd</userName>
    <password>09rbc19</password>
  </AuthenticationInfo>
</soap:Header>
<soap:Body
  soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <OpCreate>
    <Assigned_To_Group>CSCC</Assigned_To_Group>
    <Case_Type>Problem</Case_Type>
    <Category>Computer/Printer/Server</Category>

<!-- additional tags as required -->

    <txt_internalticketid/>
    <txt_requestid >40</txt_requestid >
    <Type>New Hardware Request</Type>
  </OpCreate>
</soap:Body>
</soap:Envelope>
```

JMS アダプタ

JMS 着信アダプタはリスナー アダプタであり、呼び出しに基づくポーリングはサポートされません。

JMS アダプタでは、キューからのメッセージの読み取りや書き込み、特定のトピックの公開や登録が可能です。指定のキューに対して JMS 着信エージェントを 1 つだけ設定する必要があります。同じエージェントを使用して複数のトピックを登録することはできません。トピックは、「topic.sample.exported」のように、完全に指定する必要があります。

着信アダプタのプロパティ

JMS 着信アダプタのプロパティ シートでは、次に示すプロパティ名に「JMSInboundAdapter」というプレフィックスが付けられます。

表 2-16 JMS アダプタの着信プロパティ

名前	説明
JndiProviderUrl	JMS で管理される着信エージェント用オブジェクト検索のための JNDI プロバイダーの URL。デフォルトは <code>jnp://localhost:4099</code> です。
JndiFactory	着信エージェント用の JNDI ネーミング ファクトリ。デフォルトは <code>org.jnp.interfaces.NamingContextFactory</code> です。
JmsTopicFactory	着信エージェント用の JMS トピック接続を取得するための Topic 接続ファクトリ。使用されません。
JmsQueueFactory	着信エージェント用の JMS キュー接続を取得するためのキュー接続ファクトリ。デフォルトは <code>ConnectionFactory</code> です。
MessageMode	JMS の宛先がキューであるか、トピックであるか。有効な値は <code>Queue</code> または <code>Topic</code> 。デフォルトは <code>Queue</code> です。
JmsQueue	着信エージェント用のメッセージ モードがキューの場合の名前付き JMS キュー。
JmsTopic	着信エージェント用のメッセージ モードがトピックの場合の名前付き JMS トピック。
MessageType	着信エージェントのメッセージのタイプ。有効な値は <code>Text</code> です。
Publisher.isAdapter	パブリッシャがアダプタの場合。デフォルトは <code>True</code> です。
Listener.UseCallback	コールバックを使用するかどうか。デフォルトは <code>True</code> です。
UserName	着信エージェント用の JNDI セキュリティ クレデンシャルのユーザ名。
Password	着信エージェント用の JNDI セキュリティ クレデンシャルのパスワード。

発信アダプタのプロパティ

JMS 発信アダプタのプロパティ シートでは、次に示すプロパティ名に「JMSOutboundAdapter」というプレフィックスが付けられます。

表 2-17 JMS アダプタの発信プロパティ

名前	説明
JndiProviderUrl	JMS で管理される発信エージェント用オブジェクト検索のための JNDI プロバイダーの URL。デフォルトは <code>jnp://localhost:4099</code> です。
JndiFactory	発信エージェント用の JNDI ネーミング ファクトリ。デフォルトは <code>org.jnp.interfaces.NamingContextFactory</code> です。
JmsTopicFactory	発信エージェント用の JMS トピック接続を取得するための Topic 接続ファクトリ。未使用。
JmsQueueFactory	発信エージェント用の JMS キュー接続を取得するためのキュー接続ファクトリ。デフォルトは <code>ConnectionFactory</code> です。
MessageMode	JMS の宛先がキューであるか、トピックであるか。有効な値は <code>Queue</code> または <code>Topic</code> です。

表 2-17 JMS アダプタの発信プロパティ (続き)

JmsQueue	発信エージェント用のメッセージモードがキューの場合の名前付き JMS キュー。
JmsTopic	発信エージェント用のメッセージモードがトピックの場合の名前付き JMS トピック。
MessageType	発信エージェントのメッセージのタイプ。有効な値は Text です。
Publisher.isAdapter	パブリッシャがアダプタの場合。デフォルトは True です。
UserName	発信エージェント用の JNDI セキュリティ クレデンシャルのユーザ名。
Password	発信エージェント用の JNDI セキュリティ クレデンシャルのパスワード。

MQ アダプタ

MQ 着信アダプタは、IBM WebSphere Message Queue (MQ) システムを使用するポーラー アダプタです。このアダプタでは、IBM MQ シリーズのバージョン 5.x 以上がサポートされます。統合には IBM MQ シリーズの Java API が使用されます。IBM MQ ソフトウェアは Service Portal には含まれおらず、ライセンスを IBM から入手する必要があります。

着信プロパティ

MQ 着信アダプタのプロパティシートでは、次に示すプロパティ名に「MQInboundAdapter」というプレフィックスが付けられます。

表 2-18 IBM MQ 着信アダプタのプロパティ

名前	説明
ManagerName	IBM MQ Manager の名前
HostName	IBM MQ Server のホスト名
Port	着信用の IBM MQ Server のポート
UserName	認証用のユーザ名
Password	認証用のパスワード
ChannelName	着信メッセージ用の IBM MQ チャンネル名
QueueName	着信メッセージ用のキュー名
MsgFormat	着信メッセージのメッセージ形式。デフォルトは Text です

発信プロパティ

MQ 発信アダプタのプロパティシートでは、次に示すプロパティ名に「MQOutboundAdapter」というプレフィックスが付けられます。

表 2-19 IBM MQ アダプタのプロパティ

名前	説明
ManagerName	発信メッセージ用の IBM MQ Manager の名前
HostName	IBM MQ Server のホスト名
Port	IBM MQ Server のポート

表 2-19 IBM MQ アダプタのプロパティ (続き)

UserName	認証用のユーザ名
Password	認証用のパスワード
ChannelName	発信メッセージ用の IBM MQ チャンネル名
QueueName	発信メッセージ用のキュー名
MsgFormat	発信用のメッセージ形式。デフォルト値は Text です

サービス項目リスナー アダプタ

Web サービス リスナー アダプタ ([「Web サービス リスナー アダプタ」 \(P.2-57\)](#) を参照) と同様、サービス項目リスナー アダプタは、外部システムで外部タスクへの更新の送信に使用される Web サービス (SOAP) エンド ポイントを提供します。タスクの更新に加えて、このアダプタでは Lifecycle Center サービス項目を着信 SOAP メッセージの一部として作成、更新、および削除できます。

外部システムによって送信される SOAP メッセージは、「processMessage」操作を呼び出します。SOAP 本文内のメッセージ内容は Service Link が認識できるメッセージに変換されてから、操作のタイプに基づいて分離され、それぞれ Business Engine と Service Item Import processor に転送されます。そのため、最大 2 つのメッセージが着信 SOAP メッセージの [View Transactions] ページに表示されます。1 つはタスク更新操作 (take-action、add-comments、send-parameters) 用で、1 つはサービス項目操作 (create、update、delete) です。後者のメッセージタイプは「Service Item」です。

着信プロパティ

サービス項目リスナー着信アダプタのプロパティシートでは、次に示すプロパティ名に「ServiceItemListenerInboundAdapter」というプレフィックスが付けられます。

表 2-20 サービス項目リスナー着信アダプタのプロパティ

プロパティ	説明
WsdL	<p>現在のインストール済み環境用の Request Center 着信サービス項目を記述する wsdl の URL は次の形式です。</p> <pre><Protocol>://<ServerName>:<Port>/IntegrationServer/webservices/wsdl/ServiceItemTaskService.wsdl</pre> <p>それぞれの説明は次のとおりです。</p> <p><Protocol> は http または https です。</p> <p><ServerName> は Service Link がインストールされているサーバです。</p> <p><Port> は Service Link 用に指定された通信ポートです。</p> <p>次に例を示します。</p> <pre>http://ccp-prod.cisco.com:8089/IntegrationServer/webservices/wsdl/ServiceItemTaskService.wsdl</pre> <p>このプロパティは読み取り専用です。これは、設計者が WSDL を参照できるようにするために用意されています。サービスを理解したり、Web サービスクライアントを作成するために役立ちます。</p>
RoutingURL	<p>SOAP メッセージの送信先となる WSDL は次の形式です。</p> <pre><Protocol>://<ServerName>:<Port>/IntegrationServer/services/TaskService</pre> <p>このプロパティは読み取り専用です。外部システムインテグレータが、SOAP メッセージをこの URL に登録するクライアントを作成できるように、用意されています。</p>

発信プロパティ

サービス項目リスナーアダプタは単方向（着信専用）です。したがって、発信プロパティはありません。

VMware アダプタ

VMware アダプタを使用するには、Lifecycle Center ライセンスが必要です。VMware アダプタは、VMware API を使用して vSphere 4.1 vCenter サーバと統合されます。

VMware アダプタを使用してグリッドディクショナリを処理することは、現在サポートされていません。

発信プロパティ

VMware 発信アダプタのプロパティシートでは、次に示すプロパティ名に「VMwareOutboundAdapter」というプレフィックスが付けられます。

表 2-21 VMware 発信アダプタのプロパティ

名前	説明
VMServerURL	アダプタが通信する必要がある vCenter サーバの完全な URL。例： https://<ServerName>/sdk。詳細については、以下を参照してください。
UserName	<network domain>\<network userid> 形式のログイン用ユーザ名。
Password	ユーザのパスワード
TimeOut	秒単位のタイムアウト値。
IgnoreVMServerCertificate	無効なサーバ証明書を見捨てる必要があるかどうかを示すパラメータ。 false に設定したままにしてください。
AutoInstallCertificate	true に設定したままにしてください。
ServerCertificate	SSL 接続で、無効な証明書または自己署名証明書を持つサーバを受け入れる必要がある場合に、X.509 サーバ証明書が含まれた文字列。 AutoInstallCertificate オプションが true に設定されている場合は、空白のままにできます

VMware サーバの URL

https はほとんどの vCenter のインストール済み環境で有効になっています。VMware 管理者に問い合わせ、適切なプロトコルを指定したことを確認してください。

vCenter サーバごとに個別のエージェントが必要です。さらに、同じ vCenter サーバ上のさまざまなタイプのアクティビティやさまざまなタイプのデータセンターを管理するために、複数のログインが存在している場合、ログインごとに 1 つのエージェントを定義する必要があります。

VMware アダプタとエージェントのアーキテクチャ

VMware アダプタおよびこのアダプタを使用するエージェントのアーキテクチャは、他の Service Link アダプタのアーキテクチャとは異なっています。VMware アダプタは、VMware API と直接通信を行います。API でサポートされる操作に必要なデータは、VMware 統合が含まれるサービスで使用されるディクショナリを介して提供されます。

VMware アダプタを使用するエージェントを設定する場合は、次のようにします。

- 発信メッセージの内容は「small」（デフォルト）に設定する必要があります。
- ユーザが指定した変換は使用されません。nsXML メッセージは常に「外部」メッセージと同じになります。メッセージの内容は、VMware API への呼び出しを生成するために使用されます。
- 統合の設計者は、エージェントに対して、VMware 操作に必要なフィールドが含まれるディクショナリの名前を示す 1 つの発信要求パラメータを指定する必要があります。発信パラメータ名は「Dictionary_Lookup_Name」にして、その値を適切なディクショナリ名に設定する必要があります。
- VMware アダプタは発信専用です。VMware 操作の実行結果とエラー メッセージ（該当する場合は、発信メッセージへの応答で返されます。

VMware アダプタの設定の詳細については、『Cisco Service Portal Designer Guide』を参照してください。

Web サービス リスナー アダプタ

Web サービス リスナー アダプタは、外部システムで外部タスクへの更新の送信に使用される Web サービス (SOAP) のエンド ポイントを提供します。外部システムによって送信される SOAP メッセージは、「processMessage」操作を呼び出します。SOAP 本文内のメッセージ内容は、Service Link が理解できるメッセージに変換されてから、処理のため HTTP/WS 着信アダプタに転送されます。

Web サービス リスナー アダプタは、基盤となる Web サービス リスナーを使用します。

着信プロパティ

Web サービス リスナー着信アダプタのプロパティ シートでは、次に示すプロパティ名に「WSListenerInboundAdapter」というプレフィックスが付けられます。

表 2-22 Web サービス リスナー着信アダプタのプロパティ

プロパティ	説明
WsdURL	<p>現在のインストール済み環境用の Request Center 着信 Web サービスを記述する wsdl の URL は次の形式です。</p> <pre><Protocol>://<ServerName>:<Port>/IntegrationServer/webservices/wsdl/TaskService.wsdl</pre> <p>それぞれの説明は次のとおりです。</p> <p><Protocol> は http または https です。</p> <p><ServerName> は Service Link がインストールされているサーバです。</p> <p><Port> は Service Link 用に指定された通信ポートです。</p> <p>次に例を示します。</p> <p>http://ccp-prod.cisco.com:8089/IntegrationServer/webservices/wsdl/TaskService.wsdl</p> <p>このプロパティは読み取り専用です。これは、設計者が WSDL を参照できるようにするために用意されています。サービスを理解したり、Web サービス クライアントを作成するために役立ちます。</p>
WsdRoutingURL	<p>SOAP メッセージの送信先となる WSDL は次の形式です。</p> <pre><Protocol>://<ServerName>:<Port>/IntegrationServer/services/TaskService</pre> <p>このプロパティは読み取り専用です。外部システム インテグレータが、SOAP メッセージをこの URL に登録するクライアントを作成できるように、用意されています。</p>

発信プロパティ

Web サービス リスナー アダプタは単方向 (着信専用) です。したがって、発信プロパティはありません。

Integration ウィザード

概要

Integration ウィザードでは、Request Center と Web サービスとの統合の作成に関する手順の多くが自動化されます。Integration ウィザードは、Web サービスの統合によって呼び出される wsdl と操作を取得することによって動作します。統合の定義に基づき、Integration ウィザードは統合をサポートするために必要なすべてのコンポーネントを作成します。

- 外部タスクで統合の実行に使用できる Service Link エージェントが作成され、現在のサービスの配信計画で参照されます。
- Web サービスに必要な SOAP メッセージに nsXML を変換する変換が作成され、エージェントの発信アダプタで参照されます。
- 最初の Web サービス要求と応答の両方で必要な、すべてのデータのエージェント パラメータがエージェント定義に追加されます。
- エージェント パラメータ値を格納するフィールドが含まれたディクショナリが作成され、ディクショナリ フィールドが、対応するエージェント パラメータにマッピングされます。
- ディクショナリが含まれたアクティブ フォーム コンポーネントが作成され、現在のサービスに含まれます。

Integration ウィザードは、認証方式の定義と統合の動作でいくつかのデフォルト オプションを使用します。これらの設定が適切ではない場合、または統合の作成後に変更する必要がある場合、Service Link の [Manage Integration] ページで使用できる高度なコンフィギュレーション オプションを使用してエージェント定義を編集できます。

Integration ウィザードは、Service Link エージェントと変換の作成が可能なロールを付与された、サービス設計者だけが使用できます。

Integration ウィザードからアクセスする wsdl は、Web Services Operability (WS-I) のベストプラクティスに従う必要があります。

Integration ウィザードの使用方法

Integration ウィザードを使用するには、次の手順に従います。

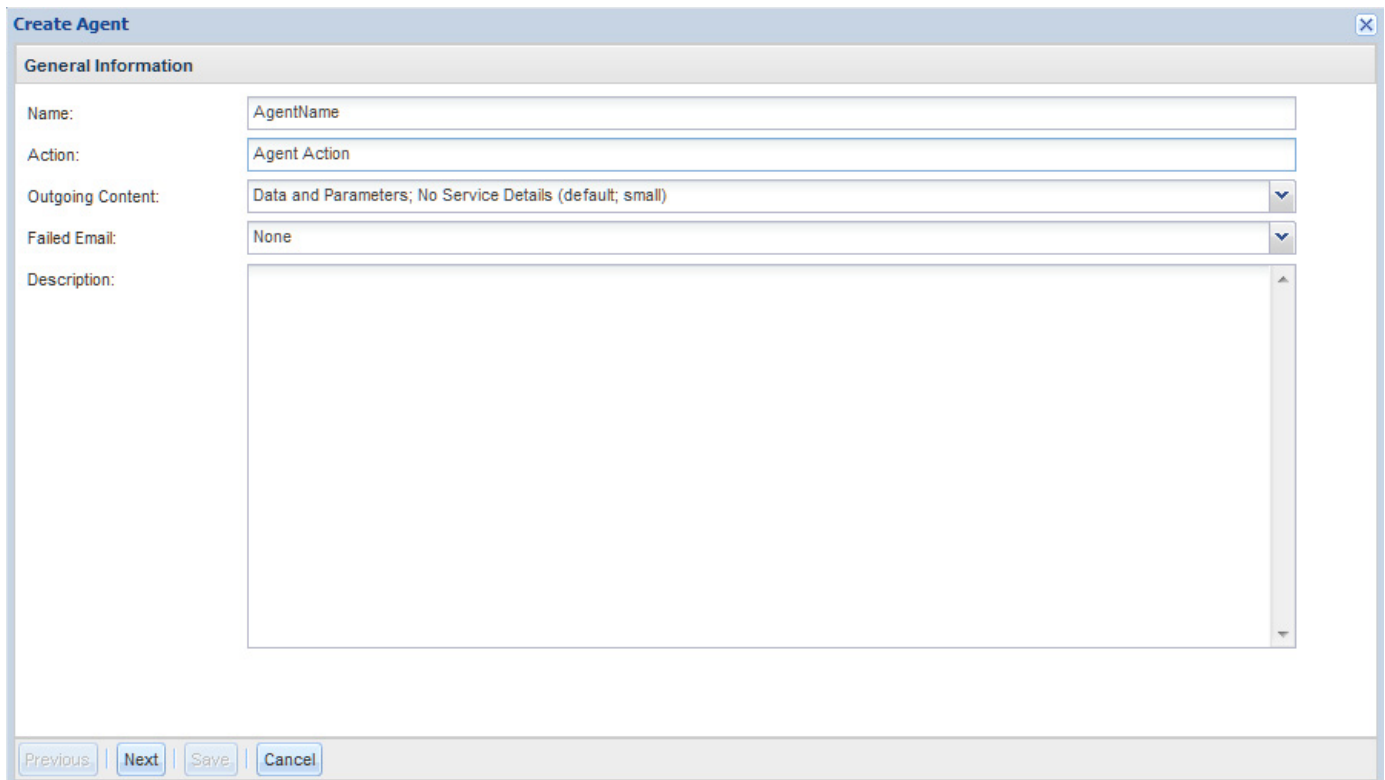
- ステップ 1** Service Designer でサービスを編集します。
- ステップ 2** サービスの [Plan] タブの [General] サブタブに移動します。任意で、タスクに関する他のデータを入力します。
- ステップ 3** [Create Agent] をクリックします。

The screenshot shows a configuration form for creating an agent. It includes a 'Save' button at the top left. The form fields are: 'Workflow Type' (dropdown menu set to 'Agent Action'), 'Task name' (text input field containing 'Task1 - Tasks for Queues'), 'Subtasks execute' (dropdown menu set to 'one after the other (sequentially)'), and 'Priority' (dropdown menu set to 'Normal'). A 'Create Agent' button is located to the right of the 'Workflow Type' dropdown.

Integration ウィザードの最初のページが表示されます。ウィザードは、エージェントの設定方法に応じて、最大 8 つのページで構成されます。各ページが完了したら、[Next] をクリックして次のページに進むか、または [Previous] をクリックして前のページに戻ります。完了したら、[Save] をクリックしてエージェント（および他の設計コンポーネント）の定義を保存するか、または [Cancel] をクリックして、作業を保存せずに終了します。

General Information

最初に、エージェントの全般情報を指定します。



The screenshot shows a 'Create Agent' dialog box with a 'General Information' tab. The form contains the following fields and values:

- Name: AgentName
- Action: Agent Action
- Outgoing Content: Data and Parameters; No Service Details (default; small)
- Failed Email: None
- Description: (empty text area)

At the bottom of the dialog, there are four buttons: Previous, Next, Save, and Cancel.

作成するディクショナリとアクティブ フォーム コンポーネントの名前は、エージェント名と同じになります。ディクショナリの命名基準はエージェントよりも厳しいため、エージェント名に使用できるのは文字、数字、アンダースコアだけで、先頭を数字にはできません。

この画面の他のすべての設定は、Service Link の [Agents] ページで使用できる設定と対応しています。

[Next] をクリックして、ウィザードの次のページに進みます。

発信プロパティ

The screenshot shows the 'Create Agent' wizard window with the following configuration:

- WSDL URL:** http://hostname/RequestCenter/webservices/wsd/RequisitionService.wsdl
- Operation:** RequisitionServiceHttpBinding.addComment
- Routing URL:** http://hostname:80/RequestCenter/services/RequisitionService
- Advanced Properties:**
 - User Name:** [Redacted]
 - Password:** [Redacted]
 - Accept Untrusted URL:** true
 - Content Type:** text/xml

Buttons at the bottom: Previous, Next, Save, Cancel.

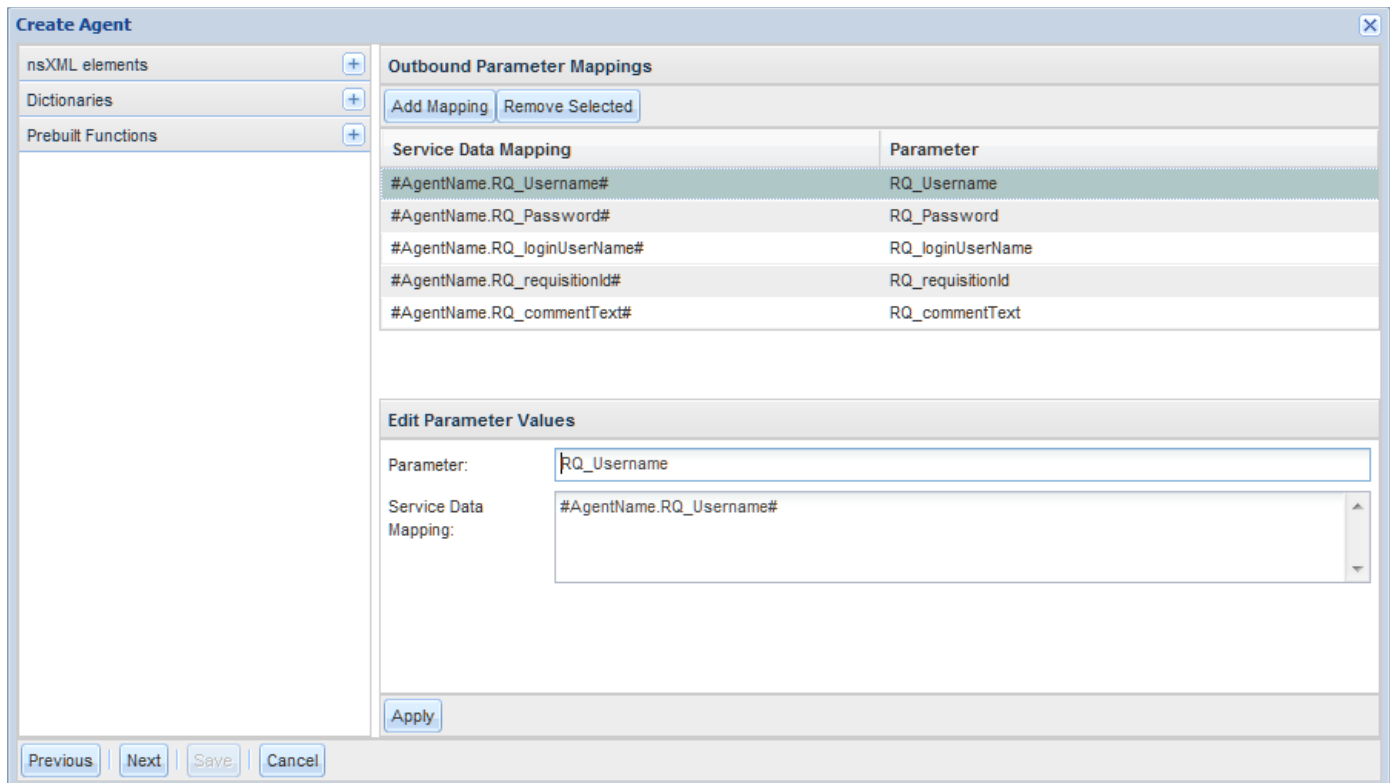
統合によって実行する操作が含まれる WSDL の場所を入力します。通常は、WSDL が存在している URL です。

Integration ウィザードは、ウィザードを読み取り、サポートされる操作のリストを表示します。目的の操作を選択してください。操作に指定される属性によって、ウィザードの後続のページでのエージェント パラメータの定義が決まります。

wsdl にルーティング url が含まれる場合、これも表示されます。

必要に応じて、[Advanced Properties] ドロップダウン ボタンをクリックして、統合の追加設定を表示します。これらはここで入力することも、Service Link から後で指定することもできます。ウィザードで指定できるのは基本的な認証だけです。

発信要求パラメータ マッピング



ウィザードは wsdl を解析し、表示されるサンプルで、Web サービス発信要求で使用する必要のある 2 つの属性が含まれているかどうかを判断します。したがって、名前が wsdl の属性名と一致する 2 つのエージェントパラメータを作成します。

エージェントパラメータがディクショナリフィールドにマッピングされます。フィールド名が wsdl の属性名と一致し、ディクショナリ名がエージェント名と一致します。このディクショナリは、エージェントを保存すると自動的に作成されます。

必要に応じて、以前に Service Designer で定義したディクショナリおよびフィールドを参照するように [Service Data Mapping] を変更します。これによって、効率的にエージェントパラメータマッピングが変更されます。ただし、ウィザードによって作成されたディクショナリには元のフィールドが残ります。ディクショナリの定義を編集すると削除できます。

関連情報として、サービスでのディクショナリの作成および使用について説明します。Request Center ディクショナリの主な目的は、サービスフォームでユーザに表示するデータを構築することです。したがって、サービスの設計者は通常、ユーザインターフェイスを考慮してディクショナリを設計し、フィールドのグループ化や並べ替えを行って、カスタマーとサービスチームメンバーの両方のエクスペリエンスを最適化します。

原則としては、発信メッセージに、多数のディクショナリのフィールドで入力（デフォルトに設定、または計算）されたデータを含める必要があります。ただし、これによってエージェントパラメータマッピングの維持がより複雑になり、エラーが発生しやすくなります。統合の設計者はサービスフォームおよびディクショナリの設計についての詳しい知識を持っている必要があります。したがって、統合データを含めることが目的である場合、ディクショナリだけを作成することを推奨します。ディクショナリの一部のフィールドは、サービスフォームに表示されるフィールドと重複する場合があります。この場合、サービスの設計者は条件付きルールを指定して、表示されたディクショナリから

フィールドの値を統合ディクショナリにコピーする必要があります。さらに、統合ディクショナリはサービス フォームに表示されません（通常は、アクティブなフォーム ルールによって非表示になります）。ただし、デバックしやすいよう、開発中は常に表示しておきます。

発信応答パラメータ マッピング

デフォルトでは、Integration ウィザードはターゲット システムから受信した応答が処理されることを前提としています。応答で送信される属性には、対応するエージェント属性があり、ディクショナリ フィールドにマッピングされます。

Parameter	Dictionary Field	Mapping	Manda...
RS_commentLevel	AgentName.RS_comme...		<input type="checkbox"/>
RS_commentLevelObj...	AgentName.RS_comme...		<input type="checkbox"/>
RS_commentSourceType	AgentName.RS_comme...		<input type="checkbox"/>
RS_commentText	AgentName.RS_comme...		<input type="checkbox"/>
RS_component	AgentName.RS_compo...		<input type="checkbox"/>
RS_dateEntered	AgentName.RS_dateEn...		<input type="checkbox"/>
RS_person	AgentName.RS_person		<input type="checkbox"/>

Parameter:

Dictionary Field:

Mapping:

Apply

Previous | Next | Save | Cancel

エージェントとフィールドの対応関係に加えて、マッピングには単純な XSLT 操作が含まれる場合があります。ページの左側にある [Prebuilt Functions] ドロップダウン矢印から使用できます。

発信パラメータの場合は、着信パラメータも代替ディクショナリ フィールドにマッピングできます。ページの左側にある [Dictionaries] ドロップダウン矢印からすべてのディクショナリを参照できます。

統合の概要

ウィザードの最後のページでは、定義された統合の概要が表示されます。前のいずれかのページに戻って修正を行うか、[Save] をクリックし、作成したエージェントおよび他のすべての統合コンポーネントを保存します。デフォルトでは、統合が保存されると、エージェントが起動します。[Start agent upon saving] チェックボックスをオフにすると、この動作を変更できます。

Name	Value
Name	AgentName
Action	Agent Action
Active Form Component Name	AgentName
Dictionary Name	AgentName
Outgoing Content	Data and Parameters; No Service Details (default; small)
Failed Email	None
Description	
Outbound Transformation	AgentName Transformation

Start agent upon saving

Outbound Properties

Outbound Parameter Mapping

Outbound Response Parameter Mapping

Previous | Next | Save | Cancel

Service Link および Service Designer の画面からすべてのコンポーネントを編集できます。これらのコンポーネントを表 2-23 に示します。

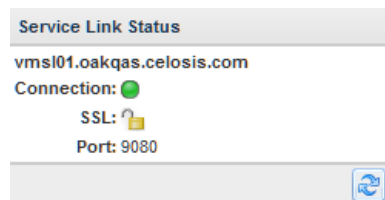
表 2-23 統合コンポーネント

統合コンポーネント	説明
Agent	http/ws 発信アダプタを使用するエージェントと、エージェント パラメータおよび関連する変換。
Transformation	発信 nsXML を、選択された WSDL 操作のために目的の形式に変換するために必要な変換。
Dictionary	すべての発信パラメータおよび着信パラメータに対応するフィールドが含まれるディクショナリ。ディクショナリが Integration グループに作成されます。必要に応じて移動できます。
Active Form Component	作成されたディクショナリが含まれるアクティブ フォーム コンポーネント。フォーム コンポーネントが統合グループに作成されます。必要に応じて移動できます。

Service Link のトラブルシューティングと管理

Service Link のステータスの確認

Service Link の動作ステータスは、最初に [Service Link Status] 表示を確認します。Service Link のステータスは、常に Service Link のホーム ページの [Common Tasks] 領域の下に表示されます。



この機能は、Service Link が、割り当てられたポートを介して Request Center サービスとの通信を行っていることを確認するために役立ちます。

[Service Link Status] は Service Link の接続ステータスが動作していることを示し、使用されるポートおよびプロトコルを示します。

エージェントの起動と停止

エージェントは、[Control Agents] ページを使用して個別に起動および停止できます。

Service Link サービスを停止して再起動すると、サービスが停止されたときに実行中だったすべてのエージェントが自動的に再起動されます。

ロギング

すべてのアダプタはそのアクティビティをサーバのログ ファイルに記録します。

さらに、Service Link¥log ディレクトリに各標準アダプタ専用のログ ファイルがあります。ログに書き込まれる詳細度は設定可能です。詳細度の設定はアプリケーションとサーバの組み合わせごとに行えます。

サーバとアダプタに固有のログ ファイルの管理に関する詳細については、『Cisco Service Portal Configuration Guide』を参照してください。

JBoss ロギング

JBoss のインストール済み環境では、サイトにあるファイル (...¥ServiceLink¥conf¥jboss-log4j.xml) で指定されているように、アダプタごとに個別のログ ファイルが自動的に維持されます。ロギング特性の設定は、log4j 設定ファイルをテキスト エディタで変更することによって調整できます。導入環境に追加アダプタがある場合、FILE_ADAPTER アペンダーおよびカテゴリをコピーし、アペンダー名、参照、ログ ファイル名、アダプタ ファイルのパッケージを変更することによって、このような新しいアダプタのログ ファイルを分割できます。

WebLogic ロギング

WebLogic ではアダプタごとにログ ファイルを分割できません。IntegrationServer コンポーネントは、デフォルトでは WebLogic ロガーを使用して設定されます。ログの分離が必要な場合は、ISEE.war/WEB-INF/classes の下の `newscalelog.properties` を編集します。ロギング メカニズムとして一般的なロギングを使用する行をアンコメントします。IntegrationServer の実行に使用されるユーザに完全な書き込みアクセス権がある場合は、`logger.directory` の有効な値をアンコメントし、有効な既存のシステム内のディレクトリに設定することも非常に重要です。ファイル `newscalelog.properties` に追加の説明があります。さらに、他のアダプタの追加設定が必要な場合はファイル `log4j.xml` を編集し、FILE_ADAPTER アペンダーとカテゴリをベースとして使用し、アペンダー名と参照（追加機能のパッケージおよびファイル名）を調整します。

WebSphere ロギング

デフォルトでは、Service Link の WebSphere ロギングは、WebSphere アプリケーション サーバに含まれている `log4j` に基づいて行われます。WebSphere での `log4j` の実装は非常に強力で、管理コンソールやその他のツールから設定可能です。ただし、ログ ファイルの分離は容易ではありません。WebSphere でアダプタごとにログ ファイルを分離する場合、ISEE.war/WEB-INF/classes の下にあるファイル `newscalelog.properties` を編集する必要があります。これを行い、ログ ファイルが保存される有効なディレクトリを入力することが非常に重要です。

メッセージのページ

Service Link メッセージをページするユーティリティは、データベース スクリプトとして用意されています。ページスクリプトを実行するには、DML ユーザとして、データベースに適した SQL ツールで RequestCenter データベースにアクセスし、プロシージャ「`sp_CleanUpSIMessageContent`」を実行します。このスクリプトの入力パラメータとして、メッセージを保持する日数を指定する必要があります。実際には、このユーティリティはメッセージデータをページしませんが、90 日以上経過した、成功または失敗したすべてのメッセージに対してメッセージの内容を「Message has been purged」に設定します。

アプリケーション サーバのコンフィギュレーション ファイル

トラブルシューティング時に次のファイルを解析できます。

- **bejms.properties** ファイル：このファイルには発信キューに関する情報が含まれます。Business Engine がメッセージを、このファイルで指定されたキューに入れます。このファイルに入力された情報は、**integrationserver.properties** ファイルの発信キューと一致している必要があります。このファイルは **ISEE.war** ファイルです。このファイルは **conf** ディレクトリに存在している必要があります。
- **bejms.properties** ファイルで次のプロパティを検索します（これはサンプル JBoss プロパティファイルです。値はインスタンスに固有なものになります）。

```
ISEEOutbound.JndiProviderUrl=jnp://localhost:4099
ISEEOutbound.JndiFactory=org.jnp.interfaces.NamingContextFactory
ISEEOutbound.JmsTopicFactory=ConnectionFactory
ISEEOutbound.JmsQueueFactory=ConnectionFactory
ISEEOutbound.JmsQueue=queue/ISEEOutboundQueue
ISEEOutbound.JmsTopic=topic/ISEEOutboundTopic
```

- **integrationserver.properties** ファイル：着信および発信の JMS キューに関する情報が含まれます。このファイルは `$deploy_home$VISEE.war$WEB-INF$classes` ディレクトリにあります。このフォルダで指定された JMS プロパティを確認してください。
- **newscaler.properties** ファイル：このファイルには `isee.base.url` のプロパティが含まれています。それが、Service Link サーバの url を指していることを確認します。

オンライン エラー ログ

サーバのログ ファイルおよびアダプタに固有のログ ファイルに加えて、Service Link によって検出されたエラーをオンラインで参照することもできます。[Messages] ページに表示される失敗したメッセージのメッセージ テキストは、そのメッセージの詳細なエラーへのハイパーリンクになっています。

エラー メッセージはサーバ ログに表示されるものとまったく同じで、技術的に高度な内容の場合があります。次に、サンプル エラー メッセージと説明を示します。

```
com.newscaler.is.core.RoutingException: Routing exception found: Reference Field not
retrieved from response
```

発信 Web サービス メッセージが送信されましたが、指定された参照フィールドが応答メッセージに含まれていなかったため、着信応答を処理できませんでした。

```
com.newscaler.is.core.TransformationException: javax.xml.transform.TransformerException:
javax.xml.transform.TransformerException: Tag is not allowed in this position in the
stylesheet!
```

変換で無効な XML メッセージが生成されました。

事前作成済みの関数

概要

事前作成済みの関数は、nsXML メッセージに含まれているエージェント パラメータの値を操作する機能を提供します。

事前作成済みの関数は、Visigoth Software Society によって開発されたオープン ソース ソフトウェアとして入手可能な FreeMarker テンプレート エンジン、バージョン 2.3.12 を使用して開発されています。シスコは以下で説明する関数だけを認定しており、エージェント パラメータの作成時にドロップダウン リストから使用できます。FreeMarker フレームワークでサポートされるその他の関数を使用できますが、十分にテストする必要があります。

関数の使用方法

基本的な関数の使用方法には、関数の式への適用、必要な場合の関数への引数リストの指定があります。一般的には、次のようになります。

```
#{Expression?function(argumentList)}
```

Service Link の場合、式は一般的に、Lightweight 名前空間構文によって指定されたディクショナリ フィールドまたは nsXML 要素になっています。次のように、引用符で囲む必要があります。

```
#{"#Customer_Information.Login_ID#"?upper_case}
```


次の構文を使用して、同じ式に 2 つ以上の関数をチェーンにして適用することができます。

```
$(Expression?function1(argumentList))${"$Parameter$"?function2}
```

たとえば、サービス データ マッピングによって、指定されたディクショナリ フィールドの先頭または末尾のスペースが削除されてから、その結果が小文字に変換されます。

Edit Parameter Values	
Parameter:	RQ_loginUserName
Service Data Mapping:	#{#Customer_Information.Login_ID#?trim}\${"\$Parameter\$"?lower_case}

次に示すように、複数の要素を組み合わせることで 1 つのマッピングにすることができます。要素は暗黙的に連結され、1 つの文字列を形成します。

Inbound Parameter Mappings			
Add Mapping Remove Selected			
Parameter	Dictionary Field	Mapping	Mandatory
Name	Temp.text1		<input type="checkbox"/>
Domain	Temp.text2		<input type="checkbox"/>
Email	Customer_Information.Email_Address	\$("#Name\$"?lower_case)@\$("#Domain\$...	<input type="checkbox"/>

Edit Parameter Values	
Parameter:	Email
Dictionary Field:	Customer_Information.Email_Address
Mapping:	\$("#Name\$"?lower_case)@\$("#Domain\$"?lower_case).com
Apply	

このシナリオでは、「一時」フィールドを使用して入力値を格納し、マッピング式でできるようにする別のコーディング テクニックも示します。

関数の概要

substring

substring 関数の構文は次のとおりです。

```
exp?substring(from, toExclusive). exp?substring(from) としても呼び出し可能
```

文字列のサブストリング。 *from* は 1 文字目のインデックスです。 0 以上、 *toExclusive* 以下の数値にする必要があります。 そうしないと、エラーとなってテンプレートの処理が中断されます。 *toExclusive* は、サブストリングの最後の文字の後にある文字の位置のインデックスです。つまり、最後の文字のインデックスよりも 1 大きい値になります。 0 以上、文字列の長さ以下にする必要があります。 そうしな

いと、エラーとなってテンプレートの処理が中断されます。*toExclusive* を省略した場合は、文字列の長さがデフォルト値になります。パラメータが整数ではない数値の場合、数値の整数部分だけが使用されます。

index_of

指定したサブストリングがその文字列で最初に出現する場所を示すインデックスを返します。たとえば、`"abcabc"?index_of("bc")` は 1 を返します (1 文字目のインデックスは 0 です)。また、検索を開始するインデックスを指定することもできます。`"abcabc"?index_of("bc", 2)` は 4 を返します。2 番目のパラメータの数値に制限はありません。負の値の場合、0 と同じ効果があり、この文字列の長さを超える場合、この文字列の長さと同じ効果があります。10 進値は切り捨てられて整数になります。

最初のパラメータがこの文字列のサブストリングとして出現しなかった場合は (2 番目のパラメータを使用した場合は、指定されたインデックスから開始)、-1 を返します。

last_index_of

指定したサブストリングがその文字列で最後 (右端) に出現する場所を示すインデックスを返します。サブストリングの最初 (左端) の文字のインデックスを返します。たとえば、`"abcabc"?last_index_of("ab")` は 3 を返します。また、検索を開始するインデックスを指定することもできます。次に例を示します。

```
"abcabc"?last_index_of("ab", 2)
```

この例では 0 を返します。最初のパラメータは、サブストリングの開始の最大インデックスを示します。2 番目のパラメータの数値に制限はありません。負の値の場合、0 と同じ効果があり、この文字列の長さを超える場合、この文字列の長さと同じ効果があります。10 進値は切り捨てられて整数になります。

最初のパラメータがこの文字列のサブストリングとして出現しなかった場合は (2 番目のパラメータを使用した場合は、指定されたインデックスの前)、-1 を返します。

length

文字列内の文字の数。

lower_case

文字列を小文字にしたもの。たとえば、「*GrEeN MoUsE*」は「*green mouse*」となります。

replace

元の文字列内で使用されている特定の文字列すべてを別の文字列に置換するために使用します。単語の境界は考慮されません。次に例を示します。

```
${"this is a car acarus"?replace("car", "bulldozer")}
```

次のように出力されます。

```
this is a bulldozer abulldozerus
```

置換は左から右の順に行われます。そのため、次のようになります。

```
${"aaaaa"?replace("aaa", "X")}
```

次のよう出力されます。

```
Xaa
```

最初のパラメータが空の文字列の場合は、空の文字列すべてが置換され、"`foo`"?`replace("", "/")` が "`|f|o|o|`" のように評価されます。

`replace` では、オプションの **flags parameter** を 3 番目のパラメータとして指定できます。

upper_case

文字列を大文字にしたもの。たとえば、「`GrEeN MoUsE`」は「`GREEN MOUSE`」となります。

