



Informatica® Cloud Application Integration
April 2023

RabbitMQ コネクタガイド

Informatica Cloud Application Integration RabbitMQ コネクタガイド
April 2023

© 著作権 Informatica LLC 2019, 2023

発行日: 2023-06-27

目次

序文	4
第 1 章 : RabbitMQ コネクタについて	5
RabbitMQ コネクタの概要.	5
RabbitMQ の概要.	5
RabbitMQ の概念.	6
第 2 章 : RabbitMQ 接続	9
RabbitMQ 接続の概要.	9
基本接続プロパティ.	10
RabbitMQ 接続のプロパティ.	10
RabbitMQ イベントソースプロパティ.	13
RabbitMQ イベントターゲットプロパティ.	15
RabbitMQ 接続のメタデータ.	16
第 3 章 : RabbitMQ プロセス	17
RabbitMQ プロセスの作成.	17
メッセージペイロード.	17
RabbitMQ プロセスのルールとガイドライン.	18
索引	19

序文

『*RabbitMQ* コネクタガイド』で、RabbitMQ 接続を設定および使用する方法を確認します。RabbitMQ キューに対してデータの読み書きを行うための、RabbitMQ 接続の設定方法と使用方法について説明しています。

第 1 章

RabbitMQ コネクタについて

この章では、以下の項目について説明します。

- [RabbitMQ コネクタの概要, 5 ページ](#)
- [RabbitMQ の概要, 5 ページ](#)
- [RabbitMQ の概念, 6 ページ](#)

RabbitMQ コネクタの概要

RabbitMQ コネクタを使用して、次のタスクを実行できます。

- RabbitMQ 接続、RabbitMQ プロデューサ、RabbitMQ コンシューマの設定。
- 特定のキューにあるメッセージのリスンおよび読み取り。
- 特定のキューへのメッセージの書き込み。

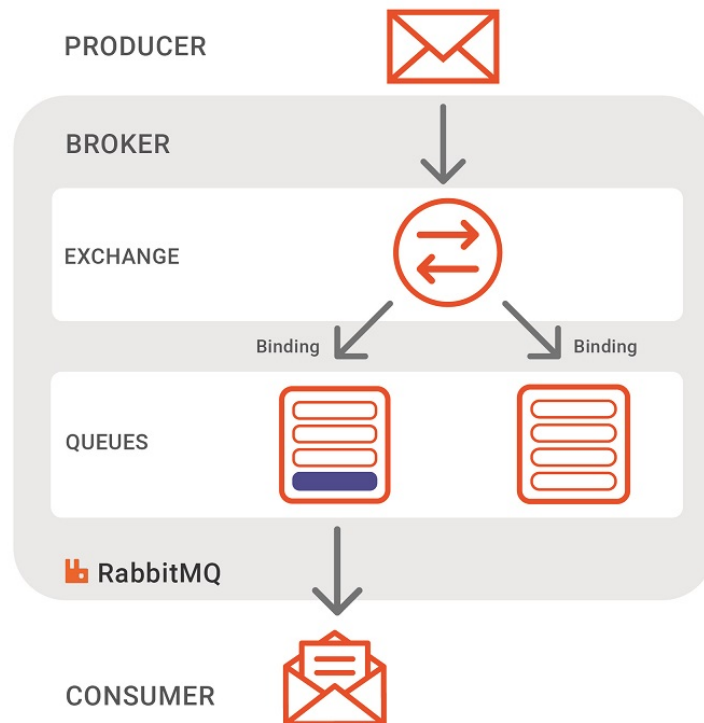
RabbitMQ の概要

RabbitMQ は、高速かつ拡張可能で堅牢な分散型のメッセージブローカシステムです。

RabbitMQ はメッセージの安全な転送を実現するために、Advanced Messaging Queuing Protocol (AMQP) を使用します。

プロデューサがメッセージをエクスチェンジにパブリッシュします。エクスチェンジがメッセージを受信したら、それらのメッセージを別のキューにルーティングします。メッセージのルーティングキーに基づいて、エクスチェンジがキューとのバインディングを作成し、メッセージをキューにルーティングします。メッセージは、コンシューマがキューに接続してメッセージをサブスクライブするまでキュー内に留まります。

次の図は、RabbitMQ のメッセージフローを示しています。



RabbitMQ の概念

RabbitMQ 接続を正しく使用するためには、RabbitMQ のメッセージングについて次のような概念を理解しておく必要があります。

- 仮想ホスト
- エクスチェンジ
- コンシューマの確認応答
- パブリッシャの確認応答
- デッドレタールーティング
- ハートビート
- 永続キュー
- ヘッダー

仮想ホスト

仮想ホストにより、同じ RabbitMQ インスタンスを使用しているアプリケーションを分離できます。それぞれのユーザーが仮想ホストごとに別々のアクセス権限を持ち、1 つの仮想ホスト内にのみに存在するキューとエクスチェンジを作成できます。

エクスチェンジ

プロデューサがメッセージを RabbitMQ ブローカにパブリッシュすると、メッセージがエクスチェンジに送信されます。エクスチェンジはメッセージを別のキューにルーティングする役割を担当します。エクスチェンジがプロデューサからのメッセージを受け付け、そのメッセージをヘッダー属性、バインディング、ルーティングキーに基づいてキューにルーティングします。

バインディングは、エクスチェンジとキューの間の接続です。ルーティングキーはメッセージ内の属性で、エクスチェンジがメッセージをキューにルーティングする際に役立ちます。

RabbitMQ では次のタイプのエクスチェンジを使用できます。

- **ダイレクトエクスチェンジ**

ダイレクトエクスチェンジは、メッセージルーティングキーに基づいてメッセージをキューに配信します。ダイレクトエクスチェンジでは、メッセージのルーティングキーと一致するバインディングキーを持つキューに、メッセージがルーティングされます。

- **ファンアウトエクスチェンジ**

ファンアウトエクスチェンジは、自分にバインドされているすべてのキューにメッセージをルーティングします。

- **トピックエクスチェンジ**

トピックエクスチェンジは、ルーティングキーと、バインディングで指定されたルーティングパターンとをワイルドカード照合します。

- **ヘッダーエクスチェンジ**

ヘッダーエクスチェンジは、メッセージのヘッダー属性をルーティングに使用し、ヘッダーの値に基づいてキューにメッセージをルーティングします。

コンシューマの確認応答

メッセージがコンシューマに配信されると、RabbitMQ はメッセージを削除対象としてマークします。RabbitMQ が確実にメッセージをコンシューマに配信し、コンシューマがメッセージを正常に処理できるようにするために、RabbitMQ は確認応答を使用します。コンシューマはメッセージが正常に受信され処理されたことを示す確認応答を RabbitMQ に送信します。その後、RabbitMQ がそのメッセージを削除します。

コンシューマ接続が失われ、コンシューマが RabbitMQ に確認応答を送信できなかった場合、RabbitMQ はメッセージが完全には処理されなかったと判断し、メッセージを再びキューに入れます。そのときにオンラインのコンシューマが他にあれば、RabbitMQ はメッセージを再配信します。

自動確認

自動確認モードでは、RabbitMQ はメッセージを送信した直後に、メッセージが正常に配信されたと見なします。自動メッセージ確認ではパフォーマンスとスループットが向上しますが、メッセージ配信の安全性は低下します。

コンシューマ接続が失われた場合、RabbitMQ が送信したメッセージは失われます。

ネガティブ応答確認

ネガティブ応答確認モードでは、コンシューマがすぐに配信を処理できなくても、他のコンシューマがそのメッセージを処理できる場合があります。RabbitMQ は、別のコンシューマがメッセージを受信して処理できるようにメッセージを再びキューに入れます。

パブリッシャの確認応答

プロデューサによってパブリッシュされたメッセージが確実に RabbitMQ サーバーに到達し、RabbitMQ サーバーがメッセージを正常に処理できるようにするために、RabbitMQ はコンシューマの確認応答で使用するメカニズムと同じメカニズムを使用して、パブリッシャに応答確認を送信します。

デッドレタールーティング

RabbitMQ は次のいずれかのイベントが発生したときに、デッドレタールーティングを使用して、エクスチェンジにメッセージを再パブリッシュします。

- コンシューマによってメッセージのネガティブ応答確認があり、RabbitMQ ブローカによるキューの入れ直しが行われなかった。
- メッセージの有効期限が切れた。
- キューの長さ制限を超えたため、メッセージがドロップした。

デッドレターになったメッセージは、同じ仮想ホストに指定されているデッドレターエクスチェンジにルーティングされます。デッドレターになったメッセージは、そのキューに対して指定されているルーティングキー、またはメッセージが最初にパブリッシュされたときに使用したルーティングキーに基づいてデッドレターエクスチェンジにルーティングされます。

ハートビート

RabbitMQ は中断された接続を検出するためにハートビートを使用し、終了するおそれのあるアイドル接続を守ります。ハートビートタイムアウトは、RabbitMQ のサーバーとクライアントが接続へのアクセスを不可と判断するまでの時間を決定します。

永続キュー

永続として定義された RabbitMQ のキューは、ディスクに永続化され、ブローカが再起動しても維持されます。永続キューは、ブローカが再起動しても、そのキューにルーティングされたメッセージが残ることを保証するものではありません。永続キュー内で、ブローカが再起動しても維持されるのは永続メッセージのみです。

ヘッダー

RabbitMQ メッセージはヘッダーと本文で構成されています。ヘッダーは、RabbitMQ キューでのメッセージ配信を制御するために RabbitMQ メッセージに設定されている属性です。ヘッダーはプロデューサがメッセージをパブリッシュしたときにプロデューサによって設定され、コンシューマがメッセージを受信した後にコンシューマによっても設定されます。

たとえば、次のようなヘッダーを使用できます。

- `rabbitmq.ROUTING_KEY`. キューにメッセージを配信するために使用されるルーティングキー。ヘッダーで指定されているルーティングキーは、RabbitMQ 接続で指定されている値をオーバーライドします。
- `rabbitmq.DELIVERY_MODE`. 値が 2 に設定されている場合、メッセージは永続キュー内で永続化され、ブローカの再起動後も維持されます。
- `rabbitmq.EXPIRATION`. メッセージがキュー内で有効期限切れになるまでの時間を指定します。
- `rabbitmq.EXCHANGE_OVERRIDE_NAME`. ターゲットエクスチェンジの名前を指定します。この名前は、RabbitMQ 接続で指定されたエクスチェンジ名をオーバーライドします。
- `rabbitmq.EXCHANGE_NAME`. コンシューマが受信したメッセージの送信元であるエクスチェンジの名前。

第 2 章

RabbitMQ 接続

この章では、以下の項目について説明します。

- [RabbitMQ 接続の概要, 9 ページ](#)
- [基本接続プロパティ, 10 ページ](#)
- [RabbitMQ 接続のプロパティ, 10 ページ](#)
- [RabbitMQ イベントソースプロパティ, 13 ページ](#)
- [RabbitMQ イベントターゲットプロパティ, 15 ページ](#)
- [RabbitMQ 接続のメタデータ, 16 ページ](#)

RabbitMQ 接続の概要

RabbitMQ 接続を使用して、RabbitMQ ブローカに接続し、キューとの間でデータの読み取りおよび書き込みを行います。

RabbitMQ 接続を作成した後、接続を検証して保存します。

次に、RabbitMQ 接続をパブリッシュして、**[メタデータ]** タブをクリックし、接続用に生成されたプロセスオブジェクトを表示できます。

基本接続プロパティ

次の表に、接続の作成ページの【プロパティ】タブで設定可能な基本プロパティを示します。

プロパティ	説明
名前	必須。Process Designer での識別に使用される、RabbitMQ 接続の名前。 名前はアルファベットで始まり、アルファベット、数値、ハイフン (-) のみを含めることができます。 同じプロジェクト内で接続に同じ名前を指定することはできません。
場所	オプション。接続を保存するプロジェクトまたはフォルダの場所。 【参照】 をクリックして場所を選択します。 [Explore (参照)] ページが現在アクティブになっていて、プロジェクトまたはフォルダが選択されている場合、接続のデフォルトの場所はその選択されているプロジェクトまたはフォルダです。そうでない場合、デフォルトの場所は直近で保存されたアセットの場所です。
説明	オプション。接続の説明。
タイプ	必須。コネクタまたはサービスコネクタに使用する接続のタイプ。 [RabbitMQ] を選択します。
実行日時	必須。接続のランタイム環境。Secure Agent グループまたは Secure Agent マシン上で接続を実行できます。
接続テスト	RabbitMQ コネクタではサポートされていません。
OData 対応	RabbitMQ コネクタではサポートされていません。

基本プロパティを設定した後、次のプロパティも定義する必要があります。

- RabbitMQ 接続タイプに適用できるプロパティ。
- RabbitMQ 接続のイベントソースプロパティとイベントターゲットプロパティ。

RabbitMQ 接続のプロパティ

接続タイプに **【RabbitMQ】** を選択すると、接続の作成ページの【プロパティ】タブで、RabbitMQ 固有の接続プロパティを設定できます。

接続設定

接続先となる RabbitMQ ホストおよび RabbitMQ ポートを指定する必要があります。TLS 認証または SSL 認証を有効にすることもできます。

次の表では、**接続設定** セクションで設定する必要がある RabbitMQ 接続のプロパティを定義しています。

プロパティ	説明
ホスト名	必須。Secure Agent が TCP 接続を使用して接続する必要がある、RabbitMQ インスタンスまたはクラスタが実行されているマシンのホスト名。
ポート	必須。RabbitMQ インスタンスまたはクラスタを実行するホストのポート番号。 デフォルトは 5762 です。
仮想ホスト	オプション。RabbitMQ チャンネルの接続先となる仮想ホストを指定します。 デフォルトは / です。
TLS プロトコル	オプション。RabbitMQ への接続に、TLS または SSL プロトコルを使用するかどうかを指定します。 次のいずれかの値を選択します。 <ul style="list-style-type: none">- TLS- SSLv3- 無効 デフォルトは 無効 です。

アクセス設定

ユーザー名とパスワードを設定すると、異なる特権と権限を使用して RabbitMQ インスタンスに接続できます。ユーザーに、特定の仮想ホストに対する権限を付与することもできます。

次の表では、**アクセス設定** セクションで設定する必要がある RabbitMQ 接続のプロパティを定義しています。

プロパティ	説明
ユーザー名	必須。RabbitMQ に接続するためのユーザー名。
パスワード	必須。RabbitMQ に接続するためのパスワード。

セキュリティプロトコル設定

RabbitMQ プロデューサ、RabbitMQ コンシューマ、RabbitMQ インスタンス間でデータを暗号化して安全に転送されるように、TLS または SSL 認証を設定できます。TLS または SSL 認証を設定するときは、認証局が証明書に署名して RabbitMQ クライアントに発行します。RabbitMQ ブローカはその証明書を使用して、クライアントの身元を確認します。

SSL 認証を有効にすると、**【セキュリティプロトコル設定】** セクションで次の認証プロパティを設定する必要があります。

プロパティ	説明
すべて信頼	接続時に任意のサーバー証明書を信頼できるようにするには、 【はい】 を選択します。 デフォルトは 【いいえ】 です。
トラストストアの場所	RabbitMQ ブローカとの安全な双方向通信を確立するための TLS または SSL 証明書を含む、Secure Agent マシン上のトラストストアファイルの絶対パスとファイル名。 例: C:/SSL/Certs_208/icrt-truststore.jks 空白のままにすると、接続は JVM 引数 <code>javax.net.ssl.trustStore</code> で設定されたデフォルトのトラストストアを使用します。
トラストストアのパスワード	TLS 証明書または SSL 証明書を含むトラストストアファイルのパスワード。 空白のままにすると、接続は JVM 引数 <code>javax.net.ssl.trustStorePassword</code> で設定されたデフォルトのパスワードを使用します。
キーストアの場所	RabbitMQ ブローカとの安全な双方向通信を確立するために必要なキーと証明書を含む、Secure Agent マシン上のキーストアファイルの絶対パスとファイル名。 例: C:/SSL/Keys/icrt-keystore.jks 空白のままにすると、接続では JVM 引数 <code>javax.net.ssl.keyStore</code> で設定したデフォルトのキーストアが使用されます。 【すべて信頼】 プロパティが 【はい】 に設定されているか 【いいえ】 に設定されているかに関係なく、この接続はサーバー証明書を信頼します。
キーストアのパスワード	セキュアな通信のための TLS または SSL 証明書を含むキーストアファイルにアクセスするためのパスワード。 空白のままにすると、接続では JVM 引数 <code>javax.net.ssl.keyStorePassword</code> で設定したデフォルトのパスワードが使用されます。

詳細設定

メッセージを正常にキューに配信するために、詳細 RabbitMQ 属性を設定します。

次の表では、**【詳細設定】** セクションで設定する必要がある RabbitMQ 接続のプロパティを定義しています。

プロパティ	説明
接続タイムアウト	必須。RabbitMQ への接続を試行中に待機する秒数。 デフォルトは 60 秒です。
ハートビートのタイムアウト	必須。RabbitMQ の接続に対して要求された、秒単位でのハートビートのタイムアウト。 デフォルトは 60 秒です。

プロパティ	説明
必須メッセージルーティングの有効化	オプション。メッセージをキューに転送できない場合に例外をスローするには、 【はい】 を選択します。 【いいえ】 を選択すると、RabbitMQ サーバーは警告なくメッセージを削除します。デフォルトは 【はい】 です。
メッセージの保証された配信の有効化	オプション。メッセージを配信できず、必須メッセージルーティングが有効になっている場合に例外をスローするには、 【はい】 を選択します。デフォルトは 【はい】 です。

RabbitMQ イベントソースプロパティ

RabbitMQ イベントソースを作成するときは、RabbitMQ コンシューマを作成して RabbitMQ メッセージを読み取ります。RabbitMQ メッセージを読み取るプロセスで、各 RabbitMQ イベントソースを使用できます。

作成する各 RabbitMQ 接続について、1 つまたは複数のイベントソースを追加できます。イベントソースは、指定した RabbitMQ キューの新しいメッセージをリスンまたは監視する開始イベントとして機能します。RabbitMQ 接続のイベントソースの定義後は、Secure Agent 上でのみ、接続をパブリッシュできます。プロセスでイベントソースを使用して、メッセージをキューからプロセスオブジェクトとしてコンシュームできます。その後、そのプロセスを、RabbitMQ 接続をパブリッシュしたときと同じ Secure Agent にデプロイできます。

RabbitMQ 接続のイベントソースを作成するには、**【イベントソース】** タブで **【イベントソースの追加】** をクリックします。イベントソースタイプで **【RabbitMQ モニタ】** を選択します。

次の表に、設定できる基本的なイベントソースプロパティを示します。

プロパティ	説明
名前	必須。Process Designer に表示されるイベントソースの名前。この名前は、この接続に一意である必要があります。 このプロパティは、接続を作成した後に変更できます。名前は 128 文字以内で指定し、空白および以下の特殊文字は使用できません。 ~`!\$%^&*()-+={} \`;"'<, > . ? /
説明	オプション。Process Designer に表示される RabbitMQ イベントソースの説明。 説明は、4,000 文字を超えることはできません。
有効	このイベントソースをパブリッシュ後即時に使用できるようにするには、 【はい】 を選択します。このイベントソースを使用準備ができるまで無効にするには、 【いいえ】 を選択します。デフォルトは 【はい】 です。

次の表に、設定できる RabbitMQ イベントソースプロパティを示します。

イベントソースプロパティ	説明
キュー	必須。メッセージをコンシュームできる RabbitMQ キューの名前。
ペイロード形式	<p>必須。宛先で想定されるメッセージペイロード形式のタイプを指定します。</p> <p>次のいずれかのオプションを選択します。</p> <ul style="list-style-type: none"> - TEXT - XML - JSON - BINARY <p>デフォルトは【TEXT】です。</p>
オブジェクトリストのフィールド名	<p>オプション。ペイロード形式が JSON の場合、キューはオブジェクト配列を含む可能性がある JSON メッセージを受け入れます。その場合は、RabbitMQ メッセージ本文のオブジェクトリストのフィールド名として使用する値をここに入力します。</p>
デッドレタールーティングの有効化	<p>オプション。問題が発生したメッセージをデッドレターキューにルーティングするには、【はい】を選択します。</p> <p>注: RabbitMQ ブローカのデッドレタールーティングを設定している場合、【デッドレタールーティングの有効化】フィールドを必ず無効にする必要があります。</p> <p>デフォルトは【いいえ】です。</p>
デッドレターエクスチェンジ	デッドレター交換の名前。空白の場合は、デフォルトの交換が使用されます。
デッドレタールーティングキー	メッセージをデッドレターキューにルーティングするためのデッドレターエクスチェンジのルーティングキー。
その他の属性	<p>オプション。イベントソースの RabbitMQ 詳細属性を設定する必要がある場合は、ここで指定できます。Informatica グローバルカスタマサポートまでお問い合わせください。</p>
メッセージの自動確認の有効化	<p>オプション。メッセージを自動的に確認する場合は、【はい】を選択します。</p> <p>注: RabbitMQ ブローカのデッドレタールーティングを設定している場合、【メッセージの自動確認の有効化】フィールドを必ず無効にする必要があります。そうしないと、メッセージがデッドレターキューにルーティングされません。</p> <p>デフォルトは【いいえ】です。</p>
プリフェッチ数	<p>オプション。RabbitMQ チャンネルで配信された、認識されていないメッセージの最大数。確認されていないメッセージを無制限に配信するには、0 を指定します。</p> <p>デフォルトは 100 です。</p>

各イベントソースのステータスは、パブリッシュされた接続で表示できます。イベントソースのステータスが【停止】の場合は、接続を再パブリッシュしてイベントソースを再開できます。接続を再パブリッシュすると、接続内のすべてのイベントソースがデフォルトで開始されます。

リスナベース接続でのイベントソースの開始と停止の詳細については、「*Cloud アプリケーション統合と Monitor 用のコネクタ*」を参照してください。

RabbitMQ イベントターゲットプロパティ

RabbitMQ イベントターゲットを作成するときは、RabbitMQ プロデューサを作成して RabbitMQ メッセージを書き込みます。RabbitMQ メッセージを書き込むプロセスで、各 RabbitMQ イベントターゲットを使用できます。

作成した各 RabbitMQ 接続に、操作を指定する 1 つまたは複数のイベントターゲットを含めて、プロセスからイベントターゲットを呼び出して RabbitMQ エクスチェンジにメッセージを送信することができます。

RabbitMQ 接続のイベントターゲットを作成するには、**【イベントターゲット】** タブで **【イベントターゲットの追加】** をクリックします。イベントターゲットタイプで **【RabbitMQ ライタ】** を選択します。

次の表に、設定できる基本的なイベントターゲットプロパティを示します。

プロパティ	説明
名前	必須。Process Designer に表示されるイベントターゲットの名前。この名前は、この接続に一意である必要があります。 名前は 128 文字以内で指定し、空白および以下の特殊文字は使用できません。 ~`!\$%^&*()-+=[] \\:;'"<>.,?/
説明	オプション。Process Designer に表示される RabbitMQ イベントターゲットの説明。 説明は、4,000 文字を超えることはできません。

次の表に、設定できる RabbitMQ イベントターゲットプロパティを示します。

イベントターゲットのプロパティ	説明
エクスチェンジ名	オプション。プロデュースされたメッセージの送信先となるエクスチェンジの名前。 空白の場合は、デフォルトの交換が使用されます。
ルーティングキー	オプション。ターゲットキューにメッセージをルーティングするためのルーティングキー。
ペイロード形式	必須。宛先で想定されるメッセージペイロード形式のタイプを指定します。 次のいずれかのオプションを選択します。 - TEXT - XML - JSON - バイナリ デフォルトは 【TEXT】 です。
その他の属性	オプション。イベントターゲットの RabbitMQ 詳細属性を設定する必要がある場合は、ここで指定できます。Informatica グローバルカスタマサポートまでお問い合わせください。

RabbitMQ 接続のメタデータ

RabbitMQ 接続をパブリッシュした後、RabbitMQ 接続に関連付けられているアクションとオブジェクトを、**[接続]** ページの **[メタデータ]** タブで確認することができます。

[アクション] セクションでは、RabbitMQ 接続に対して設定されているイベントソースとイベントターゲットを確認できます。

[オブジェクト] では、RabbitMQ 接続に関連付けられているオブジェクトを確認できます。RabbitMQ 接続のパブリッシュされたメタデータは、次のオブジェクトで構成されます。

- **ヘッダー**

[ヘッダー] オブジェクトには、RabbitMQ のヘッダー名とそのヘッダーの値を表す **[名前]** フィールドと **[値]** フィールドが含まれています。

- **メッセージ**

RabbitMQ メッセージはヘッダーと本文で構成されています。RabbitMQ メッセージのヘッダーは、RabbitMQ メッセージの複数の属性を含む名前と値のペアのリストです。RabbitMQ メッセージの本文には、タイプ `$po:$any` のプロセスオブジェクトとしてプロセスにアクセスできるメッセージペイロードが含まれます。

次の図は、RabbitMQ 接続のパブリッシュされたメタデータを示します。

RabbitMQDemoConnectionValid

SaveTest

PropertiesEvent SourcesEvent TargetsMetadata

▼ Actions

Action Name	Description
badQueueWriter	Use this event target to write messages to RabbitMQ.
ColorsWriter	Use this event target to write messages to RabbitMQ.
DefaultExchangeWriter	Use this event target to write messages to RabbitMQ.

▼ Objects

Name	Label	Type	Description
▼ Header	Header		Message header.
name	Name	string	
value	Value	string	
▼ Message	Message		RabbitMQ message.
headers	Headers	RabbitMQDemoConnection.Header[]	
body	Body	Spa:Any	

第 3 章

RabbitMQ プロセス

この章では、以下の項目について説明します。

- [RabbitMQ プロセスの作成, 17 ページ](#)
- [メッセージペイロード, 17 ページ](#)
- [RabbitMQ プロセスのルールとガイドライン, 18 ページ](#)

RabbitMQ プロセスの作成

RabbitMQ 接続、RabbitMQ コンシューマ、RabbitMQ プロデューサを作成すると、プロセスでこれらを使用できます。

RabbitMQ コンシューマプロセス

RabbitMQ コンシューマプロセスを作成するときは、次のガイドラインを使用してください。

- プロセスバインディングはイベントとして選択します。
- RabbitMQ 接続で作成した RabbitMQ コンシューマを選択します。

注: プロセスが自動的に【入力フィールド】セクションに【イベント】フィールドを作成します。

RabbitMQ プロデューサプロセス

RabbitMQ プロデューサプロセスを作成するときは、次のガイドラインを使用してください。

- サービスステップを追加して、RabbitMQ 接続に作成した RabbitMQ プロデューサを選択します。
- 【入力】フィールドで、ターゲットエクスチェンジに書き込む RabbitMQ メッセージを選択します。

メッセージペイロード

プロセスで使用されているイベントには、RabbitMQ メッセージを表すペイロードを持つプロセスオブジェクトが含まれます。次のいずれかの形式で、RabbitMQ メッセージを処理できます。

- JSON
- XML
- バイナリ
- TEXT

RabbitMQ メッセージが JSON または XML 形式の場合、RabbitMQ のメッセージペイロードはタイプ \$po:\$any のプロセスオブジェクトとしてプロセスにアクセスできるようになります。

たとえば、次の RabbitMQ メッセージは JSON 形式で表されます。

```
[{"firstName": "cindy", "lastName": "louis", "department": "qa", "id": 1}, {"firstName": "billy", "lastName": "joe", "department": "dev", "id": 12}]
```

イベントのコンテンツ形式として JSON を選択した場合、Process Designer はプロセスオブジェクトリストとして JSON メッセージペイロードを受け取ります。

```
<message>
  <headers>
    <name>rabbitmq.ROUTINGKEY</name>
    <value>defaultQueue</value>
  </headers>
  <body>
    <employee>
      <firstName>cindy</firstName>
      <lastName>louis</lastName>
      <department>qa</department>
      <id>1</id>
    </employee>
    <employee>
      <firstName>billy</firstName>
      <lastName>joe</lastName>
      <department>dev</department>
      <id>12</id>
    </employee>
  </body>
</message>
```

注: ペイロードが JSON の場合、配列には RabbitMQ メッセージで定義されているようなフィールド名がありません。【オブジェクトリストのフィールド名】プロパティで、暗黙のオブジェクトフィールド名を設定できます。上記の例では、【オブジェクトリストのフィールド名】プロパティで、イベントソースのオブジェクトフィールド名として *employee* が指定されています。

RabbitMQ プロセスのルールとガイドライン

RabbitMQ プロセスについては、次のルールとガイドラインを考慮してください。

- Rabbit イベントソースまたはイベントターゲットのメッセージが JSON または XML 形式である場合、Process Designer はメッセージをプロセスオブジェクトとして受け取ります。
- Rabbit イベントソースまたはイベントターゲットのメッセージがバイナリ形式である場合、Process Designer はメッセージをプロセスオブジェクト内の Base64 でエンコードされた文字列形式で受け取ります。
- RabbitMQ イベントソースまたはイベントターゲットのメッセージがテキスト形式である場合、Process Designer はメッセージをプロセスオブジェクト内の文字列として受け取ります。
- プロセスで RabbitMQ イベントソースを使用する場合、RabbitMQ 接続をバブリッシュしたときと同じ Secure Agent にプロセスをデプロイする必要があります。
- プロセスで RabbitMQ イベントターゲットを使用する場合、Cloud Server、Secure Agent グループ、または任意の Secure Agent マシンにプロセスをデプロイできます。

索引

R

RabbitMQ

概念 [6](#)

概要 [5](#)

RabbitMQ イベントソース

プロパティ [13](#)

概要 [13](#)

RabbitMQ イベントターゲット

プロパティ [15](#)

概要 [15](#)

RabbitMQ コネクタ

概要 [5](#)

実装 [5](#)

RabbitMQ コンシューマ

作成 [13](#)

RabbitMQ コンシューマのプロセス

ガイドライン [17](#)

RabbitMQ コンシューマプロセス

作成 [17](#)

RabbitMQ プロデューサ

作成 [15](#)

RabbitMQ プロデューサのプロセス

ガイドライン [17](#)

RabbitMQ プロデューサプロセス

作成 [17](#)

RabbitMQ 接続

プロパティ [10](#)

メタデータ [16](#)

概要 [9](#)

基本プロパティ [10](#)