Integrating PostilionPRM with ReD Shield

PostilionPRM

Release 2.0

7 Febuary 2018

ACI Worldwide

Offices in principal cities throughout the world

www.aciworldwide.com

Americas +1 402 390 7600
Asia Pacific +65 6334 4843
Europe, Middle East, Africa +44 (0) 1923 816393

Publish Date: February 2018

About ACI Worldwide

ACI Worldwide, the Universal Payments (UP) company, powers electronic payments for more than 5,100 organizations around the world. More than 1,000 of the largest financial institutions and intermediaries as well as thousands of leading merchants globally rely on ACI to execute $14 trillion each day in payments. In addition, thousands of organizations utilize our electronic bill presentment and payment services. Through our comprehensive suite of software and SaaS-based solutions, we deliver real-time, any-to-any payments capabilities and enable the industry's most complete omni-channel payments experience. To learn more about ACI, please visit http://www.aciworldwide.com. You can also find us on Twitter @ACI_Worldwide.

Contents

List of figures

About this publication

Audience

This document is intended for operators who are responsible for configuring the PostilionPRM interface to successfully integrate with an ACI ReD Shield® and Universal Payments Framework (UPF) deployment. The personnel using this information should have adequate knowledge and experience in configuring a Postilion system.

Terminology

The following acronyms are used in this document:

Acronym	Term
TM	Transaction Manager
RPRA	RetailPaymentsRiskAnalyze
MCAS	Mission Critical Application Server
UPF	Universal Payments Framework
SSR	Static Service Registry
DSR	Dynamic Service Registry
BSI	Business Service Infrastructure

The following concepts are used in this document:

Concept	Definition
External transaction processing	Processing performed by an entity external to Transaction Manager for a particular transaction. See the *Realtime Framework User Guide* for more information.
Customer transaction	The original transaction initiated by the customer involved in a transaction chain that requires external transaction processing.
External processing transaction	The auxiliary transaction generated by Transaction Manager when performing external transaction processing on a customer transaction. The external processing transaction is associated with the customer transaction.

Platform Manager	The application that enables the monitoring and management of UPF(MCAS). It is an on-line application and needs to be connected to a UPF(MCAS) instance to carry out its functions. See the “Application tools” section of the *UPF Operations Guide* for more details.

Section 1: Introduction

Overview

ReD Shield is integrated with the PostilionPRM interface using the RetailPaymentsRiskAnalyze service. A service-based architecture is used to facilitate communication between a Postilion system and a ReD Shield server via UPF (MCAS).

To integrate Red Shield with the PostilionPRM interface, do the following:

Note: The sink nodes that will be used for risk analysis must have already been configured in Transaction Manager before you start these configuration steps.

1. Configure the PostilionPRM service. (See the Service configuration section)
2. Configure the PostilionPRM interface to communicate with UPF (MCAS). (See the UPF (MCAS) configuration section)
3. Configure the required external processing and control node. See the "Configuring external processing nodes" and "Configuring control nodes" sections of the *Realtime Framework User Guide* for information on configuring these nodes.
4. Configure the interchange or interchanges (See the Interchange configuration section)
5. Configure active/active (optional) (See the Active/active configuration section)
6. Configure SSL to secure communication between PostilionPRM and UPF (MCAS) (See the SSL configuration section)
7. Configure tracing (optional) (See the Tracing configuration section)

System requirements

When integrating PostilionPRM with ReD Shield, the following operating system and minimum versions are supported:

- Windows OS
- Realtime Framework v5.6
- PostilionPRM v2.0 Patch 15
- Universal Payments Framework v3.2.2

Architectural overview

The following diagram shows how the PostilionPRM interface fits into a Postilion system when integrating with ReD Shield:

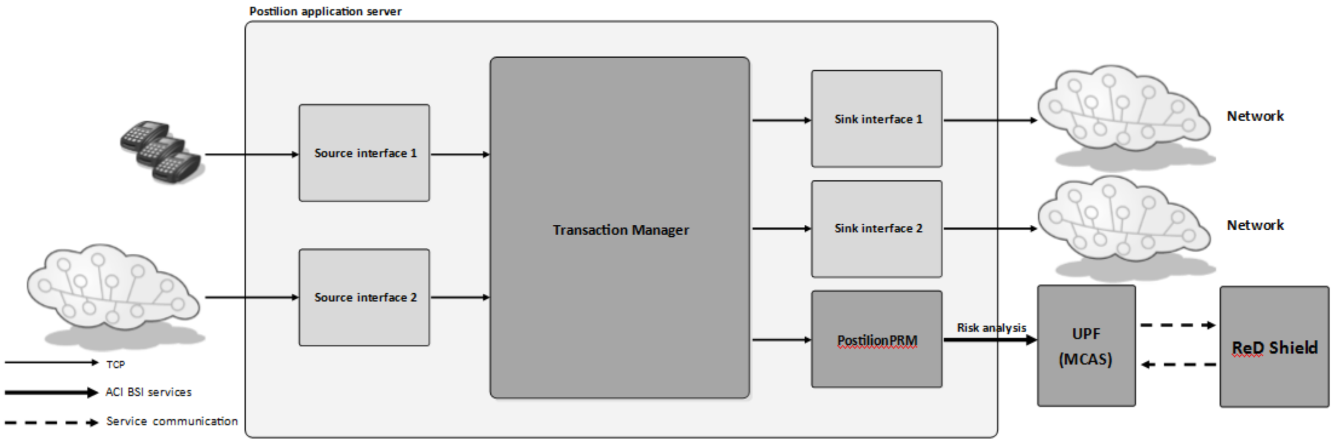


Figure 1: Architectural overview

The PostilionPRM interface is configured as a sink interchange in the Postilion system:

- On the external processing sink node, transaction request messages from Transaction Manager are sent to the ReD Shield server. These messages request the ReD Shield server to perform **real-time risk analysis** while the transaction is being authorized. Transaction Manager considers the result of the risk analysis, and may decline and possibly reverse the transaction as a result.
- On the control node, transaction notification messages from Transaction Manager are sent to the ReD Shield server. These messages notify the ReD Shield server of transactions that have already been processed by the Postilion system. The ReD Shield server performs **near real-time risk analysis** on these transactions. Because the risk analysis is performed after the transaction is already authorized, it does not affect the outcome of the authorization.

Note: When integrating PostilionPRM with ReD Shield, you must configure both real-time and near real-time risk analysis interchanges. All 0100/0200 messages (request messages) are handled by the real-time channel, while 0220 messages (standalone advices) are handled by the near real-time risk analysis channel. ReD Shield does not differentiate between real-time and near real-time requests.

Real-time risk analysis message flow

The following diagram and steps describe how Transaction Manager interfaces with PostilionPRM to perform real-time risk analysis on a 0200 message (transaction request):

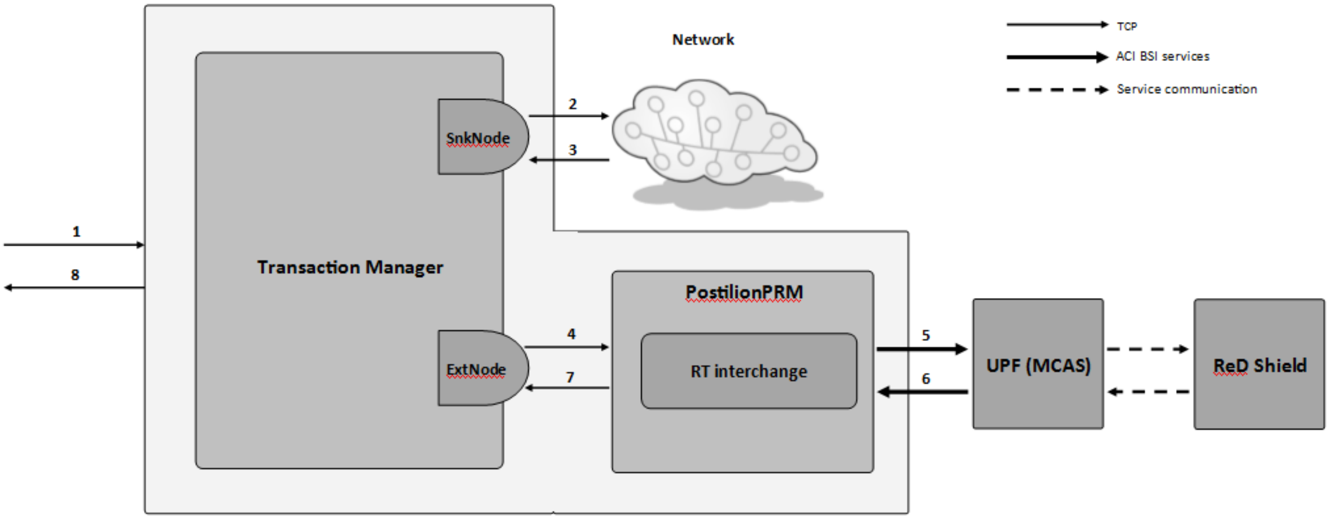


Figure 2: Real-time message flow

1. Transaction Manager receives a 0200 message (transaction request) from a source node, and then sends the 0200 message to the applicable sink node.
2. The sink node interface sends a request to the upstream entity.
3. The sink node interface receives a response from the upstream entity.
4. Transaction Manager sends a message to the applicable **external processing sink node,** which sends a message to PostilionPRM.
5. PostilionPRM sends a real-time risk analysis request message to ReD Shield through UPF (MCAS).
6. ReD Shield sends a risk analysis response. PostilionPRM receives this response from ReD Shield through UPF (MCAS).
7. PostilionPRM sends a 0210 message (transaction request response) to Transaction Manager.
8. Transaction Manager sends the 0210 message to the source node.

Note: The above flow represents an external processing sink node configured for post-authorization risk analysis. If pre-authorization risk analysis is configured, risk analysis would be performed prior to sending the request to the upstream entity.

Near real-time risk analysis message flow

The following diagram and steps describe how Transaction Manager interfaces with PostilionPRM to perform near real-time risk analysis on a 0220 message (transaction request).

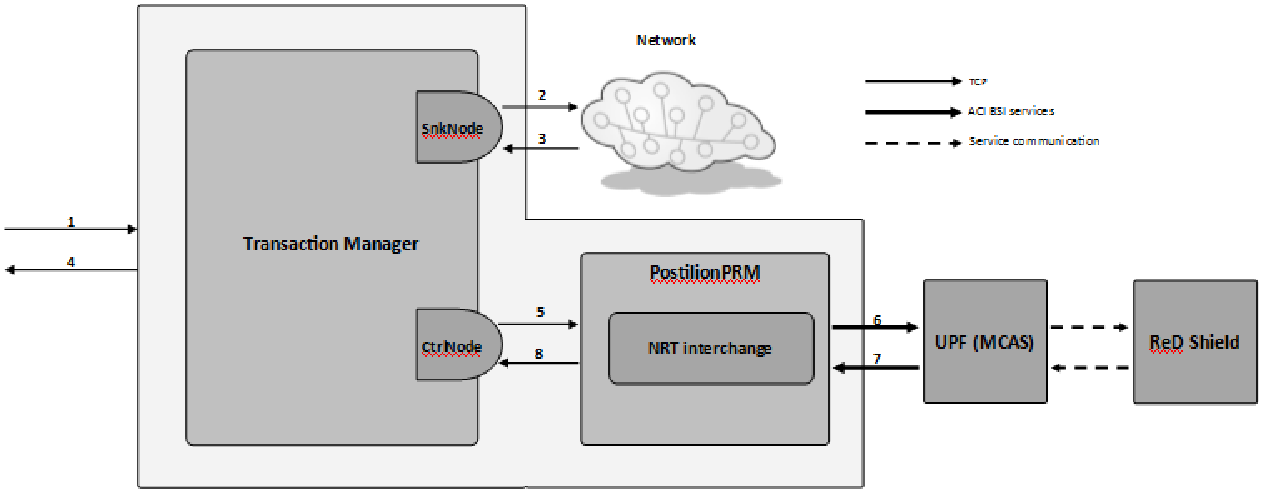


Figure 3: Near real-time message flow

1. Transaction Manager receives a 0220 message (transaction advice) from a source node, and then sends the 0220 message to the applicable sink node.
2. The sink node interface sends a request to the upstream entity.
3. The sink node interface receives a response from the upstream entity.
4. Transaction Manager sends a 0230 message to the source node.
5. Transaction Manager sends a 9220 message (transaction notification) to the PostilionPRM sink node interface. The interchange is associated with a sink node that is configured as a **control node** for the route selected for the 0220 message, and will provide a near real-time (NRT) RetailPaymentsRiskAnalyze service.
6. PostilionPRM sends a message through UPF (MCAS) to ReD Shield for risk analysis.
7. PostilionPRM receives a risk analysis response from ReD Shield through UPF (MCAS).
8. PostilionPRM sends a 9230 message (transaction notification response) to Transaction Manager.

Service overview

ReD Shield is integrated with the PostilionPRM interface using the RetailPaymentsRiskAnalyze service. This service enables the Retail Payment Risk Analysis high-level component. See the "Overview" section of the *PostilionPRM User Guide* for more details on each high-level component.

RetailPaymentsRiskAnalyze service

The RetailPaymentsRiskAnalyze service is provided by ReD Shield via UPF (MCAS) and is used by the PostilionPRM interface to perform risk analysis on transactions that are processed by the Postilion system. PostilionPRM is able to perform near real-time and real-time risk analysis when fully integrated with ReD Shield and UPF (MCAS).

Limitations

Operating system

PostilionPRM and ReD Shield integration is supported only on a Windows operating system.

Card Status Modification service

ReD Shield does **not** support the Card Status Modification service that PostilionPRM provides.

Demographic Data Uploads service

ReD Shield does **not** support demographic data extracts and uploads. As such, the scheduled job *PostilionPRM - Extract - Demographic Data*, which is installed with PostilionPRM, should be **disabled manually**. See the *Realtime Framework User Guide* for information on disabling scheduled jobs.

Record duplication prevention

ReD Shield does not support multiple risk analysis requests for the same transaction. To prevent record duplication, the control nodes configured with PostilionPRM must be configured with the *Store-and-forward-Queue Transmission Limit* sink node option set to 1.

Active/active with UPF (MCAS)

Automatic failover from one UPF (MCAS) server to another is only supported when the Static Service Registry (SSR) BSI configuration is used. The Service Configurator response file for ReD Shield only supports a single failover location. Contact your primary support provider if more failover locations are required.

When a failover location is specified in the response file, the location selection will be configured as *firstavailable*. This is an optional BSI configuration that enables the PostilionPRM interface to always use the primary UPF (MCAS) location, and only use the second or remaining locations if all preceding locations have become unavailable. This means that the UPF (MCAS) server configured as the *services.retailpaymentsriskanalyze.consumer* in the Service Configurator response file will always be used when both locations are online. The UPF (MCAS) server configured as the *services.retailpaymentsriskanalyze.consumer.failover* will only be used when the primary goes offline. Thereafter if *services.retailpaymentsriskanalyze.consumer* comes back online, all external processing requests will resume being sent to it once again.

Section 2: Service configuration

The PostilionPRM service is configured using the service configurator wizard. This wizard will create or update the configuration files required to enable communication with UPF (MCAS) and ReD Shield. See the “Service configuration” section in the *PostilionPRM User Guide* for information on configuring the PostilionPRM service. Note that running the Service Configurator will override any manual custom configuration performed on the configuration files.

The following properties must be configured in the appropriate response file:

Property	**Setting**
service.serviceregistry.type	Set to **static**. This will indicate that PostilionPRM should connect via a Static Service Registry (SSR) when consuming or providing services.
services.risk.engine.type	Set to **RS** This will indicate that PostilionPRM will be integrated with **ReD Shield**.
services.primary.siteid	Optional property to configure the ID of the site where the primary UPF (MCAS) server is located. The default value is “Site1”. This value should align with the provider site configured on Platform Manager. See the “UPF (MCAS) configuration” section for more details
services.retailpaymentsriskanalyze.providerid	Optional property to configure the ID of the provider. The default value is “RetailPaymentsRiskAnalyze1”. This value should align with the provider ID configured on Platform Manager. See the “UPF (MCAS) configuration” section for more details
services.retailpaymentsriskanalyze.consumer.address services.retailpaymentsriskanalyze.consumer.port	*address* should be the remote hostname or IP address of where PostilionPRM should send requests for this service *port* should be the port configured for the RetailPaymentsRiskAnalyze service provided by the ReD Shield server.

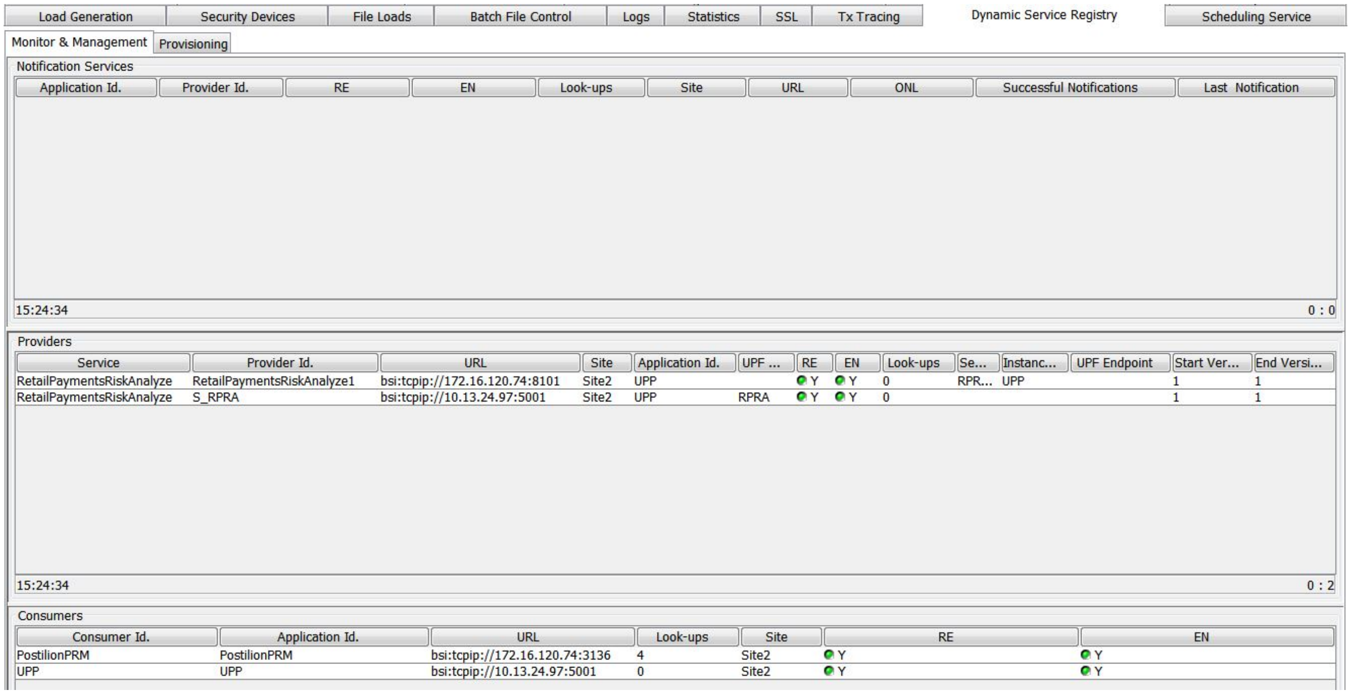
services.retailpaymentsriskanalyze.consumer.failover.address services.retailpaymentsriskanalyze.consumer.failover.port	Optional properties to configure the failover site. If failover for this service is not required, leave these property values blank. *address* should be the remote hostname or IP address of where PostilionPRM should send requests for this service *port* should be the port configured for the RetailPaymentsRiskAnalyze service provided by the ReD Shield server.
services.retailpaymentsriskanalyze.response.address services.retailpaymentsriskanalyze.response.port	*address* should be the externally accessible hostname or IP address of the system where PostilionPRM is installed. *port* should be any unused port that is accessible from the UPF (MCAS) system.

Section 3: UPF (MCAS) configuration

UPF (MCAS) communicates with the ReD Shield system to provide the PostilionPRM interface with the RetailPaymentsRiskAnalyze service. For the PostilionPRM interface to successfully integrate with ReD Shield, PostilionPRM must be configured to communicate with UPF (MCAS).

The PostilionPRM application ID must be configured on Platform Manager for UPF (MCAS) to recognize PostilionPRM as a consumer. An application ID is configured as a subset of an existing site ID using Platform Manager. The default value for PostilionPRM's application ID is "PostilionPRM".

The site ID and provider ID configured in the service configurator response file must be the same value (case sensitive) as in Platform Manager. Their values must align with what is in the "Provider Id" column and the "Site" column in Platform Manager as shown in Figure 4. See the "Service configuration" section for how to configure the site ID and provider ID in PostilionPRM.



Notification Services

Application Id.	Provider Id.	RE	EN	Look-ups	Site	URL	ONL	Successful Notifications	Last Notification

15:24:34 0 : 0

Providers

Service	Provider Id.	URL	Site	Application Id.	UPF ...	RE	EN	Look-ups	Se...	Instanc...	UPF Endpoint	Start Ver...	End Versi...
RetailPaymentsRiskAnalyze	RetailPaymentsRiskAnalyze1	bsi:tcpip://172.16.120.74:8101	Site2	UPP		● Y	● Y	0	RPR...	UPP		1	1
RetailPaymentsRiskAnalyze	S_RPRA	bsi:tcpip://10.13.24.97:5001	Site2	UPP	RPRA	● Y	● Y	0				1	1

15:24:34 0 : 2

Consumers

Consumer Id.	Application Id.	URL	Look-ups	Site	RE	EN
PostilionPRM	PostilionPRM	bsi:tcpip://172.16.120.74:3136	4	Site2	● Y	● Y
UPP	UPP	bsi:tcpip://10.13.24.97:5001	0	Site2	● Y	● Y

Figure 4: Example - Configured UPF (MCAS) Platform Manager

Note: It is possible to configure PostilionPRM to use the dynamic service registry (DSR) to connect to UPF (MCAS). However, using the static service registry (SSR) is recommended.

Connecting the Static Service Registry

Service configuration for the PostilionPRM interface is necessary for PostilionPRM to successfully connect to the static service registry (SSR) on the UPF (MCAS). The following properties must be configured in the response file to reflect the correct address and port on which the SSR service is listening on UPF (MCAS):

- `services.retailpaymentsriskanalyze.consumer.address`
- `services.retailpaymentsriskanalyze.consumer.port`

See “Service configuration” in this guide, and the “Service configuration” section in the *PostlionPRM User Guide,* for more information.

Connecting the Dynamic Service Registry

The service configuration for the PostilionPRM interface is necessary for PostilionPRM to successfully connect to the Dynamic Service Registry (DSR) on UPF (MCAS). The following properties must be configured in the response file to reflect the correct address and port on which the DSR service is listening on UPF (MCAS):

- `services.dynamicserviceregistry.address`
- `services.dynamicserviceregistry.port`

See “Service configuration” in this guide, and the “Service configuration” section in the *PostlionPRM User Guide,* for more information.

Section 4: Interchange configuration

Select the **Interchanges** page in the Realtime applications configurator to configure the PostilionPRM interface within the Postilion system.

RetailPaymentsRiskAnalyze service

PostilionPRM can be configured to consume the RetailPaymentsRiskAnalyze service to perform near real-time and real-time risk analysis. A single interchange must be configured for each service consumer.

Note: Prior to configuring PostilionPRM interchanges, the sink nodes that will be used for risk analysis must have already been configured in the *Transaction Manager* console in Realtime Framework, and the following must be taken into consideration:

- An **external processing sink node** must be configured for the real-time risk analysis channel.
- A **control node sink node** must be configured for the near real-time risk analysis channel.

Both real-time and near-real time interchanges must be configured when integrating with ReD Shield. ReD Shield supports the following message types on the given channel:

Message Type	Channel
0100	Real-time
0200	Real-time
9120	Near real-time
9220	Near real-time

Message types other than the ones specified above are not currently supported by ReD Shield.

To configure a PostilionPRM interchange, go to the **Interchanges** tab in the Applications configuration console and complete the following steps:

1. Create a new entry for the RetailPaymentsRiskAnalyze service in the Interchanges category:

Variable	**Setting**
Interface	Select *PostilionPRM*
Interchange	Provide a name for the interchange. For example, *PostAuthRiskAnalyze*.
User Parameters	**Note**: Parameters should be specified using the following template and must use the predefined keys where values must be encapsulated in single quotation marks and separated by a comma. For example, *key1='value1', key2='value2', key3='value3'.* • Provide the Acquiring Institution ID (variable length up to 11 characters) to be used to identify acquirer transactions in PostilionPRM. For example, *AcquiringInstId='ACQ00001'*. See the "Message Identification" section in the *PostilionPRM User Guide* for more information. • ***<Optional>***: Provide the server identifier value (variable length up to 4 characters) that will be used to identify the active/active partner sending the transactions to PostilionPRM. The value must be configured uniquely across the active/active partners. For example*. ServerId='PST1' and ServerId='PST2'*. • ***<Optional>***: Provide the risk management system value that will be used to identify which risk management system to use. *RiskManagementSystem='RS'* must be specified when integrating with ReD Shield. If this parameter is not specified, the default parameter value of *RiskManagementSystem='PRM'* will be used.
Sink Node	Select the name of a sink node created in the **Sink Nodes** page of the Transaction Manager Configurator. **Note:** Ensure that the correct sink node is chosen. If the interchange is configured to perform **near real-time** risk analysis, then the sink node should be configured as a **control node**. If the interchange is configured to perform **real-time** risk analysis, then the sink node should be configured as an **external processing node**. The **external processing type** on the external processing sink node can be set to *Pre-Authorization Risk Analysis* or *Post-Authorization Risk Analysis*. See the sections on configuring nodes in the "Routing Configuration" section of the *PostilionPRM User Guide* for more information.

2. Create one Service Access Point (SAP) in the **SAPs** category for each RetailPaymentsRiskAnalyze service entry configured in the previous step:

Variable	Setting
SAP	Any name. Suggestion: Use the same name as the name used for the interchange. For example, *PostAuthRiskAnalysis*.
Type	Select **client**.
Protocol	Select **Generic Protocol**.
SAP Factory Class Name	Enter `postilion.realtime.postilionprm.ipc.SapFactory`
Address	Leave this blank
Setup Data	Enter `postilion.realtime.postilionprm.service.retailpaymentsriskanalyze.RPRAProxyDriver.` This configures a consumer of the RetailPaymentsRiskAnalyze service in order to configure ReD Shield to perform risk analysis.

3. Create one connection for the configured client SAP for the RetailPaymentsRiskAnalyze service entry:

Variable	Setting
Direction	Select **Dual**
Priority	Select **High**
Connect on Demand	Do not enable this option (the default is set to **disabled**)
Connect Timeout	Enter `0`
Inactivity Timeout	Enter `0`
Address	Enter `RetailPaymentsRiskAnalyze` as the name of the service
Setup Data	Leave this blank

Generic field mapper

The locations of some source fields in messages to the PRM application (that will be copied to locations in the message to ReD Shield) are configurable. See the “Generic Field Mapper Configuration” section in the *PRM User Guide* for instructions on configuring these field mappings to suite the particular system configuration. If this configuration is not done, the default source locations for these fields will be used.

Section 5: Active/active configuration

PostilionPRM interfaces can be connected to a single UPF (MCAS) instance, or multiple UPF (MCAS) instances.

For each instance of UPF (MCAS) that PostilionPRM connects to, configuring the Postilion side entails running the service configurator to duplicate the configuration on each Realtime active/active partner. See the steps outlined in the “Service configuration” section of the *PostilionPRM User Guide* for more details.

The “Service Configurator configuration” and “Server identification” sections in this guide describe the requirements for active/active.

Note: For these changes to take effect, you will need to RESYNC. See the *Realtime Framework User Guide* for more information about performing a resync.

Service Configurator configuration

A PostilionPRM interface can be configured to communicate with multiple UPF (MCAS) instances. This configuration is optional.

Note: See the “Service configuration” section for details on configuring a failover UPF (MCAS) instance.

Server identification

For the active/active PostilionPRM instances to transact with ReD Shield, an identifier for each of the active/active partners needs to be configured. This is used internally by PostilionPRM for transactions from each of the active/active partners to be uniquely identified. A unique server identification value must be provided in the user parameter for each partner.

See the “Interchange configuration” section for more information on how to configure the user parameters for the RetailPaymentsRiskAnalyze service.

Active/active coordination and failure recovery

Automatic failover from a primary to a partner UPF (MCAS) instance is supported. However, there are some limitations to this process. See the “Limitations” section for more information.

Section 6: SSL configuration

It is recommended that all communication between PostilionPRM and UPF (MCAS) be secured as it contains data that could be considered sensitive. PostilionPRM supports the use of SSL/TLS to secure communications.

Communication for each system is divided into two distinct channels. An incoming (server) channel used to receive messages, and an outgoing (client) channel used to send messages. A single SSL certificate can be used to secure both communication channels on the PostilionPRM system. This is the recommended approach, but the use of separate SSL certificates for client and server channels is also supported.

Configuration needs to be performed on both PostilionPRM and UPF (MCAS) to perform SSL encryption. For PostilionPRM-specific configuration, see the “Service configuration” and “SSL/TLS configuration” sections in the *PostilionPRM User Guide*

For UPF (MCAS) specific configuration, see the “Configuring SSL” section of the *UPF Operations Guide*.

Section 7: Optional configuration

Tracing configuration

Tracing can be enabled for events relevant to the PostiltionPRM interface. The following types of events are recorded to the Postilion and UPF (MCAS) traces:

- Postilion events
- Business Service Infrastructure (BSI) events
- UPF (MCAS) events

See the *Realtime Framework User Guide* for more information on configuring system properties.

Postilion event tracing

To enable tracing, configure the following system property:

Application	System Property Name	Description
PostilionPRM	postilion.env.enable_tracing_at_startup	Set to **true** to enable tracing.

Business Service Infrastructure event tracing

The BSI framework is used by the PostilionPRM interface to consume and provide business services. Detailed events generated by the BSI framework can be traced by the PostilionPRM interface. The following system property is used to determine if these detailed events should be included in the application traces.

Application	System Property Name	Description
PostilionPRM	postilionprm.bsi.include_detailed_tracing	Set to **true** to indicate that detailed BSI events should be included in application tracing. **Note:** This behavior can be controlled at run time with a SET command.

Universal Payments Framework event tracing

See the “Tracing Messages” Section of the *UPF Message Tracing Guide* for more information on enabling and configuring message tracing on Platform Manager. Message tracing is the process of capturing additional information about what is occurring during message processing. The information captured during tracing helps to troubleshoot problems and confirm that messages are processed as expected and as defined by business rules. Platform Manager is the interface used to create, customize, enable, disable, initiate, and stop a trace.