

TPIR-PF[®]
RosettaNet Implementation Guide
(RIG)

For Validated 11.00.00
TPIR-PF Core Specification

Trading Partner Implementation Requirements
Presentation Format
(TPIR-PF)

Specification Information

Name	RosettaNet Implementation Guide for TPIR-PF
Version	Validated 11.00.00
Date	30 June 2006

Table of Contents

T1 Document Management	iii
1.1 Legal Disclaimer	iii
1.2 Copyright	iii
1.3 Trademarks	iii
1.4 Related Documents	iii
1.5 Purpose.....	iv
1.6 Document Version History.....	iv
1.7 Acknowledgements	iv
2 Overview	1
2.1 Business Objectives	1
2.2 Technical Objectives.....	1
2.3 Purpose of this Document	2
2.4 Background	2
2.5 Profiling Trading Partner Capabilities	3
2.5.1 MNC with Persistent Connectivity	3
2.5.2 Small and Mid-sized Enterprises with Persistent Connectivity	3
2.5.3 Small and Mid-sized Enterprises with Occasional Connectivity	3
3 Implementation Process Flow	6
3.1 Process Flow Introduction	6
3.2 Forms Design	6
3.3 Merging a PIP with a TPIR-PF form Template	8
3.4 Rendering.....	9
4 Recommendations	10
4.1 Forms Design	10
4.1.1 General Recommendations	10
4.1.2 Personnel Recommendations.....	10
4.1.3 Process Recommendations.....	10
4.1.4 Expertise Recommendations	11
4.2 Merge and Extract	11
4.3 Rendering.....	11
5 References	13
6 Glossary	15
7 Appendix A	19

7.1	XML Data Package (XDP) Rules.....	19
7.2	TPIR-PF Form Template Rules.....	20
7.3	PDF Rules.....	20
8	Appendix B.....	21
8.1	XDP Service Content	21
8.2	PDF Service Content.....	23
9	Appendix C.....	24
9.1	PIP3A4_V11.00.00_047897855_PO_01.01_RequestPurchaseOrder.zip	24
9.2	IntelPurchaseOrder_v37.pdf	24
9.3	PurchaseOrder_V11.00.pdf.....	24
9.4	POsample.xml	24

1 Document Management

1.1 Legal Disclaimer

RosettaNet™, its members, officers, directors, employees, or agents shall not be liable for any injury, loss, damages, financial or otherwise, arising from, related to, or caused by the use of this document or the specifications herein, as well as associated guidelines and schemas. The use of said specifications shall constitute your express consent to the foregoing exculpation.

1.2 Copyright

©2006 RosettaNet. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the inclusion of this copyright notice. Any derivative works must cite the copyright notice. Any public redistribution or sale of this publication or derivative works requires prior written permission of the publisher.

1.3 Trademarks

RosettaNet, Partner Interface Process, PIP and the RosettaNet logo are trademarks or registered trademarks of "RosettaNet," a non-profit organization. All other product names and company logos mentioned herein are the trademarks of their respective owners. In the best effort, all terms mentioned in this document that are known to be trademarks or registered trademarks have been appropriately recognized in the first occurrence of the term.

1.4 Related Documents

The following documents provide additional background and relevant information:

- RosettaNet Trading Partner Implementation Requirements (TPIR-PF) specification, Release 11.00.00A
- RosettaNet Automated Enablement, Trading Partner Implementation Requirements-Presentation Format [TPIR-PF] Use Models document
- RosettaNet Automated Enablement, Trading Partner Implementation Requirements-Presentation Format [TPIR-PF] Requirements document
- RosettaNet Automated Enablement, Trading Partner Implementation Requirements-Partner Interface Process [TPIR-PIP] Requirements document
- PDF Reference: Adobe Portable Document Format, Version 1.5, Adobe Systems Incorporated. Available at http://partners.adobe.com/asn/acrobat/docs/File_Format_Specifications/PDFReference.pdf
- Template 2.0 Specification, Adobe Systems Incorporated. Available at http://partners.adobe.com/asn/tech/pdf/xfa/template_2.0.pdf
- Data Handling 2.0 Specification, Adobe Systems Incorporated. Available at http://partners.adobe.com/asn/tech/pdf/xfa/data_handling_2.0.pdf
- Data Binding 2.0 Specification, Adobe Systems Incorporated. Available at http://partners.adobe.com/asn/tech/pdf/xfa/data_binding_2.0.pdf
- Data Text Handling Specification, Version 2.0, Adobe Systems Incorporated. Available at http://partners.adobe.com/asn/tech/pdf/xfa/data_text_handling_2.0.pdf
- FormCalc 2.0 Specification, Adobe Systems Incorporated. Available at http://partners.adobe.com/asn/tech/pdf/xfa/formcalc_2.0.pdf

- Picture Clause 2.0 Specification, Adobe Systems Incorporated. Available at http://partners.adobe.com/asn/tech/pdf/xfa/picture_clause_2.0.pdf
- Scripting Object Model 2.0 Specification, Adobe Systems Incorporated. Available at http://partners.adobe.com/asn/tech/pdf/xfa/som_2.0.pdf
- XML Data Package 2.0 Specification, Adobe Systems Incorporated Available at http://partners.adobe.com/asn/tech/pdf/xfa/xdp_2.0.pdf

1.5 Purpose

This document includes the following information:

- 1) Overall approach to human-readable visualization of RosettaNet Partner Interface Processes[®] (PIPs[®]) restricted by Specialized Schemas
- 2) Recommendations on how tools, techniques and methodologies for creating a TPIR-PF forms template

This document's primary audience is Solution Providers and PIP Implementers that need to create a TPIR-PF as a means of providing a human-readable, visual representation of a RosettaNet PIP in order to better support their business processes.

1.6 Document Version History

Version	Date	Notes
V11.00.00	30 Jun 2006	Publication of RIG for TPIR-PF Core Specification

1.7 Acknowledgements

This document has been prepared by the Validation Team. Listed below are the legal entities that contributed to the development of this RosettaNet (Recommended) Implementation Guide (RIG).

Adobe Systems, Inc	EPSA	Intel Corporation
E2open, Inc.	Global eXchange Services	SAP

2 Overview

2.1 Business Objectives

Organizations that have adopted RosettaNet standards have realized significant savings and benefits from implementing a process-centric approach to e-business transactions. While the savings thus far have been tangible, they fall short of the potential returns that could be achieved should RosettaNet standards be adopted across ecosystem.

While 100% compliance to current RosettaNet standards would maximize the ROI from implementation investments to date, the business reality is that one size cannot and does not fit all types of trading partners. Flexible and downwardly scalable solutions are needed to traverse a vastly diverse, multi-tier global value chain; solutions which reflect the technical capabilities of each tier to standards-based B2B integration technologies.

What is needed to foster mass adoption is a spectrum approach of RosettaNet offerings, services, and capabilities accompanied by a set of methods and guidelines that enable any trading partner, regardless of size, to uniformly implement solutions either up, down or across the value chain. A more flexible solution is needed to satisfy the information needs of both sender and receiver at either end of revenue scale, and at any point in the value chain.

- Current MNC adopters will increase the return on their RosettaNet investment by reducing the number of non-RosettaNet exchanges with a greater percentage of their trading partners, lowering the cost of future exchanges, and by streamlining customization they are required to do;
- Mid-tier and small suppliers will benefit from a low-cost, easy to deploy mechanism to comply with requirements for RosettaNet exchanges;
- Out-of-sector industries could also benefit from a more agile and more responsive high tech industry supply base. As goods are shared by multiple industries, companies without RosettaNet capability will see the value of the standards.

2.2 Technical Objectives

The technical intent of the TPIR-PF specification is to facilitate RosettaNet usage across the entire supply chain. Predictably, this will require dramatically reduced costs compared to the typical historical costs for implementing RosettaNet over the last 2-3 years. The TPIR-PF combined with the TPIR-PIP specification addresses this by augmenting current RosettaNet standards with new methods and processes that can deliver alternative forms of B2B integration suitable to different use models and trading partner profiles. The technical approach includes the following

- Define the machine-readable specification that defines changes to the schema that describes the specific implementation of a PIP for a trading partner. The TPIR-PIP specification details how a PIP can be further constrained to meet the needs of the specific business process for a specific trading partner.
- Define a non-proprietary standards-based methodology by which client software systems could be developed to view or manually create RosettaNet compliant messages. This specification is called TPIR-PF and defines the presentation format for a RosettaNet PIP. It is ideally suited to those organizations that cannot implement a complete system-to-system integration.

2.3 Purpose of this Document

This document is designed to assist e-business system implementers and solution providers who want to create or implement interoperable software application components that cooperatively execute RosettaNet PIPs using a human-readable form. The document does this by describing of the implementation approaches that was used to validate the Trading Partner Implementation Requirements - Presentation Format (TPIR-PF) specification.

Readers should already be familiar with the Adobe Portable Document Format (PDF) and XML Data Package (XDP) technical descriptions, and the RosettaNet TPIR-PF and TPIR-PIP specifications. The result of the TPIR-PF and TPIR-PIP specifications should be to enable two primary objectives for the RosettaNet community.

- Streamline Execution: Facilitate the rapid implementation of Partner Interface Processes (PIPs) particularly with small and mid-sized enterprises lacking an existing B2B infrastructure
- Accelerate Adoption: Provide for the rapid development of e-business applications that employ RosettaNet PIPs

The TPIR-PF specification combines industry-standard XML-based data exchange using RosettaNet PIPs and the RosettaNet Implementation Framework with the presentation quality, reliability, interactivity, and universal access of Adobe PDF files. Using the TPIR-PF and TPIR_PIP specifications, RosettaNet PIPs can be exchanged with trading partners using the PDF format for viewing, interacting, printing, and archiving in a form that incorporates the PIP schema defined by TPIR-PIP.

2.4 Background

The TPIR-PF consists of the metadata or instructions on how to interpret and visually render a specific type of PIP instance that conforms to a TPIR-PIP schema. The TPIR-PF metadata is a description of how each and every field described by a TPIR-PIP schema for a particular PIP will be visually presented to the user including labels, controls, physical position, fonts, and other graphical attributes. A TPIR-PF provides a visual representation of the PIP instance that fully retains the semantics of the original PIP message.

There are numerous commercial products that provide a presentation layer or user interface which has been specifically developed to receive and manage B2B messages structures. The solutions have provided a means of loosely integrating organizations that have limited backend integration capabilities. The general profile is that of an organization that has been required to support electronic B2B communication but for one or more reasons is unable to accommodate the request.

Generally, a multi-national corporation (MNC), who requires electronic B2B communication from their partners, works with a solution provider to create an acceptable presentation layer, or they create it on their own and hope it assists the user in supplying data that is required to complete a B2B process. Hopefully, the collected data is compliant with the B2B standards specifications. A web portal implementation is the typical manifestation of this type of visual integration, much more so than an exchange of electronic forms.

A portal or set of web screens is not conducive with the rapid implementation of multiple TPIR-PIP versions across separate trading partners. Solutions need to be able to dynamically accept variations in the PIP schemas as defined by a TPIR-PIP and then to display the data in a human readable format. Trading Partner Implementation Requirement – Presentation Format (TPIR-PF)

will support this effort. In addition, unlike a portal, TPIR-PF supports an offline or detached mode of integration.

A TPIR-PF provides instructions on how the TPIR-PIP data should be displayed to the user. The specification leverages available web and XML standards capabilities from other parts of the industry where useful and appropriate. It is not the intent of RosettaNet to define a new presentation format when so many formats already exist. The goal is to provide as much presentation format information as possible to allow a solution to self-enable the human interface and to successfully communicate the semantics of the original PIP message.

For solution providers who already have robust human interfaces that can handle the transformation of TPIR-PIP data, the TPIR-PFs may not be necessary and are certainly not required.

2.5 Profiling Trading Partner Capabilities

During the investigation stage, an attempt was made to identify the various types of trading partner groups among which RosettaNet adoption is low and to define the business constraints that have hindered their adoption of RosettaNet. The result was a classification of trading partners in an ecosystem into 3 broad categories. It is important to understand these categories in relation to the TPIR-PF requirements.

2.5.1 MNC with Persistent Connectivity

Current Multi-National Corporations typically have always-on, real-time B2B solutions which support robust gateway rules and choreographies with their trading partners. There is also the assumption of automatic integration into the backend with no additional processing delays or manual steps. Existing RNIF implementations, and many existing PIP processes and exchanges, assume this 'always on' state.

2.5.2 Small and Midsized Enterprises with Persistent Connectivity

Some Small and Midsized Enterprises may have robust gateway solutions or may employ value added networks that provide an 'always on' appearance although the end consumer of the data may not be 'on'. The impact to existing gateway set-up may be minimal. However, the ASP model is seldom one that supports automatic back end integration, so processing delays and network latency should be expected in some cases.

2.5.3 Small and Midsized Enterprises with Occasional Connectivity

Currently solutions to support intermittent Internet connectivity do not comply with RNIF. In addition, PIP choreography often depends on immediate response which is impossible in occasionally connected environments. The connection of the MNC to the SME with only occasional connectivity was of greatest importance to the companies participating in the RAE research. The needs of the occasionally connected user is therefore of high priority and are particularly well-suited to TPIR-PF.

The table below was constructed by the MMS foundational team. The purpose of the table is to provide more details behind the specific operating characteristics of the trading partner. This is important in understanding why a trading partner would use TPIR-PF.

Feature	Tier 1	Tier 2	Tier 3
Organization			
Company Label	Large (MNC)	Medium (SME)	Small (SME)
Company Description	Multi-national corporation with thousands of employees	Business with hundreds, even a thousand employees, sometimes multi-national	Small organization with tens or maybe a hundred people, regional presence only
Company Revenue	Hundreds of millions to billions	Hundreds of millions	Tens of millions USD
Role in the Supply Chain	Typically a large OEM or EMS, can have multiple roles as an OEM and as a supplier	Typically a supplier of components or subassemblies, occasionally classified as an EMS	Typically a supplier
Internet			
Messaging Connectivity	Consistently high quality of service	Good quality of service	Fair quality to intermittent availability
Presence	7x24x365	Split - 7x24 with a server or intermittent if manual	Occasionally connected
IP	Static IP address	Static or dynamic	Dynamic
Bandwidth	Virtually unlimited	T1 or T3	Fractional T1 to dial-up
Information Technology			
B2B Budget	Hundreds of millions	Tens of millions	Less than one million
B2B Staff	10-20 people dedicated to B2B integration both buy and sell side	1-2 people working on B2B, focus is primarily on sell-side	No IT resources on B2B, most likely network, email and app support only
Backend Integration	From B2B Server to EAI bus to virtually any type of message	Integration capability to one backend system	No backend system integration capability
Backend Systems	Multiple ERP, MRP systems, multiple implementations across many divisions	Mid-range to ERP system, single implementation	Accounting packages, sometime no backend system at all - run on spreadsheets
Data Transformation	In-house use of a high-end data transformation tools for B2B and EAI, and many formats	Variety of tools, tend towards knowledge of one B2B format	No data transformation, manual key entry
B2B Messaging			
Number of Trading Partners	Hundreds to thousands	Tens to hundreds	Tens
Transaction Volume	Thousands to tens of thousands of messages per day	Thousands of messages per day	Tens to hundreds of messages per day
% of Transactions – B2B	15-25%	5-10%	0-5%
# of B2B Processes	Many B2B processes from design, quality, forecast, order and inventory management	Typically just a few depending on biz type - forecast, inventory, order	Centered around order management or VMI model from supply chain master
# of Transaction Types	Ten to thirty different types of messages	Around a dozen types of messages	Online or paper
Largest Transaction Size	Forecast files can get to MBs, design files can get to GBs	Transaction sizes generated are less than 100k	Counted in number of pages
Supported Data Formats	Supports EDI and at least one XML standard, capability to support all	Usually just one format - older firms support EDI, newer companies tend to support XML	Fax, email, web site, spreadsheets
Supported Protocols	In high-tech market, support multiple protocols - EDI-VAN, EDI-INT, RNIF and some Web Services	Usually just one protocol - older firms more than likely supports EDI-VAN and newer firms support RNIF	Web screens, SMTP (email) and some FTP of spreadsheets
Sensitivity to Security	Highly sensitive to security requirements of message exchange	Sensitive to message exchange more along the types of messages such as those including pricing	Not typically aware of security issues, tendency to rely on larger partners to specify

Sensitivity to Reliability	High sensitively driven in large part by the message volume - no other way to really manage	Sensitive to reliability as volume and number of connections increases	Sensitive but overly as volume is low enough so can manage manually as well
Management Needs	Adequate IT staffing tends to make this value proposition not as high	Value depends on the complexity of the application and maturity of B2B skills	Management could provide an easier method for implementing the solution
Messaging Complexity	Messages can be complex but tends to be limited to less popular use models	Somewhat complex with XML two-way.	Not as complex, either posting or responding if using B2B.
Message Timing Requirement	Timing is highly dependent on process so have to enforce to most demanding requirement	Timing is dependent on the partner and the process. Variable.	Timing driven by requirements of customers. Lower volume allows rapid feedback during working hours
Intermediary	Not typically required, adequate IT capabilities, however, added services could assist	Variable perceived need depending on type of connection and volume of transactions	High value as IT capabilities are not typically conversant in B2B integration

3 Implementation Process Flow

There are a number of alternatives that can be used for implementation of the TPIR-PF and TPIR-PIP specifications. The alternatives are defined in the TPIR-PF specification. This section describes the actual process flow that was used in one of the implementations that supported the RosettaNet validation efforts.

3.1 Process Flow Introduction

The TPIR-PF process flow consists of three major parts; designing the Form Template, delivering the Electronic Document and returning the Electronic Document. The following is a list of the major functional parts of the processes.

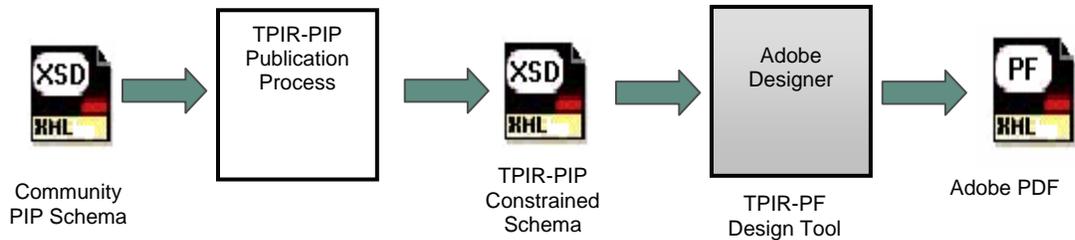
- **Design:** A TPIR-PF Form Template is created that visually represents the PIP. This controls how the PIP will be displayed on an output device, typically a monitor
- **Merge:** The PIP is combined with the TPIR-PF Form Template to create an Electronic Document. An Electronic Document can be in one of two forms; an XML Data Package (XDP) or a Portable Document Format (PDF)
- **Render:** Rendering is the process of visually displaying the Electronic Document in a human-readable format to an output device. Rendering also encompasses updating the Electronic Document as required in a two-action Partner Interface Process
- **Extract:** The PIP instance is extracted from the Electronic Document

3.2 Forms Design

The TPIR-PF can be viewed as an intermediate form that describes how the PIP instance is to be visually rendered. This intermediate form is similar to the methodology used by virtually all B2B implementations in practice today. Organizations using RosettaNet as their integration standard transform the proprietary data structures of their backend systems to and from an intermediate form called a PIP message. Other organizations use OAGI BODs, EDI flat files or other structures as the intermediate form. In each case, the trading partners on each side of the integration are free to implement technology that conforms to their individual requirements such as price, platform, capabilities, and many other factors. The only common attribute of these solutions, whether developed in-house or purchased, is that they consume and produce the intermediate form per the standards specification.

The specification calls for the specific use of the publicly available, royalty-free, Adobe XML Data Package (XDP) 2.0 format or the Adobe Portable Document Format (PDF) 1.5 format. In this implementation, we chose PDF as our intermediate form. PDF is a binary format for representing documents and is general use by millions of businesses, governments and other institutions and individuals to represent information. More importantly, PDF is interactive, providing control such as text boxes, radio button and drop-down list boxes to facilitate data collection.

The overall flow of design the TPIR-PF is illustrated below.



There are two major steps in the design process.

- In the first step, the TPIR-PIP specification guided us in constraining the RosettaNet community schema. The result is a schema that contained only the fields that were required for the integration. This is the TPIR – Trading Partner Implementation Requirements.
- In the second step, an electronic form was created. This is called a TPIR-PF and it represents each of the fields in the TPIR-PIP.

Adobe Designer v7 was the TPIR-PF design tool used to create a visual layout of the PIP using labels, controls and graphic elements in such a way that the visual representation retains the semantics of the PIP. TPIR-PIP is an XML Schema Definition (XSD) that defines the specific implementation of a PIP, that is, the specific subset of fields and constraints agreed upon by two parties. The TPIR-PIP is used by Adobe Designer to bind the contents of the schema to the TPIR-PF Form Template.

The output of Adobe Designer is an XML structure called the TPIR-PF Form Template. This is really a blank form that consists of the metadata or instructions on how to interpret and visually render a PIP instance that conforms to a TPIR-PIP schema for human interaction. The TPIR-PF Form Template provides a description of how each and every field described by a TPIR-PIP schema will be visually presented to the user including labels, controls, physical position, fonts, and other graphical and interactive attributes. In this implementation, the TPIR-PF provides a visual representation of the PIP instance that fully retained the semantics of the PIP.

Adobe Designer can be purchased from Adobe or any one of their many distributors. The package tends to be less expensive from the distribution channel. Recommendations on the usage of Adobe Designer are outlined in the next section of this document.

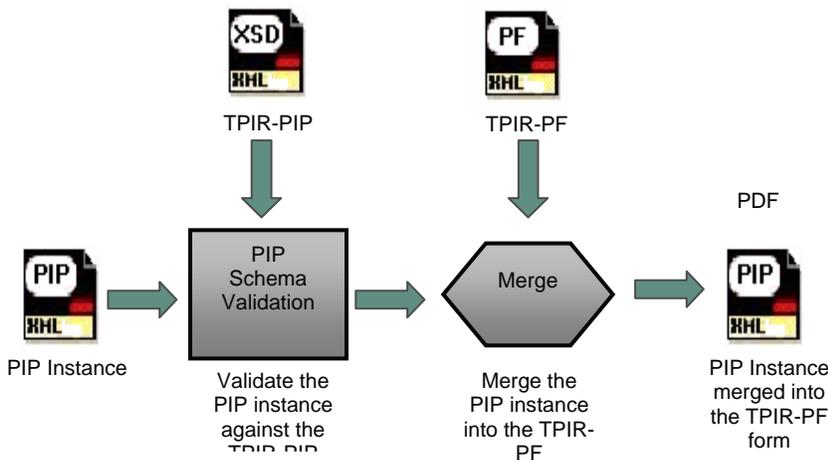
The TPIR-PF form template for a 3A4 Purchase Order Request is illustrated below.

The illustration shows the TPIR-PF form template with a 3A4 Purchase Order request instance merged into the form. Examination of the form shows the fields from the 3A4 are merged into the appropriate positions on the form.

3.3 Merging a PIP with a TPIR-PF form Template

The TPIR-PF is a form template. In a paper-based world, this is analogous to a blank paper form. It is waiting for data to be filled into the form. In this implementation, the intent is to send a purchase order request to the trading partner. This requires that the PIP instance be merged with the TPIR-PF to create a document that can be viewed by the trading partner.

The merge process is illustrated below.



Notice that an extra step was added to the typical RNIF implementation. Normally, a PIP instance is created in the private process after which the PIP is validated before it is sent to the trading partner. In this implementation, the PIP is still validated but it is also merged into the TPIR-PF form template to create the final document – a PDF. Technically, the merge can occur in one of three places

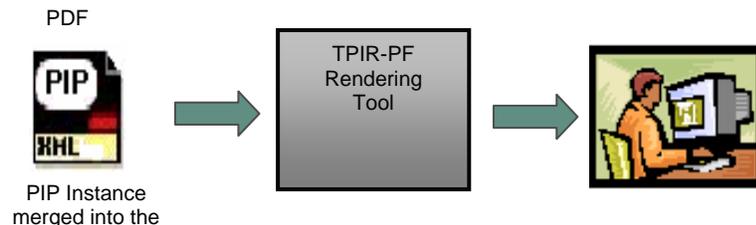
- **Originator** – the sender merges the PIP instance into the TPIR-PF to create the PDF
- **Intermediary** – the PIP instance is sent to a third party service provider that performs the merge with the TPIR-PF and then sends the PIP to the trading partner
- **Receiver** – the receiver merges the PIP instance into the TPIR-PF form template

In this implementation, an intermediary was used to merge the PIP instance into the TPIR-PF form template. The product that was used to perform the merge was the Adobe LiveCycle Server. The LiveCycle server was licensed from Adobe; alternatively, developers could build their own server. LiveCycle is also used to extract the PIP from the PDF after it has been filled in by the trading partner.

In one of the other implementations, the receiver merged the PIP instance with a TPIR-PF Form Template. This model is well suited to those trading partners that want to employ multiple TPIR-PF Form Templates for a single TPIR-PIP schema. Different TPIR-PF Form Templates may be applicable for different roles or activities to be performed by the responder as part of their private process.

3.4 Rendering

After the PIP instance has been merged with the TPIR-PF form template, the result was a PDF file. In this implementation, the originator used RNIF to send the PIP instance to the intermediary where it was merged with the TPIR-PF form. The resulting PDF was sent to the trading partner using a Web services implementation that allowed for the transport of the PDF.



In this implementation, the trading partners used Adobe Reader v7 or Adobe Acrobat v7 to view the PIP instance. In this implementation, the trading partner did not interact with the form – no data was filled into the form by the trading partner.

4 Recommendations

There are a number of alternatives that can be used for implementation of the TPIR-PF and TPIR-PIP specifications. The alternatives are defined in the TPIR-PF specification. This section describes recommendations for how to use the tools based on the experience of the individuals that participated in validation.

4.1 Forms Design

The tools selected for Forms Design was Adobe Designer version 7. The following are recommendations for the most effective use of the Adobe Designer tool.

4.1.1 General Recommendations

- Get the latest version of Adobe Designer. The product is undergoing significant changes and each successive release is more stable and has more features that make it easier to use
- Do not use an version before version 7.x.x as prior versions do not support modular schema so it is impossible to even load the complete schema into Designer
- Designer is no different from any other office desktop tool, it is unstable and will crash repeatedly, so save early and save often
- There are a number of sample documents that come with Designer under the /Samples directory under the install directory for Designer
- You can purchase Adobe Designer Professional from the distribution channel for less money than Adobe directly. The Professional package includes both Designer and Acrobat. You really need both products to develop and unit test forms development.
- There are two programming languages supported in Adobe Designer – Javascript and FormCalc. Adobe provides support for FormCalc and will not support Javascript.
- There were a significant number of issues with Javascript where it just didn't work or didn't work exactly as expected and Adobe will not provide any support, consider using FormCalc, it is pretty close in form to Javascript and it works.
- We recommend that you try building simple forms to become familiar with the tool. Adobe provides some very good examples with their software. Take a look at some of the code associated with buttons and calculated fields. In addition, several members of the RAE program are offering examples of their TPIR-PIPs and TPIR-PFs.
- Buy Acrobat's Support contract for Acrobat 7.0 Professional. It is worth it.

4.1.2 Personnel Recommendations

- Adobe Designer is a complex tool, it is strongly recommended that you get formal training on the product before attempting to use it for production. Adobe has a number of training partners that provide classes and it only takes 2-3 days
- Forms design is one of those areas where it is optimal to have a hybrid of skills – part systems analyst, part developer, part graphic design and part human usability engineer
- Join the Adobe developer program if you are going to use this tool, there are hundreds perhaps thousands of people using Designer that are sharing their tips, tricks and techniques

4.1.3 Process Recommendations

- Designer is not integrated with source control, as you develop you will need to keep track of versions of your TPIR-PF
- Treat the TPIR-PF no differently than you would for any source code development – as you develop the form you will need to save versions if you are going to be effective

- First develop the TPIR-PIP and get it perfect if you can, then and only then start designing the form
- Make a list of all the fields that need to be on the form. Then do a layout on paper of how you think the form should look visually in some simple drawing tool with each and every field. Use an existing form if you can to reduce the development time
- Design the form or better yet get an existing form and modify it accordingly
- Import the TPIR-PIP schema into the Designer then drag-and-drop the schema fields onto the fields in the form

4.1.4 Expertise Recommendations

- RosettaNet PIPs frequently use recurring structures, for example a line item on a purchase order is a recurring structure, could be one line, could be 5 lines, could be 100 lines – in Designer, subforms are used to accommodate recurring structures. The best way to get up to speed on subforms is to review existing forms
- There is a bug in Designer around the schema binding, it may be fixed after the publication of this document. When you import the TPIR-PIP into Designer you will see all of the fields, after you save, close and re-open the document, you will only see a part of the fields in the schema. Delete the Data Connection, then create the data connection again and you will see all of the fields. That's why it is recommended that you design the form first and then do the binding to the schema as the last step.
- When you import the PIP schema and bind to a field on the form, enumerated lists are not carried over. For example, a field in a schema may be restricted to a defined list of 50 states. You have to enter these manually in the field in Designer, it will not import.

4.2 Merge and Extract

The tool used in the implementation to merge the PIP and the TPIR-PF form template was Adobe LiveCycle Server. The following are recommendations for merging and extracting.

- Adobe LiveCycle provides for the merge and extract of an XML PIP instance into the TPIR-PF form template. This was the only server option used in validation.
- You can read the 2,000+ pages of Adobe technical documentation to figure out how to do the merge without purchasing LiveCycle, however, this was very time-consuming and the documentation was very obtuse. In the end, the only merge and extract server option that really worked was the LiveCycle.
- Despite repeated meetings, webinars and questions, Adobe is not very forthcoming on how to write your own merge and extract functions. Adobe wants you to purchase LiveCycle. Don't waste your time trying to get Adobe to assist you in the area of merge and extract.
- Merge and extract can be done at the client using Adobe Acrobat. Adobe Reader (the freeware) was changed in mid-cycle of validation so that merge and extract were no longer functional

4.3 Rendering

The tool used in the implementation to view PDF documents was Adobe Acrobat. Adobe Reader is free from Adobe. But Adobe has imposed limitations in what you can do with Reader using these TPIR-PF forms. For example, you cannot Export the content of the form using the standard Export function. Some of these limitations are lifted from the form if it is passed through an Adobe product called Extension Server. The process is called enabling the form – what it does is embed a code in the form that extends the capabilities of Adobe Reader automatically when Reader loads the form.

- Adobe Acrobat can be purchased from the distribution channel for less money than from Adobe directly so use the channel to buy Adobe products.
- Small trading partners were part of the validation and it was found that every one of the companies had a copy of Adobe Acrobat. None of the partners thought having to have purchased a copy of Acrobat was an issue. This is something to consider when designing a form to be used with Acrobat versus Reader.
- Forms can be edited with Acrobat, they cannot be edited and saved with Reader. You can enable the form with Reader Extension Server but this was so confusing as the capabilities of reader extension changed so often that we finally gave up. Adobe is still trying to figure out how maximize their revenue and their plans keep changing the capabilities so it is best just to stay away from it and tell your partners to use Adobe Acrobat instead.
- You can provide a digital signature on a form which can be used for non-repudiation. This worked well provided you used Acrobat and not Reader Extension Server.

5 References

Source	Description
[RFC2119]	Author: Scott Bradner Title: <i>"Key words for use in RFCs to Indicate Requirement Levels"</i> The Internet Engineering Task Force Available from: http://www.ietf.org/rfc/rfc2119.txt
[SBDH]	Title: <i>"UN/CEFACT Standard Business Document Header"</i> Revision 2.1 UN/CEFACT Available from: http://webster.disa.org/cefact-groups/atg/downloads/Generic_Header_TS_rev2.1.zip
[TPIR-PIP-REQ]	Title: <i>RosettaNet Automated Enablement, Trading Partner Implementation Requirements-Partner Interface Process™ (TPIR-PIP)</i> Available from: http://www.rosettanet.org/RosettaNet/Doc/0/PQP9UROKDV0492C7BF7A3N6VE0/RAE+TPIR-PIP+Requirement+040331Post.doc
[TPIR-PF-REQ]	Title: <i>RosettaNet Automated Enablement, Trading Partner Implementation Requirements-Presentation Format™ (TPIR-PF) Requirements</i> Available from: http://www.rosettanet.org/RosettaNet/Doc/0/PQP9UROKDV0492C7BF7A3N6VE0/RAE+TPIR-PF+Requirement+040331Post.doc
[TPIR-PF-UM]	Title: <i>RosettaNet Automated Enablement, Trading Partner Implementation Requirement -Presentation Format™ (TPIR-PF) Use Models</i> Available from: http://www.rosettanet.org/RosettaNet/Doc/0/PQP9UROKDV0492C7BF7A3N6VE0/RAE+TPIR-PF+Requirement+040331Post.doc
[PDF-REF]	Title: PDF Reference: Adobe Portable Document Format, Version 1.5, Adobe Systems Incorporated Available from: http://partners.adobe.com/asn/acrobat/docs/File_Format_Specifications/PDFReference.pdf
[TEMPLATE]	Title: Template 2.0 Specification Adobe Systems Incorporated Available from: http://partners.adobe.com/asn/tech/pdf/xfa/template_2.0.pdf
[DHS]	Title: Data Handling 2.0 Specification Adobe Systems Incorporated Available from: http://partners.adobe.com/asn/tech/pdf/xfa/data_handling_2.0.pdf
[DBS]	Title: Data Binding 2.0 Specification Adobe Systems Incorporated Available from: http://partners.adobe.com/asn/tech/pdf/xfa/data_binding_2.0.pdf
[DTHS]	Title: Data Text Handling Specification, Version 2.0 Adobe Systems Incorporated Available from: http://partners.adobe.com/asn/tech/pdf/xfa/data_text_handling_2.0.pdf

[FCS]	Title: FormCalc 2.0 Specification Adobe Systems Incorporated Available from: http://partners.adobe.com/asn/tech/pdf/xfa/formcalc_2.0.pdf
[PCS]	Title: Picture Clause 2.0 Specification Adobe Systems Incorporated Available from: http://partners.adobe.com/asn/tech/pdf/xfa/picture_clause_2.0.pdf
[SOM]	Title: Scripting Object Model 2.0 Specification Adobe Systems Incorporated Available from: http://partners.adobe.com/asn/tech/pdf/xfa/som_2.0.pdf
[XDP]	Title: XML Data Package 2.0 Specification Adobe Systems Incorporated Available from: http://partners.adobe.com/asn/tech/pdf/xfa/xdp_2.0.pdf

6 Glossary

Term	Definition
Abstract types	Allow use of complex types in such a way that a single element name can be used to represent various types in an XML document instance.
Annotation	Information for human and/or mechanical consumers. The interpretation of such information is not defined in the XML Schema specifications. The annotation element can contain one or more <documentation> or <appinfo> elements.
Attribute	A name="value" field within an XML element, providing information associated with that XML element.
Attribute Group	A set of attribute declarations, enabling re-use of the same set in several complex type definitions.
Attribute Group Definition	An attribute group definition is an association between a name and a set of attribute declarations, enabling re-use of the same set in several complex type definitions.
Built-in Datatypes	Datatypes that are defined either in the XML Schema specification (as primitive types) or in this specification, and can be either primitive or derived.
Character set	The encoding method for the data values of the document, based on Unicode format.
Complex Type	An XML element type that allows nested elements in their content and may carry attributes.
Complex Type Definition	A complex type definition is a set of attribute declarations and a content type, applicable to the attributes and children of an element information item respectively. The content type may require the children to contain neither element nor character information items (that is, to be empty), to be a string that belongs to a particular simple type or to contain a sequence of element information items that conforms to a particular model group, with or without character information items as well.
Complex type extension	Extension adds attributes, and adds elements to the end of the content model of the base type.
Complex type restriction	Restriction limits a base type to a more restrictive set of valid values.
component	Component means a basic building block of the Schema like named type, named element, named group etc.
Datatype	A datatype is a 3-tuple, consisting of 1) a set of distinct values, called its value space, 2) a set of lexical representations, called its lexical space, and 3) a set of facets that characterize properties of the value space, individual values or lexical items.
Default attribute values	Data values that imply a default value if they do not explicitly appear in the XML instance document.
Derived Data Types	Derived datatypes are those that are defined in terms of other datatypes. A datatype is said to be derived by restriction from another datatype when values for zero or more constraining facets are specified that serve to constrain its value space and/or its lexical space to a subset of those of its base type. Every datatype that is derived by restriction is defined in terms of an existing datatype, referred to as its base type . Base types can be either primitive or derived.

Electronic Document	An electronic representation of a page-oriented aggregation of the TPIR-PF form and the PIP instance that can be reproduced on screen or paper without significant loss of its information content.
Element	A fundamental unit of XML information, which has an element name, optional attributes, optional data value, and an associated type definition. Elements may be nested, one inside another.
Element Declaration	An element declaration is an association of a name with a type definition, either simple or complex, an (optional) default value and a (possibly empty) set of identity-constraint definitions.
Facet	A facet is a single defining aspect of a value space. Generally speaking, each facet characterizes a value space along independent axes or dimensions.
Fixed attribute values	An attribute value that always has the same value.
Globally defined attributes	Attribute definitions that are defined at the highest level in the XML Schema document, so that the definitions can be reused.
Globally defined elements	Element definitions that are defined at the highest level in the XML Schema document, so that the definitions can be reused.
Groups	XML Schema allows fragments of content models to be named and referenced from multiple complex types.
Main type	A reusable type that is used to define the root element of the XML instance document (PIP Action Message). In case when Schema contains only one reusable type definition than that type is by default the Schema main type.
Message Guidelines	The Message guidelines are the semantic documentation of the PIPs, which cannot be captured in Schemas.
Mixed Content	A combination of child elements and character data nested within an element.
Name	Represents names in XML. A Name is a token that begins with a letter, underscore, or colon and continues with name characters (letters, digits, and other characters). This data type is derived from token .
Namespace	An XML namespace is a collection of names identified by a URI reference, which are used in XML documents as element types and attribute names. The targetNamespace must be conformant with the URN format defined in the RosettaNet Namespace Specification and Management (NSSM).
NCName	Represents noncolonized names. This data type is the same as Name , except it cannot begin with a colon. This data type is derived from Name .
Named Types	Named types may be defined once and used many times.
Namespaces	An XML namespace is a collection of names identified by a URI reference, which are used in XML documents as element types and attribute names.
normalizedString	Represents white space normalized strings. This data type is derived from string .
PDF	Portable Document Format file as defined in the Adobe PDF Reference. PDF is one of the most recognized formats for visual representation of a document and is also a packaging format that encloses many different types and ranges of content. However, PDF can also be interactive including form controls such as text boxes, radio buttons, drop-down list boxes required for data collection. The interactivity of PDF can be enabled using the ubiquitous free Adobe Reader.
Simple Type	Simple types cannot have element content and cannot carry attributes.

Simple Type Definition	A simple type definition is a set of constraints on strings and information about the values they encode, applicable to the normalized value of an attribute information item or of an element information item with no element children. Informally, it applies to the values of attributes and the text-only content of elements.
Subassemblies	A unit of content in an XDP is referred to as a subassembly. For example, an XDP representing an interactive form instance would have a PIP form data subassembly, a TPIR-PF Form Template subassembly, a PDF subassembly and other optional subassemblies.
Substitution groups	An element can be declared to be a substitute for another element, the "head" element, allowing the new element to appear anywhere the head element may appear.
targetNamespace	The namespace of an instance document.
token	Represents tokenized strings. This data type is derived from normalizedString .
TPIR-PF	The TPIR-PF Form Template consists of the metadata or instructions on how to interpret and visually render a PIP instance that conforms to a TPIR-PIP schema for human interaction. The TPIR-PF Form Template provides a description of how each and every field described by a TPIR-PIP schema will be visually presented to the user including labels, controls, physical position, fonts, and other graphical and interactive attributes.
TPIR-PIP	An identified set of restrictions and requirements to which a PIP instance must comply. A TPIR-PIP is a subset of the Standard RosettaNet Schema. A TPIR-PIP only includes the necessary schema components required in a B2B integration while other components from the RosettaNet Standard Schemas might be removed.
Type Derivation	XML Schema allows a type to be derived from another type (its base type), either by extension or restriction.
Type Redefinition	XML Schema allows a Schema author to redefine the types or groups of another Schema document.
Type Substitution	Allows a base type to be substituted by any derived type.
PIP Umbrella Version	The PIP version (e.g., R11.01) of the whole PIP Package.
Union types	The union operation is supported by XML Schema for element types. For example, a codelist may be defined as the union of two other codelists.
User-derived Datatypes	User-derived datatypes are those derived datatypes that are defined by individual Schema designers.
Value Space	A value space is the set of values for a given datatype. Each value in the value space of a datatype is denoted by one or more literals in its lexical space.
XDP	The XDP format provides an alternate means of expressing a PDF document in a manner where the outer packaging is described in an XML-based syntax rather than a PDF-based syntax. The XML Data Package (XDP) provides a mechanism for packaging units of content called subassemblies within a surrounding XML container that consists of the PIP XML instance, the TPIR-PF XML Form Template, and a base64-encoded PDF.
XML Schema	An XML document that defines the allowable content of a class of XML documents. A class of documents refers to all possible permutations of structure in documents that will still confirm to the rules of the Schema.

XSD	Extensible Stylesheet Language for Transformations is an official recommendation of the W3C. It provides a language for transforming XML documents into something else including an HTML document, another XML document, a Portable Document Format (PDF), Java Code, a flat text file or most any other format.
XSLT	Refers to the XML Schema Definition language.

7 Apendix A

7.1 XML Data Package (XDP) Rules

This section summarizes the XDP rules. For complete information regarding particular aspect refer to the appropriate section.

Rule 3-1	The XDP format MUST be comprised of only a single element, known as the XDP element.
Rule 3-2	The XDP element MUST enclose one occurrence of content, each known as an XDP packet.
Rule 3-3	The XDP MUST be comprised of three subassemblies within the XDP element. The subassemblies include the PIP form data, the TPIR-PF Form Template and the Portable Document Format (PDF).
Rule 3-4	The XDP MAY contain additional subassemblies but they will be disregarded.
Rule 3-5	The XDP element SHOULD make use of explicitly prefixed namespace notation rather than declaring the XDP namespace as a default namespace. If the XDP element declared the XDP namespace as the default namespace it would have the unfortunate side effect of placing any content that lacks namespace information into the XDP namespace itself.
Rule 3-6	All child elements of the XDP element are considered to be XDP packets. Conversely, an XDP packet MUST be located as a child element of the XDP element.
Rule 3-7	An XDP packet MUST not belong to the XDP namespace. The application of the XDP namespace on child elements of the XDP element is reserved for future use.
Rule 3-8	The PIP form data packet MUST enclose XML data content that is a RosettaNet PIP that conforms to a TPIR-PIP XML schema. This PIP data MAY have originated from an Electronic Document form and/or may be intended to be consumed by an Electronic Document form.
Rule 3-9	The Form Template packet MUST enclose the definition of a TPIR-PF Form Template.
Rule 3-10	The PDF packet MUST enclose the definition of a PDF document that represents the combination of the PIP form data and the TPIR-PF Form Template.
Rule 3-11	XML is a text format, and is not designed to host binary content. PDF files are binary and therefore MUST be encoded into a text format before they can be enclosed within an XML format such as XDP. The most common method for encoding binary resources into a text format, and the method used by the PDF packet, is base64-encoding [RFC2045].
Rule 3-12	The <chunk> element MUST enclose a base64-encoded PDF document.
Rule 3-13	The XDP MAY enclose zero to one XSLT packets. Refer to the W3C XSL Transformations specification [XSLT] for further information on how to configure XSLT processing.

7.2 TPIR-PF Form Template Rules

This section summarizes the TPIR-PF Form Template rules. For complete information regarding particular aspect refer to the appropriate section.

Rule 4-1	The Adobe Template 2.0 Specification MUST be used for defining a Trading Partner Interface Requirement – Presentation Format (TPIR-PF).
Rule 4-2	The TPIR-PF Form Template MUST be designed with a TPIR-PIP XML schema.
Rule 4-3	The TPIR-PF Form Template MUST use all of fields in the TPIR-PIP schema tree and place them on the template canvas. The Design Tool will create subform/field structure, with properly-typed fields, and template-defined bindings.
Rule 4-4	The TPIR-PF Form Template MUST be designed such that the constraints defined in the TPIR-PIP XML schema are reflected in the TPIR-PF Form Template. These constraints will validate the data entered into the Electronic Document.
Rule 4-5	The TPIR-PF Form Template MUST conform to the following Adobe specifications: <ul style="list-style-type: none"> • Template 2.0 Specification, Adobe Systems Incorporated • Data Handling 2.0 Specification, Adobe Systems Incorporated • Data Binding 2.0 Specification, Adobe Systems Incorporated • Data Text Handling Specification, Version 2.0, Adobe Systems Incorporated • FormCalc 2.0 Specification, Adobe Systems Incorporated • Picture Clause 2.0 Specification, Adobe Systems Incorporated • Scripting Object Model 2.0 Specification, Adobe Systems Incorporated

7.3 PDF Rules

This section summarizes the PDF rules. For complete information regarding particular aspect refer to the appropriate section.

Rule 5-1	The Portable Document Format (PDF) MUST conform to the following Adobe specification other than the exceptions noted below. PDF Reference: Adobe Portable Document Format, Version 1.5 Adobe Systems Incorporated, Available at http://partners.adobe.com/asn/acrobat/docs/File_Format_Specifications/PDFReference.pdf .
Rule 5-2	The Portable Document Format MUST be comprised of two segments within the binary file. The segments subassemblies include the PIP form data and the TPIR-PF Form Template.
Rule 5-3	The keywords Encrypt SHOULD NOT be used in the trailer dictionary. Encryption of the service content will be facilitated as defined in the RosettaNet Implementation Framework.
Rule 5-4	The document information dictionary MAY be optional. The preferred method for supplying descriptive document information is in the PIP headers.
Rule 5-5	The LZWDecode filters MUST NOT be permitted. The use of the LZW decompression algorithm is subject to intellectual property constraints.
Rule 5-6	A stream object dictionary SHOULD NOT contain the F , FFilter , or FDcodeParams keys. The use of these keys would permit the existence of document content external to the file.
Rule 5-7	The document catalog dictionary MAY not contain the OCProperties key. This key is defined in PDF 1.5 for the use of optional content that can be used to generate alternative renderings of a document.

8 Appendix B

There are two TPIR-PF implementation models that conform to the current RosettaNet Implementation Framework (RNIF) 2.0 core specification as outlined in *Section 2* of this document. There are two additional models that are included in this Appendix for reference. The two additional models do not technically conform to the RNIF specification; both require a technical advisory or a change in the RNIF 2.0 specification.

Both implementation models transmit only the electronic document, either an XML Data Package (XDP) or a Portable Document Format (PDF), in the Payload, not the Partner Interface Process (PIP) instance. Current B2B gateways that conform to RNIF 2.0 cannot validate the PIP instance without modification. *Section 1.2.5.2* of the *RNIF 2.0 Specification* states that for action messages, RNIF 2.0 provides the option of shipping business action messages in a third-party defined format. The RNIF 2.0 Service Header includes additional fields that identify the standard body and the version of the specification to which the action message conforms. These messages must still be exchanged in a RosettaNet-defined PIP and must be sanctioned by RosettaNet by explicit identification of the sanctioned third-party action messages in the PIP specification.

8.1 XDP Service Content

In the XDP service content model, the B2B message consists of an XML Data Package that contains the PIP instance, the TPIR-PF Form Template and a base64-encoded PDF. The XDP may include other components as well. The RNIF B2B Server would perform transportation but only the headers would undergo validation not the PIP instance data. The XDP service content would look like the following:

```
<xfa:datasets xmlns:xfa="http://www.xfa.org/schema/xfa-data/1.0/">
  <xfa:data>
    ...PIP XML form data content...
  </xfa:data>
</xfa:datasets>

<xfa:template xmlns:xfa="http://www.xfa.org/schema/xfa-template/2.0/">
  ...TPIR-PF XML Form Template content...
</xfa:template>

<pdf xmlns="http://ns.adobe.com/xdp/pdf/">
  <document>
    <chunk>
      ...base64 encoded PDF content...
    </chunk>
  </document>
</pdf>
```

Figure 11-1

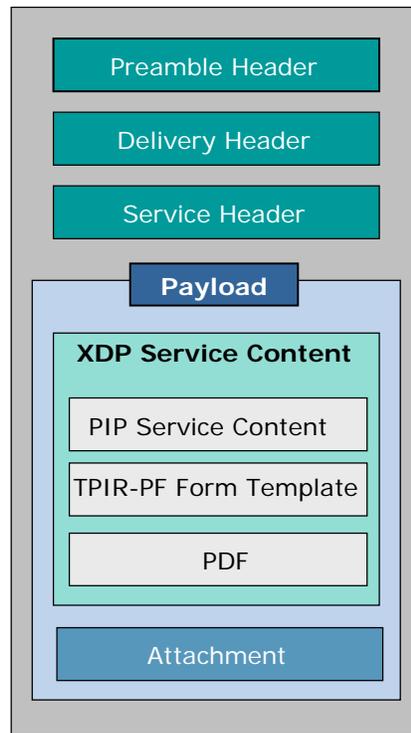


Figure 11-2: XDP Service Content

The responder would visually render the XDP using a Rendering Tool and the user would interact with the Electronic Document. If the Partner Interface Process (PIP) requires a return message signal, the user would interact with the controls as required and both the PIP data and the PDF in the XML Data Package would be updated. When the XDP is returned, the initiator would extract the PIP from the XDP.

In this model, the PIP instance is not a part of the Payload, hence, it is not validated. A change would be required to existing RNIF-compliant software products to validate the form data segment (the PIP instance) of the XDP, which is in clear text. To do this properly, the Service Header would also have to be more specific in that the Payload would be declared as XDP.

8.2 PDF Service Content

In the PDF Service Content model, the B2B message consists of a Portable Document Format (PDF) that has been base64-encoded. The PDF is generated by merging a PIP instance, the form data, with the TPIR-PF Form Template. The RNIF-compliant software would perform transportation but only the headers would undergo validation not the PIP instance data.

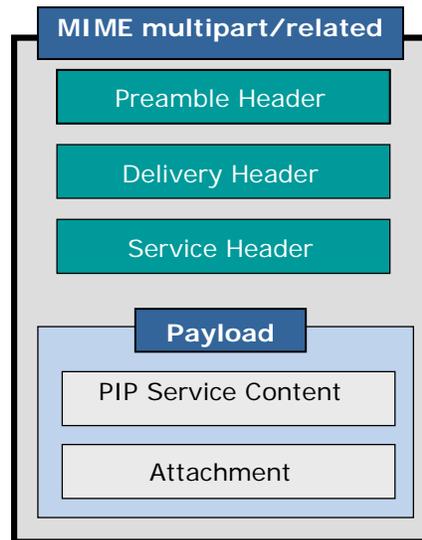


Figure 11-3: PDF Service Content

The responder would visually render the PDF using a Rendering Tool and the user would interact with the Electronic Document. If the Partner Interface Process requires a return message signal, the user would interact with the controls on the PDF form. The form data that conforms to the TPIR-PIP schema is updated inside the PDF.

When the PDF is returned, the initiator would first decode the base64-encoded service content and then extract the PIP form data from the PDF binary. The size of the message being transferred using a PDF is smaller than other implementation options. PDF is a binary format that leverages different compression schemes that result in a much smaller file.

Files included in the B2B message as attachments could be included in the PDF structure as additional subassemblies. The advantage of including file attachments in the PDF rather than the Payload is that existing Rendering Tools are capable of visually presenting the attachments. The responder could not only interact with the PIP via the TPIR-PF Form Template but also view the attachments at the same time in the same tool.

In this model, the PIP instance is not a part of the payload, hence, it is not validated. A change would be required to existing RNIF B2B Servers to validate the form data segment (the PIP instance) of the PDF, which would have to be extracted from the PDF binary. To do this properly, the service header would also have to be more specific in that the payload would be declared as PDF.

9 Appendix C

There are a number of files included with this document to assist the user in developing their own implementation of the RAE specifications. Specifically, the TPIR-PF specifications.

9.1 PIP3A4_V11.00.00_047897855_PO_01.01_RequestPurchaseOrder.zip

The ZIP package contains one of the TPIR-PIPs used in validation. The TPIR-PIP is a constrained version of the 3A4r Request Purchase Order community schema. The TPIR-PIP reflects the specific implementation requirements of the trading partners.

When designing a TPIR-PF, the TPIR-PIP was loaded into the design tool so that a binding could be established between fields on the form and fields in the PIP. Data entered by a user into the TPIR-PF form will be placed in the correct position as defined by the TPIR-PIP schema. If the TPIR-PIP is merged with a PIP instance, the binding tells where the field will be displayed on the TPIR-PF form.

9.2 IntelPurchaseOrder_v37.pdf

The v37 document was used by the receiver. In this scenario in validation, the originator sent a PIP to the receiver. The receiver opens the v37 document and used the import button to load the PIP instance into the TPIR-PF.

9.3 PurchaseOrder_V11.00.pdf

The v110.00 TPIR-PF is a presentation form template that is design for machine processing. A PIP instance was generated by the originator, then a machine process merged the PIP instance the v11.00 TPIR-PF to create an electronic document. The electronic document was viewed by the receiver.

9.4 POsample.xml

The POsample is an XML file that can be used with either of the TPIR-PFs described above.