

How To Use a Single E-mail for NF-e & Cancellation Messages in SAP GRC NFE Inbound B2B Interface



Applies to:

SAP GRC NFE 1.0, SAP Process Integration 7.0 / 7.1, For more information, visit the [Governance, Risk, and Compliance homepage](#).

Summary

By default, the inbound B2B interfaces of the SAP GRC NFE solution require each one a separate configuration scenario in Integration Directory of SAP PI, making it necessary to use separate communication channels for each message type, e.g. use different e-mail accounts for receiving NF-e and Cancellation messages. This article explains how to enable the solution to work with a single inbound interface for both types of messages, through some few modifications on SAP PI objects & configuration, making it possible to use a single e-mail to receive both the incoming NF-e and Cancellation XML messages.

Author: Henrique Pinto

Company: SAP Brazil

Created on: 10 June 2009

Author Bio



Henrique Pinto works at SAP Brazil, focusing on integration related solutions and development link for Latin-American specific products, including product management for the SAP GRC NFE solution. He is at SAP since Feb/07.

Table of Contents

Introduction	3
Preparation	3
SLD Configuration.....	3
Implementation	5
Integration Repository.....	5
Interface Mapping	5
Integration Directory.....	6
Interface Determination.....	6
Receiver Agreement	7
Testing	8
Further Considerations	9
Multiple Attachments.....	9
Error message back to the Sender	9
Related Content.....	11
Copyright.....	12

Introduction

Besides the government communication interfaces, the SAP GRC NFE solution includes several interfaces for the exchange of XML documents between the company and its Business Partners (suppliers and customers).

The set of B2B interfaces includes both Outbound and Inbound interfaces. The Outbound B2B interfaces are triggered after the government authorization process and are meant to send the NF-e XML messages to the invoice receivers (e.g. customers). The Inbound B2B interfaces, on the other hand, are the ones used in the other branch of the business process, when the company receives the NF-e XML messages from the invoice sender (e.g. suppliers).

For the receiving process, there are two Inbound B2B scenarios: one for receiving NF-e XML messages (NFB2B) and another one to receive Cancellation XML messages (CFB2B). This means that it is necessary to configure two inbound scenarios, one for each possible inbound interface.

However, there are some adapters which require one communication channel for each interface that is being processed, for example the Sender Mail adapter. In this case, it would be necessary to keep two separate communication channels to be able process these two interfaces. And since each communication channel would pool just one type of XML message (interface), it would be necessary to maintain two e-mail accounts: one to receive just NF-e XML messages and another to receive just cancellation XML messages. This would increase the maintenance efforts as well as the complexity of the integration between the company and its suppliers and customers (in case the customer is sending a return NF-e).

This article presents a simple alternative to avoid the usage of duplicated communication channels, by realizing simple customizations in the inbound interfaces. This procedure doesn't change the standard objects anyhow, so there is no impact to maintenance coverage or during the application of Support Packages.

Preparation

SLD Configuration

In the SLD, it is necessary to create a Software Component Version (SWCV) that will include our custom developments. However, in order to make our job a lot easier and avoid redefining standard objects which we'll not change, it is possible to add the standard SLL-NFE SWCV as a dependency of our customer SWCV. This way, we'll be able to refer to all objects in the standard SWCV in our custom objects.

In order to add a dependency, go to the SLD home page and click on Software Components (under the Software Catalog section). Browse your custom SWCV and, on the Dependencies tab, search and include the standard SLL-NFE SWCV, as pictured below. The Dependency Context is not really important for us, so you can leave it in the default option (Installation Time dependency).

Software Catalog

Software Type: Filter:

Software Component	Version
NFE of sap	NFE , 1.0 of sap
SAP SLL-NFE	SAP SLL-NFE 1.0
SAP SLL-NFE-JWS	SAP SLL-NFE-JWS 1.0
XI CONTENT SLL-NFE	XI CONTENT SLL-NFE 1.0
ZNFE_TEST of sdn.sap.com	ZNFE_TEST , 1.0 of sdn.sap.com

Linha 1 de 5

ZNFE_TEST , 1.0 of sdn.sap.com

General | **Products** | Dependencies | Support Packages | Release Compatibility | Release History

Context: ⓘ

Prerequisite Software Component Versions	Dependent S
SAP SLL-NFE 1.0	

Note: Make sure that your SLD CR Content Version is at least 3.10, or else you won't be able to see the SAP NFE Product and Software Components in the SLD Software Catalog. In order to check that, go to the *SLD home page* -> *Administration* -> *Details* -> *Data* tab and check the SAP CR Content Version value.

Server | **Data** | Object Manager | Browser | Data Suppliers

Model Version (sld/active): 1.5.32

SAP CR Content Version (sld/active): SAP_CR_INTERNAL 4.209 (produced 11/19/2008)

Namespace	Model Version	Classes	Instances	Associations	Total Objects
sld/active	1.5.32	960	127713	291696	420369

Finally, it is also necessary to create the Business Systems in SLD that will be part of the integration process. At least, you need the Business System of the GRC NFE System (as described in the NFE configuration guide).

Implementation

The idea behind this procedure is to find a way to process different XML messages which are received through the same e-mail account (and, consequently, are related to the same outbound Message Interface).

The main consideration to be done here is that, by default, PI does not realize any payload validation in the incoming message. The only scenario where there is a payload dependency with the message processing is when there is content based routing which requires a specific field to exist in the payload.

However, it is possible to define content-based routing which in a “generic” way that makes it possible to process different XML messages through the same processing pipeline.

The sole requirement that makes this scenario possible and that we need to consider is that the interface mapping, even though it always has the same outbound interface (the NF-e one), may have two possible outputs: the NF-e or the Cancellation inbound interfaces. Since there is no interface mapping between the Incoming B2B NFE outbound interface and the Incoming B2B Cancellation inbound interface, we need to make a simple modification on the standard interface mapping for the cancellation scenario.

The details are described below.

Integration Repository

Since we've added the standard SLL-NFE SWCV as a dependency of our customer SWCV, we can create modified versions of the standard objects without actually modifying the standard SWCV. For more details, check the *Modifying Objects* reference below.

Once you've added the dependency, a new *Basis Objects* node should appear in your custom SWCV, right below your namespaces & Imported Objects. Within this new node, you can find all the standard objects in the SLL-NFE SWCV, however here they're included in your custom SWCV. It's more or less like if the system had automatically created Z versions of all standard objects within your custom SWCV.

The only change that needs to be performed in the Integration Repository, as explained above, is to modify the Incoming B2B Cancellation Outbound interface.

Interface Mapping

Go to the CFB2B_enhProcCancNFe_TO_enhProcCancNFe interface mapping in the Basis Objects node. Enter into *Edit* mode and click on *Modify*. In the *Source Interface* list, change the existing default interface (CFB2B_procCancNFe_OB) for the NFB2B_procNFe_OB message interface. You can leave the SWCV as the default one (which should be your custom SWCV).

Save your interface mapping, click on *Modify* again and activate it. It should look something like this:

The screenshot displays the SAP Integration Repository interface. On the left, a tree view shows the 'Basis Objects' node expanded to 'Interface Mappings', with 'CFB2B_enhProcCancNFe_TO_enhProcCancNFe' selected and highlighted with a red box. The main area shows the 'Display Interface Mapping' dialog for this mapping. The 'Name' field is 'CFB2B_enhProcCancNFe_TO_enhProcCancNFe', the 'Namespace' is 'http://sap.com/xi/NFE/005a', and the 'Software Component Version' is 'ZNFE_TEST , 1.0 of sdn.sap.com', which is highlighted with a red box. Below this, the 'Source Interface' list is visible, with 'NFB2B_procNFe_OB' selected and highlighted with a red box. The 'Name' field of this source interface is 'NFB2B_procNFe_OB' and the 'Software Component Version' is 'http://sapZNFE_Tt1'.

Integration Directory

In the Integration Directory, transfer the NFB2B_WebAS_Inbound_B2B_NFe Integration Scenario into a Configuration Scenario, as you would do normally. This scenario doesn't need to be changed anyhow. The Communication Channel may also be configured normally as it would be for a simple Incoming NFE interface.

After the configuration for the Incoming B2B NFE scenario is complete, it is just necessary to include the routing to the Cancellation inbound interface; we'll be using the same configuration scenario for that, by just modifying some objects, as described below.

Interface Determination

Now that the Interface Mapping is modified, it is possible to select it in the Interface Determination for the NFE scenario. However, the idea is not to replace the standard receiver interface & interface mapping, but rather to include another receiver interface (and respective interface mapping) for the cancellation scenario, and to define conditions that will route the message either to one or the other receiver interface.

Open the Interface Determination object for the Incoming B2B NFE scenario. It should include one entry in the *Configured Inbound Interfaces* list: NFB2B_enhProcNFe_IB (with associated Interface Mapping NFB2B_procNFe_TO_enhProcNFe). Add another line in the list and input the following content:

```
Inbound Interface Name:      CFB2B_enhProcCancNFe_IB
Inbound Interface Namespace: http://sap.com/xi/NFE/005a
```

Click on the Input Help button for the Interface Mapping data for this new line. It should bring the Interface Mapping that we had modified above:

```
Interface Mapping Name:      CFB2B_enhProcCancNFe_TO_enhProcCancNFe
Interface Mapping Namespace: http://sap.com/xi/NFE/005a
Interface Mapping SWCV:      <your custom SWCV>;
```

After you have included the receiver interface & interface mapping for the cancellation inbound interface, it is necessary to maintain the routing conditions that will make the attachment from the e-mail flow through one or the other receiver interface.

In the *Condition* column, click on the Input Help button for the first row (the default one, related to the NFB2B_enhProcNFe_IB receiver interface) to open the *Condition Editor* screen. Click on the Input Help button for the Left Operand first entry to open the *Expression Editor* screen.

The condition that we are going to use cannot be dependent on particular fields of the payload (given that the payload can be different, since we're working with different messages in the same communication channel). Select the *XPath* radio button (if not selected by default) and enter the following expression in the *XPath Expression* field:

```
/p1:nfeProc
```

Click on *Ok*. Back to the *Condition Editor* screen, select the "EX" Operator (*Op.* column) from the dropdown list. Click on *Ok*. Repeat the same procedure to the second row of the list (the one related to the CFB2B_enhProcCancNFe_IB receiver interface), but this time enter the following expression in the *XPath Expression* field:

```
/p1:procCancNFe
```

After you've finished, save and activate the object. It should look like this:

Configured Inbound Interfaces			
Inbound Interface			
	Name	Namespace	Condition
1	NFB2B_enhProcNFe_IB	http://sap.com/xi/NFE/005a	(/p1:nfeProc EX)
2	CFB2B_enhProcCancNFe_	http://sap.com/xi/NFE/005a	(/p1:procCancNFe EX)

Interface Mapping			
	Name	Namespace	Software Component Version
	NFB2B_procNFe_TO_enhProcNI	http://sap.com/xi/NFE/005a	SAP SLL-NFE 1.0
	CFB2B_enhProcCancNFe_TO_e	http://sap.com/xi/NFE/005a	ZNFE_TEST , 1.0 of sdr

Tip: if you can't see the *Condition* column, add a third line and then remove it. This should make the column to be displayed.

Receiver Agreement

After manually including a new receiver interface to the Interface Determination, we need to manually create a Receiver Agreement related to that interface.

In order to save time, we can base the new Receiver Agreement creation on the data of the already existing one (for the NFB2B_enhProcNFe_IB interface). Right-click on it and select *Copy Object*. In the *Copy Receiver Agreement* screen, change the interface for the Inbound Interface we have added to the Interface Determination (CFB2B_enhProcCancNFe_TO_enhProcCancNFe). In the *Add To Scenario* field, choose the configuration scenario that you have created for the Inbound B2B NFE process. Click on *Copy*.

Activate all your changes. The new receiver agreement should look like this:

Display Receiver Agreement	
Sender	
Party	NFE_B2B_Party
Service	SendService
Receiver	
Party	
Service	PSQ_305
Interface	CFB2B_enhProcCancNFe_IB
Namespace	http://sap.com/xi/NFE/005a
Description	
Receiver Communication Channel *	WAS_XI_RCV

Testing

In order to test the scenario, it would be possible to send two emails to the same account (configured in the Sender Communication Channel). However, to make it easier, we will use the *Test Message* tab of the *Integration Engine* component, in the Runtime Workbench.

Just insert the information to test the NFE Inbound scenario, as shown below, and paste the content of the <nfeProc> XML message.

Tip: make sure to fill the Receiver Party & Service (the Party and Business Service that is used in the B2B communication, not the internal receiver Business System for A2A).

Fill the password and send the message. It'll simulate the sending from a communication channel and should be updated to the Incoming NFE Monitor of SAP NFE.

Resultado

Visão [Visão standard] | Versão de impressão | Exportação | Download XML

Chave de acesso	CNPJ do Emissor	CNPJ Recebedor	Região	Tipo NF-e
43090474544297000435550010000008260364412720	74544297000435	12345678000195	43	NF-e

To test the Cancellation, just change the payload of the test message above for the <procCancNFe> XML message which would be sent from a supplier. Even with the same input data (sender party, service & interface), due to the conditional routing, it'll automatically be pointed to the correct receiver interface and mapping, and should also appear ok in the Incoming NFE monitor.

Resultado

Visão [Visão standard] | Versão de impressão | Exportação | Download XML

Chave de acesso	CNPJ do Emissor	CNPJ Recebedor	Região	Tipo NF-e
43090474544297000435550010000008260364412720	74544297000435	12345678000195	43	NF-e
43090474544297000435550010000008260364412720	74544297000435	12345678000195	43	Cancelar NF-e

Further Considerations

Besides the standard functionalities of feeding the Incoming NFe table (which is shown in the Incoming NFe Monitor) and storing the XML files in Knowledge Provider (KPro), it is sometimes desirable to have extra functionalities in the Incoming NFe scenario.

Below, there are some simple suggestions to further enhance this scenario with SAP PI capabilities.

Multiple Attachments

By default, the XI Mail Receiver adapter will work with just one attachment as the main payload. Hence, when feeding the Incoming NFe interfaces, just one of the attachments would be transferred to GRC NFE.

This is not a big problem, since it is possible to request your suppliers to send one attachment per e-mail (just like SAP NFE works in the outbound side). However, it may be desirable to have multiple attachments when the supplier does this manually or in order to reduce the number of internet communications. In order to implement this with GRC NFE, it is possible to make use of a simple Mail-PI-File scenario prior to the actual Incoming NFe scenario.

First, you would configure a simple Mail to File scenario, selecting the *Keep Attachments* flag in the Sender Mail communication channel. In the Receiver File channel, just configure a local folder where the Mail attachments (supposedly, all of them must be valid NFe or Cancellation XML messages) will be saved to. You could create a simple Business Service to represent the System where the file folder will be placed.

In the standard Incoming NFe scenario, configure the sender adapter as a Sender File communication channel. This channel would be the one that goes in the *Transfer Integration Scenario* wizard. The sender system in this case would be the Business Service mentioned above, not the actual B2B Party.

With this simple set up, it is possible to persist all Mail attachments and read them, 1 by 1, into the incoming NFe scenario.

Error message back to the Sender

Sometimes, especially when the suppliers that send you the XML messages do it manually, it is possible to receive several garbage messages with wrongly attached or wrongly encoded files or even messages without any attachments at all. For such cases, it would be desirable to send error messages back to the mail sender informing about the errors.

This would require two things:

- I. the persistence of the Sender mail ("From" field); this is done by marking both the *Set Adapter-Specific Message Attributes* and *Variable Transport Binding* flags in the Sender Mail communication channel.
- II. an orchestrated integration scenario, otherwise known as BPM (or cc-BPM, from cross component Business Process Management), in PI. The idea is to create such a BPM that would be executed prior to the Incoming NFe scenario. The message flow would be:
 1. From B2B Party (Mail Sender adapter) to BPM
 2. From BPM, it would be possible to have two outputs:
 - 2.1. Error message back to Mail Sender
 - 2.2. Normal NFe (or Cancellation) message to GRC NFE.

It would be possible to use one single BPM to both NFe and Cancellation messages, using some decision logic in the mapping and generic interfaces (e.g. a message with the XML as a string and simple identifiers, such as the NFe Access Key (NFe Id) and a field to identify whether it is an NFe or Cancellation Message). It would be easier to implement to use two separate BPMs, though.

Within the BPM itself, it could have the following design:

1. Receiver Step, that would receive the incoming message;
2. Block step, within which you define an exception (related to the Invalid Mail Content); then create an exception branch and associate it to this exception. The other steps would be like this

2.1. Normal branch:

- 2.1.1. Transformation step, which would include a Java Mapping that would map the incoming message to the generic interface (a Java Mapping is necessary in order to map “anything” (NFe or Cancellation XML) to the <xml> string field; a check for empty or invalid payload can be implemented within the mapping, which would forcedly thrown an exception). Associate any exception of this Transformation Step with the Exception defined within the Block step;
- 2.1.2. Send step, which would send the generic message described above to the possible receivers (the interface determination (and related mappings, could be done externally to the BPM, as described throughout this article);

2.2. Exception branch:

- 2.2.1. Transformation Step, which would include a mapping (Message or Java Mapping) that would read the Sender Mail information (From field) from the Adapter Specific Message Attribute with Technical Name *SHeaderFROM* and namespace *http://sap.com/xi/XI/System/Mail* (check the *Configuring the Sender Mail Adapter* reference below for more details);
- 2.2.2. Send Step, which would send the error message (it could even include some error detail) back to the original mail sender.

With these developments, and by configuring it properly in Integration Directory, it would be possible to have such kind of persistence and failsafe mechanism.

Related Content

[Modifying an Object](#)

[Configuring the Sender Mail Adapter](#)

[SAP NFE 1.0 Weblog](#)

[Adapter-Specific Message Attributes in the Message Header](#)

[Accessing Adapter-Specific Attributes](#)

[Using SAP PI Lookup API and Dynamic Configuration in SAP GRC NFE Outbound B2B Interface for Dynamic E-mail Determination](#)

For more information, visit the [Governance, Risk, and Compliance homepage](#).

Copyright

© 2009 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, OS/2, Parallel Sysplex, MVS/ESA, AIX, S/390, AS/400, OS/390, OS/400, iSeries, pSeries, xSeries, zSeries, System i, System i5, System p, System p5, System x, System z, System z9, z/OS, AFP, Intelligent Miner, WebSphere, Netfinity, Tivoli, Informix, i5/OS, POWER, POWER5, POWER5+, OpenPower and PowerPC are trademarks or registered trademarks of IBM Corporation.

Adobe, the Adobe logo, Acrobat, PostScript, and Reader are either trademarks or registered trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Sun Microsystems, Inc.

JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

MaxDB is a trademark of MySQL AB, Sweden.

SAP, R/3, mySAP, mySAP.com, xApps, xApp, SAP NetWeaver, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

These materials are provided "as is" without a warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement.

SAP shall not be liable for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials.

SAP does not warrant the accuracy or completeness of the information, text, graphics, links or other items contained within these materials. SAP has no control over the information that you may access through the use of hot links contained in these materials and does not endorse your use of third party web pages nor provide any warranty whatsoever relating to third party web pages.

Any software coding and/or code lines/strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, except if such damages were caused by SAP intentionally or grossly negligent.