



## SAP Transportation Management Output Management Configuration Guide

CUSTOMER  
Document Version: 3.2 – December 2013

## Copyright

© Copyright 2013 SAP AG. All rights reserved.

SAP Library document classification: PUBLIC

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. National product specifications may vary.

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.

SAP 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 other countries.

Please see <http://www.sap.com/corporate-en/legal/copyright/index.epx#trademark> for additional trademark information and notices.

## Icons in Body Text

Icon	Meaning
	Caution
	Example
	Note
	Recommendation
	Syntax

Additional icons are used in SAP Library documentation to help you identify different types of information at a glance. For more information, see *Help on Help → General Information Classes and Information Classes for Business Information Warehouse* on the first page of any version of *SAP Library*.

## Typographic Conventions

Type Style	Description
<i>Example text</i>	Words or characters quoted from the screen. These include field names, screen titles, pushbuttons labels, menu names, menu paths, and menu options. Cross-references to other documentation.
<b>Example text</b> EXAMPLE TEXT	Emphasized words or phrases in body text, graphic titles, and table titles. Technical names of system objects. These include report names, program names, transaction codes, table names, and key concepts of a programming language when they are surrounded by body text, for example, SELECT and INCLUDE.
Example text	Output on the screen. This includes file and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools.
<b>Example text</b>	Exact user entry. These are words or characters that you enter in the system exactly as they appear in the documentation.
<Example text>	Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system.
EXAMPLE TEXT	Keys on the keyboard, for example, F2 or ENTER.

## Table of Contents

Output Management Configuration Guide.....	5
1 Definitions .....	6
2 Output Management Overview .....	8
3 Features of Output Management .....	13
3.1 Immediate Print .....	14
3.2 Automated Output .....	15
3.3 Originals and Copies .....	16
3.4 Mass Output .....	17
3.5 Features of the Output Management UI.....	18
4 Output Management Settings and Configuration.....	23
4.1 Maintaining PPF Settings .....	24
4.1.1 Defining Applications .....	25
4.1.2 Defining Action Profiles.....	26
4.1.3 Creating Action Definitions .....	27
4.1.4 Defining Processing Types .....	28
4.2 Creating Conditions .....	29
4.3 Output Management Adapter Configuration .....	30
4.3.1 Customizing .....	31
4.3.2 PPF Exits .....	36
4.3.3 Output Management Callbacks .....	38
4.3.3.1 Action Profile Determination .....	40
4.3.3.2 Action Determinations and Schedule Condition Checks .....	41
4.3.3.3 Start Condition Check .....	42
4.3.3.4 Determine Printer .....	43
4.3.3.5 Determine Document Partners .....	44
4.3.3.6 Determine Partner Language.....	45
4.3.3.7 Specify Output Content for PDF Forms .....	46
4.3.3.8 Specify Application Keys for Transient Action Definitions .....	47
4.3.3.9 Specify Application Keys for Aggregate Action Definitions .....	48
4.3.4 BRFplus Applications and Functions .....	51
4.4 BO Modifications for Integration with Output Management Adapter .....	52
4.5 Configuring Direct Output.....	53
4.6 Configuring the Output Management User Interface .....	54
4.7 General Recommendations and Guidelines .....	55
5 Testing Output Management in Your Application .....	57
6 Appendix .....	59
6.1 Basic Introduction to the PPF.....	60
6.1.1 Configuring Different Output Requirements .....	61
6.1.1.1 Processing Type (Medium Type) .....	62
6.1.1.1.1 Setting Up Processing Types for External Communication .....	63
6.1.1.1.2 Setting Up Processing Types for Alerts .....	64
6.1.1.1.3 Setting Up Processing Types for Method Calls.....	65
6.1.1.2 Action Definition (Trigger Type) .....	66
6.1.1.3 PPF Action Profile (Context Type) .....	67
6.1.1.4 PPF Application.....	68
6.1.2 PPF Runtime .....	69
6.1.2.1 Generating Triggers (Actions) .....	70
6.1.2.2 Processing Triggers (Actions).....	71
6.2 Design .....	72
6.3 TM-Specific Guidelines .....	73



## **Output Management Configuration Guide**



# 1 Definitions

The following terms are used throughout this document:

Term	Definition
Action	This is a runtime instance of an <i>action definition</i> .
Action definition	An action definition is metadata or a skeleton of a business task. It encompasses all possible realizations (also see <i>processing type</i> ). It determines whether business partner data is required for the execution and the business partner to which the data relates. Within the action definition, you set up the <i>merge technique</i> , that is, how many successful, failed, or unprocessed actions are allowed for a particular action definition.
Action determination	This is the runtime interface of the Post Processing Framework (PPF). When the action determination is triggered, the PPF checks the scheduling conditions of all <i>configured action definitions</i> based on the data provided by the <i>context</i> . The partner is also identified for partner-dependent <i>action definitions</i> , that is, <i>partners</i> with the specified partner function are added whenever they are provided as part of the <i>partner collection</i> . Otherwise, the partner-dependent action is deleted. Finally, the specified <i>merge technique</i> is triggered. Execution management is automatically updated for new actions.
Action profile	Smallest set of action definitions that can be investigated during one call of the <i>action determination</i> of the PPF.
Application object	The application object is the standardized proxy used to access the <i>business object</i> . The persistence of this object must be realized using the object services.
Business object	Object that contains the main data that is relevant for <i>action determination</i> and execution. Its persistence is either given as a Business Object Repository (BOR) object or as a persistent class of the object services.
Configured action definition	The <i>action definition</i> must be configured so that an action can be created during <i>action determination</i> . The combination of the selected <i>processing type</i> , scheduling condition, and starting condition is called a configured action definition. An <i>action definition</i> can have several configured instances. Moreover, almost all of the technical settings for the action definition can be configured differently.
Context	Set of data transferred to the <i>action determination</i> . The context includes the name of the application, the <i>action profile</i> , the <i>application object</i> , and a collection of partners.
Merge technique	As part of <i>action determination</i> , new actions are merged with the old actions of an action definition. You specify the merge technique to be used at <i>action definition</i> level. For example, the merge technique <i>1 successful action per action definition</i> deletes

	all newly determined actions if an action in the specified action definition is successfully executed.
Partner	<i>Action definitions</i> can be assigned to special person groups. To identify these groups, the PPF expects the partner function to be included in the address data (for example, “Responsible Person” or “Customer”). The partner number can also be entered if required. An object that contains all of these attributes is known as a <i>partner</i> . The partner functions are defined using a BAdI from the application.
Partner collection	This is the runtime collection of all partners in a business object. The collection is part of the <i>context</i> .
PPF	<i>Post Processing Framework</i> . This provides SAP applications with a standardized way of executing and managing conditioned business tasks. In contrast to the SAP Business Workflow, these business tasks are mutually independent, meaning that there is no scope for a process flow.
Processing type	The processing type refers to the technical realization of an <i>action definition</i> . For example, start a workflow, send a PDF-based form by e-mail, and so on.



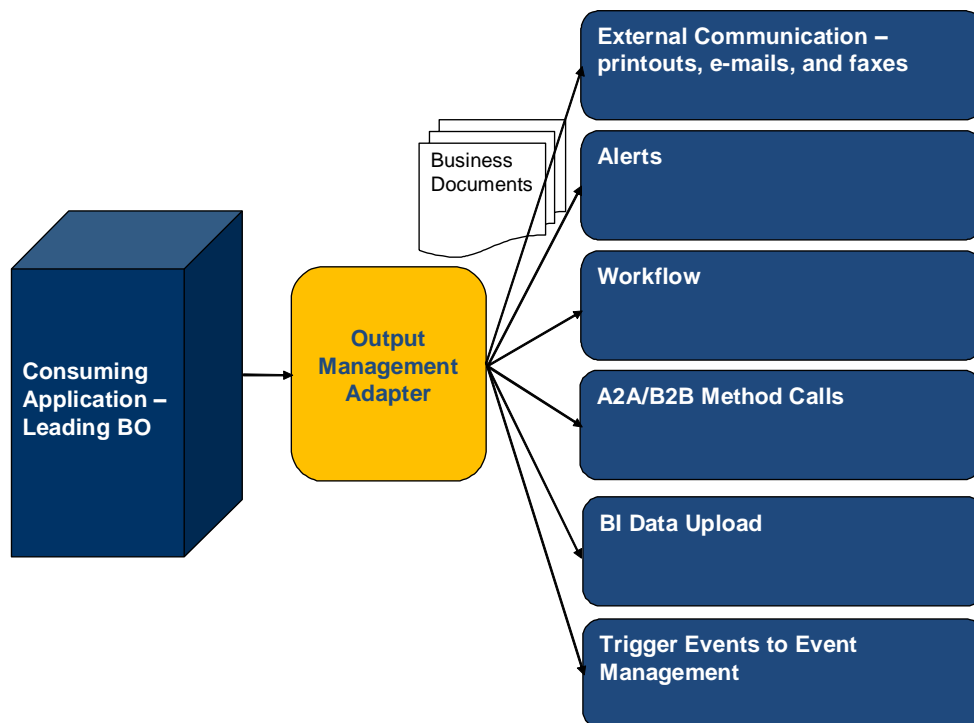
## 2 Output Management Overview

The basic functions of an output management system are to:

- Issue business documents in response to certain business events. The following can be issued:
  - External communication such as printouts, faxes, and e-mails with a print preview option
  - Alerts
  - Workflow
  - Business-to-business (B2B) and application-to-application (A2A) communication
  - BI data upload
  - Events to an Event Management system on the basis of decisions or in response to business events (for example, when a forwarding order is confirmed in a transportation scenario)
- Easily configure:
  - Print forms
  - Business decisions to schedule and execute outputs
  - Data extraction to print forms
- Output of business documents with the following capabilities:
  - Automated (back end), rule-based dispatching of outbound communications in response to business events such as the creation of an order, the release of a tendering request, and so on
  - Preview of output and manual (front end) outputs in user interface (UI) scenarios
  - Archive output dispatched in relation to business documents
  - Include actual print files (PDFs, and so on) in the archive system for bookkeeping and later references
  - Mass output from a worklist
- Optimize performance in relation to SAP Transportation Management 8.0
- Provide a comprehensive output management UI that:
  - Classifies all the outputs based on the processing status that can be audited or monitored. The system triggers and executes the outputs in relation to a business document.
  - Supports archiving, previewing, and log display
  - Enables users to generate, activate, or deactivate actions prior to execution



- Enables users to change the printer, number of copies, communication partner, output language, and recipients prior to execution
- Enables e-mail text to be added for the e-mail recipients
- Enables attachments (that are either predetermined using a BAdI, configured in Customizing, or uploaded into the attachments folder of the consuming BO) to be sent to the e-mail recipient
- Enables users to choose the mode of communication such as print, e-mail, or fax for outputting the document
- Enables users to use a default text for the subject line of an e-mail and enables them to change the subject text
- Enables users to view the list of all relevant actions for a given document
- Enables users to use Customizing settings to generate and execute actions without applying rules that would otherwise apply

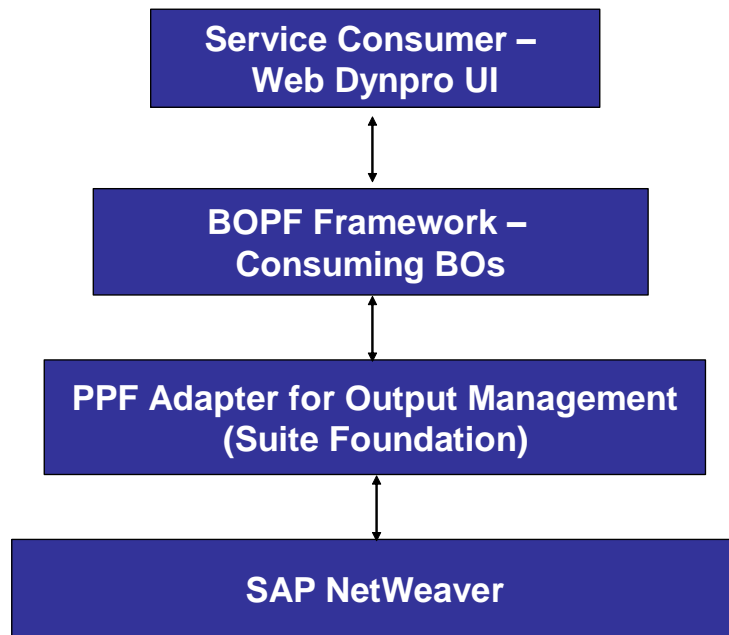


### Overview of Output Management Adapter

*Output management adapter (developed in the SAP\_BS\_FND layer) enables output to be triggered for a given BO node developed in the Business Object Processing Framework (BOPF) environment.*

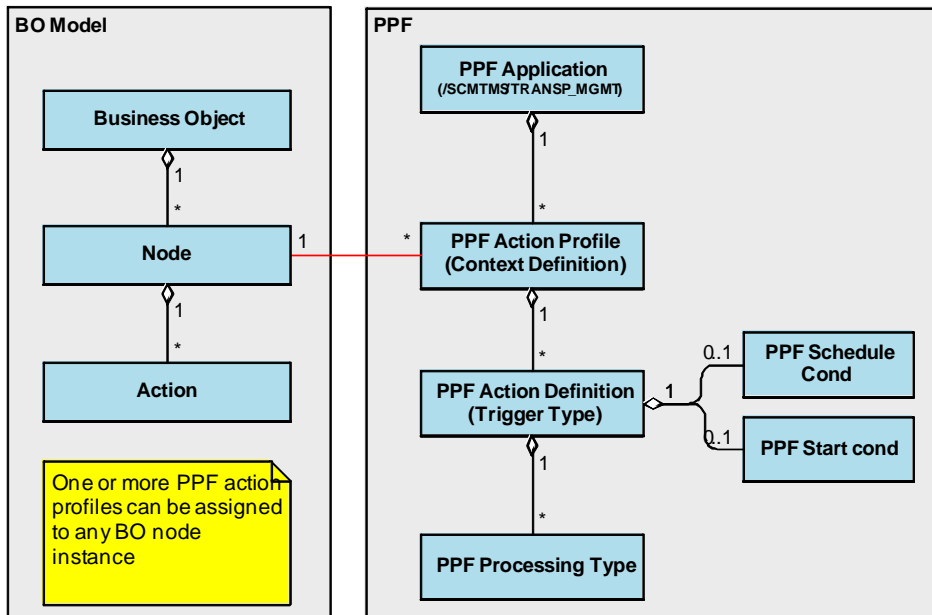
*Output management (SAP\_BS\_FND\_702 SP05) uses the Post Processing Framework (PPF).*

You can connect the PPF at the level of the individual BO nodes. You can also set up PPF triggers on a BO node-specific (use case) basis and specify whether they are critical or non-critical.



#### **Overview of Output Management Integration**

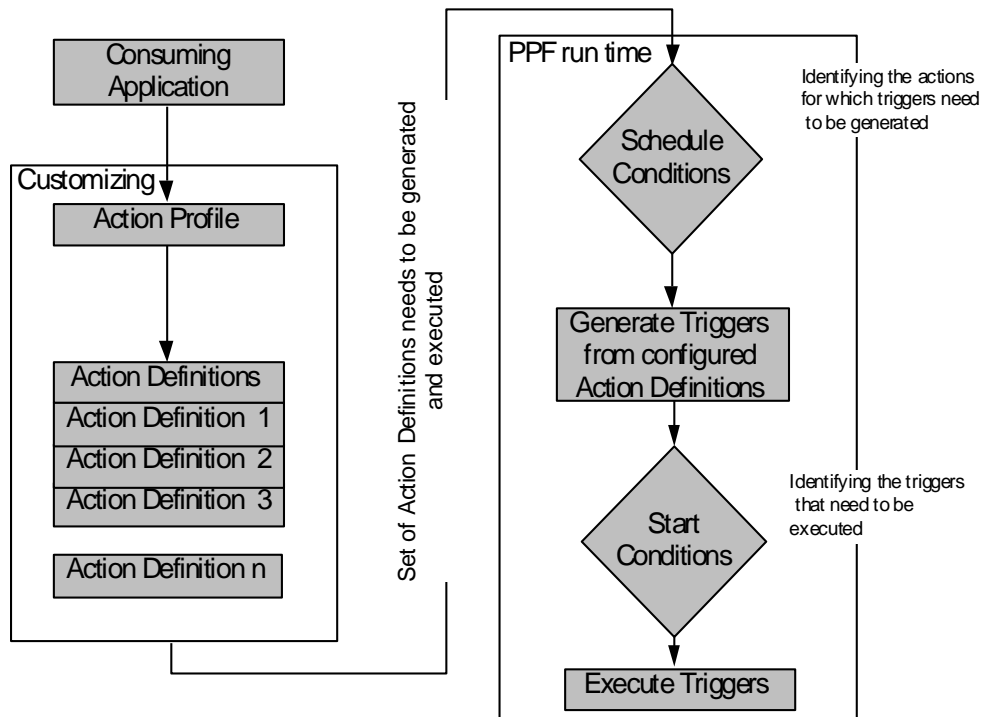
The following figure shows the logical connection of the PPF to BOPF BO nodes:



### Logical Connection of the PPF

The PPF must be connected to these nodes whenever outbound communication is required. The figure above shows where the PPF for output management must be configured. For more information about the PPF actions that are relevant for your application, see the Modeling Guide for PPF.

The following figure shows the integration of PPF Customizing and the PPF runtime to the consuming application:



### Integration of PPF Customizing and PPF Runtime



### 3 Features of Output Management

Output management contains an adapter that is developed in the BS\_FND layer. It uses the post processing framework (PPF), which provides SAP applications with a standardized way of executing and managing condition based business tasks.

The output management adapter has following capabilities:

- PPF utilities:
  - Technical connection between the PPF action profiles and BO nodes
  - Generates PPF triggers for BO node instances
  - Processes triggers in two modes depending upon the configuration settings – “before commit” and “after commit”. These enable the user to segregate critical and non-critical actions.
- BO to wrap PPF transaction data that facilitates:
  - Output management UI
  - Preview of output for individual unprocessed triggers (PDFs, alert data, and so on)
  - Read from archive and preview for processed triggers
  - Repeat of a trigger (it currently checks the merging constraints as set in the PPF configuration if required)
  - Manual generation of triggers (this currently checks for the scheduling and merging constraints set in the PPF configuration)

Basic output management features are described below in more detail.



## 3.1 Immediate Print

You can manually trigger a print action directly from the document. This is a forceful execution of the output regardless of the schedule conditions and start conditions. The schedule conditions and start conditions are ignored and the document is printed. An immediate print is not recorded in the archive.

If you choose *Print Immediately*, the system displays a dialog box that lists all of the actions that are relevant for external communication. It also displays a switch option that allows you to choose between the output details (output device and number of copies) pertaining to the user or the PPF configuration.

You must create a BOPF action in the root node of the BO and assign the name `/BOFU/S_PPf_IMMEDIATE_PRINT` to the parameter structure. Class `/BOFU/CL_A_PPf_IMMEDIATE_PRINT`, which contains the implementation steps for the immediate print function, must also be provided.

You can enter the information about the default printer and the number of copies in the following locations:

- User profile
- Condition configuration in the PPF

By default, the settings in the user profile are taken into consideration. If the BO is to use the values from the condition configuration instead, they must inherit the immediate print class and the value in the constructor must be set using constant `MC_CONDITION_DEVICE` for the variable `MV_PRINT_SETTINGS`.

Note: An alternative way of triggering an immediate print is to set the processing time parameter in Customizing for the action definition to *Processing Using Selection Report*. The action is then scheduled but not executed. You can then execute the action manually from the output management UI.

The current filtering logic removes all non-print PPF actions (such as *Send B2B message*).



## 3.2 Automated Output

You can schedule automated outputs throughout the life cycle of a business document. Outputs are scheduled and executed on the basis of the business scenarios and the life cycle status. Some of the configured action definitions are not relevant for a business document in a specific business scenario. The relevant actions and outputs are included using the schedule conditions. These outputs could be partner-dependent or partner-independent. For partner-dependent actions, the business object provides the partner-specific data and outputs are triggered on the basis of this data (you can change or override some of the parameters on the output management UI prior to execution). Automated outputs are saved in the history and can be viewed on the *Output Management* tab page.

If you want to e-mail external communication, you can send attachments along with the e-mail. Attachments can be added from the following locations:

- Customizing settings for the PPF
- Output management UI

The output management UI is supplied with a list of attachments. These attachments are added as part of the attachments folder UI.

If you want to print external communication, there are various ways in which you can specify the printer. The printer is determined in the following order of priority:

1. Redefining method `DETERMINE_PRINTER ( )` of class `/BOFU/CL_PPF_SERV_FOR_BO`. The printer can be determined using ABAP or BRFplus.
2. The static configuration setting in Customizing for the PPF
3. User Printer setting



### 3.3 Originals and Copies

The output management adapter does not differentiate between originals and copies of business documents. The print form is also the same for both originals and copies. The difference is brought about by configuring different actions for originals and copies. The originals and copies are printed in different trays of the same printer, which contain different types of paper. These trays can be modeled as two separate SAP printer configurations. For more information, see SAP Note [143375](#).



This is relevant only for external communication.





## 3.4 Mass Output

You can trigger mass outputs from a worklist. For example, you can select multiple orders from the worklist, choose the actions for which you want to trigger the mass outputs, and then perform mass execution of the unprocessed actions. The system checks the start conditions and if the outcome is successful, the actions are executed.

You have to create a BOPF action in the root node of the BO and provide class `/BOFU/CL_A_PPF_MASS_OUTPUT`, which contains the implementation measures for the mass output function.



## 3.5 Features of the Output Management UI

The header area of the output management user interface (UI) uses a table to display all of the actions for a business document. The system displays the action details, message log, communication details, and document preview on separate tab pages in the screen area below the table. Depending on the processing status, the system groups the actions as *Unprocessed*, *Successfully Processed*, or *Processed with Errors*.

### Generate Actions

You can generate all of the relevant actions for a business document. If you choose ► **Generate** ► **Actions Including Conditions Checks** , the system performs the following tasks:

- Selects all of the relevant actions for the business document
- Executes the schedule conditions
- Generates the actions if the schedule conditions are met

The system executes the actions either immediately or when you save, depending on how you have configured the action. In the *Maintain PPF Settings* Customizing activity, you can set the *Processing Time* field to *Immediate processing*, *Processing when saving document*, or *Processing using selection report*. If you validate changes to data in the scheduled condition to determine whether you create a new trigger, you may not be able to generate the trigger manually. This is because the data in the business document has not been changed since the last trigger was generated. You can validate changes to data in the scheduled condition by comparing the data before and after the change.



In the *Maintain PPF Settings* Customizing activity, if you specify an action as *Schedule Automatically* and the processing time as *Processing when saving document*, the system generates a trigger in the active state in the UI and deletes the trigger on saving. However, the system does not delete the trigger if the trigger is changed.

### List of Actions

You can view the list of all actions that are relevant for the document by choosing ► **Generate** ► **List of Actions** . A dialog box appears with the list of all the possible actions for the document, along with the configured processing types, and schedule condition status. You can view the latest status of the schedule condition by choosing *Check Schedule Conditions*. You can select one or more actions for generation. You can generate the actions by choosing the *Generate Actions* pushbutton and choosing one of the following options:

- *Include Condition checks*

You can generate the selected action after executing the relevant schedule conditions. If the schedule conditions are met, the actions are generated.

- *Skip Condition Checks*

You can generate the selected actions without checking the relevant schedule conditions. The system always generates the actions, regardless of whether or not the schedule conditions are met. You can specify which actions can be generated without checking the

schedule conditions in the *Maintain Output Management Adapter Settings* Customizing activity.



Generating an action without checking the schedule conditions may lead to possible inconsistencies in the document. We do not recommend that you do this. To restrict the actions that can be generated in this manner, you can select the actions that are safe to generate without schedule condition checks in the action settings of the *Maintain Output Management Adapter Settings* Customizing activity. Unless specified otherwise in Customizing, the system executes actions only if the schedule conditions are met.

The system merges generated actions, based on the merging method you specify for the action definition in the *Maintain PPF Settings* Customizing activity.

## Regenerate

You can regenerate specific actions by selecting existing processed actions on the *Output Management* tab page and choosing *Regenerate*. The system evaluates the schedule conditions and generates the action on the basis of the evaluation. You can use the *Regenerate* function only for actions with a processing type of *External Communication*.

## Edit

The ability to edit in output management is influenced directly by the lock status of the consuming business object (BO). If the BO is in display mode, then the output management UI is also in display mode.

You can edit all of the unprocessed actions. This enables you to enter or change the data in specific fields for all of the unprocessed actions. This function is only possible for the actions with a processing type of *External Communication*.

You can change the following fields for an unprocessed action:

- *Language*

If you schedule actions for execution when you save a business document, the consuming application can determine the language based on the business partner. The system displays the language on the *Outputs* tab page. If you want to repeat the action at a later time but in a different language, you can change the language on the UI and execute the output. This function is only possible for the actions with a processing type of *External Communication*.

- *Printer*

You can change the printer.

- *Number of Print Copies*

You can enter the number of copies to be printed. This should be a value between 1 and 255.

- *Subject and Text*

- By default, the system uses the description of an action as the subject for an unprocessed trigger.
  - You can use the *Maintain Output Management Adapter Settings* Customizing activity to create a default e-mail text.
- *Recipients*  
You can add, change, and delete the recipient types and recipients for external communication.
  - *Attachments*  
You can attach or delete attachments that you send to e-mail recipients.
  - Interactive PDF content  
You can edit the content of an interactive PDF form for an unprocessed action.
  - Deactivate an action  
You can deactivate an action if you do not want to execute the action in a particular scenario.

## Subject and Text

You can send business documents to e-mail recipients.



- By default, the system uses the subject of an unprocessed trigger as the description for an action.
- You can use the *Maintain Output Management Adapter Settings* Customizing activity to create a default e-mail text.

## Recipient

You can add and delete recipients of external communication triggers. The system supports the following recipient types:

- E-mail
- Fax
- Business partner
- SAP name



- If the system can determine the communication mode preferences for a business partner, it displays this automatically.
- For an action, it is not possible to specify a mode of communication by business partner.

## Editing Interactive PDF content

You can associate an interactive PDF with an action, and edit the PDF. You can save the changed document, and change the document later if required.

1

- In the *Maintain PPF Settings* Customizing activity, you can specify that the system archives a PDF on execution.
- The system supports the interactive feature for PDF forms only with template type ZCI.

## Attachments

You can attach or delete attachments that you send to e-mail recipients with a business document. You can add attachments that you have specified in the following places:

- *Maintain PPF Settings* Customizing activity
- BAdI
- Attachments folder of the consuming BO

1

Due to technical limitations the system writes the addition or deletion of an attachment to the database records. This means that even if you exit the consuming application without an explicit save, the system keeps the changes to the e-mail attachments.

## Activate and Deactivate

You can activate or deactivate an action on the output management UI. This enables you to prevent the system executing a particular action in a particular scenario.

## Delete

You can delete an unprocessed action.

## Execute

► *Execute Actions* ► *Include Condition Checks* ▢

You can execute unprocessed actions on the Output Management UI. The start conditions are executed and, if these are met, the action is executed.

► *Execute Actions* ► *Skip Condition Checks* ▢

You can skip the start conditions and execute unprocessed actions on the Output Management UI. Regardless of whether the start conditions are met, the system always executes the action. You can specify the appropriate settings in the *Maintain Output Management Adapter Settings* Customizing activity.

## 1

If you skip the start conditions when executing an action, inconsistencies may arise in the output document. Because of this, we would not recommend this approach. To restrict the actions that can be executed in this manner, you can select the actions that are safe to execute while skipping the start condition. For more information, see the *Processing Type Settings* view in the *Maintain Output Management Adapter Settings* Customizing activity.

By default the system executes actions only if the start conditions are met.

## Direct Outputs

You can use output management just to check conditions and trigger certain types of actions, without creating an output history. Types of actions include updating events to SAP Event Management, or updating deltas to the SAP Business Information Warehouse (BW) queue.

You can specify the *Direct Output Agent* in the Customizing for *Maintain Output Management Adapter Settings*. You do not need to specify any other Customizing settings to enable this feature.



## 4 Output Management Settings and Configuration

To access the Customizing activities for configuring Output Management, choose ► *Cross-Application Components* ► *Processes and Tools for Enterprise Applications* ► *Reusable Objects and Functions for BOPF Environment* ► *PPF Adapter for Output Management* ►.

- Executing *Maintain PPF Settings* takes you to transaction `SPPFCADM`.
- Executing *Maintain Output Management Adapter Settings* takes you to Customizing for the output management adapter.



The *Maintain PPF Settings* Customizing activity enables you to use a wizard to change PPF settings and create conditions. For more information, see [Maintaining PPF Settings](#) [Page 24] and [Creating Conditions](#) [Page 29].

You can access the wizard in the following ways:

- Enter transaction `SPPFCWIZARD` for direct access
- In transaction `SPPFCADM`, choose *Wizard for Defining and Scheduling an Action*



## 4.1 Maintaining PPF Settings

1

If you are unfamiliar with the Post Processing Framework (PPF), see the [Appendix](#) [Page 60].

If you want to use the PPF adapter for output management with archiving, you must configure the PPF. If you want to use the direct output feature of output management, without the PPF or output archiving, these settings are not required.





## 4.1.1 Defining Applications

### Procedure

To create a PPF application:

1. Call transaction `SPPFCADM`.
2. Check whether a PPF application exists.
3. If not, choose *New Entries* or press `F5` to create an application and enter an appropriate name (for example, `/BOFU/DEMO_APPLICATION` or `/SCMTMS/TRANSPORTATION`) and a description.



## 4.1.2 Defining Action Profiles

### Prerequisites

You can create one or more action profiles for a given BO-node.

Before you create an action profile for your business object and node, make sure that you have fulfilled the following prerequisites:

- You have configured the following critical outputs (actions) in one action profile:
  - Processing type intended for triggering outputs such as business-to-business communication (B2B), application-to-application (A2A) messages, and so on (triggered immediately or upon saving before the application data is committed)
  - All outputs that are to be executed when a node is deleted
- Other non-critical outputs such as sending e-mails, faxes, and printouts (triggered after the commit and processed once the application data has been saved in the background task) must be configured in a separate action profile.
- The correct namespace has been prefixed to the name of the action profile (for example, /BOFU/ACTION\_PROFILE\_UNCRITICAL).

### Procedure

1. Select the created application and choose *Define Action Profiles and Actions*.
2. If an appropriate action profile is not available, choose *New Entries* to create one.
3. Enter the name of your action profile.
4. Enter a description.
5. Specify the *Category of Object Type* as *Persistent Class* and enter the *Object Type Name* as /BOFU/CL\_PPF\_CONTAINER.
6. In the *Context Class* field, enter /BOFU/CL\_PPF\_CONTEXT.



## 4.1.3 Creating Action Definitions

### Prerequisites

You have defined an action profile. For more information, see [Defining Action Profiles](#) [Page 26].

### Procedure

1. Select your action profile and double-click *Action Definition* in the structure tree.
2. Enter a business-relevant name for the action definition (such as “Send Confirmation”). Prefix it with the correct namespace (for example, /BOFU/ACTION\_NAME).
3. Enter a description.
4. Specify the *Processing Time* as *Processing when saving document* or *Processing using selection report* (if triggers are to be executed in a batch job and not immediately after saving).
5. Select the following checkboxes:
  - *Schedule Automatically*
  - *Changeable in Dialog*
  - *Executable in Dialog*
6. If you want to configure the output for each partner (for example, if some shippers require an e-mail but others require a printout), specify the partner function for which the action is to be configured.
7. In the *Determination Technology* field, choose *Determination Using Conditions that Can Be Transported*.
8. In the *Rule Type* field for evaluating the start condition, choose *Conditions Using Business AddIn (BAI)*.
9. On the *Action Description* tab page, enter an action description.
10. If necessary, maintain the *Action Merging* tab page.



## 4.1.4 Defining Processing Types

### Prerequisites

You have created an action definition. For more information, see [Creating Action Definitions](#) [Page 27].

### Procedure

The *Permitted Processing Types of Action* table must contain the following information:

- *Method Call* for triggering outputs such as B2B, A2A, and so on, or for triggering application-specific tasks
- *External Communication* for triggering printouts, faxes, and e-mails (Adobe forms)
- *Trigger Alert* for triggering alerts
- *Workflow* for triggering workflows

To define the processing type:

- For the method call, enter the method name `/BOFU/PPF_STANDARD`.
- For external communication:
  1. Enter the name of the Adobe form as the form name.
  2. Enter the form type *PDF-Based Forms*.
  3. Enter the name of the format as `/BOFU/PPF_STANDARD`.
- For alerts:
  1. Enter the *Alert Category* (create an alert category).
  2. Select the *Communication Strategy* as *Default Communication Type*.
  3. Enter an alert exit. Create a new BAdI implementation for the BAdI `ALERT_EXIT_LOCAL_PPF`, as specified in `PPF_Exits`. For more information, see [PPF Exits](#) [Page 36].



## 4.2 Creating Conditions

### Procedure

1. Call transaction SPPFCADM.
2. Select the relevant application (for example, /BOFU/DEMO\_APPLICATION) and choose *Condition Configuration (Transportable Conditions)*.

### Defining the Schedule and Start Conditions

1. Choose the action profile that you created and double-click the relevant action definition.
2. Choose the *Schedule Condition* tab page and enter the schedule condition  
/BOFU/EVAL\_SCHEDULE\_CONDITION (BAId implementation  
/BOFU/PPF\_STD\_EV\_SCH).
3. If required, choose the *Start Condition* tab page and enter the name of the start condition  
as /BOFU/EVAL\_START\_CONDITION (BAId implementation  
/BOFU/PPF\_STD\_EV\_STA).

### Defining the Processing Details

If the processing type is *External Communication* and the output is partner-dependent (*Partner-Dependant* checkbox is selected), proceed as follows:

1. Choose the *Processing Details* tab page.
2. Choose the *Document* tab page.
3. In the *Communication Method* field, choose *Fax*, *Internet Mail*, or *Print* to send a fax, e-mail, or printout respectively.



You must enter a start condition and a schedule condition for an action. You can use the following standard filter values:

- /BOFU/EVAL\_START\_CONDITION for the start condition
- /BOFU/EVAL\_SCHEDULE\_CONDITION for the schedule condition



## 4.3 Output Management Adapter Configuration



## 4.3.1 Customizing

In the Customizing activity *Maintain Output Management Adapter Settings*, you assign output agents to different nodes of a business object.

In a transportation scenario, for example, you can maintain outputs in both the root node of a freight order BO as well as the tendering request node of the same BO.



Due to technical constraints in the current release, the node names are not displayed on the UI even though multiple assignments are possible in different nodes. You must, therefore, ensure that the action definitions used within action profiles as well as the action profiles for such assignments themselves are unique and not reused.

The Customizing activity *Maintain Output Management Adapter Settings* also enables you to assign multiple action profiles to a business object node.

Possible reasons for assigning multiple action profiles to a BO node include the following:

- For performance reasons, you may want to differentiate between critical and non-critical outputs. Critical outputs are those that must be completed within the logical unit of work (LUW) such as B2B method calls or workflows. Non-critical outputs are those that fall under the category of communication methods, such as printouts, e-mails, and faxes. You can specify whether output is critical or non-critical in the *Output Type* column.
- You may have multiple business scenarios that result in multiple action profiles for a BO node. A classic example is one action profile for each mode of transport.

Within a business scenario, therefore, it may be the case that not all of the action profiles are relevant for a BO node. It would be more efficient and effective (in terms of performance) to automatically choose only the relevant action profiles based on the business case.

The output management adapter provides two ways of making decisions and determining data: ABAP and BRFplus. To make the configuration settings at PPF level more generic and accommodate any restrictions based on the business scenarios, generic ABAP callback methods or BRF functions are provided that:

- Identify PPF action provides that are specific to the status of the BO node
- Schedule decision-making to generate triggers for a given set of action definitions, the application data record, and a given set of partners (schedule conditions)
- Start decision-making for processing the output of a trigger (start conditions)
- Determine BO node-specific document partners
- Determine the print form for a given trigger
- Initialize alert data
- Determine the printer for a given trigger
- Determine the BO partner-specific language

- Enable the system to share the same action definition among a group of business documents.

## Procedure

1. Call transaction `SPRO` and open Customizing for SAP.
2. Choose **► Cross-Application Components ► Processes and Tools for Enterprise Applications ► Reusable Objects and Functions for BOPF Environment ► PPF Adapter for Output Management ► Maintain Output Management Adapter Setting**.



You can use ABAP or BRFplus to make decisions.

### *Defining PPF Output Agents for BO Nodes*

1. Enter the name of the business object and node for which you have configured the action profile, in the relevant fields.
2. In the *Output Agent* field, enter a meaningful name for the PPF output agent. For example, `/BOFU/DEMO1`.
3. In the *Agent Class for BO Node* field, enter the class that implements class `/BOFU/CL_PPF_SERV_FOR_BO` (see [Output Management Callbacks](#) [Page 38]).
4. Administrative data in the fields *Created By*, *Created On*, *Changed by*, and *Changed On* is completed automatically.

### *Using the BRFplus Capabilities to Define PPF Output Agents for BO Nodes*

1. In the *Appl. ID for Node* field, enter the BRFplus application ID for the node. This application has BRFplus functions to select the PPF profiles for the given node and the document partner data.
2. In the *Func. for Act. Prof.* field, enter the BRFplus selection function that returns the list of PPF profiles for a given BO node.
3. In the *Func. for Partners* field, enter the BRFplus selection function that returns the list of PPF partners for a given BO and node.
4. Enter the *Catalog ID* for node: This is the catalog of other objects. In this case, the catalog ID is used for the where-used analysis in BRFplus so that accidental deletion of decision functions can be prevented. The catalog is available to business users for navigation and as a workplace. Business users can also maintain the attributes and structure of the catalog.



Class `/BOFU/CL_PPF_SERV_FOR_BO` is available for use in the default implementation, which can be used if no special application-specific implementation is required to obtain the PPF profiles. You must redefine the methods in this class.

### *Assigning PPF Profiles*



1. Select the name of the action profile that you maintained in transaction `SPPFCADM` for your BO and node (for more information, see [Defining Action Profiles](#) [Page 26]).
2. Specify the output type. As explained in the *Defining Action Profiles* document, if the output is critical, select *Has Critical O/P: Process immediate Actions before commit*. Otherwise, select *Has uncritical O/P: Process immediate Actions after commit*.
3. *Create DB Image*: If your output is not critical and you require a data snapshot for a particular BO, select this indicator.
4. Administrative data in the fields *Created By*, *Created On*, *Changed by*, and *Changed On* is completed automatically.

#### *Using BRFplus Capabilities to Assign PPF Profiles*

To use the BRFplus capabilities, in the *Appl. for Act. Prof.* field, enter the BRFplus application for the action profile.

#### *Specifying Action Settings (Optional)*

1. In the *Action Definition* field, use the input help (F4) to enter the name of the action previously created in transaction `SPPFCADM`. For more information, see [Creating Action Definitions](#) [Page 27].
2. *Can Generate Without Condition Check*: Select this indicator if you want to skip the relevant schedule conditions and allow the action to be generated.
3. Select the *Aggregate* checkbox to enable the user to use the action definition in all the business documents that you have grouped in the agent class logic (see [Output Management Callbacks](#) [Page 38]). This feature is available only if you have implemented SP05 of SAP\_BS\_FND 7.31 or note [1706788](#).



You can group documents FU1 and FU2 in the agent class logic. In Customizing, you specify an action definition as aggregate. The system uses the action definition in FU1 and FU2. When you change and save an action definition in FU1, the change appears in FU2. If you delete and save the action definition in FU1, the system deletes the action definition in FU2.

#### *Using BRFplus Capabilities to Specify Action Settings (Optional)*

1. In the *Action Gen Func.* field, enter the name of the BRFplus function that evaluates the schedule condition.
2. In the *Find Printer* field, enter the name of the BRFplus function that determines the name of the printer.

#### *Specifying Processing Type Settings (Optional)*

1. In the field for the class name, use the input help (F4) to enter the name of the processing type (class name of the context). The system enters the description automatically.

2. *Can Execute Without Condition Check*: Select this indicator if you want to skip the relevant schedule conditions and allow the action to be executed. This setting is relevant only for processing type *External Communication*.
3. *Text Template*: Use this to specify a text template maintained using the transaction SO10. This field is used to default the text of the e-mail in the *Communication Details* tab for the relevant action in the Output Management user interface (UI). This setting is relevant only for processing type *External Communication*.
4. *Proc. Type Dsc*: Use this field if at runtime you want to display a descriptive text in the Output Management UI, for actions with processing type *Method Call*.



- In the Output Management UI, the text in the field for *Processing Types* is defaulted to the processing type text, for example, *External Communication*, *Method Call*, or *Alert*.
- The texts, created through SO10, have to be manually included in a transport request if they need to be transported.

#### *Using BRFplus Capabilities to Specify Processing Type Settings*

1. In the *Action Start Func.* field, enter the name of the BRFplus function that evaluates the start condition.
2. In the *Change Form Func.* field, enter the name of the BRFplus function that determines the print form so that the print form can be changed.
3. In the *Initialize Alert Func.* field, enter the name of the BRFplus function that determines the print form so that the alert data can be initialized.

#### *Transient Action Definitions (Optional)*

1. In the agent class logic, you specify one business document as the source and another document as the target (see [Output Management Callbacks](#) [Page 38]).
2. In the *Maintain PPF Settings* screen, you assign an action definition to an action profile. In the *Assign Transient Actions* screen, you then assign the same action definition to another profile as a transient action definition. You assign the first action profile to the source business document. You assign the action profile with the transient action definition to a target business document.

When you execute and save the action definition in the source business document, the system makes the same action definition available in the target business document.

You can only edit the action definition in one document at a time. When you change and save an action definition in a particular business document, the change appears in all the business documents that use the action definition.



In the agent class logic, you specify that FU2 is the source document and FW01 is the target document. In Customizing, you specify the action definition of freight unit FU2 as transient for forwarding order FWO1. When you generate and save the action definition in FU2, the system makes the same action definition available in FWO1. When you change

and save an action definition in FWO1, the change appears in FU2. If you delete the action definition in FWO1, the system also deletes the action definition in FU2.

## 1

You can generate transient actions only in the source business document and not in the target business document. You can generate aggregate action definitions in all the business documents you have grouped in the agent class logic. The aggregate feature is available only if you have implemented SP05 of SAP\_BS\_FND 7.31 or note [1706788](#).



You have specified the following logic in the agent class:

- Business documents FU1 and FU2 grouped as aggregate
- Target business document FWO1 for source business document FU2
- Target business document FWO1 for source business document FU3

You have specified the following settings in Customizing:

- Selected the *Aggregate* checkbox for action definitions AFU1 and AFU2 for a business object of TOR and node of ROOT
- AFU2 as a transient action for document FWO1
- AFU3 as an action definition for document FU3
- AFU3 as a transient action for document FWO1

After you create business documents FWO1, FU1, FU2, and FU3, the system makes the following action definitions available:

- AFU1 and AFU2 as aggregate in FU1 and FU2
- AFU3 in FU3

FU3 is the transient source for FWO1. Therefore, when you generate action AFU3 in FU3, the system makes the action in FU3 available to FWO1.

When you generate action AFU2 in FU1, the system makes the action available in FU2. Since FU2 is the transient source for FWO1, the system also makes action AFU2 available in FWO1.



## 4.3.2 PPF Exits

PPF offers various exits to the consuming BOs to provide the relevant business information. These exits are listed in the table below:

BAdI Name	Implementation Name	Filter Value	Description
EVAL_SCHEDCOND_PP F	/BOFU/PPF_STD_EV_SCH	/BOFU/EVAL_SCHEDULE _CONDITION	Used to evaluate the business determination rules to select the actions to be generated
EVAL_STARTCOND_PPF	/BOFU/PPF_STD_EV_STA	/BOFU/EVAL_START_CO NDITION	Used to evaluate the business checks to determine when to execute the generated outputs
COMPLETE_PROC_PPF	Customer specific BAdI implementation needs to be provided	Not applicable	Used to complete processing
PRINTER_DETERM_PPF	Customer specific BAdI implementation needs to be provided	Not applicable	Used to identify the printer used to print the action
EXEC_METHODCALL_P PF	/BOFU/PPF_STD_METHOD	/BOFU/PPF_STANDARD	Used to execute actions with the method call processing type
GET_PARTN_ROLES_PP F	Customer specific BAdI implementation needs to be provided	PPF application, profile	Used to retrieve the possible partner functions of the application
DOC_PERSONALIZE_BC S	/BOFU/PPF_STD_DOCBCS	/BOFU/PPF_STANDARD	Used to retrieve data from the business object and personalize the document
ALERT_EXIT_LOCAL_PP F	Customer specific BAdI implementation needs to be provided	Action profile, action	Used to read application data for an alert

Output management handles the enhancements offered by PPF using the BAdI façade /BOFU/CL\_PPF\_BADI\_FACADE. This class contains all the BAdIs that PPF offers. Wherever possible in output management, a standard BAdI implementation is provided that calls the BAdI façade. Examples of standard BAdI implementations are START\_CONDITION and SCHEDULE\_CONDITION.

In certain cases, using standard implementations is not possible. For example, where the BAdI filter depends on PPF settings such as PPF action name and PPF action profile. In these cases, you have to create your own BAdI implementation and use the BAdI façade to plug it into output management. To do this, from your BAdI implementation, delegate the call to the appropriate method in the BAdI façade. This will ensure that the system calls the corresponding methods in your agent class.



The following is a coding example for a BAdI method complete\_mail. Identify the BAdI façade method that you need to call and make the call as given in the example. Replace the method name and parameters according to your needs.

```
METHOD if_ex_complete_proc_ppf~complete_mail. "BAdI implementation
method

DATA lo_facade TYPE REF TO /bofu/cl_ppf_badi_facade.

lo_facade ?= /bofu/cl_ppf_badi_facade=>get_instance( ).

CALL METHOD lo_facade->if_ex_complete_proc_ppf~complete_mail

EXPORTING

io_context = io_context

ip_protocol = ip_protocol

ip_no_protocol = ip_no_protocol

io_medium = io_medium

flt_val = flt_val.

ENDMETHOD. "IF_EX_COMPLETE_PROC_PPF~COMPLETE_MAIL
```



### 4.3.3 Output Management Callbacks

Since Output Management is deployed in the SAP\_BS\_FND layer, the business data required to execute outputs and make decisions must be identified by and transferred from the consuming business objects.

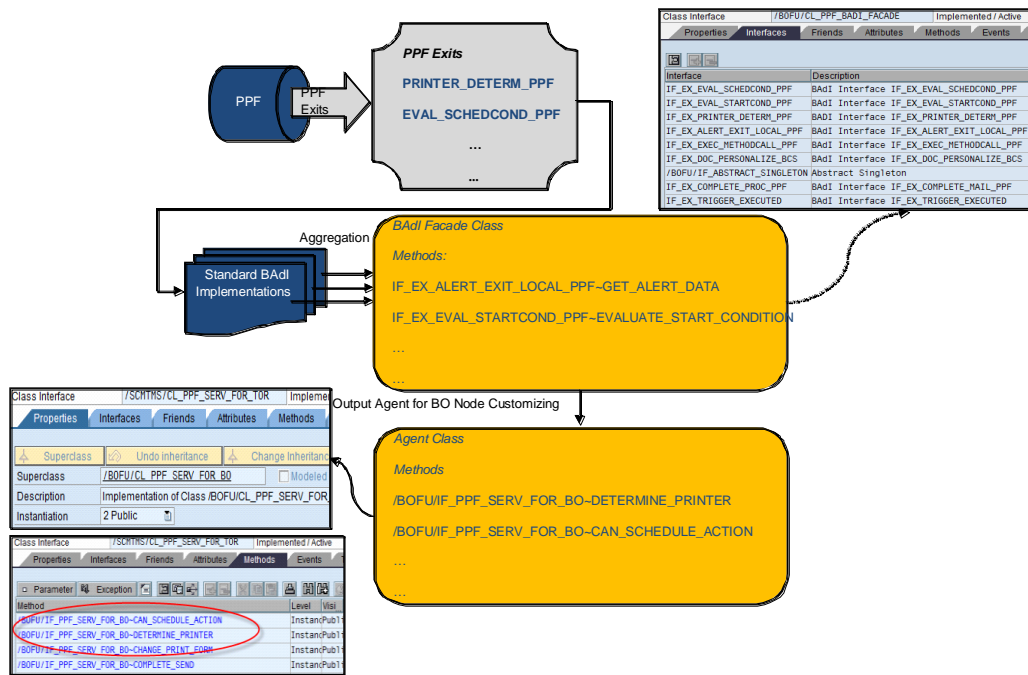
All of the PPF BAdI interfaces are aggregated into a BAdI facade class `/BOFU/CL_PPF_BADI_FACADE`, which in turn triggers the corresponding method of the agent class. The standard implementations of the PPF BAdI interfaces instantiate the BAdI facade and trigger the corresponding method in the BAdI facade.

You can retrieve business data and make decisions by extending the agent-class callback methods provided by the Output Management adapter. In this case, class `/BOFU/CL_PPF_SERV_FOR_BO` must be inherited and the necessary methods redefined in the agent class implementation.

Class `/BOFU/CL_PPF_SERV_FOR_BO` provides a generic implementation of interface `/BOFU/IF_PPF_SERV_FOR_BO`. The business objects or application users in output management must inherit this class and redefine the methods in line with the relevant business scenarios. BO information and error or information messages can be transferred to the output management layer using the method parameters. Parameter `IO_MESSAGE` is important since any exceptions caused in the BO layer would need to be communicated to the OM layer. This facilitates the debugging process should errors occur.

Example class: `/BOFU/CL_PPF_SERV_BO_DEMO_SO`, which has extended `/BOFU/CL_PPF_SERV_FOR_BO`.

It is sufficient to extend the agent class and implement all of the necessary callback methods (including BAdI methods). The figure below shows how the agent class is called when the BAdI method is called in output management.



## Output Management Callback Mechanism

The following sections explain the various determinations and condition checks that can be carried out in the methods of the agent class.



### 4.3.3.1 Action Profile Determination

In some cases, multiple action profiles are valid for a use case. In a transportation scenario, for example, you can assign multiple action profiles to a document type. However, the relevant profiles must be communicated to the output management adapter. You can select the action profiles based on the business scenario. All of the business logic and criteria for selecting the relevant action profiles can be implemented in method `GET_PROFILES()`.

Each of the interface methods can use the ABAP implementation as well as the variant for BRFplus. Implement the appropriate variant depending on the usage, that is, redefine the method with either the suffix ABAP or FDT depending on your scenario.

#### 1

You must redefine method `GET_PROFILES_BY_ABAP()`. If this method is not redefined, the OM adapter takes into account all of the assigned action profiles. This may impact performance since the PPF considers all of the action profiles assigned to the BO node.

Method: `GET_PROFILES_BY_ABAP()`

If you use an ABAP implementation, you can redefine this method to write the following:

- Appropriate application state-specific logic to select the action profile for the given input keys. The keys correspond to node instances.
- Appropriate logic to obtain the alternative key from the business object for the given input keys. If an alternative key is not available, use the node ID (GUID) as the PPF key.

Method: `GET_PROFILES_BY_FDT()`

If you use a BRFplus function, there is a method that does not require a business add-in (BAI) implementation since the function returns the required PPF profiles and alternative keys.





### 4.3.3.2 Action Determinations and Schedule Condition Checks

An action profile consists of one or more action definitions. Not all action definitions are relevant for all business scenarios. Some actions, such as B2B communication or workflows, are relevant only in certain business scenarios. You can schedule only those actions that are relevant for a scenario. This process can also be used to specify the criteria for scheduling outputs.

During execution, relevancy checks must be carried out in an exit provided by the PPF. This exit is known as a *schedule condition*. The schedule condition determines whether an action is to be scheduled for processing. An action is only generated, therefore, if the schedule condition is met.

1

You must redefine method `CAN_SCHEDULE_ACTION_BY_ABAP ()` in the inherited class.

Method: `CAN_SCHEDULE_ACTION_BY_ABAP ()`

If you use an ABAP implementation, ensure that you:

- Write appropriate logic to retrieve the BO data and check whether the criteria have been met to enable the PPF action for the given key. If the criteria have been met, this function returns the keys. These are the keys for which the system generates the PPF trigger for the output.
- Check different schedule conditions in the same redefinition if more than one PPF action is defined for a given node. You must also specify a CASE statement for the input action name.

Method: `CAN_SCHEDULE_ACTION_BY_FDT ()`

If you use a BRFplus function, do not redefine this method since it calls the BRFplus function to obtain the keys that enable a PPF action.

If this method is not redefined, all the action definitions and actions are generated by the OM adapter. This impacts performance since the PPF considers all of the actions regardless of whether they are relevant.



### 4.3.3.3 Start Condition Check

The exact time at which all relevant actions are executed is controlled by the start conditions. The system checks the start condition to determine whether it has been fulfilled, and if so, executes the action. You can use the start conditions to check the life cycle status, the time spent waiting for external events, the time of day at which an action is to be executed, and so on. This can also be defined as the criteria for executing output.



You must redefine method `EVAL_START_COND_BY_ABAP ()` in the inherited class.

Method: `EVAL_START_COND_BY_ABAP ()`

If you use an ABAP implementation, write appropriate logic to obtain the BO data and check whether the PPF action can be mass-enabled for the specified key.

Method: `EVAL_START_COND_BY_FDT ()`

If you use a BRFplus function, do not redefine this method since the function returns a status that indicates whether the system can execute the specified action.

If you do not redefine this method, the actions are executed as soon as they meet the schedule conditions or the document is saved. This is dependent on the Customizing setting that determines whether the action is set to *Process during save* or *Immediately*.



### 4.3.3.4 Determine Printer

The consuming application can identify the printer on which the business document is to be printed using the callback method `DETERMINE_PRINTER()`, which implements the PPF BAdI to determine the printer.

Method: `DETERMINE_PRINTER_BY_ABAP()`

If you use an ABAP implementation, write appropriate logic to obtain the BO data and identify the appropriate printer to print business documents.

Method: `DETERMINE_PRINTER_BY_FDT()`

If you use a BRFplus function, do not redefine this method since it calls the BRFplus function that returns the printer.



### 4.3.3.5 Determine Document Partners

The output management adapter requires partner-based information for a business document.

In Customizing for the PPF, you can configure whether an action is partner-dependent or partner-independent.

The partner function must be provided for partner-dependent actions. You can generate as many action triggers (instances of actions) as there are partners. The PPF expects the business object to provide the communication partners for the business document. The actual partners relevant for the partner function as specified in Customizing are then chosen. For the *External Communication* processing type, the communication method specific to the selected partners is chosen and the outputs are executed. To obtain the document partners, the callback method must be redefined.



Redefine method `GET_COMM_PARNR_BY_ABAP ()` in the inherited class.

**Method:** `GET_COMM_PARNR_BY_ABAP ()`

If you use an ABAP implementation, write appropriate logic to select the document partners from the business object for the given input keys. You can select the document partner data only if partner-dependent actions are supported.

If you do not redefine this method, partner-specific scenarios fail. For example, e-mail and fax scenarios require partner-specific information to execute the output.

**Method:** `GET_COMM_PARNR_BY_FDT ()`

If you use a BRFplus function, do not redefine this method since it calls the BRFplus function that returns the required document partners.



### 4.3.3.6 Determine Partner Language

In partner-specific output scenarios, you can execute output in the partner's language for the *External Communication* processing type. For partner-dependent actions, the partner function must also be provided. You can generate as many action triggers (instances of actions) as there are partners. For each partner, the partner language can be obtained from the business object. For this purpose, a generic callback method is provided to obtain the partner's language.



Redefine method `GET_LANGU_4_EXT_COMM_BY_ABAP` in the inherited class.

If you do not redefine this method, the outputs are executed in the default (logon) language. BRFplus is not supported for this determination.



### 4.3.3.7 Specify Output Content for PDF Forms

You can specify the content of forms and e-mails. This information can be obtained using the callback method to the business object. The information includes the form content, any attachments, and the document number.



Redefine method `PERSONALIZE_DOC_BY_ABAP ()` in the inherited class.

Method: `PERSONALIZE_DOC_BY_ABAP ()`

If you use an ABAP implementation, write appropriate logic to retrieve the BO data and complete the print form.

Method: `PERSONALIZE_DOC_BY_FDT ()`

If you use a BRFplus function, do not redefine this method since it calls the BRFplus function to retrieve BO data, completes the form, and calls the necessary form processing APIs.

If you do not redefine this method, the content is missing from the outputs and the outputs are not executed. BRFplus is not supported for this activity.



### 4.3.3.8 Specify Application Keys for Transient Action Definitions

Transient actions are action definitions that the system makes available in a target business document when you generate the action definition in a source business document. To enable transient actions, you must assign the relevant action definitions to the source business document, and assign the same action definitions as transient action definitions to the target business document in the Customizing activity *Maintain Output Management Adapter Settings* under ► *Cross-Application Components* ► *Processes and Tools for Enterprise Applications* ► *Reusable Objects and Functions for BOPF Environment* ► *PPF Adapter for Output Management* ►.

To display the transient actions in a target business document, the system requires the keys of the source business document that contains the action definitions.

To define the logic to determine the keys of the source business document using the input keys of the target business document, you must redefine method `APPL_KEY_4_TRANS_ACT ( )`.

### More Information

[Customizing](#) [Page 31]



### 4.3.3.9 Specify Application Keys for Aggregate Action Definitions

You can use aggregate action definitions that you specify in Customizing to share action definitions between multiple business documents. You must specify the logic that the system uses to share the definitions

To define the logic to return the aggregated action definition identifiers and their corresponding sharing logic in the form of alternate keys, you must redefine method `APPL_KEY_4_AGGR_ACT` ( ).



You want to enable the aggregate feature for action definition `AD_1` in business documents `FU_1` and `FU_2`. `FU_1` and `FU_2` are created from the same forwarding order `FWO_1`.

You select the *Aggregate* checkbox for the action definition `AD_1` in the *Action Settings* screen in the Customizing activity *Maintain Output Management Adapter Settings* under **► Cross-Application Components ► Processes and Tools for Enterprise Applications ► Reusable Objects and Functions for BOPF Environment ► PPF Adapter for Output Management ►**.

In the agent class, you specify the following sharing logic in method `APPL_KEY_4_AGGR_ACT`:



#### Syntax

```

READ TABLE it_aggr_action INTO ls_aggr_action WITH KEY aggr_act =
'AD_1'.
IF sy-subrc = 0.
" Get the service manager instance of TOR (Freight unit)
lo_tor =
/bobf/cl_tra_serv_mgr_factory=>get_service_manager(iv_bo_key =
/scmtms/if_tor_c=>sc_bo_key ).
" Get the forwarding order keys
lo_tor->retrieve_by_association(
EXPORTING
iv_node_key      = /scmtms/if_tor_c=>sc_node-root
it_key           = it_key
iv_association   = /scmtms/if_tor_c=>sc_association-root-bo_trq_root
IMPORTING
et_target_key    = lt_trq_keys
).
" Aggregation criteria: Freight units created from the same
Forwarding order share the same action
READ TABLE it_key INTO ls_key INDEX 1.
IF sy-subrc = 0.
ls_aggr_alt_key-key = ls_key-key.
" The application key (Freight unit key)
ls_aggr_alt_key-ppf_action = ls_aggr_action-aggr_act.
" Name of the aggregate action
ls_aggr_alt_key-ppf_profile = ls_aggr_action-profile.

```



```

        " Name of the action profile
READ TABLE lt_trq_keys INTO ls_key INDEX 1.
IF sy-subrc = 0.
  ls_aggr_alt_key-alt_key1 = ls_key-key.
  " Aggregation criteria
  APPEND ls_aggr_alt_key TO et_aggr_alt_key.
ENDIF.
ENDIF.
ENDIF.

```

## Syntax

In the methods such as, EVAL\_START\_COND\_BY\_ABAP, PERSONALIZE\_DOC\_BY\_ABAP, and METHOD\_CALL\_BY\_ABAP, you write the following code to determine the FU\_1 and FU\_2 keys:

```

" Aim is to derive the application keys i.e the freight unit keys
" io_container holds the alternate keys provided in the
APPL_KEY_4_AGGR_ACT method.
" Logic applicable only for aggregate actions
IF is_ppf_act-aggr_action = 'X' AND is_ppf_act-ppf_action = 'AD_1' .
  " Retrieve the alternate key from the container instance(the
  forwarding order key in our example)
  TRY.
    ls_key-key = io_container->get_doc_id_fld_1( ).
  CATCH cx_os_object_not_found.
  ENDTRY.
  APPEND ls_key TO lt_keys.
  " Get the service manager instance of the forwarding order and
  retrieve all the freight unit keys associated with the forwarding
  order
  lo_trq = /bobf/cl_tra_serv_mgr_factory=>get_service_manager(
  iv_bo_key = /scmtms/if_trq_c=>sc_bo_key ).
  lo_trq->retrieve_by_association(
  EXPORTING
    iv_node_key      = /scmtms/if_trq_c=>sc_node-root
    it_key           = lt_keys
    iv_association   = /scmtms/if_trq_c=>sc_association-root-tor_root
  IMPORTING
    et_target_key    = lt_tor_keys
  ).
ENDIF.

```

## Syntax

In the method CAN\_SCHEDULE\_ACTION\_BY\_ABAP, you write the following code to determine the FU\_1 and FU\_2 keys:

```

" Aim is to derive the application keys i.e the freight unit keys
" it_key holds the container key. Get the container instance and
  subsequently derive the alternate keys provided in the
APPL_KEY_4_AGGR_ACT method.
DATA: lo_container TYPE REF TO /bofu/cl_ppf_container.
" Logic only applicable for aggregate actions
IF is_ppf_act-aggr_action = 'X' AND is_ppf_act-ppf_action = 'AD_1'.
  " Get the container instance from the it_key
  READ TABLE it_key INTO ls_key INDEX 1.
  IF sy-subrc = 0.

```

```

TRY.
lo_container ?= /bofu/ca_ppf_container=>agent-
>if_os_ca_persistency~get_persistent_by_oid( ls_key-key ).
CATCH cx_os_object_not_found cx_os_class_not_found.
ENDTRY.
ENDIF.
" Retrieve the alternate key from the container instance (the
forwarding order key in our example)
TRY.
ls_key-key = lo_container->get_doc_id_fld_1( ).
CATCH cx_os_object_not_found.
ENDTRY.
APPEND ls_key TO lt_keys.
" Get the service manager instance of the forwarding order and
retrieve all the freight unit keys associated with the forwarding
order
lo_trq = /bobf/cl_tra_serv_mgr_factory=>get_service_manager(
iv_bo_key = /scmtms/if_trq_c=>sc_bo_key ).
lo_trq->retrieve_by_association(
EXPORTING
iv_node_key = /scmtms/if_trq_c=>sc_node-root
it_key      = lt_keys
iv_association = /scmtms/if_trq_c=>sc_association-root-tor_root
IMPORTING
et_target_key = lt_tor_keys
).
ENDIF.

```

## i

The aggregate feature is available only if you have implemented SP05 of SAP\_BS\_FND 7.31 or SAP Note [1706788](#).

## More Information

[Customizing](#) [Page 31]



## **4.3.4 BRFplus Applications and Functions**

If BRFplus is used to make decisions based on the business object data, you must ensure that you have access to the BRFplus application and its functions.

For example, if you want to evaluate schedule conditions, you require the BRFplus function that returns the keys for which the PPF triggers for the output can be generated.



## 4.4 BO Modifications for Integration with Output Management Adapter

### Alternative Key

Create an alternative key at node level of your business object (BOPF) for which the output is configured.

### Association Implementation

In transportation scenarios, you can maintain output at the level of both the root node of a freight order BO and the tendering request node of the same BO (see [Customizing](#) [Page 31]).

To implement this feature, you must implement root level association  
`/BOFU/CL_C_PPFOC_ROOT_XBO_RI`.

Once the root level association has been implemented, all of the node-level output content is aggregated and displayed on the output management UI at root level.

However, scenarios may be configured in such a way that it is not possible to specify multiple outputs in different BO nodes. In this case, you do not have to implement the association above. It is sufficient if association `/BOFU/CL_C_PPFOC_ROOT_XBO_NI` is implemented.



## 4.5 Configuring Direct Output

To configure direct output, you have to specify the settings for the output management adapter. Create a class that inherits `/BOFU/CL_PPF_DOA_BASE`.

### Procedure

1. Enter the BO name `/SCMTMS/TOR`.
2. Enter the node name `ROOT`.
3. Enter the agent name `SEND_EM_DATA_FROM_TOR`.
4. Enter a processor class that inherits data from `/BOFU/CL_PPF_DOA_BASE`.

In the inherited class, implement or extend method `DISPATCH_INT ()` to check the conditions.

(Method `GET_NODE_CHANGES ()` receives all of the transactional changes that have been created, updated, or deleted in the business object.)



## 4.6 Configuring the Output Management User Interface

You must carry out certain steps to display the output management user interface as a tab page on the main user interface.

### Procedure

1. Create a cross-BO association with the Output Management BO in BOPF.
2. Configure the main view on the *Variant-main* screen in UI component configuration. Use the OTR `/BOFU/PPF_ADAPTER_CONTENT_DO/OH` to name the main view as *Output Management*.
3. Configure TABBED UIBB
  - Configuration name `/BOFU/PPF_OUT_CONT_TAB`
4. Add Wire
  - Configuration name `/BOFU/PPF_OUTPUT_ROOT`
  - Source Configuration Name – UI Configuration name (UIBB configuration name based on the node containing the association to output history in the BO)
  - Port Identifier – CO (Collection)
  - Connector Class `/BOFU/CL_FBI_CONNECTOR`



## 4.7 General Recommendations and Guidelines

This section contains information about the runtime performance in your system and explains how you can ensure that your scenario is coherent and supportable.

We recommend that you redefine the methods specified under [Output Management Callbacks](#) [Page 38] according to the business case of the OM user. Doing so significantly improves the performance of the PPF if the subset of action profiles and actions can be identified by the application prior to execution.

Although it is technically possible to implement schedule conditions and start conditions in one location, it is more beneficial to implement them in their appropriate methods. On the output management user interface, for example, you can generate actions for a business document. During the generation process, only the schedule conditions are checked. If the criteria for executing output (start conditions) are also implemented in the exit for schedule conditions, actions would not be generated (scheduled) if the execution logic failed.

Customizing for the PPF is delivered with the standard SAP system. Note the following guidelines for improving your PPF Customizing settings:

- Use different PPF profiles for different business scenarios. This ensures that the runtime remains short since there are fewer conditions to check and fewer configured action definitions to load.
- Describe the action definition in terms of the expected function of the action.
- Do not develop custom code that uses standard Customizing entries.
- Model status checks as scheduling conditions and not starting conditions. Actions that meet the scheduling conditions persist until the application deletes them. In some cases, this can result in long runtimes when you execute the PPF selection report.
- A start condition should only delay the execution of the action. Any other part of the condition must be included in the scheduling condition.
- PPF is not an SAP workflow even if it can trigger a workflow. Therefore, you must avoid dependent actions. Technically, PPF cannot support dependent actions.
- Actions that affect or even delete each other can cause data inconsistencies and short dumps on the customer side.
- Do not use special characters (for example, Ä, Ü, Ö, À, and so on) in technical keys such as the technical name of an action definition.
- After the development phase, delete any unnecessary and unused action definitions apart from those used as templates for consultants. Deactivate any action definitions that are necessary but not currently used.
- Keep the conditions as simple as possible regardless of whether they are scheduling or starting conditions. This ensures that the runtime of the action determination and the selection report remains short.
- When modeling complex scenarios, document your reasons for modeling the scenario in the way you have. Your modeling process may be difficult for others to understand, and

so it is useful to provide customers, consultants, support personnel, and other developers with as much information as possible.





## 5 Testing Output Management in Your Application

The table below lists the basic steps that you must carry out to test output management in your system.

### Procedure

Step	Description	Expected Behavior
1	Create a BO instance (forwarding order or freight order). To do so, choose the <b>► Output Management ► Outputs ►</b> tab page and choose <i>Generate Actions</i> .	Relevant actions are generated on the basis of the configuration settings. A message is displayed that specifies the number of actions generated.
2	Save the BO instance.	<p>Actions with the processing time <i>Processing using selection report</i> are generated automatically but not processed.</p> <p>Actions for which the <i>Schedule Automatically</i> checkbox is not selected are generated but deactivated by default.</p>
3	Select an unprocessed action of type <i>External Communication</i> .	<p>The document is previewed on the <i>Document Preview</i> tab page.</p> <p>The language of the document can be changed.</p> <p>The output parameters for the printer (output device) can be changed on the <i>Processing Details</i> tab page.</p> <p>Copy recipients can be added, removed, or changed.</p> <p>Input help (F4) is available for both the printer and recipient fields. Check the input help for different recipient types.</p> <p>The number of copies to be printed can be changed along with the language.</p> <p>A note to be sent to copy recipients can be maintained.</p> <p>Actions can be activated, deactivated, and deleted.</p>
4	Select an unprocessed action of type <i>External Communication</i> and choose <i>Execute/Execute Immediately</i> .	The action is executed only if the start condition is fulfilled. If you choose <i>Execute Immediately</i> , the action is executed regardless of whether the start condition has been fulfilled.
5	Select a processed action of type <i>External</i>	The document number is displayed.

	<p><i>Communication.</i></p>	<p>The <i>Processing Details</i> tab page shows the details of the output device, recipients, number of copies, and so on.</p> <p>If the action is partner-dependent, the <i>Partner</i> tab page contains the partner details.</p> <p>The <i>Message Log</i> tab page contains the execution details.</p> <p>If the action is configured to archive the document, the <i>Archive Links</i> tab page contains the archive link of the sent document.</p> <p>An action can also be repeated.</p>
--	------------------------------	---



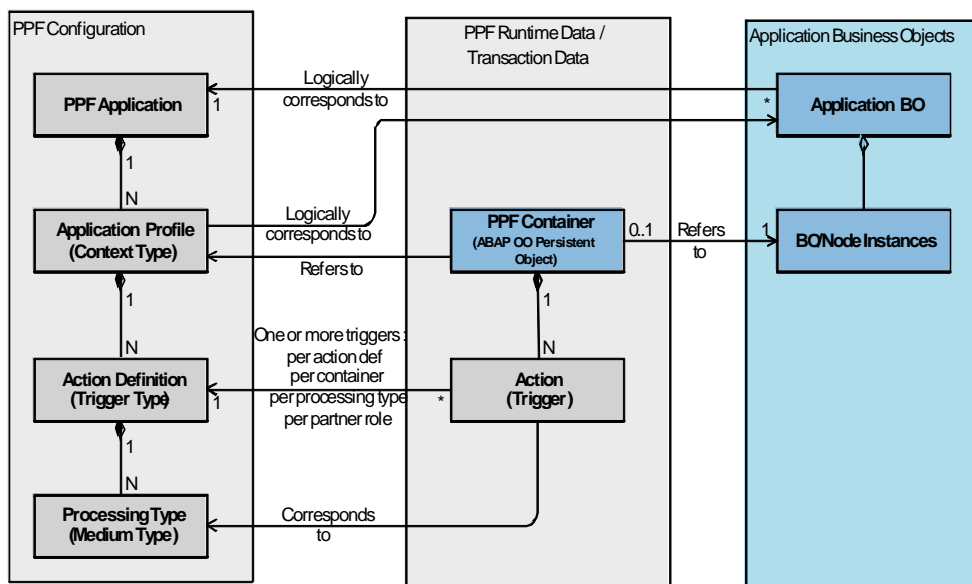
## 6 Appendix



## 6.1 Basic Introduction to the PPF

PPF is a supported framework from SAP\_BASIS that is used by multiple Business Suite applications. This section provides a brief overview of the PPF. For more information about the PPF, see the SAP NetWeaver Library under ► *SAP NetWeaver by Key Capability* ► *Application Platform by Key Capability* ► *Business Services* ► *Business Communication Services* ► *Generic Business Tools for Application Developers (BC-SRV-GBT)* ► *Post Processing Framework (BC-SRV-GBT)* ►. For detailed guidelines on how to use the PPF, see SAP Note [489334](#).

As an output management framework, the PPF helps you to configure installation-specific and customer-specific output. We can also provide you with sample implementations. The PPF generates output triggers once it has been configured for specific application data records. Based on the configuration settings, these triggers are then processed to send the actual output. The figure below shows a graphical representation of this process.



### Overview of the Post Processing Framework



## 6.1.1 Configuring Different Output Requirements

To illustrate the concepts better, the section below explains the PPF configuration objects in the following order:

1. [Processing type](#) [Page 62]
2. [Action definition](#) [Page 66]
3. [Action profile](#) [Page 67]
4. [PPF application](#) [Page 68]



### 6.1.1.1 Processing Type (Medium Type)

The PPF supports different processing types:

- External communication:
  - For print, fax, and e-mail scenarios
  - Supports PDF-based forms
  - Supports DOCX forms technology
- Alert:
  - Raising alerts using SAP NetWeaver Alert Management in a universal worklist (UWL)
- Execute method:
  - Any application-specific output processing
  - In SAP Transportation Management, this option is used to:
    - Trigger A2A/B2B outbound communication
    - Post BI content
- Print Smart Forms
- Fax Smart Forms
- E-mail Smart Forms



## **6.1.1.1.1 Setting Up Processing Types for External Communication**

### **Procedure**

To set up a processing type for printouts, e-mails, and faxes:

1. Select the print form to be used for both the cover page and main document content.
2. Specify the program module (filter-dependent BAdI implementation) to populate print forms with application data.
3. Specify default recipients for e-mail communication.
4. Specify default attachments to be sent along with the main printed document.



## 6.1.1.1.2 Setting Up Processing Types for Alerts

### Procedure

To set up a processing type for alerts:

1. Specify an alert category.
2. Specify the program module (filter-dependent BAdI implementation) to populate the alert container parameters with application data; add internal and external recipients to the alert.





### **6.1.1.1.3 Setting Up Processing Types for Method Calls**

#### **Procedure**

To set up a processing type for method calls, specify the program module (filter-dependent BAdI implementation) that contains application-specific outbound logic. In SAP Transportation Management, outbound proxies of enterprise services/BI content dispatch APIs are called from the BAdI.



## 6.1.1.2 Action Definition (Trigger Type)

An action definition groups one or more processing types. An action is a logical set of outputs than can be sent to an individual communication partner.

The following options can be configured when the action definition is set up:

- Indicate whether the action is partner-dependent or partner-independent
- For partner-dependent actions, the *Partner* role that is to be used can be specified
- Specify how transaction data action triggers are to be generated for each action definition and application data record:
  - One trigger for each action definition (for the default processing type)
  - One trigger for each action definition and partner role (for the default processing type)
  - One trigger for each action definition and processing type
  - One trigger for each action definition, partner role, and processing type
- Specify how many times a type of trigger can occur during the lifecycle of the application data record:
  - Maximum of “N” occurrences of unprocessed actions
  - Maximum of “N” occurrences of processed actions
  - Maximum of “N” occurrences of total number of actions (processed and unprocessed)
- Whether the action trigger is generated automatically or created manually
- Whether the action trigger is processed:
  - Before the commit
  - After the commit in the background process
  - Manually (selected for processing) during the report run
- Provide textual description in the action definition
- Choose a program module (filter-dependent BAdI implementation) that provides decision help when determining whether triggers are to be generated for a specific action definition
  - Evaluate Schedule Condition BAdI
- Choose a program module (filter-dependent BAdI implementation) that provides decision help when determining whether a particular trigger can send output
  - Evaluate Start Condition BAdI



### 6.1.1.3 PPF Action Profile (Context Type)

A PPF action profile aggregates the number of PPF action definitions by logically grouping them together. Multiple PPF action profiles can be created for a given application data table (or business object node in BOPF) depending on the actual output requirements.

For example, SAP Transportation Management contains one technical business object that serves multiple business uses. The TRQ category determines whether the business object is a request object or quotation object. The output requirements for a request object may differ from those of a quotation object. From a logical perspective, it may therefore be necessary to set up multiple PPF action profiles for each TRQ category. The requirements for output setup may also vary in the different BO nodes.

The following options can be configured when the PPF action profile is set up:

- Specify the connection to the BOR business object (not applicable to SAP TM or BOPF-based BOs)
- Specify the name of an ABAP OO repository class that is application-specific called *Context Class*
  - Note that PPF API functions only with an instance of a context class
  - The application-specific context class inherits data from class `CL_PPF_CONTEXT`
  - This context class instance contains:
    - Reference to the application data record – through the persistent class instance
    - Name of the PPF application
    - Name of the PPF action profile
    - Reference to the partner data collection containing communication-relevant partner identities and addresses
- Note that to use PPF API, the application has to have at least one persistent class managed by the ABAP OO object service. This enables the PPF to obtain the handle to the calling and consuming applications:
  - Generated PPF triggers choose the object ID (refer to OO object service) of this persistent class as the (logical) foreign key.
  - A persistent class can be connected to one or more application BO database tables.
  - For BOPF-based BOs, the PPF adapter provides a dummy DB table (`/BOFU/DPPFCNTR` – known as a “PPF container”). This dummy table is connected to persistent class `/BOFU/CL_PPF_CONTAINER`.



## 6.1.1.4 PPF Application

The PPF application is the application that uses the *Post Processing Framework* (PPF) to generate actions. The name of an ABAP OO PPF context class is maintained in the PPF profile. While calling the PPF using the PPF API, the instance of the PPF context class identifies the:

- Name of the PPF application
- Name of the PPF profile
- Application data record using an “ABAP OO object service Persistent class instance”. Any application that calls PPF services must use at least one dummy persistent class that is connected to application data tables.



## 6.1.2 PPF Runtime

The PPF runtime is managed by the business application that uses the PPF to call the PPF API.



## 6.1.2.1 Generating Triggers (Actions)

Based on the status of the application data record, the application determines which PPF action profile to use. It then creates an instance of the PPF context class. In the context class, the output management API transfers a reference to the application data persistent class and specifies:

- The name of the PPF profile
- The name of the PPF application
- A collection of document partners in the application data record who are eligible to receive outbound communication

The specified instance of the PPF context class `CL_MANAGER_PPF` and the PPF manager method `DETERMINE ( )` assist in generating the PPF triggers (output requirements). An instance of a trigger corresponds to an:

1. Application data record (through persistent class)
2. Action definition
3. Output type

Depending on the value of static attribute in class `CL_MANAGER_PPF`, method `LOCALE_OUTPUT`, generated triggers may be processed:

- Immediately within `DETERMINE`
- Upon saving (pending the `BEFORE-COMMIT` event from Object Services)
- After saving (pending the `AFTER-COMMIT` event from Object Services)



With the available PPF APIs, it is not possible to combine processing triggers at different points in time. For example, in a logical unit of work, you can process some triggers before the commit and other triggers after the commit.

However, you can apply mixed processing by writing application-specific trigger processing logic.



## 6.1.2.2 Processing Triggers (Actions)

When a trigger is generated, it is assigned a status. Depending on how the processing time has been configured for the action definition, the trigger is processed before or after application data is saved (before or after COMMIT).

When a trigger is processed, the start condition for BAdI evaluation is called. Based on the processing type, a suitable BAdI implementation is then called to extract business data from the application business object and supply this data to the print form or alert.



## 6.2 Design

The following section provides an example that results in a list of actions, schedule conditions, and start conditions. “All” actions for which execution is to be controlled by PPF are listed below:

What	Why	When
Send shipping list (by EDI or FAX)	Shipping list has changed	Batch job during the following night
Send claim confirmation to customer	Claim is approved	On saving
Send proposal for a follow-up contract	Always	Two weeks before the current contract expires
Send an alert to the sales manager	Sales opportunity for an important customer is lost	Immediately

PPF supports two types of conditions:

*Scheduling conditions* are checked when the PPF action is determined. The conditions specify whether an action is to be executed. The scheduling conditions are identified on the basis of the list of action definitions in the above table. To identify the scheduling conditions, we can use the entries in the “Why” column.

*Starting conditions* are checked when the action is executed, regardless of whether execution takes place immediately or upon saving.

These conditions only affect the time at which an action is executed. The “When” column indicates the technical execution time such as “on saving” or “in a background job”, as well as the business restrictions such as “Two weeks before the contract expires”. In this case, the business restrictions are relevant



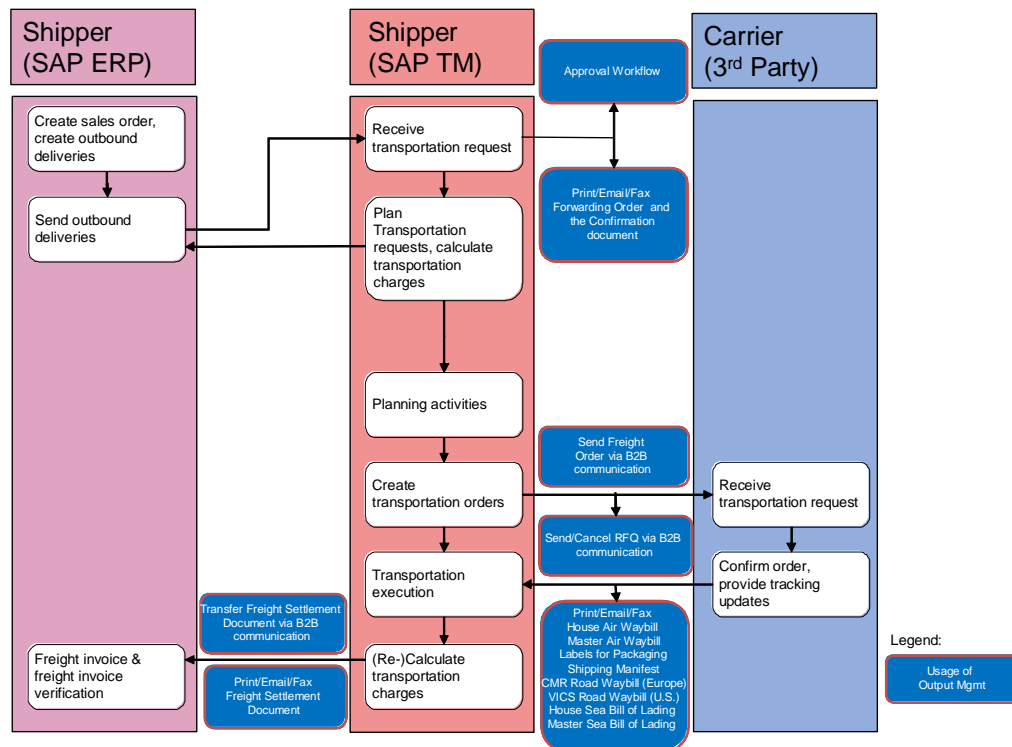


## 6.3 TM-Specific Guidelines

In addition to the connection between the PPF profile and the business object at BO node level, SAP Transportation Management (SAP TM) provides additional Customizing settings in the business document type configuration. If action profiles are configured, only the specified action profiles are used to generate actions when the PPF is called. If you select *Dynamic Determination of Output Profile*, profiles are determined based on the business scenario as specified in [Output Management Adapter Configuration](#) [Page 30].

You should take the following points into consideration when you configure for SAP TM:

- In SAP TM, PDF forms are used as standard. These forms do not support dot matrix printers. For more information, see SAP Note [1009567](#). If you want to use a dot matrix printer, you must use Smart Forms.
- A possible use of output management in an outbound shipment scenario is shown in the figure below:



### Example of Output Management in the Outbound Shipment Scenario

In Customizing for the business document type, specify the action profile setting as follows:

- In the *Output Profile* field, enter /SCMTMS/TRQ\_FWO\_PRINT.
- In the *Add. Output Profile* field, enter /SCMTMS/TRQ\_FWO.

*Authorization Concept in Output Management for SAP TM*

A `CHECK_AUTHORITY` action has been created in the BO `/BOFU/PPF_OUTPUT/CONTENT` so that output management can check authorization for objects that are specific to SAP TM. When triggers are executed using the *Execute* and *Execute Immediately* actions, the action is delegated to method `START_CONDITION ()` in the BAdI façade. Inside the method, the system checks the medium type for BCS (BCS = include PRN, MAI, and FAX) and the BO action `CHECK_AUTHORITY` is called. SAP TM can also use a plug-in class to check authorization before an action is executed. This plug-in class is also called for output management when an associated SAP TM object is involved. The call is transferred from the plug-in class to a check class specific to output management, which is specified in table `/SCMTMS/I_OBM_BO`. Execution is permitted on the basis of the authorization check performed in this class.

In addition to the standard PPF exit implementations (see [PPF Exits](#) [Page 36]), output management offers exit implementations that are specific to SAP TM. The following table provides the technical details of these exit implementations:

BAdI Name	Implementation Name	Filter Value	Generic/ SAP TM Specific	Description
EVAL_SCHEDCOND_PPF	/BOFU/PPF_STD_EV_SCH	/BOFU/EVAL_SCHEDULE_CONDITION	Generic	Used to evaluate the business determination rules, to select the actions to be generated
EVAL_STARTCOND_PPF	/BOFU/PPF_STD_EV_STA	/BOFU/EVAL_START_CONDITION	Generic	Used to evaluate the business checks, to determine when to execute the generated outputs
COMPLETE_PROC_PPF	/SCMTMS/COMPLPROCPPF	Action Profile; Action (/SCMTMS/*)	SAP TM Specific (see note below)	Used to complete processing
PRINTER_DETERM_PPF	/SCMTMS/PPF_STD_PRNT	Action (/SCMTMS/*)	SAP TM specific (see note below)	Used to identify the printer used to print the action
EXEC_METHODCALL_PPF	/BOFU/PPF_STD_METHOD	/BOFU/PPF_STANDARD	Generic	Used to execute actions with the <i>Method Call</i> processing type
GET_PARTN_ROLES_PPF	/SCMTMS/PPF_PARTN_RL	PPF Application; Profile	SAP TM specific (checked once when the PPF is	Used to retrieve the possible partner functions

			configured, which is a separate implementation. See note below.)	of the application
DOC_PERSONALIZE_BCS	/BOFU/PPF_STD_DOCBCS	/BOFU/PPF_STANDARD	Generic	Used to retrieve data from the business object and personalize the document
ALERT_EXIT_LOCAL_PPF	/BOFU/PPF_STD_ALERT	Action Profile, Action	SAP TM specific (see note below)	Used to read application data for an alert

# 1

For customer-specific BAdI implementations with filter values other than those in /SCMTMS/\*, the BAdI implementation must contain the code used in the corresponding SCMTMS implementation (the call to the BAdI façade and, therefore, the call to the agent class).