



Modeling Guide for PPF

Contents

1	Document Organization	3
1.1	Authors	3
1.2	Intended Group of Readers	3
1.3	References	3
1.4	Glossary	4
2	Modeling Guidelines - Application Analysis	6
2.1	What, Why, When ...?	6
2.2	Translation into PPF Terms	7
2.2.1	Definition of Tasks - Action Definitions & Processing Types	7
2.2.2	Grouping of Tasks - Action Profiles	8
2.2.3	Conditions	9
2.3	Communication between Application and PPF	11
2.3.1	Context	11
2.3.2	Application Object	11
3	General Hints.....	14
3.1	Rules for Setting up a Good Customizing	14
3.2	Delivering Customizing	14
3.3	Reorganisation of Actions	14

1 Document Organization

1.1 Authors

Name	Company	Project Role(s) / Comments
IMS Technology	SAP AG	Software Developer and Author

1.2 Intended Group of Readers

This document is meant as extension of the PPF Implementation Guide, see 1.3. The Implementation Guide focuses on the technical aspect of the implementation and offers a step-by-step instruction for implementing important PPF features. In contrast, this document aims to provide you with a better understanding of the modeling process and its elements.

This document is, especially, intended for

- Application developers, who are integrating PPF into their application
- Application developers, who are setting up or changing SAP-customizing.

The information might be also useful for the following groups:

- Support
- Consultants

1.3 References

Please also see the Post Processing Framework Implementation Guide on how to apply the PPF to a new application.

1.4 Glossary

Term	Definition
PPF	<p>Abbreviation for Post Processing Framework. This framework provides SAP-applications with a standardized way to execute and administrate conditioned business tasks.</p> <p>In contrast to the SAP Business Workflow, these business tasks are mutual independent, i.e. there are no means for a process flow.</p>
Action Definition	<p>An action definition is meta data or skeleton of a business task. The action definition encompasses all its possible realizations, see also processing type.</p> <p>Additionally, it carries the information if business partner data is needed for the execution and to which business partner it corresponds. Within the action definition, you setup the merge technique, i.e. how many successful, failed, or unprocessed actions are allowed for a particular action definition.</p>
Action	This is a runtime instance of an action definition
Configured Action Definition	<p>For action creation during action determination, the action definition must be configured. The triple of a chosen processing type, scheduling and starting condition is called configured action definition.</p> <p>An action definition can have several configured instances. Moreover, almost all technical settings of the action definition can be differently configured.</p>
Processing Type	The processing type corresponds to the technical realization of an action definition . For instance, start a workflow, send a PDF-based form via email, etc.
Action Profile	Smallest set of action definitions that can be investigated during one call of the action determination of the PPF.
Application Object	The application object is the standardized proxy to access the business object . The persistence of this object must be realized via the Object Services.
Business Object	Object that carries the main data relevant for the action determination and execution. Its persistence is either given as a BOR object or as a persistent class of the Object Services.
Context	Set of data handed to the action determination . The context encompasses the name of the application and the action profile , the application object and a collection of partners.
Partner	<p>Action definitions can be assigned to special person groups. In order to identify these groups, the PPF expects all address data enriched with a name of partner function, e.g. "Responsible Person", "Customer", and, optionally, a partner number. The object having all these attributes is called a partner.</p> <p>The partner functions are defined via a BAdI from the application.</p>
Partner Collection	This is the runtime collection of all partners of a business object. The collection is part of the context .

Merge Technique	<p>During the action determination, the new actions are merged with the old actions of an action definition. Which merge technique is used, is customized on action definition level. For instance, the merge technique "1 successful action per action definition" deletes all newly determined actions, if one action of the given action definition was successfully executed.</p>
Action Determination	<p>This is the runtime interface of the PPF.</p> <p>When the action determination is invoked, the PPF checks the scheduling conditions of all configured action definitions based on the data provided by the context. Moreover, a partner determination is performed for partner dependent action definitions, i.e. a partner of the customized partner functions are added whenever they are provided with the partner collection. Otherwise, a partner dependent action will be deleted. Finally, the customized merge technique is invoked.</p> <p>The administration of the execution is automatically updated for newly generated actions, too.</p>

2 Modeling Guidelines - Application Analysis

Administrating actions in a dynamical, modifiable and flexible way is an ever reoccurring challenge. One possibility of integrating a flexible output control into your application is using the **Post Processing Framework** (short PPF).

In order to successfully integrate PPF into your application, there is some need for a data modeling process before you start implementing the necessary code. The data model that you chose will heavily influence the expandability, flexibility, performance and supportability of your solution. The modelling process for PPF is not a very hard or complicated one. Nevertheless, a too sloppy approach inhibits further enhancement or raises problem whenever the customer wants to adapt the setting to his needs. This document assists you during that process and provides you with proven concepts and hints.

The implied application analysis can be a wide field. Here, we only address mutually independent actions that your application wants to trigger under various circumstances. For dependent actions, you should consider to integrate a workflow into your scenario.

As a starting point for the modelling, we have chosen your application, since this is the area which you know best. Building up a list of actions of your application will be the first step. Based on this list, we explain the analogous terms and entities of PPF. We describe the translation process as algorithmically as possible. Finally, we explain in which way your application parses information to the PPF at runtime.

Note: Before discussing your integration with the PPF colleagues, you should have worked through the section 2.1.

2.1 What, Why, When ...?

You start by setting up a list of “all” actions whose execution should be controlled by PPF. We strongly suggest investigating some effort into that step. The related effort will pay off, definitely. The following three questions serve as a good starting point.

- **What** should be done? E.g. send the shipping list to customer (via FAX and/or IDOC etc.), create a follow up order, ...
- **Why** or under which conditions should an action be executed? E.g. always, when the order is complete, when the user selects it from his work list, when an email address is provided ...
- **When** should it be executed? There are technical execution times, e.g. immediately, at the end of the application controlled local unit of work (short LUW), in a batch job, and there are business relevant execution times as “two weeks before the contract expires” or “at the end of the year” ...

What	Why	When
Send shipping list (via EDI or FAX)	Shipping list has changed	Batch job of the following night
Send claim confirmation to customer	Claim is approved	Save
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 import customer is lost	Immediately
...		

Hint: In order to arrive at an almost complete list of actions, it is useful to discuss or to analyse your application from the following view point.

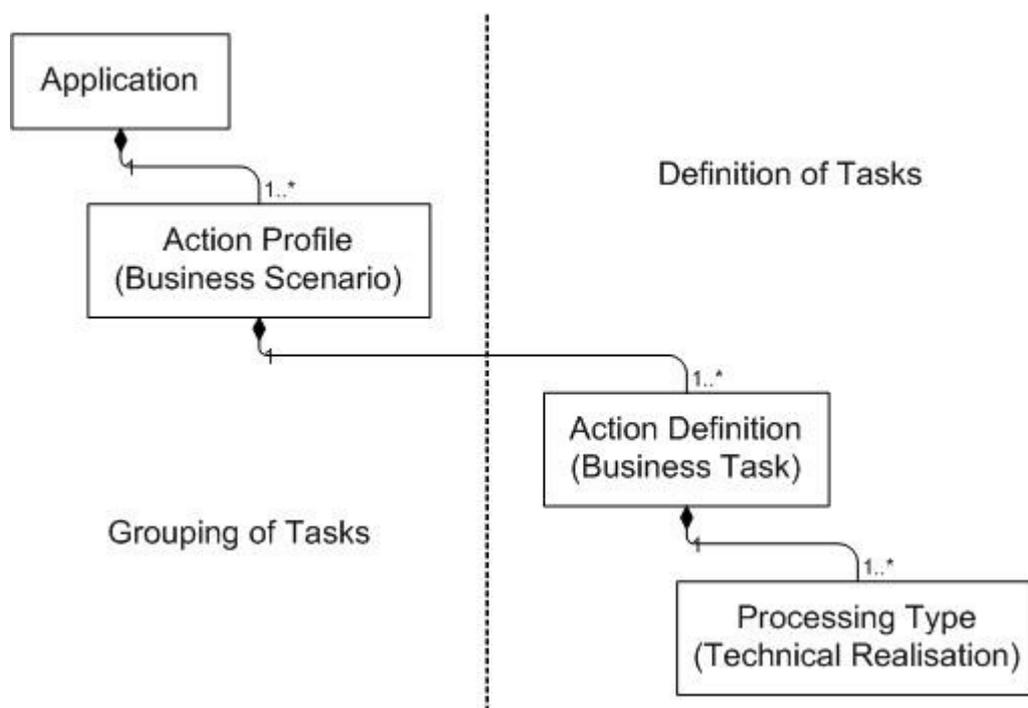
- What is expected to happen during user interaction?
- What should happen when the application is in batch processing mode?
- What might the customer want to change, add, or delete, in addition to our scenario?

2.2 Translation into PPF Terms

We have not stressed an order or hierarchy of the action list. Hence the above list can be very huge and unstructured. Now, we set up a hierarchy that will serve as a one-to-one template for setting up the PPF customizing. For the moment, we do not care about the business conditions attached to a business task. Meaning, we are only interested, for instance, if a “sales order has to be send to a business partner” and not that it is has to be complete. The set up of the conditions will be dealt with in the subsection 2.2.3.

The process for setting up the hierarchy consists of two steps:

- Definition of the tasks
- Grouping of the tasks



2.2.1 Definition of Tasks - Action Definitions & Processing Types

An action definition has the character of a business task, which might have different technical realisation, e.g. send a fax, an email, or print and send a letter. Within PPF, such technical realisations are called processing types.

Having the action list at hand, it is fairly easy to arrive at the list of action definitions.

1. You take all the entries in the column “What” of your action list.
2. You ignore the technical way in which the task is completed. This gives you a business task. The ignored techniques provide the corresponding list of technical realisations of a business task.
3. You list the technical ways
4. You delete or merge the all duplicates business tasks.

Example: Take the action list below.

What	Why	When
Send whole shipping list via EDI	Shipping list has been created	Next batch job
Send whole shipping list via fax	Shipping list has been changed	When saving the document
Send shipping list changes via fax	Shipping list has been changed	When saving the document

Applying the above algorithm to above list, we obtain the following action definition and processing types.

Action Definition	Processing Type
Send whole shipping	Method call (starting the EDI communication)
	External Communication - Channel "Fax"
Send shipping list changes	External Communication - Channel "Fax"

2.2.2 Grouping of Tasks - Action Profiles

In general, not all the business tasks belong to the same business scenario. For instance, "Sending a delivery avis" belongs to another scenario as "Sending an order cancellation".

Unfortunately, we are not aware of an algorithm as in the previous identification of the action definitions. Nevertheless, your application provides you, in a natural way, with good candidates for the action profiles. We have collected some starting points, which might be helpful for grouping the action definitions.

- Probably, your business process has several phases. These phases can serve as your action profiles.
- Or, there might be two ways in which your application is called, i.e. background mode and user interaction.
- Eventually, your service interacts with other application. This often leads to a distinction of the business scenario. Therefore, a distinction in your PPF modelling occurs almost naturally.

Sometimes you can find two or more ways for modelling your application, as the next example illustrates. Which way you choose depends on your needs and the restriction of the application.

Example: Let's assume there is an attribute or state of your business object checked in every action and that we determined the following list of actions as described in 2.1.

What	Why	When
Do something important 1	It's Monday and the business object is in state 'A'	Next batch job
Do something important 2	Business object is in state 'A'	When saving the document
Do something important 3	It is not Monday and the business object is in state 'A'	Next batch job
Do something trivial 1	It's Monday and the business object is in 'B'	When saving the document
Do something trivial 2	Business object is in state 'B'	When saving the document

This list can be ordered in different ways.

Here is the first one, with no optimisations. Everything is checked every time.

Action Profile	Action Definition	Why
All in One	Do something important 1	It's Monday and the business object is in state 'A'
	Do something important 2	Business object is in state 'A'
	Do something important 3	It is not Monday and the business object is in state 'A'
	Do something trivial 1	It's Monday and the business object is in 'B'
	Do something trivial 2	Business object is in state 'B'

The second approach takes care of the states.

Action Profile	Action Definition	Why
State A	Do something important 1	It's Monday
	Do something important 2	
	Do something important 3	It is not Monday
State B	Do something trivial 1	It's Monday
	Do something trivial 2	

The third one uses the reuse of action definitions and uses include action profile.

Action Profile	Action Definition	Why
"Anytime"	Do something important 2	Business object is in state 'A'
	Do something trivial 2	Business object is in state 'B'
"Monday" including "Anytime"	Do something important 1	Business object is in state 'A'
	Do something trivial 1	Business object is in state 'B'
'Not Monday' including "Anytime"	Do something important 3	Business object is in state 'A'

Note: The related discussion takes place in your application group. A help request to the PPF group for modelling should only focus technical subjects, e.g. possibilities for reuse, performance impacts of different solutions etc.

In case that you found further useful starting points during your discussions, you can contribute to the list above. We are more than happy to add your strategy for figuring out the action profiles.

2.2.3 Conditions

Having identified the action definitions from the action list of section 2.1, we are now focusing on the conditions in more detailed. Moreover, the previous investigation of the action profiles illustrates the connection between the different parts. In order to get a better understanding of the conditions, itself, we ignore these connections for a moment.

PPF supports two types of conditions.

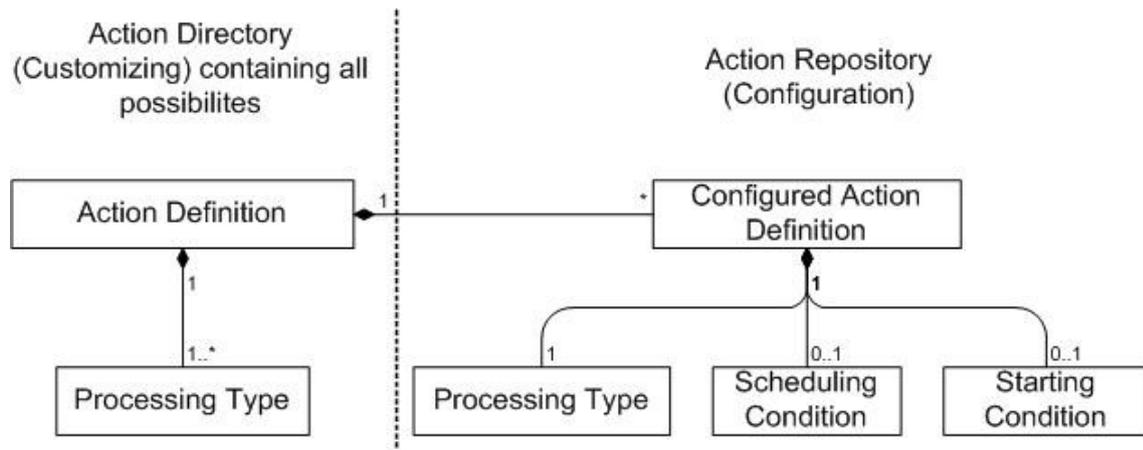
- **Scheduling conditions** are checked during the PPF action determination. They decide whether an action should be executed or not. The identification of the scheduling conditions is based on the list of all action definitions obtained in section 2.1. For identifying the scheduling conditions, we can use the entries of the "Why" column.

- Starting conditions** are checked, when just before the execution of an action will be started. These conditions should only influence the point in time when an action will be executed. The “When” column corresponds either to the technical execution time, e.g. “on save” or “in a background job”, or to business restrictions, as for instance “2 weeks before the contracts expires”. The later ones are the starting conditions that we are looking for.

In which way are conditions attached to the Action Definitions?

Before setting up the conditions from our initial action list, some words on the customizing and configuration will be helpful.

Remember, the action definitions and its processing types were identified by simplifying the actions of our action list. This implies that one action definition can belong to several actions of that list. Now, you have to look at the reverse direction. This assignment is realized by configuring your action definition. The entities storing that information are called configured action definitions.



A configured action definition is an object that references to its skeleton – its action definition. Within the configured action definition it is possible to change almost all properties of the action definition, e.g. select another execution time. Additionally, one of the processing types offered by the action definition as well as the scheduling and starting conditions are assigned to it.

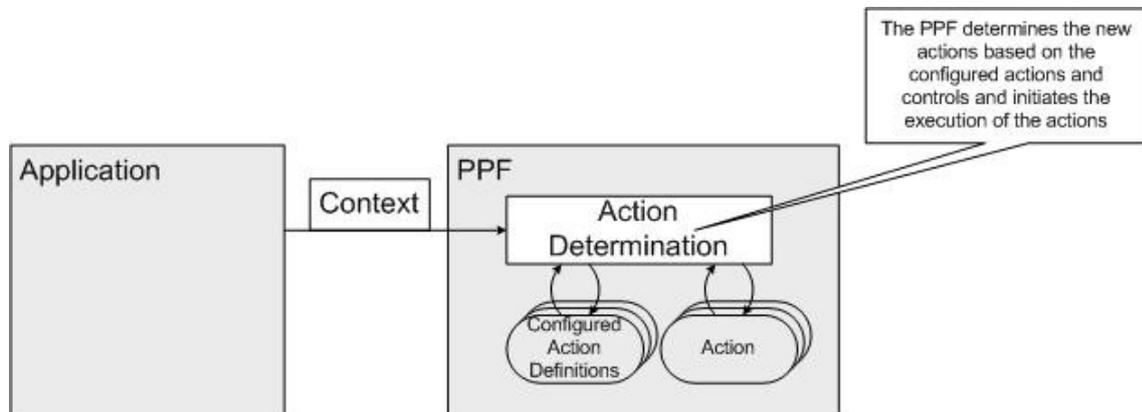
Now, it's time for an example.

Example Recall the example of subsection 2.2.1. The scenario leads to the following list of configured action definitions.

Used Action Definition	Changed Attributes	Used Processing Type	Scheduling Condition	Starting Condition
Send whole shipping	Technical execution time (Batch)	Method Call (EDI)	Shipping list is created	None
Send whole shipping		External Comm. - Fax	Shipping list is changed	None
Send shipping list changes		External Comm. - Fax	Shipping list is changed	None

2.3 Communication between Application and PPF

We recall the general runtime interaction between your application and the PPF. The PPF manages the determination and the execution of the action for your application. In order to accomplish that, the application has to parse some information to the PPF.



Based upon the context information, the PPF selects the corresponding configured action definitions. That data leads to new actions which were merged with eventually existing ones. Hence a change in the set of actions for the application object is effected.

2.3.1 Context

The central question to answer is as follows: What needs to be transferred to the PPF? Or, how does the context look like?

The next parts make up the context that is handed to the PPF for action determination.

- Name of your application.
- Action profile, in order to determine the actions this business scenario.
- The application object for which the actions should be determined.
- Collection of partners related to the business object.

The name of your application and the action profile serve to identify the set of configured action definitions whose conditions will be evaluated, see the hierarchy figure in subsection 2.2.2.

The partner collection is not discussed here. For a detailed treatment of this subject we refer to the PPF Implementation Guide (see 1.3).

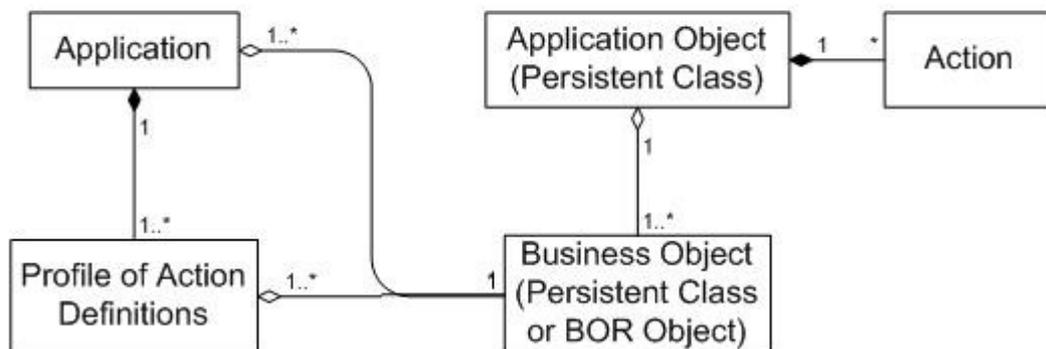
To get a clear vision of the application object, we give a short overview on its main tasks and occurrences.

- Evaluation of the conditions attached to the action definition.
- Action execution regardless whether this takes place during the application processing or in a batch job a year later.
- Some BAdIs during the action determination.

2.3.2 Application Object

Let us first give an architectural overview on the application object. In order to handle all applications in similar fashion, PPF expects the application object to apply to certain standards.

Hierarchy of the Action Definitions



Standard: The application object must be an instance of a persistent ABAP OO-class!

In general, a persistent class has a one-to-one correspondence to a database table. Its instances correspond to entries of that table and attributes correspond to fields of the table entry. Shortly, a persistent class is an object oriented, standardized encapsulation of a database table. For a detailed introduction and more advanced encapsulations, i.e. addressing more than one table with only one class, we refer to the corresponding documents of the Object Services.

Standard: Application object must be identifiable via a GUID!

As mentioned above, the PPF treats all applications the same. Therefore, the keys to identify an instance of an application class must have the same format. This is realized by a GUID. Hence there must be mapping between your business key and a GUID. This GUID must be a field of the structure for persistence mapping and marked as object identifier for the persistent class.

Standard: An interface for returning the business object must be implemented!

Depending on the situation in your application, you will run into one of the following possibilities.

- The business object for which the action determination is need is already modelled as a BOR object. In that case, you have to implement the interface `IF_BOR_OBJECT_PPF`.
- The business object is modelled as a persistent class, but the model can not be extended by a GUID. Then the application object should implement the interface `IF_BUSINESS_OBJECT_PPF` returning the corresponding instance.
- The business object is modelled as a persistent class and the model can be extended by a GUID. Then you reuse this class as the application class, too. You have to implement the interface `IF_BUSINESS_OBJECT_PPF` in a trivial way.
- There is no business object, yet. You can choose either you prefer a BOR object or a persistent class. Preferable, the business object and the application object for the PPF should be one. Then you can proceed as in the previous case.

Attributes of the Business Object

The workflow condition editor, with which you can configure your scheduling and starting conditions, only offers attributes of the business object (BOR or persistent class) for the formulation of a condition. Hence you should make an analysis of your conditions. All used elements, beside static parameters or dates, should become attributes of your business object.

In case that you cannot extend your business object, e.g. some development directions of your group, there exists the possibility to use BAdI-conditions. This condition technology is based on coding. With this kind of condition, you can use all data being accessible from your business object.

Although BAdI-conditions are a very powerful tool, you should keep in mind that BAdI implementations are not as transparent as conditions modelled by the graphical condition editor. With the workflow condition editor, you model your condition whereas you code them in the BAdI situation. This has some implications on the life cycle of your PPF integration.

- Error analysis and the investigation of the business process are easier when there is less coding to analyze.
- Implemented Conditions, which you deliver to the customer, generally serve as a template for consultants. Hence it is very likely, that there will be even more BAdI implementation at customer side.

3 General Hints

3.1 Rules for Setting up a Good Customizing

In this section, we focus on performance of the runtime, a coherent and supportable scenario. That is what we mean by a “good” customizing

- Use different PPF profiles for different business scenarios. This keeps the runtime fast, since there are fewer conditions to check and less configured action definitions to load.
- Do not program on customizing entries. See section 3.2.
- Status checks should be modelled as scheduling conditions and not as starting conditions. Actions with a fulfilled scheduling condition are persisted and exist until the application initiates their deletion. In some cases, this can lead to unnecessary long runs of the PPF selection report.

A start condition should only cause a delayed execution of the action. Any other part of the condition must be part of the scheduling condition.

- PPF is not SAP-Workflow, even if it able to start a workflow. Therefore, you must avoid dependent actions. Technically, there are no means within PPF to support dependent actions.

Actions affecting or deleting each other can cause data inconsistencies and short dumps on customer side.

- Do not use special characters as Ä, Ü, Ö, Å ...in technical keys. For example the technical name of an action definition is such a key.
- After the developing phase, you should delete all unnecessary and unused action definitions beside those serving as a template for consultants. Necessary but, temporarily, unused action definitions should be deactivated.
- Keep the conditions as simple as possible, irrespective if it is a scheduling or a starting condition. This keeps the action determination and the selection report fast.
- When modelling complex scenarios do not forget to document why you have chosen to do it that way. It might be a way, which is, possibly, hard to understand. The customer, the consultant, the supporter, the following developer, they all will appreciate this information.

3.2 Delivering Customizing

If the customer wants to change the SAP-standard PPF-customizing of an application, e.g. inactivate an action definition, change the printed form, the following way is proposed.

Application Side:

- The application must provide a place inside its customizing where the action profile used in a specific scenario can be changed. **Do not program on PPF customizing entries.**
- During runtime, the application has to read the customized name of the PPF-profile. This name should be used for building up the context that is handed over to the PPF-determination of the PPF-manager.

Customer Side:

- The customer has to copy the corresponding action profile into his name space. This step ensures that the changes will not be overwritten by delivered SAP customizing in forthcoming support packages.
- Enter the name of his new profile into the application customizing described above.
- Now, he can change due to his demands.

3.3 Reorganisation of Actions

Although the action creation and execution is controlled by PPF, they belong to an application object. Please keep in mind, that the deletion of a successful action has an influence on the next run of the action determination. Therefore the complete control is handed to the application. In case that an application wants to delete actions, the class `CL_MANAGER_PPF` provide the following method `DELETE_ALL_TRIGGERS_FOR_OBJECT`. You only have to supply a reference to you application object for which all its actions should be deleted.

Example: The action “Send invoice” belonging to the application object “Order 14532” was successfully sent. If you delete that action and open “Order 14532” in change mode, an action determination is started. Since no successful action “Send invoice” exists, a new action of the same type is created and send to the customer!