

assentis
...
...

Shipment Management Guide

Assentis:DocFamily

Shipment Management Guide: Assentis:DocFamily

Version 6.7.0

Copyright © 2020 Assentis Technologies AG, Blegistr. 1, 6343 Rotkreuz, Switzerland.

Abstract

The Shipment Management feature allows processing of multiple documents, sent to multiple recipients, over different channels, within a single business event. It extends the scope of DocFamily beyond interactive editing of single documents. Shipment Management is relevant to customers who have a need to formalize the process of communicating in an ordered manner with their customers when more than one document or recipient is involved. Claims management, client on-boarding and shipment document distribution are examples of business processes that are naturally suited to the Shipment Management feature.

This document describes the features and usage of the system by introducing various roles. It also references technical documentation with details on APIs and XML schemas.

While Shipment Management reduces the process complexity for the system end-user, its configuration may be quite complex due to its power and number of options. Thus, it is highly recommended that professional services support and / or training are provided by Assentis during the initial on-site integration phases.

Legal Notice

THIS DOCUMENTATION IS PROVIDED FOR REFERENCE PURPOSES ONLY AND COPYRIGHT PROTECTED (C) BY ASSENTIS TECHNOLOGIES. COPYRIGHTS FOR COMPONENTS OF THIS WORK OWNED BY OTHERS THAN ASSENTIS TECHNOLOGIES MUST BE HONORED. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS DOCUMENTATION, THIS DOCUMENTATION IS PROVIDED "AS IS" WITHOUT ANY WARRANTY WHATSOEVER AND TO THE MAXIMUM EXTENT PERMITTED, ASSENTIS TECHNOLOGIES DISCLAIMS ALL IMPLIED WARRANTIES, INCLUDING WITHOUT LIMITATION THE IMPLIED WARRANTIES OF MERCHANTABILITY, NONINFRINGEMENT AND FITNESS FOR A PARTICULAR PURPOSE, WITH RESPECT TO THE SAME. ASSENTIS TECHNOLOGIES SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION, DIRECT, INDIRECT, CONSEQUENTIAL OR INCIDENTAL DAMAGES, ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS DOCUMENTATION OR ANY OTHER DOCUMENTATION. NOTWITHSTANDING ANYTHING TO THE CONTRARY, NOTHING CONTAINED IN THIS DOCUMENTATION OR ANY OTHER DOCUMENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM ASSENTIS TECHNOLOGIES (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF THE APPLICABLE LICENSE AGREEMENT GOVERNING THE USE OF THIS SOFTWARE.

Table of Contents

1. Introduction	1
1.1. Prerequisites	1
1.2. Further reading	1
1.3. Enabling Shipment Management	2
2. Functionality	3
2.1. Business background	3
2.2. Shipment Management in DocFamily	3
2.2.1. Shipments, business events and recipients	4
2.2.2. Channels	4
2.2.3. User interaction and restrictions	5
2.3. Definitions vs. instances of objects	5
2.4. Flexibility of Shipment Management	6
2.4.1. Dynamic behaviour and custom data interaction	6
2.4.2. Customization and extensions	7
2.5. Roles	8
3. Configurator view	10
3.1. Basic principles	10
3.1.1. Referencing and changeability	10
3.1.2. Labels	11
3.1.3. Rules	12
3.1.4. Custom Data	17
3.2. Channel definition	18
3.3. Recipient definition	19
3.4. Event definition	19
3.5. Shipments	21
3.6. Papers and paper assignments	22
3.6.1. Assignments	22
3.6.2. Documents	25
3.6.3. Carbon Copies	25
3.6.4. Attachments	26
3.7. Recipient assignment	26
3.8. Custom Data preconfiguration	27
3.9. Document pool	27
3.10. Workflow on business event templates	30
4. Business user view	34
4.1. Staffer	34
4.1.1. Creating an event	34
4.1.2. Event	34
4.1.3. Shipment	35
4.1.4. Shipment repetition	39
4.1.5. Document pool	40
4.1.6. Recalculation	41
4.1.7. Preview	41
4.2. Workflow on business events	42
5. Integration options	44
5.1. Designing document templates	44
5.1.1. Document template creation in DocDesign	46
5.1.2. DocWrite Template preparation in DocAdmin	46
5.2. Data processing	47
5.2.1. Data processing without shipment	47
5.2.2. Shipment data processing	48
5.2.3. Shipment based serial letter processing	49
5.3. Sending Shipments as email with attachments	52
5.3.1. Preliminaries	52
5.3.2. Document action <code>textsystemPreviewEhtml</code>	52

5.3.3. Extension of global action <code>uploadToRepoAsXep</code>	53
5.3.4. Email channel configuration	54
5.3.5. Shipment based email processing	54
5.3.6. Providing sender and recipient information	55
5.4. Formatting Shipments with PDF attachments to AFP	55

List of Figures

3.1. Paper assignment configuration	23
3.2. Paper configuration	24
3.3. Extension of the workflow definition	31
3.4. "Workflow" entry in context menue	31
3.5. Specific assignment in DocAdmin	32
3.6. Specific assignment in DocWrite	32
3.7. Workflow controls in the business event template	33
4.1. Recalculation button	41
4.2. Assigning action in business event template	42
4.3. Workflow controls in the business event	43
5.1. DocWrite Template Editor	47

List of Tables

2.1. Objects with definitions:	6
3.1. Allow insertion of [object] - contained in:	14
3.2. Allow insertion of [object] - properties:	14
3.3. Specific [object] - contained in:	15
3.4. Specific [object] - properties:	15
3.5. Repetition of a specific [object] - contained in:	16
3.6. Repetition of a specific [object] - properties:	16
3.7.	29
4.1. Carbon copies in repetitions	40
5.1. List of system variables for events	45

Chapter

1

Introduction

This document gives inside into Shipment Management and its implementation in DocFamily. Shipment Management is an overall functionality.

1.1. Prerequisites

Shipment Management requires the installation and configuration of:

- DocRepo
- DocBase
- DocWrite

For configuration purpose DocDesign and DocAdmin have to be operated, and for monitoring DocAdmin Console can be used.

1.2. Further reading

Correspondingly to the operation of a full DocFamily installation, the lecture of DocFamily base documentation is recommended to obtain the necessary information to prepare, configure and use Shipment Management.

Both, staffers and configurators, will find the document “[DocWrite User Guide](#)” helpful while working with the DocWrite UI.

Configurators and integrators should refer to “[DocDesign Online Help](#)” and “[DocAdmin online Help](#)” when configuring document templates respectively DocWrite Templates.

Extended technical information for integrators can be found in the following documents:

- [DocWrite Development Guide](#)
- [TextSystem Development Guide](#)
- [DocBase Cookbook](#)
- [DocBase Tasks Reference](#)

- *DocBase Post Processing Guide*
- ComFoundation documentation

1.3. Enabling Shipment Management

Prerequisite to the activation of Shipment Management is the proper installation and configuration of DocFamily according the instructions of *DocFamily Installation Guide*.

Shipment Management is a standard part of DocFamily and is activated in the DocWrite configuration. The Shipment Management-related settings can be found in the /DocWriteConfig/ShipmentConfig section of the docwriteconfig.xml file.

Make sure the functionality in general is enabled by setting /DocWriteConfig/ShipmentConfig/UseShipment to true:

```
<UseShipment>true</UseShipment>
```

Then all shipment related objects are visible in the DocWrite UI.

For more information of further configuration options in the DocWrite read the inline documentation of the configuration file.

Chapter

2

Functionality

This chapter provides a brief overview of the main requirements in Shipment Management. It introduces the major functional areas and describes the principles of Shipment Management but is not meant to provide an exhaustive implementation guide.

2.1. Business background

The term “Shipment Management” is abstract and can be interpreted in many ways.

Output management systems often focus on offering services for producing single documents which may be produced in batch runs but do not however relate to one another.

Real world business cases on the other hand are rarely **limited** to producing a single, isolated document sent to one recipient over a single channel. More often than not there is a need to:

- produce collections of different, lightly related documents
- send them to multiple recipients
- over different channels
- within a single business event



Note

Customers adopting traditional OMS systems usually cope with this situation by developing custom code “around” the OMS to take care of the lacking functionality. More often than not, these systems are difficult to build, inflexible and expensive to maintain. If the OMS is based on proprietary technology it is usually difficult to interface with and the custom code may not always produce the required results.

The following sections offer a more detailed view of the shipment functionality using concrete examples to underpin the theory.

2.2. Shipment Management in DocFamily

Shipment Management in DocFamily provides the capability to configure a complete business case in the form of a Business Event Template. It may include several Shipments which are combinations of related documents. Shipments can be addressed to one or more recipients and

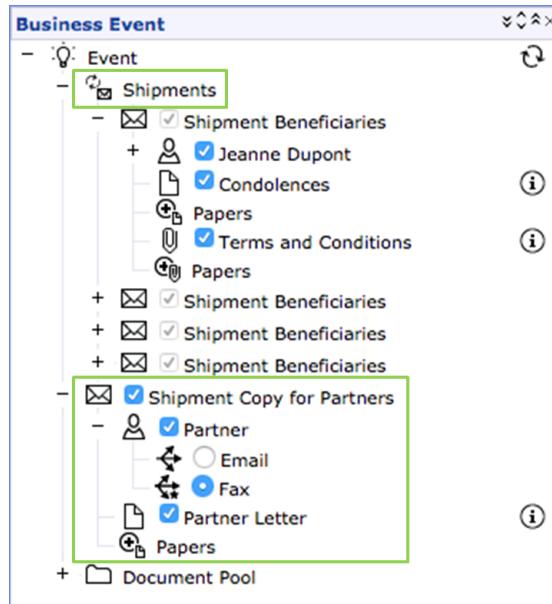
sent over multiple channels. Recipients and channels can both be modelled with Shipment Management.

For producing output documents, Business Event Instances are handled by Post Processing, packaging Shipments according to their recipients and distributing them as defined for the channel, in the format specified.

2.2.1. Shipments, business events and recipients

A shipment defines a collection of documents sent to one or more recipients through a single channel. A business event contains one or more shipments and includes a pool of documents and attachments that may be used in the context of the event.

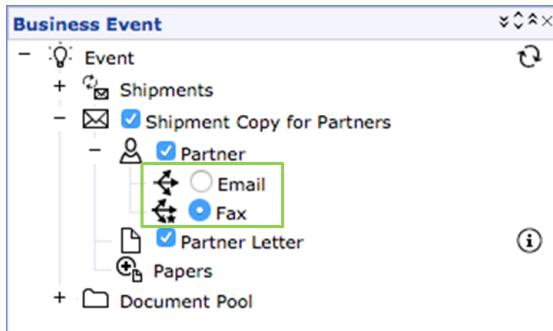
A typical example of a business event that would require the shipment functionality can be taken from the insurance branch, namely, the death of a policy holder. The insurance company informs each of the individuals listed as beneficiaries on the policy that they are beneficiaries and of the corresponding payout that they are entitled to receive. In addition, the original issuer of the policy may be a subsidiary of the insurance company and it may therefore be expected that the responsible representative also receives a copy of all the documents sent to the beneficiaries. The business event would thus include a shipment to each beneficiary plus a shipment to the subsidiary containing a copy of the shipments sent to the beneficiaries.



2.2.2. Channels

A shipment is always sent to its recipients over one channel. Sometimes the channel is predefined and sometimes it may be chosen by the staffer. In addition recipients often have a preferred channel, which should be respected whenever possible. However, there may be legal or compliance issues that require a document to be sent over a specific channel. These issues may be further impacted based on whether an original or a copy of the document is being sent.

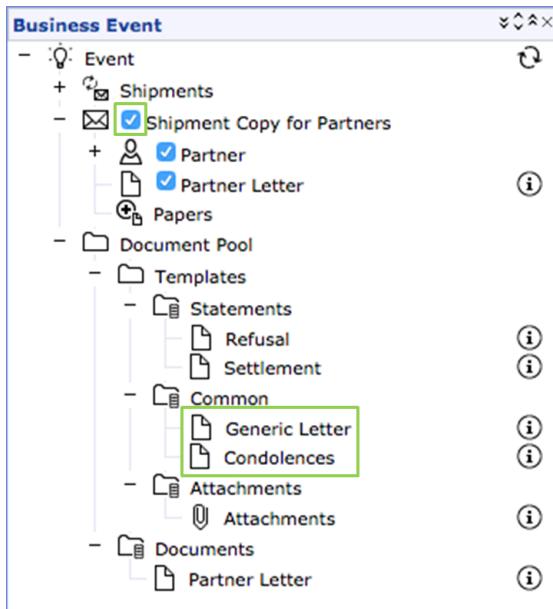
Beneficiary notifications might for example be required to be sent via physical mail in certain legislations while it may be perfectly fine to send the copy of the notifications via email.



2.2.3. User interaction and restrictions

Often the processing of single documents happens without user interaction and most organisations attempt to minimize the number of documents that require processing by a staffer. This is true for both events and shipments. Ideally, they are built and processed automatically. However, in most cases there is still a need for user interaction. The degree of interaction may in fact be very heavy for business events which are loosely defined and require a great level of judgement and flexibility by the staffer. It is therefore important, that events and shipments are configurable to be either processed automatically or with user interaction. It is also often necessary, to restrict the options a staffer has when modifying a shipment. The options may even depend on business data or the user's role.

In the case of a life insurance policy, users might require the option to include or exclude a specific shipment from the business event or to define documents that are allowed to be attached to a pre-defined document in the shipment. This requires the shipment event model to allow user actions to be based on roles.



2.3. Definitions vs. instances of objects

This section provides insight into the differences between the definition of an object (e.g. an event) and its actual instance. Thereby a definition is to understand as a template and an instance as a concrete element created from said template.

An event is created by defining its shipments, the channels, the recipients, allocating the corresponding documents etc. This can become time-consuming and sometimes quite complex for

end-users. Therefore the Shipment Management offers the option to configure “event definitions”. An event definition (icon: ) contains basic structures for a specific family of events (icon: ) and can be used like a template. Instead of defining every event from scratch, events are created based on event definitions. The same holds true for channels and recipients.

Event definitions can simply define a structure describing the shipments and whether or not they may be selected, it may not necessarily contain any actual data. An event may also define a complete event including all of its data. While it makes little sense to define a *complete* event without offering any end-user interaction, it may however be useful to configure a template that defines a fairly complete shipment and allows the end-user to make minor adaptations.



Note

The usage of definitions is a basic concept in Shipment Management. It will be explained in more detail in Chapter 4, *Business user view*.

Table 2.1. Objects with definitions:

Object	Definition	Instance
Event		
Shipment		
Recipient		
Channel		

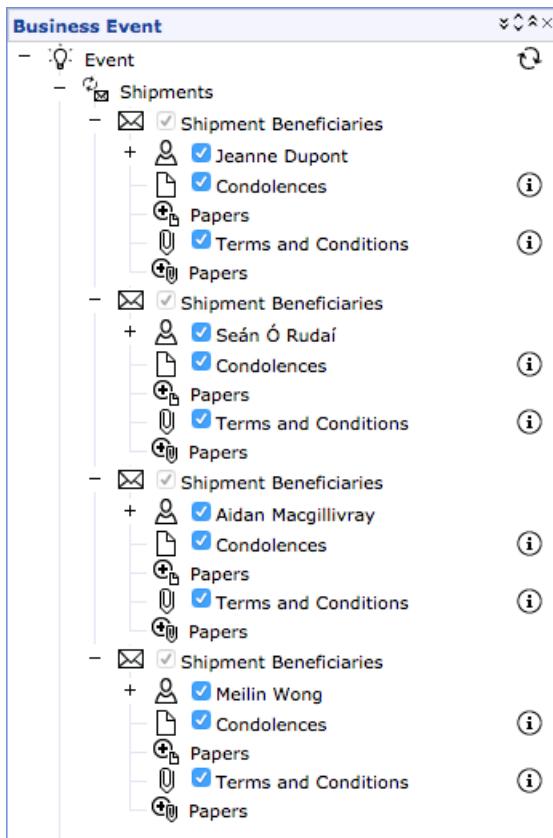
2.4. Flexibility of Shipment Management

Shipment Management provides different approaches, to deal with the big number of possible use cases and to ensure the required flexibility.

2.4.1. Dynamic behaviour and custom data interaction

Business data input can influence the shipment structure. The repetition objects are usually bound to and therefore affected by the business data. This implies that any changes in the business data may require structural changes under the repetition elements especially after an event instance has been created. This is handled by the recalculation (see Section 4.1.6, “Recalculation”) function.

If a staffer selects a contract for an event of death, the contract may contain any number of beneficiaries. The system will have to create one shipment for each beneficiary automatically, assign it the correct recipient and display the results to the staffer.



2.4.2. Customization and extensions

Shipment is a generic construct that applies to a number of business use cases across industries but is complicated by the fact that each business may use extremely different methods and formats for gathering and storing the business data. Therefore, business events, shipments, recipients etc. need to be extensible with respect to their contained data.

While the fax channel may require a fax-number as an input, the mail channel requires an email address. The central print channel might require various parameters such as the latest production date and / or the dispatch priority to be specified.

The screenshot shows the 'Custom Data' configuration dialog for a 'Fax' channel. At the top, it displays the path: Repository-Server/Repository/Documentation Resources /Shipment/Use Case Event of Death/Fax. Below this, there are sections for 'Name/Label' (set to 'Fax'), 'Variable set for output properties' (set to '\$\$Fax Variables\$\$'), and 'Custom Data'. The 'Custom Data' section contains two main groups: 'Model Class (Custom Data)' and 'View Class (Custom Data)'. Under 'Model Class', the value is 'com.assentis.docrepo.common.shipment.iface.customdata.DefaultCustomDataModelClass'. Under 'View Class', the value is 'com.assentis.docwrite.shipment.samples.customdata.ShipCustomDataViewUsingVarSet'. Both fields have an 'Add' button to the right. Below these, there is a 'Parameters of this class' field containing the URL 'getResource?uid=2049939194.1421941438378.2962392990338649.1.1974249623' with a delete 'X' button. At the bottom, there is a 'Custom Data' panel with fields for 'Fax number 1' and 'Fax number 2', both with green checkmarks. It also includes 'Apply', 'Refresh', 'Apply data', and 'Save' buttons.

2.5. Roles

The following section explains the roles involved in Shipment Management.

There are three main roles to be taken into consideration when talking about Shipment Management:

- **Staffer**

The staffer is the actual "end user" of the system in a traditional sense. She creates events based on event definitions, adds data to forms and edits documents. Clerks handling business events in banks and insurance companies typically act as staffers. They do not need to have knowledge of the detailed concepts behind Shipment Management. It is sufficient to introduce them to the usage of the predefined structures.

- **Configurator**

The configurator is a "super user" or "business administrator" in a traditional sense. She creates the object definitions, their contents and the restrictions that apply. The configurator actually prepares the system for usage by the staffer. Configurators require a far better understanding of the business requirements compared to staffers, and would usually be domain experts. The task of composing shipments, defining rules and configuring the document pool is substantially more sophisticated than just using preconfigured definitions.

- **Integrator**

The integrator is the developer of the system. As well as integrating DocFamily into the customer's system landscape, integrator responsibilities include:

- Creating custom GUIs for object definitions

In most cases, event and shipment definitions need to be customized to fit real world business cases. The interfaces are both accessible and well documented. Integrators can build the corresponding code and easily integrate it into the system without the need to recompile the DocFamily source code. The intention of Assentis is to minimize the customer's effort when migrating to a new version of the software by keeping customizations as isolated as possible.

- Creating custom validators

Shipment Management offers well defined interfaces to allow validating events. This is necessary as such validations may become very complex in the real world. They may for example depend on values submitted in a custom GUI but also involve shipment structure aspects.

- Creating custom reactions on changes

It may be desirable to change the event structure following user interaction. The number of recipients in a shipment might for example depend on a specific document being inserted into a shipment. There is a public API available to the integrator allowing for the development of custom reactions based on structural changes of the event.

- Programming event definitions and events

Instead of configuring event definitions, integrators may also programmatically create them over an API or by editing XML files. While this is associated with more effort, it offers a flexible means of using Shipment Management in various contexts. It is also possible to hide an object from staffers and configurators and process them without any user interaction.

- Designing document templates

The documents handled in Shipment Management need to be created, stored and offered for usage by the configurator and staffer. Usually it makes sense to manage a shipment-based project like a traditional document engineering and correspondence project.

- Integrating Shipment Management business systems

Shipment Management is often used as an integral part of a solution involving business systems for the capturing of business data, authentication, archiving, image provision etc. These systems need to be attached through interfaces or interaction with DocFamily.

- Configure and extend rendering

The changes made by the staffer in the UI (e.g. the selection of a specific channel) need to be handled by the document processing part of the solution. The required adaptation depends on the degree of customization in the front-end.



Note

This division into three main roles may not always suffice. There are several plausible reasons for defining a further role, namely: the document designer, a role for professionals that mainly use DocDesign to compose document templates. There are numerous real world examples where these are separate individuals. However, the intention is to keep the explanation as simple as possible (and as complex as necessary). For the sake of this document, the document designer role is merged into the integrator role.

Chapter

3

Configurator view

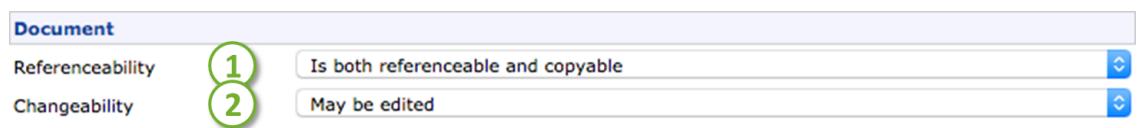
The task of configuring object definitions currently remains within the domain of the integrator. An overview of the concepts of object definition is given here. A technical description of the interfaces can be found in the corresponding JavaDoc and XML Schema definitions.

3.1. Basic principles

The configurator has a set of standard mechanisms available when composing an event definition. These mechanisms are available on most objects and have the same effect.

3.1.1. Referencing and changeability

Defines how documents, resp. attachments are inserted into the instantiated business event and if they can or have to be changed. They can be referenced and/ or copied, and they might be editable or not. In some cases editing might be required. The corresponding settings can be selected from the drop down lists.



1. When adding documents and attachments, but also shipments to a business event instance, they can either be referenced or copied. Four options can be selected:

May only be copied	Adding this object to a business event, will always create a copy of it. So changing this object has only effects inside of the event. Vice versa, changing the original object is not visible in the event.
May only be referenced	Adding this object to a business event, only creates a reference to it. So changing it has effects on the original object and vice versa, change the original object is reflected in the event.
Is both referenceable and copyable	Adding this object will first create a reference but provides a button in the business event tree which will replace the reference by a copy. This step is irreversible.

	
Cannot be copied or referenced	This object can neither be referenced or copied somewhere else in a business event. This setting would render useless documents or attachments of the document pool. It makes sense for all objects that are already a part of the business event upon its instantiation.

2. When adding objects to a business event instance, this setting defines, if it can be modified or not:

Unmodifiable, can not be edited.	In a business event, this object is immutable.
Incomplete, must be editing	On instantiation of a business event, the object must be modified. The object is always instantiated.
	<p>Warning</p> <p>The current version does not force the user to modify the object and does not inform him that a change is required.</p>
May be edited	The object may be changed in a business event. It is always instantiated.

3.1.2. Labels

Labels may be assigned to any object within an event definition. They define what text is displayed on the right hand side of the icon. This enables the configurator to name all objects according to the terms used within the company and business domain. While in one company a shipment is known as an "envelope", in another it may be known as a "communication package". Furthermore, it allows for individual information to be added to an object, e.g. one shipment named "Shipment Mr. John Doe" and another named "Mrs. Jane Anyone", depending on who it is sent to.

Labels are composed of the following information:

1. Locale information (Internationalisation)

It is possible to register various label definitions for different combinations of language and region (called "locale"). Event definitions and their contents may thus be composed in a multi-language form. If the locale information is not provided, the default label is used for all locales or rather for all locales where no local specific label has explicitly been provided.

2. Text

Simple text can be displayed in the label. The text is statically registered, i.e. it does not change between occurrences of an object. Text is used, if the object's name remains the same and does not need to be individualized.

3. Expression

These are references to a textsystem expression that will be evaluated whenever the event is "recalculated", for example, when it is instantiated. This allows for dynamic labels, like "Shipment Mr. John Doe", to be rendered as the expression may contain variables.

Label			
English	1	Event of Death	2
German		Todesfall	
		\$\$\$\$	3

Sample XML configuration:

```
<adfship:Labels>
    <adfship:Label lang="en">
        <adfship:Text>Event of Death</adfship:Text>
    </adfship:Label>
    <adfship:Label lang="de">
        <adfship:Text>Todesfall</adfship:Text>
    </adfship:Label>
    <adfship:Label>
        <adfship:Text>$$$$</adfship:Text>
        <adfship:Expression>
            <adfship:Expanded>false</adfship:Expanded>
            <adfship:LocTexts uID="++4A4Yi7Skn+e5FBKB7QWdqD">
                <adfship:LocText uID="++4A4Yi7Skj+e5FBKB7QWdqD">
                    <adfship:Text>
                        <write:html xmlns:write="http://www.assentis.com/write/html">
                            <div>
                                <write:a class="fck_protected fck_variable" href="" id="getResource?uid=435391052.1421860144204.11789502555819831.0.1974249623::RecShipmentVorName::document" style="text-decoration:none" xmlns:write="http://www.assentis.com/write/html"/>
                                <write:a class="fck_protected fck_variable" href="" id="getResource?uid=435391052.1421860144204.11789502555819831.0.1974249623::RecShipmentName::document" style="text-decoration:none" xmlns:write="http://www.assentis.com/write/html"/>
                            </div>
                        </write:html>
                    </adfship:Text>
                </adfship:LocText>
            </adfship:LocTexts>
            <adfship:Locales>
                <adfship:Locale/>
            </adfship:Locales>
            <adfship:Master>false</adfship:Master>
        </adfship:LocText>
        </adfship:LocTexts>
    </adfship:Expression>
</adfship:Label>
</adfship:Labels>
```

3.1.3. Rules

The shipment framework offers a highly flexible and configurable environment within which various situations can be implemented:

On the one hand, the configurator can create an event definition that allows the staffer to do anything she likes: Select / deselect all shipments, recipients, document and attachments. Full freedom in using all channels with every combination of document recipient may be granted and also the possibility to add documents and attachments anywhere within the event structure.

On the other hand, the configurator can create an event definition with immutable shipments, recipients, channels and contained documents having none of the flexibility defined above. Such event definitions may be implemented for use with an external system that does not require any user interaction.

While both situations exist, it is common to have event definitions that are open in some respect and restricted in others, some typical examples are:

- allow the user to add documents from the document pool to the document listed in the first shipment of an event but not to the second document.
- add a document to a shipment upon instantiation of the shipment if the data element `amount` in the business data added is greater than 100000.
- allow users with the role `teamhead` to select / deselect shipments from an event. However, users without this role (e.g. users with the role `teammember`) are not allowed to do any selection on shipments.
- add a shipment per beneficiary of a payment defined in the input data.

The mechanisms to automatically add elements and allow staffers to perform certain actions are defined using rules. A rule is the combination of an instruction or permission together with a condition. If the condition evaluates to `true`, the instruction is executed or the permission is granted.

The following instructions and permissions are currently available:

1. Allow insertion of [object] - allows the insertion of objects at the insertion point
2. Specific [object] - will insert a specific object
3. Repetition of a specific [object] - automatically adds one or many objects dynamically

Rules

- Allow insertion of Papers
- Specific Paper
- Repetition of a specific Paper

These rules apply to all objects, which can be inserted automatically, as there are shipments, papers (documents and attachments), and recipients. They are defined by a set of properties, which depend on the object type to insert.

Conditions. For all properties a condition can be defined or dragged & dropped from the *Repository Elements* tree onto the corresponding area. If the condition resolves to true, the property applies to the insertion point.

This allows configurators to graphically build conditions and optionally include "Role" variables such that any condition may also be dependent on the permissions assigned to the current user.

Caution



Using conditions including "Role" variables must always be evaluated upon instantiation of the event or recalculation and **never** during production. "Render-time" is run in a batch mode operation during which no user is logged on.

3.1.3.1. Allow insertion of [object]

This rule allows the insertion of an object to a specific insertion point for objects of this type. It is applicable whenever the staffer can influence the structure of an event directly by adding new objects, e.g. allow adding documents and attachments to shipments, adding recipients, or even create new shipments.

Table 3.1. Allow insertion of [object] - contained in:

Shipment assignment 
Recipient assignment 
Document assignment 
Attachment assignment 

Table 3.2. Allow insertion of [object] - properties:

AddableIf	defines if adding objects is allowed. Adding is a generic term which can have different meanings: Moving from another location, inserting a copy, building a reference or creating a new element.
CopyingIntoAllowedIf	defines if copies of objects may be pasted here.
InstantiableAllowIf	defines if new objects may be instantiated here.
MoveableAllowIf	defines if objects may be moved here.
ReferenceableByAllowedIf	defines if references of objects may be inserted here.
DeselectableIf	defines if the insertion point can be deselected.
PreselectedIf	defines, if the insertion point is preselected on instantiation of the business event.
RenamableIf	defines, if inserted objects can be renamed. This applies only for papers (documents and attachments) and only for copies of papers. Referenced papers can not be renamed in a business event.

Rules (Allow insertion of Papers)	
Addable If - allow to include other Papers	<input checked="" type="checkbox"/> 
CopyingIntoAllowedIf - allow to include copies of other Papers	<input checked="" type="checkbox"/> 
InstantiableAllowIf - allow to instantiate this Paper	<input type="checkbox"/> 
MovableAllowIf - allow to move Papers	<input checked="" type="checkbox"/> 
ReferenceableByAllowedIf - allow to include references of other Papers	<input checked="" type="checkbox"/> 
RenamableIf - allow to rename copied Papers	<input checked="" type="checkbox"/> 
DeselectableIf - allow to enable/disable Papers	<input checked="" type="checkbox"/> 
PreselectedIf - per default, select Paper	<input checked="" type="checkbox"/> 
<input type="button" value="Reset"/>	

3.1.3.2. Specific [object]

This rule allows the automatic insertion of a specific object at this insertion point. The assignment requires the predefinition of this specific object, which can be inserted.

Table 3.3. Specific [object] - contained in:Shipment assignment Recipient assignment Document assignment Attachment assignment **Table 3.4. Specific [object] - properties:**

AddIf	defines if the predefined object is automatically added. Setting this property statically to false, disables the addition per default and renders this assignment useless.
DeselectableIf	defines if the object can be deselected.
PreselectedIf	defines, if the object is preselected on instantiation of the business event. This property has only effect if the AddIf property evaluates to true and the object has been inserted.

Rules (Specific Paper)

Add If - automatically add Paper	<input checked="" type="checkbox"/> 
DeselectableIf - allow to enable/disable Papers	<input checked="" type="checkbox"/> 
PreselectedIf - per default, select Paper	<input checked="" type="checkbox"/> 

Reset

3.1.3.3. Repetition of a specific [object]

The repetition rule is used in the event definition to create a number of shipments or recipients either when instantiating / recalculating the event or when rendering it. This feature is useful when the number of data objects is unknown at design time but is supplied by the staffer or data at render time.

This rule does not grant the staffer any permissions, it is an instruction to the framework regarding the composition of the event structure.

Table 3.5. Repetition of a specific [object] - contained in:

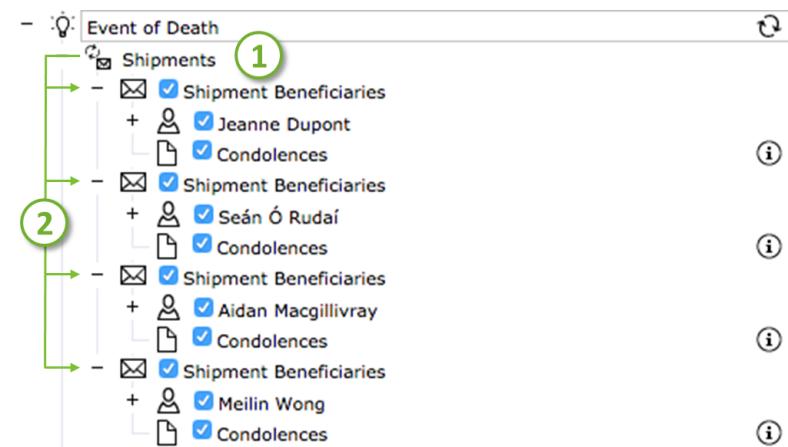
Shipment assignment

Recipient assignment

Document assignment

Attachment assignment

When the event definition is instantiated and an event is created, a number of objects are created and automatically added to the event. The next sample shows a shipment repetition element (1) and its nested shipments (2):

**Table 3.6. Repetition of a specific [object] - properties:**

Repetition variable	This variable contains the XPath with the repetition. The object is repeated for each child node of the variable.
Repetition key variable	This variable must uniquely identify an element within the repetition. It is used by the framework to "remember" repeated elements. This is important in various situations, such as: Initially, the staffer selected 5 persons from the business data. The event definition states, that each selected person is assigned a shipment. The staffer has edited the shipments of these persons. The staffer then decides to add a 6th person. The system must not overwrite the 5 existing shipments but rather simply adds the sixth. Afterwards, the staffer decides to delete the 3rd person. The system must only delete the 3rd shipment and leave all others untouched.
Production time repetition	This property defines the time at which the repetition should be evaluated. This can either occur when the event is instantiated / recalculated (check box ticked off) or when the event is rendered to a final document (check box ticked). The latter is the usual case when a large set of shipments or recipients is created and which is not changed by the staffer. The former is normally applied when a small set of shipments (20 or less) are automatically created but may be potentially modified by the user. This property does not apply for documents or attachments.
DeselectableIf	defines if the automatically inserted object can be deselected.
PreselectedIf	defines, if the object is preselected on instantiation of the business event. This property has only effect if the repetition inserts at least one object.

Rules (Repetition of a specific Shipment)

Repetition variable	<code>\$\$DFShipShipmentRI\$\$</code> <input type="button" value="X"/>
Repetition key variable	<code>\$\$DFShipShipmentRI_id\$\$</code> <input type="button" value="X"/>
Production time repetition	<input type="checkbox"/>
DeselectableIf - allow to enable/disable Shipments	<input checked="" type="checkbox"/> <input type="button" value="?"/>
PreselectedIf - per default, select Shipment	<input checked="" type="checkbox"/> <input type="button" value="?"/>

3.1.4. Custom Data

Custom data are provided by the user when working in a Business Event Instance. The data can influence the content of papers, the set of generated papers or even shipments.

In this control, the method to provide the user data is configured. Depending on the selection, further input fields are displayed which are explained in the following sections. Either way, the required data structure is created in the custom data section of the underlying object. When filling in data, they will be saved. They act as preconfigured values in case the object is used inside of a Business Event Template.

3.1.4.1. Form

A Form is dragged and dropped from the Repository Elements tree into the drop area.

Custom Data

Form Custom Data Configuration

Data input

In a Business Event Instance, the corresponding Form is displayed for the data input.

Personal data

First name	<input type="text" value="Daniel"/>	Last name	<input type="text" value="Huber"/>
Gender	<input type="text" value="male"/>	<input type="button" value="?"/>	
Address:			
Street	<input type="text" value="Schlossstrasse 22a"/>		
ZIP Code	<input type="text" value="86497"/>	City	<input type="text" value="Horgau"/>



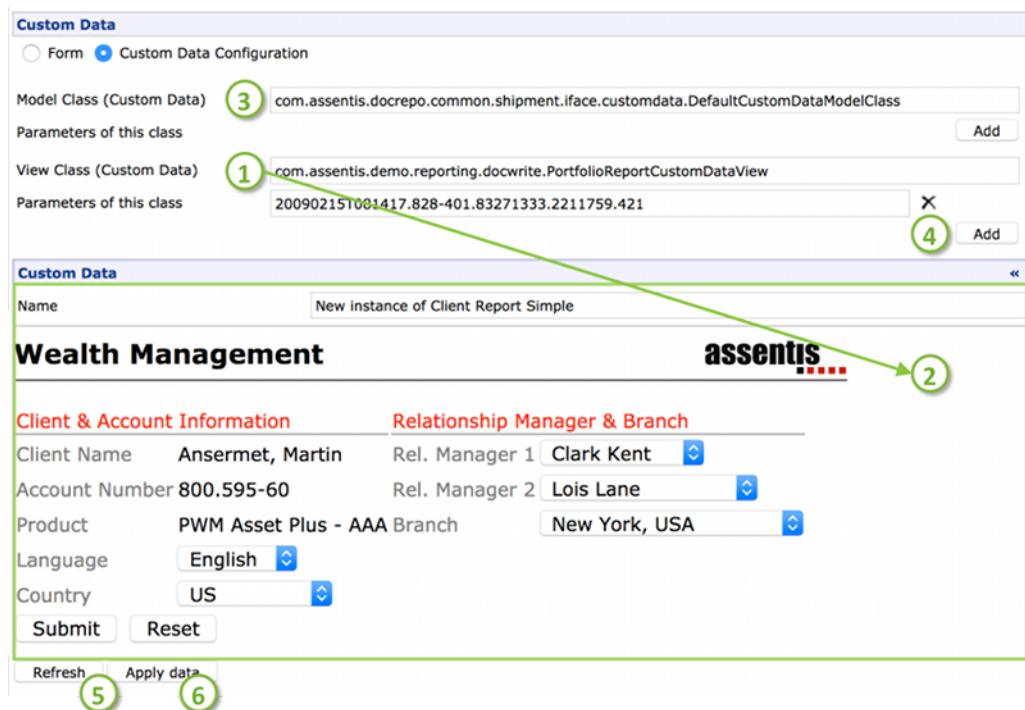
Note

Changes in Forms which are used in Business Event Instances do not trigger the refresh of the event tree. If the Form fields control the document selection, a manual recalculation of the Business Event Instance is required. The recalculation is triggered by clicking the corresponding button (see Section 4.1.6, “Recalculation”).

See *DocWrite User Guide* and *TextSystem Development Guide* for more information on the creation of Forms.

3.1.4.2. Custom data configuration

For the custom data configuration, a view class can be defined (1) and a corresponding model class (3). For both the required parameters can be added (4). When clicking the “Refresh” button (5), in the lower panel, the view is displayed (2) as it will be shown in the Business Event Instance.



For this configuration, the `Apply data` button (6) must be clicked to create the data structure in the custom data section of the underlying object.

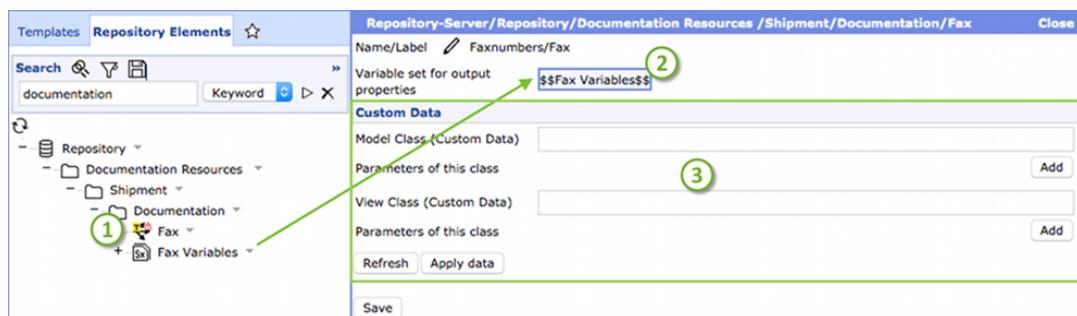
3.2. Channel definition

A channel represents a production destination. This may e.g. be central printing, email or fax. Since channels are individual across different companies, a custom data section (3) allows configuring channels according to the customer's needs (see Section 3.1.4, "Custom Data").

Channels definitions are set up separate from event definitions (1) and may be reused across various event definitions. This makes sense since the number of channels is usually limited and the channels are available to many (if not all) business events.

From a technical perspective, the channel is the central means of communication with the production system of DocFamily: through the assignment of a variable set to the channel (2), information may be delivered for further processing to DocBase.

All variables of the variable set are stored as custom attributes in the `PPE_METADATA` table. This entry references the entry in the `POSTPROCOUTDATA` table which stores the respective shipment. Custom attributes are filed as key-value pairs, where the variable name is used as key. They can be retrieved by DocBase standard mechanisms via `jobId` and the `POSTPROCOUTDATA` table entry (see *DocBase Post Processing Guide*).

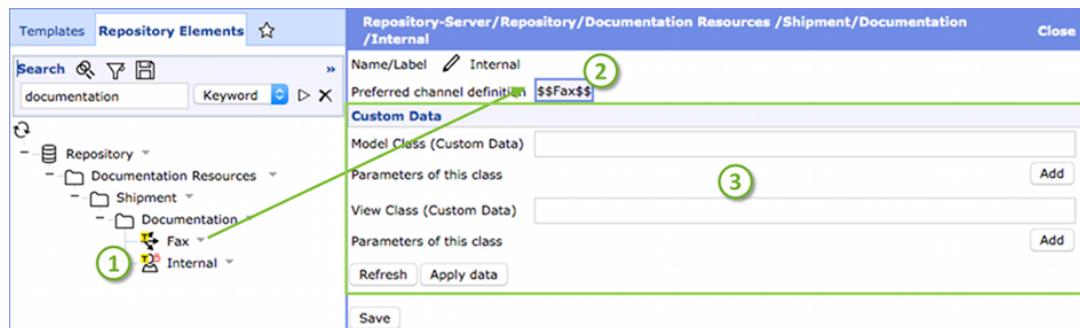


3.3. Recipient definition

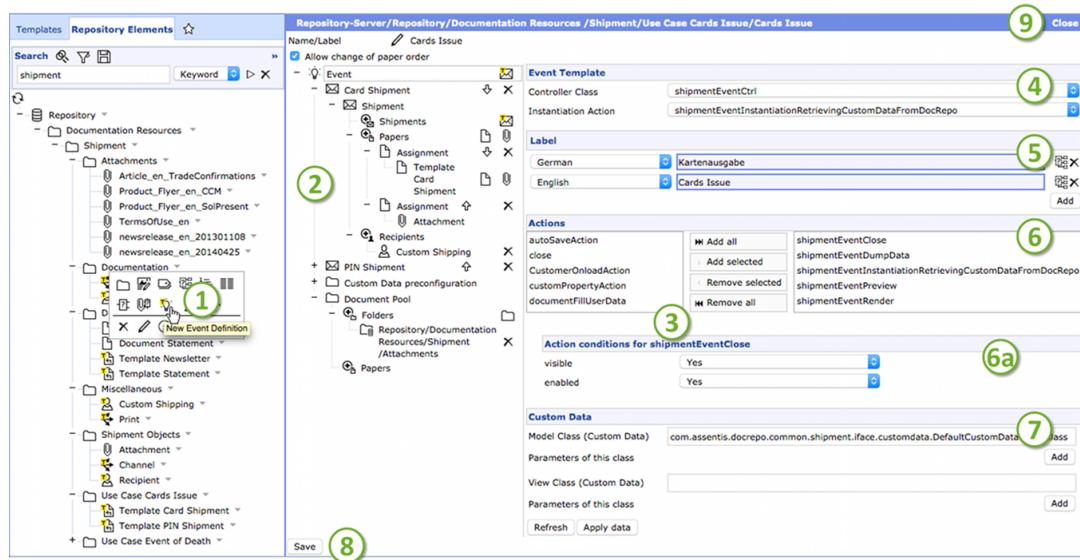
A recipient represents the receiver of a shipment. Since the actual recipient of a shipment depends by the business data, only a certain type of recipient can be assigned to a business event. Still, a type can be used in different event definitions. Therefore types can be setup separately in the system in form of a recipient definition (1).

Recipients presented by a recipient definition may have in common the channel their shipments will be sent through. The configurator can add a preferred channel definition (2). The assignment may be changed later on business event instantiation by the staffer or during production time by the business data.

Like for other objects, the custom data section (3) allows the configuration of individual views and data (see Section 3.1.4, "Custom Data").



3.4. Event definition



An event definition is the template an event is instantiated off. It configures the initial structure, which can be edited later on by the staffer or is influenced by the instantiation or runtime data.

Find details on business events in Chapter 4, *Business user view*.

1. An event definition can be created in the DocWrite *Repository Elements* tree. After providing name, keywords and labels in the opening dialog, the event definition is opened in the DocWrite workarea. This workarea is devided into:

2. Event definition tree which represents the structure of the event. In the tree various objects can be created and configured. They will be described in detail in the following sections:

- Shipments: Section 3.5, "Shipments"
- Documents: Section 3.6.2, "Documents"
- Attachments: Section 3.6.4, "Attachments"
- Recipients: Section 3.7, "Recipient assignment"
- Document pool: Section 3.9, "Document pool"
- Custom data preconfiguration: Section 3.8, "Custom Data preconfiguration"

The check box at the very top of the event structure decides, if later on the staffer can change the order of shipments and of document and attachment assignments. Single documents or attachments within an assignment can not be moved.

3. Area which shows the properties of the object, which is currently selected in the event definition tree.

4. General settings for the event template.

- Controller class

Specifies the event controller class. If no custom implementation exists, the standard controller for events is `shipmentEventCtrl`, which can be selected from the drop down list.

- Instantiation Action

An action that is called when the event is instantiated. This may be helpful if the staffer has to be prompted for some input before the event is created.

5. Label

Contains the localized labels appearing when the staffer instantiates the event (see Section 3.1.2, "Labels").

6. Actions

Actions specific to the event. The system will preconfigure the standard shipment actions. On the left side of the control every DocWrite action in the repository will be listed, so that also custom actions can be assigned to an event. In general these actions will appear in the global actions menu bar. Find more information on specific actions in "DocAdmin Online Help"

6a Action conditions

When selecting an action at the right side of the control, in this area the available action conditions will be displayed. Here the user can decide, if on event instantiation this action is visible and in case it is, if it is enabled. The following settings are possible: Yes, No and Admin mode only.

If Admin mode only is selected, the action is visible, respectively enabled only when the user has activated the mode Administrator in the user profile.

7. Custom data section where customized views and data for the event can be defined (see Section 3.1.4, "Custom Data").

8. The **Save** button applies all changes, which have been made in the whole event template. It updates the event definition tree (2) and also the business event data in the background and it triggers the validation.

9. The **Close** button closes the event. As for the **Save** button, on close the event definition is saved and the validation is triggered.

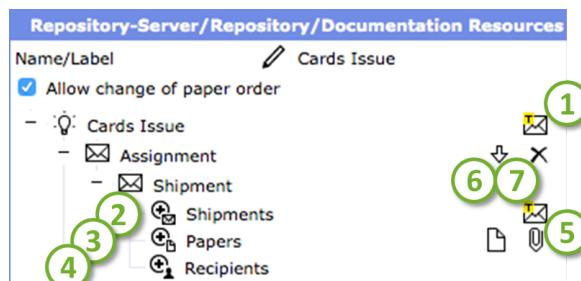
Validation warnings and errors will be displayed in the message panel of DocWrite at the top of the browser window. In case of validation errors (marked by the font color red), the event can not be closed until these errors are fixed.

3.5. Shipments

A shipment is a set of documents respectively attachments, which can be sent to a set of recipient via a specific channel. Thereby the set can consist of one or multiple documents, attachments or recipients. The number of these objects may depend on the business data.

Shipments can be repeated and they can be nested within each other. When shipments are nested, only the recipients of the outermost shipment and their individual channels are considered during production. Recipient based channels can be restricted by the channel assignment on specific documents or attachments.

A shipment is added to a business event definition by the **Add Shipment** button (1). This creates a structure with a shipment assignment with three containers: **Shipments** (2), **Papers** (3) and **Recipients** (4). The **Shipments** container again can contain multiple shipments, which will be added by the corresponding button (5). The order of shipments can be changed (6) and they can be removed (7). The usage of the **Papers** and **Recipients** container is detailed in Section 3.6.2, "Documents" respectively Section 3.6.4, "Attachments", and in Section 3.7, "Recipient assignment".



Shipments are added via assignments, where labels (1) for the shipment can be defined (see Section 3.1.2, "Labels") and a rule (2) can be selected (see Section 3.1.3, "Rules").

Shipment Assignment	
Label	(1)
Rules	
<input type="radio"/> Allow insertion of Shipments <input type="radio"/> Specific Shipment <input type="radio"/> Repetition of a specific Shipment	

On the shipment itself the following properties are configurable:

1. The referenceability defines, if the shipment can be either referenced and/ or copied, or not. In case an option other than Cannot be copied or referenced is chosen, in the business event the shipment can be dragged & droppend onto an assignment, where shipments can be inserted. In case the shipment can be copied, a button will appear in the business event tree, that will force the creation of a copy of the whole shipment. Otherwise only a reference to the shipment will appear in the business event.
2. Specific views and data can be defined in the custom data section (see Section 3.1.4, "Custom Data").

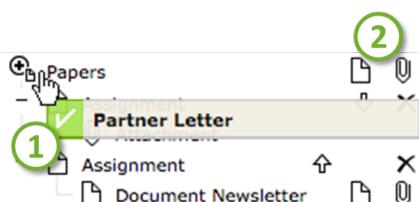
The screenshot shows the 'Shipment' configuration screen. At the top, there's a section labeled 'Referenceability' with a dropdown menu set to 'Is both referenceable and copyable'. A green circle with the number '1' highlights this field. Below it is a section labeled 'Custom Data' with a 'Model Class (Custom Data)' input field and an 'Add' button. A green circle with the number '2' highlights the 'Model Class' field. Further down are sections for 'Parameters of this class' and 'View Class (Custom Data)', each with its own 'Add' button. At the bottom are 'Refresh' and 'Apply data' buttons.

3.6. Papers and paper assignments

Papers are all objects which are sent to an output channel and directed to a recipient. They are added to the document pool and to assignments in shipments. There are three types of papers - documents, attachments, and carbon copies. The creation and properties of documents and attachments are nearly identical. Carbon copies are closely linked to documents. The following description details the specifics of documents, carbon copies, and attachments. For the time being, these types are summarized as papers, and the description speaks of papers and paper assignments. Following the general description, the specifics of documents, carbon copies and attachments are pointed out.

3.6.1. Assignments

A paper assignment can be created in two different ways. Either a paper is dragged into the business event definition tree and dropped onto the **Papers** container or onto another paper (1), or the **Add Document/ Add Attachment** button in the tree is used (2). Both creates always a structure of an assignment and a paper, where the paper can be a placeholder for documents that are either created during instantiation or later on added by the staffer.



In general the paper assignment defines properties for the contained papers. Additionally a paper assignment might later on, in the business even tree, serve as a container where either multiple papers are displayed or where the staffer can insert papers. As displayed in Figure 3.1, "Paper assignment configuration", the properties for this container are configured here:

1. *Generate as PDF Form:* When ticking this checkbox, an additional field, *Element Id of the file containing the PDF Form styles*, is displayed where the resource identifier of a PDF Form stylesheet must be inserted. The provided identifier must be of the form:

```
getResource?uid=1696771889.1521814652299.59120034790135320.3.118807
```

Then the paper is generated as PDF Form with Variables shown as PDF Form elements. In a Business Event Instance, this paper is displayed in the *PDF Form* view.



Warning

When ticking the checkbox to generate the PDF as form, it is mandatory to provide the identifier of the stylesheet. Despite the fact that it is possible to save the Business Event Template without an error, the instantiation of the Business Event Instance will be erroneous.

See *DocWrite User Guide* for further information on PDF Forms and the *PDF Form* view.

Open automatically when Event is opened: When ticking the check box, the paper appended to the assignment is opened on instantiation of the Business Event Instance. If the assignment contains multiple papers or multiple assignments were marked to be opened, the topmost paper to be found in the Business Event Instance tree is opened.

2. Label for the specific paper in this assignment. In case it contains multiple papers, all will have the same display name in the business event tree.
3. Depending by the selected rule, for the papers in this assignment, properties have to be configured (details in Section 3.1.3, "Rules").
4. The group label can only be set, in case the assignment serves as container for multiple papers. Then this label is displayed as name of the container in the business event tree.
5. Specific views and data can be defined for this paper assignment (see Section 3.1.4, "Custom Data").

Figure 3.1. Paper assignment configuration

Document Assignment

① Generate as PDF Form

Element Id of the file containing the PDF Form styles

②

Open automatically when Event is opened

③

Label

④

Rules (Repetition of a specific Paper)

⑤ Repetition variable

Repetition key variable

DeselectableIf - allow to enable/disable Papers

PreselectedIf - per default, select Paper

Add

Reset

Group label

⑥

Custom Data

⑦ Form Custom Data Configuration

Data input ↵ X

Figure 3.2, “Paper configuration” shows further settings which are adjustable directly on the paper:

1. Defines the referencing and the changeability of the paper (see details in Section 3.1.1, “Referencing and changeability”).
2. A label for the specific paper can be defined here. But it will be overwritten by any label of the same locale, which has been defined at the paper assignment. Also, in case it is the placeholder for a repetition and results in multiple papers, they all will display the same label later in the business event tree.
3. Restricts the displayed papers in this assignment. Clicking the Add button creates an input form, which corresponds to and works like the one of the advanced search. Find more information about the advanced search in “DocWrite User Guide”. Multiple restrictions can be set up for one container.

This section appears only in case the paper assignment has been created by the Add Document/ Add Attachment button and does not contain a specific paper.

4. This selection defines the channels to which the paper can be sent. It overrules the channel assignment of the recipient.

When instantiating the Business Event Template, the actual channels to which the papers of a shipment are sent are calculated as intersection of all paper channels within a shipment. If every paper has assigned a different channel, this intersection set is empty.

Correspondingly, it is possible to configure a Business Event Template which creates a Business Event Instance where no channel is available for a shipment. The channel configuration of the Business Event Template must ensure that for a shipment at least one channel is assigned to all papers that are included after the application of the rules.

5. In the custom data section views and data can be customized (see details of custom data in Section 3.1.4, “Custom Data”).

Figure 3.2. Paper configuration

The screenshot shows the 'Paper configuration' interface with five numbered sections:

1. Referenceability: A section with two dropdown menus: 'Is both referenceable and changeable' (set to 'Yes') and 'May be edited' (set to 'Yes').
2. Label: A section with an 'Add' button.
3. File Restrictions: A section with an 'Add' button.
4. Allowed Channels: A section showing a list of channels: 'channel', 'Email', 'Fax', 'Faxnumbers', and 'Print'. To the right is a toolbar with buttons: 'Add all', 'Add selected', 'Remove selected', and 'Remove all'.
5. Custom Data: A section with two input fields: 'Model Class (Custom Data)' and 'View Class (Custom Data)'. Below each field is an 'Add' button. At the bottom are 'Refresh' and 'Apply data' buttons.

3.6.2. Documents

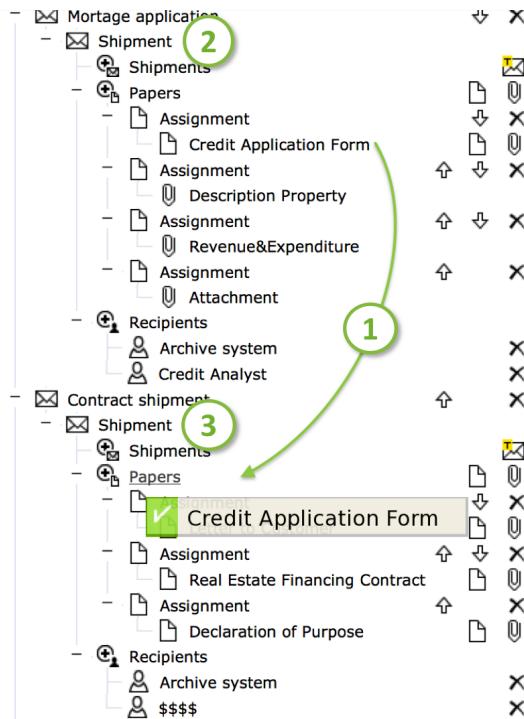
Technically, documents are DocFamily internal objects like Text System Templates, Text System Templates with Copies and Text System Documents.

Document assignments can be nested, meaning a document assignment again can contain another document assignment. This can be useful in case sending a document depends on sending another one. Then the depending document can be nested into the document assignment of the initiating document. Document assignments also can contain attachment assignments. The corresponding buttons are displayed in the business event definition tree.

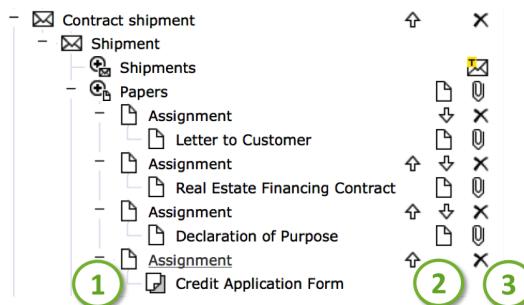
3.6.3. Carbon Copies

Carbon copies are copies of documents and correspondingly copies of DocFamily internal objects like Text System Template, Text System Templates with Copies and Text System Documents.

They are created by dragging and dropping (1) a document from a document assignment of a shipment (2) into the **Papers** container of either the same or another shipment (3):



A new assignment with the carbon copy (1) appears in the **Papers** container:



The content of the carbon copy document is an exact copy of the original document, and does not have any configuration options. A carbon copy can only be moved (2) to a new position within the **Papers** container, or be deleted (3).

A carbon copy assignment is governed by the rules defined in the document assignment of the original document. It is not possible to define different rules for the carbon copy. The configuration options are limited to defining labels and if the carbon copy is opened automatically on the instantiation of the Business Event Instance. Other than document or attachment assignments, a carbon copy assignment can not contain nested assignments.

3.6.4. Attachments

Attachments are papers of a specific type which is marked in the *Repository Elements* tree and in the search filter with the paper clip  . Attachment assignments cannot contain other paper assignments, but they can be contained in a document assignment.

In the Business Event Template, the attachment source must be selected. The setting defines if, when instantiating a Business Event Instance from this template, an attached document must be chosen from the document pool or if it can be uploaded directly.

Attachment source Document pool Upload

The documents for the document pool must be uploaded to the repository beforehand by using the **Convert** and **Upload Attachment** action. The Document pool is described in Section 4.1.5, “Document pool” and the action in *DocWrite User Guide*.

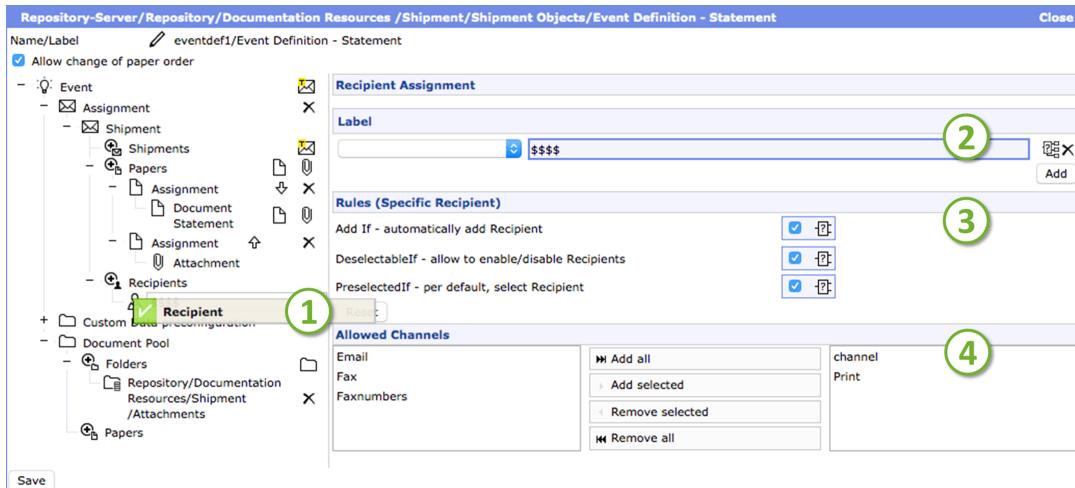
When the direct upload is enabled, the **Upload Attachment** action is displayed in the Business Event Instance structure tree. The action must be added to the Business Event Template (see Section 3.4, “Event definition”, point “Actions”). The handling of attachments in Business Event Instances is described in Section 4.1.3.6, “Attachment”.

3.7. Recipient assignment

Every shipment has at least one recipient. All recipients of a shipment get the same documents although not necessarily through the same channel. If two recipients should get different documents, separate shipments will have to be created.

Recipients can only be dragged & dropped from the *Repository Elements* tree onto the **Recipients** container of a shipment (1). The assignment is done implicitly and the recipient is inserted directly. The following properties can be configured for the recipient:

2. Label for the specific recipient. In case it creates multiple recipients and a static label is defined, all recipients will have the same display name in the business event tree (see Section 3.1.2, “Labels”).
3. Depending by the selected rule, for the recipient, properties have to be configured (details in Section 3.1.3, “Rules”).
4. For this shipment, a specific channel can be defined for the recipient assignment. It will be restricted in case a paper assignment has defined its own channels.

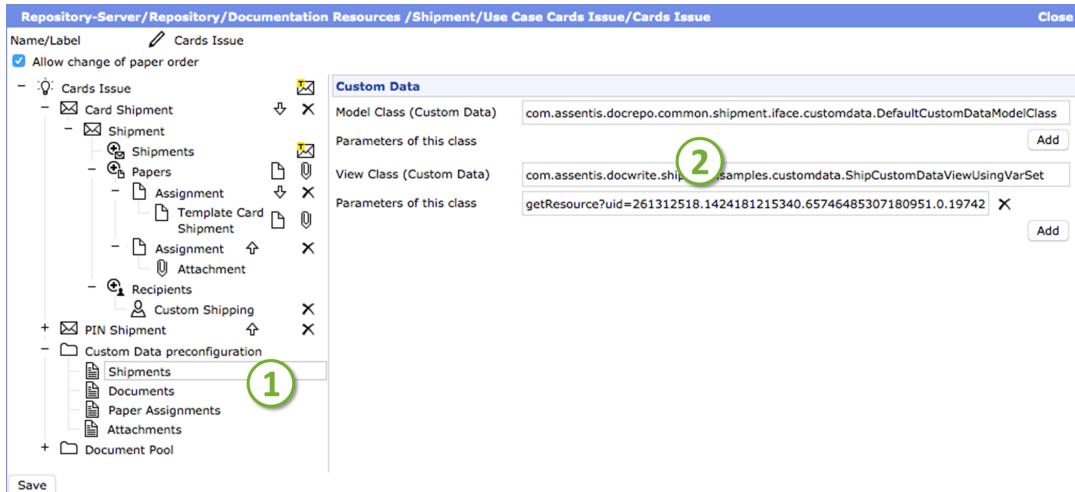


3.8. Custom Data preconfiguration

For shipments, papers assignments, documents and attachments, custom data can be preconfigured (1). As explained in Section 3.1.4, “Custom Data”, model and view classes, and required parameters can be defined (2).

In the business event definition, the corresponding form will then appear for each of these objects, which is created in the business event definition tree. It is possible to insert default values for the business event on its instantiation.

The custom data view will also appear for all objects of this type in the instantiated business event. In case, default values have been defined, they are displayed but can be overwritten by any inserted data.



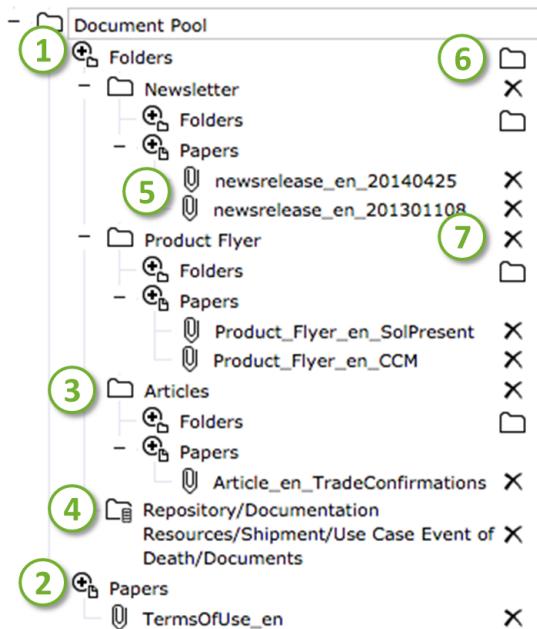
3.9. Document pool

The document pool is the section of an event, where all documents and attachments are stored that the staffer may add to shipments.

There is always exactly one document pool in every event definition.

The document pool is structured into folder assignments (1) and paper assignments (2). The folder assignment can contain logical folders (3) and repository folders (4). A paper assignment contains specific documents or attachments (5). Repository folders and papers are added via

drag & drop. Logical folders are created in the document pool structure (6). All elements can be removed by the cross-sign (7).



Logical folders. They may be set up by the configurator depending on her preferences. Usually this is done in a manner that benefits the staffer's understanding (as opposed to reflecting a technical structure).

Repository folders. are one-to-one links to a folder in the repository. They can be used when multiple document pools share common documents and attachments: a reusable repository folder structure may be set.

Properties. When clicking on an object in the document pool tree the right area shows the properties of it. The displayed properties are specific to the selected object. See the following table for more information which property is available for which object and its functionality.

Table 3.7.

No.	Description	Paper	Repository folder	Logical folder
1.	Defines the referencing and the changeability (see details in Section 3.1.1, "Referencing and changeability").	X	X	
2.	Labels which will be displayed in the document pool of an instantiated business event (see details of labels in Section 3.1.2, "Labels").	X	X	X
3.	The displayed objects can be restricted for repository folders. Clicking the Add button creates an input form, which corresponds to and works like the one of the advanced search. Find more information about the advanced search in "DocWrite User Guide". Multiple restrictions can be set up for one folder, e.g. to display documents created by user A and text components created within a certain time frame.		X	
4.	This selection defines, to which channel documents and attachments can be sent. If the selection is done for a repository folder, all documents and attachments in this folder are affected. Furthermore, this selection overrules the channel assignment of a recipient.	X		X
5.	In the custom data section views and data can be customized (see details of custom data in Section 3.1.4, "Custom Data").	X		

Attachment

Referenceability (1) Is both referenceable and copyable
Changeability (1) May be edited

Label (2) Attachments

File Restrictions (3) Add

Allowed Channels

channel (4)	Add all Add selected Remove selected Remove all	Fax Email
Faxnumbers		
Print		

Document

Referenceability (1) Is both referenceable and copyable
Changeability (1) May be edited

Label (2) Documents

File Restrictions (3) Add

Allowed Channels

channel (4)	Add all Add selected Remove selected Remove all	Print
Email		
Fax		
Faxnumbers		

Custom Data

Model Class (Custom Data) (5)
Parameters of this class

View Class (Custom Data)
Parameters of this class

Refresh

3.10. Workflow on business event templates

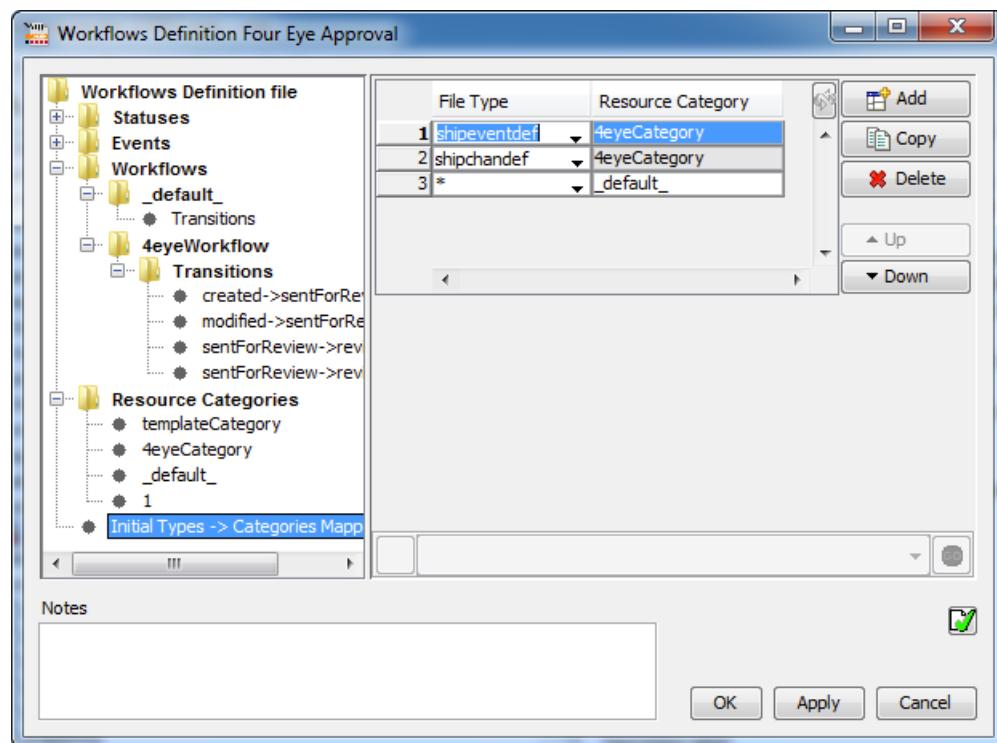
As for other resources in DocWrite also for business event templates a workflow can be defined. See "DocAdmin Online Help" to find out, how to activate the workflow for DocFamily.

A workflow for a business event template has to be set on system level in the following ways:

- **Global assignment of a workflow**

It is possible to extend the workflow definition. Therefore the workflows definition file in DocRepo has to be edited with DocAdmin. The following default editor is opened, where the tree entry "Initial Types -> Categories Mapping" can be extended. By adding in the right panel the file type "shipeventdef" and assigning a resource category each newly created business event template will have assigned automatically the corresponding workflow.

Figure 3.3. Extension of the workflow definition



- Specific assignment of a workflow

A workflow can be set specifically for every single business event template after it has been created.

In DocAdmin the dialog “Workflow” is opened on choosing the menu entry “Workflow” of the context menu of a business event template, where the resource category for the required workflow can be chosen.

Figure 3.4. “Workflow” entry in context menu

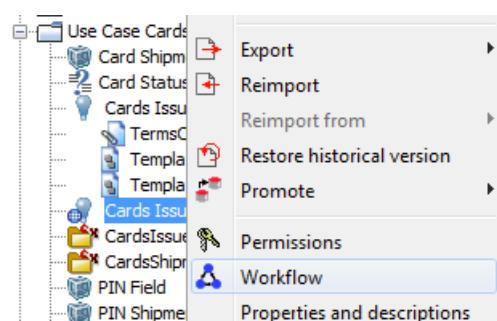
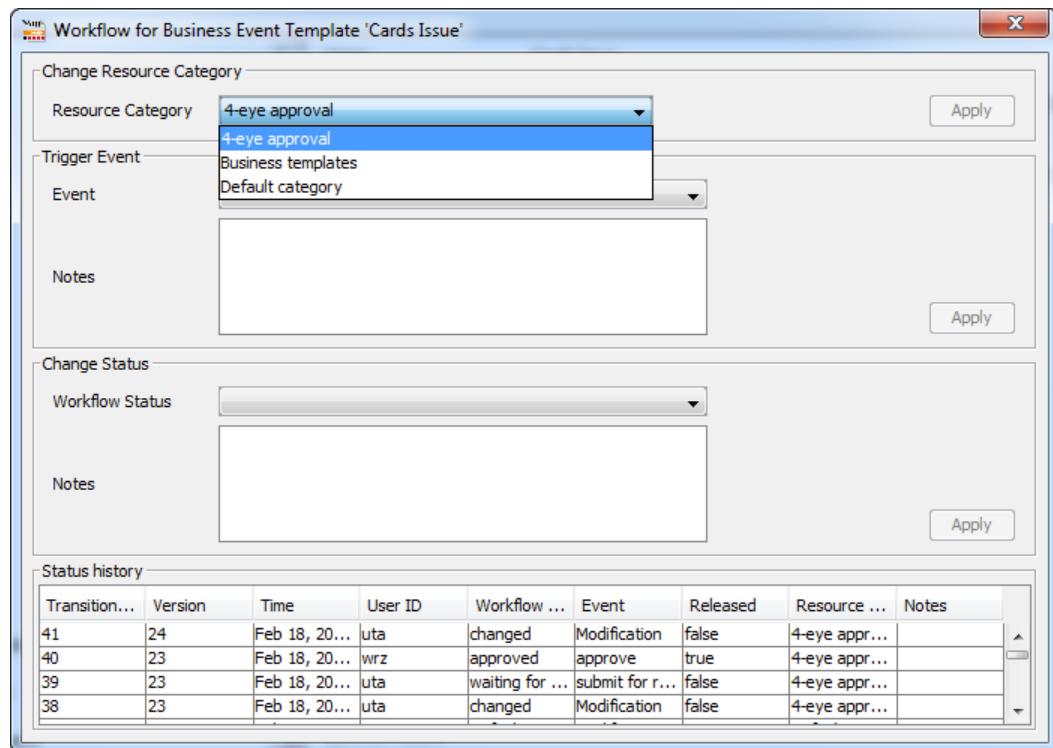
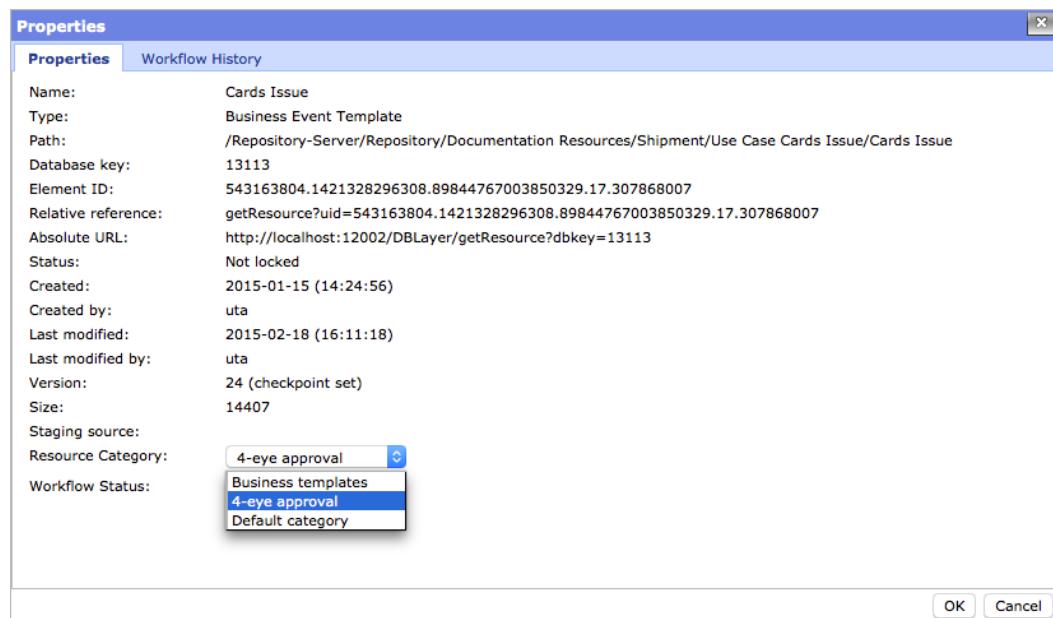


Figure 3.5. Specific assignment in DocAdmin

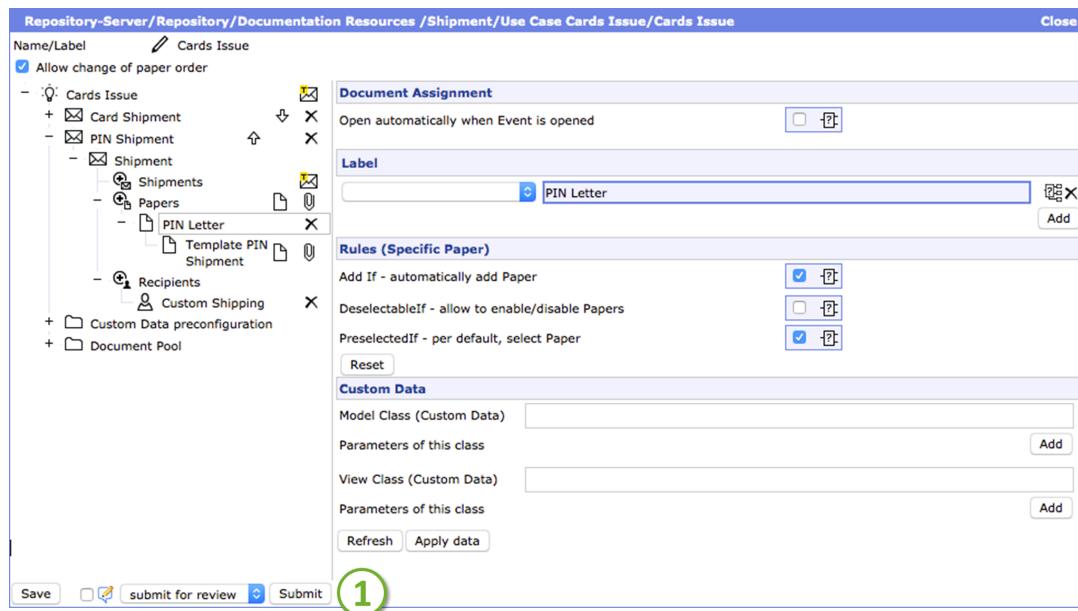
In DocWrite the workflow can be chosen on the dialog “Properties”:

Figure 3.6. Specific assignment in DocWrite

Editing a business event template after assigning a workflow will trigger a corresponding workflow event.

While creating and manipulating business event templates, controls will be displayed in case an event occurred that triggers the workflow. The control can be found at the bottom of the work area beside the “Save” button (1).

Figure 3.7. Workflow controls in the business event template



The workflow control will be displayed if the user has executed a workflow triggering action and does not close the editor at the same time.

In case he closes the editor with this action, a dialog will be displayed, that informs the user about the proceeding in the workflow.

General information about workflows can be found in "DocAdmin Online Help". Information about working with workflows in DocWrite is described in the document "DocWrite User Guide".

Chapter

4

Business user view

4.1. Staffer

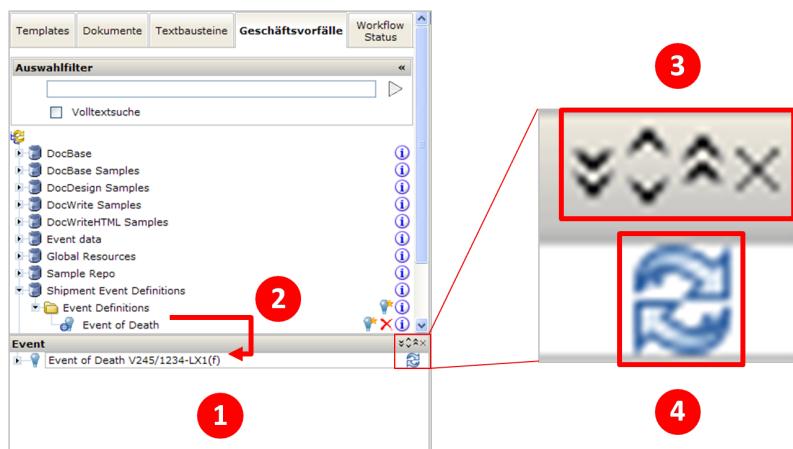
This section describes the staffer's functionality in Shipment Management.

4.1.1. Creating an event

The staffer chooses to instantiate an event based on an event definition (1). She clicks on the corresponding icon (2)



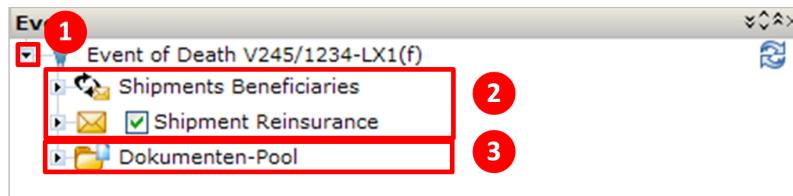
A new section (1) displaying the created event (2) is displayed. The section may be maximized, minimized, changed in size or closed using the corresponding controls (3). The event may be recalculated using control (4). The meaning of recalculating an event is explained later.



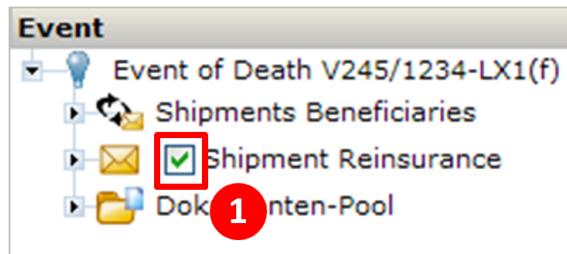
4.1.2. Event

The staffer clicks on the horizontal arrow (1) to open the structure of elements underneath the event element. There are always two types of elements below the event:

- Shipments and shipment repetitions (2)
- The document pool (3)



The staffer may choose to switch a shipment (or any other object) on or off , as long as the appropriate checkbox is available and editable (1).



4.1.3. Shipment

A shipment may contain any combination of shipments, references to shipments (1), recipients (2), references to documents (3) and attachments. A shipment may be compared to a physical envelope that is sent to a single recipient with all enclosing documents and attachments being included within the same envelope. A shipment with multiple recipients will be packed in multiple envelopes. The contents of a shipment may be changed by the staffer depending on the configured access rights. If several recipients are associated with a shipment, each receives the **same** shipment. If the shipments should differ (and even if just marginally) another shipment has to be created.



If a shipment contains other shipments, all shipments are sent to the recipient of the topmost channel in the "channel-hierarchy". The recipients and channels of the subordinate shipments are not considered anymore.

4.1.3.1. Shipment reference

A shipment reference may either be pre-configured or established by the staffer by dragging & dropping a shipment from the same event onto the corresponding icon (1).

As a result, the referenced shipment will be added with a shipment reference icon (1). The reference may be removed by clicking on (2). Clicking the green arrow (3) highlights the original shipment (i.e. the shipment where the reference "points" to). By clicking on the plus icon (4), the shipment reference is converted into a copy of the shipment.



When the staffer creates a copy of the shipment, the plus icon and the green arrow on the right hand side disappear. This makes sense since there is no shipment which is referenced anymore and there is no need to create a copy since this has already happened. Only the delete icon remains (2). Instead of a shipment reference the icon now displays a shipment, which may be opened and changed as any normal shipment. Changes in the shipment do not have an effect on the original shipment and vice versa.

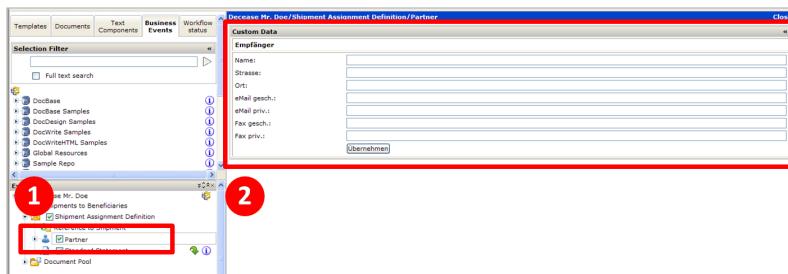


Note

A shipment reference can be converted into a copy of the referenced shipment (if the configuration of the event allows it) but the reverse is never possible: One cannot switch back to a reference of a shipment once it has been converted into a copy.

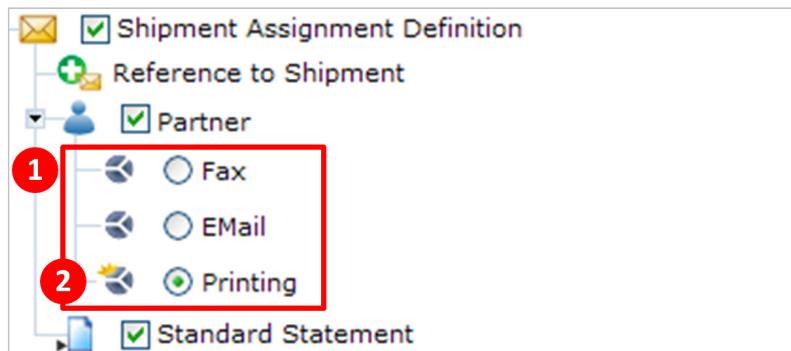
4.1.3.2. Recipient

Every shipment may have one or many recipients (1). A recipient is based on a recipient definition. Typical recipient definitions might be: Legal entity, person, beneficiary, partner etc. Usually the recipient information differs between recipient definitions. A legal entity may require the user to enter a company name while a person has a name and an address. The recipient specific information is displayed on the right hand side (2).



4.1.3.3. Channel

Each recipient has a list of channels the shipment may be sent through (1). This list is configured by the staffer or may depend on user-data entry. The user can choose one of the available channels. Her preferred channel is displayed with a specific icon (2) and is the default channel.



Each document and attachment also has a list of channels that it is allowed to be sent through. The list may even be different depending on whether the original of a document is sent or just a copy. This allows a configurator to define a shipment such that the original of a document (e.g. a policy statement) is **always** sent via physical mail, while its copies may be forwarded via email.

The available channels displayed to the user are dependent on the combination of allowed channels as defined by each document, attachment and recipient within the shipment.

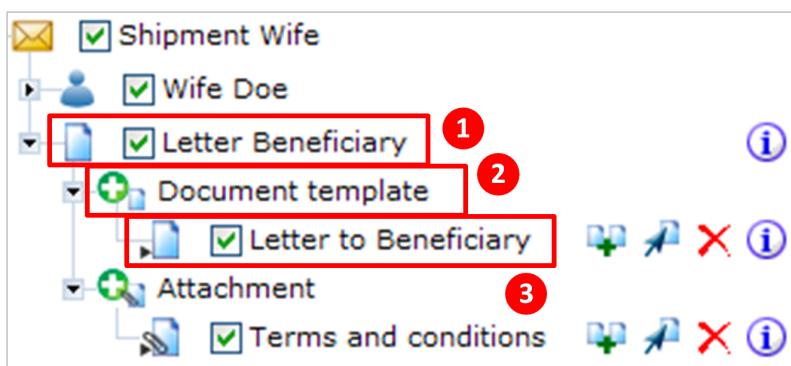


Note

It is possible, that no channel is available. This can be the case if the allowed channels of the recipient and the documents / attachments mutually exclude each other. Try removing documents to see if the channels reappear.

4.1.3.4. Document

The configurator may define documents that belong to a shipment (1). These cannot be removed in any way. In addition she may also allow the staffer to add documents if she wishes to. She therefore adds a document assignment symbol (2). The staffer may now drag & drop a document from the document pool onto the document assignment - a reference to the document is displayed (3).



Note

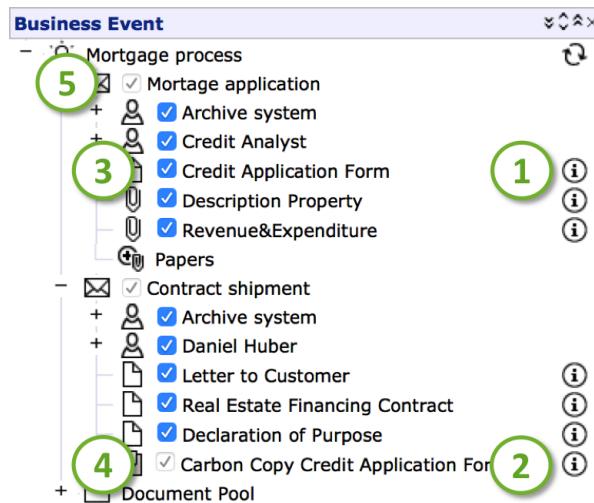
Documents that have been assigned are added as a reference by default. The mechanisms to convert the reference to a copy are identical to the ones described for Section 4.1.3.1, "Shipment reference".

4.1.3.5. Carbon Copy

In a Business Event Instance, an exact copy of the referenced document (1) is delivered for all recipients, shipments or output channels where the carbon copy has been added (2) if the ori-

ginal document is delivered, that is to say, if all rules applied to the original document evaluate to true.

Manually selecting/deselecting the original document (3) in a Business Event Instance automatically selects/deselects the carbon copy (4). In contrast, deselecting the whole Shipment (5) containing the original document does not automatically deselect the carbon copy, which be delivered.



4.1.3.6. Attachment

Depending on the attachment source described in Section 3.6.4, "Attachments", attachments are handled in two different ways:

- **Uploaded attachments:** The **Upload Attachment** action (1) opens the dialog to choose a file from the file system and to upload it. The attachment is only available in the context of the current Business Event Instance. It is not possible to re-upload a document. An incorrectly uploaded or outdated document must be removed, and the new document must be uploaded. Uploading a document with an already existing name creates two attachments with the same name.

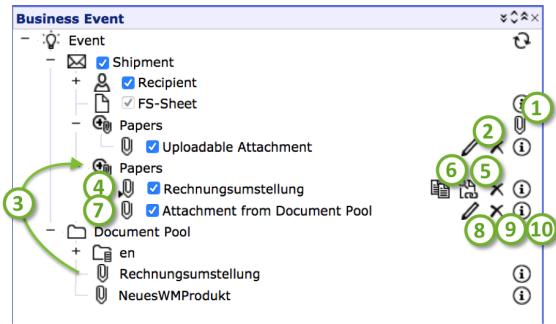
If configured in the Business Event Template, names, keywords and labels can be assigned to the attachment, or edited (2).

- **Attachments from the document pool:** Attachments from the document pool can be dragged and dropped here (3).

If the attachment is referenced (4), the original attachment can be displayed in the document pool (5), or it can be replaced with a copy (6).

If the attachment is a copy (7), names, keywords and labels can be assigned to the attachment, or edited (8).

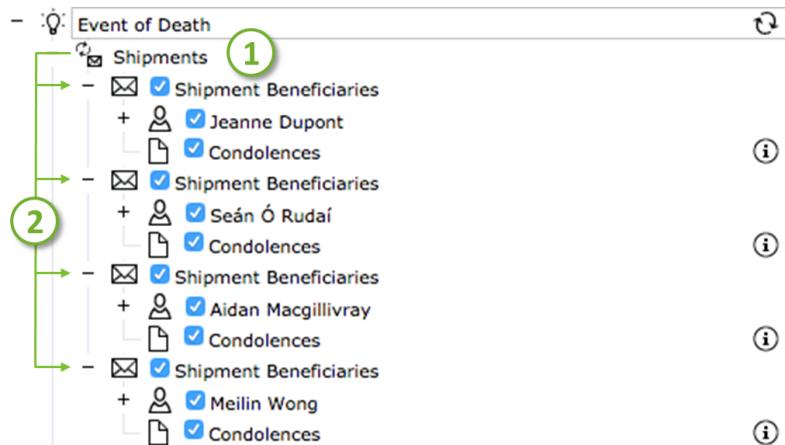
All attachments can be deleted (9), and their information can be displayed (10).



4.1.4. Shipment repetition

The number of shipments may be unknown at the time that an event definition is created. If an event triggers sending one shipment to each family member, the number of shipments equals the number of family members. However the number of family members is only known when the staffer works on a concrete case. Therefore it is possible to define a shipment repetition, that is either executed, when an event is instantiated or when it is rendered. The following description focuses on the first case.

When the event is instantiated, the business data leads to the creation of several shipments (2) which are all allocated underneath the shipment repetition symbol (1).



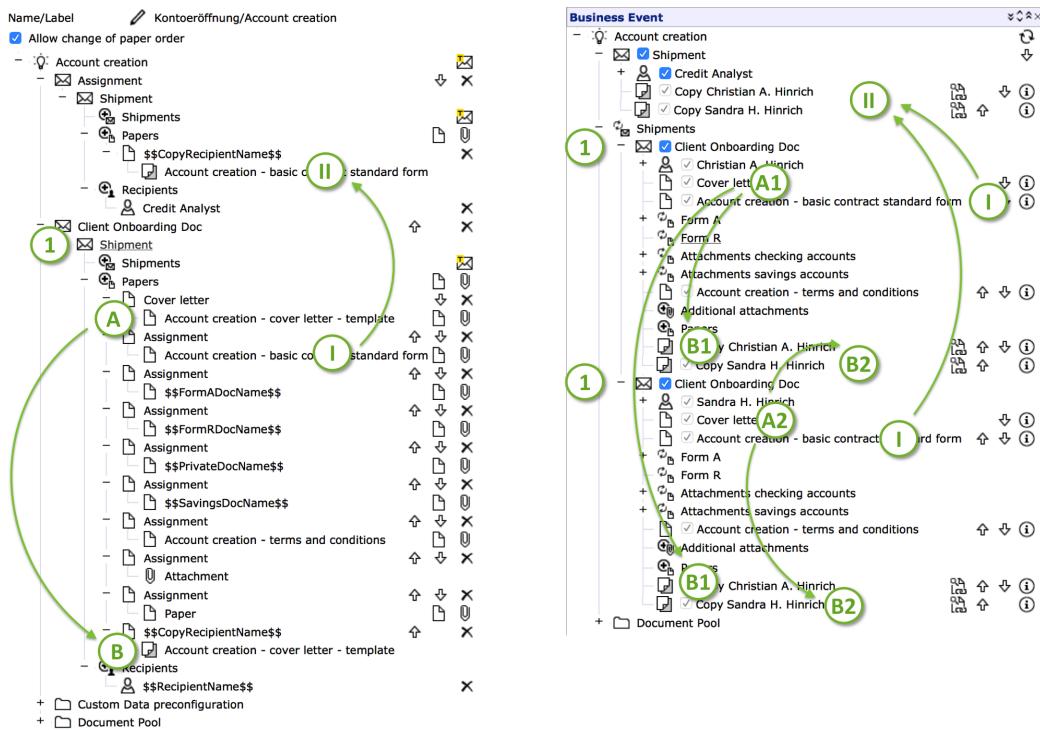
Note



Whenever a change to the business data is performed and the event is recalculated (see Section 4.1.6, "Recalculation"), new entries are created and entries might be removed. However, existing entries that have not been removed are not reset to their original state. This way, changes performed by the user are kept even in case the business data changes.

Carbon copies in repetitions. Carbon copies behave uniquely in repetitions. The instantiation of a Business Event Template into a Business Event Instance creates as many carbon copies as the number of times the shipment is repeated depending on the repetition variable. Usually, the repetition variable refers to the recipients which are either delivered by runtime data or manually provided by the Business User.

Table 4.1, "Carbon copies in repetitions" below illustrates the configuration of two carbon copies on the left side and the instantiated carbon copies on the right.

Table 4.1. Carbon copies in repetitions

The left side shows the **configuration** of a Business Event Template with two main shipments. A repetition over recipients is defined on the second shipment (1).

For the basic contract standard form (I), a carbon copy is inserted into the first shipment without repetition (II).

For the cover letter (A), a carbon copy is inserted into the second shipment with repetition (B).

The right side shows the **instantiation** into a Business Event Instance. It contains three shipments; the last two resulting from the configured repetition (1).

Since the basic contract standard form (I) has been instantiated by the repetition twice, once for Christian A. Hinrich and the other for Sandra A. Hinrich, the first, unpeated shipment contains the carbon copies of both forms (II).

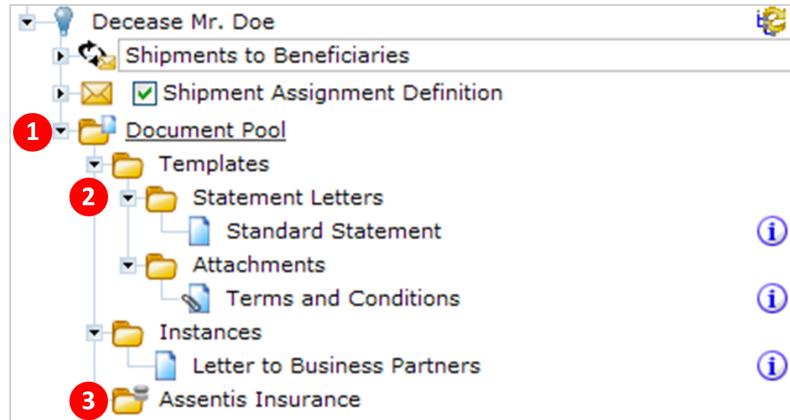
The cover letter also has been instantiated twice (A1 and A2). Since the carbon copy of the cover letter has been added to the repeated shipment configuration, each cover letter is attached to each instantiated shipment created by the repetition. This results in two carbon copies for the cover letter of Christian A. Hinrich (B1) and two carbon copies for the cover letter of Sandra H. Hinrich (B2).

4.1.5. Document pool

Every event contains a document pool (1) which includes all the documents and attachments available for use within the current event.

The document pool is subdivided into configurable logical folders (2). This logical structure may differ between event definitions to reflect differences in the business context and make document selection easier for the staffer. This allows a document to find itself in different folders depending on the type event.

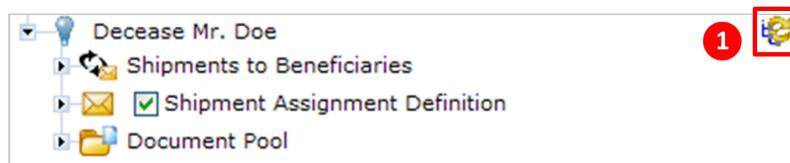
The document pool may also contain repository folders (3) that "point" directly to a folder in the repository. This means that a document referenced using a repository folder will find itself at the same location independent of the event.



4.1.6. Recalculation

The structure of an event (document, attachment, shipment and recipient assignments) is defined by rules as described in Section 3.1.3, "Rules". They can be static or rely on Conditions which evaluate values from the user data and/or the user input. To update the event structure when user data and/or user input is changed, the button marked in Figure 4.1, "Recalculation button" triggers a manual recalculation.

Figure 4.1. Recalculation button



When this button is clicked, user data are reloaded though previously provided user input from Forms is not overwritten. With the updated data, all Conditions are reevaluated, which may lead to differently resolved Condition-based rules, and a changed event structure.

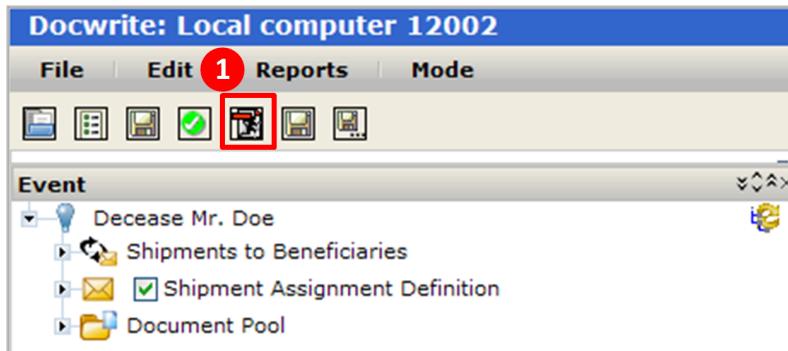


Note

If the `PreselectIf` rule is configured statically (`true` or `false`), the global system configuration defines if the user's selection is reset on recalculation or not. Ask the system administrator for details.

4.1.7. Preview

It is possible to preview the event by clicking on the corresponding menu entry (1). The preview is a PDF document containing all documents and attachments within the shipment.



4.2. Workflow on business events

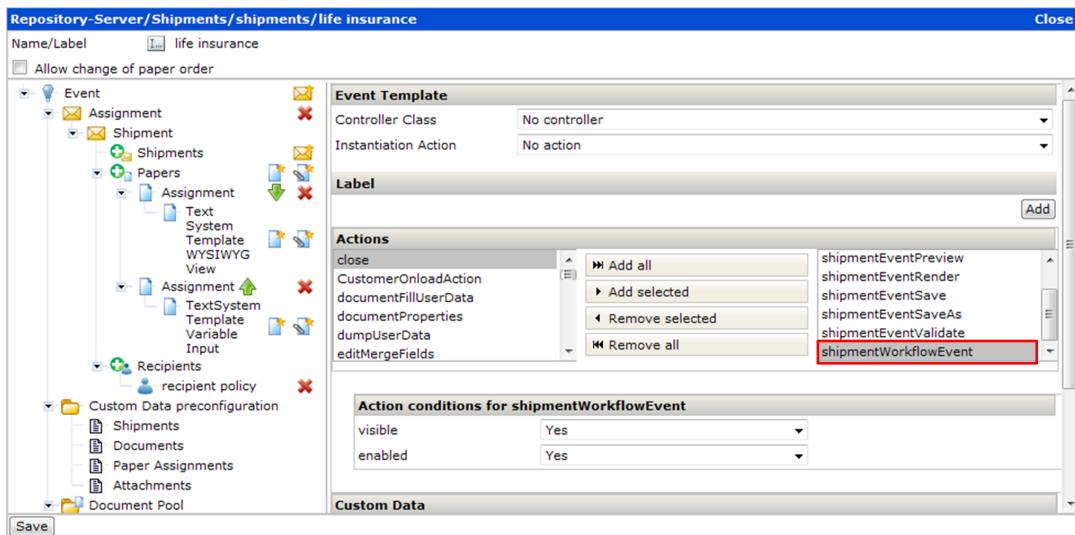
As for other resources in DocWrite also for business events a workflow can be defined. See "DocAdmin Online Help" to find out, how to activate the workflow for DocFamily.

For business events two steps are necessary to assign a workflow.

As for business event templates a workflow has to be assigned in DocAdmin or DocWrite (see Section 3.10, "Workflow on business event templates").

Furthermore during the creation of a business event template the action "shipmentWorkflowEvent" has to be added to the template definition. Only if this has been done, the workflow controls will be displayed for the business event.

Figure 4.2. Assigning action in business event template



Warning

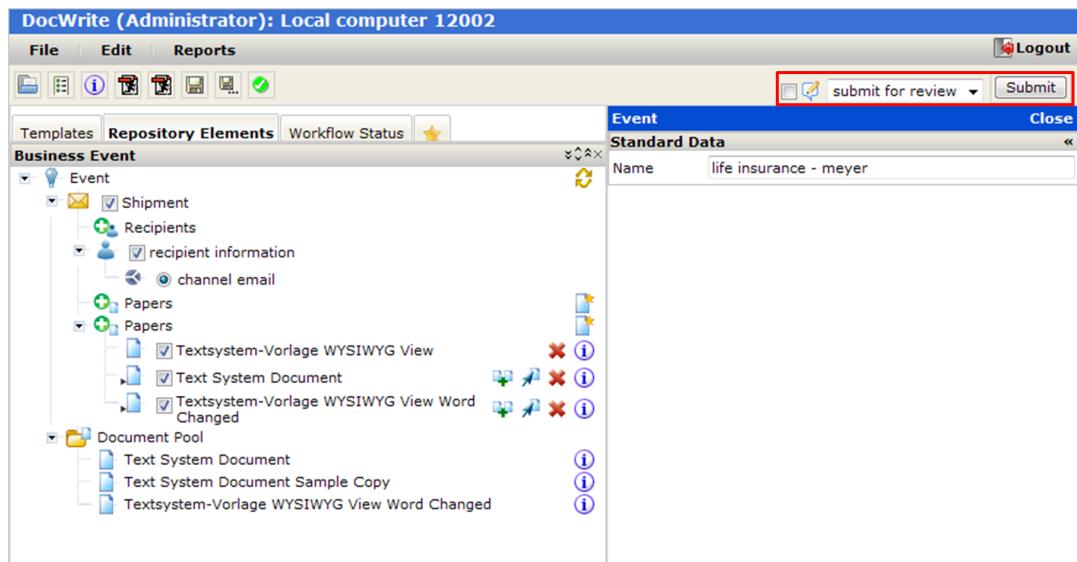


For business events only the dedicated "shipmentWorkflowEvent" action can be configured. The common action "workflowEvent" will have no effect during editing the event.

While creating and manipulating business events, controls will be displayed in case an event occurred that triggers the workflow.

The controls of the business event are located at the top right site of the browser window underneath the button "Logout".

Figure 4.3. Workflow controls in the business event



The workflow control will be displayed if the user has executed a workflow triggering action and does not close the editor at the same time.

In case he closes the editor with this action, a dialog will be displayed, that informs the user about the proceeding in the workflow.

General information about workflows can be found in "DocAdmin Online Help". Information about working with workflows in DocWrite is described in the document "DocWrite User Guide".

Chapter

5

Integration options

The role of an integrator is to perform the following tasks:

- Creating custom GUIs for object definitions
- Creating custom validators
- Creating custom reactions on changes
- Programming event definitions and events
- Designing document templates
- Integrating Shipment Management business systems

The listed tasks (with the exception of Section 5.1, “Designing document templates”) require programming skills (mainly Java) and the knowledge of how Shipment Management works in principle. API details are listed in the Javadoc and XML Schema definitions delivered with the application.

The relevant Javadocs may be found at [DocRepo Installation Directory]/Client/Javadoc/Index.html:

```
com.assentis.docrepo.common.shipment iface  
com.assentis.docrepo.common.shipment iface.callback  
com.assentis.docrepo.common.shipment iface.customdata  
com.assentis.docrepo.common.shipment iface.model
```

The XML Schemas EventDefinition.xsd, Event.xsd, ChannelDefinition.xsd and RecipientDefinition.xsd can be found in the subdirectory com/assentis/docrepo/common/shipment iface/xsd of the file DocRepo-client.jar at [DocRepo Installation Directory]/tomact/webapps/DBLayer/WEB-INF/lib

5.1. Designing document templates

Document templates for events can be built the same way as other document templates. No special precautions need to be taken as long as the document template does not refer to inform-

ation within the shipment. This is usually true for static documents such as terms and conditions or documents that are only populated with data from a delivering business system, e.g. an invoice.

If a document should include event (structure) information, the designer may use system variables. Typically, event information includes:

- The name and address of a recipient defined by the staffer.
- A list with the documents and attachments added to a cover letter by the staffer.
- A contract number stored in the custom data section of the event.

As one can imagine, these scenarios occur quite often. So although the document designer **theoretically** creates documents without event information, she will **practically** include event information most of the time.

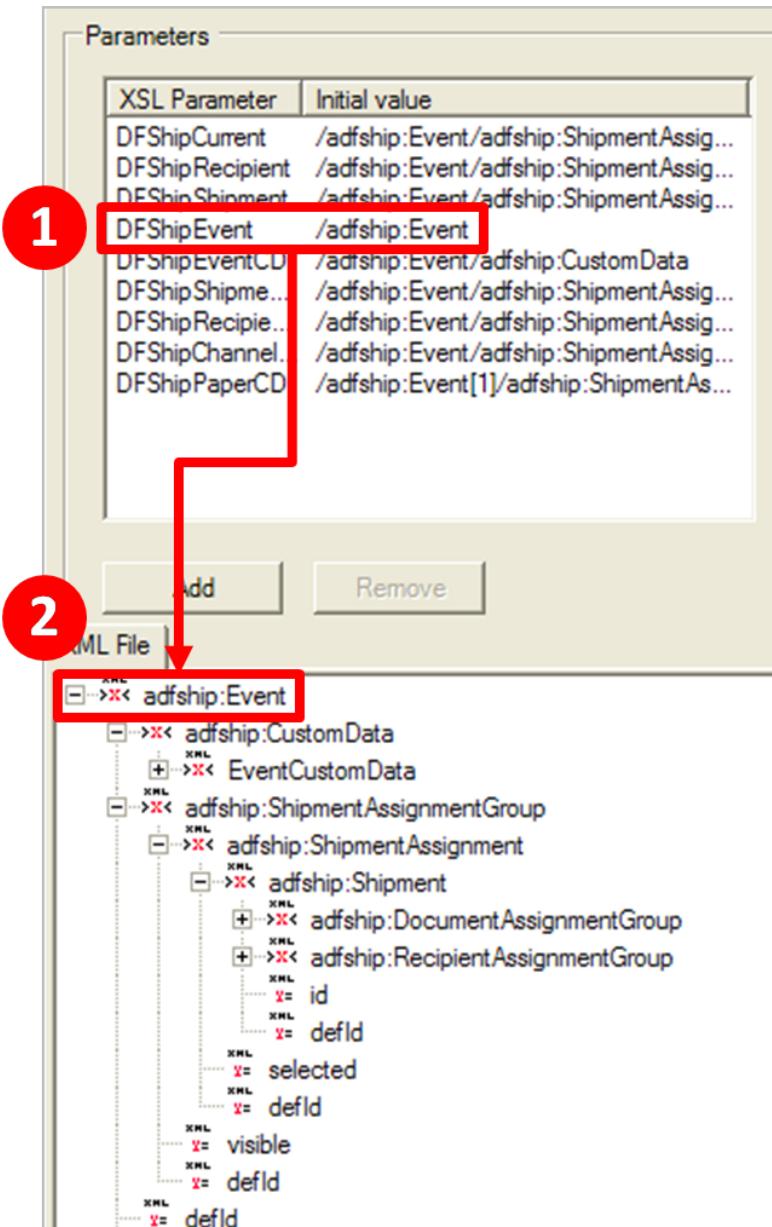
Event information is included by using a set of system variables, which may be used in combination with XPath statements.

Table 5.1. List of system variables for events

Systemvariable	Function
DFShipEvent	The root node of the current event
DFShipEventCD	The node containing the custom data of the current event
DFShipShipment	The node of the actual shipment in the event
DFShipShipmentCD	The node containing the custom data of the current shipment
DFShipShipmentRI	This variable only makes sense, if a repetition rule was used to create the shipment. A repetition rule always contains a reference to a nodeset that defines the number of occurrences of shipments. DFShipShipmentRI returns the specific node that was the reason for creating the shipment. If e.g. a set of business partners in the custom data section of the event was used to create the shipments, the variable returns the specific business partner for the actual shipment.
DFShipRecipient	The node of the actual recipient
DFShipRecipientCD	The node containing the custom data of the current recipient
DFShipRecipientRI	This variable only makes sense, if a repetition rule was used to create the recipient. A repetition rule always contains a reference to a nodeset that defines the number of occurrences of recipients. DFShipRecipientRI returns the specific node that was the reason for creating the recipient. If e.g. a set of addresses in the custom data section of the event was used to create the recipients, the variable returns the specific address for the actual recipient.
DFShipChannel	The node of the actual channel
DFShipChannelCD	The node containing the custom data of the current channel
DFShipPaperCD	The node containing the custom data of the current document or attachment node
DFShipPaperRI	The repetition item of the current document (within a repetition)
DFShipPaperSPEID	The nodes of all documents underneath the actual document. This can e.g. be used to list all attachments on a document
DFShipCurrent	The current node

5.1.1. Document template creation in DocDesign

When preparing a document template in DocDesign, these variables can be used. They need to be added manually (1) to the document as parameters before they can be used. Ideally a data source with the event structure is attached and the parameters' initial paths are mapped to it (2). This allows the document to be previewed as if they are used inside of a shipment:



The system variables may also be used to define textsystem variables that are used inside of textcomponents. This way, configurators can put together components for staffers, that allow including shipment structures into the documents sent to recipients.

Detailed information on the creation of DocDesign document templates can be found in *DocDesign Online Help*, or *HowTo - Create a document template*.

5.1.2. DocWrite Template preparation in DocAdmin

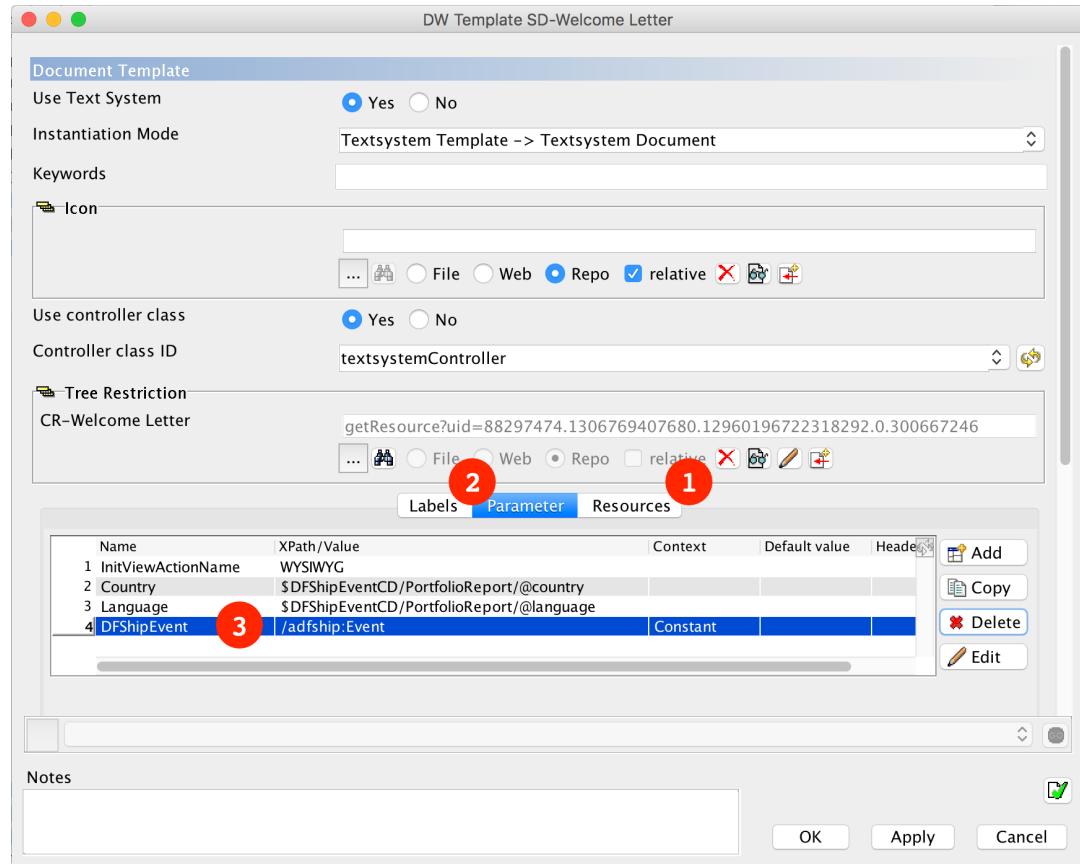
When building Business Event Templates, Text System Templates and Text System Documents are added which are based on DocDesign document templates. Usually, these Text System

Templates and Text System Documents are designed and tested separately. The DocWrite Template from which the Text System Templates and Text System Documents are created, needs to have the same configuration of parameters as the DocDesign document template. Thus, the Variables relying on the event information can be displayed in the *Variables* view in DocWrite.

In the DocWrite Template editor in DocAdmin, the DocDesign document template is selected on the *Deployment Package* tab under *Resources* (1). Likewise under *Resources*, on the *Forms* tab, the matching initial XML file is referenced. It is the same file that was used while designing the DocDesign document template, and while setting the parameter for the system variables (see Section 5.1.1, “Document template creation in DocDesign”).

The parameters for the system variables must be defined under *Parameter* (2). For every parameter in the DocDesign document template, a parameter with the exact same name has to be added (3). The XPath field also contains exactly the same XPath as the DocDesign parameter. The context must be set to **Constant**.

Figure 5.1. DocWrite Template Editor



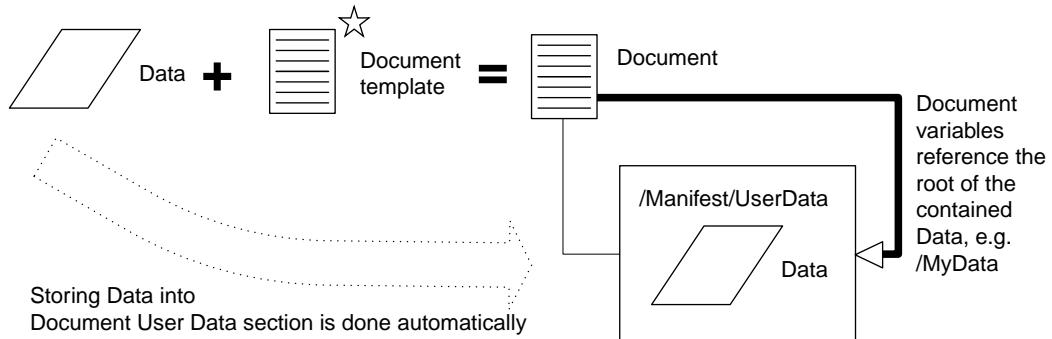
Find detailed information on the creation of DocWrite Templates in *DocAdmin Online Help*. The creation of Text System Templates and Text System Documents is described in *DocWrite User Guide*.

5.2. Data processing

5.2.1. Data processing without shipment

When a DocWrite document is created, input data is merged with information from the DocWrite template (DW template) and stored in the user data section of the document. The data is delivered by an input data provider or a default XML datafile stored within the DW template.

Any XPath statement in the document template (i.e. the deployment package) or in variables, conditions, expressions etc. refers directly to that datafile. This makes life easy to developers and configurators as they can ignore any surrounding structures:

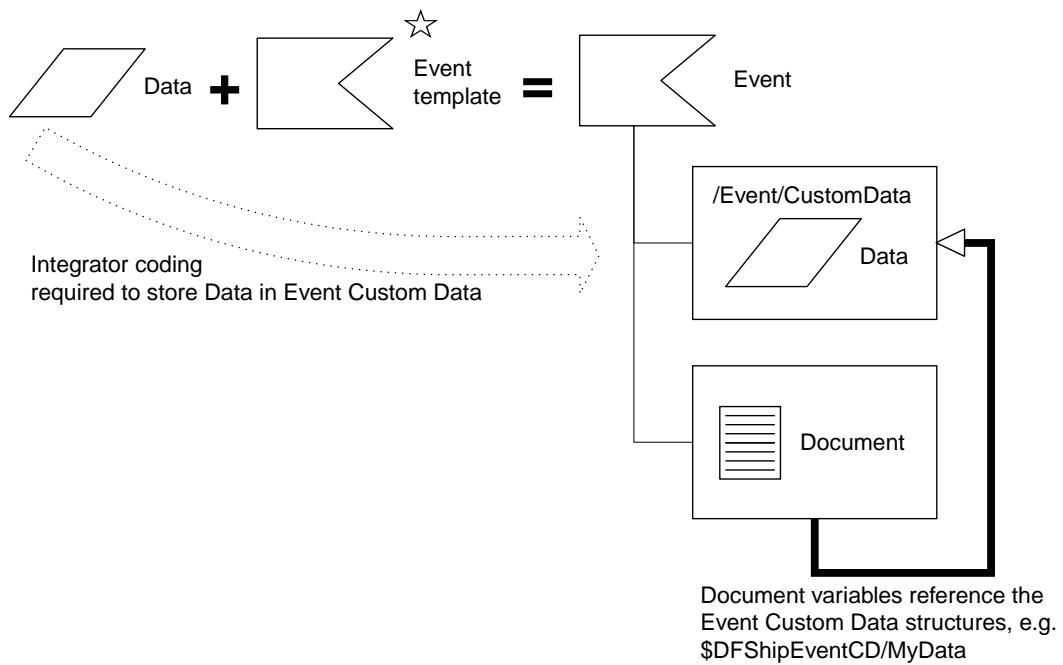


5.2.2. Shipment data processing

When documents are used in conjunction with the shipment functionality, input data needs to be handled differently.

First, the integration needs to make sure, that the data gets stored in the custom data section of the created event. Alternatively, the data may not be present when the event is created but loaded afterwards. A concrete example consists in an insurance claim case, where the end user first selects a customer from a list and clicks OK, which leads to the corresponding customer's data being loaded into the event's custom data section.

Second, the document template developer (and any other writer of an XPath statement) will have to reference the data in the event custom data section as described in Section 5.1, "Designing document templates".



While theoretically it is possible to store the data in any other section of the event, that allows storing custom data (e.g. the document assignment), it is often easiest, to just put it into the event custom data section.

Caution



It is not recommended to mix the way of handling data within a shipment with the way of handling data with documents that have no relationship with the shipment functionality, i.e. documents should reference event custom data instead of storing data in the user data section. Since events are not instantiated like traditional documents, XPaths that do not reference the shipment structure are likely to fail retrieving any data or will just pull data that is stored in the default document data set (the one configured in the DocWrite template).

Caution



Integrators and configurators must not use XForms with documents inside of events.

5.2.3. Shipment based serial letter processing

5.2.3.1. Introduction

As event custom data are stored with the event, the number of shipment assignments created by repetition rules based on the embedded custom data is limited by the available memory during shipment processing. To virtually support any number of repetition based event elements created by repetition rules, event custom data can be provided during production time of a shipment.

When using that functionality, an event still stores custom data. These will provide data during preview time, but will be replaced during production time. During production time, event custom data will be sourced from a specific configurable data source. During this process, the whole event is being recalculated so that the event structure corresponds to the custom data being inserted.

The default implementation provided with the product retrieves custom data that is to be inserted from any XML document based on a configurable element path.

If you already defined repetitions on your event template, enabling runtime repetitions is as easy as activating "production time repetition" in the event template. Data is provided, that will be used as event custom data when executing the DocBase process `adb_shipmentRuntimeSequenceProcessor` with a corresponding configuration (that references the shipment event and the data source).

5.2.3.2. Parts list

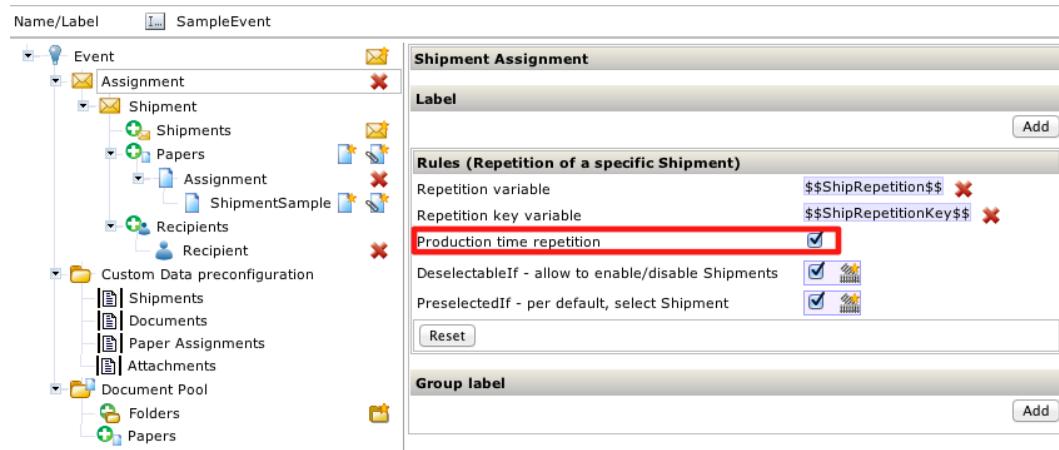
To execute a shipment based serial letter that retrieves its repetition data at runtime, you need the following:

- a configured shipment event instance stored in DocRepo
- a DocBase jobcard to execute the process `adb_shipmentRuntimeSequenceProcessor` and to provide necessary configuration details for it

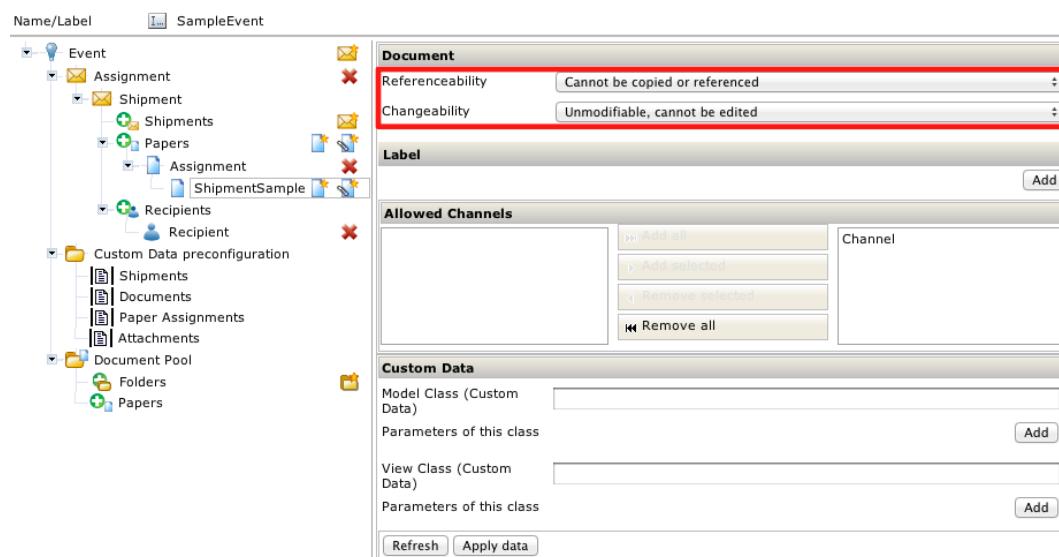
5.2.3.3. Preparing a shipment event

Shipment event instances that shall be used for serial letters at runtime must be instantiated from a shipment event definition, which must have some specific settings applied:

- "Production time repetition" must be enabled on Shipment Assignment level



- In paper assignments, the "Referenceability" must be set to "Cannot be copied or referenced" and "Changeability" must be set to "Unmodifiable, cannot be edited"



When the above mentioned conditions are met by the shipment event definition, you create a event instance from it. This instance becomes the template for the shipment serial letter during runtime.

5.2.3.4. Creating the jobcard

To run the shipment as serial letter we will use a jobcard. Instead of using a jobcard you may also use any other method to start processes and to provide configuration details in DocBase, which is covered in the DocBase documentation.

The jobcard references the process definition `adb_shipmentRuntimeSequenceProcessor` and configures the task `adb_shipmentRuntimeSplitterTask`. The task requires the id of the event instance to be used as template for the serial letter and how to obtain custom data to create serial shipments from it.

eventElementID

This property needs to be set to the element id of the shipment event instance to be used as template.

delegate.customDataModelIterator

Use this property to configure the datasource which provides custom data to be inserted as event custom data into the event referenced by the eventElementId. DocBase comes with a default implementation (`com.assentis.db.service.splitting.DefaultCustomDataModelIterator`), which allows to source custom data from a XML stream.

The `DefaultCustomDataModelIterator` works similiar as the DocBase XML splitter. It requires a URL to be specified to read XML data from, a split path to identify a single fragment to be used as custom data, and a parent element, in case the fragment must be surrounded with elements. Each time, the `DefaultCustomDataModelIterator` detects an element matching the split path, new custom data is created and inserted into the template event before it gets rendered.

In case you need to connect to other data sources to provide data for your serial shipments, you may implement the interface `com.assentis.db.api.service.splitting.ICustomDataModelIterator` and configure it to be used within the jobcard.

delegate.customDataModelIteratorScope

Defines, which structure within the event receives the custom data. The only supported value is `EVENT`.

```
<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:adb-iocwf="http://www.assentis.com/schema/adb-iocwf"
    xsi:schemaLocation="http://www.springframework.org/schema/beans
        http://foo.assentis/schemas/spring-beans.xsd
        http://www.assentis.com/schema/adb-iocwf
        http://foo.assentis/schemas/adb-iocwf.xsd">

    <adb-iocwf:jobcard id="sample_shipmentRuntimeRepetitionJC">
        <adb-iocwf:process-ref value="adb_shipmentRuntimeSequenceProcessor" />

        <adb-iocwf:beanRuntimeProperties>
            <adb-iocwf:property
                targetBeanName="adb_shipmentRuntimeSplitterTask"
                targetPropertyName="eventElementID">
                <value>747158149.1323850119812.87164882551308634.0.1166777822</value>
            </adb-iocwf:property>
            <adb-iocwf:property
                targetBeanName="adb_shipmentRuntimeSplitterTask"
                targetPropertyName="delegate.customDataModelIterator">
                <bean
                    class="com.assentis.db.service.splitting.DefaultCustomDataModelIterator"
                    scope="prototype">
                    <property name="parentBegin" value="&lt;sample>"/>
                    <property name="customDataSplitPath" value="/sample/document"/>
                    <property name="inputUrl" value="file:///tmp/customDataEventSource.xml"/>
                </bean>
            </adb-iocwf:property>
            <adb-iocwf:property
                targetBeanName="adb_shipmentRuntimeSplitterTask"
                targetPropertyName="delegate.customDataModelIteratorScope">
                <value>EVENT</value>
            </adb-iocwf:property>
        </adb-iocwf:beanRuntimeProperties>
    </adb-iocwf:jobcard>
</beans>
```

5.2.3.5. Running the jobcard

The jobcard can be run in multiple ways such as:

- using the "Start Job" feature in DocAdminConsole

- using DocBase clients (command line or Java clients)
- defining a schedule to run the jobcard

5.3. Sending Shipments as email with attachments

Shipments of business events can be delivered to different channels and rendered to different formats. Sending a shipment as email is a special case in DocFamily and requires specific configuration.

The following configuration and preparation steps are required:

- Configuration of the `textsystemPreviewEhtml` DocWrite action on DocWrite template level.
- Configuration of the global `uploadToRepoAsXEP` DocWrite action to import attachments while keeping the original files.
- Configuration of the email channel.

Based on the configuration, `EML` files can be produced, and sender and recipient information can be stored.

For details read the following sections and refer to the mentioned documentation.

5.3.1. Preliminaries

The Shipment is sent as an `EML` file and needs to be organized in a way that the first paper is a document (Text System Template, Text System Template with Copies or Text System Document). This document becomes the email body. All other papers are added as `PDF` attachments to the email.

Any other document paper will be transformed to `PDF` following the usual render process. Depending by its format, papers of type `attachment` are either `PDF` files generated from the imported `XEP` file (for Files in internal format), or the original `PDF` file is attached (for Files in internal and original format). Find detailed information on the attachment types in "DocAdmin Online Help".

When processing shipments to `EML`, for each recipient, which has the email channel configured, a separate file is created.

5.3.2. Document action `textsystemPreviewEhtml`

The action is document-specific. It is assigned shipment-independently to a DocWrite Template. Therefore in DocAdmin in the Resources tab of the DocWrite Template the action is assigned. The action does not require any parameters, but all parameters available for the `renderOnly` action, can also be configured for `textsystemPreviewEhtml`.

The screenshot shows the DocAdmin configuration interface for managing actions. At the top, there are tabs for 'Deployment Package', 'Forms', and 'Actions'. The 'Actions' tab is selected, displaying a list of actions in a table:

Name	Action class
1 textsystemPreview...	textsystemPreviewEhtml
2 close	close
3 workflowEvent	workflowEvent
4 documentFillUser...	documentFillUserData
5 dumpUserData	dumpUserData

On the right side of the table are buttons for 'Add', 'Copy', and 'Delete'. Below the table, there are four configuration fields:

- Visible:** Yes
- Enabled:** Yes
- Name:** textsystemPreviewEhtml
- Action class:** textsystemPreviewEhtml

Below these fields is a 'Parameter' section with a table:

Name	XPath/Value	Context	Default value	Namespaces
1 ShowLocaleSelection	true	Constant		

On the right side of this table are buttons for 'Add', 'Copy', 'Delete', and 'Edit'.

This change applies only to documents that are instantiated from the DocWrite Template after this configuration. Find further information on DocWrite actions in "DocAdmin Online Help".

5.3.3. Extension of global action `uploadToRepoAsXep`

The `uploadToRepoAsXep` action allows the user to upload files of specific formats (PDF, PS, etc.) and to store them in the database. This action is document-independent. It usually is configured as global action in DocWrite and available for all users, possibly restricted by roles and permissions. So configuring this action will affect all users.

Per default on upload, the upload file is transformed into the intermediate format (XEP). The file extension will be `.xep` and the type is displayed as "File in internal format" in DocAdmin.

With the following configuration on upload, the file is also transformed into the intermediate format (XEP) and additionally the original file is kept. Both, the XEP and the original file are put into a container, which is stored in the database. The file extension will be `.mfc` and the type is displayed as "File in internal and original format".

In DocAdmin edit the `uploadToRepoAsXep` action with the default editor. In the parameter group, the setting for the `keepOriginal` property has to be changed to `true`.

The screenshot shows the configuration of the `uploadToRepoAsXep` action. At the top, there is a 'Parameters' section. Below it is a table:

Name	XPath/Value	Context	Default value	Namespaces
1 keepOriginal	true	Constant		

On the right side of the table are buttons for 'Add', 'Copy', 'Delete', and 'Edit'.

To apply this configuration, DocWrite needs to be restarted. It applies only for attachments uploaded after this configuration. Find detailed information on the attachment types and the `uploadToRepoAsXep` action in "DocAdmin Online Help".

5.3.4. Email channel configuration

The configuration of the email channel is required in DocBase and in DocWrite.

DocBase. The configuration in DocBase is a global one and defines which channel induces the production of shipments as emails. When the `adb_shipmentEmailSequenceProcessor` is executed only shipments with an recipient with the email channel configured are produced in `EML` format.

The name of the channel is exposed in the configuration file `[AssentisResources-Dir]/spring/propertyConfigurer/core_placeholder.properties`. For the property `adb_shipmentInEmailDataProcessor` define the name of the email channel:

```
adb_shipmentInEmailDataProcessor.emailChannel =email
```

To apply this configuration, DocBase needs to be restarted.

DocWrite. Corresponding to the DocBase configuration, a channel element is required in DocWrite. The name of the channel has to match the value given in the “`core_placeholder.properties`” file. To define language-dependent display names, the usage of labels is necessary. Changing the channel name, would render the DocBase configuration useless and no shipment is produced as email.

Shipments are only send as email, if the recipient of the shipment has the email channel selected. This channel can either be assigned as preferred channel during recipient configuration or it is assigned in the business event on recipient or paper level.

Find more information on how to configure a channel and how to assign it to a recipient in Section 3.2, “Channel definition”, Section 3.3, “Recipient definition”, Section 3.7, “Recipient assignment” and Section 3.6, “Papers and paper assignments”.

5.3.5. Shipment based email processing

After the initial configuration of the system and of shipment, business event templates can be created and configured, which can be used for the instantiation of business events with shipments, that are supposed to be sent as email. To produce `EML` files out of a business event, a DocBase jobcard referencing the standard DocBase process `adb_shipmentEmailSequenceProcessor` can be configured.

In the following sample the most important setting is the property `eventElementID` (XPath `/beans/adb-iocwf:jobcard/adb-iocwf:beanRuntimeProperties/adb-iocwf:property[@targetPropertyName="eventElementID"]`). The value of this property needs to contain the element Id of the business event, which is supposed to be produced to `EML`.

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:adb-iocwf="http://www.assentis.com/schema/adb-iocwf"
       xsi:schemaLocation="http://www.springframework.org/schema/beans
                           http://foo.assentis/schemas/spring-beans.xsd
                           http://www.assentis.com/schema/adb-iocwf
                           http://foo.assentis/schemas/adb-iocwf.xsd">

    <adb-iocwf:jobcard class="com.assentis.db.starter.IOCWorkflowJobCard"
                       id="renderShipmentAsEMLAndAddToPostProcOut" scope="prototype">

        <adb-iocwf:process-ref value="adb_shipmentEmailSequenceProcessor"/>

        <adb-iocwf:beanRuntimeProperties>
            <adb-iocwf:property targetBeanName="adb_shipmentSplitterTask"
```

```
targetPropertyName="eventElementID">
<value>832138732.142192877717.92127627239492840.0.1166777822</value>
</adb-iocwf:property>
</adb-iocwf:beanRuntimeProperties>
</adb-iocwf:jobcard>
</beans>
```

The jobcard can be run by the DocBase Commandline client, or it is started via the DocAdmin Console "Start job" feature. From the business event, the process will create an `EML` file per shipment and recipient. It is stored in the `POSTPROCOUDATA` table of the database and therefore available for e.g. Post Processing. The execution will return a job Id, which can be used to identify the produced `EML` files navigating through the `Jobs` table of DocAdmin Console.

Creating and running jobcards is described in "DocBase Cookbook" and "DocBase Commandline Client". For more information on DocAdmin Console see "DocAdmin Console User Guide".

5.3.6. Providing sender and recipient information

To send a shipment as an email, sender and recipient information is required. This information can be provided via Variableset, assigned to the channel (see Section 3.2, "Channel definition").

The Variables of this Variableset have to point into the event user data. Using the predefined event system variables (see Section 5.1, "Designing document templates"), data files creating a business event can deliver sender and recipient information for every single shipment.

During the `EML` processing, these data fetched by the variable set will be stored as custom attributes in the `PPE_METADATA` table connected to the produced and saved `EML` file (entry in the `POSTPROCOUDATA` table). Custom attributes are filed as key-value-pairs, where the variable name is used as key. They are retrievable by DocBase standard mechanisms via job Id and `POSTPROCOUDATA` table entry.

Further information on Variablesets are to find in "DocAdmin Online Help", and on Variablesets in channels in Section 3.2, "Channel definition"

5.4. Formatting Shipments with PDF attachments to AFP

Technically, formatting a Shipment with attachments to `XEP`-based output formats, will convert the attachment to `XEP` and embed it in the Shipment `XEP` document before the formatting step. The quality of the output document depends by the reliability of the "attachment input format to `XEP`"-conversion.

For `AFP` formatting, the embedding of the original `PDF` attachment can be enabled. Then the `AFP` multi-page object container is used for the attachment.

Configuration. To enable the usage of the multi-page object container, two configuration steps are required:

1. In the DocBase configuration file `[AssentisResourcesDir]/DocBaseConfig.xml`, the section `/DocBaseConfig/DocCommon/FP` contains a property `useMultiPageObjectContainer`, which has to be set to `true`.
2. In DocAdmin the upload of attachments as `File` in `internal` and `original` format has to be configured for `DocWrite`. By setting the parameter `keepOriginal` on the `uploadToRepoAsXEP` `DocWrite` action, the attachment will save both, the file in `original` format and the file converted to `XEP`. Find more information in "DocAdmin Online Help".

Only if both configurations have been made, **AFP** formatting will use the multi-page object container and embed the original **PDF** attachment.