



SAP Policy Management Guide for Data Migration

SAP FS-PM 5.4

Version 1.0

March 2018

© Copyright 2018 SAP SE. All rights reserved.

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

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

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

IBM, DB2, DB2 Universal Database, System i, System i5, System p, System p5, System x, System z, System z10, System z9, z10, z9, iSeries, pSeries, xSeries, zSeries, eServer, z/VM, z/OS, i5/OS, S/390, OS/390, OS/400, AS/400, S/390 Parallel Enterprise Server, PowerVM, Power Architecture, POWER6+, POWER6, POWER5+, POWER5, POWER, OpenPower, PowerPC, BatchPipes, BladeCenter, System Storage, GPFS, HACMP, RETAIN, DB2 Connect, RACF, Redbooks, OS/2, Parallel Sysplex, MVS/ESA, AIX, Intelligent Miner, WebSphere, Netfinity, Tivoli and Informix are trademarks or registered trademarks of IBM Corporation.

Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.

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

Oracle and Java are registered trademarks of Oracle.

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

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

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

SAP, R/3, SAP NetWeaver, Duet, PartnerEdge, ByDesign, SAP BusinessObjects Explorer, StreamWork, SAP HANA, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and other countries.

Business Objects and the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Decisions, Web Intelligence, Xcelsius, and other Business Objects products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Business Objects Software Ltd. Business Objects is an SAP company.

Sybase and Adaptive Server, iAnywhere, Sybase 365, SQL Anywhere, and other Sybase products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Sybase, Inc. Sybase is an SAP company.

All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

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

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

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

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

SAP NetWeaver "How-to" Guides are intended to simplify the product implementation. While specific product features and procedures typically are explained in a practical business context, it is not implied that those features and procedures are the only approach in solving a specific business problem using SAP NetWeaver. Should you wish to receive additional information, clarification or support, please refer to SAP Consulting.

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

Disclaimer

Some components of this product are based on Java™. Any code change in these components may cause unpredictable and severe malfunctions and is therefore expressively prohibited, as is any decompilation of these components.

Any Java™ Source Code delivered with this product is only to be used by SAP's Support Services and may not be modified or altered in any way.

Document History

Version	Date	Description
1.0	2017-04-27	First version of this guide

Typographic Conventions

Type Style	Description
<i>Example Text</i>	Words or characters quoted from the screen. These include field names, screen titles, pushbuttons labels, menu names, menu paths, and menu options. Cross-references to other documentation
Example text	Emphasized words or phrases in body text, graphic titles, and table titles
Example text	File and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools.
Example text	User entry texts. These are words or characters that you enter in the system exactly as they appear in the documentation.
<Example text>	Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system.
EXAMPLE TEXT	Keys on the keyboard, for example, F2 or ENTER.

Icons

Icons	Description
	Caution
	Important
	Note
	Recommendation or tip
	Example

Contents

1	Disclaimer.....	1
2	Introduction.....	2
3	References and Information Sources.....	3
4	Glossary	4
4.1	Terms.....	4
4.2	Abbreviations	6
5	General Information on Legacy Data Migration	7
5.1	Process Steps for the Data Migration.....	7
5.2	Migration Scenarios	7
6	Overview of Legacy Data Migration to FS-PM	8
6.1	Basics	8
6.2	Last Version Migration	11
6.3	Historical Migration	12
6.4	Migration of Business Transactions.....	14
6.5	FS-PM Migration Interfaces.....	16
6.6	Not in the Scope of Functions	16
7	Details for the Migration of Policies to FS-PM.....	17
7.1	Overall Process for FS-PM Migration	17
7.2	Authorization and Status Concept	18
7.2.1	Authorization	18
7.2.2	Migration Status	18
7.2.3	Business Processes for Migrated Policies	21
7.3	Technical Migration	21
7.3.1	Prerequisites	21
7.3.2	Scope of Technical Migration	22
7.3.3	Migration Data.....	26
7.3.3.1	General Requirements for the Migration File	26
7.3.3.2	Display of Policy Business Object and Its Versions in Migration File	26
7.3.3.3	Preassignments and Initial Fields.....	28
7.3.3.4	Taking Account of Templates from In-Force Business Configurator in Migration	28
7.3.4	Special Features of Historical Migration	30
7.3.5	Migration of Business Transactions: Reversal.....	31
7.3.6	Migration of Business Transactions: Perform Premium Waiver (Life)	31
7.3.7	Additional Data from the Legacy System.....	31
7.3.8	Migration of Scheduling Dates.....	32
7.3.8.1	Migration of Regular Scheduling Dates.....	32
7.3.8.2	Migration of Related Dates.....	32
7.3.9	Enhancement and Settings Options	33
7.3.9.1	Enhancement of Migration Data Structures with Non-SAP Fields	33
7.3.9.2	Enhancement of the Migration Level of Functions	34
7.3.10	Historical Migration of Policies with Actual Investment.....	34
7.4	Functional Validation	36
7.4.1	Prerequisites	37

7.4.2	Scope of the Functional Validation	37
7.4.3	Transaction for Functional Validation	45
7.4.4	Logging of the Validation Run.....	47
7.4.4.1	Display of the Application Log	47
7.4.4.2	Selection of Messages to Be Logged and Considered	48
7.4.5	Update in Validation for Historical Migration.....	48
7.4.6	Deletion of Temporary Data of the Migration of Policies	49
7.4.6.1	Transaction for Deletion of Temporary Data	49
7.4.6.2	Deletion of Temporary Data in the Validation.....	49
7.4.6.3	Deletion of Temporary Data when Starting Business Processes..	50
7.4.7	Release in the Validation	50
7.4.8	Migration of Business Transactions	51
7.4.9	Document Processing in Policy Migration	53
7.4.9.1	Treatment of Cash Flow Documents in Policy Migration	53
7.4.9.2	Treatment of Non Cash Flow (G/L) Documents in Policy Migration.....	56
7.4.9.3	Treatment of Account Balances in Policy Migration	57
7.4.10	Other Comparison Values when Comparing the Calculated Policy Version with Technically Migrated Policy Version.....	57
7.4.11	Handling of Entities to Be Deleted in the Release	57
7.4.12	Treatment of Versions Migrated for Display Only	58
7.4.13	Correspondence	59
7.4.14	Enhancement and Settings Options	60
7.4.14.1	BAdI: Functional Validation of Policies – Settings (/PM0/ABQ_POL_VAL_CNTRL_BADI).....	60
7.4.14.2	BAdI: Functional Validation of Policies – Error Handling (/PM0/ABQ_POL_ERR_HANDL_BADI).....	60
7.4.14.3	BAdI: Functional Validation of Policies – Calculation Using Product Engine (/PM0/ABQ_POL_CALC_BADI)	60
7.4.14.4	BAdI: Functional Validation of Policies – Comparison After Calculation (/PM0/ABQ_POL_COMP_BADI).....	61
7.4.14.5	BAdI: Functional Validation of Policies – Statistics (/PM0/ABQ_POL_STATISTIC_BADI)	61
7.4.14.6	BAdI: Functional Validation of Policies – Migration of Business Transactions (/PM0/ABQ_POL_BT_X_MIG_BADI)	61
7.5	Mass Deletion	61
7.5.1	Enhancement and Settings Options	62
7.5.1.1	BAdI: Deletion of Policy Data (BAdI /PM0/ABQ_POL_DEL_BADI)	62
7.6	Testing the Migration of Policies.....	62
8	Details for the Migration of Other Business Objects to FS-PM	64
8.1	Migration of Insurable Objects (INBO)	65
8.1.1	Technical Migration.....	65
8.1.2	Migration of Object-Extending Data.....	65
8.1.3	Mass Deletion of Insurable Objects	66
8.1.4	Testing the Migration of Insurable Objects	67
8.2	Master Policy	67
8.3	Distribution Plan.....	67
8.3.1	Technical Migration.....	67
8.3.2	Functional Validation.....	68
8.3.3	Mass Deletion of Distribution Plans	69
8.3.4	Testing the Migration of Distribution Plans	69

8.4	Fund Management Data	69
8.5	Special Risk Notifications	69
8.6	Miscellaneous	69
9	Integration of Operational System Data in an FS-PM Migration Project.....	72
9.1	Operational Systems in General.....	72
9.2	Commission System/Portfolio Assignment.....	78
9.3	Reinsurance System	78
10	Treatment of Add-On LoBs for the Migration of Policies	80
10.1	Enhancing the Migration Structures for the Migration of Data from Entities Not Delivered in the SAP Standard System.....	80
10.2	Enhancement of the Migration Level of Functions	81
11	Performance Aspects.....	83
12	Recommendations.....	84
12.1	Check Lists	84
12.2	Recommended Procedure.....	85
13	Appendix	88

1 Disclaimer

This guide is not part of SAP product documentation. By using this guide, you agree that (unless expressly stated otherwise in your agreements with SAP) you may not infer any product claims against SAP based on this information.

2 Introduction

This guide contains information, notes and recommendations on how you can migrate in-force business management data from your legacy system to FS-PM.

All of the important steps that have to be carried out for an FS-PM migration project in various migration scenarios are covered in this guide.

You are informed which programs and tools SAP provides and how you should use them. Enhancement options that you can use to adapt the migration functions delivered to the specific requirements of your project are described in detail.

3 References and Information Sources

General cross-modal information on the procedure and method of migrating legacy data to SAP applications is documented in the ASAP Implementation Roadmap. You can find this in SAP Support Portal in the SAP Solution Manager area. In-force business management data should also be migrated from your legacy system to FS-PM in accordance with the procedure described there.

For a summary overview of the most important steps in a migration in FS-PM, see [SAP Policy Management](#) -> *Application Help* -> *SAP Library for SAP Policy Management* -> *In-Force Business Management* -> *Migration in SAP Policy Management*.

The Legacy System Migration Workbench (LSMW) available in SAP Basis also provides you with detailed help for the application.

The programs and enhancement options described in this guide and delivered with FS-PM are documented in detail in the system.

In addition to the migration tools and functions available in the FS-PM standard system, you can request SAP for consultation services for carrying out an FS-PM migration project. This enables you to take advantage of additional specialized tools and additional best practices.

4 Glossary

4.1 Terms

Terminology used in a migration context is not always used in the same way. Below are definitions of the most important migration-specific designations that are used in this guide.

Term	Definition
Key date	Date on which the new system supersedes the legacy system and commences productive operation (Go Live). Different key dates are possible for various data segments of the same application.
Migration date	Field that controls the FS-PM migration and validation run, and sets the date stamp for the data records. In most cases this date corresponds to the date on which the migration was made.

Big-Bang migration	Date of the move to the new software platform, where all of the new systems affected by the migration project are made operational on a specified key date, and the legacy systems completely replaced. This means that from the key date, all of the operational systems and data sources supply only the new systems.
Gradual migration	The movement of data to a new software platform whereby the dataset is transferred from the legacy system to the new system in phases and the legacy system is gradually phased out. In a gradual migration, the operational systems that serve the migration project can also become operational in phases on different key dates.
Policy version	Version of a policy that is valid for the period between the previous and the next processing of the policy. Every complete and released processing activity creates a new legally binding policy version.
Latest version	The chronologically latest valid policy version of a policy to be migrated that is delivered from the legacy system.
Last version migration	Movement of data to a new software platform whereby only the currently valid version of each policy to be migrated is transferred from the legacy system to the new system, where it can be processed.
Historical migration	The movement of data to a new software platform whereby the previously valid versions of each policy to be migrated as well as the currently valid version are transferred from the legacy system to the new system, where they can be processed.
Processable migrated versions	Policy versions that can be fully processed in the new system following their migration, for example, retroactive changes having an effective date prior to the key date.
Versions only migrated for display	Policy versions that can only be displayed following their migration.
Technical migration	Copying the existing data from a legacy system to the application tables of FS-PM. The legacy system data have to be converted to the FS-PM data structures.
Technically migrated version	Version of a policy following its technical migration. This version is the starting point for the functional validation.
Functional validation	Transfer of a technically migrated version of a policy to a checked, calculated, and released policy version.

4.2 Abbreviations

Below you can find an explanation of the most important abbreviations used in this guide.

Abbreviation	Explanation
ETL	Extract, Transform, Load (ETL) The necessary process steps to combine data from several data sources possibly with differing structures in one target database. • Extraction of the relevant data from various sources • Transformation of the data into the layout and format of the target database • Loading the data into the target database
LSMW	Legacy System Migration Workbench Standard SAP migration tool for carrying out the ETL process.
FPP	Framework for Parallel Processing SAP tool that is needed to improve parallel processing of background runs in FS-PM.
BAPI	Business Application Interface SAP standard technology
BAdI	Business Add-In SAP standard enhancement technology
PBT	Policy-Based Technology Development environment focusing on the requirements of in-force business management. It is contained in the scope of delivery for FS-PM.

5 General Information on Legacy Data Migration

Legacy data migration is necessary if, following the implementation of a new software platform (also known as software or application migration), you also want to process the data previously administered in the legacy system in the new software platform. This is frequently a prerequisite for being able to replace the legacy system completely.

The necessary level of resources and risks associated with a legacy data migration are regularly underestimated. Timely and careful planning and execution is decisive for preserving data consistency and a smooth transition of the functions from the old to the new application.

5.1 Process Steps for the Data Migration

Data migration always comprises the following steps:

- The extraction of the data from the legacy system that you also want to be available in the new software platform
- The transformation (or conversion) of the extracted data to the data format of the new software platform
This usually involves more than just data mapping: mandatory data may for instance have to be supplied to the new software platform that is unknown in the legacy system, or checks need to be made in the migration tool in the transformation.
- Loading the transformed data into the target system, meaning writing the data to the database tables of the new software platform.

Taken together, these three steps are usually referred to as the ETL process, and the migration tools that support these process steps as ETL tools. The LSMW is one of these migration tools.

In this guide, the term “technical migration” is used for this ETL process.



Note

When data is migrated in FS-PM, the functional validation of the technically migrated data takes place in a separate transaction in FS-PM. Due to the complexity of the business objects in in-force business management, it is not feasible to have all of the necessary checks and derivations already included in the migration tool.

5.2 Migration Scenarios

In this context, it is usual to distinguish between a Big-Bang migration and a gradual migration.

In a Big-Bang migration, all of the new systems affected by a migration project go live on a certain key date.

Contrast this with a gradual migration that is possible in two variants:

- The existing data from the legacy system is transferred to the new system in tranches. The new system goes live with these subsets at different key dates, and so gradually replaces the legacy system.
- If several legacy systems are to be switched to a new software platform in a migration project, but these go live in succession on different key dates, this is also considered to be a gradual migration.

If, in addition to FS-PM, you want to migrate other operational systems to SAP for Insurance Applications in your migration project, you are advised to migrate the operational system data such as Collections/Disbursements or Commission Management first if at all possible, and the in-force business management data last.

In addition to this, various application-specific migration scenarios are distinguished for a data migration to FS-PM.

For more information on this, see Section ‘Last Version Migration’.

6 Overview of Legacy Data Migration to FS-PM

6.1 Basics

If an insurance company wants to replace existing in-force business management systems with the SAP in-force business management system FS-PM, this usually entails a migration of data from the legacy system to the new system. To do this, the insurance in-force business data is transferred to the new in-force business management system FS-PM.

The scope of the data to be migrated depends on the actual insurance products and the selected migration scenario. In addition, the scope and quality of the legacy data depend on the capacity of the legacy system.

In the standard system, FS-PM supports the following in-force business management data:

- Policies
- Insurable objects
- Fund data
- Master policies
- Distribution plans

For each of these data types, in the standard system FS-PM has a migration interface implemented as a BAPI. For additional information about the migration of this data, see Sections 'Details for the Migration of Policies to FS-PM' and 'Details for the Migration of Other Business Objects to FS-PM'.

For additional information about the integration of data from relevant operational systems in the migration project, see Section '

Integration of Operational System Data in an FS-PM Migration Project'.

What you need to know specifically when migrating add-on lines of business, meaning lines of business other than P&C/Non-Life and Life, is described in Section '

Treatment of Add-On LoBs for the Migration of Policies’.

Data migration from a legacy system to FS-PM always encompasses the following steps:

- Execute technical migration
- Execute functional validation
- Migrate business transactions (optional)
- Complete migration

You need to perform these steps in the order specified:

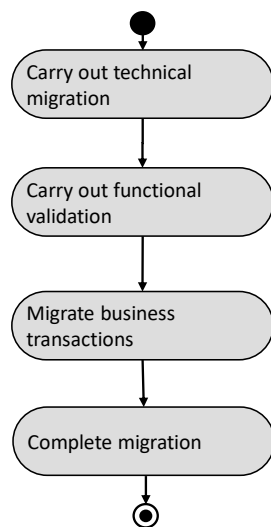


Figure: Basic processes in a legacy data migration to FS-PM

Technical Migration

The technical migration covers the following steps:

- Import in-force business data in the legacy system into a migration tool (for example, LSMW available in SAP Basis)
- Convert imported in-force business data to FS-PM compatible data structures (for example in LSMW)
- Write converted data to the FS-PM database tables using the FS-PM migration interfaces. In this step, technical keys (GUIDS) are also determined and also, if required, numbers assigned from number range objects. You can also provide data with default values from Customizing of product templates in In-Force Business Configurator in this step.
- You can make checks in the technical migration by a BAdI implementation. The standard system does not make any checks here, only derivations and defaults.

As you may have to use different migration tools for the first two steps and the LSMW is described in great detail in SAP Library, this guide mostly concentrates on the last two steps.

After the technical migration, the migrated policies can be displayed in the *Inquiry* business process.

You can schedule the business processes *Execute Reversal* and *Perform Premium Waiver (Life)* for execution for migrated policies in FS-PM in the technical migration.

For additional information about the technical migration, see Section Technical Migration.

Functional Validation

Functional validation is also abbreviated in this document to 'validation'.

The validation of technically migrated policies can cover the following steps:

- Carry out functional derivations and check migrated policies
This step is comparable with the application check in New Business. This ensures the quality of the migrated policy versions.
- Calculate the migrated policy versions using the product engine, and compare the calculated policy versions with the original migrated policy versions
- In the case of historical migration, position between the migrated policy versions.
- Migrate background-enabled business transactions using a BAdI implementation
- Update latest migrated version of a policy
- Assign imported documents to the policy versions that have arisen due to the positioning
- Release migrated policies and their migrated versions

Whether you need to carry out all of the validation steps mentioned depends on the migration scenario that you want to employ in the migration project. Further, you do not need to repeat all of the steps in their entirety when you migrate to a productive system if you have already validated the migrated policies in a test system. This means that a number of validation steps are optional.

Using the various BAdIs available, you can adapt the validation to meet the actual requirements of your project.

For additional information about the functional validation, see Section 'Functional Validation'.

Migrate Business Transactions

If changes made to a policy by a background-enabled FS-PM business transaction can be mapped in the legacy system, you can carry out the corresponding business transaction in the new system following the migration of the policy. This means the last version prior to carrying out the business transaction in the legacy system is also available in the new system, and you can reactivate it. This procedure is designated in this document as the migration of business transactions.

The migration of business transactions *Execute Reversal (Policy and Contract Reversal)* and *Perform Premium Waiver (Life)* is already supported in the FS-PM migration interface. The system schedules both business transactions during the technical migration. For additional information about this, see Sections 'Migration of Business Transactions: Reversal' and 'Migration of Business Transactions: Perform Premium Waiver (Life)'.

You can migrate other background-enabled business transactions by implementing a BAdI in the validation. For more information, see Section 'Migration of Business Transactions'.

Validation of a policy is only completed when the migration of the business transactions is also completed, providing such a migration is supported. The migration of business transactions is also a component of the validation of a policy, providing these business transactions are executed in an update of the last version.

Complete Migration

Migrated policies can only be processed by the business transactions in FS-PM if the migration of these policies has been completed in its entirety.

The following prerequisites apply to this:

- The migrated policy has been released (you do this in the validation)
- The migrated policy has the corresponding migration status (see Section 'Migration Status')

- The temporary data created during the migration is deleted (see Section ‘Deletion of Temporary Data of the Migration of Policies’; this is particularly important for an historical migration)

The following is a simplified illustration of the components involved in migrating data to FS-PM when using SAP standard migration tool LSMW:

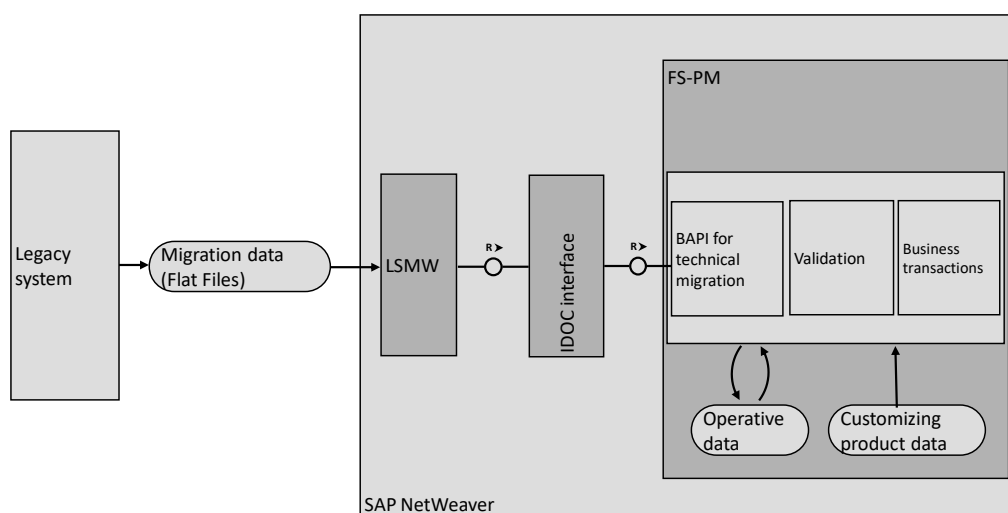


Figure: Components involved in a migration

When migration interfaces are mentioned in this document, this refers to the migration BAPIs for the corresponding FS-PM components (policy, insurable object, fund, master policy, distribution plans). Functions that involve more than importing the data, such as the functional validation, are distinguished from this by being referred to as migration functions.

By using migration BAPIs, partners and customers can migrate additional data fields for tables supplied, migrate data to additional partner or customer tables, and other data for information purposes.

6.2 Last Version Migration

With a last version migration, a single policy version is migrated so that it can be processed with the available business processes in FS-PM. This is the policy version valid on the date of the migration. In the following overview, the policy version is designated as 'Version 4'.

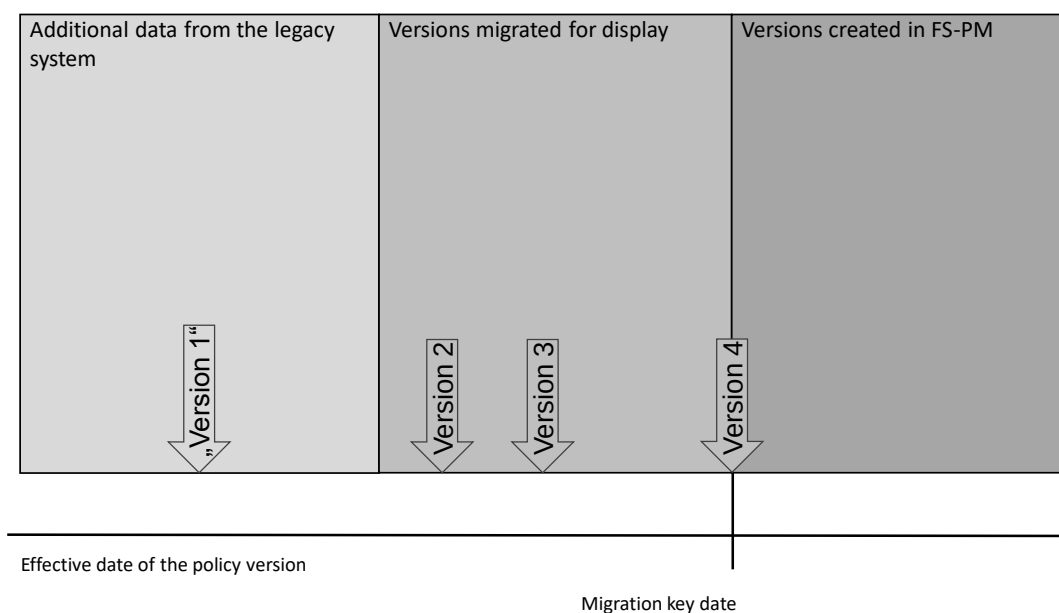


Figure: Last version migration

In addition, policy versions valid for earlier periods can be migrated in such a way that you can only display them in FS-PM. Reactivating the policy at an effective date in the periods in between is not possible in FS-PM. These policy versions are described in this document as 'policy versions migrated for display'.

In addition to this, you can store additional data from the legacy system in customer-specific tables for information purposes, and display it in the *Inquiry* business process in the contract journal. This procedure is suitable, for instance, for data processed in the legacy system a long time in the past that should not to be migrated as complete policy versions.

6.3 Historical Migration

For a historical migration, not only is the policy version valid on the migration date migrated, but also historical policy versions that were valid during earlier periods, such that they can also be processed in FS-PM. These policy versions have 'Version 4' and 'Version 5' in the figure below. They are described in this document as processable migrated policy versions.

In this document, if the historical migration is mentioned, then the migration of historical policy versions processable in FS-PM is meant unless expressly stated otherwise.

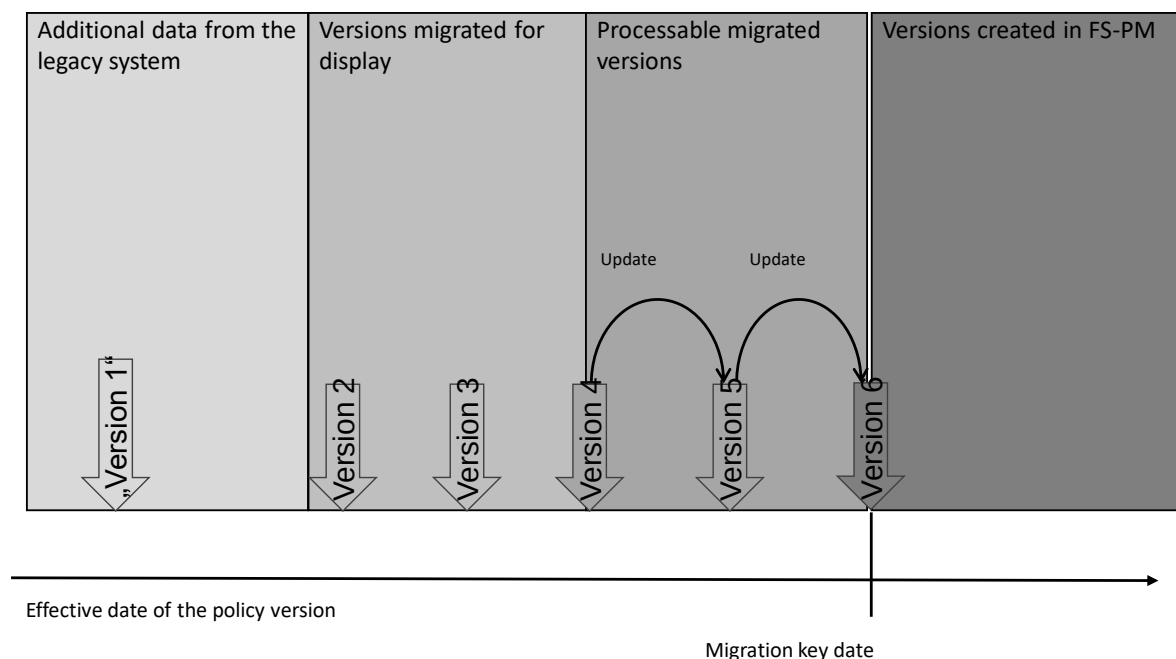


Figure: Historical migration

The migration functions delivered ensure that an update is made between the processable migrated policy versions in FS-PM. With migrated policy versions that are for display only, this is not the case which is why the migration requires significantly less runtime.

Recommendation

SAP recommends you to check carefully which policy versions really need this from a technical point of view, and to migrate only these policy versions.

If the migration project requires you to be able to process the migrated policies in FS-PM retrospectively from a certain validity date, you must migrate a processable policy version for the policies concerned for this validity date. In the figure above, 'Version 4' can be a processable policy version created especially for this purpose.

The processable and for display only migrated policy versions must be supplied to FS-PM with versioning corresponding to journal management in FS-PM. For additional information about versioning the policy versions, see Section 'Technical Migration'.

Complete policy versions are migrated for a historical migration carried out this way. The actual scheduled or unscheduled (irregular) processing from which these policy versions result in the legacy system is not relevant to this. In FS-PM, the processable migrated policy versions are treated as resulting from unscheduled (irregular) processing irrespective of their actual origin in the legacy system.

Without customer-specific adaptations, FS-PM does not have any information about the processing from which the migrated policy versions result. With a retrospective change prior to the validity of a migrated policy version, the migrated policy version cannot therefore be reimplemented.

6.4 Migration of Business Transactions

Using the delivered migration interfaces, you can migrate business transactions and this applies both in the context of a last version migration and in a historical migration context.

The following is a typical scenario for the migration of business transactions:

You want to migrate a policy reversed in the legacy system, for example in the context of a last version migration in FS-PM. If you only migrate this current reversed policy version, you cannot undertake a reinstatement after reversal in FS-PM, as the policy version prior to carrying out the reversal is not available in FS-PM.

In this instance, the procedure to follow is to migrate the last policy version as it was before carrying out the reversal, and then carry out the reversal in FS-PM. The reversal in the migration can be automated. In FS-PM, you can reinstate the reversed policy as usual in the *Reset* business process.

The sequence of the migration of business transactions in a last version migration context is illustrated in the following graphic:

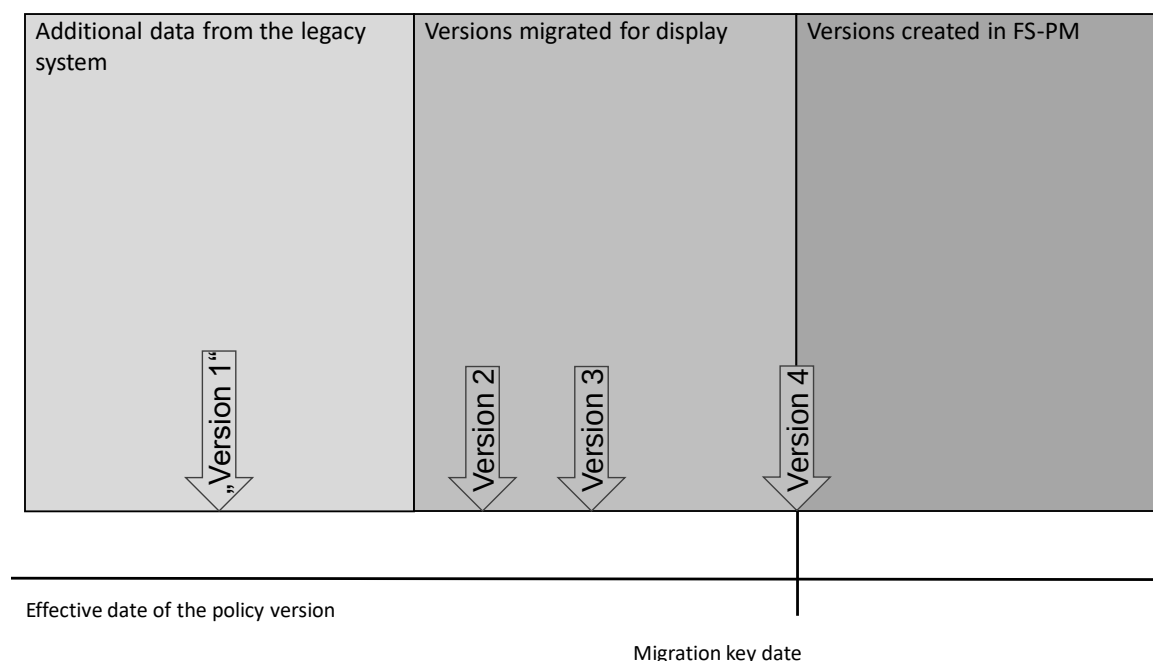


Figure: Last version migration with migration of a business transaction

'Version 4' in the above graphic represents the migrated policy status before carrying out the business transaction (for example, execution of a reversal), 'Version 5' represents the policy version created by the migration of the business transaction. Based on this depiction, it is possible to ascertain that this scenario actually involves a historical migration, since as a result two processable policy versions are available in the historical period in FS-PM. However, 'Version 5' in this case is created in FS-PM.

Consequently, the depiction of the business transaction migration in the context of a historical migration only differs slightly from this:

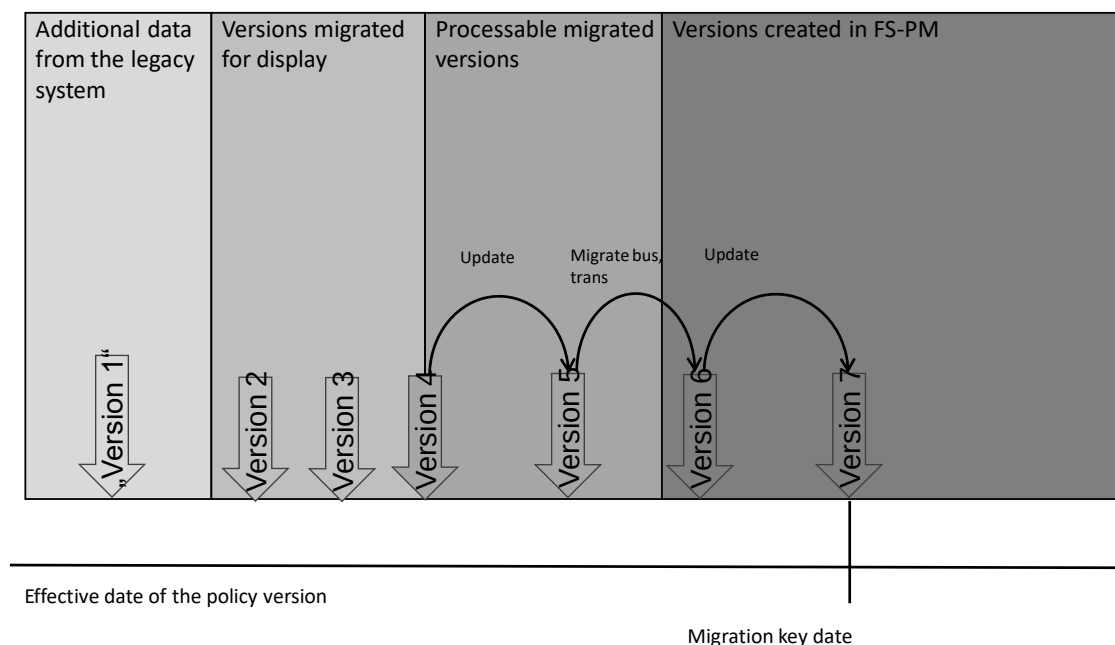


Figure: Historical migration with migration of a business transaction

'Version 4' in the graphic above represents the required version at the beginning of the historical period that should be processable in FS-PM. 'Version 5' represents the historical policy version before carrying out the business transaction to be migrated, and 'Version 6' represents the policy version created by the migration of the business transaction.

The migration of business transactions consequently provides you with the following functions:

- Automatic execution of business transactions in the validation
- Automated scheduling of business transactions with an effective date later than the migration date to be carried out in FS-PM following the migration.
- If a change between two migrated policy versions by a business transaction of the *Change* business process can be depicted, and you migrate the later version by a migration of this business transaction, you can reimplement the later version in FS-PM.

When you schedule the migration of business transactions, note the following:

- The migration of business transactions can generally only be performed based on a processable migrated policy version. The effective date of the business transaction to be migrated must be later than the validity date of the last processable migrated policy version. In the figure above, this is 'Version 5'. For more information, see Section 'Migration of Business Transactions'.
- This restriction does not apply for migrating Execute Reversal and Perform Premium Waiver (Life). For more information, see Sections 'Migration of Business Transactions: Reversal' 7.3.5 and 'Migration of Business Transactions: Perform Premium Waiver (Life)'.

6.5 FS-PM Migration Interfaces

In this section, you can find a summary of the FS-PM migration interfaces delivered for the technical migration. The function and capabilities of these interfaces are set out in more detail in this document.

Mass transfer describes the writing using mass processing of the migration files supplied by the migration tool, (for example LSMW) to the FS-PM database of the relevant component.

Mass deletion describes the mass processing deletion of objects for the relevant component.

Temporary data is also removed that was created in the technical migration. An example of using mass deletion is where errors have occurred during the migration of a large number of policies, and you want to migrate the entire package again. In this case, you begin by deleting the incorrectly migrated policies in FS-PM.

FS-PM provides interfaces for mass transfer and deletion for the following components and business transactions:

- Policy
- Insurable object
- Master contract
- Distribution plan
- Funds (life)
- Fund range (life)
- Fund price (life)
- Fund order (life)

The migration of Execute Reversal and Perform Premium Waiver (Life) is a component of the interface for the mass transfer of policies.

FS-PM provides the migration interfaces specified in the form of BAPIs with the related ALE interfaces (IDocs).

The migration interfaces do not cover all entities for the FS-PM business objects. For more information, see Section 8.6.

6.6 Not in the Scope of Functions

Note the following constraints for the delivered migration interfaces and functions for validation:

- You cannot migrate versions that are currently being processed.
- The migration of application versions is not possible. (This restriction applies to the migration interfaces delivered that are described in this guide. By making customer implementations, you can also implement this requirement.)
- The migration of postdated applications is not possible. (This restriction applies to the migration interfaces delivered. By customer implementations, you can also implement this requirement.)
- If the legacy system supports two-dimensional period management, you can only migrate the active paths to FS-PM.
- A valid system-wide central migration status is not supported; FS-PM only recognizes a policy-related migration status.
- The migration of business transactions, with the exception of the migration of Execute Reversal and Perform Premium Waiver (Life), is only possible in accordance with the latest processable migrated policy status.

7 Details for the Migration of Policies to FS-PM

In this section, the migration of policies in FS-PM is described in detail. The focus is on the status and authorization concept, the technical migration, the functional validation and the migration of business transactions, and the enhancement options available in the standard system for these areas.

If you need to take into account differences when implementing the various migration scenarios (for example, last version migration versus historical migration), this is pointed out at the relevant place.

If in the following descriptions the migration tool is explicitly included, then you can always assume this is the LSMW that is available in SAP Basis.

7.1 Overall Process for FS-PM Migration

So that the migration tool can use the FS-PM migration interfaces, the RFC modules that write the data to the FS-PM databases are encapsulated in BAPIs with related ALE interface (IDoc).

Note

The migration BAPIs are the interfaces that are available in the standard system for FS-PM. If you consider using other interfaces for loading the data to FS-PM or migration tools other than LSMW, you are advised to get in contact with SAP.

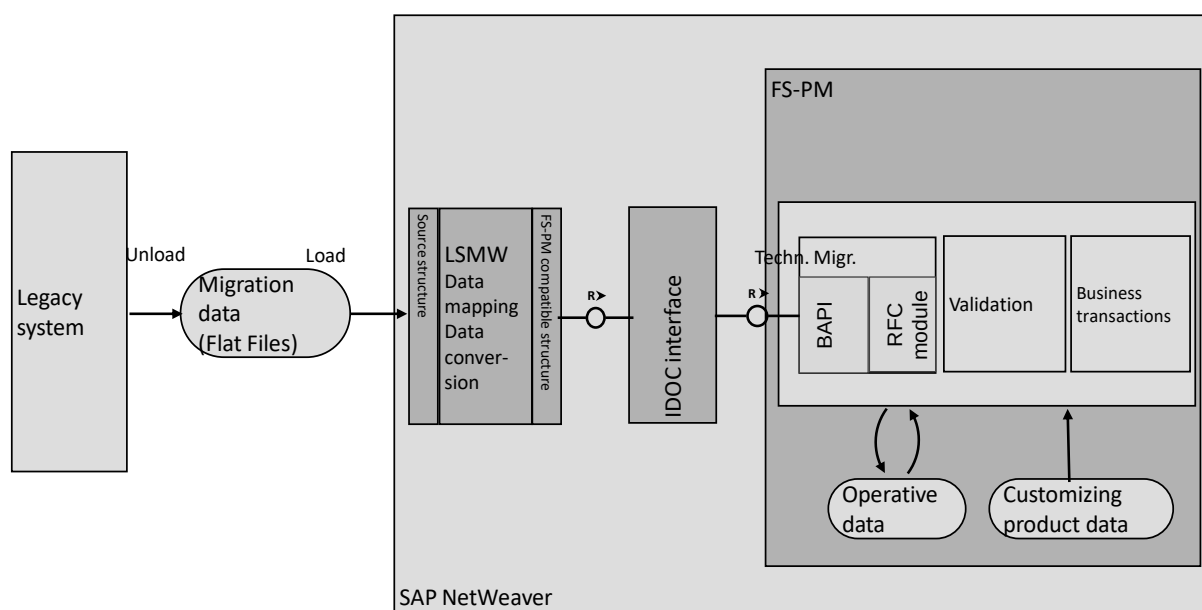


Figure: Components and interfaces for migration to FS-PM

The standard migration process for migrating insurance policies (and for the most part similar for migrating insurable objects, master policies, fund data, and distribution plans) to FS-PM looks like this:

- Extract the data from the legacy system and download it to the migration tool
- Conversion of the imported data to FS-PM data structures in the migration tool (can mostly only be done in iterative procedure)

- Writing the data to the FS-PM database tables (these first three steps are usually abbreviated to ETL – extract, transform, load)
- Functional validation (is carried out in a separate mass processing transaction due to the complex factual and technical structure of the FS-PM policy business object)
- Migration of business transactions in the update during the validation (optional)
- Completion of the migration by release of the validated policies and deletion of temporary data produced during the migration

Recommendation

The migration process described above contains numerous cycles. It often becomes clear during the functional validation that adjustments still need to be made for the legacy data extraction. For this reason, SAP strongly advises you to have a dedicated test system available for the migration as early as possible.

The following transactions are available on the *SAP Easy Access* screen for carrying out a migration in FS-PM:

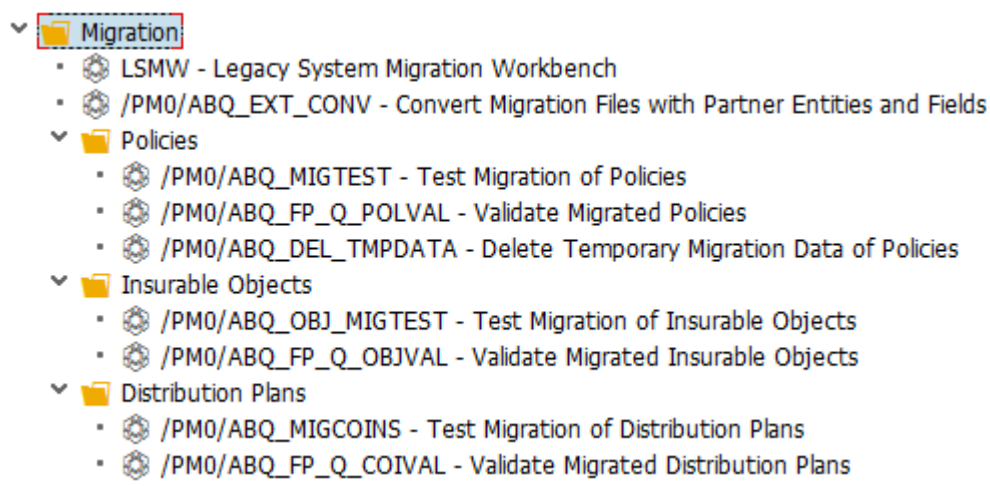


Figure: SAP Easy Access for Migration

In addition, there are service programs in the internal area menu `/PM0/CUST_INT`.

7.2 Authorization and Status Concept

7.2.1 Authorization

The composite template role for developers (`SAP_FSPM_C_TMPL_DEVELOPER`) also covers the single role `SAP_FSPM_S_TMPL_MIGRATION` delivered with FS-PM, and has the necessary authorizations to execute all of the transactions and programs needed in the migration.

If, therefore, you assign a copy of the template role `SAP_FSPM_C_TMPL_DEVELOPER` to those involved in the migration with unchanged authorization profile, they can perform all of the necessary migration tasks.

7.2.2 Migration Status

To facilitate a gradual migration (meaning a migration of legacy datasets to a productive FS-PM system), no system-wide migration status is used in FS-PM but rather only a policy-related migration status.

The following status diagram shows when and under what conditions this migration status accepts which values.

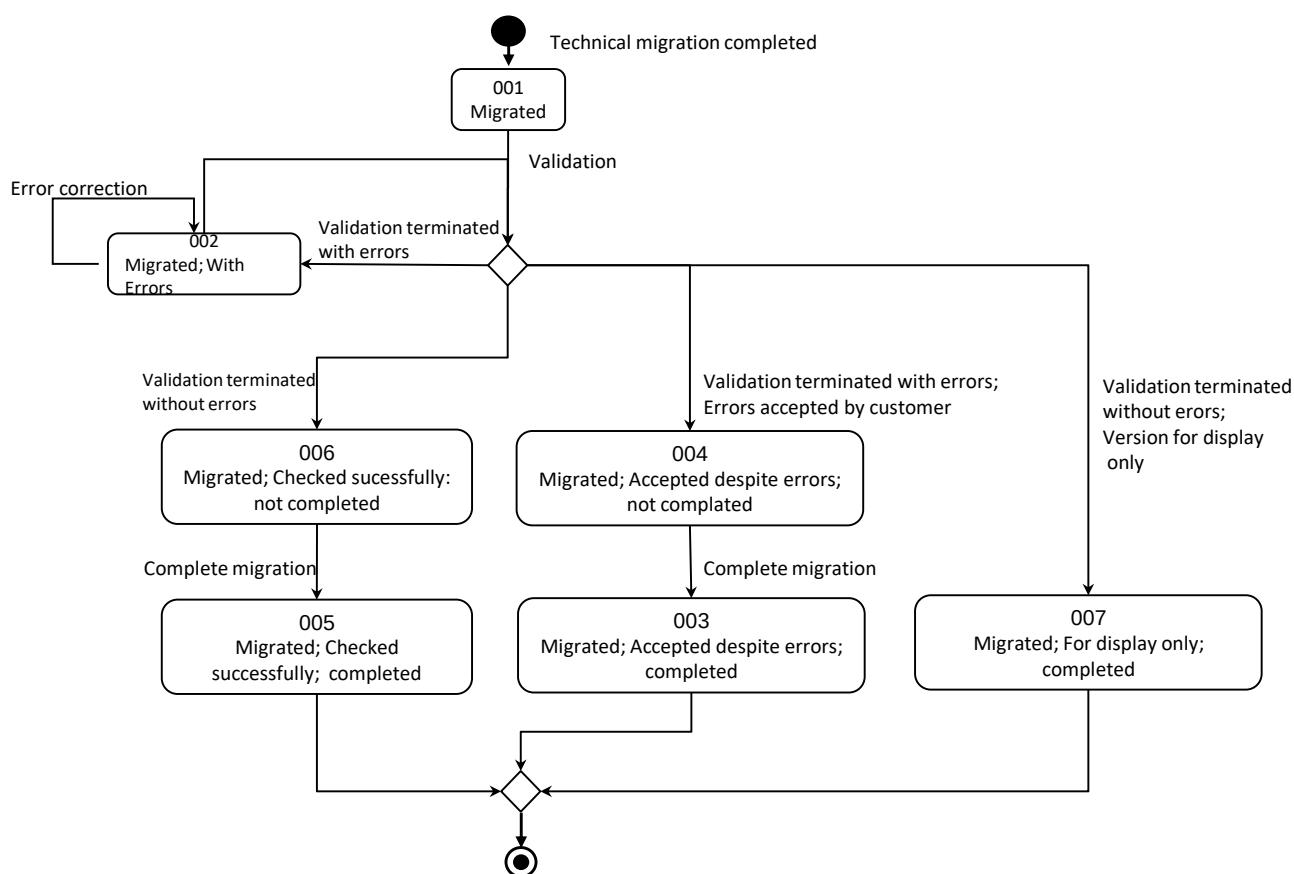


Figure: Status diagram of the migration status

Explanation of the possible status values:

Status ID	Status Text	Description
001	Migrated; Not Checked	This status is set if the data has been transferred to FS-PM in the technical migration step, but not yet validated.
002	Migrated; Error	This status is set when the validation is started and then terminated with errors. Migrated policies with this migration status are selected during the error handling run.
003	Migrated; Accepted Despite Errors; Completed	This status value is set when the validation has not run successfully, but the errors have been accepted with BAdI /PM0/ABQ_POL_ERRCORR_BADI of the error handling run, and the migration has been completed.
004	Migrated; Accepted Despite Errors; Not Completed	This status value is set when the validation has not run successfully, but the errors have been accepted with BAdI /PM0/ABQ_POL_ERRCORR_BADI of the error handling run.

005	Migrated; Checked Successfully; Completed	This status value is set when the validation has been successfully run and the migration completed.
006	Migrated; Checked Successfully; Not Completed	This status value is set when the validation has been successfully run but the migration not completed.
007	Migrated, Only for Display; Completed	This status is set when the policy version has been migrated for display only, and cannot be processed.

The migration status is persisted for each journal entry in treaty journal table /PM0/ABDJPOL in the BOMIGSTATUS_CD field.

Other migration-relevant execution parameters and process steps are persisted in migration journal table /PM0/ABDQMIGJOUR providing the migration and validation of a policy have not yet been completed.

To correct incorrect policies (policies with migration status 002), functions are available in FS-PM that are described in the Appendix.

Recommendation

SAP always recommends that you delete the incorrect policies in FS-PM rather than correct them. You can then eliminate the cause of the error (for example, incompletely delivered data, incorrect checks in the validation), and carry out the migration and validation of the policies concerned again.

By implementing BAdI /PM0/ABQ_POL_ERR_HANDL_BADI (see Section 'BAdI: Functional Validation of Policies – Error Handling (/PM0/ABQ_POL_ERR_HANDL_BADI)'), you can mark the errors reported by the delivered validation functions as not relevant. In this case, migration status 003 or 004 is set for the corresponding policy, depending on whether the release of the state foreseen this. **Note:** This occurs in method /PM0/CL_ABQ_BPU_PP_TOOLKIT->HANDLE_ERROR.

Completing a migration also includes releasing the validated policies. This means both the release of all of the policy versions for this policy, and the successful update in the validation. With the successful release, migration status 005 or 003 is set for the policies. You release the validated policies by setting the *Release Policies* indicator in transaction *Validate Migrated Policies*.

The deletion of temporary data created during the migration is also part of completing a migration. You can do this with the transaction described in Section 'Deletion of Temporary Data of the Migration of Policies'. When reference is made in this document to a completed migration, this assumes that you have already deleted the temporary data.

A non-completed policy version cannot be loaded to FS-PM for processing, neither for changing nor updating. If an attempt is made to load a request for processing, the system recognizes this, and the customer project cannot influence this.

If you require additional checks when loading migrated policy versions, you can implement the following BAdIs (for the sake of completeness, the corresponding BAdIs for loading the remaining FS-PM business objects are also detailed here):

Table: BAdIs for checking migrated policy versions

Business Object	BAdI
Policy	/PM0/ABQ_POL_STR_CHK_BADI
Insurable Object	/PM0/ABQ_OBJM_STR_CHK_BADI
Master Contract	/PM0/ABQ_MPO_STR_CHK_BADI
Distribution Plan	/PM0/ABQ_COINS_STR_CHK_BADI
Fund	/PM0/ALQ_FND_STR_CHK_BADI
Fund Prices	/PM0/ALQ_FPRC_STR_CHK_BADI
Fund Range	/PM0/ALQ_FRM_STR_CHK_BADI
Fund Order	/PM0/ALQ_ODR_STR_CHK_BADI

These BAdIs are delivered without a standard implementation, though sometimes with a fallback implementation.

The BAdI implementations return an indicator that specifies whether loading should be terminated, and also a table with error and information messages. Here you can, for example, implement customer authorization checks or display additional messages.

7.2.3 Business Processes for Migrated Policies

For technically migrated policies, you can execute the *Inquiry* business process. The following restrictions apply:

- All pushbuttons in the application toolbar are inactive, with the exception of the Collections/Disbursements locks. However, Collections/Disbursements locks may not be added, changed or deleted, but only displayed.
- All changing business transactions are inactive in the context menu. You can only display the policy.

On the other hand, the following business processes are canceled if they are executed for migrated policies for which the migration has not yet been completed:

- Change
- Universal Change
- Reversal
- Reset
- Update
- Scheduling Related Dates

As already explained, the check of the migration status ensures that a policy can only be set to *In Processing* if it is released. At the same time, a check is made whether temporary data still exists for the policy. As long as temporary data exists, data for the technically migrated policy versions (temporary versions) still exists in the database in addition to data for validated and released policies (final versions), so that database queries in FS-PM could return a temporary version.

If, therefore, you want to process a migrated policy whose migration is not yet completed in one of the business processes specified, you receive error messages in the FS-PM standard system, providing one of the following statements applies:

- The policy and all its migrated versions have not yet been released.
- The policy has been released, but the temporary data has not yet been deleted.

If you need additional checks at this point, use BAdI `/PM0/ABQ_POL_STR_CHK_BADI`.

7.3 Technical Migration

7.3.1 Prerequisites

Before you migrate policies in FS-PM, a range of prerequisites must be fulfilled. In this section on technical migration, the following prerequisites apply:

- Customer developments in FS-PM are complete and have been successfully tested.
- All Customizing activities in FS-PM are complete.
- The values of the FS-PM key fields that reference data in operational systems (for instance, the identifying data in their Collections/Disbursements system or business partner module) are known and are supplied to the migration. The operational systems are set up appropriately before the migration to FS-PM.

Recommendation

For these fields, FS-PM provides the option of internal number assignment for initial values, however SAP recommends only using these in exceptional cases. For a Big-Bang migration, the migration of operational systems should be complete before the migration of the policies. This lets you switch off the interfaces to operational systems

during the migration of policies. For more information about the integration of data from operational systems in the migration process, see Section ‘

Integration of Operational System Data in an FS-PM Migration Project'.

- The data for FS-PM components (fund data, insurable objects, master policies, distribution plans) are migrated and released. This procedure is more or less the same as that described for the migration of policies. For additional information about the migration of the FS-PM components, see Section 'Details for the Migration of Other Business Objects to FS-PM'.
- For policies that are to be (technically) migrated using the LSMW, an object is created in the LSMW project as follows:

Object Type and Import Method

☐ Standard Batch/Direct Input

Object

Method

Program Name 

Program Type

☐ Batch Input Recording

Recording  

☒ Business Object Method (BAPI)

Business Object	/PM0/ABQMP	FS-PM Migration of In-Force Business (Restricted Rel
Method	MASSMIGRATION	FS-PM: Migration of In-Force B...
Message Type	POLICYMIGRATION_MASSMIGR...	FS-PM: In-Force Business Migra...
Basic Type	/PM0/POLICYMIGRATION_MAS...	FS-PM: Migration of In-Force B...

☐ IDoc (Intermediate Document)

Message Type

Basic Type

Enhancement

☐ Allow Structure Assignment for EDIDC40

Figure: LSMW project for migration

- The legacy data is converted by the migration tool to FS-PM-compatible structures and is available to the migration BAPI as input data.

7.3.2 Scope of Technical Migration

The migration interface for mass migration of policies is the BAPI method *MassMigration* of the *PolicyMigration* business object:

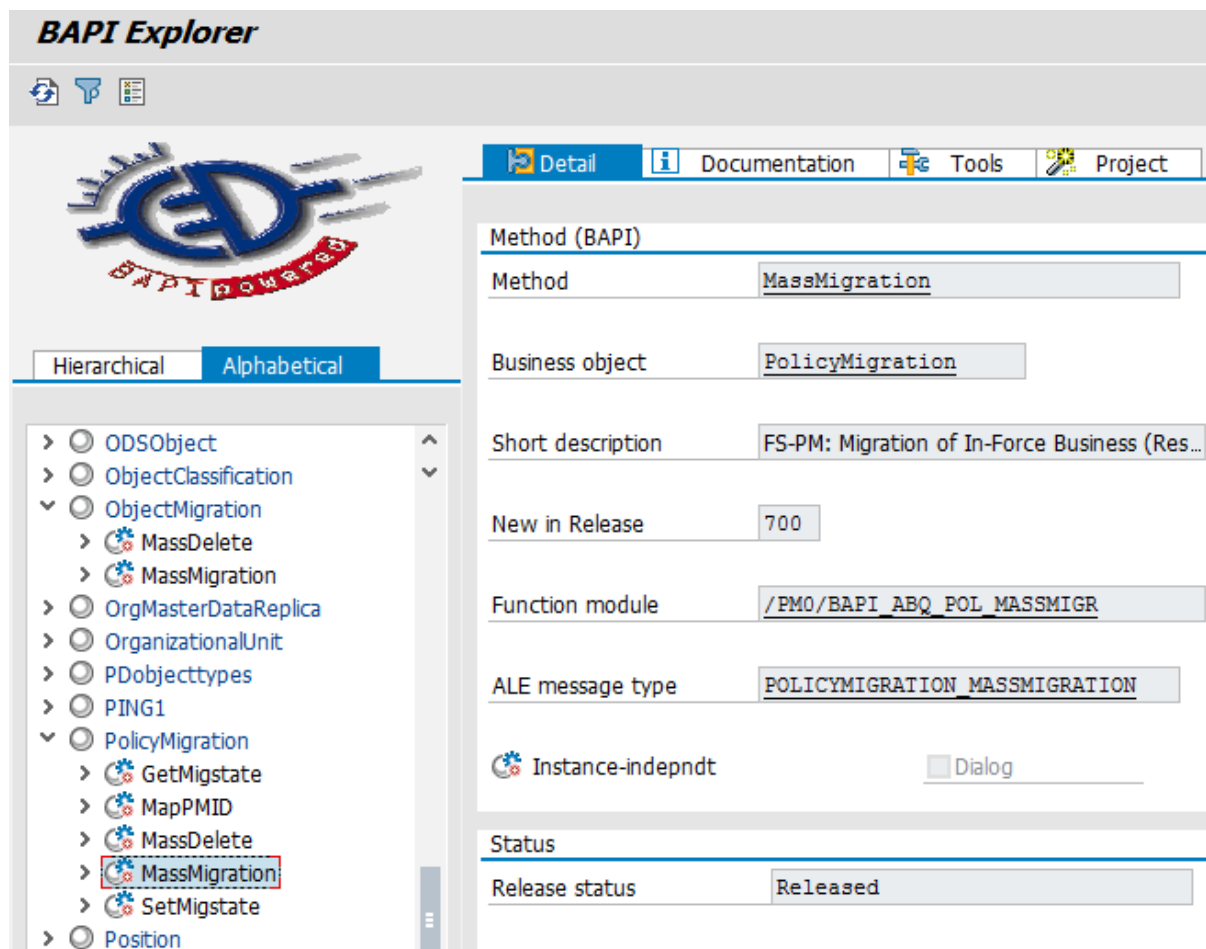


Figure: BAPI Explorer with BAPI method MassMigration

The following figure illustrates the process flow for the technical migration of policies:

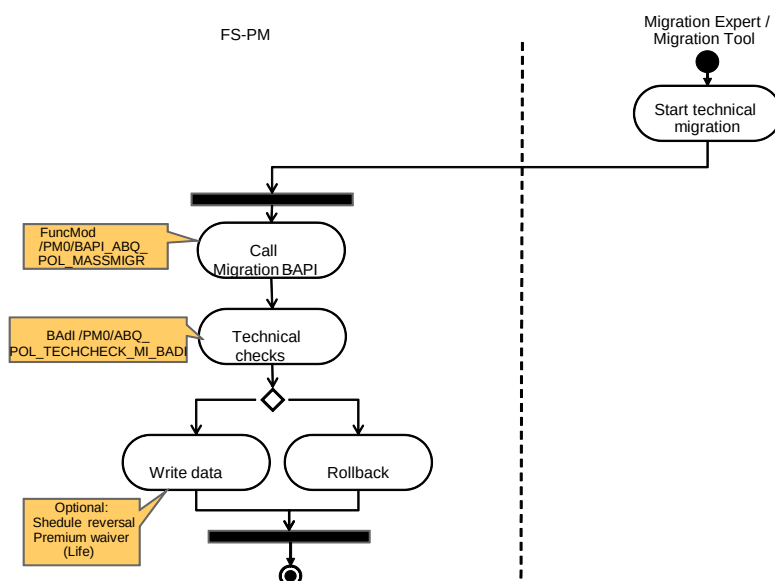


Figure: Overview of process flow of technical migration

All the repository objects specific to the migration belong to the following packages:

- /PM0/AB_MIGRATION (cross-LoB basis functions)
- /PM0/AP_MIGRATION (P&C/non-life-specific functions)
- /PM0/AL_MIGRATION (Life-specific functions)

The technical migration of policies is handled by the remote-enabled function module /PM0/BAPI_ABQ_POL_MASSMIGR.

With the BAdI /PM0/POL_TECHCHECK_MI_BADI, you can implement additional technical checks and data changes.

During the technical migration, the *Execute Reversal* and *Perform Premium Waiver (Life)* business transactions to be migrated are scheduled provided you have supplied the relevant data.

You can import additional customer-specific data into FS-PM with the migration BAPI. This data can either be persisted in FS-PM purely for information purposes or it can be stored in the FS-PM policy versions. This information, which is also stored in the technically migrated version (on which no operations have been performed in FS-PM), may also be used as comparison values for insurance mathematics to fill any delta values and to be stored in the contract. Thus, with the migration project you can, for example, ensure that the values already communicated to the policyholder also exist in FS-PM. For more information, see Section 'Functional Validation'.

The migration BAPI contains the import parameter `BapiMigrationControl` of structure type /PM0/BAPI_ABQ_MIGRATION. With the transferred values, you can control the behavior of the migration interface in the specified framework. In particular, you can trigger the validation and release of the migrated versions directly after the technical migration. For the migration of policies, the control parameters have the following functions (not all of the control parameters are supported for the migration of data of the other FS-PM components):

/PM0/BAPI_ABQ_MIGRATION-IF_LINK_VALIDATION

Depending on the value of this parameter (*X* = yes, in background; *D* = yes, in dialog mode; space = no), the functional validation is executed directly after the technical migration. In this case, the program for the functional validation is started at the end of the function module of the migration BAPI (function module /PM0/BAPI_ABQ_POL_MASSMIGR). For additional information about the functional validation, see Section 'Functional Validation'.

You can also define this parameter in the migration test tool. For additional information about the test tool, see Section 'Testing the Migration of Policies'.



Tip

You should only use this parameter for testing small policy portfolios. To migrate larger policy portfolios, you should use the transaction for validating migrated policies (/PM0/ABQ_FP_Q_POLVAL) since this fully supports parallel processing.

/PM0/BAPI_ABQ_MIGRATION-ID_MIGR_DATE

This parameter specifies the migration date in the form YYYYMMDD. The entries written to the FS-PM application database tables during migration use this date to set the time stamp. This is the date on which the technical migration was executed.

/PM0/BAPI_ABQ_MIGRATION-IF_RELEASE

Depending on the value of this parameter (*X* = yes, space = no), the validated policies can be released after the validation has run successfully in the migration BAPI.

You can also determine this parameter in the migration test tool if the functional validation has been previously selected there.



Tip

You should only use this parameter for test purposes. For migration in a live system, the release should be handled in transaction /PM0/ABQ_FP_Q_POLVAL.

/PM0/BAPI_ABQ_MIGRATION-IF_PREASSIGN

Depending on the value of this parameter (X = yes, space = no), specific preassignments are made in BAPI /PM0/BAPI_ABQ_POL_MASSMIGR.

You can also define this parameter in the migration test tool.

If this parameter is set, this is stored in the migration journal in the `PM_DEFAULT` attribute. For the technical migration, this parameter controls whether a preassignment should be made with standard values from In-Force Business Configurator. However, this is only possible if the template ID is known.

The following figure illustrates the main steps of the process flow for the technical migration of a policy:

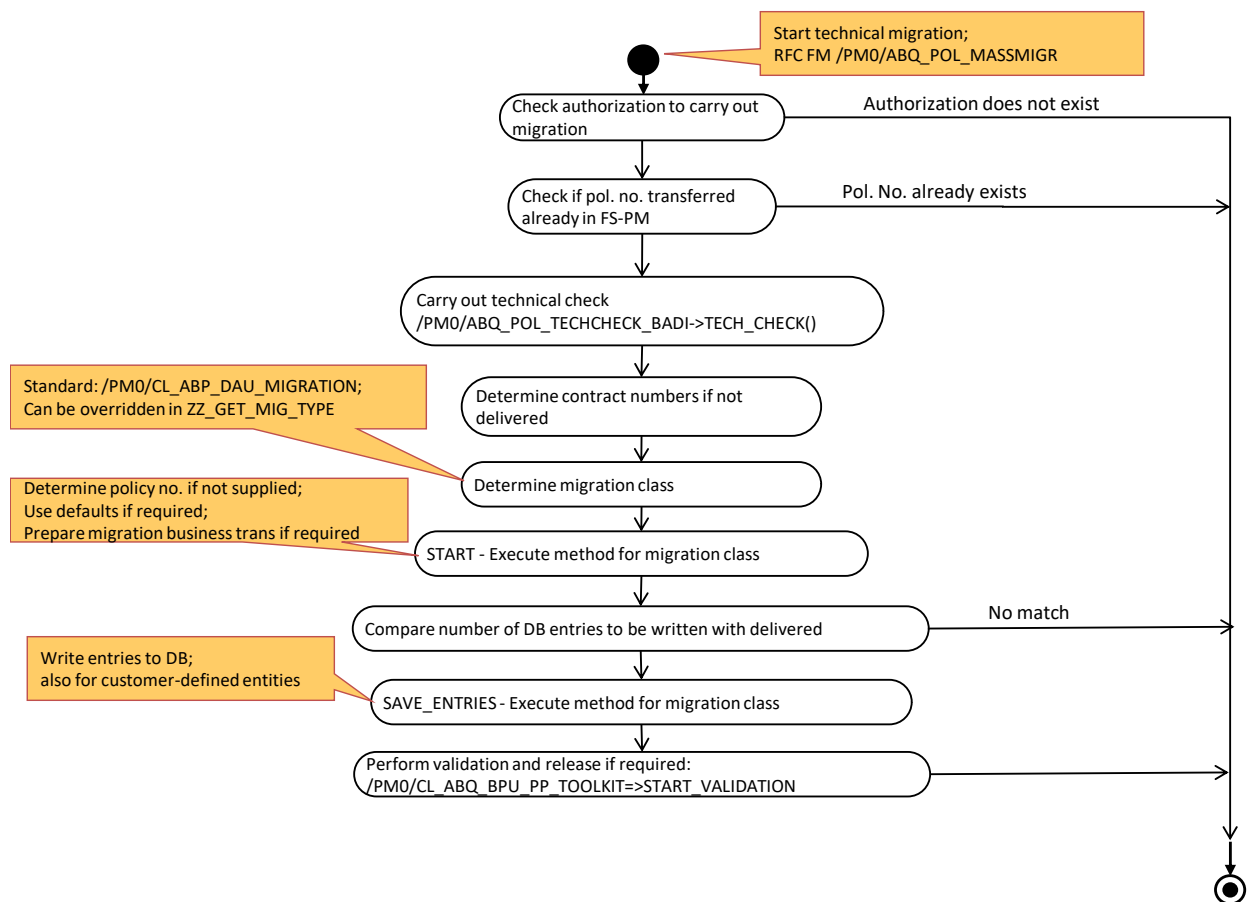


Figure: Detail of process flow of technical migration

Data transfer is basically handled by the migration class. The migration class holds all of the migration data in instance tables and writes the data to the FS-PM database tables after technical formatting.

The standard FS-PM system uses migration class /PM0/CL_ABQ_DAU_MIGRATION. You can also use your own migration class. Subprogram `ZZ_GET_MIG_TYPE` must be created in function group /PM0/ABQ_POLMGM and it must return the name of the migration class to be used.



CAUTION

Only highly experienced development partners should implement their own migration class.

For a better understanding of the process flow of the technical migration and the criteria described below that the data transferred to the migration BAPI has to meet, a thorough knowledge of how the standard migration class works is required. For additional information about the standard migration class, see ‘Figure: Process flow for technical migration of entities not delivered in the standard system.’

Obligatory contract elements are not automatically created by the system, neither for the technical migration nor the functional validation. They must be delivered in the migration file according to the cardinalities defined in In-Force Business Configurator.

7.3.3 Migration Data

7.3.3.1 General Requirements for the Migration File

All the migrated versions of a policy must be transferred to the migration BAPI in one call. Consequently, the migration file must include all of the policy versions to be migrated.

The standard migration class is capable of processing multiple policies where each policy also has multiple versions. In this case, all of the policy versions to be migrated must be available in a single BAPI call.



Tip

You can use the migration test tool to create migration files from policies that exist in FS-PM. You can use these migration files as a template for your migration project. For additional information about the migration test tool, see Section ‘Testing the Migration of Policies’.

For migrated policies in FS-PM, the following options are available for assigning the functional policy number (field `/PM0/ABQAPOLICY-POLICYNR_TT`):

- If no policy number is transferred to the migration data in the BAPI call, a new policy number is assigned on mass data transfer on the basis of the internal number range interval (the same as for New Business).
- If a policy number is transferred on the call, an external number range interval must be maintained for it in the number range object. During mass data transfer, the system checks the interval. The system also checks if the policy number already exists on the FS-PM database.

These steps are performed by the standard migration class.

7.3.3.2 Display of Policy Business Object and Its Versions in Migration File

Every entity in the migration file is identified by a `BO_ID`, which identifies an element of the policy business object. This `BO_ID` serves the purpose of a temporary key during the technical migration, and the assignment of this ID is the responsibility of the migration project. Moreover, line of business enhancements (siblings) must be delivered with the same `BO_ID` as the associated basis entity. The correct relationships between the entities in the policy business object are mapped in such a way that each entity (except for the policy data) references the `BO_ID` of the parent entity in the `PARENT_ID` field. This is the way the `BO_ID` field is used in the migration structures; it is essential to specify the entity relationships correctly for a successful migration in FS-PM. You can display the possible relationships in the policy business object in the PBT Workplace (transaction `/PM0/3FW_START`, application FS-PM, business object model `FS_PM_POLICY_MANAGEMENT`).

Multiple policy versions to be migrated are mapped with the `VERSION_ID` field. This field exists in all migration structures. The version numbers must be in ascending order of the historical order of the versions to be migrated. An entity that exists in more than one version must exist in all versions with the same value for the `BO_ID` in the migration file.

The data transferred in an IDoc must be internally consistent with regard to the BO_ID, entity relationships, and version numbers.



When assigning the BO_ID, you must therefore consider the following:

- The identification numbers must be unique within a transferred migration file. The same BO_ID can only be used for multiple versions of an entity.
- To ensure the unique assignment of the data records, the line of business enhancements must have the same identification numbers as the main nodes.
- The BO_ID of the parent entity in the policy business object must be correctly specified in the PARENT_ID field of the entities so that the relationships in the business object can be created correctly in FS-PM.

You can only use a limited number of different values for the BO_ID in a migration file. This limitation is specified by the data type (NUMC5) that the BO_ID field is based on. As using too many BO_IDs would result in incorrectly migrated policies, an error message informs you an overflow would occur for the BO_IDs during the technical migration. In this instance, you have to reduce the migration file as appropriate, and transfer fewer policies in an IDoc.

The figure below is an example of how the migration data that is transferred to the migration BAPI is structured.

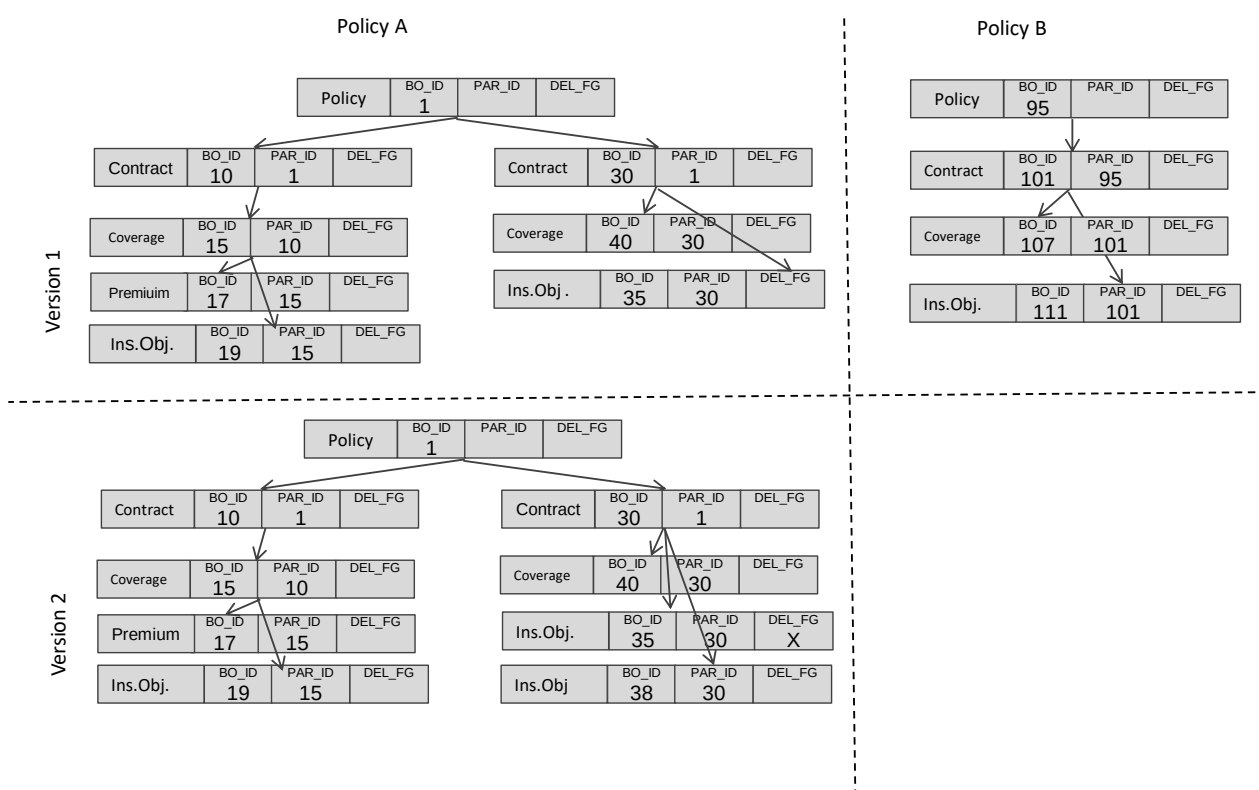


Figure: Structure of the migration data (example)

In the migration file, data for two policies is transferred, which results in two processable versions being delivered for policy A. The second version is different from the first to the extent that an insured

object (BO_ID = 35) was replaced by another one (BO_ID = 38) and data in the insured object was changed with BO_ID = 19. In the migration file, this can be displayed as in the following table (the listed line numbers merely serve the purpose of simplifying the following explanation):

Table: Example entries for insured objects

Line no.	BO_ID	Parent_ID	Version_ID	To delete_FG	Dates
1	19	15	1		Value19Old
2	19	15	2		Value19New
3	35	30	1		Value35
4	35	30	2	X	Value35
5	38	30	2		Value38
6	111	101	1		Value111

The table contains all the entries for insured objects that the migration file must provide.

- Only one version is supplied for policy B; all of the entities belonging to policy B therefore receive the Version_ID 1 (see Line 6) and only exist in this version.
- Two versions are supplied for policy A, consequently there are entries for the entities with Version_ID 1 and 2.
- Unchanged entities must also be supplied for each version since a complete policy version is expected for each version. The entry in line 2 would also have to be supplied if the insured object with BO_ID = 19 is supposed to have unchanged values in the second version (Value19Old in the example).
- Entities that no longer exist in a version in comparison to the previous version must likewise be supplied with the corresponding Version_ID and must be flagged for deletion with the ToDelete indicator (see line 4). In this connection, see the information provided in Section 7.4.11.
- All versions of a policy must always be transferred with the same call since technical keys (GUIDs, and so on) can only be assigned once.

7.3.3.3 Preassignments and Initial Fields

In this section, the use of structure /PM0/ABQAZEROFIELDS of the migration BAPI is explained.

If, on the call of the mass data transfer, the parameter for preassignments is set (field /PM0/BAPI_ABQ_MIGRATION-IF_PREASSIGN, see Section 'Scope of Technical Migration'), the standard preassignments are applied as defined in the static product information of In-Force Business Configurator if no value was supplied for the fields in question.

It is only possible to check if a value in a migration structure was supplied by using IS INITIAL. Since for NUMC fields the value 0 and for Char fields the value empty are also evaluated as INITIAL, the problem arises that a desired value 0 or empty may possibly be overwritten with standard values.

To avoid this situation, there is a special table parameter (ABQAZEROFIELDS). You can use it to determine which field is to receive the value 0 or empty for which entities and versions, and thus protect it from preassignments.

Recommendation

SAP recommends supplying all the persistent data in the migration file, if possible, so that it is not necessary to perform the preassignments at all.

7.3.3.4 Taking Account of Templates from In-Force Business Configurator in Migration

From the perspective of the FS-PM in-force business management system, we talk about product element templates that are defined in In-Force Business Configurator and that are identified by a template ID; in the policy business object this is referred to as the PM_ID (product engine), for reasons

of downward compatibility. In this document, the terms PM_ID and template ID are used synonymously.

The correct handling of these PM-IDs by the product engine is a central issue in the migration of policies from the legacy system to FS-PM:

The product engine works with product modules (these correspond to the product elements on the in-force business side), and these product modules likewise have a unique ID. From the perspective of the product engine, a concrete product is a module for instance, also a benefit object. This ID is stored in FS-PM in every entity of the policy business object in the PM_ID field. Thus, for the communication between FS-PM and the product engine, a unique assignment of product elements to product modules is created.

Consequently, during migration, values are expected in the PM_ID field in the structures of the migration interface (DDIC data type /PM0/AB*Q*) that are stored in In-Force Business Configurator (transaction /PM0/ABC_IBC) as template IDs. In the following example, for the migration of a family coverage policy the value 6940000S0000 was delivered in structure /PM0/ABQAPOLICY in the PM_ID field:

In-Force Business Configurator > Prod.Templates > Display Template

Template ID: 6940000S0000 Family Coverage: Family Coverage = PM_ID in migration structure /PM0/ABQAPOLICY

Template Status: 3 Released

Basic Data

Long Name of Template: Family Coverage

Short Name of Template: VS0001

☒ Import Lock for Template Text

Entity ID: /PM0/ABDAPOLICY Basis Policy

Sales Period From: 01.01.2000

Sales Period To: 01.01.2009

Product Assignment: 6940000SPS20000101

Template Type: 1 Product Template

Sales Product Group: P P&C/Non-Life

☐ Only in Master Policies

Figure: In-Force Business Configurator policy template

SAP assumes that the PM_ID is also transferred when the migration BAPI is called. If this is not on one occasion possible, you can proceed as follows:

- Instead of the PM_ID (which remains *initial* in this case), you transfer a text key (field ID_ATTR) in the migration structures, which is subsequently resolved in a PM_ID.
- You can call the BAPI /PM0/BAPI_ABQ_POL_MAP_PMIID in your BAdI implementation of /PM0/ABQ_POL_TECHCHECK_MI_BADI. This BAPI is not called in the standard system during migration. Customers must implement the call themselves.
- In this case, you can use class /PM0/CL_ABQ_TECHCHECK_BADI of the sample BAdI implementation as a template for your own BAdI implementation.

Recommendation

SAP recommends only using this procedure in exceptional cases. The recommended procedure is the provision of the PM_IDs in the supplied migration data.

This procedure may be used, for example, when only one module from a group of modules is ever used in a specific product and these modules share the same value for the identification attribute. Then you can easily determine the PM_ID of this module from a mapping instead of always including it in the migration data. However, for run time reasons, you should only use this method if a very small number of policies is to be migrated.

Example:

Personal liability products are identified in the product engine by the value *Personal Liability* of the identification attribute. In FS-PM, this is reflected in In-Force Business Configurator as follows:

In-Force Business Configurator > Prod.Templates > Display Template

Template ID: 5T40000S0001 PH0001 Personal Liability 2005
Template Status: 3 Released

Basic Data Chars. **Attribs** Subordinate Templates Bus.Transactions Where-Used

Static Attributes

Field Name	Description	Attribute Value	Desc. of Attribute Value
ADJUSTMENT_END	Adjustment End Date	0	Not Relevant
ADJUSTMENT_INCLUSION	Inclusion of Adjustment	0	No Restriction
ADJUSTMENT_START	Adjustment Start Date	0	Not Relevant
CONTRACT_END	Contract End	105	User-Defined; Derivation from Policy End Date
EXTDURACPER_AEND_DAYS	Accept.Period After Ctr End (Da...	15	
EXTDURACPER_AEND_MONTHS	Accept.Period After Ctr End (M...	1	
EXTDURACPER_END_DAYS	Accept.Period Before CtrEnd (D...	5	
EXTDURACPER_END_MONTHS	Accept.Period Before CtrEnd (M...	1	
EXTENSIONDURATION	Extension Duration	12	
EXTERN_REPLACEMENT_BUSINESS	Ext. Repl.Bus.Possible	0	No
IDENTIFIER_ATTRIBUTE	Identification Attribute	Privathaftpflicht	
LOB_CD	Name of Line of Business	LP	P&C (Liability and Personal Accident Insurance)
MAIN_DUE_DATE	Main Due Date	4	Contract End Date
PREMIUM_END	Premium End	2	Contract End
PRODUCTGROUP	ID of Product Group		
PRODUCT_MODULE_ID	Product Module ID	LP_PROD_1	

Figure: In-Force Business Configurator identification attribute

In the corresponding migration structure, in this case /PM0/ABQAPOLPR, the PM_ID field would remain blank and the ID_ATTR field would receive the value *Personal Liability*.

The attribute value *Personal Liability* would be mapped to PM_ID 5T40000S0001 using the BAPI /PM0/BAPI_ABQ_POL_MAP_P MID.

A mixture of the provision of the PM_ID and the provision of the ID_ATTR value is also possible, for instance the PM_ID could be provided in migration structure /PM0/ABDAPOLICY and the ID_ATTR in structure /PM0/ABQAPOLPR.

7.3.4 Special Features of Historical Migration

As explained in Section 5.3 ('Historical Migration'), during the historical migration of policies by the migration project, historical policy versions with different validity dates are provided for the migration, and FS-PM must be able to process all of them. These policy versions must be in the correct historical order with an ascending version ID, and supplied to FS-PM in a file (IDoc).

In FS-PM, the policy versions are consistently treated as change versions with the corresponding change category, regardless of whether they originate from scheduled changes or irregular changes in the legacy system. Consequently, they undergo a complete application check in the validation, in the same way as for New Business. For this reason, you must also transfer the complete business object for each policy version to the migration BAPI, as described in Section 6.3.3.2 ('Display of Policy Business Object and Its Versions in Migration File'). Without any additional measures that have to be performed by the migration project, there is no information in FS-PM as to what changes in the legacy system created the policy versions. The migration of business transactions behaves differently.

The validation ensures that an update is run between the editable, migrated policy versions that are delivered. For additional information about the functional validation, see Section 'Functional Validation'.

Recommendation

Consequently SAP recommends only migrating policy versions that were created in the legacy system as a result of irregular changes. Assuming the corresponding version data is delivered, the scheduled processing activities can usually be performed automatically during the update in the course of the validation. However, for the start date of the historical period that you want to be able to process retroactively in FS-PM, you must always migrate a policy version, regardless of whether the version can be processed in the legacy system with this validity date.

For more information about exceptions that require the delivery of additional versions, see Section 'Overview of Legacy Data Migration to FS-PM'.

Temporary journal data is written during the migration. You must delete this data before you can process the policy in FS-PM. For additional information about the deletion of temporary data, see Section 'Deletion of Temporary Data of the Migration of Policies'.

7.3.5 Migration of Business Transactions: Reversal

A frequent requirement of migration projects is the migration to FS-PM of policies that were reversed in the legacy system with the option of reinstating them in FS-PM: FS-PM takes account of this in the migration interfaces and allows an automated migration of these policies.

As a starting point for reinstating reversed policies after migration to FS-PM, it is necessary to migrate the policy reversal or contract reversal.

You do this by supplying the last valid version before the reversal was executed in the legacy system as the policy version. You also transfer in structure `/PM0/ABQACANCEL` of the migration interface the required data to execute the reversal in FS-PM, such as the execution level (contained in the value of field `PARENT_ID`) and the reversal date. This means there must be a version migrated for processing, with validity date before the reversal date.

When the required data is delivered, the execution of the reversal on the migrated policy is scheduled during the technical migration as a BTS date, and execution takes place in the functional validation during the update.

The method described can be used for reversals at both policy level and contract level.

After the technical migration, you can also use a BADl to execute additional business transactions as BTS dates during the functional validation. For more information, see Section 'Migration of Business Transactions'.

7.3.6 Migration of Business Transactions: Perform Premium Waiver (Life)

A frequent requirement of migration projects in connection with the migration of policies of the Life line of business is the migration to FS-PM of policies that were subject to a premium exemption in the legacy system with the option of annulling the premium waiver in FS-PM. FS-PM takes account of this in the migration interfaces and allows an automated migration of these policies.

As a starting point for annulling the premium waiver of contracts after the migration to FS-PM it is necessary to migrate the *Perform Premium Waiver (Life)* transaction.

You do this by supplying the last valid version before the premium waiver was performed in the legacy system as the policy version. In structure `/PM0/ALQAPXN` of the migration interface, you also transfer the required data to perform the premium waiver in FS-PM, such as the execution level (contained in the value of field `PARENT_ID`). This means there must be a version migrated for processing, with validity date before the validity date of the premium exemption.

When the required data is delivered, the premium waiver is scheduled for the migrated contract during the technical migration as a BTS date, and the premium waiver is performed in the functional validation during the update.

After the technical migration, you can also use a BADl to execute additional business transactions as BTS dates during the functional validation. For more information, see Section 'Migration of Business Transactions'.

7.3.7 Additional Data from the Legacy System

The FS-PM migration interfaces offer the option of migrating data for policy versions for information purposes that are not stored in the *Policy* business object, and that are to be persisted in customer-specific tables.

This can involve, for example, information about processing carried out in the legacy system in the past, for instance the last processor responsible or similar information.

To display this data, you can use the *Additional Data from the Legacy System* pushbutton in the overview of the contract journal overview in the FS-PM business process *Inquiry* for migrated policies. FS-PM does not deliver a dialog box for this function in the standard system, meaning if you choose this pushbutton, an empty dialog box appears. You have the option of creating your own dialog box in the PBT workplace in dialog model P_B_D_PLJ_LEGACY for channel_ID P_B_PLJ_LEGACY_MDIA_CVIEW.

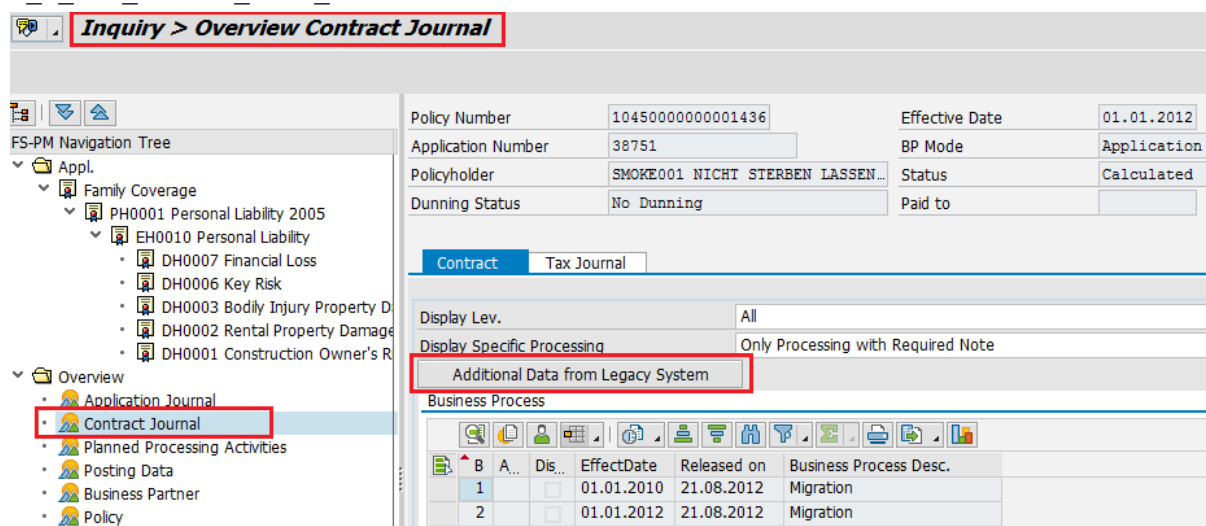


Figure: Inquiry contract journal

7.3.8 Migration of Scheduling Dates

7.3.8.1 Migration of Regular Scheduling Dates

You can deliver regular scheduling dates using the migration structure /PM0/ABQADATE of the migration interface.

During the validation, however, the standard system scheduling date update in the time model is called in the release of each migrated version, which calculates the scheduling dates to be executed using the application status data. This checks for each scheduling date that is entered for the present product module group (in dimension TIMEMODEL1) whether a change to the scheduling date delivered or an initial scheduling of the date is to be made. As following versions transfer the scheduling dates of the previous version, and automatically make a scheduling date update based on them, there is only any point in delivering the dates for the first migrated version.



Tip

SAP recommends you to deliver regular scheduling dates only for the first processable migrated version.

7.3.8.2 Migration of Related Dates

You can deliver related dates using the migration structure /PM0/ABQUTRDPTDT of the migration interface.

Providing a related date does not check its scheduling independently (method UPDATE_AFTER_UNPLANNED at the /PM0/*TMF4* class), the scheduling date is transferred. Such an independent check is not usual, and as a rule not undertaken.

The migration of related dates is mostly required in the context of the document migration. For each document type for which documents are delivered, you need to assign a related date to a document type in the Customizing activity *Assign Dates to Document Types*, and supply this related date for the processable migrated versions.

For additional information about processing documents during the migration to FS-PM, see the Section 'Document Processing in Policy Migration'.

7.3.9 Enhancement and Settings Options

The following enhancement options that do not involve modifying the system are available to you for enhancing the functions delivered for the technical migration to FS-PM:

- Option for enhancing the migration data structures
 - New data fields in the tables delivered
 - You define these in customer includes (customer attributes) or append structures (partner attributes) of the FS-PM application tables. In the migration structures, transfer this data in the `EXTENSIONIN` parameter. For information about the migration of your own data fields, see Section 'Enhancement of Migration Data Structures with Non-SAP Fields'.
 - New tables not belonging to SAP

Development partners for add-on lines of business usually require new tables. For information on the migration of data for entities not delivered in the SAP standard system, see Section '

Treatment of Add-On LoBs for the Migration of Policies’.

- Option for enhancing the migration level of functions

By implementing BADIs by partners or customers

For information, see Section ‘Enhancement of the Migration Level of Functions’.

7.3.9.1 Enhancement of Migration Data Structures with Non-SAP Fields

You can enhance the FS-PM application tables delivered with your own attributes using the attributes of customer includes or append structures. To do this, it is necessary that the FS-PM migration interface supports the downloading of this data to FS-PM.

As the enhancement of structures in BAPI interfaces is not allowed, there is a generic import mechanism for the FS-PM migration interfaces with `EXTENSIONIN` parameter in the BAPI. This means the FS-PM migration data structures can be enhanced without modifying the system. Using this import mechanism, you can add customer or partner-specific data to the FS-PM migration structures delivered.

For more cross-application information about this enhancement mechanism, see SAP Library under *Customer Enhancement and Modification of BAPIs (CA-BFA)*.

The `EXTENSIONIN` parameter is used in FS-PM as follows:

For the `EXTENSIONIN` parameter of the FS-PM migration interfaces, there is a BAPI tables extension (name: `/PM0/BAPI_TE_<Table Name>`; for example `/PM0/BAPI_TE_ABQAPOLICY`). This is used for the categorization of the data content of the `EXTENSIONIN` parameter. The fields from the customer includes or append structures you have appended to the operational FS-PM tables you also have to append to the related BAPI tables extension. Further, you also have to append your own fields to migration data structures (such as `/PM0/ABQAPOLICY`). As these fields are grouped in one customer include, you always have to include only this particular include in the relevant structures and tables. The transfer of this data from the migration interface to the actual FS-PM import structures occurs automatically.

Tip

The length of all customer/partner fields for an entity must not exceed 945 characters.

In the structures `/PM0/BAPI_TE_<table name>` you have to replace the numeric values by CHAR fields with a fixed length, because the BAPI extension structures can only be enhanced alphanumerically.

For fields with data type DEC, you have to reserve an additional place in the migration file for displaying the decimal point.

Recommendation

SAP provides the transaction `/PM0/ABQ_EXT_CONV` on the SAP Easy Access screen with which you can convert customer and partner fields automatically to an `EXTENSIONIN` parameter. Proceed as follows:

- Add the partner and customer fields to the migration structure `/PM0/A+QA*` (for instance `/PM0/ABQAPOLICY`) in the same order as you added them to the entity (for instance `/PM0/ABDAPOLICY`) by customer include or append.
- Create a migration file that contains all the fields of the extended entity in one line.
- Convert this migration file using transaction `/PM0/ABQ_EXT_CONV`. Lines of the type `BAPIPAREX;[enhanced_entity]` (for instance `BAPIPAREX; /PM0/ABQAPOLICY`) are added to the migration file. The converted file is saved in the migration file folder with the suffix `_converted`.
- Download the converted file to the migration tool (for example, the LSMW) in the normal way.

For additional information about using this program, see the program documentation in the system.

7.3.9.2 Enhancement of the Migration Level of Functions

The enhancement of the migration functions without modifying the system is ensured by BAdIs, in which partners and customers can define their own code.

Additional cross-application information about the enhancement mechanism can be found in SAP Library under *Business Add-Ins (BAdIs)*.

You can enhance the technical migration by using BAdI `/PM0/ABQ_POL_TECHCHECK_MI_BADI`.

BAdI `/PM0/ABQ_POL_TECHCHECK_MI_BADI`

This BAdI is for checking and changing the data supplied for a policy version.

There is no standard implementation.

All table parameters of function module `/PM0/ABQ_POL_MASSMIGR` from the category of the migration structures are transferred as changing parameters. This means partners and customers can check data in the technical data check, and also change data.

This BAdI is initially called in the BAPI for the mass transfer of policies. Only then are the checks and processing of the FS-PM standard carried out.

For additional information, see the BAdI documentation in the system.

The BAdIs that you have to implement for the technical migration of non-SAP related entities are described in Section 0.

7.3.10 Historical Migration of Policies with Actual Investment

If you face the challenge of having to execute a historical migration of policies based on actual investment, and the historical payments based on actual investment are to be available for the migrated policies for the complete historical time interval to be migrated, SAP suggests considering the following high-level solution idea. This is SAP's proposal on how to ensure the correct processing of out-of-sequence changes and a correct handling of historical payments in FS-PM for migrated policies. However, SAP cannot guarantee that the approach described below is complete and that it will match your specific project needs.

Provide the following data with the migration BAPI:

- Cash flow documents
- Info container records
- External dates
- Non cash flow documents

To be able to do so, the migration project needs to implement some enhancements of the basic migration infrastructure to support migration and validation of the entities listed above in the context of actual investment. Important enhancements are:

- Migration interface
 - Enhance CF document header input: Add preliminary `DOCNRCE_ID`.
 - Enhance CF document item input: Add preliminary `POSITIONNR_ID`, `POSNRCD_ID`.
 - Provide new interface for info container input.
 - Enhance external dates input: Add `INFOEXTDT_ID`.
- Technical migration
 - Implement logic to replace the preliminary `DOCNRCE_ID` and preliminary `POSITIONNR_ID` with final numbers both in the cash flow and info container accordingly (preserve the link).
- Functional validation
 - Enhance the CF document comparison variant (avoid mixing up Down Payment Request CFs with Debit Entry CFs).

When these enhancements are available, provide data for the enhanced migration interface in the technical migration:

- Down Payment Request CF document data

- Debit Entry CF document data
- Info container data
- External date data
- Non CF document data, if required

Do **not** provide Incoming Payment CF data. This will be created during the validation.

Down Payment CF document	Header		DOCNRFC_ID (preliminary)	...	...	
	Item 1		DOCNRFC_ID (preliminary)	POSITIONNR_ID (preliminary)	...	
	Item 2		DOCNRFC_ID (preliminary)	POSITIONNR_ID (preliminary)	...	
Debit Entry CF document	Header		...	...	DOCNRCO_ID (Number of incoming payment header in FS-CD)	...
	Item 1		...	..	...	POSNRCD_ID (Number of incoming payment item in FS-CD)
	Item 2		...	..	...	POSNRCD_ID (Number of incoming payment item in FS-CD)
Infocontainer		DATA 1 (Number of incoming payment header in FS-CD)	DATA 2 (Number of incoming payment item in FS-CD)	AWKEY (preliminary DOCNRFC_ID)	BELNR (preliminary POSITIONNR_ID)	INFOEXTDT_ID
Externam Date		INFOEXTDT_ID	...	...	...	...

Figure: Data to be provided in migration interface

To be considered when providing this data:

- The artefacts must be linked with each other as shown in the figure above (same box color = same value).
- DOCNRFC_ID (=AWKEY) and POSITIONNR_ID (=BELNR) might not be needed if the customer has implemented enhancement spot /PM0/GET_TYPE_ADD_DEPOSIT. It then depends on the customer implementation; the standard fallback class requires the data.

7.4 Functional Validation

You can use the functions in the functional validation process to ensure the consistency and accuracy of the migrated policy versions. In the case of historical migration, the editable, migrated versions are also updated during the validation process.

The functional validation can either occur automatically after the technical migration (see Section 'Scope of Technical Migration') or manually using the transaction *Validate Migrated Policies* (/PM0/ABQ_FP_Q_POLVAL). You can adapt the functional validation to your requirements using various BADIs.



Tip

SAP recommends that you carry out the technical migration and functional validation in two steps when migrating to the production system. This allows you to achieve a

complete parallel processing capability and use the user-friendly application log of the transaction for the validation of migrated policies.

The following figure provides an overview of the basic steps of the functional validation. Not all of the steps are important in every migration scenario.

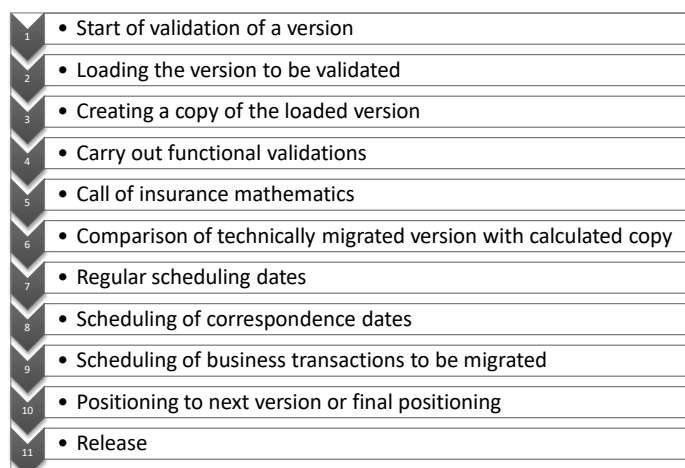


Figure: Process flow functional validation

- Creation of a copy of the migrated policy version to be validated (so that the technically migrated policy version remains unaltered)
- Carrying out derivations and checks on the copied policy version
- Calculation of the copied and checked policy version by the product engine
- Comparison of the calculated copy with the unchanged technically migrated policy version
- Scheduling of dates and business transactions to be migrated, and update to the next policy version or final positioning to be validated
- Release of the validated policy versions, and assignment of the migrated documents to the versions of the policy created by the update in FS-PM

7.4.1 Prerequisites

The policy versions to be validated must be technically migrated and show the corresponding migration status. For information about the migration status, see Section 'Migration Status'.

You must have implemented the enhancement options required for your migration project before the validation. For an overview of the most important BADIs for adapting the validation, see Section 'Enhancement and Settings Options'.

7.4.2 Scope of the Functional Validation

The following explanatory figure shows the process flow of the transaction for validating migrated policies in detail, and provides an overview of where enhancement options are available:

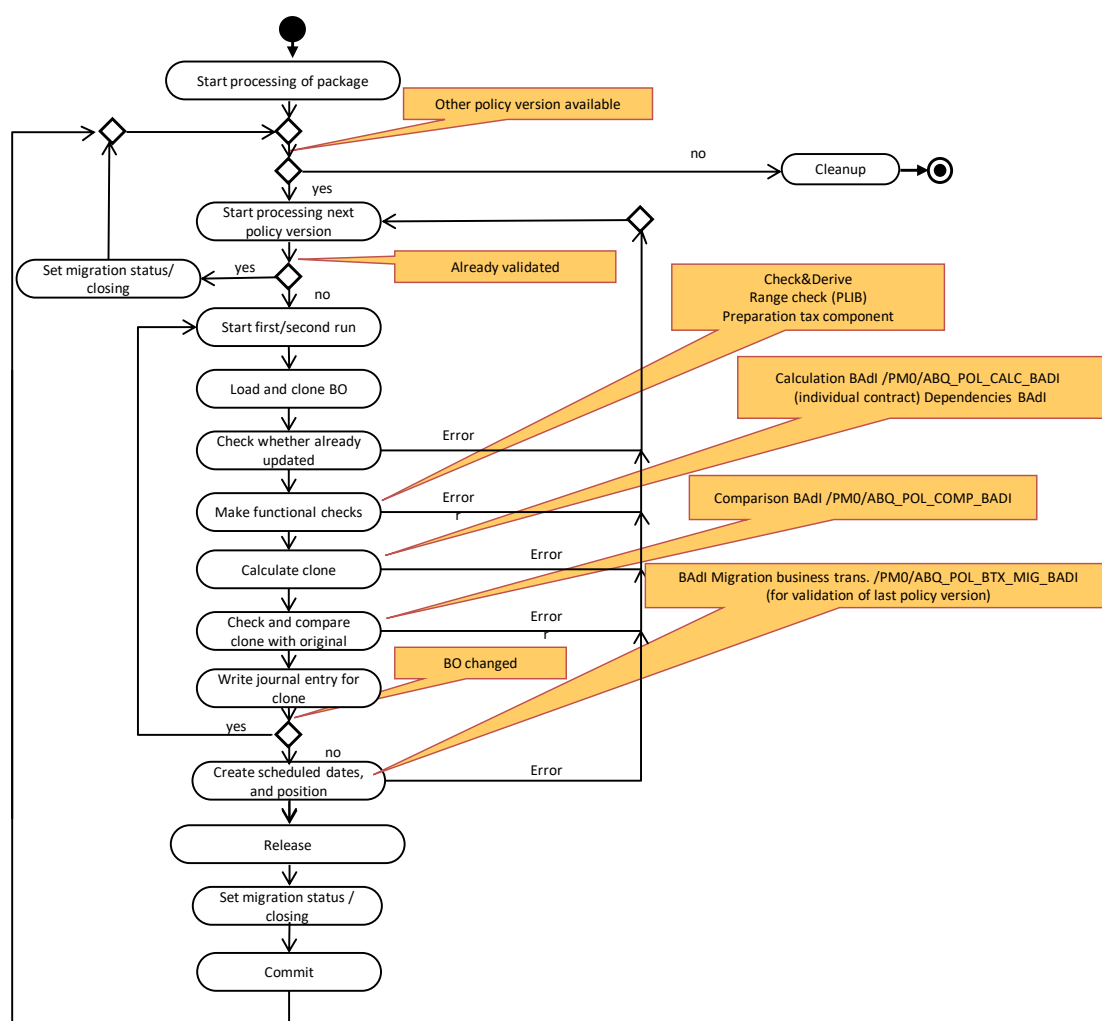


Figure: Detail process flow for functional validation

In the following overview, you can find a description of the individual steps of the functional validation. An explanation of the options for adapting the validation to your requirements using control parameters is at the end of the overview.

Table: Steps for the functional validation

Step	Description of Step from Figure Above	Can Be Influenced By Specified Control Parameters
1	START PROCESSING NEXT POLICY VERSION Initialization: Selection of data for the migration, filling of technical fields and buffer tables for later quick access. Layout of control structure specifying for all subsequent steps which version is to be processed	Cannot be influenced
2	START FIRST/SECOND RUN Set the migration status with value 001 – Migrated, not checked.	Cannot be influenced

3	LOAD AND CLONE BO Loading of the application and creation of a copy of the application on which the following processing is carried out without losing the initial technically migrated version. Continuation of Check and Derive on pressing Enter, to make derivations on the copy.	CLONE_FG DERIVECLONE_FG
4	CHECK IF UPDATED Check whether an update has already begun. If this step has been reached and an error occurred, the validation cannot be continued since no clear starting point is known.	Cannot be influenced
5	APPLY BUSINESS RULES Make checks and derivations with Check and Derive (scope as for New Business application check) and checks for current settings of In-Force Business Configurator (cardinality checks). In addition, preparatory tax treatment steps are made.	CARDCHECK_OFF_FG By implementing BAdI method /PM0/IF_EX_ABQ_POL_VAL_CNTRL~PERFORM_CREATE_RULE you can enter default values for entities. A sample implementation exists.
6	CALCULATE CLONE Call of the insurance mathematics using an LoB-dependent (filter criterion) BAdI (/PM0/ABQ_POL_CALC_BADI). Check of dependency plan and also response to possible dependency changes.	CALCULATE_FG CLONE_FG CHECKDEP_FG
7	CHECK AND COMPARE CLONE WITH ORIGINAL Make a comparison between the original technically migrated version and the copy using a BAdI (in which the calculated copy can also be changed, see /PM0/ABQ_POL_COMP_BADI). If a difference is identified, continue with step 2, and following the changes a second calculation can be made.	COMPARE_FG
8	WRITE JOURNAL ENTRY FOR CLONE Recording in journal and fixing of changes.	ONLYDISPLAY_FG
9	CREATE SCHEDULED DATES, AND POSITION Business transactions can be migrated by implementing BAdI /PM0/ABQ_POL_BT_X_MIG_BADI and scheduled as a BTS date. The time model calculates the regular scheduling dates according to its settings. If, for a historical migration, a later version is processed, the dates of the previous version are transferred to the current version, as otherwise the time model does not recognize that existing dates have to be updated and not recalculated. The positioning is then performed, up to the target date that represents the maximum from the target date and the latest migration related date for the document processing. If, for a historical migration, there is a version to be validated with a later validity date, the system updates to the next version.	UPDATE_FG TARGETDT_FG DELETECORRDT_FG

10	RELEASE (APPROVE) In the standard system, the transfer of the application to a policy version and also the release to the accounting component. Entities that are in the current version, but are missing in the following version to be functionally validated, are still processed.	Can be influenced by Customizing of the release variant for migration. You release a validated version using process type <i>CPT0040</i> and process category <i>SPC0040</i> (release of in-force business migration (for each version to be validated)). For Customizing options, see Customizing for Policy Management under <i>In-Force Business Management -> Basis -> Business Transactions, Actions, and Process Steps</i> .
11	PRESET STATUS IN MIGRATION TABLE Preparation of statistical data in a BAdI (see name: <i>/PM0/ABQ_POL_STATISTIC_BADI</i>).	Cannot be influenced.
12	SET STATUS IN MIGRATION TABLE Setting the migration status to value <i>005</i> : migrated; checked successfully; released – unless the release should not follow (meaning step 10 has not been completed): status <i>006</i> is then set. Journal and migration journal are likewise updated (set to 'successful'). If necessary, the temporary data can be deleted after successful validation of the last delivered version.	ONLYDISPLAY_FG You can define whether the temporary data should be deleted automatically by implementing method <i>/PM0/IF_EX_ABQ_POL_VAL_CNTRL~CHECK_DELETE_TEMPDATA</i> for business process Q.

Transient Check and Derive derivations are always switched off in the system in the validation.

Insured objects are not automatically synchronized during the functional validation. You need to ensure that the correct journal numbers of the insurable objects are transferred in the policy migration files.

The requirements of the steps to be made in the functional validation can be very different, depending on the migration scenario (last version migration, historic migration), on the quality of the data provided or on the migration project phase (migration tests, productive migration). FS-PM therefore provides you with flexible control of the validation options, allowing you to adapt the validation for each version to be validated in order to meet these requirements. You can control this by the corresponding values in control structure */PM0/ABQ_VALIDATION_CNTRL*, which can be supplied along with the migration file.

By setting the control parameters you can carry out or prevent individual validation steps. In doing so, you have to note that the option of preventing steps is dependent on the data quality of the policy versions supplied, and that preventing steps can affect whether you can process the validated versions.

The characteristics of the control structure have to be defined for each version to be validated. This control structure can be made available to the validation in different ways:

- Per migration interface in parameter */PM0/E1BP_ABQ_AVALCTRL*. This control structure can be transferred in the migration file under the identifying field content */PM0/ABQ_AVALCTRL* for each version (*VERSION_ID*) and policy (*PARENT_ID*).
- By implementing BAdI */PM0/ABQ_POL_VAL_CNTRL_BADI* using method */PM0/IF_EX_ABQ_POL_VAL_CNTRL~CHANGE_CNTRL_INFO*. There, the control structure transferred to the migration interface can also be overridden for the last time.
- The control parameters are not explicitly specified. The steps outlined in the next figure are then carried out in their entirety and the resulting validated version can then be processed.

The following explanatory figure provides an overview of the available control parameters and their usage.

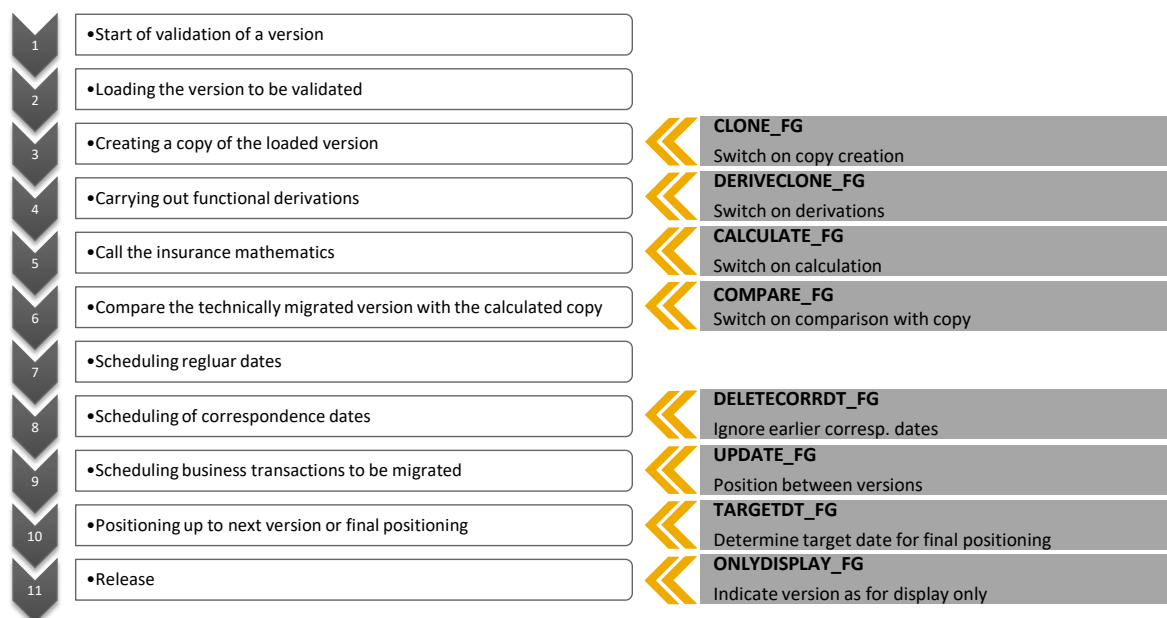


Figure: Control parameters for the functional validation

In addition to the options detailed here, you can switch off the cardinality checks and the checks made of dependencies between the contracts.

Below is a description of the prerequisites for setting the control parameters and how to set them, and the effect that setting the individual control parameters has.

ONLYDISPLAY_FG

With this indicator, you can control whether the validated policy version in FS-PM can be loaded for display only.

Effects if indicator is set:

Processing leading to changes prior to the first processable policy version in FS-PM is prevented on the validated policy. If you attempt this, you receive corresponding error messages. Policy versions migrated for display can only be displayed in the *Inquiry* business process.

Actions and business transactions are also not possible in the *Inquiry* business process, with the exception of purely display actions and business transactions. The executable actions are determined based on the entries in the S table /PM0/ABUADISPACT. The executable business transactions are determined on the basis of the entries in table /PM0/ABUUBTX. You cannot schedule unscheduled (irregular) dates that are based on these versions. In the contract journal, policy versions migrated for display only can be recognized by the *Display Only* indicator being set.

Inquiry > Initial Screen						
Execute						
Inquiry Contract Journal						
BP No.	Application Number	Display Only	EffectDate	Released on	R	
1		☒	01.01.2009	22.08.2012	H	
2		☒	01.07.2009	22.08.2012	H	
3		☒	01.10.2009	22.08.2012	H	
4		☐	01.07.2010	22.08.2012	H	

Processing						
S...	Contract Number	Jour...	Display Only	Processing Text	Eff. Start	
1	Policy	1	☒	Policy Issue - New Business	01.01.2009	
1	11400000000006056	1	☒	Policy Issue - New Business	01.01.2009	
2	Policy	2	☒	Change External Reference	01.07.2009	
2	11400000000006056	2	☒	Change External Reference	01.07.2009	
3	Policy	3	☒	Change Payment Frequency	01.10.2009	
3	11400000000006056	3	☒	Change Payment Frequency	01.10.2009	
4	Policy	4	☐	Change Duration	01.07.2010	
4	11400000000006056	4	☐	Change Duration	01.07.2010	

Figure: Contract journal display migration

CLONE_FG

You can use this indicator to control whether a copy of the technically migrated policy version is to be created during the validation.

Effects if indicator is not set:

If you do not create a copy of the technically migrated version, you cannot make a comparison of the initial migrated policy version with the version of this policy calculated by the product engine. If you want to make this comparison, which means setting the indicator `COMPARE_FG`, you also have to set `CLONE_FG`.

DERIVECLONE_FG

You can use this indicator to control whether derivations should be made once the copy has been created.

Effects if indicator is not set:

No defaults are executed on the copy once it has been created.

COMPARE_FG

You can use this indicator to control whether the version calculated by the product engine is to be compared with the initial migrated version. It only makes sense to set this indicator if `CLONE_FG` is also set, and consequently a copy is created on which the calculation is made.

Effects if indicator is not set:

In the validation, you cannot check whether unacceptable changes in comparison with processing in the legacy system have resulted from the calculation by the product engine connected to FS-PM. SAP recommends that you make sure in advance in a migration test system that only acceptable variances occur through the calculation in FS-PM.

CALCULATE_FG

You can use this indicator to control whether a calculation of the policy version to be validated is to be made by the product engine. If a copy has been made, the calculation is made in the copy, otherwise the calculation is made directly on the technically migrated version.

Effects if indicator is not set:

You cannot make a comparison of the copied, calculated version with the technically migrated version.

Recommendation

SAP recommends that you only set this indicator together with `CLONE_FG` and `COMPARE_FG`. For a calculation of the original, technically migrated policy version, there is the danger that persistent data is changed by the calculation, and the technically migrated version becomes incorrect. SAP therefore recommends that, if possible, you supply all persistent data during the technical migration, so that you can dispense with a calculation at least during the productive migration.

UPDATE_FG

You can use this indicator to control whether the validated policy version is to be positioned.

Effects if indicator is not set:

No documents or scheduled dates are produced or performed for the periods in which positioning is prevented.

It is essential that you set this indicator for policy versions that are to be migrated not just for display purposes in FS-PM.

If documents are delivered during the migration, the status must be positioned to at least the date of the most recently delivered document. If no documents are delivered during the migration, the status must be positioned to the date up to which documents were created in the legacy system. This can be ensured by corresponding related dates. For more information, see Section 'Document Processing in Policy Migration'.

The indicator must also be set for the latest policy version to be validated if business transactions are to be migrated (Execute Reversal and Perform Premium Waiver (Life) or by a BADl implementation). The target-determining BADl must in this case determine at least the latest effective date of the migrated business transaction as a target date, and `TARGETDT_FG` must be set so that the target date is determined.

TARGETDT_FG

You can use this indicator to control whether the target date determination is to be taken into account for positioning.

Effects if indicator is not set:

The latest supplied migration-related date or, for a historical migration, the effective date of the next version to be validated, represent the target date. If you migrate business transactions, you have to set this indicator. For performance reasons, you should consider positioning only as far as required in the validation, and to perform the further update in a separate, downstream update run in FS-PM.

CHECKDEPS_FG

You can use this indicator to control whether cross-contractual dependencies should be checked and, if necessary, a response made to these dependencies.

Effects if indicator is not set:

No check for cross-contractual dependencies is made, and you must ensure that such dependencies are already included in the policy versions delivered.

CARDCHECK_OFF_FG

You can use this indicator to control whether to use cardinality checks in the validation.

Effects if indicator is set:

No check is made of the cardinalities allowed that are defined in In-Force Business Configurator. If you do not wish to check the cardinalities during the migration into the productive system for runtime reasons, SAP strongly advises you to make checks in advance by carrying out a test migration of the policies.

DELETECORRDT_FG

You can use this indicator to control whether correspondence dates that are scheduled before the version currently to be validated should be deleted.

Effects if indicator is set:

No correspondence is created for the correspondence dates that had been scheduled for a date earlier than the effective date of this version.

If correspondence had already been created for the policies to be migrated in the legacy system, you should set the indicator at least for the latest policy version to be validated.

CAUTION

Note that statuses that are not positioned (`UPDATE_FG = NO`) can be used for display purposes only in the *Inquiry* business process (`ONLYDISPLAY_FG = YES`). If your BAdI implementation for `/PM0/ABQ_POL_VAL_CNTRL_BADI` delivers one of the settings that violate this condition, the validation is terminated with an ABAP runtime error. Similarly, you are only allowed to compare the initial policy version migrated with the calculated status (`COMPARE_FG = YES`) if a copy of the technical migration status is created (`CLONE_FG = YES`). The system also checks this condition.

Example

Examples of using the control parameters:

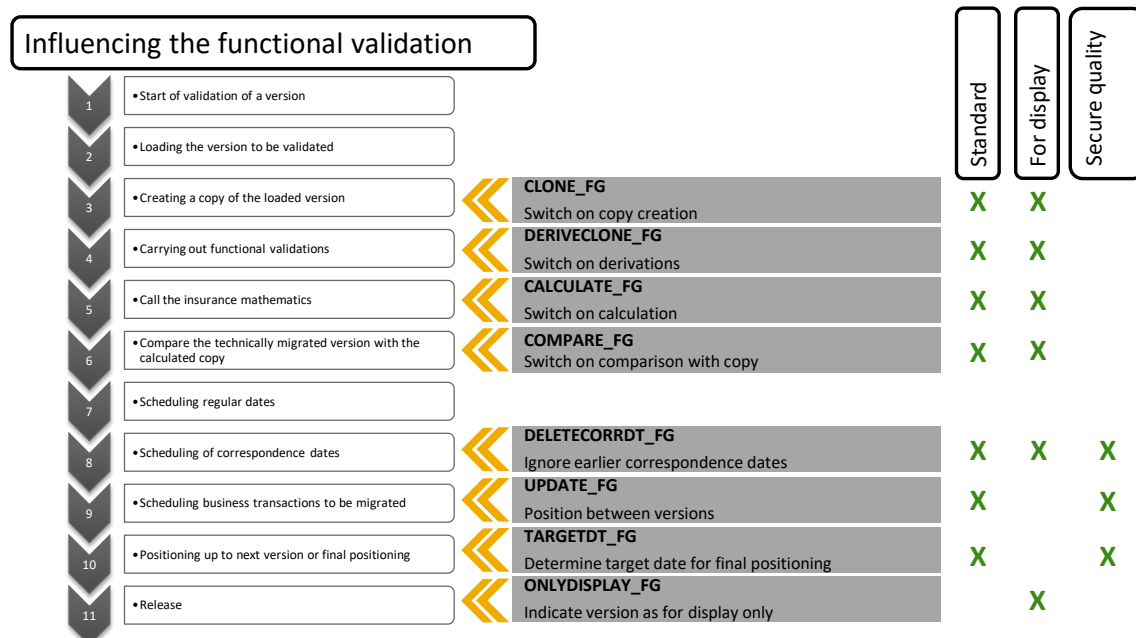


Figure: Control parameter validation – Example 1

- **Standard column:** In the standard system, all indicators are set up to ONLYDISPLAY_FG for the migration of processable policy versions. All of the validation substeps are performed.
- **For Display column:** When migrating policy versions only for display purposes, you must only set ONLYDISPLAY_FG; you must not set UPDATE_FG and TARGETDT_FG. Whether you use the other indicators depends on the requirements of the migration project.
- **Secure Quality column:** If you are able to ensure a high quality for the technically migrated policy versions, for example, by measures in the migration tests, you can deactivate some of the validation options when migrating processable policy versions.

Migration scenario	Validation	To 1.1.2010	From 1.1.2010
• Historical migration • Test migration already taken place • Test migration checked and accepted both technically and functionally • The processable versions should still be calculated • Versions from 1.1.2010 should be completely processable in FS-PM • Versions before 1.1.2010 should be present in FS-PM for display only • Positioning must only occur to the extent documents have been delivered	CLONE_FG Switch on copy creation		
	DERIVECLONE_FG Switch on derivations		
	CALCULATE_FG Switch on calculation		X
	COMPARE_FG Switch on comparison with copy		
	DELETCORRDT_FG Ignore earlier corresp. dates	X	X
	UPDATE_FG Position between versions		X
	TARGETDT_FG Determine target date for final positioning		
	ONLYDISPLAY_FG Indicate version as for display only	X	

Figure: Control parameter validation – Example 2



The values of the control parameters are persisted for the validation of each policy version in the migration journal (table /PM0/ABDQMIGJOUR).

7.4.3 Transaction for Functional Validation

Use transaction /PM0/ABQ_FP_Q_POLVAL (*Validate Migrated Policies*) to perform the functional validation of technically migrated policies. You should only carry out a validation in the technical migration as described in Section 'Scope of Technical Migration' for test activities.

See the following figure for when to start the transaction for the validation of migrated policies:

Figure: Transaction for validation of migrated policies

As for all transactions that are based on FPP in the FS-PM application, you first have to assign a run ID. Using this ID, the input data and logs are identified in the system in the course of further processing.

Using the following selection criteria, you can define the number of policies to be validated:

- Policy number (actual policy numbers or intervals of policy numbers)



Note

The collective search help for this field is not restricted to migrated policies. If you require this, you must enhance the collective search help with an elementary search help.

- Run date of the technical migration
- Run time of the technical migration



Note

There is no point in making an entry under *Run Time of Technical Migration* unless you have entered a date. Otherwise, no policies are selected.

- Release policies



Note

You can only release a policy if all migrated versions of this policy have been successfully validated.

- Number of entries per interval

Here, you define the maximum number of policies that are assigned to each policy interval to be processed during parallel processing of the functional validation. The intervals of the policies to be validated are created again for each validation run (dynamic interval creation). You can set the size of the intervals using this parameter. All migrated policies with the migration status: *Migrated, Not Checked; Migrated, With Errors; Migrated, Accepted Despite Errors, Not Released; Migrated; Checked Successfully, Not Released; Migrated; Checked Successfully, For Display* are taken into account.

Once you have saved the data entered in dialog mode, you can choose the pushbutton *Schedule* to provide the functional validation for execution in the background, for direct execution in the background, or for dialog execution.

Once the validation run has finished, you can analyze the results and messages using various logs.

Figure: Selection options for evaluation of validation run

- Job Logs
 - Display of messages from interval creation and package processing
- Application Logs
 - The system displays for each interval whether there were problems during processing, how many messages, terminations, warnings and error messages there were.
 - You can view detailed messages here by additional functions for each interval or cross-interval. For more information, see the following section.
- Counter
 - Here, you obtain a quick overview of the number of policies selected and the number of policies validated successfully or with errors.

Technically, the system first determines which policies are to be functionally validated for each interval (function module: `/PM0/ABQ_FP_Q_POLVAL_INTERVAL`). The functional validation is then started for the selected policies (method `/PM0/CL_ABQ_BPP_POL_VAL_RUN->START`). For more details on the steps to be carried out, and how to influence them, see Section 'Scope of the Functional Validation'.

The FPP run to validate migrated policies can automatically restart in case a customizable number of its jobs aborts. You can set the threshold of aborted jobs for such an FPP run. When the threshold is reached, the system triggers an ABAP runtime error in the running jobs one by one so that all jobs stop running and then the FPP run restarts. For the jobs that were forced to stop, no policies are excluded from the next run. Setting the threshold improves efficiency because a run otherwise restarts only when all the jobs finish running, whether aborted or completed successfully. This may take a long time even if most of the jobs are aborted. This automatic restart option is useful for mass migration tests. Before the migration into the productive system, of course, you should have reached a level of stability where this option is no longer required.

In Developer Customizing (transaction `/PM0/CUST_INT`), choose *Internal Customizing -> General Settings -> Define Exits for FPP*. In the Customizing activity, assign the FPP exit `ABT_THLD` and a class/interface to mass activity `FP_Q_POL`. A sample class `/PM0/CL_ABY_FPP_INITRUN_PACK` is provided. In the sample class, the threshold value is 50%. You can create a similar class and modify the `DEFINE_ABORT_THLD` method to set a different threshold value.

For additional information, see also:

- SAP Notes [2335969](#) and [2292553](#).
- [SAP Policy Management](#) -> *Application Help -> SAP Library -> In-Force Business Management -> Migration in SAP Policy Management -> Validate Migrated Policies*.

7.4.4 Logging of the Validation Run

7.4.4.1 Display of the Application Log

The messages generated in the validation run are displayed in the application log, sorted for each interval or in their entirety by policy number.

When you choose the *Display Policy* pushbutton, the *Inquiry* business process is started for the policy selected in the list.

Date/User	Number	Object	Subobject
21.08.2012 HENNRICHF	52	Applica...	S... FPI
10450000000001441	2		
10450000000001442	2		
10450000000001443	2		
10450000000001444	2		
10450000000001445	2		
10450000000001446	2		
10450000000001447	2		
10450000000001448	2		
10450000000001449	2		

Figure: Application log

When you choose the *Log for Each Message* pushbutton, the messages are displayed sorted according to their technical ID. This means, for example, that you can quickly see which messages have occurred in a validation run, and how often.

Date/User	Number	Object	Subobject
21.08.2012 HENNRICHF	52	Application Log...	Subobject
10450000000001441	40		
10450000000001442	12		

Figure: Display of messages by technical ID

In addition, you have the option of filtering the messages by whether they are *Error*, *Warning* or *Information* messages.

7.4.4.2 Selection of Messages to Be Logged and Considered

When the functional validation is finished, you can decide whether a message should appear in the application log at all by implementing BAdI `/PM0/ABQ_POL_ERR_HANDL_BADI->HANDLE_MESSAGE`. This means you can exclude predictable messages that you would expect anyway. For example, if you deactivate operational systems during the migration of interfaces to FS-PM, it may make sense initially not to log any messages originating from the operational systems.

Similarly, by implementing BAdI method `/PM0/ABQ_POL_ERR_HANDL_BADI->HANDLE_ERROR`, you can define for each functional error occurring in the validation whether the error is to be considered as an actual error in the current context. If you decide to discard error messages this way, the corresponding migration status is set. For additional information about the migration status, see Section 'Migration Status'.

7.4.5 Update in Validation for Historical Migration

The validation in the historical migration is carried out in the following steps:

- Once the policy version migrated for processing has been downloaded, the time model is initialized based on the data existing in this policy version. Scheduled changes to be executed are determined, and scheduled by the time model as dates for execution.
- The policy version is now saved, and the policy updated to the next migrated policy version. The system then performs all of the processing scheduled between the validity dates of the two migrated policy versions.

This procedure is repeated at least until the policy in FS-PM is brought up to the version valid on the key date. The precise date to which the migrated policy is updated in FS-PM in the validation depends on other parameters, for instance whether related dates for document categories have been migrated (for information, see Section 'Document Processing in Policy Migration'), on the value of the control parameters (see Section 'Scope of the Functional Validation'), or on your settings for the target date determination. You can make the latter settings in Customizing for Policy Management under *In-Force Business Management -> Basis -> Update -> Define Update Types*. Using the validation control parameters, you can in turn ignore the target date determination defined on the basis of these settings.

 Recommendation

SAP recommends that you check, in consultation with your user departments, exactly how far the migrated policies should be updated in the validation. You should then clarify how the control options given above are to be used in order to ensure the required update.

Since, for the historical migration, the update is between the policy versions migrated for processing, you will frequently find that additional processing to be recorded in the journal is carried out between the versions to be migrated. This is explained in the following example:

 Example

You migrate two policy versions for processing in FS-PM. In the context of the technical migration, these contain journal numbers 1 and 2. You can find more information about this in the description of the migration class in Figure: Process flow for technical migration of entities not delivered in the standard system.

If the first migrated policy version is now updated, new journal entries can arise. This means, for example, the first executed scheduled processing has to be recorded with journal number 2.

Consequently, for the technical migration of the policy version, a temporary technical key (GUID) is created. In the validation, application statuses are created that, following successful validation, are provided with the final technical key and saved with the final journal number.

In the technical migration, temporary data is written that has to be removed again before processing the historically migrated policies in FS-PM.

 Note

The previously mentioned conversion in the validation is logged in the migration journal (table /PM0/ABDQMIGJOUR). You can find the following information in this table:

- Functional policy number
- Journal number of migrated status
- Temporary GUID
- Final GUID
- GUID for application
- Indicator whether processing successful
- Indicator which default values and derivations were performed
- Indicator whether temporary data has already been deleted

As already mentioned, you can also find the values for the control parameters for performing the validation in this table.

 CAUTION

Before you can continue processing the validated and released migration policies in FS-PM, you have to delete the temporary data for the migration of the policies.

7.4.6 Deletion of Temporary Data of the Migration of Policies

FS-PM provides the following options for deleting the temporary migration data of policies.

7.4.6.1 Transaction for Deletion of Temporary Data

You can use transaction /PM0/ABQ_DEL_TMPDATA to delete the temporary data.

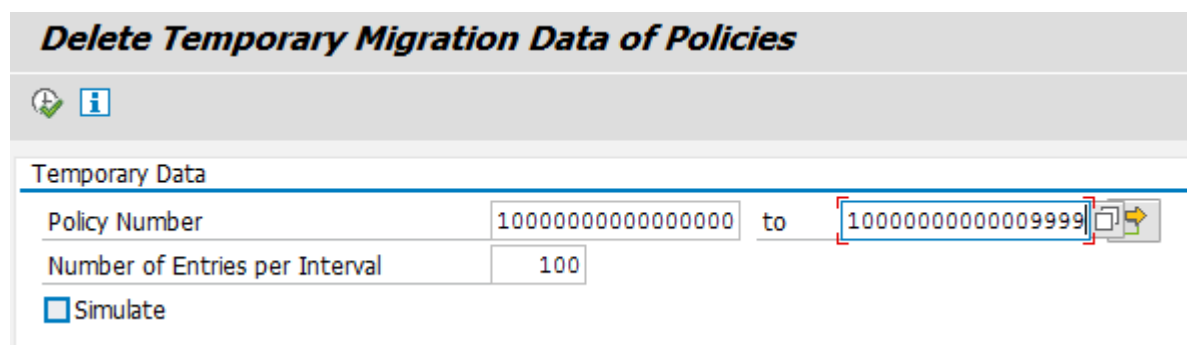


Figure: Initial screen for deleting temporary migration data

You can select policies here. If you do not make any restrictions, all temporary data is deleted. The selection and deletion of the data is made sequentially by package. Depending on the configuration of your system and your database, it may make sense to optimize the size of these packages. You can do this by an appropriate entry in the *Number of Entries per Interval* field.

 Note

The program for deleting temporary data has not been integrated in Framework for Parallel Processing (FPP). However, if necessary you can create a program in the migration project that determines the policies to be processed, divides them into packages, and then executes program `/PM0/ABQ_DEL_TEMPDATA` for each package. Depending on your system and application configuration, this can significantly reduce the runtime of the deletion program.

7.4.6.2 Deletion of Temporary Data in the Validation

It is also possible to delete the temporary data directly in the validation run, following successful validation and release of the last migrated policy version.

To do this, you have to implement method `CHECK_DELETE_TEMPDATA` of BAdI `/PM0/ABQ_POL_VAL_CNTRL_BADI` so that when you call business process Q (= Migration), the return value is filled with X (= TRUE).

With this setting, you no longer need to start program `/PM0/ABQ_DEL_TEMPDATA` explicitly.

7.4.6.3 Deletion of Temporary Data when Starting Business Processes

It is also possible to have the system delete the temporary data if the *Change* or *Universal Change* business processes are started for a migrated policy.

To do this, you have to implement method `CHECK_DELETE_TEMPDATA` of BAdI `/PM0/ABQ_POL_VAL_CNTRL_BADI` so that when you call the business processes A (for change) or V (for universal change), the return value is filled with X (= TRUE).

 Tip

SAP recommends deleting the temporary data of the migration of policies explicitly using transaction `/PM0/ABQ_DEL_TEMPDATA`. By using the transaction, you ensure that:

- You still have the original data of the technical migration available for analysis even after successful validation. You do not retain this option if you implement BAdI method `CHECK_DELETE_TEMPDATA` for business process Q.
- You have the same processing status for all policies migrated and validated. This is no longer the case if you delete the temporary data for individual policies at the start of business processes.

7.4.7 Release in the Validation

You release a validated policy version using process type *CPT0040 Release of In-Force Business Migration (for each version to be validated)* and the related process category *SPC0040*.

The Customizing delivered ensures that, for the release in the validation, the validated policy versions are implemented as legally binding and valid policy versions, and the release is executed in the accounting component.

You define further process steps to be executed in the release in Customizing for Policy Management under *In-Force Business Management -> Basis -> Business Transactions, Actions, and Process Steps -> Define Process Steps* and under *In-Force Business Management -> Basis -> Business Transactions, Actions, and Process Steps -> Define Process Types and Assign Process Step Types*:

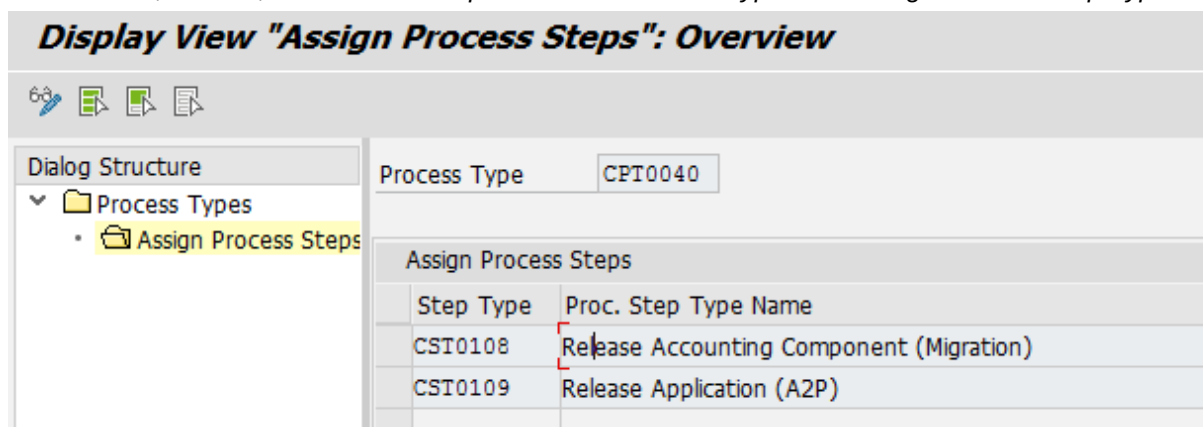


Figure: Define process types and assign process steps

CAUTION

Changes in this Customizing activity should only be made by very experienced FS-PM consultants. In particular, the reinsurance documents should not be released at this point but in a separate run subsequent to the migration. For more information, see Section 'Reinsurance System'.

In enhancement implementation /PM0/ABP_ABB_BADI_EI, you can find the BAdI implementations of BAdI /PM0/ABB_PROC_METHOD relevant here:

- BAdI implementation /PM0/ABP_ABB_APPR_MIG_AC for process method SPM0108 (*Release Accounting Component (Migration)*)
- BAdI implementation /PM0/ABP_ABB_APPR_APPL_BADI for process method SPM0045 (*Release Application (A2P)*)

7.4.8 Migration of Business Transactions

For information about the migration of policy and contract reversal, and migration of business transaction *Perform Premium Waiver (Life)*, see Sections 'Migration of Business Transactions: Reversal' and 'Migration of Business Transactions: Perform Premium Waiver (Life)'. The statements in this section do not apply to these two cases.

If you want to migrate business transactions or schedule them for later execution in FS-PM in the validation, you can achieve this by implementing BAdI /PM0/ABQ_POL_BT_X_MIG_BADI. You can use the sample implementation to ascertain how to implement the migration of the *Change Duration* business transaction.

Note the following when you implement BAdI /PM0/ABQ_POL_BT_X_MIG_BADI:

- You can only migrate background-enabled business transactions because, in the validation, scheduling is performed as a BTS date.
- The effective date of the migrated business transactions must be after the validity date of the last migrated policy version.
- The update in the validation must have been activated for this last policy version. For more information, see Section 'Scope of the Functional Validation'.

Example

Migration of business transactions in the context of retroactive changes

- Migration of policy versions without migration of business transactions

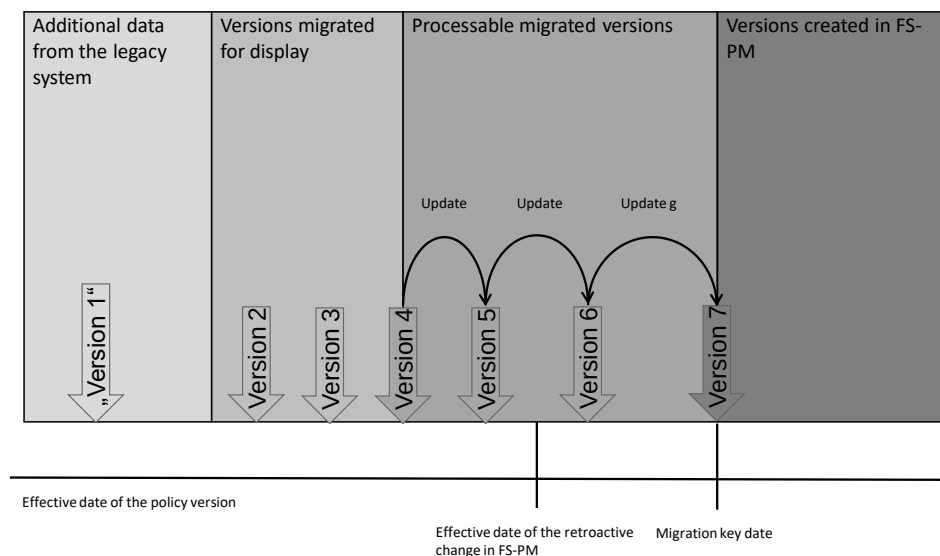


Figure: Retroactive change without migration of business transactions

The figure above shows the historic migration of a policy, where the versions designated by 4, 5, and 6 represent historical versions from the legacy system that have been migrated to FS-PM and are processable. Any policy versions that might be created by the update between these versions are not shown in the figure. The policy version designated with 7 has arisen in FS-PM by the update of the last migrated policy version. There has been no migration of business transactions here.

In the period from the validity date of the first processable migrated version, you can make retroactive changes in FS-PM. A possible effective date of such a retroactive change is depicted in the figure. If you now process the policy in FS-PM with this effective date, later versions are cancelled. An automatic reimplementation of the changes that lead to version 6 cannot be made since the data required for a reimplementation is not available. After making the retroactive change, therefore, the changes relating to version 6 must then be made manually by the processor.

- Migration of policy versions with migration of business transactions

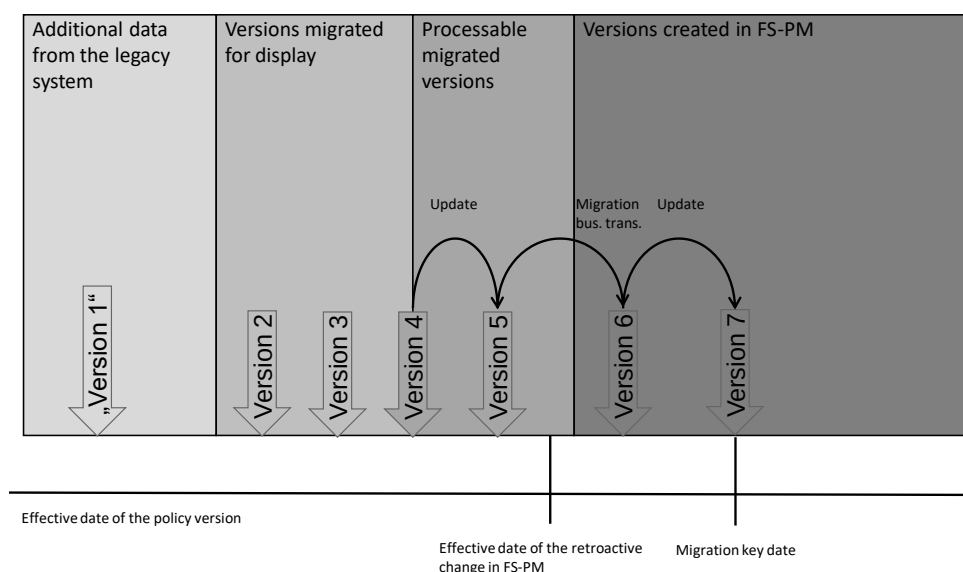


Figure: Retroactive change with migration of business transactions

The figure above shows the historic migration of a policy, where the versions designated by 4 and 5 represent historical versions from the legacy system that have been migrated to FS-PM and are processable. Any policy versions that might be created by the update between these versions are not shown in the figure. Version 6 also corresponds to a policy version in the historical period, but in this instance can be created by executing a business transaction in FS-PM. Version 6 in this example is not migrated from the legacy system, but created by migrating a business transaction to FS-PM. The corresponding business transaction is scheduled by implementing BAdI /PM0/ABQ_POL_BT_X_MIG_BADI during the validation of version 5, and executed in the update.

In the period from the validity date of the first processable migrated version, you can make retroactive changes in FS-PM. A possible effective date of such a retroactive change is depicted in the figure. If you now process the policy in FS-PM with this effective date, later versions are then cancelled. However, an automatic reimplementations of the business transaction that resulted in policy version 6 is now possible in most cases. After making the retroactive change, therefore, the changes relating to version 6 do not then need be made manually by the person processing.

7.4.9 Document Processing in Policy Migration

This section contains important information about what you need to consider during the policy migration for processing cash flow documents, non cash flow documents, and account statuses. You need to remember that, following a historic migration, where there is a retroactive change in FS-PM in the processable historical period, new documents are always created in FS-PM even for periods that have already been updated in the legacy system.

7.4.9.1 Treatment of Cash Flow Documents in Policy Migration

In your migration project, you can proceed in three different ways to handle cash flow documents:

- Variant 1: You supply none of the cash flow documents created by the legacy system with the migration.
- Variant 2: You supply cash flow documents created by the legacy system that have already been collected.
- Variant 3: You supply cash flow documents created by the legacy system that have not yet been collected.

If you decide on variant 1, you lose the option of comparing documents created by the legacy system with those created by FS-PM in the policy migration. For settlement periods that have already been collected before the key date, you have to suppress with variant 1 the transfer of the documents created by FS-PM during the policy migration to your collections/disbursements system.

With variants 2 and 3, the posting status of the delivered documents decides how these are processed in FS-PM in the validation of the technically migrated policies.

Details for variant 1

Documents are created as usual in FS-PM in the validation of the technically migrated policies. Without further measures, these are created so that they are transferred to the collections/disbursements system.

If a collection has already been made in the legacy system up to a particular point, but you do not deliver these documents with the migration, you have to ensure that, for the validation, the documents created up to this point are not transferred to your collections/disbursements system. Therefore, documents up to this date must have posting status *No Transfer* in FS-PM.

You achieve this by supplying a date with the migration. As cash flow documents are not further differentiated according to document categories, only one date is to be delivered here (`MIG_CF_END`). For this date, you have to migrate a date for each policy up to which document processing has already been completed in the legacy system. Up to this date, documents created in FS-PM in the validation are not transferred. The processing logic in FS-PM is that all cash flow documents in FS-PM created during the validation with (`Document.Accounting_from < Date_MIG_CF_END`) are not transferred to your collections/disbursements system.

For information about the migration of related dates, see Section 'Migration of Related Dates'.

In each case, the system ensures, irrespective of your implementation of the target date determining `BAdI /PM0/ABB_PROC_TARGETDATE`, that, in the validation, each migrated policy is updated to at least the latest related date defined in the above Customizing activity.

Details for variants 2 and 3

Here is a more detailed explanation of variants 2 and 3, using the example scenarios depicted in the following figure. A historic policy migration with delivery of documents is made, in which documents delivered up to (`Document.Accounting_from = DateMIG_CF_END`) have already been collected by the legacy system.

Example

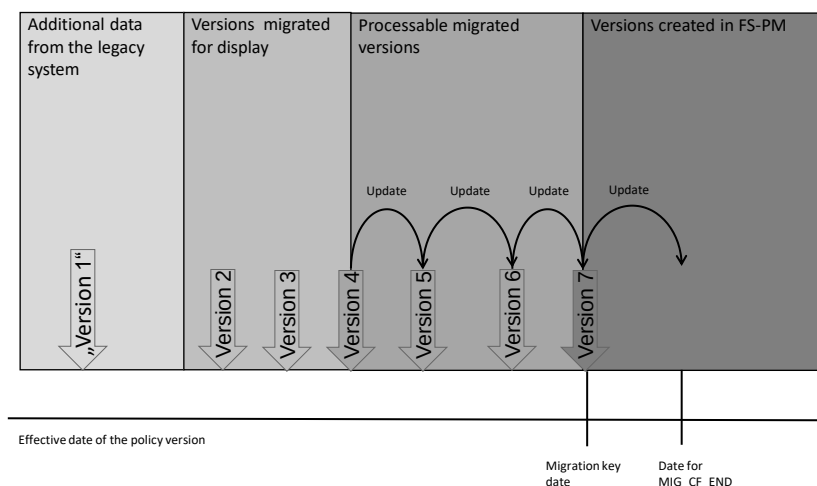


Figure: Example illustrating document migration

For each document delivered with the migration, offsetting documents are first created in FS-PM in the validation. The system also ensures that the migrated policy is updated to at least the migrated related date. During this update of the validity date of the first processable migrated policy version up to the related date, cash flow documents are created as usual by the accounting component, which can always be transferred to your collections/disbursements system.

By the migrated related date for `MIG_CF_END`, you make sure that the documents created for this policy in FS-PM by the accounting component with (*Document.Accounting_from* < *DateMIG_CF_END*) are not transferred, but documents with (*Document.Accounting_from* >= *DateMIG_CF_END*) are transferred to your collections/disbursements system.

Processing the documents created in FS-PM by the accounting component depends on whether these match the documents delivered with the migration or not. The system, therefore, compares all of the documents created in FS-PM during the validation with the corresponding documents delivered. Two documents are regarded as matching if they agree in certain field values. You can define which fields need to be considered for this comparison in Customizing for Policy Management under *Accounting Components -> Define Variants for Document Comparison*.

1. Case: Document delivered and document created in FS-PM match

If the comparison establishes that the document delivered (in the following figure represented by *Origin Q*, where Q stands for migration) and those created in FS-PM (indicated by *Origin A*) match, the offsetting document and that created in FS-PM are discarded. The delivered document is retained, but is given the journal number of the document created in FS-PM. In this way, the document delivered is assigned to processing in FS-PM and can be reset in case of a retroactive change, for example.

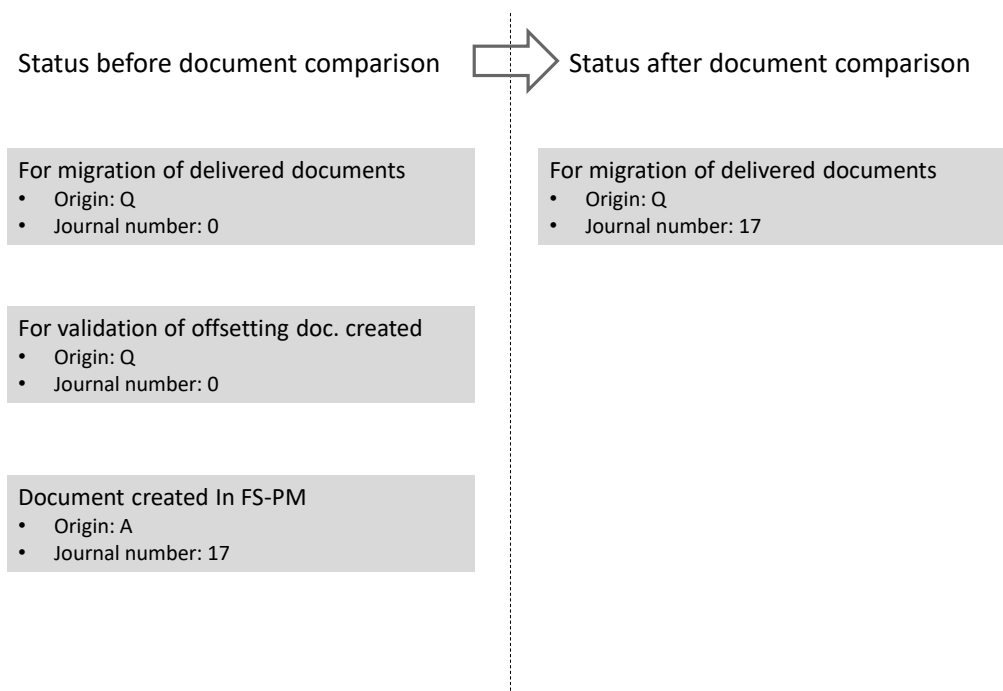


Figure: Delivered document and created document match

2. Case: Document delivered and document created in FS-PM do not match

In this instance, the delivered and offsetting documents are retained, where the offsetting document reverses the delivered document. Both are indicated by the origin Q (migration), and retain the journal number 0. This means that these two documents are not assigned to processing, and cannot be updated. The document created by FS-PM is likewise retained, and is given its regular journal number, and so is the basis for further processing, such as retroactive changes.

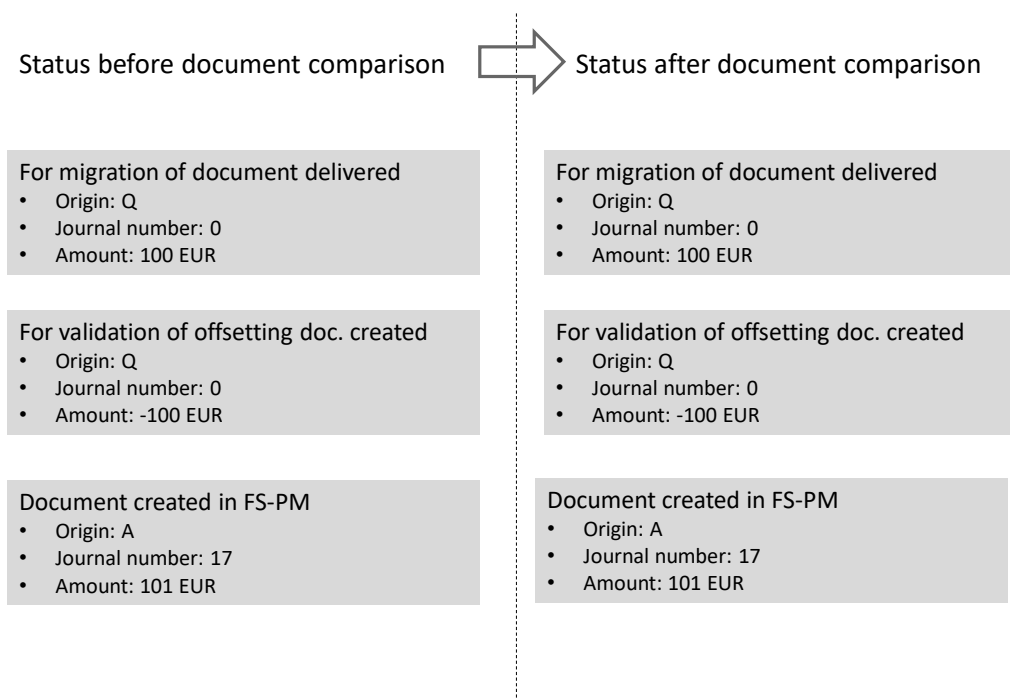


Figure: Delivered document and created document do not match

With both of these procedures, the validation ensures that, with later retroactive changes to a policy in FS-PM, the correct documents are reversed (meaning those with values that are correct in FS-PM). This works when, by delivering the date `MIG_CF_END`, you ensure that you update to at least the date up to which collection has already been made in the legacy system.

7.4.9.2 Treatment of Non Cash Flow (G/L) Documents in Policy Migration

In the same way as for the variants for the treatment of cash flow documents, you can deliver documents already transferred from the legacy system to your general ledger (G/L) system for the migration, and those already created by the legacy system but not yet transferred. The posting status of the delivered G/L documents is also decisive here for the further processing in FS-PM.

If you completely forego the delivery of non cash flow documents, you can still set the point from which the validation run documents receive the *Ready for Posting* status, by the delivery of a related date. Documents for the previous accounting period consequently receive the status *No Transfer*. Unlike cash flow documents, G/L documents are classified by various document categories.

You can, therefore, define a corresponding related date for each document category in Customizing for Policy Management under *General Settings -> Migration -> Assign Dates to Document Types*. For the dates defined here, you can migrate a related date for each policy up to which processing has already been completed in the legacy system. Up to this date, documents created in FS-PM in the validation are not transferred. The processing logic in FS-PM is that all documents of a document category created in FS-PM during the validation with `(Document.Accounting_from < RelatedDate_DocumentCategory)` are not transferred to your G/L system.

For each document category for which documents are delivered, you must likewise assign a related date to this document category. The system then ensures that the migrated policy is updated to at least the highest migrated related date. During this update of the validity date of the first processable migrated policy version up to the related date, non cash flow documents are created as usual by the accounting component that can always be transferred to your G/L system.

Delivered-related dates with a date ID that are assigned to a document category are evaluated in the accounting component in the course of the update in the validation. As already described, if a (new) creation of documents for the respective category is prevented up to the date of the related date (field `DATE_DT`), then existing delivered documents are merely assigned to the journal number of the current processing instead.

As for the cash flow documents, a comparison between documents delivered and documents created during the validation is made for the non cash documents, with a list of comparison attributes that you can configure, and the same case differentiation. The result of the comparison decides the further processing of the documents, as for the cash flow documents.

For the document categories *In-Force Business Transaction* and *Transaction Key Date*, there is no comparison between delivered documents and documents created by FS-PM in the validation. Delivered documents of these categories are **always** assigned to the respective migrated policy version. If new in-force business transactions or transaction key dates are created by calling the product engine, these represent discrepancies between the delivered data (for example, sum insured or premium), and the values calculated by the product engine. In the standard system for the migration method call of the product engine, these discrepancies are likewise saved as non cash flow documents.

7.4.9.3 Treatment of Account Balances in Policy Migration

If you want to migrate account balances, you must only deliver the account balances of the latest version. For retroactive changes in FS-PM, the account balances valid during earlier periods are calculated by evaluating the documents delivered or created in FS-PM.

7.4.10 Other Comparison Values when Comparing the Calculated Policy Version with Technically Migrated Policy Version

When you are migrating policies, it may be necessary to migrate other comparison values (for example, reserves or coverage capital) that have to be taken into account when calculating the migrated policies using the product engine. The functions for migrating policies support you here in the following two cases:

1. Case: Migration of values for comparison with values calculated in FS-PM

You can append the values to be compared to the migration structure `/PM0/ABQAADDDATA` using `APPEND` in the policies migration interface. You have to persist these values in a customer-specific table by implementing `BAdI /PM0/ABQ_POL_SAVE_BADI`. You can compare the delivered values with the values calculated by the product engine by implementing `BAdI /PM0/ABQ_POL_COMP_BADI`. Reading the values to be compared, and the comparison itself, has to be implemented in this `BAdI`.

2. Case: Migration of additional values for processing in the migration method of the product engine

If you want to process these additional values in your product engine content, then you have to append these values using `APPEND` in the policies migration interface to the migration structure `/PM0/ABQABASE`. Persisting these values and the transfer to the product engine is then automatic. All you need to do is to map these fields with the content of your product engine.

Structure `/PM0/ABQABASE` (entity `/PM0/ABDABASE`) is available in the contract model delivered by SAP at the contract model levels `/PM0/ABDAPOLPR` (contract), `/PM0/ABDACOV` (coverage), and `/PM0/ABDACOVCP` (subcoverage).

7.4.11 Handling of Entities to Be Deleted in the Release

For the functional validation of two historical policy versions that follow on from each other, in this section designated with version A or version B, you may find that an entity exists in positioned version A, but no longer in version B.

Possible reasons for this are:

- The entity was removed between policy versions A and B in the legacy system (for example, deletion of a clause or reversal of a coverage), however the policy version responsible for this deletion is not migrated. Consequently, the procedure described in Section 'Display of Policy Business Object and Its Versions in Migration File' does not apply with regard to the entity to be deleted.
- The entity is deleted during the functional validation of version A (for example, in the positioning but mostly due to the calculation by the product engine).

This means that an entity exists in the successfully validated version A that, correctly, no longer exists in version B.

In these cases, FS-PM must end the validity of such entities in the release of version A.

This occurs in the steps described below:

- Step 1
 - For all entities of version A for which the procedure described above is possible, the system checks whether they exist with the same key field values in the next migrated version B (see method `/PM0/CL_ABQ_BPP_POL_VAL_RUN->DELETE_OLD_ENTITIES`).
 - You can define the entities for which this check is to be made in Customizing for Policy Management under *General Settings -> Migration -> Define Entities for Policy Comparison*.
 - If such entities are found, then these are provisionally deleted from version A in the context of its release.
- Step 2
 - The provisional deletion of these entities is recorded in the journal with the processing ID and the effective date of version B.
 - The deleted entities are therefore valid in any event up to the validity date start of version B, even if further processing may result during the update of version A to version B.
- Step 3

The application is transferred to a contract version.
- Step 4
 - The journal entry from step 2 that causes the deletion is removed from the contract journal (see method `/PM0/IF_ABJ_0_POL_JOURNAL~POSTPROCESSING_MIGRATION`).
 - When loading the policy at the effective date of the validity of version B, the deleted entities are consequently not loaded.

If you require more detailed information for particular cases, see the Section 'Comparison of several policies pairs

' and the example described there.

7.4.12 Treatment of Versions Migrated for Display Only

You can obtain information on how to migrate historical policy versions for display in FS-PM only from Section 'Scope of the Functional Validation'.

For policy versions migrated for display only, you may only execute business transactions and actions in the *Inquiry* business process that do not result in the policy being processed.

To make sure of this for business transactions, in the *Inquiry* business process for policy versions migrated for display only, you can only execute these business transactions in the context menu for which the *Business Transaction for Display Only* indicator has been set in table `/PM0/ABUUBTX`.

Display View "Business Transaction Control Parameter": Overview

Business Transaction	Name of BusTrans	Process API Class	Backgrnd. ...	BTrans Data ...	OneT.(SAP)	Disp. Only	BackgrBT
P_B_S_COVPCO_INFO	play Subcoverage					☒	
P_B_S_COVPAC_CRT_EDIT	Create and Process Coverage Package	/PM0/CL_ABP_B...				☐	
P_B_S_COVPAC_EDIT	Process Coverage Package	/PM0/CL_ABP_B...				☐	
P_B_S_COVSHIFT	Shift Coverage Start Date	/PM0/CL_ABP_B...	/PM0/CL_AB...	/PM0/IF_ABP...		☐	☒
P_B_S_COV_BD_CRT	Create Coverage with Alternative Start	/PM0/CL_ABP_B...	/PM0/CL_AB...			☐	☒
P_B_S_COV_CRT	Create Coverage					☐	
P_B_S_COV_CRT_EDIT	Create and Process Coverage	/PM0/CL_ABP_B...				☐	
P_B_S_COV_DEL	Delete Coverage					☐	
P_B_S_COV_EDIT	Process Coverage	/PM0/CL_ABP_B...				☐	
P_B_S_COV_IFR_AMD	Change Adjustment	/PM0/CL_ABP_B...	/PM0/CL_AB...	/PM0/IF_ABP...	☒	☐	
P_B_S_COV_INFO	Display Coverage					☒	
P_B_S_COV_VDZ_CRT	Create Coverage with TCN					☐	
P_B_S_COV_VDZ_CRT_EDIT	Create and Process Coverage with TCN	/PM0/CL_ABP_B...				☐	
P_B_S_CPK_CRT	Create Coverage Package					☐	
P_B_S_CPK_DEL	Delete Coverage Package			/PM0/IF_ABP...		☐	
P_B_S_CPK_INFO	Display Coverage Package					☒	
P_B_S_CFP_ADD	Add Commission Participant					☐	
P_B_S_CFP_AMD	Change Commission Participant	/PM0/CL_ABP_B...	/PM0/CL_AB...	/PM0/IF_ABP...	☒	☐	

Figure: Overview of Business Transaction Control Parameter

To make sure of this for actions, actions in the *Inquiry* business process can only be executed for policy versions migrated for display only if they have been entered in system table /PM0/ABUADISPACT.

7.4.13 Correspondence

As the scheduling of correspondence is suppressed during the functional validation, you can proceed as follows if the successful migration is to be notified using correspondence:

1. Creation of a new ABAP object class as a subclass of /PM0/CL_ABZ_BPU_TMF7_MASTER:
To the extent that no special logic for the decision whether correspondence should be created is needed, the reimplementations of individual methods in the derived class is **not** necessary.
2. Creation of a new date and a new time model function in Customizing (*Define Dates and Time Model Functions* or SE43 -> /pm0/cust_int -> *Define Dates and Time Model Functions*):
 - a. The time model function should be of validity type FUNC7 (Single Correspondence)
 - b. Enter the ABAP object class created in step 1 as the implementing class.
3. Define an appropriate correspondence type using *Set FS-PM-Specific Correspondence Types*, taking into account the instructions in Customizing activity *Edit Basic Settings for General Correspondence*.
4. Assign the correspondence date created in step 2 with the correspondence type created in step 3 using Customizing activity *Automatically Schedule Correspondence -> Date Control*
5. Assign the correspondence date created in step 2 to the product module groups in dimension TIMEMODEL1 for those product module groups for whose contracts the successful migration is to be notified using the correspondence. You can make this setting with Customizing activity *Define Execution Criteria for Time Model Functions*.
6. Delivery of a correspondence date using the migration interface:
 - a. The date should be the date created in step 2.
 - b. The effective date of the date should be the date of the latest version delivered.

The result of carrying out these steps is that a correspondence date is scheduled for the migrated policies for the technical migration, but this date is not processed.

With the next correspondence run (meaning the update of the policies or contracts with an update type that carries out correspondence dates, for example CORR), this date is processed and the correspondence as defined in step 3 is scheduled.

7.4.14 Enhancement and Settings Options

In this section, you can find an overview of the most important Business Add-Ins (BAdIs) that you usually need to implement in the migration project for the validation.

You can find the BAdIs in Customizing for Policy Management under *General Settings -> Migration -> Business Add-Ins (BAdIs) -> Policies*. For more information about the BAdIs, see the documentation for the relevant activities in Customizing for Policy Management.

For more information about enhancement options that are relevant for the migration of data from entities not delivered in the SAP standard system, see Section 0.

7.4.14.1 BAdI: Functional Validation of Policies – Settings (/PM0/ABQ_POL_VAL_CNTRL_BADI)

By implementing this BAdI, you can define the values of the control parameters for the validation for each policy version to be validated.

You can also specify that temporary migration data following a successful validation or access to a business process is deleted by the system.

In your implementation of BAdI method

`/PM0/IF_EX_ABQ_POL_VAL_CNTRL~PERFORM_CREATE_RULE`, you can influence the execution of default settings. In the example implementation `/PM0/CL_EXM_ABQ_VAL_CNTRL_BADI` delivered, these default settings are executed for each entity that was not available in the previously migrated policy version. If you do not implement this BAdI method, the default settings are made for each entity in the validation.

Recommendation

SAP advises you to check in your migration project whether executing the default settings for each entity is necessary, and if possible to restrict it accordingly.

For additional information about using this BAdI, see Section ‘Scope of the Functional Validation’.

7.4.14.2 BAdI: Functional Validation of Policies – Error Handling (/PM0/ABQ_POL_ERR_HANDL_BADI)

By implementing the method `HANDLE_ERROR` of this BAdI, you can define if an error issued during the validation is actually relevant or if it can be ignored. You can regard the validation as successful despite certain functional errors, and set the migration status of the policies accordingly. A technical error, however, is always relevant. If the indicator `EF_NOERROR` is set even although there is a technical error, then it is ignored. If the indicator `EF_NORELEASE` is set, the release of the state with accepted functional errors will be suppressed, even if it was planned by run settings. The indicator `EF_NORELEASE` will be evaluated only if the indicator `EF_NOERROR` is set.

In addition, by implementing the method `HANDLE_MESSAGE` of this BAdI you can define whether each message should be displayed in the application log of the validation run or not.

If you do not implement this BAdI, a fallback implementation is called. You can also, for example, ascertain from this how you can log any errors.

For additional information about the use of this BAdI, see Sections ‘Migration Status’ and ‘Logging of the Validation Run’.

7.4.14.3 BAdI: Functional Validation of Policies – Calculation Using Product Engine (/PM0/ABQ_POL_CALC_BADI)

By implementing this filter BAdI (the filter value is the line of business), you can call the migration method of your product engine. The UBOI key for the contract is transferred, because a calculation is made for each contract.

The method `vtm_Migration` is called in the standard system implementation.

For additional information about using this BAdI, see Section ‘Scope of the Functional Validation’.

7.4.14.4 BAdI: Functional Validation of Policies – Comparison After Calculation (/PM0/ABQ_POL_COMP_BADI)

By implementing this BAdI you can compare a calculated copy of the technically migrated policy version with the original version, and change the copy if you need to. The UBOI key of the technically migrated policy version is transferred, and the copy.

Once the technically migrated policy version has been copied, and derivations and calculations made on the copy, you can compare the copy with the technically migrated version in this BAdI and make any necessary changes to the copy. A necessary change to the copy might be, for example, the creation of a surcharge or discount where the premium calculated by the product engine does not match the migrated premium value. The technically migrated version delivered is protected against processing, which means it cannot be changed in this BAdI.

After successful checks, the policy is issued in the following validation steps, at the legally binding and valid policy version in FS-PM.

You must implement all of your migration-specific checks and comparisons in this BAdI. In doing so, you can also make use of standard comparison methods, such as the journal comparison tool (see /PM0/IF_ABJ_JOURNAL_UBOI->COMPARE).

You can obtain the form in which this BAdI is to be implemented from example implementation /PM0/CL_ABQ_VALIDCOMP_EXAMPLE. When checking changes made to the business object, bear in mind that some entities are deleted and created again during the calculation by the product engine (see also Section 'Handling of Entities to Be Deleted in the Release'). This means you have to ensure that the deleted entity is compared with the newly created entity. These are differentiated in the value of the UBOI secondary key (/PM0/ABVA-<entity>), so that you have to use other criteria here to determine correctly the entities to be compared.

For additional information about using this BAdI, see Section 'Scope of the Functional Validation'.

7.4.14.5 BAdI: Functional Validation of Policies – Statistics (/PM0/ABQ_POL_STATISTIC_BADI)

By implementing this BAdI, you can store statistically interesting data from the validation run, for example in table /PM0/ABDQSTATIS.

SAP delivers the fallback implementation /PM0/CL_DEF_ABQ_POL_STATISTIC for this BAdI. This class implements the interface for the update service, meaning the data is located in the same SAP transaction management as the normal business object. No data is transferred since it is held by the BAdI instance itself.

7.4.14.6 BAdI: Functional Validation of Policies – Migration of Business Transactions (/PM0/ABQ_POL_BT_X_MIG_BADI)

By implementing this BAdI, you can migrate background-enabled business transactions of the *Change* business process, or schedule them to be performed in FS-PM with validity after the migration date.

You can use the example implementation /PM0/CL_POL_BT_X_MIG_SAMPLE to ascertain how to implement the migration of the *Change Duration* business transaction.

For more information about using this BAdI, see Section 'Migration of Business Transactions'.

7.5 Mass Deletion

The FS-PM migration interface provides you with the option of mass-deleting policies already migrated to FS-PM:

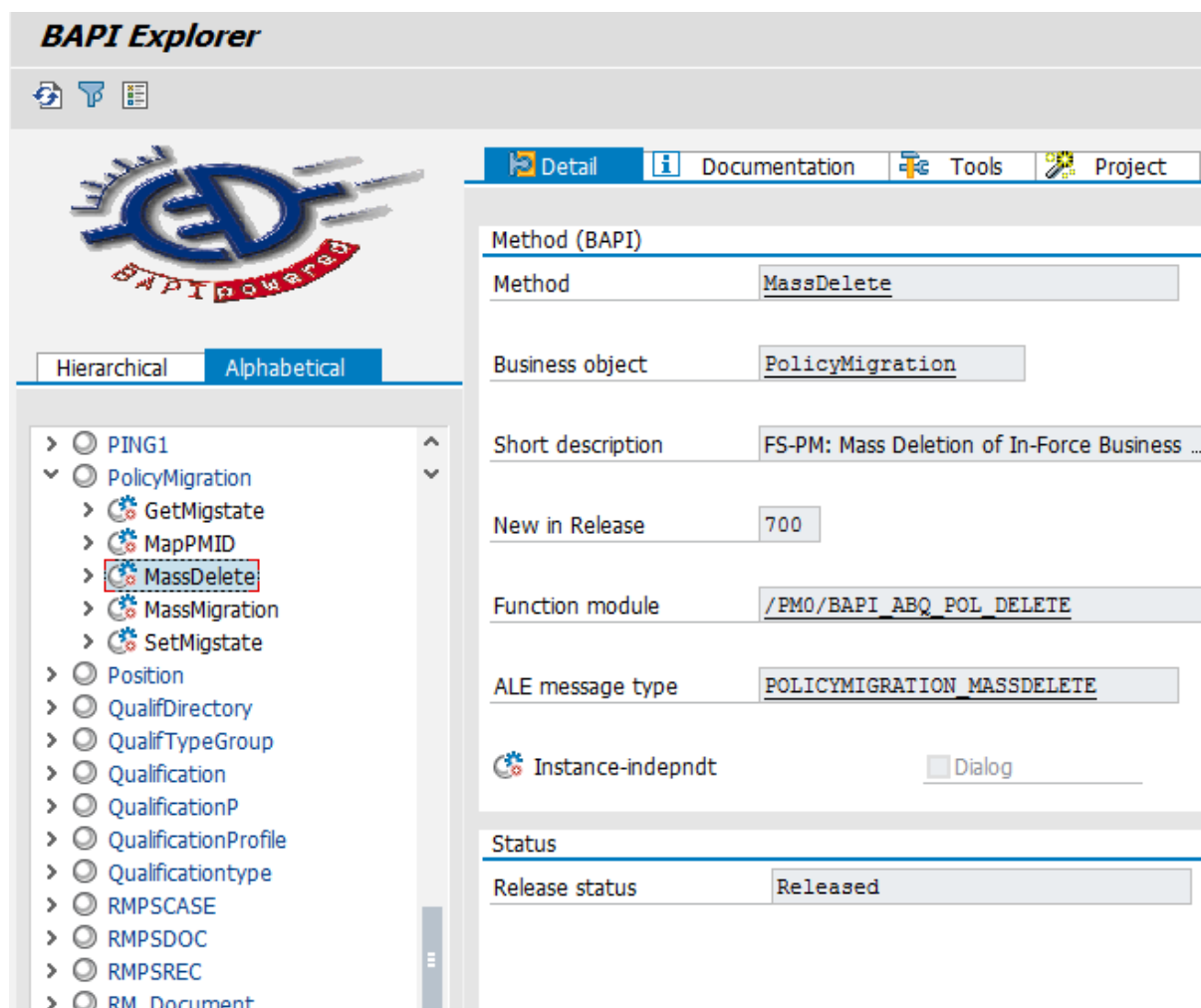


Figure: Mass deletion in the BAPI Explorer

You usually use this BAPI during the migration tests when you want to migrate large numbers of policies again, and have to delete migration policies from the previous migration run in FS-PM.

7.5.1 Enhancement and Settings Options

You can find the BAdIs in Customizing for Policy Management under *General Settings -> Migration -> Business Add-Ins (BAdIs) -> Policies*. For additional information about the BAdIs, see the documentation for the relevant activities in Customizing for Policy Management.

For more information about enhancement options that are relevant for the migration of data from entities not delivered in the SAP standard system, see Section ‘

Treatment of Add-On LoBs for the Migration of Policies’.

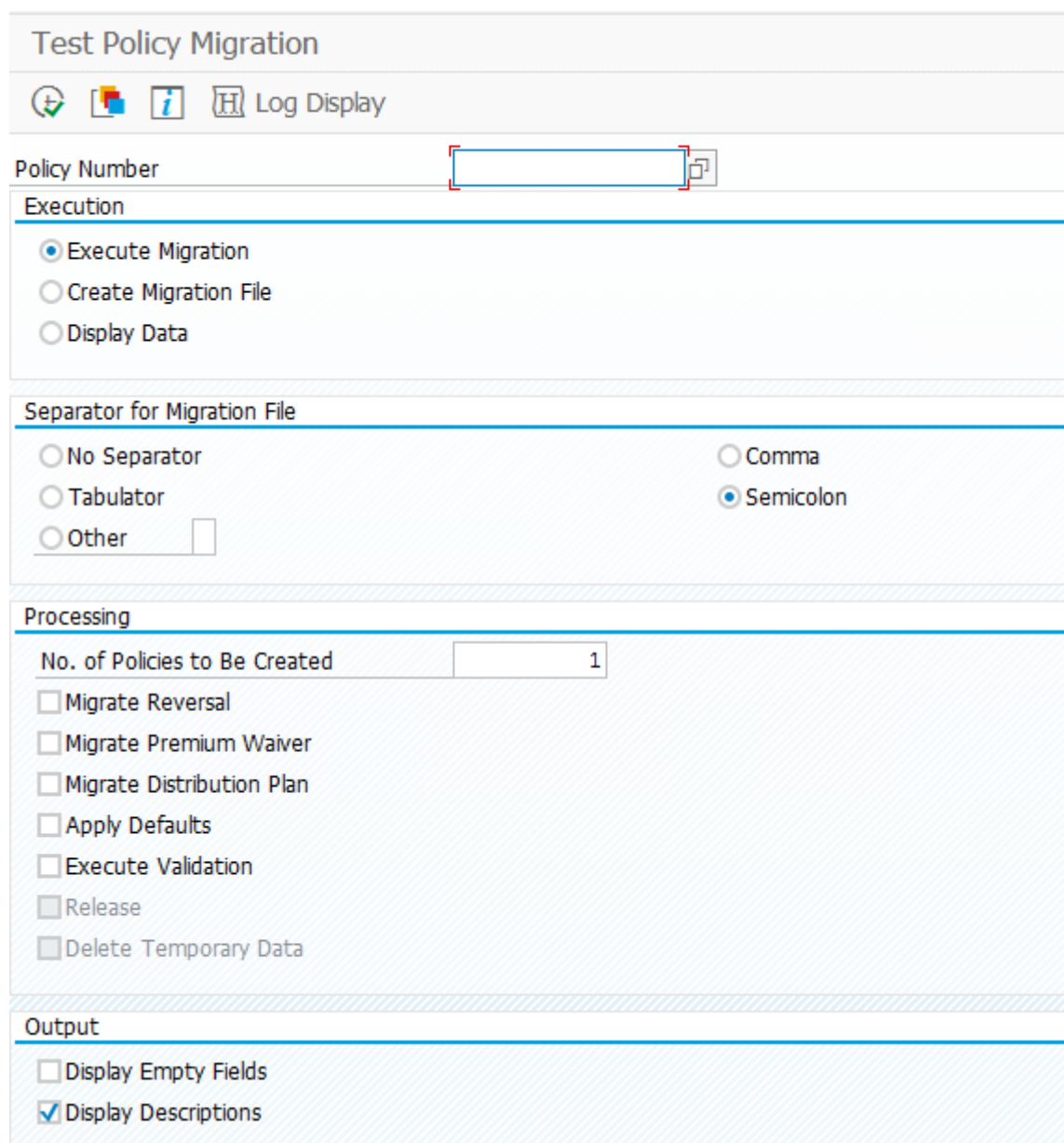
7.5.1.1 BAdI: Deletion of Policy Data (BAdI /PM0/ABQ_POL_DEL_BADI)

By implementing this BAdI, you can delete data from tables not delivered in the standard SAP system that belongs to the migrated policy versions.

By implementing this BAdI, you can also delete temporary data from tables not delivered in the standard SAP system that belongs to the migrated policy versions.

7.6 Testing the Migration of Policies

Transaction /PM0/ABQ_MIGTEST supports you in your migration project and the testing of the migration of policies:



The screenshot displays the 'Test Policy Migration' SAP transaction interface. At the top, there is a title bar 'Test Policy Migration' and a toolbar with icons for back, forward, help, and a 'Log Display' button. Below the toolbar, the 'Policy Number' field is highlighted with a red rectangle. The 'Execution' section contains three radio buttons: 'Execute Migration' (selected), 'Create Migration File', and 'Display Data'. The 'Separator for Migration File' section has four radio buttons: 'No Separator', 'Comma', 'Semicolon' (selected), and 'Tabulator'. There is also an 'Other' option with a small text input field. The 'Processing' section includes a 'No. of Policies to Be Created' field with the value '1' and a list of checkboxes: 'Migrate Reversal', 'Migrate Premium Waiver', 'Migrate Distribution Plan', 'Apply Defaults', 'Execute Validation', 'Release', and 'Delete Temporary Data'. The 'Output' section at the bottom has two checkboxes: 'Display Empty Fields' and 'Display Descriptions' (checked).

Figure: Testing the policy migration

This test tool provides you with the following capabilities:

- You can create migration files from existing policies in FS-PM. These can be very useful for you as templates for the development of conversion rules and data structures. You can define the separator to be used in the migration file.

You can test your implementation of the technical migration and the functional validation on different levels. There is additional support for testing in the form of the policy comparison described in Appendix H. The most important BADIs are discussed in the main part of this guide. For additional information about these BADIs, see the documentation for the relevant activities in Customizing for Policy Management.

- You can influence the process flow of the test program by the values in specific user parameters. By using these user parameters, you can, for example, migrate policy versions in the test tool before a certain validity date for display only in FS-PM.
- For more information about using this tool and the relevant user parameters, see the detailed program documentation in the system.

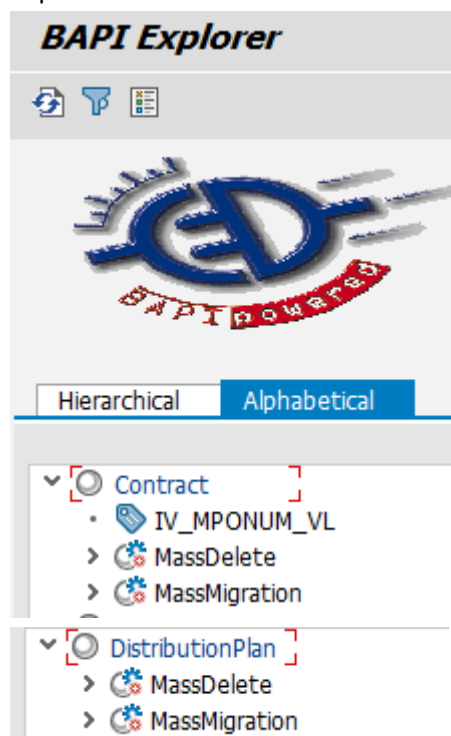
8 Details for the Migration of Other Business Objects to FS-PM

In this section, you can find information about the migration of data from the following business objects belonging to FS-PM:

- Insurable object
- Master contract
- Distribution plan
- Fund management data (funds, fund range, fund price, fund order)

You should check early in your project to determine which of these business objects you want to migrate, and in what order. You are recommended as a rule to migrate the business objects listed here first, and then the policies.

In the standard system, FS-PM has a migration interface for each of these business objects, implemented as a BAPI for the mass transfer and mass deletion of data.



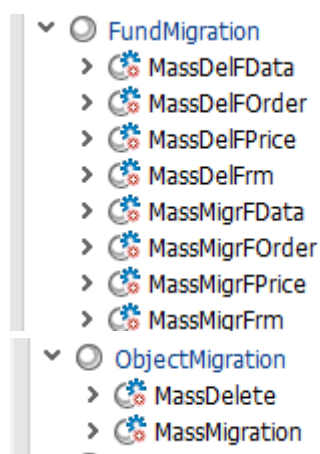


Figure: Migration of additional business objects in the BAPI Explorer

8.1 Migration of Insurable Objects (INBO)

Similar to the migration of policies, the migration of INBOs is made in two steps: first the technical migration, and then the functional validation.

The migration status of an INBO is persisted in the journal table of the INBO (table /PM0/ABDOIOJOUR) in a similar way to the migration of policies. The possible values of this status are the same as for the migration of policies.

If you migrate an INBO of the Declaration category, you also have to migrate the items contained in the declaration and their assignment to the declaration.

For the migration of INBOs, transaction *Test Migration of Insurable Objects* (transaction /PM0/ABQ_OBJ_MIGTEST) can support you.

8.1.1 Technical Migration

Similar to the migration of policies, a control structure (/PM0/ABQ_MIGRATION_OBJ) is available with which you can set parameters for the technical migration:

IF_LINK_VALIDATION: If you set this indicator, the validation is started automatically following the technical migration. You should only set the indicator in test mode.

IF_RELEASE: If you set this indicator, the release is started automatically following the validation. You should only set the indicator in test mode.

ID_MIGR_DATE: Date of the technical migration. The entries written to the FS-PM application database tables during migration use this date to set the time stamp.

You can make your own checks in the technical migration by implementing BAdI /PM0/ABQ_OBJM_TC_BADI. These are made before the technically migrated data is written to the FS-PM application tables.

8.1.2 Migration of Object-Extending Data

If you want to migrate object-extending data (tables /PM0/ABDOINSOBJ1/2/3 for the insurable object and /PM0/ABDASUBJCT1/2/3 for the insured object), simply append the extension fields in the migration file to the relevant migration structures (/PM0/ABQINSOBJ or /PM0/ABQASUBJCT). The functions delivered for the technical migration then persist the data by checking the fields have identical names in the correct FS-PM tables. Functional Validation

With the functional validation of INBOs (transaction /PM0/ABQ_FP_Q_OBJVAL), you transfer the technically migrated data of the INBO in a checked, released version recorded in the journal. The validation run for an INBO is integrated in Framework for Parallel Processing (FPP):

Figure: Validation of migrated insurable objects

The selection criteria for the INBO to be validated is used as described in Section 'Transaction for Functional Validation' for policies. The collective search help for the *Insurable Object* field, however, is not restricted to migrated insurable objects. If you require this, you have to enhance the collective search help with an appropriate elementary search help.

The intervals of the insurable objects to be validated are created again for each validation run (dynamic interval creation). You can set the size of the intervals in the *Number of Entries per Interval* field. All migrated insurable objects with the migration status: *Migrated; Not Checked, Migrated; With Errors, Migrated; Accepted Despite Errors; Not Released, Migrated; Checked Successfully; Not Released; Migrated; Checked Successfully; For Display* are taken into account. The migration and validation of insurable objects use the same migration values as the migration and validation of policies.

To display the log of a validation run of migrated insurable objects, there are similar options as described in Section 'Logging of the Validation Run'.

For more information, see [SAP Policy Management](#) -> *Application Help* -> *In-Force Business Management* -> *Migration in SAP Policy Management* -> *Validate Migrated Insurable Objects*.

An INBO is validated in the following basic steps:

- Loading the technically migrated INBO
- Executing the business rules maintained at events `INIT` and `EXECUTE`

Display View "Events for Business Rules": Overview of Selected Set

Rule ID	Model Name	Name of Process Node	Bus. Proc.	Business P...	Entity	Event	Event Name	Time Long Text
1030003001					/PM0/ABDOINSOBJ	8	INITIALIZE	Load Entity (BO Rules)
1030007001					/PM0/ABDOINSOBJ	9	CHANGED	Change Entity (BO Rules)
1030010100					/PM0/ABDOINSOBJ	11	EXECUTE	During a BTrans (Process Rules)

Figure: Events for business rules

- Data of the validated insurable object is written to the database tables.

When validating insurable objects, you have the following enhancement options:

- By implementing `BAdI /PM0/ABQ_OBJM_VALIDATE_VR_BADI`, you can define additional checks that are made following the checks delivered.
- By implementing `BAdI /PM0/ABQ_OBJM_ERR_HANDL_BADI`, you can classify error messages as not applicable. This enables you to use the migration status *Migrated; Accepted Despite Errors*.

- By implementing BAdI `/PM0/ABQ_OBJM_STATISTIC_BADI`, you can store the data occurring during the validation of the INBO that may be of interest for a statistical valuation.

For more information about these BAdIs, see the documentation for the relevant activities in Customizing for Policy Management.

Unlike the migration of policies, you do not need to delete temporary files from the system following the functional validation of insurable objects since there is no update, meaning no further insurable objects versions can occur. Insurable object versions are only also created in FS-PM when you migrate delivered insurable object versions.

If you are dealing with an insurable object of the declaration object type, you must specify both the insurable object number of the declaration and the insurable object number(s) of the item(s) in the selection dialog. The validation of a declaration does not automatically entail a validation of its item(s).

8.1.3 Mass Deletion of Insurable Objects

The migration interface enables the mass deletion of insurable objects already migrated to FS-PM.

Using import parameter `IF_DEL_RELEASED`, you can control whether insurable objects already released may also be deleted.

By implementing BAdI `/PM0/ABQ_OBJM_DEL_BADI`, you can also delete data from tables not delivered in the SAP standard system.

8.1.4 Testing the Migration of Insurable Objects

Transaction `/PM0/ABQ_OBJ_MIGTEST` supports you in the migration of insurable objects. You can use this transaction in much the same way as the transaction for testing the migration of policies described in Section 7.6 .

For more information, see [SAP Policy Management](#) -> *Application Help* -> *In-Force Business Management* -> *Migration in SAP Policy Management* -> *Test Migration of Insurable Objects*.

8.2 Master Policy

FS-PM provides you with migration interfaces for the technical migration and the mass deletion of master policies. Only master policies that have not been released are deleted during a mass deletion.

By entering values in migration control structure `/PM0/ABQ_MIGRATION_OBJ`, you can set parameters for the technical migration, for example, carry out the validation directly after it. There is no separate validation transaction available for master policies.

The BAdIs for enhancing the migration of master policies are structured in the same way as the BAdIs for enhancing the migration of insurable objects. For more information about these BAdIs, see Customizing for Policy Management under *General Settings* -> *Migration* -> *Business Add-Ins (BAdIs)* -> *Master Policies*.

8.3 Distribution Plan

Since an actual distribution plan must always be assigned to a contract element (policy, contract, coverage), it only makes sense to migrate distribution plans if the policies used are also migrated. The distribution plan relationships (structure `/PM0/ABQADISTREL`), which show the reference from the policy to the assigned distribution plans, are contained in the migration interface for the migration of policies.

Similar to the migration of policies, the migration of distribution plans is made in two steps: first the technical migration, and then the functional validation.

The migration status of a distribution plan is persisted in the journal table of the distribution plan (table `/PM0/ABDDJOUR`) in a similar way to the migration of policies. The possible values of this status are the same as for the migration of policies.

The migration of distribution plans is supported by a migration test tool (transaction `/PM0/ABQ_MIGCOINS`).

8.3.1 Technical Migration

Similar to the migration of policies, a control structure (`/PM0/ABQ_MIGRATION_OBJ`) is available with which you can set parameters for the technical migration:

IF_LINK_VALIDATION: If you set this indicator, the validation is started automatically following the technical migration. You should only set the indicator in test mode.

IF_RELEASE: If you set this indicator, the release is started automatically following the validation. You should only set the indicator in test mode.

IF_PREASSIGN: If you set this indicator and also transfer a distribution plan template number when you migrate the distribution plan, the migrated distribution plan is populated with data from the template by default. In this instance, the distribution plan template must already be available in the target system.

ID_MIGR_DATE: Date of the technical migration. The entries written to the FS-PM application database tables during migration use this date to set the time stamp.

You can make your own checks in the technical migration by implementing BAdI `/PM0/ABQ_COINS_TECHCHECK_BADI`. These are made before the technically migrated data is written to the FS-PM application tables.

By implementing BAdIs `/PM0/ABQ_COINS_ENH_ENT_BADI` and `/PM0/ABQ_COINS_SAVE_BADI`, you can also migrate entities of the *Distribution Plan* business object that are not delivered in the SAP standard system. This works in the same way as for the entities that do not belong to SAP for the *Policy* business object, as described in Section 0.

8.3.2 Functional Validation

With the functional validation of distribution plans (transaction `/PM0/ABQ_FP_Q_COIVAL`), you transfer the technically migrated data of distribution plans in a checked, released version recorded in the journal. The validation run for distribution plans is integrated in Framework for Parallel Processing (FPP):

Figure: Validate migrated distribution plans

The selection criteria for the distribution plans to be validated is used as described in Section 'Transaction for Functional Validation'. The collective search help for the *Distribution Plan Number* field, however, is not restricted to migrated distribution plans. If you require this, you have to enhance the collective search help with an appropriate elementary search help.

The intervals of the distribution plans to be validated are created again for each validation run (dynamic interval creation). You can set the size of the intervals in the *Number of Entries per Interval* field. All migrated distribution plans with the migration status: *Migrated; Not Checked, Migrated; With Errors, Migrated; Accepted Despite Errors; Not Released, Migrated; Checked Successfully; Not Released; Migrated; Checked Successfully; For Display* are taken into account. The migration and validation of distribution plans use the same migration values as the migration and validation of policies.

To display the log of a validation run of migrated distribution plans, the options are similar to those described in Section 'Logging of the Validation Run'.

For more information, see [SAP Policy Management](#) -> *Application Help* -> *In-Force Business Management* -> *Migration in SAP Policy Management* -> *Validate Migrated Distribution Plans*.

A distribution plan is validated in the following basic steps:

- Loading the technically migrated distribution plan
- Executing the business rules maintained at event INIT
- Executing checks when saving a new distribution plan
- Writing data of the validated distribution plan to the database tables

When validating distribution plans, you have the following enhancement options:

- By implementing BAdI /PM0/ABQ_COINS_VALID_VR_BADI, you can define additional checks that are made following the checks delivered.
- By implementing BAdI /PM0/ABQ_COINS_ERR_HANDL_BADI, you can classify error messages as not applicable. This enables you to use the migration status *Migrated; Accepted Despite Errors*.
- By implementing BAdI /PM0/ABQ_COINS_STATISTIC_BADI, you can store the data occurring during the validation of distribution plans that may be of interest for a statistical valuation.
- By implementing BAdI /PM0/ABQ_COINS_STR_CHK_BADI, you can prevent the processing of a distribution plan that has not yet been completely migrated or that contains errors.

For more information about these BAdIs, see the documentation for the relevant activities in Customizing for Policy Management.

Unlike the migration of policies, you do not need to delete temporary files from the system following the functional validation of distribution plans since there is no update, meaning no further distribution plan versions can occur.

8.3.3 Mass Deletion of Distribution Plans

The migration interface enables the mass deletion of distribution plans already migrated to FS-PM.

By implementing BAdI /PM0/ABQ_COINS_DEL_BADI, you can also delete data from tables not delivered in the SAP standard system.

8.3.4 Testing the Migration of Distribution Plans

Transaction /PM0/ABQ_MIGCOINS supports you in the migration of distribution plans. You can use this transaction in much the same way as the transaction for testing the migration of policies described in Section 'Testing the Migration of Policies'.

The test tool for the migration of policies (transaction /PM0/ABQ_MIGTEST) also supports you in the migration of distribution plans. If you set the *Including Distribution Plans* indicator in this test tool, the distribution plans assigned to the policy, if there are any, are likewise migrated.

For more information, see [SAP Policy Management](#) -> *Application Help* -> *In-Force Business Management* -> *Migration in SAP Policy Management* -> *Test Migration of Distribution Plans*.

8.4 Fund Management Data

FS-PM provides you with migration interfaces for the technical migration and the mass deletion of funds, fund ranges, fund prices, and fund orders.

By entering values in migration control structure /PM0/ALQ_MIGRATION_FUND, you can set parameters for the technical migration, for example, to carry out a validation directly after it. A validation transaction for business objects from funds management, however, is not available in the SAP standard system. If you want to execute your own validation transaction in parallel, you can use the transaction /PM0/3FCA_BACKG available in PBT. This guide does not include a description of how to use this transaction since SAP does not consider this of practical relevance to this use case.

The BADIs for enhancing the migration of funds management data are structured in the same way as the BADIs for enhancing the migration of insurable objects. For more information about these BADIs, see Customizing for Policy Management under *General Settings -> Migration -> Business Add-Ins (BADIs) -> Fund Management*.

8.5 Special Risk Notifications

If you want to migrate special risk notifications, you can do this when you migrate policies. Structures /PM0/BAPI_ABQAMNR (for special risk notifications submitted in the past) and /PM0/BAPI_ABQAIMPMNR (for external special risk notifications) are part of the interface for migrating policies.

However, the related entities /PM0/ABQAMNR and /PM0/ABQAIMPMNR are not part of the *Policy* business object. The special risk notifications delivered are, therefore, written to these database tables directly without further checks.

8.6 Miscellaneous

At the time of publication of this guide, the persistent entities of FS-PM in-force business management listed below are not supported by the migration interfaces delivered in the standard system. SAP does not claim that the list is complete or fully up-to-date.

Above all, the list contains data that is only created for the productive operation with FS-PM, and therefore it should not be relevant for migration. For the same reason, the migration of sample applications is not supported in the standard system.

Business Object	Entity	Short Description
FS_PM_CORRESPONDENCE	/PM0/ABDCCORR	Correspondence
FS_PM_CORRESPONDENCE	/PM0/ABDCDOCU	Documents
FS_PM_CORRESPONDENCE	/PM0/ABDCATTLIST	Attachment list
FS_PM_CORRESPONDENCE	/PM0/ABDCDOCSECT	Document sections
FS_PM_CORRESPONDENCE	/PM0/ABDCEXEMPL	Copy specifications
FS_PM_CORRESPONDENCE	/PM0/ABDCUSERTE	Free texts
FS_PM_CORRESPONDENCE	/PM0/ABDCUSERTM	Free text modules
FS_PM_CORRESPONDENCE	/PM0/ABDCUSERVAR	Free variables
FS_PM_CORRESPONDENCE	/PM0/ABDCATTLIST	Attachment list
FS_PM_EMPLOYER	/PM0/ABDMEMPLYR	Third-party collections
FS_PM_FUNDS	/PM0/ALDLFPRPR	Fund price
FS_PM_FUNDS_PALETTE	/PM0/ALDKPALLEG	Fund range data include
FS_PM_GBL_ODR	/PM0/ALDLGBLODR	Aggregating fund order
FS_PM_JOURNAL_BO	/PM0/ABDAJOURNBO	Journal BO root node
FS_PM_JOURNAL_BO	/PM0/ABDADUALCEI	Duel control check: error table
FS_PM_JOURNAL_BO	/PM0/ABDADUALCH	Duel control check: header data
FS_PM_JOURNAL_BO	/PM0/ABDARENEWAL	Renewal data
FS_PM_MASTER_AGREEMENT	/PM0/ABDGPAT	Sample application
FS_PM_POLICY_MANAGEMENT	/PM0/ABDACHAPAY	Charges payer
FS_PM_POLICY_MANAGEMENT	/PM0/ABDAMACOST	Manual cost posting
FS_PM_POLICY_MANAGEMENT	/PM0/ABDAMASTLAM	Manual settlement amount

FS_PM_POLICY_MANAGEMENT	/PM0/ABDAMACCTAX	Manual posting tax determination
FS_PM_POLICY_MANAGEMENT	/PM0/ABDACOMDIFF	Different commission participant
FS_PM_POLICY_MANAGEMENT	/PM0/ABDADEVAPPL	Application discrepancy
FS_PM_POLICY_MANAGEMENT	/PM0/ABDAWFREMCO	Application discrepancy
FS_PM_POLICY_MANAGEMENT	/PM0/ALDAPROJEC	Model calculation
FS_PM_POLICY_MANAGEMENT	/PM0/ABDAANSWER	Answers risk assessment
FS_PM_POLICY_MANAGEMENT	/PM0/ABDASBINC	Answers risk assessment
FS_PM_POLICY_MANAGEMENT	/PM0/ABDAUWDCN	Underwriting decision risk assessment
FS_PM_POLICY_MANAGEMENT	/PM0/ALDAFPERF	Performance rate
FS_PM_POLICY_MANAGEMENT	/PM0/ALDAPROJFUN	Fund distribution simulation
FS_PM_POLICY_MANAGEMENT	/PM0/ALDAPROJTAX	Model calculation tax
FS_PM_TRANSFER	/PM0/ABDSTRANS	Total transfer (root)
FS_PM_TRANSFER	/PM0/ABDSTAXPAYR	Taxable person
FS_PM_TRANSFER	/PM0/ABDSSETTLE	Accounting

9 Integration of Operational System Data in an FS-PM Migration Project

This section contains information about integrating operational systems in in-force data migration. Every migration project has its own procedures, meaning a guide such as this has to focus on more general aspects and recommendations.

You have to differentiate between two cases:

- In a Big-Bang migration, operational system data is migrated both to applications in SAP for Insurance (such as FS-CD or FS-ICM or to the central business partner), or to non-SAP applications.
- Operational system data is not migrated.

Tip

If you intend to migrate operational systems in your project, SAP recommends that you complete the migration of operational systems before you migrate policies. SAP recommends that you always switch off the operational systems during the productive migration, and if necessary catch up later in a program run following the migration.

Where an operational system is migrated either before the policy migration or not at all, you have to supply the key fields that create the link from the in-force business data to the operational system data. This requires you to have to forgo the automatic number assignment for data, such as the policy number or the number of the insured object in FS-CD, when you migrate policies.

A pre-migration is generally not feasible if you are migrating your reinsurance system. For more information, see Section 'Reinsurance System'.

The procedure for the pre-migration of operational system data is less susceptible to errors. You can validate the migration of operational systems before the policy migration, and switch off the interfaces to them during the productive migration of policies. In this way, you can achieve a significant reduction in runtime for the validation and release of the migrated policies, and ensure that good quality productive data in the operational systems is no longer changed during the migration of policies.

If you switch off interfaces for operational systems during the migration of policies, you should also indicate any messages resulting from calling these interfaces as not relevant, and not display them in the validation run application log. For more information, see Section 'Logging of the Validation Run'.

If you update migrated policies at a point after the policy version valid on the migration date, then it follows that the operational systems must be supplied again by FS-PM. This is the reason SAP recommends that you perform the update in a separate program run subsequent to the migration.

9.1 Operational Systems in General

Operational systems that are usually used in a comprehensive insurance solution, in addition to the in-force business management system, are:

- Business partner system
- Collections/disbursements system
- Commission system
- Reinsurance system
- Claims management system
- Document management system
- General ledger system

As mentioned above, you should have completely set up or migrated the operational systems used, whether SAP or non-SAP, before the migration. The values for FS-PM key fields that refer to this operational system data are to be delivered in the migration files. In the table below, you can find an overview of the particular key fields that you must take into account for the migration of policies. SAP does not claim that the list is complete or fully up-to-date.

In case of a migration to the SAP Incentive and Commission Management system FS-ICM/IFBA or to the SAP Reinsurance system, there are some particular features you need to note that are described in the following sections.

Migration Structure	Field Name	Field Description	Operational System
Representative /PM0/BAPI_ABQAAGENT	PARTNER_ID	Business Partner	Business Partner
	ADDRESSNUMBER_ID	Address	Business Partner
Beneficiary /PM0/BAPI_ABQABNFCRY	PARTNER_ID	Business Partner	Business Partner
Commission /PM0/BAPI_ABQACOMMIS	PARTICIPROLE_ID	Participation Role	Commission System/In-Force Business Range
	BSNESSPARTNER_ID	Business Partner	Business Partner
	COMMCONTRNR_ID	Commission Contract Number	Commission System
	COMCONTGRP_ID	Person Responsible	Commission System
	POL_NR_NN	Policy Number	Commission System/In-Force Business Range
Commission Participant Role /PM0/BAPI_ABQACOMROLE	COMMISROLE_CD	Commission Agent Role	Commission System/In-Force Business Range
	BUSI_BEGIN_NN	Valid for Business From	Commission System/In-Force Business Range
	BUSI_END_NN	Valid for Business From	Commission System/In-Force Business Range
	TECH_BEGIN_NN	Technically Valid From	Commission System/In-Force Business Range
	TECH_END_NN	Tech. Valid To	Commission System/In-Force Business Range
Creditor /PM0/BAPI_ABQACRDITOR	PARTNER_ID	Business Partner	Business Partner
	ADDRESSNUMBER_ID	Address	Business Partner
Related Contract /PM0/BAPI_ABQAEXTPOL	PARTNER_ID	Business Partner	Business Partner
	ADDRESSNUMBER_ID	Address	Business Partner
Policyholder /PM0/BAPI_ABQAPOLHLDR	PARTNER_ID	Business Partner	Business Partner
	ADDRESSNUMBER_ID	Address	Business Partner
Contract Data /PM0/BAPI_ABQAPOLPR	INSOJECT_ID	Insurance Object	Collections/Disbursements

	BUSOBJ_TYPE_NN	Business object type that triggers a commission	Commission System/In-Force Business Range
	BUSOBJ_ID_NN	Name of Business Object Category	Commission System/In-Force Business Range
	BUSOBJ_VERS_NN	Master Data or Customizing Object version	Commission System/In-Force Business Range
	REM_BUSCASE_TYP_NN	Remunerating Business Transaction in Operational System	Commission System/In-Force Business Range
	REM_BUSCASE_ID_NN	Name of Business Transaction Category	Commission System/In-Force Business Range
Premium Payer /PM0/BAPI_ABQAPRMP AYR	PARTNER_ID	Business Partner	Business Partner
	ADDRESSNUMBER_ID	Address	Business Partner
	BANKACC_ID	Bank Details	Business Partner
	PAYMENTCARD_ID	Payment Card Number	Business Partner
	INCPAYMETHOD_CD	Incoming Payment Method	Collections/Disbursements
	PAYMACC_CD	Collection Account	Collections/Disbursements
	PAYMPART_CD	Payment Partner for Collection Account	Collections/Disbursements
	INSOBJ_CD	Insurance Object	Collections/Disbursements
	BROKERINSOBJ_ID	Broker Insurance Object	Collections/Disbursements
	BROKERBP_ID	Broker BP Number	Business Partner
Related Contract /PM0/BAPI_ABQAREX TPOL	PARTNER_ID	Business Partner	Business Partner
Insured Object Address Assignment /PM0/BAPI_ABQASBAR	ADR_CD	Insurable Object Address	INBO Address Management
Insured Object /PM0/BAPI_ABQASUB JCT	PARTNER_ID	Business Partner	Business Partner
	INSOBJNR_TT	Insurable Object	INBO
	JOB_CD	Occupation	Business Partner
	ADDRESSNUMBER_ID	Address	Business Partner or INBO Address Management
	GENDER_CD	Gender	Business Partner

Special Deposit Payer /PM0/BAPI_ALQAADD PRPY	PARTNER_ID	Business Partner	Business Partner
	ADDRESSNUMBER_ID	Address	Business Partner
	BANKACC_ID	Bank Details	Business Partner
	PAYMENTCARD_ID	Payment Card Number	Business Partner
	PAYMPART_CD	Payment Partner for Collection Account	Collections/Disbursements
	INSOBJ_CD	Insurance Object	Collections/Disbursements
	INCPAYMETHOD_CD	Incoming Payment Method	Collections/Disbursements
Fund Assignment /PM0/BAPI_ALQAASF UND	FNDPARENT_ID	Fund Range	Fund Management
	INTNAME_TT	Fund Number	Fund Management
	PALETTEINT_ID	Fund Range	Fund Management
Coverage (Life) /PM0/BAPI_ALQACOV	FP_ID	Fund Range	Fund Management
Fund Allocation /PM0/BAPI_ALQAFUN DALL	INTNAME_TT	Fund Number	Fund Management
Interest Payer for Policy Loan /PM0/BAPI_ALQAINTPAY	PARTNER_ID	Business Partner	Business Partner
	ADDRESSNUMBER_ID	Address	Business Partner
	BANKACC_ID	Bank Details	Business Partner
	PAYMENTCARD_ID	Payment Card Number	Business Partner
	INCPAYMETHOD_CD	Incoming Payment Method	Collections/Disbursements
Policy Loan /PM0/BAPI_ALQALOA N	INSOBJECT_ID	Insurance Object	Collections/Disbursements
Payment Recipient for Policy Loan /PM0/BAPI_ALQAPAY MREC	PARTNER_ID	Business Partner	Business Partner
	ADDRESSNUMBER_ID	Address	Business Partner
	BANKACC_ID	Bank Details	Business Partner
	PAYMENTCARD_ID	Payment Card Number	Business Partner
	OUTGOINGPAY_CD	Outgoing Payment Method	Collections/Disbursements
Policy Loan Payer /PM0/BAPI_ALQAPOL OPAY	PARTNER_ID	Business Partner	Business Partner

	ADDRESSNUMBER_ID	Address	Business Partner
	BANKACC_ID	Bank Details	Business Partner
	PAYMENTCARD_ID	Payment Card Number	Business Partner
	INCPAYMETHOD_CD	Incoming Payment Method	Collections/Disbursements
Premium Deposit /PM0/BAPI_ALQAPRE MDEP	PARTNER_ID	Business Partner	Business Partner
	ADDRESSNUMBER_ID	Address	Business Partner
	BANKACC_ID	Bank Details	Business Partner
	PAYMENTCARD_ID	Payment Card Number	Business Partner
	ACCINTKEY_ID	Unique Interest Key	Collections/Disbursements
	INSOBJNO_TT	Insurance Object Number	Collections/Disbursements
	OUTGOINGPAY_CD	Outgoing Payment Method	Collections/Disbursements
ShiftSwitch /PM0/BAPI_ALQASSW ITCH	SOURCEFUND_ID	Source Fund	Fund Management
	TARGETFUND_ID	Target Fund	Fund Management
	PALETTEINT_ID	Fund Range	Fund Management
	FNDPARENT_ID	Fund Range	Fund Management
Special Risk Inventory /PM0/BAPI_ABQAIMP MNR	GENDER_CD	Gender	Business Partner
Submitted Special Risk Messages /PM0/BAPI_ABQAMNR	PARTNER_ID	Business Partner	Business Partner
	GENDER_CD	Gender	Business Partner
Subobject Commission /PM0/BAPI_ABQTICM MPBO	TRIOBJ_ID_EXT_NN	External Identification of Triggering Subobject	Commission System/In-Force Business Range
	TRI_OBJ_TYPE_NN	Subobject Type of Business Object	Commission System/In-Force Business Range
Cashflow Header /PM0/BAPI_ABQNCAS HFLH	DOCNRCID_ID	Document Number for Contract Current Account	Collections/Disbursements
	DOWNPAYREF_ID	Down Payment Reference	Collections/Disbursements
	BUSINESSTRA_ID	Business Case Number	Collections/Disbursements
Cashflow Header Item /PM0/BAPI_ABQNCASH FLI	ADDACCASSIGN_TT	Account Assignment String	Collections/Disbursements

	POSTINGTARGET_ID	Posting Target	Collections/Disbursements
	BP_ID	Business Partner	Business Partner
	ADDRNUM_ID	Address Number	Business Partner
	PYMET_KK	Payment Method	Collections/Disbursements
	BANKACC_ID	Bank Details	Business Partner
	CREDITCARDNUM_ID	Payment Card Number	Business Partner
	DIFFPARTNER_ID	Alternative Partner for Payment	Business Partner
	DIFFADDRESSNR_ID	Alternative Address Number for Payment	Business Partner
	DIFFBANKDET_ID	Alternative Bank Details for Payment	Business Partner
	DIFFPOSTTARG_ID	Different Posting Target	Collections/Disbursements
	DIFFBUPARTNER_ID	Alternative Business Partner for Posting Target	Business Partner
Non-Cashflow Item /PM0/BAPI_ABQNNOC ASHI	ADDACCASSIGN_TT	Account Assignment String	Collections/Disbursements
Policy Data /PM0/BAPI_ABQAPOL ICY	BUSOBJ_TYPE_NN	Business object type that triggers a commission	Commission System/In-Force Business Range
	BUSOBJ_ID_NN	Name of Business Object Category	Commission System/In-Force Business Range
	BUSOBJ_VERS_NN	Master Data or Customizing Object version	Commission System/In-Force Business Range
	REM_BUSCASE_TYP_NN	Remunerating Business Transaction in Operational System	Commission System/In-Force Business Range
	REM_BUSCASE_ID_NN	Name of Business Transaction Category	Commission System/In-Force Business Range
	MPOLICY_NR	Master Policy Number	Master Policy Management
Withholding Tax /PM0/BAPI_ABQAWHD TAX	BP_ID	Business Partner	Business Partner
PFO OM Interface /PM0/BAPI_ABQTPFO OM	BUSI_BEGIN	Valid for Business From	
	BUSI_END	Valid for Business From	
	REALO	Extended Object ID	
	ROLE	ID for a Role in PFO	

Cashflow Account Balance /PM0/BAPI_ABQNNOC ASHB	INTNAME_TT	Fund Number	Fund Management
Different Commission /PM0/BAPI_ABQAVAR COMM	COMMCONTRNR_ID	Commission Contract Number	Commission System/In- Force Business Range
	ROLE_ID	Participant Role	Commission System/In- Force Business Range
	REMUNERATION_CD	Remuneration Type	Commission System/In- Force Business Range
	TRIOBJTYPE_ID	Subobject Type of Business Object	Commission System/In- Force Business Range
	TRIOBJIDEXT_ID	External Identification of Triggering Subobject	Commission System/In- Force Business Range

9.2 Commission System/Portfolio Assignment

If you migrate your commission data to FS-ICM/PFO, you have to take the following into account if you have switched the portfolio assignment (PFO) of FS-ICM to active in FS-PM. You activate the portfolio assignment of ICM in Customizing for Policy Management under *Integration -> Portfolio Assignment PFO -> Define Interface to Portfolio Assignment*.

If you have activated portfolio assignment (PFO), mapping table /PM0/ABDTPFOJNO is filled by migration class /PM0/CL_ABP_DAU_MIGRATION during the migration of policies.

The central data field of mapping table /PM0/ABDTPFOJNO is field POL_NR_NN. The functional policy number from the legacy system is transferred in this during the migration in migration structure /PM0/ABQACOMMIS. An entry is written in the mapping table for each commission participant delivered with the policy version and their participant roles. This enables you to create the link to commission cases and contracts that have already been created in the legacy system under the old policy number, during the life cycle of a policy in FS-PM.

If you provide values in the import structure /PM0/BAPI_ABQTPFOOM, when calling the function module /PM0/ABQ_POL_MASSMIGR new PFO entries will be created in the PFO system for the migrated policies. If no data is provided in this structure for a policy to be migrated, SAP assumes that the migrated policy shall refer to existing PFO data for the policy. If you followed SAP's recommendation and set up or migrated the PFO data before the migration of policies is done, there should be no need to provide data for portfolio assignment in /PM0/BAPI_ABQTPFOOM when migrating policies. See also SAP Note [2301222](#).

9.3 Reinsurance System

If you also migrate your reinsurance system, it is very important to agree with the user department in detail the rules to be applied to reinsure the products to be migrated:

- Is cumulation required? If so, is it also required for all policies?
- Which rules should be used to cumulate (in the interface of the standard delivery of FS-RI, cumulation is according to the risk group number)?
- Do you need to determine a facultative reinsurance cover for migrated contracts?
- What are the rules for facultative reinsurance cover in your reinsurance system?



Tip

SAP recommends that you set up your reinsurance system well before the planned policy migration. If necessary, adapt the implementation of the reinsurance interface and

check in an FS-PM test system in dialog mode whether your reinsurance system is correctly supplied by FS-PM and the reinsurance is correctly determined. You should not use the reinsurance system during the migration of policies. Instead, you should supply the reinsurance system in an update run subsequent to the policy migration.

Even if you do not cumulate or use the risk group number for the cumulation, you should still deliver a risk group number in the migration files, in order to be able to use the SAP standard functions to their best advantage. Clarify with the user department how best to assign the risk group numbers.

 CAUTION

Only once you are sure that there is to be no cumulation for the migrated policies should you perform the mass update run to update the migrated policies in parallel. If there is cumulation, the reinsurance interface can only be supplied in parallel if this is done according to the risk group number. If there is any chance of cumulation, the update run must not be performed in parallel (all policies in an interval) in order to avoid the problem of the cumulation lock.

10 Treatment of Add-On LoBs for the Migration of Policies

10.1 Enhancing the Migration Structures for the Migration of Data from Entities Not Delivered in the SAP Standard System

With regard to the EXTENSIONIN mechanism discussed in Section 'Enhancement and Settings Options', you can also migrate to FS-PM entities or structures (though only flat data structures) that are not delivered in the SAP standard system.

CAUTION

New entities and siblings must only be implemented by SAP development partners. You can supply data from entities not delivered in the SAP standard system in the migration file in the same way as data from entities belonging to SAP. This works for both partner entities and partner siblings in the same way. Proceed as described below.

Prerequisites:

- Migration structures for partner entities are available and meet the following requirements:
- Name: /[Namespace]/A[LoB abbreviations]QA* (Example: /NSP/AMDAPOLPR)
- Field 1: BO_ID Type /PM0/ABD_BO_ID
 Field 2: PARENT_ID Type /PM0/ABD_PARENT_ID
 Field 3: VERSION_ID Type /PM0/ABD_MIGVERSION_ID
 Field 4: TODELETE_FG Type /PM0/ABQ_TODELETE
- This is followed by the data fields of the entity.

Steps for the migration of partner entities:

The steps are described on the basis of performing the technical migration using the SAP standard migration tool LSMW, to make it easier to follow. In the ETL migration tool you use, the details for the individual steps to be performed will vary.

1. Preparation of the LSMW migration project

Make the following adjustments to load the Extension-In parameters with the LSMW.

Section *Maintaining Source Structures*: Create source structure EXTENSIONIN (source structures are the data structures with which you load the data delivered from the legacy system to the ETL migration tool).

- Section *Maintaining Source Fields*: Add the following fields to the EXTENSIONIN source structure:
 - TYPE C(030) Record type: identifying field content: BAPIPAREX
 - STRUCTURE C(030) Structure name of the BAPI Table Extension
 - VALUEPART1 C(240) Data part for the BAPI extension parameter
 - VALUEPART2 C(240) Data part for the BAPI extension parameter
 - VALUEPART3 C(240) Data part for the BAPI extension parameter
 - VALUEPART4 C(240) Data part for the BAPI extension parameter
- Section *Maintaining Structural Relationships*: Link the source structure EXTENSIONIN to BAPI structure E1BPPAREX that is delivered in the SAP standard system (in your ETL migration tool, you execute this when you map your source structures to your target structures).

2. Creating the migration file

The migration file is created and it contains both partner and standard entities. You can use transaction `/PM0/ABQ_MIGTEST` to create a sample file.

3. Conversion of the migration file

The migration file partner entities are not recognized by the LSMW interface and have to be converted to generic Extension-In parameters. You can achieve this using transaction `/PM0/ABQ_EXT_CONV`. All of the files of a specified folder are converted. The converted file is stored in the same folder with the extension `_converted`.

Partner entities are converted to the structure BAPIPAREX in this step. The fields of BAPIPAREX are filled as follows:

- **STRUCTURE:** Name of migration structure
- **VALUEPART1:** Concatenated fields of key structure `/PM0/ABQ_XKEY`
- **VALUEPART2:** Concatenated fields of migration structure (max. 240 characters)
- **VALUEPART3:** Concatenated fields of migration structure (max. 240 characters)
- **VALUEPART4:** Concatenated fields of migration structure (max. 240 characters)

If the migration structure is longer than 720 characters, the fields of the structure do not fit into one line. A second line is then created for this object. The `COUNTER_ID` field of the key structure `/PM0/ABQ_XKEY` is increased by 1; `VALUEPART2-4` contain the other characters of the migration data.

4. Starting the LSMW with converted structure

Load the converted structure with the LSMW. The standard implementations of BAdIs `/PM0/ABQ_POL_ENH_ENT_BADI` and `/PM0/ABQ_POL_SAVE_BADI` create the partner entities in FS-PM, and save the corresponding migrated data to the FS-PM tables.

The LSMW always loads the entire migration file to the main memory. If there are large migration files, this may result in memory problems. In such cases, SAP recommends you to create several smaller migration files instead of one very large file.

In Appendix C - Details for the Technical Migration of Data from Entities Not Delivered in the SAP Standard System

you can find further details for processing data during the technical migration from entities that are not delivered in the SAP standard system.

10.2 Enhancement of the Migration Level of Functions

You can implement the following BAdIs if you need to perform a technical migration of data from entities not delivered in the SAP standard system, which is different from the standard implementation.

- **BAdI `/PM0/ABQ_POL_ENH_ENT_BADI`**

BAdI for the migration of entities that are not delivered in the SAP standard system.

A standard implementation is provided. You can use class `/PM0/CL_EX_ABQ_POL_ENH_ENT` as an example, should you have to create your own implementation.

When the transferred data records are being processed, this BAdI is called after each processing, because, theoretically, an entity that does not belong to SAP can be added under each entity.

- **BAdI `/PM0/ABQ_POL_SAVE_BADI`**

BAdI for storing data in tables that are not delivered in the SAP standard system.

No standard implementation is provided.

A reference to the migration object is transferred (reading; in the standard instance of migration class `/PM0/CL_ABQ_DAU_MIGRATION`) and, as changing parameter, the data of table parameter `ABQAADDDATA`. This means that, in this BAdI, you can persist data of your own entities from `ABQAADDDATA` in your own tables and also, for example, provide your own status tracking table.

- **BAdI `/PM0/ABQ_POL_DEL_BADI`**

This BAdI is required if you also need to include your own entities during the mass deletion of migrated policies and temporary data.

For more information about these BAdIs, see the documentation for the relevant activities in Customizing for Policy Management.

11 Performance Aspects

You would expect the functional validation to require a much longer runtime as compared to the technical migration. In response to this, SAP has created the following measures to enable you to speed up the validation:

- As described in detail in Section 'Scope of the Functional Validation', you can configure the scope of the validation. With the appropriate data quality for the technically migrated policies, you can switch off individual validation functions, thereby positively influencing the runtime.
- The transactions for the functional validation have been integrated in Framework for Parallel Processing (FPP). With the appropriate system setup, you can make good use of performing the validation in parallel.
- As described in Section 'Integration of Operational System Data in an FS-PM Migration Project', you should set up your project as far as practicable so the interfaces to the operational systems can be switched off during the migration.

With a historical migration, the validation runtime required increases in proportion to the number of processable policy versions to be migrated to FS-PM, and the length of the period of the historical migration. You should take this into consideration when planning the migration. For reasons of performance, SAP also recommends that, in the validation, you do not update the migrated policies using the policy version valid at the migration date, but in a subsequent update run in FS-PM.

Recommendation

SAP recommends that you measure the performance in a migration test system early on, enabling you to identify project and customer-specific runtime requirements. To do so, you can use the performance tools available in the ABAP Workbench.

Tip

In case of a high parallelization when running the migration programs, the buffer size for the number range objects may become a bottleneck that increases the overall runtime. It may be worth testing if a significant increase of the relevant buffer sizes (e.g. for /PM0/ABN31 and /PM0/ABN32) during the migration runs can improve performance. See also SAP Note [2299651](#).

12 Recommendations

This section contains useful suggestions intended to assist you in planning and carrying out your migration project.

12.1 Check Lists

SAP recommends that you clarify the following key points and general conditions of the migration project with the user department concerned and IT as early as possible:

- Define the planned migration date, taking into consideration any accounting deadlines or periods with an expected increased claims volume.
- Agree about the quantity structure to be migrated, and how long productive operation can be suspended due to the migration.
- Clarify whether the complete in-force business can be divided into packages that could be migrated separately, for example on following weekends, thus enabling a gradual migration.
- Define a roadmap for the integration of the operational systems involved (such as collections/disbursements, commission, business partners, reinsurance), and also the further FS-PM business objects to be migrated (insurable object, master policy, fund management, distribution plan) in the migration of policies.
- Clarify whether and how the interfaces to operational systems can be switched off during the migration of policies.
- Define a migration scenario, in particular whether and to what extent you want to perform a historical migration.
- Clarify whether and to what extent you want to migrate documents already created in the legacy system.
- Clarify to what extent the migration of business transactions can be used or should be used.
- Clarify whether the FS-PM target system will already be in operational production on the migration date.
- Clarify in good time whether mass processing, such as execution of adjustments pending in the period of the planned migration, should be carried out in the legacy system or in FS-PM.
- Define a planned deadline when a validated FS-PM test system should be available (if necessary, with customer or partner development), to test the migration procedure and make ongoing improvements to it.
- Agree with the user department about the acceptance criteria for the migration test, so that a program can be created that can automatically test the most important acceptance criteria (for instance, a check that the premium total of a migrated policy tranche in the legacy system matches the one in FS-PM).

Parallel to these activities, you are advised to tackle the topics of extracting the legacy data and assuring the quality of the legacy data, at the earliest stage and with the appropriate support and involvement of the user departments. SAP can support you in this with tools designed for this purpose. If you consider doing this, you should contact your SAP partner.

Once you have clarified the points above, you should be able to estimate which quantity structures, with which functional scope, and in which period your migration project can take place. This enables you to formulate requirements for performance or sizing of the systems concerned.

You are now able to define the requirements of the upcoming migration project in greater detail. FS-PM provides numerous enhancement and configuration options for the migration of policies and the other FS-PM business objects. You have to decide which of these you need to use. To make such decisions, you must clarify the following:

- Which customer-specific data fields or data from entities not delivered in the SAP standard system are to be included for the technical migration (see Section 'Enhancement and Settings Options' and 'Treatment of Add-On LoBs for the Migration of Policies')?
- Which persistent data is delivered in the migration file, and which should be calculated first, either in FS-PM by derivations or calculation by the product engine?

Recommendation

SAP recommends that the data is delivered in the migration file.

- What should the check in the technical migration achieve (see Section 'Scope of Technical Migration')?
- What should the functional validation achieve (see Section 'Scope of the Functional Validation')? Can you take suitable measures to switch off individual validation functions?
- Which error messages from the validation are irrelevant to your migration project, for example, because you have switched off operational systems (see Section 'Logging of the Validation Run')?
- What should the comparison of the calculated policy with the technically validated version achieve (see Section 'Enhancement and Settings Options')?
- Which reference policies should be migrated, and when, so that later the policy comparison can be used for regression tests? The most important BADIs are discussed in the main part of this guide. For additional information about these BADIs, see the documentation for the relevant activities in Customizing for Policy Management.
- Which related dates are to be migrated for which document categories (see Section 'Migration of Related Dates')?
- If you want to compare documents, which criteria should be used as the basis for this (see Section 'Document Processing in Policy Migration')?
- How do you want to proceed if policies are migrated with errors?

Recommendation

SAP advises you to delete the policies migrated with errors and then migrate them again (see Section 'Mass Deletion').

- Which business transactions apart from Reversal and Premium Waiver are to be migrated (see Section 7.4.8)? Is it possible to map policy versions by the migration of business transactions, since, in this case, the versions in FS-PM can be reimplemented?

12.2 Recommended Procedure

This section includes recommendations from SAP regarding the general procedure for the migration of policies to FS-PM. Since the actual requirements and options are very closely linked to the particular project, only general recommendations are made..

As mentioned in Section 12.1, the general conditions should be agreed with the user departments at the earliest opportunity. Similarly, the user departments should be involved in the creation of a test and verification environment. A migration project should always be set up so that:

- The operational systems are not supplied during the migration
- As much persistent FS-PM data structures data as possible are delivered in the migration file.
- For a historical migration, the number of policy versions to be migrated is restricted to the level of functions that are actually required.
- A migration test system, where you can also measure performance, is available in time.

If you envisage a massive migration project where the runtime is critical, you can also check whether a gradual migration is possible. In this case, the policies are migrated to the target system in a period before the migration date, so that, on the migration key date, only those policies that have been processed in the legacy system in the meantime are migrated to FS-PM.

The figure below illustrates the procedure recommended by SAP:

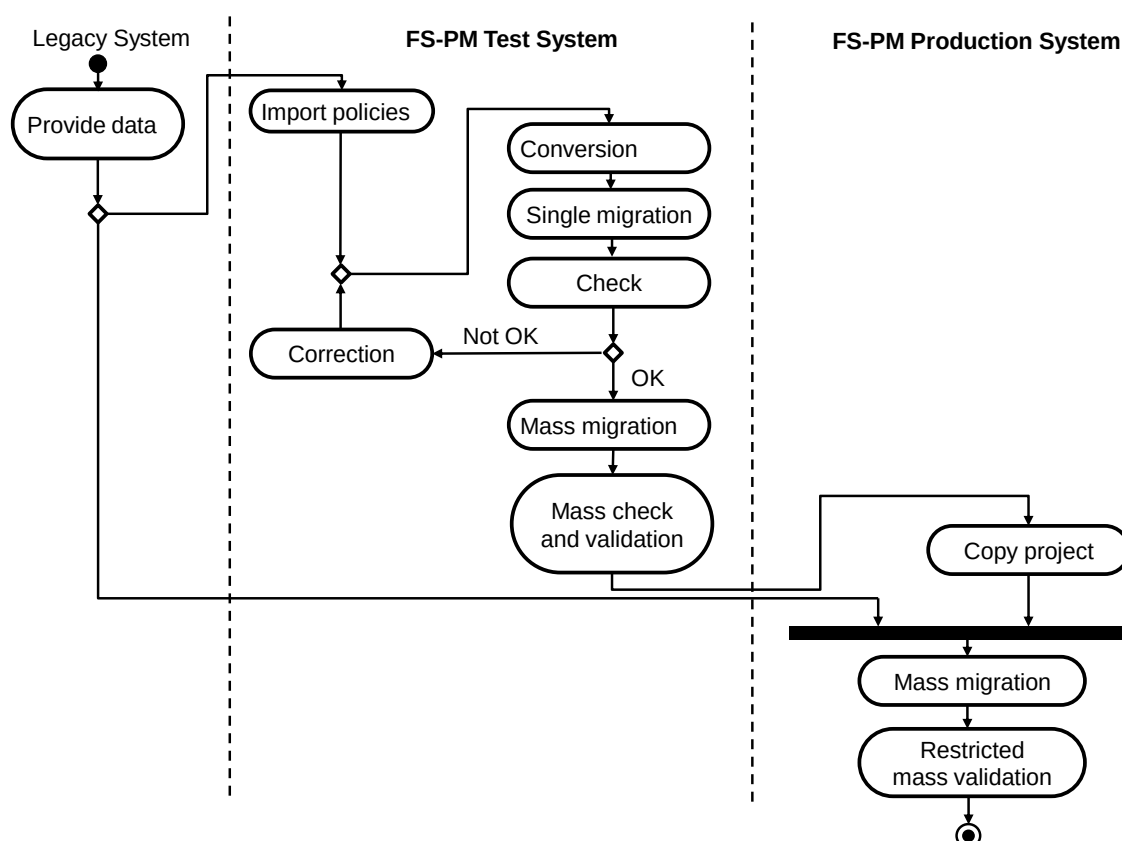


Figure: Recommended procedure for the migration

Prerequisites:

- The quality of the legacy data is ensured.
- The data conversion rules have been implemented in the migration tool, for example, LSMW.
- Any customer developments have been validated and transported to the migration test system.
- Customizing of the test system is complete.
- All migration-relevant BAdIs have been implemented, in particular, those defining which validation errors in your migration project are to be viewed as relevant.

A broad outline of the recommended procedure:

1. A small representative portfolio of policies from the legacy system is migrated to the test system, and validated.
2. In addition to the migrated policies being checked by experts, for example using the *Inquiry* business process that can also be executed on technically migrated policies, a regular check should be made of the migrated partial portfolio as agreed with the user department, if possible in automatic form.
3. For the verification by the user department, the FS-PM business processes should be carried out on the policies migrated for the test. It is very important to consider the integration aspect here: Are the documents created correctly and the interfaces supplied correctly?

 Note

Integration scenarios should definitely be included in the test activities as soon as possible.

4. Making corrections (structure/content of the legacy data, data conversion rules, implementation of migration-relevant BAdIs) and migrating the sample portfolio again, until the technical migration and functional validation deliver a satisfactory result for the project phase reached.

Migration of reference policies, in order to be able to analyze the effects of future developments or corrections later with the policy comparison. (The most important BADs are discussed in the main part of this guide. For additional information about these BADs, see the documentation for the relevant activities in Customizing for Policy Management.

5. Runtime analysis for migration or validation of a policy portfolio (> approximately 100 policies); you can make policy templates for this runtime analysis, for example, with the test tool for the migration of policies (see Section 'Testing the Migration of Policies').
6. Analysis of the runtime measurement, particularly regarding possible customer-specific runtime requirements, and carrying out measures to improve this.
7. Migrating a representative part portfolio again, and, if necessary, making corrections until verification/acceptance by the user department.
8. Performing the mass migration in package sizes best suited to the specific project, while retaining all necessary validation options (see Section 'Scope of the Functional Validation').
9. In case of errors, delete the migrated policies, make corrections, and migrate the policy packages concerned again; repeat this step if necessary.
10. Acceptance test of the migration project by the user department.
11. Migration of reference policies for later migration regression tests.
12. Mass migration to the FS-PM productive system, with interfaces to operational systems switched off. Since a complete validation of the portfolio to be migrated has already been made in the test system, for performance reasons you can consider whether you want to make only a restricted validation, or a complete validation of just a part of the portfolio, or both.
13. Following acceptance of the migration, perform a subsequent update of the policies migrated.
14. After making/importing critical corrections, new migration of the reference policies, and check using the policy comparison whether the migration procedure has not been invalidated. If various aspects of the implementation of the migration project have to be adapted, a reasonable number of these adaptations must be tested.

13 Appendix

Appendix A - LSMW

To implement the ETL process, you can use the migration tool LSMW that is freely available in the SAP standard system (transaction LSMW).

The LSMW enables you to:

- Import legacy data
- Convert the data to FS-PM data structures
- Import the data to the FS-PM application tables

The LSMW always loads the entire migration file to the main memory. If there are large migration files, this may result in memory problems. In such cases, SAP recommends you to create several smaller migration files instead of one very large file.

For more information about using the LSMW, see the Application Help for SAP NetWeaver.

Appendix B – Example for the Migration of Data from Entities Not Delivered in the SAP Standard System Using Transaction /PM0/ABQ_EXT_CONV

Example of a data conversion using the fictitious entity /NSP/AHDAEXREF.

Source structure /NSP/AHQAEXREF with the following fields:

Field	Type	Value
BO_ID	/PM0/ABD_BO_ID	00499
PARENT_ID	/PM0/ABD_PARENT_ID	00003
VERSION_ID	/PM0/ABD_MIGVERSION_ID	00001
TODELETE_FG	/PM0/ABQ_TODELETE	SPACE
REFNOM_CD	/PM0/ABD_REFCOMPANY_ID	001
REFNUMBER_TT	/PM0/ABD_REFNUMBER_TT	123456789012345678901234567890
REFTYP_CD	/PM0/ABD_REFTYPE_ID	001
PM_ID	/PM0/ABD_MODELPM_ID	EXREF0000000
NOTE1	CHAR255	aaaaaaaa...(256x)
NOTE2	CHAR255	bbbbbbbb...(256x)
NOTE3	CHAR255	cccccccc...(256x)
NOTE4	CHAR255	dddddddd...(256x)

Following step 2, the following line is created in the migration file:

[illegible]

Figure: Line in the migration file following step 2

With the data conversion in step 3, the following line is created (if specifying the semicolon as separator):

[illegible]

Migration structure

Key structure /PM0/ABO XKEY, the last two characters are line counters of the objects

VALUEPART2, concatenated fields of the migration structure

VALUEPART3, concatenated fields of the migration structure

VALUEPART4, concatenated fields of the migration structure

Figure: Line in the migration file following data conversion in step 3

Appendix C - Details for the Technical Migration of Data from Entities Not Delivered in the SAP Standard System

In the `STRUCTURE` field of the `EXTENSIONIN` parameter, the structure of the entity not delivered in the SAP standard system must be transferred, and in the related data containers `VALUEPART` the related data must be transferred. All of the data delivered in parameter `EXTENSIONIN` that is not of category `/PMO/BAPI_TE_*`, is written to instance table `MT_DBABDAADDDATA` of the migration class using migration structure `/PMO/ABQAADDDATA`. This means that this data too is available for processing through customer BAdI implementations.

 **CAUTION**

New entities and siblings must only be implemented by SAP development partners.

In detail, the migration of data from entities not belonging to SAP proceeds in the following steps:

- When you call the BAPI for the mass transfer of policies (`/PM0/BAPI_ABQ_POL_MASSMIGR`), the data from an entity that does not belong to SAP is transferred in the `EXTENSIONIN` parameter. If you want to insert this entity as a node in the *Policy* business object, the structure transferred in this parameter must be constructed in the same way as the standard system (`BO ID`, `PARENT ID`, and so on).

- Call the implementation of BAdI `/PM0/ABQ_POL_ENH_ENT_BADI` for the migration of data from entities not belonging to SAP.

You can copy the code of a `create_<entity>` method of migration class `/PM0/CL_ABQ_DAU_MIGRATION` for the implementation, and adapt it to meet your own requirements. If, for a new entity, you are dealing with a LoB enhancement (sibling) for a basis node, SAP recommends you to use the code for a sibling (such as `ALDACOV`) as a copy template here as well. As an example, you can view the code for an external reference (class `/PM0/CL_EX_ABQ_POL_ENH_ENT`), with the proviso that this is not delivered in the standard system and that it can be found under the coverage.

- Call the implementation of BAdI `/PM0/ABQ_POL_SAVE_BADI` for writing the data for additional entities to the database.

The call only occurs if no error messages have been issued in the previous activities for the migration class.

You also need to make sure to implement BAdI `/PM0/ABQ_POL_DEL_BADI` for the deletion of data from entities not belonging to SAP. This BAdI is required if you also need to include your own entities during the mass deletion of migrated policies and temporary data.

The following figure is a schematic representation of the process flow for the technical migration of data from entities not delivered in the SAP standard system. You can find more information about this in the description of the migration class in the following section.

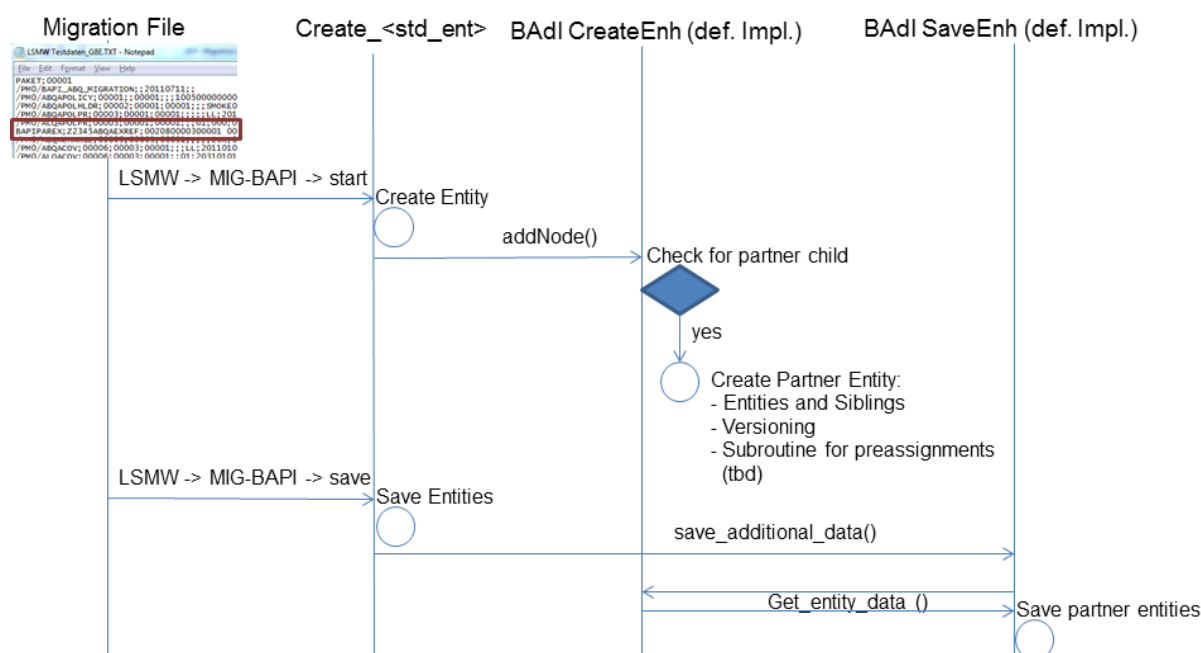


Figure: Process flow for technical migration of entities not delivered in the standard system.

Appendix D – Processing the Migration Data by Standard Migration Class `/PM0/CL_ABQ_DAU_MIGRATION`

In this section, we have summarized the basic processing steps of standard migration class `/PM0/CL_ABQ_DAU_MIGRATION` from the technical point of view.

Start method of the migration class:

- The migration data is sorted according to `VERSION_ID` and `BO_ID` in order to ensure the orderly processing of the policies and their versions. Therefore, the data does not have to be delivered already sorted.
- The policies, including all of the versions delivered for each policy, are processed one after the other.

- The following defaults and checks are made for the policy entity, where the steps not identified by 1) are relevant for all entities:
 - The attribute `JOURNALNO_ID` is filled with the attribute `VERSION_ID` of migration structure `/PM0/ABQAPOLICY` (temporary journal management).
 - If only one version is delivered, the attribute `JOURNALENDNO_ID` is filled with the constants for the highest journal number, or else with the `VERSION_ID` of the preceding version.
 - The company is derived using the `Templates_ID`. 1)
 - The policy number transferred is checked, or if not available, determined from the corresponding number range. 1)
 - The validity dates (`VALID_FROM_DT`) of the versions delivered are determined and checked. 1)
 - The GUIDS are created as necessary (for example, for the policy note, Secpol ID). 1)
 - The policy note is created. 1)
 - Private methods are called for the migration class for the creation of the policy children and BAdI `/PM0/ABQ_POL_ENH_ENT_BADI->add_note()` for partner entities.
 - The migration journal is derived or written: If `IF_PREASSIGN` has been set, `PM_DEFAULT` is set. If `IF_PREASSIGN` has been set and the `PM_ID` is transferred, the default values from In-Force Business Configurator are transferred. Table `ABDAZEROFIELDS` is also evaluated.
 - The policy journal is written with the following special features: 1)
 - `bizprc_id = 'Q'`
 - `updatetype_id = 01`
 - `bomigstatus_cd = 1. "migrated, not checked"`
 - `journalstat_cd = 110.`
 - `'DUMMY' lv_jno INTO ls_poll-application_id`
 - Provision of the structure for the mapping table for FS-PM journal entries and versions of FS-ICM (`/PM0/ABDTICMMPVR`). 1)
 - Call of method `CREATE_ABDAVARCOMM` to create the journal BO with the entity `VARCOMM` and to define the `BO_ID` for the journal BO in the journal entry. 1)
- You also create the remaining entities of the *Policy* business object according to the above pattern. The steps indicated by 1) are not necessarily relevant for all entities. The `CREATE` methods of the migration class always ensure the following:
 - The `JOURNALNO_ID` is always supplied with the `VERSION_ID` (temporary journal management).
 - The `JOURNALENDNO_ID` is filled with the `VERSION_ID` of the previous or with the journal high number in accordance with the number of versions delivered. If attribute `TODELETE_FG` of the migration structure is set, it is filled with the `VERSION_ID` of the version currently being processed.
 - `BO_CD` is filled with `P` (`P` indicates a version with policy issued).
 - The migration data is set as the time stamp `CHANGE_TS`.
- When the contract journal is created, attributes of both the policies and the contract migration structure are used.

`SAVE_ENTRIES` method of the migration class:

- The data is only written to the database if there are no error messages in the logger.
- The procedure in this migration class method is as follows:
 - Formatting of the cash flow/non cash flow documents (setting the migration status; completion of the journal number, structure of the header – item relationship).

Call of the PFO interface (In-Force Business Range) (set_com_migr; see also Section 'Commission System/Portfolio Assignment').

Writing the migration data to the FS-PM databases.

Writing the related dates required for the validation using the final policy GUID.

Call of BAdI /PM0/ABQ_POL_SAVE_BADI->save_additional_data() for saving customer/partner-specific data, which has been delivered in migration structure /PM0/ABQAADDDATA, in tables that do not belong to SAP.

Appendix E - Manual Migration Using Acquisition Type in New Business

If you want to migrate very small portfolios and individual policies, in the standard delivery of FS-PM you have the option of creating policies in the *New Business* business process using the acquisition type *Manual Policy Migration*. Apart from this special acquisition type, the procedure is the same as the normal *New Business* business process in FS-PM.

CAUTION

The acquisition type *Manual Policy Migration* is not further evaluated in the standard system, and consequently it has a purely documentary nature.

Appendix F – Error Handling Run

Using the transaction for starting the error handling run (transaction /PM0/ABQ_MIGCOR), you can automatically correct policies validated with errors to a certain extent.

CAUTION

This transaction was delivered in an earlier release. It is mentioned here briefly for the sake of completeness. However, as it is no longer being developed by SAP, you are advised against using it. SAP recommends that, instead, you delete incorrect policies, correct the implementation of the migration functions, and migrate the policies concerned again.

In the validation, you can log the error messages by implementing BAdI /PM0/ABQ_POL_ERR_HANDL_BADI, for example in table /PM0/ABDQPOLERR. In the error handling run, you can select the policies logged here, and further restrict the selection based on the messages.

You can closely control package creation in the error handling run by implementing BAdI

/PM0/ABQ_POL_BLD_PACK_CR_BADI.

You can correct errors in the error handling run by implementing BAdI

/PM0/ABQ_POL_ERRCORR_BADI.

Appendix G – Migration-Relevant BAdIs

You can find all of the BAdIs that you need to implement in a migration project in Customizing for Policy Management under *General Settings -> Migration -> Business Add-Ins (BAdIs)*.

The most important BAdIs are discussed in the main part of this guide. For additional information about these BAdIs, see the documentation for the relevant activities in Customizing for Policy Management.

You will not find the BAdIs for the error handling run in Customizing for Policy Management since SAP does not recommend using this function.

Appendix H – Policy Comparison for Support During Migration Regression Tests

If you wish to ensure that a migration process that was already accepted by the user department has not been rendered invalid again by corrections in the meantime, you can use policy comparison. A policy comparison program is available for this in FS-PM.

Program for comparing two policies

Using the *Compare Policies* (/PM0/ABX_POLICY_COMP) program, you can compare two policies including historical policy versions, and detect any differences. The policy version data that you want to include in your comparison can be adapted in Customizing to meet your particular requirements.

Note

The policy comparison program only works if the product engine msg.PM is used.

Recommendation

SAP recommends that you use the policy comparison in FS-PM migration tests as follows:

- Migrate and validate a selected, representative quantity of policies from the legacy system with the accepted migration procedure in FS-PM. Retain this quantity of legacy data in an unchanged form for later regression tests. The policies migrated to FS-PM in this step are later used as reference policies for the policy comparison.
- Check whether a migration test is absolutely necessary following corrections made in customer development or the implementation of SAP Notes. If it is necessary, migrate and validate the representative quantity of legacy policies again.
- A comparison of the policies you migrated again with the migrated reference policies should not result in any functional differences. If there are functional differences, you should check the correction again or else adapt the migration procedure as necessary and have it accepted again.

The data to be compared in the policy comparison depends closely on the migration scenario, the products used, on partner and customer developments, and other possible conditions.

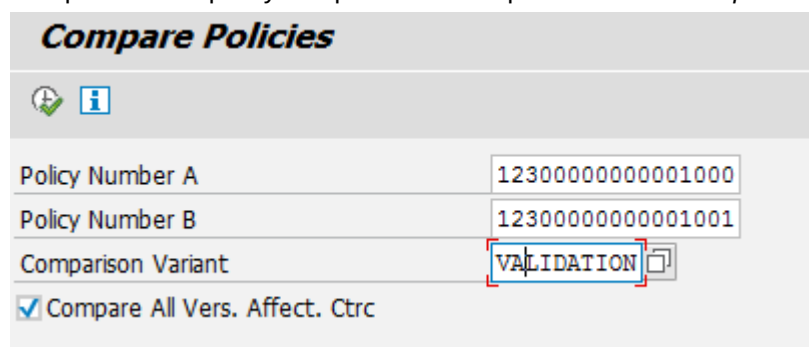
FS-PM therefore gives you the option of defining the data to be included in the comparison as freely as possible. You can find the *Compare Policies* program in Developer Customizing under transaction SE43 -> /PM0/CUST_INT -> *Developer Customizing* -> *Auxiliary Programs* -> *Policy Comparison*, where you can define comparison variants for the policy comparison.

The comparison variants are evaluated according to the exclusion principle. You define here the entities and fields that are NOT to be included in the policy comparison. If you do not define anything in the variant, a full comparison of the policies is made. With a full comparison, all of the persistent data of the *Policy* business object are included in the comparison. For some of the technical fields, such as technical keys, a difference is to be expected and is not regarded as an error. The policy comparison automatically does not include such fields.

It is often beneficial to exclude functional fields from the policy comparison. If the corrections you make result in an expected difference for certain fields, or if, for example, you expect differences between the reference migration and the migration in the regression test due to the time shift, you should exclude the relevant fields from the comparison.

SAP delivers comparison variants with sample values. These can provide you with information for the values of your variants.

You perform the policy comparison of two policies in the *Compare Policies* program as follows:



Compare Policies	
Policy Number A	12300000000001000
Policy Number B	12300000000001001
Comparison Variant	VALIDATION
☒ Compare All Vers. Affect. Ctrc	

Figure: Compare policies

You enter the policy number of the two policies to be compared, and select a comparison variant. If you set the *Compare All Versions That Affect Contract* indicator, then not only all the currently valid policy versions of the two policies are compared but all of the historical versions as well.

In the results view of the policy comparison, the differences in the number of entities are displayed separately from the differences at field level. If the system determines differing numbers of entities, there is no display at field level.

The following figure shows an example for the display of differences at field level.

	Feldname	Abw. Std.	Wert Pol.A	Wert Pol.B
Stand			Änderung 28.10.2010 14:00:29	Änderung 28.10.2010 14:25:33
Rechtsschutz,1100000000000000137	/PM0/ABDAPOLICY			
PS0301 Verkehrsrechtsschutz,1100000000000000074	/PM0/ABDAPOLPR			
Identifikation eines Versicherungsobjekts	INSOJECT_ID	🟢	IN-1100000000000000074	IN-1100000000000000076
Vorläufige Vertragsnummer	PROVAPPLNR_TT	🟢	1105000000000000146	1105000000000000148
PS0301 Verkehrsrechtsschutz,1100000000000000075	/PM0/ABDAPOLPR			
Verlängerungsvariante	EXTVARIA_CD	🔴	001	000
Identifikation eines Versicherungsobjekts	INSOJECT_ID	🔴	IN-1100000000000000075	IN-1100000000000000077
Vorläufige Vertragsnummer	PROVAPPLNR_TT	🔴	1105000000000000147	1105000000000000149

Figure: Display of differences at field level

<i>German:</i>	<i>English:</i>
<i>Abweichende Felder</i>	<i>Different Fields</i>
<i>Polizistenstand</i>	<i>Policy Version</i>
<i>Stand</i>	<i>Version</i>
<i>Feldname</i>	<i>Field Name</i>
<i>Abw. Std.</i>	<i>Difference from Policy Version</i>
<i>Wert: Pol. A</i>	<i>Field Value in Policy A</i>
<i>Wert: Pol. B</i>	<i>Field Value in Policy B</i>

Differences existing in an earlier version are clearly indicated by the red traffic light. In the above example, the difference is in the extension variant, meaning by the fourth version. If you need to find the cause of the difference, then in this example you would have to examine the fourth policy version (or an earlier one). Differences that first occur in the policy version currently being compared, in this example version 3, are marked green.

The *Compare Policies* program uses the actual comparison tool. You can obtain details of this from program /PM0/ABX_POLICY_COMPARE_TOOL. You can find the preparation of the results of the comparisons in the implementation of Include /PM0/ABX_POLICY_COMPARE_OUT, for example.

Note

When you migrate and validate a policy created in FS-PM with the test tool for the migration of policies, you can update the migrated policy in the validation up to a target date that differs from the target date of the original policy. In such cases, the account balances between the two policies for instance can differ considerably. You can only make a useful comparison of such policies if the original policy and the resulting migration policy are updated to the same date, which means the implementation of target date BADI /PM0/ABB_PROC_TARGETDATE determines the same value in both cases.

CAUTION

The policy comparison is restricted to fields of the *Policy* business object. You should also perform a comparison of the data connected with the policy following critical changes. You should pay particular attention to the data of the accounting component and of dates. You are also advised to perform central scenarios such as changes or resetting in the test system following a successful comparison using either an automated or manual test, and to verify the version after performing these scenarios for each policy and manual comparison.

Comparison of several policies pairs

In the migration tests, you will frequently want to compare more than just two policies. As the actual comparison of two policies is encapsulated in the policy comparison tool, you can write a program in your project with minimum effort that enables the comparison of several policy pairs in one program run.

Appendix I – Example of Handling of Entities to Be Deleted in the Release

In Section 'Handling of Entities to Be Deleted in the Release', there was a description of how you proceed if, by positioning a processable migrated policy version, an entity exists in the subsequent version that is correctly no longer there in the technically migrated subsequent version. This procedure is illustrated here using an example:

- Version A with effective date 1.1.2009, processing ID *Processing A* and journal number 1 has the entity *Subcoverage* with key *D1*.
- Version B with effective date 1.1.2010, processing ID *Processing B* does not have the entity *Subcoverage* with key *D1*.
- There is no version between version A and version B.
- Version A is functionally validated up to the *Release* step (complete migration, step 10 in the list in Section 7.4.2).

Entity *Subcoverage*:

Key	Effective Date	Journal Number	Journal End Number
D1	01.01.2009	1	VALID

Application Journal:

Number	Effective Date	Processing ID
1	1.1.2009	Processing A
...	1.1.2009	Processing...
19	1.12.2009	Processing X

- Step 1:
The system checks – with the appropriate Customizing – whether subcoverage *D1* also exists for version B. This is not the case. Subcoverage *D1* is therefore deleted:

Entity *Subcoverage*:

Key	Effective Date	Journal Number	Journal End Number
D1	01.01.2009	1	20
D1	01.01.2009	20	20

- Step 2:
The deletion is recorded in the journal with processing ID and effective date of version B.

Application Journal:

Number	Effective Date	Processing ID
1	1.1.2009	Processing A

...	1.1.2009	Processing...
19	1.12.2009	Processing X
20	1.1.2010	Processing B

- Step 3:
The application versions are transferred to contract versions:

Contract Journal

Number	Effective Date	Processing ID
1	1.1.2009	Processing A
...	1.1.2009	Processing...
19	1.12.2009	Processing X
20	1.1.2010	Processing B

- Step 4:
The journal entry to be deleted from step 2 is deleted from the contract journal

Contract Journal

Number	Effective Date	Processing ID
1	1.1.2009	Processing A
...	1.1.2009	Processing...
19	1.12.2009	Processing X

- Final status:
Following the functional validation of version B, the following version, for example, would result:

Entity *Subcoverage*:

Key	Effective Date	Journal Number	Journal End Number
D1	01.01.2009	1	20
D1	01.01.2009	20	20

Application Journal:

Number	Effective Date	Processing ID
1	1.1.2009	Processing A
...	1.1.2009	Processing...
19	1.12.2009	Processing X
20	1.1.2010	Processing B
21	1.1.2010	Processing B
...	1.1.2010	Processing

Contract Journal		
Number	Effective Date	Processing ID
1	1.1.2009	Processing A
...	1.1.2009	Processing...
19	1.12.2009	Processing X
21	1.1.2010	Processing B
...	1.1.2010	Processing

